On the numerical solution
of state- and control-constrained optimal control problems

Ivo Dravins\(^1\) and Maya Neytcheva\(^1\)

\(^1\)Department of Information Technology, Uppsala University, Uppsala, Sweden

April 13, 2021

Keywords: PDE-constrained optimization, state- and control-constrained problems, sparse control, block preconditioning, semismooth Newton method

Abstract
We consider the iterative solution of algebraic systems, arising in optimal control problems, constrained by a partial differential equation, with additional box constraints on the state and the control variables, and sparsity imposed on the control. A nonsymmetric two-by-two block preconditioner is analysed and tested for a wide range of problem, regularization and discretization parameters. The constraint equation characterizes convection-diffusion processes.

1 Introduction and problem setting
Optimal control problems with constraints given by partial differential equations (PDE) arise in numerous important applications, where we pose the task to control processes, governed by PDEs. In addition to the state equation constraint, in practice it is often required that the solution obeys also other types of restrictions, such as inequality or sparsity constraints. Due to their complexity, such problems can in general only be analysed via computer simulations. During the simulation process we face the necessity to solve large scale nonlinear and/or linear systems of algebraic equations, making the use of iterative solution methods and preconditioning a must.

In this paper we analyse and illustrate numerically the performance of a particular nonsymmetric two-by-two block preconditioner, suitable for PDE-constrained optimal control problems with additional inequality constraints on the state and the control variables.

The primary optimal control problem we consider in this work reads as follows: find a solution \((u,v)\) that minimizes the cost functional

\[
\mathcal{J}(u,v) = \frac{1}{2} \int_{\Omega} |u-u_d|^2 \, dx + \frac{\alpha}{2} \int_{\Omega} |v|^2 \, dx
\]  

(1)
subject to the stationary convection-diffusion equation

\[ L_u = -\epsilon \Delta u + (\nabla \cdot b)u = v + f \quad \text{in} \quad \Omega \subset \mathbb{R}^d, \quad d = 2, 3 \]

(2)

with proper boundary conditions. Above, \( u \) is the state variable, \( u_d \) is some given desired (target) state we want to achieve and \( v \) is the control variable, imposed as a source term, to attain the target; \( b \) is a given vector field and \( 0 < \epsilon \leq 1 \) is a problem parameter, which determines the strength of convection related to diffusion. The space domain \( \Omega \) is assumed to be convex polyhedral. The constant \( \alpha \) is the Tikhonov regularization parameter, which is positive and small.

We follow the well-established technique to incorporate the various constraints in a Lagrangian functional, introducing corresponding adjoint variables (Lagrange multipliers). The Lagrange functional, after including the state equation takes the form

\[ L(u, v, p) = J(u, v) + \int_{\Omega} (L_u - f - v) pdx. \]

(3)

The variable \( p \) denotes the Lagrangian multiplier to handle the state equation.

Next, we add inequality constraints for the control, referred to as box-constraints, namely, \( v \leq u \leq \bar{v}, \; v < 0 < \bar{v} \) ([1]) together with the requirement that the control variable is sparse, thus, it is zero in some parts of the domain. The latter is achieved by introducing a \( L^1 \) term into the Lagrange functional which then takes the form

\[ L(u, v, p) = J(u, v) + \int_{\Omega} (L_u - f - v) pdx + \beta \int_{\Omega} |v| dx, \]

(4)

where \( \int_{\Omega} |v| dx = \|v\|_{L^1(\Omega)} \). The \( L^1 \) norm constraint introduces one more regularization parameter \( \beta \), again positive and small. For details, see [2, 3, 4] and with emphasis on solution techniques for such problems, for instance, [5, 6, 7, 8].

Finally, we impose inequality constraints on the state, requiring that \( u \leq u \leq \bar{u} \) holds in \( \Omega \). We mention that in practice the box constraints could be imposed only on a part of the domain, \( \Omega_0 \subset \Omega \), however we assume here that \( \Omega_0 = \Omega \).

State constraints can be handled in various ways, for instance, using the Moreau-Yosida penalty function method, cf. e.g. [9], or the Lavrentiev regularization method, both approaches compared in [10]. As in [11], we choose to use the former method, which does not introduce an additional Lagrange multiplier. Instead, we modify the Lagrange functional from (4) and minimize

\[ L(y, u, p) = J(u, v) + \int_{\Omega} (L_u - f - v) pdx + \beta \int_{\Omega} |v| dx \]

\[ + \frac{1}{2\gamma} \int_{\Omega} \max\{0, u - \bar{u}\}^2 dx + \frac{1}{2\gamma} \int_{\Omega} \min\{0, u - \bar{u}\}^2 dx. \]

(5)

Here \( 0 < \gamma < 1 \) is yet another small regularization parameter.
Formulation (5) and how to efficiently compute its solution in discrete form is the main target of this work.

The area of optimal problems constrained by PDEs has been of profound research interest for decades. The published research is immense. In this paper we refer to some general sources, such as [12, 13, 14] and to particular articles and the references therein.

As stated above, we aim at developing a fast and reliable numerical solution method for the discrete counterpart of (5), having three additional constraints, namely, box constraints on the state, restrictions on the sparsity of the control, which includes also box-type constraints on the control itself. As optimization framework we choose the first order necessary (Karush-Kuhn-Tucker, KKT) conditions, entailing large scale nonlinear and linear systems of algebraic equations to be solved. The latter triggers the necessity to use iterative solution methods and robust and computationally efficient preconditioners to accelerate the convergence of the iterative methods when solving the linear systems occurring within nonlinear iterations.

Designing highly efficient solution techniques for the distributed control problem with no additional constraints on the state and the control is well studied. Optimal and near optimal preconditioning methods are available, cf., e.g., [15, 16, 17, 18, 19, 20, 21, 22]. Applying box constraints on the state is discussed in e.g. [13] and numerical methods for that formulation of the problem are considered in, e.g., [23, 24] for both state and control constraints, and later in [11, 25]. Imposing sparsity on the control variable is discussed in [2, 3, 4, 5, 6, 7]. Up to our knowledge, constructing preconditioners for the case when all additional constraints are active is not studied by other authors in the semismooth Newton framework. Preconditioning is studied in [8] but using another optimization framework, namely, interior point methods. The systems to solve in the latter setting are of dimension larger than that of KKT systems. Initial testing of the approach, discussed in this paper for the case when the state constraint is given by the Poisson equation is presented in [26].

The optimal control problem, where the constraint is given by the convection-diffusion equation is considered in, for instance in [27, 28].

The paper is organized as follows. In Section 2 we define the discrete form of the optimization problem as in (5) and its compressed two-by-two block form. In Section 3 we define the preconditioner and analyse its properties. We include also some analysis of the spectrum of the related preconditioned matrix, based on the Generalized Locally Toepants (GLT) framework, Section 4. In Section 5 we present numerical examples to illustrate the performance of the method. Some concluding remarks are given in Section 6. Appendix A contains a detailed derivation of the involved GLT symbols.

2 Discrete algebraic form of the target optimization problem

To proceed to the numerical solution of minimizing (5) one might follow either the ‘optimize-then-discretize’ or the ‘discretize-then-optimize’ framework. When using the primal ‘optimize-then-discretize’ procedure one finds the necessary optimality conditions analytically and then discretizes those in order to solve them numerically. However, by adding equality or inequality
constraints makes this approach more complex and problem-dependent. Furthermore, in some cases the discretization of the optimality conditions demands the usage of specific techniques to guarantee sufficient accuracy and stability of the discrete solution, which might lead to solving numerically optimality conditions that are not equivalent with the analytically derived ones. Therefore 'discretize-then-optimize' is the relevant approach, which we follow in this work.

For simplicity we choose to work in two space dimensions and also assume that \( f = 0 \). To discretize the problem we use the Finite Element method (FEM) on triangular or quadrilateral meshes with piece-wise linear/bilinear basis functions. To refer to the discretized variables in vector form we use lower-case bold letters. As in many related studies, we assume also that all variables are discretized using the same finite element spaces, thus, the mass matrices, corresponding to \( u \) and \( p \) (and \( \lambda \), to appear in the sequel) are the same. This is utilized in order to compress the arising algebraic systems and save computations.

The discrete constraint equation becomes then \( K \dot{u} = M v + f \). Here \( K \) represents the second order stiffness matrix, \( M \) is the mass matrix and \( f \) contains the boundary conditions. To be able to easily handle the box constraints for the state and the control, we work with the lumped mass matrix, thus, in the sequel, \( M \) is diagonal.

To solve the optimization problem, we formulate the corresponding discrete Lagrangian functional \( L \) and construct the first order necessary KKT conditions. For notational simplicity we denote the arising system matrix as \( A \) even when we reduce its dimension.

**Case 1:** When the only constraint is the state equation, the discrete form of the Lagrange functional is

\[
L(u,v,p) = \frac{1}{2}(u-u_d)^T M (u-u_d) + \alpha \frac{1}{2} v^T M v + p^T (K u - M v)
\]

and the first order necessary conditions in matrix form read

\[
A \begin{bmatrix} u \\ v \\ p \end{bmatrix} = \begin{bmatrix} M & 0 & K^T \\ K & -M & 0 \\ 0 & \alpha M & -M \end{bmatrix} \begin{bmatrix} u \\ v \\ p \end{bmatrix} = \begin{bmatrix} M u_d \\ f \\ 0 \end{bmatrix}.
\]

From the third equation we express the control via the Lagrange multiplier, \( v = \frac{1}{\alpha} p \) and obtain the reduced system

\[
\begin{bmatrix} M & K^T \\ K & -\frac{1}{\alpha} M \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} M u_d \\ f \end{bmatrix}.
\]

In order to ensure that the computed state is as close as possible to the desired state \( u_d \), the regularization parameter \( 0 < \alpha < 1 \) is supposed to be small, typically of order \( 10^{-6} - 10^{-8} \). The inverse of it introduces a bad scaling in the matrix, which would make the solution of systems with it difficult for the iterative methods. Therefore, even before constructing a preconditioner, we weaken the bad scaling by multiplying the second equation by \( \sqrt{\alpha} \) and replacing \( p \) by \( -\sqrt{\alpha} p \). The system to be solved then becomes

\[
\begin{bmatrix} M & -\sqrt{\alpha} K^T \\ \sqrt{\alpha} K & M \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} M u_d \\ \sqrt{\alpha} f \end{bmatrix}.
\]
Case 2: When we have box constraints on the control, \( \underline{v} \leq v \leq \overline{v} \) understood pointwise, the discrete Lagrangian functional becomes

\[
L(u, v, p, \lambda_u, \lambda_v) = \frac{1}{2}(u - u_d)^T M (u - u_d) + \frac{3}{2} v^T M v + p^T (K u - M v) + \lambda_u M (v - \overline{v}) - \lambda_v M (v - \underline{v}).
\] (9)

Usually the multipliers \( \lambda_u \) and \( \lambda_v \) are unified as \( \lambda = \lambda_v - \lambda_u \) and, after discretization, the first order necessary conditions read as follows (e.g., [29]):

\[
\begin{align*}
(1^0) & \quad M u + K^T p = M y_d \\
(2^0) & \quad K u - M v = 0 \\
(3^0) & \quad \lambda M v - M p + M \lambda = 0 \\
(4^0) & \quad -\lambda + \max(0, \lambda + c(v - v_b)) + \min(0, \lambda + c(v - v_a)) = 0 \quad \text{for any constant } c > 0.
\end{align*}
\] (10)

The presence of conditional terms in the KKT system (10) results in a nonlinear problem with a non-smooth Jacobian. Further, relation (4\(^0\)) is understood pointwise, e.g., the conditions are checked in every individual element of \( v \). To facilitate writing the KKT conditions in matrix form we introduce the sets \( \mathcal{I}_a^{(e)} \) and \( \mathcal{I}_i^{(e)} \), denoting corresponding gly the points, where any of the box control constraints is violated (active points) and the rest of the points, where the control satisfies the constraints (inactive points). We introduce also the notations \( M_a^{(e)} \) and \( M_i^{(e)} \), denoting diagonal matrices, equal to \( M \) in the points \( \mathcal{I}_a^{(e)} \) and zero in \( \mathcal{I}_i^{(e)} \) for \( M_a^{(e)} \), and vice versa for \( M_i^{(e)} \).

With the above assumptions, conditions (10) in matrix form read

\[
\mathfrak{A} \begin{bmatrix} u \\ v \\ p \\ \lambda \end{bmatrix} = \begin{bmatrix} M & 0 & K^T & 0 \\ K & -M & 0 & 0 \\ 0 & \alpha M & -M & M \\ 0 & c M_a^{(e)} & 0 & -M_i^{(e)} \end{bmatrix} \begin{bmatrix} u \\ v \\ p \\ \lambda \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ 0 \\ b_3 \end{bmatrix}.
\] (11)

As it becomes clear in the derivations below, the matrix \( \mathfrak{A} \) can be compressed and the constant \( c \) can be eliminated from the system matrix. It plays a role implicitly in determining the active and inactive sets in the nonlinear process.

To include the \( L^1 \) norm constraint only relation (4\(^0\)) in the first order necessary conditions (10) has to be changed, cf. e.g., [2, 26].

\[
\begin{align*}
(1^0) & \quad M u + K^T p = M y_d \\
(2^0) & \quad K u - M v = 0 \\
(3^0) & \quad \alpha M v - M p + M \lambda = 0 \\
(4^1) & \quad v - \max\{0, v + c_1 (\lambda - \beta)\} - \min\{0, v + c_1 (\lambda + \beta)\} + \\
& \quad \max\{0, c_2 (v - v_b) + c_1 (\lambda - \beta)\} + \min\{0, c_2 (v - v_a) + c_1 (\lambda + \beta)\} = 0 \\
& \quad \text{for any constants } c_1 > 0, c_2 > 0.
\end{align*}
\] (12)

Thus, the conditions how to classify an entry from \( v \) into \( \mathcal{I}_a^{(e)} \) or \( \mathcal{I}_i^{(e)} \) change, however the structure of the KKT system remains the same, as in (11). The constants \( c_{1,2} \) do not appear in the compressed KKT-system yet they affect the non-linear iterations in determining the active sets. Some choices of \( c_{1,2} \) have been observed to lead to non-convergence for the Newton iteration, see e.g., [26].
Case 3: Discretizing the Lagrangian functional (5), we add to the KKT system the box constraints for the state variable \( u \). To do this, we analogously introduce two more sets \( \mathcal{I}_a^{(u)} \) and \( \mathcal{I}_i^{(u)} \), denoting correspondingly the points, where any of the state box constraints is violated (active points) and the rest of the points, where the state satisfies the constraints (inactive points). Similarly as for the control, we introduce also the notations \( M_a^{(u)} \) and \( M_i^{(u)} \), to be equal to \( M \) in the points \( \mathcal{I}_a^{(u)} \) and zero in \( \mathcal{I}_i^{(u)} \) for \( M_a^{(u)} \) and vice versa for \( M_i^{(u)} \). We note that in general \( \mathcal{I}_a^{(u)} \cap \mathcal{I}_i^{(v)} \neq \emptyset \).

The matrix form of (12) is as follows,

\[
\begin{bmatrix}
u \\
p \\
\lambda \\
\end{bmatrix} = \begin{bmatrix}
M + \gamma^{-1} M_a^{(u)} & K^T & 0 \\
K & -M & 0 \\
0 & \alpha M & -M \\
0 & cM_a^{(v)} & -M_i^{(v)} \\
\end{bmatrix} \begin{bmatrix}
u \\
p \\
\lambda \\
\end{bmatrix} = \begin{bmatrix}
b_1 \\
b_2 \\
b_3 \\
\end{bmatrix},
\]

where now the first part of the right hand side has a new form, \( b_1 = M u_d + \gamma^{-1} (M_a^{(u)} u_b + M_i^{(u)} u_a) \). Here \( M_a^{(u)} \) and \( M_i^{(u)} \) are parts of \( M \) corresponding to the current active sets, i.e., \( M_a^{(u)} = M_a^{(u)} + M_i^{(u)} \).

We next show how we can scale and compress the system (13). We perform the following steps:

- Eliminate \( v = \frac{1}{\alpha} (p - \lambda) \) from the third equation in (13) and obtain

\[
\begin{bmatrix}
u \\
p \\
\lambda \\
\end{bmatrix} = \begin{bmatrix}
M + \gamma^{-1} M_a^{(u)} & K^T & 0 \\
K & -\frac{1}{\alpha} M & \frac{1}{\alpha} M \\
0 & \frac{c}{\alpha} M_a^{(v)} & -\frac{c}{\alpha} M_a^{(v)} - M_i^{(v)} \\
\end{bmatrix} \begin{bmatrix}
u \\
p \\
\lambda \\
\end{bmatrix} = \begin{bmatrix}
b_1 \\
b_2 \\
b_3 \\
\end{bmatrix}.
\]

- Next multiply the second equation by \( \sqrt{\alpha} \), the third equation by \( \alpha \) and scale \( p = -\sqrt{\alpha} \hat{p} \).

The system becomes

\[
\begin{bmatrix}
u \\
p \\
\lambda \\
\end{bmatrix} = \begin{bmatrix}
M + \gamma^{-1} M_a^{(u)} & -\sqrt{\alpha} K^T & 0 \\
\sqrt{\alpha} K & -\alpha M & \frac{1}{\sqrt{\alpha}} M \\
0 & -c\sqrt{\alpha} M_a^{(v)} & -cM_a^{(v)} - \alpha M_i^{(v)} \\
\end{bmatrix} \begin{bmatrix}
u \\
p \\
\lambda \\
\end{bmatrix} = \begin{bmatrix}
b_1 \\
b_2 \\
b_3 \\
\end{bmatrix}.
\]

- Observe, that the Schur complement \( A \) of \( \hat{A} \) with respect to the top pivot two-by-two block can be computed exactly and is of the form

\[
A = \begin{bmatrix}
M + \gamma^{-1} M_a^{(u)} & -\sqrt{\alpha} K^T \\
\sqrt{\alpha} K & M_i^{(v)} \\
\end{bmatrix}.
\]

In detail,

\[
A = \begin{bmatrix}
M + \gamma^{-1} M_a^{(u)} & -\sqrt{\alpha} K^T \\
\sqrt{\alpha} K & M_i^{(v)} \\
\end{bmatrix} - \begin{bmatrix}
0 \\
\frac{1}{\sqrt{\alpha}} M \\
\end{bmatrix} \begin{bmatrix}
cM_a^{(v)} + \alpha M_i^{(v)} \\
0 \\
\end{bmatrix}^{-1} \begin{bmatrix}
0 \\
c\sqrt{\alpha} M_a^{(v)} \\
\end{bmatrix}
= \begin{bmatrix}
M + \gamma^{-1} M_a^{(u)} & -\sqrt{\alpha} K^T \\
\sqrt{\alpha} K & M_i^{(v)} \\
\end{bmatrix} + \begin{bmatrix}
0 & 0 \\
0 & W \\
\end{bmatrix}
\]

6
where

\[
W = -c\sqrt{\alpha} \frac{1}{\sqrt{\beta}} M(cM_a^{(v)} + \alpha M_i^{(v)})^{-1} M_a^{(v)}
= -c(cM_a^{(v)}M^{-1} + \alpha M_i^{(v)}M^{-1})^{-1} M_a^{(v)}
= -c(cM_a^{(v)} + \alpha f_i^{(v)})^{-1} M_a^{(v)}
= -M_a^{(v)}.
\]

The matrices \( I_a^{(v)} \) and \( I_i^{(v)} \) are diagonal with entries equal to one in the active and inactive points, correspondingly. In the derivation we use the fact that \( M_a^{(v)} + M_i^{(v)} = M \), \( M_a^{(v)}M_i^{(v)} = 0 \), \( M_i^{(v)}M^{-1} = I_a^{(v)} \), \( M_i^{(v)}M^{-1} = I_i^{(v)} \) and \( I_i^{(v)}M_a^{(v)} = 0 \). Thus, the matrix of the system to solve becomes now

\[
A = \begin{bmatrix}
M + \gamma^{-1} M_a^{(u)} & -\sqrt{\alpha} K T \\
n\alpha K & M_i^{(v)}
\end{bmatrix}.
\]

• Note that \( M + \gamma^{-1} M_a^{(u)} = (I + \gamma^{-1} I_a^{(u)})M \). Denote \( I_u = (I + \gamma^{-1} I_a^{(u)}) \). It is a positive definite diagonal matrix and we can use both its square root and inverse. We introduce \( \hat{u} = \mathbb{I}^{1/2} u \) and multiply the top block-row by \( \mathbb{I}^{-1/2} \) to obtain

\[
\begin{bmatrix}
M & -\sqrt{\alpha} \mathbb{I}^{-\frac{1}{2}} K T \\
\sqrt{\alpha} K \mathbb{I}^{-\frac{1}{2}} & M_i^{(v)}
\end{bmatrix}
\begin{bmatrix}
\hat{u} \\
\hat{p}
\end{bmatrix}
= \begin{bmatrix}
\mathbb{I}^{-\frac{1}{2}} M(u_d + b_u) \\
\sqrt{\alpha} M(f + b_v)
\end{bmatrix}.
\]

Denoting \( \hat{K} = \sqrt{\alpha} K \mathbb{I}^{-\frac{1}{2}} \) we obtain the form of the system, used in the sequel,

\[
A \begin{bmatrix}
\hat{u} \\
\hat{p}
\end{bmatrix}
= \begin{bmatrix}
M & -\hat{K} T \\
\hat{K} & M_i^{(v)}
\end{bmatrix}
\begin{bmatrix}
\hat{u} \\
\hat{p}
\end{bmatrix}
= \begin{bmatrix}
\mathbb{I}^{-\frac{1}{2}} M(y_d + b_u) \\
\sqrt{\alpha} M(f + b_v)
\end{bmatrix}.
\] (16)

We note that for the implementation of the action of \( A \) on a vector we do not have to form \( \hat{K} \) explicitly.

Thus, the main aim of this work reduces to solve systems with the matrix \( A \) in (16) in an efficient and robust way, ensuring fast convergence, independently or nearly independently from all problem (\( \epsilon \)), discretization (\( h \)) and method parameters (\( \alpha, \beta, \gamma \)). The system is nonlinear and the Jacobian is not differentiable in the classical sense. Therefore, we use the semismooth Newton method, see [30, 31, 1]. Furthermore, there are two characteristics of \( A \) that make some of the established and well-studied preconditioning techniques inefficient. Namely, earlier theoretical results on preconditioning matrices of the form (15) are not applicable here, because (i) the matrix \( K \) is nonsymmetric due to the convection-diffusion state equation and (ii) the block \( M_i^{(v)} \) is singular.
3 Preconditioning techniques

To simplify the notations further, denote $M_1 = M + \gamma^{-1} M_a^{(u)}$, $M_2 = M_i^{(v)}$ and note that $M_1$ is nonsingular. Also, unless stated otherwise, $K$ is used to denote the scaled stiffness matrix $\sqrt{\alpha} K$. The assumptions regarding the matrices are that $M$ and $M_1$ are symmetric and positive definite (spd), $K$ may or may not be symmetric and $\mathcal{N}(M) \cap \mathcal{N}(K) = \{0\}$.

We consider the matrix

$$A = \begin{bmatrix} M_1 & -K^T \\ K & M_2 \end{bmatrix},$$

where $K \neq K^T$, and seek a preconditioner of the form

$$P_{PSSB} = \begin{bmatrix} M_1 & -K^T \\ K + M_2 & K + M_2 \end{bmatrix},$$

(17)

where PSSB stands for Preconditioner for Square and Singular Blocks.

The preconditioner $P_{PSSB}$ possesses the following factorization,

$$P_{PSSB} = \begin{bmatrix} I & 0 \\ 0 & K_2 + M \end{bmatrix} \begin{bmatrix} M_1 & -K^T \\ I & I \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & K + M \end{bmatrix} \begin{bmatrix} I & -K^T \\ 0 & I \end{bmatrix} \begin{bmatrix} M_1 + K^T & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} I & 0 \\ I & I \end{bmatrix}.$$  

(18)

The computational steps to solve systems with $P_{PSSB}$ are given in Algorithm 1.

**Algorithm 1** Solution of a system with the PSSB preconditioner

1: Let $F_1 = M_1 + K^T$, $F_2 = M_2 + K$,

2: Solve $F_2 h = q$

3: Set $c = p + K^T h$

4: Solve $F_1 x = c$

5: Compute $y = h - x$

Before studying the spectral properties of the generalized eigenvalue problem

$$Az = \lambda Pz$$

(19)

we first give some details to relate $P_{PSSB}$ to another efficient preconditioner in earlier studies on the problem in focus, the PRESB preconditioner.

3.1 No additional constraints on $u$ and $v$

Imposing no box or sparsity constraints means $M_a^{(u)} = 0$, $M_i^{(v)} = M$ and we obtain the classical distributed control problem and the resulting matrix in compressed two-by-two block form (cf., e.g., [32])

$$A_0 = \begin{bmatrix} M & -K^T \\ K & M \end{bmatrix}.$$
The properties below hold for any spd matrix $M$. A preconditioner that is shown to have significant advantages over other techniques, in terms of convergence and computational complexity is the PRESB preconditioner (cf., e.g., [32, 33, 34, 25] and the references therein), defined as

$$P_{PRESB} = \begin{bmatrix} M & -K^T \\ K & M + K + K^T \end{bmatrix}, \text{ or equivalently, } P_{PRESB} = \begin{bmatrix} M + K + K^T & -K^T \\ K & M \end{bmatrix}. \quad (20)$$

It is shown that all eigenvalues of $P_{PRESB}^{-1}A_0$ are real and belong to the interval $[\frac{1}{2}, 1]$. Furthermore, the PRESB preconditioner possesses a convenient factorized form,

$$\begin{bmatrix} M & -K^T \\ K & M + K + K^T \end{bmatrix} = \begin{bmatrix} I & -I \\ 0 & I \end{bmatrix} \begin{bmatrix} M + K & 0 \\ K & M + K^T \end{bmatrix} \begin{bmatrix} I & I \\ 0 & I \end{bmatrix},$$

which means that the solution of a system with $P_{PRESB}$ requires to solve two half-sized systems with $M + K$ and $M + K^T$ and to perform one matrix multiplication with $K$.

For completeness we include Algorithm 2 to solve a system

$$\begin{bmatrix} M & -K^T \\ K & M + K + K^T \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} p \\ q \end{bmatrix}.$$  

Algorithm 2 Solution of a system with the PRESB preconditioner

1: Let $F_1 = M + K^T$, $F_2 = M + K$,
2: Solve $F_1h = p + q$
3: Solve $F_2y = q - \tilde{C}h$
4: Compute $x = h - y$

In [26] the preconditioner $P$ as in (17) is introduced, which in this case reads as follows

$$P_{PSSB}^0 = \begin{bmatrix} M & -K^T \\ K + M & K + M \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} M & -K^T \\ K & K + M \end{bmatrix}. \quad (21)$$

As the factorization in (21) suggests, the solution with $P_{PSSB}^0$ has nearly the same computational cost as $P_{PRESB}$. We show below that the spectrum of $(P_{PSSB}^0)^{-1}A_0$ is also real and has the same eigenvalue bounds as $P_{PRESB}^{-1}A_0$. To this end, consider

$$\begin{bmatrix} M & -K^T \\ K & M \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \lambda \begin{bmatrix} M & -K^T \\ K + M & K + M \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

and follow the well-known approach to simplify the generalized eigenvalue problem,

$$\mu \begin{bmatrix} M & -K^T \\ K & M \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ M & K \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, \text{ or, } \begin{cases} \mu(Mx - K^Ty) = 0 \\ \mu(Kx + My) = Mx + Ky \end{cases} \quad (22)$$

with $\mu = \frac{1}{\lambda} - 1$. From the first equation in (22) we see that for $x \neq M^{-1}K^Ty$ we have that $\mu = 0$ and $\lambda = 1$. For $\mu \neq 0$ and $x = M^{-1}K^Ty$ we obtain the relation

$$\mu(KM^{-1}K^T + M)y = (K + K^T)y.$$
Clearly, $\mu$ is real and positive, which shows that $\lambda < 1$. We next multiply by $M^{-\frac{1}{2}}$ from left and right, denote $\tilde{K} = M^{-\frac{1}{2}}KM^{-\frac{1}{2}}$ and see, that $\mu$ is an eigenvalue of

$$\mu(\tilde{K}\tilde{K}^T + I)y = (\tilde{K} + \tilde{K}^T)y.$$ 

Utilising the fact that the eigenvalues of a matrix and its transpose are the same, $\mu$ is of the form $\frac{2a}{1+a^2}$. Using the assumption that $K + K^T$ is positive definite, then $a$ is positive and the algebraic expression is bounded independently of the value of $a$, namely,

$$0 \leq \frac{2a}{1+a^2} \leq 1,$$ 

thus $\frac{1}{2} \leq \lambda \leq 1$.

We prove in this way the following result.

**Theorem 3.1.** Let $M$ and $K$ be real matrices of size $n$, $M$ be symmetric positive definite, $K + K^T$ be positive definite, $\mathcal{N}(M) \cap \mathcal{N}(K) = \{0\}$ and let

$$\mathcal{A}_0 = \begin{bmatrix} M & -K^T \\ K & M \end{bmatrix}, \quad \mathcal{P}^0_{PSSB} = \begin{bmatrix} M & -K^T \\ K+M & K+M \end{bmatrix}.$$ 

Denote by $\lambda_i, i = 1, \ldots, n$ all eigenvalues of the generalized eigenvalue problem $\mathcal{A}_0v = \lambda\mathcal{P}^0_{PSSB}v$. Then all $\lambda_i$ are real and lie in the interval $[\frac{1}{2}, 1]$.

**Remark 3.1.** We note another relation between $\mathcal{P}_{PRESB}$ and $\mathcal{P}^0_{PSSB}$, namely, there holds

$$\begin{bmatrix} I & 0 \\ -I & I \end{bmatrix}\mathcal{P}^0_{PSSB} = \begin{bmatrix} I & 0 \\ -I & I \end{bmatrix} \begin{bmatrix} M & -K^T \\ K+M & K+M \end{bmatrix} = \begin{bmatrix} M+K+K^T & -K^T \\ K & M \end{bmatrix} = \mathcal{P}_{PRESB}.$$ 

### 3.2 General case

Consider now in detail the general case (19), where $\hat{K} = \sqrt{\alpha K}T_u^{-\frac{1}{2}}$.

$$\begin{bmatrix} M & -\hat{K}^T \\ \hat{K} & M_2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \lambda \begin{bmatrix} M & -\hat{K}^T \\ \hat{K} + M_2 & \hat{K} + M_2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}. \quad \text{(23)}$$ 

Here, $\hat{K} \neq \hat{K}^T$, $M_2$ is singular, and $\lambda$, $x$ and $y$ may be complex. Clearly, $\lambda$ is not zero.

We transform (23) to the equivalent problem

$$\mu \begin{bmatrix} M & -\hat{K}^T \\ \hat{K} & M_2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ M_2 & \hat{K} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix},$$ 

or, \begin{align*}
\mu(Mx - \hat{K}^Ty) &= 0 \\
\mu(\hat{K}x + M_2y) &= M_2x + \hat{K}y
\end{align*} \quad \text{(24)}$$

where $\mu = \frac{1}{2} - 1$. From (24) we see straightforwardly, as for $\mathcal{P}^0_{PSSB}$, that for $x$ and $y$ such that $x \neq M^{-1}\hat{K}^Ty$ we have $\mu = 0$, thus, $\lambda = 1$. In other words, 1 is an eigenvalue of multiplicity at least $n$, the size of $M$ and $\hat{K}$.
Assume next that \( \mu \neq 0 \), i.e., \( \mathbf{x} = M^{-1} \hat{K}^T \mathbf{y} \) and substitute in the second equation in (24),

\[
\mu \left( \frac{M_2 + \hat{K}M^{-1} \hat{K}^T}{Q} \right) \mathbf{y} = \left( \frac{\hat{K} + M_2 M^{-1} \hat{K}^T}{Q} \right) \mathbf{y}.
\] (25)

In this way we obtain another generalized eigenvalue problem with real matrices, where \( S \) is spd and \( Q \) is nonsymmetric.

Recall that \( \mu = \frac{1}{\lambda} - 1 \). Let \( \mu = \mu_1 + i \mu_2 \) and \( \lambda = \lambda_1 + i \lambda_2 \). Then, a straightforward computation shows that \( \lambda_2 = \frac{\mu_2}{\mu_1 + 1} \) and then

\[
\lambda_1 = \begin{cases} 
1 & \mu_1 + 1 \\
\frac{\mu_1 + 1}{(\mu_1 + 1)^2 + \mu_2^2} & \end{cases} \quad \lambda_2 = \begin{cases} 
0 & \mu_1 + 1 \\
\frac{-\mu_2}{(\mu_1 + 1)^2 + \mu_2^2} & \end{cases}.
\] (26)

Below we attempt to estimate \( \mu_1 \) and \( \mu_2 \) for the matrices at hand.

- We see from (26) that \( |\lambda_2| \leq 1 \) for all values of \( \mu \).
- Let \( \mu_2 \neq 0 \). Then \( |\lambda_1| < 1 \) and \( |\lambda_2| < 1 \) for any value of \( \mu \), thus, \( |\lambda| < 1 \).
- Let \( \mu_2 = 0 \) which means that \( \lambda \) is real. The undesired cases are two:
  - for \( \mu_1 > 0 \) and large we see that \( \lambda \to \infty \).
  - for \( \mu_1 < 0 \) and \( |\mu_1| \to 1 \) then \( \lambda \to \infty \).

For further analysis of \( \mu_1 \) and \( \mu_2 \) one has to consider the generalized eigenvalue problem \( Q \mathbf{y} = \mu S \mathbf{y} \). The following lemma can be of use.

**Lemma 3.1** ([35]). Let \( X \in \mathbb{R}^{n,n} \), \( Y \in \mathbb{R}^{n,n} \), \( Y = Y^T \). Then for the eigenvalues \( \mu \) of the generalized eigenvalue problem \( X \mathbf{v} = \mu Y \mathbf{v} \), \( \mu \in \mathbb{C} \), \( \mathbf{v} \in \mathbb{C}^n \) it holds that

\[
v^H \frac{1}{2}(X + X^T) \mathbf{v} = \text{Re}(\mu)v^H \mathbf{v},
\] (27)

\[
v^H \frac{1}{2i}(X - X^T) \mathbf{v} = \text{Im}(\mu)v^H \mathbf{v}.
\] (28)

Here \( i \) denotes the complex unit, and \( \ast^H \) - the complex conjugate of \( \ast \).

In our case \( X = Q \) and \( Y = S \). We note, that \( M_1 \) and \( M_2 \) are diagonal and they commute.

We obtain

\[
Q^+ = \frac{1}{2}(Q + Q^T) = \frac{1}{2}(\hat{K}(I + M^{-1}M_2) + (I + M_2M^{-1})\hat{K}^T)
\]

\[
= \frac{1}{2}(\hat{K}W^+ + W^+ \hat{K}^T).
\]

Analogously,

\[
Q^- = \frac{1}{2}(Q - Q^T) = \frac{1}{2}(M_2M^{-1}\hat{K}^T + \hat{K} - \hat{K}M^{-1}M_2 - \hat{K}^T)
\]

\[
= \frac{1}{2}(\hat{K}W^- - W^- \hat{K}^T),
\]

\[
\]
The target numerical problem involves five parameters. The problem parameter $\epsilon$ determines how singularly perturbed is the given convection-diffusion problem. The method parameters are the regularization variables $\alpha$, $\beta$ and $\gamma$ and $h$ is the discretization parameter. As is shown below, the convergence of the PSSB-preconditioned compressed KKT system depends explicitly on $\epsilon, \alpha, \gamma, h$ and implicitly on $\beta$, which determines the number of active/inactive points for the additional state and control box- and sparsity constraints. Due to the complexity of the problem and the existing mutual parameter dependencies, a convergence study cannot be done by letting these parameters individually approach their limit values. There are several considerations to take into account.

1. Since no stabilization is included in the discretization, $h$ should be chosen small enough with respect to $\epsilon$. For the tests in Figure 1 we let $2h = \epsilon$.

2. In [9, 25] it has been shown that when we impose box constraints on the state using the Moreau-Yosida approach, keeping $h$ fixed and decreasing $\gamma$ leads to a stagnation of the achieved error. In [9] some optimal relations between $h$ and $\gamma$ have been discussed, of the type $\gamma \approx h^2$ in 2D.

Interplay between problem, discretization and method parameters

The target numerical problem involves five parameters. The problem parameter $\epsilon$ determines how singularly perturbed is the given convection-diffusion problem. The method parameters are the regularization variables $\alpha$, $\beta$ and $\gamma$ and $h$ is the discretization parameter. As is shown below, the convergence of the PSSB-preconditioned compressed KKT system depends explicitly on $\epsilon, \alpha, \gamma, h$ and implicitly on $\beta$, which determines the number of active/inactive points for the additional state and control box- and sparsity constraints. Due to the complexity of the problem and the existing mutual parameter dependencies, a convergence study cannot be done by letting these parameters individually approach their limit values. There are several considerations to take into account.

1. Since no stabilization is included in the discretization, $h$ should be chosen small enough with respect to $\epsilon$. For the tests in Figure 1 we let $2h = \epsilon$.

2. In [9, 25] it has been shown that when we impose box constraints on the state using the Moreau-Yosida approach, keeping $h$ fixed and decreasing $\gamma$ leads to a stagnation of the achieved error. In [9] some optimal relations between $h$ and $\gamma$ have been discussed, of the type $\gamma \approx h^2$ in 2D.
3. A certain relation between $\epsilon$ and $\alpha$ can also be foreseen. To illustrate this interplay, consider two mathematically equivalent forms of the state equation,

$$-\epsilon \Delta u + b \cdot \nabla u = v_1 \quad \text{in } \Omega$$  
(29)

$$-\Delta u + \frac{1}{\epsilon} b \cdot \nabla u = v_2 \quad \text{in } \Omega$$  
(30)

We can regard (30) to be a dimensionless form of (29).

Then the corresponding optimization problem becomes

**Minimize:** $J_1(u,v) = \frac{1}{2} ||u-u_d||_{L^2(\Omega)}^2 + \frac{\alpha}{2} ||v_1||_{L^2(\Omega)}^2$ subject to (29),  
(31)

**Minimize:** $J_2(u,v) = \frac{1}{2} ||u_2-u_d||_{L^2(\Omega)}^2 + \frac{\alpha}{2} ||v_2||_{L^2(\Omega)}^2$ subject to (30).  
(32)

The second form is obtained by multiplying the right side of the constraining PDE by $1/\epsilon$ while leaving the objective functional unchanged. For $\epsilon \neq 1$, since the second formulation does not take the scaling into account in the target functional, the solutions to the optimization problems will differ. We can modify the target functional to take the scaling into account,

**Minimize:** $J_2^{(m)}(u,v) = \frac{1}{2} ||u_2-u_d||_{L^2(\Omega)}^2 + \frac{\alpha \epsilon^2}{2} ||v_2||_{L^2(\Omega)}^2$ subject to (30)  
(33)

In (33) the minimum of $J_2^{(m)}$ relates to the minimum of $J_1$ as $u_1 = u_2$ and $v_2 = \frac{1}{\epsilon} v_1$. That is to say, we have equivalence up to a scaling on the control for the optimization problem and given the solution to the either formulation we can trivially reconstruct the other. In this sense the solutions to both formulations contain the same information and this observation is useful when predicting solver performance. The form (33) also indicates that for small values of $\epsilon$ we have an effective $L_2$-stabilization parameter, defined as

$$\alpha_{\text{eff}} := \epsilon^2 \alpha$$

suggesting that when $\epsilon$ is small, we do not have to choose $\alpha$ very small to ensure good quality of the state. The equivalence of the two settings is illustrated in Table 2. Here we considered only the $L_2$-term yet there are analogous relations for the other control parameters; $\beta_2 = \epsilon \beta_1$ and the control box-constraints are subject to scaling $\epsilon v_a^{(1)} = v_a^{(2)}$.

As the convergence properties of the method depend primarily on the $L_2$-regularization it warrants special attention in a separate study.

4 GLT as a guideline to estimate spectrum bounds

In [37] GLT is successfully used to estimate the spectrum of a nonsymmetric Schur complement matrix. Below we attempt to use the technique again, to estimate lower and upper bounds of the spectrum of $v^H \frac{1}{2} (Q^+) v = \text{Re}(\mu)v^H S v$. Similar analysis can be done for
\(v^H \frac{1}{2}(Q^-)v = Im(\mu)v^H S v\), however, in this work we omit that and rely on (26) and the fact that \(|\lambda_2| \leq 1\) in all cases.

The theoretical justification to apply GLT in our case is given in [38, 39, 40, 41]. In order to apply GLT, a number of conditions must hold. In the first place, the matrices have to be locally Toeplitz (cf. [38]). This property holds for \(\hat{K}\). It should be noted, see also the derivations in Appendix A, that we use the GLT machinery in its (almost) full generality since the symbol, derived below, is matrix-valued and multivariate (two variables in the present context).

The lumped matrix \(M\) is not constant diagonal and cannot easily be viewed as a equispaced sampling of a piece-wise continuous function over a piece-wise smooth domain. (We mention that, as one can find in [38, 39], the most general assumption, in our setting, is that the symbol is a Riemann integrable function over a Peano-Jordan measurable set, which coincides with those Lebesgue measurable sets whose boundary has zero Lebesgue measure.) Although the GLT theory handles matrices with varying diagonals, we simplify the analysis and consider \(M = h^2 I\).

The matrix \(M_2\), in addition having some entries zero, cannot be seen as locally Toeplitz either. For the analysis we assume that \(M_2 = 0\), thus, all points of the control are inactive.

It is very important to state that the GLT theory is asymptotic and considers a sequence of matrices or given algebraic operations on matrix-sequences, associated with \(h \to 0\). Thus, GLT is by no means applicable to a single matrix. However, as it becomes evident from the numerical illustrations in Figure 1 and in the appendix, the property of the GLT symbol to qualitatively describe the spectrum of matrix expressions is observed already for reasonably small values of \(h\).

A hindrance to fully rely on GLT for obtaining spectrum bounds should be mentioned. It is known that due to the locally Toeplitz character of the matrices, the so-called symbol describes the spectrum of the related matrix sequence up to a finite number of outliers, that are mostly due to boundary conditions and discretization features. We need to bound the smallest and the largest eigenvalue of the above generalized eigenvalue problems. However, if the extreme eigenvalues we want to estimate are not in the range of the symbol but are among the outliers, then the GLT theory cannot straightforwardly be used to quantify the outliers. Nevertheless, we apply GLT to support the numerical evidence that the real part of \(\lambda\) is bounded away from zero from below and does not grow much for a wide range of the parameters \(\epsilon, \alpha, \beta, \gamma\) and \(h\).

To derive the symbol for \(S\) and \(Q^+\) we discretize the problem using a square mesh and bilinear FEM basis functions. For the analysis we do not include any stabilization of the convection-diffusion equation.

We next construct the symbols for \(\hat{K}, W, M_1\) and \(M_2\), named \(f^K, f^W, f^{M_1}\) and \(f^{M_2}\). The detailed derivations are given in Appendix A. The symbol of \(\hat{K}\) is as follows,

\[
f^K = \frac{1}{3} \sqrt{\alpha} \frac{\sqrt{\gamma}}{\sqrt{1+\gamma}} \left[ \epsilon \left( 8 - 2\cos(\theta_1) - 2\cos(\theta_2)(1+2\cos(\theta_1)) \right) 
- ih \left( b_1 \sin(\theta_1)(2+\cos(\theta_2)) + b_2 \sin(\theta_2)(2+\cos(\theta_1)) \right) \right] 
= \sqrt{\epsilon} \frac{\sqrt{\gamma}}{\sqrt{1+\gamma}} (\epsilon g_1 - ih g_2), \tag{34}
\]
(a) $\epsilon = 0.0625, \alpha = 0.001, \gamma \text{ inactive}$

(b) $\epsilon = 0.0625, \alpha = 0.001, \gamma = 0.05$

(c) $\epsilon = 0.03125, \alpha = 0.001, \gamma \text{ inactive}$

(d) $\epsilon = 0.03125, \alpha = 0.001, \gamma = 0.05$

Figure 1: The spectrum and the symbol of $S^{-1}Q^+$ for various values of $\epsilon$ and $\gamma$

where for convenience we denote

$$g_1(\theta_1, \theta_2) = \frac{1}{3}[8 - 2\cos(\theta_1) - 2\cos(\theta_2)(1 + 2\cos(\theta_1))]$$

$$g_2(\theta_1, \theta_2) = \frac{1}{3}[b_1\sin(\theta_1)(2 + \cos(\theta_2)) + b_2\sin(\theta_2)(2 + \cos(\theta_1))].$$

We choose $M = h^2 I$ and $M_2 = 0$ the latter meaning that all points for the control are active. Thus,

$$f^M = h^2, \quad f^{M_2} = 0, \quad f^{W^+} = 1, \quad f^{W^-} = 0.$$  \hspace{1cm} (35)

Since the GLT matrices form an algebra, the symbols of $S$ and $Q^+$ are then easily constructed using the same algebraic operations as to form the matrices.

$$f^S = f^{M_2} + h^{-2} f^K f^K = h^{-2} \frac{\alpha \gamma}{1 + \gamma} \left( \epsilon^2 g_1^2 + h^2 g_2^2 \right)$$  \hspace{1cm} (36)
with \( \text{f}^\text{K} \) being the complex conjugate of \( f^K \). Analogously we find

\[
\text{f}^{Q^+} = \frac{1}{2}(f^K + f^\text{K}) = \frac{\sqrt{\alpha} \sqrt{\gamma}}{\sqrt{1 + \gamma}} \epsilon g_1.
\]  

(37)

Based on the above, the symbol of \( S^{-1}Q^+ \) is found to be

\[
\text{f}^{S^{-1}Q^+} = \frac{3h^2 \sqrt{1 + \gamma} \epsilon g_1}{\sqrt{\alpha} \sqrt{\gamma} (\epsilon^2 g_1^2 + h^2 g_2^2)}.
\]  

(38)

Consider \( f^{S^{-1}Q^+} \), which determines the real part of \( \mu \). If it approaches 0 then \( \lambda \) approaches 1, which is favourable. We see from (38) that if all parameters but \( h \) are fixed and we let \( h \to 0 \), then \( \mu_1 \to 0 \) and \( \lambda_1 \to 1 \). Let now allow the other parameters to vary as well. To see when \( f^{S^{-1}Q^+} \) increases unboundedly, we can analyse \( f^{(Q^+)^{-1}S} \) and check when it goes to zero.

With the assumptions that \( h = O(\epsilon) \) and \( \gamma = O(h^2) \) we obtain

\[
f^{(Q^+)^{-1}S} = C \frac{\sqrt{\alpha} \sqrt{\gamma} (\epsilon^2 g_1^2 + h^2 g_2^2)}{h^2 \sqrt{1 + \gamma} \epsilon g_1} \approx C \frac{\sqrt{\alpha} \sqrt{\gamma} \epsilon g_1}{h^2} + C \frac{\sqrt{\alpha} \sqrt{\gamma} g_2^2}{\epsilon g_1} = O(\sqrt{\alpha}).
\]

Thus, \( \alpha \to 0 \) is the most important factor to deteriorate the convergence of the PSSB-preconditioned system. This is confirmed by the numerical experiments in the figures in Section 5.1.

In Figure 1 we show the sampled symbols \( f^{S^{-1}Q^+} \) and the eigenvalues of the corresponding matrices for various values of the parameters \( \alpha, \gamma \) and \( h \). Figure 1 shows that the upper part of the spectrum is well described by the symbol. We also see that in the lower part of the spectrum there are a number of outliers, that are not in the range of the symbol and cannot be described by that. Although we cannot quantify the lower bound of \( \mu_1 \) we provide extensive numerical tests, showing that for a broad interval of parameter values of practical interest, the spectrum of the PSSB-preconditioned matrix stays away from zero and its upper bound behaves as \( O(1) \).

5 Numerical illustrations

We illustrate the performance of the PSSB preconditioner for various values of the parameters \( h, \alpha, \beta \) and \( \gamma \).

Discretization

The optimal control problem is discretized using a triangular mesh with right-angled isosceles triangles and linear Finite Element basis functions. In this study, in order to avoid excessive complications in the analysis of PSSB, we do not consider any stabilization in the discrete convection-diffusion problem. Numerical tests, not included here, show that including a stabilization does not influence the performance of the preconditioner for the problems documented in the tables and figures below.
For the symbol computations, to fit in the GLT framework, we use a square mesh and bilinear FE basis functions. We note that although there is no algebraic relation between the matrices arising from triangular and quadrilateral meshes, the results from the GLT analysis are representative for the latter case, confirmed by the numerical results included in this paper. In [42], for the same class of problems, FEM discretizations on quadrilateral meshes have been used, showing analogous numerical performance of the preconditioned iterative solvers.

Stopping criteria and solver details
The solution procedure consists of a nonlinear solver, the semismooth Newton method. The KKT system is solved until the number of the points in the active sets is minimized (stops changing). To mitigate cases when the problem has converged and the active set changes with only very few points that are counted active or inactive due to numerical fluctuations of the values the current state and control, we check the change of the state variable. If all changes leading to a set change are less than the discretization error $h^2$, we consider the solution to be converged.

As is well known, the convergence of the Newton-type of methods depends on the quality of the initial guess. The numerical experiments are performed on a sequence of nested meshes and the initial guess for the semismooth Newton method on each next finer mesh is the prolonged solution obtained on the previous coarse mesh. Similarly, the initial guess for the linear solver is carried forward between Newton iterations and between meshes. The degrees of freedom in the discrete constraining equation correspond to $h = 2^{-k}$, namely, 25, $k=2$; 81, $k=3$; 289, $k=4$; 1089, $k=5$;... 263169, $k=9$ and 1050625, $k=10$. We start the computations from the mesh with $k=2$ but in the tables we include only results for meshes from $k \geq 5$.

The linear system with the Jacobian and the two systems in the preconditioner are solved iteratively with a stopping criterion chosen as the relative preconditioned residual $\|P^{-1}(Ax-b)\|/\|P^{-1}b\| < 10^{-6}$ for the outer solver. The stopping criterion for the inner solvers with $M_i^{(0)}+\sqrt{\alpha}\hat{K}$ (denoted AGMG1) and $M+\sqrt{\alpha}\hat{K}^T$ (denoted AGMG2) is relative $10^{-2}$. In order to avoid explicitly computing $\hat{K}$ at each non-linear step, instead of solving the systems with $\sqrt{\alpha}Ku_1+Mv_i$, AGMG1 solves with $\sqrt{\alpha}Ku_1+Mv_i$, and similarly for AGMG2. The constants $c_1$ and $c_2$ in (12) are chosen to be $c_1=c_2=1/\alpha$ in all tests.

Implementation
All experiments are performed on a Dell Latitude 7490 laptop with an Intel® Core™ i7-8650U CPU and 16 GB of RAM. The generation and refinement of the meshes and the generation of the FEM matrices are done in FEniCS ([43]). The outer linear solver is the PSSB-preconditioned generalized Conjugate Residual (GCR) method, cf. [44]. As an inner solver for the block systems we use the AGMG-preconditioned GCR ([45, 46, 47, 48]). The numerical tests are performed in Julia ([49]) and the GLT-related computations are performed and plotted in Matlab. We provide elapsed run time for the solvers on each mesh level (i.e., excluding mesh operations and FEM assembly). Details on the computations and the sampling of the symbols are outlined in Appendix A.
We minimize the cost functional \( J(u,v) \) as in (5), subject to (2) in two space dimensions using the test problems below. The problem domain is the unit square \([0,1]^2\).

**Problem P1.** Consider the optimal control problem with \( u_d = -\exp(|x_1 - 0.5| + |x_2 - 0.5|) \) and the following rotational convection field

\[
b = \begin{bmatrix} 2(2x_2 - 1)(1 - (2x_1 - 1)(2x_1 - 1)) \\ -2(2x_1 - 1)(1 - (2x_2 - 1)(2x_2 - 1)) \end{bmatrix}.
\]

We assume Dirichlet boundary conditions along the entire boundary with values defined by the desired state.

**Problem P2.** Let the desired state be \( u_d = \sin(2\pi x_1)\sin(2\pi x_2) \), the convection field be defined as \( b = [1 \ 0]^T \). We assume homogeneous Neumann boundary condition at \( x_1 = 1 \) and Dirichlet boundary conditions on the rest of the boundary, defined by the desired state.

**Comparison between PSSB and PRESB**

We consider Problem P1 without non-linear constraints, i.e., the standard distributed control problem (1). Iteration counts for the two preconditioners \( P_{PRESB} \) and \( P_{PSSB} \) is is detailed in Table 1.

<table>
<thead>
<tr>
<th>( h^{-n} )</th>
<th>DoF</th>
<th>( P_{PSSB} ) Lin. AGMG1/2 Time</th>
<th>Iter.</th>
<th>Sec.</th>
<th>( P_{PRESB} ) Lin. AGMG1/2 Time</th>
<th>Iter.</th>
<th>Sec.</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>4225</td>
<td>5</td>
<td>4/3</td>
<td>0.04</td>
<td>6</td>
<td>2/2</td>
<td>0.05</td>
</tr>
<tr>
<td>7</td>
<td>16641</td>
<td>5</td>
<td>3/3</td>
<td>0.3</td>
<td>6</td>
<td>3/3</td>
<td>0.3</td>
</tr>
<tr>
<td>8</td>
<td>66049</td>
<td>4</td>
<td>3/3</td>
<td>1.0</td>
<td>6</td>
<td>3/3</td>
<td>1.2</td>
</tr>
<tr>
<td>9</td>
<td>263169</td>
<td>2</td>
<td>2/2</td>
<td>2.1</td>
<td>5</td>
<td>3/3</td>
<td>3.8</td>
</tr>
<tr>
<td>10</td>
<td>1050625</td>
<td>2</td>
<td>2/2</td>
<td>8.8</td>
<td>3</td>
<td>3/3</td>
<td>10.3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( h^{-n} )</th>
<th>DoF</th>
<th>( P_{PSSB} ) Lin. AGMG1/2 Time</th>
<th>Iter.</th>
<th>Sec.</th>
<th>( P_{PRESB} ) Lin. AGMG1/2 Time</th>
<th>Iter.</th>
<th>Sec.</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>4225</td>
<td>4</td>
<td>1/1</td>
<td>0.01</td>
<td>3</td>
<td>1/1</td>
<td>0.01</td>
</tr>
<tr>
<td>7</td>
<td>16641</td>
<td>4</td>
<td>2/2</td>
<td>0.1</td>
<td>4</td>
<td>2/2</td>
<td>0.1</td>
</tr>
<tr>
<td>8</td>
<td>66049</td>
<td>4</td>
<td>2/2</td>
<td>0.5</td>
<td>5</td>
<td>2/2</td>
<td>0.6</td>
</tr>
<tr>
<td>9</td>
<td>263169</td>
<td>3</td>
<td>2/2</td>
<td>2.0</td>
<td>5</td>
<td>2/2</td>
<td>2.6</td>
</tr>
<tr>
<td>10</td>
<td>1050625</td>
<td>2</td>
<td>2/2</td>
<td>7.8</td>
<td>4</td>
<td>2/2</td>
<td>10.0</td>
</tr>
</tbody>
</table>

Table 1: Problem P1, without non-linear constraints on \( u \) and \( v \)

From Table 1 we see that \( P_{PRESB} \) and \( P_{PSSB} \) perform very similarly although the iteration counts are not identical.
Performance of the equivalent forms of the state equation (29) and (30)

Consider Problem P1 and denote the vector field as

\[ \mathbf{b} = c \begin{bmatrix} 2(2x_2-1)(1-(2x_1-1)(2x_1-1)) \\ -2(2x_1-1)(1-(2x_2-1)(2x_2-1)) \end{bmatrix} \]

thus, \(c=1\) corresponds to the standard formulation of the state equation as in (29) and \(c=1/\epsilon\) corresponds to the scaled formulation (30).

We consider three pairs of parameter choices where the parameters are chosen such that the problems pairwise correspond to each other in the sense outlined above. The iteration counts are listed in Table 2. We see that the iteration counts are identical across all pairs and the elapsed time for the solvers are comparable. The obtained solutions are also equivalent up to a scaling of the control. In all remaining tests we use the standard formulation.

<table>
<thead>
<tr>
<th>(2^{-n})</th>
<th>(\epsilon)</th>
<th>(\alpha)</th>
<th>(\beta)</th>
<th>(\text{iter.})</th>
<th>(\text{iter.})</th>
<th>(\text{iter.})</th>
<th>(\text{Time})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>6</td>
<td>4/4</td>
<td>0.2</td>
<td>2</td>
<td>6</td>
<td>4/4</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>6</td>
<td>4/4</td>
<td>1.0</td>
<td>2</td>
<td>6</td>
<td>4/4</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>6</td>
<td>4/4</td>
<td>2.7</td>
<td>2</td>
<td>6</td>
<td>4/4</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>5</td>
<td>4/4</td>
<td>8.2</td>
<td>2</td>
<td>5</td>
<td>4/4</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>4</td>
<td>4/4</td>
<td>32.7</td>
<td>2</td>
<td>4</td>
<td>4/4</td>
</tr>
</tbody>
</table>

| Normalized | | | | | | | | |
| 6 | 2 | 8 | 4/3 | 0.3 | 2 | 8 | 4/3 | 0.2 |
| 7 | 2 | 8 | 4/4 | 1.2 | 2 | 8 | 4/4 | 1.1 |
| 8 | 2 | 8 | 4/4 | 3.4 | 2 | 8 | 4/4 | 3.3 |
| 9 | 2 | 6 | 4/4 | 10.4 | 2 | 6 | 4/4 | 10.2 |
| 10 | 2 | 6 | 4/4 | 39.8 | 2 | 6 | 4/4 | 39.5 |

<table>
<thead>
<tr>
<th>(\epsilon)</th>
<th>(\alpha)</th>
<th>(\beta)</th>
<th>(\text{iter.})</th>
<th>(\text{iter.})</th>
<th>(\text{iter.})</th>
<th>(\text{Time})</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\epsilon=0.1, c=1, \alpha=10^{-3}, \beta=10^{-3})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>7</td>
<td>5/4</td>
<td>0.8</td>
<td>3</td>
<td>17</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>15</td>
<td>7/5</td>
<td>2.3</td>
<td>2</td>
<td>15</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>14</td>
<td>9/4</td>
<td>6.8</td>
<td>2</td>
<td>14</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>14</td>
<td>7/4</td>
<td>20.6</td>
<td>2</td>
<td>14</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>14</td>
<td>7/5</td>
<td>138.1</td>
<td>3</td>
<td>14</td>
</tr>
</tbody>
</table>

Table 2: Problem P1: Standard and scaled formulations

PSSB performance with all constraints active

To illustrate the performance of the PSSB preconditioner in the case when all additional constraints are imposed, we consider Problem P1. When \(\epsilon=0.1\) we impose control box constraints
as \( v \in [5, -2.5] \) and state-box-constraints \( u \leq -1.3 \) and \( \beta = 10^{-4} \). As in [26] \( \gamma = \alpha^{1/4} \). When \( \epsilon = 0.01 \) we impose the control box constraints as \( v \in [-2.2] \) and \( \beta = 10^{-3} \). The iteration counts are displayed in Table 3 and plots of the control and state are shown in Figures 2 and 3. We plot the eigenvalues of the preconditioned linear systems for one discretization in Figure 4.

<table>
<thead>
<tr>
<th>h ( 2^{-n} )</th>
<th>Problem P1: ( \epsilon = 0.1 )</th>
<th>Problem P1: ( \epsilon = 0.01 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( \alpha = 10^{-3} ); ( \alpha_{ef} = 10^{-5} )</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>( \alpha = 10^{-4} ); ( \alpha_{ef} = 10^{-6} )</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>11</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>( \alpha = 10^{-5} ); ( \alpha_{ef} = 10^{-7} )</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>24</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>21</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>24</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>27</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>29</td>
</tr>
</tbody>
</table>

Table 3: Problem P1: all constraints active

The performance of the PSSB preconditioner without state-constraints is shown in Table 4 and Figures 5 and 6.

PSSB performance with all constraints active: alternate field

We consider Problem P2 with state-box-constraints \(-0.5 \leq u \leq 0.5\) and \( \beta = 10^{-3} \). When \( \epsilon = 0.1 \) we have control box constraints \( v \in [10, -10] \). When \( \epsilon = 0.01 \) we have control box constraints \( v \in [3, -3] \). We fix \( \gamma = 0.05 \) for both \( \epsilon \) values. Iteration counts are listed in Table 5. Plots of the obtained control and state are shown in Figures 7 and 8, plots of the eigenvalues of the preconditioned systems are shown in Figures 9 and 10.

Assessment of the results of the numerical tests

Given the additional constraints, imposed on the state and the control, combined with a convection-diffusion equation as a general state constraint, the target problem we study in this
work poses significant level of difficulty to solve the arising algebraic systems, both nonlinear and linear.

For all experiments, the nonlinear iterations have been very few, due to two factors. On one side, the solution on a previous coarser discretization mesh provides a good initial guess for the nonlinear iteration on the next finer mesh as well as for the related (outer) linear iterations. This approach is not new and has been utilized, cf., e.g., [25] and the references therein. The second factor is the efficient PSSB preconditioner, which allows in a relatively few iterations to achieve sufficient accuracy of the linear solver with the related Jacobian system, so that the performance of the nonlinear iterations is not degraded.

As seen in the results presented in Tables 2, 3, 4, 5, the performance of the PSSB precon-
Figure 4: Problem P1 with state constraints: Eigenvalues of the preconditioned system $\alpha = 10^{-3}$, $\epsilon = 0.01$

Figure 5: Problem P1 without state constraints: control and state on the final mesh for $\alpha = 10^{-6}$, $\epsilon = 0.1$, desired state is plotted in red

Figure 6: Problem P1 without state constraints: Eigenvalues of the preconditioned system $\alpha = 10^{-3}$, $\epsilon = 0.01$
Table 4: Problem P1: without box constraints for the state

<table>
<thead>
<tr>
<th>Problem P1: $\epsilon = 0.1$</th>
<th>Problem P1: $\epsilon = 0.01$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h$</td>
<td>$2^{-n}$</td>
</tr>
<tr>
<td>$\alpha = 10^{-3}$; $\alpha_{eff} = 10^{-5}$</td>
<td>$\alpha = 10^{-1}$; $\alpha_{eff} = 10^{-5}$</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
</tr>
</tbody>
</table>

ditioner is very satisfactory across problem sizes, exhibiting nearly independent convergence with respect to $h$. The outer convergence is in general affected by the value of the regularization parameters $\alpha$, $\gamma$ and $\epsilon$. However, as commented in Section 3, provided that these parameters cannot be varied independently of each other, the convergence of the linear solver remains fast and well-controlled for a broad range of parameter values of practical interest.

As PSSB requires solutions of inner systems with some block matrices, we show that these can be solved very efficiently using an algebraic multigrid preconditioner. We also show that the relative stopping criterion of the inner systems can be chosen relatively large, $10^{-2}$, which speeds up the solution with PSSB without sacrificing the convergence of the outer iterative solution method.
Table 5: Problem P2: all constraints active

<table>
<thead>
<tr>
<th>$h = 2^{-n}$</th>
<th>Problem P2: $\epsilon = 0.1$</th>
<th>Problem P2: $\epsilon = 0.01$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Nonlin. Av. lin. AGMG 1/2 Time</td>
<td>Nonlin. Av. lin. AGMG 1/2 Time</td>
</tr>
<tr>
<td>$\alpha = 10^{-3}$; $\alpha_{eff} = 10^{-5}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>14</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>15</td>
</tr>
<tr>
<td>$\alpha = 10^{-4}$; $\alpha_{eff} = 10^{-6}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>11</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>11</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>13</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>15</td>
</tr>
<tr>
<td>$\alpha = 10^{-5}$; $\alpha_{eff} = 10^{-7}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>17</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>16</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>17</td>
</tr>
</tbody>
</table>

Figure 7: Control and state on the final mesh for $\alpha = 10^{-3}$, $\epsilon = 0.01$, desired state is plotted in red
Figure 8: Control and state on the final mesh for $\alpha = 10^{-5}$, $\epsilon = 0.1$, desired state is plotted in red.

Figure 9: Problem P2 with state constraints: Eigenvalues of the preconditioned system $\alpha = 10^{-3}$, $\epsilon = 0.01$.

Figure 10: Problem P2 with state constraints: Eigenvalues of the preconditioned system $\alpha = 10^{-5}$, $\epsilon = 0.1$.
5.1 Examples of unfavourable spectrum of the PSSB-preconditioned matrix

In Tables 3-5 we observe some increase in the number of linear iterations when $\alpha$ decreases, which is also aligned with the analysis of the spectrum of the preconditioned matrix. However, the increase in the iteration counts for the outer linear solver is not significant. To illustrate some unfavourable cases of this effect, we consider Problem P2 with $\alpha = 10^{-5}$ but we decrease $\epsilon$ from 0.1 to 0.001 and plot the eigenvalues of the preconditioned system in Figure 11. For the case with 4225 degrees of freedom the smallest eigenvalue on the real line is $\approx 0.025$.

![Figure 11: Problem P2 with state constraints: Eigenvalues of the preconditioned system, $\alpha = 10^{-5}$, $\epsilon = 0.001$.](image)

For the same problem we now decrease $\epsilon$ further and choose $\epsilon = 10^{-4}$. The spectrum plot is shown in Figure 12.

![Figure 12: Problem P2 with state constraints: Eigenvalues of the preconditioned system $\alpha = 10^{-5}$, $\epsilon = 10^{-4}$.](image)

Another case, disadvantageous for the PSSB preconditioner, would be a small value of $\gamma$. To illustrate, we revert to $\epsilon = 0.1$ and choose $\gamma = 5 \cdot 10^{-5}$. The eigenvalue plot is depicted in Figure
13. As we see in Figures 11, 12 and 13, the imaginary parts are always by modulus bounded

Figure 13: Problem P2 with state constraints: Eigenvalues of the preconditioned system, $\alpha = 10^{-5}$, $\epsilon = 0.1$ and $\gamma = 5 \cdot 10^{-5}$

by one, the largest eigenvalue stays close to one, while the smallest eigenvalue approaches zero. However, the choices of the parameters disobey the mutual interplay dependencies, discussed in Section 3.2 and would hardly occur when solving problems of practical interest.
6 Conclusions

In this work we consider the numerical solution of discrete optimal control problems, constrained by the convection-diffusion equation with three additional constraints, namely, two-sided constraints on the state and the control and required sparsity for the control. The algebraic problems arising from the KKT system are large-scale and nonlinear. The associated Jacobian matrices are of two-by-two block form with nonsymmetric off-diagonal blocks and one of the diagonal blocks is singular. We consider the solution of the related linear systems by Krylov subspace iteration methods. We derive a particular two-by-two block preconditioner, PSSB, and analyse its properties and performance as a function of problem, discretization and three regularization parameters. Because the arising matrices are nonsymmetric the standard analysis techniques to analyse the spectral properties of the preconditioned systems are not straightforwardly applicable. To aid the analysis we use the Generalized Locally Toeplitz framework. With some simplifications we illustrate that for a broad range of the involved parameters, which are also of practical interest, the spectrum of the preconditioned matrix remains bounded and well-away from zero. We provide also extensive numerical experiments showing that the solver is near optimal with respect to $h$, and highly efficient for many parameter combinations of practical interest. The timing results confirm a nearly linear computational complexity of the overall nonlinear solver, which includes inner-outer solver for the corresponding Jacobian.

We note that some of the regularization parameters, namely, $\alpha$ and $\gamma$ counteract. Due to $\alpha$ and $h$ being small and $1/\gamma$ being large, the arising matrices are in general very ill-conditioned, also due to bad scaling of the blocks. To mitigate the latter effect we introduce a particular additional scaling of the system matrix.

The solution methods consists of nonlinear semismooth Newton iterations and a linear solver for the Jacobian, which changes at each nonlinear iteration. The nonlinearity is of a particular nature as it reflects the fact that the set of active points can differ for different mesh and problem parameters. The stopping criteria for the linear and nonlinear solvers should be dynamic, reflecting the problem at hand and the accuracy of the discretization.

In order to facilitate the convergence of the nonlinear solver we use a hybrid nonlinear-linear solver, utilizing one or several (not necessarily consecutive) coarse meshes. The resulting method is in general computationally very efficient.

Finally, for the linearised problem to be solved at each nonlinear step, we propose a preconditioner, referred to as PSSB, and analyse in detail the spectrum of the corresponding preconditioned matrix. The preconditioner shows fast convergence of the (outer) iterative solver, in our case GCR. Since the preconditioner has a two-by-two block structure, to solve systems with the blocks, we use as an inner solver another iterative method, preconditioned by an efficient algebraic multigrid scheme, AGMG.

Provided the complexity of the considered optimization problem, the solution procedure exhibits nearly optimal performance for a broad range of valued of all involved parameters, that are of practical interest.
Acknowledgements

The work of the first author and partially of the second author has been supported by the Research Grant VR-2017-03749, ‘Mathematics and numerics in PDE-constrained optimization problems with state and control constraints’, financed by the Swedish Research Council.

The authors are indebted to Stefano Serra-Capizzano and Sven-Erik Ekström for the help with the GLT technology and the numerous fruitful discussions.

The authors declare that there are no conflicts of interest, related to this work.

References


Y. Notay, AGMG software and documentation, see [http://agmg.eu](http://agmg.eu)


A Appendix

The differential equation in this work is the stationary convection-diffusion equation

\[-\epsilon \Delta u + (b \cdot \nabla) u = f \text{ in } \Omega = [0,1]^2 \] (39)

We use square mesh of the unit square with mesh-size $h$ and bilinear finite element basis functions. For simplicity we choose $b=[1,1]$, which corresponds to a diagonal wind.

Notations:

- $M$ mass matrix
- $M_0$ lumped mass matrix
- $L$ stiffness matrix, corresponding to the discrete Laplacian $\Delta u$
- $Cx$ stiffness matrix, corresponding to the discrete $\frac{\partial u}{\partial x}$
- $Cy$ stiffness matrix, corresponding to the discrete $\frac{\partial u}{\partial y}$
- $K_0 = \epsilon L + Cx + Cy$ stiffness matrix, corresponding to $-\epsilon \Delta u + b_1 \frac{\partial u}{\partial x} + b_2 \frac{\partial u}{\partial y}$
- $\hat{K} = \sqrt{\alpha K_0} \| u \|^{-\frac{1}{2}}$ scaled stiffness matrix $K_0$. 

32
We are interested in estimating the order of the real and the imaginary parts of the eigenvalues of the generalized eigenvalue problem

\[ Q^+ v = \mu S v, \]

where \( S = M_2 + \hat{K} M_1^{-1} \hat{K}^T \), \( Q^+ = \frac{1}{2}(\hat{K} W + W \hat{K}^T) \), \( 0 < \alpha < 1 \), \( M_2 = 0 \), \( M_1 = h^2 I \), \( W = I \).

Symbols for the convection-diffusion problem, discretized by bilinear FEM

For completeness we include also the element stiffness matrices for the corresponding terms in the variational form of the equation.

Below, \( \theta_1 \) and \( \theta_2 \) are generic angles between \(-\pi\) and \(\pi\).

In all figures below, \( \gamma = 0 \) denotes that when computing the symbol and the eigenvalues of the corresponding matrices, no box constraints are imposed on the state variable.

The symbols of the diagonal matrices are as follows: \( f^M = h^2 \) and \( f^W = 1 \).

a The symbol of the stiffness matrix, corresponding to \( \Delta u \)

\[ L^{(e)} = \frac{1}{6} \begin{bmatrix} 4 & -1 & -2 & -1 \\ -1 & 4 & -1 & -2 \\ -2 & -1 & 4 & -1 \\ -1 & -2 & -1 & 4 \end{bmatrix}. \]

The stencil for the assembled matrix \( L \) is shown in Figure 14.

![Figure 14: The stencil of the matrix \( L (\frac{1}{3} \times) \)](image)

The matrix has also a block-tridiagonal structure and its symbol is

\[
\begin{align*}
  f_L^0(\theta) &= 2(4 - \cos(\theta)), \\
  f_L^1(\theta) &= f_L^{-1}(\theta) = -(1 + 2\cos(\theta)), \\
  f_L(\theta_1, \theta_2) &= \frac{1}{2}(8 - 2\cos(\theta_1) - 2\cos(\theta_2)(1 + 2\cos(\theta_1))).
\end{align*}
\]

For brevity, in the sequel we introduce the notation \( g_i = \frac{1}{3}[8 - 2\cos(\theta_1) - 2\cos(\theta_2)(1 + 2\cos(\theta_1))] \).
b  The symbols of the stiffness matrices, corresponding to $\frac{\partial u}{\partial x}$ and $\frac{\partial u}{\partial y}$

$$C_x^{(e)} = \frac{h}{12} \begin{bmatrix} -2 & -1 & 1 & 2 \\ -1 & -2 & 2 & 1 \\ -1 & -2 & 2 & 1 \\ -2 & -1 & 1 & 2 \end{bmatrix}, \quad C_y^{(e)} = \frac{h}{12} \begin{bmatrix} -2 & 2 & 1 & -1 \\ -1 & 1 & 2 & -2 \\ -1 & 1 & 2 & -2 \end{bmatrix}.$$ 

The stencils for the assembled matrices $C_x$ and $C_y$ are shown in Figure 16.

![Figure 16: The stencil of the matrices $C_x$ and $C_y$](image)
The matrices have also a block-tridiagonal structure and their corresponding symbols are computed as

\[
\begin{align*}
    f_{C_0}^{C_x}(\theta) &= -i 8 \sin(\theta), & f_{C_1}^{C_x}(\theta) &= f_{C_{-1}}^{C_x}(\theta) = -i 2 \sin(\theta), \\
    f_{C_x}(\theta_1, \theta_2) &= -i \frac{1}{3} h \sin(\theta_1)(2 + \cos(\theta_2)). \\
    f_{C_y}^{C_0}(\theta) &= 0, & f_{C_1}^{C_y}(\theta) &= f_{C_{-1}}^{C_y}(\theta) = 4 + 2 \cos(\theta), \\
    f_{C_y}(\theta_1, \theta_2) &= -i \frac{1}{3} h \sin(\theta_2)(2 + \cos(\theta_1)).
\end{align*}
\]

(41)

Again for notational convenience we introduce \( g_2 = \frac{1}{3} \left[ \sin(\theta_1)(2 + \cos(\theta_2)) + \sin(\theta_2)(2 + \cos(\theta_1)) \right] \). Thus, \( f^{C_x} + f^{C_y} = -i h g_2 \).

\[
\begin{align*}
    f^{K_0} &= \epsilon g_1 - i h g_2, \\
    f^K &= \frac{\sqrt{\alpha} \sqrt{\gamma}}{\sqrt{1 + \gamma}} (\epsilon g_1 - i h g_2)
\end{align*}
\]

The plots of the sampled symbol and the eigenvalues of \( \hat{K} \) (after applying some boundary conditions, is shown in Figure 18.

To enable reproducibility of the computations of the symbols, we include the Matlab code for the sampling of the symbols.

```matlab
n = 50; n2 = n^2;
t1 = linspace(-pi+2*pi/(n*n+1),-pi+n*n*2*pi/(n*n+1),n); % grid to sample the symbol
t2 = t1;
```

Figure 17: \( C_x \) and \( C_y \)

c The symbol of the stiffness matrix \( \hat{K} \)

The convection-diffusion matrix \( K_0 \) is by construction \( K_0 = \epsilon L + C_x + C_y \). Based on the properties of the GLT sequences, the symbols of \( K_0 \) and \( \hat{K} \) have the form

\[
\begin{align*}
    f^{K_0} &= \epsilon g_1 - i h g_2, \\
    f^K &= \frac{\sqrt{\alpha} \sqrt{\gamma}}{\sqrt{1 + \gamma}} (\epsilon g_1 - i h g_2)
\end{align*}
\]

The plots of the sampled symbol and the eigenvalues of \( \hat{K} \) (after applying some boundary conditions, is shown in Figure 18.

To enable reproducibility of the computations of the symbols, we include the Matlab code for the sampling of the symbols.

```matlab
n = 50; n2 = n^2;
t1 = linspace(-pi+2*pi/(n*n+1),-pi+n*n*2*pi/(n*n+1),n); % grid to sample the symbol
t2 = t1;
```
Figure 18: The spectrum and the symbol of $\hat{K}$, $\alpha=0.001$

```
%% Symbol functions
ffM = @(t1,t2)(4*(2+cos(t1)).*(2+cos(t2)));
ffM1 = @(h)(h^2);
ffL = @(t1,t2)(1/3*(8-2*cos(t1)-2*cos(t2).*1+2*cos(t1)));
ffCx = @(t1,t2,h)(-i*h/3*sin(t1).*2+cos(t2));
ffCy = @(t1,t2,h)(-i*h/3*sin(t2).*2+cos(t1));
ffK = @(t1,t2,h,ep,a,b)(a*b*(ep*ffL(t1,t2)+ffCx(t1,t2,h)+ffCy(t1,t2,h)));
ffWpls = @(h)(1+ffM2(h)/ffM1(h));
ffS = @(t1,t2,h,ep,a,b)(ffM2(h)+ffK(t1,t2,h,ep,a,b).*ffK(-t1,-t2,h,ep,a,b)/ffM1(h));
ffQpls = @(t1,t2,h,ep,a,b)(0.5*((ffK(t1,t2,h,ep,a,b)+ffK(-t1,-t2,h,ep,a,b))/ffWpls(h)));
ffQplsS = @(t1,t2,h,ep,a,b,c)(ffS(t1,t2,h,ep,a,b)/ffQpls(t1,t2,h,ep,a,b));
ffSQpls = @(t1,t2,h,ep,a,b)(ffQplsS(t1,t2,h,ep,a,b))/ffQpls(t1,t2,h,ep,a,b));
```

```
%% Sample
for k=1:n,
    for l=1:n
        fM((k-1)*n+l) = ffM(t1(k),t2(l)); % symbol mass matrix
        fM((k-1)*n+l) = ffM2(h); % symbol lumped mass matrix (h^2)
        fL((k-1)*n+l) = ffL(t1(k),t2(l)); % symbol bilinear Laplacian
        fCx((k-1)*n+l) = ffCx(t1(k),t2(l),h); % symbol convection part, x-derivative
        fCy((k-1)*n+l) = ffCy(t1(k),t2(l),h); % symbol convection part, y-derivative
        fK((k-1)*n+l) = ffK(t1(k),t2(l),h,epsx,opt_alphaS,opt_gammaR); % symbol K
        fS((k-1)*n+l) = ffS(t1(k),t2(l),h,epsx,opt_alphaS,opt_gammaR); % symbol S
        fQpls((k-1)*n+l) = ffQpls(t1(k),t2(l),h,epsx,opt_alphaS,opt_gammaR);
        fQplsS((k-1)*n+l) = ffQplsS(t1(k),t2(l),h,epsx,opt_alphaS,opt_gammaR);
        fSQpls((k-1)*n+l) = ffSQpls(t1(k),t2(l),h,epsx,opt_alphaS,opt_gammaR);
    end
end
```

```
%% Plot
figure,scatter(real(EK),imag(EigK),'*');hold;scatter(real(fK),imag(fK),'ro')
legend('Eigenvalues','symbol')
```

d  The symbol of the Schur complement matrix $S = M_2 + \hat{K}M_1^{-1}\hat{K}^T$

Under the assumptions made, we compute next the symbol of the Schur complement matrix. The symbol of $S$ has the form

$$f^S = \frac{\alpha \gamma h^{-2}}{1+\gamma} f^K \overline{f^K} = \frac{\alpha \gamma h^{-2}}{1+\gamma} (\epsilon^2 g_1^2 + h^2 g_2^2)$$

with $\overline{f^K}$ being the complex conjugate of $f^K$. The symbol and the eigenvalues of $S$ are illustrated in Figure 19. We note that by definition $S$ is symmetric and positive definite, which is correctly reflected by its symbol.

![Figure 19](image)

Figure 19: The spectrum and the symbol of $S$; $\alpha = 10^{-3}$ no sparsity constraints

e  The symbol of the matrix $Q^+ = \hat{K}W^+ + W^+ \hat{K}^T$

We recall that $W^+ = I$. Under the current assumptions the symbol of $Q^+ = \frac{1}{2} (\hat{K} + \hat{K}^T)$ becomes equal to the scaled symbol of $L$,

$$f^Q = \frac{\sqrt{\alpha} \sqrt{\gamma}}{\sqrt{1+\gamma}} \frac{1}{2} (f^K + \overline{f^K}) = \frac{\sqrt{\alpha} \sqrt{\gamma}}{\sqrt{1+\gamma}} \epsilon g_1.$$  

The spectrum of $Q^+$ and the sampled symbol $f^Q$ are plotted in Figure 20.
The symbol of the matrix $S^{-1}Q^+$

The symbol of $S^{-1}Q^+$ is computed to be

$$f^{S^{-1}Q} = \frac{3}{\sqrt{\alpha \sqrt{\gamma}} \sqrt{1+\gamma \epsilon g_1}} \frac{h^{-2} \sqrt{1+\gamma \epsilon g_1}}{\sqrt{\alpha \sqrt{\gamma} (\epsilon^2 g_1^2 + h^2 g_2^2)}}.$$  

For completeness we include Figure 22 for the spectrum and the symbol of the matrix $(Q^+)^{-1}S$. 

Figure 20: The spectrum and the symbol of $Q^+$; $\alpha = 10^{-3}$, no sparsity constraints
Figure 21: The spectrum and the symbol of $S^{-1}Q^+$; $\alpha = 10^{-3}$, no sparsity constraints

Figure 22: The spectrum and the symbol of $S^{-1}Q^+$; $\alpha = 10^{-3}$, no sparsity constraints