Fast iterative solvers for time-dependent PDE-constrained optimization problems

Habilitation

zur Erlangung des akademischen Grades

doctor rerum naturalium habilitatus
(Dr. rer. nat. habil.)

von Martin Stoll
(akademischer Grad, Vorname, Name)

geb. am 16.02.1980 in Bützow

genehmigt durch die Fakultät für Mathematik
der Otto-von-Guericke-Universität Magdeburg

Gutachter: Prof. Dr. Peter Benner
(akademischer Grad, Vorname, Name)

Prof. Dr. Roland Herzog
(akademischer Grad, Vorname, Name)

Prof. Dr. Jörg Liesen
(akademischer Grad, Vorname, Name)

eingereicht am: _____

Verteidigung am: 30.06.2016

Fast iterative solvers for time-dependent PDE-constrained optimization problems

MAX PLANCK INSTITUTE
FOR DYNAMICS OF COMPLEX
TECHNICAL SYSTEMS
MAGDEBURG

Martin Stoll

Max Planck Institut Magdeburg

Otto-von-Guericke Universität

Kumulative Habilitation

2016

To my beautiful family

CONTENTS

ACKNOWLEDGMENTS

This thesis presents results from several years of research and hence the list of people I owe gratitude to is not short. First I would like to thank Andy Wathen my DPhil and first postdoctoral supervisor, who has introduced me to the problem of PDE-constrained optimization and challenged me to think about solving parabolic problems. Leaving Oxford my new mentor Peter Benner allowed me the freedom to follow my ideas and always encouraged me to go a step further.

Mostly, I am indebted to my co-authors who have made working on these problems a fun ride and teaching me many things on the way. Besides Andy, I would like to especially thank my two 'little brothers' Tyrone Rees and John Pearson. I stirred both Andrew Barker's and Tobias Breiten's interest in PDE-constrained optimization problems and the work with them has been great fun.

I would also like to thank my group as they have always inspired me to live up to their spirit. Thank you Jessica, Akwum, Wei, Hamdullah, and Sergey! I also thank the whole CSC group at the MPI Magdeburg for their input. In particular my office mate Sara Grundel.

There are also researches who have provided constant input on how to solve PDE-constrained optimization problems. In particular, I am indebted to Walter Zulehner, Roland Herzog, Ekkehard Sachs, Eldad Haber, Michele Benzi among others. I thank Jen Pestana for her input on nonstandard inner products.

Last but not least, I need to thank the people who showed me that there is more to life than preconditioning. I am always indebted to my mom for supporting me even though my research could not be more foreign to her. Most importantly, my gratitude goes to Anke, my wife, and our three kids. I love you!

SUMMARY

The optimal control of partial differential equations is a crucial topic in computational science and engineering. Many techniques from functional analysis are combined with state-of-the-art numerical schemes and efficient software technology to efficiently compute solutions to optimal control problems.

One of the most challenging tasks is the solution of linear systems in structured form. These systems often represent the computational bottleneck of an optimization problem. Linear systems can be found representing the first order optimality conditions, at the heart of an outer nonlinear solver, or even in methods for non-smooth problems. One of the key features that these problems have in common is the large-scale nature of the systems, as the computation of accurate solutions requires small mesh-parameters. This problem becomes even more pronounced when the constraining PDE is of parabolic type as now the curse of dimensionality strikes and the computational work drastically increases.

The approach that we promote in this thesis is one of discretizing the problem in both space and time and then simultaneously solving the corresponding optimization problem. Assuming a linear PDE and a convex objective, this means we need to solve a large-scale KKT system representing the first order conditions. Such a space-time system can only be solved using iterative solvers. For this we propose methods of Krylov type and discuss tailored preconditioners. These are needed because the convergence of the unpreconditioned scheme can be very slow as it will depend on the system parameters, such as mesh and regularization parameter.

In our proposed methodology we need an efficient approximation of the leading block and the Schur-complement of the system representing the first order conditions. Typically, the approximation of the leading block is straightforward due to the nature of its construction. The Schur-complement typically involves the sum of potentially complicated operators such as the

discretized PDE operators. We illustrate an approach where the Schur-complement is approximated by a simpler operator that is the product of three matrices that match all terms of the original Schur-complement. It is often possible to theoretically underpin this approach by proving the robustness of the eigenvalues of the preconditioned Schur-complement with respect to variations of the system parameters.

While this approach in many cases provides optimal or nearly optimal results it can suffer from the storage requirements of the space-time vectors not that space-time matrix. For this we illustrate an elegant solution that utilizes low-rank techniques to avoid the curse of dimensionality when dealing with parabolic control systems.

ZUSAMMENFASSUNG

Die optimale Steuerung von partiellen Differentialgleichungen ist von enormer Bedeutung in allen Bereichen der computergestützen Wissenschaften oder den Ingenieurwissenschaften. Viele Techniken aus der Funktionalanalysis werden mit modernsten numerischen Verfahren und neuester Software kombiniert, um die Lösungen der Steuerungsprobleme genau und effizient zu berechnen.

Eine der größten Herausforderungen ist das effiziente Lösen von linearen Gleichungssystemen in strukturierter Form, welche das Herzstück vieler Optimierungsalgorithmen bilden. Solche Systeme repräsentieren dabei entweder die Optimalitätsbedingungen erster Ordnung oder sind der Kern von äußeren, nichtlinearen Lösern, wie dem populären SQP-Verfahren. In all diesen Fällen ist die Systemmatrix sehr groß, da akkurate Lösungen zumeist kleine Netzparameter erfordern. Diese Eigenschaft ist noch stärker ausgeprägt, wenn zeitabhängige, parabolische Probleme betrachtet werden und der Curse-of-dimensionality zuschlägt und daraus resultierend der Arbeitsaufwand drastisch steigt.

Die in dieser Arbeit präsentierte Technik löst dabei das Optimierungsproblem simultan in Raum und Zeit. Bei linearer PDE-Nebenbedingung und einer konvexen Zielfunktion bedeutet dies, dass ein großes KKT- oder Sattelpunktsystem gelöst werden muss. Systeme dieser Größe können nur von iterativen Verfahren effizient gelöst werden. Zu diesem Zweck wird das Verwenden von Krylov-Unterraumverfahren vorgeschlagen und die Entwicklung von speziellen Vorkonditionierern diskutiert.

Diese sind extrem wichtig, um die Konvergenzgeschwindigkeit zu beschleunigen und robuste Konvergenz unabhängig von den Systemparametern, wie den Netz- oder Regularisierungsparametern, zu garantieren.

Dazu wird eine effiziente Approximation des $(1,1)$-Blocks und des Schurkomplements verwendet. Der $(1,1)$-Block ist dabei einfach zu approximieren, was in der Natur der Formulierung liegt, aber nicht prinzipiell für alle Prob-

leme gilt. Das Schurkomplement besteht typischerweise aus einer Summe von komplizierten, diskretisierten Differentialoperatoren. Dabei ist ein Ansatz erfolgreich der möglichst alle Terme des ursprünglichen Schurkomplements widerspiegelt, aber dabei einfach zu invertieren ist. Hierbei kann in einigen Fällen die Robustheit dieser Approximation bewiesen werden und es zeigt sich dabei, dass die Eigenwerte der vorkonditionierten Matrix robust bezüglich Parameterverwänderungen sind.

Dieser Ansatz zeigt häufig optimales oder nahezu optimales Konvergenzverhalten. Ein möglicher Nachteil ist der Speicherbedarf für die Raum-Zeit-Vektoren. Die Raum-Zeit-Matrizen erfordern dabei nahezu den gleichen Speicherbedarf wie im stationären Fall. Diese Arbeit präsentiert daher eine elegante Lösung, welcher Niedrigrangansätze aufzeigt, die es erlauben den Curse-of-dimensionality zu durchbrechen.

CHAPTER 1

INTRODUCTION

The numerical solution of partial differential equations (PDEs) has been a core question of numerical analysis from the beginning. Much progress has been made on understanding the equations, existence of solutions, constructing tailored discretizations, developing fast solvers and much more. This has enabled researchers from all subjects to ask more fundamental questions. One that is typically asked is to compute the 'optimal setup' of complex PDE models that describe measured or desired data. The field of optimal control with PDEs or PDE-constrained optimization has therefore received much attention over the last decades with fantastic progress on all fronts. The contribution of this thesis is mainly concerned with the progress that has been made for the solution of the discretized linear systems that arise as the solution of first order optimality conditions or are at the core of a nonlinear solution technique. The next sections follow the path of such an optimal control problem from formulation to discretization and further to solution via iterative solvers. Beyond that, modern compressed formats are lastly presented and we speculate on what problems will be considered in the future. This thesis is not intended to give a general introduction to the subject but rather give a small overview of issues related to the fast iterative solution of linear systems in saddle point form that arise in PDE-constrained optimization. There exist excellent introductions to PDE-constrained optimization see [39,41,87] and in particular the theses [60,67] for linear algebra focused introductions.

The field of PDE-constrained optimization is a research area with problems coming from all areas of science and engineering. Our strategy in this thesis is to illustrate our developed methodology on a model problem that is a representative of a wider class of challenges. We will in places refer to more general introductions or different applications.

The goal of the following overview is to establish the contribution made

in the attached papers (cf. Appendix A.1 to A.9), which focus on the fast and robust solution of the linear systems that arise in the discretization of the infinite-dimensional optimization problems. In particular, this thesis focuses on time-dependent PDEs treated in a simultaneous or all-at-once approach for which little work existed until recently.

1.1 A PDE-constrained optimization model problem

The core problem that will be discussed in this work and the associated papers shown in the appendix is an optimization problem where the objective function is given by

$$J(y,u) := \frac{1}{2}\int_{\Omega_1} (\mathrm{y(x)} - \bar{\mathrm{y}}\mathrm{(x)})^2\, dx + \frac{\beta}{2}\int_{\Omega_2} \mathrm{u(x)}^2 dx \tag{1.1}$$

in stationary form or in the transient case defined by

$$J(y,u) := \frac{1}{2}\iint_{Q} (y(x,t) - \bar{y}(x,t))^2\, dxdt + \frac{\beta}{2}\iint_{\Sigma} u(x,t)^2 dxdt, \tag{1.2}$$

where $Q = \Omega_1 \times (0,T), \Sigma = \Omega_2 \times (0,T)$ are space-time cylinders. Here T is the final time and we have a given desired state $\bar{y}$ which is specified for each problem. The goal of the optimization process is to drive the *state* y as close as possible to the desired state using the *control* u. Note, that both Ω_1 and Ω_2 are subdomains of a Lipschitz domain $\Omega \in \mathbb{R}^d$. We will frequently use $\Omega_2 = \Omega$ and occasionally $\Omega_1 \subsetneq \Omega$. So far we have only introduced the objective function but the state y and the control u need to be linked in a meaningful way. For this it is crucial that the underlying 'problem physics' are well represented. Often naturally, we use as a model problem a partial differential equation called the *state equation* that links both quantities. We consider the following parabolic PDE for the course of this work

$$y_t - \triangle y = u \tag{1.3}$$

in $\Omega \times (0,T)$, with boundary conditions $y = 0$ on the spatial boundary $\partial\Omega$ and initial condition $y(x,0) = y_0(x)$. We here simply use the heat equation because it illustrates many of the desired features that require our special attention later on. In practice, more complex models are needed and in many instances additional constraints often of algebraic nature are imposed. We later briefly come back to such cases.

Assuming that Ω is a Lipschitz domain and $y, \bar{y} \in L^2(Q)$, Theorem 3.16 in [86] gives the existence of an optimal control u^*, which is unique for $\beta > 0$. We also later need the adjoint PDE to the heat equation constraint defined

above. For (1.2), the adjoint equation is given by

$$\begin{aligned} -p_t - \triangle p &= \chi_{\Omega_1}(y - \bar{y}) \\ p &= 0 \text{ on } \partial\Omega \\ p(x,T) &= 0, \end{aligned} \tag{1.4}$$

where χ_{Ω_1} is an indicator function for the domain Ω_1 (see Chapter 3.6.4 in [86] for more details). The adjoint equation with adjoint state p is a crucial quantity in PDE constrained optimization and p acts as the Lagrange multiplier used in a Lagrangian (see [41, 87] for details).

In order to determine the optimal solution to the optimization problem described above one would typically follow a Lagrangian approach. The stationary points of the corresponding Lagrangian function reflect the first order necessary optimality conditions [41] of our problem. There are two ways to arrive at such an approximate solution. The first one discretizes the objective function and PDE-constraint, then considers a discrete Lagrangian and builds the first order conditions of the discretized problem. This is the so-called *discretize-then-optimize* approach.

The second approach first builds a Lagrangian function based on the infinite dimensional problem and then discretizes the resulting optimality conditions. This is the so-called *optimize-then-discretize* approach. In this approach [87] the optimality conditions are given by the state equation (1.3), the adjoint PDE (1.4), and the gradient equation

$$\beta u + p = 0, \tag{1.5}$$

all formulated in function space. Both approaches have favorable properties and much research has be done to find suitable discretization such that both approaches commute [39]. Additionally, it can be advantageous not to discretize the control variable at all but remove it using the gradient equation (1.5). This is often referred to as the variational discretization concept [36]. As the focus of our work is on the efficient design of fast solvers for the discretized problem the elimination of the control is often not a crucial step in this setup and we will not employ the variational discretization.

In this work we follow a *discretize-then-optimize* strategy for the simple reason that the resulting linear systems representing the first order conditions are symmetric, which simplifies the choice of the iterative solver. Nevertheless, the techniques we present are mostly applicable in the non-symmetric case that could arise when an *optimize-then-discretize* framework is employed.

We now discuss this approach in more detail. We begin by discretizing the functional given in (1.2) using a standard finite-element approach in space and a trapezoidal rule for the temporal discretization. The discretization in time of the PDE uses an implicit Euler scheme and finite elements

in space. Note that of course other temporal discretization schemes are possible and our methodology typically applies in these cases. We here refer to [2,52] for more details on space-time discretizations and believe that this area will receive more attention in the coming years.

In more detail, the discretization of (1.2) leads to

$$J(\mathbf{y},\mathbf{u}) = \frac{\tau}{2}(\mathbf{y}-\bar{\mathbf{y}})^T\mathcal{M}_{1/2}(\mathbf{y}-\bar{\mathbf{y}}) + \frac{\beta\tau}{2}\mathbf{u}^T\mathcal{M}_{1/2}\mathbf{u}, \tag{1.6}$$

where

$$\mathcal{M}_{1/2} = \begin{bmatrix} \frac{1}{2}M & & & & \\ & M & & & \\ & & \ddots & & \\ & & & M & \\ & & & & \frac{1}{2}M \end{bmatrix}. \tag{1.7}$$

Note that we use bold notation for the vectors representing the state, control, adjoint state and so on. Using the rectangular rule instead would give

$$J(\mathbf{y},\mathbf{u}) = \frac{\tau}{2}(\mathbf{y}-\bar{\mathbf{y}})^T\mathcal{M}_{0}(\mathbf{y}-\bar{\mathbf{y}}) + \frac{\beta\tau}{2}\mathbf{u}^T\mathcal{M}_{0}\mathbf{u}, \tag{1.8}$$

with $\mathcal{M}_0 = \text{blkdiag}(M, M, \ldots, M, 0)$. Here, $\mathbf{y} = [\mathbf{y}_1^T, \mathbf{y}_2^T, \ldots, \mathbf{y}_N^T]^T$ is a space-time vector representing the discrete state at time-steps 1 to N of a backward Euler scheme (see below). Next, we perform a time discretization of the PDE (1.3) using a backward Euler scheme

$$\frac{y^k - y^{k-1}}{\tau} - \triangle y^k = u^k \tag{1.9}$$

with time step τ. A spatial discretization using finite elements leads to

$$M\mathbf{y}_k + \tau K\mathbf{y}_k = M\mathbf{y}_{k-1} + \tau M\mathbf{u}_k. \tag{1.10}$$

Here, K is the finite element stiffness matrix and M the finite element mass matrix. Putting all of Equation (1.10) together, the one-shot or space-time discretization for N time-steps becomes

$$\underbrace{\begin{bmatrix} M+\tau K & & & & \\ -M & M+\tau K & & & \\ & -M & M+\tau K & & \\ & & \ddots & \ddots & \\ & & & -M & M+\tau K \end{bmatrix}}_{\mathcal{K}} \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \mathbf{y}_3 \\ \vdots \\ \mathbf{y}_N \end{bmatrix} - \tau\mathcal{M}\mathbf{u} = d, \tag{1.11}$$

where

$$d = \begin{bmatrix} M\mathbf{y}_0 + c \\ c \\ \vdots \\ c \end{bmatrix}, \quad \mathbf{u} = [\mathbf{u}_1^T, \mathbf{u}_2^T, \ldots, \mathbf{u}_N^T]^T, \text{ and } \mathcal{M} = \text{blkdiag}(M, \ldots, M).$$

The vector c represents the boundary conditions of the state equation. Using this we can write down the discrete Lagrangian for the functional $J(y, u)$ as

$$\mathcal{L}(\mathbf{y}, \mathbf{u}, \mathbf{p}) = J(\mathbf{y}, \mathbf{u}) + \mathbf{p}^T(-\mathcal{K}\mathbf{y} + \tau\mathcal{M}\mathbf{u} + d), \tag{1.12}$$

with the Lagrange multiplier given as $\mathbf{p} = [\mathbf{p}_1^T, \mathbf{p}_2^T, \ldots, \mathbf{p}_N^T].^T$ In this framework, the Lagrange multiplier also represents a discrete version of the adjoint state satisfying the adjoint equation. Finally, the first order conditions can now be written as

$$\begin{bmatrix} \tau\mathcal{M}_{1/2} & 0 & -\mathcal{K}^T \\ 0 & \beta\tau\mathcal{M}_{1/2} & \tau\mathcal{M} \\ -\mathcal{K} & \tau\mathcal{M} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \tau\mathcal{M}_{1/2}\bar{\mathbf{y}} \\ 0 \\ d \end{bmatrix}. \tag{1.13}$$

The first equation in (1.13) is referred to as the adjoint equation, for the simple reason as it represents a discretization of the adjoint PDE. Namely, we get

$$\tau\mathcal{M}_{1/2}\mathbf{y} - \mathcal{K}^T\mathbf{p} = \tau\mathcal{M}_{1/2}\bar{\mathbf{y}}.$$

Note that an Euler discretization would be given by

$$-\frac{p^k - p^{k-1}}{\tau} - \triangle p^{k-1} = y^{k-1} - \bar{y}^{k-1}. \tag{1.14}$$

In order to return to the discussion about whether to optimize first and then discretize or vice versa we look at the final condition for the adjoint equation. The final condition for the adjoint PDE has to be represented by the first equation in (1.13). Note that the last block-line of the first-equation in (1.13) gives

$$\frac{1}{2}M(\mathbf{y}^N - \bar{\mathbf{y}}^N) = (M + \tau K)p^N,$$

which does not necessarily coincide with final conditions of the adjoint PDE

$$-\frac{p^N - p^{N-1}}{\tau} - \triangle p^{N-1} = y^{N-1} - \bar{y}^{N-1}.$$

Note that for the steady version of the heat equation, discretize-then-optimize and optimize-then-discretize typically coincide while for the transient problem this is not necessarily the case. For $\tau \to 0$ the final condition is fulfilled; indeed the final condition of the adjoint equation is satisfied to first order

accuracy in τ.

We have now in much detail described the discretization of the optimization problem constrained by the heat equation. Before we discuss how to solve the system (1.13), we comment on possible extensions and more general cases. Obviously, in many applications one needs to consider more complex objective functions, more difficult PDEs, systems of PDEs, or additional constraints.

Among the vast amount of research on this topic, we want to point to the control of the Stokes equations [37,46,70,83,84] (see also Appendix A.5) and convection diffusion equations [1,15,64,67]. In these examples the PDE-constraint is a linear PDE, which when ignoring additional constraints on the state or the control means solving the KKT conditions is sufficient to determine the optimal control and state.

In contrast, many examples found in the sciences and engineering are of nonlinear type and it becomes even more important to be able to solve such problems efficiently. Many of the techniques we later present can be used for nonlinear problems. Here we do not discuss nonlinear problems and the numerical techniques needed in any detail. For introductions to nonlinear PDE-constrained optimization we refer to [39, 41, 87]. A typical way to solve such problems consists of forming the first order conditions for the Lagrangian function, which itself is now a nonlinear system of equations. Such a system can then be solved using Newton techniques, this is the so-called Lagrange-Newton or SQP scheme see [11, 12, 41, 42, 45, 56, 66, 93] for more details and Appendix A.7 and A.8 for reaction-diffusion systems employing Lagrange-Newton techniques. Another very efficient scheme is the so-called interior point method [75, 89, 91]. Introductions to these types of methods are found in [23, 43, 57] for general optimization problems and in [11, 31, 39, 41, 88] for the particular case of nonlinear PDE-constraints. Such a nonlinear solver again requires the solution of a linear system similar to the one shown in (1.13). One technique often used in combination with SQP schemes is the so-called trust region approach that has recently received some attention [30, 93].

Additionally, one can obtain nonlinear problems when the model also requires the control to be bounded via

$$u_a \leq u \leq u_b$$

where we assume that $u_a, u_b \in L^2(\Sigma)$. This additional constraint presents a new challenge as the optimality conditions for such a system now involve variational inequalities. For handling problems of this type, non-smooth Newton methods have shown great potential [9,35,41] and have been studied extensively. Non-smooth Newton schemes are also well-suited for the more

challenging case when the state itself is constrained

$$y_a \leq y \leq y_b$$

[10, 16, 41]. We refrain from discussing the last two cases here and refer to the excellent work found in the literature, see [39, 41, 86, 88] as points of reference. We refer to [33, 74] for functional analysis motivated approaches to these problems. Our results for the development of iterative solvers are found in Appendix A.1 and A.2 representing our work in [81, 82].

We now discuss methods that allow the efficient solution of these large-scale saddle point systems such as the one given (1.13). Our discussion will focus on iterative solvers of Krylov type. There are also other efficient schemes such as multigrid methods that we do not discuss but refer to [13, 14, 38].

1.2 Krylov subspace solvers

The linear system presented in (1.13) is a system in saddle point form with system matrix

$$\mathcal{A} = \begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix},$$

where

$$A = \begin{bmatrix} \mathcal{M}_{1/2} & 0 \\ 0 & \beta\tau\mathcal{M}_{1/2} \end{bmatrix} \quad B = \begin{bmatrix} -\mathcal{K} & \tau\mathcal{M} \end{bmatrix}.$$

A system of this form is typically referred to as a KKT system as it represents the first order of an optimization problem [56]. Systems of this type have been studied extensively within the numerical analysis community, see [7, 20] for superb introductions to the numerical solution of saddle point problems. While often, especially for two-dimensional systems, direct solvers provide outstanding performance, these method's performance typically deteriorates for higher dimensional or highly structured systems. Hence, there is a crucial need to construct efficient iterative methods.

The most effective methods to iteratively solve large and sparse linear systems are so-called preconditioned Krylov subspace solvers. We briefly discuss the basics needed in our case but refer to [22, 24, 27, 53, 72] for more thorough introductions. There exist many different Krylov subspace solvers and the choice which one to employ typically depends on the properties of the system matrix. As our systems are symmetric and indefinite we focus on the MINRES method [59], which we explain in the following. In the case of a symmetric positive definite system the conjugate gradient method [34] is the method of choice. For nonsymmetric problems it is much less clear which method to choose [55] and we here refer to GMRES [73] and BICGSTAB [90] for two of the many available nonsymmetric Krylov methods.

For MINRES and all the just mentioned methods, the underlying Krylov-subspace of dimension k

$$\mathcal{K}_k(\mathcal{A}, r_0) = \text{span}\left\{r_0, \mathcal{A}r_0, \ldots \mathcal{A}^{k-1}r_0\right\}$$

is of utmost importance. Here, $r_0 = b - \mathcal{A}x_k$ is the initial residual but r_0 can be an arbitrary starting vector. With every step of an iterative procedure the chosen method uses an increased dimension of $\mathcal{K}_k(\mathcal{A}, r_0)$ to find an approximation to the solution x of the linear system. The method uses an optimality criterion for the selection of approximation to the solution x and the quantity that is minimized by a particular method depends mainly on the properties of the systems matrix $\mathcal{A}$.

The minimal residual method (MINRES) [59] is an iterative solver for symmetric systems where, as the name suggests, the minimization of the residual is its defining characteristic. We start our discussion of it from a purely algebraic viewpoint. We will later discuss it using the knowledge that the matrix $\mathcal{A}$ comes from the discretization of a partial differential equation. The discussion is then based on the fact that we have a good understanding of the mapping properties of the continuous operator underlying the matrix $\mathcal{A}$.

The MINRES method is based on the symmetric Lanczos procedure, which constructs an orthogonal basis for the Krylov-subspace $\mathcal{K}_k(\mathcal{A}, r_0)$. The Lanczos method is then expressed as

$$\mathcal{A}V_k = V_k T_k + \beta_{k+1} v_{k+1} e_k^T = V_{k+1} T_{k+1,k}$$

with

$$T_{k+1,k} = \begin{bmatrix} \alpha_1 & \beta_2 & & \\ \beta_2 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_k \\ & & \beta_k & \alpha_k \\ & & & \beta_{k+1} \end{bmatrix}.$$

This scheme can be derived from a tridiagonalization of the matrix $\mathcal{A}$, i.e., $\mathcal{A} = VTV^T$, and its column-wise consideration. The approximate solution x_k is of the form

$$x_k = x_0 + V_k z_k \tag{1.15}$$

for some vector z_k, where the columns of V_k form an orthogonal basis of the Krylov subspace $\mathcal{K}_k(\mathcal{A}, r_0)$. We refer to the condition (1.15) as the space condition because the current approximation x_k is a linear combination of the starting vector x_0 and the current basis of the Krylov space $\mathcal{K}_k(\mathcal{A}, r_0)$. The vector z_k of coefficients is computed such that the 2-norm of the current residual $r_k = b - \mathcal{A}x_k$ is minimized. Mathematically, this is expressed as

(ignoring the minimization)

$$\begin{aligned} \|r_k\|_2 &= \|b - \mathcal{A}x_k\|_2 \\ &= \|b - \mathcal{A}(x_0 + V_k z_k)\|_2 \\ &= \|r_0 - \mathcal{A}V_k z_k\|_2 \\ &= \|r_0 - V_{k+1}T_{k+1,k}z_k\|_2 \end{aligned} \tag{1.16}$$

and with the typical choice of $v_1 = r_0/\|r_0\|_2$ we get

$$\begin{aligned} \|r_k\|_2 &= \|V_{k+1}(\|r_0\|_2 e_1 - T_{k+1,k}z_k)\|_2 \\ &= \|\,\|r_0\|_2 e_1 - T_{k+1,k}z_k\|_2. \end{aligned} \tag{1.17}$$

The term V_{k+1} inside the norm can be ignored because its columns are orthogonal in exact arithmetic. In order to compute the vector z_k, we have to minimize (1.17), i.e., a least squares problem

$$\min \|r_k\|_2 = \min \|\,\|r_0\|_2 e_1 - T_{k+1,k}z_k\|_2.$$

A well-known technique to solve least squares systems of this type is the QR decomposition (cf. [59]). As computing the QR decomposition at every step could be very costly, we need an alternative. Fortunately, since the matrix $T_{k+1,k}$ changes from step to step simply by adding one column and one row, its QR decomposition can be updated at every step. This can be done by simply using one Givens rotation [24,85]. In more detail, we assume that the QR factorization of $T_{k,k-1} = Q_{k-1}R_{k-1}$ is given with

$$R_{k-1} = \begin{bmatrix} \hat{R}_{k-1} \\ 0 \end{bmatrix}$$

and $\hat{R}_{k-1}$ an upper triangular matrix. To obtain the QR factorization of $T_{k+1,k}$ we eliminate the element β_{k+1} from

$$\begin{aligned} \begin{bmatrix} Q_{k-1}^T & 0 \\ 0 & 1 \end{bmatrix} T_{k+1,k} &= \begin{bmatrix} Q_{k-1}^T & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} T_{k,k-1} & \alpha_k e_k \\ 0 & \beta_{k+1} \end{bmatrix} \\ &= \begin{bmatrix} R_{k-1} & Q_{k-1}^T \alpha_k e_k \\ 0 & \beta_{k+1} \end{bmatrix} \end{aligned} \tag{1.18}$$

by using one Givens rotation in rows $k, k+1$. There is no need to store the whole basis V_k in order to update the solution. The matrix R_k of the QR decomposition of the tridiagonal matrix $T_{k+1,k}$ has only three non-zero diagonals. Let us define $C_k = [c_0, c_1, \ldots, c_{k-1}] = V_k\hat{R}_k^{-1}$. Note that c_0 is a multiple of v_1 and we can compute successive columns using that $C_k\hat{R}_k = V_k$, i.e.

$$c_{k-1} = (v_k - \hat{r}_{k-1,k}c_{k-2} - \hat{r}_{k-2,k}c_{k-3})/\hat{r}_{k,k}, \tag{1.19}$$

where the $\hat{r}_{i,j}$ are elements of $\hat{R}_k$. Therefore, we can update the solution

$$x_k = x_0 + \|r_0\|_2 C_k \left(Q_k^T e_1\right)_{k\times 1} = x_{k-1} + a_{k-1}c_{k-1}, \tag{1.20}$$

where a_{k-1} is the kth entry of $\|r_0\|_2 Q_k^T e_1$. Implementations of MINRES are found in most libraries for scientific computing and even many finite element packages come with a MINRES implementation under the hood. We refer to [20, 22] for more details on the implementation.

We have introduced MINRES as a method to solve the system (1.13) but need to discuss its convergence as this is typically dependent on the number of distinct eigenvalues of the system matrix $\mathcal{A}$ [27, 49]. Loosely speaking, the fewer the number of distinct eigenvalues or number of eigenvalues the faster the convergence. For the PDE-constrained optimization problems, the eigenvalues of the system matrix $\mathcal{A}$ depend on the system parameters such as the mesh-parameter and the regularization parameter. This means for many interesting values of these parameters, the eigenvalues are spread out over the real line with no hope for fast convergence. The goal is now to overcome this shortcoming and for this reason we are modifying the problem using a so-called preconditioner such that we can still apply MINRES but with enhanced convergence properties. In more detail, the preconditioning matrix $\mathcal{P}$ is used in the following way

$$\mathcal{P}^{-1}\mathcal{A}x = \mathcal{P}^{-1}b \tag{1.21}$$

with the ultimate goal that $\mathcal{P}^{-1}\mathcal{A}$ has very few distinct eigenvalues or its eigenvalues are found in a small number of distinct clusters. While $\mathcal{P}^{-1}\mathcal{A}$ is not symmetric anymore, a symmetric positive definite preconditioner $\mathcal{P}$ still allows the use of MINRES using the spectral equivalence of the matrix $\mathcal{P}^{-1}\mathcal{A}$ to a centrally preconditioned system $R^{-1}\mathcal{A}R^{-T}$ with R being the Cholesky factor of $\mathcal{P}$. The usual way to obtain the preconditioned version of MINRES as it is shown in [19] is to plug in the centrally preconditioned system into the unpreconditioned MINRES method and then see that the expensive Cholesky decomposition is not necessary. Compared to the unpreconditioned method, the resulting preconditioned scheme only needs little extra storage and the additional cost of solving a system with the preconditioner $\mathcal{P}$. The derivation of efficient preconditioners is postponed until the next section.

We now want to discuss a different way of arriving at the preconditioned MINRES method. The minimization of the residual that defines the method is in its standard from done in the 2-norm, but this is not necessary for the success of the scheme. We here take our motivation from some recent work in operator preconditioning [28, 32, 74, 76] where the preconditioner is a crucial ingredient in the well-posedness of MINRES.

The derivation by Herzog and coauthors who consider the MINRES method from a function space perspective [28,32] is based on the mapping properties

of the operator $\mathcal{A}$, which is defined as $\mathcal{A} : X \to X^*$. Here X is a Hilbert space and X^* its dual space.

A crucial role in the derivation of the operator preconditioned MINRES method is played by the Riesz representer $\mathcal{H}$, which is a bounded linear operator that maps from the dual space X^* back to the original space X. For the minimization of $\min \|r_k\|_{\mathcal{H}}$ the authors in [28] give the Lanczos relation

$$\mathcal{A}\mathcal{H}V = VT$$

where the matrix of basis vectors V_k is now $\mathcal{H}-$orthogonal. This can be seen as a tridiagonalization of the matrix

$$V^T\mathcal{H}\mathcal{A}\mathcal{H}V = T.$$

The initial residual is again of importance as the first vector in V is $v_1 = r_0/\|r_0\|_{\mathcal{H}}$. The Lanczos iteration at step k then uses the k-th column of

$$\mathcal{A}\mathcal{H}Ve_k = VTe_k$$

and in more detail

$$\mathcal{A}\mathcal{H}v_k = \left(\beta_{k+1}v_{k+1} + \alpha_k v_k + \beta_{k-1}v_{k-1}\right).$$

To determine the coefficient α_k, we multiply the previous equation on the left by $v_k^T\mathcal{H}$ to get

$$\begin{aligned} v_k^T\mathcal{H}\mathcal{A}\mathcal{H}v_k &= \beta_{k+1}v_k^T\mathcal{H}v_{k+1} + \alpha_k v_k^T\mathcal{H}v_k + \beta_{k-1}v_k^T\mathcal{H}v_{k-1} && (1.22)\\ &= \alpha_k. && (1.23)\end{aligned}$$

Introducing $\mathcal{H}v_k =: z_k$, the parameter is computed via $\alpha_k = z_k^T\mathcal{A}z_k$. The parameters β_k are determined via the $\mathcal{H}$-norm of the vectors. The matrix form of the Lanczos process is then given by

$$\mathcal{A}\mathcal{H}V_k = V_{k+1}T_{k+1,k}$$

and we can use this to derive the appropriate result for MINRES. The residual norm in the $\mathcal{H}$-inner product then becomes

$$\begin{aligned} \|r_k\|_{\mathcal{H}} &= \|b - \mathcal{A}x_k\|_{\mathcal{H}} \\ &= \|b - \mathcal{A}x_0 - \mathcal{A}\mathcal{H}V_k y_k\|_{\mathcal{H}} \\ &= \|r_0 - V_{k+1}T_{k+1}y_k\|_{\mathcal{H}} \\ &= \|\|r_0\|_{\mathcal{H}}\, e_1 - T_{k+1}y_k\|_{\mathcal{H}}, \end{aligned} \qquad (1.24)$$

using the fact that $x_k = x_0 + \mathcal{H}V_k y_k$.

Note also that the Riesz representer and the inner product matrix $\mathcal{H}$ given above take the role of the preconditioner, i.e., $\mathcal{H} = \mathcal{P}^{-1}$. For more

recent work regarding the derivation of preconditioners based on the underlying PDE operators we refer to [28, 32, 40, 51, 76, 94] among others.

Here the notion is slightly different than the one presented in [65] where the author starts with the tridiagonalization of $\hat{\mathcal{A}}$

$$\hat{\mathcal{A}}V = VT$$

and an $\mathcal{H}-$ orthogonal matrix V. This would then result in

$$V^T\mathcal{H}\hat{\mathcal{A}}V = T$$

which is not identical to the formulation of Herzog and co-authors if $\hat{\mathcal{A}} = \mathcal{A}$. The mapping properties of $\mathcal{A}$ and considering Pestana's approach with preconditioning it can then be seen that the matrix $\hat{\mathcal{A}} = \mathcal{A}\mathcal{H}$, i.e. the right preconditioned matrix, represents the correct choice. Note that for this discussion to remain true we assume that $\hat{\mathcal{A}}$ is self-adjoint in the $\mathcal{H}-$inner product.

From the previous discussion and the fact that the inner product matrix acts as a preconditioner, the residual at step k is expressed via the following relation

$$r_k = p_k(\mathcal{A}\mathcal{H})r_0$$

where $p_k \in \Pi_0^k$ is the polynomial of degree at most k satisfying $p_k(0) = 1$. The convergence of MINRES is then given via

$$\|r_k\|_{\mathcal{H}} \leq \min_{p_k \in \Pi_0^k} \max_{\lambda \in \rho(\mathcal{A}\mathcal{H})} |p_k(\lambda)| \, \|r_0\|_{\mathcal{H}} \tag{1.25}$$

where $\rho(\mathcal{A}\mathcal{H})$ denotes the spectrum of the matrix $\mathcal{A}\mathcal{H}$. Herzog and Sachs [32] present a nice convergence analysis for MINRES defined in function space. For the linear algebra formulation we refer to [22, 49, 50].

We have now realized that the preconditioner $\mathcal{P}$ can be identified with the inner product $\mathcal{H}$, which we will elaborate on in the next section. Additionally, we introduce a preconditioning framework that has been widely used for designing preconditioners for saddle point problems.

1.3 A preconditioning framework

One important result from the above made observations is that the preconditioner needs to define an inner product or in other words a symmetric positive definite matrix. While other preconditioners can also be used when different iterative solvers are employed, we here only focus on block-diagonal preconditioners. Our preconditioning strategy mainly follows a result presented in [54] for the general construction of preconditioners for saddle point systems. Similar results are found in [4, 48].

Lemma 1.1 (Proposition 1, [54]). *For the saddle point system*

$$\mathcal{A} = \begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix}$$

preconditioned by

$$\mathcal{P} = \begin{bmatrix} A & 0 \\ 0 & S \end{bmatrix}$$

with $S = BA^{-1}B^T$ the negative Schur-complement, the following holds

$$\mathcal{T}\left(\mathcal{T} - I\right)\left(\mathcal{T}^2 - \mathcal{T} - I\right) = 0$$

with $\mathcal{T} = \mathcal{P}^{-1}\mathcal{A}$ the preconditioned system matrix.

Lemma 1.1 implies that a good preconditioner is found when the $(1,1)$-block and the Schur-complement of the matrix are well approximated. A thorough analysis concerning the behaviour of the eigenvalues of the preconditioned system when both the $(1,1)$-block and the Schur-complement are approximated is given in [58]. We now focus on approximating the $(1,1)$-block and the Schur-complement of the saddle point system (1.13) where it holds that

$$A = \begin{bmatrix} \tau\mathcal{M}_{1/2} & 0 \\ 0 & \beta\tau\mathcal{M}_{1/2} \end{bmatrix}, \quad B = \begin{bmatrix} -\mathcal{K} & \tau\mathcal{M} \end{bmatrix}.$$

We start by discussing the approximation of the $(1,1)$-block of the matrix (1.13), i.e.,

$$A = \begin{bmatrix} \tau\mathcal{M}_{1/2} & 0 \\ 0 & \beta\tau\mathcal{M}_{1/2} \end{bmatrix}.$$

Note that the matrix $\mathcal{M}_{1/2}$ in the case of a distributed observation ($\Omega_1 = \Omega$) over all time-steps is a block-diagonal matrix with mass matrices as diagonal blocks. This means that in order to approximate $\mathcal{M}_{1/2}$, we need to approximate the mass matrix M efficiently. In the case when M is lumped, i.e., diagonal, this is a trivial task. For consistent mass matrices one can either resort to a diagonal approximation of M or one can use the Chebyshev semi-iteration [25, 26] shown in Algorithm 1.1. In the case that we do not observe the desired state on the full domain Ω, the matrix $\mathcal{M}_{1/2}$ is a block-diagonal matrix consisting of matrices M_1 where this matrix only contains contributions from the domain Ω_1. This means that the matrix M_1 is semi-definite. For the preconditioning of this matrix we introduce a small parameter ν to obtain the preconditioning matrix $\tilde{M}_1 = M_1 + \nu\tilde{I}_1$, where $\tilde{I}_1$ is an identity matrix associated with the degrees of freedom in the part $\Omega\backslash\Omega_1$. A more detailed description of this parameter is found in [8, 83]. In general ν will depend on the mesh parameter h, the time-step τ, and the regularization parameter β.

1: Set $D = diag(M)$
2: Set relaxation parameter ω
3: Compute $g = \omega D^{-1}\hat{b}$
4: Set $S = (I - \omega D^{-1}M)$ (this can be used implicitly)
5: Set $w_0 = 0$ and $w_1 = Sw_{k-1} + g$
6: $c_0 = 2$ and $c_1 = \omega$
7: **for** $k = 2, \ldots, l$ **do**
8: $\quad c_{k+1} = \omega c_k - \frac{1}{4}c_{k-1}$
9: $\quad \eta_{k+1} = \omega \frac{c_k}{c_{k+1}}$
10: $\quad w_{k+1} = \eta_{k+1}(Sw_k + g - w_{k-1}) + w_{k-1}$
11: **end for**

Algorithm 1.1: Chebyshev semi-iterative method to solve $Mw = \hat{b}$ for a number of l steps

For nonlinear problems, the $(1,1)$-block is typically more complicated in structure and therefore harder to approximate. Different nonlinear solvers result in different structures of the $(1,1)$-block and tailored preconditioners need to be devised.

Typically the more challenging part comes from the approximation of the Schur-complement S. For reasons of exposition we will in this chapter occasionally discuss solvers for the steady control problem, which in its discretized form is written as

$$\begin{bmatrix} M & 0 & -K^T \\ 0 & \beta M & M \\ -K & M & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} M\bar{\mathbf{y}} \\ 0 \\ \mathbf{c} \end{bmatrix}. \tag{1.26}$$

Preconditioners for this problem have been studied extensively in the literature (cf. [19, 33, 68, 69, 76]). Following the above approach, we see that approximating the block-diagonal matrix

$$\begin{bmatrix} M & 0 \\ 0 & \beta M \end{bmatrix}$$

is essential and can in this case easily be done using the techniques mentioned earlier. The Schur-complement

$$S = KM^{-1}K^T + \frac{1}{\beta}M$$

is typically harder to approximate. For larger values of β ignoring the second term $\frac{1}{\beta}M$ in S can be successful [68]. A more robust approach for

approximating S was presented in [63] where the approximation

$$\hat{S} = (K + \frac{1}{\sqrt{\beta}}M)M^{-1}(K + \frac{1}{\sqrt{\beta}}M)^T$$

was presented. Here, we note that K and M are symmetric and positive definite which makes the transpose used redundant[1]. The effectiveness of this approach can be seen by analyzing the eigenvalues of $\hat{S}^{-1}S$. This can be done looking at the Rayleigh-quotient

$$\frac{v^T S v}{v^T \hat{S} v} = \frac{v^T K M^{-1} K^T v + \frac{1}{\beta} v^T M v}{v^T (K + \frac{1}{\sqrt{\beta}}M) M^{-1} (K + \frac{1}{\sqrt{\beta}}M)^T v} = \frac{a^T a + b^T b}{a^T a + b^T b + 2a^T b}$$

with $a = M^{-1/2}K^T v$ and $b = M^{1/2}v$. We can now easily show that the Rayleigh quotient is bounded independent of all the system parameters. For this we use the fact that

$$(a-b)^T(a-b) \geq 0 \Rightarrow a^T a + b^T b \geq 2a^T b$$

and this gives a lower bound of $\frac{1}{2}$ for the Rayleigh quotient. The upper bound of 1 that we want to establish follows from the fact that

$$2a^T b = 2v^T K v \geq 0$$

since K is positive semi-definite and in the case of a Dirichlet boundary condition even positive definite. The result is that the eigenvalues of $\hat{S}^{-1}S$ lie in the interval $[\frac{1}{2}, 1]$ independent of all system parameters [63]. In a similar fashion one can obtain an eigenvalue result for the control of the heat equation. We here only state the result from [62] and refer to the paper shown in Appendix A.3 for a more detailed discussion.

Theorem 1.2. *If*

$$S = \frac{1}{\tau}\mathcal{K}\mathcal{M}_{1/2}^{-1}\mathcal{K}^T + \frac{\tau}{\beta}\mathcal{M}\mathcal{M}_{1/2}^{-1}\mathcal{M}, \tag{1.27}$$

and

$$\widehat{S} = \frac{1}{\tau}\left(\mathcal{K} + \frac{\tau}{\sqrt{\beta}}\mathcal{M}\right)\mathcal{M}_{1/2}^{-1}\left(\mathcal{K} + \frac{\tau}{\sqrt{\beta}}\mathcal{M}\right)^T, \tag{1.28}$$

then

$$\lambda(\widehat{S}^{-1}S) \in [\frac{1}{2}, 1].$$

[1] We nevertheless use the transpose as the matrices are not necessarily symmetric for other problems.

If we now look at the structure of the matrix

$$\mathcal{K} + \frac{\tau}{\sqrt{\beta}}\mathcal{M} = \begin{bmatrix} L & & & \\ -M & L & & \\ & \ddots & \ddots & \\ & & -M & L \end{bmatrix}$$

with $L = (1 + \frac{\tau}{\sqrt{\beta}})M + \tau K$, we see that the inversion of this matrix and also its transpose are forward and backward substitutions. We thus require the solution or approximation solution with the diagonal blocks $(1+\frac{\tau}{\sqrt{\beta}})M+\tau K$. As this itself is a costly process due to K and M being large-scale finite element matrices, we employ an approximate solve for the diagonal blocks using multigrid techniques. For this both geometric [29, 92] and algebraic [21,71] multigrid techniques can be used. We here illustrate the performance of this scheme by showing results from Appendix A.3 in Table 1.1. It can be seen that the proven robustness can be observed in a practical scenario. The degrees of freedom listed in Table 1.1 are only the spatial degrees of freedom. This means the system size is much larger.

DoF	MINRES(T)	MINRES(T)	MINRES(T)
	$\beta = 1e-2$	$\beta = 1e-4$	$\beta = 1e-6$
4913	10(2)	12(2)	12(2)
35937	10(14)	12(17)	12(18)
274625	10(148)	12(171)	12(170)

Table 1.1: Results for the robust Schur-complement approximation and an optimal control problem with distributed control for the heat equation. Various mesh-sizes and different values for β are computed. We have chosen 20 time steps and a tolerance of 10^{-4} for convergence of MINRES.

Employing such a robust Schur-complement approach has proven efficient in many cases. We point to [5, 64, 78, 83] for some examples involving linear PDEs. For nonlinear problems this approach has also shown promising results but typically it is harder to theoretically underpin the performance as was done in Theorem 1.2 for the heat equation with distributed control. We point to [61, 80] for some first results on reaction-diffusion-like systems. For this we point to Appendices A.6, A.7, and A.8. Similar results could be proven for time-periodic systems shown in Appendix A.4 and [78].

Another technique to develop preconditioners that has recently received more attention is the so-called *operator preconditioning*. For an introduction to this approach we point to [2, 28, 51, 76, 94]. We here briefly introduce the idea for the steady-state problem and note that for the transient case, first efforts have been made in [2, 74]. For now consider the steady state optimal

control problem. Following the notation used in [32] we write the steady control problem with $\Omega_1 = \Omega_2 = \Omega$ as the following linear system

$$\begin{aligned}\langle U(y,u,p),(q,g,v)\rangle = &\int_\Omega yq + \beta\int_\Omega ug + \int_\Omega \nabla y\cdot\nabla v - \int_\Omega uv \\ &- \int_\Omega \nabla p\cdot\nabla q + \int_\Omega pg\end{aligned}$$

with right hand side

$$\langle b,(q,g,v)\rangle = \int_\Omega \bar{y}q$$

assuming zero Dirichlet conditions and test functions (q,g,v). The operator A acts on

$$X = H_0^1(\Omega)\times L_2(\Omega)\times H_0^1(\Omega)$$

into the dual space

$$\begin{aligned}X^* &= H_0^1(\Omega)^*\times L_2(\Omega)^*\times H_0^1(\Omega)^* \\ &= H^{-1}(\Omega)\times L_2(\Omega)\times H^{-1}(\Omega).\end{aligned}$$

The inner product that we need from the previous section is now given by the Riesz representer

$$\mathcal{P}^{-1} = \begin{bmatrix} -\Delta^{-1} & 0 & 0 \\ 0 & I^{-1} & 0 \\ 0 & 0 & -\Delta^{-1}\end{bmatrix},$$

where $I : L_2(\Omega)\to L_2(\Omega)^*$. Herzog and Sachs discuss the convergence behaviour of MINRES for this case in [32], and by construction, this preconditioner shows mesh-independent behaviour. We here additionally want to present a technique presented by Mardal and Winther in [51], which also shows robustness with respect to the regularization parameter. The idea presented in [51] is that regularization-parameter dependent spaces are introduced. For this they consider the space

$$X_\beta = \left(L_2(\Omega)\cap\beta^{1/4}H_0^1(\Omega)\right)\times\beta^{1/2}L_2(\Omega)\times\left(\beta^{-1/2}L_2(\Omega)\cap\beta^{-1/4}H_0^1(\Omega)\right).$$

One needs to establish coercivity of the bilinear form representing the control problem by showing there exists an $\alpha > 0$ such that

$$\|y\|^2 + \beta\|u\|^2 \geq \alpha\left(\|y\|^2 + \beta^{1/2}\|\nabla y\|^2 + \beta\|u\|^2\right) \tag{1.29}$$

for all $y\in H_0^1(\Omega)$ and $u\in L_2(\Omega)$ that satisfy the state equation in the elliptic case

$$(\nabla y,\nabla v) = (u,v)\quad \forall v\in H_0^1(\Omega).$$

Here, the left hand side of (1.29) represents the bilinear form representing the contributions of state and control to the objective function and the right-hand side of the equation α times the norm on the space $\left(L_2(\Omega) \cap \beta^{1/4} H_0^1(\Omega)\right) \times \beta^{1/2} L_2(\Omega)$. The constraint for $v = y$ now results in

$$(\nabla y, \nabla y) = (u, y) \Rightarrow \|\nabla y\|^2 \leq \|y\| \, \|u\| \, .$$

We start by using

$$\left(\|y\| + \sqrt{\beta} \, \|u\|\right)^2 \geq 0 \tag{1.30}$$

$$\|y\|^2 + \beta \, \|u\|^2 \geq 2\sqrt{\beta} \, \|y\| \, \|u\| \tag{1.31}$$

$$\frac{1}{2} \|y\|^2 + \frac{1}{2} \beta \, \|u\|^2 \geq \sqrt{\beta} \, \|y\| \, \|u\| \geq \frac{\sqrt{\beta}}{2} \, \|y\| \, \|u\| \geq \frac{\sqrt{\beta}}{2} \, \|\nabla y\|^2 \, . \tag{1.32}$$

Now adding $\frac{1}{2} \|y\|^2 + \frac{1}{2} \beta \|u\|^2$ to both sides of (1.32) gives

$$\|y\|^2 + \beta \, \|u\|^2 \geq \frac{1}{2} \left(\sqrt{\beta} \, \|\nabla y\|^2 + \|y\|^2 + \beta \, \|u\|^2 \right)$$

and we see that for $\alpha = \frac{1}{2}$ the coercivity is satisfied. Furthermore, we need to check whether the inf-sup condition is also satisfied as then the choice of the above spaces is well justified. It reads in this case as

$$\sup_{(y,u) \in H_0^1 \times L_2} \frac{(\nabla y, \nabla v) - (u, v)}{\left(\|y\|^2 + \beta^{1/2} \, \|\nabla y\|^2 + \beta \, \|u\|^2 \right)^{1/2}} \geq \tag{1.33}$$

$$\eta \left(\beta^{-1} \, \|v\|^2 + \beta^{-1/2} \, \|\nabla v\|^2 \right)^{1/2} \tag{1.34}$$

where η is the inf-sup constant. On the other hand, if we choose $y = \beta^{-1/2} v$ and $u = -\beta^{-1} v$, then

$$\sup_{(y,u) \in H_0^1 \times L_2} \frac{(\nabla y, \nabla v) - (u, v)}{\left(\|y\|^2 + \beta^{1/2} \, \|\nabla y\|^2 + \beta \, \|u\|^2 \right)^{1/2}} \geq \tag{1.35}$$

$$\frac{\beta^{-1/2} \, \|\nabla v\|^2 + \beta^{-1} \, \|v\|}{\left(\beta^{-1} \, \|v\|^2 + \beta^{-1/2} \, \|\nabla v\|^2 + \beta^{-1} \, \|v\|^2 \right)^{1/2}} \geq \tag{1.36}$$

$$\frac{\beta^{-1/2} \, \|\nabla v\|^2 + \beta^{-1} \, \|v\|}{\sqrt{2} \left(\beta^{-1} \, \|v\|^2 + \beta^{-1/2} \, \|\nabla v\|^2 \right)^{1/2}} = \tag{1.37}$$

$$\frac{1}{\sqrt{2}} \left(\beta^{-1} \, \|v\|^2 + \beta^{-1/2} \, \|\nabla v\|^2 \right)^{1/2} . \tag{1.38}$$

Since this relation holds for all v including its infimum the inf-sup constant

in this case is given by $\eta = \frac{1}{\sqrt{2}}$. With this result we can now establish the design for our preconditioner in this case. We start by noting that the dual space is given as

$$X_\beta^* = \left(L_2(\Omega) + \beta^{-1/4} H_0^1(\Omega)^*\right) \times \beta^{-1/2} L_2(\Omega) \times \left(\beta^{1/2} L_2(\Omega) + \beta^{1/4} H_0^1(\Omega)^*\right).$$

The operator mapping

$$X_\beta = \left(L_2(\Omega) \cap \beta^{1/4} H_0^1(\Omega)\right) \times \beta^{1/2} L_2(\Omega) \times \left(\beta^{-1/2} L_2(\Omega) \cap \beta^{-1/4} H_0^1(\Omega)\right)$$

to X_β^* is then given by

$$\begin{bmatrix} I - \beta^{1/2}\Delta & & \\ & \beta I & \\ & & \beta^{-1}(I - \beta^{1/2}\Delta) \end{bmatrix}$$

and induces the following preconditioner

$$\mathcal{P}^{-1} = \begin{bmatrix} (I - \beta^{1/2}\Delta)^{-1} & & \\ & \beta^{-1} I & \\ & & \beta(I - \beta^{1/2}\Delta)^{-1} \end{bmatrix}.$$

Again as in the result taken from Herzog and Sachs, this preconditioner needs to be implemented efficiently. The realization of the individual blocks via a spectrally equivalent approximation such as a multigrid process typically shows outstanding performance. The resulting blocks obtained in the preconditioner resemble the blocks obtained in the robust Schur-complement approximation introduced by Pearson and Wathen [63].

In the case of parabolic control problems first steps for the design of operator based preconditioners have been made (cf. [2, 74]).

1.4 Data-compressed approximations

We have so far presented methodology to efficiently simultaneous space-time problems. Convergence theory and numerical results indicate the efficiency of this approach. Nevertheless, a possible shortcoming is the storage required for the space-time vectors. We point out that the storage of the matrix system is only slightly higher than for the steady problem. We now introduce a technique that we proposed in [79] also described in detail in Appendix A.9.

We recall the definition of the Kronecker product

$$W \otimes V = \begin{bmatrix} w_{11}V & \dots & w_{1m}V \\ \vdots & \ddots & \vdots \\ w_{n1}V & \dots & w_{nm}V \end{bmatrix}$$

and note that we can write our linear system (1.13) using Kronecker notation and get

$$\underbrace{\begin{bmatrix} D_1 \otimes \tau M_1 & 0 & -\left(I_N \otimes L + C^T \otimes M\right) \\ 0 & D_2 \otimes \beta\tau M_2 & D_3 \otimes \tau M \\ -\left(I_N \otimes L + C \otimes M\right) & D_3 \otimes \tau M & 0 \end{bmatrix}}_{\mathcal{A}} \begin{bmatrix} y \\ u \\ p \end{bmatrix} = \tag{1.39}$$

$$\begin{bmatrix} D_1 \otimes \tau M_1 y_{obs} \\ 0 \\ d \end{bmatrix},$$

where $D_1 = D_2 = \operatorname{diag}\left(\frac{1}{2}, 1, \dots, 1, \frac{1}{2}\right)$, $L = M + \tau K$, and $D_3 = I_N$. Additionally, the matrix $C \in \mathbb{R}^{N,N}$ is given by

$$C = \begin{bmatrix} 0 & & & \\ -1 & 0 & & \\ & \ddots & \ddots & \\ & & -1 & 0 \end{bmatrix}$$

and represents the implicit Euler scheme. It is of course possible to use a different discretization in time. So far we have simply reformulated the previously given system (1.13). But our goal was to derive a scheme that allows for a reduction in storage requirement for the vectors y, u, and p. For this we remind the reader of the definition of the vec operator via

$$\operatorname{vec}(W) = \operatorname{vec}([w_1, \dots, w_N]) = \begin{bmatrix} w_1 \\ \vdots \\ w_N \end{bmatrix}$$

as well as the relation

$$\left(W^T \otimes V\right) \operatorname{vec}(Y) = \operatorname{vec}(VYW).$$

Now employing these definitions and using the following notation

$$Y = [\ y_1, y_2, \dots, y_N\ ], \quad U = [\ u_1, u_2, \dots, u_N\ ], \quad P = [\ p_1, p_2, \dots, p_N\ ]$$

we get for the matrix vector multiplication that

$$\begin{bmatrix} D_1 \otimes \tau M_1 & 0 & -\left(I_N \otimes L + C^T \otimes M\right) \\ 0 & D_2 \otimes \beta\tau M_2 & D_3 \otimes \tau N^T \\ -\left(I_N \otimes L + C \otimes M\right) & D_3 \otimes \tau N & 0 \end{bmatrix} \begin{bmatrix} \mathrm{vec}(Y) \\ \mathrm{vec}(U) \\ \mathrm{vec}(P) \end{bmatrix} \tag{1.40}$$

$$= \mathrm{vec}\left(\begin{bmatrix} \tau M_1 Y D_1^T - LPI_N^T - MPC \\ \tau\beta M_2 U D_2^T + \tau N^T P D_3^T \\ -LYI_N^T - MYC^T + \tau N U D_3^T \end{bmatrix} \right).$$

So far nothing is gained from rewriting the problem in this form. As was previously done in [6] we assume for now that Y, U, and P are approximations to the true solutions and that these approximations are represented by a low-rank representation. Then any iterative Krylov subspace solver can be implemented using a low-rank version of (1.40) for the matrix vector multiplication. To see this, we denote the low-rank representations by

$$Y = W_Y V_Y^T \text{ with } W_Y \in \mathbb{R}^{n_1,k_1}, V_Y \in \mathbb{R}^{N,k_1} \tag{1.41}$$

$$U = W_U V_U^T \text{ with } W_U \in \mathbb{R}^{n_2,k_2}, V_U \in \mathbb{R}^{N,k_2} \tag{1.42}$$

$$P = W_P V_P^T \text{ with } W_P \in \mathbb{R}^{n_1,k_3}, V_P \in \mathbb{R}^{N,k_3} \tag{1.43}$$

with $k_{1,2,3}$ being small in comparison to N and rewrite (1.40) accordingly to get

$$\begin{bmatrix} \tau M_1 W_Y V_Y^T D_1^T - LW_P V_P^T I_N^T - MW_P V_P^T C \\ \tau\beta M_2 W_U V_U^T D_2^T + \tau N^T W_P V_P^T D_3^T \\ -LW_Y V_Y^T I_N^T - MW_Y V_Y^T C^T + \tau N W_U V_U^T D_3^T \end{bmatrix}. \tag{1.44}$$

Note that we skipped the vec operator and instead use matrix-valued unknowns. We can write the block-rows of (1.44) as

$$\text{(first block-row)} \quad \begin{bmatrix} \tau M_1 W_Y & -LW_P & -MW_P \end{bmatrix} \begin{bmatrix} V_Y^T D_1^T \\ V_P^T I_N^T \\ V_P^T C \end{bmatrix},$$

$$\text{(second block-row)} \quad \begin{bmatrix} \tau\beta M_2 W_U & \tau N^T W_P \end{bmatrix} \begin{bmatrix} V_U^T D_2^T \\ V_P^T D_3^T \end{bmatrix},$$

$$\text{(third block-row)} \quad \begin{bmatrix} -LW_Y & -MW_Y & \tau N W_U \end{bmatrix} \begin{bmatrix} V_Y^T I_N^T \\ V_Y^T C^T \\ V_U^T D_3^T \end{bmatrix}. \tag{1.45}$$

We have now shown that the matrix vector product with the KKT matrix of our optimization problem can be performed using low-rank methodology.

This is easily seen from (1.45). While the rank of the matrix

$$\begin{bmatrix} \tau M_1 W_Y & -LW_P & -MW_P \end{bmatrix}$$

grows with this multiplication, one can condense the result again using for example a truncated SVD or a QR reduction [47, 79].

Using such truncation approaches for the approximation to state, control, and adjoint state, we can perform a full cycle of an iterative scheme. As our system matrix is indefinite, we employ MINRESonce more. Nevertheless, all other Krylov solvers can be used. In Algorithm 1.2, we show a skeleton version of MINRES in the full format and highlight the work intensive parts of the algorithm. These involve the matrix vector product and the preconditioning step. If both can be performed maintaining the low-rank form the algorithm can then be rewritten for a low-rank approximation.

Zero-Initialization of $v^{(0)}, w^{(0)}$, and $w^{(1)}$.
Choose $u^{(0)}$
Set $v^{(1)} = b - \mathcal{A}u^{(0)}$
Solve $\mathcal{P}z^{(1)} = v^{(1)}$
...
for $j = 1$ until convergence **do**
  $z^{(j)} = z^{(j)}/\gamma_j$
  $\delta_j = \langle \mathcal{A}z^{(j)}, z^{(j)} \rangle$
  Compute $v^{(j+1)} = \mathcal{A}z^{(j)} - \delta_j/\gamma_j v^{(j)} - \gamma_j/\gamma_{j-1} v^{(j-1)}$
  Solve $\mathcal{P}z^{(j+1)} = v^{(j+1)}$
  $\gamma_{j+1} = \sqrt{\langle z^{(j+1)}, v^{(j+1)} \rangle}$
  ...
  if Convergence criterion fulfilled **then**
    Compute approximate solution
    stop
  end if
end for

Algorithm 1.2: Skeleton of MINRES algorithm with gray areas illustrating the computationally expensive parts.

The matrix vector product is a crucial component of any Krylov subspace solver and (1.45) enables us to proceed to a low-rank Krylov scheme as shown in Algorithm 1.3. This algorithm is almost identical to the classical scheme but working with matrix valued unknowns and maintaining the low-rank nature. One of the crucial steps of the algorithm is illustrated by the parentheses $\{\}$, which indicate the concatenation and truncation process. To recall this again, relation (1.45) can be truncated using efficient QR or SVD based reduction techniques where even benign rank increase can be avoided [47, 79], see also Appendix A.9. Again, preconditioning is crucial for this scheme and we also refer to Appendix A.9 for details on potential preconditioners.

The use of low-rank techniques for linear systems coming from high-dimensional problems has seen a lot of interest as of late [3,47,79] with a vast

Zero-Initialization of $V_{11}^{(0)}, \ldots, W_{11}^{(0)}, \ldots$, and $W_{11}^{(1)}, \ldots$.
Choose $U_{11}^{(0)}, U_{12}^{(0)}, U_{21}^{(0)}, U_{22}^{(0)}, U_{31}^{(0)}, U_{32}^{(0)}$
Set $V_{11}, V_{12}, \ldots$ to normalized residual
while residual norm $>$ tolerance **do**
$Z_{11}^{(j)} = Z_{11}^{(j)}/\gamma_j$, $Z_{21}^{(j)} = Z_{21}^{(j)}/\gamma_j$, $Z_{31}^{(j)} = Z_{31}^{(j)}/\gamma_j$,
$[F_{11}, F_{12}, F_{21}, F_{22}, F_{31}, F_{32}] = \texttt{Amult}(Z_{11}^{(j)}, Z_{12}^{(j)}, Z_{21}^{(j)}, Z_{22}^{(j)}, Z_{31}^{(j)}, Z_{32}^{(j)})$
$\delta_j = \texttt{traceproduct}(F_{11}, F_{12}, F_{21}, F_{22}, F_{31}, F_{32}, Z_{11}^{(j)}, Z_{12}^{(j)}, Z_{21}^{(j)}, Z_{22}^{(j)}, Z_{31}^{(j)}, Z_{32}^{(j)})$
$V_{11}^{(j+1)} = \left\{ \begin{array}{ccc} F_{11} & -\frac{\delta_j}{\gamma_j} V_{11}^{(j)} & -\frac{\gamma_j}{\gamma_{j-1}} V_{11}^{(j-1)} \end{array} \right\}$, $V_{12}^{(j+1)} = \left\{ \begin{array}{ccc} F_{12} & V_{12}^{(j)} & V_{12}^{(j-1)} \end{array} \right\}$
$V_{21}^{(j+1)} = \left\{ \begin{array}{ccc} F_{21} & -\frac{\delta_j}{\gamma_j} V_{21}^{(j)} & -\frac{\gamma_j}{\gamma_{j-1}} V_{21}^{(j-1)} \end{array} \right\}$, $V_{22}^{(j+1)} = \left\{ \begin{array}{ccc} F_{22} & V_{22}^{(j)} & V_{22}^{(j-1)} \end{array} \right\}$
$V_{31}^{(j+1)} = \left\{ \begin{array}{ccc} F_{31} & -\frac{\delta_j}{\gamma_j} V_{31}^{(j)} & -\frac{\gamma_j}{\gamma_{j-1}} V_{31}^{(j-1)} \end{array} \right\}$, $V_{32}^{(j+1)} = \left\{ \begin{array}{ccc} F_{32} & V_{32}^{(j)} & V_{32}^{(j-1)} \end{array} \right\}$
$\left\{ Z_{11}^{(j+1)}, Z_{12}^{(j+1)}, Z_{21}^{(j+1)}, Z_{22}^{(j+1)}, Z_{31}^{(j+1)}, Z_{32}^{(j+1)} \right\} =$
$\texttt{Aprec}(V_{11}^{(j+1)}, V_{12}^{(j+1)}, V_{21}^{(j+1)}, V_{22}^{(j+1)}, V_{31}^{(j+1)}, V_{32}^{(j+1)})$
$\gamma_{j+1} = \sqrt{\texttt{tracepoduct}(Z_{11}^{(j+1)}, \ldots, V_{11}^{(j+1)}, \ldots)}$
$\alpha_0 = c_j \delta_j - c_{j-1} s_j \gamma_j$
$\alpha_1 = \sqrt{\alpha_0^2 + \gamma_{j+1}^2}$
$\alpha_2 = s_j \delta_j + c_{j-1} c_j \gamma_j$
$\alpha_3 = s_{j-1} \gamma_j$
$c_{j+1} = \frac{\alpha_0}{\alpha_1}$
$s_{j+1} = \frac{\gamma_{j+1}}{\alpha_1}$
$W_{11}^{(j+1)} = \left\{ \begin{array}{ccc} Z_{11}^{(j)} & -\alpha_3 W_{11}^{(j-1)} & -\alpha_2 W_{11}^{(j)} \end{array} \right\}$, $W_{12}^{(j+1)} = \left\{ \begin{array}{ccc} Z_{12}^{(j)} & W_{12}^{(j-1)} & W_{12}^{(j)} \end{array} \right\}$
$W_{21}^{(j+1)} = \left\{ \begin{array}{ccc} Z_{21}^{(j)} & -\alpha_3 W_{21}^{(j-1)} & -\alpha_2 W_{21}^{(j)} \end{array} \right\}$, $W_{22}^{(j+1)} = \left\{ \begin{array}{ccc} Z_{22}^{(j)} & W_{22}^{(j-1)} & W_{22}^{(j)} \end{array} \right\}$
$W_{31}^{(j+1)} = \left\{ \begin{array}{ccc} Z_{31}^{(j)} & -\alpha_3 W_{31}^{(j-1)} & -\alpha_2 W_{31}^{(j)} \end{array} \right\}$, $W_{32}^{(j+1)} = \left\{ \begin{array}{ccc} Z_{32}^{(j)} & W_{32}^{(j-1)} & W_{32}^{(j)} \end{array} \right\}$
if Convergence criterion fulfilled **then**
Compute approximate solution
stop
end if
end while

Algorithm 1.3: Low-rank MINRES

majority of the work considering not just matrix-valued but tensor-valued equations [18, 44]. As is clear from the algorithm, the initial rank comes from the right-hand side of the equation as this is the starting vector for our Krylov subspace. For optimization problems this can often be expected to have a small rank. A more detailed discussion can be found in [79].

Additionally, one needs to worry about preconditioning. For this, we recall the system matrix

$$\begin{bmatrix} D_1 \otimes \tau M_1 & 0 & -\left(I_N \otimes L + C^T \otimes M\right) \\ 0 & D_2 \otimes \beta\tau M_2 & D_3 \otimes \tau M \\ -\left(I_N \otimes L + C \otimes M\right) & D_3 \otimes \tau M & 0 \end{bmatrix}$$

and our previously discussed strategy to approximate the $(1,1)$-block

$$\begin{bmatrix} D_1 \otimes \tau M_1 & 0 \\ 0 & D_2 \otimes \beta\tau M_2 \end{bmatrix}$$

and the Schur-complement

$$\begin{aligned} S = &\left(I_N \otimes L + C \otimes M\right)\left(D_1 \otimes \tau M_1\right)^{-1}\left(I_N \otimes L + C^T \otimes M\right) \\ &+ (D_3 \otimes \tau M)(D_2 \otimes \beta\tau M_2)^{-1} D_3 \otimes \tau M. \end{aligned}$$

The inversion of the $(1,1)$-block is again easy and maintains the low-rank structure of the approximations. The Schur-complement can in a similar manner to before be approximated using

$$\hat{S} = \left(I_N \otimes \hat{L} + C \otimes M\right)\left(D_1 \otimes \tau M_1\right)^{-1}\left(I_N \otimes \hat{L} + C^T \otimes M\right),$$

where $\hat{L}$ is chosen according to the matching approach presented earlier. In order to use our presented low-rank approach, the evaluation of $\hat{S}^{-1}$ needs to maintain this structure. Let us note that the first and last term in $\hat{S}$ are equations resembling generalized Sylvester equations for which low-rank techniques are available (see [77] and the references mentioned therein).

It is also possible to extend this methodology to the case of a nonlinear PDE-constraint. Namely, the Navier-Stokes equations, which then results in a much more complex structure of the matrix blocks. First results for this approach seem to be promising [17].

1.5 Conclusion

Our introduction presented here was based on a linear algebra view of part of the field of PDE-constrained optimization. In order to get to this point we have seen that sophisticated discretization schemes are needed. We showed that one can find a large-scale, highly structured linear system at the core

of the optimization procedure. We have focused on linear constraints with a convex objective function such that the first order conditions are sufficient to solve the optimization problem. Nevertheless, similar systems are found at the heart of optimization schemes when nonlinear problems need to be tackled.

We have discussed the use of MINRES as an iterative solver and motivated how the preconditioned version of this algorithm can be derived from looking at nonstandard inner products. We also discussed that any iterative solver is enhanced by preconditioning and in the case of MINRES preconditioners needs to be symmetric and positive definite. In our application, they can be thought of as inner product representations for the underlying function spaces. In particular, we have discussed possible preconditioning strategies for the optimization problems. We have followed a Schur-complement approach and illustrated how robustness for this method can be proven for some cases. We have additionally illustrated that similar constructs are obtained when operator preconditioning is employed.

While the results obtained for such a space-time approach mostly show good convergence behaviour, the possible downside of a high-storage demand remains. In order to avoid this potential pitfall, we discussed a low-rank framework that allows an optimal low-rank representation of the full solution and can be efficiently combined with iterative solvers such as MINRES.

BIBLIOGRAPHY

[1] T. Akman and B. Karasözen, *Variational time discretization methods for optimal control problems governed by diffusion–convection–reaction equations*, J. Comput. Appl. Math., 272 (2014), pp. 41–56.

[2] R. Andreev, *Stability of space-time Petrov-Galerkin discretizations for parabolic evolution equations*, PhD thesis, Diss., Eidgenössische Technische Hochschule ETH Zürich, Nr. 20842, 2012, 2012.

[3] R. Andreev and C. Tobler, *Multilevel preconditioning and low-rank tensor iteration for space–time simultaneous discretizations of parabolic pdes*, Numer. Linear. Algebr. , 22 (2015), pp. 317–337.

[4] R. Bank, B. Welfert, and H. Yserentant, *A class of iterative methods for solving saddle point problems*, Numer. Math., 56 (1989), pp. 645–666.

[5] A. T. Barker, T. Rees, and M. Stoll, *A fast solver for an H1 regularized optimal control problem*, Commun. Comput. Phys., 19 (2016), pp. 143–167.

[6] P. Benner and T. Breiten, *Low rank methods for a class of generalized Lyapunov equations and related issues*, Numer. Math., 124 (2013), pp. 441–470.

[7] M. Benzi, G. H. Golub, and J. Liesen, *Numerical solution of saddle point problems*, Acta Numer., 14 (2005), pp. 1–137.

[8] M. Benzi, E. Haber, and L. Taralli, *A preconditioning technique for a class of PDE-constrained optimization problems*, Adv. Comput. Math., 35 (2011), pp. 149–173.

[9] M. Bergounioux, K. Ito, and K. Kunisch, *Primal-dual strategy for constrained optimal control problems*, SIAM J. Control Optim., 37 (1999), pp. 1176–1194.

[10] M. Bergounioux and K. Kunisch, *Primal-dual strategy for state-constrained optimal control problems*, Comput. Optim. Appl., 22 (2002), pp. 193–224.

[11] L. T. Biegler, O. Ghattas, M. Heinkenschloss, D. Keyes, and B. van Bloemen Waanders, *Real-time PDE-constrained Optimization*, vol. 3, SIAM, 2007.

[12] G. Biros and O. Ghattas, *Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization. I. The Krylov-Schur solver*, SIAM J. Sci. Comput., 27 (2005), pp. 687–713.

[13] A. Borzi, *Multigrid methods for parabolic distributed optimal control problems*, J. Comput. Appl. Math., 157 (2003), pp. 365–382.

[14] A. Borzì and V. Schulz, *Multigrid methods for PDE optimization*, SIAM Rev., 51 (2009), pp. 361–395.

[15] E. Burman, *Crank-Nicolson finite element methods using symmetric stabilization with an application to optimal control problems subject to transient advection-diffusion equations*, Commun. Math. Sci., 9 (2011), pp. 319–329.

[16] E. Casas, *Control of an elliptic problem with pointwise state constraints*, SIAM J. Control Optim., 24 (1986), pp. 1309–1318.

[17] S. Dolgov and M. Stoll, *Low-rank solutions to an optimization problem constrained by the Navier-Stokes equations*, Submitted, (2015).

[18] S. V. Dolgov and D. V. Savostyanov, *Alternating minimal energy methods for linear systems in higher dimensions*, SIAM J. Sci. Comput., 36 (2014), pp. A2248–A2271.

[19] H. C. Elman, D. Silvester, and A. Wathen, *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics*, Oxford University Press, 2014.

[20] H. C. Elman, D. J. Silvester, and A. J. Wathen, *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics*, Numerical Mathematics and Scientific Computation, Oxford University Press, New York, 2005.

[21] R. Falgout, *An Introduction to Algebraic Multigrid*, Computing in Science and Engineering, 8 (2006), pp. 24–33. Special Issue on Multigrid Computing.

[22] B. Fischer, *Polynomial based iteration methods for symmetric linear systems*, Wiley-Teubner Series Advances in Numerical Mathematics, John Wiley & Sons Ltd, Chichester, 1996.

[23] R. Fletcher, *Practical methods of optimization*, A Wiley-Interscience Publication, John Wiley & Sons Ltd., Chichester, second ed., 1987.

[24] G. H. Golub and C. F. Van Loan, *Matrix computations*, Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, MD, third ed., 1996.

[25] G. H. Golub and R. S. Varga, *Chebyshev semi-iterative methods, successive over-relaxation iterative methods, and second order Richardson iterative methods. I*, Numer. Math., 3 (1961), pp. 147–156.

[26] ——, *Chebyshev semi-iterative methods, successive over-relaxation iterative methods, and second order Richardson iterative methods. II*, Numer. Math., 3 (1961), pp. 157–168.

[27] A. Greenbaum, *Iterative methods for solving linear systems*, vol. 17 of Frontiers in Applied Mathematics, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.

[28] A. Günnel, R. Herzog, and E. Sachs, *A note on preconditioners and scalar products for Krylov methods in Hilbert space*, Electron. Trans. Numer. Anal., 40 (2014), pp. 13–20.

[29] W. Hackbusch, *Multigrid methods and applications*, vol. 4 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 1985.

[30] M. Heinkenschloss and D. Ridzal, *A matrix-free trust-region SQP method for equality constrained optimization*, SIAM J. Optim., 24 (2014), pp. 1507–1541.

[31] R. Herzog and K. Kunisch, *Algorithms for PDE-constrained optimization*, GAMM-Mitt., 33 (2010), pp. 163–176.

[32] R. Herzog and E. Sachs, *Superlinear Convergence of Krylov Subspace Methods for Self-Adjoint Problems in Hilbert Space*, SIAM J. Numer. Anal., 53 (2015), pp. 1304–1324.

[33] R. Herzog and E. W. Sachs, *Preconditioned conjugate gradient method for optimal control problems with control and state constraints*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 2291–2317.

[34] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. Nat. Bur. Stand, 49 (1952), pp. 409–436 (1953).

[35] M. Hintermüller, K. Ito, and K. Kunisch, *The primal-dual active set strategy as a semismooth Newton method*, SIAM J. Optim., 13 (2002), pp. 865–888.

[36] M. Hinze, *A variational discretization concept in control constrained optimization: the linear-quadratic case*, Comput. Optim. Appl., 30 (2005), pp. 45–61.

[37] M. Hinze, M. Köster, and S. Turek, *A Hierarchical Space-Time Solver for Distributed Control of the Stokes Equation*, tech. rep., SPP1253-16-01, 2008.

[38] ——, *A Space-Time Multigrid Solver for Distributed Control of the Time-Dependent Navier-Stokes System*, tech. rep., SPP1253-16-02, 2008.

[39] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich, *Optimization with PDE Constraints*, Mathematical Modelling: Theory and Applications, Springer-Verlag, New York, 2009.

[40] R. Hiptmair, *Operator preconditioning.*, Comput. Math. Appl., 52 (2006), pp. 699–706.

[41] K. Ito and K. Kunisch, *Lagrange multiplier approach to variational problems and applications*, vol. 15 of Advances in Design and Control, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2008.

[42] K. Ito, K. Kunisch, V. Schulz, and I. Gherman, *Approximate nullspace iterations for KKT systems*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 1835–1847.

[43] C. T. Kelley, *Iterative methods for optimization*, vol. 18 of Frontiers in Applied Mathematics, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1999.

[44] B. N. Khoromskij and C. Schwab, *Tensor-structured Galerkin approximation of parametric and stochastic elliptic PDEs*, SIAM J. Sci. Comput., 33 (2011), pp. 364–385.

[45] H. K. Kim and A. H. Hielscher, *A PDE-constrained SQP algorithm for optical tomography based on the frequency-domain equation of radiative transfer*, Inverse Probl., 25 (2009), p. 015010.

[46] M. Kollmann and W. Zulehner, *A robust preconditioner for distributed optimal control for Stokes flow with control constraints*, in Numerical Mathematics and Advanced Applications 2011, Springer, 2013, pp. 771–779.

[47] D. Kressner and C. Tobler, *Krylov subspace methods for linear systems with tensor product structure*, SIAM J. Matrix Anal. Appl, 31 (2010), pp. 1688–1714.

[48] Y. A. Kuznetsov, *Efficient iterative solvers for elliptic finite element problems on nonmatching grids*, Russian J. Numer. Anal. Math. Modelling, 10 (1995), pp. 187–211.

[49] J. Liesen and Z. Strakoš, *Krylov subspace methods: principles and analysis*, Oxford University Press, 2012.

[50] J. Liesen and P. Tichý, *Convergence analysis of Krylov subspace methods*, GAMM-Mitt., 27 (2004), pp. 153–173.

[51] K. Mardal and R. Winther, *Preconditioning discretizations of systems of partial differential equations*, Numer. Linear Algebr., 18 (2011), pp. 1–40.

[52] D. Meidner and B. Vexler, *Adaptive space-time finite element methods for parabolic optimization problems*, SIAM J. Control Optim., 46 (2007), pp. 116–142.

[53] G. Meurant, *Computer solution of large linear systems*, vol. 28 of Studies in Mathematics and its Applications, North-Holland Publishing Co, Amsterdam, 1999.

[54] M. F. Murphy, G. H. Golub, and A. J. Wathen, *A note on preconditioning for indefinite linear systems*, SIAM J. Sci. Comput, 21 (2000), pp. 1969–1972.

[55] N. M. Nachtigal, S. C. Reddy, and L. N. Trefethen, *How fast are nonsymmetric matrix iterations?*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 778–795.

[56] J. Nocedal and S. J. Wright, *Numerical optimization*, Springer Series in Operations Research, Springer-Verlag, New York, 1999.

[57] ——, *Numerical optimization*, Springer Series in Operations Research and Financial Engineering, Springer, New York, second ed., 2006.

[58] Y. Notay, *A new analysis of block preconditioners for saddle point problems*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 143–173.

[59] C. C. Paige and M. A. Saunders, *Solutions of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal, 12 (1975), pp. 617–629.

[60] J. W. Pearson, *Fast iterative solvers for PDE-constrained optimization problems*, PhD thesis, University of Oxford, 2013.

[61] J. W. Pearson and M. Stoll, *Fast Iterative Solution of Reaction-Diffusion Control Problems Arising from Chemical Processes*, SIAM J. Sci. Comput., 35 (2013), pp. 987–1009.

[62] J. W. Pearson, M. Stoll, and A. J. Wathen, *Regularization-robust preconditioners for time-dependent PDE-constrained optimization problems*, SIAM J. Matrix Anal. Appl., 33 (2012), pp. 1126–1152.

[63] J. W. Pearson and A. J. Wathen, *A new approximation of the Schur complement in preconditioners for PDE-constrained optimization*, Numer. Linear Algebr. , 19 (2012), pp. 816–829.

[64] J. W. Pearson and A. J. Wathen, *Fast iterative solvers for convection-diffusion control problems*, Electron. Trans. Numer. Anal., 40 (2013), pp. 294–310.

[65] J. Pestana, *Nonstandard inner products and preconditioned iterative methods*, PhD thesis, University of Oxford, 2011.

[66] E. E. Prudencio, R. Byrd, and X.-C. Cai, *Parallel full space SQP Lagrange–Newton–Krylov–Schwarz algorithms for PDE-constrained optimization problems*, SIAM J. Sci. Comput., 27 (2006), pp. 1305–1328.

[67] T. Rees, *Preconditioning iterative methods for PDE constrained optimization*, PhD thesis, Oxford University, 2010.

[68] T. Rees, H. S. Dollar, and A. J. Wathen, *Optimal solvers for PDE-constrained optimization*, SIAM J. Sci. Comput., 32 (2010), pp. 271–298.

[69] T. Rees and M. Stoll, *Block-triangular preconditioners for PDE-constrained optimization*, Numer. Linear Algebr., 17 (2010), pp. 977–996.

[70] T. Rees and A. J. Wathen, *Preconditioning iterative methods for the optimal control of the Stokes equations*, SIAM J. Sci. Comput., 33 (2011), pp. 2903–2926.

[71] J. W. Ruge and K. Stüben, *Algebraic multigrid*, in Multigrid methods, vol. 3 of Frontiers Appl. Math., SIAM, Philadelphia, PA, 1987, pp. 73–130.

[72] Y. Saad, *Iterative methods for sparse linear systems*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2003.

[73] Y. Saad and M. H. Schultz, *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput, 7 (1986), pp. 856–869.

[74] A. Schiela and S. Ulbrich, *Operator preconditioning for a class of inequality constrained optimal control problems*, SIAM J. Optim., 24 (2014), pp. 435–466.

[75] A. Schiela and M. Weiser, *Superlinear convergence of the control reduced interior point method for pde constrained optimization*, Comput. Optim. Appl., 39 (2008), pp. 369–393.

[76] J. Schöberl and W. Zulehner, *Symmetric indefinite preconditioners for saddle point problems with applications to pde-constrained optimization problems*, SIAM J. Matrix Anal. Appl, 29 (2007), pp. 752–773.

[77] V. Simoncini, *Computational methods for linear matrix equations*, To appear SIAM Rev., (2016).

[78] M. Stoll, *All-at-once solution of a time-dependent time-periodic PDE-constrained optimization problems*, IMA J. Numer. Anal., 34 (2014), pp. 1554–1577.

[79] M. Stoll and T. Breiten, *A low-rank in time approach to PDE-constrained optimization*, SIAM J. Sci. Comput., 37 (2015), pp. B1–B29.

[80] M. Stoll, J. W. Pearson, and P. K. Maini, *Fast Solvers for Optimal Control Problems from Pattern Formation*, J. Comput. Phys., 304 (2016), 27-45.

[81] M. Stoll, J. W. Pearson, and A. Wathen, *Preconditioners for state constrained optimal control problems with Moreau-Yosida penalty function*, Numer. Lin. Alg. Appl., 21 (2014), pp. 81–97.

[82] M. Stoll and A. Wathen, *Preconditioning for partial differential equation constrained optimization with control constraints*, Numer. Lin. Alg. Appl., 19 (2012), pp. 53–71.

[83] M. Stoll and A. Wathen, *All-at-once solution of time-dependent Stokes control*, J. Comput. Phys., 232 (2013), pp. 498–515.

[84] S. Takacs, *Efficient smoothers for all-at-once multigrid methods for Poisson and Stokes control problems*, in System Modeling and Optimization, Springer, 2014, pp. 337–347.

[85] L. N. Trefethen and D. Bau, III, *Numerical linear algebra*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.

[86] F. Tröltzsch, *Optimale Steuerung partieller Differentialgleichungen: Theorie, Verfahren und Anwendungen*, Vieweg Verlag, Wiesbaden, 2005.

[87] F. Tröltzsch, *Optimal Control of Partial Differential Equations: Theory, Methods and Applications*, American Mathematical Society, 2010.

[88] M. Ulbrich, *Semismooth Newton Methods for Variational Inequalities and Constrained Optimization Problems*, SIAM Philadelphia, 2011.

[89] M. Ulbrich and S. Ulbrich, *Primal-dual interior-point methods for pde-constrained optimization*, Math. Program., 117 (2009), pp. 435–485.

[90] H. A. Van der Vorst, *BiCGSTAB: A fast and smoothly converging variant of BiCG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 13 (1992), pp. 631–644.

[91] M. Weiser and A. Schiela, *Function space interior point methods for pde constrained optimization*, PAMM, 4 (2004), pp. 43–46.

[92] P. Wesseling, *An introduction to multigrid methods*, Pure and Applied Mathematics (New York), John Wiley & Sons Ltd., Chichester, 1992.

[93] J. C. Ziems and S. Ulbrich, *Adaptive multilevel inexact SQP methods for PDE-constrained optimization*, SIAM J. Optim., 21 (2011), pp. 1–40.

[94] W. Zulehner, *Non-standard norms and robust estimates for saddle point problems*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 536–560.

APPENDIX A
SELECTED PAPERS

A.1 Preconditioning for control constraints

This paper is published as M. Stoll and A. Wathen, ***Preconditioning for partial differential equation constrained optimization with control constraints*, Numer. Lin. Alg. Appl., 19 (2012), pp. 53–71.**

Result from the paper

Table A.1 shows results for $3D$ example with number of iterations for a nonstandard CG method (BPCG) and MINRES for the case of no bound constraints. The iteration numbers for the active set (AS) method with total number of CG iterations are shown and timings for all methods are given in brackets. Robust iteration numbers can be observed for a varied mesh-parameter.

	Unconstrained Iterations		Simple bounds
	BPCG(T)	MINRES(T)	AS(# BPCG/T)
4913	7(0.65)	7(0.62)	2(15/1.51)
35937	7(6.06)	7(5.84)	5(39/35.36)
274625	7(52.01)	9(62.7)	4(32/247.55)
2146689	8(476.47)	11(609.85)	5(43/2652.73)

Table A.1: Results for $3D$ example with number of iterations for the Bramble-Pasciak CG (BPCG) and MINRES for the case of no bound constraints. The iteration numbers for the active set (AS) method with total number of CG iterations are shown and timings for all methods are given in brackets.

NUMERICAL LINEAR ALGEBRA WITH APPLICATIONS
Numer. Linear Algebra Appl. 2012; **19**:53–71
Published online 18 October 2011 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/nla.823

Preconditioning for partial differential equation constrained optimization with control constraints

Martin Stoll [1,*,†] and Andy Wathen [2]

[1] *Oxford Centre for Collaborative Applied Mathematics, Mathematical Institute, 24–29 St Giles', Oxford OX1 3LB, U.K.*
[2] *Numerical Analysis Group, Mathematical Institute, 24–29 St Giles', Oxford OX1 3LB, U.K.*

SUMMARY

Optimal control problems with partial differential equations play an important role in many applications. The inclusion of bound constraints for the control poses a significant additional challenge for optimization methods. In this paper, we propose preconditioners for the saddle point problems that arise when a primal–dual active set method is used. We also show for this method that the same saddle point system can be derived when the method is considered as a semismooth Newton method. In addition, the projected gradient method can be employed to solve optimization problems with simple bounds, and we discuss the efficient solution of the linear systems in question. In the case when an acceleration technique is employed for the projected gradient method, this again yields a semismooth Newton method that is equivalent to the primal–dual active set method. We also consider the Moreau–Yosida regularization method for control constraints and efficient preconditioners for this technique. Numerical results illustrate the competitiveness of these approaches. Copyright © 2011 John Wiley & Sons, Ltd.

Received 20 April 2010; Revised 22 August 2011; Accepted 28 August 2011

KEY WORDS: PDE-constrained optimization; saddle point systems; preconditioning; Newton method; Krylov subspace solver

1. INTRODUCTION

Advances in algorithms and hardware have enabled optimization with constraints given by partial differential equations (PDEs). Problems of this type arise in a variety of applications [1]. Comprehensive introductions to this field now titled PDE-constrained optimization can be found in [1–3]. Problems of this type arising in many applications pose significant challenges to optimization algorithms and numerical methods in general.

In this paper, we will focus on three numerical optimization algorithms: an active set method (Section 3), a projected gradient method (Section 4), and a Moreau–Yosida regularization technique. These are standard methods in the field of practical optimization [4, 5], but within the framework of PDE-constrained optimization, one has to carefully analyze the numerical schemes to solve these problems as the constraining PDEs typically result in very large dimensional discrete systems. In particular, we focus on the linear systems arising in these methods and preconditioners to be employed with iterative methods for their solution. We show in the course of this paper that all three

*Correspondence to: Martin Stoll, Oxford Centre for Collaborative Applied Mathematics, Mathematical Institute, 24–29 St Giles', Oxford OX1 3LB, U.K.
†E-mail: martin.stoll80@gmail.com

methods lead to systems in saddle point form, that is

$$\underbrace{\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix}}_{\mathcal{K}} x = b, \tag{1}$$

where we assume that $A \in \mathbb{R}^{n,n}$ is symmetric and positive definite and $B \in \mathbb{R}^{m,n}$, $m < n$, is a matrix of full rank. Under these assumptions, the linear system given in (1) is well defined and has a unique solution. The system matrix $\mathcal{K}$ is symmetric and indefinite, and a variety of methods exists to solve problems of this type efficiently (see [6] for a survey). In practice, the linear system $\mathcal{K}x = b$ usually is of sufficiently high dimension that iterative solution methods are needed, and it is never solved without the application of a preconditioner $\mathcal{P}$ chosen to enhance the convergence behavior of the iterative method. A variety of preconditioners exists to tackle saddle point problems of the form (1). The aim of this paper is for the different optimization methods to present preconditioners that are tailored toward an efficient solution of the linear system arising from the discretization of an optimal control problem involving a PDE.

The problem we are interested in will be presented in detail in Section 2. Our focus in this paper is to derive efficient preconditioners for the optimal control problems; hence, we introduce all methods from a linear algebra perspective. The area of PDE-constrained optimization has received an enormous amount of attention over the last few years, but only a few contributions for the efficient solution of the corresponding linear systems have been made. We show how for each method the saddle point system can be preconditioned and efficiently solved using a Krylov subspace technique. We also derive bounds for the eigenvalues of the preconditioned matrix in some idealized cases, and the numerical results presented in Section 6 illustrate the competitiveness of this approach.

2. THE PROBLEM

The functional to be minimized over a domain $\Omega \in \mathbb{R}^d$ with $d = 2, 3$ is given by

$$J(y, u) := \frac{1}{2}\mathrm{y} - \bar{\mathrm{y}}^2_{L^2(\Omega)} + \frac{\beta}{2}\|\mathrm{u}\|^2_{L^2(\Omega)}, \tag{2}$$

where $\beta \in \mathbb{R}^+$ is a regularization parameter and $\bar{\mathrm{y}}$ is a given function that represents the desired state. In many practical applications, this function describes design criteria in the underlying application; hence, problems of this type are often called design optimization problems. The state y and the control u are linked via a PDE problem that we shall take throughout this paper to be the Poisson equation:

$$-\triangle \mathrm{y} = \mathrm{u} \text{ in } \Omega. \tag{3}$$

Additionally, we allow for the control to be bounded by so-called *box constraints*

$$\mathrm{u_a}(x) \leqslant \mathrm{u}(x) \leqslant \mathrm{u_b}(x) \text{ a.e in } \Omega, \tag{4}$$

where we assume that $\mathrm{u_a}(x) < \mathrm{u_b}(x)$ a.e in Ω and define

$$U_{\mathrm{ad}} := \left\{\mathrm{u} \in L^2(\Omega) : \mathrm{u_a}(x) \leqslant \mathrm{u}(x) \leqslant \mathrm{u_b}(x) \text{ a.e in } \Omega\right\}$$

to be the set of admissible functions. The presented setup can be summarized in the following optimization system:

$$\begin{cases} \min \frac{1}{2}\|\mathrm{y} - \bar{\mathrm{y}}\|^2_{L^2(\Omega)} + \frac{\beta}{2}\|\mathrm{u}\|^2_{L^2(\Omega)} & \text{s.t.} \\ -\triangle \mathrm{y} = \mathrm{u} & \text{in } \Omega \\ \mathrm{y} = \bar{\mathrm{y}} & \text{on } \Gamma \\ \mathrm{u_a}(x) \leqslant \mathrm{u}(x) \leqslant \mathrm{u_b}(x) & \text{a.e in } \Omega. \end{cases} \tag{5}$$

Problems of this type are well studied, and we refer to [1–3, 3, 7–9] for more details. Note that it is not necessary to restrict y to $\bar{\mathrm{y}}$ on the boundary.

DOI: 10.1002/nla

We want to discuss the discretization of our problem as we will follow a *discretize-then-optimize* approach as we discretize first and then look at the first-order conditions of the discrete problem. One way to discretize the optimization problem (5) is to use the finite element method [10].

Our choice is here **Q1** elements using the deal.II [11] framework; to guarantee that necessary optimality conditions are satisfied, we use lumped mass matrices. Note that using piecewise constant finite elements for the control u and the Lagrange multipliers μ_a and μ_b would be sufficient. Our approach leads to a more convenient notation, but everything also holds for the case when piecewise constants elements are used for the control and the Lagrange multipliers. Now, the resulting discrete optimization problem is given by

$$\begin{cases} \min \frac{1}{2}(y-\bar{y})^T M (y-\bar{y}) + \frac{\beta}{2} u^T M u \quad \text{s.t.} \\ Ky = Mu - d \\ \underline{u}_a \leqslant u \leqslant \overline{u}_b, \end{cases} \tag{6}$$

where K and M represent the stiffness and lumped mass matrices of the appropriate finite element space, respectively. The vector d represents the boundary data and $\underline{u}_a$, $\overline{u}_b$ represent projections onto the finite elements space in an analogous way to u. We want to solve this problem using the Lagrange multiplier approach (cf. [2,3,9]). The Lagrange function is given by

$$\begin{aligned} L(y,u,\lambda) = & \ \tfrac{1}{2}(y-\bar{y})^T M (y-\bar{y}) \\ & + \tfrac{\beta}{2} u^T M u - \lambda^T (Ky - Mu + d), \end{aligned} \tag{7}$$

and the stationarity conditions for the Lagrange function $L(y,u,\lambda)$ are

$$\nabla_y L(y^*,u^*,\lambda^*) = My^* - M\bar{y} - K^T\lambda^* = 0 \tag{8}$$

and

$$\nabla_\lambda L(y^*,u^*,\lambda^*) = -Ky^* + Mu^* - d = 0. \tag{9}$$

The last optimality condition considering the box constraints on the control can be written as

$$(u-u^*)^T \nabla_u L(y^*,u^*,\lambda^*) = (u-u^*)^T(\beta M u^* + M\lambda^*) \geqslant 0 \quad \forall u \in U_{\text{ad}}. \tag{10}$$

Condition (10) follows from the variational inequality

$$F'(u^*)(u-u^*) \geqslant 0 \quad \forall u \in U_{\text{ad}},$$

where $F(u)$ can be obtained by using the state equation to remove y from the objective function and to solely work with the control u. In more detail, we obtain the reduced discretized optimization problem as

$$\min_{U_{\text{ad}}} F(u) = \frac{1}{2}\left(K^{-1}(Mu-d) - \bar{y}\right) M \left(K^{-1}(Mu-d) - \bar{y}\right) + \frac{\beta}{2} u^T M u, \tag{11}$$

where we define the admissible set as $U_{\text{ad}} = \{u \in \mathbb{R}^n : \underline{u}_a \leqslant u \leqslant \overline{u}_b\}$. Note that in the absence of box constraints on the control u, (10) would reduce to $\beta M u^* + M\lambda^* = 0$, and the solution can be found by solving a linear system in saddle point form (see [7] for details). Now, we want to further rewrite the optimality system to get rid of the condition (10).

With U_{ad} as defined previously, we obtain the component-wise expression of u^*

$$(u^*)_i = \begin{cases} = (\overline{u}_b)_i & \text{if } (\beta u^* + \lambda^*)_i < 0 \\ \in U_{\text{ad}} & \text{if } (\beta u^* + \lambda^*)_i = 0 \\ = (\underline{u}_a)_i & \text{if } (\beta u^* + \lambda^*)_i > 0 \end{cases} \tag{12}$$

as M is a lumped mass matrix. From the second equation in (12), we have that

$$u^* = -\frac{\lambda^*}{\beta} \text{ whenever } \beta u^* + \lambda^* = 0. \tag{13}$$

DOI: 10.1002/nla

Relation (12) can be used to define a new Lagrange function by introducing two Lagrange multipliers:

$$\mu_{\mathrm{a}} := \left(\beta u^* + \lambda^*\right)^+ \text{ and } \mu_{\mathrm{b}} := \left(\beta u^* + \lambda^*\right)^-, \tag{14}$$

where

$$(\mu_{\mathrm{a}})_i = \left(\beta u^* + \lambda^*\right)_i$$

whenever $(\beta u^* + \lambda^*)_i$ is positive; otherwise $(\mu_{\mathrm{a}})_i$ will be zero. Similarly,

$$(\mu_{\mathrm{b}})_i = \left|\left(\beta u^* + \lambda^*\right)_i\right|$$

if $(\beta u^* + \lambda^*)_i$ is negative; otherwise $(\mu_{\mathrm{b}})_i$ is zero. This gives the complementary slackness condition

$$\begin{array}{lll} \mu_{\mathrm{a}} \geqslant 0, & \underline{u}_{\mathrm{a}} - u^* \leqslant 0, & \left(\underline{u}_{\mathrm{a}} - u^*\right)^T \mu_{\mathrm{a}} = 0 \\ \mu_{\mathrm{b}} \geqslant 0, & u^* - \overline{u}_{\mathrm{b}} \leqslant 0, & \left(u^* - \overline{u}_{\mathrm{b}}\right)^T \mu_{\mathrm{b}} = 0. \end{array} \tag{15}$$

3. ACTIVE SET METHOD

On the basis of the description in the last section, we introduce numerical methods to solve problem (6). The first is an active set method. Active set methods have a long history in optimization for linear programming in terms of the simplex method [12] or in quadratic programming [13]. The approach we follow was introduced in [14], and we follow its derivation in [8]. For reasons of convenience, we use a new Lagrange multiplier μ instead of μ_{a} and μ_{b} that is defined as follows:

$$\mu := \mu_{\mathrm{a}} - \mu_{\mathrm{b}} = \beta u + \lambda, \tag{16}$$

and we have for the optimal control

$$(u^*)_i = \begin{cases} = (\underline{u}_{\mathrm{a}})_i & \text{if } \mu_i^* > 0 \\ \in U_{\mathrm{ad}} & \text{if } \mu_i^* = 0 \\ = (\overline{u}_{\mathrm{b}})_i & \text{if } \mu_i^* < 0. \end{cases} \tag{17}$$

The first equation in (17) gives that $(u^*)_i = (\underline{u}_{\mathrm{a}})_i$, and hence, $(\mu^*)_i > 0$ that results in $(u^* - \mu^*)_i < (\underline{u}_{\mathrm{a}})_i$. Analogously, the third equation in (17) gives that $(\mu^*)_i < 0$ and $(u^* - \mu^*)_i > (\overline{u}_{\mathrm{b}}))_i$. The second equation in (17) shows that for $(\mu^*)_i = 0$ the relation $(u^* - \mu^*)_i = (u^*)_i$ holds.

Thus, the quantity $u - \mu$ is an indicator whether a constraint is active or inactive; with this, an active set strategy can be implemented. For a general introduction to active set methods, we refer to [4, 5]; in the particular case of a primal–dual active set strategy for PDE-constrained optimization, we recommend [2, 3, 9, 14].

First, we define the active sets as

$$\mathcal{A}_+ = \{i \in \{1, 2, \ldots, N\} : (u - \mu)_i > (\overline{u}_{\mathrm{b}})_i\} \tag{18}$$

$$\mathcal{A}_- = \{i \in \{1, 2, \ldots, N\} : (u - \mu)_i < (\underline{u}_{\mathrm{a}})_i\} \tag{19}$$

$$\mathcal{A}_I = \{1, 2, \ldots, N\} \setminus (\mathcal{A}_+ \cup \mathcal{A}_-)\,. \tag{20}$$

We now introduce the control $u^{(k)}$ at step k of an iterative procedure as the approximation to the optimal solution u^*. In a straightforward fashion, this notation will be used for the state and the adjoint state. The active sets $\mathcal{A}_-^{(k)}$, $\mathcal{A}_+^{(k)}$, and $\mathcal{A}_I^{(k)}$ are defined using $u^{(k-1)}$ and $\mu^{(k-1)}$. The following conditions have to hold in each step of an iterative procedure (see Algorithm 3)

$$My^{(k)} - M\bar{y} - K^T\lambda^{(k)} = 0 \tag{21}$$

$$-Ky^{(k)} + Mu^{(k)} = d \tag{22}$$

DOI: 10.1002/nla

1: Choose initial values for $u^{(0)}$, $y^{(0)}$. $\lambda^{(0)}$ and $\mu^{(0)}$
2: Set the active sets $\mathcal{A}_+^{(0)}$, $\mathcal{A}_-^{(0)}$ and $\mathcal{A}_I^{(0)}$ by using $u^{(0)}$ and $\mu^{(0)}$ in (18), (19) and (20)
3: **for** $k = 1, 2, \ldots$ **do**
4: Solve (21) to (23) on the free variables from the previous iteration ($\mathcal{A}_I^{(k-1)}$)
5: Update $\mu^{(k)}$
6: Set the active sets $\mathcal{A}_+^{(k)}$, $\mathcal{A}_-^{(k)}$ and $\mathcal{A}_I^{(k)}$ by using $u^{(k)}$ and $\mu^{(k)}$ as given in (18), (19) and (20)
7: **if** $\mathcal{A}_+^{(k)} = \mathcal{A}_+^{(k-1)}$, $\mathcal{A}_-^{(k)} = \mathcal{A}_-^{(k-1)}$, and $\mathcal{A}_I^{(k)} = \mathcal{A}_I^{(k-1)}$ **then**
8: STOP (Algorithm converged)
9: **end if**
10: **end for**

Algorithm 1: Active Set algorithm

$$\beta M u^{(k)} + M\lambda^{(k)} - M\mu^{(k)} = 0 \tag{23}$$

$$\mu^{(k)} = 0 \text{ on } \mathcal{A}_I^{(k)} \tag{24}$$

$$u^{(k)} = \underline{u}_{\mathrm{a}} \text{ on } \mathcal{A}_-^{(k)} \tag{25}$$

$$u^{(k)} = \overline{u}_{\mathrm{b}} \text{ on } \mathcal{A}_+^{(k)}. \tag{26}$$

Following a technique given in [8], we partition the control u according to the dimension of the sets $\mathcal{A}_I^{(k)}$, $\mathcal{A}_-^{(k)}$, and $\mathcal{A}_+^{(k)}$ and using the fact that the u is known on the sets $\mathcal{A}_+^{(k)}$ and $\mathcal{A}_-^{(k)}$ to obtain

$$\left[\begin{array}{c|ccc|c} M & 0 & 0 & 0 & -K \\ \hline 0 & \beta M^{\mathcal{A}_I^{(k)},\mathcal{A}_I^{(k)}} & 0 & 0 & M^{\mathcal{A}_I^{(k)},:} \\ 0 & 0 & \beta M^{\mathcal{A}_+^{(k)},\mathcal{A}_+^{(k)}} & 0 & M^{\mathcal{A}_+^{(k)},:} \\ 0 & 0 & 0 & \beta M^{\mathcal{A}_-^{(k)},\mathcal{A}_-^{(k)}} & M^{\mathcal{A}_-^{(k)},:} \\ \hline -K & M^{:,\mathcal{A}_I^{(k)}} & M^{:,\mathcal{A}_+^{(k)}} & M^{:,\mathcal{A}_-^{(k)}} & 0 \end{array}\right] \left[\begin{array}{c} y^{(k)} \\ u^{\mathcal{A}_I^{(k)}} \\ \overline{u}_{\mathrm{b}} \\ \underline{u}_{\mathrm{a}} \\ \lambda^{(k)} \end{array}\right] = \left[\begin{array}{c} M\bar{y} \\ 0 \\ (M\mu)^{\mathcal{A}_+^{(k)}} \\ (M\mu)^{\mathcal{A}_-^{(k)}} \\ d \end{array}\right]. \tag{27}$$

This can now be reduced to the final linear system

$$\left[\begin{array}{ccc} M & 0 & -K \\ 0 & \beta M^{\mathcal{A}_I^{(k)},\mathcal{A}_I^{(k)}} & M^{\mathcal{A}_I^{(k)},:} \\ -K & M^{:,\mathcal{A}_I^{(k)}} & 0 \end{array}\right] \left[\begin{array}{c} y^{(k)} \\ u^{\mathcal{A}_I^{(k)}} \\ \lambda^{(k)} \end{array}\right] = \left[\begin{array}{c} M\bar{y} \\ 0 \\ -M^{:,\mathcal{A}_+^{(k)}}\overline{u}_{\mathrm{b}} - M^{:,\mathcal{A}_-^{(k)}}\underline{u}_{\mathrm{a}} + d \end{array}\right]. \tag{28}$$

Once this system is solved, we can update the Lagrange multipliers associated with the sets $\mathcal{A}_+^{(k)}$ and $\mathcal{A}_-^{(k)}$:

$$\begin{aligned} (M\mu)^{\mathcal{A}_+^{(k)}} &= \beta M^{\mathcal{A}_+^{(k)},\mathcal{A}_+^{(k)}}\overline{u}_{\mathrm{b}} + M^{\mathcal{A}_+^{(k)},:}\lambda^{(k)} \\ (M\mu)^{\mathcal{A}_-^{(k)}} &= \beta M^{\mathcal{A}_-^{(k)},\mathcal{A}_-^{(k)}}\underline{u}_{\mathrm{a}} + M^{\mathcal{A}_-^{(k)},:}\lambda^{(k)} \end{aligned} \tag{29}$$

In [14], it is shown that when the active sets stay unchanged in two consecutive steps, the method has found a local minimum and the algorithm can be terminated.

3.1. Equivalence to a semismooth Newton method

In this section, we want to emphasize the connection of the active set method to a semismooth Newton method as given in [15]. For an introduction to semismooth Newton methods, we refer to [1, 9, 16]. Recall that we introduced the reduced optimization problem for $F(u)$ in (11). To use Newton's method for solving (11), we have to compute the gradient of F, which is given by

$$\nabla F(u) = MK^{-T}MK^{-1}Mu - MK^{-T}MK^{-1}d - MK^{-T}M\bar{y} + \beta Mu$$

DOI: 10.1002/nla

Again, we can use $Ky = Mu - d$ to simplify this and obtain

$$MK^{-T}My - MK^{-T}M\bar{y} + \beta Mu. \tag{30}$$

If we introduce λ as the solution of the adjoint system

$$K^T\lambda = M(y - \bar{y}), \tag{31}$$

the gradient finally becomes

$$\nabla F(u) = M\lambda + \beta Mu. \tag{32}$$

Note the equivalence to the definition of μ in the active set method presented in the previous section. The Lagrange multiplier μ represents the gradient of the function $F(u)$. The Hessian of F can now easily be obtained as

$$\nabla^2 F(u) = MK^{-T}MK^{-1}M + \beta M, \tag{33}$$

which is a symmetric and positive definite matrix.

It is possible to show that the active set method is equivalent to a semismooth Newton method. In [15], Hintermüller *et al.* show that the primal–dual active set method is a semismooth Newton method. On the basis of [1], we want to derive the equivalence of the active set method for the optimality system (5) and a semismooth Newton method solely written in linear algebra terms as this will provide useful information for developing good preconditioners. For more details on nonsmooth Newton methods, we refer to [1, 9, 16–18]. In our case, the optimality condition for $F(u)$ becomes

$$\Phi(u) := P_{[\underline{u}_a, \overline{u}_b]}(u - D(\beta Mu + H'(u))) - u = 0,$$

where D is a diagonal matrix with positive entries and $H'(u) = MK^{-T}MK^{-1}Mu - MK^{-T}M\bar{y}$ (see Theorem 5.2.4 in [19]). Note that the gradient of $F(u)$ is given by $\nabla F(u) = \beta Mu + H'(u)$; with the choice $D = M^{-1}$, we obtain

$$\Phi(u) := P_{[\underline{u}_a, \overline{u}_b]}(u - \beta u - M^{-1}H'(u)) - u = 0.$$

As $\Phi(u)$ now represents a nonsmooth functional, the Newton system becomes

$$M_k s_u^{(k)} = -\Phi(u^{(k-1)}), \tag{34}$$

where the generalized differential is given by

$$M_k = G - \beta G - GK^{-T}MK^{-1}M - I$$

with

$$(G)_{jj} = \begin{cases} 0 & u - \mu \notin (\underline{u}_a, \overline{u}_b) \\ 1 & \text{otherwise} \end{cases}$$

and $\mu = \beta u + M^{-1}H'(u)$ (this is equivalent to (16)). Without loss of generality, we can assume that the variables are ordered such that

$$G = \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix}.$$

Note that solving the system (34) with M_k can be achieved by solving

$$\begin{bmatrix} M & 0 & -K^T \\ 0 & -I + G - \beta G & -G \\ -K & M & 0 \end{bmatrix} \begin{bmatrix} s_y^{(k)} \\ s_u^{(k)} \\ s_\lambda^{(k)} \end{bmatrix} = \begin{bmatrix} 0 \\ -\Phi(u^{(k-1)}) \\ 0 \end{bmatrix}. \tag{35}$$

We could stop here for the implementation of a semismooth Newton method; however, as we want to obtain the implementation of the active set method presented earlier as well as obtain a

DOI: 10.1002/nla

symmetric linear system that can be solved much more efficiently than the one given in (35), we use the definition

$$\begin{aligned}\Phi(u^{(k-1)}) &= P_{[\underline{u}_a,\overline{u}_b]}(u^{(k-1)} - (\beta u^{(k-1)} + M^{-1}H'(u^{(k-1)})) - u^{(k-1)}\\ &= P_{[\underline{u}_a,\overline{u}_b]}(u^{(k-1)} - \mu^{(k-1)}) - u^{(k-1)}\end{aligned}$$

with $\mu^{(k-1)}$ defined as in (16). We now use this equation to rewrite (35) to have

$$\begin{bmatrix} M & 0 & -K^T \\ 0 & -I+G-\beta G & G \\ -K & M & 0 \end{bmatrix}\begin{bmatrix} y^{(k)}-y^{(k-1)} \\ u^{(k)}-u^{(k-1)} \\ \lambda^{(k)}-\lambda^{(k-1)} \end{bmatrix} = \begin{bmatrix} 0 \\ -\Phi(u^{(k-1)}) \\ 0 \end{bmatrix}, \tag{36}$$

which is also equivalent to

$$\begin{aligned}\begin{bmatrix} M & 0 & -K^T \\ 0 & -I+G-\beta G & -G \\ -K & M & 0 \end{bmatrix}\begin{bmatrix} y^{(k)} \\ u^{(k)} \\ \lambda^{(k)} \end{bmatrix} &= \begin{bmatrix} -K^T\lambda^{(k-1)} + My^{(k-1)} \\ -\Phi(u^{(k-1)}) + (-I+G-\beta G)u^{(k-1)} - G\lambda^{(k-1)} \\ -Ky^{(k-1)} + Mu^{(k-1)} \end{bmatrix}\\ &= \begin{bmatrix} M\bar{y} \\ -\Phi(u^{(k-1)}) \\ d \end{bmatrix}.\end{aligned} \tag{37}$$

We now have to take care of the part in (37) that corresponds to the control u. For that we are splitting the control in its parts corresponding to the active sets based in $u^{(k-1)}$ and consider the following three cases:

$$\Phi(u^{(k-1)})\begin{cases} = (\underline{u}_a - u^{(k-1)})_i & \text{for all } i \in \mathcal{A}_-^{(k-1)} \\ = (u^{(k-1)} - \mu^{(k-1)} - u^{(k-1)})_i & \text{for all } i \in \mathcal{A}_I^{(k-1)} \\ = (\overline{u}_b - u^{(k-1)})_i & \text{for all } i \in \mathcal{A}_+^{(k-1)}, \end{cases} \tag{38}$$

where $\mathcal{A}_-^{(k-1)}$, $\mathcal{A}_+^{(k-1)}$, and $\mathcal{A}_I^{(k-1)}$ are defined as in Section 3. For convenience, we neglect the indices of the active sets in the linear systems. We can equivalently split up $-\Phi(u^{(k-1)}) + (-I + G - \beta G)u^{(k-1)} - G\lambda^{(k-1)}$ using (38) and the definition of μ to have

$$-\Phi(u^{(k-1)}) + (-I+G-\beta G)u^{(k-1)} - G\lambda^{(k-1)}\begin{cases} = -\underline{u}_a & \text{for all } i \in \mathcal{A}_-^{(k-1)} \\ = 0 & \text{for all } i \in \mathcal{A}_I^{(k-1)} \\ = -\overline{u}_b & \text{for all } i \in \mathcal{A}_+^{(k-1)}. \end{cases} \tag{39}$$

Putting this together into a linear system now gives

$$\begin{bmatrix} M & 0 & 0 & 0 & -K^T \\ 0 & -I & 0 & 0 & 0 \\ 0 & 0 & -I & 0 & 0 \\ 0 & 0 & 0 & \beta I & G^{\mathcal{A}_I,:} \\ -K & M^{:,\mathcal{A}_+} & M^{:,\mathcal{A}_-} & M^{:,\mathcal{A}_I} & 0 \end{bmatrix}\begin{bmatrix} y^{(k)} \\ u^{(k)}_{\mathcal{A}_+} \\ u^{(k)}_{\mathcal{A}_-} \\ u^{(k)}_{\mathcal{A}_I} \\ \lambda^{(k)} \end{bmatrix} = \begin{bmatrix} M\bar{y} \\ -\overline{u}_b \\ -\underline{u}_a \\ 0 \\ d \end{bmatrix}. \tag{40}$$

We are almost there and now eliminate the rows corresponding to $u^{(k)}_{\mathcal{A}_+}$ and $u^{(k)}_{\mathcal{A}_-}$ and also multiply the row corresponding to $u^{(k)}_{\mathcal{A}_I}$ by $M^{\mathcal{A}_I,\mathcal{A}_I}$, a diagonal matrix, and using the fact that $M^{\mathcal{A}_I,\mathcal{A}_I}(G)_{\mathcal{A}_I,:} = M^{\mathcal{A}_I,:}$ to have

$$\begin{bmatrix} M & 0 & -K^T \\ 0 & \beta M^{\mathcal{A}_I,\mathcal{A}_I} & M^{\mathcal{A}_I,:} \\ -K & M^{:,\mathcal{A}_I} & 0 \end{bmatrix}\begin{bmatrix} y^{(k)} \\ u^{(k)}_{\mathcal{A}_+} \\ u^{(k)}_{\mathcal{A}_-} \\ u^{(k)}_{\mathcal{A}_I} \\ \lambda^{(k)} \end{bmatrix} = \begin{bmatrix} M\bar{y} \\ 0 \\ d - M^{:,\mathcal{A}_+}\overline{u}_b - M^{:,\mathcal{A}_-}\underline{u}_a \end{bmatrix}. \tag{41}$$

DOI: 10.1002/nla

This shows that the active set method is a semismooth Newton method. For the convergence properties of the active set or equivalently semismooth Newton method, we refer to [14, 15]. Note that the semismooth Newton method converges superlinearly if the initial guess is sufficiently close to the solution of the optimality system (see [1, 9] for more details).

3.2. Solving the linear system

The costly part of the active set method presented earlier is the solution of the saddle point system (28) where the system matrix is

$$\mathcal{K} = \begin{bmatrix} M & 0 & -K \\ 0 & \beta M^{\mathcal{A}_I^{(k)},\mathcal{A}_I^{(k)}} & M^{\mathcal{A}_I^{(k)},:} \\ -K & M^{:,\mathcal{A}_I^{(k)}} & 0 \end{bmatrix}. \tag{42}$$

It has to be noted that working with this matrix is not convenient in a practical environment as the matrix $M^{\mathcal{A}_I^{(k)},\mathcal{A}_I^{(k)}}$ will change with every step of the semismooth Newton method.

We will now show that we can implicitly work with the matrix

$$\mathcal{K} = \begin{bmatrix} M & 0 & -K \\ 0 & \beta M & M \\ -K & M & 0 \end{bmatrix} \tag{43}$$

for every step of the Newton method (see also [20]). In a Krylov subspace method, only matrix vector multiplications with the system matrix $\mathcal{K}$ are required; if the right hand side is chosen carefully, then the matrix vector product with (42) in Algorithm 2 can be replaced by the matrix product with (43)

$$\mathcal{K}p = \left[\begin{array}{c|ccc|c} M & 0 & 0 & 0 & -K \\ \hline 0 & \beta M^{\mathcal{A}_I^{(k)},\mathcal{A}_I^{(k)}} & 0 & 0 & M^{\mathcal{A}_I^{(k)},:} \\ 0 & 0 & \beta M^{\mathcal{A}_+^{(k)},\mathcal{A}_+^{(k)}} & 0 & M^{\mathcal{A}_+^{(k)},:} \\ 0 & 0 & 0 & \beta M^{\mathcal{A}_-^{(k)},\mathcal{A}_-^{(k)}} & M^{\mathcal{A}_-^{(k)},:} \\ \hline -K & M^{:,\mathcal{A}_I^{(k)}} & M^{:,\mathcal{A}_+^{(k)}} & M^{:,\mathcal{A}_-^{(k)}} & 0 \end{array}\right] \begin{bmatrix} p_{y^{(k)}} \\ p_{u^{\mathcal{A}_I^{(k)}}} \\ 0 \\ 0 \\ p_{\lambda^{(k)}} \end{bmatrix} \tag{44}$$

and then annihilating the entries in $\mathcal{K}p$ corresponding to the positions of the variables in $\mathcal{A}_+^{(k)}$ and $\mathcal{A}_-^{(k)}$. This means that if the initial residual r_0 has zeros in the positions corresponding to the active sets $\mathcal{A}_+^{(k)}$ and $\mathcal{A}_-^{(k)}$, the matrix vector multiplication will not take any contributions from the matrix blocks associated with these variables, and the annihilation of the additional entries after multiplication in the blocks corresponding to $\mathcal{A}_+^{(k)}$ and $\mathcal{A}_-^{(k)}$ will guarantee that this property holds in all steps of the algorithm. We will now discuss how to solve a linear system with this system matrix efficiently.

The matrix in (43) is symmetric and indefinite, and thus, we could employ MINRES [21] to solve the linear system. For MINRES to be applicable when the matrix $\mathcal{A}$ is preconditioned, we need the preconditioner $\mathcal{P}$ to be symmetric and positive definite. For a general survey of how to precondition saddle point problems, we refer to [6, 10]. In the case when no constraints are imposed on the control u, that is, $M^{\mathcal{A}_I^{(k-1)},\mathcal{A}_I^{(k-1)}} = M$, Rees *et al.* presented a block-diagonal preconditioner that can be used to solve the saddle point system [7]. Recently, Rees and Stoll showed that block triangular preconditioners can be employed for the solution of the linear system with $\mathcal{K}$ in case no constraints are given for the control [22]. In this section, we show that both techniques can efficiently be used to solve the system arising as part of the active set method.

DOI: 10.1002/nla

We consider here two preconditioners: the block-diagonal preconditioner

$$\mathcal{P}_{\mathrm{BD}} = \begin{bmatrix} A_0 & 0 & 0 \\ 0 & A_1 & 0 \\ 0 & 0 & S_0 \end{bmatrix} \tag{45}$$

and the block triangular preconditioner

$$\mathcal{P}_{\mathrm{BT}} = \begin{bmatrix} A_0 & 0 & 0 \\ 0 & A_1 & 0 \\ -K & M & -S_0 \end{bmatrix}. \tag{46}$$

The preconditioner $\mathcal{P}_{\mathrm{BD}}$ is a preconditioner that is typically used within the MINRES framework because it is symmetric and positive definite whenever the blocks A_0, A_1, and S_0 are chosen to be symmetric and positive definite. The block triangular preconditioner $\mathcal{P}_{\mathrm{BT}}$ is typically used when the now nonsymmetric preconditioned matrix $\widehat{\mathcal{K}} = \mathcal{P}_{\mathrm{BT}}^{-1}\mathcal{K}$ is used with a symmetric and positive definite inner product defined by $\langle x, y\rangle_{\mathcal{H}_{\mathrm{BT}}} = x^T \mathcal{H}_{\mathrm{BT}} y$ where

$$\mathcal{H}_{\mathrm{BT}} = \begin{bmatrix} M - A_0 & 0 & 0 \\ 0 & \beta M - A_1 & 0 \\ 0 & 0 & S_0 \end{bmatrix}. \tag{47}$$

It is easily seen that the definition of $\mathcal{H}_{\mathrm{BT}}$ imposes the restriction on the preconditioners A_0 and A_1 that $M - A_0$ and $\beta M - A_1$ both have to be symmetric and positive definite. If this criterion is fulfilled, methods such as the Bramble–Pasciak CG method can be used [23–26]. Assuming that $\mathcal{H}_{\mathrm{BT}}$ defines an inner product, these methods use the fact that the preconditioned matrix $\widehat{\mathcal{K}} = \mathcal{P}_{\mathrm{BT}}^{-1}\mathcal{K}$ is symmetric and positive definite in $\langle ., .\rangle_{\mathcal{H}_{\mathrm{BT}}}$, and hence, a CG method can be used [25, 27, 28] (see Algorithm 2).

We will now motivate the use of the preconditioner by considering an idealized case with $A_0 = M$, $A_1 = \beta M$, and $S_0 = KM^{-1}K^T$ as an approximation to the Schur complement of the matrix $S = KM^{-1}K^T + \beta^{-1}M$. Note that in this case the inner product would be degenerate ($\mathcal{H}$ is singular), but we expect the eigenvalues for a more realistic setup to be close to the ones that we analyze in the following. With this choice, the eigenvalues of $\mathcal{P}^{-1}\mathcal{K}$ can be read off the diagonal blocks of the upper triangular matrix

$$\mathcal{P}_{\mathrm{BT}}^{-1}\mathcal{K} = \begin{bmatrix} I & 0 & -M^{-1}K^T \\ 0 & I & \beta^{-1}I \\ 0 & 0 & I + \beta^{-1}K^{-T}MK^{-1}M \end{bmatrix}. \tag{48}$$

In particular, we have $2n$ eigenvalues at 1, and the remaining eigenvalues satisfy $(I + \beta^{-1}K^{-T}MK^{-1}M)x = \lambda x$. A similarity transformation

$$M^{\frac{1}{2}}(I + \beta^{-1}K^{-T}MK^{-1}M)M^{-\frac{1}{2}} = I + \beta^{-1}M^{\frac{1}{2}}K^{-T}MK^{-1}M^{\frac{1}{2}}$$

Given $x_0 = 0$, set $r_0 = \mathcal{P}^{-1}(b - \mathcal{K}x_0)$ and $p_0 = r_0$
for $k = 0, 1, \ldots$ **do**
  $\alpha = \frac{\langle r_k, r_k\rangle_{\mathcal{H}}}{\langle \mathcal{P}^{-1}\mathcal{K}p_k, p_k\rangle_{\mathcal{H}}}$
  $x_{k+1} = x_k + \alpha p_k$
  $r_{k+1} = r_k - \alpha\mathcal{P}^{-1}\mathcal{K}p_k$
  $\beta = \frac{\langle r_{k+1}, r_{k+1}\rangle_{\mathcal{H}}}{\langle r_k, r_k\rangle_{\mathcal{H}}}$
  $p_{k+1} = r_{k+1} + \beta p_k$
end for

Algorithm 2: Bramble and Pasciak CG

DOI: 10.1002/nla

reveals that the eigenvalues of the last diagonal block are the eigenvalues of the symmetric matrix $I + \beta^{-1} M^{\frac{1}{2}} K^{-T} M K^{-1} M^{\frac{1}{2}}$. With a field of value[‡] analysis, we want to obtain bounds on the eigenvalues of $\mathcal{P}^{-1}\mathcal{K}$. Thus, the upper bound can be obtained from

$$\frac{x^T \beta^{-1} M^{\frac{1}{2}} K^{-T} M K^{-1} M^{\frac{1}{2}} x}{x^T x} = \frac{\beta^{-1} \left(z^T M z\right) \left(x^T M x\right) \left(y^T K^{-T} K^{-1} y\right)}{\left(x^T x\right) \left(y^T y\right) \left(z^T z\right)} \tag{49}$$

with $y = M^{\frac{1}{2}} x$, $z = K^{-1} y$. And as the matrices K and M are symmetric, the eigenvalues of $I + \beta^{-1} K^{-T} M K^{-1} M$ are bound from below by 1 and from above by $1 + \beta^{-1} \left(\lambda_{\max}^{(M)}\right)^2 \lambda_{\max}^{(K^{-T} K^{-1})}$. For this, we need bounds on the eigenvalues of the mass matrix and the stiffness matrix. Proposition 1.29 and Theorem 1.32 in [10] provide these bounds for **Q1** elements, that is, consistent mass matrix and the stiffness matrix. We have $ch^2 \leqslant x^T M x / x^T x \leqslant Ch^2$ for the mass matrix with the constants c and C independent of the mesh size h. For the stiffness matrix K, we obtain $dh^2 \leqslant x^T K x / x^T x \leqslant D$ for mesh-independent constants d and D. We now get that the largest eigenvalue of $K^{-T} K^{-1}$ as $K = K^T$ is bounded by

$$\lambda_{\max}^{(K^{-T} K^{-1})} \leqslant \frac{1}{d^2 h^4}.$$

Hence, the eigenvalues of $\beta^{-1} K^{-T} M K^{-1} M$ are bounded above by

$$\lambda_{\max}^{(\beta^{-1} K^{-T} M K^{-1} M)} \leqslant \frac{\beta^{-1} C^2 h^4}{d^2 h^4} = \frac{\beta^{-1} C^2}{d^2},$$

which is a constant independent of h. This results in the following Theorem.

Theorem 3.1
For the consistent mass matrix M and the stiffness matrix K of a **Q1** finite element space, the eigenvalues of the matrix

$$I + \beta^{-1} K^{-T} M K^{-1} M$$

lie in the interval $\left[1, 1 + (\beta^{-1} C^2 / d^2)\right]$.

This illustrates that for the idealized case, the eigenvalue lies in an interval independent of the mesh size, and good approximations A_0, A_1, and S_0 should result in a similar behavior. Thus, we now have to discuss the choice of these preconditioners.

Note that methods based on a nonstandard inner product given by $\mathcal{H}_{\mathrm{BT}}$ usually suffer from the drawback of appropriately scaling A_0 and A_1. In [29], Wathen and Rees emphasize the fact that for the approximation A_0 to M, a linear operator should be chosen to guarantee that the overall process; that is, the employed Krylov subspace method is a linear operator itself. On the basis of the eigenvalue bounds for the mass matrix given in [30], they identify that the appropriate Chebyshev iteration gives a highly effective linear operator in this context. Building further on these results, Rees and Stoll show in [22] that an appropriate scaling for the preconditioning blocks A_0 and A_1 can easily be obtained when one is interested in solving problems from PDE-constrained optimization. Rees *et al.* [7] propose to employ the Chebyshev semi-iteration for the blocks A_0 and A_1 (see Algorithm 3). For more details on the Chebyshev semi-iterative method, we refer to [29, 31–33]. It has to be noted that the analysis in [22, 29] uses consistent mass matrices; as for lumped mass matrices, preconditioning and scaling are not an issue.

The block S_0 should represent a good approximation to the Schur complement

$$K M^{-1} K^T + \frac{1}{\beta} M. \tag{50}$$

[‡]The field of values of a matrix A is given by $x^T A x / x^T x \; \forall x \neq 0 \in \mathbb{R}^n$.

DOI: 10.1002/nla

Set $D = diag(M)$
Set relaxation parameter ω
Compute $g = \omega D^{-1}\hat{b}$
Set $S = (I - \omega D^{-1}M)$ (this can be used implicitly)
Set $z_{k-1} = 0$ and $z_k = Sz_{k-1} + g$
$c_{k-1} = 2$ and $c_k = \omega$
for $k = 2, \ldots, l$ **do**
  $c_{k+1} = \omega c_k - \frac{1}{4}c_{k-1}$
  $\vartheta_{k+1} = \omega \frac{c_k}{c_{k+1}}$
  $z_{k+1} = \vartheta_{k+1}(Sz_k + g - z_{k-1}) + z_{k-1}$
end for

Algorithm 3: Chebyshev semi-iterative method for a number of l steps

Some choices proposed in Rees *et. al.* [7] are to neglect the term $(1/\beta)M$ in (50) as already carried out in Theorem 3.1 and to only use an S_0 that approximates the term $KM^{-1}K^T$. One typical choice for $S_0 = \hat{K}M^{-1}\hat{K}^T$ is to use a fixed number of algebraic multigrid V cycles from the Trilinos ML package [34] to approximate the stiffness matrix K by $\hat{K}$. It is also important that this is also a linear operator. The cost of MINRES with $\mathcal{P}_{\mathrm{BD}}$ and the Bramble–Pasciak CG with $\mathcal{P}_{\mathrm{BT}}$ are the same when it comes to the number of applications of preconditioners A_0, A_1, and S_0. The Bramble–Pasciak CG is slightly more expensive as it requires one additional multiplication by the $(2,1)$-block of the 2×2 saddle point matrix [20, 35]. A comparison of these two methods for the unconstrained case is given in [22]. In addition, we want to mention the fact that for the evaluation of the inner products with $\mathcal{H}_{\mathrm{BT}}$ the preconditioners never need to be known explicitly; that is, the application of the inverse or rather the solution of a linear system with $\mathcal{P}$ as coefficient matrix is all that is required in practice [23, 25].

4. THE PROJECTED GRADIENT METHOD

In [4, Section 16.6], Nocedal and Wright present a projected gradient method to solve a quadratic optimization problem with box constraints. Similar approaches using a projection of the gradient are also described in [2, 3, 19, 36]. We focus on the approach used by Nocedal and Wright in more detail and will describe different variants on how to choose the search direction.

4.1. Steepest descent direction

In (11), we identified the problem of minimizing $J(y,u)$ with the problem of minimizing $F(u)$ where y and u are linked via the state equation. Hence, the minimization problem $\min J(u,y)$ can be identified with the discrete optimality system

$$\begin{cases} \min F(u) & \text{s.t.} \\ \underline{u}_{\mathrm{a}} \leqslant u \leqslant \overline{u}_{\mathrm{b}} \end{cases}$$

(see also (11)). The projected gradient method takes steps toward the steepest descent direction $p = -\nabla F(u^{(k-1)})$ and maintains the feasibility of the iterate at step k by projecting the Cauchy point onto the box given by U_{ad}. We explain this procedure here step by step. To calculate the gradient $\nabla F(u^{(k-1)})$, we have to check the definition of $F(u)$ in more detail. The gradient of F is given by

$$\nabla F(u) = M\lambda + \beta M u \tag{51}$$

(cf. (32)). By introducing indices k indicating the step k of an algorithmic procedure, we can compute the gradient at every iteration once the forward Poisson equation

$$Ky^{(k)} = Mu^{(k-1)} - d \tag{52}$$

DOI: 10.1002/nla

is solved and also the solution to the adjoint equation given by

$$K^T \lambda^{(k)} = M(y^{(k)} - \bar{y}) \tag{53}$$

is computed.

On the basis of the previous observations, we can now compute $u^{(k)}$ by the following process using projected gradients. The new iterate $u^{(k)}$ at step k is given by the projection of

$$u^{(k-1)} - \alpha p^{(k-1)}$$

onto U_{ad} where α denotes the step size and $p^{(k-1)}$ represents the gradient at step k. Hence, the feasibility of the next iterate $u^{(k)}$ is guaranteed. Following Section 16.6 in [4], the optimal step size α can be explicitly determined.

For a stopping criterion for this method, we refer the interested reader to [19, Section 5.4]. The results shown in Section 6 were computed by simply using the difference between two consecutive controls as an indicator for convergence. This will be sufficient here as we want to improve on this method in the next Section and will see there that a stopping criterion is given naturally.

4.2. Scaled and Newton directions

It is well known that when the steepest descent method is used to solve linear systems, the convergence can be very slow because it is possible to take steps into previously chosen search directions. In the case of a linear system, the Conjugate Gradient method introduced in [37] is an alternative that guarantees for the search directions to be orthogonal with respect to the positive definite system matrix.

Here, we now want to look at an approach where the steepest descent direction is scaled by a matrix L (see [38]). In more detail, instead of moving into the direction $-p^{(k-1)}$ where $p^{(k-1)}$ is the gradient of F, we move into the direction $-Lp^{(k-1)}$. In [38], Bertsekas requires the matrix L to be positive definite and diagonal with respect to the active sets $\mathcal{A}_-$ and $\mathcal{A}_+$; that is, L is diagonal for every index in $\mathcal{A}_-$ and $\mathcal{A}_+$, and positive definite otherwise.

In our case, we will choose L to be a reduced Hessian (see [19] for more information). In more detail, the search direction $p_h^{(k-1)}$ is the solution to the linear system

$$R(u^{(k-1)}, \nabla^2 F(u^{(k-1)}))\, p_h^{(k-1)} = -\nabla F(u^{(k-1)})$$

where

$$R(u^{(k-1)}, \nabla^2 F(u^{(k-1)})) = \begin{cases} \delta_{ij} & \text{if } i \in \mathcal{A}_+ \cup \mathcal{A}_- \text{ or } j \in \mathcal{A}_- \cup \mathcal{A}_- \\ (\nabla^2 F(u^{(k-1)}))_{ij} & \text{otherwise} \end{cases}$$

and $\nabla^2 F(u^{(k-1)})$ is the Hessian as derived in (33). The sets $\mathcal{A}_-$ and $\mathcal{A}_+$ are defined as follows:

$$\mathcal{A}_+ = \left\{ i : (\overline{u}_{\text{b}})_i - (u^{(k-1)})_i < 0 \right\} \text{ and } \mathcal{A}_- = \left\{ i : (u^{(k-1)})_i - (\underline{u}_{\text{a}})_i < 0 \right\}.$$

Equipped with this new $p_h^{(k-1)}$, we can employ the same technique that we already used for the projection of the steepest descent direction onto the box defined by U_{ad}.

For reduced Hessians $R(u^{(k-1)}, \nabla^2 F(u^{(k-1)}))$ that are uniformly positive definite, a global convergence theorem for the scaled steepest descent direction is given by Theorem 5.5.2 in [19]. Whenever the reduced Hessian is symmetric and positive definite, which in our case is trivially fulfilled, the aforementioned procedure can be modified. In more detail, we define the reduced Hessian (cf. [19]) by

$$\nabla_R^2 F(u) = \begin{cases} \delta_{ij} & \text{if } i \in \mathcal{A}_+ \cup \mathcal{A}_- \text{ or } j \in \mathcal{A}_- \cup \mathcal{A}_- \\ \left(\nabla^2 F(u)\right)_{ij} & \text{otherwise.} \end{cases} \tag{54}$$

DOI: 10.1002/nla

We then have to solve the following equation

$$\nabla^2_R F(u)\, p_h^{(k-1)} = -\nabla F(u^{(k-1)})$$

using that $\nabla^2 F(u) = MK^{-T}MK^{-1}M + \beta M$. Without loss of generality, we assume that $\nabla^2_R F(u)$ can be written as

$$\nabla^2_R F(u) = \begin{bmatrix} I & 0 \\ 0 & \Pi^T \nabla^2 F(u) \Pi \end{bmatrix},$$

where Π is a projection consisting of columns of the identity corresponding to entries in $\mathcal{A}_I$. We look at the block $\Pi^T \nabla^2 F(u)\Pi$ in more detail using MATLAB notation:

$$\begin{aligned} \Pi^T \nabla^2 F(u) \Pi = &\ \Pi^T M K^{-T} M K^{-1} M \Pi + \beta \Pi^T M \Pi \\ = &\ M^{\mathcal{A}_I,:} K^{-T} M K^{-1} M^{:,\mathcal{A}_I} + \beta M^{\mathcal{A}_I,\mathcal{A}_I} \end{aligned} \tag{55}$$

The part corresponding to free variables u_I in the linear system $\nabla^2_R F(u)\, p_h^{(k-1)} = -\nabla F(u^{(k-1)})$ can now be written as

$$M^{\mathcal{A}_I,:} K^{-T} M K^{-1} M^{:,\mathcal{A}_I} + \beta M^{\mathcal{A}_I,\mathcal{A}_I} p_{u_I}^{(k)} = -\mu_{\mathcal{A}_I}^{(k-1)} \tag{56}$$

using the definition that $\mu = \nabla F(u) = M\lambda + \beta M u$. We can then, similar to before, build a corresponding saddle point system

$$\begin{bmatrix} M & 0 & -K^T \\ 0 & \beta M^{\mathcal{A}_I,\mathcal{A}_I} & M^{\mathcal{A}_I,:} \\ -K & M^{:,\mathcal{A}_I} & 0 \end{bmatrix} \begin{bmatrix} s_y^{(k)} \\ s_{u_I}^{(k)} \\ s_\lambda^{(k)} \end{bmatrix} = \begin{bmatrix} 0 \\ -\mu_{\mathcal{A}_I}^{(k-1)} \\ 0 \end{bmatrix}.$$

The last system can now equivalently be rewritten as

$$\begin{bmatrix} M & 0 & -K^T \\ 0 & \beta M^{\mathcal{A}_I,\mathcal{A}_I} & M^{\mathcal{A}_I,:} \\ -K & M^{:,\mathcal{A}_I} & 0 \end{bmatrix} \begin{bmatrix} y^{(k)} \\ u_I^{(k)} \\ \lambda^{(k)} \end{bmatrix} = \begin{bmatrix} M\bar{y} \\ -\mu_{\mathcal{A}_I}^{(k-1)} \\ d - M^{:,\mathcal{A}_+} u_{\mathcal{A}_+}^{(k-1)} - M^{:,\mathcal{A}_-} u_{\mathcal{A}_-}^{(k-1)} \end{bmatrix},$$

which is similar to the linear system at the heart of the active set method introduced in Section 3 (see (28)), and for a critical point, we obtain $\mu_{\mathcal{A}_I}^{(k-1)} = 0$ [19]. This shows that the active set method and the projected gradient method with Newton acceleration are related depending on the choice of the active sets and the step size in the projected gradient method. This will be when the projected gradient algorithm diverges from the implementation described in Algorithm 3. The superlinear convergence for the projected gradient method can be found in [38, Proposition 4.]. More details can also be found in [39].

5. MOREAU–YOSIDA FOR CONTROL CONSTRAINED PROBLEMS

The Moreau–Yosida regularization (see [40, 41] and the references mentioned therein) is a popular technique for the case when state constraints are present. As discussed in [41], this case provides challenging problems from the linear algebra point of view, and we refer to [42] for preconditioning strategies.

To complete the discussion of possible methods for control constraint problems, we briefly introduce the Moreau–Yosida regularization for control constraints and discuss the arising linear systems. The problem of minimizing (2) when there are bound constraints on the control using the Moreau–Yosida penalty function is expressed as minimization of

$$J(y,u) := \frac{1}{2}\|\mathrm{y} - \bar{\mathrm{y}}\|^2 + \frac{\beta}{2}\|\mathrm{u}\|^2 + \frac{1}{2\varepsilon}\|\max\{0, \mathrm{u} - u_\mathrm{b}\}\|^2 + \frac{1}{2\varepsilon}\|\min\{0, \mathrm{u} - u_\mathrm{a}\}\|^2. \tag{57}$$

DOI: 10.1002/nla

The Moreau–Yosida approach then leads to the following linear system [41]:

$$\begin{bmatrix} M & 0 & -K^T \\ 0 & \beta M + \varepsilon^{-1} G_{\mathcal{A}} M G_{\mathcal{A}} & M \\ -K & M & 0 \end{bmatrix} \begin{bmatrix} y \\ u \\ \lambda \end{bmatrix} = \begin{bmatrix} M\bar{y} \\ \varepsilon^{-1}\left(G_{\mathcal{A}_+} M G_{\mathcal{A}_+} \overline{u}_{\mathrm{b}} + G_{\mathcal{A}_-} M G_{\mathcal{A}_-} \underline{u}_{\mathrm{a}}\right) \\ d \end{bmatrix}, \tag{58}$$

where the sets $\mathcal{A}_+ = \{i : u_i > (\overline{u}_{\mathrm{b}})_i\}$, $\mathcal{A}_- \{i : u_i < (\underline{u}_{\mathrm{a}})_i\}$ are the active sets associated with the bound constraints on the control u in a similar way to before (we have $\mathcal{A} = \mathcal{A}_+ \cup \mathcal{A}_-$), and G is a matrix variant of the characteristic function for the corresponding sets (see also Section 3.1).

Our focus is again on the efficient solution of the linear systems in (58), which is of saddle point type. Note that the block $\mathrm{blkdiag}(M, \beta M + \varepsilon^{-1} G_{\mathcal{A}_I} M G_{\mathcal{A}_I})$ is symmetric and positive definite as we have a mass matrix on the one hand and a mass matrix plus a submatrix of a mass matrix on the other. Again, considering an idealized preconditioner, we obtain

$$\mathcal{P} = \begin{bmatrix} M & 0 & 0 \\ 0 & L & 0 \\ -K & M & -S_0 \end{bmatrix},$$

where $L = \beta M + \varepsilon^{-1} G_{\mathcal{A}_I} M G_{\mathcal{A}_I}$. The preconditioned matrix is given by

$$\begin{aligned} \mathcal{P}^{-1}\mathcal{K} = &\begin{bmatrix} M^{-1} & 0 & 0 \\ 0 & L^{-1} & 0 \\ -S_0^{-1} K M^{-1} & S_0^{-1} M L^{-1} & -S_0^{-1} \end{bmatrix} \begin{bmatrix} M & 0 & -K^T \\ 0 & L & M \\ -K & M & 0 \end{bmatrix} \\ &\begin{bmatrix} I & 0 & -M^{-1}K^T \\ 0 & I & L^{-1} I \\ 0 & 0 & S_0^{-1}(K M^{-1} K^T + M L^{-1} M) \end{bmatrix}, \end{aligned} \tag{59}$$

which shows that it has $2n$ eigenvalues at 1 and n eigenvalues are given by the eigenvalues of $S_0^{-1}(K M^{-1} K^T + M L^{-1} M)$. Assuming $S_0 = K M^{-1} K^T$, this simplifies to

$$I + K^{-T} M K^{-1} M L^{-1} M.$$

We again use a similarity transformation

$$L^{-\frac{1}{2}} M (I + K^{-T} M K^{-1} M L^{-1} M) M^{-1} L^{\frac{1}{2}} = I + L^{-\frac{1}{2}} M K^{-T} M K^{-1} M L^{-\frac{1}{2}}.$$

The eigenvalues of this symmetric matrix can now be estimated in a similar way to Theorem 3.1.

$$\frac{x^T L^{-\frac{1}{2}} M K^{-T} M K^{-1} M L^{-\frac{1}{2}} x}{x^T x} = \frac{(x^T L^{-1} x)(y^T M^2 y)(z^T K^{-T} K^{-1} z)(w^T M w)}{(x^T x)(y^T y)(z^T z)(w^T w)(y^T y)} \tag{60}$$

with $y = L^{-1/2} x$, $z = My$, $w = K^{-1} z$. From (60), we see that we can use eigenvalue bounds from Section 3.2 for both mass matrix and stiffness matrix. Note the only term that has not been analyzed before is given by $x^T L^{-1} x = x^T \left(\beta M + \varepsilon^{-1} G_{\mathcal{A}} M G_{\mathcal{A}}\right)^{-1} x$. We will start analyzing $x^T L x / x^T x$ to have bounds for the eigenvalues of L and use these to establish bounds for L^{-1}. With $ch^2 \leqslant x^T M x / x^T x \leqslant C h^2$, we see that

$$\beta c h^2 \leqslant \frac{x^T (\beta M + \varepsilon^{-1} G_{\mathcal{A}} M G_{\mathcal{A}}) x}{x^T x} \leqslant \left(\beta + \varepsilon^{-1}\right) C h^2.$$

Now this finally gives for L^{-1}

$$\frac{x^T (\beta M + \varepsilon^{-1} G_{\mathcal{A}} M G_{\mathcal{A}})^{-1} x}{x^T x} \leqslant \frac{1}{\beta c h^2}.$$

DOI: 10.1002/nla

The upper bound for the eigenvalues can now be established as

$$\lambda_{\max}^{(K^{-T}MK^{-1}ML^{-1}M)} \leqslant \frac{C^3}{\beta c d^2}. \tag{61}$$

We have thus proven the following Theorem.

Theorem 5.1
For the consistent mass matrix M and the stiffness matrix K, the eigenvalues of the matrix

$$I + K^{-T}MK^{-1}ML^{-1}M$$

lie in the interval $\left[1, 1 + (C^3\beta c d^2)\right]$.

We remark that the bounds are similar to the ones obtained in Theorem 3.1 and again the interval depends on the parameter β. We will compare this approach with the active set method from Section 3 in Section 6.

6. NUMERICAL EXPERIMENTS

In this section, we will illustrate the effectiveness of the presented methods for a simple problem.

6.1. Setup

We illustrate our method using the following example, which is Example 5.2 in [7]. Let $\Omega = [0,1]^m$, where $m = 2, 3$, and consider the problem

$$\min_{\mathrm{y,u}} \frac{1}{2}\|\mathrm{y} - \bar{\mathrm{y}}\|^2_{L_2(\Omega)} + \frac{\beta}{2}\|\mathrm{u}\|^2_{L_2(\Omega)}$$

$$\text{s.t.} \qquad -\nabla^2 \mathrm{y} = \mathrm{u} \quad \text{in } \Omega \tag{62}$$

$$\mathrm{y} = \bar{\mathrm{y}} \quad \text{on } \partial\Omega \tag{63}$$

where

$$\bar{\mathrm{y}} = \begin{cases} \exp\left(-64\left((x_1 - \frac{1}{2})^2 + (x_2 - \frac{1}{2})^2\right)\right) & \text{if } (x_1, x_2) \in [0,1]^2 \\ \exp\left(-64\left((x_1 - \frac{1}{2})^2 + (x_2 - \frac{1}{2})^2 + (x_3 - \frac{1}{2})^2\right)\right) & \text{if } (x_1, x_2, x_3) \in [0,1]^3. \end{cases}$$

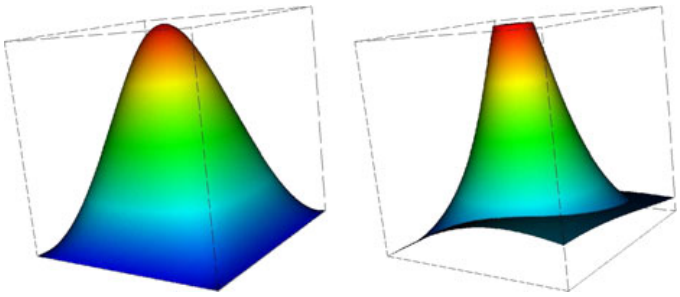
That is, $\bar{\mathrm{y}}$ is Gaussian with peak at unit height at the center of the unit cube. Figure 1 shows the state and the control for the optimal control problem without control constraints with $\beta = 1e-2$.

For the active set method, Bergounioux *et al.* [14] use different initial setups to start the method. We only introduce the setup that proved best for the examples analyzed in [14], that is,

$$\begin{cases} u^{(0)} = \overline{u}_{\mathrm{b}} \\ Ky^{(0)} = Mu^{(0)} \\ K^T\lambda^{(0)} = My^{(0)} - M\bar{y} \\ \mu^{(0)} = \beta M u^{(0)} + M\lambda^{(0)}. \end{cases} \tag{64}$$

We do not show any results for the projected gradients method without scaling as this is known to converge slowly and was also observed when implemented. Our goal is to demonstrate the efficiency of the semismooth Newton method in combination with efficient preconditioning strategies as will be shown now for various examples.

DOI: 10.1002/nla



(a) State (b) Control

Figure 1. State and control for 2D case.

Table I. Results for 2D example with number of iterations for the Bramble–Pasciak CG and MINRES for the case of no bound constraints.

	Unconstrained iterations		Simple bounds
	BPCG(T)	MINRES(T)	AS(#BPCG/T)
289	8(0.02)	9(0.02)	3(24/0.08)
1089	8(0.09)	9(0.08)	5(38/0.42)
4225	8(0.32)	11(0.4)	4(36/1.54)
16641	10(1.84)	13(2.24)	3(33/6.36)
66049	11(9.55)	15(12.32)	5(53/48.56)
263169	13(47.49)	19(65.97)	4(51/193.54)
1050625	17(251.44)	25(352.3)	4(69/1038.97)

The iteration numbers for the active set (AS) method with total number of CG iterations are shown, and timings for all methods are given in brackets. BPCG, Bramble–Pasciak CG.

6.2. *Results*

6.2.1. Semismooth Newton method. We now want to present the results for the semismooth Newton method to illustrate its superiority over the projected gradient method with steepest descent direction presented in Section 4.1. The lower bound is given by

$$u_{\mathrm{a}} = \begin{cases} 0.5x_1 \exp\left(-x_1^2 - x_2^2\right)\big) & \text{in 2D} \\ 0.1x_1 \exp\left(-x_1^2 - x_2^2 - x_3^2\right)\big) & \text{in 3D.} \end{cases} \tag{65}$$

The upper bound is defined as

$$u_{\mathrm{b}} = \begin{cases} 1 & \text{in 2D} \\ 0.5 & \text{in 3D.} \end{cases} \tag{66}$$

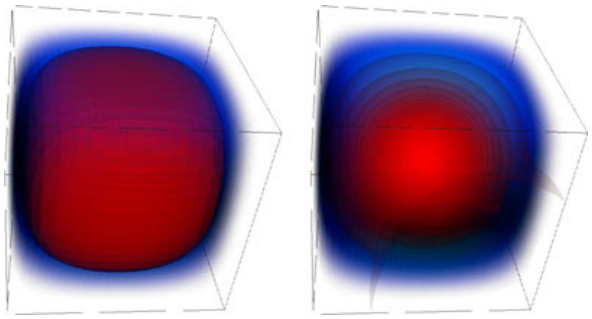
Table I shows the results for the computations performed in two dimensions, and Figure 1 shows the control and state for this setup. In Table II, we can see the results for the computations in three space dimensions, and the corresponding state and control are shown in Figure 2. For the results shown in this section, we used $\beta = 1e-2$ and the matrices are formed on a $2^N \times 2^N$ grid. We compare the unconstrained problem and the constrained problem with active set method. Each stiffness matrix K is here approximated by two V cycles of the algebraic multigrid method. The tolerance for MINRES as well as the Bramble–Pasciak CG is set to 10^{-6} for the relative residual given in the 2-norm. The results given in Tables I and II illustrate the mesh-independent performance of both MINRES and Bramble–Pasciak CG for the unconstrained problem. Note that the Bramble–Pasciak method needs less iterations than MINRES. The active set (semismooth Newton) method shows mesh-independent convergence for the problems considered here.

DOI: 10.1002/nla

Table II. Results for 3D example with number of iterations for the Bramble–Pasciak CG and MINRES for the case of no bound constraints.

	Unconstrained iterations		Simple bounds
	BPCG(T)	MINRES(T)	AS(# BPCG/T)
125	7(0.01)	9(0.01)	7(48/0.08)
729	7(0.08)	7(0.07)	2(14/0.18)
4913	7(0.65)	7(0.62)	2(15/1.51)
35937	7(6.06)	7(5.84)	5(39/35.36)
274625	7(52.01)	9(62.7)	4(32/247.55)
2146689	8(476.47)	11(609.85)	5(43/2652.73)

The iteration numbers for the active set (AS) method with total number of CG iterations are shown and timings for all methods are given in brackets. BPCG, Bramble–Pasciak CG.



(a) State (b) Control

Figure 2. State and control for 3D case.

6.2.2. Moreau–Yosida regularized Newton method. We finally show results that indicate the performance of the Moreau–Yosida regularized method described in Section 5 as a competitor to the primal–dual active set method. We here restrict ourselves to the $2D$ case with the following setup:

$$u_a = 0 \text{ and } u_b = \sin(2\pi x_1 x_2) \tag{67}$$

Table III shows results for $\varepsilon = 1e-6$ and $\beta = 1e-2$ for both the active set method and the Moreau–Yosida approach. As can be seen, the results for the Moreau–Yosida approach and the active set method are nearly indistinguishable. Figure 3 shows the state and control for the setup used here.

7. CONCLUSIONS

In this paper. we presented efficient preconditioning strategies that can be employed when a PDE-constrained optimization problem has to be solved. Our main focus was the solution of linear systems in saddle point form that arise when bound constraints for the control are introduced. We presented results for different preconditioners and considered different optimization algorithms that are well suited for problems in optimal control with PDEs subject to control constraints.

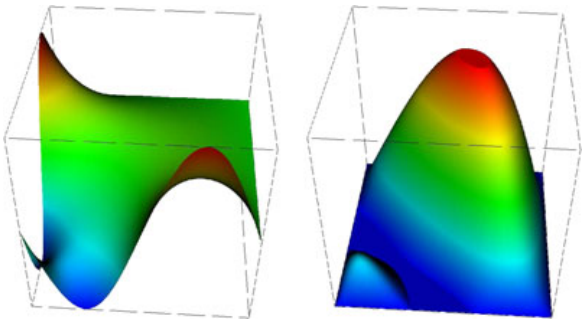
We first considered a primal–dual active set method and derived the same linear system from its interpretation as a semismooth Newton method. On the other hand, we looked at a projected gradient method with steepest descent direction. Because the results for this method were not very encouraging, we considered a Newton acceleration and were able to show that this leads to the same semismooth Newton method that is represented by the active set method. For completeness, we also looked at the Moreau–Yosida approach to the control constraint case and found very similar

DOI: 10.1002/nla

Table III. Results for Moreau–Yosida and PDAS with $\varepsilon = 1e-8$ and $\beta = 1e-2$.

n	MY(# BPCG)	PDAS(# BPCG)	$\|u_{\text{MY}} - u_{\text{PDAS}}\|_2$
289	4(26)	2(14)	$2.862e-11$
1089	4(26)	2(14)	$5.795e-11$
4225	4(28)	3(22)	$5.884e-12$
16641	3(20)	3(20)	$1.146e-11$
66049	3(22)	6(44)	$1.387e-09$
263169	3(23)	3(23)	$3.516e-11$
1050625	3(27)	3(27)	$1.398e-10$

Shown are the number of iterations plus the total number of CG solves. MY, Moreau–Yosida; PDAS, primal–dual active set; BPCG, Bramble–Pasciak CG.



(a) State (b) Control

Figure 3. State and control for 2D case from the Moreau–Yosida regularized method.

results to the active set method. All numerical results given indicate the competitiveness of these approaches.

ACKNOWLEDGEMENTS

The first author would like to thank Tyrone Rees and Nick Gould for sharing their knowledge. The authors would also like to thank the anonymous referee for helping to improve this publication. This publication is partially based on work supported by Award No. KUK-C1-013-04, made by King Abdullah University of Science and Technology (KAUST).

REFERENCES

1. Hinze M, Pinnau R, Ulbrich M, Ulbrich S. *Optimization with PDE constraints.* Mathematical Modelling: Theory and Applications. Springer-Verlag: New York, 2009.
2. Tröltzsch F. *Optimale Steuerung partieller Differentialgleichungen: Theorie, Verfahren und Anwendungen*. Vieweg Verlag: Wiesbaden, 2005.
3. Tröltzsch F. *Optimal Control of Partial Differential Equations: Theory, Methods and Applications*. American Mathematical Society: Providence, Rhode Island, 2010.
4. Nocedal J, Wright SJ. *Numerical optimization.* Springer Series in Operations Research. Springer-Verlag: New York, 1999.
5. Gill PE, Murray W, Wright MH. *Practical optimization*. Academic Press Inc. [Harcourt Brace Jovanovich Publishers]: London, 1981.
6. Benzi M, Golub GH, Liesen J. Numerical solution of saddle point problems. *Acta Numerica* 2005; **14**:1–137.
7. Rees T, Dollar HS, Wathen AJ. Optimal solvers for PDE-constrained optimization. *SIAM Journal on Scientific Computing* 2010; **32**(1):271–298. DOI: http://dx.doi.org/10.1137/080727154.
8. Engel M, Griebel M. A multigrid method for constrained optimal control problems. *Journal of Computational and Applied Mathematics* 2011; **235**(15):4368–4388. DOI: 10.1016/j.cam.2011.04.002.
9. Ito K, Kunisch K. *Lagrange multiplier approach to variational problems and applications, Advances in Design and Control*, Vol. 15. Society for Industrial and Applied Mathematics (SIAM): Philadelphia, PA, 2008.

DOI: 10.1002/nla

10. Elman HC, Silvester DJ, Wathen AJ. *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics.* Numerical Mathematics and Scientific Computation. Oxford University Press: New York, 2005.
11. Bangerth W, Hartmann R, Kanschat G. deal.II—a general-purpose object-oriented finite element library. *ACM Transactions on Mathematical Software* 2007; **33**(4):Art. 24, 27.
12. Dantzig GB. *Linear programming and extensions*. Princeton University Press: Princeton, N.J., 1963.
13. Fletcher R. *Practical methods of optimization.* 2nd edn. A Wiley-Interscience Publication, John Wiley & Sons Ltd.: Chichester, 1987.
14. Bergounioux M, Ito K, Kunisch K. Primal–dual strategy for constrained optimal control problems. *SIAM Journal on Control and Optimization* 1999; **37**(4):1176–1194.
15. Hintermüller M, Ito K, Kunisch K. The primal–dual active set strategy as a semismooth Newton method. *SIAM Journal on Optimization* 2002; **13**(3):865–888.
16. Qi LQ, Sun J. A nonsmooth version of Newton's method. *Mathematical Programming* 1993; **58**(3, Ser. A):353–367.
17. Clarke FH. *Optimization and nonsmooth analysis.* Canadian Mathematical Society Series of Monographs and Advanced Texts. A Wiley-Interscience Publication, John Wiley & Sons Inc.: New York, 1983.
18. Qi LQ. Convergence analysis of some algorithms for solving nonsmooth equations. *Mathematics of Operations Research* 1993; **18**(1):227–244.
19. Kelley CT. *Iterative methods for optimization, Frontiers in Applied Mathematics*, Vol. 18. Society for Industrial and Applied Mathematics (SIAM): Philadelphia, PA, 1999.
20. Rees T, Stoll M, Wathen A. All-at-once preconditioners for PDE-constrained optimization. *Kybernetika* 2010; **46**:341–360.
21. Paige CC, Saunders MA. Solutions of sparse indefinite systems of linear equations. *SIAM Journal on Numerical Analysis* 1975; **12**(4):617–629.
22. Rees T, Stoll M. Block-triangular preconditioners for PDE-constrained optimization. *Numerical Linear Algebra with Applications* 2010; **17**(6):977–996. DOI: 10.1002/nla.693.
23. Bramble JH, Pasciak JE. A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems. *Mathematics of Computation* 1988; **50**(181):1–17.
24. Stoll M, Wathen A. Combination preconditioning and the Bramble–Pasciak$^+$ preconditioner. *SIAM Journal on Matrix Analysis and Applications* 2008; **30**(2):582–608. DOI: 10.1137/070688961.
25. Stoll M. Solving linear systems using the adjoint. *PhD Thesis*, University of Oxford, 2009.
26. Simoncini V. Block triangular preconditioners for symmetric saddle-point problems. *Applied Numerical Mathematics* 2004; **49**(1):63–80.
27. Ashby SF, Holst MJ, Manteuffel TA, Saylor PE. The role of the inner product in stopping criteria for conjugate gradient iterations. *BIT* 2001; **41**(1):26–52.
28. Ashby S, Manteuffel T, Saylor P. A taxonomy for conjugate gradient methods. *SIAM Journal on Numerical Analysis* 1990; **27**(6):1542–1568.
29. Wathen AJ, Rees T. Chebyshev semi-iteration in preconditioning for problems including the mass matrix. *Electronic Transactions in Numerical Analysis* 2008; **34**:125–135.
30. Wathen AJ. Realistic eigenvalue bounds for the Galerkin mass matrix. *IMA Journal of Numerical Analysis* 1987; **7**(4):449–457.
31. Varga RS. *Matrix Iterative Analysis*. Prentice-Hall Inc.: Englewood Cliffs, N.J., 1962.
32. Golub GH, Varga RS. Chebyshev semi-iterative methods, successive over-relaxation iterative methods, and second order Richardson iterative methods. I. *Numerische Mathematik* 1961; **3**:147–156.
33. Golub GH, Varga RS. Chebyshev semi-iterative methods, successive over-relaxation iterative methods, and second order Richardson iterative methods. II. *Numerische Mathematik* 1961; **3**:157–168.
34. Heroux M, Bartlett R, Hoekstra VHR, Hu J, Kolda T, Lehoucq R, Long K, Pawlowski R, Phipps E, Salinger A. *et al*. An overview of Trilinos. *Technical Report SAND2003-2927*, Sandia National Laboratories, 2003.
35. Elman HC. Multigrid and Krylov subspace methods for the discrete Stokes equations. In *Seventh Copper Mountain Conference on Multigrid Methods*, Vol. CP 3339, Melson ND, Manteuffel TA, McCormick SF, Douglas CC (eds). NASA: Hampton, VA, 1996; 283–299.
36. Birgin EG, Martínez JM. Large-scale active-set box-constrained optimization method with spectral projected gradients. *Computational Optimization and Applications* 2002; **23**(1):101–125.
37. Hestenes MR, Stiefel E. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards* 1952; **49**:409–436.
38. Bertsekas DP. Projected Newton methods for optimization problems with simple constraints. *SIAM Journal on Control and Optimization* 1982; **20**(2):221–246.
39. Bertsekas D, Hager W, Mangasarian O. *Nonlinear programming*. Athena Scientific: Belmont, MA, 1999.
40. Ito K, Kunisch K. Semi-smooth Newton methods for state-constrained optimal control problems. *Systems & Control Letters* 2003; **50**(3):221–228. DOI: 10.1016/S0167-6911(03)00156-7.
41. Herzog R, Sachs EW. Preconditioned conjugate gradient method for optimal control problems with control and state constraints. *SIAM Journal on Matrix Analysis and Applications* 2010; **31**(5):2291–2317.
42. Stoll M, Pearson JW, Wathen A. Preconditioners for state constrained optimal control problems with Moreau–Yosida penalty function, 2010. Submitted.

A.2 Preconditioning for state constraints

This paper is published as

J. W. Pearson,M. Stoll, and A. Wathen, ***Preconditioners for state constrained optimal control problems with Moreau-Yosida penalty function*, Numer. Lin. Alg. Appl., 21 (2014), pp. 81–97.**

Result from the paper

In this paper we derive robust preconditioners for the problem when the state is constrained. Table A.2 shows the number of CG iterations per Newton step for different values of β and ε, a parameter needed for a regularization of the state-constrained problem.

$\varepsilon \downarrow \beta \rightarrow$	1e-2	1e-4	1e-6
1e-4	19	27	26
1e-6	26	34	28
1e-8	32	36	28

Table A.2: Number of CG iterations per Newton step for different values of β and ε, using a direct factorization of $K + \widehat{M}$. The example was again the $2D$-results for non-zero Dirichlet boundary and upper box constraint at 0.1.

NUMERICAL LINEAR ALGEBRA WITH APPLICATIONS
Numer. Linear Algebra Appl. 2014; **21**:81–97
Published online 21 November 2012 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/nla.1863

Preconditioners for state-constrained optimal control problems with Moreau–Yosida penalty function

John W. Pearson [2], Martin Stoll [1,*,†] and Andrew J. Wathen [2]

[1] *Computational Methods in Systems and Control Theory, Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, 39106 Magdeburg, Germany*
[2] *Numerical Analysis Group, Mathematical Institute, 24–29 St Giles', Oxford OX1 3LB, U.K.*

SUMMARY

Optimal control problems with partial differential equations as constraints play an important role in many applications. The inclusion of bound constraints for the state variable poses a significant challenge for optimization methods. Our focus here is on the incorporation of the constraints via the Moreau–Yosida regularization technique. This method has been studied recently and has proven to be advantageous compared with other approaches. In this paper, we develop robust preconditioners for the efficient solution of the Newton steps associated with the fast solution of the Moreau–Yosida regularized problem. Numerical results illustrate the efficiency of our approach. Copyright © 2012 John Wiley & Sons, Ltd.

Received 13 July 2010; Revised 1 October 2012; Accepted 15 October 2012

KEY WORDS: state-constrained problems; PDE-constrained optimization; saddle point systems; preconditioning; Newton method; Krylov subspace solver

1. INTRODUCTION

Optimization problems with constraints given by PDEs arise in a variety of applications (see [1]). Comprehensive introductions to this field can be found in [1,2]. Throughout this paper, we consider the minimization of a functional $J(y,u)$ defined as

$$J(y,u) := \frac{1}{2}\|y - y_d\|^2_{L^2(\Omega)} + \frac{\beta}{2}\|u\|^2_{L^2(\Omega)}, \tag{1}$$

with $\Omega \subset \mathbb{R}^{\bar{d}}$, $\bar{d} \in \{2,3\}$. In (1), $\beta \in \mathbb{R}^+$ represents a regularization parameter, and y_d is a given function that represents the desired state. The state y and the control u are linked via the Poisson equation

$$-\triangle y = u \text{ in } \Omega, \tag{2}$$

with boundary conditions $y = g$ on $\partial\Omega$ or

$$-\triangle y + y = u \text{ in } \Omega, \tag{3}$$

with boundary conditions $\partial y/\partial n = 0$ on $\partial\Omega$. We decide to consider both (2) and (3) as both play a significant role in the literature. The choice of g will typically be 0 or the projection of y_d onto the box defined by constraints. The introduction of box constraints on the control and the state, that is,

$$\underline{u} \leqslant u \leqslant \overline{u} \tag{4}$$

*Correspondence to: Martin Stoll, Computational Methods in Systems and Control Theory, Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, 39106 Magdeburg, Germany.
†E-mail: martin.stoll80@gmail.com

and

$$\underline{y} \leqslant y \leqslant \overline{y}, \tag{5}$$

is of practical interest. In this paper, we will focus on the numerical solution of the optimization problem given when state constraints are present. Effective preconditioning strategies for the control constrained case can be found in [3,4]. Recently, operator preconditioning for the state-constrained case has also received more attention [5].

We will later show that a semismooth Newton method applied to the Moreau–Yosida regularization of (1) leads to a linear system of saddle point form. The saddle point matrix is symmetric and indefinite, and a variety of methods exists to solve problems of this type efficiently (see [6] for a survey). In practice, the linear system is usually of sufficiently high dimension that iterative solution methods are needed, and it is never solved without the application of a preconditioner $\mathcal{P}$, which is chosen to enhance the convergence behavior of the iterative method. A variety of preconditioners exists to tackle saddle point problems. The aim of this paper is to present preconditioners that are tailored towards the efficient solution of the linear system arising from the discretization of an optimal control problem involving a PDE and state constraints. In general, the state-constrained problem is a considerably harder problem (see Section 2) than the control constrained problem. In this paper, we will introduce preconditioning strategies that allow for robust solution of the linear system with respect to both the regularization parameter β and the parameter coming from the Moreau–Yosida penalty term.

The paper is organized as follows. The problem we are interested in will be presented in detail in Section 2. Our focus in this paper is to derive efficient preconditioners for the optimal control problems, and hence, our focus is to introduce all methods from a linear algebra perspective. We show how for each method the saddle point system can be preconditioned and efficiently solved using a Krylov subspace technique. We successively introduce three preconditioners, where the first is derived from previous results for PDE-constrained optimization and the others follow a recent technique focusing on robustness with respect to the regularization parameters. The numerical results presented in Section 4 illustrate the performance of the presented methods.

2. THE MOREAU–YOSIDA FORMULATION

We consider the case when state constraints are introduced and assume that the functional $J(y,u)$ (1) has to be minimized for functions y and u defined over a domain $\Omega \subset \mathbb{R}^{\bar{d}}$. The problem of minimizing (1) when bound constraints on the state are given is more complicated than the control constrained case [1,7,8] as in general the Lagrange multiplier is only a measure. Several remedies have been proposed for this problem. In [9], Meyer *et al.* consider regularized state constraints, that is,

$$\underline{y} \leqslant \varepsilon u + y \leqslant \overline{y}. \tag{6}$$

An alternative approach is given by changing the objective function (1) using the Moreau–Yosida penalty function [10] to give

$$\begin{aligned} J(y,u) := &\frac{1}{2}\|y - y_d\|^2_{L^2(\Omega)} + \frac{\beta}{2}\|u\|^2_{L^2(\Omega)} + \frac{1}{2\varepsilon}\|\max\{0, y - \overline{y}\}\|^2_{L^2(\Omega)} \\ &+ \frac{1}{2\varepsilon}\|\min\left\{0, y - \underline{y}\right\}\|^2_{L^2(\Omega)}, \end{aligned} \tag{7}$$

subject to the aforementioned state equations with appropriate boundary conditions. For the remainder of this manuscript, we will assume that the state equations and hence $J(y,u)$ are considered in discretized form using an appropriate finite element discretization [3].

DOI: 10.1002/nla

The discretized version of state Equations (2) and (1) is given by

$$\begin{aligned} \text{Minimize } & \frac{1}{2}(y-y_d)^T M(y-y_d) + \frac{\beta}{2}u^T M u \\ & + \frac{1}{2\varepsilon}\max\{0, y-\overline{y}\}^T\, M \max\{0, y-\overline{y}\} \\ & + \frac{1}{2\varepsilon}\min\{0, y-\overline{y}\}^T\, M \min\{0, y-\overline{y}\} \\ \text{subject to } & Ky = Mu - f. \end{aligned} \tag{8}$$

Here, K represents the finite element stiffness matrix and M the mass matrix. We will only consider the lumped mass matrix here but comment later on how to precondition for the consistent mass matrix. Note that y, u, y_d, $\overline{y}$, and $\underline{y}$ now represent vectors. The optimality system of (8) looks as follows:

$$-K^T\lambda = -M(y-y_d) - \varepsilon^{-1}\chi_{\mathcal{A}_+} M\max\{0, y-\overline{y}\} - \varepsilon^{-1}\chi_{\mathcal{A}_-} M\min\left\{0, y-\underline{y}\right\} \tag{9}$$

$$\beta M u + M\lambda = 0 \tag{10}$$

$$-Ky + Mu = f \tag{11}$$

with $\chi_{\mathcal{A}_+}$ being the characteristic function for the indices where $y-\overline{y} > 0$ and $\chi_{\mathcal{A}_-}$ the characteristic function for the region where $y-\underline{y} < 0$. Note that $\mathcal{A}_+ = \{i : y_i > \overline{y}_i\}$ and $\mathcal{A}_- = \left\{i : y_i < \underline{y}_i\right\}$ are the active sets associated with the bound constraints on the state y at step k. If we now wish to apply a semismooth Newton method to (9)–(11), we must solve the following system at every step:

$$\begin{bmatrix} M+\varepsilon^{-1}G_{\mathcal{A}}MG_{\mathcal{A}} & 0 & -K^T \\ 0 & \beta M & M \\ -K & M & 0 \end{bmatrix} \begin{bmatrix} y^{(k+1)} \\ u^{(k+1)} \\ \lambda^{(k+1)} \end{bmatrix} = \begin{bmatrix} c_{\mathcal{A}} \\ 0 \\ f \end{bmatrix}, \tag{12}$$

where $c_{\mathcal{A}} = My_d + \varepsilon^{-1}\left(G_{\mathcal{A}_+}MG_{\mathcal{A}_+}\overline{y} + G_{\mathcal{A}_-}MG_{\mathcal{A}_-}\underline{y}\right)$ defines part of the right hand side, $\mathcal{A} = \mathcal{A}_-\cup\mathcal{A}_+$, and the G matrices are projections onto the active sets defined by $\mathcal{A}$. The application of the semismooth Newton method to these problems has been previously studied (see [3, 10, 11]). Our task is the efficient solution of the linear system in (12), which is of saddle point form. Note that we do not focus on the discussion of the inexact semismooth Newton method here but rather refer to [12] where it was observed that with suitable preconditioning this method performed just as well as the exact semismooth Newton method.

In the case of the state equation being defined by (3), we define $K := K_N + M$, where K_N is the stiffness matrix for a pure Neumann problem, and obtain the same formulation as shown previously.

The Moreau–Yosida regularization has also recently been analyzed for semilinear elliptic problems (see [13]).

3. SOLUTION OF THE LINEAR SYSTEM AND EIGENVALUE ANALYSIS

The system matrix

$$\mathcal{K} := \begin{bmatrix} L & 0 & -K^T \\ 0 & \beta M & M \\ -K & M & 0 \end{bmatrix} \tag{13}$$

is symmetric and indefinite; we define $L = M + \varepsilon^{-1}G_{\mathcal{A}}MG_{\mathcal{A}}$ for the remainder of this paper. Note that the block $blkdiag(L, \beta M)$ is symmetric and positive definite, as we have a mass matrix as one term and a mass matrix plus a submatrix of a mass matrix as the other. The matrix K is the stiffness matrix associated with the weak formulation of (2) or (3) – it is symmetric and positive definite.

DOI: 10.1002/nla

Benzi *et al.* [6] discuss properties and numerical methods to solve matrices of saddle point form. As $\mathcal{K}$ is a large and sparse, symmetric and indefinite matrix, a Krylov subspace solver [14–16] will be our method of choice. For smaller (and typically 2D) examples, direct methods [17, 18] will prove very efficient, but for large and/or 3D problems, these methods are likely to run out of memory.

The choice of preconditioners that we mention in this section is motivated by an observation about the eigenvalues of the preconditioned system $\mathcal{P}^{-1}\mathcal{K}$ for certain preconditioners $\mathcal{P}$. Murphy *et al.* show in [19] that for some idealized preconditioners, the matrix $\mathcal{P}^{-1}\mathcal{K}$ has only a small number of eigenvalues (3 for a block-diagonal preconditioner and 2 for a block-triangular preconditioner).

One method that is a standard choice for symmetric and indefinite systems is the minimal residual method (MINRES) introduced in [20], which is a method for minimizing the residual $\|r_k\|_2 = \|\mathcal{K}x_k - b\|_2$ over the current Krylov subspace

$$\text{span}\left\{r_0, \mathcal{K}r_0, \mathcal{K}^2 r_0, \ldots, \mathcal{K}^{k-1} r_0\right\}.$$

To be able to use MINRES, we need the preconditioner to be symmetric and positive definite, and hence, block-diagonal preconditioners would present a natural choice [14, 21]. A preconditioner for MINRES and the aforementioned problem could look like the following:

$$\mathcal{P} = \begin{bmatrix} A_0 & 0 & 0 \\ 0 & A_1 & 0 \\ 0 & 0 & \widehat{S} \end{bmatrix}, \tag{14}$$

with A_0, A_1, and $\widehat{S}$ being approximations to the $(1,1)$-block, the $(2,2)$-block, and the Schur complement, respectively. The use of MINRES for optimal control problems has been recently investigated in [22–26]. Note that MINRES is also applicable in the case of a semidefinite $(1,1)$-block, which is the case if we were to consider the minimization of $J(y,u)$ as in (1), but with the $\|y - y_d\|^2$ term given on some subdomain $\Omega_1 \subset \Omega$ (as opposed to Ω itself). This problem was investigated in [27]. We believe that the results presented here can be applied to the subdomain case when MINRES is employed with a block-diagonal preconditioner.

Another class of methods that has proven to be of interest is based on the fact that for some preconditioners, the preconditioned saddle point matrix $\mathcal{P}^{-1}\mathcal{K}$ is symmetric and positive definite in an inner product defined by a matrix $\mathcal{H}$, that is, $\langle \mathcal{P}^{-1}Ax, y\rangle_{\mathcal{H}} = \langle x, \mathcal{P}^{-1}Ay\rangle_{\mathcal{H}}$ where $\langle x, y\rangle_{\mathcal{H}} = x^T\mathcal{H}y$. There exists a variety of such methods [28–33], which can also be combined to give rise to new methods [34, 35]. Herzog and Sachs [3] analyzed the method of Schöberl and Zulehner [33] for state and control constrained optimal control problems.

We wish to focus our attention on the so-called Bramble–Pasciak conjugate gradient (CG) method introduced in [29], a method that uses a block-triangular preconditioner

$$\mathcal{P} = \begin{bmatrix} A_0 & 0 & 0 \\ 0 & A_1 & 0 \\ -K & M & -\widehat{S} \end{bmatrix}, \tag{15}$$

with A_0, A_1, and $\widehat{S}$ being approximations just as previously mentioned. Once the preconditioner is applied to $\mathcal{K}$, the resulting preconditioned matrix $\widehat{\mathcal{K}} = \mathcal{P}^{-1}\mathcal{K}$ is not symmetric anymore but self adjoint in a nonstandard inner product defined by

$$\mathcal{H} = \begin{bmatrix} L - A_0 & 0 & 0 \\ 0 & \beta M - A_1 & 0 \\ 0 & 0 & \widehat{S} \end{bmatrix}. \tag{16}$$

It is clear that for $\mathcal{H}$ to define an inner product, the diagonal blocks have to be symmetric and positive definite. Although this is in general a rather tricky issue requiring an eigenvalue estimation problem, in the case of (lumped) mass matrices, scaling is straightforward [4]. Further, for the case of a consistent mass matrix, Rees and Stoll showed that the scaling issues can be easily removed [24]. For more details on the implementation and properties of the nonstandard inner product solver, we refer to [3, 24, 29, 33, 34, 36, 37] and Algorithm 1.

DOI: 10.1002/nla

1: Given $\mathbf{x}_0 = 0$, set $\mathbf{r}_0 = \mathcal{P}^{-1}(\mathbf{b} - \mathcal{K}\mathbf{x}_0)$ and $\mathbf{p}_0 = \mathbf{r}_0$
2: **for** $k = 0, 1, \ldots$ **do**
3: $\quad \alpha = \frac{\langle \mathbf{r}_k, \mathbf{r}_k \rangle_{\mathcal{H}}}{\langle \mathcal{P}^{-1}\mathcal{K}\mathbf{p}_k, \mathbf{p}_k \rangle_{\mathcal{H}}}$
4: $\quad \mathbf{x}_{k+1} = \mathbf{x}_k + \alpha \mathbf{p}_k$
5: $\quad \mathbf{r}_{k+1} = \mathbf{r}_k - \alpha \mathcal{P}^{-1}\mathcal{K}\mathbf{p}_k$
6: $\quad \beta = \frac{\langle \mathbf{r}_{k+1}, \mathbf{r}_{k+1} \rangle_{\mathcal{H}}}{\langle \mathbf{r}_k, \mathbf{r}_k \rangle_{\mathcal{H}}}$
7: $\quad \mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \beta \mathbf{p}_k$
8: **end for**

Algorithm 1: Nonstandard inner product conjugate gradient.

3.1. First preconditioner

The Schur complement of $\mathcal{K}$ is given by

$$S = KL^{-1}K^T + \beta^{-1}M. \tag{17}$$

For the case $L = M$, it was proposed [23] to neglect the term $\beta^{-1}M$, which would in our case result in an approximation $\widehat{S}_0 = KL^{-1}K^T$ to S. For a symmetric system, the clustering of the eigenvalues will govern the convergence of the iterative scheme, and we want to analyze the eigenvalue distribution of $\widehat{\mathcal{K}} = \mathcal{P}^{-1}\mathcal{K}$ for an idealized case. We consider now the block-triangular preconditioner with the choice $A_0 = L$, $A_1 = \beta M$, and $\widehat{S}_0 = KL^{-1}K^T$– in this case, the eigenvalues of the preconditioned matrix $\mathcal{P}^{-1}\mathcal{K}$ can be read off the diagonal blocks, that is,

$$\mathcal{P}^{-1}\mathcal{K} = \begin{bmatrix} I & 0 & -L^{-1}K^T \\ 0 & I & \beta^{-1}I \\ 0 & 0 & I + \beta^{-1}K^{-T}LK^{-1}M \end{bmatrix}, \tag{18}$$

which shows that there are $2n$ eigenvalues equal to 1 and n eigenvalues are given by the eigenvalues of $I + \beta^{-1}K^{-T}LK^{-1}M$. Thus, we wish to find eigenvalue bounds for $I + \beta^{-1}K^{-T}LK^{-1}M$. The eigenvalue bounds may be obtained from a field of value analysis.[‡] Note that the matrix $I + \beta^{-1}K^{-T}LK^{-1}M$ is similar to the symmetric matrix $M^{1/2}\left(I + \beta^{-1}K^{-T}LK^{-1}M\right)M^{-1/2} = I + \beta^{-1}M^{1/2}K^{-T}LK^{-1}M^{1/2}$ and

$$\frac{x^T x + \beta^{-1}x^T M^{1/2}K^{-T}LK^{-1}M^{1/2}x}{x^T x} = 1 + \frac{\beta^{-1}(z^T Lz)(x^T Mx)\left(y^T K^{-T}K^{-1}y\right)}{(x^T x)(y^T y)(z^T z)} \tag{19}$$

with $y = M^{1/2}x$ and $z = K^{-1}y$. The second term on the right-hand side of (19) can be bounded using the results of Proposition 1.29 and Theorem 1.32 in [14], which provide bounds for the eigenvalues of the consistent mass matrix and the stiffness matrix. Namely, with h being the mesh size of our finite element, we have that

$$ch^2 \leqslant \frac{x^T Mx}{x^T x} \leqslant Ch^2 \text{ and } dh^2 \leqslant \frac{x^T Kx}{x^T x} \leqslant D$$

with c, C, d, and D being mesh-independent constants. Note that these are the bounds for a 2D problem. For 3D bounds, we also refer to [14] but do not discuss them here. This directly gives bounds for almost all the terms in (19), and the only term that we need to analyze further is $z^T Lz/z^T z$. Using the definition of L, we obtain $\left(z^T Mz + \varepsilon^{-1}z^T G_{\mathcal{A}}MG_{\mathcal{A}}z\right)/z^T z$, which obviously can be bounded above by $(1 + \varepsilon^{-1})Ch^2$. Hence, the overall bound is given by

$$\lambda_{\max}^{(I+\beta^{-1}K^{-T}LK^{-1}M)} \leqslant 1 + \frac{C^2}{\beta d^2} + \frac{C^2}{\beta \varepsilon d^2}. \tag{20}$$

[‡]The field of values of a matrix $A \in \mathbb{R}^{n,n}$ is a set given by $\frac{x^T Ax}{x^T x}$ $\forall x \neq 0, x \in \mathbb{R}^n$.

DOI: 10.1002/nla

Similarly, the minimum eigenvalues are given by

$$1 \leqslant 1 + \frac{c^2 h^4}{\beta D^2} + \frac{c^2 h^4}{\beta \varepsilon D^2} \leqslant \lambda_{\min}^{(I+\beta^{-1}K^{-T}LK^{-1}M)}. \tag{21}$$

Theorem 3.1
For the consistent mass matrix M and the stiffness matrix K of a $Q1$ finite element space, the eigenvalues of the matrix

$$I + \beta^{-1} K^{-T} L K^{-1} M$$

lie in the interval $\left[1, 1 + \frac{C^2}{\beta d^2} + \frac{C^2}{\beta \varepsilon d^2}\right]$.

We remark that the eigenvalue distribution depends on the regularization parameter β as was previously observed for other cases (see [4, 23]). It also depends on the value of the penalty parameter ε: with decreasing value of ε, the upper bound for the eigenvalues in Theorem 3.1 will increase.

We used the block $\widehat{S}_0 = KM^{-1}K$ as an approximation for the Schur complement of the system matrix $\mathcal{K}$. This choice results in good clustering of the eigenvalues but is too expensive for practical purposes as $\widehat{S}_0^{-1}$ involves the term K^{-1} (the discretized PDE) twice. One now has to approximate the matrix K as best as possible. For this, it is very important to take the structure of the infinite-dimensional problem into account. For both PDEs (2) and (3), the underlying operators are elliptic PDEs, and hence, multigrid provides a suitable and optimal preconditioner. The most efficient method would certainly be a geometric multigrid method as described in [38, 39]. It is well known that algebraic multigrid (AMG) provides very good approximations to the aforementioned operators while allowing greater flexibility than their geometric counterparts [40, 41]. As we implemented our method within the deal.II framework [42], we use the available interface to Trilinos [43] and the smoothed aggregation AMG method implemented there [44]. Our choice will be to approximate K by a small number of V-cycles and a fixed number of steps of a Chebyshev smoother. The mass matrix M can be efficiently approximated using a variety of methods. In our case, as we only work with lumped mass matrices, we can solve for M cheaply. For consistent mass matrices, the Chebyshev semi-iteration [45, 46] provides a powerful preconditioner [24, 47].

3.2. Two improved preconditioners

When testing the preconditioner in the previous section, we observe, both theoretically and in practice, strong dependence on the regularization parameters β and ε. We therefore wish to introduce ideas for modifying the preconditioner, so as to improve the performance of our iterative solvers for small values of β and ε. We base our ideas on recent efforts [48–50] for PDE-constrained optimization problems without state constraints.

We now motivate our first modified preconditioner. It is based on the observation that if all mass matrices are lumped, the matrix L can be split up in the following way:

$$L = \begin{bmatrix} M_{\mathcal{I}} & 0 \\ 0 & (1+\varepsilon^{-1})M_{\mathcal{A}} \end{bmatrix},$$

where $M_{\mathcal{I}}$ is the part of the mass matrix that corresponds to the free variables and $M_{\mathcal{A}}$ analogously to the active sets. Our aim is to propose a Schur complement preconditioner of the form

$$\widehat{S}_1 = \left(K + \widehat{M}\right) L^{-1} \left(K + \widehat{M}\right), \tag{22}$$

where we have constructed $\widehat{S}_1$ to approximate the Schur complement $S = KL^{-1}K + \beta^{-1}M$ better than $\widehat{S}_0$. Hence, we examine $\widehat{S}_1$ in more detail,

$$\widehat{S}_1 = KL^{-1}K + \widehat{M}L^{-1}\widehat{M} + KL^{-1}\widehat{M} + \widehat{M}L^{-1}K, \tag{23}$$

DOI: 10.1002/nla

and look for a way for $\widehat{M}L^{-1}\widehat{M}$ to approximate the term $\beta^{-1}M$ in the best possible manner. Writing

$$\widehat{M} = \begin{bmatrix} \alpha M_{\mathcal{I}} & 0 \\ 0 & \gamma M_{\mathcal{A}} \end{bmatrix},$$

for some parameters α and γ, gives that if $\widehat{M}L^{-1}\widehat{M} = \beta^{-1}M$, then

$$\begin{bmatrix} \alpha^2 M_{\mathcal{I}} & 0 \\ 0 & \gamma^2(1+\varepsilon^{-1})^{-1}M_{\mathcal{A}} \end{bmatrix} = \widehat{M}L^{-1}\widehat{M} = \beta^{-1}M = \begin{bmatrix} \beta^{-1}M_{\mathcal{I}} & 0 \\ 0 & \beta^{-1}M_{\mathcal{A}} \end{bmatrix}. \tag{24}$$

This yields that

$$\alpha = \frac{1}{\sqrt{\beta}} \text{ and } \gamma = \frac{\sqrt{1+\varepsilon^{-1}}}{\sqrt{\beta}}, \tag{25}$$

which we then use for $\widehat{M}$ in our approximation to $\widehat{S}_1$ in (22).

We find that the Schur complement approximation derived leads to much improved convergence properties when the resulting preconditioner is used in conjunction with MINRES. This is what was observed in [49] and [51] for the Poisson control and convection–diffusion control problems, respectively, without state constraints. For these cases, it was possible to prove robust eigenvalue bounds using simple algebraic manipulation and Rayleigh quotient arguments, respectively.[§] We find that in this case, proof of such a rigorous result is not as straightforward because of the ill-conditioning of the $(1,1)$-block for small values of ϵ – we present below a Rayleigh quotient analysis on the basis of that of the previous section and [51] in 2D (the 3D case is similar).

We note that the eigenvalues of the matrix $\widehat{S}_1^{-1}S$ are bounded by the extreme values of the Rayleigh quotient

$$\frac{v^T S v}{v^T \widehat{S}_1 v} = \frac{v^T K L^{-1} K v + \beta^{-1} v^T M v}{v^T K L^{-1} K v + \beta^{-1} v^T M v + v^T \widehat{M} L^{-1} K v + v^T K L^{-1} \widehat{M} v} \tag{26}$$

$$= \left(1 + \frac{v^T \widehat{M} L^{-1} K v + v^T K L^{-1} \widehat{M} v}{v^T K L^{-1} K v + \beta^{-1} v^T M v}\right)^{-1} =: R. \tag{27}$$

The term of interest here is

$$\frac{v^T \widehat{M} L^{-1} K v + v^T K L^{-1} \widehat{M} v}{v^T K L^{-1} K v + \beta^{-1} v^T M v} = \frac{b^T a + a^T b}{a^T a + b^T b},$$

with $a = L^{-1/2} K v$ and $b = L^{-1/2}\widehat{M} v$. We note first that we may write

$$(a-b)^T(a-b) \geqslant 0 \Leftrightarrow \frac{a^T b + b^T a}{a^T a + b^T b} \leqslant 1$$

for any a, b. Using this along with the fact that $a^T a + b^T b > 0$ gives immediately that $R \geqslant 1/2$ for all v.

For the upper bound of R, we first note that excluding multiplicative constants of $\mathcal{O}(1)$, $\lambda(K) \in [h^2, 1]$, $\lambda(M) = h^2$, $\lambda(L) = [1, 1+\epsilon^{-1}]$. Further, as

$$\widehat{M}L^{-1} = L^{-1}\widehat{M} = \frac{1}{\sqrt{\beta}} \begin{bmatrix} I & 0 \\ 0 & (1+\epsilon^{-1})^{-1/2} I \end{bmatrix},$$

[§]For these cases, it was shown that the eigenvalues of the preconditioned Schur complement were contained within the interval $[\frac{1}{2}, 1]$ independently of the two parameters involved in the problem: h and β. It is in some sense unsurprising that the same cannot be rigorously proved for this problem, as there are now three parameters involved: h, β, and ε.

DOI: 10.1002/nla

we have that $\lambda\left(\widehat{M}L^{-1}\right) = \lambda\left(L^{-1}\widehat{M}\right) \in \left[\beta^{-1/2}, \beta^{-1/2}(1+\epsilon^{-1})^{-1/2}\right]$. We are now in a position to consider the upper bound of R, which corresponds to the largest negative value of $\left(b^T a + a^T b\right)/\left(a^T a + b^T b\right)$. We write (using the eigenvalue bounds stated)

$$\frac{b^T a + a^T b}{a^T a + b^T b} \geqslant -\frac{2\beta^{-1/2}\chi\eta^{-1/2}}{\chi^2\eta^{-1}h^{-2} + \beta^{-1}h^2} =: -\frac{2}{\omega + \omega^{-1}}, \quad \text{where } \omega = \beta^{1/2}\chi\eta^{-1/2}h^{-2} > 0,$$

where $\chi \in [dh^2, D]$, $\eta \in [1, 1+\epsilon^{-1}]$, and $\omega = \beta^{1/2}\chi\eta^{-1/2}h^{-2} > 0$, again excluding multiplicative constants. Therefore,

$$R \leqslant \left(1 - \frac{2}{\omega + \omega^{-1}}\right)^{-1},$$

and hence,

$$\lambda(\widehat{S}_1^{-1}S) \in \left[\frac{1}{2}, \left(1 - \frac{2}{\omega + \omega^{-1}}\right)^{-1}\right].$$

We note that from this analysis that the lower bound for $\lambda\left(\widehat{S}_1^{-1}S\right)$ is concrete for all values of h, β, and ε. However, we observe that we cannot prove a universal, clean upper bound for $\lambda\left(\widehat{S}_1^{-1}S\right)$, because of the nonsymmetry of the matrices $\widehat{M}L^{-1}K$ and its transpose, and the ill-conditioning of the matrix L. As such, the presentation of the upper bound should be regarded as heuristic guidance as to the combination of parameters our approximation should work best for, as opposed to rigorous proof. The obvious worst-case scenario of this analysis occurs when $\omega = 1$, in which case $\widehat{S}_1^{-1}S$ could have an infinitely large eigenvalue. However, we can easily argue that this will not happen, as it would either correspond to the quantity $v^T S v$ in the aforementioned analysis being infinite (which clearly cannot happen) or the quantity $v^T \widehat{S}_1 v$ being equal to 0 (which will not occur because the matrices $K + \widehat{M}$ and L, which make up $\widehat{S}$, are both invertible).

We observe that if the value denoted ω is not too close to 1, the heuristic upper bound presented earlier will be close to 1, and we observe that this occurs in many practical cases. In Figure 1, we present graphs of eigenvalues of $\widehat{S}_1^{-1}S$ for a variety of values of h, β, and ϵ to demonstrate that in almost all cases of practical interest, our Schur complement approximation is highly effective. These figures not only validate the lower bound we have proved but also indicate that the upper bound of $\lambda(\widehat{S}_1^{-1}S)$ is of $\mathcal{O}(1)$ in the majority of cases. (The figures consist of parameter regimes that are close to the worst case in terms of the largest eigenvalue of $\widehat{S}_1^{-1}S$.) As numerical evidence indicates the effectiveness of this Schur complement, but theoretical study indicates that it is difficult to rigorously prove this, we would describe the resulting preconditioner as *parameter robust* as opposed to *parameter independent*.

We find that another potentially potent Schur complement approximation is given by

$$\widehat{S}_2 = \left[K + \frac{1}{\sqrt{\beta}}M\left(I + \frac{1}{\sqrt{\epsilon}}G_{\mathcal{A}}\right)\right]\mathcal{M}_G^{-1}\left[K + \frac{1}{\sqrt{\beta}}M\left(I + \frac{1}{\sqrt{\epsilon}}G_{\mathcal{A}}\right)\right]^T,$$

where $\mathcal{M}_G = \left(I + \left(1/\sqrt{\epsilon}\right)G_{\mathcal{A}}\right)M\left(I + \left(1/\sqrt{\epsilon}\right)G_{\mathcal{A}}\right)$. Note that the matrices $G_{\mathcal{A}}$ are projections onto the active sets used in the semismooth Newton method. We perform a Rayleigh quotient analysis on this approximation, aiming to demonstrate that

$$S \approx K\left[\left(I + \frac{1}{\sqrt{\epsilon}}G_{\mathcal{A}}\right)M\left(I + \frac{1}{\sqrt{\epsilon}}G_{\mathcal{A}}\right)\right]^{-1}K^T + \frac{1}{\beta}M := \widetilde{S} \approx \widehat{S}_2,$$

which we do by considering the eigenvalues of $\widetilde{S}^{-1}S$ and $\widehat{S}_2^{-1}\widetilde{S}$, using Rayleigh quotients.

DOI: 10.1002/nla

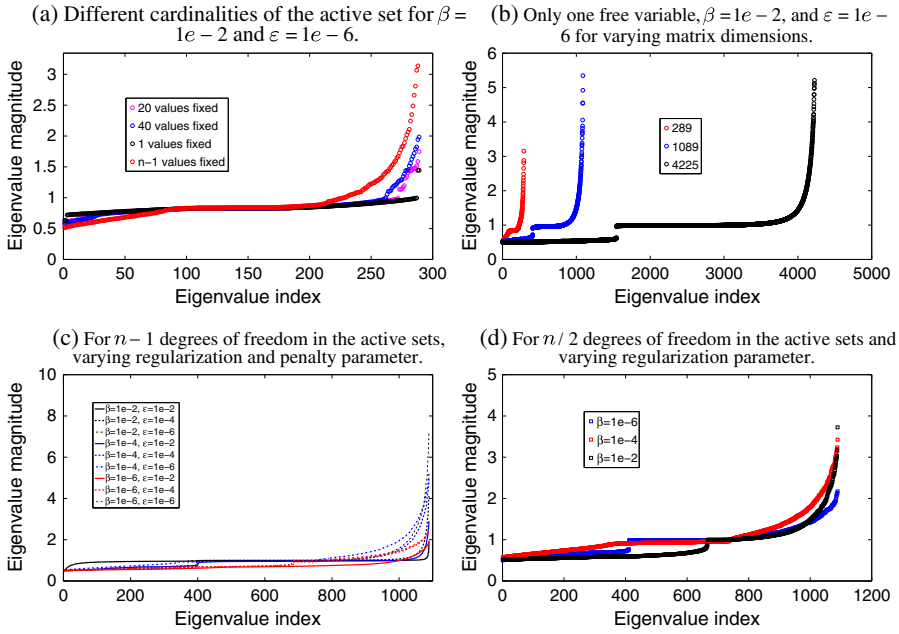


Figure 1. Eigenvalue distributions of $\widehat{S}_1^{-1} S$ for various problem set-ups.

We first examine the Rayleigh quotient

$$R_1 := \frac{v^T \left(M + \epsilon^{-1} G_{\mathcal{A}} M G_{\mathcal{A}}\right) v}{v^T \left(I + \epsilon^{-1/2} G_{\mathcal{A}}\right) M \left(I + \epsilon^{-1/2} G_{\mathcal{A}}\right) v} = \frac{a_1^T a_1 + b_1^T b_1}{(a_1 + b_1)^T (a_1 + b_1)},$$

where $a_1 = M^{1/2} v$ and $b_1 = \epsilon^{-1/2} M^{1/2} G_{\mathcal{A}} v$. We can see by straightforward algebra (using that $a_1^T a_1 > 0$ by positive definiteness of M) that $R_1 \geqslant 1/2$. Further, using the fact that $a_1^T b_1 = b_1^T a_1 \geqslant 0$ (by virtue of the diagonal and positive definite structure of the lumped mass matrix M and the diagonal and positive semidefinite structure of the projection matrix $G_{\mathcal{A}}$), we can show that $R_1 \leqslant 1$. From these bounds, it is a simple matter to show that the Rayleigh quotient $v^T S v / v^T \widetilde{S} v \in [1, 2]$.

Looking now at the eigenvalues of $\widehat{S}_2^{-1} \widetilde{S}$, we examine the Rayleigh quotient $v^T \widetilde{S} v / v^T \widehat{S}_2 v$, writing

$$R_2 := \frac{v^T \widetilde{S} v}{v^T \widehat{S}_2 v} = \frac{a_2^T a_2 + b_2^T b_2}{(a_2 + b_2)^T (a_2 + b_2)} = \left(1 + \frac{a_2^T b_2 + b_2^T a_2}{a_2^T a_2 + b_2^T b_2}\right)^{-1},$$

where $a_2 = M^{-1/2}(I + \epsilon^{-1/2} G_{\mathcal{A}})^{-1} K^T v$ and $b_2 = \beta^{-1/2} M^{1/2} v$. By algebraic manipulation (using that $b_2^T b_2 > 0$), it is clear that $\left(a_2^T b_2 + b_2^T a_2\right) / \left(a_2^T a_2 + b_2^T b_2\right) \leqslant 1$ and hence that $R_2 \geqslant 1/2$. For the upper bound of R_2, we consider the maximum negative value of $\left(a_2^T b_2 + b_2^T a_2\right) / \left(a_2^T a_2 + b_2^T b_2\right)$. Using that (excluding multiplicative constants of $\mathcal{O}(1)$) $\lambda(K) \in [h^2, 1]$, $\lambda(M) = h^2$, and $\lambda(I + \epsilon^{-1/2} G_{\mathcal{A}}) \in [1, 1 + \epsilon^{-1/2}]$, we may use a very similar approach as for $\widehat{S}_1$

$$\frac{a_2^T b_2 + b_2^T a_2}{a_2^T a_2 + b_2^T b_2} \geqslant -\frac{2\beta^{-1/2} \chi \zeta^{-1}}{\chi^2 \zeta^{-2} h^{-2} + \beta^{-1} h^2} = -\frac{2}{\nu + \nu^{-1}}, \quad \text{where } \nu = \beta^{1/2} \chi \zeta^{-1} h^{-2} > 0$$

and therefore that $R_2 \leqslant \left(1 - (2/(\nu + \nu^{-1}))\right)^{-1}$. Here, χ is as defined, $\zeta \in [1, 1 + \epsilon^{-1/2}]$, and $\nu = \beta^{1/2} \chi \zeta^{-1} h^{-2} > 0$ up to a multiplicative constant of $\mathcal{O}(1)$.

DOI: 10.1002/nla

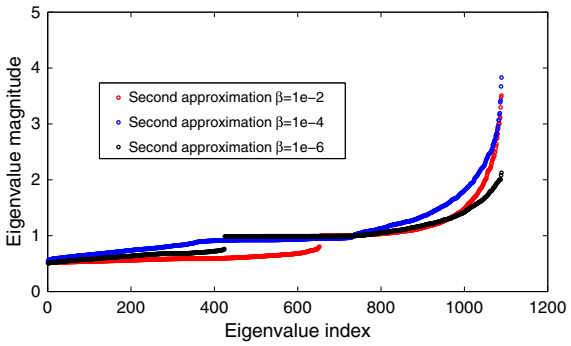


Figure 2. Eigenvalues for $n/2$ variables in the active sets and varying values of β.

As for the Schur complement approximation $\widehat{S}_1$, we are able to demonstrate a clean lower bound for the eigenvalues of the preconditioned Schur complement but not a concrete upper bound. Similarly, as for $\widehat{S}_1$, the limiting case $\nu = 1$ is potentially a problem here, but we can easily argue that this value will never be attained in the same way (as the matrices $K + (1/\sqrt{\beta})M\left(I + (1/\sqrt{\varepsilon})G_{\mathcal{A}}\right)$ and $\mathcal{M}_G$ are both invertible). It is clear that the bound shown is tight if the value denoted ν is far from 1. We again find that this is frequently the case in practical situations and provide numerical evidence to demonstrate that the eigenvalue distribution in practice is tight for a wide range of practical situations. Figure 2 shows the eigenvalues for a small example using the Schur complement approximation $\widehat{S}_2$. Once again, we describe the preconditioner resulting from our Schur complement approximation as *parameter robust* as opposed to *parameter independent*.

We therefore believe that both $\widehat{S}_1$ and $\widehat{S}_2$ are viable and often effective Schur complement approximations for the problem we are considering.

As we only focus on lumped mass matrices in this paper, we refrain from showing results for the nonlumped case in Section 4, although some results still hold for consistent mass matrices.

We note that the analytical results of this section were obtained for an idealized case where we use approximations of the form

$$\widehat{S} = (K + \widehat{M})L^{-1}(K + \widehat{M}) \text{ with } \widehat{S}^{-1} = (K + \widehat{M})^{-1}L(K + \widehat{M})^{-1}.$$

However, in practice, we always use

$$\widehat{S}^{-1} = \widehat{(K + \widehat{M})^{-1}}L\widehat{(K + \widehat{M})^{-1}},$$

where

$$\widehat{(K + \widehat{M})^{-1}}$$

denotes the application of an algebraic or geometric multigrid method to the matrix $K + \widehat{M}$. Note that as $\widehat{M}$ changes with every Newton iteration, we must recompute it at the beginning of each Newton step. Nevertheless, the reduction in iteration numbers is so significant that this is clearly the preferred approach, especially for small values of the regularization parameters.

3.3. Nested approach

A strategy that will prove useful in the context of solving state-constrained problems is the so-called nested approach [3]. This technique starts by computing the solution to the state-constrained problem on a very coarse grid. In the next step, a uniform refinement is performed for the mesh, and the

DOI: 10.1002/nla

solution from the coarse level is prolonged onto the fine mesh. This solution is then used as an initial guess for the Newton method on the fine level. Once the solution is computed to a desired accuracy, we can proceed in the same way onto the next finer grid. It is hoped (and will be shown in the next section) that this strategy reduces the number of Newton steps significantly.

4. NUMERICAL EXPERIMENTS

All results shown in this section were computed using the deal.II [42] framework with an implementation of the Bramble–Pasciak CG method that uses the 2-norm of the relative preconditioned residual (10^{-6}) as the stopping criterion. The Newton method is stopped whenever the active sets remain unchanged [52]. For the approximation via AMG, we use 10 steps of a Chebyshev smoother and four V-cycles of the smoothed aggregation AMG method implemented in Trilinos [44]. As for the domain Ω, we consider the unit square or cube. All results are performed on a Centos Linux machine with Intel(R) Xeon(R) CPU X5650 at 2.67 GHz CPUs and 48 GB of RAM.

4.1. Results for Dirichlet problems

2D results. The first example we compute is a Dirichlet problem with boundary condition $y = P_{[\underline{y},\overline{y}]}(y_d)$ on $\partial\Omega$ defined by

$$P_{[\underline{y},\overline{y}]}y_i = \begin{cases} y_{d\,i} & \text{if } \underline{y}_i < y_i < \overline{y}_i \\ \overline{y}_i & \text{if } y_i \geqslant \overline{y}_i \\ \underline{y}_i & \text{if } y_i \leqslant \underline{y}_i. \end{cases}$$

Figure 3 shows the desired state y_d, computed control u, and state y for the case without bound constraints. In Figure 4, we show the computed state and control for a bound constrained problem. The problem is unconstrained from below, and the upper bound is given by $\overline{y} = 0.1$. It can be seen that there is a small active set where y_d is attained, which results in the 'hole' in the control (see Figure 4(b) and the active set (black contour) in Figure 4(a)). Here, the desired state is given by

$$y_d = \sin(2\pi x_1 x_2).$$

Table I shows results for $\beta = 1e-2$, $\varepsilon = 1e-6$, and the upper bound $\overline{y} = 0.1$. It can be seen that for this set-up, the preconditioner as well as the Newton method behave almost independently of the mesh parameter. In our experience, the performance of the AMG preconditioner deteriorates for meshes with smaller mesh size h. The increase in iteration numbers could not be observed if a factorization of $K + \widehat{M}$ was used; however, for large problems, this is not feasible. Hence, we chose a rather large number of V-cycles, namely 4, to approximate the matrix well. A parameter-independent approximation of $K + \widehat{M}$ should be investigated in future research. Note that the timings shown also include the set-up of the preconditioner for each Newton step in the improved preconditioner. As we can see from Table II where we show results for the same set-up but with the nonrobust preconditioner presented in Section 3.1, the improvement is substantial as for the set-up with $\beta = 1e-2$, $\varepsilon = 1e-6$; however, the Newton method did not converge within 50 iterations. For $\beta = 1e-2$, $\varepsilon = 1e-4$, we show the results with four multigrid cycles in Table III and observe good convergence for this set-up of parameters. As for the nonrobust preconditioner, the AMG only has to approximate K, a smaller number of V-cycles produces the results shown in Table IV. Note that as observed in [12], the quality of the preconditioner determines the convergence of the semismooth Newton method with inexact solves.

The next comparison we wish to make is that of the quality of the preconditioner for different values of the parameters. As we mentioned earlier, some dependence of the AMG routine on the parameters could be observed. Hence, our choice is a factorization of $K + \widehat{M}$ for a smaller mesh with 16,641 degrees of freedom. The results shown in Table V show that having no deterioration in the approximation of $K + \widehat{M}$ results in almost constant low iteration numbers for the CG steps per iteration. In practice, one should of course use approximations to $K + \widehat{M}$.

DOI: 10.1002/nla

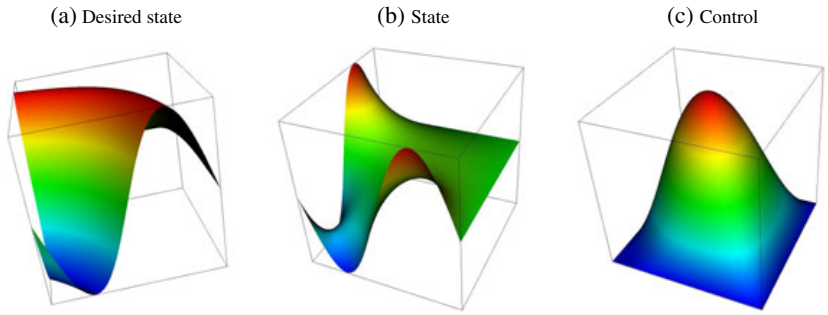


Figure 3. Desired state, state, and control for unconstrained problem.

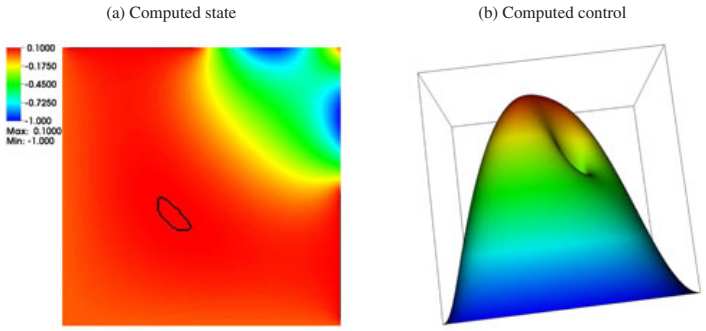


Figure 4. Computed state and control for constrained problem.

Table I. 2D results for nonzero Dirichlet boundary, $\beta = 1e-2$, $\varepsilon = 1e-6$, and $\overline{y} = 0.1$.

Degrees of freedom	Newton steps	Total conjugate gradient	Conjugate gradient per Newton	Time for Newton
1089	14	259	18	11.27
4225	8	138	17	12.63
16,641	7	108	15	30.3
66,049	7	118	16	123.54
263,169	6	103	17	462.78
1,050,625	7	183	26	3267.28

Table II. 2D results for nonzero Dirichlet boundary, $\beta = 1e-2$, $\varepsilon = 1e-6$, and $\overline{y} = 0.1$.

Degrees of freedom	Newton steps	Total conjugate gradient	Conjugate gradient per Newton	Time for Newton
1089	14	780	55	29.31
4225	2	14	7	1.25
16,641	5	59	11	13.63
66,049	10	170	17	131.98
263,169	No convergence after 50 Newton steps			

DOI: 10.1002/nla

Table III. 2D results for nonzero Dirichlet boundary, $\beta = 1e-2$, $\varepsilon = 1e-4$, and $\overline{y} = 0.1$ with four V-cycles.

Degrees of freedom	Newton steps	Total conjugate gradient	Conjugate gradient per Newton	Time for Newton
1089	9	145	16	5.55
4225	3	21	7	1.87
16,641	4	34	8	7.75
66,049	4	39	9	31.26
263,169	9	172	19	530.38

Table IV. 2D results for nonzero Dirichlet boundary, $\beta = 1e-2$, $\varepsilon = 1e-4$, and $\overline{y} = 0.1$ with two V-cycles.

Degrees of freedom	Newton steps	Total conjugate gradient	Conjugate gradient per Newton	Time for Newton
1089	9	150	16	5.64
4225	3	24	8	1.76
16,641	7	124	17	22.63
66,049	8	307	38	191.54
263,169	10	1059	105	2507.48

Table V. Number of conjugate gradient iterations per Newton step for different values of β and ε, using a direct factorization of $K + \widehat{M}$. The example was again the 2D results for nonzero Dirichlet boundary and $\overline{y} = 0.1$.

$\varepsilon \downarrow \beta \rightarrow$	$1e-2$	$1e-4$	$1e-6$
$1e-4$	19	27	26
$1e-6$	26	34	28
$1e-8$	32	36	28

Table VI. 3D results for zero Dirichlet boundary and $\overline{y} = 0.1$.

Degrees of freedom	Newton steps	Total conjugate gradient	Conjugate gradient per Newton	Time for Newton
729	4	48	12	1.05
4913	4	53	13	6.17
35,937	4	53	13	37.78
274,625	3	41	13	228.05

3D results. We now wish to show results for the 3D example with the desired state given by

$$y_d = \sin(2\pi x_1 x_2 x_3)$$

and a zero Dirichlet boundary condition. In this case, we again consider the upper bound $\overline{y} = 0.1$ and the parameters $\beta = 1e-2$ and $\varepsilon = 1e-4$. The results shown in Table VI show that the iteration numbers per Newton step as well as the number of Newton steps stays almost constant.

To illustrate the robustness of our approach, we present in Table VII iteration numbers for the Newton method and average number of CG iterations for a 3D problem. The desired state is given by

$$y_d = \sin(2\pi x_1 x_2 x_3)$$

with a lower bound fixed at $\underline{y} = 0$. All results are obtained for a fixed mesh with 4913 degrees of freedom, and for this to resemble a realistic scenario, we refrain from using a factorization as

DOI: 10.1002/nla

Table VII. Number of conjugate gradient iterations per Newton step for different values of β and ε, using an algebraic multigrid for the approximation of $K + \widehat{M}$. The example is 3D with 4913 degrees of freedom.

$\varepsilon \downarrow \beta \rightarrow$	$1e-2$ AS(∅ conjugate gradient)	$1e-4$ AS(∅ conjugate gradient)	$1e-6$ AS(∅ conjugate gradient)
$1e-4$	2(14)	4(20)	3(22)
$1e-6$	2(14)	5(21)	3(22)
$1e-8$	2(14)	5(21)	3(22)

Table VIII. Number of conjugate gradient iterations per Newton step for different values of β and ε, using an algebraic multigrid for the approximation of $K + \widehat{M}$. The example is 3D with 35, 937 degrees of freedom.

$\varepsilon \downarrow \beta \rightarrow$	$1e-2$ AS(∅ conjugate gradient)	$1e-4$ AS(∅ conjugate gradient)	$1e-6$ AS(∅ conjugate gradient)
$1e-4$	2(14)	7(23)	5(25)
$1e-6$	2(14)	7(26)	5(25)
$1e-8$	2(15)	7(27)	5(25)

Table IX. 3D results for Neumann boundary and $\underline{y} = 0.2$.

Degrees of freedom	Newton steps	Total conjugate gradient	Conjugate gradient per Newton	Time for Newton
729	6	78	13	1.95
4913	4	63	15	7.49
35,937	4	72	18	51.61
274,625	4	75	18	413.21
2,146,689	5	104	20	4458.58

was carried out in [5] but rather use the AMG with four V-cycles and 10 steps of a Chebyshev smoother. As was pointed out in [12], the convergence of the outer Newton iteration depends on the quality of the preconditioner, and to allow for a fair comparison of the parameter set-ups, we solve rather accurately to a tolerance of $1e-10$ for all parameter values. We can see that for both mesh sizes, the iteration numbers are very low and almost constant. We believe that the slight increase in Newton iterations can be avoided using the nested approach, which we did not employ for Tables VII and VIII.

4.2. Results for Neumann boundary

In this section, we only consider 3D results for the problem with the state equation given by (3). We start with the desired state given by

$$y_d = \sin(2\pi x_1 x_2 x_3)$$

and the lower bound $\underline{y} = 0$ – the results are shown in Table IX. Here, we take $\beta = 1e-3$ and $\varepsilon = 1e-5$. An illustration of the desired state and the constrained state is shown in Figure 5.

We next compute an example presented in [3] where the desired state is given by

$$y_d = \begin{cases} 1 & \text{if } x_0 < 0.5 \\ -2 & \text{otherwise.} \end{cases} \tag{28}$$

The upper bound is given by $\overline{y} = 0$ and $\beta = 1e-2$ and $\varepsilon = 1e-4$. The results are shown in Table X where, in contrast to [3], we observe parameter-robust convergence.

DOI: 10.1002/nla

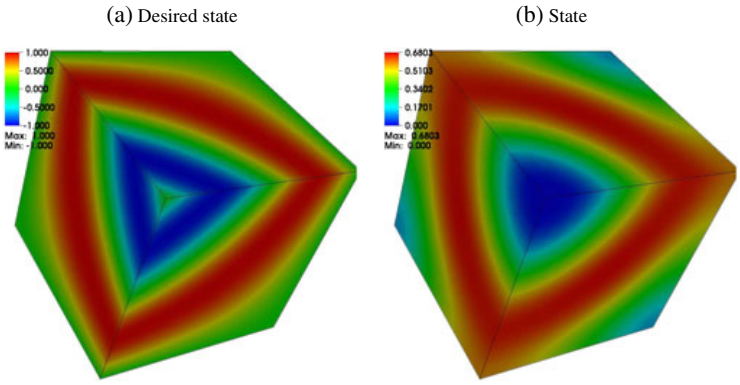


Figure 5. Desired state and state with lower bound $\underline{y} = 0$ for problem in 3D.

Table X. 3D results for Neumann boundary and $\overline{y}$ from Herzog and Sachs.

Degrees of freedom	Newton steps	Total conjugate gradient	Conjugate gradient per Newton	Time for Newton
729	5	52	10	1.32
4913	1	13	13	1.67
35,937	2	30	15	22.19
274,625	2	32	16	181.56
2,146,689	1	16	16	758.11

5. CONCLUSIONS AND OUTLOOK

In this paper, we introduced preconditioners for a state-constrained PDE-constrained optimization problem when solved using the Moreau–Yosida penalization. The Krylov subspace solvers we used showed very promising performance as we could observe robust convergence of the preconditioned Krylov solver for a wide range of parameters.

In the future, it would be useful to investigate the problem where the L^2 norm of $y - y_d$ is measured on a subdomain of Ω, as opposed to Ω itself. This, like the problem considered in this manuscript, could be solved using the preconditioned MINRES algorithm. Also, the choice of multilevel method for the parameter dependent matrix $K + \widehat{M}$ should be reconsidered, as we could observe dependence on the parameters for smaller meshes within the AMG preconditioner. Geometric multigrid and more advanced AMG preconditioners should be investigated, and incorporation of a more sophisticated scheme for the parameter ε would also be desirable for future implementations. Finally, a significant piece of future work would be to extend the results presented here to time-dependent problems, as well as to more difficult PDEs.

ACKNOWLEDGEMENTS

The authors would like to thank the anonymous referees for their careful reading of the manuscript and helpful comments. The authors would also like to thank Anton Schiela for useful conversations about this work. The first author was supported for this work by the Engineering and Physical Sciences Research Council (UK), Grant EP/P505216/1. This publication is partially based on work performed when the second author was supported by Award No. KUK-C1-013-04, made by King Abdullah University of Science and Technology (KAUST).

DOI: 10.1002/nla

REFERENCES

1. Hinze M, Pinnau R, Ulbrich M, Ulbrich S. *Optimization with PDE Constraints*, Mathematical Modelling: Theory and Applications. Springer-Verlag: New York, 2009.
2. Tröltzsch F. *Optimale Steuerung partieller Differentialgleichungen: Theorie, Verfahren und Anwendungen*. Vieweg Verlag: Wiesbaden, 2005.
3. Herzog R, Sachs EW. Preconditioned conjugate gradient method for optimal control problems with control and state constraints. *SIAM Journal on Matrix Analysis and Applications* 2010; **31**:2291–2317.
4. Stoll M, Wathen A. Preconditioning for partial differential equation constrained optimization with control constraints. *Numerical Linear Algebra with Applications* 2012; **19**:53–71.
5. Schiela A, Ulbrich S. Operator preconditioning for a class of constrained optimal control problems 2012. Submitted.
6. Benzi M, Golub GH, Liesen J. Numerical solution of saddle point problems. *Acta Numerica* 2005; **14**:1–137.
7. Casas E. Control of an elliptic problem with pointwise state constraints. *SIAM Journal on Control and Optimization* 1986; **24**:1309–1318.
8. Ito K, Kunisch K. *Lagrange Multiplier Approach to Variational Problems and Applications*, vol. 15 of Advances in Design and Control. Society for Industrial and Applied Mathematics (SIAM): Philadelphia, PA, 2008.
9. Meyer C, Prüfert U, Tröltzsch F. On two numerical methods for state-constrained elliptic control problems. *Optimization Methods and Software* 2007; **22**:871–899.
10. Ito K, Kunisch K. Semi-smooth Newton methods for state-constrained optimal control problems. *Systems Control Letters* 2003; **50**:221–228.
11. Bergounioux M, Kunisch K. Primal-dual strategy for state-constrained optimal control problems. *Computational Optimization and Applications* 2002; **22**:193–224.
12. Kanzow C. Inexact semismooth Newton methods for large-scale complementarity problems. *Optimization Methods and Software* 2004; **19**:309–325.
13. Krumbiegel K, Neitzel I, Rösch A. Sufficient optimality conditions for the Moreau–Yosida-type regularization concept applied to the semilinear elliptic optmimal control problems with pointwise state constraints. *Technical Report 1503/2010*, WIAS, 39 · 10117 Berlin Germany, 2010.
14. Elman HC, Silvester DJ, Wathen AJ. *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*, Numerical Mathematics and Scientific Computation. Oxford University Press: New York, 2005.
15. Greenbaum A. *Iterative Methods for Solving Linear Systems*, vol. 17 of Frontiers in Applied Mathematics. Society for Industrial and Applied Mathematics (SIAM): Philadelphia, PA, 1997.
16. Saad Y. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics: Philadelphia, PA, 2003.
17. Davis T. Umfpack version 4.4 user guide. *Technical Report*, Dept. of Computer and Information Science and Engineering Univ. of Florida, Gainesville, FL, 2005.
18. Duff I. Sparse numerical linear algebra: direct methods and preconditioning. *Technical Report TR/PA/96/22*, CERFACS, Toulouse, France, 1996. Also RAL Report RAL 96-047.
19. Murphy MF, Golub GH, Wathen AJ. A note on preconditioning for indefinite linear systems. *SIAM Journal on Scientific Computing* 2000; **21**:1969–1972.
20. Paige CC, Saunders MA. Solutions of sparse indefinite systems of linear equations. *SIAM Journal on Numerical Analysis* 1975; **12**:617–629.
21. Fischer B. *Polynomial Based Iteration Methods for Symmetric Linear Systems*, Wiley-Teubner Series Advances in Numerical Mathematics. John Wiley & Sons Ltd: Chichester, 1996.
22. Blank L, Sarbu L, Stoll M. Preconditioning for Allen–Cahn variational inequalities with non-local constraints. *Journal of Computational Physics* 2012; **231**:5406–5420.
23. Rees T, Dollar HS, Wathen AJ. Optimal solvers for PDE-constrained optimization. *SIAM Journal on Scientific Computing* 2010; **32**:271–298.
24. Rees T, Stoll M. Block-triangular preconditioners for PDE-constrained optimization. *Numerical Linear Algebra with Applications* 2010; **17**:977–996.
25. Stoll M. All-at-once solution of a time-dependent time-periodic PDE-constrained optimization problems July 2011. Submitted.
26. Stoll M, Wathen A. All-at-once solution of time-dependent Stokes control. *Accepted Journal of Computational Physics* 2012. DOI: http://dx.doi.org/10.1016/j.jcp.2012.08.039.
27. Stoll M, Wathen A. All-at-once solution of time-dependent PDE-constrained optimization problems 2010. Submitted.
28. Benzi M, V. On the eigenvalues of a class of saddle point matrices. *Numerische Mathematik* 2006; **103**:173–196.
29. Bramble JH, Pasciak JE. A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems. *Mathematics of Computation* 1988; **50**:1–17.
30. Dohrmann CR, Lehoucq RB. A primal-based penalty preconditioner for elliptic saddle point systems. *SIAM Journal on Numerical Analysis* 2006; **44**:270–282.
31. Fischer B, Ramage A, Silvester DJ, Wathen AJ. Minimum residual methods for augmented systems. *BIT* 1998; **38**:527–543.
32. Liesen J, Parlett BN. On nonsymmetric saddle point matrices that allow conjugate gradient iterations. *Numerische Mathematik* 2008; **108**:605–624.
33. Schöberl J, Zulehner W. Symmetric indefinite preconditioners for saddle point problems with applications to PDE-constrained optimization problems. *SIAM Journal on Matrix Analysis and Applications* 2007; **29**:752–773.

DOI: 10.1002/nla

34. Stoll M. Solving linear systems using the adjoint. *PhD Thesis*, University of Oxford, 2009.
35. Stoll M, Wathen A. Combination preconditioning and the Bramble–Pasciak$^+$ preconditioner. *SIAM Journal on Matrix Analysis and Applications* 2008; **30**:582–608.
36. Elman HC. Multigrid and Krylov subspace methods for the discrete Stokes equations. In *Seventh Copper Mountain Conference on Multigrid Methods*, Vol. CP 3339, Melson ND, Manteuffel TA, McCormick SF, Douglas CC (eds). NASA: Hampton, VA, 1996; 283–299.
37. Rees T, Stoll M, Wathen A. All-at-once preconditioners for PDE-constrained optimization. *Kybernetika* 2010; **46**:341–360.
38. Hackbusch W. *Multigrid Methods and Applications*, vol. 4 of Springer Series in Computational Mathematics. Springer-Verlag: Berlin, 1985.
39. Wesseling P. *An Introduction to Multigrid Methods*, Pure and Applied Mathematics (New York). John Wiley & Sons Ltd.: Chichester, 1992.
40. Falgout R. An introduction to algebraic multigrid. *Computing in Science and Engineering* 2006; **8**:24–33. Special Issue on Multigrid Computing.
41. Ruge JW, Stüben K. *Algebraic Multigrid, in Multigrid Methods*, vol. 3 of Frontiers Applied Mathematics. SIAM: Philadelphia, PA, 1987. 73–130.
42. Bangerth W, Hartmann R, Kanschat G. deal.II – a general-purpose object-oriented finite element library. *ACM Transactions on Mathematical Software* 2007; **33**:Art. 24, 27.
43. Heroux M, Bartlett R, Hoekstra VHR, Hu J, Kolda T, Lehoucq R, Long K, Pawlowski R, Phipps E, Salinger A, Thornquist H, Tuminaro R, Willenbring J, Williams A. An overview of Trilinos. *Technical Report SAND2003-2927*, Sandia National Laboratories, USA, 2003.
44. Gee M, Siefert C, Hu J, Tuminaro R, Sala M. ML 5.0 smoothed aggregation user's guide. *Tech. Rep. SAND2006-2649*, Sandia National Laboratories, USA, 2006.
45. Golub GH, Varga RS. Chebyshev semi-iterative methods, successive over-relaxation iterative methods, and second order Richardson iterative methods. I. *Numerische Mathematik* 1961; **3**:147–156.
46. Golub GH, Varga RS. Chebyshev semi-iterative methods, successive over-relaxation iterative methods, and second order Richardson iterative methods. II. *Numerische Mathematik* 1961; **3**:157–168.
47. Wathen AJ, Rees T. Chebyshev semi-iteration in preconditioning for problems including the mass matrix. *Electronic Transactions in Numerical Analysis* 2008; **34**:125–135.
48. Pearson JW, Stoll M, Wathen AJ. Regularization-robust preconditioners for time-dependent PDE-constrained optimization problems. *To appear SIAM Journal on Matrix Analysis and Applications* 2012.
49. Pearson JW, Wathen AJ. A new approximation of the Schur complement in preconditioners for PDE-constrained optimization. *Numerical Linear Algebra with Applications* 2012; **19**:816–829.
50. Takacs S, Zulehner W. Convergence analysis of multigrid methods with collective point smoothers for optimal control problems. *Computing and Visualization in Science* 2011; **14**:131–141.
51. Pearson JW, Wathen AJ. Fast iterative solvers for convection–diffusion control problems 2011. Submitted.
52. Bergounioux M, Ito K, Kunisch K. Primal-dual strategy for constrained optimal control problems. *SIAM Journal on Control and Optimization* 1999; **37**:1176–1194.

DOI: 10.1002/nla

A.3 Regularization Robust Preconditioning

This paper is published as

J. W. Pearson, M. Stoll, and A. J. Wathen, ***Regularization-robust preconditioners for time-dependent PDE-constrained optimization problems*, SIAM J. Matrix Anal.Appl, 33 (2012), pp. 1126–1152.**

Result from the paper

We develop a robust Schur-complement approximation for the time-dependent optimization problem. The proven eigenvalue bounds are independent of all system parameters. Table A.3 shows the iteration numbers and computing time for 20 time-steps and various mesh and regularization parameters.

DoF	MINRES(T)	MINRES(T)	MINRES(T)
	$\beta = 1e-2$	$\beta = 1e-4$	$\beta = 1e-6$
98,260	10(2)	12(2)	12(2)
1,918,740	10(14)	12(17)	12(18)
5,492,500	10(148)	12(171)	12(170)

Table A.3: Results for Discretize-then-Optimize approach via trapezoidal rule.

SIAM J. MATRIX ANAL. APPL.
Vol. 33, No. 4, pp. 1126–1152

REGULARIZATION-ROBUST PRECONDITIONERS FOR TIME-DEPENDENT PDE-CONSTRAINED OPTIMIZATION PROBLEMS*

JOHN W. PEARSON†, MARTIN STOLL‡, AND ANDREW J. WATHEN†

Abstract. In this article, we motivate, derive, and test effective preconditioners to be used with the MINRES algorithm for solving a number of saddle point systems which arise in PDE-constrained optimization problems. We consider the distributed control problem involving the heat equation and the Neumann boundary control problem involving Poisson's equation and the heat equation. Crucial to the effectiveness of our preconditioners in each case is an effective approximation of the Schur complement of the matrix system. In each case, we state the problem being solved, propose the preconditioning approach, prove relevant eigenvalue bounds, and provide numerical results which demonstrate that our solvers are effective for a wide range of regularization parameter values, as well as mesh sizes and time-steps.

Key words. PDE-constrained optimization, saddle point systems, time-dependent PDE-constrained optimization, preconditioning, Krylov subspace solver

AMS subject classifications. Primary, 65F10, 65N22, 65F50; Secondary, 76D07

DOI. 10.1137/110847949

1. Introduction.

The development of fast iterative solvers for saddle point problems from a variety of applications is a subject attracting considerable attention in numerical analysis [12, 46, 57, 14]. As such problems become more complex, a natural objective in creating efficient solvers is to ensure that the computation time taken by the solver grows as close to linearly as possible with the mesh parameter of the discretized problem. In more detail, it is desirable that if the problem size doubles due to refinement of the mesh, then the computation time roughly doubles as well.

Recently, due to the development of efficient algorithms and increased computing power, the solution of optimal control problems with PDE constraints has become an increasingly active field [55, 27, 29]. The goal is to find efficient methods that solve the discretized problem with the objective in mind of creating preconditioners that again scale linearly with decreasing mesh size. The interested reader is referred to [48, 22, 40, 44, 50] and the references therein for steady (time-independent) problems and to [52, 53, 39, 51, 4] for unsteady (time-dependent) problems. There are also multigrid [20] approaches to both time-dependent and time-independent optimal control problems [25, 26, 54, 6, 7, 1, 19, 18].

Often, designing solvers that are insensitive to the mesh size is found to compromise the performance of the solver for small values of the regularization parameter inherent in PDE-constrained optimization problems, unless the approximation of the Schur complement of the matrix system is chosen carefully. Therefore, recently

*Received by the editors September 14, 2011; accepted for publication (in revised form) by M. Benzi June 28, 2012; published electronically October 11, 2012.
http://www.siam.org/journals/simax/33-4/84794.html

†Numerical Analysis Group, Mathematical Institute, 24–29 St Giles', Oxford, OX1 3LB, United Kingdom (john.pearson@worc.ox.ac.uk, wathen@maths.ox.ac.uk). The first author's work was supported by the Engineering and Physical Sciences Research Council (UK), grant EP/P505216/1.

‡Computational Methods in Systems and Control Theory, Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, 39106 Magdeburg, Germany (stollm@mpi-magdeburg.mpg.de).

research has gone into developing preconditioners which are insensitive to the regularization as well as the mesh size; see [37, 48] for instance for such solvers for the Poisson control problem.

Here, we consider whether it is possible to build solvers for the time-dependent analogue of this problem, that is, the optimal control of the heat equation. We consider the distributed control problem and attempt to minimize a functional that is commonly used in the literature [55]. We also investigate solvers for the boundary control problem, first in the time-independent Poisson control case and then in the time-dependent heat equation control case. Further, we develop a solver for a distributed subdomain problem of this type.

This paper is structured as follows. In section 2, we outline some prerequisite saddle point theory, state the problems that we consider the iterative solution of, and describe a solver for the distributed Poisson control problem (originally detailed in [37]) that we base our methods on. In section 3, we motivate and derive the preconditioners that we apply for the problems stated, proving relevant eigenvalue bounds of the preconditioned Schur complements of the matrix systems when our recommended approximations are used. In section 4, we provide numerical results for a variety of test problems to demonstrate the effectiveness of our approaches, and in section 5 we make some concluding remarks.

2. Problems and discretization. This section is structured as follows. In section 2.1, we briefly detail elements of saddle point theory that we utilize throughout the remainder of this paper. In section 2.2, we describe work that has been undertaken on the (time-independent) distributed Poisson control problem and state the formulations of the time-dependent problem that we consider. In section 2.3, we describe the time-independent and time-dependent Neumann boundary control problems we consider in this paper.

2.1. Saddle point theory. The problems we discuss in this paper are all of *saddle point structure*, i.e., of the form

$$\underbrace{\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix}}_{\mathcal{A}} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}, \tag{2.1}$$

where $A \in \mathbb{R}^{m\times m}$ is symmetric and positive definite or semidefinite, $B \in \mathbb{R}^{p\times m}$ with $m \geq p$ and the matrix $\mathcal{A}$ is nonsingular. The properties and solution methods for such systems have been an active field of research for two decades. State-of-the-art numerical methods for solving saddle point problems can be found in [3, 12] and the references therein.

Throughout this paper, we consider block diagonal preconditioners for such saddle point systems of the form

$$\mathcal{P} = \begin{bmatrix} \widehat{A} & 0 \\ 0 & \widehat{S} \end{bmatrix},$$

which is symmetric and positive definite. To apply this preconditioner, we therefore require a good approximation $\widehat{A}$ to the $(1,1)$-block of the matrix system, A, and $\widehat{S}$ as an approximation to the (negative) *Schur complement*, $S := BA^{-1}B^T$. Note that in general we are only interested in the application of $\widehat{A}^{-1}$ and $\widehat{S}^{-1}$, which allows the use of multigrid [20] or algebraic multigrid (AMG) [45, 13] methods, for example.

Such a preconditioner is known to be effective because the spectrum of the matrix $\mathcal{P}^{-1}\mathcal{A}$ is given by

$$\lambda(\mathcal{P}^{-1}\mathcal{A}) = \left\{1, \frac{1}{2}(1 \pm \sqrt{5})\right\},$$

provided $\mathcal{P}^{-1}\mathcal{A}$ is nonsingular, when $\widehat{A} = A$, and $\widehat{S} = S$ (see [34] for details). In this case, an appropriate Krylov subspace method applied to the system (2.1) will converge in three iterations with this preconditioner. Throughout the remainder of this paper, we apply the MINRES algorithm of Paige and Saunders [35] to saddle point systems of the form $\mathcal{A}$, with preconditioner $\mathcal{P}$ as in (2.2).

Note that many other preconditioners are possible such as block triangular preconditioners [34, 8, 42, 49] or constraint preconditioners [11, 30, 59]. These usually have to be combined with different iterative solvers, either symmetric ones [8, 16] or nonsymmetric ones such as GMRES [47].

2.2. Distributed control problems. One of the most common problems employed in PDE-constrained optimization for the study of numerical techniques is the *distributed Poisson control problem* with Dirichlet boundary conditions [55]. This is written as

$$\begin{aligned} \min_{y,u} \ & \frac{1}{2}\|y - \bar{y}\|^2_{L_2(\Omega_1)} + \frac{\beta}{2}\|u\|^2_{L_2(\Omega_2)} \\ \text{s.t} \quad & -\nabla^2 y = u \quad \text{in } \Omega, \\ & y = f \quad \text{on } \partial\Omega, \end{aligned} \tag{2.2}$$

where y is referred to as the *state* variable with $\bar{y}$ some known *desired state* and u as the *control variable*. Here Ω_1 and Ω_2 are subsets of the domain $\Omega \subset \mathbb{R}^d$, where $d \in \{2, 3\}$, on which the problem is defined with boundary $\partial\Omega$, and $\beta > 0$ is the (Tikhonov) regularization parameter. Note that we will limit ourselves to the cases $\Omega_2 = \Omega$ and $\Omega_2 = \partial\Omega$—the boundary control problem is addressed in the next section.

There are two common approaches for solving this optimization problem. One can consider the infinite-dimensional problem, write down the Lagrangian, and then discretize the first order conditions, which is referred to as the *optimize-then-discretize* approach, or one can first discretize the objective function and then build a discrete Lagrangian with corresponding first order conditions. The latter is the *discretize-then-optimize* approach. Recently, the paradigm that both approaches should coincide was used to derive discretization schemes for PDE-constrained optimization (see, for example, [25]).

The problem (2.2) represents a *steady* problem, i.e., $y = y(\mathbf{x})$, where $\mathbf{x}$ denotes the spatial variable. Using a Galerkin finite element method [12] and a discretize-then-optimize strategy, with the state y, control u, and *adjoint* state or Lagrange multiplier p all discretized using the same basis functions [40, 37], leads to the following first order system:

$$\begin{bmatrix} M_1 & 0 & K \\ 0 & \beta M & -M \\ K & -M & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} M\bar{\mathbf{y}} \\ \mathbf{0} \\ \mathbf{c} \end{bmatrix}, \tag{2.3}$$

where $\mathbf{y}$, $\mathbf{u}$, and $\mathbf{p}$ denote the vectors of coefficients in the finite element expansion in terms of the basis functions $\{\phi_j,\ j = 1, \ldots, n\}$ of y, u, and p, respectively, $\bar{\mathbf{y}}$ is the

vector corresponding to $\bar{y}$, and $\mathbf{c}$ corresponds to the Dirichlet boundary conditions imposed. Here, M denotes a finite element *mass matrix* over the domain Ω; similarly, M_1 is the finite element mass matrix for the domain Ω_1 and K a *stiffness matrix* over Ω. The matrices are of dimension $n \times n$ with n being the degrees of freedom of the finite element approximation. These are defined by

$$M = \{m_{ij},\ i,j = 1,\dots,n\}, \quad m_{ij} = \int_\Omega \phi_i \phi_j \ \mathrm{d}\Omega, \tag{2.4}$$
$$K = \{k_{ij},\ i,j = 1,\dots,n\}, \quad k_{ij} = \int_\Omega \nabla\phi_i \cdot \nabla\phi_j \ \mathrm{d}\Omega.$$

Note that we often consider M to be a lumped mass matrix, that is,

$$M = \mathrm{diag}(m_{ii}), \quad m_{ii} = \sum_{j=1}^{n} \left| \int_\Omega \phi_i \phi_j \ \mathrm{d}\Omega \right|.$$

The matrix M_1 can be obtained analogously to the above by replacing Ω by Ω_1.

In literature such as [37, 48], solvers are designed which solve (2.3) in computational time independent of the mesh size h and any choice of regularization parameter β. The solver that we consider is based on the block diagonal preconditioner discussed in [37], in which the system (2.3) is written in classical saddle point form (2.1) with $A = \begin{bmatrix} M & 0 \\ 0 & \beta M \end{bmatrix}$ and $B = \begin{bmatrix} K & -M \end{bmatrix}$. The $(1,1)$-block is then approximated by the application of Chebyshev semi-iteration to each mass matrix for consistent mass matrices [58] or by simple inversion for lumped mass matrices, and the (negative) Schur complement

$$S = BA^{-1}B^T = KM^{-1}K + \frac{1}{\beta}M$$

is approximated by

$$\widehat{S} = \left(K + \frac{1}{\sqrt{\beta}}M\right) M^{-1} \left(K + \frac{1}{\sqrt{\beta}}M\right).$$

It is shown in [37] that $\lambda(\widehat{S}^{-1}S) \in [\frac{1}{2}, 1]$ for any choice of step-size h and regularization parameter β when this approximation is used. Using a multigrid process to approximate the inverse of the matrix $K + \frac{1}{\sqrt{\beta}}M$ gives a viable solution strategy.

In this paper, we attempt to extend this preconditioning framework to time-dependent analogues of the above problem. Specifically, we will consider the optimal control of the heat equation. This problem may be written as

$$\begin{aligned} &\min_{y,u} \ J(y,u) \\ &\text{s.t} \quad y_t - \nabla^2 y = u, \quad \text{for } (\mathbf{x},t) \in \Omega \times [0,T], \\ &\qquad\qquad y = f \quad \text{on } \partial\Omega, \\ &\qquad\qquad y = y_0 \quad \text{at } t = 0 \end{aligned} \tag{2.5}$$

for some functional $J(y,u)$, where f and y_0 may depend on $\mathbf{x}$ but not t. The functional that we consider here is a functional where we have observations (desired state) on the whole time-interval

$$J_1(y,u) = \frac{1}{2}\int_0^T \int_{\Omega_1} \left(y(\mathbf{x},t) - \bar{y}(\mathbf{x},t)\right)^2 \ \mathrm{d}\Omega_1 \mathrm{d}t + \frac{\beta}{2}\int_0^T \int_{\Omega_2} \left(u(\mathbf{x},t)\right)^2 \ \mathrm{d}\Omega_2 \mathrm{d}t. \tag{2.6}$$

Note that it is also possible to consider a problem where the desired state is only defined at a more limited set at times, for example, at only $t = T$, which would correspond to a functional of the form [36]

$$J_2(y,u) = \frac{1}{2}\int_{\Omega_1} \left(y(\mathbf{x},T) - \bar{y}(\mathbf{x})\right)^2 \, \mathrm{d}\Omega_1 + \frac{\beta}{2}\int_0^T \int_{\Omega_2} \left(u(\mathbf{x},t)\right)^2 \, \mathrm{d}\Omega_2 \mathrm{d}t.$$

We consider here only the problem relating to the functional $J_1(y,u)$, which we refer to as the "all-times case." Note that the state, control, and adjoint state are all now time-dependent functions. For now we again assume that $\Omega_2 = \Omega$.

As illustrated in [52], the matrix system arising from solving the problem (2.5) with $J(y,u) = J_1(y,u)$ varies according to whether a *discretize-then-optimize* or *optimize-then-discretize* strategy is applied. Applying the discretize-then-optimize approach, using the trapezoidal rule and the backward Euler scheme with N_t time steps of (constant) size τ to discretize the PDE in time, gives the matrix system [52]

$$\begin{bmatrix} \tau\mathcal{M}_{1/2}^{(1)} & 0 & \mathcal{K}^T \\ 0 & \beta\tau\mathcal{M}_{1/2} & -\tau\mathcal{M} \\ \mathcal{K} & -\tau\mathcal{M} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \tau\mathcal{M}_{1/2}^{(1)}\bar{\mathbf{y}} \\ \mathbf{0} \\ \mathbf{d} \end{bmatrix}, \tag{2.7}$$

where $\mathbf{y}$, $\mathbf{u}$, $\bar{\mathbf{y}}$, and $\mathbf{p}$ are vectors corresponding to the state, control, desired state, and adjoint at all time-steps $1, 2, \ldots, N_t$, and

$$\mathcal{M}_{1/2} = \begin{bmatrix} \frac{1}{2}M & & & & \\ & M & & & \\ & & \ddots & & \\ & & & M & \\ & & & & \frac{1}{2}M \end{bmatrix}, \quad \mathcal{M} = \begin{bmatrix} M & & & & \\ & M & & & \\ & & \ddots & & \\ & & & M & \\ & & & & M \end{bmatrix}, \tag{2.8}$$

$$\mathcal{M}_{1/2}^{(1)} = \begin{bmatrix} \frac{1}{2}M_1 & & & & \\ & M_1 & & & \\ & & \ddots & & \\ & & & M_1 & \\ & & & & \frac{1}{2}M_1 \end{bmatrix},$$

$$\mathcal{K} = \begin{bmatrix} M+\tau K & & & & \\ -M & M+\tau K & & & \\ & \ddots & \ddots & & \\ & & -M & M+\tau K & \\ & & & -M & M+\tau K \end{bmatrix}, \quad \mathbf{d} = \begin{bmatrix} M\mathbf{y}_0 + \mathbf{c} \\ \mathbf{c} \\ \vdots \\ \mathbf{c} \\ \mathbf{c} \end{bmatrix}.$$

Note that if n is the number of degrees of freedom in the spatial representation only, then each of the matrices in (2.8) belongs to $\mathbb{R}^{nN_t \times nN_t}$ with blocks as indicated, where $M, M_1, K \in \mathbb{R}^{n\times n}$. The overall coefficient matrix in (2.7) is of dimension $3nN_t \times 3nN_t$.

If, alternatively, the optimize-then-discretize approach is used with $J(y,u) = J_1(y,u)$, the matrix system becomes [52]

$$\begin{bmatrix} \tau\mathcal{M}_0 & 0 & \mathcal{K}^T \\ 0 & \beta\tau\mathcal{M}_{1/2} & -\tau\mathcal{M} \\ \mathcal{K} & -\tau\mathcal{M} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \tau\mathcal{M}_0\bar{\mathbf{y}} \\ \mathbf{0} \\ \mathbf{d} \end{bmatrix}, \tag{2.9}$$

where

$$\mathcal{M}_0 = \begin{bmatrix} M_1 & & & & \\ & M_1 & & & \\ & & \ddots & & \\ & & & M_1 & \\ & & & & 0 \end{bmatrix} \in \mathbb{R}^{nN_t \times nN_t}.$$

The matrix systems (2.7) and (2.9) are the systems corresponding to the time-dependent distributed control problem. The efficient solution of these saddle point systems will be considered in this paper.

2.3. Neumann boundary control problems. Another important problem in the field of PDE-constrained optimization is the class of *Neumann boundary control problems*. Note that this problem corresponds to $\Omega_2 = \partial\Omega$ in (2.2). In practical applications, these are perhaps the most useful class of problems. We start once more by considering the boundary control of Poisson's equation written as

$$\begin{aligned} \min_{y,u} \ & \frac{1}{2}\left\|y-\bar{y}\right\|^2_{L_2(\Omega)} + \frac{\beta}{2}\left\|u\right\|^2_{L_2(\partial\Omega)} \\ \text{s.t} \quad & -\nabla^2 y = f \quad \text{in } \Omega, \\ & \frac{\partial y}{\partial n} = u \quad \text{on } \partial\Omega, \end{aligned} \tag{2.10}$$

where f is the known source term, which may be zero, and the control, u, is applied in the form of a Neumann boundary condition. As for the distributed control case, we discretize y, u, and p using the same finite element basis functions.

The first order optimality conditions of a discretize-then-optimize approach yield the following matrix system:

$$\begin{bmatrix} M & 0 & K \\ 0 & \beta M_b & -N^T \\ K & -N & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} M\bar{\mathbf{y}} \\ \mathbf{0} \\ \mathbf{f} \end{bmatrix}, \tag{2.11}$$

where M and K are as before (see (2.4)), M_b here denotes the boundary mass matrix over $\partial\Omega$, and N corresponds to entries arising from terms within the integral $\int_{\partial\Omega} u\mathrm{tr}(v)\mathrm{d}s$ (with u the boundary control and $\mathrm{tr}(v)$ denoting the trace function acting on a member of the Galerkin test space). The vector $\mathbf{f}$ corresponds to f, the source term of Poisson's equation. The matrix in (2.11) is essentially of dimension $(2n + n_b) \times (2n + n_b)$, where n is the number of degrees of freedom for y and n_b the number of degrees of freedom for the boundary control, u.

As well as this problem, we also investigate the time-dependent analogue, that is, the Neumann boundary control of the heat equation. We write the problem that we consider as

$$\begin{aligned} \min_{y,u} \ & \frac{1}{2}\int_0^T \int_\Omega \left(y(\mathbf{x},t)-\bar{y}(\mathbf{x},t)\right)^2 \ \mathrm{d}\Omega\mathrm{d}t + \frac{\beta}{2}\int_0^T \int_{\partial\Omega} \left(u(\mathbf{x},t)\right)^2 \ \mathrm{d}s\mathrm{d}t, \\ \text{s.t.} \quad & y_t - \nabla^2 y = f \quad \text{for } (\mathbf{x},t) \in \Omega \times [0,T], \\ & \frac{\partial y}{\partial n} = u \quad \text{on } \partial\Omega. \end{aligned} \tag{2.12}$$

Note that this is related to the distributed control problem (2.5) with $J(y,u) = J_1(y,u)$. Although we could seek to solve the optimize-then-discretize formulation of this problem in a similar way as for the distributed control problem, we focus our attention on the discretize-then-optimize formulation. In this case, applying the backward Euler scheme in time and the trapezoidal rule, we obtain the matrix system

$$\begin{bmatrix} \tau\mathcal{M}_{1/2} & 0 & \mathcal{K}^T \\ 0 & \beta\tau\mathcal{M}_{1/2,b} & -\tau\mathcal{N}^T \\ \mathcal{K} & -\tau\mathcal{N} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \tau\mathcal{M}_{1/2}\bar{\mathbf{y}} \\ \mathbf{0} \\ \mathbf{g} \end{bmatrix}, \tag{2.13}$$

where $\mathcal{M}$ and $\mathcal{K}$ are as defined in (2.8), and

$$\mathcal{M}_{1/2,b} = \begin{bmatrix} \frac{1}{2}M_b & & & & & \\ & M_b & & & & \\ & & \ddots & & & \\ & & & M_b & & \\ & & & & \frac{1}{2}M_b \end{bmatrix},$$

$$\mathcal{N} = \begin{bmatrix} N & & & & \\ & N & & & \\ & & \ddots & & \\ & & & N & \\ & & & & N \end{bmatrix}, \quad \mathbf{g} = \begin{bmatrix} M\mathbf{y}_0 + \mathbf{f} \\ \mathbf{f} \\ \vdots \\ \mathbf{f} \\ \mathbf{f} \end{bmatrix}.$$

We will consider the iterative solution of the matrix systems (2.11) and (2.13), in addition to the distributed control problems previously stated, in section 3.

2.4. Possible extensions. In this section we wish to introduce some extensions of the above problems that in one form or another frequently appear in the field of optimization with PDE constraints. In many applications so-called box constraints for the state and/or the control have to be included. Here we highlight pointwise control constraints

$$u_a(x) \leq u(x) \leq u_b(x)$$

as well as pointwise state constraints

$$y_a(x) \leq y(x) \leq y_b(x).$$

These additional constraints can be handled very efficiently by so-called semismooth Newton methods [27, 23, 56, 28], whereas due to the reduced regularity of the Lagrange multiplier the state-constrained problem presents a more difficult problem [9]. It is also possible to include different or additional regularization terms in the objective function. A popular choice is the inclusion of a so-called sparsity term where the control u is given in the L_1-norm for which we write $\|\mathbf{u}\|_1$. This term can efficiently be treated as part of the semismooth Newton method (see [22]). Another possibility is to have differential operators acting on the control as part of the objective function, for which we write $\|Lu\|_2$. In this case efficient preconditioning depends on the nature of the operator L and how well it can be approximated. Recent examples for this can be found in [43, 4]. Combinations of all the above are of course possible and we address some possibilities in the next section.

3. Preconditioning. In this section, we motivate and discuss our proposed preconditioners for the matrix systems stated in section 2. These will be applied within the MINRES algorithm [35]. This section is structured as follows. In section 3.1.1, we propose a preconditioner for the matrix system (2.7) corresponding to a time-dependent distributed control problem, minimizing (2.6) and using a discretize-then-optimize formulation. We start with the case $\Omega_1 = \Omega$ and discuss the subdomain case next. In section 3.1.2, we motivate a preconditioner for (2.9), which is the same problem except with an optimize-then-discretize strategy employed. We then consider Neumann boundary control problems for the case $\Omega_1 = \Omega$; in section 3.2, we discuss the time-independent case corresponding to (2.11), and in section 3.3 we extend this theory to the time-dependent case, relating to (2.13). We only discuss the subdomain case $\Omega_1 \subset \Omega$ for the time-dependent problem in section 3.4. In section 4, we present numerical results to demonstrate that all our proposed solvers are effective in practice.

3.1. Time-dependent distributed control.

3.1.1. Minimizing J_1 with discretize-then-optimize. We start by considering the case $\Omega_1 = \Omega$, which gives $\mathcal{M}^{(1)}_{1/2} = \mathcal{M}_{1/2}$. Equation (2.7), which is the discretize-then-optimize formulation of (2.5) with $J(y,u) = J_1(y,u)$, can be written as a saddle point system with

$$A = \begin{bmatrix} \tau\mathcal{M}_{1/2} & 0 \\ 0 & \beta\tau\mathcal{M}_{1/2} \end{bmatrix}, \quad B = \begin{bmatrix} \mathcal{K} & -\tau\mathcal{M} \end{bmatrix},$$

in the notation of (2.1). The (negative) Schur complement of this system is therefore given by

$$S = \frac{1}{\tau}\mathcal{K}\mathcal{M}^{-1}_{1/2}\mathcal{K}^T + \frac{\tau}{\beta}\mathcal{M}\mathcal{M}^{-1}_{1/2}\mathcal{M}. \tag{3.1}$$

For this matrix system, we seek a (symmetric block diagonal) preconditioner of the form

$$\widehat{\mathcal{P}} = \begin{bmatrix} \widehat{A} & 0 \\ 0 & \widehat{S} \end{bmatrix} \tag{3.2}$$

to be used with MINRES.

For the approximation $\widehat{A}$, we apply a similar approach as for the Poisson control problem and take

$$\widehat{A} = \begin{bmatrix} \tau\widehat{\mathcal{M}}_{1/2} & 0 \\ 0 & \beta\tau\widehat{\mathcal{M}}_{1/2} \end{bmatrix}, \tag{3.3}$$

where $\widehat{\mathcal{M}}_{1/2}$ denotes the approximation of $\mathcal{M}_{1/2}$. Here a Chebyshev semi-iteration process is again taken to approximate consistent mass matrices or a simple inversion for lumped mass matrices.

We now wish to develop a result which enables us to find an accurate approximation to (3.1), as well as to approximate Schur complements that we will consider in section 3.1.2.

We start by noting that the matrix system (2.7) is of the form

$$\begin{bmatrix} \Phi_1 & 0 & \mathcal{K}^T \\ 0 & \beta\Phi_1 & -\Phi_2 \\ \mathcal{K} & -\Phi_2 & 0 \end{bmatrix}$$

with Schur complement given by

$$S = \mathcal{K}\Phi_1^{-1}\mathcal{K}^T + \frac{1}{\beta}\Phi_2\Phi_1^{-1}\Phi_2, \tag{3.4}$$

where Φ_1 and Φ_2 are symmetric positive definite, as they are block matrices solely consisting of mass matrices. (In section 3.1.2, we will consider approximations of Schur complements of the form (3.4), where Φ_1 and Φ_2 have the same such properties.)

We note that in all the cases we consider, the matrix $\Phi_1^{-1}\Phi_2$ simply involves scaled (positive) multiples of identity matrices. That is, all the relevant blocks are scalings of the same matrix $I \in \mathbb{R}^{n\times n}$. We may use the straightforward resulting observation that $\mathcal{M}\Phi_1^{-1}\Phi_2 = \Phi_1^{-1}\Phi_2\mathcal{M}$ with $\mathcal{M}$ defined as in (2.8) to demonstrate one further property that we will require in our analysis: that $\mathcal{K}\Phi_1^{-1}\Phi_2 + \Phi_1^{-1}\Phi_2\mathcal{K}^T$ is positive definite. We show this by applying Theorem 1 below with $\Delta = \Phi_1^{-1}\Phi_2$.

THEOREM 1. *The matrix $\mathcal{K}\Delta + \Delta\mathcal{K}^T$, where $\Delta = blkdiag(\alpha_1 I, \alpha_2 I, \ldots, \alpha_{N_t} I)$, $\alpha_1, \ldots, \alpha_{N_t} > 0$, $I \in \mathbb{R}^{n\times n}$, and $\mathcal{K}$ is as defined in* (2.8)*, is positive definite.*

Proof. We show that $\mathbf{w}^T(\mathcal{K}\Delta+\Delta\mathcal{K}^T)\mathbf{w} > 0$ for all $\mathbf{w} := \begin{bmatrix} \mathbf{w}_1^T & \mathbf{w}_2^T & \cdots & \mathbf{w}_{N_t-1}^T & \mathbf{w}_{N_t}^T \end{bmatrix}^T$ with $\mathbf{w}_1, \ldots, \mathbf{w}_{N_t} \in \mathbb{R}^n$, and

$$\Delta = \begin{bmatrix} \Delta_1 & & \\ & \ddots & \\ & & \Delta_{N_t} \end{bmatrix}, \quad \Delta_j \in \mathbb{R}^{n\times n}, \quad j = 1, \ldots, N_t,$$

with $\Delta_j = \alpha_j I$, $j = 1, \ldots, N_t$.

Using the symmetry of the mass and stiffness matrices M and K,

$$\mathcal{K}\Delta + \Delta\mathcal{K}^T = \begin{bmatrix} \Lambda_1 & -\Delta_1 M & & & & \\ -M\Delta_1 & \Lambda_2 & -\Delta_2 M & & & \\ & \ddots & \ddots & \ddots & & \\ & & -M\Delta_{N_t-2} & \Lambda_{N_t-1} & -\Delta_{N_t-1}M & \\ & & & -M\Delta_{N_t-1} & \Lambda_{N_t} \end{bmatrix},$$

where $\Lambda_j = (M + \tau K)\Delta_j + \Delta_j(M + \tau K)$ for $j = 1, \ldots, N_t$ and therefore by straightforward manipulation that

$$\begin{aligned} \mathbf{w}^T(\mathcal{K}\Delta + \Delta\mathcal{K}^T)\mathbf{w} &= \sum_{j=1}^{N_t} \mathbf{w}_j^T[M\Delta_j + \Delta_j M + \tau K\Delta_j + \tau\Delta_j K]\mathbf{w}_j \\ &\quad - \sum_{j=1}^{N_t-1} \mathbf{w}_j^T(M\Delta_j)\mathbf{w}_{j+1} - \sum_{j=2}^{N_t} \mathbf{w}_j^T(\Delta_{j-1}M)\mathbf{w}_{j-1} \\ &= 2\tau\sum_{j=1}^{N_t} \mathbf{w}_j^T(K\Delta_j)\mathbf{w}_j + \sum_{j=1}^{N_t-1} (\mathbf{w}_j - \mathbf{w}_{j+1})^T(M\Delta_j)(\mathbf{w}_j - \mathbf{w}_{j+1}) \\ &\quad + \mathbf{w}_1^T(M\Delta_1)\mathbf{w}_1 + \mathbf{w}_{N_t}^T(M\Delta_{N_t})\mathbf{w}_{N_t}, \end{aligned} \tag{3.5}$$

where we have used the facts that $M\Delta_j = \Delta_j M$ and $K\Delta_j = \Delta_j K$ for $j = 1, \ldots, N_t$, which are clear by the definition of Δ.

As we now have that $\mathbf{w}^T(\mathcal{K}\Delta+\Delta\mathcal{K}^T)\mathbf{w}$ is a sum of positive multiples of (symmetric positive definite) mass and stiffness matrices, we deduce that $\mathbf{w}^T(\mathcal{K}\Delta+\Delta\mathcal{K}^T)\mathbf{w} > 0$ and hence that $\mathcal{K}\Delta + \Delta\mathcal{K}^T$ is positive definite. ∎

Having demonstrated the properties required, we are now in a position to prove a result bounding the eigenvalues of $\widehat{S}^{-1}S$, where

$$\widehat{S} = \left(\mathcal{K} + \frac{1}{\sqrt{\beta}}\Phi_2\right)\Phi_1^{-1}\left(\mathcal{K} + \frac{1}{\sqrt{\beta}}\Phi_2\right)^T \tag{3.6}$$

and S is given by (3.4). To do this, we consider the Rayleigh quotient $R := \frac{\mathbf{v}^T S\mathbf{v}}{\mathbf{v}^T \widehat{S}\mathbf{v}}$. This may be written as

$$R = \frac{\mathbf{a}^T\mathbf{a} + \mathbf{b}^T\mathbf{b}}{\mathbf{a}^T\mathbf{a} + \mathbf{b}^T\mathbf{b} + \mathbf{a}^T\mathbf{b} + \mathbf{b}^T\mathbf{a}}, \tag{3.7}$$

where

$$\mathbf{a} = \Phi_1^{-1/2}\mathcal{K}^T\mathbf{v}, \quad \mathbf{b} = \frac{1}{\sqrt{\beta}}\Phi_1^{-1/2}\Phi_2\mathbf{v}.$$

Now, as $\mathbf{a}^T\mathbf{b} + \mathbf{b}^T\mathbf{a} = \frac{1}{\sqrt{\beta}}\mathbf{v}^T[\mathcal{K}\Phi_1^{-1}\Phi_2 + \Phi_2\Phi_1^{-1}\mathcal{K}^T]\mathbf{v} > 0$ due to Theorem 1 with $\Delta_j = \Phi_1^{-1}\Phi_2 = \Phi_2\Phi_1^{-1}$, it is clear from (3.7) that $R < 1$.

Further, showing that $R \geq \frac{1}{2}$ is a simple algebraic task, which requires only the fact that $\mathbf{b}^T\mathbf{b} > 0$ because of the positive definiteness of Φ_1 and Φ_2. (See [38] for further details.)

We have hence proved the next theorem.

THEOREM 2. *If S and $\widehat{S}$ are of the form stated in* (3.4) *and* (3.6) *respectively, with Φ_1, Φ_2 symmetric positive definite and $\Phi_1^{-1}\Phi_2 = blkdiag(\alpha_1 I, \alpha_2 I, \ldots, \alpha_{N_t} I)$, $\alpha_1, \ldots, \alpha_{N_t} > 0$, $I \in \mathbb{R}^{n\times n}$, then*

$$\lambda(\widehat{S}^{-1}S) \in \left[\frac{1}{2}, 1\right].$$

We note that Theorem 2 is an extension to a result discussed in [38] concerning convection-diffusion control.

We may now apply Theorem 2 with $\Phi_1 = \tau\mathcal{M}_{1/2}$ and $\Phi_2 = \tau\mathcal{M}$, as Φ_1 and Φ_2 defined in this way are clearly symmetric and positive definite and are such that $\Delta = \Phi_1^{-1}\Phi_2$ is symmetric positive definite and satisfies $\mathcal{M}\Delta = \Delta\mathcal{M}$. We therefore deduce that

$$\widehat{S} = \frac{1}{\tau}\left(\mathcal{K} + \frac{\tau}{\sqrt{\beta}}\mathcal{M}\right)\mathcal{M}_{1/2}^{-1}\left(\mathcal{K} + \frac{\tau}{\sqrt{\beta}}\mathcal{M}\right)^T \tag{3.8}$$

is an effective approximation to the Schur complement of the matrix system (2.7). We note that applying the inverses of the matrix $\mathcal{K} + \frac{\tau}{\sqrt{\beta}}\mathcal{M}$ and its transpose would not be feasible as this essentially means solving the PDE directly, which in itself is a computationally expensive task. Hence, for a practical algorithm we approximate $\widehat{S}$ using multigrid techniques for $\mathcal{K} + \frac{\tau}{\sqrt{\beta}}\mathcal{M}$ and its transpose, that is, we require a multigrid process for each of the diagonal blocks $M + \tau K + \frac{\tau}{\sqrt{\beta}}M \in \mathbb{R}^{n\times n}$. We apply a few cycles of such a multigrid process N_t times to approximate the inverse of $\mathcal{K} + \frac{\tau}{\sqrt{\beta}}\mathcal{M}$ and N_t times to approximate the inverse of $\left(\mathcal{K} + \frac{\tau}{\sqrt{\beta}}\mathcal{M}\right)^T$.

In conclusion, for an effective iterative method for solving (2.7), we recommend a MINRES method with a preconditioner of the form (3.2), with $\widehat{A}$ and $\widehat{S}$ as in (3.3) and (3.8). In section 4, we provide numerical results to demonstrate the effectiveness of our proposed preconditioner.

3.1.2. Minimizing J_1 with optimize-then-discretize. We now turn our attention to (2.9), the optimize-then-discretize formulation of (2.3) with $J(y,u) = J_1(y,u)$. Again, we may write this as a saddle point system of the form (2.1) with

$$A = \begin{bmatrix} \tau\mathcal{M}_0 & 0 \\ 0 & \beta\tau\mathcal{M}_{1/2} \end{bmatrix}, \quad B = \begin{bmatrix} \mathcal{K} & -\tau\mathcal{M} \end{bmatrix}.$$

We note that the $(1,1)$-block of this system, A, is not invertible, due to the rank-deficiency of $\mathcal{M}_0$, so when prescribing an approximation for a preconditioner, we recommend considering a perturbation of the matrix $\mathcal{M}_0$

$$\mathcal{M}_0^\gamma = \begin{bmatrix} M & & & & \\ & M & & & \\ & & \ddots & & \\ & & & M & \\ & & & & \gamma M \end{bmatrix}$$

for some constant γ such that $0 < \gamma \ll 1$, and taking as our approximation to A the following:

$$\widehat{A} = \begin{bmatrix} \tau\widehat{\mathcal{M}}_0 & 0 \\ 0 & \beta\tau\widehat{\mathcal{M}}_{1/2} \end{bmatrix}, \tag{3.9}$$

where $\widehat{\mathcal{M}}_0$ and $\widehat{\mathcal{M}}_{1/2}$ denote approximations to $\mathcal{M}_0^\gamma$ and $\mathcal{M}_{1/2}$, generated by using Chebyshev semi-iteration in the case of consistent mass matrices, or, in the case of lumped mass matrices, themselves.

Now, due to the noninvertibility of $\mathcal{M}_0$, the Schur complement of the matrix system (2.9) does not exist. Therefore it is less obvious what the $(2,2)$-block of our block diagonal preconditioner of the form (3.2) should be. The heuristic we use is to examine the perturbed saddle point system $\begin{bmatrix} \widehat{A} & B^T \\ B & 0 \end{bmatrix}$ and consider the Schur complement of this matrix system. This is given by the quantity

$$\tilde{S} := \frac{1}{\tau}\mathcal{K}\widehat{\mathcal{M}}_0^{-1}\mathcal{K}^T + \frac{\tau}{\beta}\mathcal{M}\mathcal{M}_{1/2}^{-1}\mathcal{M}.$$

Now, by simple manipulation, we observe that

$$\tilde{S} = \frac{1}{\tau}\mathcal{K}\widehat{\mathcal{M}}_0^{-1}\mathcal{K}^T + \frac{\tau}{\beta}\Gamma_1\widehat{\mathcal{M}}_0^{-1}\Gamma_1,$$

where

$$\Gamma_1 = \begin{bmatrix} \sqrt{2}M & & & & \\ & M & & & \\ & & \ddots & & \\ & & & M & \\ & & & & \sqrt{2\gamma}M \end{bmatrix}.$$

By applying Theorem 2 with $\Phi_1 = \tau\widehat{\mathcal{M}}_0$ and $\Phi_2 = \tau\Gamma_1$, we therefore deduce that

$$\widehat{S} = \frac{1}{\tau}\left(\mathcal{K} + \frac{\tau}{\sqrt{\beta}}\Gamma_1\right)\mathcal{M}_0^{-1}\left(\mathcal{K} + \frac{\tau}{\sqrt{\beta}}\Gamma_1\right)^T \tag{3.10}$$

satisfies

$$\lambda(\widehat{S}^{-1}\tilde{S}) \in \left[\frac{1}{2}, 1\right],$$

which tells us that $\widehat{S}$ is a good Schur complement approximation to the perturbed matrix system we have considered. As the matrix system (2.9) is very similar in structure to this perturbed system, it seems that this would also be a pragmatic choice for the $(2,2)$-block of our block diagonal preconditioner for this system.

Therefore, within the MINRES algorithm for solving (2.9), we again recommend a preconditioner of the form (3.2) with $\widehat{A}$ and $\widehat{S}$ as in (3.9) and (3.10). The numerical results of section 4 demonstrate that this is indeed an effective approach.

3.2. Time-independent Neumann boundary control. We now consider preconditioning the system (2.11), which arises when solving the time-independent Poisson boundary control problem. If we write the saddle point system in the form (2.1) with

$$A = \begin{bmatrix} M & 0 \\ 0 & \beta M_b \end{bmatrix}, \quad B = \begin{bmatrix} K & -N \end{bmatrix},$$

then constructing an approximation $\widehat{A}$ to the $(1,1)$-block A is relatively straightforward, as we treat both mass matrices M and M_b as before. However, an issue arises when we consider the effective approximation of the Schur complement of (2.11)

$$S = KM^{-1}K + \frac{1}{\beta}NM_b^{-1}N^T.$$

Because of the rank-deficiency of the $\frac{1}{\beta}NM_b^{-1}N^T$ term of the Schur complement, it is not as simple to find a clean and easy-to-invert approximation $\widehat{S}$ to S such that the eigenvalues of $\widehat{S}^{-1}S$ may be pinned down into an interval independent of both h and β, as for the distributed control case in section 2.2. We therefore seek an approximation which is robust for a range of h and β. We first wish to motivate our choices before analyzing them in more detail.

We assume now that all mass matrices are lumped. It is then easy to see that $NM_b^{-1}N^T$ is a diagonal matrix with nonzero entries on the diagonal for every boundary node. For simplicity we assume the degrees of freedom are ordered in such a way that the nodes located on the boundary can be found in the lower right corner of $NM_b^{-1}N^T$, i.e.,

$$NM_b^{-1}N^T = \begin{bmatrix} 0 & 0 \\ 0 & M_b \end{bmatrix}.$$

Now our task is to approximate the Schur complement S via

$$\widehat{S} = \left(K + \frac{1}{\sqrt{\beta}}\widehat{M}\right) M^{-1} \left(K + \frac{1}{\sqrt{\beta}}\widehat{M}\right)$$

for some matrix $\widehat{M}$ in such a way that the structure of the original Schur complement is maintained as much as possible. If we look at the last equation we see this gives

$$\widehat{S} = KM^{-1}K^T + \frac{1}{\beta}\widehat{M}M^{-1}\widehat{M} + \frac{1}{\sqrt{\beta}}\left(KM^{-1}\widehat{M} + \widehat{M}M^{-1}K\right).$$

We now look at the terms separately. The first one is part of the original Schur complement. The second one needs to be looked at more carefully. Hence

$$\begin{bmatrix} 0 & 0 \\ 0 & \alpha M_b \end{bmatrix} \begin{bmatrix} M_{y,i}^{-1} & 0 \\ 0 & M_{y,b}^{-1} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & \alpha M_b \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & \alpha^2 M_b M_{y,b}^{-1} M_b \end{bmatrix}$$

with i and b denoting interior and boundary, respectively, and for some constant α. This tells us now that if

$$\alpha^2 M_b M_{y,b}^{-1} M_b \approx M_b,$$

we have found a good approximation to the Schur complement of the original matrix, which can be evaluated efficiently. A simplification will now motivate our choice of α as, if we approximate $M_b = hI_b$ (where I_b is the identity matrix of dimension equal to the number of boundary nodes) and $M_{y,b} = h^2 I$, we obtain that

$$\alpha^2 M_b M_{y,b}^{-1} M_b = M_b \iff \alpha^2 h h^{-2} h I = \alpha^2 I \approx hI, \tag{3.11}$$

and hence a good choice for α seems to be $\alpha = \sqrt{h}$. As a result, our recommended Schur complement approximation is now defined as

$$\widehat{S}_1 = \left(K + \sqrt{\frac{h}{\beta}}M_\Gamma\right) M^{-1} \left(K + \sqrt{\frac{h}{\beta}}M_\Gamma\right),$$

i.e., the matrix $\widehat{M}$ introduced earlier is given by $\sqrt{h}M_\Gamma$. We note that because of the diagonal nature of the mass matrices the matrix $M_\Gamma = NM_b^{-1}N^T$ is simple to evaluate. Another choice with a similar motivation is given by

$$\widehat{S}_2 = \left(K + \sqrt{\frac{h}{\beta}}M_\Gamma\right) (h\widehat{M}_\Gamma)^{-1} \left(K + \sqrt{\frac{h}{\beta}}M_\Gamma\right).$$

Here $\widehat{M}_\Gamma$ is given by the matrix M_b in the boundary components and a small scalar of order h for all nodes corresponding to the degrees of freedom on the interior, i.e.,

$$\widehat{M}_\Gamma = M_\Gamma + hI_i,$$

with I_i a diagonal matrix with ones on the diagonal for all interior degrees of freedom and zeros elsewhere. We now wish to analyze these two preconditioners in more detail by considering the eigenvalue distributions of $\widehat{S}_1^{-1}S$ and $\widehat{S}_2^{-1}S$. Our analysis is based on the two-dimensional problem, however it can be easily extended to the three-dimensional case.

Eigenvalues of $\widehat{S}_1^{-1}S$. Here we must consider the Rayleigh quotient

$$\begin{aligned}\frac{\mathbf{v}^T S\mathbf{v}}{\mathbf{v}^T\widehat{S}_1\mathbf{v}} &= \frac{\mathbf{v}^T KM^{-1}K\mathbf{v} + \frac{1}{\beta}\mathbf{v}^T M_\Gamma\mathbf{v}}{\mathbf{v}^T KM^{-1}K\mathbf{v} + \frac{h}{\beta}\mathbf{v}^T M_\Gamma M^{-1}M_\Gamma\mathbf{v} + \sqrt{\frac{h}{\beta}}\mathbf{v}^T\left[M_\Gamma M^{-1}K + KM^{-1}M_\Gamma\right]\mathbf{v}} \\ &= \frac{\mathbf{v}^T KM^{-1}K\mathbf{v} + \mathbf{v}^T\left(\frac{1}{\beta}M_\Gamma\right)\mathbf{v}}{\mathbf{v}^T KM^{-1}K\mathbf{v} + \frac{h}{\beta}\mathbf{v}^T M_\Gamma M^{-1}M_\Gamma\mathbf{v} + 2\sqrt{\frac{h}{\beta}}\mathbf{v}^T M_\Gamma M^{-1}K\mathbf{v}},\end{aligned}$$

which will provide us with the eigenvalues of $\widehat{S}_1^{-1}S$.

If $\mathbf{v} \in \text{null}(M_\Gamma)$, then $\frac{\mathbf{v}^T S \mathbf{v}}{\mathbf{v}^T \widehat{S}_1 \mathbf{v}} = 1$. If not, then we can write the above also as

$$\frac{\mathbf{v}^T S \mathbf{v}}{\mathbf{v}^T \widehat{S}_1 \mathbf{v}} = \frac{1}{\frac{\mathbf{v}^T K M^{-1} K \mathbf{v} + \frac{h}{\beta} \mathbf{v}^T M_\Gamma M^{-1} M_\Gamma \mathbf{v}}{\mathbf{v}^T K M^{-1} K \mathbf{v} + \frac{1}{\beta} \mathbf{v}^T M_\Gamma \mathbf{v}} + \frac{2\sqrt{\frac{h}{\beta}} \mathbf{v}^T M_\Gamma M^{-1} K \mathbf{v}}{\mathbf{v}^T K M^{-1} K \mathbf{v} + \frac{1}{\beta} \mathbf{v}^T M_\Gamma \mathbf{v}}}. \tag{3.12}$$

Using the fact that $M_\Gamma \left(\frac{1}{h} M\right)^{-1} M_\Gamma = h M_\Gamma M^{-1} M_\Gamma$ and M_Γ are spectrally equivalent, we can see that

$$0 < \frac{\mathbf{v}^T K M^{-1} K \mathbf{v} + \frac{h}{\beta} \mathbf{v}^T M_\Gamma M^{-1} M_\Gamma \mathbf{v}}{\mathbf{v}^T K M^{-1} K \mathbf{v} + \frac{1}{\beta} \mathbf{v}^T M_\Gamma \mathbf{v}} =: D_1 = \mathcal{O}(1),$$

where D_1 is a mesh and β-independent constant.

We now examine the term

$$\frac{2\sqrt{\frac{h}{\beta}} \mathbf{v}^T M_\Gamma M^{-1} K \mathbf{v}}{\mathbf{v}^T K M^{-1} K \mathbf{v} + \frac{1}{\beta} \mathbf{v}^T M_\Gamma \mathbf{v}} =: \frac{T_1}{T_2},$$

in particular its maximum and minimum values, more carefully. We assume now that $M \approx h^2 I$ and $M_\Gamma \approx hI$, ignoring all multiplicative constants. Furthermore, we note that the eigenvalues of K are within the interval $\left[c_K h^2, C_K\right]$, where c_K and C_K are constants independent of h and β (apart from a single zero eigenvalue with a corresponding eigenvector of ones—this corresponds to an arbitrary constant being a solution of the continuous Neumann problem for Poisson's equation).

As we work with lumped mass matrices throughout our work on Neumann boundary control, we observe that $T_1 \geq 0$, as it relates to a positive constant multiplied by the product of two matrices ($M_\Gamma M^{-1}$, which we have assumed to be approximately $h^{-1} I$, and K, which is symmetric positive definite). We also note that T_2 must be strictly positive.[1]

We now consider the maximum and minimum values of $\frac{T_1}{T_2}$. We consider the maximum such value by writing

$$\frac{T_1}{T_2} = \frac{\beta^{-1/2} h^{1/2} h^1 h^{-2} c}{h^{-2} c^2 + \beta^{-1} h} = \frac{\beta^{-1/2} h^{-1/2} c}{h^{-2}(c^2 + \beta^{-1} h^3)} = \frac{ac}{c^2 + a^2}$$

with $a = h^{3/2} \beta^{-1/2}$ and c corresponding to the relevant eigenvalue of K. Here, both a and c are positive. Therefore, in this case, $\frac{ac}{c^2+a^2} \leq \frac{1}{2}$ by straightforward algebraic manipulation. This means that the denominator in (3.12) will be bounded above by a constant independent of h, β, and τ, as both terms are of $\mathcal{O}(1)$. This gives us a lower bound for $\lambda_{\min}$.

As T_1 and T_2 are both nonnegative, we may write that $\frac{T_1}{T_2} \geq 0$ and hence that $\frac{\mathbf{v}^T S \mathbf{v}}{\mathbf{v}^T \widehat{S}_1 \mathbf{v}} \geq \frac{1}{D_1}$, giving us an upper bound for $\lambda_{\max}$.

Putting our analysis together, and reinstating multiplicative constants, we conclude that

$$\lambda_{\min}(\widehat{S}_1^{-1} S) = c_1, \quad \lambda_{\max}(\widehat{S}_1^{-1} S) = C_1,$$

where c_1 and C_1 are positive constants independent of h, β, and τ.

[1] This may be argued as follows. Both $\mathbf{v}^T K M^{-1} K \mathbf{v}$ and $\frac{1}{\beta} \mathbf{v}^T M_\Gamma \mathbf{v}$ are nonnegative terms. The former will be strictly positive unless $\mathbf{v}$ is the vector of ones, which corresponds to the zero eigenvalue of K. In this case, it is clear that the $\mathbf{v}^T M_\Gamma \mathbf{v}$ term will be strictly positive, as none of the entries of M_Γ are negative. So for each $\mathbf{v}$, at least one term will be strictly positive.

Eigenvalues of $\widehat{S}_2^{-1}S$. We may carry out a similar analysis for the approximation $\widehat{S}_2$ of S by considering the Rayleigh quotient

$$\frac{\mathbf{v}^T S\mathbf{v}}{\mathbf{v}^T\widehat{S}_2\mathbf{v}} = \frac{\mathbf{v}^T K M^{-1}K\mathbf{v} + \mathbf{v}^T\left(\frac{1}{\beta}M_\Gamma\right)\mathbf{v}}{\mathbf{v}^T K(h\widehat{M}_\Gamma)^{-1}K\mathbf{v} + \frac{h}{\beta}\mathbf{v}^T M_\Gamma(h\widehat{M}_\Gamma)^{-1}M_\Gamma\mathbf{v} + 2\sqrt{\frac{h}{\beta}}\mathbf{v}^T M_\Gamma(h\widehat{M}_\Gamma)^{-1}K\mathbf{v}},$$

and writing that $M \approx h^2 I$, $\widehat{M}_\Gamma \approx hI$, and $M_\Gamma \approx \text{blkdiag}(0, hI_b)$.

Proceeding as we did for the analysis of $\widehat{S}_1$, we obtain that

$$\lambda_{\min}(\widehat{S}_2^{-1}S) = c_2, \quad \lambda_{\max}(\widehat{S}_2^{-1}S) = C_2,$$

where c_2 and C_2 are positive constants independent of h, β, and τ, provided we use lumped mass matrices.

We emphasize that due to the rank-deficient nature of the $\frac{1}{\beta}NM_b^{-1}N^T$ term of the Schur complement S, it is more difficult to obtain a complete picture of the eigenvalue distributions of $\widehat{S}_1^{-1}S$ and $\widehat{S}_2^{-1}S$ than for the preconditioned Schur complement in the distributed control case. Consequently, the bounding of $\lambda(\widehat{S}_1^{-1}S)$ and $\lambda(\widehat{S}_2^{-1}S)$ by constants of $\mathcal{O}(1)$ is less descriptive than the more specific bound outlined for distributed control in [37] and discussed in section 2.2.

However, the conclusion that the eigenvalues of $\widehat{S}_1^{-1}S$ and $\widehat{S}_2^{-1}S$ are certainly real and bounded above and below by constants of $\mathcal{O}(1)$, independently of h, β, and τ, indicates that either S_1 or S_2 should serve as an effective approximation of S—a hypothesis which is verified by the numerical results presented in section 4. We note that in the above analysis, we have assumed that lumped mass matrices are being used; however, numerical tests indicate that we still obtain a clean bound when using consistent mass matrices.

3.3. Time-dependent Neumann boundary control. In the case of the time-dependent boundary control problem, we are interested in approximating the Schur complement

$$S = \frac{1}{\tau}\mathcal{K}\mathcal{M}_{1/2}^{-1}\mathcal{K}^T + \frac{\tau}{\beta}\mathcal{N}\mathcal{M}_{1/2,b}^{-1}\mathcal{N}^T \tag{3.13}$$

of the saddle point matrix $\mathcal{A}$. We want to approximate the above by

$$\widehat{S}_3 = \tau^{-1}\left(\mathcal{K} + \frac{\tau}{\sqrt{\beta}}\widehat{\mathcal{M}}\right)\mathcal{M}_{1/2}^{-1}\left(\mathcal{K}^T + \frac{\tau}{\sqrt{\beta}}\widehat{\mathcal{M}}\right), \tag{3.14}$$

and for this to be a good approximation the choice of $\widehat{\mathcal{M}}$ is again crucial. We recall that we assumed $\mathcal{M}_{1/2,b}$ to be a block diagonal matrix of lumped boundary mass matrices and also that $\mathcal{M}_{1/2}$ consists of lumped mass matrices over the domain Ω. Hence the first term in (3.14) is given by $\tau^{-1}\mathcal{K}\mathcal{M}_{1/2}^{-1}\mathcal{K}^T$, which means that the first term in the Schur complement (3.13) is well represented in our approximation. We then obtain the next term from (3.14) as

$$\frac{\tau^{-1}\tau\tau}{\sqrt{\beta}\sqrt{\beta}}\widehat{\mathcal{M}}\mathcal{M}_{1/2}^{-1}\widehat{\mathcal{M}} = \frac{\tau}{\beta}\widehat{\mathcal{M}}\mathcal{M}_{1/2}^{-1}\widehat{\mathcal{M}}.$$

To understand how this approximates $\mathcal{N}\mathcal{M}_{1/2,b}^{-1}\mathcal{N}^T$, we need to study the structure of both matrix products more carefully. We recall that $\mathcal{M}_{1/2,b} = \text{blkdiag}(\frac{1}{2}M_b, M_b, \ldots,$

$M_b, \frac{1}{2}M_b$) and that with some abuse of notation $\mathcal{N} = \text{blkdiag}_{rec}(N, \ldots, N)$, giving for the overall structure

$$\mathcal{N}\mathcal{M}_{1/2,b}^{-1}\mathcal{N}^T = \begin{bmatrix} 2NM_b^{-1}N^T & & & & & \\ & NM_b^{-1}N^T & & & & \\ & & \ddots & & & \\ & & & NM_b^{-1}N^T & & \\ & & & & 2NM_b^{-1}N^T \end{bmatrix}.$$

We see that as $\mathcal{M}_{1/2} = \text{blkdiag}\left(\frac{1}{2}M, M, \ldots, M, \frac{1}{2}M\right)$ and $\widehat{M} = \text{blkdiag}(\widehat{M}, \ldots, \widehat{M})$, the structure of the large problem looks as follows:

$$\widehat{\mathcal{M}}\mathcal{M}_{1/2}^{-1}\widehat{\mathcal{M}} = \begin{bmatrix} 2\widehat{M}M^{-1}\widehat{M} & & & & & \\ & \widehat{M}M^{-1}\widehat{M} & & & & \\ & & \ddots & & & \\ & & & \widehat{M}M^{-1}\widehat{M} & & \\ & & & & 2\widehat{M}M^{-1}\widehat{M} \end{bmatrix}.$$

This indicates that it is important for $\widehat{M}M^{-1}\widehat{M} \approx NM_b^{-1}N^T$, which we split up even further now. Consider an ordering of the degrees of freedom on the boundary and in the interior as before,

$$\widehat{M}M^{-1}\widehat{M} = \begin{bmatrix} 0 & 0 \\ 0 & \alpha M_b \end{bmatrix} \begin{bmatrix} M_{y,i}^{-1} & 0 \\ 0 & M_{y,b}^{-1} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & \alpha M_b \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & \alpha^2 M_b M_{y,b}^{-1} M_b \end{bmatrix},$$

and now note that

$$NM_b^{-1}N^T = \begin{bmatrix} 0 & 0 \\ 0 & M_b \end{bmatrix},$$

where $M_{y,i}$ and $M_{y,b}$ denote the splitting of the mass matrix M into its interior and boundary parts, respectively. Similar to before, we can show that $\alpha = \sqrt{h}$ is a good choice. A choice not very different from the above is given by the approximation

$$\widehat{S}_4 = \tau^{-1}\left(\mathcal{K} + \tau\sqrt{\frac{h}{\beta}}\widehat{\mathcal{M}}\right)(h\widehat{\mathcal{M}}_\Gamma)^{-1}\left(\mathcal{K} + \tau\sqrt{\frac{h}{\beta}}\widehat{\mathcal{M}}\right)^T, \tag{3.15}$$

where $\widehat{\mathcal{M}}_\Gamma$ consists of block diagonal matrices that have the boundary mass matrix for the boundary nodes and a suitably scaled identity matrix for the interior nodes. (See also the time-independent case.)

Eigenvalues of $\widehat{S}_4^{-1}S$. We now search for the eigenvalues of $\widehat{S}_4^{-1}S$, where

$$\widehat{S}_4 = \tau^{-1}\left(\mathcal{K} + \tau\sqrt{\frac{h}{\beta}}\widehat{\mathcal{M}}\right)(h\widehat{\mathcal{M}}_\Gamma)^{-1}\left(\mathcal{K} + \tau\sqrt{\frac{h}{\beta}}\widehat{\mathcal{M}}\right)^T, \tag{3.16}$$

by considering the Rayleigh quotient

$$\frac{\mathbf{v}^T S\mathbf{v}}{\mathbf{v}^T\widehat{S}_4\mathbf{v}} = \frac{\tau^{-1}\mathbf{v}^T\mathcal{K}\mathcal{M}_{1/2}^{-1}\mathcal{K}^T\mathbf{v} + \tau\beta^{-1}\mathbf{v}^T\widehat{\mathcal{M}}\mathbf{v}}{\tau^{-1}\mathbf{v}^T\mathcal{K}(h\widehat{\mathcal{M}}_\Gamma)^{-1}\mathcal{K}\mathbf{v} + \frac{\tau h}{\beta}\mathbf{v}^T\widehat{\mathcal{M}}(h\widehat{\mathcal{M}}_\Gamma)^{-1}\widehat{\mathcal{M}}\mathbf{v} + 2\sqrt{\frac{h}{\beta}}\mathbf{v}^T\mathcal{K}(h\widehat{\mathcal{M}}_\Gamma)^{-1}\widehat{\mathcal{M}}\mathbf{v}},$$

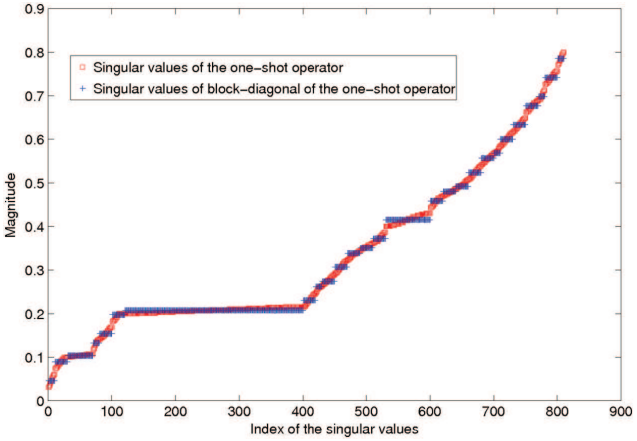


FIG. 3.1. *Singular values of $\mathcal{L}$ and $\mathcal{K}$ for a small example.*

using the fact that $\widehat{\mathcal{M}}_\Gamma = \mathcal{N}\mathcal{M}_{1/2,b}^{-1}\mathcal{N}^T$. Assuming that $\mathbf{v} \in \text{null}(\widehat{\mathcal{M}})$, we obtain that

$$\frac{\mathbf{v}^T S \mathbf{v}}{\mathbf{v}^T \widehat{S}_4 \mathbf{v}} = \mathcal{O}(1).$$

So we now consider the case where $\mathbf{v}$ is not in this nullspace; we then examine the term

$$\frac{1}{\frac{\tau^{-1}\mathbf{v}^T\mathcal{K}(h\widehat{\mathcal{M}}_\Gamma)^{-1}\mathcal{K}\mathbf{v}+\frac{\tau h}{\beta}\mathbf{v}^T\widehat{\mathcal{M}}(h\widehat{\mathcal{M}}_\Gamma)^{-1}\widehat{\mathcal{M}}\mathbf{v}}{\tau^{-1}\mathbf{v}^T\mathcal{K}\mathcal{M}_{1/2}^{-1}\mathcal{K}^T\mathbf{v}+\tau\beta^{-1}\mathbf{v}^T\widehat{\mathcal{M}}\mathbf{v}} + \sqrt{\frac{h}{\beta}}\frac{\mathbf{v}^T\left(\mathcal{K}(h\widehat{\mathcal{M}}_\Gamma)^{-1}\widehat{\mathcal{M}}+\widehat{\mathcal{M}}(h\widehat{\mathcal{M}}_\Gamma)^{-1}\mathcal{K}^T\right)\mathbf{v}}{\tau^{-1}\mathbf{v}^T\mathcal{K}\mathcal{M}_{1/2}^{-1}\mathcal{K}^T\mathbf{v}+\tau\beta^{-1}\mathbf{v}^T\widehat{\mathcal{M}}\mathbf{v}}}}.$$

So if we now assume (neglecting constants for now) that $h\widehat{\mathcal{M}}_\Gamma \approx \mathcal{M}_{1/2} \approx h^2 I$ and $\widehat{\mathcal{M}} \approx \widehat{\mathcal{M}}(h\widehat{\mathcal{M}}_\Gamma)^{-1}\widehat{\mathcal{M}} \approx hI$, we see that

$$\frac{\tau^{-1}\mathbf{v}^T\mathcal{K}(h\widehat{\mathcal{M}}_\Gamma)^{-1}\mathcal{K}\mathbf{v} + \frac{\tau h}{\beta}\mathbf{v}^T\widehat{\mathcal{M}}(h\widehat{\mathcal{M}}_\Gamma)^{-1}\widehat{\mathcal{M}}\mathbf{v}}{\tau^{-1}\mathbf{v}^T\mathcal{K}\mathcal{M}_{1/2}^{-1}\mathcal{K}^T\mathbf{v} + \tau\beta^{-1}\mathbf{v}^T\widehat{\mathcal{M}}\mathbf{v}} = \mathcal{O}(1).$$

In order to simplify the analysis at this stage we simply assume that $\mathcal{K}$ is approximated by its block diagonal, i.e., $\mathcal{L} \approx \mathcal{K}$ (see Figure 3.1). We use this to approximate the above by

$$\sqrt{\frac{h}{\beta}}\frac{\mathbf{v}^T\left(\mathcal{L}(h\widehat{\mathcal{M}}_\Gamma)^{-1}\widehat{\mathcal{M}} + \widehat{\mathcal{M}}(h\widehat{\mathcal{M}}_\Gamma)^{-1}\mathcal{L}\right)\mathbf{v}}{\tau^{-1}\mathbf{v}^T\mathcal{L}\mathcal{M}_{1/2}^{-1}\mathcal{L}\mathbf{v} + \tau\beta^{-1}\mathbf{v}^T\widehat{\mathcal{M}}\mathbf{v}} =: \frac{T_1}{T_2}.$$

We may proceed as in section 3.2 for the time-independent boundary control case to obtain (neglecting constants)

$$\frac{T_1}{T_2} = \frac{h^{1/2}\beta^{-1/2}h^{-1}c}{\tau^{-1}h^{-2}c^2 + \tau\beta^{-1}h} = \frac{\beta^{-1/2}h^{-1/2}c}{h^{-2}\tau^{-1}(c^2 + \tau^2\beta^{-1}h^3)} = \frac{\tau\beta^{-1/2}h^{3/2}c}{c^2 + \tau^2\beta^{-1}h^3} = \frac{ac}{c^2 + a^2} \leq \frac{1}{2}$$

with $a = \tau\beta^{-1/2}h^{3/2}$ and $c \in \left[c_K\tau h^2 + c_M h^2, C_k\tau + C_M h^2\right]$. This shows that the results for the time-independent case can be used here as well. For the minimum value of $\frac{T_1}{T_2}$, we may apply a similar analysis as in the case of $\widehat{S}_1^{-1}S$ and working once more with lumped mass matrices. We obtain that (reintroducing constants)

$$\lambda_{\min}(\widehat{S}_4^{-1}S) = c_4, \quad \lambda_{\max}(\widehat{S}_4^{-1}S) = C_4,$$

where c_4 and C_4 are positive constants independent of h, β and τ.

A similar analysis can be carried out for $\widehat{S}_3^{-1}S$. As for the time-independent case, it is more difficult to develop a complete picture of the eigenvalue distribution of the preconditioned Schur complement than for the distributed control case; however, it is useful to see that we may bound the eigenvalues by constants of $\mathcal{O}(1)$ independently of the parameters h, β, and τ. Indeed, the results shown in section 4 show that the performance for the preconditioners for the time-dependent and time-independent boundary control problems is quite similar, and we find that both approximations $\widehat{S}_3$ and $\widehat{S}_4$ are effective for this problem for a wide range of parameters.

3.4. The subdomain case. We now wish to address the case when the desired state is only given on a subdomain Ω_1 of Ω. The saddle point system is then defined by

$$A = \begin{bmatrix} \tau\mathcal{M}_{1/2}^{(1)} & 0 \\ 0 & \beta\tau\mathcal{M}_{1/2} \end{bmatrix}, \quad B = \begin{bmatrix} \mathcal{K} & -\tau\mathcal{M} \end{bmatrix},$$

and we note that the matrix A is only positive semidefinite as the matrix $\mathcal{M}_{1/2}^{(1)}$ is semidefinite. However, we wish to obtain an invertible approximation of A, as well as an effective Schur complement approximation, as in previous sections. For that purpose we introduce a parameter $\gamma \in \mathbb{R}$ such that the matrix $\mathcal{M}_{1/2}^{(1,\gamma)} = \text{blkdiag}\left(\frac{1}{2}M_1^\gamma, M_1^\gamma, \ldots, M_1^\gamma, \frac{1}{2}M_1^\gamma\right)$ with M_1^γ defined as

$$(M_1^\gamma)_{\Omega\setminus\Omega_1} = \gamma I \quad \text{or} \quad (M_1^\gamma)_{\Omega\setminus\Omega_1} = \gamma M_{\Omega\setminus\Omega_1}.$$

Note that we use the same notation for the small parameter, namely, γ, dealing with the zero parts of the $(1,1)$-block and believe it will be clear from the context what γ represents. The $(1,1)$-block of this perturbed problem may now be approximated by $\widehat{A} = \text{blkdiag}(\tau\widehat{\mathcal{M}}_{1/2}^{(1,\gamma)}, \beta\tau\widehat{\mathcal{M}}_{1/2})$, where $\widehat{\mathcal{M}}_{1/2}^{(1,\gamma)}$ now denotes the relevant approximation of mass matrices (Chebyshev semi-iteration or diagonal solves) within the matrix $\mathcal{M}_{1/2}^{(1,\gamma)}$. The Schur complement of this perturbed problem that we now wish to approximate is given by

$$\widetilde{S} = \frac{1}{\tau}\mathcal{K}(\mathcal{M}_{1/2}^{(1,\gamma)})^{-1}\mathcal{K}^T + \frac{\tau}{\beta}\mathcal{M}\mathcal{M}_{1/2}^{-1}\mathcal{M}.$$

Again our goal is to derive an approximation to the Schur complement that exhibits robustness with respect to the regularization parameter. For this we consider

$$\widehat{S} = \frac{1}{\tau}(\mathcal{K} + \widehat{\mathcal{M}})(\mathcal{M}_{1/2}^{(1,\gamma)})^{-1}(\mathcal{K} + \widehat{\mathcal{M}})^T,$$

where we have to define $\widehat{\mathcal{M}}$. Ideally, we have agreement between the terms $\frac{1}{\tau}\widehat{\mathcal{M}}(\mathcal{M}_{1/2}^{(1,\gamma)})^{-1}\widehat{\mathcal{M}} \approx \frac{\tau}{\beta}\mathcal{M}\mathcal{M}_{1/2}^{-1}\mathcal{M}$. Assuming now that all mass matrices are lumped we can give an elementwise description of what we wish to achieve, i.e.,

$$\widehat{m}_{ii}^2 = \frac{\tau^2}{\beta}\left(m^{(1,\gamma)}\right)_{ii} m_{ii},$$

that is,

$$\widehat{m}_{ii} = \frac{\tau}{\sqrt{\beta}} \sqrt{\left(m^{(1,\gamma)}\right)_{ii}} \sqrt{m_{ii}}. \tag{3.17}$$

We now have to distinguish between indices i that represent degrees of freedom within Ω_1 or in $\Omega \setminus \Omega_1$. In more detail,

$$\left(m^{(1,\gamma)}\right)_{ii} = \begin{cases} m_{ii} & \text{if } i \in \Omega_1, \\ \gamma & \text{otherwise.} \end{cases} \tag{3.18}$$

We have now established an expression for the elements of $\widehat{\mathcal{M}}$ in the case of the distributed control problem. We find that the resulting Schur complement approximation works well in practice—we demonstrate this once again with numerical results in section 4.

Choice of γ. We now explain how we select in practice the "perturbation parameter" γ that we have utilized in previous sections. We start by deriving the parameter γ for the case when optimize-then-discretize is used for the distributed control problem. We assume that we want both terms of the Schur complement

$$S = \mathcal{K}\widehat{\mathcal{M}}_0^{-1}\mathcal{K}^T + \tau\beta^{-1}\mathcal{M}_2$$

with $\widehat{\mathcal{M}}_0 = \text{blkdiag}(M, \ldots, M, \gamma M)$, $\mathcal{M}_2 = \text{blkdiag}(2M, M. \ldots, M, 2M)$ to be "balanced" (see [4, 52]). We simplify this task by replacing $\mathcal{K}$ by its block diagonal $\mathcal{L} := \text{blkdiag}(L, \ldots, L)$, where $L = M + \tau K$. We now wish to balance the terms in this new approximation with a particular focus on the parameter γ, i.e.,

$$\widehat{S} = \mathcal{L}\widehat{\mathcal{M}}_0^{-1}\mathcal{L}^T + \tau\beta^{-1}\mathcal{M}_2.$$

Comparing the blocks in $\widehat{S}$ that involve γ, we obtain

$$\gamma^{-1}h^{-2}L^2 \approx \tau\beta^{-1}h^2 I, \tag{3.19}$$

using the approximation $M = h^2 I$ for a two-dimensional problem. In this heuristic, we want to balance the smallest eigenvalues of both terms; for $L^2 = \tau^2 K^2 + \tau KM + \tau MK + M^2$ these will be of the order $\tau^2 h^4$ (neglecting constants). In order for γ to balance both terms in (3.19), we get

$$\gamma^{-1}h^{-2}\tau^2 h^4 \approx \tau\beta^{-1}h^2$$

and therefore that

$$\tau\beta \approx \gamma. \tag{3.20}$$

Note that the above heuristic holds for the two-dimensional case. In complete analogy, we can derive that

$$\tau\beta \approx \gamma \tag{3.21}$$

is also a good choice for problems in three dimensions. If one wants to balance the largest eigenvalues in both terms the parameter γ might not be small, depending on the choice of τ and β. In a very similar way we can derive a heuristic for the parameter

γ in the subdomain case. To solve the distributed control problem we replace the zero entries by γ to give

$$\gamma^{-1}\tau^2 h^4 \approx \tau\beta^{-1}h^2 \Rightarrow \gamma = \tau\beta h^2. \tag{3.22}$$

Rees and Greif [41] also introduce a similar parameter γ that is part of a perturbation of the $(1,1)$-block of a saddle point problem coming from the treatment of a quadratic program using interior point methods. They construct a preconditioner with an augmented $(1,1)$-block, i.e., $A+\gamma^{-1}BB^T$, using the classical saddle point notation, where their parameter γ is chosen to balance the two summands A and BB^T, similar to our heuristic above.

4. Numerical results. The results presented in this section are based on an implementation of the above described algorithms within the deal.II [2] framework using $Q1$ finite elements. For the AMG preconditioner, we used the Trilinos ML package [15] that implements a smoothed aggregation AMG. Within the AMG we typically used 10 steps of a Chebyshev smoother in combination with the application of two V-cycles. Our implementation of MINRES was taken from [12] and was stopped with a tolerance of 10^{-4} for the relative pseudoresidual. Our experiments are performed for $T = 1$ and $\tau = 0.05$, i.e., 20 time-steps. We consider homogeneous Dirichlet conditions for distributed control problems, though we are of course not limited to them, and also a zero forcing term $f = 0$ for Neumann boundary control problems. We carried out the examples on the domain $\Omega = [0,1]^3$. Whenever we show the degrees of freedom these are only the degrees of freedom for one grid point in time (i.e., for a single time-step). Implicitly, we are solving a linear system of dimension three times the number of time-steps (N_t) times the degrees of freedom of the spatial discretization (n). For example, a spatial discretization with 274,625 unknowns and 20 time-steps corresponds to an overall linear system of dimension 16,477,500. All results are performed on a Centos Linux machine with Intel Xeon CPU X5650 at 2.67 GHz CPUs and 48 GB of RAM.

4.1. Distributed control. We start by giving results for the distributed control examples presented earlier. For the distributed control problems we impose a zero Dirichlet condition. This results in the computed state not matching the desired state quite as well very close to the boundary. Another option would be to impose a Dirichlet condition where the state corresponds to the desired state on $\partial\Omega$.

4.1.1. The all-times case—whole domain. The example we consider for the distributed control problem is given by the all-times case, where the functional $J(y,u)$ contains observations for all time-steps. We have the choice of using the trapezoidal rule (which corresponds to the discretize-then-optimize formulation) or the rectangular rule (which corresponds to the optimize-then-discretize formulation) for the discretization of the state integral. We will show results for both cases that desired to drive the state close to the desired state given by

$$\bar{\mathbf{y}} = 64t\sin\left(2\pi\left((x_0-0.5)^2+(x_1-0.5)^2+(x_2-0.5)^2\right)\right)$$

with a zero initial value. An illustration of the desired state, the computed state, and the control is shown in Figure 4.1 for one particular point in time, i.e., one particular time-step. The results with the Schur complement approximation as presented in section 3.1.1 (trapezoidal rule) are shown in Table 4.1 and the results for the approach presented in section 3.1.2 (rectangular rule) are shown in Table 4.2. We can see that the number of iterations remains constant with varying mesh size and regularization parameter β.

4.1.2. The all-times case—subdomain problem. We now show results for the subdomain problem when the desired state is again defined by

$$\bar{\mathbf{y}} = 64t \sin\left(2\pi\left((x_0 - 0.5)^2 + (x_1 - 0.5)^2 + (x_2 - 0.5)^2\right)\right)$$

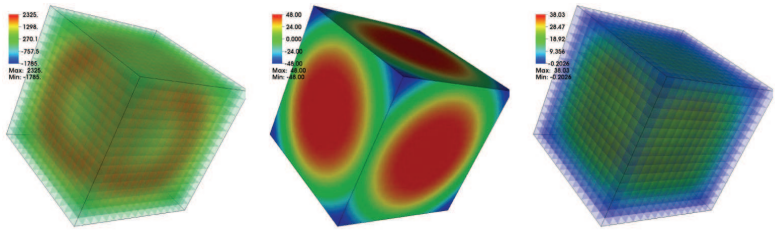
and the domain Ω_1 is defined by

$$\Omega_1 = \left\{x \in [0,1]^3 : 0.4 \leq x_1, x_2 \leq 0.7\right\}.$$

Results for this case are shown in Table 4.3, where we can again see that the iteration numbers are small and robust with respect to the mesh parameter and the regularization parameter. The timings are slightly higher than in the case for the whole domain. This is because in our experience the AMG approximation sometimes deteriorates for small parameters and as we now include γ in our approximation we decided to use four V-cycles instead of two.

4.2. Boundary control. We now present results for the time-independent and time-dependent Neumann boundary control problems discussed earlier.

4.2.1. Time-independent boundary control. The time-independent boundary control problem example that we present starts from initial value zero, matching



(a) Control (b) Desired state (c) Computed state

FIG. 4.1. *Control, desired state, and state for distributed control with* $\beta = 1e-4$ *at grid point* 15 *in time.*

TABLE 4.1
Results for discretize-then-optimize approach via trapezoidal rule.

DoF	MINRES(T)	MINRES(T)	MINRES(T)
	$\beta = 1e-2$	$\beta = 1e-4$	$\beta = 1e-6$
4913	10(2)	12(2)	12(2)
35937	10(14)	12(17)	12(18)
274625	10(148)	12(171)	12(170)

TABLE 4.2
Results for optimize-then-discretize approach via rectangular rule.

DoF	MINRES(T)	MINRES(T)	MINRES(T)
	$\beta = 1e-2$	$\beta = 1e-4$	$\beta = 1e-6$
4913	12(3)	10(2)	8(1)
35937	12(16)	10(14)	10(14)
274625	14(196)	10(152)	10(147)

TABLE 4.3
Results for discretize-then-optimize approach via trapezoidal rule for a subdomain problem.

DoF	MINRES(T)	MINRES(T)	MINRES(T)
	$\beta = 1e-2$	$\beta = 1e-4$	$\beta = 1e-6$
4913	12(5)	13(5)	15(5)
35937	12(28)	15(35)	17(38)
274625	12(332)	15(386)	19(495)

TABLE 4.4
Results obtained with Schur complement approximation $\widehat{S}_1$.

DoF	MINRES(T)	MINRES(T)	MINRES(T)
	$\beta = 1e-2$	$\beta = 1e-4$	$\beta = 1e-6$
4913	26(1)	28(1)	22(1)
35937	32(2)	38(2)	30(2)
274625	34(22)	48(31)	46(29)
2146689	38(211)	60(289)	64(314)

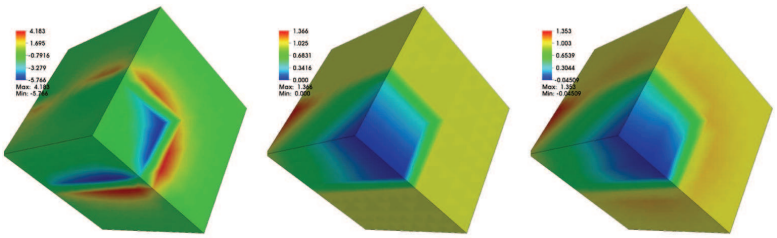
the desired state given by

$$\bar{\mathbf{y}} = \begin{cases} \sin(x_1) + x_2 x_0 & \text{if } x_0 > 0.5 \text{ and } x_1 < 0.5, \\ 1 & \text{otherwise.} \end{cases}$$

The desired state, computed state, and control are shown in Figure 4.2. The CPU times and iteration numbers for the MINRES algorithm with varying mesh size and regularization parameter are shown in Table 4.4 for the Schur complement approximation $\widehat{S}_1$ and in Table 4.5 for $\widehat{S}_2$. We see that $\widehat{S}_1$ performs better in all cases, although the results for $\widehat{S}_2$ are not dramatically different. We see for both approaches a slow growth in the iteration numbers, which is expected when dealing with a pure Neumann problem (see [5]). We observe some rather small growth with decreasing β, especially for small meshes, but with the iteration numbers still reasonably small. We also observe improved preformance when h^3 and β are further apart. The results we experience matched our expectations based on the theory detailed in section 3.2.

4.2.2. Time-dependent boundary control. The setup for the example time-dependent boundary control problem we present again starts with an initial value of zero and the following time-dependent desired state:

$$\bar{\mathbf{y}} = \begin{cases} \sin(t) + x_0 x_1 x_2 & \text{if } x_0 > 0.5 \text{ and } x_1 < 0.5, \\ 1 & \text{otherwise.} \end{cases}$$



(a) Control (b) Desired state (c) Computed state

FIG. 4.2. *Control, desired state, and state for boundary control with $\beta = 1e-4$.*

TABLE 4.5
Results obtained with Schur complement approximation $\widehat{S}_2$.

DoF	MINRES(T)	MINRES(T)	MINRES(T)
	$\beta = 1e-2$	$\beta = 1e-4$	$\beta = 1e-6$
4913	38(1)	38(1)	30(1)
35937	44(3)	54(3)	44(3)
274625	48(31)	74(48)	70(44)
2146689	54(263)	98(466)	108(513)

TABLE 4.6
Results obtained with Schur complement approximation $\widehat{S}_3$.

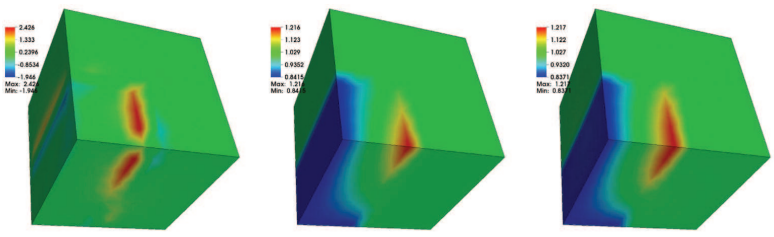
DoF	MINRES(T)	MINRES(T)	MINRES(T)
	$\beta = 1e-2$	$\beta = 1e-4$	$\beta = 1e-6$
4913	34(7)	38(7)	28(6)
35937	38(49)	48(62)	38(48)
274625	48(620)	62(800)	58(725)

TABLE 4.7
Results obtained with Schur complement approximation $\widehat{S}_4$.

DoF	MINRES(T)	MINRES(T)	MINRES(T)
	$\beta = 1e-2$	$\beta = 1e-4$	$\beta = 1e-6$
4913	40(8)	42(8)	36(7)
35937	50(65)	59(73)	42(54)
274625	62(808)	80(1002)	68(855)

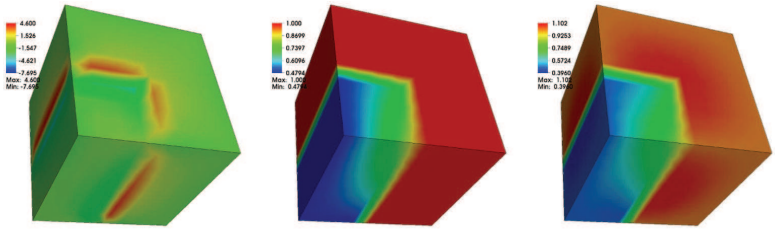
The desired state as well as the computed state and control are depicted for grid point 20 in time (i.e., the 20th time step) in Figure 4.3 and for grid point 10 (the 10th time step) in Figure 4.4. We again computed results for both Schur complement approximations presented earlier; the results are given in Table 4.6 for $\widehat{S}_3$ and in Table 4.7 for $\widehat{S}_4$. We again see higher iteration numbers for the second approximation $\widehat{S}_4$ and benign growth with respect to the mesh size, but again with improved results if h^3 and β are far apart. The results here reflect the results for the time-independent case, which we expect due to our theoretical study presented in section 3.3.

5. Concluding remarks and outlook. We have presented various setups for the optimal control of the heat equation. We derived the discretized first order conditions for the distributed and boundary control cases and showed that both problems



(a) Control (b) Desired state (c) Computed state

FIG. 4.3. *Control, desired state, and state for boundary control with* $\beta = 1e-6$ *at grid point* 20 *in time.*



(a) Control (b) Desired state (c) Computed state

FIG. 4.4. *Control, desired state, and state for boundary control with* $\beta = 1e-6$ *at grid point* 10 *in time.*

lead to a linear system with saddle point structure. We then extended the analysis for a regularization-robust preconditioner from the time-independent distributed control case to the time-dependent distributed control case. We also provided some bounds for the case of Neumann boundary control for the time-dependent and time-independent setups. We then gave an extensive numerical study of the preconditioners derived earlier and showed that the dependence with respect to the mesh size, regularization parameter, and time-step could be removed for the distributed control case. The numerical results for the pure Neumann control problem illustrated a benign dependence on the mesh size (similar to the forward problem) and very little dependence with respect to the regularization parameter β. These results have already been used in a different work on time-periodic parabolic problems with control constraints (see [51]), where good numerical results were obtained. The work presented in this paper also serves as a framework for the consideration of other time-dependent optimal control problems. The techniques presented could be adapted for the case where the control is only applied in a subdomain, or examples with additional constraints such as box constraints being imposed on the state or control.

A possible future extension of this work would be to develop robust preconditioners for more complicated PDEs with respect to all parameters involved. As well, one could generate solvers for the subdomain case, as discussed in this manuscript, or the boundary control setting. Furthermore, one drawback of the procedure described in this paper, which could be tackled in future work, is the necessary storage requirement for the vectors corresponding to the control, state, and adjoint. Although it is possible to condense the system by, for example, eliminating the control, more research would be required here. Various approaches have been applied to time-dependent PDE-constrained optimization in the past. For instance, checkpointing [17], a method which involves storing only certain checkpoints of the state and computing the adjoint state based on these, has been investigated. We note that our one-shot approach is not ideally suited for this method but rather could be treated using ideas based on instantaneous control [10, 24, 21], multiple shooting [21], and parareal schemes [31, 32, 33]. Possibly the simplest idea of all is to split up the interval into subintervals and use our approach to solve the relevant subproblems, for which all the analysis presented here carries over. However, we note that the solution obtained using this approach is suboptimal [21]. Parareal and shooting methods maintain the splitting into subintervals but ensure agreement of the control and state where the time-slices meet each other. We believe that the techniques presented here can be

used within multiple shooting methods such as that presented in [21]—this is another area of further work which will be investigated.

Acknowledgement. The authors would like to thank an anonymous referee for a careful reading of the manuscript and helpful comments.

REFERENCES

[1] U. Ascher and E. Haber, *A multigrid method for distributed parameter estimation problems.*, Electron. Trans. Numer. Anal., 15 (2003), pp. 1–17.
[2] W. Bangerth, R. Hartmann, and G. Kanschat, *deal.* II—*a general-purpose object-oriented finite element library*, ACM Trans. Math. Software, 33 (2007), pp. Art. 24, 27.
[3] M. Benzi, G. H. Golub, and J. Liesen, *Numerical solution of saddle point problems*, Acta Numer, 14 (2005), pp. 1–137.
[4] M. Benzi, E. Haber, and L. Taralli, *A preconditioning technique for a class of PDE-constrained optimization problems*, Adv. Comput. Math., 35 (2011), pp. 149–173.
[5] P. Bochev and R. Lehoucq, *On the finite element solution of the pure Neumann problem*, SIAM Rev., 47 (2005), pp. 50–66.
[6] A. Borzì, *Multigrid methods for parabolic distributed optimal control problems.*, J. Comput. Appl. Math., 157 (2003), pp. 365–382.
[7] A. Borzì and V. Schulz, *Multigrid methods for PDE optimization.*, SIAM Rev., 51 (2009), pp. 361–395.
[8] J. H. Bramble and J. E. Pasciak, *A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems*, Math. Comp., 50 (1988), pp. 1–17.
[9] E. Casas, *Control of an elliptic problem with pointwise state constraints*, SIAM J. Control Optim., 24 (1986), pp. 1309–1318.
[10] H. Choi, M. Hinze, and K. Kunisch, *Instantaneous control of backward-facing step flows*, Appl. Numer. Math., 31 (1999), pp. 133–158.
[11] S. Dollar, *Iterative Linear Algebra for Constrained Optimization*, Ph.D. thesis, University of Oxford, 2005; also available online from http://web.comlab.ox.ac.uk/oucl/research/na/thesis/thesisdollar.pdf.
[12] H. C. Elman, D. J. Silvester, and A. J. Wathen, *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*, Numer. Math. Sci. Comput., Oxford University Press, New York, 2005.
[13] R. Falgout, *An Introduction to Algebraic Multigrid*, Comput. Sci. Engrg., 8 (2006), pp. 24–33.
[14] B. Fischer, *Polynomial Based Iteration Methods for Symmetric Linear Systems*, Ser. Adv. Numer. Math., John Wiley & Sons, Chichester, UK, 1996.
[15] M. Gee, C. Siefert, J. Hu, R. Tuminaro, and M. Sala, *ML* 5.0 *Smoothed Aggregation User's Guide*, Tech. rep. SAND2006-2649, Sandia National Laboratories, 2006.
[16] 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.
[17] A. Griewank and A. Walther, *Algorithm* 799*: revolve: An implementation of checkpointing for the reverse or adjoint mode of computational differentiation*, ACM Trans. Math. Software, 26 (2000), pp. 19–45.
[18] E. Haber, *A parallel method for large scale time domain electromagnetic inverse problems.*, Appl. Numer. Math., 58 (2008), pp. 422–434.
[19] E. Haber, U. M. Ascher, and D. Oldenburg, *On optimization techniques for solving nonlinear inverse problems.*, Inverse Problems, 16 (2000), pp. 1263–1280.
[20] W. Hackbusch, *Multigrid methods and applications*, Springer Ser. Comput. Math. 4, Springer-Verlag, Berlin, 1985.
[21] M. Heinkenschloss, *A time-domain decomposition iterative method for the solution of distributed linear quadratic optimal control problems.*, J. Comput. Appl. Math., 173 (2005), pp. 169–198.
[22] R. Herzog and E. W. Sachs, *Preconditioned conjugate gradient method for optimal control problems with control and state constraints*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 2291–2317.
[23] M. Hintermüller, K. Ito, and K. Kunisch, *The primal-dual active set strategy as a semismooth Newton method*, SIAM J. Optim., 13 (2002), pp. 865–888.
[24] M. Hinze, *Optimal and Instantaneous Control of the Instationary Navier-Stokes Equations*, Habilitation, TU Berlin, 2000.

[25] M. Hinze, M. Köster, and S. Turek, *A Hierarchical Space-Time Solver for Distributed Control of the Stokes Equation*, Priority Programme 1253, Tech. rep. SPP1253-16-01, 2008.
[26] M. Hinze, M. Köster, and S. Turek, *A Space-Time Multigrid Solver for Distributed Control of the Time-Dependent Navier-Stokes System*, Priority Programme 1253, Tech. rep. SPP1253-16-02, 2008.
[27] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich, *Optimization with PDE Constraints*, in Mathematical Modelling: Theory and Applications, Springer-Verlag, New York, 2009.
[28] K. Ito and K. Kunisch, *Semi-smooth Newton methods for state-constrained optimal control problems*, Systems Control Lett., 50 (2003), pp. 221–228.
[29] K. Ito and K. Kunisch, *Lagrange multiplier approach to variational problems and applications*, Adv. Des. Control 15, SIAM, Philadelphia, 2008.
[30] C. Keller, N. Gould, and A. Wathen, *Constraint Preconditioning for Indefinite Linear Systems*, SIAM J. Matrix Anal. Appl, 21 (2000), pp. 1300–1317.
[31] J. Lions, Y. Maday, and G. Turinici, *A "parareal" in time discretization of PDE's*, C. R. Math. Acad. Sci. Ser. Paris, 332 (2001), pp. 661–668.
[32] Y. Maday and G. Turinici, *A parareal in time procedure for the control of partial differential equations*, C. R. Math., 335 (2002), pp. 387–392.
[33] T. P. Mathew, M. Sarkis, and C. E. Schaerer, *Analysis of block parareal preconditioners for parabolic optimal control problems.*, SIAM J. Sci. Comput., 32 (2010), pp. 1180–1200.
[34] M. F. Murphy, G. H. Golub, and A. J. Wathen, *A note on preconditioning for indefinite linear systems*, SIAM J. Sci. Comput, 21 (2000), pp. 1969–1972.
[35] C. C. Paige and M. A. Saunders, *Solutions of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal, 12 (1975), pp. 617–629.
[36] J. W. Pearson, M. Stoll, and A. J. Wathen, *Robust Iterative Solution of a Class of Time-Dependent Optimal Control Problems*, submitted.
[37] J. W. Pearson and A. J. Wathen, *A New Approximation of the Schur Complement in Preconditioners for PDE Constrained Optimization*, Numer. Linear Algebra Appl., (2011), DOI 10.1002/nla.814.
[38] J. W. Pearson and A. J. Wathen, *Fast Iterative Solvers for Convection-Diffusion Control Problems*, submitted.
[39] A. Potschka, M. Mommer, J. Schlöder, and H. Bock, *A Newton-Picard Approach for Efficient Numerical Solution of Time-Periodic Parabolic PDE Constrained Optimization Problems*, Interdisciplinary Center for Scientific Computing, Heidelberg University, 2010.
[40] T. Rees, H. S. Dollar, and A. J. Wathen, *Optimal solvers for PDE-constrained optimization*, SIAM J. Sci. Comput., 32 (2010), pp. 271–298.
[41] T. Rees and C. Greif, *A preconditioner for linear systems arising from interior point optimization methods.*, SIAM J. Sci. Comput., 29 (2007), pp. 1992–2007.
[42] T. Rees and M. Stoll, *Block-triangular preconditioners for PDE-constrained optimization*, Numeri. Linear Algebra Appl., 17 (2010), pp. 977–996.
[43] T. Rees and M. Stoll, *A fast solver for an H_1 regularized optimal control problem*, submitted.
[44] T. Rees, M. Stoll, and A. Wathen, *All-at-once preconditioners for PDE-constrained optimization*, Kybernetika, 46 (2010), pp. 341–360.
[45] J. W. Ruge and K. Stüben, *Algebraic multigrid*, in Multigrid Methods, Frontiers Appl. Math. 3, SIAM, Philadelphia, 1987, pp. 73–130.
[46] Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, Philadelphia, 2003.
[47] Y. Saad and M. H. Schultz, *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput, 7 (1986), pp. 856–869.
[48] J. Schöberl and W. Zulehner, *Symmetric indefinite preconditioners for saddle point problems with applications to PDE-constrained optimization problems*, SIAM J. Matrix Anal. Appl, 29 (2007), pp. 752–773.
[49] V. Simoncini, *Block triangular preconditioners for symmetric saddle-point problems*, Appl. Numer. Math, 49 (2004), pp. 63–80.
[50] V. Simoncini, *Reduced order solution of structured linear systems arising in certain PDE-constrained optimization problems*, Comput. Optim. Appl., to appear.
[51] M. Stoll, *All-at-once solution of a time-dependent time-periodic PDE-constrained optimization problems*, submitted.
[52] M. Stoll and A. Wathen, *All-at-once solution of time-dependent PDE-constrained optimization problems*, Technical Report 1017, Mathematical Institute, University of Oxford, 2010.
[53] M. Stoll and A. Wathen, *All-at-once solution of time-dependent Stokes control*, J. Comput. Phys., to appear.
[54] S. Takacs and W. Zulehner, *Convergence analysis of multigrid methods with collective point smoothers for optimal control problems*, Comput. Vis. Sci., 14 (2011), pp.131–141.

[55] F. Tröltzsch, *Optimal Control of Partial Differential Equations: Theory, Methods and Applications*, AMS, Providence, RI, 2010.
[56] M. Ulbrich, *Semismooth Newton Methods for Variational Inequalities and Constrained Optimization Problems*, SIAM Philadelphia, 2011.
[57] H. A. van der Vorst, *Iterative Krylov methods for large linear systems*, Cambridge Monogr. Appl. Comput. Math. 13, Cambridge University Press, Cambridge, UK, 2003.
[58] A. J. Wathen and T. Rees, *Chebyshev semi-iteration in preconditioning for problems including the mass matrix*, Electron. Trans. Numer. Anal., 34 (2008), pp. 125–135.
[59] W. Zulehner, *Analysis of iterative methods for saddle point problems: a unified approach*, Math. Comp., 71 (2002), pp. 479–505.

A.4 Time-periodic heat equation and preconditioning

This paper is published as

M. Stoll, ***All-at-once solution of a time-dependent time-periodic PDE-constrained optimization problems*, IMA J. Numer Anal,, 34 (2014), pp. 1554–1577.**

Result from the paper

In this paper we develop robust preconditioners for time-periodic problems. In some instances we make use of the circulant structure within the discretization. For such a circulant preconditioner Table A.4 shows MINRES iterations and timings for various meshes and values of the regularization parameter β applied to a boundary control problem.

DoF	MINRES(t) $\beta = 1e-2$	MINRES(t) $\beta = 1e-4$	MINRES(t) $\beta = 1e-6$
36880	30(14)	26(12)	20(9)
227280	32(83)	32(82)	26(67)
1560400	34(384)	40(450)	36(403)
11476560	36(2671)	*	*

Table A.4: Circulant preconditioner: MINRES iterations and timings for various meshes and values of the regularization parameter β applied to a boundary control problem with only one step for the Uzawa iteration.

IMA Journal of Numerical Analysis (2014) **34**, 1554–1577
doi:10.1093/imanum/drt019
Advance Access publication on October 11, 2013

One-shot solution of a time-dependent time-periodic PDE-constrained optimization problem

Martin Stoll
Computational Methods in Systems and Control Theory, Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, 39106 Magdeburg, Germany
stollm@mpi-magdeburg.mpg.de

[Received on 19 July 2011; revised on 14 March 2013]

In this paper we describe the efficient solution of a partial differential equation (PDE)-constrained optimization problem subject to the time-periodic heat equation. We propose a space-time formulation for which we develop a monolithic solver. We present preconditioners well suited to approximating the Schur complement of the saddle point system associated with the first-order conditions. This means that, in addition to a Richardson iteration-based preconditioner, we also introduce a preconditioner based on the tensor product structure of the PDE discretization, which allows the use of an FFT-based preconditioner. We also consider additional bound constraints that can be treated using a semismooth Newton method. Moreover, we introduce robust preconditioners with respect to the regularization parameter. Numerical results will illustrate the competitiveness and flexibility of our approach.

Keywords: PDE-constrained optimization; saddle point systems; time-dependent PDE-constrained optimization; preconditioning; Krylov subspace solver.

1. Introduction

For many years the solution of partial differential equation (PDE) problems has been the focus of the numerical analysis and scientific computing community. The progress made over recent decades has enabled the search for, in some sense, optimal solutions of PDEs. This task in the field often labelled PDE-constrained optimization is to minimize an objective function subject to constraints given by PDEs. Introductions to the field can be found in Ito & Kunisch (2008), Hinze *et al.* (2009) and Tröltzsch (2010).

A typical example will look like the following:

$$\min \quad J(y,u) \tag{1.1}$$

$$\text{s.t} \quad \mathcal{L}(y,u) = 0, \tag{1.2}$$

where $J(y,u)$ is the function we want to minimize and $\mathcal{L}(y,u) = 0$ represents an equation constraint, typically a PDE, that links the state y and the control u. We assume that suitable boundary conditions are given and in the case of time-dependent problems initial conditions are specified. Often the introduction of additional constraints, such as bound constraints on the control and/or the state poses additional challenges (see Ito & Kunisch, 2008; Hinze *et al.*, 2009; Tröltzsch, 2010 and the references mentioned therein for suitable methods to deal with this).

Our focus in this paper is to solve a problem of the above type where the PDE constraint is equipped with appropriate boundary conditions and the state y exhibits time periodicity, i.e., $y(0,\cdot) = y(T,\cdot)$,

where T is the final time we are interested in. Problems of this type have recently been analysed, for example, in Abbeloos *et al.* (2011) and Potschka *et al.* (2012). The motivation for optimal control problems of this type is that many applications incorporate time periodicity; see, e.g., Kawajiri & Biegler (2006) for so-called simulated moving bed reactors.

The paper is organized as follows. In the next section, we introduce the problem formulation and the PDE constraint with time periodicity. We will discuss the discretization and then give the first order optimality system, which shows additional structure due to the time periodicity. In Section 3 we briefly introduce a technique based on the semismooth Newton method (cf. Kummer, 1988; Pang, 1990; Qi & Sun, 1993; Bergounioux *et al.*, 1999; Hintermüller *et al.*, 2002) that will allow us to handle box constraints on the control. We will then briefly motivate our choice of the Krylov subspace solver. In Section 5, we discuss the preconditioners that are suitable for our approach as the matrix structure is different from the case when nonperiodic boundary conditions are given. Note that parts of our approach are similar to previous work by the author (see Pearson *et al.*, 2012b) but the special structure in the time-periodic case introduces additional structure that can be exploited within a numerical treatment. Namely, we discuss a stationary iteration-based preconditioner and also a preconditioner using the circulant structure of the discretized PDE. We discuss fast solvers for the case of a boundary control problem focusing on the numerical issues that arise for a varying regularization parameter. Numerical results in Section 6 illustrate the efficiency of our approach.

2. Problem and discretization

In this paper, we analyse tracking-type functionals subject to a time-periodic PDE. The functional that we want to minimize is given by

$$J_1(y,u) := \frac{1}{2}\int_0^T \int_{\Omega_1} (y(x,t) - \bar{y}(x,t))^2 \,\mathrm{d}x\,\mathrm{d}t + \frac{\beta}{2}\int_0^T \int_{\Omega_2} (u(x,t))^2 \,\mathrm{d}x\,\mathrm{d}t, \tag{2.1}$$

where $\Omega_{1,2} \subseteq \Omega$ are domains in $\mathbb{R}^d$ with $d = \{2,3\}$, y is the state, $\bar{y}$ is the desired state and u is the control. We want to minimize this functional subject to the time-periodic heat equation that links the state and the control and is hence called the state equation. In more detail, the equation now reads as

$$y_t - \triangle y = \chi_{\Omega_2} u \tag{2.2}$$

defined over $\Omega \times [0,T]$ with χ_{Ω_2} the characteristic function of the domain Ω_2. Also we impose the Dirichlet boundary condition $y = 0$ on the spatial boundary $\partial\Omega$ and the time-periodic condition $y(x,0) = y(x,T)$. In addition, we allow for variations of this problem. The first is the so-called boundary control problem given by

$$J_{\mathrm{bnd}}(y,u) := \frac{1}{2}\int_0^T \int_{\Omega_1} (y(x,t) - \bar{y}(x,t))^2 \,\mathrm{d}x\,\mathrm{d}t + \frac{\beta}{2}\int_0^T \int_{\partial\Omega} (u(x,t))^2 \,\mathrm{d}x\,\mathrm{d}t, \tag{2.3}$$

subject to

$$y_t - \triangle y = f, \tag{2.4}$$

with Neumann boundary condition

$$\frac{\partial y}{\partial \mathrm{n}} = u,$$

and some fixed forcing term f that we assume to be zero. Additionally, the introduction of bound constraints on the control and/or the state poses additional challenges to numerical algorithms. Bounds such as

$$u_a \leqslant u \leqslant u_b \quad \text{and} \quad y_a \leqslant y \leqslant y_b$$

have to be accounted for by more sophisticated algorithms (Ito & Kunisch, 2008; Hinze *et al.*, 2009). We discuss the necessary approaches for control constraints in later parts of this paper (see Section 3). There are two ways to proceed from the above problems. First, one can write down the infinite-dimensional first-order conditions and then discretize them; this is the so-called *optimize-then-discretize* approach. The other approach is to discretize first and then write down the first order or Karush-Kuhn-Tucker (KKT) conditions; this is the so-called *discretize-then-optimize* approach. For many problems it is desirable that these two approaches coincide, which is taken into account when devising discretization schemes such as the ones derived in Hinze *et al.* (2008). We here follow the discretize-then-optimize approach. Hence, we discretize both the functional and the PDE using standard Galerkin **Q1** finite elements, rectangular in our case because of the underlying use of deal.II (see Bangerth *et al.*, 2007), which does not use triangular elements.

For the time discretization of the PDE we use a backward Euler scheme that leads to the semidiscretized form of (2.2),

$$\frac{y^k - y^{k-1}}{\tau} - \triangle y^k = u^k, \tag{2.5}$$

with τ being the time step, and the number of grid points in time is denoted by n_t. The second PDE (2.4) is treated similarly. The finite element discretization in space is straightforward and putting all time steps into one system, a so-called one-shot approach, leads to

$$\underbrace{\begin{bmatrix} M+\tau K & & & & -M \\ -M & M+\tau K & & & \\ & -M & M+\tau K & & \\ & & \ddots & \ddots & \\ & & & -M & M+\tau K \end{bmatrix}}_{\mathcal{K}} \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \mathbf{y}_3 \\ \vdots \\ \mathbf{y}_N \end{bmatrix} - \tau \mathcal{N} \mathbf{u} = d, \tag{2.6}$$

where M and K are the finite element (lumped) mass and stiffness matrix, respectively, and d the right-hand side representing the boundary conditions and forcing terms. The finite element stiffness and mass matrices are constructed from

$$K_{ij} = \int_\Omega \nabla \phi_i \cdot \nabla \phi_j \quad \text{and} \quad M_{ij} = \int_\Omega \phi_i \phi_j,$$

respectively, where the ϕ functions are the basis functions that are used both for the trial and test space (see Elman *et al.*, 2005). The mass lumping is obtained by a particular choice of numerical quadrature, i.e., the trapezoidal rule for our **Q1** elements. Note that $\mathcal{K}$ exhibits a circulant structure, which we will discuss in more detail later. Further, we have the matrix $\mathcal{N} = \text{blkdiag}(N, N, \ldots, N)$,[1] where N can be a rectangular matrix, depending on the nature of the optimal control problem and its discretization. For the distributed control problem, as discretized here N is a square mass matrix because u is discretized with the same trial functions as y and these are also the basis for the test functions. In the case of u being

[1] Here we use the MATLAB notation blkdiag to describe a matrix in block-diagonal form.

discretized using different finite element trial functions from the ones used for the state y, e.g., piecewise constant elements for the control and linear finite elements for the state, or if u is a boundary control, then N is a rectangular matrix. In the case of a boundary control N, will consist of entries coming from the integral $\int_{\partial\Omega} u\,\mathrm{tr}(v)$, where u is the boundary control and tr is the trace operator acting on the test function v from the test space used for the discretization of the state y.

We now need to discretize the objective function $J(y,u)$ and for this we use the trapezoidal rule to get the discretized objective function as

$$J(\mathbf{y},\mathbf{u}) = \frac{\tau}{2}(\mathbf{y}-\bar{\mathbf{y}})^{\mathrm{T}}\mathcal{M}_y(\mathbf{y}-\bar{\mathbf{y}}) + \frac{\tau\beta}{2}\mathbf{u}^{\mathrm{T}}\mathcal{M}_u\mathbf{u}, \tag{2.7}$$

where $\mathcal{M}_y = \mathrm{blkdiag}(1/2M_y, M_y, \ldots, 1/2M_y)$, where M_y is the mass matrix over the domain Ω_1, and $\mathcal{M}_u = \mathrm{blkdiag}(1/2M_u, M_u, \ldots, 1/2M_u)$, where M_u represents the mass matrix for the domain Ω_2.

Once all these ingredients are available, we can combine them into a Lagrangian and write down the first-order conditions, which can be written as the following KKT system:

$$\begin{bmatrix} \tau\mathcal{M}_y & 0 & -\mathcal{K}^{\mathrm{T}} \\ 0 & \beta\tau\mathcal{M}_u & \tau\mathcal{N}^{\mathrm{T}} \\ -\mathcal{K} & \tau\mathcal{N} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \boldsymbol{p} \end{bmatrix} = \begin{bmatrix} \mathcal{M}_y\bar{\mathbf{y}} \\ 0 \\ d \end{bmatrix}. \tag{2.8}$$

Note that (2.8) represents a saddle point system with a symmetric and positive semidefinite $(1,1)$ block given by $\mathrm{blkdiag}(\tau\mathcal{M}_y, \beta\tau\mathcal{M}_u)$ and a full rank block $[-\mathcal{K} \quad \tau\mathcal{N}]$. For the case of a singular $\mathcal{K}$ we refer the reader to Section 5.5. Here, $\boldsymbol{p}$ represents the discrete Lagrange multiplier or equivalently the solution to the adjoint PDE. These conditions are sufficient for the invertibility of the saddle point system. In a optimize-then-discretize approach one would obtain the adjoint PDE from the Lagrangian formulation. Its existence and uniqueness is discussed in Tröltzsch (2010, Lemma 3.17). Here, the first row in (2.8) represents a discretization of the adjoint PDE obtained from a discretize-then-optimize approach and the existence of its solution follows from the existence of the solution to the state equation.

Note that in the case of a discretization of the objective function via the rectangular rule the block $\mathrm{blkdiag}(\tau\mathcal{M}_y, \beta\tau\mathcal{M}_u)$ would contain zero blocks. In the case of only a final time observation where the first term in $J(y,u)$ is given by

$$\frac{1}{2}\int_{\Omega_1} (y(x,T) - \bar{y}(x,T))^2\,\mathrm{d}x,$$

the $(1,1)$ block of the saddle point system, will be highly singular (see Benzi *et al.*, 2011; Simoncini, 2012; Stoll & Wathen, 2010, 2013). Note that the above linear system will typically be of very large dimension, i.e., the dimension is given by $3nn_t$, where n is the number of the degrees of freedom of the PDE discretization.

Note that a reduction in the dimensionality of the above system is possible by eliminating the control $\mathbf{u}$, a technique that is also discussed in Hinze (2005) and Simoncini (2012). In the case where the $(1,1)$ block is positive definite, we can also only work with the Schur-complement reduction as the mass matrices are lumped and hence the evaluation of $\mathcal{M}_y^{-1}$ and $\mathcal{M}_u^{-1}$ is trivial. This case would enable the use of the classical conjugate gradient (CG) method (see Hestenes & Stiefel, 1952) but the challenge in developing good preconditioners for the Schur complement stays intact. Note this does not apply if $\mathcal{M}_y$ and $\mathcal{M}_u$ are not invertible, e.g., $\Omega_1 \subsetneq \Omega$.

3. Bound constraints

Bound constraints on the control and the state represent additional challenges with regard to the design of numerical methods. For this purpose the development of semismooth Newton methods has received much attention over the last decade. In the context of PDE-constrained optimization this method was first introduced as a primal–dual active set method (see Bergounioux *et al.*, 1999) for problems with control constraints. It was later shown in Hintermüller *et al.* (2002) that this method is equivalent to a semismooth Newton method. Since then the semismooth Newton method has been employed for many problems in PDE-constrained optimization. The field of nonsmooth Newton methods is vast and we refer the reader to Kummer (1988), Pang (1990), Qi & Sun (1993), Hintermüller *et al.* (2002) and Ulbrich (2011) and the references mentioned therein.

In Stoll & Wathen (2012) the efficient solution of the linear systems arising within the semismooth Newton method for steady control-constrained problems was discussed. Here, we want to use a technique that again employs the semismooth Newton method but applied to a slightly different set-up. In more detail, we replace the objective function (2.1) and the control constraints by a penalized objective function, which for PDE-constrained optimization problems is often referred to as the Moreau–Yosida regularization (see Bergounioux & Kunisch, 2002). This function looks as follows:

$$\tilde{J}(y,u) := J(y,u) + \frac{1}{2\varepsilon}\int_0^T \int_{\Omega_2} (\max\{0, u(x,t) - \bar{u}(x,t)\})^2 + (\min\{0, u(x,t) - \underline{u}(x,t)\})^2 \,\mathrm{d}x\,\mathrm{d}t, \quad (3.1)$$

again subject to the time-periodic heat equation that links the state and the control. The parameter ε is assumed to be small and can be interpreted as a penalization parameter associated with the box constraints on the control. We again proceed with a discretization utilizing finite elements and the implicit Euler scheme to obtain a discretized version of the Moreau–Yosida regularized problem. It was shown previously (see Herzog & Sachs, 2010) that the semismooth Newton method can be used to solve the Moreau–Yosida regularized problem. At the heart of this method again lies the solution of a linear system in saddle point form, i.e.,

$$\begin{bmatrix} \mathcal{M}_y & 0 & -\mathcal{K}^{\mathrm{T}} \\ 0 & \beta\mathcal{M}_u + \varepsilon^{-1}\mathcal{G}_{\mathcal{A}^{(k)}}\mathcal{M}_u\mathcal{G}_{\mathcal{A}^{(k)}} & \mathcal{N}^{\mathrm{T}} \\ -\mathcal{K} & \mathcal{N} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y}^{(k)} \\ \mathbf{u}^{(k)} \\ \boldsymbol{p}^{(k)} \end{bmatrix} = \begin{bmatrix} \mathcal{M}_y\bar{\mathbf{y}} \\ \varepsilon^{-1}(\mathcal{G}_{\mathcal{A}_+^{(k)}}\mathcal{M}_u\mathcal{G}_{\mathcal{A}_+^{(k)}}\underline{u} + \mathcal{G}_{\mathcal{A}_-^{(k)}}\mathcal{M}_u\mathcal{G}_{\mathcal{A}_-^{(k)}}\bar{u}) \\ d \end{bmatrix}. \quad (3.2)$$

The matrices $\mathcal{G}$ are block-diagonal matrices with blocks G that are again diagonal matrices. These matrices represent the generalized derivatives with respect to the nonsmooth parts of the objective function associated with the control constraints at each grid point in time where we define the active sets as

$$\mathcal{A}_+^{(k)} := \{i : \mathbf{u}_i \geqslant \bar{u}_i\} \quad \text{and} \quad \mathcal{A}_-^{(k)} := \{i : \mathbf{u}_i \leqslant \underline{u}_i\}.$$

We note that these sets consist of n_t different active sets for each grid point in time. Additionally, we have that

$$\mathcal{A}^{(k)} = \mathcal{A}_+^{(k)} \cup \mathcal{A}_-^{(k)}.$$

To briefly summarize, the active sets defined above simply store the indices where the control attains the upper or lower bound. Note that the matrix $\beta\mathcal{M}_u + \varepsilon^{-1}\mathcal{G}_{\mathcal{A}^{(k)}}\mathcal{M}_u\mathcal{G}_{\mathcal{A}^{(k)}}$ is a block-diagonal matrix where each block is the sum of a scaled (by β) mass matrix and a scaled (by ε^{-1}) submatrix of a mass matrix defined on the domain Ω_2 (the control domain). The goal now is to solve the system (3.2) efficiently and we come back to this in Section 5.

4. Choice of Krylov solver

As the dimensionality of the linear system is very large and the applications are likely to be three-dimensional, direct methods based on a factorization of the saddle point system (cf. Duff, 1996; Davis, 2005) will not be applicable for realistic scenarios of the above-described problem. Therefore, we apply iterative Krylov solvers. These methods build up a so-called Krylov subspace,

$$\mathcal{K}_k(\mathcal{A}, r_0) = \text{span}\{r_0, \mathcal{A}r_0, \ldots, \mathcal{A}^{k-1}r_0\},$$

and then construct an approximation to the solution of the linear system based upon some optimality criteria for the current iteration. In the case of a symmetric and positive-definite upper left block, CG methods (see Hestenes & Stiefel, 1952) can be applied, typically with a nonstandard inner product. There are a number of candidates that are based upon a nonstandard inner product, usually employing different preconditioners and hence different inner products. The Bramble–Pasciak CG method introduced in Bramble & Pasciak (1988) is a very successful method coming from finite element solutions of the Stokes problem and has recently been used for optimal control problems (see Rees & Stoll, 2010). Schöberl & Zulehner (2007) proposed another method that also has been used successfully for optimal control problems by Herzog & Sachs (2010). Our method of choice here will be the minimal residual method (MINRES) of Paige & Saunders (1975), which minimizes the residual $r_k = b - \mathcal{K}x_k$ over the current Krylov space. This method needs a symmetric and positive-definite preconditioner, which typically would look like

$$\mathcal{P} = \begin{bmatrix} A_0 & 0 \\ 0 & S_0 \end{bmatrix}, \tag{4.1}$$

where A_0 approximates the upper left block and S_0 approximates the Schur complement of the saddle point system. These choices within $\mathcal{P}$ are motivated by a result given in Murphy *et al.* (2000), where it is shown that the choices of A_0 as the unchanged upper left block and S_0 as the negative Schur complement lead to three distinct eigenvalues in the preconditioned system. Our goal is hence to find good approximations to both the Schur complement and the upper left block.

The problem of solving time-periodic PDE problems is not a new one and a variety of methods have been proposed to solve the forward problem; see Vandewalle & Piessens (1992) for a multigrid approach or Bomhof & van der Vorst (2001) and Bomhof (2001) for a GMRES technique applicable to cyclic systems. The method given in Vandewalle & Piessens (1992) has been used for the optimal control problem studied in Abbeloos *et al.* (2011). In Ernst (2000) an overview of iterative methods that apply to p-cyclic matrices is presented.

5. Preconditioners

As we have seen in the previous section the choice of approximations for the $(1, 1)$ block of our system and the Schur complement

$$\tau^{-1}\mathcal{K}\mathcal{M}_y^{-1}\mathcal{K}^{\mathrm{T}} + \tau\beta^{-1}\mathcal{N}\mathcal{M}_u^{-1}\mathcal{N}^{\mathrm{T}}, \tag{5.1}$$

Set $D = diag(M)$
Set relaxation parameter ω
Compute $g = \omega D^{-1}\hat{b}$
Set $S = (I - \omega D^{-1}M)$ (this can be used implicitly)
Set $z_{k-1} = 0$ and $z_k = Sz_{k-1} + g$
$c_{k-1} = 2$ and $c_k = \omega$
for $k = 2, \ldots, l$ **do**
  $c_{k+1} = \omega c_k - \frac{1}{4}c_{k-1}$
  $\vartheta_{k+1} = \omega(c_k/c_{k+1})$
  $z_{k+1} = \vartheta_{k+1}(Sz_k + g - z_{k-1}) + z_{k-1}$
end for

Algorithm 1: Chebyshev semiiterative method for a number of l steps

where we assume that $\mathcal{M}_y$ and $\mathcal{M}_u$ are both invertible, is crucial. Note that this is not the case if $\Omega_1 \subsetneq \Omega$ or a rectangular rule is used for the approximation of the time integral, but even in that case, we can get good preconditioners that approximate an equation that somewhat resembles (5.1). Note that we initially follow a strategy used in Rees *et al.* (2010b) to drop the second term $\tau\beta^{-1}\mathcal{N}\mathcal{M}_u^{-1}\mathcal{N}^{\mathrm{T}}$, but we later comment on and introduce alternatives. For expository purposes we will introduce all approximations for the first approach and later introduce a slight change in these approximations that allows for a more robust approach but can in a straightforward manner be used with the previously introduced techniques.

5.1 $(1,1)$ *Block*

Our goal in this section is to derive effective approximations to the upper left block and the Schur complement. The upper left block is given by $\mathrm{blkdiag}(\tau\mathcal{M}_y, \beta\tau\mathcal{M}_u)$. This leaves us with the problem of efficiently approximating mass matrices. This is a trivial task once the mass matrices are lumped and hence diagonal. If we did not employ the trapezoidal rule, we would obtain a nondiagonal mass matrix, the so-called consistent mass matrix, which can be handled via the Chebyshev semiiteration (see Algorithm 1). This method is a viable tool for preconditioning and has been used successfully for optimal control applications (see Rees *et al.*, 2010a,b). Going to the case of control constraints we note again that the $(2,2)$ block consists of blocks of the form

$$\beta M_u + \varepsilon^{-1} G_{\mathcal{A}_i^{(k)}} M_u G_{\mathcal{A}_i^{(k)}},$$

where the index i in the active set refers to the ith block corresponding to the ith grid point in time. Note that these blocks are again symmetric positive-definite matrices and the Chebyshev semiiteration can be applied. We frequently consider diagonal (lumped) mass matrices, which means that this matrix can be easily inverted at almost no computational cost.

5.2 *Schur complement*: *stationary iteration*

In Stoll & Wathen (2010) we studied all-at-once approaches for the heat equation. In contrast to our previous results where the matrix representing the one-shot discretization was a lower block-triangular matrix, we now have an additional term in the upper right corner of $\mathcal{K}$ coming from the periodicity condition. Our goal is to derive preconditioners that deal with the Schur-complement approximation $\hat{S} = \mathcal{K}\hat{\mathcal{M}}^{-1}\mathcal{K}^{\mathrm{T}}$, where $\hat{\mathcal{M}}$ represents a symmetric positive-definite approximation to $\tau\mathcal{M}_y$, e.g., in the

case of the trapezoidal rule and $\Omega_1 = \Omega$ this will simply be $\tau \mathcal{M}_y$. We now approximate $\hat{S}^{-1}$ by approximating $\mathcal{K}^{-1}$ and $\mathcal{K}^{-T}$ using the stationary iteration. The idea of a stationary iteration is rather simple as we can use a trivial identity

$$\mathcal{K}x = Ix + (\mathcal{K} - I)x = b,$$

and, rearranging the last part, we get

$$x = (I - \mathcal{K})x + b. \tag{5.2}$$

We can now turn this into an iterative method in the following way:

$$x^{(k+1)} = (I - \mathcal{K})x^{(k)} + b. \tag{5.3}$$

It is well known (see Saad, 2003) that this method converges if the eigenvalues of the matrix $I - \mathcal{K}$ lie within the unit disc; this is easily seen by subtracting (5.2) from (5.3). To improve the convergence of this approach a preconditioner $\mathcal{P}$ can be introduced:

$$\mathcal{P}^{-1}\mathcal{K}x = Ix + (\mathcal{P}^{-1}\mathcal{K} - I)x = \mathcal{P}^{-1}b,$$

resulting in the iteration

$$x^{(k+1)} = (I - \mathcal{P}^{-1}K)x^{(k)} + \mathcal{P}^{-1}b,$$

or equivalently

$$x^{(k+1)} = x^{(k)} - \mathcal{P}^{-1}r_k,$$

with $r_k = \mathcal{K}x_k - b$ the residual. Many well-known methods fit into this scheme and we refer the reader to Saad (2003) for details. For our problem we decide to use the preconditioner

$$P = \begin{bmatrix} \hat{L} & & & & \\ -M & \hat{L} & & & \\ & -M & \hat{L} & & \\ & & \ddots & \ddots & \\ & & & -M & \hat{L} \end{bmatrix}$$

for the forward PDE, where $\hat{L}$ is an approximation to the matrix $L = \tau K + M$. We will use a fixed number of algebraic multigrid (AMG) cycles as $\hat{L}^{-1}$. This approach is feasible as we are not interested in the solution of the PDE problem but only in an approximation as part of the preconditioner. Note that, for $\hat{L} = L$, this simply is the block Gauss–Seidel method. Similarly, we proceed for the adjoint PDE represented by $\mathcal{K}^{\mathrm{T}}$ with the preconditioner P^{T}. Note that, for the Schur-complement approximation to be symmetric, we need to use a right-preconditioned stationary iteration for the adjoint PDE.

It would also be possible to use other preconditioners such as the one used in Jacobi's method, i.e., a block diagonal with the blocks given by $\hat{L}$.

5.3 *Schur complement: circulant approach*

We will now focus on a different approximation of the Schur complement. For this we study the structure of the discretized forward problem defined by the one-shot operator

$$\begin{bmatrix} M+\tau K & & & & -M \\ -M & M+\tau K & & & \\ & -M & M+\tau K & & \\ & & \ddots & \ddots & \\ & & & -M & M+\tau K \end{bmatrix},$$

which can be written as

$$\mathcal{K} = I \otimes \tau K + C \otimes M, \tag{5.4}$$

with

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & -1 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & \ddots & \ddots & 0 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

a circulant matrix. It is well known from Chen (1987) that the matrix C can be diagonalized using the Fourier matrix F, i.e.,

$$C = F \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_{n_t}) F^{\mathrm{H}}.$$

If we apply the fast fourier transform (FFT) to the matrix $\mathcal{K}y = g$, we get

$$(F^{\mathrm{H}} \otimes I_{n_t})\mathcal{K}(F \otimes I_{n_t})(F^{\mathrm{H}} \otimes I_{n_t})y = (F^{\mathrm{H}} \otimes I_{n_t})g,$$

and, using the definition of $\mathcal{K}$, this becomes

$$\begin{aligned} (F^{\mathrm{H}} \otimes I_{n_t})\mathcal{K}(F \otimes I_{n_t}) &= (F^{\mathrm{H}} \otimes I_{n_t})(I \otimes \tau K + C \otimes M)(F \otimes I_{n_t}) && (5.5)\\ &= F^{\mathrm{H}}F \otimes \tau K + F^{\mathrm{H}}CF \otimes M && (5.6)\\ &= I \otimes \tau K + \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_{n_t}) \otimes M. && (5.7) \end{aligned}$$

The eigenvalues λ_j can be determined via

$$\lambda_j = c_0 + [c_1 + c_{n-1}] \cos\left(\frac{(j-1)2\pi}{k}\right) + \mathrm{i}[c_1 - c_{n-1}] \sin\left(\frac{(j-1)2\pi}{k}\right),$$

for $j = 1, \ldots, n_t$ (see Chen, 1987). In our case we get $c_1 = 0$, $c_0 = 1$ and $c_{n-1} = -1$, and hence

$$\lambda_j = 1 - \cos\left(\frac{(j-1)2\pi}{k}\right) + \mathrm{i} \sin\left(\frac{(j-1)2\pi}{k}\right).$$

All of this results in a block-diagonal matrix with the diagonal elements in the form

$$W_j = \tau K + \lambda_j M = \tau K + \left(1 - \cos\left(\frac{(j-1)2\pi}{k}\right)\right) M + \mathrm{i} \sin\left(\frac{(j-1)2\pi}{k}\right) M.$$

The matrix W_j represents one of the blocks of the block-diagonal matrix that we now have to solve for. First, we have to point out that the application of the Fourier transform will in general result in complex-valued systems. In more detail, the diagonal blocks mentioned above represent n_t complex-valued linear systems, i.e.,

$$\left(\tau K + \left(1 - \cos\left(\frac{(j-1)2\pi}{k}\right)\right) M + \mathrm{i}\sin\left(\frac{(j-1)2\pi}{k}\right) M\right)(y_\mathrm{r} + \mathrm{i}y_\mathrm{c}) = (g_\mathrm{r} + \mathrm{i}g_\mathrm{c}) \quad \forall j, \tag{5.8}$$

or equivalently

$$\begin{bmatrix} U & -V \\ V & U \end{bmatrix} \begin{bmatrix} y_\mathrm{r} \\ y_\mathrm{c} \end{bmatrix} = \begin{bmatrix} g_\mathrm{r} \\ g_\mathrm{c} \end{bmatrix}, \tag{5.9}$$

using

$$U = \tau K + \left(1 - \cos\left(\frac{(j-1)2\pi}{k}\right)\right) M \quad \text{and} \quad V = \sin\left(\frac{(j-1)2\pi}{k}\right) M.$$

The linear system can also be written in symmetric form to give

$$\begin{bmatrix} U & -V \\ -V & -U \end{bmatrix} \begin{bmatrix} y_\mathrm{r} \\ y_\mathrm{c} \end{bmatrix} = \begin{bmatrix} g_\mathrm{r} \\ -g_\mathrm{c} \end{bmatrix}. \tag{5.10}$$

Once the solution to the system (5.10) is computed we have to transform the solution back using the Fourier transform F. Note that the same has to be done for the adjoint PDE as we have to approximate the solution of both the forward and the adjoint PDE in order to approximate the Schur complement. Note that the one-shot discretization of the adjoint PDE is characterized by

$$\begin{bmatrix} M + \tau K & -M & & & & \\ & M + \tau K & -M & & & \\ & & M + \tau K & \ddots & & \\ & & & \ddots & -M & \\ -M & & & & M + \tau K & \end{bmatrix},$$

which can be written as

$$\mathcal{K} = I \otimes \tau K + \tilde{C} \otimes M, \tag{5.11}$$

with

$$\tilde{C} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & \ddots & \\ 0 & 0 & 0 & \ddots & -1 \\ -1 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Similarly to the forward PDE we see that $\tilde{C}$ is a circulant matrix, which means that we can diagonalize it using the Fourier matrix to get

$$I \otimes \tau K + \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_{n_t}) \otimes M,$$

where the eigenvalues λ are determined from

$$\lambda_j = c_0 + [c_1 + c_{n-1}]\cos\left(\frac{(j-1)2\pi}{k}\right) + \mathrm{i}[c_1 - c_{n-1}]\sin\left(\frac{(j-1)2\pi}{k}\right),$$

with $c_0 = 1$, $c_1 = -1$ and $c_{n-1} = 0$ to give

$$\lambda_j = 1 - \cos\left(\frac{(j-1)2\pi}{k}\right) - \mathrm{i}\sin\left(\frac{(j-1)2\pi}{k}\right).$$

These are simply the complex conjugates of the eigenvalues of the forward circulant matrix. Again, we have to solve a complex linear system and we use the above-presented approach to get

$$\left(\tau K + \left(1 - \cos\left(\frac{(j-1)2\pi}{k}\right)\right)M - \mathrm{i}\sin\left(\frac{(j-1)2\pi}{k}\right)M\right)(y_r + \mathrm{i}y_c) = (g_r + \mathrm{i}g_c), \tag{5.12}$$

or equivalently

$$\begin{bmatrix} U & V \\ V & -U \end{bmatrix}\begin{bmatrix} y_r \\ y_c \end{bmatrix} = \begin{bmatrix} g_r \\ -g_c \end{bmatrix}. \tag{5.13}$$

Again, the solution to the complex linear system has to be transformed back using F via the FFT. We now discuss how to solve the linear systems associated with the complex-valued system. Note that preconditioning a matrix of block-circulant type was also recently studied for the solution of a forward time-periodic PDE (see Greidanus, 2010).

5.4 *Solving the complex linear system*

As we have already seen in the previous section, the circulant approach to both the forward and the adjoint problem leads to a complex-valued linear system. We want to solve the complex systems in their real form shown in (5.10) and (5.13). As these systems arise within an outer MINRES iteration we need the iterative solver for both systems to represent a linear operator. Note that a changing preconditioner would require a flexible method as an outer iteration (Saad, 1993). This would not be achieved in the case when a Krylov solver is used, due to its nonlinearity. Instead, we propose to use a fixed number of steps of an inexact Uzawa-type method as has already been proposed for Stokes control in both the steady (see Rees & Wathen, 2011) and the unsteady (see Stoll & Wathen, 2013) case. The main iteration of the inexact Uzawa method can be cast in the form

$$x_{k+1} = x_k + \omega\mathcal{P}^{-1}r_k,$$

which means that we need to multiply by the system matrices

$$\begin{bmatrix} U & V \\ V & -U \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} U & -V \\ -V & -U \end{bmatrix}, \tag{5.14}$$

with $U = \tau K + (1-c)M$ and $V = sM$, where s and c are abbreviations for the sine and cosine values used before. Matrices that resemble the ones used in (5.14) can be found in the numerical solution of the bidomain equations (see Pennacchio & Simoncini, 2009, 2011). There we have a two-by-two block matrix that has mass matrix plus stiffness matrix terms as diagonal blocks and the off diagonals are mass matrices. We want to use a preconditioner $\mathcal{P} = \mathrm{blkdiag}(A_0, A_1)$, where A_0 approximates the

$(1,1)$ block of (5.10) or (5.13). For the bidomain equations the choice of A_1 approximating the $(2,2)$ block also gives good results. As the Schur complement is rather complicated, one way could be to use $U + V\,\mathrm{diag}(U)V^{\mathrm{T}}$ as an approximation. Here, we will stay with the choice of an AMG method for both the $(1,1)$ and the $(2,2)$ block. In fact, in our computations we simply use the AMG that we need for the $L = \tau K + M$ block as an approximation of U. This is because we would otherwise need n_t different preconditioners for one space-time solve as all the diagonal blocks in the circulant approach are different; this would be infeasible.

5.5 *Singular constraints*

As we also consider boundary control, it has to be noted that this means that the stiffness matrix of the Laplace part can only be positive semidefinite. The operator and hence the matrix K have a one-dimensional kernel that, for the matrix K, is written as $\mathbf{1}$, the vector of all 1s of the appropriately chosen dimension, i.e., $K\mathbf{1} = 0$. It can easily be seen that the vector $[\mathbf{1}^{\mathrm{T}}, \mathbf{1}^{\mathrm{T}}, \ldots, \mathbf{1}^{\mathrm{T}}]^{\mathrm{T}}$ is in the null space of the one-shot discretization $\mathcal{K}$ of the time-dependent PDE (see (2.2)–(2.4)). Note that the saddle point system is well defined as the $(1,1)$ bock is positive definite on the null space of the constraint matrix. Problems of a similar type occur in applications such as the treatment of the hydrostatic pressure in the solution of Stokes flow (see Elman *et al.*, 2005). Hence, for singular problems we refer the reader to Elman *et al.* (2005, Section 2.3), where it is stated that iterative methods will be able to handle the singularity since any stationary method will converge as long as all nonzero eigenvalues of the iteration matrix are inside the unit disc.

In the case of the circulant approach we note that the first of the diagonal blocks will become a pure Neumann problem for both the real and the complex parts. Hence, the system we want to solve is given by

$$\begin{bmatrix} \tau K & 0 \\ 0 & \tau K \end{bmatrix} \begin{bmatrix} y_r \\ y_c \end{bmatrix} = \begin{bmatrix} g_r \\ g_c \end{bmatrix}, \tag{5.15}$$

with K a Neumann Laplacian. As a consequence we have to solve two uncoupled pure Neumann problems. The solution of pure Neumann problems is a fundamental problem in many applications and has to be treated carefully since the system matrix τK has a one-dimensional kernel spanned by $\mathbf{1}$. Bochev & Lehoucq (2005) present a review of techniques to overcome this dilemma, i.e., we instead solve the system

$$\Pi^{\mathrm{T}} K \Pi x = \Pi^{\mathrm{T}} b,$$

where Π is the projection operator

$$\Pi = I - \frac{wc^{\mathrm{T}}}{c^{\mathrm{T}} w},$$

with c being in the span of $\mathbf{1}$, and $w \in \mathbb{R}^n$ is chosen such that $c^{\mathrm{T}} w > 0$.

5.6 *Dependence on the regularization parameter*

There has recently been a surge in the development of preconditioners that show not only mesh-independent convergence behaviour but also have independence with respect to the regularization parameter β (see Schöberl & Zulehner, 2007; Takacs & Zulehner, 2011; Pearson & Wathen, 2012;

Pearson *et al.*, 2012b) and also the penalization parameter ε (see Pearson *et al.*, 2012a; Schiela & Ulbrich, 2012). In the case of distributed control, the dependence on β for many values of the parameter could be observed to be very benign for the Schur-complement approximation given by

$$\hat{S} = \tau^{-1}\hat{\mathcal{K}}\mathcal{M}\hat{\mathcal{K}}^{\mathrm{T}}.$$

For boundary control and smaller values of the regularization parameter this observation is no longer true.

In Pearson & Wathen (2012) a different approximation of the Schur complement of a time-dependent distributed control problem was introduced, namely, the approximation

$$\hat{S} = \left(K + \frac{1}{\sqrt{\beta}}M\right) M^{-1} \left(K + \frac{1}{\sqrt{\beta}}M\right), \tag{5.16}$$

which can be obtained by dropping the term $-(2/\sqrt{\beta})K$ from the Schur complement

$$S = \left(K + \frac{1}{\sqrt{\beta}}M\right) M^{-1} \left(K + \frac{1}{\sqrt{\beta}}M\right) - \frac{2}{\sqrt{\beta}}K.$$

Note that, for our problem, we can also use this technique which was recently shown to be effective for the heat equation in the standard non-time-periodic set-up (see Pearson *et al.*, 2012b). We illustrate this for the distributed control example over the whole domain $M_y = M_u = M$ and use a result of Pearson *et al.* (2012b) to show its effectiveness. The Schur complement of the time-dependent problem is given by

$$S = \tau^{-1}\mathcal{K}\mathcal{M}_y^{-1}\mathcal{K}^T + \tau\beta^{-1}\mathcal{N}\mathcal{M}_u^{-1}\mathcal{N}^{\mathrm{T}},$$

and this will now be approximated by

$$\hat{S} = \tau^{-1}(\mathcal{K} + \hat{\mathcal{M}})\mathcal{M}_y^{-1}(\mathcal{K}^{\mathrm{T}} + \hat{\mathcal{M}}). \tag{5.17}$$

The choice of $\hat{\mathcal{M}}$ will be explained now. In the distributed control case we have $\mathcal{M}_u = \mathcal{M}_y$ and $\mathcal{N} = \mathcal{M} := \mathrm{blkdiag}(M, \ldots, M)$ being block matrices containing the mass matrix M. Note that, using this notation, we have $\tau\beta^{-1}\mathcal{N}\mathcal{M}_u^{-1}\mathcal{N}^{\mathrm{T}} = \tau\beta^{-1}\mathcal{M}\mathcal{M}_y^{-1}\mathcal{M}^{\mathrm{T}}$. We want to determine $\hat{\mathcal{M}}$ such that the two terms of the Schur complement S are exactly matched. This leads us to the choice $\hat{\mathcal{M}} = (\tau/\sqrt{\beta})\mathcal{M}$ such that $\tau^{-1}\hat{\mathcal{M}}\mathcal{M}_y^{-1}\hat{\mathcal{M}} = \tau\beta^{-1}\mathcal{M}\mathcal{M}_u^{-1}\mathcal{M}^{\mathrm{T}}$. The effectiveness of this approach depends on the quality of the approximation $\hat{S}^{-1}S$ and, using Pearson *et al.* (2012b, Theorems 1 and 2), we can show that the eigenvalues of $\hat{S}^{-1}S$ are confined in the interval $[\frac{1}{2}, 1)$. In order to prove such a result we need to show that the matrix $\mathcal{K}\Delta + \Delta\mathcal{K}^{\mathrm{T}}$ is positive definite. Note that we now have $\Delta = \tau^{-1}\hat{\mathcal{M}}\mathcal{M}_y^{-1} = \tau^{-1}\mathcal{M}_y^{-1}\hat{\mathcal{M}} = \mathrm{blkdiag}(\alpha_1 I, \alpha_2 I, \ldots, \alpha_{n_t} I)$, $\alpha_j > 0$, $I \in \mathbb{R}^{n\times n}$. Here, we repeat (Pearson *et al.*, 2012b, Theorem 1) adjusted for the time-periodic set-up.

THEOREM 5.1 The matrix $\mathcal{K}\Delta + \Delta\mathcal{K}^{\mathrm{T}}$, where $\Delta = \mathrm{blkdiag}(\alpha_1 I, \alpha_2 I, \ldots, \alpha_{n_t} I)$, $\alpha_1, \ldots, \alpha_{n_t} > 0$, $\alpha_1 = \alpha_{n_t}$, $I \in \mathbb{R}^{n\times n}$, and $\mathcal{K}$ is as defined previously, is positive definite.

Proof. We show that $\mathbf{w}^{\mathrm{T}}(\mathcal{K}\Delta + \Delta\mathcal{K}^{\mathrm{T}})\mathbf{w} > 0$ for all $\mathbf{w} := [\mathbf{w}_1^{\mathrm{T}}\ \mathbf{w}_2^{\mathrm{T}}\ \cdots\ \mathbf{w}_{n_t-1}^{\mathrm{T}}\ \mathbf{w}_{n_t}^{\mathrm{T}}]^{\mathrm{T}}$ with $\mathbf{w}_1, \ldots, \mathbf{w}_{n_t} \in \mathbb{R}^n$, and

$$\Delta = \begin{bmatrix} \alpha_1 I & & \\ & \ddots & \\ & & \alpha_{n_t} I \end{bmatrix}.$$

Using the symmetry of the mass and stiffness matrices M and K,

$$\mathcal{K}\Delta + \Delta\mathcal{K}^{\mathrm{T}} = \begin{bmatrix} \Lambda_1 & -\alpha_1 M & & & & -\alpha_{n_t} M \\ -\alpha_1 M & \Lambda_2 & -\alpha_2 M & & & \\ & \ddots & \ddots & \ddots & & \\ & & -\alpha_{n_t-2} M & \Lambda_{n_t-1} & -\alpha_{n_t-1} M & \\ -\alpha_{n_t} M & & & -\alpha_{n_t-1} M & \Lambda_{n_t} \end{bmatrix},$$

where $\Lambda_j = 2\alpha_j(M + \tau K)$ for $j = 1, \ldots, n_t$, and therefore by straightforward manipulation,

$$\begin{aligned} \mathbf{w}^{\mathrm{T}}(\mathcal{K}\Delta + \Delta\mathcal{K}^{\mathrm{T}})\mathbf{w} &= \sum_{j=1}^{n_t} 2\alpha_j \mathbf{w}_j^{\mathrm{T}}[M + \tau K]\mathbf{w}_j - \sum_{j=1}^{n_t-1} \alpha_j \mathbf{w}_j^{\mathrm{T}} M \mathbf{w}_{j+1} - \sum_{j=2}^{n_t} \alpha_{j-1} \mathbf{w}_j^{\mathrm{T}} M \mathbf{w}_{j-1} \\ &\quad - \alpha_{n_t} \mathbf{w}_1^{\mathrm{T}} M \mathbf{w}_{n_t} - \alpha_{n_t} \mathbf{w}_{n_t}^{\mathrm{T}} M \mathbf{w}_1 \qquad (5.18) \\ &= 2\tau \sum_{j=1}^{n_t} \alpha_j \mathbf{w}_j^{\mathrm{T}}(K)\mathbf{w}_j + \sum_{j=1}^{n_t-1} \alpha_j (\mathbf{w}_j - \mathbf{w}_{j+1})^{\mathrm{T}} M (\mathbf{w}_j - \mathbf{w}_{j+1}) \\ &\quad + \alpha_1 \mathbf{w}_1^{\mathrm{T}} M \mathbf{w}_1 + \alpha_{n_t} \mathbf{w}_{n_t}^{\mathrm{T}} M \mathbf{w}_{n_t} - \alpha_{n_t} \mathbf{w}_1^{\mathrm{T}} M \mathbf{w}_{n_t} - \alpha_{n_t} \mathbf{w}_{n_t}^{\mathrm{T}} M \mathbf{w}_1 \qquad (5.19) \\ &= 2\tau \sum_{j=1}^{n_t} \alpha_j \mathbf{w}_j^{\mathrm{T}} K \mathbf{w}_j + \sum_{j=1}^{n_t-1} \alpha_j (\mathbf{w}_j - \mathbf{w}_{j+1})^{\mathrm{T}} M (\mathbf{w}_j - \mathbf{w}_{j+1}) \\ &\quad + \alpha_1 (\mathbf{w}_1 - \mathbf{w}_{n_t})^{\mathrm{T}} M (\mathbf{w}_1 - \mathbf{w}_{n_t}), \qquad (5.20) \end{aligned}$$

where we have used the fact that $\alpha_1 = \alpha_{n_t}$. As we now have that $\mathbf{w}^{\mathrm{T}}(\mathcal{K}\Delta + \Delta\mathcal{K}^{\mathrm{T}})\mathbf{w}$ is a sum of positive multiples of (symmetric positive-definite) mass and stiffness matrices, we deduce that $\mathbf{w}^{\mathrm{T}}(\mathcal{K}\Delta + \Delta\mathcal{K}^{\mathrm{T}})\mathbf{w} > 0$, and hence that $\mathcal{K}\Delta + \Delta\mathcal{K}^{\mathrm{T}}$ is positive definite. □

We have now shown that the cross terms from the Schur-complement approximation generate a positive-definite matrix and (Pearson *et al.*, 2012b, Theorem 2) can be applied, which means that the eigenvalues of the preconditioned Schur complement $\hat{S}^{-1}S$ are confined in $[\frac{1}{2}, 1)$ independently of the parameters h and β.

The set-up for the boundary control case is much more intricate and the results tend to be less rigorous. Here, we only introduce the approximations for this case, and illustrate their competitiveness with our numerical results in Section 6. We again construct a preconditioner of the form

$$\hat{S} = \tau^{-1}(\mathcal{K} + \hat{\mathcal{M}})\mathcal{M}_y^{-1}(\mathcal{K}^{\mathrm{T}} + \hat{\mathcal{M}}), \qquad (5.21)$$

where $\hat{\mathcal{M}} = \mathrm{blkdiag}(0, \tau\sqrt{(h/\beta)}\mathcal{M}_u)$ if we assume that the degrees of freedom corresponding to nodes on the boundary are ordered so that they appear in the last components. Note that h is the mesh size and

this scaling has to be introduced to compensate for the different order of the boundary mass matrix and the mass matrix over the whole domain. We remember at this stage that $\mathcal{N}\mathcal{M}_u^{-1}\mathcal{N}^{\mathrm{T}}$ will be a diagonal matrix with $\mathcal{M}_u$ a block-diagonal matrix consisting of lumped boundary mass matrices.

One important fact about these new Schur-complement approximations is that the preconditioning strategies mentioned earlier, i.e., stationary iteration and the circulant approach, apply with almost no changes to this case since the matrices $\hat{\mathcal{M}}$ only introduce positive-definite perturbations of the diagonal blocks of $\mathcal{K}$. Hence, we will not discuss how to implement these new approximations efficiently but rather use them immediately in Section 6.

5.7 *Preconditioning the control constraints*

We have already introduced a preconditioner for the $(1,1)$ block in the case of control constraints but now need to focus on an efficient approximation of the Schur complement

$$S = \tau^{-1}\mathcal{K}\mathcal{M}_y^{-1}\mathcal{K}^{\mathrm{T}} + \tau\mathcal{N}\bar{\mathcal{M}}_u^{-1}\mathcal{N}^{\mathrm{T}},$$

with $\bar{\mathcal{M}}_u := \beta\mathcal{M}_u + \varepsilon^{-1}\mathcal{G}_{\mathcal{A}^{(k)}}\mathcal{M}_u\mathcal{G}_{\mathcal{A}^{(k)}}$. As in the previous section we want to construct an approximation

$$\hat{S} = \tau^{-1}(\mathcal{K} + \hat{\mathcal{M}})\mathcal{M}_y^{-1}(\mathcal{K}^{\mathrm{T}} + \hat{\mathcal{M}}),$$

where $\hat{\mathcal{M}}$ is chosen such that the terms $\tau^{-1}\hat{\mathcal{M}}\mathcal{M}_y^{-1}\hat{\mathcal{M}}$ and $\tau\mathcal{N}\bar{\mathcal{M}}_u^{-1}\mathcal{N}^{\mathrm{T}}$ match. For the individual blocks of these block-diagonal matrices this means that

$$\tau^{-1}\hat{M}_i M_y^{-1}\hat{M}_i \approx \tau N\bar{M}_{i,u}^{-1}N^{\mathrm{T}}, \tag{5.22}$$

where $\bar{M}_{i,u} = \beta M_u + \varepsilon^{-1}G_{\mathcal{A}_i^{(k)}}M_u G_{\mathcal{A}_i^{(k)}}$. The matrices $G_{\mathcal{A}_i^{(k)}}$ represent the generalized derivatives with respect to the active sets associated with grid point i in time. Note that we also introduced the index i for the blocks of $\hat{\mathcal{M}}$ since each of these will vary corresponding to the active sets associated with grid point i in time. Note that $N\bar{M}_{i,u}^{-1}N^{\mathrm{T}}$ is a diagonal matrix with nonzero entries corresponding to the boundary degrees of freedom. As all the matrices in (5.22) are diagonal, and ignoring for now the different scalings between boundary mass matrices and mass matrices over the whole domain, we get the following expression for the nonzero entries of $\hat{M}_i$:

$$m_{y,jj}^{-1}\hat{m}_{i,jj}^2 = \frac{\tau^2 m_{u,jj}^2}{\bar{m}_{ui,jj}} \quad \Rightarrow \quad \hat{m}_{i,jj} = \tau\sqrt{\frac{m_{y,jj}m_{u,jj}^2}{\bar{m}_{ui,jj}}},$$

where the subscripts y and u refer to entries from the state and control mass matrices, respectively. We have already mentioned that the boundary mass matrix scales differently compared with the mass matrix on the whole domain by one order of h, e.g., using the approximations $M_y \approx h^2 I$ and $M_u \approx hI$. Using the fact that, roughly speaking, $m_{y,jj} \approx hm_{ujj}$, we change the previous approximation to

$$\hat{m}_{i,jj} = \tau\sqrt{h}\sqrt{\frac{m_{u,jj}^3}{\bar{m}_{ui,jj}}},$$

with $\bar{m}_{ui,jj} = \beta m_{u,jj}$ for the free variables and $\bar{m}_{ui,jj} = \beta m_{u,jj} + \varepsilon^{-1}m_{u,jj}$ for degrees of freedom in the ith active set.

6. Numerical experiments

6.1 *Set-up and implementation details*

In this section we provide numerical experiments for the methods presented above. For the discretization we use the deal.II library (see Bangerth *et al.*, 2007), which is implemented in C++ using quadrilateral elements. As deal.II provides easy access to the Trilinos ML AMG package, our multigrid approximations were performed using Trilinos' smoothed aggregation preconditioners (see Gee *et al.*, 2006). We approximated the blocks involving L as the sum of a mass matrix and stiffness matrix, also the matrices possibly involving scalar factors in front of M or K, by a fixed number of steps of an AMG V-cycle and typically 10 steps of a Chebyshev smoother, which has proved to be effective for symmetric matrices representing discretized elliptic operators (see Gee *et al.*, 2006). The stationary iteration-based preconditioner uses a fixed number of steps for both the adjoint and the forward problem. The circulant-based preconditioner also uses a small but fixed number of steps of the inexact Uzawa method for every complex linear system. The application of the FFT needed for the circulant approach was provided by employing FFTW (see Frigo & Johnson, 1998, 2005). We use a relative tolerance of 10^{-4} for the pseudo-residual $\|r_k\|_{P^{-1}}$ and $\tau = 0.05$ with $T = 1$, i.e., $n_t = 20$ unless mentioned otherwise. All experiments are performed on a Centos Linux machine with Intel(R) Xeon(R) CPU X5650 @ 2.67 GHz CPUs and 48 GB of RAM.

6.2 *Distributed control*

The first example is a distributed control example with zero Dirichlet boundary condition. The desired state is given by

$$\bar{\mathbf{y}}(t) = 2^{10t} x_0 x_2 x_1 (x_0 - 1)(x_1 - 1)(x_2 - 1),$$

which for $\bar{\mathbf{y}}_{10}$ (the desired state at grid point 10 in time) is depicted in Fig. 1(a). Figure 1(b) shows a spherical slice of the computed state that approximates $\bar{\mathbf{y}}_{10}$.

Note that the Dirichlet boundary condition $y = 0$ on $\partial\Omega$ could force the state to differ drastically from the desired state on the boundary. Better results are typically obtained if the desired state and the Dirichlet boundary condition coincide, i.e., in Rees *et al.* (2010b) the Dirichlet boundary condition is chosen to match boundary values of the desired state. These choices clearly depend on the underlying problem and the requirements coming from an application and cannot be altered for the sake of better numerical behaviour. The boundary control problem is often more relevant for practical applications and we show results for this case in the next subsection. Here, we compare the results of the stationary iteration-based preconditioner and the circulant preconditioner, both in their more robust form introduced in Section 5.6. Table 1 gives the results for the stationary iteration and Table 2 for the circulant-based approach. In both cases we look at the performance over a variety of meshes and different values of the regularization parameter β. It can be seen that both methods perform in a robust manner with respect to mesh size and β values. Also, both methods give comparable iteration numbers. The number of degrees of freedom given in the Tables 1 and 2 represents the size of the whole-saddle point system. The difference in timings is due to the fact that we perform two stationary iterations versus two inexact Uzawa steps for every complex system. The circulant preconditioner needs to solve n_t complex systems and each of these solves uses a Uzawa method with a double-sized matrix and two applications of the preconditioner.

In Table 3 we show results for a fixed mesh with 294780 unknowns for a large variety of values of the regularization parameter β.

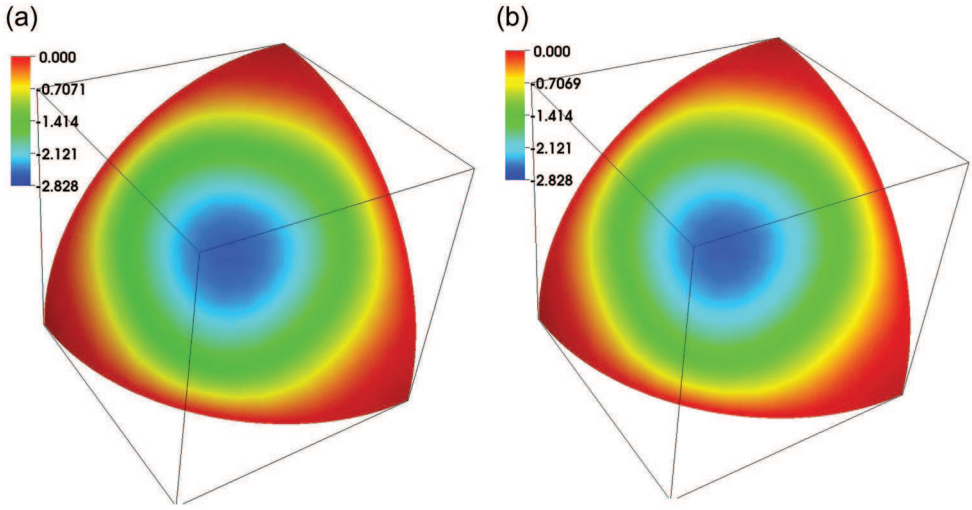


FIG. 1. Spherical slice of the desired and computed state at grid point 10τ in time for the distributed control problem evaluated with the regularization parameter $\beta = 1e{-}4$. (a) $\bar{y}_{10}$ and (b) y_{10}.

TABLE 1 *Stationary iteration preconditioner*: *MINRES iterations and timings for various meshes and values of the regularization parameter β applied to a distributed control problem*

Degrees of freedom	MINRES(t), $\beta = 1e{-}2$,	MINRES(t), $\beta = 1e{-}4$	MINRES(t), $\beta = 1e{-}6$
43740	8 (1)	6 (1)	4 (1)
294780	8 (4)	8 (4)	6 (4)
2156220	8 (29)	8 (28)	6 (23)
16477500	8 (297)	8 (298)	6 (213)

TABLE 2 *Circulant preconditioner*: *MINRES iterations and timings for various meshes and values of the regularization parameter β applied to a distributed control problem*

Degrees of freedom	MINRES(t), $\beta = 1e{-}2$	MINRES(t), $\beta = 1e{-}4$	MINRES(t), $\beta = 1e{-}6$
43740	12 (2)	8 (2)	4 (1)
294780	12 (12)	8 (9)	6 (6)
2156220	12 (86)	8 (61)	6 (51)
16477500	12 (924)	8 (673)	6 (550)

TABLE 3 *Stationary iteration and circulant preconditioner*: *MINRES iterations and timings for various values of the regularization parameter β applied to a distributed control problem on a system with* 294780 *unknowns*

Degrees of freedom	MINRES(t), $\beta = 1e{-}2$	MINRES(t), $\beta = 1e{-}4$	MINRES(t), $\beta = 1e{-}6$	MINRES(t), $\beta = 1e{-}8$	MINRES(t), $\beta = 1e{-}10$
Stationary iteration	8 (12)	8 (13)	6 (9)	2 (5)	2 (5)
Circulant	12 (48)	8 (34)	6 (28)	2 (13)	2 (14)

TABLE 4 *Stationary iteration preconditioner: MINRES iterations and timings for various meshes and values of the regularization parameter β applied to a distributed control problem*

	Stationary iteration		Circulant	
Degrees of freedom	MINRES(t), $\beta = 1e{-}4$	MINRES(t), $\beta = 1e{-}6$	MINRES(t), $\beta = 1e{-}4$	MINRES(t), $\beta = 1e{-}6$
143740	28 (7)	22 (6)	26 (12)	20 (9)
294780	40 (37)	30 (28)	32 (81)	26 (67)
2156220	50 (198)	40 (163)	40 (443)	36 (405)

In Table 4 we show the comparison of both Schur-complement approximations when the number of stationary iterations is reduced to one and also the Uzawa method for the complex linear systems using only one iteration.

6.3 *Boundary control*

Our next task is to illustrate the performance of the preconditioners proposed for the boundary control problem presented earlier. In the case of boundary control it was noted previously (see Pearson *et al.*, 2012b) that the approximation for the Schur complement given by

$$\tau^{-1}\mathcal{K}\mathcal{M}^{-1}\mathcal{K}^{\mathrm{T}}$$

is often not sufficient to guarantee convergence within a reasonable number of iterations. Hence, we developed the preconditioners presented in Section 5.6 that can use all the techniques presented for the original approximation as the structure for both the stationary iteration and also the circulant structure remain untouched. We will now illustrate the performance of the new Schur-complement approximation again for the stationary iteration and the circulant approaches. We compare the two different approximations for S given a three-dimensional example defined by the following desired state:

$$\bar{\mathbf{y}} = \begin{cases} \sin(t) + x_0 x_1 x_2, & x_0 > 0.5 \text{ and } x_1 < 0.5, \\ 1 & \text{otherwise.} \end{cases}$$

Figure 2(a) shows the desired state $\bar{\mathbf{y}}_{10}$, and Fig. 2(b,c) show the computed state $\mathbf{y}_{10}$ and control $\mathbf{u}_{10}$, respectively. In Tables 5 and 6 we again show results for the stationary and the circulant iteration approach, respectively. Again, both methods perform rather robustly with respect to the mesh parameter and the regularization parameter. We believe that the growth in iteration numbers is due to the low rank nature of the second term in the Schur complement involving the rectangular matrices $\mathcal{N}$. Also, the discretized Laplacian is now only positive semidefinite as we are dealing with a pure Neumann problem. Nevertheless, the iteration numbers are consistently low for linear systems with several million unknowns in three dimensions.

We now want to investigate the effect of reducing the number of stationary iterations/Uzawa iterations for both Schur-complement approximations. We need to have a fixed number of steps during one iteration to solve the linear system since otherwise we would have a preconditioner that changes in between two steps, which requires the use of flexible Krylov solvers. Thus, we will now investigate reducing the number of iteration steps to one for both the stationary iteration (see Table 7) and the Uzawa method for the complex linear system coming from the circulant preconditioner (see Table 8).

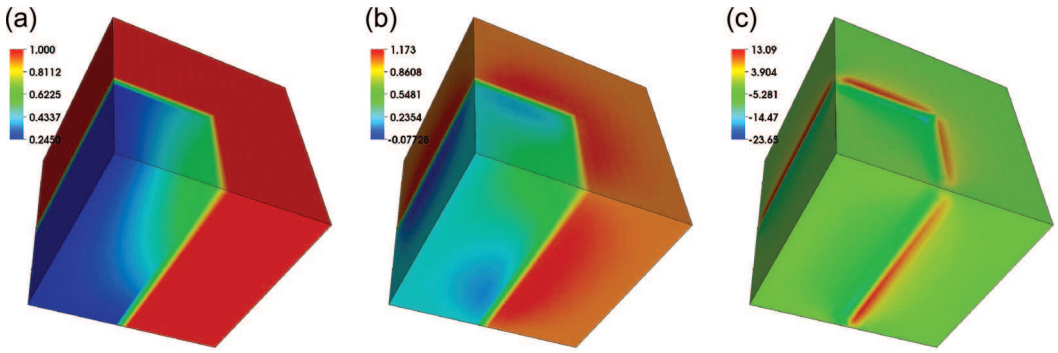


FIG. 2. Desired state, computed state and computed control at grid point 10τ in time for the boundary control problem. The regularization parameter was fixed at $\beta = 1e{-}4$. (a) $\bar{\mathbf{y}}_{10}$, (b) $\mathbf{y}_{10}$ and (c) $\mathbf{u}_{10}$.

TABLE 5 *Stationary iteration preconditioner*: *MINRES iterations and timings for various meshes and values of the regularization parameter β applied to a boundary control problem*

Degrees of freedom	MINRES(t), $\beta = 1e{-}2$	MINRES(t), $\beta = 1e{-}4$	MINRES(t), $\beta = 1e{-}6$
36880	30 (10)	26 (9)	20 (7)
227280	32 (45)	38 (54)	28 (40)
1560400	38 (233)	48 (290)	38 (233)
11476560	48 (2519)	62 (3216)	58 (2995)

TABLE 6 *Circulant preconditioner*: *MINRES iterations and timings for various meshes and values of the regularization parameter β applied to a boundary control problem. The $*$ indicates a segmentation fault from the n1_7 function of FFTW* (Frigo & Johnson, 2005), *which might be triggered by our implementation*

Degrees of freedom	MINRES(t), $\beta = 1e{-}2$	MINRES(t), $\beta = 1e{-}4$	MINRES(t), $\beta = 1e{-}6$
36880	26 (19)	24 (18)	18 (13)
227280	30 (130)	36 (158)	26 (113)
1560400	36 (656)	48 (866)	38 (696)
11476560	42 (4775)	*	*

TABLE 7 *Stationary iteration preconditioner*: *MINRES iterations and timings for various meshes and values of the regularization parameter β applied to a boundary control problem with only one step for the stationary iteration*

Degrees of freedom	MINRES(t), $\beta = 1e{-}2$	MINRES(t), $\beta = 1e{-}4$	MINRES(t), $\beta = 1e{-}6$
36880	30 (7)	26 (7)	20 (6)
227280	34 (32)	38 (36)	28 (26)
1560400	38 (154)	48 (194)	40 (161)
11476560	48 (1575)	62 (2022)	60 (2124)

TABLE 8 *Circulant preconditioner*: *MINRES iterations and timings for various meshes and values of the regularization parameter β applied to a boundary control problem with only one step for the Uzawa iteration*

Degrees of freedom	MINRES(t), $\beta = 1e{-}2$	MINRES(t), $\beta = 1e{-}4$	MINRES(t), $\beta = 1e{-}6$
36880	30 (14)	26 (12)	20 (9)
227280	32 (83)	32 (82)	26 (67)
1560400	34 (384)	40 (450)	36 (403)
11476560	36 (2671)	*	*

TABLE 9 *Circulant preconditioner*: *average number of MINRES iterations per Newton step and timings for various meshes and values of the penalty parameter ε applied to a control-constrained boundary control problem. The upper bound is $\bar{u} = 0.1$ and $\beta = 1\text{e}{-}2$ is fixed throughout. We AS(t) denotes the number of active set iterations and the time taken*

Degrees of freedom	AS(t)	MINRES average	AS(t)	MINRES average	AS(t)	MINRES average
	$\varepsilon = 1e{-}2$		$\varepsilon = 1e{-}4$		$\varepsilon = 1e{-}6$	
36880	6 (451)	41.2	6 (399)	41.2	6 (321)	34.0
227280	6 (2476)	55.0	6 (1964)	43.3	6 (1654)	36.0
1560400	7 (16794)	60.5	7 (12857)	46.0	7 (10673)	37.4

6.4 *Boundary control with box constraints*

We finally want to present results for the case of boundary control in the presence of box constraints. The desired state $\bar{\mathbf{y}}$ is defined by

$$\begin{cases} \sin(t(1-t)) + x_0 x_1 x_2 & \text{for } x_0 > 0.5 \text{ and } x_1 < 0.5, \\ 1 & \text{otherwise.} \end{cases} \tag{6.1}$$

This is the same set-up as was shown in Fig. 2. As the control changes with varying β we will, in this section, work with a fixed regularization parameter $\beta = 1\text{e}{-}2$. We decide to only work with an upper bound fixed at $u_b = 0.10$. Since the performance of the Newton iteration generally depends on the quality of the preconditioner (see Kanzow, 2004) we use the relative tolerance of $1\text{e}{-}6$ for which we always observed good behaviour of the outer Newton iteration. In the case where strict complementarity is encountered, which we will not discuss here, a strategy proposed in Bergounioux *et al.* (1999) is to replace the upper bound $>\bar{u}$ by $>\bar{u} - \delta$ where δ is of the order of machine precision. As our focus is on the solution of the linear systems we believe that these algorithmic changes will not greatly affect the performance of the iterative solver for the inner linear system but rather the outer Newton iteration. Based on Kanzow (2004) this might require a more accurate solve of the linear system, which we show in this section tends to produce satisfying results.

Please note that the timings for the results shown here are worse than the timings without control constraints. This is due to the fact that as a proof of concept we have recomputed the AMG preconditioner for every application involving a different active set. To effectively use this preconditioner in the future the recomputation of the preconditioner needs to be avoided and other strategies such as stationary iterations should be used.

TABLE 10 *Stationary iteration preconditioner: average number of MINRES iterations per Newton step and timings for various meshes and values of the penalty parameter ε applied to a control-constrained boundary control problem. The upper bound is $\bar{u} = 0.1$ and $\beta = 1e{-}2$ is fixed throughout*

Degrees of freedom	AS(t)	MINRES average	AS(t)	MINRES average	AS(t)	MINRES average
	$\varepsilon = 1e{-}2$		$\varepsilon = 1e{-}4$		$\varepsilon = 1e{-}6$	
36880	6 (428)	40.3	6 (372)	34.7	6 (320)	29.7
227280	6 (2002)	48.3	6 (1760)	41.7	6 (1454)	34.7
1560400	7 (17006)	57.4	7 (14998)	50.0	7 (11897)	39.7

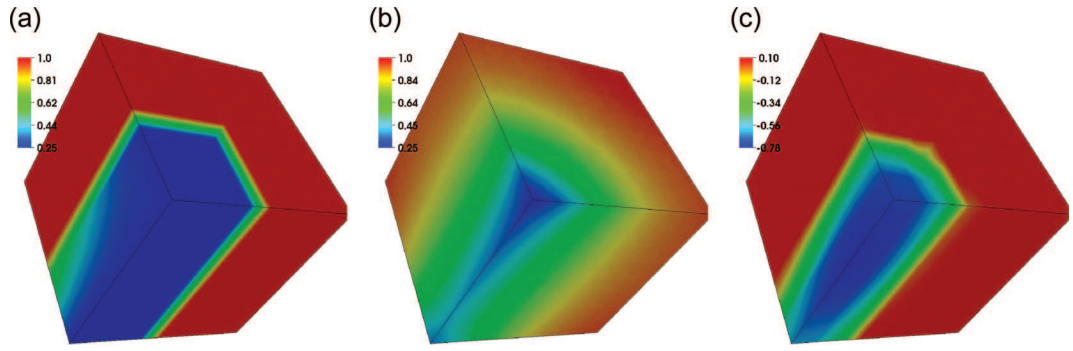


FIG. 3. Desired state, computed state and computed control at grid point 10τ in time for the control-constrained boundary control problem. The upper bound is given by $\bar{u} = 0.1$. The regularization parameter was fixed at $\beta = 1e{-}2$. (a) $\bar{\mathbf{y}}_{10}$, (b) $\mathbf{y}_{10}$ and (c) $\mathbf{u}_{10}$.

In Table 9 we show results for the active set method using the circulant preconditioner on a combination of various meshes and various values of the penalty parameter ε. The same set-up is shown in Table 10 when the stationary iteration preconditioner is used. Both methods perform very similarly in terms of iteration numbers and timings. Here, we used three steps for the stationary iteration preconditioner and only two steps of the Uzawa method for the circulant approach. The iteration numbers are constantly low and very little mesh dependency can be observed [Fig. 3(a,b,c)]. The results for an example with control constraints are shown in Fig. 3.

7. Conclusions and outlook

In this paper we presented a monolithic solver for the space-time discretization of an optimal control problem subject to the time-periodic heat equation. We formulated a one-shot approach that resulted in a huge linear system in saddle point form. With our choice of MINRES as the Krylov subspace solver we were focusing on the task of devising good preconditioners for the Schur complement of the saddle point matrix. We proposed two techniques, one based on a stationary iteration and the other on a circulant formulation that allowed the use of the FFT. Both methods performed competitively for distributed and boundary control problems. We introduced a Schur-complement approximation that allowed more flexibility with respect to the regularization parameter β. The efficient solution of the control problem also enabled the fast solution of the minimization when control constraints are present. We showed the results for boundary control with box constraints on the control and illustrated the flexibility of our approach.

One point that needs to be addressed in future research is a more efficient implementation of the Schur-complement preconditioner that avoids recomputation of the AMG approximation. Also, it would be interesting to investigate scenarios where the circulant preconditioner is able to outperform the stationary iteration approximation. We believe that higher-order discretizations in time that lead to a more complicated matrix structure can benefit from the circulant approach as, in this case, a larger number of stationary iterations might be needed to efficiently approximate the Schur complement.

Acknowledgements

The author would like to thank Andy Wathen for his comments and Daniel Kressner for pointing him to some references. He would also like to thank the anonymous referees for their valuable comments that have helped to improve the quality of the paper.

References

Abbeloos, D., Diehl, M., Hinze, M. & Vandewalle, S. (2011) Nested multigrid methods for time-periodic, parabolic optimal control problems. *Comput. Vis. Sci.*, **14**, 27–38.

Bangerth, W., Hartmann, R. & Kanschat, G. (2007) Deal. II – a general-purpose object-oriented finite element library. *ACM Trans. Math. Softw.*, **33**, Art. 24, 27.

Benzi, M., Haber, E. & Taralli, L. (2011) A preconditioning technique for a class of PDE-constrained optimization problems. *Adv. Comput. Math.*, **35**, 149–173.

Bergounioux, M., Ito, K. & Kunisch, K. (1999) Primal-dual strategy for constrained optimal control problems. *SIAM J. Control Optim.*, **37**, 1176–1194.

Bergounioux, M. & Kunisch, K. (2002) Primal-dual strategy for state-constrained optimal control problems. *Comput. Optim. Appl.*, **22**, 193–224.

Bochev, P. & Lehoucq, R. (2005) On the finite element solution of the pure Neumann problem. *SIAM Rev.*, **47**, 50–66.

Bomhof, C. (2001) Iterative and parallel methods for linear systems, with applications in circuit simulation. *Ph.D. Thesis*, Universiteit Utrecht.

Bomhof, W. & van der Vorst, H. (2001) A parallelizable GMRES-type method for *p*-cyclic matrices, with applications in circuit simulation. *Scientific Computing in Electrical Engineering: Proceedings of the 3rd International Workshop*, 20–23 August 2000, Warnemünde, Berlin, Germany. Springer, p. 293.

Bramble, J. H. & Pasciak, J. E. (1988) A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems. *Math. Comp.*, **50**, 1–17.

Chen, M. (1987) On the solution of circulant linear systems. *SIAM J. Numer. Anal.*, **24**, 668–683.

Davis, T. (2005) Umfpack version 4.4 user guide. *Technical Report*. Dept. of Computer and Information Science and Engineering Univ. of Florida, Gainesville, FL.

Duff, I. (1996) Sparse numerical linear algebra: direct methods and preconditioning. *Technical Report* TR/PA/96/22. CERFACS, Toulouse, France. Also RAL Report RAL 96-047.

Elman, H. C., Silvester, D. J. & Wathen, A. J. (2005) Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics. *Numerical Mathematics and Scientific Computation*. New York: Oxford University Press, pp. xiv+400.

Ernst, O. (2000) Equivalent iterative methods for *p*-cyclic matrices. *Numer. Algorithms*, **25**, 161–180.

Frigo, M. & Johnson, S. G. (1998) FFTW: an adaptive software architecture for the FFT. *Proc. 1998 IEEE International Conference on Acoustics Speech and Signal Processing*, vol. 3. Seattle, IEEE, pp. 1381–1384.

Frigo, M. & Johnson, S. G. (2005) The design and implementation of FFTW3. *Proc. IEEE*, **93**, 216–231. Special issue on 'Program Generation, Optimization, and Platform Adaptation'.

Gee, M., Siefert, C., Hu, J., Tuminaro, R. & Sala, M. (2006) ML 5.0 smoothed aggregation user's guide. *Technical Report* SAND2006-2649. Sandia National Laboratories.

Greidanus, J. W. (2010) Efficient computation of periodic orbits in space-time discretized nonlinear dynamical systems. *Master's Thesis*, University of Groningen.

Herzog, R. & Sachs, E. W. (2010) Preconditioned conjugate gradient method for optimal control problems with control and state constraints. *SIAM J. Matrix Anal. Appl.*, **31**, 2291–2317.

Hestenes, M. R. & Stiefel, E. (1952) Methods of conjugate gradients for solving linear systems. *J. Res. Nat. Bur. Stand.*, **49**, 409–436.

Hintermüller, M., Ito, K. & Kunisch, K. (2002) The primal-dual active set strategy as a semismooth Newton method. *SIAM J. Optim.*, **13**, 865–888.

Hinze, M. (2005) A variational discretization concept in control constrained optimization: the linear-quadratic case. *Comput. Optim. Appl.*, **30**, 45–61.

Hinze, M., Köster, M. & Turek, S. (2008) A hierarchical space-time solver for distributed control of the Stokes equation. *Technical Report*. TU Dortmund, Germany, SPP1253-16-01.

Hinze, M., Pinnau, R., Ulbrich, M. & Ulbrich, S. (2009) Optimization with PDE constraints. *Mathematical Modelling: Theory and Applications*. New York: Springer.

Ito, K. & Kunisch, K. (2008) *Lagrange Multiplier Approach to Variational Problems and Applications*. Advances in Design and Control, vol. 15. Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM), pp. xviii+341.

Kanzow, C. (2004) Inexact semismooth Newton methods for large-scale complementarity problems. *Optim. Methods Softw.*, **19**, 309–325.

Kawajiri, Y. & Biegler, L. (2006) Optimization strategies for simulated moving bed and powerfeed processes. *AIChE J.*, **52**, 1343–1350.

Kummer, B. (1988) Newton's method for non-differentiable functions. *Math. Res.*, **45**, 114–125.

Murphy, M. F., Golub, G. H. & Wathen, A. J. (2000) A note on preconditioning for indefinite linear systems. *SIAM J. Sci. Comput.*, **21**, 1969–1972.

Paige, C. C. & Saunders, M. A. (1975) Solutions of sparse indefinite systems of linear equations. *SIAM J. Numer. Anal.*, **12**, 617–629.

Pang, J.-S. (1990) Newton's method for B-differentiable equations. *Math. Oper. Res.*, **15**, 311–341.

Pearson, J. W., Stoll, M. & Wathen, A. (2012a) Preconditioners for state constrained optimal control problems with Moreau–Yosida penalty function. *Numer. Linear Algebra Appl.*, to appear.

Pearson, J. W., Stoll, M. & Wathen, A. J. (2012b) Regularization-robust preconditioners for time-dependent PDE-constrained optimization problems. Numerical Analysis Group, University of Oxford, NA-10-13. *SIAM J. Matrix Anal. Appl.*, **33**, 1126–1152.

Pearson, J. W. & Wathen, A. J. (2012) A new approximation of the Schur complement in preconditioners for PDE-constrained optimization. *Numer. Linear Algebra Appl.*, **19**, 816–829.

Pennacchio, M. & Simoncini, V. (2009) Algebraic multigrid preconditioners for the bidomain reaction–diffusion system. *Appl. Numer. Math.*, **59**, 3033–3050.

Pennacchio, M. & Simoncini, V. (2011) Fast structured AMG preconditioning for the bidomain model in electrocardiology. *SIAM J. Sci. Comput.*, **33**, 721–745.

Potschka, A., Mommer, M., Schlöder, J. & Bock, H. (2012) A Newton–Picard approach for efficient numerical solution of time-periodic parabolic PDE constrained optimization problems. *SIAM J. Sci. Comput.*, **34**, A1214–A1239.

Qi, L. Q. & Sun, J. (1993) A nonsmooth version of Newton's method. *Math. Program.*, **58**, 353–367.

Rees, T. & Stoll, M. (2010) Block-triangular preconditioners for PDE-constrained optimization. *Numer. Linear Algebra Appl.*, **17**, 977–996.

Rees, T., Stoll, M. & Wathen, A. (2010a) All-at-once preconditioners for PDE-constrained optimization. *Kybernetika*, **46**, 341–360.

Rees, T., Dollar, H. S. & Wathen, A. J. (2010b) Optimal solvers for PDE-constrained optimization. *SIAM J. Sci. Comput.*, **32**, 271–298.

Rees, T. & Wathen, A. (2011) Preconditioning iterative methods for the optimal control of the Stokes equation. *SIAM J. Sci. Comput.*, **33**, 2903–2926.

SAAD, Y. (1993) A flexible inner-outer preconditioned GMRES algorithm. *SIAM J. Sci. Comput.*, **14**, 461–469.
SAAD, Y. (2003) *Iterative Methods for Sparse Linear Systems*. Philadelphia, PA: Society for Industrial and Applied Mathematics, pp. xviii+528.
SCHIELA, A. & ULBRICH, S. (2012) Operator preconditioning for a class of constrained optimal control problems (submitted).
SCHÖBERL, J. & ZULEHNER, W. (2007) Symmetric indefinite preconditioners for saddle point problems with applications to PDE-constrained optimization problems. *SIAM J. Matrix Anal. Appl.*, **29**, 752–773.
SIMONCINI, V. (2012) Reduced order solution of structured linear systems arising in certain PDE-constrained optimization problems. *Comput. Optim. Appl.*, to appear.
STOLL, M. & WATHEN, A. (2010) All-at-once solution of time-dependent PDE-constrained optimization problems. Numerical Analysis Group, University of Oxford, NA-10-13.
STOLL, M. & WATHEN, A. (2012) Preconditioning for partial differential equation constrained optimization with control constraints. *Numer. Linear Algebra Appl.*, **19**, 53–71.
STOLL, M. & WATHEN, A. (2013) All-at-once solution of time-dependent Stokes control. *J. Comput. Phys.*, **232**, 498–515.
TAKACS, S. & ZULEHNER, W. (2011) Convergence analysis of multigrid methods with collective point smoothers for optimal control problems. *Comput. Vis. Sci.*, **14**, 131–141.
TRÖLTZSCH, F. (2010) *Optimal Control of Partial Differential Equations: Theory, Methods and Applications*. Providence, RI: American Mathematical Society.
ULBRICH, M. (2011) *Semismooth Newton Methods for Variational Inequalities and Constrained Optimization Problems*. Philadelphia: SIAM.
VANDEWALLE, S. & PIESSENS, R. (1992) Efficient parallel algorithms for solving initial-boundary value and time-periodic parabolic partial differential equations. *SIAM J. Sci. Statist. Comput.*, **13**, 1330.

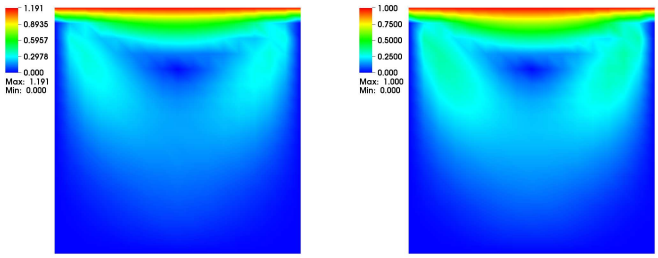
A.5 Preconditioning for the Stokes equations

This paper is published as

M. Stoll and A. Wathen, ***All-at-once solution of time-dependent Stokes control*, J. Comput. Phys., 232 (2013), pp. 498–515.**

Result from the paper

In this paper we develop robust iterative solvers for the optimal control of Stokes equations. The techniques are then applied to two and three-dimensional examples. A two-dimensional setup is shown in Figure A.1 where we consider a lid-driven cavity example.



(a) State (b) Desired state

Figure A.1: State and desired state

Journal of Computational Physics 232 (2013) 498–515

Contents lists available at SciVerse ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

All-at-once solution of time-dependent Stokes control

Martin Stoll [a,*], Andy Wathen [b,1]

[a] Computational Methods in Systems and Control Theory, Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, 39106 Magdeburg, Germany
[b] Numerical Analysis Group, Mathematical Institute, 24–29 St Giles', Oxford, OX1 3LB, United Kingdom

ARTICLE INFO

Article history:
Received 8 June 2011
Received in revised form 24 July 2012
Accepted 22 August 2012
Available online 6 September 2012

Keywords:
Saddle point problems
Unsteady Stokes equation
PDE-constrained optimization
Preconditioning

ABSTRACT

The solution of time-dependent PDE-constrained optimization problems subject to unsteady flow equations presents a challenge to both algorithms and computing power. In this paper we present an all-at-once approach where we solve for all time-steps of the discretized unsteady Stokes problem at once. The most desirable feature of this approach is that for all steps of an iterative scheme we only need approximate solutions of the discretized Stokes operator. This leads to an efficient scheme which exhibits mesh-independent behaviour.

1. Introduction

The solution of complex flow problems is one of the most interesting and demanding problems in applied mathematics and scientific computing. Over the last decades the numerical solution of problems such as Stokes flow has received a lot of attention both from applied scientists and mathematicians alike. The discretization of the Stokes equation via finite elements [13,1,10] as well as the efficient solution of the corresponding linear systems in saddle point form [13,48,43,3] are well established. In recent years, with the advances of computing power and algorithms, the solution of optimal control problems with partial differential equation (PDE) constraints such as Stokes or Navier–Stokes flow problems have become a topic of great interest [22,25,38,7,12].

In this paper, we want to address the issue of efficiently solving the linear systems that arise when the optimal control of the time-dependent Stokes problem is considered. We here want to employ the so-called all-at-once approach, which is a technique previously used in [23,24,5,44]. In detail, the discretization of the problem is constructed in the space–time domain and then solved for all time-steps at once. One of the advantages of this approach is that the PDE-constraint does not need to be satisfied until convergence of the overall system is reached. We will come back to this later.

One of the crucial ingredients to derive efficient preconditioners that show robustness with respect to the important problem parameters such as the mesh-parameter and the regularization parameter is the construction of efficient Schur-complement approximations. Recently, Pearson and Wathen [34] introduced an approximation that satisfies these criteria for the Poisson control problem. We here extend their result to a time-dependent problem subject to Stokes equation. In contrast to [34] the new approximation cannot simply be handled by a multigrid scheme but has to be embedded in a stationary iteration due to the indefiniteness of the discrete Stokes system.

* Corresponding author. Tel.: +49 391 6110 805; fax: +49 391 6110 500.
E-mail addresses: martin.stoll80@gmail.com (M. Stoll), wathen@maths.ox.ac.uk (A. Wathen).
[1] Tel.: +44 1865 615309; fax: +44 1865 273583.

0021-9991/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jcp.2012.08.039

The paper is organized as follows, we first discuss the control problem and how it can be discretized. In Section 3 we discuss the choice of the Krylov solver that should be employed. We then discuss the preconditioners for the various parts of the saddle point problem. This is followed by numerical experiments for two different objective functions in both two and three space dimensions and time.

2. Problem and discretization

In the following we consider the tracking-type functional

$$J(y,u) = \frac{1}{2}\int_0^T \int_{\Omega_1} (y-\bar{y})^2 dxdt + \frac{\beta}{2}\int_0^T \int_{\Omega_2} (u)^2 dxdt + \frac{\gamma}{2}\int_{\Omega_1} (y(T)-\bar{y}(T))^2 dx \tag{1}$$

where $\Omega_{1/2} \subseteq \Omega$ are bounded domains in $\mathbb{R}^d$ with $d = 2, 3$. Additionally, for the state y and the control u the time-dependent Stokes equation has to be satisfied

$$y_t - \nu\triangle y + \nabla p = u \quad \text{in} \quad [0,T]\times\Omega \tag{2}$$

$$-\nabla\cdot y = 0 \quad \text{in} \quad [0,T]\times\Omega \tag{3}$$

$$y(t,\cdot) = g(t) \quad \text{in} \quad \partial\Omega, t\in[0,T] \tag{4}$$

$$y(0,\cdot) = y^0 \quad \text{in} \quad \Omega, \tag{5}$$

with y the state representing the velocity and p the pressure. Here, $\bar{y}$ is the so-called desired (velocity) state. The goal of the optimization is to compute the control u in such a way that the velocity field y will be as close as possible to $\bar{y}$. One might impose additional constraints both on the control u and the state y. One of the most common constraints in practice are the so-called box constraints given by

$$u_a \leqslant u \leqslant u_b \quad \text{and} \quad y_a \leqslant y \leqslant y_b,$$

which will not be discussed further (see [45,32] for simpler PDEs).

There are two techniques used to solve the above problem. The first is the so-called *Discretize-then-Optimize* approach, where we discretize the objective function to get $J_h(y,u)$ and also discretize the PDE (Eqs. (2)–(5) written as $\mathcal{B}_h(y,u) = 0$. This allows us to form the discrete Lagrangian

$$\mathcal{L}_h(y,u) = J_h(y,u) + \lambda^T\mathcal{B}_h(y,u),$$

stationarity conditions for which would lead to a system of first order or KKT conditions. The second approach follows a *Optimize-then-Discretize* principle where we write Eqs. (2)–(5) in the form $\mathcal{B}(y,u) = 0$ and then formulate the Lagrangian of the continuous problem as

$$\mathbf{L}(y,u) = J(y,u) + \langle\mathcal{B}(y,u),\lambda\rangle$$

where $\langle\cdot,\cdot\rangle$ defines a duality product (see [26] for details). Based on the continuous Lagrangian, first order conditions are considered, which are then discretized and solved. There is no recipe as to which of these approaches has to be preferred (see the discussion in [25]). Recently, discretization schemes have been devised so that both approaches lead to the same discrete optimality system (e.g [23]).

We begin by considering the first order conditions of the above infinite dimensional problem. We obtain the forward problem described in (2)–(5) from $\mathbf{L}_\lambda = 0$, the relation

$$\beta u + \lambda = 0 \tag{6}$$

follows from $\mathbf{L}_u = 0$ and is usually referred to as the gradient equation, and we obtain also the adjoint PDE

$$-\lambda_t - \nu\triangle\lambda + \nabla\xi = y - \bar{y} \quad \text{in} \quad [0,T]\times\Omega \tag{7}$$

$$-\nabla\cdot\lambda = 0 \quad \text{in} \quad [0,T]\times\Omega \tag{8}$$

$$\lambda(t,\cdot) = 0 \quad \text{on} \quad \partial\Omega, t\in[0,T] \tag{9}$$

$$\lambda(0,\cdot) = \gamma(y(T)-\bar{y}(T)) \quad \text{in} \quad \Omega, \tag{10}$$

from $\mathbf{L}_y = 0$. For more information see [47,46,23,24].

The question is now whether we can find a discretization scheme such that the Discretize-then-Optimize and the Optimize-then-Discretize approach coincide. We start the Discretize-then-Optimize approach by using a backward Euler scheme in time to obtain for the forward Stokes problem

$$\frac{y_k - y_{k-1}}{\tau} - \nu\triangle y_k + \nabla p_k = u_k \tag{11}$$

$$-\nabla\cdot y_k = 0 \tag{12}$$

and similarly for the adjoint PDE we get

$$\frac{\lambda_k - \lambda_{k+1}}{\tau} - \nu\triangle\lambda_k + \nabla\xi_k = y_k - \bar{y}_k \tag{13}$$

$$-\nabla\cdot\lambda_k = 0. \tag{14}$$

Note that this is not an explicit method but as the adjoint PDE is going backwards in time this also represents an implicit scheme.

The matrix representation for these two expressions after a finite element space discretization is now given by

$$\frac{M\mathbf{y}_k - M\mathbf{y}_{k-1}}{\tau} + \nu K\mathbf{y}_k + B^T\mathbf{p}_k = M\mathbf{u}_k \tag{15}$$

$$B\mathbf{y}_k = 0 \tag{16}$$

$$\mathbf{y}_0 = y^0 \tag{17}$$

and

$$\frac{M\boldsymbol{\lambda}_k - M\boldsymbol{\lambda}_{k+1}}{\tau} + \nu K\boldsymbol{\lambda}_k + B^T\boldsymbol{\xi}_k = M\mathbf{y}_k - M\bar{\mathbf{y}}_k \tag{18}$$

$$B\boldsymbol{\lambda}_k = 0 \tag{19}$$

$$\boldsymbol{\lambda}_N = \gamma(\mathbf{y}_N - \bar{\mathbf{y}}_N) \tag{20}$$

with M being the mass matrix (for the sake of simplicity we assume M to be the lumped mass matrix), K the finite element stiffness matrix and $\mathbf{y}_0 = y^0$ is the projection of the initial condition onto the finite element space. We will later use M_p for the mass matrix on the pressure space but refrain from adding the index y to the mass matrix on the velocity space.

The appropriate all-at-once form for the forward PDE is now given by

$$\begin{bmatrix} \mathcal{L} & 0 & 0 & 0 & 0 \\ -\mathcal{M}_0 & \mathcal{L} & 0 & 0 & 0 \\ 0 & -\mathcal{M}_0 & \ddots & 0 & 0 \\ 0 & 0 & \ddots & \mathcal{L} & 0 \\ 0 & 0 & 0 & -\mathcal{M}_0 & \mathcal{L} \end{bmatrix} \begin{bmatrix} \mathbf{y}_0 \\ \mathbf{p}_0 \\ \mathbf{y}_1 \\ \mathbf{p}_1 \\ \vdots \\ \mathbf{y}_N \\ \mathbf{p}_N \end{bmatrix} \tag{21}$$

$$- \begin{bmatrix} M & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \\ 0 & M & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & \dots & M & 0 \\ 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & M \end{bmatrix} \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_N \end{bmatrix} = \begin{bmatrix} L\mathbf{y}_0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \tag{22}$$

$$\text{or } \mathcal{K}\mathbf{y}^+ - \mathcal{N}\mathbf{u}^+ = d, \tag{23}$$

where

$$\mathcal{L} = \begin{bmatrix} L & B^T \\ B & 0 \end{bmatrix},$$

$L = \tau^{-1}M + K$ and $\mathcal{M}_0 = \text{blkdiag}(\tau^{-1}M, 0)$. In Eq. (23) we use the notation $\mathbf{y}^+ = [\mathbf{y}_0, \mathbf{p}_0, \dots, \mathbf{y}_N, \mathbf{p}_N], \mathbf{u}^+ = [\mathbf{u}_0, \dots, \mathbf{u}_N]$ for the vectors containing the state and control variables. The matrix $\mathcal{K}$ defines the block-lower triangular matrix in (21) and $\mathcal{N}$ the rectangular matrix acting on the discretized control in (22). The right-hand side d is chosen in such a way to guarantee that the Discretize-then-Optimize and Optimize-then-Discretize approaches coincide (neglecting boundary contributions at this stage).

The scheme presented by (21) and (22) represents a discretization of the forward PDE; as already pointed out in [44] the adjoint of (22) will represent the time-evolution described by (18) but the initial condition for the adjoint PDE might make for a difference between the Discretize-then-Optimize and Optimize-then-Discretize approaches.

In more detail, the discretization of the functional $J(y,u)$ (using $\mathbf{y}_i, \mathbf{u}_i, \bar{\mathbf{y}}_i$ for $i = 0,\dots,N$ at the different time-steps) via a rectangle rule for the time and finite elements for the space leads to

$$J(\mathbf{y},\mathbf{u}) = \frac{\tau}{2}\sum_{i=1}^{N}(\mathbf{y}_i - \bar{\mathbf{y}}_i)^T M(\mathbf{y}_i - \bar{\mathbf{y}}_i) + \frac{\beta\tau}{2}\sum_{i=1}^{N}\mathbf{u}_i^T M_u \mathbf{u}_i + \frac{\gamma}{2}(\mathbf{y}_N - \bar{\mathbf{y}}_N)^T M_u(\mathbf{y}_N - \bar{\mathbf{y}}_N). \tag{24}$$

Here M is the standard or lumped mass matrix for the velocity space as before and M_u is the mass matrix for the control space, which in our case with a distributed control using the same finite element space leads to $M_u = M$. Note that N denotes the number of time-steps. In fact, we are using a slightly different approximation (note the indices of the first sums)

$$J(\mathbf{y},\mathbf{u}) = \frac{\tau}{2}\sum_{i=0}^{N}(\mathbf{y}_i - \bar{\mathbf{y}}_i)^T M(\mathbf{y}_i - \bar{\mathbf{y}}_i) + \frac{\beta\tau}{2}\sum_{i=0}^{N}\mathbf{u}_i^T M_u \mathbf{u}_i + \frac{\gamma}{2}(\mathbf{y}_N - \bar{\mathbf{y}}_N)^T M_u(\mathbf{y}_N - \bar{\mathbf{y}}_N), \tag{25}$$

which will later give that the approaches optimize-then-discretize and discretize-then-optimize coincide. Note that this changes the functional $J(y,u)$ only by constant terms involving the initial values for $\mathbf{y}_0, \bar{\mathbf{y}}_0$ and $\mathbf{u}_0$ but the location of the minimum will not be changed.

The Lagrangian of the discrete problem can now be written as

$$L_h(\mathbf{y}^+,\mathbf{u}^+,\boldsymbol{\lambda}^+) = J(\mathbf{y}^+,\mathbf{u}^+) + (\boldsymbol{\lambda}^+)^T(-\mathcal{K}\mathbf{y}^+ + \mathcal{N}\mathbf{u}^+ + d) \tag{26}$$

where we use $\mathbf{y}^+ = [\mathbf{y}_0, \mathbf{p}_0, \dots, \mathbf{y}_N, \mathbf{p}_N]$ as before and similarly $\boldsymbol{\lambda}^+ = [\boldsymbol{\lambda}_0, \xi_0, \dots, \boldsymbol{\lambda}_N, \xi_N]$. The first order or KKT conditions for $L_h(\mathbf{y}^+,\mathbf{u}^+,\boldsymbol{\lambda}^+)$ are now given by the following system

$$\begin{bmatrix} \tau\mathcal{M} & 0 & -\mathcal{K}^T \\ 0 & \beta\tau\mathcal{M}_u & \mathcal{N}^T \\ -\mathcal{K} & \mathcal{N} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y}^+ \\ \mathbf{u}^+ \\ \boldsymbol{\lambda}^+ \end{bmatrix} = \begin{bmatrix} \mathcal{M}\bar{\mathbf{y}}^+ \\ 0 \\ d \end{bmatrix}, \tag{27}$$

with $\mathcal{M} = \mathrm{blkdiag}(M, 0, \dots, M, 0)$ and $\mathcal{M}_u = \mathrm{blkdiag}(M_u, \dots, M_u)$. We will discuss appropriate solvers and possible preconditioners for system (27) in Section 3.

Our aim now is to discuss the Optimize-then-Discretize approach and how to ensure it coincides with Discretize-then-Optimize. Here we follow the results presented in [23] for the Stokes equation (see [24] for Navier–Stokes). Hinze et al. start with the infinite-dimensional KKT system. A straightforward discretization of the infinite dimensional problems will in general not result in agreement of both optimization-discretization approaches. Hinze et al.[23] however employ a technique that uses the projection of y^0 onto the finite element space to guarantee that y_0 is divergence free and has the correct boundary conditions. The initial condition is then formulated as

$$y_0 - \tau\triangle y_0 = y^0 - \tau\triangle y^0.$$

Now, writing down the Lagrangian for the semi-discretized problem the following first-order system can be obtained

$$\frac{y_k - y_{k-1}}{\tau} - \nu\triangle y_k + \nabla p_k = u_k \tag{28}$$

$$-\nabla\cdot y_k = 0 \tag{29}$$

$$y_0 - \tau\triangle y_0 = y^0 - \tau\triangle y^0 \tag{30}$$

$$\beta u_k + \lambda_k = 0 \tag{31}$$

$$\frac{\lambda_k - \lambda_{k+1}}{\tau} - \nu\triangle\lambda_k + \nabla\xi_k = y_k - \bar{y}_k \tag{32}$$

$$-\nabla\cdot\lambda_k = 0 \tag{33}$$

$$\lambda_N - \tau\triangle\lambda_N = (\tau+\gamma)(y_N - \bar{y}_N). \tag{34}$$

More details can be found in [24,23].

Using standard mixed finite elements to discretize in space we obtain the same discrete first order system for the Optimize-then-Discretize approach as for the Discretize-then-Optimize procedure. Note that with the changes we made earlier to the discretization of the cost functional $J(y,u)$, we get agreement of the initial values of the Optimize-then-Discretize approach and the Discretize-then-Optimize procedure.

In addition to the above considered problem, we will also discuss the numerical solution of a PDE-constrained optimization problem that has an added pressure term in the functional $J(y,u)$, i.e.,

$$J(y,u) = \frac{1}{2}\int_0^T\int_{\Omega_1}(y-\bar{y})^2 dxdt + \frac{1}{2}\int_0^T\int_{\Omega_1}(p-\bar{p})^2 dxdt + \frac{\beta}{2}\int_0^T\int_{\Omega}(u)^2 dxdt + \frac{\gamma}{2}\int_{\Omega_1}(y(T)-\bar{y}(T))^2 dx \tag{35}$$

subject to the unsteady Stokes equation as shown above. Here p is the pressure and $\bar{p}$ is the desired pressure. The discretization follows the above procedure and the first order conditions $L_h(\mathbf{y}^+,\mathbf{u},\boldsymbol{\lambda}^+)$ are now given by the following system

$$\begin{bmatrix} \tau\mathcal{M} & 0 & -\mathcal{K}^T \\ 0 & \beta\tau\mathcal{M}_u & \mathcal{N}^T \\ -\mathcal{K} & \mathcal{N} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y}^+ \\ \mathbf{u}^+ \\ \boldsymbol{\lambda}^+ \end{bmatrix} = \begin{bmatrix} \mathcal{M}\bar{\mathbf{y}}^+ \\ 0 \\ d \end{bmatrix}, \tag{36}$$

where the only difference to the system given in (27) is the matrix $\mathcal{M} = \text{blkdiag}(M, M_p, \ldots, M, M_p)$. Note that for reasons of convenience we use the same notation $\mathcal{M}$ whether the pressure mass matrix is present or not. We will specify when it is important to consider the two cases separately.

3. Krylov solver and preconditioning

After having derived the linear system corresponding to the solution of the optimal control problem, we now want to discuss how to solve this system efficiently. For a reasonable sized spatial discretization even in two dimensions a direct solver might run out of memory fairly quickly as the dimensionality of the overall system crucially depends on the temporal discretization. Hence, we dismiss the possibility of using a direct solver for the overall system and rather decide to employ Krylov-subspace solvers. Because of the nature of the problem, $\mathcal{A}$ being symmetric and indefinite, we will use MINRES [31] as it is often the method of choice for saddle point problems. In more detail, for a linear system $\mathcal{A}x = b$, initial guess x_0, and initial residual $r_0 = b - Ax_0$, MINRES (and also other Krylov subspace solvers) will build up Krylov subspaces

$$\text{span}\left\{r_0, \mathcal{A}r_0, \mathcal{A}^2 r_0, \ldots, \mathcal{A}^{k-1} r_0\right\}$$

by multiplying with the system matrix at each step. Here $\mathcal{A}$ denotes the saddle point system shown in (27) or (36). The approximation to the solution of the linear system will then be computed such that the 2-norm (in the unpreconditioned case) of the residual, $\|r_k\|_2$, is minimized over the current Krylov subspace. Naturally, MINRES will only be used with a preconditioner and we refer to [13] for implementation details. In order for the preconditioned system to maintain the symmetric and indefinite nature of the problem, we need the preconditioner to be symmetric and positive definite. Hence, our choice will be a symmetric block-diagonal preconditioner. Before we mention the details of the preconditioner we will discuss alternative choices for the iterative scheme. In case the upper-left block ($\text{blkdiag}(\tau\mathcal{M}, \beta\tau\mathcal{M}_u)$ is positive-definite, as is the case for the added pressure term or the forward Stokes problem, we could employ a non-standard CG method also known as the Bramble–Pasciak CG [8], which also has been successfully used for optimal control problems [45,37]. It is also possible to use the projected CG method [19] with the so-called constraint preconditioners [27], which was demonstrated to also work well for control problems [35,42,20,50]. For the use of indefinite preconditioners we would have to use non-symmetric methods such as GMRES [41], BICG [14] or QMR [15] but we will refrain from using these methods in the course of this paper. Benzi et al. [6] use Krylov methods within the preconditioner, which means that as an outer method a flexible method such as FGMRES [40] has to be employed.

We will now discuss the choice of preconditioner best suited to be used with MINRES. Our choice is a block-diagonal preconditioner of the following form

$$\mathcal{P} = \begin{bmatrix} A_0 & 0 & 0 \\ 0 & A_1 & 0 \\ 0 & 0 & \hat{S} \end{bmatrix}, \tag{37}$$

where A_0 is an approximation to $\tau\mathcal{M}$, A_1 approximates the $(2,2)$-block of the saddle point system, which we can afford to invert in case the mass matrices are lumped, and $\hat{S}$ is a Schur complement approximation. First, we want to comment on the blocks involving mass matrices. If we decide to use a consistent mass matrix, good preconditioners are available; Namely, the Chebyshev semi-iteration [17,18], which is an easy-to-use but nevertheless very efficient method for systems involving the mass matrix as illustrated in [49]. The blocks corresponding to the zero-bocks in $\tau\mathcal{M}$, can be approximated by ηI with $\eta > 0$ as was done in [5]. Appropriate choices of the parameter η will be discussed in Section 3.4.

3.1. The Schur complement approximation

The choice of the Schur complement approximation is more tricky as the $(1,1)$-block of $\mathcal{A}$ is semi-definite. Assuming for now that $\tau\mathcal{M}$ is definite, the Schur complement of the system matrix would look like the following

$$\tau^{-1}\mathcal{K}\mathcal{M}^{-1}\mathcal{K}^T + \tau^{-1}\beta^{-1}\mathcal{N}\mathcal{M}_u^{-1}\mathcal{N}^T. \tag{38}$$

In the case $\tau\mathcal{M}$ is only semi-definite, we use the Schur complement approximation

$$\tau^{-1}\mathcal{K}\bar{\mathcal{M}}^{-1}\mathcal{K}^T + \tau^{-1}\beta^{-1}\mathcal{N}\mathcal{M}_u^{-1}\mathcal{N}^T \tag{39}$$

where the zero blocks in $\tau\mathcal{M}$ are replaced by ηI with small η. We will use an approach presented in [35–38] where we drop the second term ($\tau^{-1}\beta^{-1}\mathcal{N}\mathcal{M}_u^{-1}\mathcal{N}^T$) in (38). Hence, our approximation to the Schur complement is given by

$$\hat{S}_1 = \tau^{-1}\mathcal{K}\mathcal{M}^{-1}\mathcal{K}^T.$$

We will also employ a second approach that has recently been applied to PDE-constrained optimization problems with a special emphasis on robustness with respect to regularization parameters is given by

$$\hat{S}_2 = \tau^{-1}(\mathcal{K} + \hat{\mathcal{M}})\mathcal{M}^{-1}(\mathcal{K}^T + \hat{\mathcal{M}}^T).$$

Here, $\hat{M}$ is chosen in such a way that not only the first term of the Schur complement but also the second term is matched. In more detail, this means that ideally

$$\hat{\mathcal{M}}\mathcal{M}^{-1}\hat{\mathcal{M}} = \beta^{-1}\mathcal{N}\mathcal{M}_u^{-1}\mathcal{N}^T$$

by ignoring the cross terms in $\hat{S}_2$, which are hopefully small. In order to derive $\hat{\mathcal{M}}$ we need to study $\mathcal{N}\mathcal{M}_u^{-1}\mathcal{N}^T$ and recalling the structure of the block matrices, we can see that this is simply

$$\mathcal{N}\mathcal{M}_u^{-1}\mathcal{N}^T = \begin{bmatrix} M & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & M & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Based on this observation we can now define $\hat{\mathcal{M}}$ as

$$\hat{\mathcal{M}} = \begin{bmatrix} \frac{1}{\sqrt{\beta}}M & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\sqrt{\beta}}M & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

For both Schur complement approximations $hatS_1$ and $\hat{S}_2$ it is infeasible to invert the block-triangular systems involving $\mathcal{K}$ as this would require the exact solution of all of the discretized Stokes systems. We will therefore approximate the Stokes systems further, which is described now.

3.2. Approximation of the Stokes system

For the simpler problem of the state equation being the heat equation the authors suggest in [44] that one can replace the solution with the discretized PDE operator by an appropriately chosen algebraic multigrid (AMG) preconditioner. We want to do something similar for the Stokes problem but as already pointed out in [38] the approximation of the Schur complement in the case of the Stokes problem is more involved than for the simpler heat equation. In [38] the authors show that a preconditioner for the Schur complement, namely the block-diagonal preconditioner $\mathcal{P} = \text{blkdiag}(A_0, M_p)$, is a good preconditioner for the forward Stokes equations but in the case of Stokes control where a fourth-order operator (inverting also the adjoint) has to be approximated, the contraction of the block-diagonal preconditioner is not sufficient for the Schur complement approximation of the control problem. Hence, Rees and Wathen suggest the use of an inexact Uzawa method using a block-triangular preconditioner

$$\mathcal{P} = \begin{bmatrix} A_0 & 0 \\ B & -M_p \end{bmatrix},$$

where A_0 is an approximation to the discretized Laplacian, in general a multigrid operator, and M_p is the mass matrix on the pressure space (see [13]). A fixed number of Uzawa steps to approximate the discrete Stokes operator represents a linear operator and provides a good enough contraction rate such that the approximation to the Schur complement will be sufficient to guarantee convergence of the overall outer MINRES iteration. For the steady case this was shown in [38]. Algorithm 1 shows a version of the inexact Uzawa method. Note that in the case of enclosed flow the Stokes-system matrix will be singular due to the hydrostatic pressure [13] but a consistent right-hand-side still enables the use of iterative solvers. As we need to apply a forward and a backward solve with the inexact Uzawa method a scaling[2] to make the right-hand-side sufficiently close to a consistent right-hand-side always worked very well in our numerical experiments. In the case the Stokes system is invertible these issues do not arise.

[2] In Matlab notation: Scaling b such that Pb is close to a consistent right hand side, with $P = \text{speye}(n) - \frac{\alpha}{n}\text{ones}(n,n)$ with α close to one, e.g. 0.9.

Algorithm 1. Inexact Uzawa method

1: Select x_0.
2: **for** $k = 1, 2, \ldots$ **do**
3: $x_{k+1} = x_k + \mathcal{P}^{-1}(b - \mathcal{L}x_k)$
4: **end for**

There is only a change in the preconditioner when moving from the steady system to the transient one. We note that the Schur complement approximation

$$\hat{\mathcal{K}}\mathcal{M}^{-1}\hat{\mathcal{K}}^T$$

involves a forward and backward substitution where we have to approximate the inverse of the matrix

$$\mathcal{L} = \begin{bmatrix} \tau^{-1}M + K & B^T \\ B & 0 \end{bmatrix} \tag{40}$$

for the evaluation of $\hat{\mathcal{K}}$ and $\hat{\mathcal{K}}^T$ at each time-step. We propose to use the inexact Uzawa algorithm (see Algorithm 1) for the matrix (40) with a block-triangular preconditioner defined as

$$\mathcal{P} = \begin{bmatrix} \hat{A} & 0 \\ B & -\overline{S} \end{bmatrix}.$$

We can now simply use an algebraic or geometric multigrid for the preconditioner $\hat{A}$ approximating $\tau^{-1}M + K$ but the choice of $\overline{S}$ is not so straightforward. In the case of steady Stokes problem the pressure mass matrix will allow for a suitable approximation to the Schur complement. In our case, we have a different $(1,1)$-block to the steady case and we derive a suitable preconditioner using a technique for the steady Navier–Stokes equation. We follow [Chapter 8 [13]] by looking at the least squares commutator (see [13]) defined by

$$\mathcal{E} = (\mathbb{L})\nabla - \nabla(\mathbb{L}_p)$$

where $\mathbb{L} = \tau^{-1}I + \triangle$ and $\mathbb{L}_p = (\tau^{-1}I + \triangle)$ is defined on the pressure space. These operators are only used for the purpose of deriving matrix preconditioners and no function spaces or boundary conditions are defined here. We expect the least squares commutator to be small as was previously done for the derivation of Navier–Stokes preconditioners [13]. Using the finite element method we obtain the discretization of the differential operators (see Chapter 8.2 [13]) and put this into the discretized version of the above to get

$$\mathcal{E}_h = (M^{-1}L)M^{-1}B^T - M^{-1}B^T(M_p^{-1}L_p)$$

where $L = \tau^{-1}M + K$. We now pre-multiply the last equation by $BL^{-1}M$ and post-multiply by $L_p^{-1}M_p$ to get

$$BM^{-1}B^TL_p^{-1}M_p - BL^{-1}B^T \approx 0, \tag{41}$$

under the assumption that the least squares commutator is small. The expression (41) gives

$$BM^{-1}B^TL_p^{-1}M_p \approx BL^{-1}B^T, \tag{42}$$

which allows us to use the Schur-complement approximation $BM^{-1}B^TL_p^{-1}M_p$. We do not want to use the matrix $BM^{-1}B^T$, which is invertible. For our implementation, we would have to form this matrix, which would be infeasible in the case of a consistent mass matrix. Hence, we rather use the fact that $BM^{-1}B^T$ is spectrally equivalent to the Laplacian formed on the pressure space K_p to give

$$\overline{S} = K_pL_p^{-1}M_p. \tag{43}$$

Note that as we are only interested in the application of $\overline{S}^{-1}$ we can further obtain

$$\overline{S}^{-1} = M_p^{-1}L_pK_p^{-1} = M_p^{-1}(\tau^{-1}M_p + K_p)K_p^{-1} = \tau^{-1}K_p^{-1} + M_p^{-1}. \tag{44}$$

With the approximation (44) we are now able to provide efficient preconditioners for the solution of the time-dependent Stokes problem within the Uzawa method. The preconditioner $\overline{S}^{-1}$ was first derived in [11] by Cahouet and Chabard and is hence often referred to as the Cahouet–Chabard preconditioner. It was extensively used, analyzed and extended to for example the Navier–Stokes case (more information can be found in [4,28,9,6,30]). $\hat{A}$ will in our case be an algebraic multigrid method applied to $\tau^{-1}M + K$ and for $\overline{S}^{-1}$ we need the approximation to K_p^{-1}, which can be done using algebraic multigrid as well. Additionally, we need to approximate M_p^{-1}, which can be efficiently approximated using the Chebyshev semi-iteration [17,18,49] (see Algorithm 2).

Algorithm 2. Chebyshev semi-iterative method for a number of l steps

1: Set $D = diag(M_p)$
2: Set relaxation parameter ω
3: Compute $g = \omega D^{-1}\hat{b}$, (with $\hat{b}$ the input right-hand-side)
4: Set $S = (I - \omega D^{-1}M_p)$ (this can be used implicitly)
5: Set $z_{k-1} = 0$ and $z_k = Sz_{k-1} + g$
6: $c_{k-1} = 2$ and $c_k = \omega$
7: **for** $k = 2, \ldots, l$ **do**
8: $\quad c_{k+1} = \omega c_k - \frac{1}{4}c_{k-1}$
9: $\quad \vartheta_{k+1} = \omega \frac{c_k}{c_{k+1}}$
10: $\quad z_{k+1} = \vartheta_{k+1}(Sz_k + g - z_{k-1}) + z_{k-1}$
11: **end for**

The identical analysis can be performed for the case when the Schur complement is approximated by $\overline{S} = \tau^{-1}(\mathcal{K} + \hat{\mathcal{M}})\mathcal{M}^{-1}(\mathcal{K}^T + \hat{\mathcal{M}}^T)$ as only the Cahouet–Chabard preconditioner now has to be derived for $\mathbb{L} = \tau^{-1}I + \frac{1}{\sqrt{\beta}}I + \triangle$, which in turn leads to

$$\overline{S}^{-1} = \left(\tau^{-1} + \frac{1}{\sqrt{\beta}}\right)K_p^{-1} + M_p^{-1}. \tag{45}$$

3.3. Eigenvalue analysis

In this section, we study the eigenvalues of the preconditioned matrix. We closely follow an earlier analysis presented in [44].

We analyze the eigenvalues of the preconditioned matrix for a somewhat idealized case. We assume that the preconditioner is given by

$$\mathcal{P} = \begin{bmatrix} \bar{\mathcal{M}} & 0 & 0 \\ 0 & \beta\tau\mathcal{M}_u & 0 \\ 0 & 0 & \hat{S}_1 \end{bmatrix}$$

with $\hat{S}_1 = \mathcal{K}\bar{\mathcal{M}}^{-1}\mathcal{K}^T$. A congruence transformation $\mathcal{P}^{-1/2}\mathcal{A}\mathcal{P}^{-1/2}$, now reveals the following matrix

$$\mathcal{P}^{-1/2}\mathcal{A}\mathcal{P}^{-1/2} = \begin{bmatrix} D & 0 & B_1^T \\ 0 & I & B_2^T \\ B_1 & B_2 & 0 \end{bmatrix} \tag{46}$$

where $B_1 = \hat{S}_1^{-1/2}\mathcal{K}\bar{\mathcal{M}}^{-1/2}$, and $B_2 = \tau^{-1/2}\beta^{-1/2}\hat{S}_1^{-1/2}\mathcal{N}\mathcal{M}_u^{-1/2}$. We will switch to the notation $A = \text{blkdiag}(D, I)$ and $B = [B_1\ B_2]$ as for the classical saddle point problem. It is a well-known result [39] that for such a saddle point problem with symmetric and positive-definite $(1,1)$-block, the eigenvalues of $\mathcal{P}^{-1/2}\mathcal{A}\mathcal{P}^{-1/2}$ lie in the intervals

$$\mathcal{I}^- = \left[\frac{1}{2}\left(\lambda_{\min}^{(A)} - \sqrt{\left(\lambda_{\min}^{(A)}\right)^2 + \sigma_{\max}^2}\right), \frac{1}{2}\left(\lambda_{\max}^{(A)} - \sqrt{\left(\lambda_{\max}^{(A)}\right)^2 + \sigma_{\max}^2}\right)\right]$$

and

$$\mathcal{I}^+ = \left[\lambda_{\min}^{A}, \frac{1}{2}\left(\lambda_{\max}^{(A)} + \sqrt{\left(\lambda_{\max}^{(A)}\right)^2 + \sigma_{\max}^2}\right)\right],$$

where $\sigma_{\max}$ denotes the maximal singular value of B. This is true for both problems presented here. In the case when the pressure is included in the objective function (35), the resulting saddle point system (36) has a positive definite $(1,1)$-block, which will lead to $\lambda_{\min}^A > 0$. We will now discuss the bounds for $\mathcal{I}^+$ and $\mathcal{I}^-$ in more detail. In both cases, we need bounds for eigenvalues of A. The structure of A reveals that we have an identity block and the matrix $D = \text{blkdiag}(I, 0, I, 0, \ldots, I, 0)$ for the objective function (1) and $D = \text{blkdiag}(I, I_p, I, I_p, \ldots, I, I_p)$ for (35). It is therefore easy to read off the eigenvalues of D. The estimation of the singular values of B is a bit more involved and we use the fact that the eigenvalues of BB^T are the square of the singular values of B. The structure of B now gives

$$BB^T = B_1B_1^T + B_2B_2^T = \hat{S}_1^{-1/2}\mathcal{K}\bar{\mathcal{M}}^{-1}\mathcal{K}^T\hat{S}_1^{-1/2} + \tau^{-1}\beta^{-1}\hat{S}_1^{-1/2}\mathcal{N}\mathcal{M}_u^{-1}\mathcal{N}^T\hat{S}_1^{-1/2}$$

and note that the last matrix is similar to

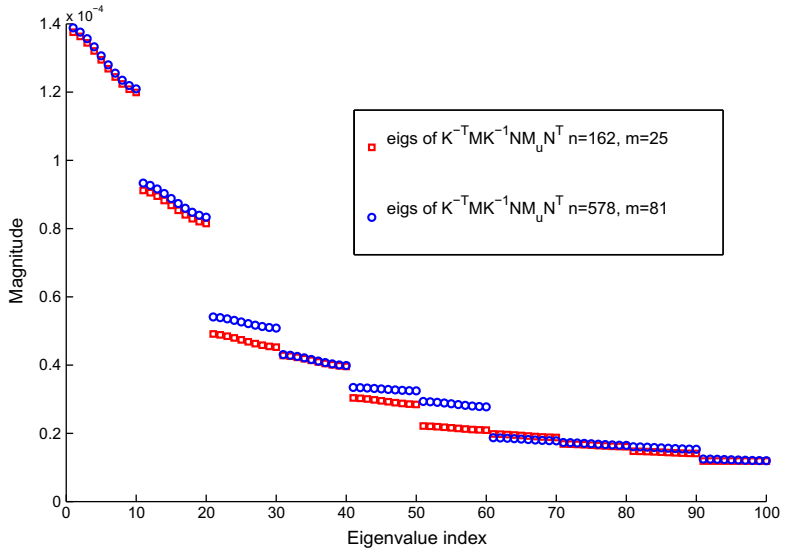


Fig. 1. Largest 100 eigenvalues of $\mathcal{K}^{-T}\bar{\mathcal{M}}\mathcal{K}^{-1}\mathcal{N}\mathcal{M}_u^{-1}\mathcal{N}^T$ for two small problems.

$$\hat{S}_1^{-1}\left(\mathcal{K}\bar{\mathcal{M}}^{-1}\mathcal{K}^T+\tau^{-1}\beta^{-1}\mathcal{N}\mathcal{M}_u^{-1}\mathcal{N}^T\right).$$

This indicates that if $\hat{S}_1^{-1}$ is chosen to be $\mathcal{K}^{-T}\bar{\mathcal{M}}\mathcal{K}^{-1}$ the above takes the following form

$$I+\tau^{-1}\beta^{-1}\mathcal{K}^{-T}\bar{\mathcal{M}}\mathcal{K}^{-1}\mathcal{N}\mathcal{M}_u^{-1}\mathcal{N}^T. \tag{47}$$

Similar equations to (47) have been analyzed before for stationary problems [35,44]. For the transient case in (47), we have to show that for a more refined mesh, smaller mesh parameter h, the eigenvalues of $\mathcal{K}^{-T}\bar{\mathcal{M}}\mathcal{K}^{-1}\mathcal{N}\mathcal{M}_u^{-1}\mathcal{N}^T$ do not change. We are at this stage not able to prove the mesh-independence of the term $\tau^{-1}\beta^{-1}\mathcal{K}^{-T}\bar{\mathcal{M}}\mathcal{K}^{-1}\mathcal{N}\mathcal{M}_u^{-1}\mathcal{N}^T$. In general the term $\bar{\mathcal{M}}$ will include a multiplication by τ which removes the dependency of the eigenvalue bounds on τ since no other matrix involves τ. In Fig. 1 we show the largest 100 eigenvalues computed by the MATLAB `eigs`}command of the matrix $\mathcal{K}^{-T}\bar{\mathcal{M}}\mathcal{K}^{-1}\mathcal{N}\mathcal{M}_u^{-1}\mathcal{N}^T$ where $\bar{\mathcal{M}}=\text{blkdiag}(M,\eta I,\ldots,M,\eta I)$ with $\eta=10^{-6}$. The Stokes problem is for simplicity chosen with a Neumann boundary at the bottom and Dirichlet on the remaining sides of the domain to have an invertible Stokes matrix.[3] In Fig. 1 we show the 100 largest eigenvalues for two relatively small meshes with the DoF for one Stokes system given by $n=578,\ m=81$ and for the second Stokes system $n=162,\ m=25$, where n is the number of discrete velocity variables and m the number of discrete pressure variables. Note that these are the degrees of freedom for one instance of the unsteady problem. We chose a fixed number of time-steps $N=10$ and see that the eigenvalues for these problems do not depend on h; we expect this behaviour to continue for smaller h as in our numerical experiments (see Section 4) we do not observe mesh-dependent behaviour.

We further want to analyze the behaviour of the second Schur complement approximation provided by

$$\hat{S}_2=\tau^{-1}(\mathcal{K}+\hat{\mathcal{M}})\mathcal{M}^{-1}(\mathcal{K}^T+\hat{\mathcal{M}}^T).$$

In a similar way to the results presented in [33,34], we have to consider the eigenvalues of the matrix $\hat{S}_2^{-1}S$, where S is the Schur complement. For this we consider the Rayleigh quotient with a normalized vector v of appropriate dimension, i.e.,

$$\frac{v^TSv}{v^T\hat{S}_2v}=\frac{v^T(\tau^{-1}\mathcal{K}\mathcal{M}^{-1}\mathcal{K}^T+\tau^{-1}\beta^{-1}\mathcal{N}\mathcal{M}_u^{-1}\mathcal{N}^T)v}{v^T(\tau^{-1}(\mathcal{K}+\hat{\mathcal{M}})\mathcal{M}^{-1}(\mathcal{K}^T+\hat{\mathcal{M}}^T))v},$$

which we can also write as

$$\frac{1}{\frac{v^T(\tau^{-1}\mathcal{K}\mathcal{M}^{-1}\mathcal{K}^T+\tau^{-1}\beta^{-1}\mathcal{N}\mathcal{M}_u^{-1}\mathcal{N}^T)v}{v^T(\tau^{-1}\mathcal{K}\mathcal{M}^{-1}\mathcal{K}^T+\tau^{-1}\beta^{-1}\mathcal{N}\mathcal{M}_u^{-1}\mathcal{N}^T)v}+\frac{\tau^{-1}v^T(\hat{\mathcal{M}}\mathcal{M}^{-1}\mathcal{K}^T+\mathcal{K}\mathcal{M}^{-1}\hat{\mathcal{M}})v}{v^T(\tau^{-1}\mathcal{K}\mathcal{M}^{-1}\mathcal{K}^T+\tau^{-1}\beta^{-1}\mathcal{N}\mathcal{M}_u^{-1}\mathcal{N}^T)v}}.$$

Assuming that $\hat{\mathcal{M}}\mathcal{M}^{-1}\hat{\mathcal{M}}=\beta^{-1}\mathcal{N}\mathcal{M}_u^{-1}\mathcal{N}^T$, this can also be written as

$$\frac{1}{1+\frac{\tau^{-1}v^T(\hat{\mathcal{M}}\mathcal{M}^{-1}\mathcal{K}^T+\mathcal{K}\mathcal{M}^{-1}\hat{\mathcal{M}})v}{v^T(\tau^{-1}\mathcal{K}\mathcal{M}^{-1}\mathcal{K}^T+\tau^{-1}\beta^{-1}\mathcal{N}\mathcal{M}_u^{-1}\mathcal{N}^T)v}}:=R.$$

[3] Note that in the enclosed flow case we have a one-dimensional kernel and this cannot be used for the illustration in Fig. 1.

We see that the value R can be bounded below by $\frac{1}{2}$ as

$$(a-b)^T(a-b) \geqslant 0 \Longleftrightarrow \frac{a^Tb+b^Ta}{a^Ta+b^Tb} \leqslant 1$$

for any a, b and $a^Ta + b^Tb > 0$ and hence for $a = \mathcal{M}^{-1/2}\mathcal{K}^T v$ and $b = \mathcal{M}_u^{-1/2}\mathcal{N}^T v$ we get that $R \geqslant \frac{1}{2}$. For previous results [34] the boundedness of R from above used the positive definiteness of the matrix $(\hat{\mathcal{M}}\mathcal{M}^{-1}\mathcal{K}^T + \mathcal{K}\mathcal{M}^{-1}\hat{\mathcal{M}})$, which for the problem presented in [34] is trivially given. Here the structure of the matrices does not give that $(\hat{\mathcal{M}}\mathcal{M}^{-1}\mathcal{K}^T + \mathcal{K}\mathcal{M}^{-1}\hat{\mathcal{M}})$ is positive definite, i.e.,

$$\hat{\mathcal{M}}\mathcal{M}^{-1} = \frac{1}{\sqrt{\beta}}\begin{bmatrix} I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

results in

$$(\hat{\mathcal{M}}\mathcal{M}^{-1}\mathcal{K}^T + \mathcal{K}\mathcal{M}^{-1}\hat{\mathcal{M}}) = \frac{1}{\sqrt{\beta}}\begin{bmatrix} 2L & B^T & -M & 0 & 0 & 0 & 0 & 0 \\ B & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -M & 0 & 0 & 0 & \ddots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \ddots & 0 & \ddots & 0 & -M & 0 \\ 0 & 0 & 0 & 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 & -M & & 2L & B^T \\ 0 & 0 & 0 & 0 & 0 & 0 & B & 0 \end{bmatrix}. \tag{48}$$

At this point we are unable to prove the robustness for the upper bounds but we will illustrate the behaviour of the eigenvalues based on numerical experiments. Fig. 2 is showing the eigenvalues for the case with pressure term of two small problems ($n = 578, m = 81$ and $n = 162, m = 25$) and for two different values of the regularization parameter. It can be seen that the eigenvalues of $\hat{S}_2^{-1}S$ depend on the regularization parameter β but seem to be independent with respect to the mesh-parameter h as for both matrix sizes the eigenvalues are very similar for the same regularization parameter. The situation is different when we are dealing with a semi-definite $(1,1)$-block (no pressure term) as the matrix $\bar{\mathcal{M}}$ now includes the parameter η. Again, the lower bound $\frac{1}{2}$ can be obtained in the same way as in the above and the upper bound can be motivated by the fact that now the term

$$\frac{\tau^{-1}v^T(\hat{\mathcal{M}}\bar{\mathcal{M}}^{-1}\mathcal{K}^T + \mathcal{K}\bar{\mathcal{M}}^{-1}\hat{\mathcal{M}})v}{v^T(\tau^{-1}\mathcal{K}\bar{\mathcal{M}}^{-1}\mathcal{K}^T + \tau^{-1}\beta^{-1}\mathcal{N}\mathcal{M}_u^{-1}\mathcal{N}^T)v} \tag{49}$$

is η dependent. Due to the structure of $\hat{\mathcal{M}}\bar{\mathcal{M}}^{-1} = \frac{1}{\sqrt{\beta}}\text{blkdiag}(I, 0, \ldots, I, 0)$ the parameter η is not influencing the term in the numerator of (49) but only influences the value of its denominator, which helps to explain the eigenvalue distribution shown in Fig. 3, i.e., η balances the effect of the negative eigenvalues of $(\hat{\mathcal{M}}\bar{\mathcal{M}}^{-1}\mathcal{K}^T + \mathcal{K}\bar{\mathcal{M}}^{-1}\hat{\mathcal{M}})$. In Fig. 3 we again show the eigenvalues for the two different problem sizes used earlier and two values of the regularization parameter.

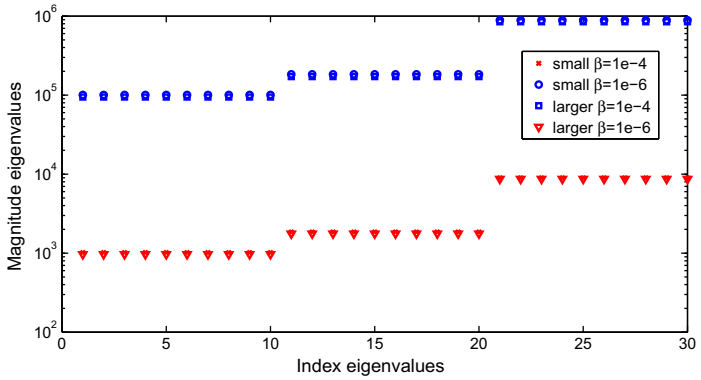


Fig. 2. Largest 30 eigenvalues of $\hat{S}_2^{-1}S$ for two small problems with pressure term.

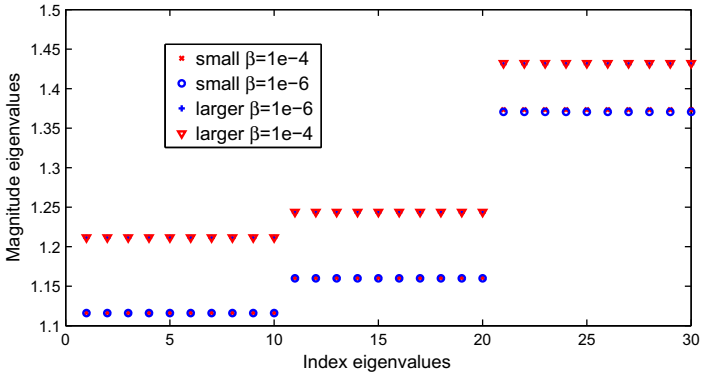


Fig. 3. Largest 30 eigenvalues of $\hat{S}_2^{-1}S$ for two small problems without pressure term.

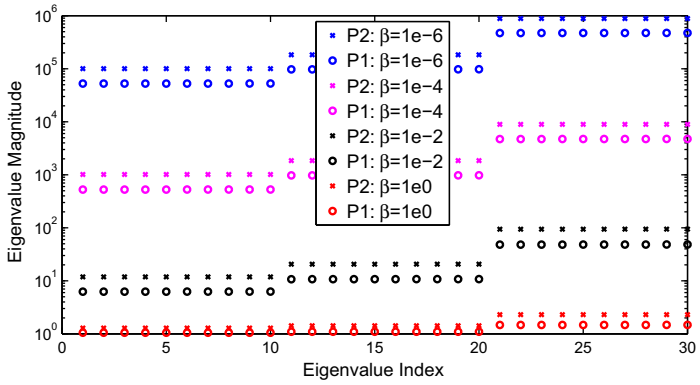


Fig. 4. Largest 30 eigenvalues of $\hat{S}^{-1}S$ for a small problem with pressure term and varying β. $P1$ and $P2$ refer to the use of $\hat{S}_1$ and S_2, respectively.

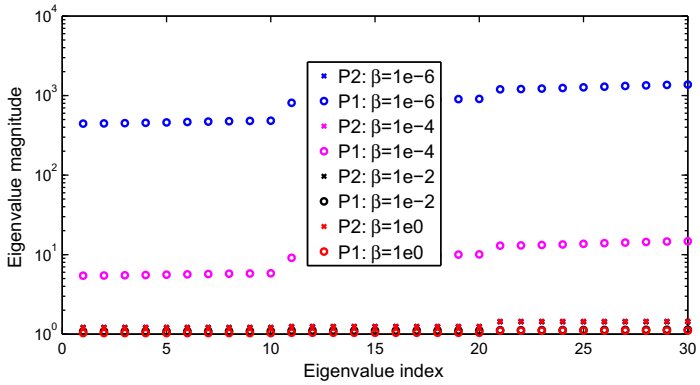


Fig. 5. Largest 30 eigenvalues of $\hat{S}^{-1}S$ for a small problem without pressure term and varying β. $P1$ and $P2$ refer to the use of $\hat{S}_1$ and S_2, respectively. Only the eigenvalues of $\hat{S}_1^{-1}S$ for small values of β ($\beta = 10^{-4}$ and $\beta = 10^{-4}$) are significantly larger than one.

In Fig. 4 and Fig. 5 we compare the largest 30 eigenvalues of $\hat{S}^{-1}S$ for both choices of $\hat{S}$ for the problem with and without pressure term. In Fig. 4 we use the notation $P1$ for the use of $\hat{S}_1$ and P_2 whenever $\hat{S}_2$ is used for a small problem with a variety of regularization parameters. It can be seen that the eigenvalues in both cases strongly depend on the regularization parameter β. This is in contrast to Fig. 5 where we again compare $\hat{S}_1$ and $\hat{S}_2$ for the same small problem without pressure term and varying regularization parameter.

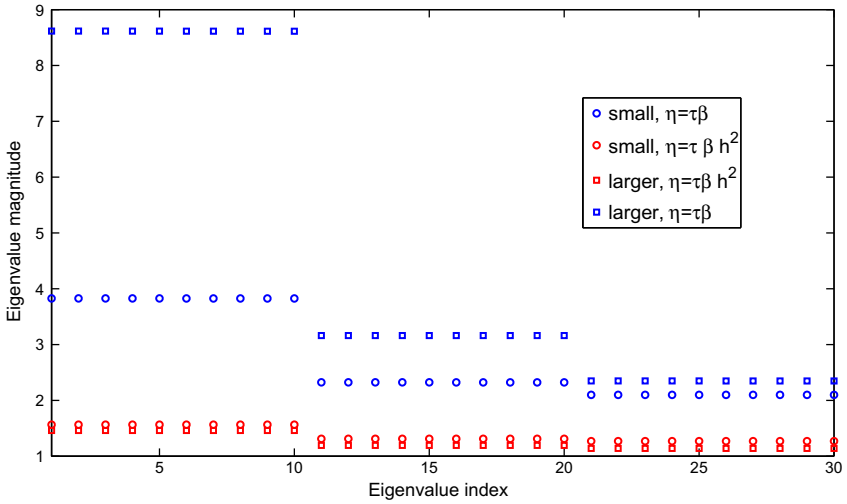


Fig. 6. Largest 30 eigenvalues of $\hat{S}_2^{-1}S$ for two small problems without pressure term and varying η.

3.4. Choice of η

We now want to motivate a heuristic for the choice of the parameter η. Note that the results below are for a two-dimensional problem. We follow a strategy presented in [5] that was also used in [33] to balance the two terms in the Schur-complement

$$S = \tau^{-1}\mathcal{K}\mathcal{M}^{-1}\mathcal{K}^T + \tau^{-1}\beta^{-1}\mathcal{N}\mathcal{M}_u^{-1}\mathcal{N}^T.$$

We have previously observed [44,33] that a heuristic choosing $\eta \approx \tau\beta h^d$ where h is the mesh-parameter and $d = 2, 3$ is the dimension of the domain often yields good results. In Fig. 6 we show the dependence of the 30 largest eigenvalues of the two already considered problems with respect to varying η. Namely, we consider the choice $\eta = \tau\beta$, which can be motivated only using the diagonal blocks of $\mathcal{K}$ in a Schur complement approximation. This in turn leads to a balancing of the terms $LM^{-1}L + \eta^{-1}B^TB$ against $\tau^{-1}\beta^{-1}M$, where we can use that the eigenvalues of B^TB scale like h^2 in 2D. We also show results for the value $\eta = \tau\beta h^2$ that was previously used in [33]. It can be seen that this choice outperforms the first one as no dependence on the mesh-parameter could be observed. This is also used in the numerical experiments presented later, where we always obtained satisfying results using this heuristic. Note that so far we only considered the two-dimensional case but this heuristic can easily be extended to three space dimensions to get $\eta = \tau\beta h^3$.

4. Numerical experiments

The numerical tests are all performed using deal.II [2]. We use a **Q2**/**Q1** discretization of the Stokes problem. The inverse of the pressure Laplacian is approximated by 6 steps of an AMG V-cycle and 10 steps of a Chebyshev smoother as part of the ML Trilinos package [16]. The inverse of the Laplacian plus mass matrix block is approximated by 10 steps of a Chebyshev smoother and 2 steps of an AMG V-cycle. In general we use a relative tolerance of 10^{-4} for the pseudo-residual and mention explicitly if any other tolerance is used. Within the smoothed aggregation AMG we use a Jacobi solver for the coarsest level.[4] With the setup 10 steps of a Uzawa method seem to produce good results for all our test cases. All experiments are performed on a 64 bit Centos Linux machine with Intel (R) Xeon (R) CPU X5650 @ 2.67 GHz CPUs and 48 GB of RAM. No parallelism was exploited in our implementation. The example we look at in this section is taken from the paper by Hinze et al. [23]. The spatial domain is defined as $\Omega = [0,1]^d$ and the time domain is given as $[0,1]$. As we have not used special multigrid methods devised for parameter-dependent problems it has been observed in [29] that general purpose preconditioners might loose the independence with respect to τ. This behaviour could not be observed if τ scaled with the mesh-parameter and hence we are often choosing $\tau \approx h$. We will also present results for a fixed $\tau = 0.05$.

The target flow is the solution for the unsteady Stokes equation with Dirichlet boundary conditions, i.e. $y = (1,0)$ when the second spatial component $x_2 = 1$ and $y = (0,0)$ on the remaining boundary for the two-dimensional case. In the three dimensional case we set $y = (1,0,0)$ when $x_2 = 1$ and $y = (0,0,0)$ on the remaining boundary. The viscosity was chosen to be $\nu = 1$. Fig. 7 shows the desired state at $t = 0.63$. For the control problem we now consider the following time-dependent boundary conditions. For the top-boundary where $x_2 = 1$ we get $y = (1 + \frac{1}{2}\cos(4\pi t - \pi), 0)$ and zero elsewhere in two

[4] The standard direct solver worked well for the Neumann problem on 32 bit machines but failed completely on 64bit machines.

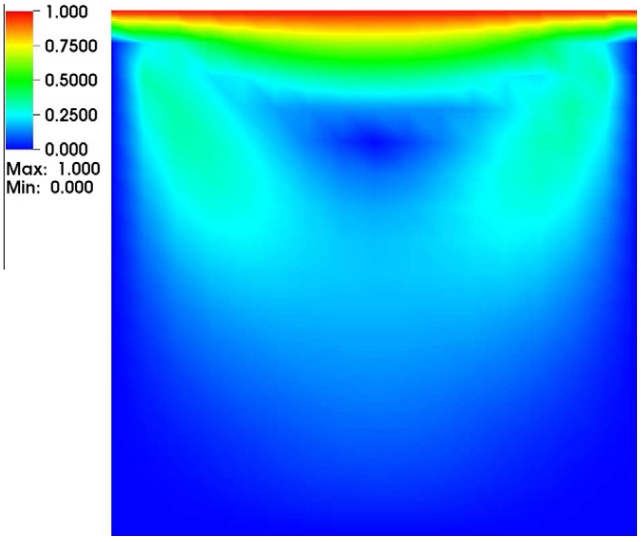


Fig. 7. Desired state at $t = 0.63$.

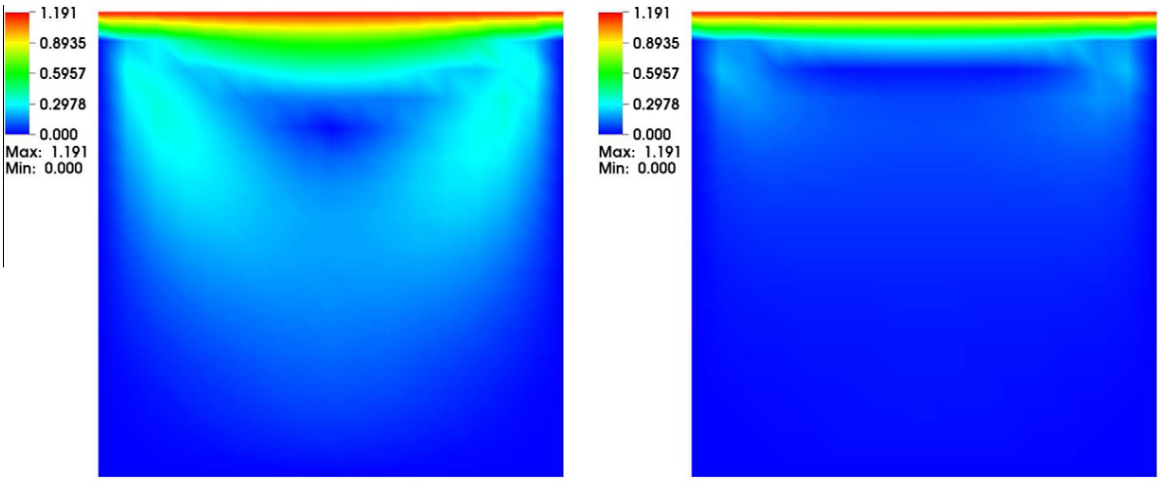


Fig. 8. Uncontrolled vs. controlled state.

space dimensions. For this example the viscosity is set to $1/100$. We also take $\gamma = 0$. Fig. 8 shows both the computed controlled state and the uncontrolled state for the above system at $t = 0.63$. For the choice of the scaling parameter η dealing with the zero-blocks in the $(1,1)$ block of the saddle point system we follow the heuristic proposed above. To illustrate the performance of our preconditioner it is imperative to consider three-dimensional results and we choose the boundary condition for $x_2 = 1$ to be $y = (t + sin(0.1x_1), t + cos(0.5x_2), 0), y = (0,0,0)$ on the rest of the domain, and $\nu = \frac{1}{100}$. This is a somewhat arbitrary choice but nevertheless exhibits all the complications expected in a realistic problem. For simplicity the initial condition y_0 is chosen to be zero within the domain and satisfying the boundary conditions on $\partial\Omega$ for the corresponding problem.

4.1. Without pressure term

We begin our numerical experiments by computing the approximate solution to the above problems on a variety of meshes. Fig. 8 shows the controlled and the uncontrolled state at the time $t = 0.63$. The control for this time is shown in Fig. 9. We denote by *DoF* the degrees of freedom used for the discretization of the PDE at one time-step, similar to a steady problem, then we show the number of time-steps *N*. This means that for the finest mesh we are implicitly solving a linear

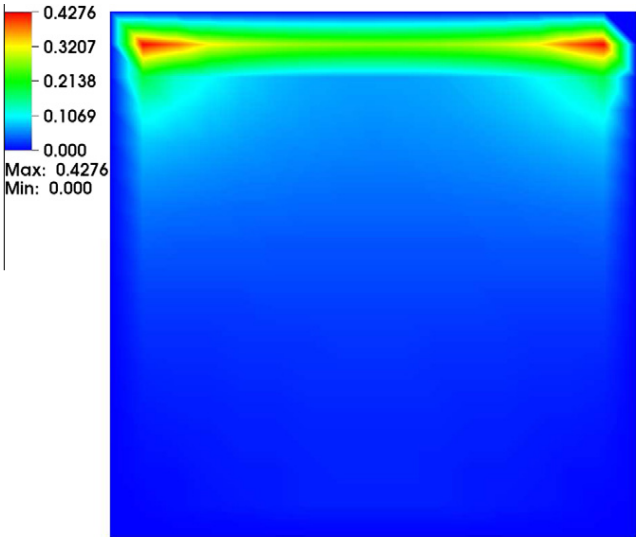


Fig. 9. Control at $t = 0.63$.

system of dimension $3 * 37507 * 129 \approx 14$million unknowns. Timings are given in seconds. Table 1 shows the results for a relative tolerance of 10^{-4} when $\tau \approx h$. In Table 2 we show the results for the same setup just the number of time-steps is now fixed. As can be seen from the results in Table 1 the iteration numbers with both preconditioners do not increase with refinement in space and time. We see that the approximation $\hat{S}_1$ produces mesh-independent iterates but the results are not as satisfactory as the ones obtained for the preconditioner $\hat{S}_2$, which comes essentially at the same cost. Thus, we will refrain from showing the results of the $\hat{S}_1$ preconditioner for the remainder of this section. We see in Table 2 that the iteration counts with respect to β are almost constant. The slight increase in the iteration numbers for smaller β and h is in our opinion due to the performance of the algebraic multigrid method, which for parameter dependent problems does not always perform equally for decreasing parameter values. We believe that if the parameter decrease is taken into account by algebraic or geometric multigrid, then the iteration numbers will stay constant. In fact, we can see in Table 2, where the time-step τ is constant, that the iteration numbers are constant with respect to β and h.

Table 3 shows the results for a boundary condition that is more oscillatory than the one previously used, i.e., $y = (1 + \frac{1}{2}\cos(50.5\pi t - \pi), 0)$ whenever $x_2 = 1$. It can be seen that iteration numbers do not change compared to the iteration numbers presented in Table 1. It might be necessary to refine further in time and space to capture the essence of the more oscillatory problem but as our preconditioners are robust with respect to temporal and spatial refinement we would not expect any difficulties.

The results shown in Table 4 are computed for a tolerance of 10^{-6} and $\tau \approx h$ with the setup in three dimensions. Again, we see a very moderate number of MINRES iterations for this case. We show results for the three-dimensional problem in Fig. 10

Table 1
Number of MINRES steps with CPU-time $\tau \approx h$ for different values of β. Results are shown for the preconditioner $\hat{S}_1$ and the preconditioner $\hat{S}_2$.

DoF	N	MINRES (Time) $\hat{S}_1$ ($\beta = 10^{-2}$)	MINRES (Time) $\hat{S}_2$ ($\beta = 10^{-4}$)	MINRES (Time) $\hat{S}_2$ ($\beta = 10^{-6}$)
2467	33	51(1010)	15(323)	11(247)
9539	65	49(7447)	15(2454)	17(2777)
37507	129	49(48668)	15(16008)	24(24521)

Table 2
Number of MINRES steps with CPU-time for different values of β and $\tau = 0.05$. Results are shown for the preconditioner $\hat{S}_2$ with a stopping tolerance of 10^{-4}.

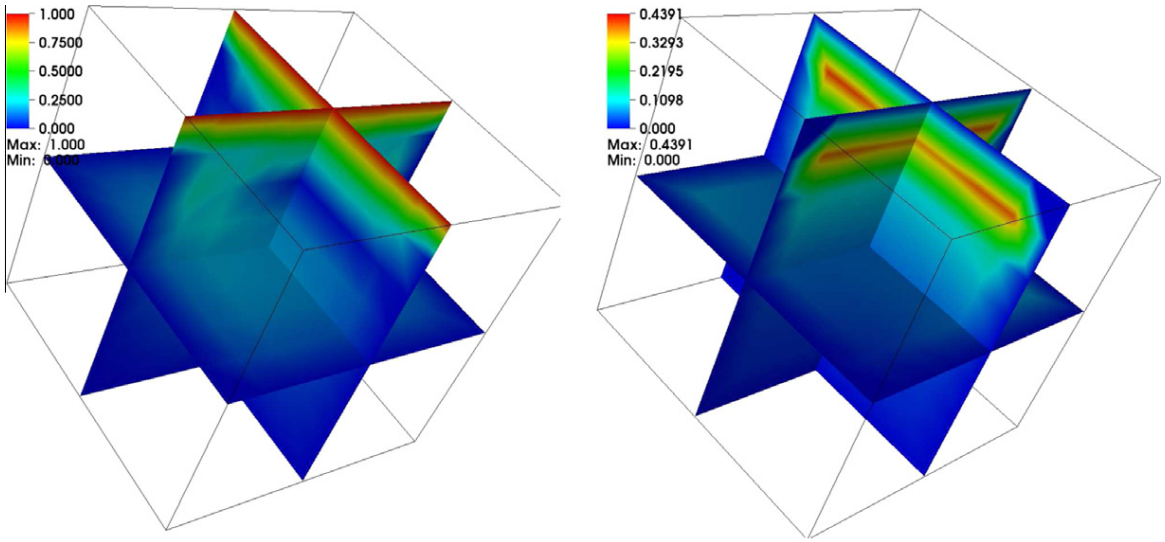
DoF	N	MINRES (Time) $\hat{S}_2$ ($\beta = 10^{-4}$)	MINRES (Time) $\hat{S}_2$ ($\beta = 10^{-6}$)
2467	21	15(336)	12(277)
9539	21	14(1201)	14(1202)
37507	21	14(4277)	16(4832)

Table 3
Number of MINRES steps with CPU-time for different values of β and $\tau \approx h$. The boundary condition is changed to $y = (1 + \frac{1}{2}\cos(50.5\pi t - \pi), 0)$ for the top boundary. Results are shown for the preconditioner $\hat{S}_2$ with a stopping tolerance of 10^{-4}.

DoF	N	MINRES (Time) $\hat{S}_2$ ($\beta = 10^{-4}$)	MINRES (Time) $\hat{S}_2$ ($\beta = 10^{-6}$)
2467	33	14(322)	8(201)
9539	65	14(2367)	13(2227)

Table 4
Number of MINRES steps and CPU-time with $\tau \approx h$ using different values of β for a three-dimensional problem. Results are shown for the preconditioner $\hat{S}_2$. The tolerance is set to 10^{-6}.

DoF	N	MINRES (Time) $\hat{S}_2$ ($\beta = 10^{-4}$)	MINRES (Time) $\hat{S}_2$ ($\beta = 10^{-6}$)
2312	9	23(364)	17(277)
15468	17	20(4711)	12(2996)



(a) Desired state at $t = 0.5$ (b) Control at $t = 0.5$

Fig. 10. Desired state and control.

and Fig. 11 with Fig. 10(a) showing the computed state, Fig. 10(b) showing the computed control, Fig. 11(b) showing the uncontrolled state, and Fig. 11(a) showing the desired state.

Finally, in Table 5 we compute a $2D$ solution with tolerance 10^{-4} and the above setup and only change the viscosity ν to be equal to one. Again, the iteration numbers are low and robust with respect to the parameters β and h.

4.2. With pressure term

In this section we show results for the problem including a pressure term in the objective function. The desired state and the desired pressure are obtained from solving the previously mentioned unsteady flow problem. In our case we now simply invert the mass matrix coming from the velocity space (as it is lumped) and use the Chebyshev semi-iteration for the mass matrix on the pressure space that corresponds to the pressure terms in the objective function, i.e., 20 steps of this method are typically employed.

Table 6 shows results for both preconditioners in the case of the problem including pressure terms in the objective function. As indicated in the eigenvalue analysis presented in Section 3.3 both preconditioners perform reasonably for large values of β but perform badly when β becomes small.

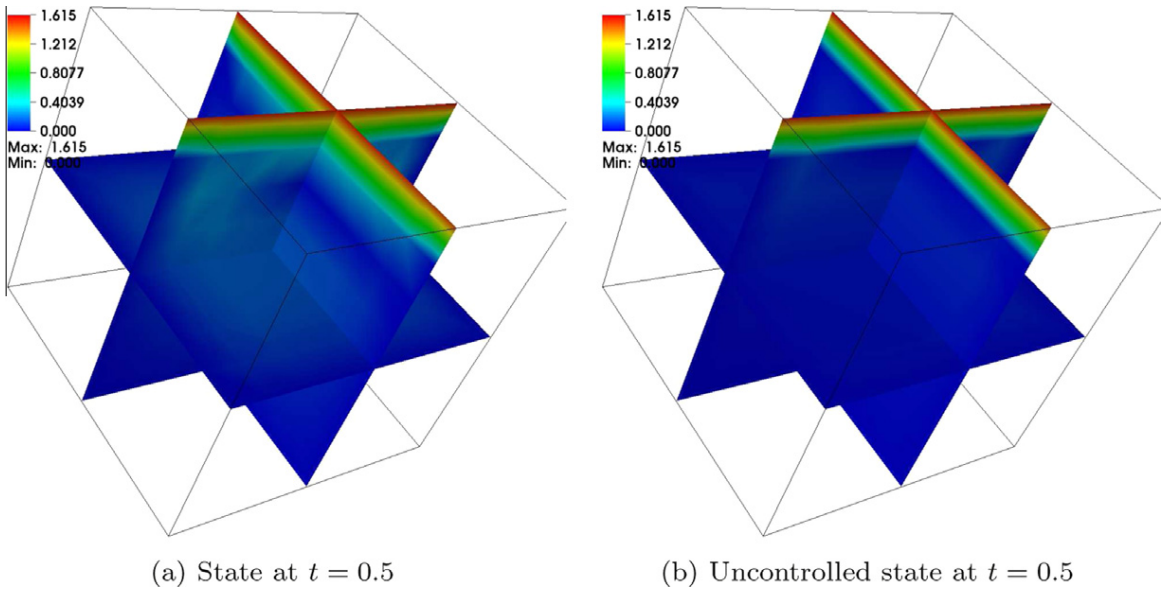


Fig. 11. Uncontrolled vs. controlled state.

Table 5
Number of MINRES steps and CPU-time with $\tau = 0.05$ and varying different values of β. Here the viscosity is set to $\nu = 1$. Results are shown for the preconditioner $\hat{S}_2$.

DoF	N	MINRES (Time) $\hat{S}_2$ ($\beta = 10^{-4}$)	MINRES (Time) $\hat{S}_2$ ($\beta = 10^{-6}$)
2467	21	16(356)	14(305)
9539	21	16(1352)	14(1221)

Table 6
Number of MINRES steps with CPU-time $\tau \approx h$ for different values of β in 3D. Results are shown for the preconditioner $\hat{S}_1$ and the preconditioner $\hat{S}_2$ using a tolerance 10^{-4}.

DoF	N	MINRES (Time) $\hat{S}_1$ ($\beta = 10^{0}$)	MINRES (Time) $\hat{S}_1$ ($\beta = 10^{-2}$)	MINRES (Time) $\hat{S}_2$ ($\beta = 10^{0}$)	MINRES (Time) $\hat{S}_2$ ($\beta = 10^{-2}$)
2312	9	24(442)	63(1107)	32(579)	91(1585)
15468	17	26(7012)	79(20112)	38(9931)	131(33821)
		$\hat{S}_1$ ($\beta = 10^{-4}$)	$\hat{S}_1$ ($\beta = 10^{-6}$)	$\hat{S}_2$ ($\beta = 10^{-4}$)	$\hat{S}_2$ ($\beta = 10^{-6}$)
2312	9	357(5989)	695(11619)	407(6817)	572(9779)

5. Conclusions and future work

We have shown that the discretization of the PDE-constrained optimal control problem involving unsteady Stokes flow as a PDE constraint can be efficiently cast using a Lagrangian technique into an all-at-once saddle point problem. As the dimensions of these type of problems are extremely large the use of iterative solvers is imperative. We have proposed the use of MINRES as the outer solver and block preconditioners. The Schur complement can efficiently be approximated using an inexact Uzawa method for which we have shown that the well-known Cahouet–Chabard preconditioner can be used. The iteration numbers for the outer MINRES method are always very low. We were able to introduce a preconditioner that showed robustness with respect to the regularization parameter β.

We believe that the results for the computation of the Stokes control problem will be very similar to the ones presented here if control constraints are present. In that case an outer Newton-type [21] method can be used and the linear systems that have to be solved at each step of the active set iteration are similar in nature to the ones for the problem with no bound constraints [45]. It might also be good to apply a nested approach where the solution is first approximated on a coarse mesh and then transferred to a fine discretization (see [20]). Another interesting aspect of the above problem is to consider the

more realistic scenario of boundary control for which we believe the regularization robust preconditioner can be extended. Of course, these problems should also be analyzed and tested numerically in the future.

Acknowledgement

The first author would like to thank Michael Köster for his advice on the discretization of the control problem. He would also like to thank Michele Benzi and Kent–Andre Mardal for pointing out references for the Cahouet–Chabard preconditioner. The authors are indebted to the anonymous referees for their comments, which helped to improve the paper a great deal.

References

[1] D. Arnold, F. Brezzi, M. Fortin, A stable finite element for the Stokes equations, Calcolo 21 (1984) 337–344.
[2] W. Bangerth, R. Hartmann, G. Kanschat, Deal. IIa general-purpose object-oriented finite element library, ACM Trans. Math. Software 33 (2007) 27. Art. 24.
[3] R. Bank, B. Welfert, H. Yserentant, A class of iterative methods for solving saddle point problems, Numer. Math. 56 (1989) 645–666.
[4] M. Benzi, G.H. Golub, J. Liesen, Numerical solution of saddle point problems, Acta Numer. 14 (2005) 1–137.
[5] M. Benzi, E. Haber, L. Taralli, A preconditioning technique for a class of PDE-constrained optimization problems, Adv. Comput. Math. 35 (2011) 149–173.
[6] M. Benzi, M.A. Olshanskii, Z. Wang, Modified augmented Lagrangian preconditioners for the incompressible Navier–Stokes equations, Int. J. Numer. Methods Fluids 66 (2011) 486–508.
[7] A. Borzi, K. Ito, K. Kunisch, An optimal control approach to optical flow computation, Int. J. Numer. Methods Fluids 40 (2002).
[8] J.H. Bramble, J.E. Pasciak, A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems, Math. Comput. 50 (1988) 1–17.
[9] J.H. Bramble, J.E. Pasciak, Iterative techniques for time dependent Stokes problems, Comput. Math. Appl. 33 (1997) 13–30.
[10] F. Brezzi, M. Fortin, Mixed and hybrid finite element methods, Springer Series in Computational Mathematics, Springer-Verlag, New York, 1991.
[11] J. Cahouet, J. Chabard, Some fast 3D finite element solvers for the generalized Stokes problem, Int. J. Numer. Methods Fluids 8 (1988) 869–895.
[12] H. Choi, M. Hinze, K. Kunisch, Instantaneous control of backward-facing step flows, Appl. Numer. Math. 31 (1999) 133–158.
[13] H.C. Elman, D.J. Silvester, A.J. Wathen, Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics, Numerical Mathematics and Scientific Computation, Oxford University Press, New York, 2005.
[14] R. Fletcher, Conjugate gradient methods for indefinite systems, in: Numerical analysis Proceedings 6th Biennial Dundee Conference, University, Lecture Notes in Mathematics, Dundee, Dundee 1975, Springer, Berlin, vol. 506, 1976, pp. 73–89.
[15] R.W. Freund, N.M. Nachtigal, QMR: a quasi-minimal residual method for non-Hermitian linear systems, Numer. Math. 60 (1991) 315–339.
[16] M. Gee, C. Siefert, J. Hu, R. Tuminaro, M. Sala, ML 5.0 Smoothed Aggregation User's Guide, Technical Report SAND2006-2649, Sandia National Laboratories, 2006.
[17] G.H. Golub, R.S. Varga, Chebyshev semi-iterative methods, successive over-relaxation iterative methods, and second order Richardson iterative methods. I, Numer. Math. 3 (1961) 147–156.
[18] G.H. Golub, R.S. Varga, Chebyshev semi-iterative methods, successive over-relaxation iterative methods, and second order Richardson iterative methods. II, Numer. Math. 3 (1961) 157–168.
[19] N.I.M. Gould, M.E. Hribar, J. Nocedal, On the solution of equality constrained quadratic programming problems arising in optimization, SIAM J. Sci. Comput. 23 (2001) 1376–1395.
[20] R. Herzog, E.W. Sachs, Preconditioned conjugate gradient method for optimal control problems with control and state constraints, SIAM J. Matrix Anal. Appl. 31 (2010) 2291–2317.
[21] M. Hintermüller, K. Ito, K. Kunisch, The primal-dual active set strategy as a semismooth Newton method, SIAM J. Optim. 13 (2002) 865–888.
[22] M. Hinze, Optimal and instantaneous control of the instationary Navier–Stokes equations, Habilitation, TU Berlin, 2000.
[23] M. Hinze, M. Köster, S. Turek, A hierarchical space-time solver for distributed control of the Stokes equation, Technical, Report, SPP1253-16-01, 2008a.
[24] M. Hinze, M. Köster, S. Turek, A space-time multigrid solver for distributed control of the time-dependent Navier–Stokes system, Technical, Report, SPP1253-16-02, 2008b.
[25] M. Hinze, R. Pinnau, M. Ulbrich, S. Ulbrich, Optimization with PDE Constraints, Mathematical Modelling: Theory and Applications, Springer-Verlag, New York, 2009.
[26] K. Ito, K. Kunisch, Lagrange multiplier approach to variational problems and applications, Advances in Design and Control, 15, Society for Industrial and Applied Mathematics (SIAM), PA, Philadelphia, 2008.
[27] C. Keller, N. Gould, A. Wathen, Constraint preconditioning for indefinite linear systems, SIAM J. Matrix Anal. Appl. 21 (2000) 1300–1317.
[28] Y. Maday, D. Meiron, A. Patera, E. Rnquist, Analysis of iterative methods for the steady and unsteady Stokes problem: application to spectral element discretizations, SIAM J. Sci. Comput. 14 (1993) 310.
[29] K. Mardal, R. Winther, Construction of preconditioners by mapping properties. Construction of preconditioners by mapping properties, Bentham Science Publishers, 2010. pp. 65–84.
[30] K. Mardal, R. Winther, Preconditioning discretizations of systems of partial differential equations, Numer. Linear Algebra Appl. 18 (2011) 1–40.
[31] C.C. Paige, M.A. Saunders, Solutions of sparse indefinite systems of linear equations, SIAM J. Numer. Anal. 12 (1975) 617–629.
[32] J.W. Pearson, M. Stoll, A. Wathen, Preconditioners for state constrained optimal control problems with Moreau–Yosida penalty function, Submitted for publication.
[33] J.W. Pearson, M. Stoll, A.J. Wathen, Regularization-robust preconditioners for time-dependent PDE-constrained optimization problems, Submitted for publication.
[34] J.W. Pearson, A.J. Wathen, A new approximation of the Schur complement in preconditioners for PDE constrained optimization, Numer. Linear Algebra Appl. in press.
[35] T. Rees, H.S. Dollar, A.J. Wathen, Optimal solvers for PDE-constrained optimization, SIAM J. Sci. Comput. 32 (2010) 271–298.
[36] T. Rees, M. Stoll, Block-triangular preconditioners for pde-constrained optimization, Numer. Linear Algebra Appl. 17 (2010) 977–996.
[37] T. Rees, M. Stoll, A. Wathen, All-at-once preconditioners for PDE-constrained optimization, Kybernetika 46 (2010) 341–360.
[38] T. Rees, A. Wathen, Preconditioning iterative methods for the optimal control of the Stokes equation, SIAM J. Sci. Comput. 33 (2011) 2903–2926.
[39] T. Rusten, R. Winther, A preconditioned iterative method for saddlepoint problems, SIAM J. Matrix Anal. Appl. 13 (1992) 887.
[40] Y. Saad, A flexible inner-outer preconditioned GMRES algorithm, SIAM J. Sci. Comput. 14 (1993). 461-461.
[41] Y. Saad, M.H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Statist. Comput. 7 (1986) 856–869.
[42] J. Schöberl, W. Zulehner, Symmetric indefinite preconditioners for saddle point problems with applications to pde-constrained optimization problems, SIAM J. Matrix Anal. Appl. 29 (2007) 752–773.
[43] D. Silvester, A. Wathen, Fast iterative solution of stabilised Stokes systems. II. Using general block preconditioners, SIAM J. Numer. Anal. 31 (1994) 1352–1367.

[44] M. Stoll, A. Wathen, All-at-once solution of time-dependent PDE-constrained optimization problems, Submitted for publication.
[45] M. Stoll, A. Wathen, Preconditioning for partial differential equation constrained optimization with control constraints, Numer. Linear Alg. Appl. 19 (2012) 53–71.
[46] F. Tröltzsch, Optimale Steuerung partieller Differentialgleichungen: Theorie, Verfahren und Anwendungen, Vieweg Verlag, Wiesbaden, 2005.
[47] F. Tröltzsch, Optimal Control of Partial Differential Equations: Theory Methods and Applications, American Mathematical Society, 2010.
[48] A. Wathen, D. Silvester, Fast iterative solution of stabilised Stokes systems. I. Using simple diagonal preconditioners, SIAM J. Numer. Anal. 30 (1993) 630–649.
[49] A.J. Wathen, T. Rees, Chebyshev semi-iteration in preconditioning for problems including the mass matrix, Electron. Trans. Numer. Anal. 34 (2008) 125–135.
[50] W. Zulehner, Non-standard norms and robust estimates for saddle point problems, SIAM J. Matrix Anal. Appl. 32 (2011) 536–560.

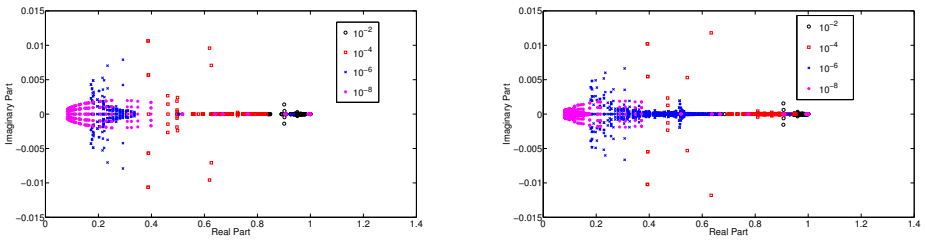
A.6 Preconditioning for the H1 norm

This paper is published as

A. T. Barker, T. Rees, and M. Stoll, ***A fast solver for an H1 regularized optimal control problem*, Commun. Comput. Phys., 19 (2016), pp. 143–167.**

Result from the paper

We develop robust solvers for optimization problems with H_1-regularization for the control. Figure A.2 shows eigenvalues for two different meshes and a variety of regularization parameters. We show coarse mesh on the left and slightly finer mesh on the right.



(a) Coarse (b) Finer

Figure A.2: Eigenvalues depending on regularization parameter.

Commun. Comput. Phys.
doi: 10.4208/cicp.190914.080415a

Vol. **19**, No. 1, pp. 143-167
January 2016

A Fast Solver for an $\mathcal{H}_1$ Regularized PDE-Constrained Optimization Problem

Andrew T. Barker[1], Tyrone Rees[2,*] and Martin Stoll[3]

[1] *Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Mail Stop L-561, Livermore, CA 94551, USA.*
[2] *Numerical Analysis Group, Scientific Computing Department, Rutherford Appleton Laboratory, Chilton, Didcot, Oxfordshire, OX11 0QX, United Kingdom.*
[3] *Computational Methods in Systems and Control Theory, Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, 39106 Magdeburg, Germany.*

Received 19 September 2014; Accepted (in revised version) 8 April 2015

Abstract. In this paper we consider PDE-constrained optimization problems which incorporate an $\mathcal{H}_1$ regularization control term. We focus on a time-dependent PDE, and consider both distributed and boundary control. The problems we consider include bound constraints on the state, and we use a Moreau-Yosida penalty function to handle this. We propose Krylov solvers and Schur complement preconditioning strategies for the different problems and illustrate their performance with numerical examples.

AMS subject classifications: 49M25, 49K20, 65F10, 65N22, 65F50, 65N55

Key words: Preconditioning, Krylov methods, PDE-constrained optimization, optimal control of PDEs.

1 Introduction

As methods for numerically solving partial differential equations (PDEs) become more accurate and well-understood, some focus has shifted to the development of numerical methods for optimization problems with PDE constraints: see, e.g., [41,44,69] and the references mentioned therein. The canonical PDE-constrained optimization problem takes a given *desired state*, $\bar{y}$, and finds a *state*, y, and a *control*, u, to minimize the functional

$$\|y-\bar{y}\|_{\mathcal{Y}}^2+\frac{\beta}{2}R(u) \tag{1.1}$$

*Corresponding author. *Email addresses:* barker29@llnl.gov (A. T. Barker), tyrone.rees@stfc.ac.uk (T. Rees), stollm@mpi-magdeburg.mpg.de (M. Stoll)

subject to the constraints

$$\begin{aligned}&\mathcal{A}y = u,\\ &u_a \le u \le u_b,\\ &y_a \le y \le y_b,\end{aligned}$$

where $\|\cdot\|_{\mathcal{Y}}$ is some norm and $R(u)$ is a regularization functional. We are free to choose both the norm and the regularization functional here; appropriate choices often depend on the properties of the underlying application. In the description above $\mathcal{A}$ denotes a PDE with appropriate boundary conditions and β denotes a scalar regularization parameter. The focus of this manuscript is regularization based on the H_1 norm of the control, which we motivate below.

The simplest choice of $R(u)$ is $\|u\|^2_{L_2(\Omega)}$, where Ω denotes the domain on which the PDE is posed. This case has been well-studied in the literature, both from a theoretical and algorithmic perspective. However, the requirements of real-world problems has necessitated the application of alternative regularization terms.

One area where there has been much interest is in regularization using L_1 norms, see, e.g., the recent articles [12, 73]. A related norm is the total variation norm $R(u) = \|\nabla u\|_{L_1(\Omega)}$, has also aroused excitement recently – see e.g. [14, 59] and the references therein. These L_1 norms have the benefit that they allow discontinuous controls, which can be important in certain applications.

For certain applications it is desirable to have a smooth control – for this reason the $\mathcal{H}_1$ semi-norm, $R(u) = \|\nabla u\|^2_{L_2(\Omega)}$, has long been studied in the context of parameter-estimation problems [10, 46, 76], image-deblurring [13, 17, 48], image reconstruction [49], and flow control [18, 34], for example. Recently van den Doel, Ascher and Haber [19] argued that this norm can be a superior choice to its L_1-based cousin, total variation, for problems with particularly noisy data due to the smooth nature of controls which arise. The test problems in PDE constrained optimization by Haber and Hanson [31], which were designed to get academics solving problems more in-line with the needs of the real-world, suggest a regularization functional of the form $R(u) = \|u\|^2_{L_2(\Omega)} + \alpha\|\nabla u\|^2_{L_2(\Omega)}$ for a given α. Indeed, this form of regularization is commonly used in the ill-posed and inverse problem communities. Another example of a field where the standard L_2 regularization may not be appropriate is flow control – see, e.g., Gunzburger [28, Chapter 4].

At the heart of many techniques for solving the optimization problem, whether it is a linear problem or the linearization of a non-linear problem, lies the solution of a linear system [35, 41, 44, 70]. These systems are very often so-called saddle point matrices [4, 23], which have the form

$$\mathcal{A} = \begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix}, \tag{1.2}$$

where A represents the misfit and regularization terms in (1.1) and B represents the PDE constraint. In the systems we consider in this paper, A is symmetric positive semi-definite. Such saddle point matrices are invertible if B has full rank and $\ker(A) \cap \ker(B) =$

$\{\mathbf{0}\}$: this condition holds for most of the examples we consider here, and in the cases where it doesn't – e.g. (2.3-2.4) – there is a well-understood one dimensional null-space that can be straightforwardly dealt with [4, Section 3.2]. We are then left with the challenge of efficiently solving linear systems of the form (1.2).

Direct solvers based on factorizations [21] can be effective, but for large and, in particular, three-dimensional problems these are no longer sufficient. In such cases we turn to iterative Krylov subspace methods, which can deal with these large and sparse systems efficiently provided that they employ a preconditioner which enhances the convergence behaviour, ideally independent of problem-dependent parameters such as the mesh-size or the regularization parameter. For a general overview of preconditioners we refer to [29,61], and in the particular case of saddle point problems see [4,23,77].

A number of preconditioners which are robust with respect to regularization parameters and mesh-parameters have recently been developed for PDE-constrained optimization [1,15,20,36,47,52,53,65]. However, these methods are tailored for an optimization problem with $R(u)=\|u\|^2_{L_2(\Omega)}$ and heavily rely on the corresponding presence of a mass matrix in the A block of (1.2). Benzi, Haber and Taralli [5] consider a block preconditioner with of $R(u)$ given by (a variant of) the $\mathcal{H}_1-$norm, but their approach is general enough to work with most regularization and the form of this term is not exploited in the method. To the authors' knowledge there have been no other attempts to apply block preconditioners – which have proved so successful with L_2 regularization – in the case of other choices of $R(u)$. We address this issue here by considering a cost-functional where

$$R(u)=\|u\|^2_{L_2}+\|\nabla u\|^2_{L_2},$$

and we present preconditioners that show robustness with respect to the regularization parameter for this problem, which is more challenging from a linear algebra perspective.

In the following we use the heat equations as an example PDE. In principle the approaches described here can be extended to other PDEs, as for the L_2 regularization case. We deliberately choose to focus on the simplest PDE example to highlight the issues corresponding directly to the regularization, not the difficulties involved in using a more complicated model, which is discussed elsewhere.

The structure of the paper is as follows. We begin in Section 2 by stating the optimal control problem in the time-dependent and time-independent cases with both distributed and boundary control. We illustrate how to obtain discretized first order conditions from a so-called discretize-then-optimize approach. In Section 3 we describe how – following a method first proposed by Ito and Kunisch [43] – the state constraints can be handled using a Moreau-Yosida penalty approach and show how to incorporate this into possible preconditioning strategies. Sections 2 and 3, which describe the application of well known techniques for solving such optimal control problems, show how the bottleneck for such codes is the solution of a very large linear system.

In Section 4 we discuss the choice of possible Krylov solvers and introduce preconditioning strategies for both the time-dependent and time-independent control problem, with an emphasis on how to handle the $\mathcal{H}_1$ regularization term. This builds on the work

in the literature that has been used to efficiently solve L_2 regularized problems, but the use of the $\mathcal{H}_1$ norm in the cost functional causes difficulties which require novel techniques to overcome. The development of such techniques is the main contribution of the paper. Our numerical results shown in Section 6 illustrate the efficiency of our approach.

2 Problem setup and discretization

2.1 A stationary control problem

Before describing the time-dependent control problem we fix ideas by considering a stationary optimal control problem. We wish to minimize the functional

$$\begin{aligned}\mathcal{J}_1(y,u) &= \frac{1}{2}\|y-\bar{y}\|^2_{L_2(\Omega_1)} + \frac{\beta}{2}\|u\|^2_{\mathcal{H}_1(\Omega_2)} \\ &= \frac{1}{2}\|y-\bar{y}\|^2_{L_2(\Omega_1)} + \frac{\beta}{2}\|u\|^2_{L_2(\Omega_2)} + \frac{\beta}{2}\|\nabla u\|^2_{L_2(\Omega_2)},\end{aligned} \tag{2.1}$$

where both Ω_1 and Ω_2 are subdomains of $\Omega \in \mathbb{R}^d$ with $d=2,3$. The constraint is given by the following elliptic PDE

$$-\triangle y = \begin{cases} u & \text{in } \Omega_2, \\ 0 & \text{in } \Omega \backslash \Omega_2, \end{cases} \tag{2.2}$$

together with Dirichlet boundary conditions, $y=g$ on $\partial\Omega$. We refer to y as the state and u as the corresponding control, which is used to drive the state variable as close as possible to the desired state (or observations) $\bar{y}$. The above problem is the distributed control problem, as u defines the forcing of the PDE over the interior subdomain Ω_2. Another important case is given by the Neumann boundary control problem, where $\Omega_2 = \partial\Omega$ together with the PDE constraint

$$-\triangle y = f \quad \text{in } \Omega, \tag{2.3}$$

$$\frac{\partial y}{\partial n} = u \quad \text{on } \partial\Omega, \tag{2.4}$$

where f represents a fixed forcing term.

In practice, physical characteristics of the application will require *box constraints* on the control and/or the state. Typical bounds would be

$$u_a \leq u \leq u_b$$

for the control and

$$y_a \leq y \leq y_b$$

for the state. The numerical treatment of these constraints is by now well established [6,7,38] but nevertheless represents a computational challenge, in particular for the state constraints [11].

We follow the discretize-then-optimize paradigm and discretize the PDE and the objective function using Q1 finite elements [23, 67]; we employ the deal.II [2] finite element package for our numerical experiments.

We derive the discrete optimality system for the cost functional (2.1) with the PDE constraint (2.2), together with homogeneous Dirichlet boundary conditions for ease of exposition – the extension to other boundary conditions proceeds similarly. Let $\phi_1,\cdots\phi_n$ be a finite element basis for the interior of Ω, and suppose we extend this by $\phi_{n+1},\cdots,\phi_{n+\partial n}$ to include the boundary. Let $Y_0^h=\langle\phi_1\cdots\phi_n\rangle$, $U^h=\langle\phi_1\cdots\phi_n,\phi_{n+1},\phi_{n+\partial n}\rangle$. Furthermore, let $Y_{\Omega_1}:=\langle\hat{\phi}_1\cdots\hat{\phi}_{\hat{m}}\rangle$ and $U_{\Omega_2}:=\langle\bar{\phi}_1\cdots\bar{\phi}_{\bar{m}}\rangle$ denote the subsets of U^h with support on Ω_1 and Ω_2 respectively.

The finite dimensional analogue to (2.1), (2.2) is to find $y_h\in Y_0^h\subset\mathcal{H}_0^1(\Omega)$ and $u_h\in U^h\subset\mathcal{H}_1(\Omega)$ which satisfy

$$\min_{y_h\in Y_{\Omega_1},u_h\in U_{\Omega_2}}\frac{1}{2}||y_h-\bar{y}||^2_{L_2(\Omega_1)}+\frac{\beta}{2}||u_h||^2_{\mathcal{H}_1(\Omega_2)},$$
$$\text{s.t.}\quad\int_\Omega\nabla y_h\cdot\nabla v_h=\int_{\Omega_2}u_hv_h,\ \ \forall v_h\in Y_0^h.$$

We can write the optimization problem in terms of matrices as

$$\min_{\mathbf{y},\mathbf{u}}\frac{1}{2}\mathbf{y}^TM_y\mathbf{y}-\mathbf{y}^T\mathbf{b}+\frac{\beta}{2}\mathbf{u}^TM_u\mathbf{u}+\frac{\beta}{2}\mathbf{u}^TK_u\mathbf{u}, \tag{2.5}$$
$$\text{s.t.}\quad K\mathbf{y}=M\mathbf{u}, \tag{2.6}$$

where

$$(M_y)_{i,j}=\int_\Omega\hat{\phi}_i\hat{\phi}_j,\ \ i,j=1,\cdots,\hat{m},\qquad (K_u)_{i,j}=\int_\Omega\nabla\bar{\phi}_i\cdot\nabla\bar{\phi}_j,\ \ i,j=1,\cdots,\bar{m},$$
$$(M_u)_{i,j}=\int_\Omega\bar{\phi}_i\bar{\phi}_j,\ \ i,j=1,\cdots,\bar{m},\qquad (K)_{i,j}=\int_\Omega\nabla\phi_i\cdot\nabla\phi_j,\ \ i,j=1,\cdots,n,$$
$$(M)_{i,j}=\int_\Omega\phi_i\bar{\phi}_j,\ \ i=1,\cdots,n,\ \ j=1,\cdots,\bar{m},\qquad \mathbf{b}_i=\int_\Omega\bar{y}\phi_i,\ i=1,\cdots,\hat{m}.$$

Note that in this paper we only discuss the case where $\Omega_2=\partial\Omega$ or $\Omega_2=\Omega$ and $\Omega_1=\Omega$. Other choices influence the matrix properties of M_y,M_u,K_u,M for which the techniques presented here are still applicable.

In the distributed control case the first order optimality conditions lead to the following saddle point system:

$$\begin{bmatrix}M_y & 0 & -K^T\\ 0 & \beta M_u+\beta K_u & M^T\\ -K & M & 0\end{bmatrix}\begin{bmatrix}\mathbf{y}\\ \mathbf{u}\\ \mathbf{p}\end{bmatrix}=\begin{bmatrix}\mathbf{b}\\ \mathbf{0}\\ \mathbf{0}\end{bmatrix}. \tag{2.7}$$

Note that the addition of an $\mathcal{H}_1$ norm in the regularization leads to an optimality system with substantially different properties compared to the L_2 case; in particular, if $\mathbf{p}=\mathbf{0}$

on the boundary, we do not necessarily have that $\mathbf{u}=\mathbf{0}$ on the boundary here, which is known to be true if we use L_2 regularization (see [58, 69, Section 2.8]). If we were to use non-homogeneous boundary conditions the 3rd entry of the right hand side would hold the boundary data, as the state equation (2.6) would become $K\mathbf{y}=M\mathbf{u}-\mathbf{d}$ for some non-zero vector $\mathbf{d}$.

We treat the boundary control problem similarly. Here we get

$$\mathcal{J}_1(\mathbf{y},\mathbf{u})=\frac{1}{2}\mathbf{y}^T M_y\mathbf{y}-\mathbf{b}^T\mathbf{y}+\frac{\beta}{2}\mathbf{u}^T M_{u,b}\mathbf{u}+\frac{\beta}{2}\mathbf{u}^T K_{u,b}\mathbf{u} \tag{2.8}$$

together with

$$\widehat{K}\mathbf{y}=\widehat{N}\mathbf{u}+\mathbf{f}. \tag{2.9}$$

Here $M_{u,b}$ and $K_{u,b}$ are the boundary mass matrix and Laplacian, respectively, i.e.

$$(K_{u,b})_{i,j}=\int_{\partial\Omega}\nabla\mathrm{tr}(\phi_i)\cdot\nabla\mathrm{tr}(\phi_j),\ (M_{u,b})_{i,j}=\int_{\partial\Omega}\mathrm{tr}(\phi_i)\mathrm{tr}(\phi_j),\quad i,j=n+1,\cdots,n+\partial n,$$

where $\mathrm{tr}(\cdot)$ is the trace operator, which we use here to give us a finite element discretization of the boundary. The vector $\mathbf{f}$ represents the discretized forcing term, which for simplicity we take to be zero for the remainder of the paper. The matrix $\widehat{K}$ is the stiffness matrix, including the boundary nodes, and $\widehat{N}$ connects interior and boundary basis functions, in particular

$$(\widehat{N})_{ij}=\int_{\partial\Omega}\phi_i\mathrm{tr}(\phi_j),\quad i=1,\cdots,n+\partial n,\quad j=1,\cdots\partial n.$$

We obtain the following first order optimality system

$$\begin{bmatrix} M_y & 0 & -\widehat{K}^T \\ 0 & \beta M_{u,b}+\beta K_{u,b} & \widehat{N}^T \\ -\widehat{K} & \widehat{N} & 0 \end{bmatrix}\begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \mathbf{p} \end{bmatrix}=\begin{bmatrix} \mathbf{b} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}. \tag{2.10}$$

2.2 Time-dependent problem

We now present a time-dependent version, which is of wide practical interest and will be the focus of our numerical tests. The objective function is now given by

$$\mathcal{J}_2(y,u)=\frac{1}{2}\int_0^T\int_{\Omega_1}(y-\bar{y})^2dxdt+\frac{\beta}{2}\int_0^T\int_{\Omega_2}u^2dxdt+\frac{\beta}{2}\int_0^T\int_{\Omega_2}(\nabla u)^2dxdt, \tag{2.11}$$

where all functions are simply time-dependent versions of their steady counterparts presented above. For the distributed control problem we apply the time-dependent parabolic constraint

$$\begin{aligned} &y_t-\triangle y=\begin{cases} u, & \text{for } (\mathbf{x},t)\in\Omega_2\times[0,T], \\ 0, & \text{for } (\mathbf{x},t)\in\Omega\backslash\Omega_2\times[0,T], \end{cases} \\ &y=g,\quad \text{on } \partial\Omega, \\ &y=y_0,\quad \text{at } t=0, \end{aligned}$$

for some prescribed functions g, y_0. In case of a boundary control problem, where $\Omega_2 = \partial\Omega$ and again take the heat equation as our PDE constraint:

$$y_t - \triangle y = f \quad \text{for } (\mathbf{x},t) \in \Omega \times [0,T], \tag{2.12}$$

$$\frac{\partial y}{\partial n} = u \quad \text{on } \partial\Omega, \tag{2.13}$$

$$y = y_0, \quad \text{at } t = 0. \tag{2.14}$$

For the discretization of the time-dependent objective function we use the trapezoidal rule for the time integral and finite elements in space to give

$$\mathcal{J}_2(\mathbf{y},\mathbf{u}) = \frac{1}{2}\mathbf{y}^T\mathcal{M}_y\mathbf{y} + \widehat{\mathbf{b}}^T\mathbf{y} + \frac{\beta}{2}\mathbf{u}^T\mathcal{M}_u\mathbf{u} + \frac{\beta}{2}\mathbf{u}^T\mathcal{K}_u\mathbf{u}, \tag{2.15}$$

where $\widehat{\mathbf{b}} = [1/2\mathbf{b}^T, \mathbf{b}^T, \cdots, \mathbf{b}^T, 1/2\mathbf{b}^T]^T$,

$$\begin{aligned}
&\mathcal{M} = \text{blkdiag}(M_y, \cdots, M_y),\\
&\mathcal{M}_y = \text{blkdiag}(1/2M_y, M_y, \cdots, M_y, 1/2M_y),\\
&\mathcal{M}_u = \text{blkdiag}(1/2M_u, M_u, \cdots, M_u, 1/2M_u), \quad \text{and}\\
&\mathcal{K}_u = \text{blkdiag}(1/2K_u, K_u, \cdots, K_u, 1/2K_u),
\end{aligned}$$

which are simply block-variants of the previously defined matrices over the domains Ω_1 and Ω_2. Note that in the time-dependent case we abuse the notation $\mathbf{y}$, $\mathbf{u}$ defined earlier, i.e., $\mathbf{y} = \left[\mathbf{y}_1^T, \mathbf{y}_2^T, \cdots, \mathbf{y}_{N_T}^T\right]^T$, etc.; we believe it will be clear from the context which of the two we are currently considering. Using this notation and a backward Euler scheme, we can write down a one-shot discretization of the time-dependent PDE as follows

$$-\underbrace{\begin{bmatrix} L & & & \\ -M & L & & \\ & \ddots & \ddots & \\ & & -M & L \end{bmatrix}}_{\mathcal{K}} \mathbf{y} + \tau\mathcal{M}\mathbf{u} = \mathbf{d} \tag{2.16}$$

with $L = M + \tau K$ and $\mathbf{d}$ holding the initial conditions for the heat equation. For more details see [5, 20, 66].

We form the Lagrangian and write down the first order conditions in a linear system,

$$\begin{bmatrix} \tau\mathcal{M}_y & 0 & -\mathcal{K}^T \\ 0 & \tau\beta(\mathcal{M}_u + \mathcal{K}_u) & \tau\mathcal{M} \\ -\mathcal{K} & \tau\mathcal{M} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \tau\widehat{\mathbf{b}} \\ 0 \\ \mathbf{d} \end{bmatrix}, \tag{2.17}$$

in the case of the distributed control problem, and

$$\begin{bmatrix} \tau\mathcal{M}_y & 0 & -\mathcal{K}^T \\ 0 & \tau\beta(\mathcal{M}_{u,b} + \mathcal{K}_{u,b}) & \tau\mathcal{N}^T \\ -\mathcal{K} & \tau\mathcal{N} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} M_y\bar{\mathbf{y}} \\ 0 \\ \mathbf{d} \end{bmatrix} \tag{2.18}$$

for boundary control, where $\mathcal{N} = \text{blkdiag}(N, \cdots, N)$.

3 Handling the state constraints

Box constraints for the state $\mathbf{y}$ can be dealt with efficiently using a penalty term. The Moreau-Yosida penalty function has proven to be a viable tool: see [36, 43, 52] and the references mentioned therein. One can also use the Moreau-Yosida technique for box constraints on the control but the primal-dual active set method [38] is mostly the method of choice. We briefly describe the Moreau-Yosida technique for the distributed control problem. A more thorough discussion can be found in the references mentioned earlier. The modified objective function becomes

$$\mathcal{J}_{MY}(y,u)=\mathcal{J}_2(y,u)+\frac{1}{2\varepsilon}\|\max\{0,y-y_b\}\|_Q^2+\frac{1}{2\varepsilon}\|\min\{0,y-y_a\}\|_Q^2 \tag{3.1}$$

for the state constrained case. Here $Q=\Omega_1\times[0,T]$ is the space-time cylinder. In accordance with [36], we can employ a semi-smooth Newton scheme that leads to the following linear system

$$\begin{aligned}&\begin{bmatrix}\tau\mathcal{M}_y+\varepsilon^{-1}G_{\mathcal{A}}\mathcal{M}_yG_{\mathcal{A}} & 0 & -\mathcal{K}^T\\ 0 & \tau\beta(\mathcal{M}_{u,b}+\mathcal{K}_{u,b}) & \tau\mathcal{N}^T\\ -\mathcal{K} & \tau\mathcal{N} & 0\end{bmatrix}\begin{bmatrix}\mathbf{y}\\ \mathbf{u}\\ \mathbf{p}\end{bmatrix}\\ =&\begin{bmatrix}\mathcal{M}_y\bar{\mathbf{y}}+\varepsilon^{-1}\left(G_{\mathcal{A}_+}\mathcal{M}_yG_{\mathcal{A}_+}y_b+G_{\mathcal{A}_-}\mathcal{M}_yG_{\mathcal{A}_-}y_a\right)\\ 0\\ \mathbf{d}\end{bmatrix},\end{aligned} \tag{3.2}$$

where the block-diagonal matrix

$$G_{\mathcal{A}}\mathcal{M}_yG_{\mathcal{A}}=\text{blkdiag}(G_{\mathcal{A}^1}M_yG_{\mathcal{A}^1},\cdots,G_{\mathcal{A}^{N_T}}M_yG_{\mathcal{A}^{N_T}})$$

defines the contribution of the penalty term with the active set $\mathcal{A}^k$ for time-step k defined as follows. We set where we define the active sets as $\mathcal{A}_+^k=\{i:\mathbf{y}_i^k>(y_b)_i^k\}$, and $\mathcal{A}_-^k\{i:\mathbf{y}_i^k<(y_a)_i^k\}$, and $\mathcal{A}^k=\mathcal{A}_+^k\cup\mathcal{A}_-^k$; the matrices G are diagonal matrix variants of the characteristic function for the corresponding sets, i.e.,

$$(G_{\mathcal{A}^k})_{ii}=\begin{cases}1 & \text{for } i\in\mathcal{A}^k,\\ 0 & \text{otherwise.}\end{cases}$$

Our focus is on the efficient solution of the linear systems (3.2), which are of saddle point type. Note that the active sets defined above within an iterative process such as the semi-smooth Newton scheme are computed based on the state at the previous iteration, but for simplicity we neglect the iteration index. For more details of semi-smooth Newton methods we refer to [41, 44, 70]; there is also recent theory introducing path-following approaches for the penalty parameter ε [39].

4 Preconditioning

4.1 Choice of Krylov solver and Schur complement preconditioning

As mentioned in the introduction, the linear systems that arise from PDE-constrained optimization are very often too large for direct solvers to be effective, and for scalable and efficient solution of these linear systems the combination of a state-of-the-art solver with an efficient preconditioning technique is crucial. In this section we derive preconditioners for each of the problems presented earlier, but first mention the choice of the iterative scheme. Krylov solvers are for many applications the method of choice [64], as they are cheap to apply; at each step they only require a matrix vector product, the evaluation of the preconditioners, and the evaluation of inner products. These methods build up a low-dimensional subspace that can be used to approximate the solution to the linear system.

There are a variety of Krylov subspace methods, and the most effective to use depends on the properties of the linear system. Here we focus on the development of effective preconditioners and we will focus less on the choice of linear solver.

Schur-complement based preconditioners, based on approximations to $S := BA^{-1}B^T$, have proved to be effective. Popular choices are a block diagonal preconditioner $\mathcal{P}_1 = \text{blkdiag}(A,S)$, or a nonsymmetric preconditioner,

$$\mathcal{P}_2 = \begin{pmatrix} A & 0 \\ B & -S \end{pmatrix}.$$

Naturally, these are too expensive for any realistic problem, but if we can approximate both the $(1,1)$-block and the Schur-complement of $\mathcal{A}$, then the underlying Krylov method will converge in a small number of steps. In the following sections we describe how to find good approximations to these blocks for the application considered here.

4.2 The $(1,1)$-block

Our first goal is to efficiently approximate the $(1,1)$-block of the saddle point matrix. Parts of the $(1,1)$-block here consist of lumped mass matrices, which are diagonal and can simply be inverted. If, on the other hand, the user prefers to use consistent mass matrices they can use the Chebyshev semi-iteration [72]. If the $(1,1)$-block part corresponding to the discretization of the state misfit part of the objective function is only semi-definite, e.g., via a partial observation operator, we can add a small perturbation to the zero blocks within the preconditioned and hence make this part positive definite so the above applies. In more detail, we replace the zero blocks in A by blocks of the form ηI with η a small parameter greater than zero. Note that this technique can also be used for an approximation of the Schur-complement in case the $(1,1)$-block is semi-definite [5,66].

The matrix part corresponding to the discretization of the $\mathcal{H}_1$ term in the objective function is more complicated as it is not diagonal. The good news in this case is that

the operator and the corresponding matrix representation are not only symmetric but also positive definite. This allows the use of either geometric [32,75] or algebraic [24,60] multigrid techniques.

4.3 Schur-complement approximation

The methods described in Section 4.2 efficiently approximate the $(1,1)$-block, A, of the saddle point system; we use $\widehat{A}$ to represent such an approximation to A for the remainder of this paper. Our goal now is to introduce efficient approximations $\hat{S}$ to the Schur-complement S.

The Schur complement of the system matrix (2.17) is

$$S=\tau^{-1}\mathcal{K}\mathcal{M}_y^{-1}\mathcal{K}+\tau\beta^{-1}\mathcal{M}(\mathcal{M}_u+\mathcal{K}_u)^{-1}\mathcal{M}^T. \tag{4.1}$$

There are various ways to approximate S; one of the simplest is

$$S\approx\tau^{-1}\mathcal{K}\mathcal{M}_y^{-1}\mathcal{K},$$

which for larger β often performs well but is not robust with respect to this parameter.

In order to develop a more robust method we look for a more sophisticated approximation inspired by [53] that more accurately mirrors S by also including the second term in (4.1). We have two options here, either a symmetric version,

$$\hat{S}_1=\tau^{-1}\left(\mathcal{K}+\hat{\mathcal{M}}\right)\mathcal{M}_y^{-1}\left(\mathcal{K}+\hat{\mathcal{M}}\right)^T,$$

which can be used within MINRES [51], or a non-symmetric approximation

$$\hat{S}_2=\tau^{-1}\left(\mathcal{K}+\hat{\mathcal{M}}_1\right)\mathcal{M}_y^{-1}\left(\mathcal{K}+\hat{\mathcal{M}}_2\right)^T$$

to be employed with a non-symmetric solver, e.g. GMRES [63] or BICG [25]. The goal is now to find $\hat{\mathcal{M}}_1$, $\hat{\mathcal{M}}_2$, and $\hat{\mathcal{M}}$ such that

$$\tau^{-1}\hat{\mathcal{M}}_1\mathcal{M}_y^{-1}\hat{\mathcal{M}}_2^T=\tau\beta^{-1}\mathcal{M}(\mathcal{M}_u+\mathcal{K}_u)^{-1}\mathcal{M}^T$$

and

$$\tau^{-1}\hat{\mathcal{M}}\mathcal{M}_y^{-1}\hat{\mathcal{M}}^T=\tau\beta^{-1}\mathcal{M}(\mathcal{M}_u+\mathcal{K}_u)^{-1}\mathcal{M}^T.$$

We start by deriving the symmetric approximation to S using

$$\hat{\mathcal{M}}:=\frac{\tau}{\sqrt{\beta}}\mathcal{M}(\mathcal{M}_u+\mathcal{K}_u)^{-1/2}\mathcal{M}_y^{1/2}.$$

We then obtain the following Schur-complement approximation

$$\hat{S}_1=\tau^{-1}\left(\mathcal{K}+\frac{\tau}{\sqrt{\beta}}\mathcal{M}(\mathcal{M}_u+\mathcal{K}_u)^{-1/2}\mathcal{M}_y^{1/2}\right)\mathcal{M}_y^{-1}\left(\mathcal{K}+\frac{\tau}{\sqrt{\beta}}\mathcal{M}(\mathcal{M}_u+\mathcal{K}_u)^{-1/2}\mathcal{M}_y^{1/2}\right)^T.$$

This has the advantage that the approximation is symmetric and positive definite, which would allow us to use MINRES. However, the drawback is that this expression involves the square root of large-scale non-diagonal matrices, $\mathcal{K}_u$.

We now turn our attention to the non-symmetric approximation. Using

$$\hat{\mathcal{M}}_1 := \frac{\tau}{\sqrt{\beta}}\mathcal{M}(\mathcal{M}_u+\mathcal{K}_u)^{-1}\mathcal{M}_y, \tag{4.2}$$

$$\hat{\mathcal{M}}_2 := \frac{\tau}{\sqrt{\beta}}\mathcal{M}, \tag{4.3}$$

we introduce the non-symmetric approximation

$$\hat{S}_2 = \tau^{-1}\left(\mathcal{K}+\frac{\tau}{\sqrt{\beta}}\mathcal{M}(\mathcal{M}_u+\mathcal{K}_u)^{-1}\mathcal{M}_y\right)\mathcal{M}_y^{-1}\left(\mathcal{K}+\frac{\tau}{\sqrt{\beta}}\mathcal{M}\right)^T.$$

This configuration does not require the square root of a potentially very large matrix.

For any preconditioner to be effective we must be able to evaluate the inverse of the Schur-complement approximation quickly. We now focus on the non-symmetric approximation but discuss the symmetric approximation in Section 5 when we analyze the approximation quality of both Schur-complement approximations.

The second part $\left(\mathcal{K}+\frac{\tau}{\sqrt{\beta}}\mathcal{M}\right)$ of $\hat{S}_2$ is easy to approximate as this is simply a block-triangular matrix with symmetric positive definite matrices along the diagonal. We therefore use an algebraic multigrid approximation for the diagonal blocks and then proceed backwards approximating the inverse of $\left(\mathcal{K}+\frac{\tau}{\sqrt{\beta}}\mathcal{M}\right)$ requiring the application of N_T algebraic multigrid operators.

The approximation of the inverse of $\left(\mathcal{K}+\frac{\tau}{\sqrt{\beta}}\mathcal{M}(\mathcal{M}_u+\mathcal{K}_u)^{-1}\mathcal{M}_y\right)$ is more involved. We are interested in solving systems of the form

$$\left(\mathcal{K}+\frac{\tau}{\sqrt{\beta}}\mathcal{M}(\mathcal{M}_u+\mathcal{K}_u)^{-1}\mathcal{M}_y\right)u = f$$

and interpret this as the Schur-complement of the auxiliary system

$$\begin{bmatrix} \mathcal{K} & \mathcal{M} \\ \mathcal{M}_y & -\frac{\sqrt{\beta}}{\tau}(\mathcal{M}_u+\mathcal{K}_u) \end{bmatrix}\begin{bmatrix} u \\ * \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix}. \tag{4.4}$$

Recalling the block-structure of the involved matrices, it is easy to see that we can proceed with a forward substitution that requires the solution of diagonal blocks given by

$$\begin{bmatrix} M+\tau K & M \\ M_y & -\frac{\sqrt{\beta}}{\tau}(M_u+K_u) \end{bmatrix}. \tag{4.5}$$

Even this block is not suitable to be inverted directly and we use a stationary iteration to approximate the solution to this system. Such an iteration proceeds by computing

$$u^{k+1} = u^k + \omega W^{-1} r_k,$$

where r_k is the residual for the system matrix used in (4.5) and a right-hand-side used within the preconditioner application. The matrix

$$W = \begin{bmatrix} \widehat{M+\tau K} & \\ & \frac{\sqrt{\beta}}{\tau}(\widehat{M_u+K_u}) \end{bmatrix}$$

is the preconditioner for (4.5). Here $\widehat{(\cdots)}$ signifies the algebraic multigrid approximation to the corresponding matrix.

Boundary control

The matrix structure in the case of a boundary control problem is very similar to the distributed control problem but nevertheless there are significant differences in the properties of some of the blocks. Therefore, we now discuss a Schur complement approximation for the boundary control problem driven by the system matrix (2.18). The Schur complement is now given by

$$S = \tau^{-1}\mathcal{K}\mathcal{M}_y^{-1}\mathcal{K}^T + \frac{\tau}{\beta}\mathcal{N}(\mathcal{M}_{u,b}+\mathcal{K}_{u,b})^{-1}\mathcal{N}^T.$$

For the reasons described above, we again focus on the non-symmetric approximation

$$\hat{S} = \tau^{-1}\left(\mathcal{K}+\frac{\tau}{\sqrt{\beta}}\mathcal{N}(\mathcal{M}_{u,b}+\mathcal{K}_{u,b})^{-1}\mathcal{N}^T\right)\mathcal{M}_y^{-1}\left(\mathcal{K}+\frac{\tau}{\sqrt{\beta}}\mathcal{M}_y\right)^T.$$

Again, the evaluation of the preconditioner $\hat{S}^{-1}$ needs to be discussed. While the term $\left(\mathcal{K}+\frac{\tau}{\sqrt{\beta}}\mathcal{M}_y\right)^{-T}$ can easily be approximated using multigrid techniques in combination with backward substitution, the term $\left(\mathcal{K}+\frac{\tau}{\sqrt{\beta}}\mathcal{N}(\mathcal{M}_{u,b}+\mathcal{K}_{u,b})^{-1}\mathcal{N}^T\right)^{-1}$ is more complicated to approximate. Note again that we only need to focus on the diagonal blocks of this matrix which correspond to the system

$$\begin{bmatrix} M+\tau K & N \\ N^T & -\frac{\sqrt{\beta}}{\tau}(M_{u,b}+K_{u,b}) \end{bmatrix}. \tag{4.6}$$

We proposed earlier the use of a stationary iteration, but found choice of damping parameter to be much more critical here; the value needed to be tuned by hand, which is

not desirable. We therefore use the non-linear iterative method GMRES [63] to evaluate the system (4.6), together with a preconditioner

$$W=\begin{bmatrix} \widehat{M+\tau K} & 0 \\ N^T & -(\widehat{M_{u,b}+K_{u,b}}) \end{bmatrix}. \tag{4.7}$$

Here $\widehat{\cdots}$ again represents that the action of the inverse of these blocks is given by a fixed number of steps of an algebraic multigrid method. Note that, because of the use of GMRES as an inner iteration, the preconditioner $\mathcal{P}_2$ is nonlinear, and theory dictates that we should use a flexible outer method such as FGMRES [62]. By using a rather small tolerance to stop GMRES we seem to avoid convergence difficulties, allowing us to use a standard Krylov method; see Section 6 for details. An alternative would be to use a sparse direct method [22, 42] to solve for the sub-problem (4.6), giving us a hybrid solution method.

State constraints

The situation is not much different in the case when state constraints are present. Here the system matrix is

$$\begin{bmatrix} \tau\mathcal{M}_\varepsilon & 0 & -\mathcal{K}^T \\ 0 & \tau\beta(\mathcal{M}_u+\mathcal{K}_u) & \tau\mathcal{M} \\ -\mathcal{K} & \tau\mathcal{M} & 0 \end{bmatrix}, \tag{4.8}$$

where each of the blocks of $\mathcal{M}_\varepsilon$ is now given by $\mathcal{M}_y+\varepsilon^{-1}G_{\mathcal{A}_i}M_yG_{\mathcal{A}_i}$. The Schur-complement is now

$$S=\tau^{-1}\mathcal{K}\mathcal{M}_\varepsilon^{-1}\mathcal{K}^T+\frac{\tau}{\beta}\mathcal{M}(\mathcal{M}_u+\mathcal{K}_u)^{-1}\mathcal{M}.$$

We can now proceed as n the absence of state constraints. An approximation of S is chosen to be

$$\widehat{S}=\tau^{-1}\left(\mathcal{K}+\frac{\tau}{\sqrt{\beta}}\mathcal{M}(\mathcal{M}_u+\mathcal{K}_u)^{-1}\mathcal{M}_\varepsilon^{1/2}\right)\mathcal{M}_\varepsilon^{-1}\left(\mathcal{K}^T+\frac{\tau}{\sqrt{\beta}}\mathcal{M}_\varepsilon^{1/2}\mathcal{M}\right).$$

We use the symmetric matrix $\mathcal{M}_\varepsilon^{1/2}$ because this makes all factors of the approximation $\hat{S}$ dependent on ε. A solve with $\widehat{S}$ is approximated as before, where the diagonal blocks of $\left(\mathcal{K}^T+\frac{\tau}{\sqrt{\beta}}\mathcal{M}_\varepsilon^{1/2}\mathcal{M}\right)^{-1}$ are approximated by an algebraic multigrid technique. Note that due to the matrix $\mathcal{M}_\varepsilon$ the diagonal blocks of this matrix are different at each outer iteration, and we need to recompute the algebraic multigrid approximation; update techniques to exploiting this structure should be investigated in the future to streamline the solver.

The term $\left(\mathcal{K}+\frac{\tau}{\sqrt{\beta}}\mathcal{M}(\mathcal{M}_u+\mathcal{K}_u)^{-1}\mathcal{M}_\varepsilon^{1/2}\right)$ is again harder to deal with and as previously we use an auxiliary system

$$\begin{bmatrix} \mathcal{K} & \mathcal{M} \\ \mathcal{M}_\varepsilon^{1/2} & \frac{-\sqrt{\beta}}{\tau}(\mathcal{M}_u+\mathcal{K}_u) \end{bmatrix},$$

which we can permute to be of block-triangular form. We are then left with approximately solving a system for

$$\begin{bmatrix} M+\tau K & M \\ M_{\varepsilon,i}^{1/2} & \frac{-\sqrt{\beta}}{\tau}(M_u+K_u) \end{bmatrix},$$

where i indicates the i-th block corresponding to the i-th point in time and its corresponding structure coming from the active set. Our strategy is again to use an accurate solution via a preconditioner GMRES method employing the preconditioner

$$W_i = \begin{bmatrix} \widehat{[M+\tau K]} & 0 \\ M_{\varepsilon,i}^{1/2} & -\widehat{[M_u+K_u]} \end{bmatrix}.$$

Here $\widehat{[\cdots]}$ indicates the use of an algebraic multigrid in the inversion of this matrix.

5 Eigenvalue analysis

Our goal here is to analyze the quality of the preconditioners proposed earlier. As described in the previous section, the approximation of the $(1,1)$-block is relatively straightforward using standard tools, such as multigrid, which are well understood. We therefore focus solely on the quality of the Schur-complement approximation.

We use the methodology introduced by Pearson and Wathen [54] for the stationary case that was later generalized for the time-dependent case (see [53]). There a symmetric Schur-complement approximation was chosen, and the quality of the approximation was measured by bounding the eigenvalues of $\hat{S}^{-1}S$ via the Rayleigh quotient

$$R := \frac{v^T S v}{v^T \hat{S}_1 v}.$$

We here briefly illustrate their argument in order to assess what parts carry over here. One can write

$$\frac{v^T S v}{v^T \hat{S}_1 v} = \frac{a^T a + b^T b}{a^T a + b^T b + b^T a + a^T b} \tag{5.1}$$

with suitably chosen vectors a and b. It is easy to see from

$$0 \leq (a-b)^T (a-b) = a^T a + b^T b - a^T b - b^T a \tag{5.2}$$

that $R \geq \frac{1}{2}$. Pearson and co-authors then proceeded by showing that $a^T b + b^T a$ is positive to conclude that the R is bounded by 1 from above. For the distributed control case this is both true in the steady [54] and transient [53] case.

Our goal is to carry this analysis over to our setup. We focus on the distributed control case here, where

$$a := \tau^{-1/2} \mathcal{M}_y^{-1/2} \mathcal{K}^T v, \tag{5.3}$$

$$b := \tau^{1/2} \beta^{-1/2} (\mathcal{M}_u + \mathcal{K}_u)^{-1/2} \mathcal{M}^T v. \tag{5.4}$$

Using

$$v^T S v = a^T a + b^T b$$

as well as (5.2) we can see that the lower bound for the approximation $\hat{S}_1$ is still valid, i.e., $R \geq \frac{1}{2}$ regardless of the mesh-parameter and the regularization parameter.

The interesting question from now on is whether the upper bound $R \leq 1$ is still valid. For this to be true we immediately see from (5.1) that $a^T b + b^T a$ needs to be positive. We proceed by considering the simpler time-independent case for which the matrix structure is very similar. In that case we obtain

$$a^T b + b^T a = \beta^{-1/2} v^T \left(K M_y^{-1/2} (M_u + K_u)^{-1/2} M^T + M (M_u + K_u)^{-1/2} M_y^{-1/2} K \right) v.$$

To see that in the L_2 norm case this was positive we set $K_u = 0$ and $M_u = M_y = M$ and obtain

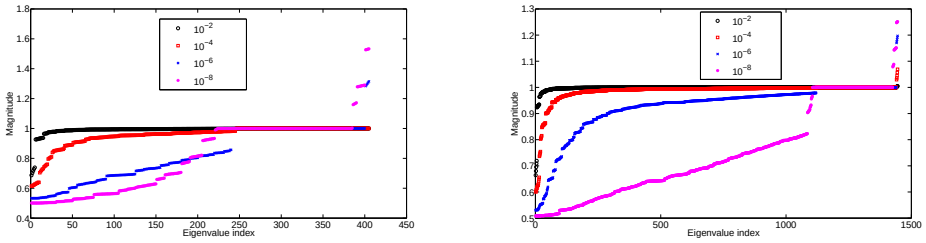
$$\beta^{-1/2} v^T \left(K + K^T \right) v$$

which is obviously positive. The same is true for the time-dependent problem without $\mathcal{H}_1$-norm (see [53] for a proof). Unfortunately, once the $\mathcal{H}_1$-norm is considered the positivity of $a^T b + b^T a$ is lost.

So why does the $\mathcal{H}_1$-norm cause a problem as its discretization only introduces a symmetric and positive definite matrix $M_u + K_u$? This is clear from the fact that in general $WV + VW \not> 0$ even when both V and W are symmetric and positive definite matrices. A simple example is given when W and V correspond to a Dirichlet and Neumann Laplacian, respectively.

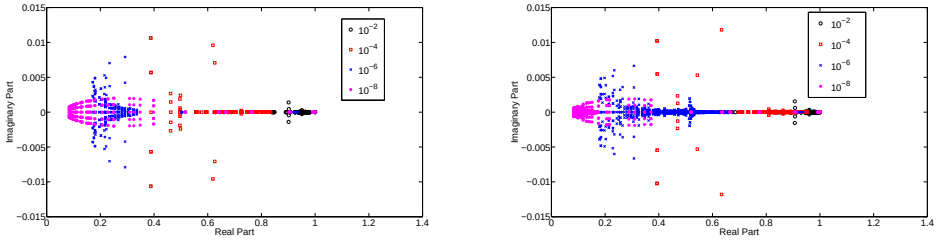
Note that in our case $a^T b + b^T a$ is of precisely this form and the computation of the eigenvalues of $\left(K M_y^{-1/2} (M_u + K_u)^{-1/2} M^T + M (M_u + K_u)^{-1/2} M_y^{-1/2} K \right)$ reveals several negative eigenvalues.

Nevertheless, the spread of the eigenvalues above the desired value of 1 with varying β and mesh-parameter is not severe as illustrated by the eigenvalues shown in Fig. 1. Fig. 1(a) shows the eigenvalues of $\hat{S}_1$ for a coarse mesh and four values of the regularization parameter β and Fig. 1(b) shows the eigenvalues for the same values of β but on a finer mesh. From these pictures we can see that the magnitude of the eigenvalues does not increase for the finer mesh and that most of the eigenvalues are contained in the interval $\left[\frac{1}{2}, 1\right]$ with some outliers that do not move much beyond 1 when the regularization parameter is decreased. The major disadvantage of the approximation $\hat{S}_1$ is the use of the matrix square roots, which is infeasible for large systems. We hence move to the



(a) Coarse (b) Finer

Figure 1: Eigenvalues for two different meshes and a variety of regularization parameters. We show coarse mesh on the left and slightly finer mesh on the right. We use $N_T = 5$.



(a) Coarse (b) Finer

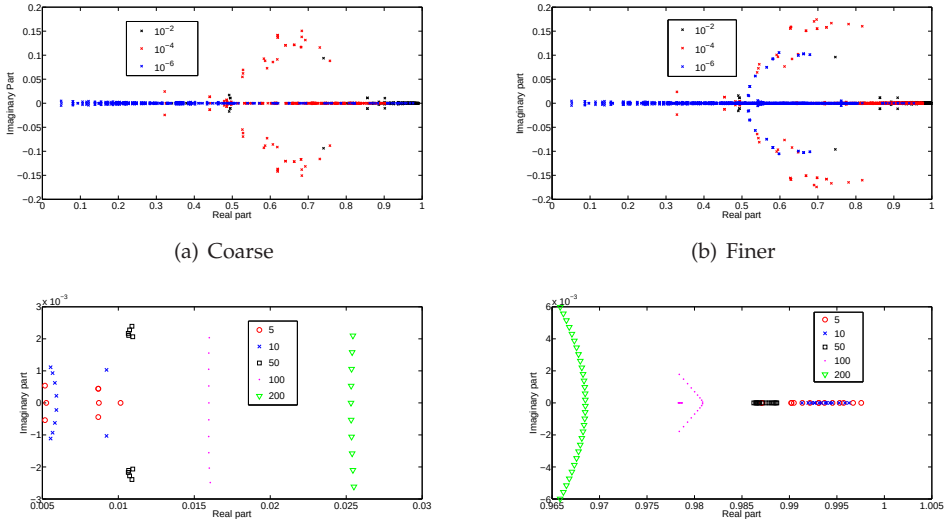
Figure 2: Eigenvalues for two different meshes and a variety of regularization parameters. We show coarse mesh on the left and slightly finer mesh on the right. We use $N_T = 5$.

nonsymmetric approximation $\hat{S}_2$ for which the above used Rayleigh quotient analysis is unfortunately not applicable.

Nevertheless, we expect the eigenvalues of the pencil $(S, \hat{S}_2)$ to provide guidance on the speed of convergence of our iterative scheme. We here want to numerically study the eigenvalues of $\hat{S}_2^{-1}S$ to obtain information that can allow us to understand the convergence of a nonsymmetric solver using the nonsymmetric Schur-complement approximation $\hat{S}_2$.

Fig. 2 shows eigenvalue distributions of $\hat{S}_2^{-1}S$ for two different mesh-sizes and a variety of regularization parameters. The comparison of both plots 2(a) for the coarse mesh and 2(b) for the refined one indicates that for very small values of β the eigenvalues move closer towards the origin but stay sufficiently far away from zero. Additionally, this behaviour does not change when the mesh is refined so we expect robust iteration numbers with respect to a refinement in space. We computed approximations to the eigenvalues closest to the origin of $\hat{S}_2^{-1}S$ for one further mesh and found these to be in the same region as the smallest eigenvalues shown in Fig. 2.

Our numerical results given in Section 6 indicate that this choice of Schur complement approximation allows for good convergence with relatively robust iteration numbers.



(a) Coarse (b) Finer

(c) Leftmost eigenvalues (d) Rightmost eigenvalues

Figure 3: Eigenvalues for two different meshes and a variety of regularization parameters. We show a coarse mesh on the left and slightly finer mesh on the right. The top row shows the eigenvalues for $\hat{S}_2^{-1}S$ for $N_T=5$. The lower figures illustrate the dependency of the smallest and largest eigenvalues of $\hat{S}_2^{-1}S$ on the number of time-steps and hence τ.

6 Numerical results

We now want to illustrate how the preconditioners presented above perform when applied to a variety of problems. As mentioned earlier we employ a finite element discretization, here done with the finite element package deal.II [2]. We discretize the state, control and adjoint state variables using **Q1** elements. For symmetric methods the stopping criterion is often inherent to the problem [74]. In the nonsymmetric context the debate is much more open an we decide to use the relative residual with $x_0=0$ based on the discussion in [3]. Hence, we present results for both a tolerance of 10^{-4} and a tighter tolerance of 10^{-6} for the relative residual within BICG using the preconditioner $\mathcal{P}_2$. For the algebraic multigrid preconditioner we use the Trilinos ML package [27] that implements a smoothed aggregation AMG. Within the algebraic multigrid we used 6 steps of a Chebyshev smoother in combination with the application of two V-cycles. For time-dependent problems we show the degrees of freedom only for one grid point in time (i.e. for a single time-step) and we are implicitly solving a linear system of dimension 3 times the number of time-steps (N_t) times the degrees of freedom of the spatial discretization (n). For example, a spatial discretization with 274625 spatial unknowns and 20 time-steps corresponds to an overall linear system of dimension 16 477 500.

6.1 Distributed control

No state constraints

In this section we show results for the time-dependent case. First, we consider the case when no state constraints are present. Here, we work with a fixed time-step $\tau = 0.05$, which results in 20 time-steps. In all tables we only show the degrees of freedom associated with the discretization of the spatial domain. The desired state is now given by

$$\bar{\mathbf{y}} = \exp\left(-64\left((x_0-0.5)^2+(x_1-0.5)^2\right)\right)$$

and $\mathbf{y} = \bar{\mathbf{y}}$ on $\partial\Omega$, where the domain is $[0,1]^2$. The results for this setup are shown in Table 1 for various mesh-parameters and values of the regularization parameter β.

Table 1: Results for the distributed control problem and varying mesh and regularization parameter. This table shows iteration numbers and timings for BICG with a nonsymmetric Schur complement approximation using 10 Uzawa steps and a damping parameter $\omega = 0.1$. The tolerance of the iterative solver is set to 10^{-6}.

DoF	$\beta = 10^{-2}$	$\beta = 10^{-4}$	$\beta = 10^{-6}$
	# it(t)	# it(t)	# it(t)
1089	15(40.1)	17(45.9)	28(72.9)
4225	15(129.2)	18(153.5)	29(242.1)
16641	18(554.2)	22(669.7)	31(932.2)
66049	19(1627.6)	27(2280.1)	36(2995.4)
263169	23(5922.8)	28(7203.9)	44(11389.2)

Table 2: Results for the distributed control problem and varying mesh and regularization parameter. This table shows iteration numbers and timings for BICG with a nonsymmetric Schur complement approximation using 10 Uzawa steps and a damping parameter $\omega = 0.1$. The tolerance of the iterative solver is set to 10^{-6}.

DoF	$\beta = 10^{-2}$	$\beta = 10^{-4}$	$\beta = 10^{-6}$
	# it(t)	# it(t)	# it(t)
1089	13(35.1)	13(35.2)	22(57.3)
4225	13(112.6)	15(128.8)	22(184.8)
16641	15(462.3)	15(462.2)	25(756.1)
66049	17(1442.6)	20(1691.4)	31(2578.7)
263169	19(4928.3)	22(5843.9)	34(8368.3)

State constraints

We now consider the problem with state constraints. The defining parameters are given by the desired state

$$\bar{\mathbf{y}} = -tx_0\exp\left(-\left((x_0-0.5)^2+(x_1-0.5)^2\right)\right)$$

Table 3: Results for the distributed control problem and varying mesh and regularization parameter. This table shows iteration numbers and timings for BICG with a nonsymmetric Schur complement approximation using 10 Uzawa steps and a damping parameter $\omega=0.1$. The tolerance of the iterative solver is set to 10^{-6}.

DoF	$\beta=10^{-2}$	$\beta=10^{-4}$	$\beta=10^{-6}$
	# it(t)	# it(t)	# it(t)
1089	13(35.1)	13(35.2)	22(57.3)
4225	13(112.6)	15(128.8)	22(184.8)
16641	15(462.3)	15(462.2)	25(756.1)
66049	17(1442.6)	20(1691.4)	31(2578.7)
263169	19(4928.3)	22(5843.9)	34(8368.3)

Table 4: Results for the state-constrained problem. We here vary the penalization parameter ε. Shown are the Newton iteration numbers for a Newton tolerance of 10^{-3} for the first two columns and a tolerance for 10^{-2} for the case $\varepsilon=10^{-4}$. As the number of Newton iterations increased we here only show iteration numbers for a stopping tolerance of 10^{-2} for the outer iteration. Further we give the average number of BICG iterations and the maximal number of GMRESiterations needed for the evaluation of the preconditioner. The tolerance of the iterative solver is set to 10^{-6}.

DoF	$\varepsilon=10^{0}$	$\varepsilon=10^{-2}$	$\varepsilon=10^{-4}$
	AS/BICG/GMRES	AS/BICG/GMRES	AS/BICG/GMRES
81	3/23.7/12	7/21.3/19	6/25.8/41
289	3/32.7/16	7/26.9/23	6/35.2/52
1089	3/51.3/19	6/37.0/27	2/45.5/75
4225	3/74.0/23	6/53.3/43	2/56.5/109

with zero initial and boundary condition. We then consider a fixed regularization parameter $\beta=10^{-4}$, which then allows us to consider the lower bound $-0.1\leq\mathbf{y}$ for all time-steps. The results are shown in Table 4, where we vary the penalization parameter from 1 to 10^{-4}. The iteration numbers obtained show a small increase with respect to the mesh-size. This might be due to the approximation quality of the diagonal blocks used within the evaluation of the preconditioner W. We observed that we needed to increase the number of V-cycles within the AMG method to 8 to obtain a robust performance. Future research should be devoted to obtaining preconditioners that allow updating to deal with the changing blocks involving components from the active sets and also show more robustness with respect to parameter-dependent matrices (here in particular β and ε).

6.2 Boundary control

We now show results for the boundary control case where the desired state is given by

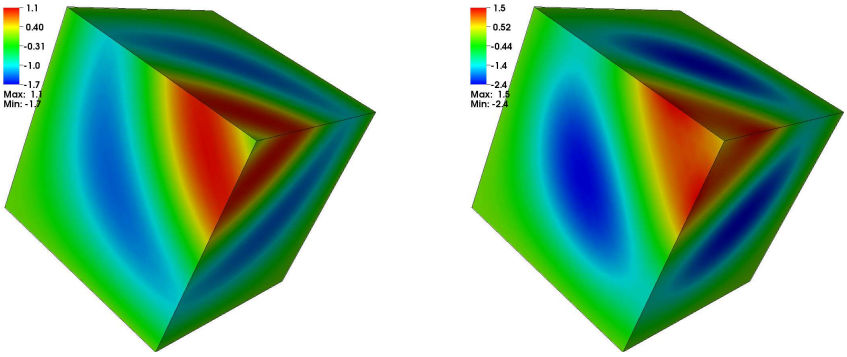
$$\bar{\mathbf{y}}=-\exp(t)\sin(2\pi x_0x_1x_2)\exp\left(-\left((x_0-0.5)^2+(x_1-0.5)^2+(x_2-0.5)^2\right)\right)$$

on the three-dimensional domain $\Omega=[0,1]^3$. The results with the Schur complement approximation $\hat{S}$ with varying mesh-size and regularization parameter β are shown in

Table 5: Results for the boundary control problem and varying mesh and regularization parameter. This table shows iteration numbers for BICG and the maximal number of GMRES iterations used for the preconditioner. The tolerance of the iterative solver is set to 10^{-6}.

DoF	$\beta=10^{-2}$ BICG/GMRES	$\beta=10^{-4}$ BICG/GMRES
729	18(21)	19(38)
4913	18(22)	17(40)
35937	19(24)	17(44)
274625	19(25)	19(47)

Table 5. We again want to emphasize that we use preconditioned GMRES to evaluate the diagonal-blocks of the Schur-complement approximation. The tolerance is set rather tight on the one hand to guarantee that as an outer iteration BICG is still suited and on the other hand to guarantee that we obtain robustness with respect to parameter changes. We additionally state for every problem the maximal number of iterations that was needed for GMRES. It can be seen that the number of BICGiterations are robust with respect to parameter changes. The number of iterations for the GMRES preconditioner increases slightly with a decrease of the regularization parameter.



(a) Desired state $\bar{\mathbf{y}}_{12}$ (b) Computed state $\mathbf{y}_{12}$

Figure 4: Desired state and computed state for boundary control problem. Here the regularization parameter was set to $\beta=10^{-6}$.

7 Conclusions and outlook

In this paper we presented optimal control problems subject to the Poisson equation or the heat equation in a distributed or boundary control setting. The control was added to the objective function as a regularization term in the $\mathcal{H}_1$ norm. We introduced the corre-

sponding discrete optimality system and introduced preconditioners for both the steady as well as the transient problem. Due to the Laplacian term coming from the $\mathcal{H}_1$ norm we were not able to introduce preconditioners that are fully independent of the regularization parameter but for the simple preconditioners we introduced the dependence on the regularization parameter seemed rather weak. We also showed that our approach works for state-constrained problems, which were treated using a Moreau-Yosida penalty approach. Numerical results showed that our preconditioners provided satisfactory results when applied to three-dimensional test problems.

The method presented here has not focused on the storage efficiency of our all-at-once approach. One might employ checkpointing [30] techniques when alternately solving forward and adjoint PDEs. Multiple shooting approaches are one way of splitting up the time-interval [33] and can lead to the same type of system. A possible way forward is to compute suboptimal solutions on a sequential splitting of the time-interval [33] or to use a parallel implementation of our approach. It is also possible to reduce the storage requirements by performing block-eliminations of some form, usually via a Schur-complement approach.

References

[1] Bai, Zhong-Zhi: Block preconditioners for elliptic PDE-constrained optimization problems. Computing **91**(4), 379–395 (2011)
[2] Bangerth, W., Hartmann, R., Kanschat, G.: deal.II—a general-purpose object-oriented finite element library. ACM Trans. Math. Software **33**(4), Art. 24, 27 (2007)
[3] Barrett, R., Berry, M., Chan,T. F., Demmel, J., Donato,J., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C., and der Vorst, H. V.: Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods. 2nd Edition, SIAM, Philadelphia, PA, 1994.
[4] Benzi, M., Golub, G.H., Liesen, J.: Numerical solution of saddle point problems. Acta Numer **14**, 1–137 (2005)
[5] Benzi, M., Haber, E., Taralli, L.: A preconditioning technique for a class of PDE-constrained optimization problems. Advances in Computational Mathematics **35**, 149–173 (2011)
[6] Bergounioux, M., Ito, K., Kunisch, K.: Primal-dual strategy for constrained optimal control problems. SIAM J. Control Optim. **37**(4), 1176–1194 (1999)
[7] Bergounioux, M., Kunisch, K.:Primal-dual strategy for state-constrained optimal control problems, Comput. Optim. Appl. **22**(2), 193–224 (2002) DOI 10.1023/A:1015489608037
[8] Bochev, P., Lehoucq, R.: On the finite element solution of the pure neumann problem. SIAM review **47**(1), 50–66 (2005)
[9] Bramble, J.H., Pasciak, J.E.: A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems. Math. Comp **50**(181), 1–17 (1988)
[10] Cai, Xiao-Chuan and Liu, Si and Zou, Jun: An overlapping domain decomposition method for parameter identification problems. Domain Decomposition Methods in Science and Engineering XVII, **60**, 451–458 (2008)
[11] Casas, E.: Control of an elliptic problem with pointwise state constraints. SIAM J. Control Optim. **24**(6), 1309–1318 (1986). DOI 10.1137/0324078. URL `http://dx.doi.org/10.1137/0324078`

[12] Casas, E., Herzog, R. and Wachsmuth, G.: Approximation of sparse controls in semilinear equations by piecewise linear functions. Numerische Mathematik **122**, 645–669 (2012)
[13] Chan, R.H. and Chan, T.F. and Wan, WL and others: Multigrid for differential-convolution problems arising from image processing. Proc. Workshop on Scientific Computing, pp. 58–72 (1997)
[14] Chan, T.F. and Tai, X.C.: Identification of discontinuous coefficients in elliptic problems using total variation regularization. SIAM Journal on Scientific Computing **25**(3) 881–904 (2003)
[15] Choi, Y., Farhat, C., Murray, W. and Saunders, M.: A practical factorization of a Schur complement for PDE-constrained Distributed Optimal Control. arXiv preprint arXiv:1312.5653 (2013)
[16] Christofides, P.: Nonlinear and robust control of PDE systems: Methods and applications to transport-reaction processes. Birkhauser (2001)
[17] Cimrák, I. and Melicher, V.: Mixed Tikhonov regularization in Banach spaces based on domain decomposition submitted to Applied Mathematics and Computation (2012)
[18] Collis, S.S., Ghayour, K., Heinkenschloss, M., Ulbrich, M. and Ulbrich, S.: Numerical solution of optimal control problems governed by the compressible Navier-Stokes equations International series of numerical mathematics, 43–56 (2002)
[19] van den Doel, K. and Ascher, U. and Haber, E.: The lost honour of l2-based regularization Submitted (2012)
[20] Du, X., Sarkis, M., Schaerer, C.E., Szyld, D.B.: Inexact and truncated parareal-in-time Krylov subspace methods for parabolic optimal control problems. Tech. Rep. 12-02-06, Department of Mathematics, Temple University (2012)
[21] Duff, I.S., Erisman, A.M., Reid, J.K.: Direct methods for sparse matrices. Monographs on Numerical Analysis. The Clarendon Press Oxford University Press, New York (1989)
[22] Duff, Iain S.: MA57—a code for the solution of sparse symmetric definite and indefinite systems. ACM Transactions on Mathematical Software (TOMS) **30**(2) 118-144 (2004)
[23] Elman, H.C., Silvester, D.J., Wathen, A.J.: Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics. Numerical Mathematics and Scientific Computation. Oxford University Press, New York (2005)
[24] Falgout, R.: An Introduction to Algebraic Multigrid. Computing in Science and Engineering, 8 (2006), pp. 24–33. Special Issue on Multigrid Computing.
[25] Fletcher, R.: Conjugate gradient methods for indefinite systems, *Lecture Notes in Mathematics*, vol. 506. Springer–Verlag, Berlin (1976), 73–89.
[26] Freund, R.W., Nachtigal, N.M.: QMR: a quasi-minimal residual method for non-Hermitian linear systems. Num. Math. **60**(1) 315–339 (1991)
[27] Gee, M., Siefert, C., Hu, J., Tuminaro, R., Sala, M.: ML 5.0 smoothed aggregation user's guide. Tech. Rep. SAND2006-2649, Sandia National Laboratories (2006)
[28] Gunzburger, Max D.: Perspectives in flow control and optimization, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA (2003)
[29] Greenbaum, A.: Iterative methods for solving linear systems, *Frontiers in Applied Mathematics*, vol. 17. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA (1997)
[30] Griewank, A., Walther, A.: Algorithm 799: revolve: an implementation of checkpointing for the reverse or adjoint mode of computational differentiation. ACM Transactions on Mathematical Software (TOMS) **26**(1), 19–45 (2000)
[31] Haber, E. and Hanson, L.: Model Problems in PDE-Constrained Optimization Emory University TR-2007-009 (2007)

[32] Hackbusch,W.:Multigrid methods and applications. vol. 4 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 1985.
[33] Heinkenschloss, M.: A time-domain decomposition iterative method for the solution of distributed linear quadratic optimal control problems. J. Comput. Appl. Math. **173**(1), 169–198 (2005). DOI 10.1016/j.cam.2004.03.005
[34] Heinkenschloss, M.: Formulation and analysis of a sequential quadratic programming method for the optimal Dirichlet boundary control of Navier-Stokes flow Optimal Control, Theory, Algorithms, and Applications (1998)
[35] Heinkenschloss, Matthias, Denis Ridzal: A matrix-free trust-region SQP method for equality constrained optimization. Technical Report 11-17, Department of Computational and Applied Mathematics, Rice University, Houston, Texas, 2011.
[36] Herzog, R., Sachs, E.W.: Preconditioned conjugate gradient method for optimal control problems with control and state constraints. SIAM J. Matrix Anal. Appl. **31**(5), 2291–2317 (2010)
[37] Hestenes, M.R., Stiefel, E.: Methods of conjugate gradients for solving linear systems. J. Res. Nat. Bur. Stand **49**, 409–436 (1953) (1952)
[38] Hintermüller, M., Ito, K., Kunisch, K.: The primal-dual active set strategy as a semismooth Newton method. SIAM J. Optim. **13**(3), 865–888 (2002)
[39] Hintermüller, M., Kunisch, K.: Path-following methods for a class of constrained minimization problems in function space. SIAM J. Optim. **17**(1), 159–187 (2006)
[40] Hinze, M., Köster, M., Turek, S.: A Hierarchical Space-Time Solver for Distributed Control of the Stokes Equation. Tech. rep., SPP1253-16-01 (2008)
[41] Hinze, M., Pinnau, R., Ulbrich, M., Ulbrich, S.: Optimization with PDE Constraints. Mathematical Modelling: Theory and Applications. Springer-Verlag, New York (2009)
[42] Hogg, Jonathan D., and Jennifer A. Scott. HSL_MA97: A bit-compatible multifrontal code for sparse symmetric systems. Science and Technology Facilities Council, 2011.
[43] Ito, K., Kunisch, K.: Semi-smooth Newton methods for state-constrained optimal control problems. Systems Control Lett. **50**(3), 221–228 (2003). DOI 10.1016/S0167-6911(03)00156-7
[44] Ito, K., Kunisch, K.: Lagrange multiplier approach to variational problems and applications, *Advances in Design and Control*, vol. 15. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA (2008)
[45] Kanzow, C.: Inexact semismooth Newton methods for large-scale complementarity problems. Optimization Methods and Software **19**(3-4), 309–325 (2004).
DOI 10.1080/10556780310001636369.
URL `http://www.tandfonline.com/doi/abs/10.1080/10556780310001636369`
[46] Keung, Y.L. and Zou, J.: Numerical identifications of parameters in parabolic systems Inverse Problems **14**(1), 83–100 (1999)
[47] Kollmann, M., Kolmbauer, M.: A Preconditioned MinRes Solver for Time-Periodic Parabolic Optimal Control Problems. Sumitted,Numa-Report 2011-06 (August 2011)
[48] Li, F. and Shen, C. and Li, C.: Multiphase Soft Segmentation with Total Variation and H_1 Regularization Journal of Mathematical Imaging and Vision **37**(2), 98–111 (2010)
[49] Ng, M.K. and Chan, R.H. and Chan, T.F. and Yip, A.M.: Cosine transform preconditioners for high resolution image reconstruction Linear Algebra and its Applications **316**(1) 89–104 (2000)
[50] Murphy, M.F., Golub, G.H., Wathen, A.J.: A note on preconditioning for indefinite linear systems. SIAM J. Sci. Comput **21**(6), 1969–1972 (2000)
[51] Paige, C.C., Saunders, M.A.: Solutions of sparse indefinite systems of linear equations. SIAM

J. Numer. Anal **12**(4), 617–629 (1975)
[52] Pearson, J.W., Stoll, M., Wathen, A.: Preconditioners for state constrained optimal control problems with Moreau-Yosida penalty function. Numerical Linear Algebra with Applications **21**(1), 81-97, (2014)
[53] Pearson, J.W., Stoll, M., Wathen, A.J.: Regularization-robust preconditioners for time-dependent PDE-constrained optimization problems. SIAM J. Matrix Anal.Appl **33**, 1126–1152 (2012)
[54] Pearson, J.W., Wathen, A.J.: A new approximation of the Schur complement in preconditioners for PDE-constrained optimization. Numerical Linear Algebra with Applications **19**, 816–829 (2012). DOI 10.1002/nla.814. URL `http://dx.doi.org/10.1002/nla.814`
[55] Peirce, A., Dahleh, M., Rabitz, H.: Optimal control of quantum-mechanical systems: Existence, numerical approximation, and applications. Physical Review A **37**(12), 4950 (1988)
[56] Pironneau, O.: Optimal shape design for elliptic systems. System Modeling and Optimization pp. 42–66 (1982)
[57] Rees, T., Dollar, H.S., Wathen, A.J.: Optimal solvers for PDE-constrained optimization. SIAM Journal on Scientific Computing **32**(1), 271–298 (2010). DOI http://dx.doi.org/10.1137/080727154
[58] Rees, T., Stoll, M., Wathen, A.: All-at-once preconditioners for PDE-constrained optimization. Kybernetika **46**, 341–360 (2010)
[59] De los Reyes, Juan-Carlos and Schönlieb, Carola-Bibiane: Image denoising: learning noise distribution via PDE-constrained optimization, http://arxiv.org/abs/1207.3425 (2012)
[60] Ruge, J. W. and Stüben, K.: Algebraic multigrid. in Multigrid methods, vol. 3 of Frontiers Appl. Math., SIAM, Philadelphia, PA, 1987, pp. 73–130.
[61] Saad, Y.: Iterative methods for sparse linear systems. Society for Industrial and Applied Mathematics, Philadelphia, PA (2003)
[62] Saad, Y.: A flexible inner-outer preconditioned GMRES algorithm. SIAM Journal on Scientific Computing, 14 (1993), pp. 461–461.
[63] Saad, Y., Schultz, M.H.: GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Stat. Comput., **7**(3), 856–869 (1986).
[64] Simoncini, V., Szyld, D.: Recent computational developments in Krylov subspace methods for linear systems. Numer. Linear Algebra Appl **14**(1), 1–61 (2007).
[65] Stoll, M.: All-at-once solution of a time-dependent time-periodic PDE-constrained optimization problems. IMA J Numer Anal (2013)
[66] Stoll, M., Wathen, A.: All-at-once solution of time-dependent PDE-constrained optimization problems. Technical Report, University of Oxford, (2010)
[67] Strang, G., Fix, G.: An Analysis of the Finite Element Method 2nd Edition, 2nd edn. Wellesley-Cambridge (2008)
[68] Takacs, S., Zulehner, W.: Convergence analysis of multigrid methods with collective point smoothers for optimal control problems. Computing and Visualization in Science **14**, 131–141 (2011)
[69] Tröltzsch, F.: Optimal Control of Partial Differential Equations: Theory, Methods and Applications. Amer Mathematical Society (2010)
[70] Ulbrich, M.: Semismooth Newton Methods for Variational Inequalities and Constrained Optimization Problems. SIAM Philadelphia (2011)
[71] Van Der Vorst, H.A.: BI–CGSTAB: A fast and smoothly converging variant of BI–CG for the solution of nonsymmetric linear systems. SIAM J. Sci. Stat. Comput. **13**(2), 631–634 (1992).
[72] Wathen, A.J., Rees, T.: Chebyshev semi-iteration in preconditioning for problems including

the mass matrix. Electronic Transactions in Numerical Analysis **34**, 125–135 (2008)
[73] Wachsmuth, D. and Wachsmuth, G.: Necessary conditions for convergence rates of regularizations of optimal control problems, RICAM Report **4** (2012)
[74] Wathen, A. J.: Preconditioning and convergence in the right norm. International Journal of Computer Mathematics, 84 (2007), pp. 1199–1209.
[75] Wesseling,P. : An introduction to multigrid methods. Pure and Applied Mathematics (New York), John Wiley & Sons Ltd., Chichester, 1992.
[76] Wilson, J. and Patwari, N. and Vasquez, F.G.: Regularization methods for radio tomographic imaging. 2009 Virginia Tech Symposium on Wireless Personal Communications (2009)
[77] Zulehner, W.: Analysis of iterative methods for saddle point problems: a unified approach. Math. Comp **71**(238), 479–505 (2002)

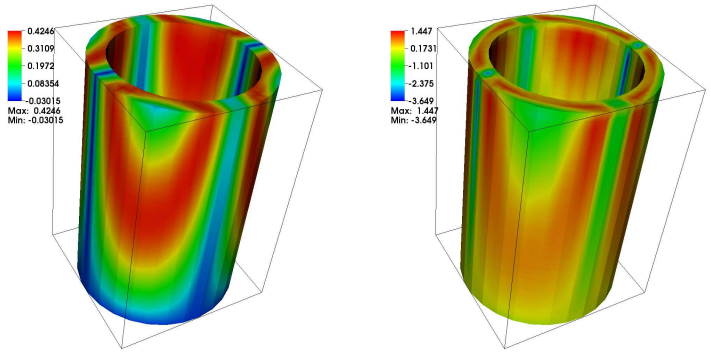
A.7 Preconditioning for reaction-diffusion problems

This paper is published as

J. W. Pearson and M. Stoll, ***Fast Iterative Solution of Reaction-Diffusion Control Problems Arising from Chemical Processes***, **SIAM J. Sci. Comput., 35 (2013), pp. 987–1009.**

Result from the paper

In this paper we consider the numerical solution of a nonlinear reaction-diffusion model where we construct efficient preconditioners. Figure A.3 shows the three-dimensional results for a computed state and control.



(a) Computed state for first reactant at time step 7

(b) Computed control at time step 7

Figure A.3

SIAM J. Sci. Comput.
Vol. 35, No. 5, pp. B987–B1009

FAST ITERATIVE SOLUTION OF REACTION-DIFFUSION CONTROL PROBLEMS ARISING FROM CHEMICAL PROCESSES*

JOHN W. PEARSON† AND MARTIN STOLL‡

Abstract. PDE-constrained optimization problems, and the development of preconditioned iterative methods for the efficient solution of the arising matrix systems, is a field of numerical analysis that has recently been attracting much attention. In this paper, we analyze and develop preconditioners for matrix systems that arise from the optimal control of reaction-diffusion equations, which themselves result from chemical processes. Important aspects of our solvers are saddle point theory, mass matrix representation, and effective Schur complement approximation, as well as the incorporation of control constraints and application of the outer (Newton) iteration to take into account the nonlinearity of the underlying PDEs.

Key words. PDE-constrained optimization, reaction-diffusion, chemical processes, Newton iteration, preconditioning, Schur complement

AMS subject classifications. 65F08, 65F10, 65F50, 92E20, 93C20

DOI. 10.1137/120892003

1. Introduction.

A class of problems which has numerous applications within mathematical and physical problems is that of PDE-constrained optimization problems. One field in which these problems can be posed is that of chemical processes [4, 19, 20, 21, 22]. In this case the underlying PDEs are reaction-diffusion equations, and therefore the PDE constraints in our formulation are nonlinear PDEs.

When solving such reaction-diffusion control problems using a finite element method, and employing a Lagrange–Newton iteration to take account of the nonlinearity involved in the PDEs, the resulting matrix system for each Newton iteration will be large, sparse, and of saddle point structure. It is therefore desirable to devise preconditioned iterative methods to solve these systems efficiently and in such a way that the structure of the matrix is exploited. Work in constructing preconditioners for PDE-constrained optimization problems has been considered for simpler problems previously, for instance, Poisson control [46, 47, 53], convection-diffusion control [45], Stokes control [36, 50, 56], and heat equation control [44, 54].

In this paper, we will consider an optimal control formulation of a reaction-diffusion problem, which generates a symmetric matrix system upon each Newton iteration. (Such an iteration is required to take into account the nonlinear terms within the underlying PDEs.) We will generally search for block triangular preconditioners for the matrix systems we examine, to be used in conjunction with a suitable iterative solver. In order to do this, we will need to approximate the $(1,1)$-block by accurately representing the inverse of mass matrices amongst other things, as well as

*Submitted to the journal's Computational Methods in Science and Engineering section September 20, 2012; accepted for publication (in revised form) June 26, 2013; published electronically September 10, 2013. This work was supported by the award of a European Science Foundation (ESF) Exchange Grant under the OPTPDE program.
http://www.siam.org/journals/sisc/35-5/89200.html
†Numerical Analysis Group, Mathematical Institute, University of Oxford, 24–29 St Giles', Oxford, OX1 3LB, UK (john.pearson@worc.ox.ac.uk). This author's work was supported by the Engineering and Physical Sciences Research Council (UK), grant EP/P505216/1.
‡Computational Methods in Systems and Control Theory, Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, 39106 Magdeburg, Germany (stollm@mpi-magdeburg.mpg.de).

devise an effective approximation of the Schur complement of the matrix system. We demonstrate with numerical tests why the choices we make are sensible for a number of practical problems.

This paper is structured as follows. In section 2, we discuss the underlying chemical problem (detailing the statement of the problem without and with control constraints included) and represent it in terms of matrix systems. In section 3, we introduce some basic saddle point theory and use this to devise effective preconditioners for the matrices which arise. In section 4, we present numerical results to demonstrate the performance of our iterative solvers in practice. Finally in section 5, we make some concluding remarks.

2. Problem formulation and discretization. Throughout this paper we consider an optimal control problem based on that considered in [4]. The objective function that has to be minimized is given by

$$
\begin{aligned}
J(u,v,c) = &\frac{\alpha_u}{2}\left\|u-u_Q\right\|^2_{L_2(Q)} + \frac{\alpha_v}{2}\left\|v-v_Q\right\|^2_{L_2(Q)} \\
&+ \frac{\alpha_{TU}}{2}\left\|u(\mathbf{x},T)-u_\Omega\right\|^2_{L_2(\Omega)} + \frac{\alpha_{TV}}{2}\left\|v(\mathbf{x},T)-v_\Omega\right\|^2_{L_2(\Omega)} + \frac{\alpha_c}{2}\left\|c\right\|^2_{L_2(\Sigma)},
\end{aligned} \tag{2.1}
$$

where u and v refer to concentrations of reactants (which in this problem are *state variables*), and c is the *control variable*, which also influences the underlying reaction. The spatial domain on which the problem is solved is given by $\Omega \subset \mathbb{R}^d$ with $d \in \{2,3\}$, and the time domain is taken to be the interval $t \in [0,T]$. We then have the space-time domain Q given by $Q := \Omega \times [0,T]$, as well as the space-time boundary given by $\Sigma := \partial\Omega \times (0,T)$. The goal of the optimization problem is to compute the quantities u, v, and c in such a way that they are close in the L_2-norm to what are often referred to as the *desired states* (u_Q, v_Q, u_Ω, v_Ω). Note that we have four desired states in this problem—two which are defined at all time points and two which are solely defined at the final time at which the problem is being solved. These are known quantities, which are typically determined from measurements and observations. In order for the objective function to resemble a physical or chemical process the variables need to satisfy the physics of the process of interest, which is typically modeled using one or more PDEs alongside additional constraints. In our case the constraints subject to which the objective function $J(u,v,c)$ is minimized are given by the following reaction-diffusion equations:

$$u_t - D_1\Delta u + k_1 u = -\gamma_1 uv \quad \text{in } Q, \tag{2.2}$$

$$v_t - D_2\Delta v + k_2 v = -\gamma_2 uv \quad \text{in } Q, \tag{2.3}$$

$$D_1\partial_\nu u + b(\mathbf{x},t,u) = c \quad \text{on } \Sigma, \tag{2.4}$$

$$D_2\partial_\nu v + \widetilde{\varepsilon} v = 0 \quad \text{on } \Sigma, \tag{2.5}$$

$$u(\mathbf{x},0) = u_0(\mathbf{x}) \quad \text{in } \Omega, \tag{2.6}$$

$$v(\mathbf{x},0) = v_0(\mathbf{x}) \quad \text{in } \Omega, \tag{2.7}$$

$$c \in C_{ad} = \{c \in L_\infty(\Sigma) : c_a \le c \le c_b \text{ a.e. on } \Sigma\}. \tag{2.8}$$

The quantities α_u, α_v, α_{TU}, α_{TV}, α_c, D_1, D_2, k_1, k_2, γ_1, γ_2, and $\widetilde{\varepsilon}$ are nonnegative constants. The function c describing the boundary condition (2.4) is the control variable defined above, and ∂_ν denotes the normal derivative. Equations (2.6) and (2.7) define the initial conditions for both concentrations. Additionally, we can impose

so-called box constraints on the control as stated in (2.8). In [22] Griesse and Volkwein also consider an integral constraint on c, which we do not discuss here. In some cases it might also be sensible to include state constraints for the concentrations u and v, which would be described by

$$u_a \le u \le u_b, \quad v_a \le v \le v_b.$$

State constraints typically bring additional difficulties to optimal control problems (see [10, 33]) and are not considered further in this present paper. For the remainder of this paper we will also follow the assumptions of $b(\mathbf{x}, t, u) = 0$ and $\widetilde{\varepsilon} = 0$, as studied in [22]. There are two approaches for solving the above problem. The first is the so-called discretize-then-optimize approach, where we discretize the objective function and constraint to build a discrete Lagrangian, and then impose the optimality conditions in the discrete setting. The second is known as the optimize-then-discretize approach, where we instead build a Lagrangian for the infinite dimensional problem and then discretize the first order conditions. There is no preferred approach and we refer to [30] for a discussion of the two cases. We note that recently it has become a paradigm to create discretization schemes such that both approaches lead to the same discrete first order system. We also need to deal with the nonlinearity of the PDE constraint. We here apply a simple sequential quadratic programming (SQP) or Lagrange–Newton method. Before we proceed to the derivation of optimality conditions and discretization, we split the problem into two stages: solving the nonlinear PDEs without control constraints and solving the system with the additional control constraints incorporated.

Newton system without control constraints. In this section we wish to further describe how the above problem can be examined and in particular focus on how to treat the nonlinearity of the PDEs. We proceed by formally building the (continuous) Lagrangian subject to the reaction-diffusion system

$$\begin{aligned}
u_t - D_1\Delta u + k_1 u &= -\gamma_1 uv \quad \text{in } Q,\\
v_t - D_2\Delta v + k_2 v &= -\gamma_2 uv \quad \text{in } Q,\\
D_1\partial_\nu u &= c \quad \text{on } \Sigma,\\
D_2\partial_\nu v &= 0 \quad \text{on } \Sigma,\\
u(\mathbf{x}, 0) &= u_0(\mathbf{x}) \quad \text{in } \Omega,\\
v(\mathbf{x}, 0) &= v_0(\mathbf{x}) \quad \text{in } \Omega,
\end{aligned}$$

giving

$$\begin{aligned}
\mathcal{L}(u, v, c, p, q) = J(u, v, c) &+ \int_Q p(u_t - D_1\Delta u + k_1 u + \gamma_1 uv)\\
&+ \int_Q q(v_t - D_2\Delta v + k_2 v + \gamma_2 uv)\\
&+ \int_\Sigma p_\Sigma(D_1\partial_\nu u - c) + \int_\Sigma q_\Sigma(D_2\partial_\nu v).
\end{aligned}$$

Here we have split up the *adjoint variables* p and q into interior and boundary parts (p and p_Σ, and q and q_Σ). We note that for brevity, when constructing $\mathcal{L}$, we included only the PDE part without boundary and initial conditions, which of course also need to be incorporated. We also make the assumption $\alpha_{TU} = \alpha_{TV} = 0$ in the working

below; the case where this is not so may be treated similarly. A rigorous derivation of the first order conditions can be found in [4, 22], to which we refer the interested reader. By taking the Fréchet derivatives with respect to state, control, and adjoint variables and equating the resulting expressions to zero, we obtain the first order conditions, or Karush–Kuhn–Tucker (KKT) conditions, given by

$$\begin{aligned}
-p_t - D_1\Delta p + k_1 p + \gamma_1 pv + \gamma_2 qv + \alpha_u(u - u_Q) &= 0 \quad \text{in } Q,\\
-q_t - D_2\Delta q + k_2 q + \gamma_2 qu + \gamma_1 pu + \alpha_v(v - v_Q) &= 0 \quad \text{in } Q,\\
\partial_\nu p = \partial_\nu q &= 0 \quad \text{on } \Sigma,\\
\alpha_c c - p &= 0 \quad \text{on } \Sigma,\\
u_t - D_1\Delta u + k_1 u + \gamma_1 uv &= 0 \quad \text{in } Q,\\
v_t - D_2\Delta v + k_2 v + \gamma_2 uv &= 0 \quad \text{in } Q,\\
\partial_\nu u - D_1^{-1} c &= 0 \quad \text{on } \Sigma,\\
\partial_\nu v &= 0 \quad \text{on } \Sigma.
\end{aligned}$$

We may abbreviate this set of nonlinear equations describing the first order conditions, using the notation $\Phi(\mathbf{x}) = \mathbf{0}$. We can use Newton's method to solve this problem via the relation $\Phi'(\mathbf{x}_k)\mathbf{s}_k = -\Phi(\mathbf{x}_k)$.

We now construct the Fréchet derivative of Φ, obtaining

$$-(s_p)_t - D_1\Delta s_p + k_1 s_p + \gamma_1(ps_v + s_p v) + \gamma_2(qs_v + s_q v) + \alpha_u s_u = b_1, \tag{2.9}$$

$$-(s_q)_t - D_2\Delta s_q + k_2 s_q + \gamma_2(qs_u + s_q u) + \gamma_1(ps_u + s_p u) + \alpha_v s_v = b_2, \tag{2.10}$$

$$\alpha_c s_c - s_p = b_3, \tag{2.11}$$

$$(s_u)_t - D_1\Delta s_u + k_1 s_u + \gamma_1(vs_u + s_v u) = b_4, \tag{2.12}$$

$$(s_v)_t - D_2\Delta s_v + k_2 s_v + \gamma_2(us_v + s_u v) = b_5. \tag{2.13}$$

Here we denote with $\mathbf{b} = [b_1, b_2, b_3, b_4, b_5]^T := -\Phi(\mathbf{x}_k)$ the right-hand side of the Newton system. Note that we did not write down the boundary conditions; however, they naturally carry through to the Newton system. If we now write all the equations together into an infinite dimensional system, the matrix describing the Newton process is given by

$$\begin{bmatrix}
\alpha_u \mathrm{Id} & \gamma_1 p + \gamma_2 q & 0 & \mathcal{L}_u' & \gamma_2 v \\
\gamma_2 q + \gamma_1 p & \alpha_v \mathrm{Id} & 0 & \gamma_1 u & \mathcal{L}_v' \\
0 & 0 & \alpha_c D_1^{-1}\mathrm{Id} & -D_1^{-1}\mathrm{Id} & 0 \\
\mathcal{L}_u & \gamma_1 u & -D_1^{-1}\mathrm{Id} & 0 & 0 \\
\gamma_2 v & \mathcal{L}_v & 0 & 0 & 0
\end{bmatrix}, \tag{2.14}$$

where

$$\mathcal{L}_u = \frac{\partial}{\partial t} - D_1\Delta + k_1\mathrm{Id} + \gamma_1 v, \quad \mathcal{L}_u' = -\frac{\partial}{\partial t} - D_1\Delta + k_1\mathrm{Id} + \gamma_1 v,$$
$$\mathcal{L}_v = \frac{\partial}{\partial t} - D_2\Delta + k_2\mathrm{Id} + \gamma_2 u, \quad \mathcal{L}_v' = -\frac{\partial}{\partial t} - D_2\Delta + k_2\mathrm{Id} + \gamma_2 u,$$

and Id denotes the identity operator.

In order to numerically solve the above problem we need to discretize the system (2.14) and the right-hand side $-\Phi(\mathbf{x}_k)$.

We first note that the system (2.14) is in saddle point form (as defined in section 3), and its discrete counterpart (using a backward Euler time-stepping scheme) is given by

$$\underbrace{\begin{bmatrix} \tau\mathcal{M}_1 & 0 & \mathcal{K}^T \\ 0 & \alpha_c\tau D_1^{-1}\mathcal{M}_c & -\tau D_1^{-1}\mathcal{N}^T \\ \mathcal{K} & -\tau D_1^{-1}\mathcal{N} & 0 \end{bmatrix}}_{\mathcal{A}} \begin{bmatrix} \mathbf{y} \\ \mathbf{c} \\ \boldsymbol{\lambda} \end{bmatrix} = \mathbf{b} \tag{2.15}$$

with

$$\mathcal{M}_1 = \text{blkdiag}\left(M_1^{(1)}, M_1^{(2)}, \ldots, M_1^{(N_t-1)}, M_1^{(N_t)}\right),$$
$$\mathcal{M}_c = \text{blkdiag}(M_c, M_c, \ldots, M_c, M_c),$$
$$\mathcal{N} = \begin{bmatrix} N & & & & & & \\ & 0 & & & & & \\ & & N & & & & \\ & & & 0 & & & \\ & & & & \ddots & & \\ & & & & & N & \\ & & & & & & 0 \end{bmatrix},$$

where

$$M_1^{(i)} = \begin{bmatrix} \alpha_u M & \gamma_1 M_{p_{(i)}} + \gamma_2 M_{q_{(i)}} \\ \gamma_1 M_{p_{(i)}} + \gamma_2 M_{q_{(i)}} & \alpha_v M \end{bmatrix}, \quad i = 1, \ldots, N_t.$$

Here, M denotes a standard finite element mass matrix, M_c is a boundary mass matrix, and the matrix N consists of evaluations of inner products from the term $\int_{\partial\Omega} w\text{tr}(z)$ with w a function on the boundary $\partial\Omega$, z a test function for the domain Ω, and tr the trace operator. The matrices $M_{p_{(i)}}$ and $M_{q_{(i)}}$ are mass-like matrices the entries of which are terms of the form $\int_\Omega \bar{p}\phi_j\phi_l$ and $\int_\Omega \bar{q}\phi_j\phi_l$, respectively (where $\bar{p}$ and $\bar{q}$ represent the previous Newton iterates of the adjoint variables—or Lagrange multipliers—p and q), and the vectors $\mathbf{y}$ and $\boldsymbol{\lambda}$ correspond to the discretized state $(\mathbf{u}, \mathbf{v})$ and adjoint $(\mathbf{p}, \mathbf{q})$ variables, respectively. The quantity N_t denotes the number of time-steps used, with τ the size of the time-step.

Finally, the matrix $\mathcal{K}$ represents the discretized PDE and can be written as

$$\mathcal{K} = \begin{bmatrix} L^{(1)} & & & & \\ -M_d & L^{(2)} & & & \\ & \ddots & \ddots & & \\ & & -M_d & L^{(N_t-1)} & \\ & & & -M_d & L^{(N_t)} \end{bmatrix},$$

where

$$M_d = \begin{bmatrix} M & 0 \\ 0 & M \end{bmatrix}$$

and

$$L^{(i)} = \begin{bmatrix} M + \tau(D_1 K + k_1 M + \gamma_1 M_{v_{(i)}}) & \tau\gamma_1 M_{u_{(i)}} \\ \tau\gamma_2 M_{v_{(i)}} & M + \tau(D_2 K + k_2 M + \gamma_2 M_{u_{(i)}}) \end{bmatrix}$$

with K the standard finite element stiffness matrix, and $M_{u_{(i)}}$ and $M_{v_{(i)}}$ mass-like matrices with terms of the form $\int_\Omega \bar{u}\phi_j\phi_l$ and $\int_\Omega \bar{v}\phi_j\phi_l$, where $\bar{u}$ and $\bar{v}$ correspond to the previous Newton iterates of the state variables u and v.

Note that we can solve for the updated states, control, and adjoints directly, which also makes the computation of the right-hand side cheaper, that is,

$$\mathcal{A}\begin{bmatrix} u^{(k+1)} \\ v^{(k+1)} \\ c^{(k+1)} \\ p^{(k+1)} \\ q^{(k+1)} \end{bmatrix} = \mathcal{A}\begin{bmatrix} u^{(k)} \\ v^{(k)} \\ c^{(k)} \\ p^{(k)} \\ q^{(k)} \end{bmatrix} + \widetilde{\mathbf{b}} = \begin{bmatrix} \alpha_u u_Q + (\gamma_1 p^{(k)} + \gamma_2 q^{(k)})v^{(k)} \\ \alpha_v v_Q + (\gamma_2 q^{(k)} + \gamma_1 p^{(k)})u^{(k)} \\ 0 \\ \gamma_1 u^{(k)} v^{(k)} \\ \gamma_2 v^{(k)} u^{(k)} \end{bmatrix},$$

in continuous form.

So far we have only discussed the Newton method to solve the KKT conditions. Note that for certain values of the states, Lagrange multipliers, and parameters we might run into the problem of obtaining an indefinite $(1,1)$-block of $\mathcal{A}$, caused by an indefinite matrix $\mathcal{M}_1$ [15]. For this reason we briefly highlight that for this purpose different techniques within the SQP step can be employed, such as line-search or trust region approaches—these may explored in future research into this subject area. One alternative that we also mention within the numerical results of section 4 is a Gauss–Newton approach (see [24]), where we ignore all mixed derivatives of the Hessian with respect to the Lagrange multipliers, resulting in a matrix system defined by the matrix

$$\begin{bmatrix} \alpha_u \mathrm{Id} & 0 & 0 & \mathcal{L}_u' & \gamma_2 v \\ 0 & \alpha_v \mathrm{Id} & 0 & \gamma_1 u & \mathcal{L}_v' \\ 0 & 0 & \alpha_c D_1^{-1}\mathrm{Id} & -D_1^{-1}\mathrm{Id} & 0 \\ \mathcal{L}_u & \gamma_1 u & -D_1^{-1}\mathrm{Id} & 0 & 0 \\ \gamma_2 v & \mathcal{L}_v & 0 & 0 & 0 \end{bmatrix}. \tag{2.16}$$

We find that preconditioners for the matrix (2.16) can be derived using the methodology presented in section 3.

Problem with control constraints. The problem we have discussed so far did not include any additional constraints on the control c. We now wish to discuss how pointwise constraints on the control, i.e.,

$$c_a(\mathbf{x},t) \le c(\mathbf{x},t) \le c_b(\mathbf{x},t),$$

may be dealt with. The treatment of control constraints can typically be carried out using a semismooth Newton method introduced in [7]. (For further information we refer to [27, 30, 58].) For the special case of the reaction-diffusion system we point to literature such as [4, 19, 20, 21, 22] for discussions on control constraints. In general the gradient equation of the Lagrangian becomes a variational inequality, which is in turn solved using the semismooth Newton method or equivalently [27] a primal-dual active set method. In contrast to [7] we employ a penalty technique, which has been applied very successfully to state-constrained optimal control problems, called the Moreau–Yosida penalty function [25, 32, 37]—this approach has also been applied to control-constrained problems [55]. The advantage of this approach is that the method does not need to work on submatrices corresponding to the free variables, which would

require a reassembly of matrices for every Newton step and would make preconditioning the matrix systems more difficult. From experience [55], the performance of this approach is comparable to the approach that directly uses the semismooth Newton method. In the Moreau–Yosida framework, the constraints

$$c_a(\mathbf{x},t) \leq c(\mathbf{x},t) \leq c_b(\mathbf{x},t)$$

are incorporated into the objective function via a penalization term, that is, we instead minimize the functional

$$J(u,v,c) + \frac{1}{2\varepsilon}\left\|\max\left\{0, c-c_b\right\}\right\|^2_{L_2(\Sigma)} + \frac{1}{2\varepsilon}\left\|\min\left\{0, c-c_a\right\}\right\|^2_{L_2(\Sigma)}$$

subject to the state equations detailed above. We can now proceed using the semismooth Newton approach, solving linear systems of the form

$$\begin{bmatrix} \tau\mathcal{M}_1 & 0 & \mathcal{K}^T \\ 0 & \alpha_c\tau D_1^{-1}\mathcal{L}_c & -\tau D_1^{-1}\mathcal{N}^T \\ \mathcal{K} & -\tau D_1^{-1}\mathcal{N} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{c} \\ \boldsymbol{\lambda} \end{bmatrix} = \widetilde{\mathbf{b}} \tag{2.17}$$

at each Newton step, where

$$\mathcal{L}_c = \begin{bmatrix} M_c + \varepsilon^{-1} G_{\mathcal{A}^{(1)}} M_c G_{\mathcal{A}^{(1)}} & & \\ & \ddots & \\ & & M_c + \varepsilon^{-1} G_{\mathcal{A}^{(N_t)}} M_c G_{\mathcal{A}^{(N_t)}} \end{bmatrix}.$$

Here $\mathcal{A}^{(i)} = \mathcal{A}_+^{(i)} \cup \mathcal{A}_-^{(i)}$ defines the active sets for every time-step of the discretized problem, that is,

$$\mathcal{A}_+^{(i)} = \left\{j \in \{1,2,\ldots,N\} : (c_i)_j > (c_b)_{i,j}\right\}, \tag{2.18}$$

$$\mathcal{A}_-^{(i)} = \left\{j \in \{1,2,\ldots,N\} : (c_i)_j < (c_a)_{i,j}\right\}, \tag{2.19}$$

using the control $\mathbf{c}$ from the previous iteration. The quantities $(c_i)_j$, $(c_b)_{i,j}$, and $(c_a)_{i,j}$ denote the values of c, c_b, and c_a at the ith time-step and the jth node, with N representing the total number of nodes. This method is schematically shown in Algorithm 1, where we assume here that the problem is already discretized.

ALGORITHM 1. Active set algorithm.

1: Choose initial values for $\mathbf{c}^{(0)}$, $\mathbf{p}^{(0)}$, $\mathbf{q}^{(0)}$, $\mathbf{u}^{(0)}$, $\mathbf{v}^{(0)}$
2: Set the active sets $\mathcal{A}_+^{(0)}$, $\mathcal{A}_-^{(0)}$ and $\mathcal{A}_I^{(0)}$ by using $\mathbf{c}^{(0)}$ in (2.18), (2.19)
3: **for** $k = 1, 2, \ldots$ **do**
4: Solve (2.17) (a system on the free variables from the previous iteration ($\mathcal{A}_I^{(k-1)}$))
5: Set the active sets $\mathcal{A}_+^{(k)}$, $\mathcal{A}_-^{(k)}$ and $\mathcal{A}_I^{(k)}$ by using $\mathbf{c}^{(k)}$ as given in (2.18), (2.19)
6: **if** $\mathcal{A}_+^{(k)} = \mathcal{A}_+^{(k-1)}$, $\mathcal{A}_-^{(k)} = \mathcal{A}_-^{(k-1)}$, and $\mathcal{A}_I^{(k)} = \mathcal{A}_I^{(k-1)}$ **then**
7: STOP (Algorithm converged)
8: **end if**
9: **end for**

3. Solving the linear systems.

Krylov solvers. We now wish to discuss how to efficiently solve linear systems of the form $\mathcal{A}\mathbf{x} = \mathbf{b}$ that arise at the heart of the Lagrange–Newton method discussed in the previous section. We here decide to employ Krylov subspace methods, which have previously been found to be very efficient for a number of optimal control problems subject to PDE constraints [47, 48, 49, 53]. In our case, as the system matrix is symmetric and indefinite, one option would be to employ the MINRES [43] method introduced by Paige and Saunders. This is a short-term recurrence method [12], requiring only a minimal amount of storage and involving one matrix-vector multiplication per iteration. MINRES minimizes the 2-norm of the residual $\mathbf{r}_k = \mathbf{b} - \mathcal{A}\mathbf{x}_k$ over the current Krylov subspace, where $\mathbf{x}_k$ is the approximation to $\mathbf{x}$ at step k of this procedure. Alternatively, there are many widely used nonsymmetric solvers such as GMRES [52] and biconjugate gradients (BICG) [13] which could be used. Of course, any Krylov method should only be effective if a preconditioner $\mathcal{P}$ is introduced such that the properties of the left-preconditioned system

$$\mathcal{P}^{-1}\mathcal{A}\mathbf{x} = \mathcal{P}^{-1}\mathbf{b}$$

are better than that of the unpreconditioned system $\mathcal{A}\mathbf{x} = \mathbf{b}$. Specifically, $\mathcal{P}$ is constructed in order to capture the properties of the matrix $\mathcal{A}$ well and so that it is easy to invert. For excellent introductions to the topic of constructing preconditioners for saddle point problems, we refer to [5, 11] and the references mentioned therein. As a guideline for constructing good preconditioners we use the known results that if the saddle point matrix

$$\mathcal{A} = \begin{bmatrix} A & B^T \\ B & -C \end{bmatrix}$$

is invertible, then the (ideal) block preconditioners

$$\mathcal{P}_1 = \begin{bmatrix} A & 0 \\ 0 & S \end{bmatrix}, \quad \mathcal{P}_2 = \begin{bmatrix} A & 0 \\ B & -S \end{bmatrix},$$

where A is the unchanged $(1,1)$-block of the saddle point matrix and $S = C + BA^{-1}B^T$ is the (negative) *Schur complement* of $\mathcal{A}$, satisfy $\lambda(\mathcal{P}_1^{-1}\mathcal{A}) \in \{1, \frac{1\pm\sqrt{5}}{2}\}$ provided $C = 0$ [38, 39], and $\lambda(\mathcal{P}_2^{-1}\mathcal{A}) \in \{1\}$ for any matrix C [31]. Therefore, although $\mathcal{P}_2$ is nondiagonalizable, both $\mathcal{P}_1$ and $\mathcal{P}_2$ are extremely effective preconditioners for $\mathcal{A}$. Of course in practice, we would not wish to explicitly invert A and S to apply the ideal preconditioner; however, if we construct good approximations to the $(1,1)$-block and the Schur complement of the system (2.15), an appropriate iterative solver is likely to converge rapidly when used with a preconditioner consisting of these approximations. As pointed out earlier the $(1,1)$-block of the preconditioner may be indefinite—in this case we cannot employ a symmetric Krylov subspace solver. Now faced with the decision of choosing a nonsymmetric Krylov method, we wish to point out that it is not straightforward to pick the "best method" (see [40]) and the convergence of the Krylov subspace solver might not be adequately described by the eigenvalues of the preconditioned matrix system [18]. Nevertheless in practice a good clustering of the eigenvalues often leads to fast convergence of the iterative scheme, and it can be seen that with a good preconditioner many methods behave in a similar way.

It is also possible to employ multigrid approaches to such saddle point problems. This class of methods has previously been shown to demonstrate good performance when applied to solve a number of PDE-constrained optimization problems, subject to both steady and transient PDEs [1, 2, 8, 9, 23, 24, 28, 29, 57].

We emphasize once more that the matrix systems we seek to solve fit into this saddle point framework. For the problem described in section 2 without control constraints, for instance,

$$A = \begin{bmatrix} \tau\mathcal{M}_1 & 0 \\ 0 & \alpha_c\tau D_1^{-1}\mathcal{M}_c \end{bmatrix}, \quad B = \begin{bmatrix} \mathcal{K} & -\tau D_1^{-1}\mathcal{N} \end{bmatrix}, \quad C = \begin{bmatrix} 0 \end{bmatrix}.$$

We may therefore employ the theory of saddle point systems to develop preconditioners for this problem.

Approximating the (1, 1)-block. In the case of a PDE-constrained optimization problem with a linear PDE as the constraint and a cost functional of the form discussed in section 2, the $(1,1)$-block of the resulting matrix system is a block diagonal matrix containing mass matrices (see [47, 49, 53], for instance), which can be handled very efficiently. In this case, however, we have to take into account that the $(1,1)$-block now contains blocks of the form

$$\begin{bmatrix} \alpha_u M & \gamma_1 M_{p_{(i)}} + \gamma_2 M_{q_{(i)}} \\ \gamma_1 M_{p_{(i)}} + \gamma_2 M_{q_{(i)}} & \alpha_v M \end{bmatrix}, \tag{3.1}$$

which demonstrates one of the major complexities encountered when attempting to solve such nonlinear problems numerically. When we seek to approximate these blocks, we use the saddle point theory as stated above to take as our approximation

$$\begin{bmatrix} \alpha_u M - \alpha_v^{-1}\left(\gamma_1 M_{p_{(i)}} + \gamma_2 M_{q_{(i)}}\right) M^{-1}\left(\gamma_1 M_{p_{(i)}} + \gamma_2 M_{q_{(i)}}\right) & 0 \\ \gamma_1 M_{p_{(i)}} + \gamma_2 M_{q_{(i)}} & \alpha_v M \end{bmatrix}$$
$$=: \begin{bmatrix} A_0^{(i)} & 0 \\ \gamma_1 M_{p_{(i)}} + \gamma_2 M_{q_{(i)}} & \alpha_v M \end{bmatrix}.$$

Using the saddle point result concerning block triangular preconditioners, we observe that each preconditioned block has eigenvalues all equal to 1 using this approximation. Note that these complicated looking matrices are actually straightforward to handle as we assume that the mass matrices are lumped for our work.[1] The block $\mathcal{M}_c$, which also forms part of the $(1,1)$-block of our matrix systems, may be approximated using Chebyshev semi-iteration [16, 17, 59] for consistent mass matrices or by simple inversion for lumped mass matrices.

Approximating the Schur complement. We now focus on approximating the Schur complement of the matrix system, which is given by

$$S = \frac{1}{\tau}\mathcal{K}\mathcal{M}_1^{-1}\mathcal{K}^T + \frac{\tau}{\alpha_c D_1}\mathcal{N}\mathcal{M}_c^{-1}\mathcal{N}^T.$$

One approach that we would predict to prove successful for moderate values of the parameter α_c (motivated by work undertaken in [47], for instance) is to use the approximation

$$\widehat{S}_1 = \frac{1}{\tau}\mathcal{K}\mathcal{M}_1^{-1}\mathcal{K}^T, \tag{3.2}$$

[1] In the case where mass matrices are not lumped, we believe that we may take a similar approximation but replace the mass matrices by their diagonals within the preconditioner.

that is, to drop the second term of the exact Schur complement for our approximation. However for smaller values of α_c we find this approximation does not produce satisfactory results. Hence, approximations that provide robustness with respect to the crucial problem parameters have been investigated (see [35, 44, 46, 53, 60]). The idea introduced in [44, 46] for simpler problems uses an approximation of the form

$$\widehat{S}_2 = \frac{1}{\tau}\left(\mathcal{K} + \widehat{\mathcal{M}}\right)\mathcal{M}_1^{-1}\left(\mathcal{K} + \widehat{\mathcal{M}}\right)^T,$$

where $\widehat{\mathcal{M}}$ is chosen to accommodate a better approximation of the term that was initially dropped from S. To discover an approach for finding such an approximation, we first study $\widehat{S}_2$ more closely:

$$\widehat{S}_2 = \frac{1}{\tau}\left(\mathcal{K}\mathcal{M}_1^{-1}\mathcal{K}^T + \widehat{\mathcal{M}}\mathcal{M}_1^{-1}\widehat{\mathcal{M}} + \mathcal{K}\mathcal{M}_1^{-1}\widehat{\mathcal{M}} + \widehat{\mathcal{M}}\mathcal{M}_1^{-1}\mathcal{K}^T\right).$$

Our goal is for the second term of the Schur complement approximation $\widehat{S}_2$ to accurately approximate the second term of the exact Schur complement S, that is,

$$\frac{1}{\tau}\widehat{\mathcal{M}}\mathcal{M}_1^{-1}\widehat{\mathcal{M}} \approx \frac{\tau}{\alpha_c D_1}\mathcal{N}\mathcal{M}_c^{-1}\mathcal{N}^T. \tag{3.3}$$

We now consider a block diagonal approximation $\widehat{\mathcal{M}}$ and recall the block structure of the other matrices involved. The most complex term which needs to be considered is the $\mathcal{M}_1^{-1}$ term, which involves inverting 2×2 block matrices of the form (3.1). To carry out this task, we observe that, given suitable invertibility conditions, the inverse of a 2×2 block matrix $\left[\begin{smallmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{smallmatrix}\right]$ may be expressed as

$$\begin{bmatrix} (A_{11} - A_{12}A_{22}^{-1}A_{21})^{-1} & -(A_{11} - A_{12}A_{22}^{-1}A_{21})^{-1}A_{12}A_{22}^{-1} \\ -(A_{22} - A_{21}A_{11}^{-1}A_{12})^{-1}A_{21}A_{11}^{-1} & (A_{22} - A_{21}A_{11}^{-1}A_{12})^{-1} \end{bmatrix},$$

which can be easily checked. We may use this expression to note that the problem of finding a suitable approximation (3.3) can be reduced to finding a matrix $\widehat{\mathcal{M}} = \text{blkdiag}(\widehat{M}_1^{(1)}, \widehat{M}_2^{(1)}, \widehat{M}_1^{(2)}, \widehat{M}_2^{(2)}, \ldots, \widehat{M}_1^{(N_t)}, \widehat{M}_2^{(N_t)})$ such that

$$\begin{bmatrix} \tau^{-1}\widehat{M}_1^{(i)}A_0^{-(i)}\widehat{M}_1^{(i)} & 0 \\ 0 & \tau^{-1}\alpha_v^{-1}\widehat{M}_2^{(i)}M^{-1}\widehat{M}_2^{(i)} \end{bmatrix} \approx \begin{bmatrix} \tau\alpha_c^{-1}D_1^{-1}NM_c^{-1}N^T & 0 \\ 0 & 0 \end{bmatrix}$$

for $i = 1, \ldots, N_t$, where $A_0^{-(i)} := \left(A_0^{(i)}\right)^{-1}$.

We may therefore conclude that it is appropriate to take $\widehat{M}_2^{(1)} = \widehat{M}_2^{(2)} = \cdots = \widehat{M}_2^{(N_t)} = 0$, with $\widehat{M}_1^{(i)}$ chosen such that

$$\frac{1}{\tau}\widehat{M}_1^{(i)}A_0^{-(i)}\widehat{M}_1^{(i)} \approx \frac{\tau}{\alpha_c D_1}M_\Gamma,$$

where $M_\Gamma := NM_c^{-1}N^T$. Given that the matrices $A_0^{-(i)}$ and M_Γ are diagonal, the above criterion will be satisfied if $\widehat{M}_1^{(i)}$ is a diagonal matrix, with diagonal entries given by

$$\widehat{m}_{1,jj}^{(i)} = \frac{\tau}{\sqrt{D_1\alpha_c}}\,|a_{0,jj}^{(i)}|^{1/2}m_{\Gamma,jj}^{1/2},$$

where $a^{(i)}_{0,jj}$ and $m_{\Gamma,jj}$ are the jth diagonal entries of $A^{(i)}_0$ and M_Γ, respectively. It would also be appropriate to make the selection

$$\widehat{m}^{(i)}_{1,jj} = \frac{\tau}{\sqrt{D_1\alpha_c}}\, h^{\frac{1}{2}(d-1)}|a^{(i)}_{0,jj}|^{1/2}$$

when j corresponds to a node on $\partial\Omega$, using the fact that M_c (which is equal to the nonzero part of M_Γ) is spectrally equivalent to $h^{d-1}I$, where d is the dimension of Ω and h the mesh-size used.

We may build these choices of $\widehat{M}^{(i)}_1$ into the matrix $\widehat{\mathcal{M}}$ and in turn into the Schur complement approximation $\widehat{S}_2$. We can also check heuristically that these choices of $\widehat{\mathcal{M}}$ ensure that $\tau^{-1}\widehat{\mathcal{M}}\mathcal{M}_1^{-1}\widehat{\mathcal{M}} \approx \tau\alpha_c^{-1}D_1^{-1}\mathcal{N}\mathcal{M}_c^{-1}\mathcal{N}^T$ by taking the approximations $M \approx h^d I$, $M_c \approx h^{d-1}I$ (where I are identity matrices of different dimensions), and writing

$$\begin{aligned}\left(\frac{1}{\tau}\widehat{\mathcal{M}}\mathcal{M}_1^{-1}\widehat{\mathcal{M}}\right)_{jj} &\approx \frac{1}{\tau}\cdot\frac{\tau}{\sqrt{D_1\alpha_c}}\, h^{\frac{1}{2}(d-1)}|a^{(i)}_{0,jj}|^{1/2}\cdot a^{-(i)}_{0,jj}\cdot\frac{\tau}{\sqrt{D_1\alpha_c}}\, h^{\frac{1}{2}(d-1)}|a^{(i)}_{0,jj}|^{1/2}\\ &= \frac{\tau}{\alpha_c D_1}h^{d-1} \approx \left(\frac{\tau}{\alpha_c D_1}\mathcal{N}\mathcal{M}_c^{-1}\mathcal{N}^T\right)_{jj}\end{aligned}$$

whenever the index j corresponds to a boundary node. (Both sides of the expression would be equal to zero otherwise.) In the above work, we have assumed that $a^{(i)}_{0,jj} \neq 0$.

Let us now consider how our approximations of the $(1,1)$-block A and (negative) Schur complement S may be applied. Due to the potential indefiniteness of A, as well as the nonsymmetry of the preconditioner used, a nonsymmetric solver, such as GMRES or BICG, needs to be applied. Given that this is the case, we recommend that a block triangular preconditioner of the form $\mathcal{P}_2$ be used, of the following structure:

$$\mathcal{P}_2 = \begin{bmatrix} \tau\widehat{\mathcal{M}}_1 & 0 & 0 \\ 0 & \alpha_c\tau D_1^{-1}\mathcal{M}_c & 0 \\ \mathcal{K} & -\tau D_1^{-1}\mathcal{N} & -\widehat{S}_2 \end{bmatrix},$$

where $\widehat{\mathcal{M}}_1$ denotes the approximation of $\mathcal{M}_1$ described above.

Alternative Schur complement approximation. We note at this point that as we apply a nonsymmetric iterative method to solve the matrix system discussed, we see no reason a nonsymmetric Schur complement approximation could not be used. For instance, it seems feasible to utilize an approximation

$$\widehat{S}_3 = \frac{1}{\tau}\left(\mathcal{K}+\widehat{\mathcal{M}}\right)\mathcal{M}_1^{-1}\left(\mathcal{K}+\widetilde{\mathcal{M}}\right)^T,$$

where in general $\widehat{\mathcal{M}}$ is not equal to $\widetilde{\mathcal{M}}$. We may consider block diagonal matrices

$$\begin{aligned}\widehat{\mathcal{M}} &= \text{blkdiag}\left(\widehat{M}^{(1)}_{11}, 0, \widehat{M}^{(2)}_{11}, 0, \ldots, \widehat{M}^{(N_t)}_{11}, 0\right),\\ \widetilde{\mathcal{M}} &= \text{blkdiag}\left(\widehat{M}^{(1)}_{21}, 0, \widehat{M}^{(2)}_{21}, 0, \ldots, \widehat{M}^{(N_t)}_{21}, 0\right)\end{aligned}$$

and, similarly to above, select $\widehat{M}^{(i)}_{11}$ and $\widehat{M}^{(i)}_{21}$ to be diagonal matrices. Their diagonal entries may be given by

$$\begin{aligned}\widehat{m}^{(i)}_{11,jj} &= \frac{\tau}{\sqrt{D_1\alpha_c}}\,|a^{(i)}_{0,jj}|,\\ \widehat{m}^{(i)}_{21,jj} &= \frac{\tau}{\sqrt{D_1\alpha_c}}\, m_{\Gamma,jj}, \quad \text{or} \quad \widehat{m}^{(i)}_{21,jj} = \frac{\tau}{\sqrt{D_1\alpha_c}}\, h^{\frac{1}{2}(d-1)},\end{aligned}$$

ALGORITHM 2. Inexact Uzawa method.
1: Select $\mathbf{x}_0$.
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: $\quad \mathbf{x}_{k+1} = \mathbf{x}_k + \mathcal{P}^{-1}\left(\mathbf{b} - \left(L + \text{blkdiag}(\widehat{M}, 0)\right)\mathbf{x}_k\right)$
4: **end for**

for example. Such choices of $\widehat{\mathcal{M}}$ and $\widetilde{\mathcal{M}}$ should ensure that the approximation $\tau^{-1}\widehat{\mathcal{M}}\mathcal{M}_1^{-1}\widetilde{\mathcal{M}} \approx \tau\alpha_c^{-1}D_1^{-1}\mathcal{N}\mathcal{M}_c^{-1}\mathcal{N}^T$ holds, as for the Schur complement approximation $\widehat{S}_2$.

We note that whether the Schur complement approximation $\widehat{S}_2$ or $\widehat{S}_3$ is used, the inverses of matrices of the form $\mathcal{K}+\widehat{\mathcal{M}}$ need to be approximated for every application of the Schur complement. We here use a Uzawa scheme [51] that approximately solves for diagonal blocks of the form $L + \text{blkdiag}(\widehat{M}, 0)$ of $\mathcal{K} + \widehat{\mathcal{M}}$. (See Algorithm 2 for a sketch of the routine used.) The preconditioner $\mathcal{P}$ is of block diagonal form and for each of these matrices applies an algebraic multigrid (AMG) technique to approximate the diagonal blocks of $L + \text{blkdiag}(\widehat{M}, 0)$.

Preconditioning for Gauss–Newton system. Let us now consider whether the approach detailed above may be applied to the matrix systems arising from a Gauss–Newton method. The matrix involved is given by (2.16) in continuous form, which in discrete form results in the same matrix (2.15) as for the Newton method, except now with

$$\mathcal{M}_1 = \text{blkdiag}\left(\alpha_u M, \alpha_v M, \alpha_u M, \alpha_v M, \ldots, \alpha_u M, \alpha_v M\right).$$

For this matrix, we may approximate the $(1,1)$-block $A = \text{blkdiag}\left(\tau\mathcal{M}_1, \tau\alpha_c D_1^{-1}\mathcal{M}_c\right)$ exactly. When developing an approximation of the form

$$\widehat{S}_2 = \frac{1}{\tau}\left(\mathcal{K} + \widehat{\mathcal{M}}\right)\mathcal{M}_1^{-1}\left(\mathcal{K} + \widehat{\mathcal{M}}\right)^T$$

to the Schur complement

$$S = \frac{1}{\tau}\mathcal{K}\mathcal{M}_1^{-1}\mathcal{K}^T + \frac{\tau}{\alpha_c D_1}\mathcal{N}\mathcal{M}_c^{-1}\mathcal{N}^T,$$

we may therefore look again for a matrix of the form $\widehat{\mathcal{M}}$ such that

$$\frac{1}{\tau}\widehat{\mathcal{M}}\mathcal{M}_1^{-1}\widehat{\mathcal{M}} \approx \frac{\tau}{\alpha_c D_1}\mathcal{N}\mathcal{M}_c^{-1}\mathcal{N}^T.$$

As for the Newton system, this problem reduces to finding an alternating block diagonal matrix $\widehat{\mathcal{M}} = \text{blkdiag}(\widehat{M}_1^{(1)}, 0, \widehat{M}_1^{(2)}, 0, \ldots, \widehat{M}_1^{(N_t)}, 0)$ such that

$$\frac{1}{\tau}\begin{bmatrix} \widehat{M}_1^{(i)} & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} \alpha_u M & 0 \\ 0 & \alpha_v M \end{bmatrix}^{-1}\begin{bmatrix} \widehat{M}_1^{(i)} & 0 \\ 0 & 0 \end{bmatrix} \approx \frac{\tau}{\alpha_c D_1}\begin{bmatrix} NM_c^{-1}N^T & 0 \\ 0 & 0 \end{bmatrix}$$

for $i = 1, \ldots, N_t$. This suggests that we should take

$$\frac{1}{\tau\alpha_u}\widehat{M}_1^{(i)}M^{-1}\widehat{M}_1^{(i)} \approx \frac{\tau}{\alpha_c D_1}M_\Gamma,$$

which is achieved if $\widehat{M}_1^{(1)} = \cdots = \widehat{M}_1^{(N_t)} = \widehat{M}_1$, where $\widehat{M}_1$ is a diagonal matrix with diagonal entries

$$\widehat{m}_{1,jj} = \tau\sqrt{\frac{\alpha_u}{D_1\alpha_c}}\, m_{jj}^{1/2} m_{\Gamma,jj}^{1/2} \quad \text{or} \quad \tau\sqrt{\frac{\alpha_u}{D_1\alpha_c}}\, h^{d-\frac{1}{2}}.$$

Here, m_{jj} and $m_{\Gamma,jj}$ denote the jth diagonal entries of M and M_Γ, respectively.

These approximations of A and S may be incorporated into a block diagonal or block triangular saddle point preconditioner. For the numerical results of section 4, we will once again consider block triangular preconditioners for matrix systems of this form.

Preconditioning for control constraints. We also wish to examine how the system (2.17), which incorporates inequality constraints on the control variable, may be preconditioned effectively. The $(1,1)$-block now contains the matrix $\alpha_c\tau D_1^{-1}\mathcal{L}_c$, which is a simple block diagonal matrix that can be treated in the same way as the $(1,1)$-block of the problem without control constraints. Approximating the Schur complement

$$S = \frac{1}{\tau}\mathcal{K}\mathcal{M}_1^{-1}\mathcal{K}^T + \frac{\tau}{\alpha_c D_1}\mathcal{N}\mathcal{L}_c^{-1}\mathcal{N}^T$$

is again the more challenging task. We now wish to use the technique employed earlier and write

$$\widehat{S}_2 = \frac{1}{\tau}\left(\mathcal{K} + \widehat{\mathcal{M}}\right)\mathcal{M}_1^{-1}\left(\mathcal{K} + \widehat{\mathcal{M}}\right)^T,$$

where $\widehat{\mathcal{M}}$ is chosen such that

$$\frac{1}{\tau}\widehat{\mathcal{M}}\mathcal{M}_1^{-1}\widehat{\mathcal{M}} \approx \frac{\tau}{\alpha_c D_1}\mathcal{N}\mathcal{L}_c^{-1}\mathcal{N}^T.$$

Note that $\frac{\tau}{\alpha_c D_1}\mathcal{N}\mathcal{L}_c^{-1}\mathcal{N}^T$ is a block diagonal matrix with blocks of the form

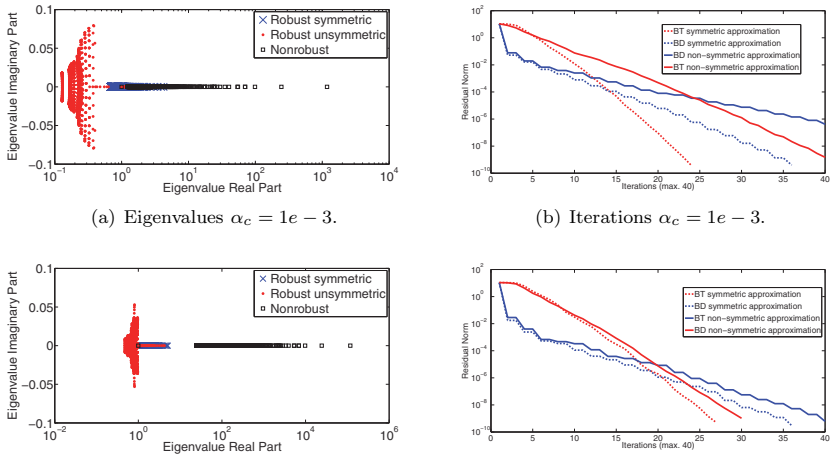
$$\frac{\tau}{\alpha_c D_1} N\left(M_c + \varepsilon^{-1} G_{\mathcal{A}^{(i)}} M_c G_{\mathcal{A}^{(i)}}\right)^{-1} N^T, \quad i = 1, \ldots, N_t,$$

alternating with zero blocks. Hence, we see that $\widehat{\mathcal{M}}$ should again have an alternating block diagonal structure, that is, $\widehat{\mathcal{M}} = \text{blkdiag}(\widehat{M}_1^{(1)}, 0, \widehat{M}_1^{(2)}, 0, \ldots, \widehat{M}_1^{(N_t)}, 0)$, as before.

Let us now employ the notation $l_{\Gamma,jj}^{(i)}$ for the diagonal entries of the matrix $N\left(M_c + \varepsilon^{-1} G_{\mathcal{A}^{(i)}} M_c G_{\mathcal{A}^{(i)}}\right)^{-1} N^T$, which will be nonzero on the diagonals corresponding to boundary nodes and zero otherwise. Then, in complete analogy to the case without control constraints, we may motivate the following choice for the diagonal entries of $\widehat{M}_1^{(i)}$:

$$\widehat{m}_{1,jj}^{(i)} = \frac{\tau}{\sqrt{D_1\alpha_c}}\, |a_{0,jj}^{(i)}|^{1/2} \left(l_{\Gamma,jj}^{(i)}\right)^{1/2}.$$

These choices of $\widehat{M}_1^{(i)}$ may again be built into the matrix $\widehat{\mathcal{M}}$ and hence the Schur complement approximation $\widehat{S}_2$.



(a) Eigenvalues $\alpha_c = 1e-3$. (b) Iterations $\alpha_c = 1e-3$.

(c) Eigenvalues $\alpha_c = 1e-5$. (d) Iterations $\alpha_c = 1e-5$.

FIG. 3.1. *Eigenvalues of* $S\mathbf{v} = \lambda\widehat{S}\mathbf{v}$ *for various approximations of the Schur complement (left) including a nonrobust approximation.* GMRES *iterations for the saddle point problem using the different Schur complement approximations and also block diagonal and block triangular preconditioners (right). The problem size is relatively small (*$\dim(S) = 5000$*).*

Effectiveness of Schur complement approximations. To motivate our choices of Schur complement approximation, we wish to illustrate the properties of the preconditioned matrix systems when the approximations derived in this section are used. We note that good clustering of these eigenvalues alone will not guarantee rapid convergence of a nonsymmetric iterative solver such as BICG or GMRES; however, it should at least indicate the prudence of our selections.[2]

In Figure 3.1, we aim to demonstrate this effectiveness in two different ways. In plots (a) and (c) we show, for different values of α_c, the eigenvalues of the preconditioned Schur complement when robust symmetric (that is, $\widehat{S}_2$), robust unsymmetric ($\widehat{S}_3$), and nonrobust ($\widehat{S}_1$) approximations are used. The plots indicate that whereas the eigenvalues of both $\widehat{S}_2^{-1}S$ and $\widehat{S}_3^{-1}S$ seem to be fairly clustered, the results when $\widehat{S}_2$ is taken as the Schur complement approximation seem to be the best.

In plots (b) and (d), we show GMRES convergence plots for a test problem when a range of preconditioning choices are made, for different values of α_c. Specifically, we show results when the symmetric approximation (that is, $\widehat{S}_2$) and nonsymmetric approximation ($\widehat{S}_3$) are taken to the Schur complement and when a BT (block triangular preconditioner we considered in this section) and BD (analogous block diagonal preconditioner) are used within the GMRES method. Whereas the plots indicate that all four choices of preconditioner behave reasonably well, the block triangular preconditioner with Schur complement approximation $\widehat{S}_2$ yields the best results. We therefore use this preconditioner for the numerical results presented in the next section.

[2]Figure 3.1 considers the case without control constraints, but we note that the behavior of our preconditioners is very similar when control constraints are present.

4. Numerical experiments. In this section we present numerical results for the iterative methods we have described, using the Schur complement approximation $\widehat{S}_2$. We implement this methodology using the finite element package deal.II [3] with $Q1$ finite element basis functions on a quadrilateral mesh. For the AMG preconditioner, we use the Trilinos ML package [14] that applies a smoothed aggregation AMG. Within the multigrid routine we typically apply a Chebyshev smoother (10 steps) in combination with the application of 6 V-cycles. We currently regard our implementation as a proof of concept, as at present we reinitialize the AMG preconditioner upon each application. A possible alternative would be to store various preconditioners, however this is prohibitive from a computer memory point of view. Therefore, we believe that the development of an efficient technique using multigrid or a fixed number of steps of a simple iterative solver such as a Gauss–Seidel or Jacobi method should be investigated in the future. Consequently, we wish to emphasize that the timings presented here are not as rapid as they would be were the recomputation of the preconditioner at each application not required. If the varying preconditioners are handled efficiently, this could also lead to the relatively larger number of V-cycles being reduced—we choose to use this number of V-cycles as we wish to avoid the performance of the AMG routine being sensitive to parameter changes. Our implementation of BICG is stopped with a tolerance of 10^{-4} or smaller for the relative residual. Additionally, we stop the SQP method whenever the relative change between two consecutive solutions is smaller than a given tolerance, as specified in our examples. More sophisticated techniques [41] for carrying this out could be employed in the future. We feel that as our purpose is to illustrate the performance of our preconditioner the choice made here is appropriate. Our experiments are performed with $T = 1$ and $\tau = 0.05$, that is, with 20 time-steps. We take the parameters $\alpha_{TU} = \alpha_{TV} = 0$ in all our numerical experiments, though we find it makes little difference computationally if this is not the case. We only consider three-dimensional examples here and specify $\Omega \subset \mathbb{R}^3$ for each example. All results are performed on a Centos Linux machine with Intel Xeon CPU X5650 at 2.67 GHz CPUs and 48 GB of RAM. We present overall timings for the solution process in seconds.

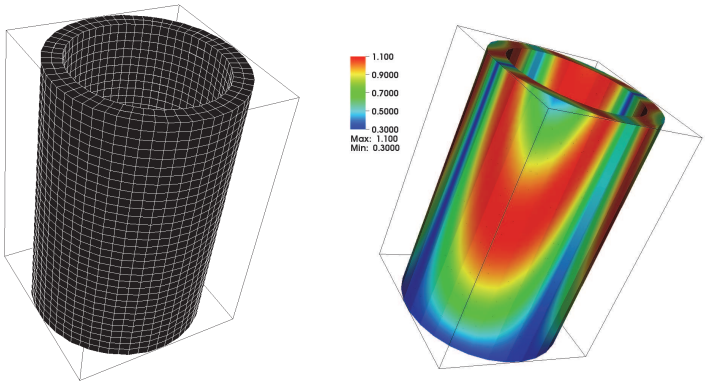
No control constraints.

Example 1. The first example we consider involves a cylindrical shell domain shown in Figure 4.1(a) with inner radius 0.8, outer radius 1.0, and height 3.0. The parameter setup for this problem is as follows: the desired state for the first reactant is shown in Figure 4.1(b) and is given by

$$u_Q = 2t\,|\sin(2x_1x_2x_3)| + 0.3,$$

where $\mathbf{x} = [x_1, x_2, x_3]^T$, and the desired state for the second reactant is given by $v_Q = 0$. Additionally, we have $D_1 = D_2 = k_1 = k_2 = 1$ and $\gamma_1 = \gamma_2 = 0.15$. Figure 4.2 demonstrates computed state and control variables for this problem at a particular time-step.

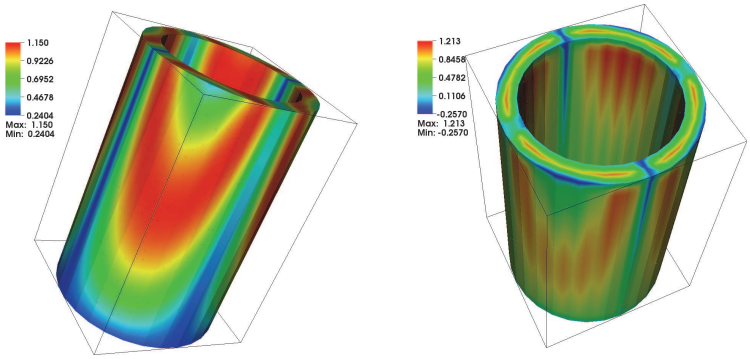
In Table 4.1 we show for each step of the SQP method the number of BICG iterations needed to achieve the required convergence. The first column indicates the number of degrees of freedom (i.e., the dimension of the matrix systems being solved), with the second and fifth columns providing the timings for all SQP steps at the level of mesh refinement. The third and sixth columns give the number of SQP steps needed to reach convergence, and the remaining fourth and seventh columns give the iteration numbers needed for BICG to converge to the desired tolerance. For this setup we require three SQP steps to reach the tolerance of 10^{-6}. The results



(a) Domain

(b) Desired state for first reactant at eighth time-step

FIG. 4.1. *Cylindrical shell domain for computations and desired state for the first reactant at the eighth time-step.*



(a) Computed state for first reactant at eighth time-step

(b) Computed control at eighth time-step

FIG. 4.2. *Computed control and state for the first reactant at the eighth time-step with* $\alpha_c = 1e-5$ *and* $\alpha_u = \alpha_v = 1.0$.

indicate a benign mesh-dependence of the preconditioner, which from our experience can often be observed for boundary control problems. We also observe nearly constant iteration numbers when the regularization parameter is varied.

Example 2. The setup used for the second example is similar to that for the first. Here, however, we take the desired states

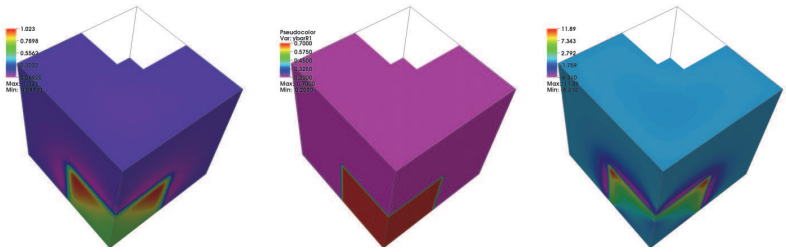
$$u_Q = \begin{cases} 0.7 & \text{for } (x_1, x_2, x_3) \in [0, 0.5]^3, \\ 0.2 & \text{otherwise,} \end{cases} \qquad v_Q = 0,$$

with the parameters $D_1 = D_2 = k_1 = k_2 = 1$ and $\gamma_1 = \gamma_2 = 0.15$. In contrast to the previous example we solve the optimization problem on a Hyper L domain consisting

TABLE 4.1

Results on the cylindrical shell domain for varying mesh-size and regularization parameter α_c. SQP steps are shown in columns 3 and 6 with the corresponding BICG *iteration numbers in columns 4 and 7. The timings (in seconds) measure the total time for convergence of the SQP scheme.*

DoF	Time	SQP step	BICG	Time	SQP step	BICG
		$\alpha_c = 1e-5$			$\alpha_c = 1e-3$	
538 240	1726	step 1	16	1995	step 1	17
		step 2	16		step 2	20
		step 3	16		step 3	20
3 331 520	14904	step 1	28	14757	step 1	28
		step 2	27		step 2	31
		step 3	34		step 3	29



(a) Computed state for first reactant (b) Desired state (c) Computed control

FIG. 4.3. *Desired state, computed control, and state for the first reactant at eighth time-step with $\alpha_c = 1e-5$ and $\alpha_u = \alpha_v = 1.0$.*

of the cube on $[-1,1]^3$ with the cube $(0,1]^3$ removed (see Figure 4.3). Again, we wish to vary the control regularization parameter α_c and the mesh-size. Table 4.2 shows the results for the setup presented here, including timings and iteration numbers as explained in Table 4.1. We again observe a mild growth in iteration numbers with varying mesh-size and robustness for our selection of α_c values. We find all iteration numbers are very reasonable considering the complexity of the matrix system being solved.

TABLE 4.2

Results on the Hyper L domain for varying mesh-size and regularization parameter α_c. SQP steps are shown in columns 3 and 6 with the corresponding BICG *iteration numbers in columns 4 and 7. The timings (in seconds) measure the total time for convergence of the SQP scheme.*

DoF	Time	SQP	BICG	Time	SQP	BICG
		$\alpha_c = 1e-5$			$\alpha_c = 1e-3$	
60920	457	step 1	23	369	step 1	19
		step 2	25		step 2	20
		step 3	25		step 3	20
382 840	2819	step 1	29	2624	step 1	27
		step 2	35		step 2	30
		step 3	33		step 3	33
2 670 200	22976	step 1	46	19128	step 1	36
		step 2	52		step 2	44
		step 3	53		step 3	44

Varying parameters. We next consider a problem where the desired states are given by

$$u_Q = \begin{cases} 0.3 & \text{for } (x_1, x_2, x_3) \in [0, 0.5]^3, \\ 0.2 & \text{otherwise,} \end{cases} \qquad v_Q = 0,$$

and vary some values that have previously been assumed to be fixed. The default setup is again $D_1 = D_2 = k_1 = k_2 = 1$ and $\gamma_1 = \gamma_2 = 0.15$, with the stopping tolerance for the SQP method set as 10^{-4}. In the remainder of this section we vary one or two of these parameters and keep the others fixed. Clearly this does not cover all the relevant choices that might be possible, but this should indicate the effectiveness of our approach for a large range of parameter regimes. All computations are carried out on a fixed mesh that leads to a saddle point system of dimension 382840. We note that each of the problems tested represents a completely different setup of the PDE and the optimization problem. The purpose of presenting the results in Table 4.3 is to show that the iteration numbers for these scenarios are reasonable and sometimes very low. There are some specific parameter regimes for which this approach is not as effective as for the cases presented, but for a wide range of parameters (h, τ, α_u, α_v, α_c, D_1, D_2, k_1, k_2, γ_1, γ_2) we find that our approach works very well. Also presented in the table are results for the case when the tolerance of the iterative solver is decreased—it can be seen that the increase in iteration numbers is not dramatic for a decreased tolerance.

We may see from the results in Table 4.3 that especially for increasing values of γ_1 and γ_2 it is possible that the convergence deteriorates slightly due to the $(1,1)$-block having larger negative eigenvalues (as the increasing indefiniteness of the $(1,1)$-block in this case is not captured by our preconditioner). One way to overcome this issue is by switching from a Newton method to a Gauss–Newton scheme [6, 42]. This means that the off-diagonal blocks in (2.14) are ignored, which results in a typically slower convergence of the nonlinear iteration but provides better matrix properties during

TABLE 4.3

Results for varying parameters on the Hyper L domain with fixed dimension 382840 *and varying regularization parameter* α_c. *The timings (in seconds) measure the total time for convergence of the SQP scheme. We show the number of* BICG *iterations and the number of SQP steps.*

Parameter	Time	SQP step	BICG	Time	SQP step	BICG
		$\alpha_c = 1e-5$			$\alpha_c = 1e-3$	
$D_1 = D_2 = 100$	1783	step 1	28	1161	step 1	16
		step 2	33		step 2	22
$D_1 = D_2 = 0.1$	2083	step 1	19	1744	step 1	18
		step 2	27		step 2	20
		step 3	20		step 3	19
$\gamma_1 = \gamma_2 = 0.05$	2426	step 1	25	2199	step 1	22
		step 2	29		step 2	25
		step 3	29		step 3	25
$\gamma_1 = \gamma_2 = 0.75$	3240	step 1	20	3796	step 1	24
		step 2	60		step 2	36
		step 3	32		step 3	72
tol $= 1e-6$	3226	step 1	34	2702	step 1	27
		step 2	38		step 2	33
		step 3	38		step 3	33
tol $= 1e-8$	3749	step 1	39	3289	step 1	33
		step 2	46		step 2	39
		step 3	46		step 3	42

TABLE 4.4

Results for varying parameters γ_1 and γ_2 on the Hyper L domain with fixed dimension 382840 *and varying regularization parameter α_c for the Gauss–Newton scheme. The timings (in seconds) measure the total time for convergence of the SQP scheme. We show the number of* BICG *iterations and the number of GN steps.*

Parameter	Time	GN Step	BICG	Time	GN Step	BICG
		$\alpha_c = 1e-5$			$\alpha_c = 1e-3$	
$\gamma_1 = \gamma_2 = 0.75$	3107	step 1	25	2848	step 1	22
		step 2	27		step 2	25
		step 3	27		step 3	25
		step 4	27		step 4	24
$\gamma_1 = \gamma_2 = 1.75$	3249	step 1	25	2973	step 1	22
		step 2	28		step 2	26
		step 3	28		step 3	26
		step 4	26		step 4	26

the solution process. We observe that all our preconditioners can be applied in this case. Table 4.4 provides some results for this case with larger γ_1 and γ_2 and the desired states

$$u_Q = \begin{cases} 0.7 & \text{for } (x_1, x_2, x_3) \in [0, 0.5]^3, \\ 0.2 & \text{otherwise,} \end{cases} \qquad v_Q = 0.$$

It can be seen that we have a small increase in the number of Gauss–Newton (GN in the table) iterations compared to the number of SQP steps to reach the tolerance of 10^{-4} but that the number of BICG iterations remains (almost) constant.

Control constraints. We now present results for the case where control constraints are present. The domain of interest is again the Hyper L domain presented earlier, with the desired states given by

$$u_Q = t\,|\sin(2x_1x_2x_3)\cos(2x_1x_2x_3)|\,, \quad v_Q = 0,$$

and $D_1 = D_2 = k_1 = k_2 = 1$, $\gamma_1 = \gamma_2 = 0.15$. We work only with an upper bound on the control given by

$$c_b = 1.5.$$

The results for varying α_c and different mesh-sizes are shown in Table 4.5. We note that the convergence of the Newton method dealing with the control constraints (CCNM in the tables) seems to depend on the tolerance used for the solution of

TABLE 4.5

Results on the Hyper L domain for varying mesh-size and regularization parameter α_c. Shown are the number of SQP steps, the number of the Newton iterations for the CCNM, and the average number of BICG *iterations per step of the CCNM method.*

DoF	Time	SQP step	CCNM/BICG	Time	SQP step	CCNM/BICG
		$\alpha_c = 1e-5$			$\alpha_c = 1e-3$	
60 920	859	step 1	3/22.0	1066	step 1	3/18.0
		step 2	2/25.5		step 2	3/21.0
					step 3	3/21.0
382 840	13358	step 1	5/28.6	5498	step 1	2/26.0
		step 2	5/32.6		step 2	2/36.0
		step 3	5/32.8		step 3	2/35.0

TABLE 4.6

Results on the Hyper L domain for varying penalty parameter ε. Shown are the number of SQP steps, the number of the Newton iterations for the CCNM, and the average number of BICG *iterations per step of the CCNM method.*

DoF	SQP step	CCNM/ av.BICG	SQP step	CCNM/ av.BICG	SQP step	CCNM/ av.BICG
		$\varepsilon = 1e-2$		$\varepsilon = 1e-4$		$\varepsilon = 1e-6$
60 920	step 1	3/32.3	step 1	3/24.6	step 1	3/20.6
	step 2	3/36.3	step 2	3/26.6	step 2	3/22.3

the linear system (see [34]). The smaller value of α_c shown in Table 4.5 requires the tolerance for the iterative solver to be reduced, as otherwise we do not observe convergence of the Newton method to deal with the control constraints. Our stopping criterion for the Newton method is based on the coincidence of subsequent active sets, but a more sophisticated stopping criterion might be able to avoid the convergence issue of the Newton method [26, 41]. Table 4.5 shows the number of SQP steps, the number of semismooth Newton steps (CCNM) for the control constraints, and the average number of BICG iterations at each SQP step. We find that it is also feasible to handle the nonlinearity of the PDEs and the control constraints within a single Newton loop, and the matrix systems obtained using this approach are of the same structure as that derived in section 2. We see that there is a benign growth in BICG iteration numbers with respect to the mesh-size. The difference in iteration numbers for the two different values of α_c is likely to be due to the fact that as we change α_c the values for the control c change, which means that more nodes belong to the active sets than in the case with the larger value of α_c.

In addition, we wish to illustrate robustness with respect to the penalty parameter ε. We here keep the mesh-size, as well as the regularization parameter ($\alpha_c = 1e-3$), fixed and consider different values of ε. Table 4.6 illustrates that once again the resulting iteration numbers are very reasonable given the complexity of the problem. We also observe that the performance of the Newton method depends on the tolerance with which the linear systems were solved. For the rather low tolerance of 10^{-9} we find that the Newton scheme and the SQP method often converge in very few iterations. We sometimes observe that for smaller values of the penalty parameter the convergence of the outer SQP method is slower than for larger values. This may be caused by the use of our simple SQP scheme—as we mentioned previously more sophisticated schemes may be able to avoid this. Observe that the residual of the iterative solver depends on ε, and thus from our experience small tolerances are typically required to ensure convergence of the outer iteration. We note that it is also possible to replace the SQP scheme by a Gauss–Newton iteration to possibly avoid indefinite Hessians.

5. Conclusions. In this paper we have considered a PDE-constrained optimization problem based on reaction-diffusion equations used to model chemical processes. We devised nonlinear solvers to solve these problems, at the heart of which lay the solution of large-scale linear systems of saddle point structure. We have shown that these systems can be solved using efficient preconditioning techniques for a wide range of cases. We have introduced a preconditioner that efficiently approximates the $(1,1)$-block of the saddle point systems, and we additionally derived approximations of the Schur complement which were intended to be robust with respect to parameters within the construction of the problem. Our numerical results illustrated that for a variety

of problem setups (including problems with box constraints on the control variable) our method solves the matrix systems in low BICG iteration numbers. To summarize, the method presented here not only enables the accurate solution of chemical process models but also provides fast and robust techniques to do this.

Acknowledgments. The authors gratefully acknowledge the Max Planck Institute in Magdeburg for their hospitality. The authors would like to thank Andy Wathen for fruitful discussions regarding the presented work.

REFERENCES

[1] S. S. ADAVANI AND G. BIROS, *Multigrid algorithms for inverse problems with linear parabolic PDE constraints*, SIAM J. Sci. Comput., 31 (2008), pp. 369–397.
[2] U. ASCHER AND E. HABER, *A multigrid method for distributed parameter estimation problems*, Electron. Trans. Numer. Anal., 15 (2003), pp. 1–17.
[3] W. BANGERTH, R. HARTMANN, AND G. KANSCHAT, *deal.*II. *A general-purpose object-oriented finite element library*, ACM Trans. Math. Software, 33 (2007), pp. 24-1–24-7.
[4] W. BARTHEL, C. JOHN, AND F. TRÖLTZSCH, *Optimal boundary control of a system of reaction diffusion equations*, ZAMM Z. Angew. Math. Mech., 90 (2010), pp. 966–982.
[5] M. BENZI, G. H. GOLUB, AND J. LIESEN, *Numerical solution of saddle point problems*, Acta Numer., 14 (2005), pp. 1–137.
[6] M. BENZI, E. HABER, AND L. TARALLI, *A preconditioning technique for a class of PDE-constrained optimization problems*, Adv. Comput. Math., 35 (2011), pp. 149–173.
[7] M. BERGOUNIOUX, K. ITO, AND K. KUNISCH, *Primal-dual strategy for constrained optimal control problems*, SIAM J. Control Optim., 37 (1999), pp. 1176–1194.
[8] A. BORZÌ, *Multigrid methods for parabolic distributed optimal control problems*, J. Comput. Appl. Math., 157 (2003), pp. 365–382.
[9] A. BORZÌ AND V. SCHULZ, *Multigrid methods for PDE optimization*, SIAM Rev., 51 (2009), pp. 361–395.
[10] E. CASAS, *Control of an elliptic problem with pointwise state constraints*, SIAM J. Control Optim., 24 (1986), pp. 1309–1318.
[11] H. C. ELMAN, D. J. SILVESTER, AND A. J. WATHEN, *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*, in Numerical Mathematics and Scientific Computation, Oxford University Press, New York, 2005.
[12] V. FABER AND T. MANTEUFFEL, *Necessary and sufficient conditions for the existence of a conjugate gradient method*, SIAM J. Numer. Anal., 21 (1984), pp. 352–362.
[13] R. FLETCHER, *Conjugate gradient methods for indefinite systems*, in Numerical Analysis, Lecture Notes in Math., 506, Springer, Berlin, 1976, pp. 73–89.
[14] M. GEE, C. SIEFERT, J. HU, R. TUMINARO, AND M. SALA, *ML* 5.0 *Smoothed Aggregation User's Guide*, Tech. report SAND2006-2649, Sandia National Laboratories, 2006.
[15] P. E. GILL AND W. MURRAY, *Newton-type methods for unconstrained and linearly constrained optimization*, Math. Program., 7 (1974), pp. 311–350.
[16] G. H. GOLUB AND R. S. VARGA, *Chebyshev semi-iterative methods, successive over-relaxation iterative methods, and second order Richardson iterative methods.* I, Numer. Math., 3 (1961), pp. 147–156.
[17] G. H. GOLUB AND R. S. VARGA, *Chebyshev semi-iterative methods, successive over-relaxation iterative methods, and second order Richardson iterative methods.* II, Numer. Math., 3 (1961), pp. 157–168.
[18] A. GREENBAUM, V. PTÁK, AND Z. STRAKOŠ, *Any nonincreasing convergence curve is possible for GMRES*, SIAM J. Matrix Anal. Appl. 17 (1996), pp. 465–470.
[19] R. GRIESSE, *Parametric sensitivity analysis in optimal control of a reaction diffusion system.* I*: Solution differentiability*, Numer. Funct. Anal. Optim., 25 (2004), pp. 93–117.
[20] R. GRIESSE, *Parametric sensitivity analysis in optimal control of a reaction diffusion system.* II*: Practical methods and examples*, Optim. Methods Softw., 19 (2004), pp. 217–242.
[21] R. GRIESSE AND S. VOLKWEIN, *A primal-dual active set strategy for optimal boundary control of a nonlinear reaction-diffusion system*, SIAM J. Control Optim., 44 (2005), pp. 467–494.
[22] R. GRIESSE AND S. VOLKWEIN, *Parametric Sensitivity Analysis for Optimal Boundary Control of a 3D Reaction-Diffusion System*, Springer, New York, 2006.
[23] E. HABER, *A parallel method for large scale time domain electromagnetic inverse problems*, Appl. Numer. Math., 58 (2008), pp. 422–434.

[24] E. Haber, U. M. Ascher, and D. Oldenburg, *On optimization techniques for solving nonlinear inverse problems*, Inverse Problems, 16 (2000), pp. 1263–1280.
[25] R. Herzog and E. W. Sachs, *Preconditioned conjugate gradient method for optimal control problems with control and state constraints*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 2291–2317.
[26] M. Hintermüller, M. Hinze, and M. Tber, *An adaptive finite-element Moreau-Yosida-based solver for a non-smooth Cahn-Hilliard problem*, Optim. Methods Softw., 26 (2011), pp. 777–811.
[27] M. Hintermüller, K. Ito, and K. Kunisch, *The primal-dual active set strategy as a semismooth Newton method*, SIAM J. Optim., 13 (2002), pp. 865–888.
[28] M. Hinze, M. Köster, and S. Turek, *A Hierarchical Space-Time Solver for Distributed Control of the Stokes Equation*, Tech. report SPP1253-16-01, Priority Programme 1253, 2008.
[29] M. Hinze, M. Köster, and S. Turek, *A Space-Time Multigrid Solver for Distributed Control of the Time-Dependent Navier-Stokes System*, Tech. report, SPP1253-16-02, Priority Programme 1253, 2008.
[30] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich, *Optimization with PDE Constraints*, in Mathematical Modelling: Theory and Applications, Springer, New York, 2009.
[31] I. Ipsen, *A note on preconditioning non-symmetric matrices*, SIAM J. Sci. Comput., 23 (2001), pp. 1050–1051.
[32] K. Ito and K. Kunisch, *The primal-dual active set method for nonlinear optimal control problems with bilateral constraints*, SIAM J. Control Optim., 43 (2004), pp. 357–376.
[33] K. Ito and K. Kunisch, *Lagrange Multiplier Approach to Variational Problems and Applications*, Adv. Des. Control 15, SIAM, Philadelphia, 2008.
[34] C. Kanzow, *Inexact semismooth Newton methods for large-scale complementarity problems*, Optim. Methods Softw., 19 (2004), pp. 309–325.
[35] M. Kollmann and M. Kolmbauer, *A Preconditioned MinRes Solver for Time-Periodic Parabolic Optimal Control Problems*, Numer. Linear Algebra Appl., DOI: 10.1002/nla.1842, 2012.
[36] M. Kollmann and W. Zulehner, *A Robust Preconditioner for Distributed Optimal Control for Stokes Flow with Control Constraints*, Numer. Math. Adv. Appl., (2011), pp. 771–779.
[37] K. Krumbiegel, I. Neitzel, and A. Rösch, *Sufficient Optimality Conditions for the Moreau-Yosida-Type Regularization Concept Applied to the Semilinear Elliptic Optimal Control Problems with Pointwise State Constraints*, Tech. report 1503/2010, *Weierstrass Institute for Applied Analysis and Stochastics (Berlin)*, 2010.
[38] Y. A. Kuznetsov, *Efficient iterative solvers for elliptic finite element problems on nonmatching grids*, Russian J. Numer. Anal. Math. Modelling, 10 (1995), pp. 187–211.
[39] M. F. Murphy, G. H. Golub, and A. J. Wathen, *A note on preconditioning for indefinite linear systems*, SIAM J. Sci. Comput, 21 (2000), pp. 1969–1972.
[40] N. M. Nachtigal, S. C. Reddy, and L. N. Trefethen, *How fast are nonsymmetric matrix iterations?*, SIAM J. Matrix Anal. Appl, 13 (1992), pp. 778–795.
[41] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer Ser. Oper. Res., Springer, New York, 1999.
[42] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer Ser. Oper. Res. Financial Engineering, 2nd ed., Springer, New York, 2006.
[43] C. C. Paige and M. A. Saunders, *Solutions of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal, 12 (1975), pp. 617–629.
[44] J. W. Pearson, M. Stoll, and A. J. Wathen, *Regularization-robust preconditioners for time-dependent PDE-constrained optimization problems*, SIAM J. Matrix Anal. Appl., 33 (2012), pp. 1126–1152.
[45] J. W. Pearson and A. J. Wathen, *Fast Iterative Solvers for Convection-Diffusion Control Problems*, Electron. Trans. Numer. Anal., 40, (2013), pp. 294–310.
[46] J. W. Pearson and A. J. Wathen, *A new approximation of the Schur complement in preconditioners for PDE-constrained optimization*, Numer. Linear Algebra Appl., 19 (2012), pp. 816–829.
[47] T. Rees, H. S. Dollar, and A. J. Wathen, *Optimal solvers for PDE-constrained optimization*, SIAM J. Sci. Comput., 32 (2010), pp. 271–298.
[48] T. Rees and M. Stoll, *Block-triangular preconditioners for PDE-constrained optimization*, Numer. Linear Algebra Appl., 17 (2010), pp. 977–996.
[49] T. Rees, M. Stoll, and A. Wathen, *All-at-once preconditioners for PDE-constrained optimization*, Kybernetika, 46 (2010), pp. 341–360.

[50] T. Rees and A. Wathen, *Preconditioning iterative methods for the optimal control of the Stokes equation*, SIAM J. Sci. Comput., 33 (2011), pp. 2903–2926.
[51] Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, Philadelphia, 2003.
[52] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.
[53] J. Schöberl and W. Zulehner, *Symmetric indefinite preconditioners for saddle point problems with applications to PDE-constrained optimization problems*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 752–773.
[54] M. Stoll and A. Wathen, *All-at-once solution of time-dependent PDE-constrained optimization problems*, Oxford Centre for Collaborative and Applied Mathematics Technical Report 10/47, 2010.
[55] M. Stoll and A. Wathen, *Preconditioning for partial differential equation constrained optimization with control constraints*, Numer. Linear Algebra Appl., 19 (2012), pp. 53–71.
[56] M. Stoll and A. Wathen, *All-at-once solution of time-dependent Stokes control*, J. Comput. Phys., 232 (2013), pp. 498–515.
[57] S. Takacs and W. Zulehner, *Convergence analysis of multigrid methods with collective point smoothers for optimal control problems*, Comput. Vis. Sci., 14 (2011), pp. 131–141.
[58] M. Ulbrich, *Semismooth Newton Methods for Variational Inequalities and Constrained Optimization Problems*, SIAM, Philadelphia, 2011.
[59] A. J. Wathen and T. Rees, *Chebyshev semi-iteration in preconditioning for problems including the mass matrix*, Electron. Trans. Numer. Anal., 34 (2008), pp. 125–135.
[60] W. Zulehner, *Nonstandard norms and robust estimates for saddle point problems*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 536–560.

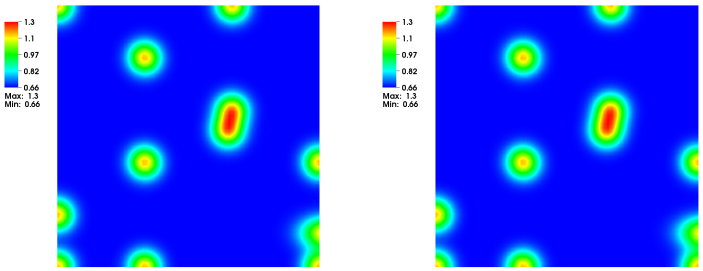
A.8 Preconditioning for pattern formation

This paper is published as

M. Stoll, J. W. Pearson, and P. K. Maini, ***Fast Solvers for Optimal Control Problems from Pattern Formation*, J. Comput. Phys., 304 (2016), pp. 27-45. (2015).**

Result from the paper

Pattern formation models are considered in this paper where we compare Newton and Gauss-Newton approaches and equip them with efficient preconditioners. Figure A.4 shows the computed vs. the desired state for a two-dimensional model.



(a) Desired State $\widehat{u}_{12}$ (b) Computed State u_{12}

Figure A.4: Desired and computed state

Journal of Computational Physics 304 (2016) 27–45

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Fast solvers for optimal control problems from pattern formation

Martin Stoll [a], John W. Pearson [b,*], Philip K. Maini [c]

[a] Computational Methods in Systems and Control Theory, Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, 39106 Magdeburg, Germany
[b] School of Mathematics, Statistics and Actuarial Science, University of Kent, Cornwallis Building (East), Canterbury, CT2 7NF, UK
[c] Wolfson Centre for Mathematical Biology, Mathematical Institute, University of Oxford, Radcliffe Observatory Quarter, Woodstock Road, Oxford, OX2 6GG, UK

ARTICLE INFO

Article history:
Received 24 March 2014
Received in revised form 5 July 2015
Accepted 1 October 2015
Available online 13 October 2015

Keywords:
PDE-constrained optimization
Reaction–diffusion
Pattern formation
Newton iteration
Preconditioning
Schur complement

ABSTRACT

The modeling of pattern formation in biological systems using various models of reaction–diffusion type has been an active research topic for many years. We here look at a parameter identification (or PDE-constrained optimization) problem where the Schnakenberg and Gierer–Meinhardt equations, two well-known pattern formation models, form the constraints to an objective function. Our main focus is on the efficient solution of the associated nonlinear programming problems via a Lagrange–Newton scheme. In particular we focus on the fast and robust solution of the resulting large linear systems, which are of saddle point form. We illustrate this by considering several two- and three-dimensional setups for both models. Additionally, we discuss an image-driven formulation that allows us to identify parameters of the model to match an observed quantity obtained from an image.

1. Introduction

One of the fundamental problems in developmental biology is to understand how spatial patterns, such as pigmentation patterns, skeletal structures, and so on, arise. In 1952, Alan Turing [42] proposed his theory of pattern formation in which he hypothesized that a system of chemicals, reacting and diffusing, could be driven unstable by diffusion, leading to spatial patterns (solutions which are steady in time but vary in space). He proposed that these chemical patterns, which he termed morphogen patterns, set up pre-patterns which would then be interpreted by cells in a concentration-dependent manner, leading to the patterns that we see.

These models have been applied to a very wide range of areas (see, for example, Murray [27]) and have been shown to exist in chemistry [6,30]. While their applicability to biology remains controversial, there are many examples which suggest that Turing systems may be underlying key patterning processes (see [2,8,40] for the most recent examples). Two important models which embody the essence of the original Turing model are the Gierer–Meinhardt [14] and Schnakenberg models [39] and it is upon these models which we focus.[1] In light of the fact that, to date, no Turing morphogens have

* Corresponding author.
E-mail addresses: stollm@mpi-magdeburg.mpg.de (M. Stoll), j.w.pearson@kent.ac.uk (J.W. Pearson), maini@maths.ox.ac.uk (P.K. Maini).

[1] Although the second model is commonly referred to as the Schnakenberg model, it was actually first proposed by Gierer and Meinhardt in [14] along with the model usually referenced as the Gierer–Meinhardt model – we therefore refer to the first and second models as '*GM*1' and '*GM*2' within our working.

http://dx.doi.org/10.1016/j.jcp.2015.10.006
0021-9991/

been unequivocally demonstrated, we do not have model parameter values so a key problem in mathematical biology is to determine parameters that give rise to certain observed patterns. It is this problem that the present study investigates.

More recently, an area in applied and numerical mathematics that has generated much research interest is that of PDE-constrained optimization problems (see [41] for an excellent introduction to this field). It has been found that one key application of such optimal control formulations is to find solutions to pattern formation problems [11,12], and so it is natural to explore this particular application here.

In this paper, we consider the numerical solution of optimal control (in this case parameter identification) formulations of these Turing models – in particular we wish to devise preconditioned iterative solvers for the matrix systems arising from the application of Newton and Gauss–Newton methods to the problems. The crucial aspect of the preconditioners is the utilization of saddle point theory to obtain effective approximations to the $(1,1)$-block and Schur complement of these matrix systems. The solvers incorporate aspects of iterative solution strategies developed by the first and second authors to tackle simpler optimal control problems in literature such as [32–35].

This paper is structured as follows. In Section 2 we introduce the Gierer–Meinhardt ($GM1$) and Schnakenberg ($GM2$) models that we consider, and outline the corresponding optimal control problems. In Section 3 we discuss the outer (Newton-type) iteration that we employ for these problems, and state the resulting matrix systems at each iteration. We then motivate and derive our preconditioning strategies in Section 4. In Section 5 we present numerical results to demonstrate the effectiveness of our approaches, and finally in Section 6 we make some concluding remarks.

2. A parameter identification problem

Parameter identification problems are crucial in determining the setup of a mathematical model, often given by a system of differential equations, that is best suited to describe measured data or an observed phenomenon. These problems are often posed as PDE-constrained optimization problems [20,41]. We here want to minimize an objective function of misfit type, i.e., the function is designed to penalize deviations of the function values from the observed or measured data. The particular form is given by [11,12]:

$$\begin{aligned}\mathcal{J}(u,v,a,b) = &\frac{\beta_1}{2}\|u(\mathbf{x},t)-\widehat{u}(\mathbf{x},t)\|^2_{L_2(\Omega\times[0,T])} + \frac{\beta_2}{2}\|v(\mathbf{x},t)-\widehat{v}(\mathbf{x},t)\|^2_{L_2(\Omega\times[0,T])}\\ &+\frac{\beta_{T,1}}{2}\|u(\mathbf{x},T)-\widehat{u}_T(\mathbf{x})\|^2_{L_2(\Omega)} + \frac{\beta_{T,2}}{2}\|v(\mathbf{x},T)-\widehat{v}_T(\mathbf{x})\|^2_{L_2(\Omega)}\\ &+\frac{\nu_1}{2}\|a(\mathbf{x},t)\|^2_{L_2(\Omega\times[0,T])} + \frac{\nu_2}{2}\|b(\mathbf{x},t)\|^2_{L_2(\Omega\times[0,T])},\end{aligned} \tag{2.1}$$

where u, v are the *state variables*, and a, b the *control variables*, in our formulation. This is to say we wish to ensure that the state variables are as close as possible in the L_2-norm to some observed or desired states $\widehat{u}$, $\widehat{v}$, $\widehat{u}_T$, $\widehat{v}_T$, but at the same time penalize the enforcement of controls that have large magnitudes in this norm. The space–time domain on which this problem is considered is given by $\Omega\times[0,T]$, where $\Omega\subset\mathbb{R}^d$, $d\in\{2,3\}$.

Our goal is to identify the parameters of classical pattern formation equations such that the resulting optimal parameters allow the use of these models for real-world data. We here use models of reaction–diffusion type typically exploited to generate patterns seen in biological systems. The two formulations we consider are the $GM1$ model [14,27]:

$$\begin{aligned} u_t - D_u\Delta u - \frac{ru^2}{v} + au &= r, \quad \text{on } \Omega\times[0,T],\\ v_t - D_v\Delta v - ru^2 + bv &= 0, \quad \text{on } \Omega\times[0,T],\\ u(\mathbf{x},0)=u_0(\mathbf{x}),\quad v(\mathbf{x},0)&=v_0(\mathbf{x}),\quad \text{on } \Omega,\\ \frac{\partial u}{\partial\nu}=\frac{\partial v}{\partial\nu}&=0,\quad \text{on } \partial\Omega\times[0,T],\end{aligned} \tag{2.2}$$

and the $GM2$ model [14,27,39]:

$$\begin{aligned} u_t - D_u\Delta u + \gamma(u-u^2v) - \gamma a &= 0, \quad \text{on } \Omega\times[0,T],\\ v_t - D_v\Delta v + \gamma u^2v - \gamma b &= 0, \quad \text{on } \Omega\times[0,T],\\ u(\mathbf{x},0)=u_0(\mathbf{x}),\quad v(\mathbf{x},0)&=v_0(\mathbf{x}),\quad \text{on } \Omega,\\ \frac{\partial u}{\partial\nu}=\frac{\partial v}{\partial\nu}&=0,\quad \text{on } \partial\Omega\times[0,T],\end{aligned} \tag{2.3}$$

where r and γ are non-negative parameters involved in the respective models.

Both the $GM1$ and $GM2$ formulations are models of reaction–diffusion processes occurring in many types of pattern formation and morphogenesis processes [14,27,39]. The $GM1$ model relates to an "activator–inhibitor" system, whereas the $GM2$ model represents substrate-depletion. Within both models the variables u and v, the state variables in our formulation, represent the concentrations of chemical products. The parameters D_u and D_v denote the diffusion coefficients – typically

it is assumed that v diffuses faster than u, so $D_u < D_v$ [14]. The (given) parameters r and γ are positive: the value r in the *GM*1 model denotes the (small) production rate of the activator [14], and the parameter γ in the *GM*2 model is the *Hill coefficient*, which describes the cooperativity within a binding process. The variables a and b, the control variables in our problem, represent the rates of decay for u and v, respectively. The initial conditions u_0 and v_0 are known.

Throughout the remainder of this article we consider the minimization of the cost functional (2.1), with PDE constraints taking the form of the *GM*1 model or the *GM*2 model. PDE-constrained optimization problems of similar form have been considered in the literature, such as in [11,12]. When solving these problems we consider a range of values of the parameters involved in the PDE models, as well as variations in the coefficients β_i, $\beta_{T,i}$, ν_i $(i = 1, 2)$ within $\mathcal{J}(u, v, a, b)$. One typically chooses β_i (and frequently $\beta_{T,i}$) to be larger than ν_i in order for the states to closely resemble the desired states, due to the input of control not being severely penalized – the case of larger ν_i relates to the control variables being very small, with the state variables therefore failing to match the desired states as closely. However it is possible to formulate these problems using a wide range of parameters, and so within the numerical results of Section 5 we vary the parameter setup to demonstrate the robustness of our methods.

Note that the PDE constraints, for either model (2.2) or (2.3), are nonlinear, and as a result the optimization problem $\min_{(u,v,a,b)} \mathcal{J}(u, v, a, b)$ is itself nonlinear (although the cost functional is quadratic). To solve this problem we are therefore required to apply nonlinear programming [29] algorithms. Many of these are generalizations of Newton's method [29]. We here focus on a Lagrange–Newton (or basic SQP) scheme and a Gauss–Newton method. At the heart of both approaches lies the solution of large linear systems, which are often in saddle point form [4,9], that represent the Hessian or an approximation to it. In order to be able to solve these large linear systems we need to employ iterative solvers [9,37], which can be accelerated using effective preconditioners.

3. Nonlinear programming

A standard way of how to proceed with the above nonlinear program is to consider a classical Lagrangian approach [41]. In our case, with a nonlinear constraint, we apply a nonlinear solver to the first order conditions. We hence start by deriving the first order conditions, or Karush–Kuhn–Tucker conditions, of the Lagrangian

$$\mathcal{L}(u, v, a, b, p, q) = \mathcal{J}(u, v, a, b) + (p, \mathcal{R}_1(u, v, a, b)) + (q, \mathcal{R}_2(u, v, a, b)),$$

where $\mathcal{R}_1(u, v, a, b)$, $\mathcal{R}_2(u, v, a, b)$ represent the first two equations of both *GM*1 and *GM*2 models, and p, q denote the *adjoint variables* (or *Lagrange multipliers*). Note that for convenience our Lagrangian ignores the boundary and initial conditions. In general form the first order conditions are given by

$$\begin{aligned} &\mathcal{L}_u = 0, \quad \mathcal{L}_v = 0, \\ &\mathcal{L}_a = 0, \quad \mathcal{L}_b = 0, \\ &\mathcal{L}_p = 0, \quad \mathcal{L}_q = 0. \end{aligned}$$

The equations are in general nonlinear and a standard Newton method can be applied to them to give the following Lagrange–Newton or SQP scheme:

$$\begin{bmatrix} \mathcal{L}_{uu} & \mathcal{L}_{uv} & \mathcal{L}_{ua} & \mathcal{L}_{ub} & \mathcal{L}_{up} & \mathcal{L}_{uq} \\ \mathcal{L}_{vu} & \mathcal{L}_{vv} & \mathcal{L}_{va} & \mathcal{L}_{vb} & \mathcal{L}_{vp} & \mathcal{L}_{vq} \\ \mathcal{L}_{au} & \mathcal{L}_{av} & \mathcal{L}_{aa} & \mathcal{L}_{ab} & \mathcal{L}_{ap} & \mathcal{L}_{aq} \\ \mathcal{L}_{bu} & \mathcal{L}_{bv} & \mathcal{L}_{ba} & \mathcal{L}_{bb} & \mathcal{L}_{bp} & \mathcal{L}_{bq} \\ \mathcal{L}_{pu} & \mathcal{L}_{pv} & \mathcal{L}_{pa} & \mathcal{L}_{pb} & \mathcal{L}_{pp} & \mathcal{L}_{pq} \\ \mathcal{L}_{qu} & \mathcal{L}_{qv} & \mathcal{L}_{qa} & \mathcal{L}_{qb} & \mathcal{L}_{qp} & \mathcal{L}_{qq} \end{bmatrix} \begin{bmatrix} \delta u \\ \delta v \\ \delta a \\ \delta b \\ \delta p \\ \delta q \end{bmatrix} = - \begin{bmatrix} \mathcal{L}_u \\ \mathcal{L}_v \\ \mathcal{L}_a \\ \mathcal{L}_b \\ \mathcal{L}_p \\ \mathcal{L}_q \end{bmatrix}, \tag{3.1}$$

where δu, δv, δa, δb, δp, δq denote the Newton updates for u, v, a, b, p, q.

Note that our formulation does not include any globalization techniques such as trust region or line search approaches [28]. In order for the optimization algorithm to converge these should in general be incorporated. As our focus here is on large-scale linear systems we do not focus on these approaches now. At this stage we simply state the systems obtained for both *GM*1 and *GM*2 models and refer the interested reader to Appendix A, where all quantities are derived in detail. The system given in (3.1) represents the most general Newton system but it is often possible to only use approximations to this system. The Gauss–Newton method [16] is often used as the corresponding system matrix in (3.1) – this ignores the mixed derivatives with respect to the primal variables, i.e. with the system matrix given by

$$\begin{bmatrix} \widehat{\mathcal{L}}_{uu} & 0 & 0 & 0 & \mathcal{L}_{up} & \mathcal{L}_{uq} \\ 0 & \widehat{\mathcal{L}}_{vv} & 0 & 0 & \mathcal{L}_{vp} & \mathcal{L}_{vq} \\ 0 & 0 & \widehat{\mathcal{L}}_{aa} & 0 & \mathcal{L}_{ap} & \mathcal{L}_{aq} \\ 0 & 0 & 0 & \widehat{\mathcal{L}}_{bb} & \mathcal{L}_{bp} & \mathcal{L}_{bq} \\ \mathcal{L}_{pu} & \mathcal{L}_{pv} & \mathcal{L}_{pa} & \mathcal{L}_{pb} & \mathcal{L}_{pp} & \mathcal{L}_{pq} \\ \mathcal{L}_{qu} & \mathcal{L}_{qv} & \mathcal{L}_{qa} & \mathcal{L}_{qb} & \mathcal{L}_{qp} & \mathcal{L}_{qq} \end{bmatrix}, \tag{3.2}$$

where the matrices denoted by $\widehat{\mathcal{L}}_{:,:}$ do not contain second derivative information (see [16,29] for more details). Additionally, to derive the infinite-dimensional Newton system we discretize the resulting equations using finite elements in space and a backward Euler scheme in time. The resulting system for the $GM1$ model is given by

$$\underbrace{\begin{bmatrix} \mathbf{A}_{u,GM1} & -2\tau r\mathbf{M}_{up/v^2} & -\tau\mathbf{M}_p & 0 & -\mathbf{L}_{u,GM1}^T & 2\tau r\mathbf{M}_u \\ -2\tau r\mathbf{M}_{up/v^2} & \mathbf{A}_{v,GM1} & 0 & -\tau\mathbf{M}_q & -\tau r\mathbf{M}_{u^2/v^2} & -\mathbf{L}_{v,GM1}^T \\ -\tau\mathbf{M}_p & 0 & \tau\nu_1\mathbf{M} & 0 & -\tau\mathbf{M}_u & 0 \\ 0 & -\tau\mathbf{M}_q & 0 & \tau\nu_2\mathbf{M} & 0 & -\tau\mathbf{M}_v \\ -\mathbf{L}_{u,GM1} & -\tau r\mathbf{M}_{u^2/v^2} & -\tau\mathbf{M}_u & 0 & 0 & 0 \\ 2\tau r\mathbf{M}_u & -\mathbf{L}_{v,GM1} & 0 & -\tau\mathbf{M}_v & 0 & 0 \end{bmatrix}}_{\mathcal{A}} \begin{bmatrix} \boldsymbol{\delta u} \\ \boldsymbol{\delta v} \\ \boldsymbol{\delta a} \\ \boldsymbol{\delta b} \\ \boldsymbol{\delta p} \\ \boldsymbol{\delta q} \end{bmatrix} = \mathbf{f},$$

where

$$\mathbf{A}_{u,GM1} = \tau\beta_1\mathbf{M} + \beta_{T,1}\mathbf{M}_T + 2\tau r\mathbf{M}_{p/v} + 2\tau r\mathbf{M}_q,$$

$$\mathbf{A}_{v,GM1} = \tau\beta_2\mathbf{M} + \beta_{T,2}\mathbf{M}_T + 2\tau r\mathbf{M}_{u^2p/v^3},$$

$$\mathbf{L}_{u,GM1} = \mathbf{M}_E + \tau D_u\mathbf{K} - 2\tau r\mathbf{M}_{u/v} + \tau\mathbf{M}_a,$$

$$\mathbf{L}_{v,GM1} = \mathbf{M}_E + \tau D_v\mathbf{K} + \tau\mathbf{M}_b.$$

Note that M and K denote standard finite element mass and stiffness matrices, respectively. Here the matrices

$$\mathbf{M}_E := \begin{bmatrix} M & & & & \\ -M & M & & & \\ & -M & M & & \\ & & \ddots & \ddots & \\ & & & -M & M \end{bmatrix}, \quad \mathbf{M}_T := \begin{bmatrix} 0 & & & & \\ & 0 & & & \\ & & \ddots & & \\ & & & 0 & \\ & & & & M \end{bmatrix},$$

correspond to, respectively, the time-stepping scheme used, and the values at the final time $t = T$. All other mass matrices $\mathbf{M}_\psi = \text{blkdiag}(M_\psi, \ldots, M_\psi)$ are obtained from evaluating integrals of the form $[M_\psi]_{ij} = \int \psi\phi_i\phi_j$ for each matrix entry and for every time-step, where ϕ_i denote the finite element basis functions used (see the group finite element method in [23]). Furthermore, the matrix $\mathbf{K} = \text{blkdiag}(K, \ldots, K)$. The parameter τ denotes the (constant) time-step used. The vector $\mathbf{f}$ is the discrete representation at each Newton step of the following vector function:

$$\begin{bmatrix} \beta_1\int(\widehat{u} - \bar{u}) + \int(-\bar{p}_t - D_u\Delta\bar{p} - 2r\frac{\bar{u}}{\bar{v}}\bar{p} + \bar{a}\bar{p} - 2r\bar{u}\bar{q}) \\ \beta_2\int(\widehat{v} - \bar{v}) + \int(-\bar{q}_t - D_v\Delta\bar{q} + r\frac{\bar{u}^2}{\bar{v}^2}\bar{p} + \bar{b}\bar{q}) \\ \int(\bar{u}\bar{p} - \nu_1\bar{a}) \\ \int(\bar{v}\bar{q} - \nu_2\bar{b}) \\ \int(\bar{u}_t - D_u\Delta\bar{u} - \frac{r\bar{u}^2}{\bar{v}} + \bar{a}\bar{u} - r) \\ \int(\bar{v}_t - D_v\Delta\bar{v} - r\bar{u}^2 + \bar{b}\bar{v}) \end{bmatrix},$$

where $\bar{u}$, $\bar{v}$, $\bar{a}$, $\bar{b}$, $\bar{p}$, $\bar{q}$ denote the previous Newton iterates for u, v, a, b, p, q.

The Gauss–Newton type matrix for this problem now becomes

$$\underbrace{\begin{bmatrix} \beta_1\tau\mathbf{M} & 0 & 0 & 0 & -\mathbf{L}_{u,GM1}^T & 2\tau r\mathbf{M}_u \\ 0 & \beta_2\tau\mathbf{M} & 0 & 0 & -\tau r\mathbf{M}_{u^2/v^2} & -\mathbf{L}_{v,GM1}^T \\ 0 & 0 & \nu_1\tau\mathbf{M} & 0 & -\tau\mathbf{M}_u & 0 \\ 0 & 0 & 0 & \nu_2\tau\mathbf{M} & 0 & -\tau\mathbf{M}_v \\ -\mathbf{L}_{u,GM1} & -\tau r\mathbf{M}_{u^2/v^2} & -\tau\mathbf{M}_u & 0 & 0 & 0 \\ 2\tau r\mathbf{M}_u & -\mathbf{L}_{v,GM1} & 0 & -\tau\mathbf{M}_v & 0 & 0 \end{bmatrix}}_{\mathcal{A}} \begin{bmatrix} \boldsymbol{\delta u} \\ \boldsymbol{\delta v} \\ \boldsymbol{\delta a} \\ \boldsymbol{\delta b} \\ \boldsymbol{\delta p} \\ \boldsymbol{\delta q} \end{bmatrix} = \mathbf{f}_{GN},$$

with all matrices as previously defined (see [5,16] for details on the Gauss–Newton matrix structure). We consider this matrix system as well as the "pure Newton" formulation of the $GM1$ model, as we find that the Gauss–Newton method often results in favorable properties from an iterative solver point-of-view.

Moving on to the *GM2* model, Appendix A reveals the following structure of the Newton system:

$$\underbrace{\begin{bmatrix} \mathbf{A}_{u,GM2} & -2\tau\gamma\mathbf{M}_{u(q-p)} & 0 & 0 & -\mathbf{L}_{u,GM2}^T & -2\tau\gamma\mathbf{M}_{uv} \\ -2\tau\gamma\mathbf{M}_{u(q-p)} & \mathbf{A}_{v,GM2} & 0 & 0 & \tau\gamma\mathbf{M}_{u^2} & -\mathbf{L}_{v,GM2}^T \\ 0 & 0 & \tau\nu_1\mathbf{M} & 0 & \tau\gamma\mathbf{M} & 0 \\ 0 & 0 & 0 & \tau\nu_2\mathbf{M} & 0 & \tau\gamma\mathbf{M} \\ -\mathbf{L}_{u,GM2} & \tau\gamma\mathbf{M}_{u^2} & \tau\gamma\mathbf{M} & 0 & 0 & 0 \\ -2\tau\gamma\mathbf{M}_{uv} & -\mathbf{L}_{v,GM2} & 0 & \tau\gamma\mathbf{M} & 0 & 0 \end{bmatrix}}_{\mathcal{A}} \begin{bmatrix} \boldsymbol{\delta}\mathbf{u} \\ \boldsymbol{\delta}\mathbf{v} \\ \boldsymbol{\delta}\mathbf{a} \\ \boldsymbol{\delta}\mathbf{b} \\ \boldsymbol{\delta}\mathbf{p} \\ \boldsymbol{\delta}\mathbf{q} \end{bmatrix} = \mathbf{g},$$

with

$$\begin{aligned} \mathbf{A}_{u,GM2} &= \tau\beta_1\mathbf{M} + \beta_{T,1}\mathbf{M}_T + 2\tau\gamma\mathbf{M}_{v(q-p)}, \\ \mathbf{A}_{v,GM2} &= \tau\beta_2\mathbf{M} + \beta_{T,2}\mathbf{M}_T, \\ \mathbf{L}_{u,GM2} &= \mathbf{M}_E + \tau D_u\mathbf{K} + \tau\gamma\mathbf{M} - 2\tau\gamma\mathbf{M}_{uv}, \\ \mathbf{L}_{v,GM2} &= \mathbf{M}_E + \tau D_v\mathbf{K} + \tau\gamma\mathbf{M}_{u^2}, \end{aligned}$$

and $\mathbf{g}$ the discrete representation of the vector function:

$$\begin{bmatrix} \beta_1 \int(\widehat{u} - \bar{u}) + \int(-\bar{p}_t - D_u\Delta\bar{p} + 2\gamma\bar{u}\bar{v}(\bar{q} - \bar{p}) + \gamma\bar{p}) \\ \beta_2 \int(\widehat{v} - \bar{v}) + \int(-\bar{q}_t - D_v\Delta\bar{q} + \gamma\bar{u}^2(\bar{q} - \bar{p})) \\ -\int(\nu_1\bar{a} + \gamma\bar{p}) \\ -\int(\nu_2\bar{b} + \gamma\bar{q}) \\ \int(\bar{u}_t - D_u\Delta\bar{u} + \gamma(\bar{u} - \bar{u}^2\bar{v}) - \gamma\bar{a}) \\ \int(\bar{v}_t - D_v\Delta\bar{v} + \gamma\bar{u}^2\bar{v} - \gamma\bar{b}) \end{bmatrix}.$$

The main challenge is now the numerical evaluation of the discretized problems. As we here opt for an all-at-once approach where we discretize in space and time and then solve the resulting linear system for all time steps simultaneously, we need to be able to perform this operation efficiently. Similar approaches have recently been considered in [33]. The goal of the next section is to introduce the appropriate methodology.

4. Preconditioning and Krylov subspace solver

The solution of large-scale linear systems of the *saddle point* form:

$$\mathcal{A}\mathbf{x} = \mathbf{b}, \qquad \text{with} \quad \mathcal{A} = \begin{bmatrix} A & B^T \\ B & -C \end{bmatrix}, \tag{4.1}$$

where $A \in \mathbb{R}^{m\times m}$, $B \in \mathbb{R}^{n\times m}$ (with $m \geq n$), $C \in \mathbb{R}^{n\times n}$, is a topic of major interest within the numerical analysis community [4,9]. Due to the vast dimensionality of the systems derived earlier we cannot use factorization-based approaches [7]. We hence employ a Krylov subspace method [37] where we construct a Krylov subspace of the form

$$\mathcal{K}_k(\mathcal{A}, \mathbf{r}_0) = \text{span}\left\{\mathbf{r}_0, \mathcal{A}\mathbf{r}_0, \mathcal{A}^2\mathbf{r}_0, \ldots, \mathcal{A}^k\mathbf{r}_0\right\},$$

within which we seek an approximation to the solution of a given linear system. These methods are cheap as they only require multiplication with the system matrix, which is often possible to perform in a matrix-free way, that is to say the matrix $\mathcal{A}$ can be a black-box that only computes $\mathcal{A}\mathbf{w}$ for some vector $\mathbf{w}$. As a rule-of-thumb (rigorously in the case of symmetric $\mathcal{A}$) the eigenvalues of $\mathcal{A}$ determine how fast the approximate solution converges towards the true solution.

It is very well recognized that the eigenvalues of a saddle point matrix $\mathcal{A}$ depend strongly on the eigenvalues of the individual blocks A, B, C.[2] Clearly the eigenvalues of the individual matrices within these blocks depend on the mesh-size and time-step used, as well as all the other parameters describing the PDE and the objective function. [To give one example, the eigenvalues of K are contained within $[c_1h^d, c_2h^{d-2}]$ for constants c_1, c_2, where a constant mesh-size h is taken.] As a result, for our problem, the eigenvalues of $\mathcal{A}$ depend on these problem parameters. The convergence of an iterative method

[2] For illustrative purposes, a fundamental result [36] is as follows: if A is symmetric positive definite, B is full rank, and $C = 0$, the eigenvalues of $\mathcal{A}$ are contained within the intervals

$$\lambda(\mathcal{A}) \in \left[\frac{1}{2}\left(\mu_m - \sqrt{\mu_m^2 + 4\sigma_1^2}\right), \frac{1}{2}\left(\mu_1 - \sqrt{\mu_1^2 + 4\sigma_n^2}\right)\right] \cup \left[\mu_m, \frac{1}{2}\left(\mu_1 + \sqrt{\mu_1^2 + 4\sigma_1^2}\right)\right],$$

where μ_1, μ_m are the largest and smallest eigenvalues of A, and σ_1, σ_n denote the largest and smallest singular values of B.

applied directly to $\mathcal{A}$ can therefore be prohibitively slow, especially for large problems where h and τ are small. Our goal is therefore to find a preconditioning matrix $\mathcal{P}$ such that we can solve the equivalent preconditioned system

$$\mathcal{P}^{-1}\mathcal{A}\mathbf{x} = \mathcal{P}^{-1}\mathbf{b},$$

and $\mathcal{P}$ captures the properties of $\mathcal{A}$ well. If this can be achieved, the dependence of eigenvalues of $\mathcal{P}^{-1}\mathcal{A}$ on the problem parameters can be mitigated, and in the best case removed.

For a saddle point problem of the form (4.1), this is typically achieved by preconditioners of the form

$$\mathcal{P} = \begin{bmatrix} \widetilde{A} & 0 \\ 0 & \widetilde{S} \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} \widetilde{A} & 0 \\ B & -\widetilde{S} \end{bmatrix}, \tag{4.2}$$

where $\widetilde{A}$ approximates the $(1,1)$-block A of the saddle point matrix $\mathcal{A}$, and $\widetilde{S}$ approximates the (negative) Schur complement $S := C + BA^{-1}B^T$. This is motivated by results obtained in [22,26] where it is shown that the exact preconditioners $\widetilde{A} = A$ and $\widetilde{S} = S$ lead to a very small number of eigenvalues for the preconditioned system, and hence iteration numbers. The choice of the outer Krylov subspace solver typically depends on the nature of the system matrix and the preconditioner. For symmetric indefinite systems such as the ones presented here we usually choose MINRES [31] based on a three-term recurrence relation. However as MINRES typically requires a symmetric positive definite preconditioner, in the case of an indefinite preconditioner $\mathcal{P}$ we cannot use this method. We then need to apply a nonsymmetric solver of which there exist many, and it is not obvious which of them is best suited to any particular problem. Our rule-of-thumb is that if one carefully designs a preconditioner such that the eigenvalues of the preconditioned system are tightly clustered (or are contained within a small number of clusters), many different solvers perform in a fairly similar way. For simplicity we here choose BICG [10], which is the extension of CG [17] to nonsymmetric problems and is based on the nonsymmetric Lanczos process [15].

4.1. *GM*1 model

We now wish to derive preconditioners for all of the above linear systems. When examining the *GM*1 model using a Newton method the matrix $\mathcal{A}$ is written in the form of the saddle point system (4.1), with

$$A = \begin{bmatrix} \mathbf{A}_{u,GM1} & -2\tau r\mathbf{M}_{up/v^2} & -\tau\mathbf{M}_p & 0 \\ -2\tau r\mathbf{M}_{up/v^2} & \mathbf{A}_{v,GM1} & 0 & -\tau\mathbf{M}_q \\ -\tau\mathbf{M}_p & 0 & \tau\nu_1\mathbf{M} & 0 \\ 0 & -\tau\mathbf{M}_q & 0 & \tau\nu_2\mathbf{M} \end{bmatrix},$$

$$B = \begin{bmatrix} -\mathbf{L}_{u,GM1} & -\tau r\mathbf{M}_{u^2/v^2} & -\tau\mathbf{M}_u & 0 \\ 2\tau r\mathbf{M}_u & -\mathbf{L}_{v,GM1} & 0 & -\tau\mathbf{M}_v \end{bmatrix}, \qquad C = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Consider first approximating the matrix A. We observe that its block structure means we can also write it in saddle point type form. The matrix blkdiag$(\tau\nu_1\mathbf{M}, \tau\nu_2\mathbf{M})$ is comparatively straightforward to work with, as it is a block diagonal matrix consisting solely of mass matrices, so we may devise saddle point approximations with this as the leading block, i.e.,

$$\widetilde{A} = \begin{bmatrix} \widetilde{\mathbf{A}}_1 & 0 & 0 & 0 \\ 0 & \widetilde{\mathbf{A}}_2 & 0 & 0 \\ 0 & 0 & \tau\nu_1\mathbf{M} & 0 \\ 0 & 0 & 0 & \tau\nu_2\mathbf{M} \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} -\widetilde{\mathbf{A}}_1 & 0 & 0 & 0 \\ 0 & -\widetilde{\mathbf{A}}_2 & 0 & 0 \\ -\tau\mathbf{M}_p & 0 & \tau\nu_1\mathbf{M} & 0 \\ 0 & -\tau\mathbf{M}_q & 0 & \tau\nu_2\mathbf{M} \end{bmatrix}, \tag{4.3}$$

for suitable choices of $\widetilde{\mathbf{A}}_1$ and $\widetilde{\mathbf{A}}_2$. To make these selections, we seek to approximate the Schur complement of A, that is:

$$\begin{bmatrix} \widetilde{\mathbf{A}}_1 & 0 \\ 0 & \widetilde{\mathbf{A}}_2 \end{bmatrix} \approx \underbrace{\begin{bmatrix} \mathbf{A}_{u,GM1} & -2\tau r\mathbf{M}_{up/v^2} \\ -2\tau r\mathbf{M}_{up/v^2} & \mathbf{A}_{v,GM1} \end{bmatrix}}_{=:\widetilde{\mathbf{A}}_{(1,2)}} - \begin{bmatrix} \tau\nu_1^{-1}\mathbf{M}_p\mathbf{M}^{-1}\mathbf{M}_p & 0 \\ 0 & \tau\nu_2^{-1}\mathbf{M}_q\mathbf{M}^{-1}\mathbf{M}_q \end{bmatrix}.$$

We then utilize the heuristic of approximating $\widetilde{\mathbf{A}}_{(1,2)}$ by its own (block diagonal) saddle point approximation, leading to the following candidates for $\widetilde{\mathbf{A}}_1$ and $\widetilde{\mathbf{A}}_2$:

$$\widetilde{\mathbf{A}}_1 = \mathbf{A}_{u,GM1} - (2\tau r)^2\mathbf{M}_{up/v^2}\mathbf{A}_{v,GM1}^{-1}\mathbf{M}_{up/v^2} - \tau\nu_1^{-1}\mathbf{M}_p\mathbf{M}^{-1}\mathbf{M}_p,$$
$$\widetilde{\mathbf{A}}_2 = \mathbf{A}_{v,GM1} - \tau\nu_2^{-1}\mathbf{M}_q\mathbf{M}^{-1}\mathbf{M}_q,$$

which we apply within (4.3). Within our implementation we replace the matrix $\mathbf{M}^{-1}$ with diagonal approximations, in order for the application of our solver to be computationally feasible.

Turning our attention now to the Schur complement of the entire matrix $\mathcal{A}$:

$$S = \begin{bmatrix} -\mathbf{L}_{u,GM1} & -\tau r \mathbf{M}_{u^2/v^2} \\ 2\tau r \mathbf{M}_u & -\mathbf{L}_{v,GM1} \end{bmatrix} \widetilde{\mathbf{A}}_{(1,2)}^{-1} \begin{bmatrix} -\mathbf{L}_{u,GM1}^T & 2\tau r \mathbf{M}_u \\ -\tau r \mathbf{M}_{u^2/v^2} & -\mathbf{L}_{v,GM1}^T \end{bmatrix} + \tau \begin{bmatrix} \nu_1^{-1} \mathbf{M}_u \mathbf{M}^{-1} \mathbf{M}_u & 0 \\ 0 & \nu_2^{-1} \mathbf{M}_v \mathbf{M}^{-1} \mathbf{M}_v \end{bmatrix},$$

we construct an approximation

$$\widetilde{S} = \begin{bmatrix} -\mathbf{L}_{u,GM1} + \widehat{\mathbf{M}}_1^{(1)} & \tau r \mathbf{M}_{u^2/v^2} \\ -2\tau r \mathbf{M}_u & -\mathbf{L}_{v,GM1} + \widehat{\mathbf{M}}_2^{(1)} \end{bmatrix} \widetilde{\mathbf{A}}_{(1,2)}^{-1} \begin{bmatrix} -\mathbf{L}_{u,GM1}^T + \widehat{\mathbf{M}}_1^{(2)} & -2\tau r \mathbf{M}_u \\ \tau r \mathbf{M}_{u^2/v^2} & -\mathbf{L}_{v,GM1}^T + \widehat{\mathbf{M}}_2^{(2)} \end{bmatrix},$$

for suitably chosen matrices $\widehat{\mathbf{M}}_1^{(1)}$, $\widehat{\mathbf{M}}_2^{(1)}$, $\widehat{\mathbf{M}}_1^{(2)}$ and $\widehat{\mathbf{M}}_2^{(2)}$. To do this we apply a 'matching strategy', where we seek an additional (outer) term of the Schur complement approximation to match the second term of the exact Schur complement,[3] as follows:

$$\begin{bmatrix} \widehat{\mathbf{M}}_1^{(1)} & 0 \\ 0 & \widehat{\mathbf{M}}_2^{(1)} \end{bmatrix} \widetilde{\mathbf{A}}_{(1,2)}^{-1} \begin{bmatrix} \widehat{\mathbf{M}}_1^{(2)} & 0 \\ 0 & \widehat{\mathbf{M}}_2^{(2)} \end{bmatrix} \approx \tau \begin{bmatrix} \nu_1^{-1} \mathbf{M}_u \mathbf{M}^{-1} \mathbf{M}_u & 0 \\ 0 & \nu_2^{-1} \mathbf{M}_v \mathbf{M}^{-1} \mathbf{M}_v \end{bmatrix}.$$

By examining the diagonal blocks of $\widetilde{\mathbf{A}}_{(1,2)}^{-1}$, we see that this strategy motivates the approximations:

$$\widehat{\mathbf{M}}_1^{(1)} \widetilde{\mathbf{A}}_1^{-1} \widehat{\mathbf{M}}_1^{(2)} \approx \frac{\tau}{\nu_1} \mathbf{M}_u \mathbf{M}^{-1} \mathbf{M}_u, \qquad \widetilde{\mathbf{A}}_1 := \mathbf{A}_{u,GM1} - (2\tau r)^2 \mathbf{M}_{up/v^2} \mathbf{A}_{v,GM1}^{-1} \mathbf{M}_{up/v^2},$$
$$\widehat{\mathbf{M}}_2^{(1)} \widetilde{\mathbf{A}}_2^{-1} \widehat{\mathbf{M}}_2^{(2)} \approx \frac{\tau}{\nu_2} \mathbf{M}_v \mathbf{M}^{-1} \mathbf{M}_v, \qquad \widetilde{\mathbf{A}}_2 := \mathbf{A}_{v,GM1} - (2\tau r)^2 \mathbf{M}_{up/v^2} \mathbf{A}_{u,GM1}^{-1} \mathbf{M}_{up/v^2}.$$

To achieve such an approximation, we again recommend selecting diagonal matrices $\widehat{\mathbf{M}}_1^{(1)}$, $\widehat{\mathbf{M}}_2^{(1)}$, $\widehat{\mathbf{M}}_1^{(2)}$ and $\widehat{\mathbf{M}}_2^{(2)}$, with diagonal entries given by

$$\left[\widehat{\mathbf{M}}_1^{(1)}\right]_{jj} = \sqrt{\frac{\tau}{\nu_1}} \, [\mathbf{M}_u]_{jj} \cdot [\mathbf{M}]_{jj}^{-1/2} \cdot \left|\left[\widetilde{\mathbf{A}}_1\right]_{jj}\right|, \qquad \left[\widehat{\mathbf{M}}_1^{(2)}\right]_{jj} = \sqrt{\frac{\tau}{\nu_1}} \, [\mathbf{M}]_{jj}^{-1/2} \cdot [\mathbf{M}_u]_{jj},$$
$$\left[\widehat{\mathbf{M}}_2^{(1)}\right]_{jj} = \sqrt{\frac{\tau}{\nu_2}} \, [\mathbf{M}_v]_{jj} \cdot [\mathbf{M}]_{jj}^{-1/2} \cdot \left|\left[\widetilde{\mathbf{A}}_2\right]_{jj}\right|, \qquad \left[\widehat{\mathbf{M}}_2^{(2)}\right]_{jj} = \sqrt{\frac{\tau}{\nu_2}} \, [\mathbf{M}]_{jj}^{-1/2} \cdot [\mathbf{M}_v]_{jj}.$$

Now, for any practical method, we are only interested in the inverse of the Schur complement approximation $\widetilde{S}$. We therefore evaluate the inverse of the first and last block using a fixed number of steps of an Uzawa method [37] with preconditioners

$$\text{blkdiag}\left(\left(-\mathbf{L}_{u,GM1} + \widehat{\mathbf{M}}_1^{(1)}\right)_{\text{AMG}}, \left(-\mathbf{L}_{v,GM1} + \widehat{\mathbf{M}}_2^{(1)}\right)_{\text{AMG}} \right) \quad \text{or}$$
$$\text{blkdiag}\left(\left(-\mathbf{L}_{u,GM1}^T + \widehat{\mathbf{M}}_1^{(2)}\right)_{\text{AMG}}, \left(-\mathbf{L}_{v,GM1}^T + \widehat{\mathbf{M}}_2^{(2)}\right)_{\text{AMG}} \right),$$

where $(\cdot)_{\text{AMG}}$ denotes the application of an algebraic multigrid method to the relevant matrix.

For the Gauss–Newton case the derivation of the preconditioners is more straightforward. The approximation of the Hessian is typically not as good as in the Newton setting but the Gauss–Newton matrices are easier to handle from a preconditioning viewpoint. To approximate A we write

$$\begin{bmatrix} \beta_1 \tau \mathbf{M} & 0 & 0 & 0 \\ 0 & \beta_2 \tau \mathbf{M} & 0 & 0 \\ 0 & 0 & \nu_1 \tau \mathbf{M} & 0 \\ 0 & 0 & 0 & \nu_2 \tau \mathbf{M} \end{bmatrix} \approx \begin{bmatrix} \beta_1 \tau \widetilde{\mathbf{M}} & 0 & 0 & 0 \\ 0 & \beta_2 \tau \widetilde{\mathbf{M}} & 0 & 0 \\ 0 & 0 & \nu_1 \tau \widetilde{\mathbf{M}} & 0 \\ 0 & 0 & 0 & \nu_2 \tau \widetilde{\mathbf{M}} \end{bmatrix} =: \widetilde{A},$$

where $\widetilde{\mathbf{M}}$ is equal to $\mathbf{M}$ for (diagonal) lumped mass matrices. If consistent mass matrices are used instead, some approximation such as the application of Chebyshev semi-iteration [43] is chosen. The inverse of the Schur complement approximation

$$\widetilde{S} := \begin{bmatrix} -\mathbf{L}_{u,GM1} + \widehat{\mathbf{M}}_1^{(1)} & \tau r \mathbf{M}_{u^2/v^2} \\ -2\tau r \mathbf{M}_u & -\mathbf{L}_{v,GM1} + \widehat{\mathbf{M}}_2^{(1)} \end{bmatrix} \widetilde{\mathbf{A}}_{(1,2)}^{-1} \begin{bmatrix} -\mathbf{L}_{u,GM1}^T + \widehat{\mathbf{M}}_1^{(2)} & -2\tau r \mathbf{M}_u \\ \tau r \mathbf{M}_{u^2/v^2} & -\mathbf{L}_{v,GM1}^T + \widehat{\mathbf{M}}_2^{(2)} \end{bmatrix},$$

[3] This matching strategy was originally developed by the authors [33–35] to generate approximations for more fundamental PDE-constrained optimization problems. Considering the Poisson control problem, for instance, the Schur complement and its approximation take the form [34]:

$$S = KM^{-1}K + \frac{1}{\beta} M, \qquad \widetilde{S} = \left(K + \frac{1}{\sqrt{\beta}} M\right) M^{-1} \left(K + \frac{1}{\sqrt{\beta}} M\right),$$

whereupon it can be shown that the eigenvalues of $\widetilde{S}^{-1}S$ are contained within $[\frac{1}{2}, 1]$, independently of problem parameters and matrix dimension. This bound can be proved due to the comparatively simple structures of the matrices involved, and thus cannot be replicated for the complex systems studied here, however we have found this strategy to be very effective for a range of PDE-constrained optimization problems.

where here $\widetilde{\mathbf{A}}_{(1,2)} = \text{blkdiag}(\beta_1\tau\mathbf{M}, \beta_2\tau\mathbf{M})$, $\widehat{\mathbf{M}}_1^{(1)} = \widehat{\mathbf{M}}_1^{(2)} = \tau\sqrt{\frac{\beta_1}{\nu_1}}\,\mathbf{M}_u$, and $\widehat{\mathbf{M}}_2^{(1)} = \widehat{\mathbf{M}}_2^{(2)} = \tau\sqrt{\frac{\beta_2}{\nu_2}}\,\mathbf{M}_v$, is applied at each step of our iterative method.

4.2. GM2 model

In a completely analogous way we can derive saddle point preconditioners for the *GM*2 model, for which

$$A = \begin{bmatrix} \mathbf{A}_{u,GM2} & -2\tau\gamma\mathbf{M}_{u(q-p)} & 0 & 0 \\ -2\tau\gamma\mathbf{M}_{u(q-p)} & \mathbf{A}_{v,GM2} & 0 & 0 \\ 0 & 0 & \tau\nu_1\mathbf{M} & 0 \\ 0 & 0 & 0 & \tau\nu_2\mathbf{M} \end{bmatrix},$$

$$B = \begin{bmatrix} -\mathbf{L}_{u,GM2} & \tau\gamma\mathbf{M}_{u^2} & \tau\gamma\mathbf{M} & 0 \\ -2\tau\gamma\mathbf{M}_{uv} & -\mathbf{L}_{v,GM2} & 0 & \tau\gamma\mathbf{M} \end{bmatrix}, \qquad C = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix},$$

in the notation of (4.1).

We may approximate the matrix A, in this case using the saddle point type structure of its upper sub-matrix, by

$$\widetilde{A} = \begin{bmatrix} \widetilde{\mathbf{A}}_1 & 0 & 0 & 0 \\ 0 & \mathbf{A}_{v,GM2} & 0 & 0 \\ 0 & 0 & \tau\nu_1\mathbf{M} & 0 \\ 0 & 0 & 0 & \tau\nu_2\mathbf{M} \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} -\widetilde{\mathbf{A}}_1 & 0 & 0 & 0 \\ -2\tau\gamma\mathbf{M}_{u(q-p)} & \mathbf{A}_{v,GM2} & 0 & 0 \\ 0 & 0 & \tau\nu_1\mathbf{M} & 0 \\ 0 & 0 & 0 & \tau\nu_2\mathbf{M} \end{bmatrix},$$

where

$$\widetilde{\mathbf{A}}_1 = \mathbf{A}_{u,GM2} - (2\tau\gamma)^2\mathbf{M}_{u(q-p)}\mathbf{A}_{v,GM2}^{-1}\mathbf{M}_{u(q-p)}.$$

We follow a similar strategy as before to approximate the Schur complement

$$S = \begin{bmatrix} -\mathbf{L}_{u,GM2} & \tau\gamma\mathbf{M}_{u^2} \\ -2\tau\gamma\mathbf{M}_{uv} & -\mathbf{L}_{v,GM2} \end{bmatrix}\widetilde{\mathbf{A}}_{(1,2)}^{-1}\begin{bmatrix} -\mathbf{L}_{u,GM2}^T & -2\tau\gamma\mathbf{M}_{uv} \\ \tau\gamma\mathbf{M}_{u^2} & -\mathbf{L}_{v,GM2}^T \end{bmatrix} + \tau\gamma^2\begin{bmatrix} \nu_1^{-1}\mathbf{M} & 0 \\ 0 & \nu_2^{-1}\mathbf{M} \end{bmatrix},$$

where for this problem

$$\widetilde{\mathbf{A}}_{(1,2)} := \begin{bmatrix} \mathbf{A}_{u,GM2} & -2\tau\gamma\mathbf{M}_{u(q-p)} \\ -2\tau\gamma\mathbf{M}_{u(q-p)} & \mathbf{A}_{v,GM2} \end{bmatrix}.$$

Again applying our matching strategy, we obtain the following approximation:

$$\widetilde{S} = \begin{bmatrix} -\mathbf{L}_{u,GM2} + \widehat{\mathbf{M}}_1^{(1)} & -\tau\gamma\mathbf{M}_{u^2} \\ 2\tau\gamma\mathbf{M}_{uv} & -\mathbf{L}_{v,GM2} + \widehat{\mathbf{M}}_2^{(1)} \end{bmatrix}\widetilde{\mathbf{A}}_{(1,2)}^{-1}\begin{bmatrix} -\mathbf{L}_{u,GM2}^T + \widehat{\mathbf{M}}_1^{(2)} & 2\tau\gamma\mathbf{M}_{uv} \\ -\tau\gamma\mathbf{M}_{u^2} & -\mathbf{L}_{v,GM2}^T + \widehat{\mathbf{M}}_2^{(2)} \end{bmatrix}.$$

Examining the diagonal blocks of $\widetilde{\mathbf{A}}_{(1,2)}^{-1}$ (as for the *GM*1 model), and applying a matching strategy to approximate the second term of S, we motivate the following approximations:

$$\widehat{\mathbf{M}}_1^{(1)}\widetilde{\mathbf{A}}_1^{-1}\widehat{\mathbf{M}}_1^{(2)} \approx \frac{\tau\gamma^2}{\nu_1}\mathbf{M}, \qquad \widetilde{\mathbf{A}}_1 := \mathbf{A}_{u,GM2} - (2\tau\gamma)^2\mathbf{M}_{u(q-p)}\mathbf{A}_{v,GM2}^{-1}\mathbf{M}_{u(q-p)},$$

$$\widehat{\mathbf{M}}_2^{(1)}\widetilde{\mathbf{A}}_2^{-1}\widehat{\mathbf{M}}_2^{(2)} \approx \frac{\tau\gamma^2}{\nu_2}\mathbf{M}, \qquad \widetilde{\mathbf{A}}_2 := \mathbf{A}_{v,GM2} - (2\tau\gamma)^2\mathbf{M}_{u(q-p)}\mathbf{A}_{u,GM2}^{-1}\mathbf{M}_{u(q-p)}.$$

These approximations may be achieved by constructing diagonal matrices $\widehat{\mathbf{M}}_1^{(1)}$, $\widehat{\mathbf{M}}_2^{(1)}$, $\widehat{\mathbf{M}}_1^{(2)}$, $\widehat{\mathbf{M}}_2^{(2)}$ with diagonal entries given by

$$\left[\widehat{\mathbf{M}}_1^{(1)}\right]_{jj} = \sqrt{\frac{\tau}{\nu_1}}\,\gamma\,[\mathbf{M}]_{jj}^{1/2}\cdot\left|\left[\widetilde{\mathbf{A}}_1\right]_{jj}\right|, \qquad \left[\widehat{\mathbf{M}}_1^{(2)}\right]_{jj} = \sqrt{\frac{\tau}{\nu_1}}\,\gamma\,[\mathbf{M}]_{jj}^{1/2},$$

$$\left[\widehat{\mathbf{M}}_2^{(1)}\right]_{jj} = \sqrt{\frac{\tau}{\nu_2}}\,\gamma\,[\mathbf{M}]_{jj}^{1/2}\cdot\left|\left[\widetilde{\mathbf{A}}_2\right]_{jj}\right|, \qquad \left[\widehat{\mathbf{M}}_2^{(2)}\right]_{jj} = \sqrt{\frac{\tau}{\nu_2}}\,\gamma\,[\mathbf{M}]_{jj}^{1/2}.$$

We can again build these choices of $\widehat{\mathbf{M}}_1^{(1)}$, $\widehat{\mathbf{M}}_2^{(1)}$, $\widehat{\mathbf{M}}_1^{(2)}$, $\widehat{\mathbf{M}}_2^{(2)}$ into the approximation $\widetilde{S}$ within our preconditioner.

For each of our suggested iterative methods, we insert our approximations of A and $S = BA^{-1}B^T$ into the general preconditioners for saddle point systems stated in (4.2).

4.3. Computational cost

When implementing our preconditioned iterative methods, the vast majority of the computational expense occurs when applying the inverse of our preconditioners. We therefore now wish to detail the solves that we are required to carry out when applying our preconditioner to each problem.

To enact our preconditioner for the *GM*1 model, we are required to perform the following:

- Solves for $\widetilde{\mathbf{A}}_1$ and $\widetilde{\mathbf{A}}_2$ ('mass-like' matrices),
- Chebyshev semi-iteration/diagonal solves for $\tau\nu_1\mathbf{M}$, $\tau\nu_2\mathbf{M}$ (mass matrices),
- 1 multigrid per Uzawa iteration for each of $-\mathbf{L}_{u,GM1}+\widehat{\mathbf{M}}_1^{(1)}$, $-\mathbf{L}_{v,GM1}+\widehat{\mathbf{M}}_2^{(1)}$, $-\mathbf{L}_{u,GM1}^T+\widehat{\mathbf{M}}_1^{(2)}$, $-\mathbf{L}_{v,GM1}^T+\widehat{\mathbf{M}}_2^{(2)}$ (to apply $\widetilde{S}^{-1}$).

From a computational point-of-view, the most straightforward operations involve inverting mass matrices, with the multigrid operations the most expensive.

For the Gauss–Newton approach for the same problem, the preconditioner is cheaper to apply, with the following operations dominating the computational cost:

- Chebyshev semi-iteration/diagonal solves for $\beta_1\tau\mathbf{M}$, $\beta_2\tau\mathbf{M}$, $\tau\nu_1\mathbf{M}$, $\tau\nu_2\mathbf{M}$ (mass matrices),
- 1 multigrid per Uzawa iteration for each of $-\mathbf{L}_{u,GM1}+\widehat{\mathbf{M}}_1^{(1)}$, $-\mathbf{L}_{v,GM1}+\widehat{\mathbf{M}}_2^{(1)}$, $-\mathbf{L}_{u,GM1}^T+\widehat{\mathbf{M}}_1^{(2)}$, $-\mathbf{L}_{v,GM1}^T+\widehat{\mathbf{M}}_2^{(2)}$ (for the new choices of $\widehat{\mathbf{M}}_1^{(1)}$, $\widehat{\mathbf{M}}_2^{(1)}$, $\widehat{\mathbf{M}}_1^{(2)}$, $\widehat{\mathbf{M}}_2^{(2)}$).

Further, the dominant computational operations for solving the *GM*2 model are:

- Solves for $\widetilde{\mathbf{A}}_1$, $\mathbf{A}_{u,GM2}$ ('mass-like' matrices),
- Chebyshev semi-iteration/diagonal solves for $\tau\nu_1\mathbf{M}$, $\tau\nu_2\mathbf{M}$ (mass matrices),
- 1 multigrid per Uzawa iteration for each of $-\mathbf{L}_{u,GM1}+\widehat{\mathbf{M}}_1^{(1)}$, $-\mathbf{L}_{v,GM1}+\widehat{\mathbf{M}}_2^{(1)}$, $-\mathbf{L}_{u,GM1}^T+\widehat{\mathbf{M}}_1^{(2)}$, $-\mathbf{L}_{v,GM1}^T+\widehat{\mathbf{M}}_2^{(2)}$ (to apply $\widetilde{S}^{-1}$).

We believe the amount of computational work required to apply our preconditioner is satisfactory when taking into account the complex structure and large dimension of the matrix systems.

4.4. Alternative methods

Before presenting numerical results we wish to briefly discuss alternative approaches for the solution of the optimization problem, or the linear systems at the heart of the nonlinear solver. Due to the highly complex structure of the problem statements and associated matrix systems, we are not aware of any robust methods that have previously been developed for solving these systems. We believe that this underlines the value of investigating preconditioned iterative methods for this important class of problems. However we wish to outline other classes of methods which could potentially be applied to these problems, in many cases building on the work described in this article.

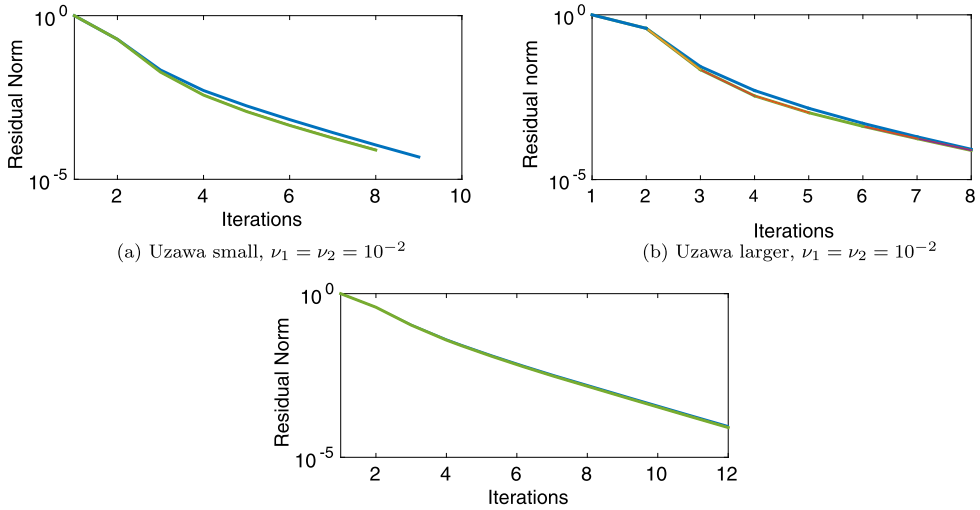
The method we have presented is applied when using either a Newton or a Gauss–Newton approach for the nonlinear program. Alternatively, we could apply a simple gradient descent coupled with a line search procedure [29], which typically converges very slowly. Another alternative would be to employ an interior point scheme [44], which also requires the solution of saddle point problems, and we believe that many of our proposed techniques could be carried over to this case. In [21] the authors follow a so-called one-shot method that can be viewed as a stationary iteration of the form $\mathbf{x}_{k+1}=\mathbf{x}_k+\mathcal{P}^{-1}\mathbf{r}_k$. The preconditioner in this case is given by a block matrix that requires the (approximate) inversion of the adjoint and forward PDE operator, as well as the solution of a complicated Schur complement system (here written for the Gauss–Newton system (3.2) with the block containing second derivatives with respect to Lagrange multipliers equal to zero):

$$S_A \approx \begin{bmatrix} \widehat{\mathcal{L}}_{aa} & \\ & \widehat{\mathcal{L}}_{bb} \end{bmatrix} + \begin{bmatrix} \mathcal{L}_{ap} & \mathcal{L}_{aq} \\ \mathcal{L}_{bp} & \mathcal{L}_{bq} \end{bmatrix} \begin{bmatrix} \mathcal{L}_{up} & \mathcal{L}_{uq} \\ \mathcal{L}_{vp} & \mathcal{L}_{vq} \end{bmatrix}^{-1} \begin{bmatrix} \widehat{\mathcal{L}}_{uu} & \\ & \widehat{\mathcal{L}}_{vv} \end{bmatrix} \begin{bmatrix} \mathcal{L}_{pu} & \mathcal{L}_{pv} \\ \mathcal{L}_{qu} & \mathcal{L}_{qv} \end{bmatrix}^{-1} \begin{bmatrix} \mathcal{L}_{pa} & \mathcal{L}_{pb} \\ \mathcal{L}_{qa} & \mathcal{L}_{qb} \end{bmatrix}.$$

Note that this is in general harder to approximate than the Schur complements obtained using our approach, as we here have the sum of mass matrices and inverse PDE operators within the Schur complement approximation. As it is the action of S_A^{-1} that is important for preconditioning purposes, this approximation is extremely difficult to apply in practice.

Recently, operator preconditioning approaches have proven successful for many PDE preconditioning problems (see [25, 45]). Their use for nonlinear problems has recently attracted attention within the field [1,38], and we are currently investigating how to extend these approaches to the reaction–diffusion type setting.

Of the approaches currently within reach, the stationary iteration approaches were found not to converge, and the above approximation S_A is of a very complex nature and is infeasible to apply. The only alternative approach which we found



(a) Uzawa small, $\nu_1 = \nu_2 = 10^{-2}$ (b) Uzawa larger, $\nu_1 = \nu_2 = 10^{-2}$

(c) Uzawa larger, $\nu_1 = \nu_2 = 10^{-4}$

Fig. 4.1. Iteration numbers for Uzawa scheme with a block triangular version of our preconditioner. Plots are shown for every SQP step, however they can often not be distinguished due to very similar convergence behavior. The smaller example uses 729 degrees of freedom in space, and the larger example 4096.

to generate sensible results for a range of parameter regimes is that of an Uzawa method, using a preconditioner of the form derived in this paper. In Fig. 4.1 we show the iteration numbers that are required using such an approach for two different meshes and two regularization parameters. We note that, while the results show that this method performs well, we are required to move from the block diagonal preconditioner derived in this section to a more expensive block triangular preconditioner, as the Uzawa method diverges when a block diagonal approach is taken. Apart from this change, the major computational operations required per Uzawa iteration are largely similar to those required to apply our preconditioner with a single Uzawa step. We also highlight that this method is itself only feasible due to the preconditioners derived in this paper.

5. Numerical results

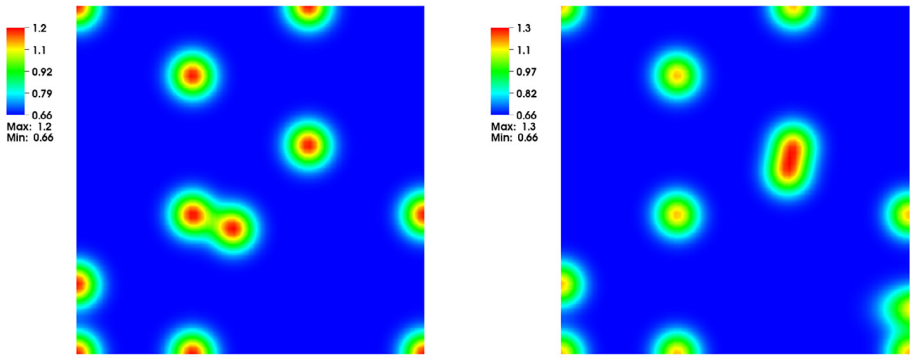
We now wish to apply our methodology to a number of test problems. All results presented in this section are based on an implementation of the presented algorithms and (block diagonal) preconditioners within the deal.II [3] framework using $Q1$ finite elements. The AMG preconditioner we use is part of the Trilinos ML package [13] that implements a smoothed aggregation AMG. Within the algebraic multigrid routine we typically apply 10 steps of a Chebyshev smoother in combination with the application of two V-cycles. Typically we apply 4 iterations of the Uzawa scheme within our Schur complement approximation, to guarantee high accuracy. For our implementation of BiCG we use a stopping tolerance of 10^{-4}. Our experiments are performed for $T = 1$ and $\tau = 0.05$, i.e. 20 time-steps. Typically, the spatial domain Ω is considered to be the unit square or cube. All results are performed on a Centos Linux machine with Intel(R) Xeon(R) CPU X5650 @ 2.67 GHz CPUs and 48 GB of RAM.

5.1. *GM2 model*

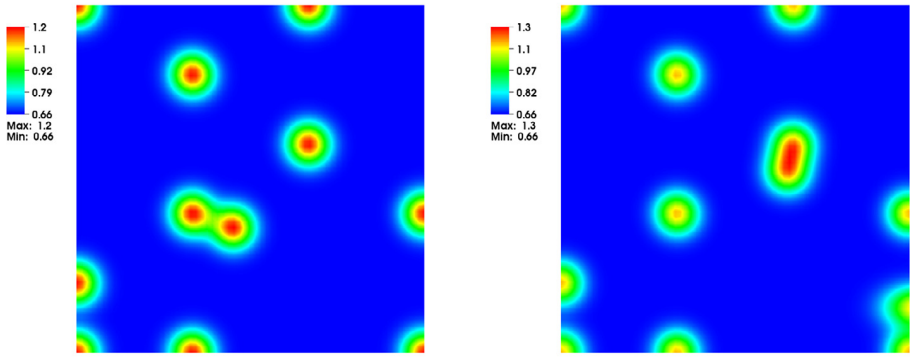
For both *GM2* and *GM1* models we start creating desired states using Gaussians placed at different positions in the unit square/cube that might depend on the time t. In Fig. 5.1 we illustrate two instances of the desired state and computed results for the *GM2* formulation, with the parameters set to $D_u = 1$, $D_v = 10$, $\beta_1 = \beta_2 = 1$, $\gamma = 50$, and $\nu_1 = \nu_2 = 10^{-6}$. As the regularization parameters become smaller we see that the desired and computed states are very close. This is reflected in the third set of images within Fig. 5.1, where the control is shown with sometimes rather high values. In Table 5.1 we present BiCG iteration numbers for solving this test problem for a range of degrees of freedom and regularization parameters – the results indicate that our solver is robust in a large number of problem setups.

5.2. *GM1 model with Newton and Gauss–Newton methods*

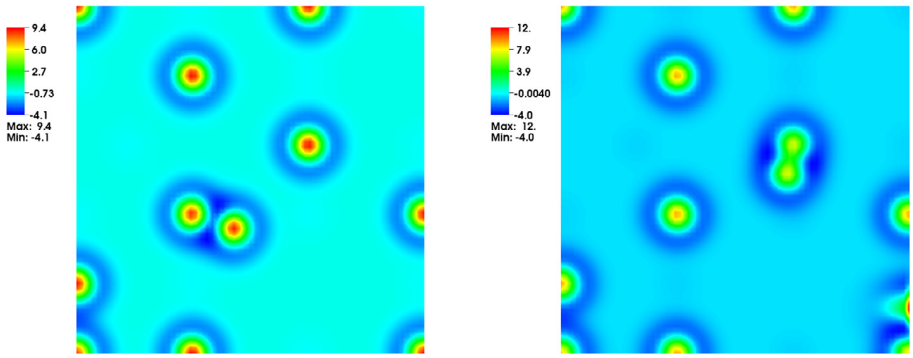
For the next problem we examine, the desired state for the *GM1* model is created using Gaussian functions placed in the unit cube. This is illustrated in Fig. 5.2, where we present the desired state for the first component, the computed first state variable, and the corresponding control variable. The parameters for this case are chosen to be $\beta_1 = \beta_2 = 10^2$, $\nu_1 = \nu_2 = 10^{-2}$, $D_u = 1$, $D_v = 10$, and $r = 10^{-2}$. For many interesting parameter setups (including for a range of values



(a) First Desired State $\widehat{u}_8$ (b) First Desired State $\widehat{u}_{12}$



(c) First Computed State u_8 (d) First Computed State u_{12}

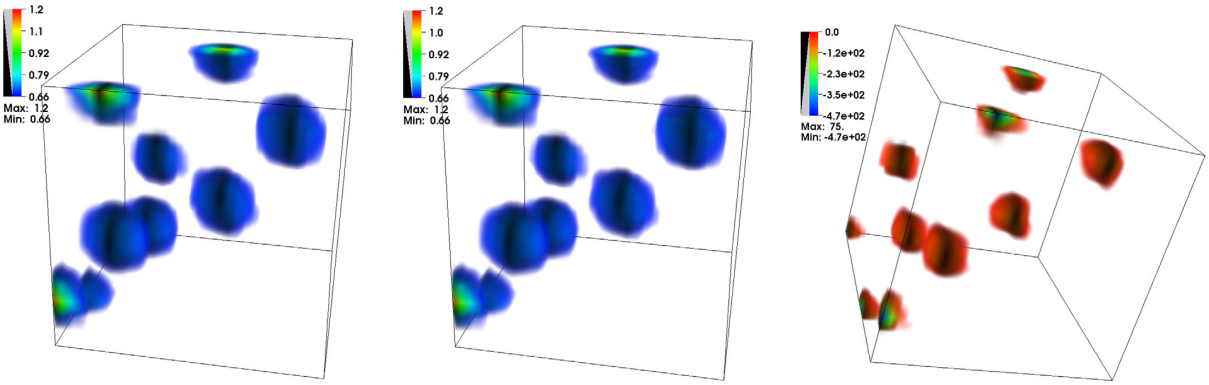


(e) First Computed Control a_8 (f) First Computed Control a_{12}

Fig. 5.1. Desired state for 8th and 12th grid points in time (upper two), computed state using the *GM2* model (middle two), and the computed control (lower two) for two reactants using the *GM2* model. The parameters are set to be $D_u = 1$, $D_v = 10$, $\beta_1 = \beta_2 = 1$, $\gamma = 50$, and $\nu_1 = \nu_2 = 10^{-6}$.

Table 5.1
Results on unit square with $D_u = 1$, $D_v = 10$, $\beta_1 = \beta_2 = 1$, and $\gamma = 50$. Stated are BiCG iteration numbers for each Newton step.

DoF	$\nu_1 = \nu_2 = 10^{-2}$		$\nu_1 = \nu_2 = 10^{-4}$		$\nu_1 = \nu_2 = 10^{-6}$	
	Newton	BiCG	Newton	BiCG	Newton	BiCG
507,000	step 1	18	step 1	16	step 1	16
	step 2	20	step 2	15	step 2	15
	step 3	20	step 3	15	step 3	15
	step 4	20	step 4	15	step 4	15
	step 5	20	step 5	15		
1,996,920	step 1	23	step 1	17	step 1	17
	step 2	23	step 2	18	step 2	16
	step 3	24	step 3	18	step 3	16
	step 4	23	step 4	18	step 4	16
	step 5	23	step 5	18		



(a) First Desired State (b) Computed First State (c) Computed Control

Fig. 5.2. Desired state, computed state and computed control for the first reactant in the *GM*1 model with parameters at $\beta_1 = \beta_2 = 10^2$, $\nu_1 = \nu_2 = 10^{-2}$, $D_u = 1$, $D_v = 10$, and $r = 10^{-2}$.

Table 5.2
Results on unit cube with $\beta_1 = \beta_2 = 10^2$, $D_u = 1$, $D_v = 10$, and $r = 10^{-2}$. We here vary the mesh-size and the regularization parameters ν_1 and ν_2. Stated are BICG iteration numbers for each Gauss–Newton step.

DoF	$\nu_1 = \nu_2 = 10^{-2}$		$\nu_1 = \nu_2 = 10^{-4}$		$\nu_1 = \nu_2 = 10^{-6}$	
	GN	BICG	GN	BICG	GN	BICG
87,480	step 1	11	step 1	11	step 1	9
	step 2	11	step 2	11	step 2	9
	step 3	11	step 3	11	step 3	9
	step 4	11	step 4	11		
	step 5	11	step 5	11		
589,560	step 1	11	step 1	13	step 1	11
	step 2	11	step 2	12	step 2	11
	step 3	11	step 3	12	step 3	11
	step 4	11	step 4	12	step 4	11
	step 5	11	step 5	12		
4,312,440	step 1	11	step 1	13	step 1	11
	step 2	11	step 2	12	step 2	11
	step 3	11	step 3	12	step 3	11
	step 4	11	step 4	12	step 4	11
	step 5	11	step 5	12		

of r) it is not trivial to find a configuration of the Newton scheme that demonstrates satisfying convergence properties. We instead focus on the Gauss–Newton method here, and we illustrate the BICG iteration numbers achieved for a range of problems in Table 5.2 – these results demonstrate much greater robustness, with rapid convergence of the inner solver.

As we have already highlighted, the complex structure of the linear systems makes the design of efficient preconditioners harder when the Newton scheme is applied compared to the Gauss–Newton scheme. We use the results presented in Table 5.3 to illustrate the performance of both Newton and Gauss–Newton schemes. We observe that the Newton method and our associated preconditioner perform well when the regularization parameters are chosen to be rather large. In this case, whereas the Gauss–Newton scheme generates low iteration numbers, it seems to generate less meaningful numerical results. Whereas, in the case of smaller regularization parameters, the Gauss–Newton scheme requires a greater number of outer iterations, the number of inner BICG iterations remains low. For this setup the preconditioner for the Newton scheme does not allow the method to converge, and therefore the scheme failed. This makes clear that careful choices concerning the outer iteration and preconditioner need to be made, in order to achieve good performance of the method for a particular parameter case.

Additionally, we illustrate in Table 5.4 the performance of the Gauss–Newton scheme when the tolerance of the inner solver is relaxed. We can see that for this mesh the choice of a tolerance decrease from 10^{-4} to 10^{-2} does not have a significant influence on the output of the optimization routine and the number of outer iterations.

We also wish to highlight that it is possible to include additional control constraints $\underline{a} \leq a \leq \bar{a}$ and $\underline{b} \leq b \leq \bar{b}$, to be enforced along with the systems of PDEs (2.2) or (2.3). Our approach to deal with these additional bounds is to include a Moreau–Yosida penalization [18] that can be used with a non-smooth Newton scheme. The structure of the Newton system is very similar to the one without control constraints, and we refer to [32] for more details on the derivation of the non-smooth Newton system and the choice of preconditioner. In Table 5.5 we present results for the setup $0 \leq a$ and $0 \leq b$, where the Gauss–Newton scheme is used in conjunction with BICG.

Table 5.3
Results on unit cube with $\beta_1 = \beta_2 = 1$, $D_u = 1$, $D_v = 10$, and $r = 10^{-4}$. We here vary the mesh-size and the regularization parameters ν_1 and ν_2. We show the iteration numbers for BICG, the value of the data misfit term $\|\mathbf{y} - \widehat{\mathbf{y}}\|^2$ in the objective function, and the relative change in the optimization variable between two consecutive Gauss–Newton iterations (GN_Δ). The outer iteration is stopped if GN_Δ is smaller than 10^{-4}.

DoF		Newton	GN	GN
87,480		$\nu_1 = \nu_2 = 10^2$ BICG/$\|\mathbf{y} - \widehat{\mathbf{y}}\|^2$/$GN_\Delta$	$\nu_1 = \nu_2 = 10^2$ BICG/$\|\mathbf{y} - \widehat{\mathbf{y}}\|^2$/$GN_\Delta$	$\nu_1 = \nu_2 = 10^{-3}$ BICG/$\|\mathbf{y} - \widehat{\mathbf{y}}\|^2$/$GN_\Delta$
	step 1	2/4.67/–	2/38.53/–	2/38.53/–
	step 2	$5/4.27/7.4 \times 10^{-2}$	$7/38.45/3.6 \times 10^{-3}$	$16/5.1 \times 10^{-1}/9.8 \times 10^{-1}$
	step 3	$5/4.27/1.2 \times 10^{-2}$	$7/38.45/9.7 \times 10^{-4}$	$19/3.8 \times 10^{-2}/2.1 \times 10^{0}$
	step 4	$5/4.27/1.9 \times 10^{-3}$	$7/38.45/3.3 \times 10^{-6}$	$16/3.3 \times 10^{-2}/4.3 \times 10^{-1}$
	step 5	$5/4.27/2.2 \times 10^{-4}$		$15/3.3 \times 10^{-2}/2.7 \times 10^{-3}$
	step 6	$5/4.27/1.8 \times 10^{-5}$		$15/3.2 \times 10^{-2}/7.9 \times 10^{-5}$

Table 5.4
Results on unit cube with $\beta_1 = \beta_2 = 10^2$, $D_u = 1$, $D_v = 10$, and $r = 10^{-2}$. We here vary the mesh-size and the regularization parameters ν_1 and ν_2. We show the iteration numbers for BICG and the value of the objective function $\mathcal{J}(\cdot)$. The outer iteration was stopped if the relative difference between two consecutive iterates was smaller than 10^{-4}.

DoF		Newton	GN	GN
87,480		$\text{tol} = 10^{-4}$ BICG/$\mathcal{J}(\cdot)$	$\text{tol} = 10^{-2}$ BICG/$\mathcal{J}(\cdot)$	$\text{tol} = 10^{-1}$ BICG/$\mathcal{J}(\cdot)$
	step 1	2/1926.98	2/1926.98	2/1926.98
	step 2	11/13.57	7/13.58	3/12.63
	step 3	11/1.78	7/1.79	3/1.19
	step 4	11/2.20	5/2.20	3/2.30
	step 5	11/2.20	5/2.22	3/2.17
	step 6	11/2.20	5/2.22	3/2.18
	step 7		5/2.22	3/2.18
	step 8			3/2.18
	step 9			3/2.18

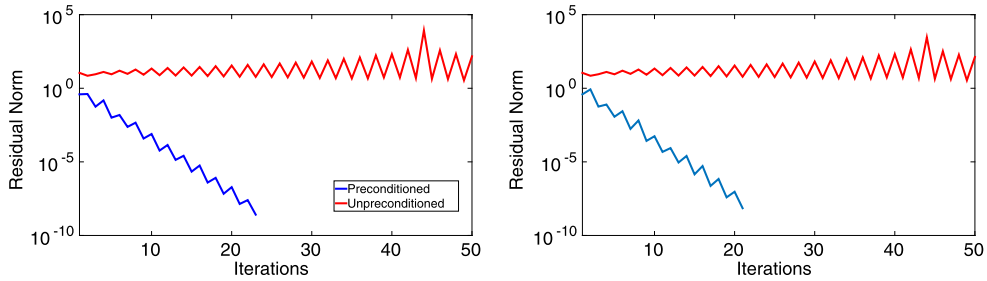
Table 5.5
Results on unit cube with $\beta_1 = \beta_2 = 10^2$, $D_u = 1$, $D_v = 10$, and $r = 10^{-2}$. We here vary the mesh-size and the regularization parameters ν_1 and ν_2. Stated are BICG iteration numbers for each Gauss–Newton step. The tolerance for the Gauss–Newton method is 10^{-2}.

DoF	$\nu_1 = \nu_2 = 10^{-2}$		$\nu_1 = \nu_2 = 10^{-4}$	
	GN	BICG	GN	BICG
130,680	step 1	2	step 1	2
	step 2	9	step 2	13
	step 3	13	step 3	14
507,000	step 1	4	step 1	4
	step 2	9	step 2	15
	step 3	10	step 3	15
1,996,920	step 1	10	step 1	10
	step 2	14	step 2	17
	step 3	14	step 3	17

We now wish to compare the performance of our preconditioned BICG approach with an unpreconditioned version of the same solver. We do so in order to demonstrate the clearly superior convergence properties of the preconditioned method, and show the very high accuracy to which our solver is able to solve the matrix systems. Fig. 5.3 illustrates the residual error of the unpreconditioned and preconditioned approaches for two test problems: it is clear that running the preconditioned method for only a few iterations easily outperforms the unpreconditioned version. Indeed the unpreconditioned method appears to diverge in many cases, demonstrating the need to construct effective preconditioners for the complex matrix systems involved.

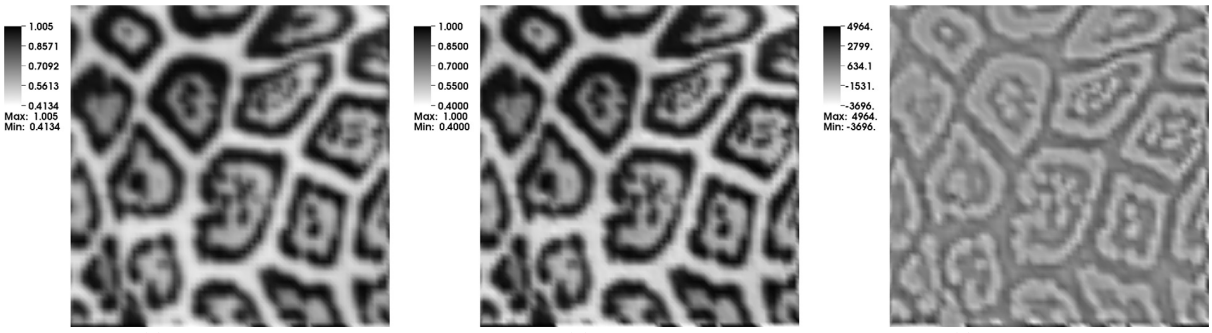
5.3. *Image-driven desired state and GM1 model*

An attractive feature of this methodology is that it is also possible to obtain desired states by reading in pattern information from an image. This may be done for the *GM*1 and *GM*2 models, whether or not control constraints are included. Image-driven parameter estimation techniques can also be found in [19]. For this problem, we choose to take an image similar to those used in [24] – this involves reading in a pattern found on a mature jaguar. As this problem is not necessarily time-dependent we wish to illustrate the performance of our method by scaling the desired pattern by τ_i, where i denotes



(a) $\nu_1 = \nu_2 = 10^{-2}$ (b) $\nu_1 = \nu_2 = 10^{-4}$

Fig. 5.3. Residual errors of unpreconditioned (red) and preconditioned (blue) BiCG method for two test problems (DoFs 4096). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) Computed State (b) Desired State (c) Computed Control

Fig. 5.4. Results for image-driven model: Shown are computed state, desired state, and computed control for the parameter setups using $\beta_1 = \beta_2 = 10^2$, $\nu_1 = \nu_2 = 10^{-7}$, $D_u = 1$, $D_v = 10$, and $r = 10^{-5}$.

the relevant index in time. The results for applying the Gauss–Newton scheme to this image-driven problem are shown in Fig. 5.4.

In Fig. 5.4(b) we show the desired state used, with the computed state presented in Fig. 5.4(a) and the associated control in Fig. 5.4(c).

The parameters for this setup are $\beta_1 = \beta_2 = 10^2$, $\nu_1 = \nu_2 = 10^{-7}$, $D_u = 1$, $D_v = 10$, and $r = 10^{-5}$. For the computations from which Fig. 5.4 is generated, a tolerance of 10^{-2} is taken for the Gauss–Newton scheme. Within these computations 8 steps of the Gauss–Newton iteration are required, with an average of 20.5 BiCG iterations per Gauss–Newton step.

Overall the numerical results presented for the above experiments indicate that we are able to solve a wide range of parameter identification problems from pattern formation, with our observed BiCG iteration numbers (as well as computation times) being low for a large number of parameter regimes. Furthermore, the iteration numbers behave in a fairly robust way as the parameters involved in the problem are varied.

6. Concluding remarks and future work

In this article, we have considered the development of preconditioned iterative methods for the numerical solution of parameter identification problems arising from pattern formation. We have constructed our methods using effective strategies for approximating the $(1,1)$-block and Schur complement of the saddle point systems that result from these problems.

The numerical results we have obtained when applying our techniques to a number of test examples (using both *GM*1 and *GM*2 models) indicate that our proposed solvers are effective ones for a wide range of parameter setups. Another key aspect of our methodology is that we are able to feed desired states (or "target patterns") into our implementation using experimental or computational data, and use this to obtain appropriate solutions to the Turing model in question. Furthermore, our solvers are found to be effective at handling additional inequality constraints on the control variables.

There are a number of related areas of research which we hope to consider, including the incorporation of additional constraints on the state or control variables (for instance integral constraints, or bounds on the state variables), different time-stepping schemes, and possibly different techniques for the outer iteration. We also wish to investigate a version of the problem where the L_2-distance between the states and desired states is only measured at the final time $t = T$ (i.e. where $\beta_1 = \beta_2 = 0$), as we find that such problems have considerable physical applicability. Furthermore, we now hope to tackle other problems of significant interest to the mathematical biology community using the methodology presented in this paper.

Acknowledgements

We would like to thank two anonymous referees for their helpful and insightful comments. The second author was supported for this work in part by the Engineering and Physical Sciences Research Council (EPSRC) Grant EP/P505216/1, and by an EPSRC Doctoral Prize (EP/K503113/1). The authors are grateful for the award of a European Science Foundation (ESF) Exchange Grant OPTPDE-3997 under the OPTPDE programme, and express their thanks to the Max Planck Institute in Magdeburg.

Appendix A. Derivation of the Newton systems

For the Gierer–Meinhardt ($GM1$) formulation, we examine the forward equations

$$\begin{aligned} u_t - D_u\Delta u - \frac{ru^2}{v} + au &= r, \quad \text{on } \Omega\times[0,T],\\ v_t - D_v\Delta v - ru^2 + bv &= 0, \quad \text{on } \Omega\times[0,T],\\ u(\mathbf{x},0) = u_0(\mathbf{x}), \quad v(\mathbf{x},0) &= v_0(\mathbf{x}), \quad \text{on } \Omega,\\ \frac{\partial u}{\partial \nu} = \frac{\partial v}{\partial \nu} &= 0, \quad \text{on } \partial\Omega\times[0,T], \end{aligned}$$

and the adjoint equations (see [11])

$$\begin{aligned} -p_t - D_u\Delta p - 2r\frac{u}{v}p + ap - 2ruq &= \beta_1(u-\widehat{u}), \quad \text{on } \Omega\times[0,T],\\ -q_t - D_v\Delta q + r\frac{u^2}{v^2}p + bq &= \beta_2(v-\widehat{v}), \quad \text{on } \Omega\times[0,T],\\ p(\mathbf{x},T) = \beta_{T,1}(u(\mathbf{x},T)-\widehat{u}(\mathbf{x},T)), \quad q(\mathbf{x},T) &= \beta_{T,2}(v(\mathbf{x},T)-\widehat{v}(\mathbf{x},T)), \quad \text{on } \Omega,\\ \frac{\partial p}{\partial \nu} = \frac{\partial q}{\partial \nu} &= 0, \quad \text{on } \partial\Omega\times[0,T], \end{aligned}$$

where p and q denote the *adjoint variables*.

We now employ a Newton iteration, by writing at each Newton step

$$u = \bar{u}+\delta u, \quad v = \bar{v}+\delta v, \quad a = \bar{a}+\delta a, \quad b = \bar{b}+\delta b, \quad p = \bar{p}+\delta p, \quad q = \bar{q}+\delta q,$$

where $\bar{u}$, $\bar{v}$, $\bar{a}$, $\bar{b}$, $\bar{p}$, $\bar{q}$ denote the most recent iterates of u, v, a, b, p, q, with δu, δv, δa, δb, δp, δq denoting the changes in the solutions at each Newton step.

Applying this to the forward equations yields

$$\begin{aligned} (\bar{u}+\delta u)_t - D_u\Delta(\bar{u}+\delta u) - \frac{r(\bar{u}+\delta u)^2}{\bar{v}+\delta v} + (\bar{a}+\delta a)(\bar{u}+\delta u) &= r, \quad \text{on } \Omega\times[0,T],\\ (\bar{v}+\delta v)_t - D_v\Delta(\bar{v}+\delta v) - r(\bar{u}+\delta u)^2 + (\bar{b}+\delta b)(\bar{v}+\delta v) &= 0, \quad \text{on } \Omega\times[0,T],\\ (\bar{u}+\delta u)(\mathbf{x},0) = u_0(\mathbf{x}), \quad (\bar{v}+\delta v)(\mathbf{x},0) &= v_0(\mathbf{x}), \quad \text{on } \Omega,\\ \frac{\partial(\bar{u}+\delta u)}{\partial \nu} = \frac{\partial(\bar{v}+\delta v)}{\partial \nu} &= 0, \quad \text{on } \partial\Omega\times[0,T], \end{aligned}$$

whereupon we can use the assumption $(\bar{u}+\delta u)^2 \approx \bar{u}^2 + 2\bar{u}\cdot\delta u$ and the resulting derivation

$$\frac{(\bar{u}+\delta u)^2}{\bar{v}+\delta v} \approx \frac{\bar{v}-\delta v}{\bar{v}^2}(\bar{u}^2+2\bar{u}\cdot\delta u) \approx \frac{\bar{u}^2\bar{v}-\bar{u}^2\cdot\delta v+2\bar{u}\bar{v}\cdot\delta u}{\bar{v}^2}$$

to write

$$(\delta u)_t - D_u\Delta(\delta u) + r\frac{\bar{u}^2\cdot\delta v - 2\bar{u}\bar{v}\cdot\delta u}{\bar{v}^2} + \bar{u}\cdot\delta a + \bar{a}\cdot\delta u = r - \left(\bar{u}_t - D_u\Delta\bar{u} - \frac{r\bar{u}^2}{\bar{v}} + \bar{a}\bar{u}\right), \quad \text{on } \Omega\times[0,T], \tag{A.1}$$

$$(\delta v)_t - D_v\Delta(\delta v) - 2r\bar{u}\cdot\delta u + \bar{v}\cdot\delta b + \bar{b}\cdot\delta v = -(\bar{v}_t - D_v\Delta\bar{v} - r\bar{u}^2 + \bar{b}\bar{v}), \quad \text{on } \Omega\times[0,T], \tag{A.2}$$

$$(\delta u)(\mathbf{x},0) = (\delta v)(\mathbf{x},0) = 0, \quad \text{on } \Omega, \tag{A.3}$$

$$\frac{\partial(\delta u)}{\partial \nu} = \frac{\partial(\delta v)}{\partial \nu} = 0, \quad \text{on } \partial\Omega\times[0,T]. \tag{A.4}$$

Considering now a Newton iteration applied to the adjoint equations, we have

$$
\begin{aligned}
&-(\bar{p}+\delta p)_t - D_u\Delta(\bar{p}+\delta p) - 2r\frac{\bar{u}+\delta u}{\bar{v}+\delta v}(\bar{p}+\delta p) + (\bar{a}+\delta a)(\bar{p}+\delta p) - 2r(\bar{u}+\delta u)(\bar{q}+\delta q)\\
&\quad = \beta_1((\bar{u}+\delta u)-\widehat{u}), \quad \text{on } \Omega\times[0,T],\\
&-(\bar{q}+\delta q)_t - D_v\Delta(\bar{q}+\delta q) + r\frac{(\bar{u}+\delta u)^2}{(\bar{v}+\delta v)^2}(\bar{p}+\delta p) + (\bar{b}+\delta b)(\bar{q}+\delta q)\\
&\quad = \beta_2((\bar{v}+\delta v)-\widehat{v}), \quad \text{on } \Omega\times[0,T],\\
&(\bar{p}+\delta p)(\mathbf{x},T) = \beta_{T,1}((\bar{u}+\delta u)(\mathbf{x},T)-\widehat{u}(\mathbf{x},T)), \quad \text{on } \Omega,\\
&(\bar{q}+\delta q)(\mathbf{x},T) = \beta_{T,2}((\bar{v}+\delta v)(\mathbf{x},T)-\widehat{v}(\mathbf{x},T)), \quad \text{on } \Omega,\\
&\quad \frac{\partial(\bar{p}+\delta p)}{\partial\nu} = \frac{\partial(\bar{q}+\delta q)}{\partial\nu} = 0, \quad \text{on } \partial\Omega\times[0,T].
\end{aligned}
$$

Now, using the approximations

$$
\begin{aligned}
\frac{\bar{u}+\delta u}{\bar{v}+\delta v}(\bar{p}+\delta p) &\approx \frac{(\bar{u}+\delta u)(\bar{v}-\delta v)(\bar{p}+\delta p)}{\bar{v}^2}\\
&\approx \frac{\bar{u}\bar{v}\bar{p}+\bar{v}\bar{p}\cdot\delta u-\bar{u}\bar{p}\cdot\delta v+\bar{u}\bar{v}\cdot\delta p}{\bar{v}^2},\\
\frac{(\bar{u}+\delta u)^2}{(\bar{v}+\delta v)^2}(\bar{p}+\delta p) &\approx \frac{(\bar{u}+2\bar{u}\cdot\delta u)(\bar{v}^2-2\bar{v}\cdot\delta v)(\bar{p}+\delta p)}{\bar{v}^4}\\
&\approx \frac{\bar{u}}{\bar{v}^3}(\bar{u}\bar{v}\bar{p}+2\bar{v}\bar{p}\cdot\delta u-2\bar{u}\bar{p}\cdot\delta v+\bar{u}\bar{v}\cdot\delta p),
\end{aligned}
$$

we may write

$$
\begin{aligned}
&-(\delta p)_t - D_u\Delta(\delta p) - 2r\frac{\bar{u}\bar{p}\cdot\delta v-\bar{v}\bar{p}\cdot\delta u-\bar{u}\bar{v}\cdot\delta p}{\bar{v}^2} + \bar{p}\cdot\delta a+\bar{a}\cdot\delta p - 2r(\bar{u}\cdot\delta q+\bar{q}\cdot\delta u) - \beta_1\delta u\\
&\quad = \beta_1(\bar{u}-\widehat{u}) - \left(-\bar{p}_t - D_u\Delta\bar{p} - 2r\frac{\bar{u}}{\bar{v}}\bar{p}+\bar{a}\bar{p}-2r\bar{u}\bar{q}\right), \quad \text{on } \Omega\times[0,T], \qquad \text{(A.5)}\\
&-(\delta q)_t - D_v\Delta(\delta q) + r\bar{u}\frac{2\bar{v}\bar{p}\cdot\delta u+\bar{u}\bar{v}\cdot\delta p-2\bar{u}\bar{p}\cdot\delta v}{\bar{v}^2} + \bar{q}\cdot\delta b+\bar{b}\cdot\delta q-\beta_2\delta v\\
&\quad = \beta_2(\bar{v}-\widehat{v}) - \left(-\bar{q}_t - D_v\Delta\bar{q} + r\frac{\bar{u}^2}{\bar{v}^2}\bar{p}+\bar{b}\bar{q}\right), \quad \text{on } \Omega\times[0,T], \qquad \text{(A.6)}\\
&(\delta p)(\mathbf{x},T) = \beta_{T,1}(\delta u)(\mathbf{x},T), \quad (\delta q)(\mathbf{x},T) = \beta_{T,2}(\delta v)(\mathbf{x},T), \quad \text{on } \Omega, \qquad \text{(A.7)}\\
&\quad \frac{\partial(\delta p)}{\partial\nu} = \frac{\partial(\delta q)}{\partial\nu} = 0, \quad \text{on } \partial\Omega\times[0,T]. \qquad \text{(A.8)}
\end{aligned}
$$

Now, the forward and adjoint equations can clearly be derived by differentiating the Lagrangian

$$
\begin{aligned}
\mathcal{J}_{GM1}(u,v,a,b,p,q) = &\frac{\beta_1}{2}\|u-\widehat{u}\|^2_{L_2(\Omega\times[0,T])} + \frac{\beta_2}{2}\|v-\widehat{v}\|^2_{L_2(\Omega\times[0,T])}\\
&+\frac{\beta_{T,1}}{2}\|u-\widehat{u}_T\|^2_{L_2(\Omega)} + \frac{\beta_{T,2}}{2}\|v-\widehat{v}_T\|^2_{L_2(\Omega)}\\
&+\frac{\nu_1}{2}\|a\|^2_{L_2(\Omega\times[0,T])} + \frac{\nu_2}{2}\|b\|^2_{L_2(\Omega\times[0,T])}\\
&-\int_{\Omega\times[0,T]} p\left(u_t - D_u\Delta u - \frac{ru^2}{v} + au - r\right)\\
&-\int_{\Omega\times[0,T]} q\left(v_t - D_v\Delta v - ru^2 + bv\right),
\end{aligned}
$$

with respect to the adjoint variables p, q and the state variables u, v, respectively. Within this cost functional, we have excluded the constraints on the boundary conditions for readability reasons. To obtain the gradient equations we require for a closed system of equations, we also need to differentiate the above cost functional with respect to the control variables a and b. Differentiating with respect to a gives the requirement

$$\int_{\Omega\times[0,T]} (up - \nu_1 a) = 0,$$

and differentiating with respect to b yields similarly that

$$\int_{\Omega\times[0,T]} (vq - \nu_2 b) = 0.$$

Applying a Newton iteration to these equations gives constraints of the form

$$\int_{\Omega\times[0,T]} (\bar{p}\cdot\delta u + \bar{u}\cdot\delta p - \nu_1\delta a) = -\int_{\Omega\times[0,T]} (\bar{u}\bar{p} - \nu_1\bar{a}), \tag{A.9}$$

$$\int_{\Omega\times[0,T]} (\bar{q}\cdot\delta v + \bar{v}\cdot\delta q - \nu_2\delta b) = -\int_{\Omega\times[0,T]} (\bar{v}\bar{q} - \nu_2\bar{b}), \tag{A.10}$$

at each Newton step.

Therefore the complete system which we need to solve at each Newton step corresponds to the adjoint equations (A.5)–(A.8), the gradient equations (A.9) and (A.10), and the forward equations (A.1)–(A.4).

We now turn our attention to the Schnakenberg ($GM2$) model, where we wish to deal with the forward equations

$$\begin{aligned}
u_t - D_u\Delta u + \gamma(u - u^2 v) - \gamma a &= 0, \quad \text{on } \Omega\times[0,T],\\
v_t - D_v\Delta v + \gamma u^2 v - \gamma b &= 0, \quad \text{on } \Omega\times[0,T],\\
u(\mathbf{x},0) = u_0(\mathbf{x}), \quad v(\mathbf{x},0) &= v_0(\mathbf{x}), \quad \text{on } \Omega,\\
\frac{\partial u}{\partial \nu} = \frac{\partial v}{\partial \nu} &= 0, \quad \text{on } \partial\Omega\times[0,T],
\end{aligned}$$

and the adjoint equations (see [11])

$$\begin{aligned}
-p_t - D_u\Delta p + 2\gamma uv(q-p) + \gamma p &= \beta_1(u - \widehat{u}), \quad \text{on } \Omega\times[0,T],\\
-q_t - D_v\Delta q + \gamma u^2(q-p) &= \beta_2(v - \widehat{v}), \quad \text{on } \Omega\times[0,T],\\
p(\mathbf{x},T) = \beta_{T,1}(u(\mathbf{x},T) - \widehat{u}(\mathbf{x},T)), \quad q(\mathbf{x},T) &= \beta_{T,2}(v(\mathbf{x},T) - \widehat{v}(\mathbf{x},T)), \quad \text{on } \Omega,\\
\frac{\partial p}{\partial \nu} = \frac{\partial q}{\partial \nu} &= 0, \quad \text{on } \partial\Omega\times[0,T].
\end{aligned}$$

Now, substituting

$$u = \bar{u} + \delta u, \quad v = \bar{v} + \delta v, \quad a = \bar{a} + \delta a, \quad b = \bar{b} + \delta b, \quad p = \bar{p} + \delta p, \quad q = \bar{q} + \delta q,$$

into the forward equations at each Newton step gives

$$\begin{aligned}
(\bar{u}+\delta u)_t - D_u\Delta(\bar{u}+\delta u) + \gamma((\bar{u}+\delta u) - (\bar{u}+\delta u)^2(\bar{v}+\delta v)) - \gamma(\bar{a}+\delta a) &= 0, \quad \text{on } \Omega\times[0,T],\\
(\bar{v}+\delta v)_t - D_v\Delta(\bar{v}+\delta v) + \gamma(\bar{u}+\delta u)^2(\bar{v}+\delta v) - \gamma(\bar{b}+\delta b) &= 0, \quad \text{on } \Omega\times[0,T],\\
(\bar{u}+\delta u)(\mathbf{x},0) = u_0(\mathbf{x}), \quad (\bar{v}+\delta v)(\mathbf{x},0) &= v_0(\mathbf{x}), \quad \text{on } \Omega,\\
\frac{\partial(\bar{u}+\delta u)}{\partial \nu} = \frac{\partial(\bar{v}+\delta v)}{\partial \nu} &= 0, \quad \text{on } \partial\Omega\times[0,T],
\end{aligned}$$

which we may expand and simplify to give

$$\begin{aligned}
&(\delta u)_t - D_u\Delta(\delta u) + \gamma(\delta u - \bar{u}^2\cdot\delta v - 2\bar{u}\bar{v}\cdot\delta u) - \gamma\delta a\\
&\quad = -(\bar{u}_t - D_u\Delta\bar{u} + \gamma(\bar{u} - \bar{u}^2\bar{v}) - \gamma\cdot\bar{a}), \quad \text{on } \Omega\times[0,T], \qquad (A.11)\\
&(\delta v)_t - D_v\Delta(\delta v) + \gamma(\bar{u}^2\cdot\delta v + 2\bar{u}\bar{v}\cdot\delta u) - \gamma\delta b = -(\bar{v}_t - D_v\Delta\bar{v} + \gamma\bar{u}^2\bar{v} - \gamma\cdot\bar{b}), \quad \text{on } \Omega\times[0,T], \qquad (A.12)\\
&(\delta u)(\mathbf{x},0) = (\delta v)(\mathbf{x},0) = 0, \quad \text{on } \Omega, \qquad (A.13)\\
&\frac{\partial(\delta u)}{\partial \nu} = \frac{\partial(\delta v)}{\partial \nu} = 0, \quad \text{on } \partial\Omega\times[0,T]. \qquad (A.14)
\end{aligned}$$

Applying the same substitutions to the adjoint equations gives

$$
\begin{aligned}
-(\bar{p}+\delta p)_t - D_u\Delta(\bar{p}+\delta p) + 2\gamma\bar{u}\bar{v}((\bar{q}+\delta q)-(\bar{p}+\delta p)) + \gamma(\bar{p}+\delta p) &= \beta_1((\bar{u}+\delta u)-\widehat{u}), \quad \text{on } \Omega\times[0,T],\\
-(\bar{q}+\delta q)_t - D_v\Delta(\bar{q}+\delta q) + \gamma\bar{u}^2((\bar{q}+\delta q)-(\bar{p}+\delta p)) &= \beta_2((\bar{v}+\delta v)-\widehat{v}), \quad \text{on } \Omega\times[0,T],\\
(\bar{p}+\delta p)(\mathbf{x},T) &= \beta_{T,1}((\bar{u}+\delta u)(\mathbf{x},T)-\widehat{u}(\mathbf{x},T)), \quad \text{on } \Omega,\\
(\bar{q}+\delta q)(\mathbf{x},T) &= \beta_{T,2}((\bar{v}+\delta v)(\mathbf{x},T)-\widehat{v}(\mathbf{x},T)), \quad \text{on } \Omega,\\
\frac{\partial(\bar{p}+\delta p)}{\partial\nu} = \frac{\partial(\bar{q}+\delta q)}{\partial\nu} &= 0, \quad \text{on } \partial\Omega\times[0,T],
\end{aligned}
$$

which may then be expanded and simplified to give

$$
\begin{aligned}
&-(\delta p)_t - D_u\Delta(\delta p) + 2\gamma(\bar{v}\bar{q}\cdot\delta u + \bar{u}\bar{q}\cdot\delta v + \bar{u}\bar{v}\cdot\delta q - \bar{v}\bar{p}\cdot\delta u - \bar{u}\bar{p}\cdot\delta v - \bar{u}\bar{v}\cdot\delta p) + \gamma\delta p - \beta_1\delta u\\
&\quad = \beta_1(\bar{u}-\widehat{u}) - (-\bar{p}_t - D_u\Delta\bar{p} + 2\gamma\bar{u}\bar{v}(\bar{q}-\bar{p}) + \gamma\bar{p}), \quad \text{on } \Omega\times[0,T], \qquad \text{(A.15)}\\
&-(\delta q)_t - D_v\Delta(\delta q) + \gamma(\bar{u}^2\cdot\delta q + 2\bar{u}\bar{q}\cdot\delta u - \bar{u}^2\delta p - 2\bar{u}\bar{p}\cdot\delta u) - \beta_2\delta v\\
&\quad = \beta_2(\bar{v}-\widehat{v}) - (-\bar{q}_t - D_v\Delta\bar{q} + \gamma\bar{u}^2(\bar{q}-\bar{p})), \quad \text{on } \Omega\times[0,T], \qquad \text{(A.16)}\\
&(\delta p)(\mathbf{x},T) = \beta_{T,1}(\delta u)(\mathbf{x},T), \quad (\delta q)(\mathbf{x},T) = \beta_{T,2}(\delta v)(\mathbf{x},T), \quad \text{on } \Omega, \qquad \text{(A.17)}\\
&\frac{\partial(\delta p)}{\partial\nu} = \frac{\partial(\delta q)}{\partial\nu} = 0, \quad \text{on } \partial\Omega\times[0,T]. \qquad \text{(A.18)}
\end{aligned}
$$

The forward and adjoint equations can be derived by differentiating the Lagrangian

$$
\begin{aligned}
\mathcal{J}_{GM2}(u,v,a,b,p,q) = &\frac{\beta_1}{2}\|u-\widehat{u}\|^2_{L_2(\Omega\times[0,T])} + \frac{\beta_2}{2}\|v-\widehat{v}\|^2_{L_2(\Omega\times[0,T])}\\
&+ \frac{\beta_{T,1}}{2}\|u-\widehat{u}_T\|^2_{L_2(\Omega)} + \frac{\beta_{T,2}}{2}\|v-\widehat{v}_T\|^2_{L_2(\Omega)}\\
&+ \frac{\nu_1}{2}\|a\|^2_{L_2(\Omega\times[0,T])} + \frac{\nu_2}{2}\|b\|^2_{L_2(\Omega\times[0,T])}\\
&- \int_{\Omega\times[0,T]} p\left(u_t - D_u\Delta u + \gamma(u-u^2v) - \gamma a\right)\\
&- \int_{\Omega\times[0,T]} q\left(v_t - D_v\Delta v + \gamma u^2 v - \gamma b\right),
\end{aligned}
$$

with respect to u, v, p and q, similarly as for the $GM1$ model. The gradient equations for this problem may be derived by differentiating this Lagrangian with respect to the control variables a and b, which gives the conditions

$$
\int_{\Omega\times[0,T]} (\nu_1 a + \gamma p) = 0, \qquad \int_{\Omega\times[0,T]} (\nu_2 b + \gamma q) = 0.
$$

Applying Newton iteration to these equations gives

$$
\int_{\Omega\times[0,T]} (\nu_1\delta a + \gamma\delta p) = -\int_{\Omega\times[0,T]} (\nu_1\bar{a} + \gamma\bar{p}), \qquad \text{(A.19)}
$$

$$
\int_{\Omega\times[0,T]} (\nu_2\delta b + \gamma\delta q) = -\int_{\Omega\times[0,T]} (\nu_2\bar{b} + \gamma\bar{q}), \qquad \text{(A.20)}
$$

at each Newton step.

Hence the system of equations which need to be solved at each Newton step are the adjoint equations (A.15)–(A.18), the gradient equations (A.19) and (A.20), and the forward equations (A.11)–(A.14).

References

[1] O. Axelsson, J. Karátson, Equivalent operator preconditioning for elliptic problems, Numer. Algorithms 50 (2009) 297–380.
[2] A. Badugu, C. Kraemer, P. Germann, D. Menshykau, D. Iber, Digit patterning during limb development as a result of the BMP-receptor interaction, Sci. Rep. 991 (2012) 1–13.
[3] W. Bangerth, R. Hartmann, G. Kanschat, deal.II—a general-purpose object-oriented finite element library, ACM Trans. Math. Softw. 33 (2007), Art. 24, 27 pp.
[4] M. Benzi, G.H. Golub, J. Liesen, Numerical solution of saddle point problems, Acta Numer. 14 (2005) 1–137.
[5] M. Benzi, E. Haber, L. Taralli, A preconditioning technique for a class of PDE-constrained optimization problems, Adv. Comput. Math. 35 (2011) 149–173.

[6] V. Castets, E. Dulos, J. Boissonade, P.D. Kepper, Experimental evidence of a sustained standing Turing-type nonequilibrium chemical pattern, Adv. Comput. Math. 35 (2011) 2953–2956.
[7] I.S. Duff, A.M. Erisman, J.K. Reid, Direct Methods for Sparse Matrices, Monogr. Numer. Anal., The Clarendon Press, Oxford University Press, New York, 1989.
[8] A.D. Economou, A. Ohazama, T. Porntaveetus, P.T. Sharpe, S. Kondo, M.A. Basson, A. Gritli-Linde, M.T. Coburne, J.B.A. Green, Periodic stripe formation by a Turing mechanism operating at growth zones in the mammalian palate, Nat. Genet. 44 (2012) 348–351.
[9] H.C. Elman, D.J. Silvester, A.J. Wathen, Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics, Numer. Math. Sci. Comput., Oxford University Press, New York, 2005.
[10] R. Fletcher, Conjugate gradient methods for indefinite systems, in: Numerical Analysis, Proc. 6th Biennial Dundee Conf., Univ. Dundee, Dundee, 1975, in: Lect. Notes Math., vol. 506, Springer, Berlin, 1976, pp. 73–89.
[11] M.R. Garvie, P.K. Maini, C. Trenchea, An efficient and robust numerical algorithm for estimating parameters in Turing systems, J. Comput. Phys. 229 (2010) 7058–7071.
[12] M.R. Garvie, C. Trenchea, Identification of space–time distributed parameters in the Gierer–Meinhardt reaction–diffusion system, SIAM J. Appl. Math. 74 (2014) 147–166.
[13] M.W. Gee, C.M. Siefert, J.J. Hu, R.S. Tuminaro, M.G. Sala, ML 5.0 smoothed aggregation user's guide, Tech. Rep. SAND2006-2649, Sandia National Laboratories, 2006.
[14] A. Gierer, H. Meinhardt, A theory of biological pattern formation, Biol. Cybern. 12 (1972) 30–39.
[15] G.H. Golub, C.F.V. Loan, Matrix Computations, third ed., Johns Hopkins Stud. Math. Sci., Johns Hopkins University Press, Baltimore, MD, 1996.
[16] E. Haber, U.M. Ascher, D. Oldenburg, On optimization techniques for solving nonlinear inverse problems, Inverse Probl. 16 (2000) 1263–1280.
[17] M.R. Hestenes, E. Stiefel, Methods of conjugate gradients for solving linear systems, J. Res. Natl. Bur. Stand. 49 (1952) 409–436, 1953.
[18] M. Hinze, R. Pinnau, M. Ulbrich, S. Ulbrich, Optimization with PDE Constraints, Math. Model. Theor. Appl., Springer-Verlag, New York, 2009.
[19] C. Hogea, C. Davatzikos, G. Biros, An image-driven parameter estimation problem for a reaction–diffusion glioma growth model with mass effects, J. Math. Biol. 56 (2008) 793–825.
[20] K. Ito, K. Kunisch, Lagrange Multiplier Approach to Variational Problems and Applications, Adv. Des. Control, vol. 15, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2008.
[21] K. Ito, K. Kunisch, V. Schulz, I. Gherman, Approximate nullspace iterations for KKT systems, SIAM J. Matrix Anal. Appl. 31 (2010) 1835–1847.
[22] Y.A. Kuznetsov, Efficient iterative solvers for elliptic finite element problems on nonmatching grids, Russ. J. Numer. Anal. Math. Model. 10 (1995) 187–211.
[23] H.P. Langtangen, Computational Partial Differential Equations. Numerical Methods and Diffpack Programming, second ed., Springer, Berlin, 2003.
[24] R.T. Liu, S.S. Liaw, P.K. Maini, Two-stage Turing model for generating pigment patterns on the leopard and the jaguar, Phys. Rev. E 74 (2006) 011914.
[25] K.A. Mardal, R. Winther, Preconditioning discretizations of systems of partial differential equations, Numer. Linear Algebra Appl. 18 (2011) 1–40.
[26] M.F. Murphy, G.H. Golub, A.J. Wathen, A note on preconditioning for indefinite linear systems, SIAM J. Sci. Comput. 21 (2000) 1969–1972.
[27] J.D. Murray, Mathematical Biology, vol. 2: Spatial Models and Biomedical Applications, third ed., Springer, New York, NY, 2003.
[28] J. Nocedal, S.J. Wright, Numerical Optimization, Springer Ser. Oper. Res., Springer-Verlag, New York, 1999.
[29] J. Nocedal, S.J. Wright, Numerical Optimization, second ed., Springer Ser. Oper. Res. Financ. Eng., Springer, New York, 2006.
[30] Q. Ouyang, H.L. Swinney, Transition from a uniform state to hexagonal and striped Turing patterns, Nature 352 (1991) 610–612.
[31] C.C. Paige, M.A. Saunders, Solutions of sparse indefinite systems of linear equations, SIAM J. Numer. Anal. 12 (1975) 617–629.
[32] J.W. Pearson, M. Stoll, Fast iterative solution of reaction–diffusion control problems arising from chemical processes, SIAM J. Sci. Comput. 35 (2013) B987–B1009.
[33] J.W. Pearson, M. Stoll, A.J. Wathen, Regularization-robust preconditioners for time-dependent PDE-constrained optimization problems, SIAM J. Matrix Anal. Appl. 33 (2012) 1126–1152.
[34] J.W. Pearson, A.J. Wathen, A new approximation of the Schur complement in preconditioners for PDE-constrained optimization, Numer. Linear Algebra Appl. 19 (2012) 816–829.
[35] J.W. Pearson, A.J. Wathen, Fast iterative solvers for convection–diffusion control problems, Electron. Trans. Numer. Anal. 40 (2013) 294–310.
[36] T. Rusten, R. Winther, A preconditioned iterative method for saddle point problems, SIAM J. Matrix Anal. Appl. 13 (1992) 887–904.
[37] Y. Saad, Iterative Methods for Sparse Linear Systems, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2003.
[38] A. Schiela, S. Ulbrich, Operator preconditioning for a class of inequality constrained optimal control problems, SIAM J. Optim. 24 (2014) 435–466.
[39] J. Schnakenberg, Simple chemical reaction systems with limit cycle behaviour, J. Theor. Biol. 81 (1979) 389–400.
[40] R. Sheth, L. Marcon, M.F. Bastida, M. Junco, L. Quintana, R. Dahn, M. Kmita, J. Sharpe, M.A. Ros, Hox genes regulate digit patterning by controlling the wavelength of a Turing-type mechanism, Science 338 (2012) 1476–1480.
[41] F. Tröltzsch, Optimal Control of Partial Differential Equations: Theory, Methods and Applications, American Mathematical Society, 2010.
[42] A. Turing, The chemical basis of morphogenesis, Philos. Trans. R. Soc. Lond. B 237 (1952) 37–72.
[43] A.J. Wathen, T. Rees, Chebyshev semi-iteration in preconditioning for problems including the mass matrix, Electron. Trans. Numer. Anal. 34 (2008) 125–135.
[44] S.J. Wright, Primal–Dual Interior-Point Methods, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.
[45] W. Zulehner, Non-standard norms and robust estimates for saddle point problems, SIAM J. Matrix Anal. Appl. 32 (2011) 536–560.

A.9 Low-rank in time method for PDE-constrained optimization

This paper is published as

M. Stoll and T. Breiten, ***A low-rank in time approach to pde-constrained optimization*, SIAM J. Sci. Comput., 37 (2015), pp. B1–B29.**

Result from the paper

Results for full-rank (FR) MINRES vs. low-rank (LR) MINRES with both the stationary iteration (SI) and IKPIK for a fixed mesh with 16641 unknowns in space. We show varying time-steps and additionally the rank of the state/control/adjoint state. Both iteration numbers and computing times in seconds are listed. OoM indicates Out of Memory in MATLAB®. Results are shown for $\beta = 10^{-4}$.

DoF 16641	20 # it(t)	100 # it(t)	200 # it(t)	400 # it(t)	600 # it(t)
LR(SI)	19(108.2)	21(307.8)	25(432.7)	43(671.9)	61(937.3)
LR(IKPIK)	19(115.1)	19(288.9)	19(296.8)	21(335.3)	21(357.1)
Rank (SI)	8/10/10	10/11/11	12/13/13	11/14/14	14/15/15
FR	21(18.3)	35(124.0)	63(434.3)	OoM	OoM

Table A.5: Full-rank versus low-rank scheme implemented with two different preconditioners.

SIAM J. SCI. COMPUT.
Vol. 37, No. 1, pp. B1–B29

A LOW-RANK IN TIME APPROACH TO PDE-CONSTRAINED OPTIMIZATION*

MARTIN STOLL† AND TOBIAS BREITEN‡

Abstract. The solution of time-dependent PDE-constrained optimization problems is a challenging task in numerical analysis and applied mathematics. All-at-once discretizations and corresponding solvers provide efficient methods to robustly solve the arising discretized equations. One of the drawbacks of this approach is the high storage demand for the vectors representing the discrete space-time cylinder. Here we introduce a low-rank in time technique that exploits the low-rank nature of the solution. The theoretical foundations for this approach originate in the numerical treatment of matrix equations and can be carried over to PDE-constrained optimization. We illustrate how three different problems can be rewritten and used within a low-rank Krylov subspace solver with appropriate preconditioning.

Key words. PDE-constrained optimization, low-rank methods, space-time methods, preconditioning, Schur-complement, matrix equations

AMS subject classifications. 65F08, 65F10, 65F50, 92E20, 93C20

DOI. 10.1137/130926365

1. Introduction. Many complex phenomena in the natural, engineering, and life sciences are modeled using partial differential equations (PDEs). To obtain optimal configurations of these equations one typically formulates this as a PDE-constrained optimization problem of the form

$$\min \mathcal{J}(y,u)$$

subject to

$$\mathcal{L}(y,u) = 0$$

with $\mathcal{J}(y,u)$ the functional of interest and $\mathcal{L}(y,u)$ representing the differential operator. Problems of this type have been carefully analyzed in the past (see [49, 84] and the references therein).

Recently with the advancement of algorithms and technology, research has focused on the efficient numerical solution of these problems. In this paper we focus on the efficient solution of the discretized first order conditions in a space-time framework. The KKT conditions when considered in an all-at-once approach, i.e., simultaneous discretization in space and time, are typically of vast dimensionality. Matrix-free approaches have recently been developed to guarantee the (nearly) optimal convergence of iterative Krylov subspace solvers. The focus for both steady [68, 73] and transient problems [61, 81] has been on the development of efficient preconditioning

*Submitted to the journal's Computational Methods in Science and Engineering section June 26, 2013; accepted for publication (in revised form) October 16, 2014; published electronically January 8, 2015.

http://www.siam.org/journals/sisc/37-1/92636.html

†Numerical Linear Algebra for Dynamical Systems, Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, 39106 Magdeburg, Germany (stollm@mpi-magdeburg.mpg.de).

‡Institute for Mathematics and Scientific Computing, Heinrichstr. 36/III, University of Graz, Austria (tobias.breiten@uni-graz.at). Most of this work was completed while this author was with the Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg.

strategies for the linear system that typically are of structured form (see [16, 27] for introductions to the numerical solution of saddle point systems).

One of the obstacles using a space-time discretization is the storage requirement for the large vectors needed to represent the solution at all times. Approaches such as checkpointing [38] or multiple shooting [42] are possible methods to solve these problems. Here we want to introduce an alternative to these schemes that can for certain problems provide an efficient representation with a minimal amount of storage. We are basing our methodology on recent developments within the solution of large and sparse matrix equations; see, e.g., [4, 13, 25, 28, 36, 50, 51, 54, 70, 74, 77, 85] and references therein. One classical representative in this category is the Lyapunov equation

$$AX + XA^T = -\tilde{C}\tilde{C}^T,$$

where we are interested in approximating the matrix-valued unknown X. Solving this system is equivalent to solving the linear system

$$(I \otimes A + A \otimes I)x = \tilde{c},$$

where x and $\tilde{c}$ are related to X and $\tilde{C}\tilde{C}^T$, respectively. For details on the relevance of this equation within control theory, see [3, 44, 52]. In [15, 56, 64, 65, 70] the authors have introduced low-rank iterative schemes that approximate intermediate iterates X_k in a low-rank fashion that is maintained until convergence. We can exploit these technologies for problems coming from PDE-constrained optimization. It is not expected that these techniques outperform optimal solvers with only a few time-steps. The more crucial component is that they enable computations with many time-steps that would otherwise not be possible.

The paper is structured as follows. In section 2 we introduce the heat equation as our model problem and discuss its discretization. Section 3 illustrates how this problem can be reformulated using Kronecker technology and how we need to adapt a standard Krylov-subspace solver to be able to solve this problem efficiently. As we need a preconditioner for fast convergence we next discuss possible preconditioners in section 3. We provide some theoretical results in section 4. Section 5 is devoted to illustrating that our methodology can be carried over to other state equations such as Stokes equations and the convection-diffusion equation. Finally, in section 6 we illustrate the competitiveness of our approach.

2. A PDE-constrained optimization model problem. We start the derivation of the low-rank in time method by considering an often used model problem in PDE-constrained optimization (see [47, 49, 84]) that nevertheless reflects the crucial structure exhibited by many problems of similar type. Our goal is the minimization of a misfit functional that aims at bringing the state y as close as possible to a desired or observed state y_{obs} while using a control u, i.e.,

$$\min_{y,u} \frac{1}{2}\|y - y_{obs}\|^2_{L_2(\Omega_1)} + \frac{\beta}{2}\|u\|^2_{L_2(\Omega_2)}, \tag{2.1}$$

subject to a partial differential equation that connects both state and control, referred to as the state equation. We start by considering the heat equation with a distributed control term,

$$\begin{aligned} y_t - \nabla^2 y &= u \quad \text{in } \Omega, \\ y &= f \quad \text{on } \partial\Omega, \end{aligned} \tag{2.2}$$

or equipped with Neumann-boundary control,

$$\begin{aligned} y_t - \nabla^2 y &= f \quad \text{in } \Omega, \\ \frac{\partial y}{\partial \mathrm{n}} &= u \quad \text{on } \partial\Omega. \end{aligned} \tag{2.3}$$

For a more detailed discussion on the well-posedness, existence of solutions, etc., we refer the interested reader to [47, 49, 84]. Classically these problems are solved using a Lagrangian to incorporate the constraints and then consider the first order optimality conditions or KKT conditions [49, 58, 84]. This can be done either by forming a discrete Lagrangian and then performing the optimization procedure or by first considering an infinite-dimensional Lagrangian for whose first order conditions we employ a suitable discretization. Here we perform the first approach, although much of what we state in this paper is valid for both cases. Our goal is to build a discrete Lagrangian using an all-at-once approach [61, 81, 40] using a discrete problem within the space-time cylinder $\Omega \times [0, T]$. Using the trapezoidal rule in time and finite elements in space leads to the discrete objective function

$$J(y, u) = \frac{\tau}{2} (y - y_{obs})^T \mathcal{M}_1 (y - y_{obs}) + \frac{\tau\beta}{2} u^T \mathcal{M}_2 u \tag{2.4}$$

with $\mathcal{M}_1 = \text{blkdiag}(\frac{1}{2}M_1, M_1, \ldots, M_1, \frac{1}{2}M_1), \mathcal{M}_2 = \text{blkdiag}(\frac{1}{2}M_2, M_2, \ldots, M_2, \frac{1}{2}M_2)$ being space-time matrices where M_1 is the mass matrix associated with the domain Ω_1 and M_2 is the corresponding mass matrix for Ω_2. The vectors $y = [y_1^T \ldots y_{n_t}^T]^T$ and $u = [u_1^T \ldots u_{n_t}^T]^T$ are of vast dimensionality and represent a collection of spatial vectors for all time-steps collected into one single vector.

The all-at-once discretization of the state equation using finite elements in space and an implicit Euler scheme in time is given by

$$\mathcal{K} y - \tau \mathcal{N} u = d, \tag{2.5}$$

where

$$\mathcal{K} = \begin{bmatrix} L & & & \\ -M & L & & \\ & \ddots & \ddots & \\ & & -M & L \end{bmatrix}, \quad \mathcal{N} = \begin{bmatrix} N & & & \\ & N & & \\ & & \ddots & \\ & & & N \end{bmatrix}, \quad d = \begin{bmatrix} M_1 y_0 + f \\ f \\ \vdots \\ f \end{bmatrix}.$$

Here, M is the mass matrix for the domain Ω, the matrix L is defined as $L = M + \tau K$, the matrix N represents the control term either via a distributed control (square matrix) or via the contributions of a boundary control problem (rectangular matrix), and the right-hand-side d consists of a contribution from the initial condition y_0 and a vector f representing forcing terms and contributions of boundary conditions. The first order conditions using a Lagrangian formulation with Lagrange multiplier p leads to the following system:

$$\underbrace{\begin{bmatrix} \tau\mathcal{M}_1 & 0 & -\mathcal{K}^T \\ 0 & \beta\tau\mathcal{M}_2 & \tau\mathcal{N}^T \\ -\mathcal{K} & \tau\mathcal{N} & 0 \end{bmatrix}}_{\mathcal{A}} \begin{bmatrix} y \\ u \\ p \end{bmatrix} = \begin{bmatrix} \tau\mathcal{M}_1 y_{obs} \\ 0 \\ d \end{bmatrix}. \tag{2.6}$$

Systems of this form have previously been studied in [81, 61, 82, 57]. As these systems are of vast dimensionality it is crucial to find appropriate preconditioners together with Krylov subspace solvers to efficiently obtain an approximation to the solution. The vast dimensionality of system matrices does not allow the use of direct solvers [26, 23] but we can employ Krylov subspace solvers in a matrix-free way by never forming the matrix $\mathcal{A}$ and only implicitly performing the matrix vector product. The main bottleneck of this approach is the storage requirement for the space-time vectors which can be reduced by working on the Schur-complement if it exists of the matrix $\mathcal{A}$ or removing the control from the system matrix [76, 45]. Other approaches that can be employed are checkpointing schemes [38] or multiple shooting approaches [42]. In the following we want to present an alternative that uses the underlying tensor structure of the first order conditions.

3. A Kronecker view. We noticed earlier that the linear system in (2.6) is of vast dimensionality and that we need only very few matrices to efficiently perform the matrix vector multiplication with $\mathcal{A}$, and we can approach this in a matrix-free form by never forming $\mathcal{A}$. Nevertheless, the vectors y, u, and p themselves are enormous and every storage reduction would help to improve the performance of an optimization scheme. The goal now is to employ the structure of the linear system to reduce the storage requirement for the iterative method. Our approach is based on recent developments for matrix equations [10, 54, 36]. Using the definition of the Kronecker product

$$W \otimes V = \begin{bmatrix} w_{11}V & \dots & w_{1m}V \\ \vdots & \ddots & \vdots \\ w_{n1}V & \dots & w_{nm}V \end{bmatrix}$$

we note that (2.6) can also be written as

$$\underbrace{\begin{bmatrix} D_1 \otimes \tau M_1 & 0 & -\left(I_{n_t} \otimes L + C^T \otimes M\right) \\ 0 & D_2 \otimes \beta\tau M_2 & D_3 \otimes \tau N^T \\ -\left(I_{n_t} \otimes L + C \otimes M\right) & D_3 \otimes \tau N & 0 \end{bmatrix}}_{\mathcal{A}} \begin{bmatrix} y \\ u \\ p \end{bmatrix} = \begin{bmatrix} D_1 \otimes \tau M_1 y_{obs} \\ 0 \\ d \end{bmatrix}, \tag{3.1}$$

where $D_1 = D_2 = \operatorname{diag}\left(\frac{1}{2}, 1, \dots, 1, \frac{1}{2}\right)$ and $D_3 = I_{n_t}$. Additionally, the matrix $C \in \mathbb{R}^{n_t, n_t}$ is given by

$$C = \begin{bmatrix} 0 & & & \\ -1 & 0 & & \\ & \ddots & \ddots & \\ & & -1 & 0 \end{bmatrix}$$

and represents the implicit Euler scheme. It is of course possible to use a different discretization in time. So far we have simply reformulated the previously given system. But our goal was to derive a scheme that allows for a reduction in storage requirement for the vectors y, u, and p. For this we remind the reader of the definition of the vec operator via

$$\operatorname{vec}(W) = \begin{bmatrix} w_{11} \\ \vdots \\ w_{n1} \\ \vdots \\ w_{nm} \end{bmatrix}$$

as well as the relation

$$\left(W^T \otimes V\right) \operatorname{vec}(Y) = \operatorname{vec}(VYW).$$

Now employing this and using the notation

$$Y = [\ y_1, y_2, \ldots, y_{n_t}\ ], \quad U = [\ u_1, u_2, \ldots, u_{n_t}\ ], \quad P = [\ p_1, p_2, \ldots, p_{n_t}\ ]$$

we get that

$$\begin{aligned} &\begin{bmatrix} D_1 \otimes \tau M_1 & 0 & -\left(I_{n_t} \otimes L + C^T \otimes M\right) \\ 0 & D_2 \otimes \beta\tau M_2 & D_3 \otimes \tau N^T \\ -\left(I_{n_t} \otimes L + C \otimes M\right) & D_3 \otimes \tau N & 0 \end{bmatrix} \begin{bmatrix} \operatorname{vec}(Y) \\ \operatorname{vec}(U) \\ \operatorname{vec}(P) \end{bmatrix} \\ &= \operatorname{vec}\left(\begin{bmatrix} \tau M_1 Y D_1^T - LPI_{n_t}^T - MPC \\ \tau\beta M_2 U D_2^T + \tau N^T P D_3^T \\ -LYI_{n_t}^T - MYC^T + \tau N U D_3^T \end{bmatrix}\right). \end{aligned} \tag{3.2}$$

So far nothing is gained from rewriting the problem in this form. As was previously done in [10] we assume for now that if Y, U, and P can be represented by a low-rank approximation, any iterative Krylov subspace solver can be implemented using a low-rank version of (3.2). We denote the low-rank representations by

$$Y = W_Y V_Y^T \text{ with } W_Y \in \mathbb{R}^{n_1,k_1}, V_Y \in \mathbb{R}^{n_t,k_1}, \tag{3.3}$$

$$U = W_U V_U^T \text{ with } W_U \in \mathbb{R}^{n_2,k_2}, V_U \in \mathbb{R}^{n_t,k_2}, \tag{3.4}$$

$$P = W_P V_P^T \text{ with } W_P \in \mathbb{R}^{n_1,k_3}, V_P \in \mathbb{R}^{n_t,k_3}, \tag{3.5}$$

with $k_{1,2,3}$ being small in comparison to n_t, and we rewrite (3.2) accordingly to get

$$\begin{bmatrix} \tau M_1 W_Y V_Y^T D_1^T - LW_P V_P^T I_{n_t}^T - MW_P V_P^T C \\ \tau\beta M_2 W_U V_U^T D_2^T + \tau N^T W_P V_P^T D_3^T \\ -LW_Y V_Y^T I_{n_t}^T - MW_Y V_Y^T C^T + \tau N W_U V_U^T D_3^T \end{bmatrix}, \tag{3.6}$$

where we skipped the vec operator and instead used matrix-valued unknowns. Note that we can write the block-rows of (3.6) as

$$\begin{aligned} &\text{(first block-row)} \quad \begin{bmatrix} \tau M_1 W_Y & -LW_P & -MW_P \end{bmatrix} \begin{bmatrix} V_Y^T D_1^T \\ V_P^T I_{n_t}^T \\ V_P^T C \end{bmatrix}, \\ &\text{(second block-row)} \quad \begin{bmatrix} \tau\beta M_2 W_U & \tau N^T W_P \end{bmatrix} \begin{bmatrix} V_U^T D_2^T \\ V_P^T D_3^T \end{bmatrix}, \\ &\text{(third block-row)} \quad \begin{bmatrix} -LW_Y & -MW_Y & \tau N W_U \end{bmatrix} \begin{bmatrix} V_Y^T I_{n_t}^T \\ V_Y^T C^T \\ V_U^T D_3^T \end{bmatrix}. \end{aligned} \tag{3.7}$$

We obtain a significant storage reduction if we can base our approximation of the solution using the low-rank factors (3.7). It is easily seen that due to the low-rank nature of the factors we have to perform fewer multiplications with the submatrices by also maintaining smaller storage requirements. As the usage of a direct solver is out of the question we here rely on a preconditioned Krylov subspace solver, namely, MINRES introduced in [59] as the underlying matrix is symmetric and indefinite. Before explaining all the intricacies of the method we state the resulting algorithm and carefully explain the necessary details afterward. Algorithm 1 shows a low-rank implementation of the classical preconditioned MINRES method as presented in [59]. Note that due to the truncation to low-rank the application of the preconditioner is not identical for every step of the iteration and the use of a flexible solver needs to be investigated in the future. Here we use a rather small truncation tolerance to try to maintain a very accurate representation of what the full-rank representation would look like.

It is hard to hide the fact that the low-rank version presented here seems much messier than its vector-based relative. This is due to the fact that we want to maintain the structure of the saddle point system, which is reflected in low-rank representations associated with the state (all matrices with indices 11 and 12), the control (all matrices with indices 21 and 22), and the Lagrange multiplier (all matrices with indices 31 and 32). Please keep in mind that

$$\operatorname{vec}\left(\begin{bmatrix} Z_{11}Z_{12}^T \\ Z_{21}Z_{22}^T \\ Z_{31}Z_{32}^T \end{bmatrix}\right) = z$$

corresponds to the associated vector z from a vector-based version of MINRES.

For Algorithm 1 to be accessible to the reader, we need to dissect its different parts. Starting with the inner products of the classical MINRES method we see that we can efficiently evaluate the inner product $\left(z^{(j)}, v^{(j)}\right)$. In more detail, we use

$$\operatorname{vec}\left(\begin{bmatrix} Z_{11}^{(j)}(Z_{12}^{(j)})^T \\ Z_{21}^{(j)}(Z_{22}^{(j)})^T \\ Z_{31}^{(j)}(Z_{32})^T \end{bmatrix}\right) = z^{(j)} \text{ and } \operatorname{vec}\left(\begin{bmatrix} V_{11}^{(j)}(V_{12}^{(j)})^T \\ V_{21}^{(j)}(V_{22}^{(j)})^T \\ V_{31}^{(j)}(V_{32})^T \end{bmatrix}\right) = v^{(j)}$$

and the relation for the trace

$$\operatorname{trace}\left(A^T B\right) = \operatorname{vec}(A)^T \operatorname{vec}(B)$$

to compute the inner product $\left(z^{(j)}, v^{(j)}\right)$ (for convenience ignoring the index j) via

$$\begin{aligned} \left(z^{(j)}, v^{(j)}\right) &= \operatorname{trace}\left(\left(Z_{11}Z_{12}^T\right)^T \left(V_{11}V_{12}^T\right)\right) \\ &\quad + \operatorname{trace}\left(\left(Z_{21}Z_{22}^T\right)^T \left(V_{21}V_{22}^T\right)\right) \\ &\quad + \operatorname{trace}\left(\left(Z_{31}Z_{32}^T\right)^T \left(V_{31}V_{32}^T\right)\right), \end{aligned} \tag{3.8}$$

where $z^{(j)}$ and $v^{(j)}$ are the vectorization of the stacked V and Z matrices. Note that so far we have rewritten the vector problem in matrix form, but the interested reader might have noted that the matrices formed as part of (3.8) are of the full dimensionality $n \times n_t$ in the case of a distributed control problem. Due to the properties of the trace operator we are in luck as

$$\operatorname{trace}\left(\left(Z_{11}Z_{12}^T\right)^T \left(V_{11}V_{12}^T\right)\right) = \operatorname{trace}\left(Z_{11}^T V_{11} V_{12}^T Z_{12}\right)$$

Algorithm 1: Low-rank MINRES.

Zero-Initiliazation of $V_{11}^{(0)}, \dots$, $W_{11}^{(0)}, \dots$, and $W_{11}^{(1)}, \dots$.
Choose $U_{11}^{(0)}, U_{12}^{(0)}, U_{21}^{(0)}, U_{22}^{(0)}, U_{31}^{(0)}, U_{32}^{(0)}$
Set $V_{11}, V_{12}, \dots$ to normalized residual
while residual norm $>$ tolerance **do**
$Z_{11}^{(j)} = Z_{11}^{(j)}/\gamma_j$, $Z_{21}^{(j)} = Z_{21}^{(j)}/\gamma_j$, $Z_{31}^{(j)} = Z_{31}^{(j)}/\gamma_j$,
$[F_{11}, F_{12}, F_{21}, F_{22}, F_{31}, F_{32}] = \texttt{Amult}(Z_{11}^{(j)}, Z_{12}^{(j)}, Z_{21}^{(j)}, Z_{22}^{(j)}, Z_{31}^{(j)}, Z_{32}^{(j)})$
$\delta_j = \texttt{traceproduct}(F_{11}, F_{12}, F_{21}, F_{22}, F_{31}, F_{32}, Z_{11}^{(j)}, Z_{12}^{(j)}, Z_{21}^{(j)}, Z_{22}^{(j)}, Z_{31}^{(j)}, Z_{32}^{(j)})$
$V_{11}^{(j+1)} = \left\{F_{11} \quad -\frac{\delta_j}{\gamma_j} V_{11}^{(j)} \quad -\frac{\gamma_j}{\gamma_{j-1}} V_{11}^{(j-1)}\right\}$, $V_{12}^{(j+1)} = \left\{F_{12} \quad V_{12}^{(j)} \quad V_{12}^{(j-1)}\right\}$
$V_{21}^{(j+1)} = \left\{F_{21} \quad -\frac{\delta_j}{\gamma_j} V_{21}^{(j)} \quad -\frac{\gamma_j}{\gamma_{j-1}} V_{21}^{(j-1)}\right\}$, $V_{22}^{(j+1)} = \left\{F_{22} \quad V_{22}^{(j)} \quad V_{22}^{(j-1)}\right\}$
$V_{31}^{(j+1)} = \left\{F_{31} \quad -\frac{\delta_j}{\gamma_j} V_{31}^{(j)} \quad -\frac{\gamma_j}{\gamma_{j-1}} V_{31}^{(j-1)}\right\}$, $V_{32}^{(j+1)} = \left\{F_{32} \quad V_{32}^{(j)} \quad V_{32}^{(j-1)}\right\}$
$\left\{Z_{11}^{(j+1)}, Z_{12}^{(j+1)}, Z_{21}^{(j+1)}, Z_{22}^{(j+1)}, Z_{31}^{(j+1)}, Z_{32}^{(j+1)}\right\} =$
$\texttt{Aprec}(V_{11}^{(j+1)}, V_{12}^{(j+1)}, V_{21}^{(j+1)}, V_{22}^{(j+1)}, V_{31}^{(j+1)}, V_{32}^{(j+1)})$
$\gamma_{j+1} = \sqrt{\texttt{tracepoduct}(Z_{11}^{(j+1)}, \dots, V_{11}^{(j+1)}, \dots)}$
$\alpha_0 = c_j \delta_j - c_{j-1} s_j \gamma_j$
$\alpha_1 = \sqrt{\alpha_0^2 + \gamma_{j+1}^2}$
$\alpha_2 = s_j \delta_j + c_{j-1} c_j \gamma_j$
$\alpha_3 = s_{j-1} \gamma_j$
$c_{j+1} = \frac{\alpha_0}{\alpha_1}$
$s_{j+1} = \frac{\gamma_{j+1}}{\alpha_1}$
$W_{11}^{(j+1)} = \left\{Z_{11}^{(j)} \quad -\alpha_3 W_{11}^{(j-1)} \quad -\alpha_2 W_{11}^{(j)}\right\}$, $W_{12}^{(j+1)} = \left\{Z_{12}^{(j)} \quad W_{12}^{(j-1)} \quad W_{12}^{(j)}\right\}$
$W_{21}^{(j+1)} = \left\{Z_{21}^{(j)} \quad -\alpha_3 W_{21}^{(j-1)} \quad -\alpha_2 W_{21}^{(j)}\right\}$, $W_{22}^{(j+1)} = \left\{Z_{22}^{(j)} \quad W_{22}^{(j-1)} \quad W_{22}^{(j)}\right\}$
$W_{31}^{(j+1)} = \left\{Z_{31}^{(j)} \quad -\alpha_3 W_{31}^{(j-1)} \quad -\alpha_2 W_{31}^{(j)}\right\}$, $W_{32}^{(j+1)} = \left\{Z_{32}^{(j)} \quad W_{32}^{(j-1)} \quad W_{32}^{(j)}\right\}$
if Convergence criterion fulfilled **then**
Compute approximate solution
stop
end if
end while

allows us to to compute the trace of small matrices rather than of the ones from the full temporal/spatial discretization. We denote the reformulation of the trace in Algorithm 1 by the term `tracepoduct`.

We have now defined the matrix vector multiplication denoted by `Amult` in Algorithm 1 and shown in detail in Algorithm 2 as well as the efficient computation of the inner products within the low-rank MINRES algorithm. We have not yet defined the brackets $\{\}$. The brackets $U := \{U_1 \quad V_1 \quad W_1\}$ and $\{U_2 \quad V_2 \quad W_2\}$ can be understood as a concatenation and truncation by the way of an abstract function `trunc` that takes as inputs the matrices $[U_1 \quad V_1 \quad W_1]$ and $[U_2 \quad V_2 \quad W_2]$ and gives back low-rank approximations to these matrices, i.e., $\tilde{Z}_1 \approx [U_1 \quad V_1 \quad W_1]$ and $\tilde{Z}_2 \approx [U_2 \quad V_2 \quad W_2]$. We now briefly discuss how the `trunc` function could be designed.

We want to perform the truncation of two matrices V and U that represent the low-rank representation of $Z = VU^T$. As discussed in [54] we can perform skinny QR factorizations of both matrices, i.e., $V = Q_v R_v$ and $U = Q_u R_u$. We then note that $Z = Q_v R_v R_u^T Q_u^T$. A singular value decomposition [32] of the matrix

Algorithm 2: Matrix multiplication: Amult.

Input: $W_{11}, W_{12}, W_{21}, W_{22}, W_{31}, W_{32}$
Output: $Z_{11}, Z_{12}, Z_{21}, Z_{22}, Z_{31}, Z_{32}$
$Z_{11} = \begin{bmatrix} \tau M_1 W_{11} & -LW_{31} & -MW_{31} \end{bmatrix}$
$Z_{21} = \begin{bmatrix} \tau\beta M_2 W_{21} & \tau N W_{31} \end{bmatrix}$
$Z_{31} = \begin{bmatrix} -LW_{11} & -MW_{31} & \tau N W_{21} \end{bmatrix}$
$Z_{12} = \begin{bmatrix} D_1 W_{12} & I_{n_t} W_{32} & C^T W_{32} \end{bmatrix}$
$Z_{22} = \begin{bmatrix} D_2 W_{22} & D_3 W_{32} \end{bmatrix}$
$Z_{32} = \begin{bmatrix} I_{n_t} W_{12} & C W_{12} & D_3 W_{22} \end{bmatrix}$

$R_v R_u^T = B\Sigma C^T$ provides the means to reduce the rank by dropping small (depending on some tolerance) singular values. Using MATLAB notation we get a low-rank approximation via the truncated expression $B(:, 1:k)\Sigma(1:k, 1:k)C(:, 1:k)^T$. This leads to the overall low-rank approximation $V_{new} = Q_v B(:, 1:k)$ and $U_{new} = Q_v C(:, 1:k)\Sigma(1:k, 1:k)$, which in turn gives $Z \approx V_{new} U_{new}^T$. We have implemented this approach in MATLAB but noted that the computation of the skinny QR factorization was rather slow. Alternatively, we exploited the MATLAB function `svds` to directly compute a truncated singular value decomposition of VU^T by passing a function handle that allowed the implicit application of the $Z = VU^T$ without ever forming this matrix. This approach proved advantageous in terms of the time needed for the truncation. Note that alternative ways to compute the truncated SVD are of course possible [48, 6, 79].

Before discussing the possible preconditioners, employed via the `Aprec` function in Algorithm 1, we state that the vector update formulas given in Algorithm 1 are straightforward versions of vector versions of MINRES.

We additionally want to briefly comment on some of the alternative approaches to the presented methodology. One can of course reduce the dimensionality by eliminating the control when possible and obtain a system that is still vast and can be cast using our low-rank methodology. The use of reduced Hessian approaches typically leads to a symmetric positive-definite system for which CG would be applicable. But these formulations usually involve the inverse of the discretized PDE and this means that in order to simply apply the system matrix to a vector one has to very accurately solve for the PDE, as otherwise the matrix vector product does not represent the original KKT system. We refer to [39] for more details. Many algorithms employ the checkpointing technique [38], where only snapshots in time are stored. This is often done when the KKT system is treated in a block-Gauss–Seidel fashion, i.e., solving adjoint PDE, gradient equation, and forward PDE in an alternating manner. For such an iteration convergence might be slow or without proper scaling the method might not converge at all. The multiple shooting approach presented in [42] is in spirit very similar to the full-rank system we introduced here. Heinkenschloss [42] introduced an augmented Lagrangian formulation that leads to a large-scale linear system than can be reordered to obtain a system similar to (2.6). Our approach implicitly picks the "correct" number of vectors needed to accurately represent the solution in time. On the other hand we are currently limited to a formulation that can be written in the tensor form shown above. This is not true for the full-space approach and techniques based on the checkpointing methodology.

While the storage requirements can be reduced dramatically we still need to precondition the linear systems as we still have to deal with possibly very large matrices

from the discretization in space. We show in the next section that we can use many of the techniques from the full-order space-time system for the low-rank scheme.

Preconditioning for low-rank MINRES. The study of preconditioners for the optimal control subject to parabolic PDEs has recently seen developments that were aimed at providing robust performance with respect to the many system parameters such as mesh-size or regularization parameters (see [73, 61, 60, 53]). More results can be found in [40, 17] and for multigrid techniques we recommend [18] and the references therein. We start our derivation of suitable preconditioners based on an approach presented by Pearson and colleagues [61, 63], where we start with a block-diagonal preconditioner

$$\mathcal{P} = \begin{bmatrix} A_0 & & \\ & A_1 & \\ & & \hat{S} \end{bmatrix}. \tag{3.9}$$

Here $A_0 \approx \tau\mathcal{M}_1$ and $A_1 \approx \tau\beta\mathcal{M}_2$ are approximations to the upper left block of $\mathcal{A}$ and $\hat{S}$ is an approximation to the Schur-complement

$$S = \tau^{-1}\mathcal{K}\mathcal{M}_1^{-1}\mathcal{K}^T + \frac{\tau}{\beta}\mathcal{N}\mathcal{M}_2^{-1}\mathcal{N}^T.$$

One approximation that has proved to be very effective [61, 63] is of the form

$$\hat{S} = \tau^{-1}\left(\mathcal{K} + \hat{\mathcal{M}}\right)\mathcal{M}_1^{-1}\left(\mathcal{K} + \hat{\mathcal{M}}\right)^T,$$

where in the case of a distributed control problem the matrix $\hat{\mathcal{M}}$ is given by

$$\hat{\mathcal{M}} = \frac{\tau}{\sqrt{\beta}}\,\mathrm{blkdiag}(M, \ldots, M).$$

Note that for for simplicity we assumed $D_1 = D_2 = D_3 = I_{n_t}$ during the Schur-complement approximation. It is of course possible to obtain robust approximations for other choices, but they would make the presentation of the Schur-complement approximation less accessible (see [61] for details using different Ds). This approach will be the basis for the derivation of efficient preconditioners for the low-rank version of MINRES. For this we need the preconditioner $\mathcal{P}$ to maintain the low-rank structure as described in (3.7). Due to the nature of the upper left block of $\mathcal{A}$ given by

$$\begin{bmatrix} D_1 \otimes \tau M_1 & 0 \\ 0 & D_2 \otimes \beta\tau M_2 \end{bmatrix}$$

we see that an efficient preconditioner given, for example, by

$$\begin{bmatrix} D_1 \otimes \tau\hat{M}_1 & 0 \\ 0 & D_2 \otimes \beta\tau\hat{M}_2 \end{bmatrix},$$

where the mass matrices are approximated by the Chebyshev semi-iteration [86], will naturally maintain the desired structure. But what can be said about the Schur-complement S of the above system? Starting from the previously used approximation

$$\hat{\mathcal{S}} = \tau^{-1}\left(I_{n_t} \otimes \hat{L} + C \otimes M\right)\mathcal{M}_1^{-1}\left(I_{n_t} \otimes \hat{L} + C \otimes M\right)^T,$$

where $\hat{L} = ((1 + \frac{\tau}{\sqrt{\beta}})M_1 + \tau K)$, we see that there already exists an inherent tensor structure within this approximation. In [81] the authors observe that such a system can be easily solved as the matrix $(I_{n_t} \otimes \hat{L} + C \otimes M)$ is of block-triangular nature. This means one can sequentially pass through the vectors associated with each grid-point in time. For our purpose the block-triangular nature will not be sufficient to guarantee the low-rank preserving nature of our algorithm. In simple terms, a low-rank factorization in time does not allow for a temporal decoupling of the time-steps as the vectors for each time-step are not readily identified. In mathematical terms we can see that it is not possible to explicitly write down the inverse of $(I_{n_t} \otimes \hat{L} + C \otimes M)$. Our starting point is a Block–Jacobi version of the Schur-complement approximation. This procedure is motivated by the fact that we can simply write

$$\left(I_{n_t} \otimes \hat{L}\right)^{-1} = \left(I_{n_t} \otimes \hat{L}^{-1}\right).$$

The last expression assures us that this preconditioner applied to any vector $v = \operatorname{vec}\left(RS^T\right)$ can be written as

$$\operatorname{vec}\left(\hat{L}^{-1} R S^T I_{n_t}\right).$$

We can now simply use the Schur-complement approximation

$$\hat{\mathcal{S}} = \tau^{-1}\left(I_{n_t} \otimes \hat{L}\right) \mathcal{M}_1 \left(I_{n_t} \otimes \hat{L}\right)^T$$

or when using $\hat{\mathcal{S}} = \tau^{-1}(I_{n_t} \otimes \hat{L} + C \otimes M)\, \mathcal{M}_1 (I_{n_t} \otimes \hat{L} + C \otimes M)^T$ approximate the inverse of $(I_{n_t} \otimes \hat{L} + C \otimes M)$ by a small, fixed number of steps of a stationary iteration with the block-diagonal preconditioner $(I_{n_t} \otimes \hat{L})$.

Another possibility is to employ a matrix equation approach to approximately solve for the Schur-complement system with $\hat{\mathcal{S}}$, where we use that $(I_{nt} \otimes \hat{L} + C \otimes M)$ is the Kronecker representation of the generalized Sylvester operator

$$\mathcal{S}(X) = \hat{L}X + MXC^T.$$

As mentioned in the beginning, there exist several low-rank methods such as the ADI iteration (see, e.g., [14]) and projection-based methods (see, e.g., [74]) that allow us to approximately solve linear matrix equations of this type. For our purposes, we use the method IKPIK, which is an inexact version of the method KPIK developed in [74]. We employ this method with a small and fixed number of steps to approximately solve the two matrix equations $(I_{n_t} \otimes L + \hat{C} \otimes M)$ and its transpose found in

$$\hat{\mathcal{S}} = \tau^{-1}\left(I_{n_t} \otimes L + \hat{C} \otimes M\right) \mathcal{M}_1^{-1} \left(I_{n_t} \otimes L + \hat{C} \otimes M\right)^T.$$

Note that due to the nature of the problem we have rewritten $\hat{\mathcal{S}}$ using

$$\hat{C} = \begin{bmatrix} 1 + \frac{\tau}{\sqrt{\beta}} & & & \\ -1 & 1 + \frac{\tau}{\sqrt{\beta}} & & \\ & \ddots & \ddots & \\ & & -1 & 1 + \frac{\tau}{\sqrt{\beta}} \end{bmatrix}.$$

This was done as IKPIK needs to work with the inverse of $\hat{C}$ and in the old formulation C could not be inverted. As this method is designed for the classical Sylvester equation we transform the two systems during the preconditioning to become

$$(I_{n_t} \otimes M)^{-1} \left(I_{n_t} \otimes L + \hat{C} \otimes M \right) = \left(I_{n_t} \otimes M^{-1} L + \hat{C} \otimes I \right)$$

and similarly for the transpose equation. The inexactness in IKPIK [77] allows us to approximately solve the systems for the two matrices $M^{-1}L$ and $\hat{C}$. Note that in our case the matrix $\hat{C}$ is trivial to solve for and we employ algebraic multigrid combined with a few steps of a stationary iteration [19] for the solution with $M^{-1}L$. Note that this approach is not yet ideal as one should use methods design for generalized Sylvester equations. However, due to the limitation of the scope of this paper, here we refrain from these latter ideas and instead propose them as possible topics of future research.

These preconditioners are then embedded into Algorithm 1 via the preconditioning function outlined in Algorithm 3.

As was noted in [80], a time-periodic control problem where $y(0,.) = y(T,.)$ results in the matrix C having circulant structure and we can then make use of the Fourier transform to transform the Schur-complement system to a system with only block-diagonal matrices that are now of complex nature, which simplifies the Sylvester equation.

Similar matrix structures are obtained in [1] for the simultaneous discretization in space and time. Preconditioning results using tensor structures are found in [2, 24].

Algorithm 3: Preconditioner application: Aprec.

Input: $W_{11}, W_{12}, W_{21}, W_{22}, W_{31}, W_{32}$
Output: $Z_{11}, Z_{12}, Z_{21}, Z_{22}, Z_{31}, Z_{32}$
Solve $\tau M Z_{11} = W_{11}$
Solve $D_1 Z_{12} = W12$
Solve $\tau\beta M Z_{21} = W_{21}$
Solve $D_2 M Z_{22} = W_{22}$
Compute Z_{31} and Z_{32} as the low rank solution of $\hat{S}$ with right-hand side defined by W_{31} and W_{32}.

We have now obtained an overall low-rank algorithm for the computation of low-rank in time solutions to the PDE-constrained optimization problems. We want to comment on some of the algorithms properties before moving on to the existence of low-rank solutions in the next section. The computational cost of the algorithm is again dominated by the matrix vector product and the application of the preconditioner. Both of these are needed in any other iterative solver and the dimensionality of the low-rank factors determines how expensive the matrix vector products are. The cost of the preconditioner is here dominated by how efficiently we can solve the Schur-complement system. The stationary iteration suffers from the fact that the matrix structure does not result in parameter independent convergence. The solver based on standard Sylvester equations IKPIK typically shows much more parameter independent convergence behavior. We refer to section 6 for the numerical results. Nevertheless, more research is needed to employ solvers for generalized Sylvester equations. Additionally, the algorithm has to compute the truncation of the resulting matrices. This step is not free and one should use the full-space method for a small set of time-steps.

Nevertheless, the computation via a truncated SVD proved to be typically quite fast as we only needed to multiply with the large and skinny low-rank factor matrices.

4. Existence result of low-rank solutions. The previously derived low-rank method of course is competitive only if the solution to the optimal control problem exhibits a fast singular value decay, allowing us to replace it by a low-rank approximation. It thus remains to show that this is indeed a reasonable assumption for problems of the form (2.6). For this reason, in this section we establish a direct connection between (2.6) and the more prominent Sylvester equation

$$AX + XB = \tilde{C}, \tag{4.1}$$

where $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{m\times m}$, and $\tilde{C} \in \mathbb{R}^{n\times m}$. For the case that $\tilde{C}$ is of low-rank, i.e., $\tilde{C} = W_{\tilde{C}}V_{\tilde{C}^T}$, $W_{\tilde{C}} \in \mathbb{R}^{n\times k}$, $V_{\tilde{C}} \in \mathbb{R}^{m\times k}$, and $k \ll n, m$, it is well-known (see, e.g., [36, 35, 55]) that there exist approximations $X_r = W_X V_X^T \approx X$ with $W_X \in \mathbb{R}^{n\times r}, V_X \in \mathbb{R}^{m\times r}$, and $r \ll n, m$. Moreover, recently there has been an increased interest in numerical methods that, rather than computing the true solution and computing an approximation afterward, solely work on low-rank factors and iteratively construct approximations X_r converging to the true solution X, making these approaches feasible for dimensions $n, m \sim 10^6$. Popular methods are projection-based methods (see [9, 30, 75]), ADI-based methods (see [8, 14, 12, 78]), and multigrid methods (see [37]).

Let us now consider the second block-row of (2.6), for which we obtain that

$$(D_2 \otimes \beta\tau M_2)\, u + \left(D_3 \otimes \tau N^T\right) p = 0.$$

Solving this equation for u and inserting the result into the third block-row of (2.6) gives

$$-\left(I_{n_t} \otimes L + C \otimes M\right) y - \frac{1}{\beta}\left(D_3 \otimes \tau N\right)\left(D_2^{-1} \otimes M_2^{-1}\right)\left(D_3 \otimes N^T\right) p = d,$$

which, due to the properties of the Kronecker product and the definition of D_3, can be simplified to

$$-\left(I_{n_t} \otimes L + C \otimes M\right) y - \frac{\tau}{\beta}\left(D_2^{-1} \otimes N M_2^{-1} N^T\right) p = d.$$

Together with the first blow-rock, we thus can reformulate (2.6) in matrix notation as

$$\begin{aligned} \tau M_1 Y D_1 - LP - MPC &= \tau M_1 Y_{obs} D_1, \\ -LY - MYC^T - \frac{\tau}{\beta} N M_2^{-1} N^T P D_2^{-1} &= D. \end{aligned}$$

So far, we have only eliminated the second block-row and rewritten the problem in its matrix form. For the connection to (4.1), we have to make some additional assumptions on our initial setup (2.1). Typically, in real-life applications we can only observe a small portion $\tilde{y}$ of the state rather than the full y. In other words, the mass matrix M_1 in this case can be replaced by a low-rank matrix $\tilde{C}_{obs}\tilde{C}_{obs}^T = M_1$, with $\tilde{C}_{obs} \in \mathbb{R}^{n_1\times \ell}$ determining the observable parts of y. Note that in the context of classical control theory, $\tilde{C}_{obs}$ simply denotes the measurable output quantity of interest within the state-space representation of a linear dynamical system; see [3, 44, 52].

Similarly, in the case of boundary control, the rectangular matrix $N \in \mathbb{R}^{n_1 \times m}$ usually contains significantly fewer columns than rows. In summary, this means that we are often interested in the solution $\begin{bmatrix} Y & P \end{bmatrix}$ of the linear matrix equation

(4.2)
$$L \begin{bmatrix} Y & P \end{bmatrix} \begin{bmatrix} 0 & -I \\ -I & 0 \end{bmatrix} + M \begin{bmatrix} Y & P \end{bmatrix} \begin{bmatrix} 0 & -C^T \\ -C & 0 \end{bmatrix} + \tilde{C}_{obs} \tilde{C}_{obs}^T \begin{bmatrix} Y & P \end{bmatrix} \begin{bmatrix} \tau D_1 & 0 \\ 0 & 0 \end{bmatrix} + N M_2^{-1} N^T \begin{bmatrix} Y & P \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & -\frac{\tau}{\beta} D_2^{-1} \end{bmatrix} = \begin{bmatrix} \tau M_1 Y_{obs} D_1 \\ D \end{bmatrix}.$$

Pre- and postmultiplying the previous equation by M^{-1} and $\left[\begin{smallmatrix} 0 & -I \\ -I & 0 \end{smallmatrix}\right]$ leads to a generalized Sylvester equation of the form

$$\mathcal{A}\mathcal{X} + \mathcal{X}\mathcal{B} + \mathcal{Q}_1 \mathcal{X} \mathcal{R}_1 + \mathcal{Q}_2 \mathcal{X} \mathcal{R}_2 = \mathcal{E}_1 \mathcal{F}_2^T,$$

where

$$\mathcal{A} = M^{-1}L,\ \mathcal{B} = \begin{bmatrix} C^T & 0 \\ 0 & C \end{bmatrix},\ \mathcal{Q}_1 = M^{-1}\tilde{C}_{obs}\tilde{C}_{obs}^T,$$

$$\mathcal{R}_1 = \begin{bmatrix} 0 & -\tau D_1 \\ 0 & 0 \end{bmatrix},\ \mathcal{Q}_2 = M^{-1} N M_2^{-1} N^T,\ \mathcal{R}_2 = \begin{bmatrix} 0 & 0 \\ -\frac{\tau}{\beta} D_2^{-1} & 0 \end{bmatrix},$$

$$\mathcal{E}_1 = \begin{bmatrix} M_1 W_{Y_{obs}} & W_D \end{bmatrix},\ \mathcal{F}_1 = \begin{bmatrix} D_1 V_{Y_{obs}} & 0 \\ 0 & V_D \end{bmatrix}.$$

Note that we assumed $Y_{obs} = W_{Y_{obs}} V_{Y_{obs}}^T$ and $Y_D = W_D V_D^T$ to be the low-rank representations for the right-hand side.

In what follows, we proceed as in [10, 22] and use the Sherman–Morrison–Woodbury formula [32] to simplify the previous equation. Since $\mathcal{Q}_1 = \mathcal{U}_1 \mathcal{V}_1^T$ and $\mathcal{Q}_2 = \mathcal{U}_2 \mathcal{V}_2^T$, for the Kronecker structured linear system, we subsequently obtain

$$\left(\mathcal{I} \otimes \mathcal{A} + \mathcal{B}^T \otimes \mathcal{I} + \mathcal{R}_1^T \otimes \mathcal{Q}_1 + \mathcal{R}_2^T \otimes \mathcal{Q}_2\right) \operatorname{vec}(\mathcal{X}) = \operatorname{vec}\left(\mathcal{E}_1 \mathcal{F}_1^T\right),$$

which can be rewritten as

$$\left(\underbrace{\mathcal{I} \otimes \mathcal{A} + \mathcal{B}^T \otimes \mathcal{I}}_{\tilde{A}} + \underbrace{\begin{bmatrix} \mathcal{I} \otimes \mathcal{U}_1 & \mathcal{I} \otimes \mathcal{U}_2 \end{bmatrix}}_{\tilde{U}} \underbrace{\begin{bmatrix} \mathcal{R}_1^T \otimes \mathcal{V}_1^T \\ \mathcal{R}_2^T \otimes \mathcal{V}_2^T \end{bmatrix}}_{\tilde{V}^T} \right) \operatorname{vec}(\mathcal{X}) = \operatorname{vec}\left(\mathcal{E}_1 \mathcal{F}_1^T\right).$$

According to the Sherman–Morrison–Woodbury formula, we alternatively get

$$\tilde{A} \operatorname{vec}(\mathcal{X}) = \operatorname{vec}\left(\mathcal{E}_1 \mathcal{F}_1^T\right) - \tilde{U} \underbrace{\left(I + \tilde{V}^T \tilde{A}^{-1} \tilde{U}\right)^{-1} \tilde{V}^T \tilde{A}^{-1} \operatorname{vec}\left(\mathcal{E}_1 \mathcal{F}_1^T\right)}_{\operatorname{vec}(\mathcal{Y})}.$$

Since we have

$$\begin{aligned} \tilde{U} \operatorname{vec}(\mathcal{Y}) &= \tilde{U} \operatorname{vec}\left(\begin{bmatrix} \mathcal{Y}_1 \\ \mathcal{Y}_2 \end{bmatrix} \right) \\ &= \begin{bmatrix} \mathcal{I} \otimes \mathcal{U}_1 & \mathcal{I} \otimes \mathcal{U}_2 \end{bmatrix} \operatorname{vec}\left(\begin{bmatrix} \mathcal{Y}_1 \\ \mathcal{Y}_2 \end{bmatrix} \right) \\ &= \operatorname{vec}(\mathcal{U}_1 \mathcal{Y}_1) + \operatorname{vec}(\mathcal{U}_2 \mathcal{Y}_1) \end{aligned}$$

we can conclude that

$$\tilde{U}\,\mathrm{vec}\,(\mathcal{Y}) =: \mathrm{vec}\left(\mathcal{E}_2\mathcal{F}_2^T\right)$$

with $\mathcal{E}_2 \in \mathbb{R}^{n_1\times(l+m)}$, $\mathcal{F}_2 \in \mathbb{R}^{2n_t\times(l+m)}$. In particular, this implies

$$\tilde{A}\,\mathrm{vec}\,(\mathcal{X}) = \mathrm{vec}\left(\mathcal{E}_1\mathcal{F}_1^T\right) - \mathrm{vec}\left(\mathcal{E}_2\mathcal{F}_2^T\right)$$

or, in other words, $\mathcal{X}$ can also be derived as the solution to a regular Sylvester equation of the form

$$\mathcal{A}\mathcal{X} + \mathcal{X}\mathcal{B} = \begin{bmatrix}\mathcal{E}_1 & -\mathcal{E}_2\end{bmatrix}\begin{bmatrix}\mathcal{F}_1^T\\ \mathcal{F}_2^T\end{bmatrix}.$$

We have now established that the PDE-constrained optimization problem can be written in form of a classical Sylvester equation for which we can use the existence results for a low-rank solution introduced in [36]. Note that we do not claim to actually proceed this way in order to compute the solution matrix $\mathcal{X}$. Obviously, determining the intermediate solution $\mathrm{vec}\,(\mathcal{Y})$ would be a challenge on its own. The previous steps rather should be understood as a theoretical evidence for the assumption that $\mathcal{X}$ indeed exhibits a very strong singular value decay. Keep in mind that we had to assume that the desired state Y_{obs} as well as D are of low-rank and that $l, m \ll n_1$, which is a reasonable assumption for realistic control problems.

A special case. One might argue that for the distributed control case, i.e., N begin square together with an (almost) entirely observable state, i.e., $\tilde{C}_{obs}\tilde{C}_{obs}^T = M_1$, the previous low-rank assumptions no longer hold true. Consequently, applying the Sherman–Morrison–Woodbury formula will not simplify (4.2) and we thus will have to deal with a linear matrix equation of the form

$$\sum_{i=1}^{4}\mathcal{A}_i\mathcal{X}\mathcal{B}_i = \mathcal{E}_1\mathcal{F}_1^T, \tag{4.3}$$

where we cannot benefit from additional structure in $\mathcal{A}_i$ and $\mathcal{B}_i$. Still, as has already been (numerically) observed and partially discussed in [10, 11, 22] for the special Lyapunov type case, i.e., $\mathcal{B}_i = \mathcal{A}_i^T$, the solution matrix $\mathcal{X}$ still seems to exhibit similar low-rank properties.

Although the most general case certainly is an interesting topic of future research, we want to conclude by pointing out that for the special case $M_2 = M_1 = N = M$ we immediately get an analogous (in fact even stronger) low-rank existence result for (4.2). This is due to the fact that here (4.2) is equivalent to the Sylvester equation

$$L\begin{bmatrix}Y & P\end{bmatrix}\begin{bmatrix}0 & -I\\ -I & 0\end{bmatrix} + M\begin{bmatrix}Y & P\end{bmatrix}\begin{bmatrix}\tau D_1 & -C^T\\ -C & -\frac{\tau}{\beta}D_2^{-1}\end{bmatrix} = \begin{bmatrix}\tau M Y_{obs} D_1\\ D\end{bmatrix}$$

for which we again can apply the low-rank existence results from [36].

5. Other state equations.

Stokes equation. In addition to the heat equation as a test problem we here also consider the Stokes equation. The discretization of the Stokes control problem can

be performed similarly to the case for the heat equation and we refer the interested reader to [27]. The Stokes equations are given by

$$y_t - \nu \triangle y + \nabla p = u \text{ in } [0,T] \times \Omega, \tag{5.1}$$

$$-\nabla \cdot y = 0 \text{ in } [0,T] \times \Omega, \tag{5.2}$$

$$y(t,.) = g(t) \text{ on } \partial\Omega, t \in [0,T], \tag{5.3}$$

$$y(0,.) = y^0 \text{ in } \Omega, \tag{5.4}$$

and the functional we are interested in is defined as

$$J(y,u) = \frac{1}{2}\int_0^T \int_{\Omega_1} (y - \bar{y})^2 \, dxdt + \frac{\beta}{2}\int_0^T \int_{\Omega_2} u^2 dxdt. \tag{5.5}$$

Our goal is to build a discrete Lagrangian using an all-at-once approach [82], where we set up a discrete problem within the space-time cylinder. Using the trapezoidal rule and $Q2/Q1$ finite elements in space leads to the following discrete objective function:

$$J(y,u) = \frac{\tau}{2}\left(y - y_{obs}\right)^T \mathcal{M}_1 \left(y - y_{obs}\right) + \frac{\tau\beta}{2} u^T \mathcal{M}_2 u \tag{5.6}$$

with

$$\mathcal{M}_1 = \text{blkdiag}\left(\frac{1}{2}M_1, 0, M_1, 0, \ldots, M_1, 0, \frac{1}{2}M_1, 0\right),$$

$$\mathcal{M}_2 = \text{blkdiag}\left(\frac{1}{2}M_2, M_2, \ldots, M_2, \frac{1}{2}M_2\right),$$

where we reuse the notation for the heat equation. Note that for the Stokes case the vectors y_i are split into a velocity v part with $d = 2, 3$ components and pressure part p, i.e.,

$$y_i = \begin{bmatrix} y_i^v \\ y_i^p \end{bmatrix}.$$

The all-at-once discretization of the state equation using finite elements in space and an implicit Euler scheme in time is given by

$$\mathcal{K}y - \tau\mathcal{N}u = d, \tag{5.7}$$

where

$$\mathcal{K} = \begin{bmatrix} \mathcal{L} & & & \\ -\mathcal{M} & \mathcal{L} & & \\ & \ddots & \ddots & \\ & & -\mathcal{M} & \mathcal{L} \end{bmatrix}, \quad \mathcal{N} = \begin{bmatrix} N & 0 & & & & \\ 0 & 0 & 0 & & & \\ 0 & N & 0 & & & \\ & 0 & & & & \\ & & & \ddots & & \\ & & & & 0 & N \\ & & & & 0 & 0 \end{bmatrix}, \quad d = \begin{bmatrix} Ly_0 + f \\ 0 \\ f \\ \vdots \\ f \\ 0 \end{bmatrix}.$$

Here,

$$\mathcal{L} = \begin{bmatrix} L & B^T \\ B & 0 \end{bmatrix}$$

represents an instance of a time-dependent Stokes problem with B the discrete divergence, M is the mass matrix for the domain Ω, the matrix L is defined as $L = \tau^{-1}M + K$, the matrix N is a rectangular matrix that can be written as

$$D_3 \otimes \mathcal{N}_s \text{ with } \mathcal{N}_s = \begin{bmatrix} N \\ 0 \end{bmatrix}$$

which represents the distributed control term control term where $N = M$, the matrix

$$\mathcal{M} = \begin{bmatrix} \tau^{-1}M & 0 \\ 0 & 0 \end{bmatrix}$$

is associated with the discretization in time via the implicit Euler scheme, and the right-hand side d consists of a contribution from the initial condition y_0 and a vector f representing forcing terms and contributions of boundary conditions. Note that all matrices here correspond to the ones introduced for the heat equation but equipped with a block form corresponding to the components for the velocity y_v and pressure y_p. The first order conditions using a Lagrangian with Lagrange multiplier p lead to the system

$$\underbrace{\begin{bmatrix} \tau\mathcal{M}_1 & 0 & -\mathcal{K}^T \\ 0 & \beta\tau\mathcal{M}_2 & \mathcal{N}^T \\ -\mathcal{K} & \mathcal{N} & 0 \end{bmatrix}}_{\mathcal{A}} \begin{bmatrix} y \\ u \\ p \end{bmatrix} = \begin{bmatrix} \tau\mathcal{M}_1 y_{obs} \\ 0 \\ d \end{bmatrix}, \tag{5.8}$$

where again we can switch to a Kronecker structure defined by

$$\begin{bmatrix} D_1 \otimes \tau\mathcal{M} & 0 & -\left(I_{n_t} \otimes \mathcal{L} + C^T \otimes \mathcal{M}\right) \\ 0 & D_2 \otimes \beta\tau M_2 & D_3 \otimes \mathcal{N}_s^T \\ -\left(I_{n_t} \otimes \mathcal{L} + C \otimes \mathcal{M}\right) & D_3 \otimes \mathcal{N}_s & 0 \end{bmatrix}. \tag{5.9}$$

We can now in a similar way as earlier use the low-rank MINRES method. Again, here we apply a block-diagonal preconditioner of the form

$$P = \begin{bmatrix} D_1 \otimes \tau\hat{M}_1 & 0 & 0 \\ 0 & D_2 \otimes \beta\tau\hat{M}_2 & 0 \\ 0 & 0 & \hat{S} \end{bmatrix}. \tag{5.10}$$

Here $\hat{M} = \text{blkdiag}\,(\hat{M}_1, \gamma I)$ with $\gamma = \beta\tau h^d$ (see [82] for details). Here d is the dimension of the problem ($d = 2, 3$) and h the mesh parameter. The matrices M_1 and M_2 are approximated via a Chebyshev semi-iteration [33, 34, 86], or in the case of lumped mass matrices we trivially have $\hat{M}_{1,2} = M_{1,2}$. The approximation of the Schur-complement is much more tricky in this case as for the indefinite $\mathcal{M}_1$ the Schur-complement is not well-defined. Thus, we again use the approximation $\hat{M} = \text{blkdiag}\,(\hat{M}_1, \gamma I)$ to form an approximate Schur-complement

$$S = \tau^{-1}\mathcal{K}\hat{\mathcal{M}}_1^{-1}\mathcal{K}^T + \tau^{-1}\beta^{-1}\mathcal{N}\mathcal{M}_2^{-1}\mathcal{N}^T$$

with $\hat{\mathcal{M}}_1$ a block-diagonal involving $\hat{M}$. We in turn approximate this via

$$\hat{S} = \tau^{-1}\left(\mathcal{K} + \hat{\mathcal{M}}\right)\mathcal{M}_1^{-1}\left(\mathcal{K} + \hat{\mathcal{M}}\right)^T,$$

where $\hat{\mathcal{M}} = \text{blkdiag}(\frac{1}{\sqrt{\beta}}M_1, 0, \ldots, \frac{1}{\sqrt{\beta}}M_1, 0)$ for the distributed control case. As in section 3 we note that the matrix

$$\left(\mathcal{K} + \hat{\mathcal{M}}\right) = \left(I_{n_t} \otimes \tilde{\mathcal{L}} + C \otimes \mathcal{M}\right)$$

with $\tilde{\mathcal{L}} = \left[\begin{smallmatrix} (\tau^{-1}+\beta^{-1/2})M_1+K & B^T \\ B & 0 \end{smallmatrix}\right]$. We now proceed in the following way. A stationary iteration scheme with a fixed number of steps is used to approximately solve $(I_{n_t} \otimes \tilde{\mathcal{L}} + C \otimes M)$ with preconditioner $(I_{n_t} \otimes \tilde{\mathcal{L}})$ and within this preconditioner systems with $\tilde{\mathcal{L}}$ are approximately solved using another Uzawa scheme with a fixed but small number of iterations. For this inner Uzawa iteration the inverse of the preconditioner is given by

$$\begin{bmatrix} \left[\left(\tau^{-1} + \beta^{-1/2}\right) M_1 + K\right]^{-1}_{MG} & 0 \\ 0 & \left(\tau^{-1} + \beta^{-1/2}\right) [K_p]^{-1}_{MG} + [M_p]^{-1}_{MG} \end{bmatrix},$$

where the $[\ldots]^{-1}_{MG}$ indicates the use of a geometric [41, 87] or algebraic multigrid method [69, 29]. Preconditioners of this type are of so-called Cahouet–Chabard form [21] and the derivation can be done using a least squares commutator approach [27, 82].

Again, for more robustness, more sophisticated Sylvester solvers should be used in the future to guarantee robustness with respect to the system parameters.

Convection-diffusion equation. Before coming to the numerical results we quickly want to introduce the last state equation considered here. The PDE constraint is now given by the convection-diffusion equation

$$y_t - \varepsilon \triangle y + w \cdot \nabla y = u \text{ in } \Omega, \tag{5.11}$$

$$y(:, x) = f \text{ on } \partial\Omega, \tag{5.12}$$

$$y(0, :) = y_0 \tag{5.13}$$

as the constraint to the following objective function:

$$J(y, u) = \frac{1}{2}\int_0^T \int_{\Omega_1} (y - \bar{y})^2 \, dxdt + \frac{\beta}{2}\int_0^T \int_{\Omega_2} u^2 dxdt. \tag{5.14}$$

Note that the parameter ε is crucial to the convection-diffusion equation as a decrease in its value is adding more hyperbolicity to the PDE where the wind w is predefined. Such optimization problems have recently been discussed in [67, 43, 62] and for brevity we do not discuss the possible pitfalls regarding the discretization. Here we focus on a discretize-then-optimize scheme using a streamline upwind Galerkin (SUPG) approach introduced in [20]. Note that other schemes, such as discontinuous Galerkin methods [83] or local projection stabilization [62], are typically more suitable discretizations for the optimal control setup as they often provide the commutation between optimize first or discretize first for the first order conditions. Nevertheless, our approach will also work for these discretizations. Once again we employ a trapezoidal rule in connection with finite elements and now additionally use a SUPG stabilization. The discretized objective function and state equation are given by

$$J(y, u) = \frac{\tau}{2}(y - y_{obs})^T \mathcal{M}_1 (y - y_{obs}) + \frac{\tau\beta}{2} u^T \mathcal{M}_2 u,$$

which is the same as for the heat equation case. For the all-at-once discretization of the convection-diffusion equation we get the same structure as before,

$$\mathcal{K}y - \tau\mathcal{N}u = d, \tag{5.15}$$

with

$$\mathcal{K} = \begin{bmatrix} L_s & & & \\ -M_s & L_s & & \\ & \ddots & \ddots & \\ & & -M_s & L_s \end{bmatrix}, \quad \mathcal{N} = \begin{bmatrix} M_s & & & \\ & M_s & & \\ & & \ddots & \\ & & & M_s \end{bmatrix}, \quad d = \begin{bmatrix} M_1 y_0 + f \\ f \\ \vdots \\ f \end{bmatrix}.$$

Note that due to the SUPG test functions used we now have M_s, which is obtained entrywise from evaluating the integrals

$$(M_s)_{ij} = \int_\Omega \phi_i \phi_j + \delta \int_\Omega \phi_i \, (w \cdot \nabla \phi_j),$$

where ϕ are the finite element test functions and δ is a parameter coming from the use of SUPG [20, 27]. We then have $L_s = M_s + \tau K_s$, where K_s is the standard nonsymmetric matrix representing the SUPG discretization of the convection-diffusion equation. We can now see that this again is of the desired Kronecker form

$$\begin{bmatrix} D_1 \otimes \tau M_1 & 0 & -\left(I_{n_t} \otimes L_s + C \otimes M_s\right)^T \\ 0 & D_2 \otimes \beta\tau M_2 & D_3 \otimes \tau M_s^T \\ -\left(I_{n_t} \otimes L_s + C \otimes M_s\right) & D_3 \otimes \tau M_s & 0 \end{bmatrix} \begin{bmatrix} y \\ u \\ p \end{bmatrix} = \begin{bmatrix} D_1 \otimes \tau M_1 y_{obs} \\ 0 \\ d \end{bmatrix}. \tag{5.16}$$

Again, we employ the low-rank version of MINRES to solve this system. Note that for nonsymmetric formulations such as the one obtained from an optimize-then-discretize strategy we can also use low-rank versions of nonsymmetric Krylov solvers such as GMRES [72] or BICG [31]. A preconditioner is of the form

$$P = \begin{bmatrix} D_1 \otimes \tau \hat{M}_1 & 0 & 0 \\ 0 & D_2 \otimes \beta\tau \hat{M}_2 & 0 \\ 0 & 0 & \hat{S} \end{bmatrix}, \tag{5.17}$$

where the two blocks involving mass matrices are as before and the Schur-complement of $\mathcal{A}$

$$\begin{aligned} S = {} & \left(I_{n_t} \otimes L_s + C \otimes M_s\right) \left(D_1^{-1} \otimes \tau^{-1} M_1^{-1}\right) \left(I_{n_t} \otimes L_s + C \otimes M_s\right)^T \\ & + \left(D_3 \otimes \tau M_s\right) \left(D_2^{-1} \otimes \beta^{-1}\tau^{-1} M_2^{-1}\right) \left(D_3 \otimes \tau M_s^T\right). \end{aligned} \tag{5.18}$$

As before with the heat and Stokes problem the aim for an efficient approximation of S is by not dropping terms but rather to create an approximation that matches both terms in S. In a similar way to the technique introduced in [63] for the steady case can now be extended by introducing a term $\hat{D} \otimes \hat{M}$ so that

$$\begin{aligned} & \left(\hat{D} \otimes \hat{M}\right) \left(D_1^{-1} \otimes \tau^{-1} M_1^{-1}\right) \left(\hat{D} \otimes \hat{M}\right)^T \\ & \quad \approx \left(D_3 \otimes \tau M_s\right) \left(D_2^{-1} \otimes \beta^{-1}\tau^{-1} M_2^{-1}\right) \left(D_3 \otimes \tau M_s^T\right). \end{aligned}$$

If we now assume $D_1 = D_2 = D_3 = I_{n_t}$ one can obtain

$$\hat{S} = \left(I_{n_t} \otimes \hat{L}_s + C \otimes M_s\right) \left(D_1^{-1} \otimes \tau^{-1} M_1^{-1}\right) \left(I_{n_t} \otimes \hat{L}_s + C \otimes M_s\right)^T, \tag{5.19}$$

where

$$\hat{L}_s = \left(1 + \frac{1}{\sqrt{\beta}}\right) M_s + \tau K_s.$$

Note that it is of course possible to construct such an approximation without assuming $D_1 = D_2 = D_3 = I_{n_t}$. We can now proceed with a stationary iteration for the two Sylvester equations in $\hat{S}$, or if we want to employ IKPIK again we rewrite the approximation as

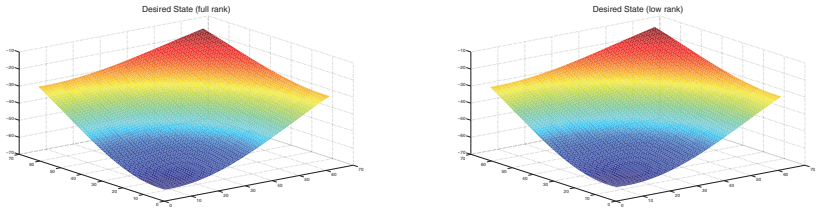
$$\hat{S} = \left(I_{n_t} \otimes L_s + \hat{C} \otimes M_s\right) (D_1^{-1} \otimes \tau^{-1} M_1^{-1}) \left(I_{n_t} \otimes L_s + \hat{C} \otimes M_s\right)^T \tag{5.20}$$

with

$$\hat{C} = \begin{bmatrix} 1 + \frac{\tau}{\sqrt{\beta}} & & & \\ -1 & 1 + \frac{\tau}{\sqrt{\beta}} & & \\ & \ddots & \ddots & \\ & & -1 & 1 + \frac{\tau}{\sqrt{\beta}} \end{bmatrix}.$$

Now when solving with $\hat{S}$ we can approximately solve both $(I_{n_t} \otimes L_s + \hat{C} \otimes M_s)$ and its transpose using IKPIK, where we use an algebraic multigrid for the approximate solution with the stiffness matrix L_s. These preconditioners need to be adapted when other discretizations are used, but we expect our results to carry over to these cases as well.

6. Numerical results. We are now going to present results for the low-rank in time solution of certain PDE-constrained optimization problems. The results presented in this section are based on an implementation of the above described algorithms within MATLAB, whereas we perform the discretization of the PDE-operators within the deal.II [7] framework using $Q1$ finite elements for the heat equation and the convection-diffusion equation. The Stokes equation is discretized with $Q2$ elements for the velocity and $Q1$ elements for the pressure component. For the algebraic multigrid approximation, we used HSL MI_20 [19]. For some preconditioner we used backslash within MATLAB for the innermost solution within the preconditioner. Our implementation of MINRES is based on a vector version presented in [27] and was stopped with a tolerance of 10^{-4} or 10^{-6} for the relative pseudoresidual. Please note that we ideally should couple the stopping criterion for the iterative solver to the discretization error. We also give results for the stationary iteration and IKPIK used for the approximation within the preconditioner. Our experiments are performed for a final time $T = 1$ with a varying number of time-steps. As the domain Ω we consider the unit cube, but other domains are of course possible. We specify the boundary conditions for each problem separately. Throughout the results section we fixed the truncation at 10^{-8}, for which we observed good results. Additionally, we also performed not-listed experiments with a tolerance of 10^{-10} for which we also observed fast convergence. Larger tolerances should be combined with a deeper analysis of the algorithms and a combination with flexible outer solvers. All results are performed on a standard Ubuntu desktop Linux machine with Intel Xeon CPU W3503 @ 2.40 GHz and 6 GB of RAM.



(a) Desired state $\bar{\mathbf{y}}_{10}$—full rank (b) Desired state $\bar{\mathbf{y}}_{10}$—low rank

FIG. 1. *Desired state in full-rank and low-rank form.*

TABLE 1

Results for full-rank MINRES versus low-rank (LR) MINRES with stationary iteration preconditioner for 20 *or* 100 *time-steps and a variety of different meshes. Both iteration numbers and computing times in seconds are listed. DoF = degrees of freedom. OoM indicates out of memory in MATLAB. Results are shown for* $\beta = 10^{-4}$.

DoF	FR (20) # it(t)	LR (20) # it(t)	FR (100) # it(t)	LR (100) # it(t)
289	17(0.2)	15(5.6)	31(1.1)	19(7.6)
1089	19(0.7)	17(8.9)	33(5.56)	21(11.9)
4225	19(3.5)	17(19.7)	35(26.6)	23(26.3)
16641	21(17.5)	19(72.1)	35(125.8)	23(97.6)
66049	23(81.8)	19(324.8)	OoM	25(427.4)

6.1. The heat equation.

Distributed control. As the first example shown in this section we use the heat equation with a distributed control term. We choose the boundary conditions for this problem to be of zero Dirichlet type. We first show how well the desired state

$$y_{obs} = -64 \exp\left(-\left((x_0 - 0.5t)^2 + (x_1 - 0.5t)^2\right)\right)$$

is approximated in low-rank form. Figure 1 illustrates this for grid point 10 in time where the right-hand side $\mathrm{vec}^{-1}(\tau \mathcal{M}_1 y_{obs}) = B_{11} B_{12}^T$ is approximated by low-rank factors of rank 2.

Table 1 shows first results for the comparison of the full-rank MINRES versus the low-rank version. We want to point out that here we use the backslash operator in MATLAB to evaluate the matrix L within the preconditioner, but this can easily be replaced by a multigrid approximation and in fact is done later. For the full-rank version, we only used a block-diagonal approximation for the matrix $\mathcal{K}$ and hence the robustness with respect to changes in the number of time-steps is not given. This would typically be the case and our results using deal.II and C++ in [81, 61] show robustness with respect to the number of time-steps. Nevertheless, every increase in the number of time-steps also results in an increase in the matrix size and so one would expect when the number of time-steps is increased fivefold that the same happens for the time needed to solve the linear system. Going back to the results in Table 1, where both the timings and iteration numbers are shown for a variety of mesh-sizes and two different orders of grid points in time, both methods perform mesh-independent and we can see that the low-rank method shows almost no increase when the number of

TABLE 2

Results for full-rank MINRES versus low-rank MINRES with both the stationary iteration (SI) and IKPIK *for a fixed mesh with* $16,641$ *unknowns in space. We show varying time-steps and additionally the rank of the state/control/adjoint state. Both iteration numbers and computing times in seconds are listed. Results are shown for* $\beta = 10^{-4}$.

DoF	20	100	200	400	600
16641	# it(t)	# it(t)	# it(t)	# it(t)	# it(t)
LR(SI)	19(108.2)	21(307.8)	25(432.7)	43(671.9)	61(937.3)
LR(IKPIK)	19(115.1)	19(288.9)	19(296.8)	21(335.3)	21(357.1)
Rank (SI)	8/10/10	10/11/11	12/13/13	11/14/14	14/15/15
FR	21(18.3)	35(124.0)	63(434.3)	OoM	OoM

time-steps is drastically increased. Note also the degrees of freedom given here are only for the spatial discretization. The overall dimension of the linear system is then given by $3nn_t$, where n represents the spatial degrees of freedom. We see that the iteration times for the full rank solver go up, and using the nonoptimal preconditioners we additionally see that the times increase more than just by a factor of five. We also see that due to the cost of performing a low-rank truncation the full-rank method always outperforms the low-rank scheme for a small number of time-steps. Nevertheless, the low-rank method can easily solve problems that are no longer tractable for full-rank methods.

Next we compare how both the full-rank and the low-rank method perform when the number of time-steps is further increased. We therefore consider a fixed mesh for a varying time-discretization. Table 2 shows the results for both the full-rank and the low-rank method. We additionally show the rank of the three components of the state, control, and adjoint state. We started computing the truncation process using a maximum size of the truncated SVD of 20, which was sufficient for all discretizations in time using a truncation tolerance of 10^{-8}. In order to keep the iteration numbers from growing too much with an increase in the number of time-steps we increased the number of stationary iterations for the preconditioner from two to three for the last two columns in Table 2. Additionally, we show the results for the Schur-complement approximation when IKPIK is employed. For this we employ IKPIK with a fixed number of steps. We used four steps for the results shown in Table 2. For the evaluation of the inverse of the stiffness matrix within IKPIK we used a five steps of a stationary iteration with one cycle of an algebraic multigrid as a preconditioner. The algebraic multigrid is also used within the stationary iteration approximation to the Sylvester equation. We see again that the full-rank method exceeds the memory limit in MATLAB. It can also be seen that the increase in rank and computing time is typically moderate. Note that the system dimension considering a full-rank solution is ranging from $998,460$ to $29,953,800$ unknowns.

In order to illustrate the distribution of the singular values we show in Figure 2 how the relative value of the singular values behaves throughout the iteration. Shown are the scaled singular values (σ_j/σ_1) of the approximation to the state for the problem with 4225 unknowns and 100 grid points in time. In Table 3 we illustrate the performance of our scheme when different values for the regularization parameter are considered. So far the preconditioners introduced based on the stationary iteration have used a direct solver for the solution of the systems with

$$\hat{L} = \left(\left(1 + \frac{\tau}{\sqrt{\beta}} \right) M_1 + \tau K \right)$$

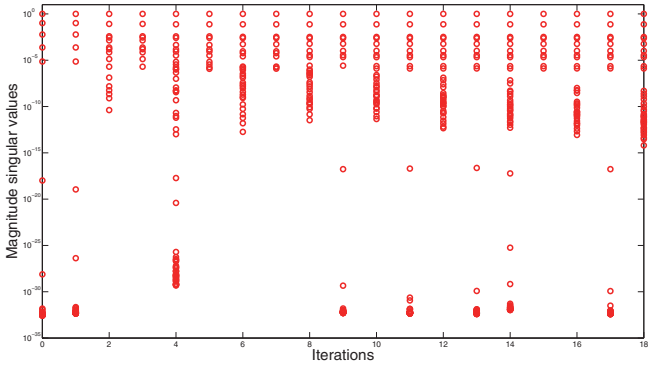


FIG. 2. *Singular values of the approximate solution during the iteration before truncation.*

TABLE 3

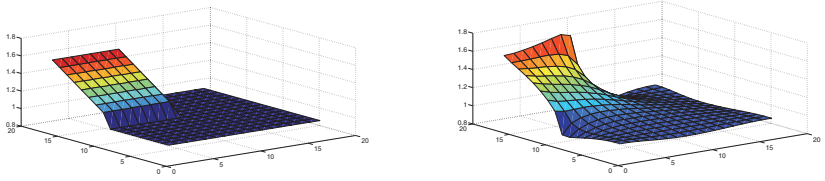
Results for low-rank MINRES with 100 *time-steps and a varying regularization parameter on three different meshes. Both iteration numbers and computing times in seconds are listed. The results shown use the* IKPIK *approximation of the Sylvester-type operators.*

DoF β	1089 (100) # it(t)	4225 (100) # it(t)	16641 (100) # it(t)	16641 (100) $\|\mathbf{y}-\bar{\mathbf{y}}\| \,/\, \|\bar{\mathbf{y}}\|$
10^{-4}	17(25.8)	19(80.1)	19(311.9)	0.3979
10^{-6}	17(23.8)	17(64.7)	19(267.5)	0.2019
10^{-8}	15(20.3)	17(56.7)	19(227.2)	0.0809

both in the full-rank method and the low-rank one. We now illustrate that we can easily approximate this matrix using an algebraic multigrid technique by also showing that our preconditioner performs robustly with respect to the regularization parameter β. We here compute the truncated singular value decomposition up to order 20 and then cut off corresponding to the truncation tolerance. We additionally increased the number of stationary iteration steps for the matrix $(I_{n_t} \otimes \hat{L} + C \otimes M)$ with preconditioner $(I_{n_t} \otimes \hat{L})$ to 4.

Boundary control. In the following we demonstrate that our approach also works for the case of a boundary control problem. The desired state is shown in Figure 3(a) and the computed state needed to approximate this in Figure 3(b). In Table 4 we show results for the low-rank MINRES approximation for a variety of mesh-parameters and regularization parameters. Details on the preconditioners used can be found in [61]. As in the last example for the distributed control case we choose four Uzawa iterations and a tolerance of 10^{-4} for the iterative solver. Here we evaluate $\hat{L}$ again using the backslash operator in MATLAB but the use of AMG is straightforward. We do not employ IKPIK as its use for this setup needs to be further investigated.

6.2. Stokes equation. The configuration for the Stokes equation is taken from [82] and originally appeared in [46]. The spatial domain is the unit cube $\Omega = [0,1]^d$ with a time domain $[0,1]$. The target flow is the solution for an unsteady Stokes



(a) Desired state at grid point 65 in time.

(b) Computed state at grid point 65 in time for $\beta = 10^{-6}$.

FIG. 3. *Desired state and computed state for a boundary control problem.*

TABLE 4

Results for low-rank MINRES with 100 *time-steps and a varying regularization parameter on three different meshes for a boundary control example. Both iteration numbers and computing times in seconds are listed.*

DoF	289 (100)	4225 (100)	16641 (100)
β	# it(t)	# it(t)	# it(t)
10^{-2}	49(137.3)	61(236.18)	79(802.7)
10^{-4}	67(179.8)	99(406.6)	151(1510.6)
10^{-6}	63(169.2)	95(380.4)	147(1448.6)

TABLE 5

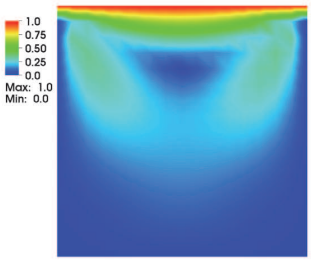
Results for low-rank MINRES with 100 *time-steps and a varying regularization parameter on three different meshes for a Stokes control example. Both iteration numbers and computing times in seconds are listed.*

DoF	578+81 (100)	2178+289 (100)	8450+1089 (100)
β	# it(t)	# it(t)	# it(t)
10^{-1}	11(224.4)	12(624.8)	14(3601.9)
10^{-5}	15(290.2)	15(737.6)	17(4091.5)

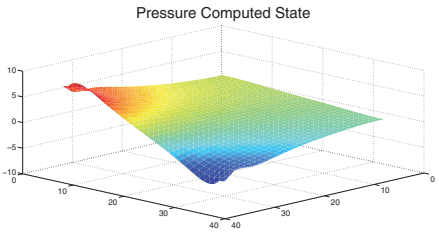
equation with Dirichlet boundary conditions, i.e., $y = (1, 0)$ when the second spatial component $x_2 = 1$ and $y = (0, 0)$ on the remaining boundary for the two-dimensional case. For the control problem we now consider the following time-dependent boundary conditions. For the top-boundary where $x_2 = 1$ we get $y = (1 + \frac{1}{2}\cos(4\pi t - \pi), 0)$ and zero elsewhere in two space dimensions and we set viscosity to $1/100$. Figure 4(a) depicts the desired state and the corresponding computed pressure is shown in Figure 4(b). For the results shown in Table 5 we note that we needed to set the number of stationary iteration steps for the outer iteration to 30 and for the inner one for the small saddle point system to 5. We believe that the outer iteration can be replaced by a robust Sylvester solver.

Apart from the approximation of the Neumann–Laplacian on the pressure space whose inverse was evaluated using an algebraic multigrid scheme, we simply used the backslash operator to evaluate the remaining components. A further increase in computational efficiency can be gained when these are replaced by multigrid approximations.

6.3. Convection-diffusion equation. The configuration for the convection-diffusion equation is taken from [27] and is typically referred to as the double glazing problem. The spatial domain is the unit cube $\Omega = [-1, 1]^2$ with a time domain $[0, 1]$.



(a) Desired state at grid point 10 in time.



(b) Pressure of the computed state for $\beta = 1e-1$.

FIG. 4. *Desired state and computed pressure for the Stokes flow problem.*

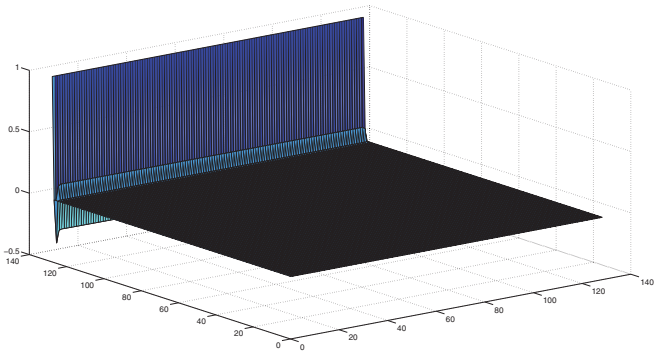


FIG. 5. *Computed state for* $\beta = 10^{-6}$ *at grid point* 10 *in time.*

The wind w is given by

$$w = \left(2y(1-x^2), -2x(1-y^2)\right).$$

Here we set the parameter to ε to $1/200$ and the boundary condition is a Dirichlet zero condition with the exception of $y = 1$ when $x_2 = 1$. The desired state is set to zero throughout the domain [62]. In Figure 5 we show the computed state for grid point 10 in time. Due to the nonsymmetric nature of the PDE operator we have not employed the recommended multigrid technique [66] and simply used the backslash operator here. The results shown in Table 6 indicate a robust performance of the low-rank MINRES method. We here set the number of stationary iterations to 15. We additionally show iteration numbers for the use of IKPIK with the use of a direct solver and a fixed number of IKPIK steps. The tolerance for MINRES is set to 10^{-6}.

Additionally, we show in Table 7 the numerical ranks that we obtain for the computed state. We hereby change the parameter ε in order to make the problem more hyperbolic. The tolerance for the low-rank truncation is chosen to be 10^{-8}.

TABLE 6

Results for low-rank MINRES with 100 *or* 200 *time-steps and a varying regularization parameter on a variety of meshes for a convection-diffusion control example. Both iteration numbers and computing times in seconds are listed. The tolerance for convergence is* $1e-6$.

	DoF	1089 (100)	4225 (100)	16641 (100)	4225 (200)	16641 (200)
	β	# it(t)	# it(t)	# it(t)	# it(t)	# it(t)
SI	10^{-8}	4(17.36)	6(66.1)	6(291.1)	6(68.8)	6(277.6)
IKPIK	10^{-8}	4(8.5)	6(35.6)	6(151.0)	6(36.7)	6(161.4)
SI	10^{-6}	10(40.4)	10(109.7)	14(639.5)	12(128.1)	14(646.3)
IKPIK	10^{-6}	10(22.3)	10(72.3)	14(520.3)	12(98.7)	14(568.6)

TABLE 7

Results for low-rank MINRES with 100 *time-steps and a fixed mesh with* 4225 *DoFs. We here show the ranks of the low-rank factors with respect to a varying* ε *within the convection-diffusion equation. The tolerance for convergence is* $1e-6$.

DoF	1/50	1/500	1/5000	1/50000
$\bar{\mathbf{y}}_1$ (trunctol $= 1e-8$)	9/7/7	7/6/6	7/5/5	7/5/5
$\bar{\mathbf{y}}_2$ (trunctol $= 1e-8$)	30/35/35	29/49/49	29/50/50	29/50/50
$\bar{\mathbf{y}}_2$ (trunctol $= 1e-5$)	4/3/3	3/4/4	3/4/4	3/4/4

We show ranks for both the zero desired state $\bar{\mathbf{y}}_1$ and a different desired state with a higher frequency $\bar{\mathbf{y}}_2$. We note that the control and adjoint states both need more terms for the low-rank representation when the desired state is nontrivial. The maximum number of vectors stored was limited to 50, which was not sufficient for small values of ε. This does not indicate that the method fails in this case but rather that it is crucial to investigate the relation between the discretization error, the algebraic error, and the truncation error. The truncation tolerance of 10^{-8} could have simply been too tight for the level of discretization. Hence, we additionally show the results for the truncation level 10^{-5}, which in this case is smaller than h^2.

7. Outlook. We believe that the research presented here opens some interesting angles that should be studied in the future. The incorporation of additional constraints such as control and state constraints is typically very important for real-world scenarios. We plan to investigate a technique introduced in [45] where the state and adjoint state are computed first and hence is amenable to low-rank techniques, and then the constrained control is computed. It is further desired to investigate more complicated discretizations in time. Of particular interest, we want to study backward differentiation formulas [5] as these can be easily incorporated simply modifying the C matrix in (3.1). We further plan to incorporate more sophisticated generalized Sylvester equation solvers for $(I_{n_t} \otimes L + C \otimes M)$, which we believe allows for more robustness with respect to the system parameters and should be combined with a flexible outer method [71]. It is further crucial to investigate how the low-rank techniques can be extended to incorporate nonlinearities of both the objective function and the PDE constraint such as [17].

8. Conclusions. In this paper we proposed the use of a low-rank methodology for the solution to PDE-constrained optimization problems. In particular we introduced a low-rank in time approach that allows us to significantly reduce the storage requirements in time for a one-shot solution of the optimal control problem. We were also able to rewrite the problem in such a way that we can obtain low-rank existence results from classical Sylvester equation theory. We additionally discussed a stationary

iteration as a preconditioner for the Schur-complement approximation within the overall block-diagonal preconditioner. We further illustrated that this technique can be used for many well-known PDEs. Our numerical results illustrated that even with the rather crude Schur-complement approximation a rather robust performance could be obtained. The low-rank method presented enabled computations that are no longer possible to perform with the full-rank approach, which we see as a crucial feature of our methodology.

REFERENCES

[1] R. Andreev, *Space-Time Discretization of the Heat Equation. A Concise MATLAB Implementation*, preprint, arXiv:1212.6037, 2012.

[2] R. Andreev and C. Tobler, *Multilevel Preconditioning and Low Rank Tensor Iteration for Space-Time Simultaneous Discretizations of Parabolic PDES*, Technical report 2012–16, Seminar for Applied Mathematics, ETH Zürich, 2012.

[3] A. Antoulas, *Approximation of Large-Scale Dynamical Systems*, SIAM, Philadelphia, 2005.

[4] A. Antoulas, D. Sorensen, and Y. Zhou, *On the decay rate of Hankel singular values and related issues*, Systems Control Lett., 46 (2002), pp. 323–342.

[5] U. M. Ascher, R. M. Mattheij, and R. D. Russell, *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*, Classics in Appl. Math. 13, SIAM, Philadelphia, 1955.

[6] J. Baglama and L. Reichel, *Augmented implicitly restarted Lanczos bidiagonalization methods*, SIAM J. Sci. Comput, 27 (2005), pp. 19–42.

[7] W. Bangerth, R. Hartmann, and G. Kanschat, *deal.*II*—a general-purpose object-oriented finite element library*, ACM Trans. Math. Software, 33 (2007), pp. Art. 24, 27.

[8] U. Baur and P. Benner, *Factorized solution of Lyapunov equations based on hierarchical matrix arithmetic*, Computing, 78 (2006), pp. 211–234.

[9] P. Benner and T. Breiten, *On Optimality of Interpolation-Based Low-Rank Approximations of Large-Scale Matrix Equations*, MPI Magdeburg Preprint MPIMD/11-10, 2011.

[10] P. Benner and T. Breiten, *Low rank methods for a class of generalized Lyapunov equations and related issues*, Numer. Math., 124 (2013), pp. 441–470.

[11] P. Benner and T. Damm, *Lyapunov equations, energy functionals, and model order reduction of bilinear and stochastic systems*, SIAM J. Control Optim., 49 (2011), pp. 686–711.

[12] P. Benner and P. Kürschner, *Computing Real Low-Rank Solutions of Sylvester Equations by the Factored ADI Method*, MPI Magdeburg Preprint MPIMD/13-05, May 2013.

[13] P. Benner, J.-R. Li, and T. Penzl, *Numerical solution of large Lyapunov equations, Riccati equations, and linear-quadratic control problems*, Numer. Linear Algebra Appl., 15 (2008), pp. 755–777.

[14] P. Benner, R. Li, and N. Truhar, *On the ADI method for Sylvester equations*, J. Comput. Appl. Math., 233 (2009), pp. 1035–1045.

[15] P. Benner and E. Quintana-Ortí, *Solving stable generalized Lyapunov equations with the matrix sign function*, Numer. Algorithms, 20 (1999), pp. 75–100.

[16] M. Benzi, G. H. Golub, and J. Liesen, *Numerical solution of saddle point problems*, Acta Numer., 14 (2005), pp. 1–137.

[17] M. Benzi, E. Haber, and L. Taralli, *A preconditioning technique for a class of PDE-constrained optimization problems*, Adv. Comput. Math., 35 (2011), pp. 149–173.

[18] A. Borzì and V. Schulz, *Multigrid methods for PDE optimization*, SIAM Rev., 51 (2009), pp. 361–395.

[19] J. Boyle, M. D. Mihajlovic, and J. A. Scott, *HSL_MI*20*: An Efficient AMG Preconditioner*, Tech. report RAL-TR-2007-021, Rutherford Appleton Laboratory (CCLRC), 2007.

[20] A. N. Brooks and T. J. Hughes, *Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations*, Comput. Methods Appl. Mech. Engrg., 32 (1982), pp. 199–259.

[21] J. Cahouet and J. Chabard, *Some fast* 3*D finite element solvers for the generalized Stokes problem*, Internat. J. Numer. Methods Fluids, 8 (1988), pp. 869–895.

[22] T. Damm, *Direct methods and ADI-preconditioned Krylov subspace methods for generalized Lyapunov equations*, Numer. Linear Algebra Appl., 15 (2008), pp. 853–871.

[23] T. Davis, *UMFPACK Version* 4.4 *User Guide*, Technical report, Department of Computer and Information Science and Engineering, University of Florida, Gainesville, 2005.

[24] S. Dolgov, B. N. Khoromskij, I. Oseledets, and E. Tyrtyshnikov, *A reciprocal preconditioner for structured matrices arising from elliptic problems with jumping coefficients*, Linear Algebra Appl., 436 (2012), pp. 2980–3007.
[25] V. Druskin, L. Knizhnerman, and V. Simoncini, *Analysis of the rational Krylov subspace and ADI methods for solving the Lyapunov equation*, SIAM J. Numer. Anal., 49 (2011), pp. 1875–1898.
[26] I. S. Duff, A. M. Erisman, and J. K. Reid, *Direct Methods for Sparse Matrices*, Monogr. Numer. Anal., Oxford University Press, New York, 1989.
[27] H. C. Elman, D. J. Silvester, and A. J. Wathen, *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*, Numer. Math. Sci. Comput., Oxford University Press, New York, 2005.
[28] A. Eppler and M. Bollhöfer, *A structure preserving FGMRES method for solving large Lyapunov equations*, in Progress in Industrial Mathematics at ECMI 2010, M. Günther, A. Bartel, M. Brunk, S. Schöps, and M. Striebel, eds., Math. Ind., Springer-Verlag, Berlin, 2012, pp. 131–136.
[29] R. Falgout, *An introduction to algebraic multigrid*, Comput. Sci. Eng., 8 (2006), pp. 24–33.
[30] G. Flagg and S. Gugercin, *On the ADI method for the Sylvester equation and the optimal-$\mathcal{H}_2$ points*, Appl. Numer. Math., 64 (2013), pp. 50–58.
[31] R. Fletcher, *Conjugate gradient methods for indefinite systems*, in Numerical Analysis, Lecture Notes in Math. 506, Springer, Berlin, 1976, pp. 73–89.
[32] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins Stud. Math. Sci., Johns Hopkins University Press, Baltimore, MD, 1996.
[33] G. H. Golub and R. S. Varga, *Chebyshev semi-iterative methods, successive over-relaxation iterative methods, and second order Richardson iterative methods.* I, Numer. Math., 3 (1961), pp. 147–156.
[34] G. H. Golub and R. S. Varga, *Chebyshev semi-iterative methods, successive over-relaxation iterative methods, and second order Richardson iterative methods.* II, Numer. Math., 3 (1961), pp. 157–168.
[35] L. Grasedyck, *Existence and computation of low Kronecker-rank approximations for large linear systems of tensor product structure*, Computing, 72 (2004), pp. 247–265.
[36] L. Grasedyck, *Existence of a low rank or -matrix approximant to the solution of a Sylvester equation*, Numer. Linear Algebra Appl., 11 (2004), pp. 371–389.
[37] L. Grasedyck and W. Hackbusch, *A multigrid method to solve large scale Sylvester equations*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 870–894.
[38] A. Griewank and A. Walther, *Algorithm* 799: *Revolve: An implementation of checkpointing for the reverse or adjoint mode of computational differentiation*, ACM Trans. Math. Software, 26 (2000), pp. 19–45.
[39] E. Haber and U. M. Ascher, *Preconditioned all-at-once methods for large, sparse parameter estimation problems*, Inverse Problems, 17 (2001), pp. 1847–1864.
[40] E. Haber, U. M. Ascher, and D. W. Oldenburg, *Inversion of 3d electromagnetic data in frequency and time domain using an inexact all-at-once approach*, Geophysics, 69 (2004), pp. 1216–1228.
[41] W. Hackbusch, *Multigrid Methods and Applications*, Springer Ser. Comput. Math. 4, Springer-Verlag, Berlin, 1985.
[42] M. Heinkenschloss, *A time-domain decomposition iterative method for the solution of distributed linear quadratic optimal control problems*, J. Comput. Appl. Math., 173 (2005), pp. 169–198.
[43] M. Heinkenschloss and D. Leykekhman, *Local error estimates for SUPG solutions of advection-dominated elliptic linear-quadratic optimal control problems*, SIAM J. Numer. Anal., 47 (2010), pp. 4607–4638.
[44] D. Hinrichsen and A. Pritchard, *Mathematical Systems Theory* I: *Modelling, State Space Analysis, Stability and Robustness*, Vol. 1, Springer-Verlag, Berlin, 2005.
[45] M. Hinze, *A variational discretization concept in control constrained optimization: The linear-quadratic case*, Comput. Optim. Appl., 30 (2005), pp. 45–61.
[46] M. Hinze, M. Köster, and S. Turek, *A Hierarchical Space-Time Solver for Distributed Control of the Stokes Equation*, Technical report SPP1253-16-01, 2008.
[47] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich, *Optimization with PDE Constraints*, Math. Model. Theory Appl., Springer-Verlag, New York, 2009.
[48] M. E. Hochstenbach, *A Jacobi-Davidson type SVD method*, SIAM J. Sci. Comput, 23 (2001), pp. 606–628.
[49] K. Ito and K. Kunisch, *Lagrange Multiplier Approach to Variational Problems and Applications*, Adv. Design Control 15, SIAM, Philadelphia, 2008.

[50] I. Jaimoukha and E. Kasenally, *Krylov subspace methods for solving large Lyapunov equations*, SIAM J. Numer. Anal., 31 (1994), pp. 227–251.
[51] K. Jbilou and A. J. Riquet, *Projection methods for large Lyapunov matrix equations*, Linear Algebra Appl., 415 (2006), pp. 344–358.
[52] H. Knobloch and H. Kwakernaak, *Lineare Kontrolltheorie*, Springer-Verlag, Berlin, 1985.
[53] M. Kollmann and M. Kolmbauer, *A preconditioned MinRes solver for time-periodic parabolic optimal control problems*, Numer. Linear Algebra Appl., 20 (2013), pp. 761–784.
[54] D. Kressner and C. Tobler, *Krylov subspace methods for linear systems with tensor product structure*, SIAM J. Matrix Anal. Appl, 31 (2010), pp. 1688–1714.
[55] D. Kressner and C. Tobler, *Low-rank tensor Krylov subspace methods for parametrized linear systems*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 1288–1316.
[56] J.-R. Li and J. White, *Low rank solution of Lyapunov equations*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 260–280.
[57] T. P. Mathew, M. Sarkis, and C. E. Schaerer, *Analysis of block parareal preconditioners for parabolic optimal control problems*, SIAM J. Sci. Comput., 32 (2010), pp. 1180–1200.
[58] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed., Springer Ser. Oper. Res. Financ. Eng., Springer, New York, 2006.
[59] C. C. Paige and M. A. Saunders, *Solutions of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal, 12 (1975), pp. 617–629.
[60] J. W. Pearson, M. Stoll, and A. Wathen, *Preconditioners for state constrained optimal control problems with Moreau-Yosida penalty function*, Numer. Linear Algebra Appl., 21 (2014), pp. 81–97.
[61] J. W. Pearson, M. Stoll, and A. J. Wathen, *Regularization-robust preconditioners for time-dependent PDE-constrained optimization problems*, SIAM J. Matrix Anal. Appl., 33 (2012), pp. 1126–1152.
[62] J. W. Pearson and A. J. Wathen, *Fast Iterative Solvers for Convection-Diffusion Control Problems*, Electron. Trans. Numer. Anal., 40 (2013), pp. 294–310.
[63] J. W. Pearson and A. J. Wathen, *A new approximation of the Schur complement in preconditioners for PDE-constrained optimization*, Numer. Linear Algebra Appl., 19 (2012), pp. 816–829.
[64] T. Penzl, *A Cyclic Low Rank Smith Method for Large, Sparse Lyapunov Equations with Applications in Model Reduction and Optimal Control*, Technical report SFB393/98-6, Fakultät für Mathematik, TU Chemnitz, Germany, 1998.
[65] T. Penzl, *Eigenvalue decay bounds for solutions of Lyapunov equations: The symmetric case*, Systems Control Lett., 40 (2000), pp. 139–144.
[66] A. Ramage, *A multigrid preconditioner for stabilised discretisations of advection-diffusion problems*, J. Comput. Appl. Math., 110 (1999), pp. 187–203.
[67] T. Rees, *Preconditioning Iterative Methods for PDE Constrained Optimization*, Ph.D. thesis, University of Oxford, UK, 2010.
[68] T. Rees, H. S. Dollar, and A. J. Wathen, *Optimal solvers for PDE-constrained optimization*, SIAM J. Sci. Comput., 32 (2010), pp. 271–298.
[69] J. W. Ruge and K. Stüben, *Algebraic multigrid*, in Multigrid Methods, Frontiers in Appl. Math. 3, SIAM, Philadelphia, 1987, pp. 73–130.
[70] Y. Saad, *Numerical solution of large Lyapunov equation*, in Signal Processing, Scattering, Operator Theory and Numerical Methods, M. A. Kaashoek, J. H. van Schuppen, and A. C. M. Ran, eds., Birkhauser, Basel, 1990, pp. 503–511.
[71] Y. Saad, *A flexible inner-outer preconditioned GMRES algorithm*, SIAM J. Sci. Comput., 14 (1993), pp. 461–461.
[72] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.
[73] J. Schöberl and W. Zulehner, *Symmetric indefinite preconditioners for saddle point problems with applications to PDE-constrained optimization problems*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 752–773.
[74] V. Simoncini, *A new iterative method for solving large-scale Lyapunov matrix equations*, SIAM J. Sci. Comput., 29 (2007), pp. 1268–1288.
[75] V. Simoncini, *The extended Krylov subspace for parameter dependent systems*, Appl. Numer. Math., 60 (2010), pp. 550–560.
[76] V. Simoncini, *Reduced order solution of structured linear systems arising in certain PDE-constrained optimization problems*, Comput. Optim. Appl., to appear.
[77] V. Simoncini, *Computational Methods for Linear Matrix Equations*, Technical report, Università di Bologna, 2013.

[78] D. Sorensen and A. Antoulas, *The Sylvester equation and approximate balanced reduction*, Linear Algebra Appl., 351–352 (2002), pp. 671–700.
[79] M. Stoll, *A Krylov-Schur approach to the truncated SVD*, Linear Algebra Appl., 436 (2012), pp. 2795–2806.
[80] M. Stoll, *All-at-once solution of a time-dependent time-periodic PDE-constrained optimization problems*, IMA J. Numer. Anal., 34 (2014), pp. 1554–1577.
[81] M. Stoll and A. Wathen, *All-at-Once Solution of Time-Dependent PDE-Constrained Optimization Problems*, Technical report, University of Oxford, UK, 2010.
[82] M. Stoll and A. Wathen, *All-at-once solution of time-dependent Stokes control*, J. Comput. Phys., 232 (2013), pp. 498–515.
[83] T. Sun, *Discontinuous Galerkin finite element method with interior penalties for convection diffusion optimal control problem*, Int. J. Numer. Anal. Model, 7 (2010), pp. 87–107.
[84] F. Tröltzsch, *Optimal Control of Partial Differential Equations: Theory, Methods and Applications*, AMS, Providence, RI, 2010.
[85] B. Vandereycken and S. Vandewalle, *A Riemannian optimization approach for computing low-rank solutions of Lyapunov equations*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 2553–2579.
[86] A. J. Wathen and T. Rees, *Chebyshev semi-iteration in preconditioning for problems including the mass matrix*, Electron. Trans. Numer. Anal., 34 (2008), pp. 125–135.
[87] P. Wesseling, *An Introduction to Multigrid Methods*, Pure Appl. Math. (N.Y.), John Wiley, New York, 1992.

EHRENERKLÄRUNG

Ich versichere hiermit, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; verwendete fremde und eigene Quellen sind als solche kenntlich gemacht. Ich habe insbesondere nicht wissentlich:

- Ergebnisse erfunden oder widersprüchliche Ergebnisse verschwiegen,
- statistische Verfahren absichtlich missbraucht, um Daten in ungerechtfertigter Weise zu interpretieren,
- fremde Ergebnisse oder Veröffentlichungen plagiiert oder verzerrt wiedergegeben.

Die Arbeit wurde bisher weder im Inland noch im Ausland in gleicher oder ähnlicher Form als Habilitationsschrift oder ähnliche Prüfungsarbeit eingereicht und ist als Ganzes auch noch nicht veröffentlicht.

Magdeburg, 30.6.2016

Martin Stoll