

PRECONDITIONED CONJUGATE GRADIENT METHOD FOR OPTIMAL CONTROL PROBLEMS WITH CONTROL AND STATE CONSTRAINTS*

ROLAND HERZOG[†] AND EKKEHARD SACHS[‡]

Abstract. Optimality systems and their linearizations arising in optimal control of partial differential equations with pointwise control and (regularized) state constraints are considered. The preconditioned conjugate gradient (PCG) method in a nonstandard inner product is employed for their efficient solution. Preconditioned condition numbers are estimated for problems with pointwise control constraints, mixed control-state constraints, and of Moreau–Yosida penalty type. Numerical results for elliptic problems demonstrate the performance of the PCG iteration. Regularized state-constrained problems in three dimensions with more than 750,000 variables are solved.

Key words. preconditioned conjugate gradient method, saddle point problems, optimal control of partial differential equations, control and state constraints, multigrid method

AMS subject classifications. 49K20, 49M15, 65F08, 65N55

DOI. 10.1137/090779127

1. Introduction. In their seminal paper [4], Bramble and Pasciak introduced the idea of applying the preconditioned conjugate gradient (PCG) iteration to symmetric indefinite saddle point problems

$$(1.1) \quad \begin{pmatrix} A & B^\top \\ B & 0 \end{pmatrix} \begin{pmatrix} x \\ q \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix},$$

where A is a real, symmetric, and positive definite matrix, and B is a real m -by- n matrix with full rank $m \leq n$. The application of the PCG iteration becomes possible through the clever use of a suitable scalar product, which renders the preconditioned saddle point matrix symmetric and positive definite. Meanwhile, this approach has been further developed and has become a widespread technique for solving partial differential equations (PDEs) in mixed formulation. Very often in these applications, A is positive definite on the whole space.

Recently, Schöberl and Zulehner [19] proposed a symmetric indefinite preconditioner with the above properties for the case that A is positive definite only on the nullspace of B :

$$x^\top Ax > 0 \quad \text{for all } x \in \ker B \setminus \{0\}.$$

This situation is natural when (1.1) is viewed as the first-order optimality system of the optimization problem

$$\text{Minimize } 1/2 x^\top Ax - f^\top x \quad \text{subject to (s.t.) } Bx = g.$$

In fact, the authors in [19] show the applicability of their approach to discretized optimal control problems for elliptic PDEs with no further constraints. In this paper,

*Received by the editors December 3, 2009; accepted for publication (in revised form) by V. Simoncini April 22, 2010; published electronically July 1, 2010.

<http://www.siam.org/journals/simax/31-5/77912.html>

[†]Faculty of Mathematics, Chemnitz University of Technology, D-09107 Chemnitz, Germany (roland.herzog@mathematik.tu-chemnitz.de).

[‡]Department of Mathematics, University of Trier, D-54286 Trier, Germany (sachs@uni-trier.de).

we extend their technique to optimal control problems with PDEs in the presence of control and state constraints, which are usually present in practical examples.

Since even linear inequality constraints give rise to nonlinear optimality systems, we apply a Newton-type approach for their solution. Every Newton step is a saddle point problem of type (1.1). Our focus is on the efficient solution of these linear systems. Therefore we address a key component present in practically every algorithm for the solution of constrained optimal control problems. We efficiently solve the underlying linear systems by employing a nonstandard inner product PCG method.

Following [19], the building blocks of the preconditioner are preconditioners for those matrices which represent the scalar products in the spaces X and Q , in which the unknowns x and q are sought. In the context of constrained optimal control problems, x represents the state and control variables, and q comprises the adjoint state and additional Lagrange multipliers. Optimal preconditioners for scalar product matrices are readily available, e.g., using multigrid or domain decomposition approaches. Due to their linear complexity and grid independent spectral qualities, the PCG solver also becomes grid independent and has linear complexity in the number of variables. We use elliptic optimal control problems as examples and proof of concept, but the extension to time dependent problems is straightforward.

The efficient solution of saddle point problems is of paramount importance in solving large scale optimization problems, and it is receiving an increasing amount of attention. Let us put our work into perspective. Various preconditioned Krylov subspace methods for an accelerated solution of optimality systems in PDE-constrained optimization are considered in [1, 2, 3, 12, 14, 15]. None of these employs conjugate gradient iterations to the system (1.1). The class of constraint preconditioners approximates only A in (1.1) and is often applied within a projected preconditioned conjugate gradient (PPCG) method. However, preconditioners of this type frequently require the inversion of B_1 in a partition $B = [B_1, B_2]$; see [6, 8]. In the context of PDE-constrained optimization, this amounts to the inversion of the PDE operator, which should be avoided. Recently, in [17] approximate constraint preconditioners for PPCG iterations were considered in the context of optimal control. It seems, however, that their approach lacks extensions to situations where the control acts only on part of the domain or on its boundary. An approach similar to ours is presented in [18], where a nonstandard inner product conjugate gradient iteration with a block triangular preconditioner is devised and applied to an optimal control problem without inequality constraints. However, positive definiteness of the (1,1) block is required, which fails to hold, for instance, as soon as the observation of the state variable in the objective is reduced to a subdomain. The technique considered here does not have these limitations. Finally, we mention [7], where a number of existing approaches are interpreted in the framework of PCG iterations in nonstandard inner products.

In the present paper, we apply PCG iterations in a nonstandard scalar product to optimal control problems with control and regularized state constraints. Through a comparison of the preconditioned condition numbers, we find the Moreau–Yosida penalty approach preferable to regularization by mixed control-state constraints. This is confirmed by numerical results.

The solution of state-constrained problems in three dimensions is considered computationally challenging, and numerical results can hardly be found in the literature. Using Moreau–Yosida regularization and PCG iterations, the approach presented here pushes the frontier towards larger problems. The largest three-dimensional (3D) problem solved has 250,000 state variables, and the same number of control and adjoint state variables, plus Lagrange multipliers. Hence the overall size of the optimality

system exceeds 750,000.

The plan of the paper is as follows. The remainder of this section contains the standing assumptions. The PCG method is described in section 2. We also briefly recall the main results from [19] there. In section 3, we address its application to the Newton system arising in elliptic optimal control problems. We distinguish problems with pointwise control constraints and regularized state constraints by way of mixed control-state constraints and a Moreau–Yosida-type penalty approach. Extensive numerical examples are provided in section 4. Concluding remarks follow in section 5.

Assumptions. We represent (1.1) in terms of the underlying bilinear forms: Find $x \in X$ and $q \in Q$ satisfying

$$(1.2) \quad \begin{aligned} a(x, w) + b(w, q) &= \langle F, w \rangle \quad \text{for all } w \in X, \\ b(x, r) &= \langle G, r \rangle \quad \text{for all } r \in Q, \end{aligned}$$

where X and Q are real Hilbert spaces. The following conditions are well known to be sufficient to ensure the existence and uniqueness of a solution $(x, q) \in X \times Q$:

1. a is bounded:

$$(1.3a) \quad a(x, w) \leq \|a\| \|x\|_X \|w\|_X \quad \text{for all } x, w \in X.$$

2. a is coercive on $\ker B = \{x \in X : b(x, r) = 0 \text{ for all } r \in Q\}$: There exists $\alpha_0 > 0$ such that

$$(1.3b) \quad a(x, x) \geq \alpha_0 \|x\|_X^2 \quad \text{for all } x \in \ker B.$$

3. b is bounded:

$$(1.3c) \quad b(x, r) \leq \|b\| \|x\|_X \|r\|_Q \quad \text{for all } x \in X, r \in Q.$$

4. b satisfies the inf–sup condition: There exists $k_0 > 0$ such that

$$(1.3d) \quad \sup_{x \in X} b(x, r) \geq k_0 \|x\|_X \|r\|_Q \quad \text{for all } r \in Q.$$

5. a is symmetric and nonnegative, i.e.,

$$(1.3e) \quad a(x, w) = a(w, x) \quad \text{and} \quad a(x, x) \geq 0 \quad \text{for all } x, w \in X.$$

In constrained optimal control applications, x will correspond to the state and control variables, while q comprises the adjoint state and Lagrange multipliers associated with inequality constraints.

Notation. For symmetric matrices, $A \preceq B$ means that $B - A$ is positive semidefinite, and $A \prec B$ means that $B - A$ is positive definite.

2. PCG method. In this section, we recall the main results from [19] and give some algorithmic details concerning the PCG iteration. We consider a Galerkin discretization of (1.2) with finite-dimensional subspaces $X_h \subset X$ and $Q_h \subset Q$. Here and throughout, \mathcal{X} and \mathcal{Q} are the matrices representing the scalar products in the subspaces X_h and Q_h , respectively. Moreover, A and B are the matrices representing the bilinear forms a and b , respectively, w.r.t. a chosen basis on these subspaces. It is

assumed that conditions (1.3a)–(1.3e) are also satisfied on the discrete level; i.e., the following hold (see [19, equations (3.2)–(3.5)]):

$$\begin{aligned}
 (2.1a) \quad & A \preceq \|a\| \mathcal{X}, \\
 (2.1b) \quad & \vec{x}^\top A \vec{x} \geq \alpha_0 \vec{x}^\top \mathcal{X} \vec{x} \quad \text{for all } \vec{x} \in \ker B, \\
 (2.1c) \quad & B\mathcal{X}^{-1}B^\top \preceq \|b\|^2 \mathcal{Q}, \\
 (2.1d) \quad & B\mathcal{X}^{-1}B^\top \succeq k_0^2 \mathcal{Q}, \\
 (2.1e) \quad & A = A^\top, \quad A \succeq 0.
 \end{aligned}$$

Note that the validity of (2.1b) and (2.1d) depends on a proper choice of the discretization.

The following class of symmetric indefinite preconditioners for (1.1) is analyzed in [19]:

$$(2.2) \quad \widehat{\mathcal{K}} = \begin{pmatrix} \widehat{A} & B^\top \\ B & B\widehat{A}^{-1}B^\top - \widehat{S} \end{pmatrix} = \begin{pmatrix} I & 0 \\ B\widehat{A}^{-1} & I \end{pmatrix} \begin{pmatrix} \widehat{A} & B^\top \\ 0 & -\widehat{S} \end{pmatrix},$$

where \widehat{A} and \widehat{S} are symmetric and nonsingular matrices that we define below. The application of $\widehat{\mathcal{K}}^{-1}$ amounts to the solution of two linear systems with \widehat{A} and one with \widehat{S} :

$$\widehat{\mathcal{K}} \begin{pmatrix} \vec{r}_x \\ \vec{r}_q \end{pmatrix} = \begin{pmatrix} \vec{s}_x \\ \vec{s}_q \end{pmatrix} \Leftrightarrow \widehat{A}\vec{r}_x' = \vec{s}_x, \quad \widehat{S}\vec{r}_q = B\vec{r}_x' - \vec{s}_q, \quad \widehat{A}\vec{r}_x = \vec{s}_x - B^\top \vec{r}_q.$$

It stands out as a feature of this approach that one can simply use properly scaled *preconditioners for the scalar product matrices* as building blocks for the preconditioner $\widehat{\mathcal{K}}$:

$$\widehat{A} = \frac{1}{\sigma} \widehat{\mathcal{X}}, \quad \widehat{S} = \frac{\sigma}{\tau} \widehat{\mathcal{Q}},$$

where $\widehat{\mathcal{X}}$ and $\widehat{\mathcal{Q}}$ are suitable approximations to \mathcal{X} and \mathcal{Q} , respectively.

Remark 2.1. Preconditioners for scalar product matrices are usually easy to obtain. For instance, if $X = H^1(\Omega)$ is discretized by the finite element method, then \mathcal{X} is given by the finite element stiffness matrix associated with the weak form of the problem

$$-\Delta y + y = 0 \quad \text{in } \Omega, \quad \partial_n y = 0 \quad \text{on } \partial\Omega.$$

In other words, $\mathcal{X}_{ij} = \int_\Omega (\nabla\varphi_j \cdot \nabla\varphi_i + \varphi_j \varphi_i) dx$, where $\{\varphi_i\}$ are the basis functions of the discrete subspace $X_h \subset X$. The standard multigrid cycle yields a suitable preconditioner with linear complexity. In case $X = L^2(\Omega)$, a symmetric Gauss–Seidel iteration for the mass matrix $\mathcal{X}_{ij} = \int_\Omega \varphi_j \varphi_i dx$ can be applied instead.

We emphasize that the preconditioners \widehat{A} and \widehat{S} depend only on the scalar products in the spaces X_h (which comprises the state and control space for optimal control applications) and Q_h (which includes the adjoint state and additional Lagrange multipliers). In particular, the preconditioners *do not depend on the PDE* appearing as a constraint in optimal control problems. For example, whether or not the state equation contains convection terms or discontinuous coefficients, etc., the same preconditioner can be used, albeit possibly with different scaling parameters σ and τ .

In view of the availability of good preconditioners for \mathcal{X} and \mathcal{Q} , it can be assumed that the spectral estimates

$$(1 - q_X) \widehat{\mathcal{X}} \preceq \mathcal{X} \preceq \widehat{\mathcal{X}}, \quad (1 - q_Q) \widehat{\mathcal{Q}} \preceq \mathcal{Q} \preceq \widehat{\mathcal{Q}}$$

are valid, with mesh independent constants q_X and q_Q close to zero. We recall the following result from [19, Theorem 2.2, Lemma 3.1].

THEOREM 2.2. *Suppose that (2.1a)–(2.1e) hold. Let σ and τ be chosen such that*

$$(2.3) \quad \sigma < \frac{1}{\|a\|} \quad \text{and} \quad \tau > \frac{1}{(1 - q_X)(1 - q_Q)} \frac{1}{k_0^2}$$

are satisfied. Then the estimates

$$\bar{x}^\top A \bar{x} \geq \alpha \bar{x}^\top \widehat{A} \bar{x} \quad \text{for all } \bar{x} \in \ker B \quad \text{and} \quad \widehat{A} \succ A$$

as well as

$$\widehat{S} \prec B \widehat{A}^{-1} B^\top \preceq \beta \widehat{S}$$

hold with constants

$$\alpha = \sigma (1 - q_X) \alpha_0 \leq 1 \quad \text{and} \quad \beta = \tau \|b\|^2 \geq 1.$$

COROLLARY 2.3. *Under the assumptions of the previous theorem,*

$$(2.4) \quad \langle (\vec{t}_x, \vec{t}_q), (\vec{r}_x, \vec{r}_q) \rangle_{\mathcal{D}} = (\vec{t}_x, \vec{t}_q) \begin{pmatrix} \widehat{A} - A & 0 \\ 0 & B \widehat{A}^{-1} B^\top - \widehat{S} \end{pmatrix} \begin{pmatrix} \vec{r}_x \\ \vec{r}_q \end{pmatrix}$$

defines a scalar product. Moreover, the preconditioned matrix $\widehat{\mathcal{K}}^{-1} \mathcal{K}$ is symmetric and positive definite w.r.t. $\langle \cdot, \cdot \rangle_{\mathcal{D}}$. Its condition number can be estimated by

$$(2.5) \quad \kappa(\widehat{\mathcal{K}}^{-1} \mathcal{K}) \leq \frac{(1 + \sqrt{5})^2}{2} \frac{\beta}{\alpha}.$$

The estimates above indicate that σ should be chosen as large as possible and τ as small as possible, while still satisfying condition (2.3). A simple calculation shows that

$$(2.6) \quad \kappa(\widehat{\mathcal{K}}^{-1} \mathcal{K}) \sim \frac{\|a\|}{\alpha_0} \left(\frac{\|b\|}{k_0} \right)^2$$

can be expected for the preconditioned condition number. This result implies that the effectiveness of the preconditioner will depend more on the properties of the bilinear form b and less on the properties of a .

Finding good choices for σ and τ requires some a priori knowledge of the norm $\|a\|$ and the inf-sup constant k_0 , which may be available in some situations (see the examples in section 3). Estimates of $\|a\|$ and k_0 can be found by computing the extreme eigenvalues of a generalized eigenvalue problem; see section 4.1 for details. In any case, too large a choice of σ or too small a choice of τ will reveal itself by negative values of the residual norm δ^+ during the PCG iteration (Algorithm 1). We thus employ a simple safeguard strategy (Algorithm 2) to correct unsuitable scaling

parameters.

For the convenience of the reader, the PCG method is given in Algorithm 1. It is to be noted that the scalar product $\langle \cdot, \cdot \rangle_{\mathcal{D}}$ cannot be evaluated for arbitrary pairs of vectors. The reason is that matrix-vector products with \widehat{A} and \widehat{S} are usually not available (in contrast to products with \widehat{A}^{-1} and \widehat{S}^{-1} , which are realized by applications of the preconditioners). And thus $\widehat{A}\vec{r}_x$ and $\widehat{S}\vec{r}_q$ cannot be evaluated unless $(\vec{r}_x, \vec{r}_q) = \widehat{\mathcal{K}}^{-1}(\vec{s}_x, \vec{s}_q)$ holds. That is, the evaluation of the scalar product is possible if one of the factors is known to be the preconditioner applied to another pair of vectors. We denote this situation by $\langle (\vec{t}_x, \vec{t}_q), (\vec{r}_x, \vec{r}_q); (\vec{s}_x, \vec{s}_q) \rangle_{\mathcal{D}}$. In this case, the scalar product can be evaluated as follows:

$$\langle (\vec{t}_x, \vec{t}_q), (\vec{r}_x, \vec{r}_q); (\vec{s}_x, \vec{s}_q) \rangle_{\mathcal{D}} = (\vec{t}_x)^\top (\vec{s}_x - B^\top \vec{r}_q - A \vec{r}_x) + (\vec{t}_q)^\top (\vec{s}_q - B \vec{r}_x).$$

As a consequence, it is necessary to maintain the relations $\vec{r} = \widehat{\mathcal{K}}^{-1}\vec{s}$ and $\vec{q} = \widehat{\mathcal{K}}^{-1}\vec{e}$ throughout the iteration, which requires the storage of one extra vector compared to common conjugate gradient implementations w.r.t. the standard inner product.

3. Application to constrained optimal control problems. In this section, we elaborate on the preconditioned numerical solution of optimal control problems with various types of inequality constraints. Although our examples are of linear quadratic type, the inequality constraints render the corresponding optimality systems nonlinear. We apply a generalized (semismooth) Newton scheme for their solution. The resulting linear system in every Newton step is a saddle point problem of the form (1.1).

We consider the following representative examples:

- a linear quadratic elliptic optimal control problem with pointwise control constraints (section 3.1),
- a similar problem with state constraints, regularized by mixed control-state constraints (section 3.2),
- and an alternative regularization by the Moreau–Yosida (penalty) approach (section 3.3).

These regularizations were introduced in [16] and [13], respectively, and they are well-established techniques to approximate optimal control problems with pointwise state constraints. The need to regularize problems with state constraints arises due to the low function space regularity of the associated Lagrange multipliers, which also has an impact on the linear algebra. We make some further comments on this issue in section 3.4.

Problems with more general nonlinear objective functions, state equations, or inequality constraints can be considered as well. In this case, their first- and second-order derivatives enter the blocks A and B . However, one has to be aware of the fact that A may not need to be positive semidefinite, and the coercivity of A on the nullspace of B may hold only in the vicinity of a local minimum (which is typically guaranteed by second-order sufficient optimality conditions). In early Newton steps, it may thus be necessary to restore the positive semidefiniteness of A and its coercivity on the nullspace of B , e.g., by manipulating the (1,1) block, in order for the Newton system to satisfy conditions (1.3b) and (1.3e).

In all of our examples, Ω denotes a bounded domain in \mathbb{R}^2 or \mathbb{R}^3 with Lipschitz boundary Γ , respectively.

ALGORITHM 1. Conjugate gradient method for $\widehat{\mathcal{K}}^{-1}\mathcal{K}$ w.r.t. to scalar product \mathcal{D} .

Input: right-hand side (\vec{b}_x, \vec{b}_q) and initial iterate (\vec{x}_x, \vec{x}_q)

Output: solution (\vec{x}_x, \vec{x}_q) of $\mathcal{K}(\vec{x}_x, \vec{x}_q) = (\vec{b}_x, \vec{b}_q)$

1. Set $n := 0$ and compute initial residual

$$\begin{pmatrix} \vec{s}_x \\ \vec{s}_q \end{pmatrix} := \begin{pmatrix} \vec{b}_x - A\vec{x}_x - B^\top\vec{x}_q \\ \vec{b}_q - B\vec{x}_x \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \vec{d}_x \\ \vec{d}_q \end{pmatrix} := \begin{pmatrix} \vec{r}_x \\ \vec{r}_q \end{pmatrix} := \widehat{\mathcal{K}}^{-1} \begin{pmatrix} \vec{s}_x \\ \vec{s}_q \end{pmatrix}$$

2. Set $\delta_0 := \delta^+ := \langle (\vec{r}_x, \vec{r}_q), (\vec{r}_x, \vec{r}_q); (\vec{s}_x, \vec{s}_q) \rangle_{\mathcal{D}}$
3. **while** $n < n_{max}$ and $\delta^+ > \varepsilon_{rel}^2 \delta_0$ and $\delta^+ > \varepsilon_{abs}^2$ **do**
4. Set

$$\begin{pmatrix} \vec{e}_x \\ \vec{e}_q \end{pmatrix} := \begin{pmatrix} A\vec{d}_x + B^\top\vec{d}_q \\ B\vec{d}_x \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \vec{q}_x \\ \vec{q}_q \end{pmatrix} := \widehat{\mathcal{K}}^{-1} \begin{pmatrix} \vec{e}_x \\ \vec{e}_q \end{pmatrix}$$

5. Set $\alpha := \delta^+ / \langle (\vec{d}_x, \vec{d}_q), (\vec{q}_x, \vec{q}_q); (\vec{e}_x, \vec{e}_q) \rangle_{\mathcal{D}}$
6. Update the solution

$$\begin{pmatrix} \vec{x}_x \\ \vec{x}_q \end{pmatrix} := \begin{pmatrix} \vec{x}_x \\ \vec{x}_q \end{pmatrix} + \alpha \begin{pmatrix} \vec{d}_x \\ \vec{d}_q \end{pmatrix}$$

7. Update the residual

$$\begin{pmatrix} \vec{r}_x \\ \vec{r}_q \end{pmatrix} := \begin{pmatrix} \vec{r}_x \\ \vec{r}_q \end{pmatrix} - \alpha \begin{pmatrix} \vec{q}_x \\ \vec{q}_q \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \vec{s}_x \\ \vec{s}_q \end{pmatrix} := \begin{pmatrix} \vec{s}_x \\ \vec{s}_q \end{pmatrix} - \alpha \begin{pmatrix} \vec{e}_x \\ \vec{e}_q \end{pmatrix}$$

8. Set $\delta := \delta^+$ and $\delta^+ := \langle (\vec{r}_x, \vec{r}_q), (\vec{r}_x, \vec{r}_q); (\vec{s}_x, \vec{s}_q) \rangle_{\mathcal{D}}$
9. Set $\beta := \delta^+ / \delta$
10. Update the search direction

$$\begin{pmatrix} \vec{d}_x \\ \vec{d}_q \end{pmatrix} := \begin{pmatrix} \vec{r}_x \\ \vec{r}_q \end{pmatrix} + \beta \begin{pmatrix} \vec{d}_x \\ \vec{d}_q \end{pmatrix}$$

11. Set $n := n + 1$
12. **end while**
13. **return** (\vec{x}_x, \vec{x}_q)

ALGORITHM 2. Safeguard strategy for selection of scaling parameters σ, τ .

1. Determine an underestimate $\|a\|' \leq \|a\|$ and an overestimate $k'_0 \geq k_0$
2. Set $\sigma = 0.9/\|a\|'$ and $\tau = 1.2/(k'_0)^2$ and **scaling_rejected** := **true**
3. **while** **scaling_rejected** **do**
4. Run the PCG iteration (Algorithm 1)
5. **if** it fails with $\delta^+ < 0$ **then**
6. Set $\sigma := \sigma/\sqrt{2}$ and $\tau := \sqrt{2}\tau$
7. **else**
8. Set **scaling_rejected** := **false**
9. **end if**
10. **end while**

3.1. Control constraints. We consider the following optimal control problem with pointwise control constraints:

$$(CC) \quad \begin{aligned} & \text{Minimize} && \frac{1}{2} \|y - y_d\|_{L^2(\Omega)}^2 + \frac{\nu}{2} \|u\|_{L^2(\Omega)}^2 \\ & \text{s.t.} && \begin{cases} -\Delta y + y = u & \text{in } \Omega, \\ \partial_n y = 0 & \text{on } \Gamma \end{cases} \\ & \text{and} && u_a \leq u \leq u_b \quad \text{a.e. in } \Omega. \end{aligned}$$

Here, $\nu > 0$ denotes the control cost parameter, ∂_n is the normal derivative on the boundary, and u_a and u_b are the lower and upper bounds for the control variable u . The first-order system of necessary and sufficient optimality conditions of (CC) can be expressed as follows (compare [21, section 2.8.2]):

$$\begin{aligned} (3.1a) \quad & -\Delta p + p = -(y - y_d) \quad \text{in } \Omega, \quad \partial_n p = 0 \quad \text{on } \Gamma, \\ (3.1b) \quad & \nu u - p + \xi = 0 \quad \text{a.e. in } \Omega, \\ (3.1c) \quad & -\Delta y + y = u \quad \text{in } \Omega, \quad \partial_n y = 0 \quad \text{on } \Gamma, \\ (3.1d) \quad & \xi - \max\{0, \xi + c(u - u_b)\} - \min\{0, \xi - c(u_a - u)\} = 0 \quad \text{a.e. in } \Omega \end{aligned}$$

for any $c > 0$. In (3.1), p denotes the adjoint state, and ξ is the Lagrange multiplier associated with the control constraints. Note that (3.1d) is equivalent to the two pointwise complementarity systems

$$\begin{aligned} 0 &\leq \xi^+, \quad u - u_b \leq 0, \quad \xi^+(u - u_b) = 0, \\ 0 &\leq \xi^-, \quad u_a - u \leq 0, \quad \xi^-(u_a - u) = 0. \end{aligned}$$

It is well known that (3.1) enjoys the Newton differentiability property [11], at least for $c = \nu$. Hence, a generalized (semismooth) Newton iteration can be applied. We focus on the preconditioned solution of the generalized Newton steps. Due to the structure of the nonsmooth part (3.1d), the Newton iteration can be expressed in terms of an active set strategy. Given an iterate (y_k, u_k, p_k, ξ_k) , the active sets are determined by

$$(3.2) \quad \begin{aligned} \mathcal{A}_k^+ &= \{x \in \Omega : \xi_k + c(u_k - u_b) > 0\}, \\ \mathcal{A}_k^- &= \{x \in \Omega : \xi_k - c(u_a - u_k) < 0\}, \end{aligned}$$

and the inactive set is $\mathcal{I}_k = \Omega \setminus (\mathcal{A}_k^+ \cup \mathcal{A}_k^-)$. The Newton step for the solution of (3.1), given in terms of the new iterate, reads as follows:

$$(3.3) \quad \begin{pmatrix} I & \cdot & L^* & \cdot \\ \cdot & \nu I & -I & I \\ L & -I & \cdot & \cdot \\ \cdot & c\chi_{\mathcal{A}_k} & \cdot & \chi_{\mathcal{I}_k} \end{pmatrix} \begin{pmatrix} y_{k+1} \\ u_{k+1} \\ p_{k+1} \\ \xi_{k+1} \end{pmatrix} = \begin{pmatrix} y_d \\ 0 \\ 0 \\ c(\chi_{\mathcal{A}_k^+} u_b + \chi_{\mathcal{A}_k^-} u_a) \end{pmatrix},$$

where $\chi_{\mathcal{A}_k^+}$, $\chi_{\mathcal{A}_k^-}$, and $\chi_{\mathcal{A}_k}$ denote the characteristic functions of \mathcal{A}_k^+ , \mathcal{A}_k^- , and $\mathcal{A}_k = \mathcal{A}_k^+ \cup \mathcal{A}_k^-$, respectively, and L represents the differential operator of the PDE constraint in (CC). In the present example, we have $L = -\Delta + I$ with homogeneous Neumann boundary conditions in weak form, considered as an operator from $H^1(\Omega)$ into $H^1(\Omega)^*$. We emphasize that the Newton system (3.3) changes from iteration to iteration due

to changes in the active sets. Since, however, we focus here on the efficient solution of individual Newton steps, we drop the iteration index from now on.

From (3.3) one infers $\xi_{\mathcal{I}} = 0$ (the restriction of ξ to the inactive set \mathcal{I}), and we eliminate this variable from the problem. The Newton system then attains an equivalent *symmetric* saddle point form:

$$(3.4) \quad \begin{pmatrix} I & \cdot & L^* & \cdot \\ \cdot & \nu I & -I & \chi_{\mathcal{A}} \\ L & -I & \cdot & \cdot \\ \cdot & \chi_{\mathcal{A}} & \cdot & \cdot \end{pmatrix} \begin{pmatrix} y \\ u \\ p \\ \xi_{\mathcal{A}} \end{pmatrix} = \begin{pmatrix} y_d \\ 0 \\ 0 \\ \chi_{\mathcal{A}^+} u_b + \chi_{\mathcal{A}^-} u_a \end{pmatrix},$$

which fits into our framework (1.2) with the identifications

$$\begin{aligned} x &= (y, u) \in X = H^1(\Omega) \times L^2(\Omega), \\ q &= (p, \xi_{\mathcal{A}}) \in Q = H^1(\Omega) \times L^2(\mathcal{A}) \end{aligned}$$

and bilinear forms

$$(3.5a) \quad a((y, u), (z, v)) := (y, z)_{L^2(\Omega)} + \nu (u, v)_{L^2(\Omega)},$$

$$(3.5b) \quad b((y, u), (p, \xi_{\mathcal{A}})) := (y, p)_{H^1(\Omega)} - (u, p)_{L^2(\Omega)} + (u, \xi_{\mathcal{A}})_{L^2(\mathcal{A})}.$$

LEMMA 3.1. *The bilinear forms $a(\cdot, \cdot)$ and $b(\cdot, \cdot)$ satisfy assumptions (1.3a)–(1.3e) with constants $\|a\| = \max\{1, \nu\}$, $\alpha_0 = \nu/2$, $\|b\| = 2$, and $k_0 = 1/2$, independent of the active set \mathcal{A} .*

Proof. The proof uses the Cauchy–Schwarz and Young inequalities. The boundedness of a follows from

$$\begin{aligned} a((y, u), (z, v)) &\leq \|y\|_{L^2(\Omega)} \|z\|_{L^2(\Omega)} + \nu \|u\|_{L^2(\Omega)} \|v\|_{L^2(\Omega)} \\ &\leq (\|y\|_{L^2(\Omega)}^2 + \nu \|u\|_{L^2(\Omega)}^2)^{1/2} (\|z\|_{L^2(\Omega)}^2 + \nu \|v\|_{L^2(\Omega)}^2)^{1/2} \\ &\leq \max\{1, \nu\} \|(y, u)\|_X \|(z, v)\|_X. \end{aligned}$$

To verify the coercivity of a , let $(y, u) \in \ker B$. Then, in particular, $(y, p)_{H^1(\Omega)} = (u, p)_{L^2(\Omega)}$ holds for all $p \in H^1(\Omega)$, and from $p = y$ we obtain the a priori estimate $\|y\|_{H^1(\Omega)} \leq \|u\|_{L^2(\Omega)}$. This implies

$$\begin{aligned} a((y, u), (y, u)) &= \|y\|_{L^2(\Omega)}^2 + \nu \|u\|_{L^2(\Omega)}^2 \\ &\geq \frac{\nu}{2} \|y\|_{H^1(\Omega)}^2 + \frac{\nu}{2} \|u\|_{L^2(\Omega)}^2 = \frac{\nu}{2} \|(y, u)\|_X^2. \end{aligned}$$

The boundedness of b follows from

$$\begin{aligned} b((y, u), (p, \xi_{\mathcal{A}})) &\leq \|y\|_{H^1(\Omega)} \|p\|_{H^1(\Omega)} + \|u\|_{L^2(\Omega)} \|p\|_{L^2(\Omega)} + \|u\|_{L^2(\mathcal{A})} \|\xi\|_{L^2(\mathcal{A})} \\ &\leq (\|y\|_{H^1(\Omega)} + \|u\|_{L^2(\Omega)}) (\|p\|_{H^1(\Omega)} + \|\xi\|_{L^2(\mathcal{A})}) \\ &\leq 2 \|(y, u)\|_X \|(p, \xi_{\mathcal{A}})\|_Q. \end{aligned}$$

Finally, we obtain the inf–sup condition for b as follows: For given $(p, \xi_{\mathcal{A}}) \in Q$, choose

$y = p$ and $u = \xi_{\mathcal{A}}$ (by extending it by zero outside of \mathcal{A}). Then

$$\begin{aligned} b((y, u), (p, \xi_{\mathcal{A}})) &\geq \|p\|_{H^1(\Omega)}^2 - \|\xi_{\mathcal{A}}\|_{L^2(\Omega)}\|p\|_{L^2(\Omega)} + \|\xi_{\mathcal{A}}\|_{L^2(\mathcal{A})}^2 \\ &\geq \|p\|_{H^1(\Omega)}^2 - \frac{1}{2}\|\xi_{\mathcal{A}}\|_{L^2(\Omega)}^2 - \frac{1}{2}\|p\|_{L^2(\Omega)}^2 + \|\xi_{\mathcal{A}}\|_{L^2(\mathcal{A})}^2 \\ &\geq \frac{1}{2}\|p\|_{H^1(\Omega)}^2 + \frac{1}{2}\|\xi_{\mathcal{A}}\|_{L^2(\mathcal{A})}^2 \\ &= \frac{1}{2}(\|y\|_{H^1(\Omega)}^2 + \|u\|_{L^2(\Omega)}^2)^{1/2}(\|p\|_{H^1(\Omega)}^2 + \|\xi_{\mathcal{A}}\|_{L^2(\Omega)}^2)^{1/2} \\ &= \frac{1}{2}\|(y, u)\|_X\|(p, \xi_{\mathcal{A}})\|_Q. \quad \square \end{aligned}$$

The leading term in the estimate for the preconditioned condition number (see (2.5) and (2.6)) is thus

$$\kappa_{\text{CC}}(\widehat{\mathcal{K}}^{-1}\mathcal{K}) \sim \frac{\beta}{\alpha} = 32\nu^{-1} \max\{1, \nu\}.$$

Remark 3.2. The treatment of an additional term $\gamma \|u\|_{L^1(\Omega)}$ in the objective of (CC) is easily possible. For positive γ , this term promotes so-called sparse optimal controls, which are zero on nontrivial parts of the domain. The corresponding optimality system can be found in [20, Lemma 2.2]. The Newton iteration for the solution of this extended problem requires two changes: First, the active sets are determined from

$$\begin{aligned} \mathcal{A}^+ &= \{x \in \Omega : \xi - \gamma + c(u - u_b) > 0\}, \\ \mathcal{A}^- &= \{x \in \Omega : \xi + \gamma - c(u_a - u) < 0\}, \\ \mathcal{A}^0 &= \{x \in \Omega : -\xi - \gamma \leq cu \leq -\xi + \gamma\}, \end{aligned}$$

and $\mathcal{A} := \mathcal{A}^+ \cup \mathcal{A}^- \cup \mathcal{A}^0$. Note that \mathcal{A}^0 is the set where the updated control is zero. Second, at the end of each Newton step, $\xi_{\mathcal{I}}$ is updated according to $\xi_{\mathcal{I}} = \chi_{\mathcal{I}^+} \gamma - \chi_{\mathcal{I}^-} \gamma$, where \mathcal{I}^+ is the subset of Ω where $cu + \xi - \gamma$ is between 0 and u_b , and similarly for \mathcal{I}^- .

In particular, the Newton system (3.4) remains the same, and thus problems with an additional sparsity term can be solved just as efficiently as problems with control constraints alone.

Discretization. We now turn to the discretization of (CC) and the Newton step (3.4) by a Galerkin method. We introduce

$$\begin{aligned} (3.6a) \quad M_h &= (\varphi_i, \varphi_j)_{L^2(\Omega)} && \text{(mass matrix),} \\ (3.6b) \quad L_h &= K_h = (\nabla\varphi_i, \nabla\varphi_j)_{L^2(\Omega)} + (\varphi_i, \varphi_j)_{L^2(\Omega)} && \text{(stiffness matrix),} \end{aligned}$$

where $\{\varphi_i\}_{i=1}^n$ is a basis of a discrete subspace of $H^1(\Omega)$. The coordinate vector of y w.r.t. this basis is denoted by \vec{y} . Here and throughout, L_h corresponds to the differential operator, and K_h represents the scalar product in the state space $H^1(\Omega)$. For simplicity, we discretize all variables by piecewise linear Lagrangian finite elements. A straightforward modification would allow a different discretization of the control space, e.g., by piecewise constants.

The discrete counterpart of (CC) is

$$\begin{aligned}
 \text{(CC}_h\text{)} \quad & \text{Minimize} \quad \frac{1}{2}(\vec{y} - \vec{y}_d)^\top M_h(\vec{y} - \vec{y}_d) + \frac{\nu}{2}\vec{u}^\top M_h\vec{u} \\
 & \text{s.t.} \quad L_h\vec{y} - M_h\vec{u} = \vec{0} \\
 & \text{and} \quad \vec{u}_a \leq \vec{u} \leq \vec{u}_b.
 \end{aligned}$$

Its optimality conditions are a discrete variant of (3.1):

$$\begin{aligned}
 \text{(3.7a)} \quad & L_h^\top \vec{p} = -M_h(\vec{y} - \vec{y}_d), \\
 \text{(3.7b)} \quad & \nu M_h\vec{u} - M_h\vec{p} + \vec{\mu} = 0, \\
 \text{(3.7c)} \quad & L_h\vec{y} = M_h\vec{u}, \\
 \text{(3.7d)} \quad & \vec{\mu} - \max\{0, \vec{\mu} + c(\vec{u} - \vec{u}_b)\} - \min\{0, \vec{\mu} - c(\vec{u}_a - \vec{u})\} = 0,
 \end{aligned}$$

and a Newton step applied to (3.7) leads to the following discrete linear system:

$$\begin{pmatrix} M_h & \cdot & L_h^\top & \cdot \\ \cdot & \nu M_h & -M_h & I \\ L_h & -M_h & \cdot & \cdot \\ \cdot & c\chi_{\mathcal{A}} & \cdot & \chi_{\mathcal{I}} \end{pmatrix} \begin{pmatrix} \vec{y} \\ \vec{u} \\ \vec{p} \\ \vec{\mu} \end{pmatrix} = \begin{pmatrix} M_h\vec{y}_d \\ 0 \\ 0 \\ c(\chi_{\mathcal{A}^+}\vec{u}_b + \chi_{\mathcal{A}^-}\vec{u}_a) \end{pmatrix}.$$

On the discrete level, $\chi_{\mathcal{A}}$ is a diagonal 0-1 matrix. As in the continuous setting, we infer $\vec{\mu}_{\mathcal{I}} = \vec{0}$ and eliminate this variable to obtain

$$\text{(3.8)} \quad \begin{pmatrix} M_h & \cdot & L_h^\top & \cdot \\ \cdot & \nu M_h & -M_h & P_{\mathcal{A}}^\top \\ L_h & -M_h & \cdot & \cdot \\ \cdot & P_{\mathcal{A}} & \cdot & \cdot \end{pmatrix} \begin{pmatrix} \vec{y} \\ \vec{u} \\ \vec{p} \\ \vec{\mu}_{\mathcal{A}} \end{pmatrix} = \begin{pmatrix} M_h\vec{y}_d \\ 0 \\ 0 \\ P_{\mathcal{A}^+}\vec{u}_b + P_{\mathcal{A}^-}\vec{u}_a \end{pmatrix}.$$

$P_{\mathcal{A}}$ is a rectangular matrix consisting of those rows of $\chi_{\mathcal{A}}$ which belong to the active indices, and similarly for $P_{\mathcal{A}^\pm}$.

Some comments concerning the discrete system (3.8) are in order. The variable $\vec{\mu}$ is the Lagrange multiplier associated with the discrete constraint $\vec{u}_a \leq \vec{u} \leq \vec{u}_b$, and the relations $\vec{\mu} = P_{\mathcal{A}}^\top \vec{\mu}_{\mathcal{A}}$ and $\vec{\mu}_{\mathcal{A}} = P_{\mathcal{A}}\vec{\mu}$ hold. If we set $\vec{\xi} = M_h^{-1}\vec{\mu}$, then $\vec{\xi}$ is the coordinate vector of a function in $L^2(\Omega)$ which approximates the multiplier ξ in the continuous system (3.1d). This observation is reflected by the choice of norms on the discrete level; see (3.9).

With the settings

$$\begin{aligned}
 \vec{x} &= (\vec{y}, \vec{u}) \in X_h = \mathbb{R}^n \times \mathbb{R}^n, \\
 \vec{q} &= (\vec{p}, \vec{\mu}_{\mathcal{A}}) \in Q_h = \mathbb{R}^n \times \mathbb{R}^{n_{\mathcal{A}}}
 \end{aligned}$$

and bilinear forms

$$\begin{aligned}
 a_h((\vec{y}, \vec{u}), (\vec{z}, \vec{v})) &:= \vec{z}^\top M_h\vec{y} + \nu \vec{v}^\top M_h\vec{u}, \\
 b_h((\vec{y}, \vec{u}), (\vec{p}, \vec{\mu}_{\mathcal{A}})) &:= \vec{p}^\top L_h\vec{y} - \vec{p}^\top M_h\vec{u} + \vec{\mu}_{\mathcal{A}}^\top (P_{\mathcal{A}}\vec{u}),
 \end{aligned}$$

the discrete problem fits into the framework (1.2). As mentioned above, care has to be taken in choosing an appropriate norm for the discrete multiplier $\vec{\mu}_{\mathcal{A}}$. We use scalar products in the spaces X_h and Q_h represented by the following matrices:

$$\text{(3.9)} \quad \mathcal{X} = \begin{pmatrix} K_h & \\ & M_h \end{pmatrix} \quad \text{and} \quad \mathcal{Q} = \begin{pmatrix} K_h & \\ & P_{\mathcal{A}}^\top M_h^{-1} P_{\mathcal{A}} \end{pmatrix}$$

with M_h and K_h defined in (3.6).

Due to the use of a conforming Galerkin approach, the constants $\|a\|$ and $\|b\|$ on the discrete level are the same as in Lemma 3.1, and, in particular, they are independent of the mesh size h . The same can be shown for α_0 and k_0 .

Note that the equation $P_{\mathcal{A}}M_h^{-1}P_{\mathcal{A}}^{\top}\vec{\mu}_{\mathcal{A}} = \vec{b}_{\mathcal{A}}$ is equivalent to the linear system

$$(3.10) \quad \begin{pmatrix} M_h & P_{\mathcal{A}}^{\top} \\ P_{\mathcal{A}} & 0 \end{pmatrix} \begin{pmatrix} \vec{r} \\ \vec{\mu}_{\mathcal{A}} \end{pmatrix} = - \begin{pmatrix} \vec{0} \\ \vec{b}_{\mathcal{A}} \end{pmatrix}.$$

This fact is exploited in the construction of the preconditioner below.

Preconditioner. We recall that the preconditioner has the form

$$\widehat{\mathcal{K}} = \begin{pmatrix} I & 0 \\ B\widehat{A}^{-1} & I \end{pmatrix} \begin{pmatrix} \widehat{A} & B^{\top} \\ 0 & -\widehat{S} \end{pmatrix}$$

and that the framework of [19] requires only correctly scaled preconditioners for the scalar product matrices defined in (3.9). We use

$$(3.11) \quad \widehat{A} = \frac{1}{\sigma} \begin{pmatrix} \widehat{K}_h & \\ & \widehat{M}_h \end{pmatrix} \quad \text{and} \quad \widehat{S} = \frac{\sigma}{\tau} \begin{pmatrix} \widehat{K}_h & \\ & P_{\mathcal{A}}M_h^{-1}P_{\mathcal{A}}^{\top} \end{pmatrix}$$

with scaling parameters

$$\sigma = 0.9/\|a\|, \quad \tau = 1.2/k_0^2,$$

similarly as in [19]. For the present example, valid constants $\|a\|$ and k_0 are known from Lemma 3.1. For more complicated examples, an estimation strategy for these constants on the basis of generalized eigenvalue problems is described in section 4.1. Algorithm 3 describes in detail the application of the preconditioner (3.11) in terms of

$$(3.12) \quad \widehat{\mathcal{K}} \begin{pmatrix} \vec{r}_x \\ \vec{r}_q \end{pmatrix} = \begin{pmatrix} \vec{s}_x \\ \vec{s}_q \end{pmatrix},$$

where $\vec{r}_x = (\vec{r}_y, \vec{r}_u)$, $\vec{r}_q = (\vec{r}_p, \vec{r}_{\mu_{\mathcal{A}}})$, and $\vec{s}_x = (\vec{s}_y, \vec{s}_u)$, $\vec{s}_q = (\vec{s}_p, \vec{s}_{\mu_{\mathcal{A}}})$ hold. The building blocks of the preconditioner are as follows:

- $(\widehat{K}_h)^{-1}\vec{b}$ is realized by one multigrid V-cycle applied to the linear system with the scalar product matrix K_h (representing the discrete $H^1(\Omega)$ scalar product) and right-hand side \vec{b} . A number of ν_{GS} forward and reverse Gauss–Seidel smoothing steps are used, starting from an initial guess of $\vec{0}$. In Algorithm 3, the evaluation of $(\widehat{K}_h)^{-1}\vec{b}$ is denoted by `multigrid`(\vec{b}).
- $(\widehat{M}_h)^{-1}\vec{b}$ corresponds to ν_{SGS} symmetric Gauss–Seidel steps for the mass matrix M_h (representing the scalar product in $L^2(\Omega)$) with right-hand side \vec{b} and with an initial guess $\vec{0}$. This is denoted by `SGS`(\vec{b}) in Algorithm 3.
- As noted above, the evaluation of $(P_{\mathcal{A}}M_h^{-1}P_{\mathcal{A}}^{\top})^{-1}\vec{b}_{\mathcal{A}}$ is equivalent to solving the linear system (3.10). This is achieved efficiently by the preconditioned *projected* conjugate gradient (PPCG) method [9] in the standard scalar product, where

$$\begin{pmatrix} \text{diag}(M_h) & P_{\mathcal{A}}^{\top} \\ P_{\mathcal{A}} & 0 \end{pmatrix}$$

serves as a preconditioner. In Algorithm 3, this corresponds to the call $\text{PPCG}(\vec{b}_A, \mathcal{A}^+, \mathcal{A}^-)$. In practice, we use a relative termination tolerance of 10^{-12} for the residual in PPCG, which took at most 13 steps to converge in all examples. The reason for solving (3.10) practically to convergence is that intermediate iterates in conjugate gradient iterations depend nonlinearly on the right-hand side, and thus early termination would yield a nonlinear preconditioner \widehat{S} . Note that the projected conjugated gradient method requires an initial iterate consistent with the second equation $P_A \vec{r} = -\vec{b}_A$ in (3.10). Due to the structure of P_A , we can simply use $\vec{r}_A = -\vec{b}_A$, $\vec{r}_I = \vec{0}$, and $\vec{\mu}_A = \vec{0}$ as initial iterate.

Before moving on to problems with different types of constraints, we briefly summarize the main features of the PCG iteration applied to the solution of the Newton step (3.8).

Remark 3.3.

1. The main efforts in every application of the preconditioner are the three multigrid cycles. No PDEs need to be solved. Moreover, the method is of linear complexity in the number of unknowns.
2. In the case, where the control u and thus the Lagrange multiplier ξ are discretized by piecewise constant functions, the mass matrix M_h in the scalar product \mathcal{X} becomes a diagonal matrix, and preconditioning for the blocks M_h and $P_A M_h^{-1} P_A^\top$ becomes trivial.
3. We recall that the matrix B in Algorithm 3 denotes the (2,1) block of (3.8), i.e.,

$$B = \begin{pmatrix} L_h & -M_h \\ \cdot & P_A \end{pmatrix}$$

in the control-constrained case presently considered. For the subsequent examples, B changes, but no other modifications to Algorithm 3 are required.

ALGORITHM 3. Application of the preconditioner according to (3.12).

Input: right-hand sides $\vec{s}_x = (\vec{s}_y, \vec{s}_u)$ and $\vec{s}_q = (\vec{s}_p, \vec{s}_{\mu_A})$, scaling parameters σ, τ , and active sets $\mathcal{A}^+, \mathcal{A}^-$

Output: solution $\vec{r}_x = (\vec{r}_y, \vec{r}_u)$ and $\vec{r}_q = (\vec{r}_p, \vec{r}_{\mu_A})$ of (3.12)

1. $\vec{r}'_y := \text{multigrid}(\sigma \vec{s}_y)$
 2. $\vec{r}'_u := \text{SGS}(\sigma \vec{s}_u)$
 3. $\begin{pmatrix} \vec{s}'_p \\ \vec{s}'_{\mu_A} \end{pmatrix} := B \begin{pmatrix} \vec{r}'_y \\ \vec{r}'_u \end{pmatrix} - \begin{pmatrix} \vec{s}_p \\ \vec{s}_{\mu_A} \end{pmatrix}$
 4. $\vec{r}_p := \text{multigrid}(\tau \vec{s}'_p / \sigma)$
 5. $\vec{r}_{\mu_A} := \text{PPCG}(\tau \vec{s}'_{\mu_A} / \sigma, \mathcal{A}^+, \mathcal{A}^-)$
 6. $\begin{pmatrix} \vec{s}'_y \\ \vec{s}'_u \end{pmatrix} := \begin{pmatrix} \vec{s}_y \\ \vec{s}_u \end{pmatrix} - B \begin{pmatrix} \vec{r}_p \\ \vec{\mu}_A \end{pmatrix}$
 7. $\vec{r}_y := \text{multigrid}(\sigma \vec{s}'_y)$
 8. $\vec{r}_u := \text{SGS}(\sigma \vec{s}'_u)$
 9. **return** $\vec{r}_y, \vec{r}_u, \vec{r}_p, \vec{r}_{\mu_A}$
-

3.2. Regularized state-constrained problems: Mixed constraints. In this section, we address optimal control problems with mixed control-state constraints. They can be viewed as one way of regularizing problems with pure state constraints; see [16]:

$$(MC) \quad \begin{aligned} & \text{Minimize} && \frac{1}{2} \|y - y_d\|_{L^2(\Omega)}^2 + \frac{\nu}{2} \|u\|_{L^2(\Omega)}^2 \\ & \text{s.t.} && \begin{cases} -\Delta y + y = u & \text{in } \Omega, \\ \partial_n y = 0 & \text{on } \Gamma \end{cases} \\ & \text{and} && y_a \leq \varepsilon u + y \leq y_b \quad \text{a.e. in } \Omega. \end{aligned}$$

We point out the main differences from the control-constrained case. The first-order system of necessary and sufficient optimality conditions of (MC) can be expressed as follows:

$$(3.13a) \quad -\Delta p + p = -(y - y_d) - \xi \quad \text{in } \Omega, \quad \partial_n p = 0 \quad \text{on } \Gamma,$$

$$(3.13b) \quad \nu u - p + \varepsilon \xi = 0 \quad \text{a.e. in } \Omega,$$

$$(3.13c) \quad -\Delta y + y = u \quad \text{in } \Omega, \quad \partial_n y = 0 \quad \text{on } \Gamma,$$

$$(3.13d) \quad \xi - \max\{0, \xi + c(\varepsilon u + y - y_b)\} - \min\{0, \xi - c(y_a - \varepsilon u - y)\} = 0 \quad \text{a.e. in } \Omega.$$

A Newton step for the solution of (3.13) reads (in its symmetric form) as

$$\begin{pmatrix} I & \cdot & L^* & \chi_{\mathcal{A}} \\ \cdot & \nu I & -I & \varepsilon \chi_{\mathcal{A}} \\ L & -I & \cdot & \cdot \\ \chi_{\mathcal{A}} & \varepsilon \chi_{\mathcal{A}} & \cdot & \cdot \end{pmatrix} \begin{pmatrix} y \\ u \\ p \\ \xi_{\mathcal{A}} \end{pmatrix} = \begin{pmatrix} y_d \\ 0 \\ 0 \\ \chi_{\mathcal{A}^+} y_b + \chi_{\mathcal{A}^-} y_a \end{pmatrix}$$

with active sets similar to as in (3.2). The Newton system fits into our framework (1.2) with the identifications

$$\begin{aligned} x &= (y, u) \in X = H^1(\Omega) \times L^2(\Omega), \\ q &= (p, \xi_{\mathcal{A}}) \in Q = H^1(\Omega) \times L^2(\mathcal{A}) \end{aligned}$$

and bilinear forms

$$\begin{aligned} a((y, u), (z, v)) &:= (y, z)_{L^2(\Omega)} + \nu (u, v)_{L^2(\Omega)}, \\ b((y, u), (p, \xi_{\mathcal{A}})) &:= (y, p)_{H^1(\Omega)} - (u, p)_{L^2(\Omega)} + (y, \xi_{\mathcal{A}})_{L^2(\mathcal{A})} + \varepsilon (u, \xi_{\mathcal{A}})_{L^2(\mathcal{A})}. \end{aligned}$$

LEMMA 3.4. *The bilinear forms $a(\cdot, \cdot)$ and $b(\cdot, \cdot)$ satisfy assumptions (1.3a)–(1.3e) with constants $\|a\| = \max\{1, \nu\}$, $\alpha_0 = \nu/2$, $\|b\| = 2 \max\{1, \varepsilon\}$, and $k_0 = \min\{1, \varepsilon\}$, independent of the active set \mathcal{A} .*

Proof. The bilinear form a is the same as in (3.5a); hence we refer the reader to the proof of Lemma 3.1 for $\|a\|$ and α_0 . (Although $\ker B$ has changed, we used only the condition $Ly - u = 0$ in the proof of Lemma 3.1, which remains valid.) The

boundedness of b follows from

$$\begin{aligned} b((y, u), (p, \xi_{\mathcal{A}})) &\leq \|y\|_{H^1(\Omega)}\|p\|_{H^1(\Omega)} + \|u\|_{L^2(\Omega)}\|p\|_{L^2(\Omega)} + \|y\|_{L^2(\mathcal{A})}\|\xi_{\mathcal{A}}\|_{L^2(\mathcal{A})} \\ &\quad + \varepsilon \|u\|_{L^2(\mathcal{A})}\|\xi_{\mathcal{A}}\|_{L^2(\mathcal{A})} \\ &\leq (\|y\|_{H^1(\Omega)} + \max\{1, \varepsilon\} \|u\|_{L^2(\Omega)}) (\|p\|_{H^1(\Omega)} + \|\xi_{\mathcal{A}}\|_{L^2(\mathcal{A})}) \\ &\leq 2 \max\{1, \varepsilon\} \|(y, u)\|_X \|(p, \xi_{\mathcal{A}})\|_Q. \end{aligned}$$

Finally, we obtain the inf-sup condition for b as follows: For given $(p, \xi_{\mathcal{A}}) \in Q$, choose $y = p$ and $u = \xi_{\mathcal{A}}$ (by extending it by zero outside of \mathcal{A}). Then

$$\begin{aligned} b((y, u), (p, \xi_{\mathcal{A}})) &= (p, p)_{H^1(\Omega)} - (\xi_{\mathcal{A}}, p)_{L^2(\Omega)} + (p, \xi_{\mathcal{A}})_{L^2(\mathcal{A})} + \varepsilon (\xi_{\mathcal{A}}, \xi_{\mathcal{A}})_{L^2(\mathcal{A})} \\ &\geq \|p\|_{H^1(\Omega)}^2 + \varepsilon \|\xi_{\mathcal{A}}\|_{L^2(\mathcal{A})}^2 \\ &\geq \min\{1, \varepsilon\} \|(y, u)\|_X \|(p, \xi_{\mathcal{A}})\|_Q. \quad \square \end{aligned}$$

The leading term in the estimate for the preconditioned condition number is thus

$$\kappa_{\text{MC}}(\widehat{\mathcal{K}}^{-1}\mathcal{K}) \sim \frac{\beta}{\alpha} = 8 \frac{\max\{1, \nu\}}{\nu} \left(\frac{\max\{1, \varepsilon\}}{\min\{1, \varepsilon\}} \right)^2,$$

which scales like ε^{-2} for small ε .

Discretization. The discretization is carried out like in section 3.1. The discrete Newton step becomes

$$\begin{pmatrix} M_h & \cdot & L_h^\top & P_{\mathcal{A}}^\top \\ \cdot & \nu M_h & -M_h & \varepsilon P_{\mathcal{A}}^\top \\ L_h & -M_h & \cdot & \cdot \\ P_{\mathcal{A}} & \varepsilon P_{\mathcal{A}} & \cdot & \cdot \end{pmatrix} \begin{pmatrix} \vec{y} \\ \vec{u} \\ \vec{p} \\ \vec{\mu}_{\mathcal{A}} \end{pmatrix} = \begin{pmatrix} M_h \vec{y}_d \\ 0 \\ 0 \\ P_{\mathcal{A}^+} \vec{y}_b + P_{\mathcal{A}^-} \vec{y}_a \end{pmatrix}.$$

The scalar products and thus the preconditioner are the same (with different constants σ and τ) as in the control-constrained case; see section 3.1. We point out only that now

$$B = \begin{pmatrix} L_h & -M_h \\ P_{\mathcal{A}} & \varepsilon P_{\mathcal{A}} \end{pmatrix}$$

has to be used in Algorithm 3.

3.3. Regularized state-constrained problems: Moreau–Yosida approach.

An alternative regularization approach for state-constrained problems is given in terms of the Moreau–Yosida penalty function (see [13])

$$\begin{aligned} \text{Minimize} \quad & \frac{1}{2} \|y - y_d\|_{L^2(\Omega)}^2 + \frac{\nu}{2} \|u\|_{L^2(\Omega)}^2 + \frac{1}{2\varepsilon} \|\max\{0, y - y_b\}\|_{L^2(\Omega)}^2 \\ \text{(MY)} \quad & + \frac{1}{2\varepsilon} \|\min\{0, y - y_a\}\|_{L^2(\Omega)}^2 \\ \text{s.t.} \quad & \begin{cases} -\Delta y + y = u & \text{in } \Omega, \\ \partial_n y = 0 & \text{on } \Gamma. \end{cases} \end{aligned}$$

The first-order system of necessary and sufficient optimality conditions of (MY) can be expressed as follows:

$$(3.14a) \quad -\Delta p + p = -(y - y_d) - \xi \quad \text{in } \Omega, \quad \partial_n p = 0 \quad \text{on } \Gamma,$$

$$(3.14b) \quad \nu u - p = 0 \quad \text{a.e. in } \Omega,$$

$$(3.14c) \quad -\Delta y + y = u \quad \text{in } \Omega, \quad \partial_n y = 0 \quad \text{on } \Gamma$$

with the abbreviation $\xi = \varepsilon^{-1} \max\{0, y - y_b\} + \varepsilon^{-1} \min\{0, y_a - y\}$. A Newton step for the solution of (3.14) reads as

$$\begin{pmatrix} I + \frac{\chi_{\mathcal{A}}}{\varepsilon} & \cdot & L^* \\ \cdot & \nu I & -I \\ L & -I & \cdot \end{pmatrix} \begin{pmatrix} y \\ u \\ p \end{pmatrix} = \begin{pmatrix} y_d + \frac{1}{\varepsilon}(\chi_{\mathcal{A}^+} y_b + \chi_{\mathcal{A}^-} y_a) \\ 0 \\ 0 \end{pmatrix}$$

with active sets

$$\begin{aligned} \mathcal{A}^+ &= \{x \in \Omega : y - y_b > 0\}, \\ \mathcal{A}^- &= \{x \in \Omega : y_a - y < 0\}. \end{aligned}$$

The variables and bilinear forms are now given by

$$\begin{aligned} x &= (y, u) \in X = H^1(\Omega) \times L^2(\Omega), \\ q &= p \in Q = H^1(\Omega) \end{aligned}$$

and

$$\begin{aligned} a((y, u), (z, v)) &:= (y, z)_{L^2(\Omega)} + \varepsilon^{-1} (y, z)_{L^2(\mathcal{A})} + \nu (u, v)_{L^2(\Omega)}, \\ b((y, u), p) &:= (y, p)_{H^1(\Omega)} - (u, p)_{L^2(\Omega)}. \end{aligned}$$

Note that, in contrast to the mixed-constrained approach (section 3.2), the regularization parameter ε now appears in the bilinear form a instead of b .

LEMMA 3.5. *The bilinear forms $a(\cdot, \cdot)$ and $b(\cdot, \cdot)$ satisfy assumptions (1.3a)–(1.3e) with constants $\|a\| = \max\{1 + \varepsilon^{-1}, \nu\}$, $\alpha_0 = \nu/2$, $\|b\| = \sqrt{2}$, and $k_0 = 1$, independent of the active set \mathcal{A} .*

The proof is similar to that of Lemmas 3.1 and 3.4 and is therefore omitted. The leading term in the estimate for the preconditioned condition number is now

$$\kappa_{\text{MY}}(\widehat{\mathcal{K}}^{-1}\mathcal{K}) \sim \frac{\beta}{\alpha} = 8 \frac{\max\{1 + \varepsilon^{-1}, \nu\}}{\nu},$$

which scales like ε^{-1} for small ε .

As mentioned before, problems (MC) and (MY) can be viewed as regularized versions of the state-constrained problem (SC) below. Provided that the regularization errors are comparable for identical values of ε , the estimates for the preconditioned condition number clearly favor the Moreau–Yosida approach; see Figure 1. This is confirmed by the numerical results in section 4.

Discretization. The discrete counterpart of (MY) is

$$\begin{aligned} \text{Minimize} \quad & \frac{1}{2}(\vec{y} - \vec{y}_d)^\top M_h(\vec{y} - \vec{y}_d) + \frac{\nu}{2}\vec{u}^\top M_h\vec{u} \\ & + \frac{1}{2\varepsilon} \max\{0, \vec{y} - \vec{y}_b\}^\top M_h \max\{0, \vec{y} - \vec{y}_b\} \\ & + \frac{1}{2\varepsilon} \min\{0, \vec{y} - \vec{y}_a\}^\top M_h \min\{0, \vec{y} - \vec{y}_a\} \\ \text{s.t.} \quad & L_h\vec{y} = M_h\vec{u}, \end{aligned} \tag{MY}_h$$

and its optimality conditions read as

$$L_h^\top \vec{p} = -M_h(\vec{y} - \vec{y}_d) - \chi_{\mathcal{A}^+} \varepsilon^{-1} M_h \max\{0, \vec{y} - \vec{y}_b\} - \chi_{\mathcal{A}^-} \varepsilon^{-1} M_h \min\{0, \vec{y} - \vec{y}_a\}, \tag{3.15a}$$

$$\nu M_h\vec{u} - M_h\vec{p} = 0, \tag{3.15b}$$

$$L_h\vec{y} = M_h\vec{u}, \tag{3.15c}$$

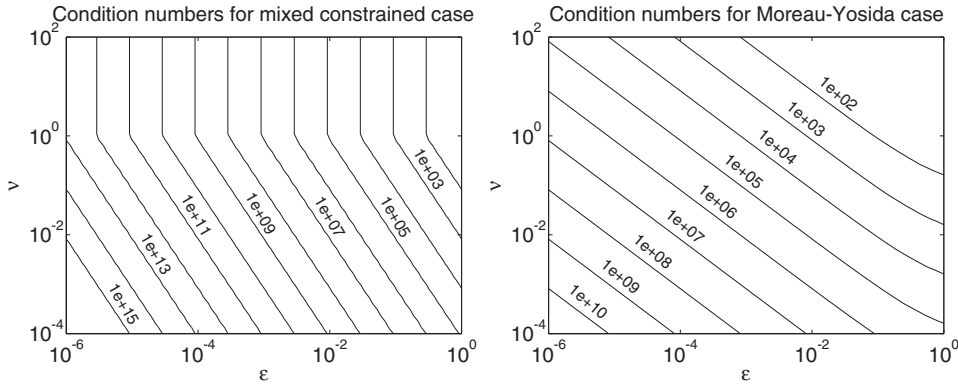


FIG. 1. Comparison of preconditioned condition numbers for mixed control-state constraints (left) and Moreau–Yosida regularization (right) as functions of the control cost coefficient ν and the regularization parameter ε .

where χ_{A^+} are the indices where $\bar{y} - \bar{y}_b > 0$, and similarly for χ_{A^-} . A Newton step for (3.15) amounts to solving

$$\begin{pmatrix} M_h + \frac{\chi_{A^+}}{\varepsilon} M_h \chi_{A^+} & \cdot & L_h^\top \\ \cdot & \nu M_h & -M_h \\ L_h & -M_h & \cdot \end{pmatrix} \begin{pmatrix} \bar{y} \\ \bar{u} \\ \bar{p} \end{pmatrix} = \begin{pmatrix} M_h \bar{y}_d + \frac{1}{\varepsilon} (\chi_{A^+} M_h \chi_{A^+} \bar{y}_b + \chi_{A^-} M_h \chi_{A^-} \bar{y}_a) \\ 0 \\ 0 \end{pmatrix}.$$

The preconditioner is the same as described in Algorithm 3, with the exception that now

$$B = \begin{pmatrix} L_h & -M_h \end{pmatrix}$$

is used, and step 5 can be omitted since no Lagrange multipliers associated with inequality constraints are present in the Moreau–Yosida case.

3.4. State-constrained problems. We briefly address an optimal control problem with pointwise state constraints:

$$\begin{aligned} \text{(SC)} \quad & \text{Minimize} \quad \frac{1}{2} \|y - y_d\|_{L^2(\Omega)}^2 + \frac{\nu}{2} \|u\|_{L^2(\Omega)}^2 \\ & \text{s.t.} \quad \begin{cases} -\Delta y + y = u & \text{in } \Omega, \\ \partial_n y = 0 & \text{on } \Gamma \end{cases} \\ & \text{and} \quad y_a \leq y \leq y_b \quad \text{in } \bar{\Omega}. \end{aligned}$$

It is well known that (SC) does not permit the same function space setting as in the previous sections. The associated Lagrange multiplier is only a measure [5], and the adjoint state belongs to $W^{1,s}(\Omega)$, where $s < N/(N - 1)$. Moreover, there is no theoretical foundation of using Newton-type methods in function space for the solution of (SC). Problems (MC) and (MY) can be considered regularized versions of (SC) which have better regularity properties.

In fact, the well posedness of a saddle point problem (1.2) is equivalent to the Lipschitz dependence of its unique solutions on the right-hand side (F, G) . Lipschitz stability for the adjoint state p , however, can be expected only w.r.t. $L^2(\Omega)$; see [10].

For completeness, we state the discrete optimality system

$$\begin{aligned} L_h^\top \vec{p} &= -M_h(\vec{y} - \vec{y}_d) + \vec{\mu}, \\ \nu M_h \vec{u} - M_h \vec{p} &= 0, \\ L_h \vec{y} &= M_h \vec{u}, \\ \vec{\mu} - \max\{0, \vec{\mu} + c(\vec{y} - \vec{y}_b)\} - \min\{0, \vec{\mu} - c(\vec{y}_a - \vec{y})\} &= 0 \end{aligned}$$

and the discrete Newton step

$$\begin{pmatrix} M_h & \cdot & L_h^\top & P_A^\top \\ \cdot & \nu M_h & -M_h & \cdot \\ L_h & -M_h & \cdot & \cdot \\ P_A & \cdot & \cdot & \cdot \end{pmatrix} \begin{pmatrix} \vec{y} \\ \vec{u} \\ \vec{p} \\ \vec{\mu}_A \end{pmatrix} = \begin{pmatrix} M_h \vec{y}_d \\ 0 \\ 0 \\ P_{A^+} \vec{y}_b + P_{A^-} \vec{y}_a \end{pmatrix}.$$

Note that the bilinear forms associated with the Newton step allow the same constants $\|a\|$, α_0 , and $\|b\|$ as in the control-constrained example, Lemma 3.1. However, k_0 tends to zero as the mesh size decreases.

4. Numerical results. In this section, we present several numerical experiments. Each of the examples emphasizes one particular aspect of our previous analysis.

4.1. Algorithmic details. We begin by describing the algorithmic details which are common to all examples, unless otherwise mentioned. The implementation was done in MATLAB.

Discretization. In the two-dimensional (2D) case, the discretization was carried out using piecewise linear and continuous (P_1) finite elements on polyhedral approximations of the unit circle $\Omega \subset \mathbb{R}^2$. The assembly of mass and stiffness matrices, as well as the construction of the prolongation operator (linear interpolation) for the multigrid preconditioner, was left to the PDE TOOLBOX. The restriction operator is the transpose of the prolongation.

In the 3D case, the underlying domain is $\Omega = (-1, 1)^3 \subset \mathbb{R}^3$ and the discretization is based on the standard second-order finite difference stencil on a uniform grid. The prolongation operator is again obtained by linear interpolation, with restriction chosen as its transpose. In all 3D examples, we used homogeneous Dirichlet (instead of Neumann) boundary conditions and the differential operator $-\Delta$ (instead of $-\Delta + I$). As a consequence, the constant k_0 in the inf-sup condition for the bilinear (constraint) form b is different from those specified in the examples in section 3. In any case, the constants $\|a\|$ and k_0 relevant for the scaling of the preconditioner were estimated numerically; see below.

Preconditioner. As described in section 3, the preconditioner \mathcal{K} is composed of individual preconditioners for the matrices representing the scalar products in the spaces $X = H^1(\Omega) \times L^2(\Omega)$ and $Q = H^1(\Omega) \times L^2(\mathcal{A})$, or $Q = H^1(\Omega)$ in the Moreau–Yosida case. We used a geometric multigrid approach as a building block in the preconditioner for the stiffness matrix K_h representing the $H^1(\Omega)$ scalar product; compare Algorithm 3. Each application of the preconditioner consisted of one V-cycle. The matrices on the coarser levels were generated by Galerkin projection (two

dimensions) or by reassembling (three dimensions), respectively. A direct solver was used on the coarsest level, which held 33 degrees of freedom in the 2D case and 27 in the 3D case. We used $\nu_{GS} = 3$ forward Gauss–Seidel sweeps as pre-smoothers and the same number of backward Gauss–Seidel sweeps as post-smoothers, which retain the symmetry of the preconditioner. The mass matrix belonging to $L^2(\Omega)$ was preconditioned using $\nu_{SGS} = 1$ symmetric Gauss–Seidel sweep. As pointed out in section 3, the appropriate scalar product for $L^2(\mathcal{A})$ is given by $P_{\mathcal{A}}M_h^{-1}P_{\mathcal{A}}^{\top}$. We solve systems with this matrix in their equivalent form (3.10) by the PPCG method [9], where (3.1) serves as a preconditioner. The relative termination tolerance for the residual was set to 10^{-12} . In fact, we found the `cgs` implementation in MATLAB preferable to PCG here, although every iteration requires two matrix-vector products.

Estimation of $\|a\|$ and k_0 and selection of scaling parameters σ and τ .

As described in section 2, the positive definiteness of the preconditioned saddle point matrix relies on a proper scaling of the preconditioner building blocks \mathcal{X} and \mathcal{Q} . In turn, a proper choice of scaling parameters σ and τ requires an estimate of $\|a\|$ and k_0 . Such estimates may be available analytically in some situations (see the examples in section 3). If they are not, they can be easily obtained as follows. We recall that $\|a\|$ and k_0 have to satisfy

$$A \preceq \|a\| \mathcal{X} \quad \text{and} \quad B\mathcal{X}^{-1}B^{\top} \succeq k_0^2 \mathcal{Q},$$

and smaller values of $\|a\|$ and larger values of k_0 are preferred. On the discrete level, viable estimates are obtained by computing the extreme eigenvalues of two generalized eigenvalue problems:

$$(4.1) \quad \begin{aligned} \|a\|' &:= \text{eigs}(A, X, 1, 'lm'); \\ (k_0')^2 &:= 1/\text{eigs}(Q, B*\text{inv}(X)*B', 1, 'lm'); \end{aligned}$$

in MATLAB notation. Naturally, these computations should be carried out for coarse level matrices A , \mathcal{X} , B , and \mathcal{Q} . Estimates for the scaling parameters are then computed from $\sigma := 0.9/\|a\|'$ and $\tau := 1.2/(k_0')^2$.

This adaptive parameter selection strategy was applied in all numerical examples, combined with the safeguarded choice of σ and τ (Algorithm 2), which seldomly became relevant. Note that the constants $\|a\|$ and k_0 can change with the active sets. We thus estimated them in every step of the outer Newton iteration to take full advantage of the adaptive strategy.

Remark 4.1. The quantities $\|a\|$ and k_0 are properties of the bilinear forms $a(\cdot, \cdot)$ and $b(\cdot, \cdot)$. Given appropriate discretizations, the same constants are also viable on all discrete levels. Nevertheless, the *optimal* constants, i.e., the smallest possible $\|a\|$ and the largest possible k_0 , will be slightly different for different grid levels.

In the following numerical examples, it turns out that this dependence is most pronounced for the problem with convection (Example 4.3) and in the mixed-constrained approximation of a purely state-constrained problem (Example 4.4). Since (4.1) finds the optimal constants on a coarse grid, they may not be valid on a fine grid for these examples. And thus the safeguarding strategy (Algorithm 2) may reject the initial scaling parameters σ and τ .

PCG method. We used our own implementation of the PCG method according to Algorithm 1, with an absolute termination tolerance of $\varepsilon_{\text{abs}} = 10^{-10}$. In every 50th iteration, the update of the residual \vec{s} in step 7 is replaced by a fresh evaluation of the residual, and the update of the preconditioned residual \vec{r} is replaced by $\widehat{\mathcal{K}}^{-1}\vec{s}$, in order to prevent rounding errors from accumulating.

Newton iteration. Since the optimality systems for all examples in section 3 are nonlinear, Newton's method was applied for their solution. The parameter in the active set strategy was chosen as $c = 1$. We recall that we focus here entirely on the efficient solution of individual Newton steps. Nevertheless, Newton's method was iterated until the convergence criterion $\|r\|_{L^2} \leq 10^{-8}$ was met, where r denotes the residual of the nonlinear equations in the respective optimality system. The reported run times and iteration numbers are averaged over all Newton steps taken for one particular example.

Computational environment. All computations were carried out on a compute server with 4 Intel Xeon CPUs (3 GHz) and 64 GB of RAM under MATLAB Version 7.9 (R2009b).

4.2. Examples.

Example 4.2 (comparison with direct solver). In this example, we compare the performance of the PCG solver with the sparse direct solver in MATLAB. The control-constrained example (CC) was set up with parameter $\nu = 10^{-2}$ and data $u_a = 0$, $u_b = 2$, and $y_d = 1$, where $|x_1| \leq 0.5$ and $y_d = 0$ elsewhere in the 2D case. In the 3D case, $u_a = 0$, $u_b = 2.5$, and $y_d = 1$, where $|x_1| \leq 0.5$ and $y_d = -2$ elsewhere. This data was chosen such that both upper and lower active sets are present at the solution. A hierarchy of uniformly refined grids was used, which resulted in a dimension of the state variable up to 100,000 in the 2D case and up to 250,000 in the 3D case. The overall number of unknowns for the direct solver applied to (3.4) is between three and four times these numbers, depending on the size of the active sets.

The plots in Figure 2 show that the direct solver is preferable to PCG in the 2D case for the discretization sizes used. For problem sizes larger than those shown, memory issues will eventually work in favor of the PCG solver. In the 3D case, the PCG iteration is clearly superior even for moderate discretization levels. This was to be expected due to an increased amount of fill in when factoring 3D finite element and finite difference matrices. The plots also confirm the linear complexity of the PCG iteration in terms of the problem size. Approximately 100 (45) PCG iterations per Newton step were required in the 2D (3D) cases. The convergence history is shown in Figure 3.

In what follows, we consider only 3D examples. There will be no further comparison with direct solvers due to their memory and run time limitations with large 3D problems.

Example 4.3 (a problem with convection). We consider an optimal control problem with an additional convection term in the bilinear form

$$b((y, u), (p, \xi_A)) := (y, p)_{H^1(\Omega)} + (\beta \cdot \nabla y, p)_{L^2(\Omega)} - (u, p)_{L^2(\Omega)} + (u, \xi_A)_{L^2(\mathcal{A})}.$$

Note that we do not change the scalar products, and thus there is no need to change the preconditioner. The boundedness constant becomes $\|b\| = 2(1 + \|\beta\|_{L^\infty(\Omega)})$, and thus we expect a deterioration of the convergence behavior for large values of β . Upwind finite differences were used in L_h to obtain stable discretizations. Note that this also stabilizes the adjoint operator L_h^\top , which has convection direction $-\beta$. The problem data is $\nu = 10^{-2}$, y_d as in Example 4.2, and the bounds were chosen as $-u_a = u_b = \infty$. We took

$$\beta \in \left\{ \begin{pmatrix} 10 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 100 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1000 \\ 0 \\ 0 \end{pmatrix} \right\}$$

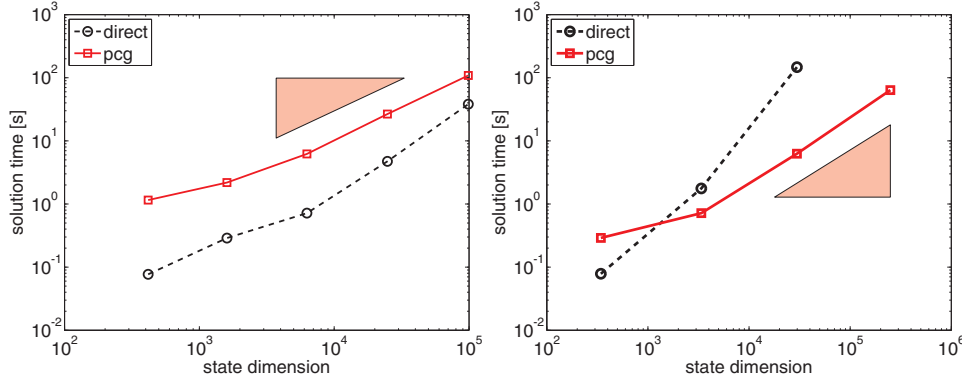


FIG. 2. The plots show the average solution time per Newton step vs. the dimension of the discretized state space. We compare the PCG method to the sparse direct solver of MATLAB applied to the linearized optimality system (3.4) of (CC) in the 2D (left) and 3D (right) cases. The setup is described in Example 4.2. The triangle has slope 1, and it visualizes the linear complexity of the PCG solver w.r.t. the number of unknowns.

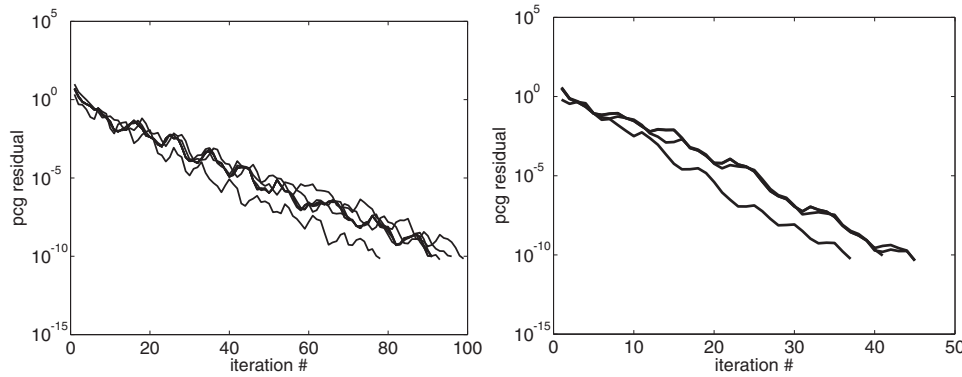


FIG. 3. The plots show the convergence history of the PCG residual in the 2D (left) and 3D (right) cases on the finest grid for all Newton steps. The setup is described in Example 4.2.

as convection directions.

We begin by observing that the optimal values of k_0 in (2.1d) on the discrete level depend on the mesh size h in this example. This seems to be an effect introduced by the upwind discretization, as it is not present in (unstable) discretizations by central differences. As a consequence, the safeguarding strategy (Algorithm 2) tended to reject the initial estimates (4.1) of $\|a\|$ and k_0 , which are obtained from coarse grid matrices, several times. With β and/or the number of unknowns increasing, the number of necessary corrections to these initial estimates also increased. And hence, appropriate values of the scaling parameters σ and τ depend noticeably on the mesh size in this example.

Figure 4 shows the convergence behavior for various discretization levels and values of β . For large values of β , we see a pronounced mesh dependence of the convergence history. As mentioned above, this is due to the upwind stabilization, which leads to a mesh dependent k_0 and thus to a deterioration of the preconditioned condition number on refined grids.

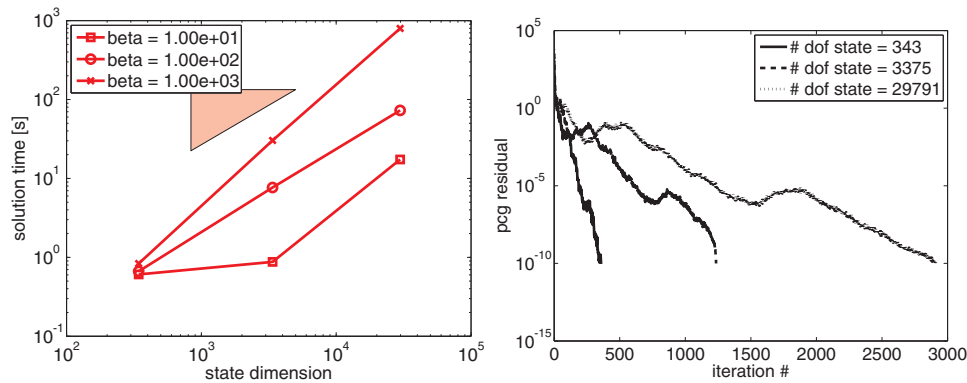


FIG. 4. The left plot shows the solution time for the single Newton step vs. the dimension of the discretized state space in a series of unconstrained problems with convection. The setup is described in Example 4.3. The right plot displays the convergence history of the PCG residual on various grid levels for $\beta = (1000, 0, 0)^T$.

Example 4.4 (comparison of mixed constraints and the Moreau–Yosida approach). In this example, we compare the performance of the preconditioned conjugate gradient solver for problems (MC) and (MY) in three dimensions. Both problems are regularized versions of the state-constrained problem (SC), and they are considered for a decreasing sequence of values for ε . The problem data is as follows: We choose $y_b = 0.0$ and $y_a = -\infty$, $\nu = 10^{-2}$, and y_d as in Example 4.2.

TABLE 1

Comparison of errors introduced by the (MC) and (MY) regularizations of the state-constrained problem (SC). See Example 4.4. J^{MC} and J^{MY} denote the values of the objective functionals in problems (MC) and (MY), and J denotes the value of the objective for the unregularized problem (SC) solved with a direct solver.

ε	$\ u^{\text{MC}} - u\ _{L^2}$	$\ u^{\text{MY}} - u\ _{L^2}$	$ J^{\text{MC}} - J $	$ J^{\text{MY}} - J $
1.00e-01	1.91e+00	1.02e+00	7.04e-02	2.62e-02
1.00e-02	5.25e-01	3.65e-01	9.32e-03	8.56e-03
1.00e-03	9.23e-02	8.05e-02	1.01e-03	1.34e-03
1.00e-04	1.18e-02	1.18e-02	1.46e-04	1.46e-04
1.00e-05	1.29e-03	1.29e-03	1.48e-05	1.48e-05

Table 1 shows the error introduced by regularization. The error was computed using a reference solution of the purely state-constrained problem, obtained on an intermediate grid level. We infer that, for identical values of ε , the errors in the optimal control and in the objective function are of comparable size for both approaches. However, the preconditioned condition numbers are dramatically different. We recall from section 3 that

$$\begin{aligned}\kappa_{\text{MC}}(\widehat{\mathcal{K}}^{-1}\mathcal{K}) &\sim \nu^{-1}\varepsilon^{-2}, \\ \kappa_{\text{MY}}(\widehat{\mathcal{K}}^{-1}\mathcal{K}) &\sim \nu^{-1}(1 + \varepsilon^{-1}).\end{aligned}$$

We thus expect the PCG iteration numbers for the (MC) case to be significantly higher than for the (MY) case. This is confirmed by our numerical experiments; see Figures 5 and 6.

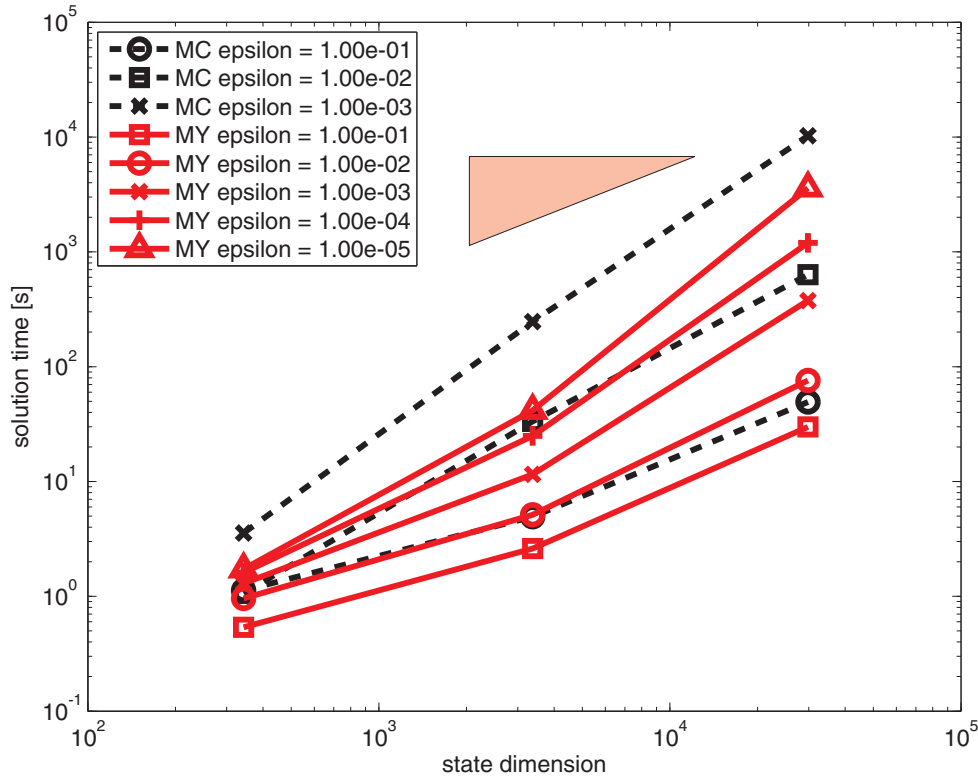


FIG. 5. The plot shows the average solution time per Newton step vs. the dimension of the discretized state space. Both regularization approaches (MC) and (MY) are compared for various levels of the regularization parameter ϵ . The setup is described in Example 4.4.

In the mixed-constrained case (MC), the magnitude of the preconditioned condition number at $\epsilon = 10^{-3}$ has several negative effects. First, we had to restrict the number of PCG steps to 3000. Beyond this iteration number, there was no further reduction of the residual. This is turn led to an increased number of Newton steps (not shown) and thus to an overall increased solution time. Moreover, as discussed in Remark 4.1, the optimal constant k_0 in the mixed-constrained problem depends on the mesh size in a noticeable way for small values of ϵ . And thus the estimated scaling parameters σ and τ had to be corrected several times before the PCG iteration went through. This led to the superlinear behavior of the run time w.r.t. the number of state variables in Figure 4.

The situation was significantly better in the Moreau–Yosida case (MY). It was computationally feasible to reduce ϵ to 10^{-5} for this setup. Moreover, the estimated constants $\|a\|'$ and k'_0 on a coarse level were viable for the fine levels as well.

In Example 4.4, the largest problem solved for $\epsilon = 10^{-5}$ has about 30,000 state variables (and the same number of control and adjoint state variables). The solution required 11 Newton steps with an average CPU time of about 5.5 minutes per Newton step. In a practical algorithm, a nested approach should be used, where early Newton steps operate on a coarse grid. Then the solution of larger problems becomes feasible, as demonstrated in the following and final example.

Example 4.5 (nested approach for Moreau–Yosida regularization). The setup in

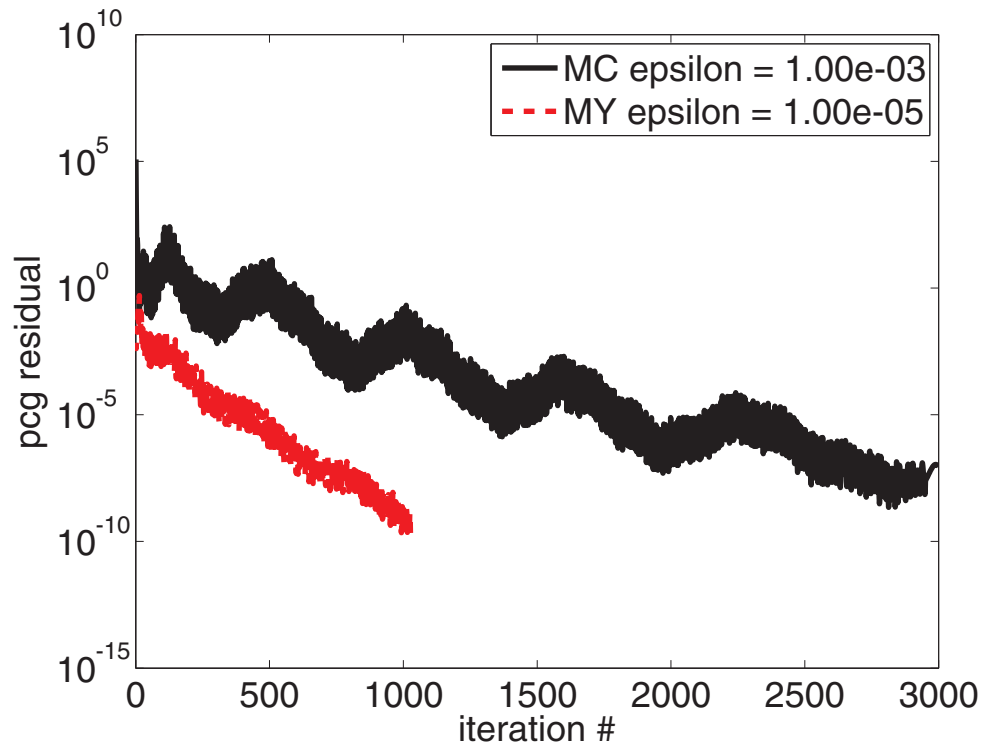


FIG. 6. The plot shows the convergence history of the PCG residual on the finest grid level for all Newton steps and $\varepsilon = 10^{-3}$ in the mixed-constrained case and $\varepsilon = 10^{-5}$ in the Moreau–Yosida case. The setup is described in Example 4.4.

this example is the same as in Example 4.4. We fix $\varepsilon = 10^{-4}$ and solve the Moreau–Yosida approximation problem on a sequence of grid levels. On each level, we use the prolongation of the previous solution as an initial guess. The Newton iteration is driven to convergence on each level.

With this nested approach, it was computationally feasible to increase the number of state variables to 250,000 (and the same number of control and adjoint state variables, plus Lagrange multipliers). Three or four Newton steps were carried out on each level, with an average number of about 450 PCG iterations each. The overall solution time was approximately 7 hours and 30 minutes, almost all of which was spent on the finest grid level; see Figure 7 and Table 2 for details. The maximum constraint violation at the solution is $\|\max\{0, y_h - y_b\}\|_{L^\infty(\Omega)} = 8.62 \cdot 10^{-4}$.

5. Concluding remarks.

Summary and conclusions. In this paper, we studied the application of the PCG method to linearized optimality systems arising in optimal control problems with constraints. This becomes possible through the use of appropriate symmetric indefinite preconditioners and a related scalar product. Problems with elliptic PDEs and pointwise control and regularized state constraints were considered as prototypical examples.

It stands out as a feature of this approach that one can simply use properly scaled preconditioners for the scalar product matrices as building blocks for the precondi-

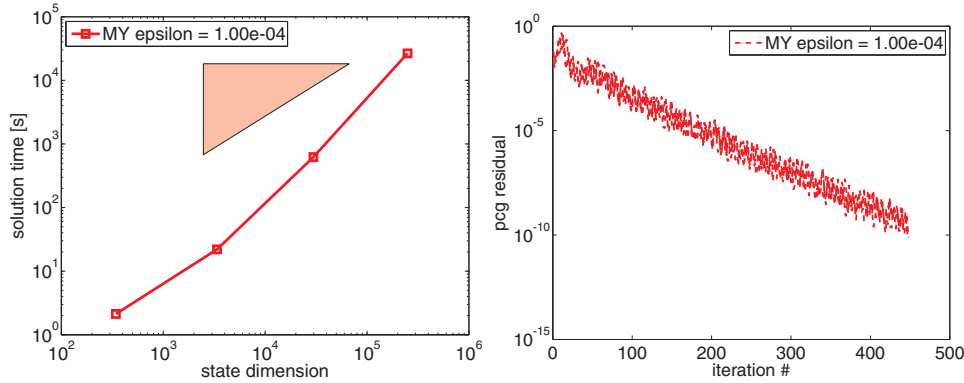


FIG. 7. The left plot shows the average solution time per Newton step vs. the dimension of the discretized state space in a nested approach for the Moreau–Yosida regularization. The setup is described in Example 4.5. The right plot shows the convergence history of the PCG residual on the finest grid level for all Newton steps.

TABLE 2

The table shows the convergence behavior in a nested approach. The setup is described in Example 4.5. We show the number of degrees of freedom for the state variable on each grid level, the residual after each Newton step, and the number of PCG iterations required to solve that particular Newton step. We also show the combined CPU time for all Newton steps on a particular grid level.

Level	# dofs	$\ r\ _{L^2}$	# PCG	CPU time
2	343	4.37e+00	33	2s
		6.74e-04	245	
		5.13e-04	207	
		1.68e-10	168	
3	3375	5.63e-01	359	22s
		3.66e-03	597	
		2.30e-11	429	
4	29791	7.31e-04	357	621s
		6.24e-05	455	
		7.96e-06	408	
		3.01e-12	436	
5	250047	5.47e-05	446	26545s
		1.75e-06	446	
		7.12e-09	447	

tioner. Such preconditioners are readily available. In our computational experiments, we used multigrid cycles for those variables whose natural scalar products involve derivatives, i.e., the state and adjoint variables. With these preconditioners, the solution of the linearized optimality systems is of linear complexity in the number of unknowns, and no PDEs need to be solved in the process.

In order to regularize the state-constrained problems, mixed control-state constraints and a Moreau–Yosida penalty approach were considered and compared. It turned out that, from our computational point of view, the penalty approach is clearly superior to the mixed-constrained approach. At the same level of regularization error, the preconditioned condition numbers are of order $1/\varepsilon$ for the penalty approach but of order $1/\varepsilon^2$ with mixed constraints. This was confirmed by numerical experiments.

The solution of state-constrained optimal control problems in three dimensions is computationally challenging. In fact, numerical results are hardly found in the

literature. The approach presented here pushes the frontier towards larger problems. In our computational experiments, the largest problem solved has about 250,000 state variables, and the same number of control and adjoint state variables, plus Lagrange multipliers.

Outlook. The main effort in every iteration of the PCG loop (Algorithm 1) is the application of the preconditioner. In our examples, this essentially amounts to three multigrid cycles and the solution of $P_{\mathcal{A}} M_h^{-1} P_{\mathcal{A}}^T \vec{\mu}_A = \vec{b}_A$ (see Algorithm 3). Thus every PCG iteration is relatively inexpensive. Profiling experiments on the grid with 250,000 unknowns shows that every call to the preconditioner in our MATLAB implementation took about 1.4 seconds. Clearly, there is room for improvement in using another computational environment and parallel preconditioners, e.g., based on domain decomposition.

The key issue, however, is to reduce the preconditioned condition number and thus the number of required PCG iterations. In [19], the authors used scalar products which depend on the control cost parameter ν . In this way, they found very low condition numbers independent of ν for an unconstrained optimal control problem. They do not discuss, however, the impact of these norms on accuracy and error tolerances. The investigation of ε dependent norms for regularized state-constrained problems in order to reduce the iteration numbers would be worthwhile and is postponed to follow-up work.

The analysis in [19] relies on the positive semidefiniteness of the (1,1) block A ; see (1.3e). In nonlinear optimal control problems, this condition may not be satisfied, not even in the vicinity of an optimal solution satisfying second-order sufficient conditions. Quasi-Newton techniques could be applied to overcome this problem, but this issue deserves further investigation.

Finally, we found that upwind stabilization of convection dominated problems can render the inf-sup constant k_0 mesh dependent, which results in a mesh dependent convergence behavior. Other stabilization techniques without this deficiency should thus be preferred.

REFERENCES

- [1] A. BATTERMANN AND M. HEINKENSCHLOSS, *Preconditioners for Karush-Kuhn-Tucker matrices arising in the optimal control of distributed systems*, in Control and Estimation of Distributed Systems, Internat. Ser. Numer. Math. 126, W. Desch, F. Kappel, and K. Kunisch, eds., Birkhäuser, Basel, 1998, pp. 15–32.
- [2] A. BATTERMANN AND E. SACHS, *Block preconditioners for KKT systems in PDE-governed optimal control problems*, in Fast Solution of Discretized Optimization Problems, Internat. Ser. Numer. Math. 138, Birkhäuser, Basel, 2001, pp. 1–18.
- [3] G. BIROS AND O. GHATTAS, *Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization. Part I: The Krylov-Schur solver*, SIAM J. Sci. Comput., 27 (2005), pp. 687–713.
- [4] J. BRAMBLE AND J. PASCIAK, *A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems*, Math. Comp., 50 (1988), pp. 1–17.
- [5] E. CASAS, *Control of an elliptic problem with pointwise state constraints*, SIAM J. Control Optim., 24 (1986), pp. 1309–1318.
- [6] H. S. DOLLAR, N. I. M. GOULD, W. H. A. SCHILDERS, AND A. J. WATHEN, *Implicit-factorization preconditioning and iterative solvers for regularized saddle-point systems*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 170–189.
- [7] H. S. DOLLAR, N. I. M. GOULD, M. STOLL, AND A. J. WATHEN, *Preconditioning saddle-point systems with applications in optimization*, SIAM J. Sci. Comput., 32 (2010), pp. 249–270.
- [8] S. DOLLAR, N. GOULD, AND A. WATHEN, *On implicit-factorization constraint preconditioners*, in Large-Scale Nonlinear Optimization, Nonconvex Optim. Appl. 83, G. D. Pillo and M. Roma, eds., Springer, Berlin, 2006, pp. 61–82.

- [9] 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.
- [10] R. GRIESSE, *Lipschitz stability of solutions to some state-constrained elliptic optimal control problems*, Z. Anal. Anwend., 25 (2006), pp. 435–455.
- [11] M. HINTERMÜLLER, K. ITO, AND K. KUNISCH, *The primal-dual active set strategy as a semi-smooth Newton method*, SIAM J. Optim., 13 (2003), pp. 865–888.
- [12] M. HINTERMÜLLER, I. KOPACKA, AND S. VOLKWEIN, *Mesh-independence and preconditioning for solving control problems with mixed control-state constraints*, ESAIM Control Optim. Calc. Var., 15 (2009), 626–652.
- [13] K. ITO AND K. KUNISCH, *Semi-smooth Newton methods for state-constrained optimal control problems*, Systems Control Lett., 50 (2003), pp. 221–228.
- [14] 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.
- [15] T. MATHEW, M. SARKIS, AND C. SCHAEERER, *Analysis of block matrix preconditioners for elliptic optimal control problems*, Numer. Linear Algebra Appl., 14 (2007), pp. 257–279.
- [16] C. MEYER, U. PRÜFERT, AND F. TRÖLTZSCH, *On two numerical methods for state-constrained elliptic control problems*, Optim. Methods Software, 22 (2007), pp. 871–899.
- [17] T. REES, H. S. DOLLAR, AND A. J. WATHEN, *Optimal solvers for PDE-constrained optimization*, SIAM J. Sci. Comput., 32 (2010), pp. 271–298.
- [18] T. REES AND M. STOLL, *Block triangular preconditioners for PDE constrained optimization*, Numer. Linear Algebra Appl., to appear.
- [19] 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.
- [20] G. STADLER, *Elliptic optimal control problems with L^1 -control cost and applications for the placement of control devices*, Comput. Optim. Appl., 44 (2009), pp. 159–181.
- [21] F. TRÖLTZSCH, *Optimale Steuerung partieller Differentialgleichungen*, Vieweg, Wiesbaden, 2005.