Numer Algor (2016) 73:631–663
DOI 10.1007/s11075-016-0111-1

ORIGINAL PAPER

Comparison of preconditioned Krylov subspace iteration methods for PDE-constrained optimization problems

Poisson and convection-diffusion control

Owe Axelsson[1] · Shiraz Farouq[2] · Maya Neytcheva[2]

Received: 11 August 2015 / Accepted: 11 February 2016 / Published online: 23 February 2016

Abstract Saddle point matrices of a special structure arise in optimal control problems. In this paper we consider distributed optimal control for various types of scalar stationary partial differential equations and compare the efficiency of several numerical solution methods. We test the particular case when the arising linear system can be compressed after eliminating the control function. In that case, a system arises in a form which enables application of an efficient block matrix preconditioner that previously has been applied to solve complex-valued systems in real arithmetic. Under certain assumptions the condition number of the so preconditioned matrix is bounded by 2. The numerical and computational efficiency of the method in terms of number of iterations and elapsed time is favourably compared with other published methods.

Keywords PDE-constrained optimization problems · Finite elements · Iterative solution methods · Preconditioning

✉ Maya Neytcheva
maya.neytcheva@it.uu.se

Owe Axelsson
owe.axelsson@it.uu.se

Shiraz Farouq
shiraz.farouq@gmail.com

[1] Institute of Geonics AS CR, Ostrava, The Czech Republic

[2] Department of Information Technology, Uppsala University, Uppsala, Sweden

1 Introduction

The numerical solution of partial differential equations (PDEs) leads normally to algebraic problems of large dimensions, for which iterative solution methods are most efficient. In distributed optimal control problems, additional variables, the control function and the Lagrangian multiplier, appear on the scene thereby increasing the dimensions of the problem even further. Thus, a crucial task is to construct an efficient preconditioner that gives close eigenvalue bounds of the preconditioned matrix. The contemporary scientific literature of preconditioning optimal control systems describes various methods for constructing efficient preconditioners, including operator preconditioning using special norms [19], and the classical Schur complement approximation [14–16]. In addition to those, in this paper we use a strategy based on a particular matrix structure, used earlier in different contexts, [1–3, 7].

The paper has the following structure. In Section 2 we present our method and derive its properties. In Section 3 we outline the formulation of control problems, constrained by PDEs. Here we consider optimal control of processes governed by scalar equations, namely, by the Poisson equation and by the convection-diffusion equation. In Section 4 we describe various preconditioning techniques and solution approaches for the considered problems that have already been published, as well as how our method is applicable in the context of PDE-constrained optimization problems. Finally, in Section 5 we present numerical results on the experiments followed by some concluding remarks. We provide performance comparison of the various preconditioning strategies considered here for our test problems: for the Poisson case, we consider the optimal control of the heat flow. In case of the optimal control of the convection-diffusion control problem, we consider the objective function as in the case of the Poisson control problem. Both these examples are also dealt with in [16].

2 A preconditioner for two-by-two block matrices of a special form

Consider two-by-two block matrices in the form

$$\mathscr{A} = \begin{bmatrix} A & -bB_2 \\ aB_1 & A \end{bmatrix}. \tag{1}$$

where A, B_i, $i = 1, 2$ are square matrices. We assume that A is positive semidefinite and a and b are nonzero scalars that have the same sign. Note, that in this section A, B, $\mathscr{A}$ and $\mathscr{B}$ denote generic matrices, not related to any particular application.

A matrix in a similar form, such as $\begin{bmatrix} A & B_2 \\ B_1 & -cA \end{bmatrix}$, where $c \geq 0$, can readily be transformed to the form (1) by scaling and transformation of the variables in the system to be solved. In many applications $B_2 = B_1^T$.

Such matrices arise in various applications such as when solving complex-valued systems (see, e. g. [2, 3] and the references therein), in some approximations of

matrices arising in discrete phase-field models (see e.g. [1, 7]) and, as shown below, also when solving certain optimal control problems. We analyze an efficient non-symmetric preconditioner for $\mathscr{A}$, blue namely,

$$\mathscr{B} = \begin{bmatrix} A & -bB_2 \\ aB_1 & A + \sqrt{ab}(B_1 + B_2) \end{bmatrix}. \tag{2}$$

It turns out that the solution of systems with $\mathscr{B}$ only involves solutions of systems with the matrices $H_i = \alpha A + \sqrt{ab}B_i$, $i = 1, 2$ at each iteration. Here $\alpha > 0$ is a method parameter.

We show that the eigenvalues of the correspondingly preconditioned matrix $\mathscr{B}^{-1}\mathscr{A}$ are real and positive, and are contained in the interval $\frac{1}{2} \leq \lambda_{\min} \leq \lambda \leq \lambda_{\max} \leq 1$. We also show that under certain conditions the lower bound can be further improved by a proper choice of the method parameter α. The preconditioner is then applied in the context of some optimal control problems.

We present below some properties of $\mathscr{B}$ and the preconditioned matrix $\mathscr{B}^{-1}\mathscr{A}$.

2.1 Efficient implementation of the preconditioner $\mathscr{B}$

As has been shown in earlier papers (e.g. [1, 3]), the inverse of the matrix $\mathscr{B}$ has the form shown in Proposition 1 and the following result holds.

Proposition 1 *Consider a matrix $\mathscr{B}$ of the form (2). Let $H_i = A + \sqrt{ab}\, B_i$, $i = 1, 2$ be nonsingular. Then*

$$\mathscr{B}^{-1} = \begin{bmatrix} H_1^{-1} + H_2^{-1} - H_2^{-1} A H_1^{-1} & \sqrt{\frac{b}{a}} \left(I - H_2^{-1} A \right) H_1^{-1} \\ -\sqrt{\frac{b}{a}}\, H_2^{-1} \left(I - A H_1^{-1} \right) & H_2^{-1} A H_1^{-1} \end{bmatrix}.$$

Proof The validity of this expression can easily be established by a matrix multiplication $\mathscr{B}^{-1}\mathscr{B} = I$. □

It follows that the solution of a system $\mathscr{B} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{bmatrix}$ can be readily performed.

Proposition 2 *Assume that $A + \sqrt{ab}B_i, i = 1, 2$ are nonsingular. Then $\mathscr{B}$ is nonsingular and a linear system with the preconditioner $\mathscr{B}$,*

$$\begin{bmatrix} A & -bB_2 \\ aB_1 & A + \sqrt{ab}(B_1 + B_2) \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_1 \\ \boldsymbol{f}_2 \end{bmatrix}$$

can be solved with only one solution with $A + \sqrt{ab}B_1$ and one with $A + \sqrt{ab}B_2$.

Proof It follows from Proposition 1 that an action of the inverse of $\mathscr{B}$ can be written in the form

$$\begin{aligned}\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} &= \begin{bmatrix}A & -bB_2\\ aB_1 & A+\sqrt{ab}(B_1+B_2)\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{f}_1\\ \mathbf{f}_2\end{bmatrix}\\ &= \begin{bmatrix} H_1^{-1}\mathbf{f}_1 + H_2^{-1}\mathbf{f}_1 - H_2^{-1}AH_1^{-1}\mathbf{f}_1 + \sqrt{\frac{b}{a}}\left(I - H_2^{-1}A\right)H_1^{-1}\mathbf{f}_2\\ -\sqrt{\frac{a}{b}}H_2^{-1}\left(I - AH_1^{-1}\right)\mathbf{f}_1 + H_2^{-1}AH_1^{-1}\mathbf{f}_2\end{bmatrix}\\ &= \begin{bmatrix} H_2^{-1}\mathbf{f}_1 + \mathbf{g} - H_2^{-1}A\mathbf{g}\\ -\sqrt{\frac{a}{b}}H_2^{-1}\mathbf{f}_1 + \sqrt{\frac{a}{b}}H_2^{-1}A\mathbf{g}\end{bmatrix}\\ &= \begin{bmatrix}\mathbf{g} + H_2^{-1}(\mathbf{f}_1 - A\mathbf{g})\\ -\sqrt{\frac{a}{b}}H_2^{-1}(\mathbf{f}_1 - A\mathbf{g})\end{bmatrix} = \begin{bmatrix}\mathbf{g}+\mathbf{h}\\ -\sqrt{\frac{a}{b}}\mathbf{h}\end{bmatrix}\end{aligned}$$

where

$$\mathbf{g} = H_1^{-1}\left(\mathbf{f}_1 + \sqrt{\frac{b}{a}}\mathbf{f}_2\right), \quad \mathbf{h} = H_2^{-1}(\mathbf{f}_1 - A\mathbf{g}).$$

□

The computation can take place using the following algorithm.

Algorithm 1 Solving the factorized operator

1: Solve $H_1\mathbf{g} = \mathbf{f}_1 + \sqrt{\frac{b}{a}}\mathbf{f}_2$.
2: Compute $A\mathbf{g}$ and $\mathbf{f}_1 - A\mathbf{g}$.
3: Solve $H_2\mathbf{h} = \mathbf{f}_1 - A\mathbf{g}$.
4: Compute $\mathbf{x} = \mathbf{g} + \mathbf{h}$ and $\mathbf{y} = -\sqrt{\frac{a}{b}}\mathbf{h}$.

2.2 Spectral properties of $\mathscr{B}^{-1}A$

Consider the generalized eigenvalue problem

$$\lambda\,\mathscr{B}\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} = \mathscr{A}\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}, \quad \|\mathbf{x}\| + \|\mathbf{y}\| \neq 0.$$

We study first the case where A is symmetric, $B_1 = B^T$, $B_2 = B$ and A and $B + B^T$ are positive semidefinite or A is positive definite. This case has been already analyzed and tested in earlier research and applied to problems of different origin, see [1–3, 7]. For completeness, we include here the related, but somewhat extended theoretical results.

2.2.1 The symmetric case: A is positive semidefinite, $B_2 = B_1^T$

Assume that $B + B^T$ is positive semidefinite and

$$ker(A) \cap ker(B) = \{0\}. \tag{3}$$

Note that if $B\mathbf{x} = 0$, $\mathbf{x} \neq 0$, then $\mathbf{x}^{\mathbf{T}}(B + B^T)x = 0$ and since $B + B^T$ is positive semidefinite, $(B + B^T)\mathbf{x} = 0$ and therefore also $B^T\mathbf{x} = 0$. Then it follows that $A + \sqrt{ab}B$ and $A + \sqrt{ab}B^T$, and hence also $\mathscr{B}$, since $\mathscr{B}^{-1}$ only involves inverses of those matrices, are nonsingular. We show first that then $\mathscr{A}$ is also nonsingular.

Proposition 3 *Let condition (3) hold and let also $a, b \in \mathbb{R}$ be nonzero scalars with the same sign. Then $\mathscr{A}$ is nonsingular.*

Proof If

$$\mathscr{A}\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} \equiv \begin{bmatrix} A & -bB^T \\ aB & A \end{bmatrix}\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} = \begin{bmatrix}\mathbf{0}\\ \mathbf{0}\end{bmatrix}, \tag{4}$$

then

$$\begin{aligned} &\mathbf{x}^* A\mathbf{x} - b\mathbf{x}^* B^T\mathbf{y} = \mathbf{0},\\ &a\mathbf{y}^* B\mathbf{x} + \mathbf{y}^* A\mathbf{y} = \mathbf{0} \end{aligned}$$

so, $\frac{1}{b}\mathbf{x}^* A\mathbf{x} + \frac{1}{a}\mathbf{y}^* A\mathbf{y} = 0$. Since A is positive semidefinite, it follows that $\mathbf{x}$ and $\mathbf{y} \in ker\, A$. But it follows then from (4) that $B^T\mathbf{y} = \mathbf{0}$ and $B\mathbf{x} = \mathbf{0}$, implying by (3) that $\mathscr{A}\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} = \begin{bmatrix}\mathbf{0}\\ \mathbf{0}\end{bmatrix}$ has only the trivial solution. □

Proposition 4 *Let $\mathscr{A} = \begin{bmatrix} A & -bB^T \\ aB & A \end{bmatrix}$, where a, b are nonzero and have the same sign and let $\mathscr{B} = \begin{bmatrix} A & -bB^T \\ aB & A + \sqrt{ab}\,(B + B^T) \end{bmatrix}$. If condition (3) holds then the eigenvalues of $\mathscr{B}^{-1}\mathscr{A}$, are contained in the interval $[\frac{1}{2}, 1]$.*

Proof For the generalized eigenvalue problem

$$\lambda\mathscr{B}\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} = \mathscr{A}\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}, \qquad \|\mathbf{x}\| + \|\mathbf{y}\| \neq 0$$

it follows from Proposition 3 that $\lambda \neq 0$. It holds

$$\left(\frac{1}{\lambda} - 1\right)\mathscr{A}\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} = \begin{bmatrix} 0 \\ \sqrt{ab}\,(B + B^T)\mathbf{y} \end{bmatrix}$$

Here $\lambda = 1$ if $\mathbf{y} \in ker(B + B^T)$. If $\lambda \neq 1$, then $A\mathbf{x} = bB^T\mathbf{y}$ and

$$\left(\frac{1}{\lambda} - 1\right)(\mathbf{y}^* A\mathbf{y} + a\mathbf{y}^* B\mathbf{x}) = \sqrt{ab}\,\mathbf{y}^*(B + B^T)\mathbf{y},$$

i.e.,

$$\left(\frac{1}{\lambda}-1\right)\left(\mathbf{y}^*A\mathbf{y}+\frac{a}{b}\mathbf{x}^*A\mathbf{x}\right)=\sqrt{ab}\,\mathbf{y}^*(B+B^T)\mathbf{y}\,.$$

Since both A and $B+B^T$ are positive semidefinite, it follows that $\lambda \le 1$.

Further, as $A\mathbf{x}=bB^T\mathbf{y}$ it holds that $\mathbf{y}^*A\mathbf{x}=b\mathbf{y}^*B^T\mathbf{y}$ so

$$\left(\frac{1}{\lambda}-1\right)(b\mathbf{y}^*B^T\mathbf{y}+a\mathbf{x}^*B\mathbf{x})=\sqrt{ab}\,\mathbf{x}^*(B+B^T)\mathbf{y}$$

or

$$\left(\frac{1}{\lambda}-1\right)(b\mathbf{y}^*(B+B^T)\mathbf{y}+a\mathbf{x}^*(B+B^T)\mathbf{x})=2\sqrt{ab}\,\mathbf{x}^*(B+B^T)\mathbf{y}\,.$$

Using that $B+B^T$ is positive semidefinite, $\|\mathbf{x}\|+\|\mathbf{y}\|\neq 0$, a and b have the same sign, and $\lambda \le 1$, it follows from Cauchy–Schwarz inequality that

$$\frac{1}{\lambda}-1\le\frac{2\sqrt{ab}\mid \mathbf{x}^*(B+B^T)\mathbf{y}\mid}{\mid b\mid \mathbf{y}^*(B+B^T)\mathbf{y}+\mid a\mid \mathbf{x}^*(B+B^T)\mathbf{x}}\le 1\,,$$

that is, $\lambda\ge\frac{1}{2}$. □

2.2.2 A is positive definite

We assume now that A is positive definite and consider the parameter dependent preconditioner

$$\mathscr{B}_\alpha=\begin{bmatrix}A & -bB_2\\ aB_1 & \alpha^2A+\alpha\sqrt{ab}\,(B_1+B_2)\end{bmatrix}.$$

We let the parameter α be larger or equal to 1. The generalized eigenvalue problem takes here form

$$\lambda\begin{bmatrix}A & -bB_2\\ aB_1 & \alpha^2A+\alpha\sqrt{ab}\,(B_1+B_2)\end{bmatrix}\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}=\begin{bmatrix}A & -bB_2\\ aB_1 & A\end{bmatrix}\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}.$$

Let $\widetilde{B}_i=\sqrt{ab}\,A^{-1/2}B_iA^{-1/2}$, $i=1,2$. By a transformation with $\begin{bmatrix}A^{-1/2} & 0\\ 0 & A^{-1/2}\end{bmatrix}$ from both sides, the eigenvalue problem takes the form

$$\lambda\begin{bmatrix}I & -\sqrt{\frac{b}{a}}\widetilde{B}_2\\ \sqrt{\frac{a}{b}}\widetilde{B}_1 & \alpha^2I+\alpha(\widetilde{B}_1+\widetilde{B}_2)\end{bmatrix}\begin{bmatrix}\tilde{\mathbf{x}}\\ \tilde{\mathbf{y}}\end{bmatrix}=\begin{bmatrix}I & -\sqrt{\frac{b}{a}}\widetilde{B}_2\\ \sqrt{\frac{a}{b}}\widetilde{B}_1 & I\end{bmatrix}\begin{bmatrix}\tilde{\mathbf{x}}\\ \tilde{\mathbf{y}}\end{bmatrix},$$

where $\widetilde{\mathbf{x}}=A^{1/2}\mathbf{x}$, $\widetilde{\mathbf{y}}=A^{1/2}\mathbf{y}$. It follows that $\lambda=1$ if $[(\alpha^2-1)I+\alpha(\widetilde{B}_1+\widetilde{B}_2)]\widetilde{\mathbf{y}}=0$, in particular if $\mathbf{y}=\mathbf{0}$, $\mathbf{x}\neq\mathbf{0}$.

If $\lambda\neq 1$, then $\widetilde{\mathbf{x}}=\sqrt{\frac{b}{a}}\widetilde{B}_2\widetilde{\mathbf{y}}$ and

$$\lambda\left[\widetilde{B}_1\widetilde{B}_2+\alpha^2I+\alpha(\widetilde{B}_1+\widetilde{B}_2)\right]\widetilde{\mathbf{y}}=(I+\widetilde{B}_1\widetilde{B}_2)\widetilde{\mathbf{y}}.$$

We let now $B_1=B^T$, $B_2=B$ and assume that $B+B^T$ is positive semidefinite.

Then

$$\lambda\,\widetilde{\mathbf{y}}^*\left[\widetilde{B}^T\widetilde{B}+\alpha^2 I+\alpha(\widetilde{B}+\widetilde{B}^T)\right]\widetilde{\mathbf{y}}=\widetilde{\mathbf{y}}^*(I+\widetilde{B}^T\widetilde{B})\widetilde{\mathbf{y}} \tag{5}$$

where $\widetilde{B} = \sqrt{ab}\,A^{-1/2}BA^{-1/2}$. Since the matrices within brackets in (5) are symmetric and positive definite the eigenvalues are real and positive.

Let $\widetilde{\mathbf{y}}$ be an eigenvector of $\widetilde{B}$, i.e. $\widetilde{B}\widetilde{\mathbf{y}}=\mu\widetilde{\mathbf{y}}$, $\widetilde{\mathbf{y}}\neq\mathbf{0}$. Since $\widetilde{\mathbf{y}}^*\widetilde{B}^T=\overline{\mu}\widetilde{\mathbf{y}}^*$, where $\overline{\mu}$ is the complex conjugate of μ, it follows from (5) that

$$\lambda=\lambda(\mu):=\frac{1+|\mu|^2}{\alpha^2+|\mu|^2+2\alpha Re(\mu)}.$$

Hence, since $Re(\mu)\le|\mu|$,

$$\lambda\ge\frac{1+|\mu|^2}{(\alpha+|\mu|)^2}.$$

An elementary computation shows that

$$\min_{\mu}\lambda\ge\frac{1}{1+\alpha^2}$$

and is taken for $|\mu|=\frac{1}{\alpha}$. For a fixed value of α, the function $\lambda(\mu)$ varies as shown in Fig. 1, where $\lambda=1/\alpha^2$ for $|\mu|=0$ and $|\mu|=\mu_0$.

If $Re(\mu)>0$, for not too large values of $|\mu|\le\mu_0$, namely if

$$\alpha^2(1+|\mu|^2)\le\alpha^2+|\mu|^2+2\,\alpha\,Re(\mu),$$

that is,

$$|\mu|^2\le\mu_0^2=\frac{2\alpha}{\alpha^2-1}Re(\mu), \tag{6}$$

then $1/\alpha^2$ is also an upper bound for all values of μ satisfying (6). The condition number as a function of α is then bounded by

$$\kappa(\mathscr{B}_\alpha^{-1}\mathscr{A})\le\frac{1}{\alpha^2}(1+\alpha^2)=1+\frac{1}{\alpha^2}.$$

We have thus proved the following proposition.

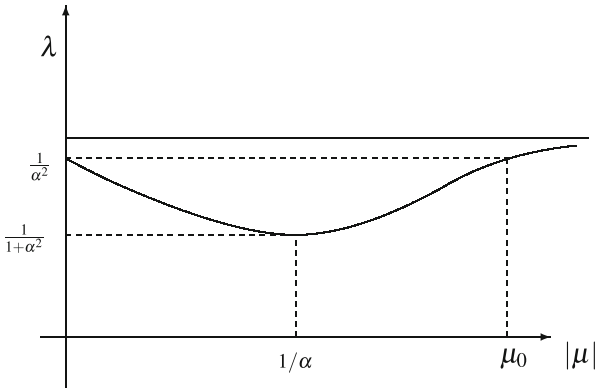


Fig. 1 Function $\lambda(\mu)$ for a fixed value of $\alpha>1$

Proposition 5 *Assume that A is positive definite and $B + B^T$ are positive semidefinite and let μ be an eigenvalue of $\widetilde{B} = \sqrt{ab}\, A^{-1/2} B A^{-1/2}$.*

If $\alpha \geq 1$ and $|\mu|^2 \leq \frac{2\alpha}{\alpha^2-1} Re(\mu)$ then $\kappa(\mathscr{B}_\alpha^{-1}\mathscr{A}) \leq 1 + \frac{1}{\alpha^2}$.

If $\alpha = 1$ the bound $\kappa(\mathscr{B}_\alpha^{-1}\mathscr{A}) \leq 2$ holds for all values of $|\mu|$.

Remark 1 If $\alpha = 2$, $Re(\mu) \geq \frac{3}{4}|\mu|$ and $|\mu| \leq \frac{2\alpha}{\alpha^2-1}\frac{3}{4} = 1$, then $\kappa\left(\mathscr{B}_\alpha^{-1}\mathscr{A}\right) \leq 1 + \frac{1}{\alpha^2} = 1.25$. For still larger values of α the condition number decreases even further if $|\mu|$ is correspondingly bounded.

Corollary 1 *If B is symmetric and positive definite and $\mu_{\max} = \frac{2\alpha}{\alpha^2-1}$, that is, $\alpha = \alpha_{opt} = \frac{1}{\mu_{\max}} + \sqrt{\frac{1}{\mu_{\max}^2} + 1}$, then*

$$\kappa(\mathscr{B}_\alpha^{-1}\mathscr{A}) = 1 + \frac{1}{\left(\frac{1}{\mu_{\max}} + \sqrt{\frac{1}{\mu_{\max}^2} + 1}\right)^2}.$$

If $\mu_{\max} = 1$ then $\alpha_{opt} = 1 + \sqrt{2}$ and $\kappa(\mathscr{B}_\alpha^{-1}\mathscr{A}) = 1 + \frac{1}{(\sqrt{2}+1)^2} \approx 1.17$.

We note that by a matrix multiplication with $\begin{bmatrix} I & 0 \\ 0 & \frac{1}{\alpha} I \end{bmatrix}$ from both sides, the preconditioner

$$\begin{bmatrix} A & -\tilde{b} B_2 \\ \tilde{a} B_1 & \alpha^2 A + \alpha\sqrt{\tilde{a}\tilde{b}}\,(B_1 + B_2) \end{bmatrix}$$

is transformed to the matrix $\mathscr{B}$ as in (2), where $a = \frac{1}{\alpha}\tilde{a}$, $b = \frac{1}{\alpha}\tilde{b}$ and systems with it can be solved as described in Algorithm 1.

2.3 Convergence of the iterative solution method

We show next that even though $\mathscr{B}$ is non-symmetric, the preconditioned matrix $\mathscr{B}^{-1}\mathscr{A}$ is normal.

The preconditioned system with the matrix $\mathscr{H} = \mathscr{B}^{-1}\mathscr{A}$ is solved by the GMRES method or its flexible form, FGMRES. At each iteration, the GMRES method minimizes the Euclidean norm of the residual vector $\mathbf{r}^k = \mathbf{b} - \mathscr{H}\mathbf{x}^k$ with respect to vectors $\mathbf{x}^k \in \mathbf{x}^0 + span\{\mathscr{H}\mathbf{r}^0, \mathscr{H}^2\mathbf{r}^0, \cdots \mathscr{H}^{k-1}\mathbf{r}^0\}$ where $\mathbf{r}^0 = \mathbf{b} - \mathscr{H}\mathbf{x}^0$ is the initial residual. This means that $\mathbf{r}^k = P_k(\mathscr{H})\mathbf{r}^0$ for some polynomial $P_k(\cdot)$ of degree k, normalized as $P_k(0) = 1$. Hence, if $\mathscr{H}$ has a complete eigenvector space $\{\mathbf{v}_j\}_{j=1}^n$, then $\mathbf{r}^0 = \sum_{j=1}^n \alpha_j \mathbf{v}_j$ and $\mathbf{v}_j$ are linearly independent, and the rate of convergence is determined by a best polynomial approximation on the set of eigenvalues $\{\lambda_j\}_{j=1}^n$ of $\mathscr{H}$ and $min_{\{P_k(\cdot)\}} max_{\{\lambda_j\}} |P_k(\lambda)|$ gives an upper bound of the rate of convergence. Since the property of complete eigenvalue space is equivalent to normality of the matrix, it is important that $\mathscr{H}$ is normal. More generally, however, if $\mathbf{r}^0 \in V_m = \{\mathbf{v}_j^m\}$, $m < n$ and V_m contains all linearly independent vectors, then the

polynomial bound still holds. In this case, however, rounding errors occurring in the method may cause that residual components outside V_m arise and convergence may slow down. As has been shown in [9], if the set of eigenvectors is incomplete, any slow rate of convergence can be obtained for proper vectors $\mathbf{r}^0$ and one can in general not judge the rate of convergence on polynomial approximation properties on the set of eigenvalues. To show that $\mathscr{H}$ is normal, we use the splitting

$$\mathscr{A} = \mathscr{B} - \sqrt{ab}\begin{bmatrix} 0 & 0 \\ 0 & B_1 + B_2 \end{bmatrix}.$$

We assume that $B_1 = B$ and $B_2 = B^T$. Then, using the explicit expression of $\mathscr{B}^{-1}$, we obtain

$$\mathscr{H} = \mathscr{B}^{-1}\mathscr{A} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} - \begin{bmatrix} 0 & C \\ 0 & E \end{bmatrix},$$

where $C = \sqrt{\frac{b}{a}}\left(I - H_2^{-1}A\right)H_1^{-1}(B_1 + B_2)$ and $E = \sqrt{ab}H_2^{-1}AH_1^{-1}(B_1 + B_2)$. Here E can be symmetrized by the similarity transformation

$$A^{-\frac{1}{2}}H_2EH_2^{-1}A^{\frac{1}{2}} = \sqrt{ab}A^{\frac{1}{2}}H_1^{-1}(B_1 + B_2)H_2^{-1}A^{\frac{1}{2}}.$$

Further, since $H_1A^{-1}H_2 = \left(A + \sqrt{ab}B\right)A^{-1}\left(A + \sqrt{ab}B^T\right) = A + \sqrt{ab}(B + B^T) + abBA^{-1}B^T$, it follows that E has eigenvalues in the interval $\left[0, \frac{1}{2}\right]$. Hence, $\mathscr{H} = \begin{bmatrix} I & -C \\ 0 & I - E \end{bmatrix}$ has eigenvalues in the interval $\left[\frac{1}{2}, 1\right]$. To find its eigenvectors, consider

$$\begin{bmatrix} I & -C \\ 0 & I - E \end{bmatrix}\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \lambda\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}.$$

Here $\lambda = 1$ if $\mathbf{x} \neq \mathbf{0}$, $\mathbf{y} = \mathbf{0}$ and $\mathbf{x} = \mathbf{0}$, $E\mathbf{y} = \mathbf{0}$, $\mathbf{y} \neq \mathbf{0}$, if such vectors $\mathbf{y}$ exist. For $\lambda \neq 1$, it holds that $E\mathbf{y} = (1 - \lambda\mathbf{y}$ and $\mathbf{x} = \frac{1}{1-\lambda}C\mathbf{y}$. Since E has a complete eigenvector space, so has $\mathscr{H}$. As in this case $\mathscr{H}$ is normal and all its eigenvalues are real and positive, $\mathscr{H}$ is positive definite.

3 Control problems constrained by PDEs

The general form of a distributed control problem consists of a cost functional of the form (7) to be minimized subject to a partial differential equation posed on a domain $\Omega \subset \mathbb{R}^d$, $d = 1, 2, 3$:

$$\begin{aligned} &\min_{y,u} \mathscr{J}(y,u) && = \frac{1}{2}\|y - \widehat{y}\|^2_{L_2(\Omega)} + \frac{1}{2}\beta|u|^2_{L_2(\Omega)} \\ &\text{such that } \mathscr{L}(y) && = u, \\ &y = g_D \text{ on} && \partial\Omega_D, \qquad \frac{\partial y}{\partial n} = g_N \text{ on } \partial\Omega_N, \\ &\partial\Omega = \partial\Omega_D && \cup\partial\Omega_N, \qquad \partial\Omega_D \cap \partial\Omega_N = \{\emptyset\}. \end{aligned} \tag{7}$$

Here $\mathscr{L}$ is some scalar or vector partial differential operator, y is the state function, u is the distributed control function, $\widehat{y}$ is the target (desired) solution we want to

approach, $\beta > 0$ is the regularization parameter (also called the cost parameter) which in practice is usually chosen to be small. The PDE-constraint that models the underlying process that needs to be controlled, is referred to as the *state equation*.

One way to solve the minimization problem is through the *first order optimality* conditions, also known as the Karush-Kuhn-Tucker (KKT) conditions. This results in the involvement of another function, the Lagrange multiplier λ (also referred as the dual variable or adjoint state). The existence and the uniqueness of the optimal solution is not discussed here and the reader is referred to [11, 13, 18].

We consider the optimal control of processes governed by two types of scalar PDE's:

> Problem (1): the optimal control of processes governed by the Poisson equation,
> Problem (2): the optimal control of processes governed by the convection-diffusion equation.

Each control problem is stated as a continuous minimization problem which then is dealt with in two steps: optimization and discretization. An important issue, related to the order in which the two steps are carried out, can have consequences on the resulting algebraic system and well as its solution. Therefore, which step is taken first is determined by the following two philosophies:

- the optimize-then-discretize philosophy,
- the discretize-then-optimize philosophy.

In cases where the underlying PDE is self-adjoint, it does not matter which philosophy is followed. The Poisson equation is an example of a self-adjoint PDE. The convection-diffusion equation, however, is not self-adjoint due to the presence of additional terms resulting from stabilization schemes such as the streamline upwind Petrov-Galerkin stabilization (SUPG).

We describe next the basic mathematical framework for our optimal control problems. Let Ω be a bounded domain in $\mathbb{R}^d$, $d = 1, 2$ or 3 and let $\partial\Omega$ be its boundary which is assumed to be sufficiently smooth. Let $L^2(\Omega)$ and $H^1(\Omega)$ denote the standard Lebesgue and Hilbert spaces of functions defined on Ω. We introduce another Hilbert space, namely $H_0^1(\Omega)$, to incorporate functions with homogeneous Dirichlet boundary values at $\partial\Omega$. Further, let $(\cdot,\cdot)$ and $\|\cdot\|$ denote the inner product and norm in $L^2(\Omega)$, respectively, both for scalar and vector functions. Extending [19], and based on [13], we consider now two optimal control problems.

3.1 The control problem constrained by the Poisson equation

Problem 1 Find the state $y \in H_0^1(\Omega)$ and the control $u \in L^2(\Omega)$ that minimize the cost functional

$$\begin{aligned} \min_{y,u} \mathcal{J}(y,u) &= \frac{1}{2}\|y-\widehat{y}\|^2_{L^2(\Omega)} + \frac{1}{2}\beta\|u\|^2_{L^2(\Omega)} \\ -\Delta y &= u \qquad \text{in } \Omega \\ y &= g \qquad \text{on } \partial\Omega. \end{aligned} \tag{8}$$

where $\widehat{y} \in L^2(\Omega)$ is a given target state, $\beta > 0$ is chosen a priori and is sufficiently small to obtain a solution close to the target state but not too small and also not too large as this leads to ill–conditioning. The forcing term $\frac{\beta}{2}\|y\|^2$ acts as a control of the solution to the state equation. By including the control in the cost functional, the problem becomes well-posed.

Using the standard Galerkin finite element method (FEM) and introducing the adjoint variable $\lambda \in H_0^1(\Omega)$ corresponding to the PDE-constraint via the first order optimality conditions and within the framework of either optimize-then-discretize or discretize-then optimize, we obtain the symmetric linear system

$$\begin{bmatrix} M & 0 & K^T \\ 0 & \beta M & -M \\ K & -M & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \\ \mathbf{d} \end{bmatrix}. \tag{9}$$

Here, M is the standard mass matrix, K is the standard stiffness matrix, $\mathbf{b}$ contains the discretized terms of the target state and $\mathbf{d}$ contains the boundary terms. Details of the derivation of the optimality system (9) can be found in [14–16, 19].

Note that if we discretize the state, the control and the adjoint state using the same finite element basis functions, we can reduce the system using the relation

$$\mathbf{u} = \frac{1}{\beta}\boldsymbol{\lambda},$$

resulting in

$$\begin{bmatrix} M & K^T \\ K & -\frac{1}{\beta}M \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{d} \end{bmatrix}. \tag{10}$$

By scaling $\boldsymbol{\lambda}$ we can rewrite (10) in the (non-symmetric) form

$$\begin{bmatrix} M & -\beta K^T \\ K & M \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ -\frac{1}{\beta}\boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{d} \end{bmatrix} \tag{11}$$

and then can directly apply the preconditioner from Section 2.

3.2 The control problem constrained by the convection-diffusion equation

Problem 2 Find the state $y \in H_0^1(\Omega)$ and the control $u \in L^2(\Omega)$ that minimize the cost functional

$$\begin{aligned} \min_{y,u} \mathcal{J}(y,u) &= \frac{1}{2}\|y - \widehat{y}\|^2_{L^2(\Omega)} + \frac{1}{2}\beta\|u\|^2_{L^2(\Omega)} \\ s.t. \; -\varepsilon\Delta y + (\mathbf{w}\cdot\nabla)y &= u \quad \text{in } \Omega \\ y &= g \quad \text{on } \partial\Omega, \end{aligned} \tag{12}$$

where $\mathbf{w}$ is some vector field (for instance, direction of the wind) and g is the Dirichlet boundary data. Further, we assume that $\mathbf{w}$ is divergence free, i.e., $\nabla \cdot \mathbf{w} = 0$. The

scalar quantity ε (for instance, viscosity) satisfies $0 < \varepsilon \ll 1$ and the smaller the value it takes, the more convection-dominated the problem becomes.

As already mentioned, Problem 2 requires some stabilization and the type of that stabilization may lead to differences in the arising linear systems in the 'optimize-then-discretize' or 'discretize-then-optimize' case. A detailed analysis of the SUPG scheme applied to the optimal control of convection-diffusion equation is found in [10]. There, the following main issues are raised.

- The optimize-then-discretize philosophy leads to a strongly consistent but non-symmetric system. The non-symmetry implies that there is no finite dimensional problem for which the discretized system is an optimality system.
- The discretize-then-optimize philosophy leads to an inconsistent but symmetric system.

These issues are addressed in [6] by using the so-called *local projection stabilization* (LPS) scheme. This scheme leads to a symmetric optimality system with an optimal convergence order. While commenting on the issues raised in [10] on the SUPG scheme, in [15], using the LPS scheme proposed in [6], a symmetric and consistent optimality system with both the optimize-then discretize and discretize-then optimize approaches, is presented.

We employ LPS by introducing a projection operator π_h as discussed in [6, 15]. We use a standard Galerkin finite element method and introduce the adjoint variable $\lambda \in H_0^1(\Omega)$ corresponding to the PDE-constraint via first order optimality conditions. Within the framework of either optimize-then-discretize or discretize-then optimize, we obtain

$$\begin{bmatrix} M & 0 & F^T \\ 0 & \beta M & -M \\ F & -M & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \\ \mathbf{d} \end{bmatrix}. \tag{13}$$

Here M is the mass matrix, $F = \varepsilon K + N + T$, where K is the stiffness matrix defined before and

$$N = \int_\Omega (\mathbf{w}\nabla\phi_j).\phi_i, \qquad i, j = 1, ..., n$$

$$T = \delta \int_\Omega (\mathbf{w}\cdot\nabla\phi_i - \pi_h(\mathbf{w}\cdot\nabla\phi_i) \times (\mathbf{w}\cdot\nabla\phi_j - \pi_h(\mathbf{w}\cdot\nabla\phi_j))), \qquad i, j = 1, ..., n.$$

Here π_h is a local L_2-orthogonal projection operator, $\pi : L_2(\Omega) \to V_{2h}$, where V_{2h} is the set of basis functions on the coarser mesh. The stabilization parameter δ is defined locally on individual elements (cells Ω_h^k) and depends on the Peclet number

$$P_\varepsilon^k = \frac{h_k \|\mathbf{w}\|}{\varepsilon};$$

h_k is the maximal diameter of the corresponding cell. Following [6], the stabilization parameter is applied only if P_ε^k is larger than unity, i.e.,

$$\delta_k = \begin{cases} \dfrac{h}{\|\mathbf{w}\|}, & \text{if } P_\varepsilon^k \geq 1, \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

For details on the derivation on the optimality system (13) using either discretize-then-optimize or optimize-then-discretize, see [15].

Using the relation $\mathbf{u} = \frac{1}{\beta}\boldsymbol{\lambda}$ and scaling $\boldsymbol{\lambda}$, again we can rewrite (13) in the form

$$\begin{bmatrix} M & -\beta K^T \\ K & M \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ -\frac{1}{\beta}\boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{d} \end{bmatrix}$$

and directly apply the preconditioner from Section 2.

4 Preconditioners for PDE-constrained optimization problems

We first briefly describe some previously used preconditioning techniques. The notations used to name the preconditioners are as follows. The preconditioners for Problem (1) are denoted by $\widehat{\mathscr{P}}$ and those for Problem 2 - by $\widetilde{\mathscr{P}}$.

4.1 Preconditioners for the distributed optimal control problem constrained by the Poisson equation

Recall that numerically tackling the distributed Poisson control problem leads to an optimality system

$$\mathscr{A}_F = \begin{bmatrix} M & 0 & K^T \\ 0 & \beta M & -M \\ K & -M & 0 \end{bmatrix} \tag{15}$$

with its reduced form given by

$$\mathscr{A}_R = \begin{bmatrix} M & K^T \\ K & -\frac{1}{\beta}M \end{bmatrix}. \tag{16}$$

Extending the work in [16], a Schur complement approximation is derived in [14] that is independent of the mesh size parameter h and the regularization parameter β. This Schur complement approximation is then used for constructing the block-diagonal and the block lower-triangular preconditioner for preconditioning the saddle point system (15). The reduced version of this preconditioner can be applied to the reduced optimality system (16).

For operator preconditioning technique based on standard and non-standard norms, see [19] and [12]. Operator preconditioning using interpolation operator, see [20] and [12], are other methods to solve the reduced optimality system (16). All these methods lead to preconditioners that exhibit h- and β-independent convergence.

We now present a basic overview of some of the preconditioners developed using the approaches discussed above.

4.1.1 Operator preconditioning with standard norms

For the reduced optimality system (16), cf. [12, 20], operator preconditioning with standard norms results in a symmetric positive definite block-diagonal preconditioner

$$\widehat{\mathscr{P}}_{sn} = \begin{bmatrix} M + \sqrt{\beta}K & 0 \\ 0 & M + \sqrt{\beta}K \end{bmatrix}. \tag{17}$$

4.1.2 Operator preconditioning with non-standard norms

Operator preconditioning with non-standard norms results in a symmetric positive definite block-diagonal preconditioner

$$\widehat{\mathscr{P}}_{nsn} = \begin{bmatrix} M + \sqrt{\beta}K & 0 \\ 0 & \frac{1}{\beta}(M + \sqrt{\beta}K) \end{bmatrix}, \tag{18}$$

again for the reduced optimality system (16). This preconditioner is robust with respect to the underlying parameters, i.e., h and β.

Using a more direct derivation than in [19], we show here that the bound on the condition number of the square of the matrix is given by

$$\kappa\left(\left(\widehat{\mathscr{P}}_{nsn}^{-1}\mathscr{A}_R\right)^2\right) \leq 2.$$

Let $\mathscr{G} = \widehat{\mathscr{P}}_{nsn}^{-1}\mathscr{A}_R$ and $\widetilde{M} = (M+\sqrt{\beta}K)^{-1}M$. Since $K = \frac{1}{\sqrt{\beta}}(M+\sqrt{\beta}K-M)$, i.e., $(M + \sqrt{\beta}K)^{-1}K = \frac{1}{\sqrt{\beta}}(I - \widetilde{M})$, it holds

$$\mathscr{G} = \begin{bmatrix} \widetilde{M} & \frac{1}{\sqrt{\beta}}(I - \widetilde{M}) \\ \sqrt{\beta}(I - \widetilde{M}) & -\widetilde{M} \end{bmatrix}.$$

Hence,

$$\mathscr{G}^2 = \begin{bmatrix} \widetilde{M}^2 + (I - \widetilde{M})^2 & 0 \\ 0 & \widetilde{M}^2 + (I - \widetilde{M})^2 \end{bmatrix},$$

that is, $\mathscr{G}$ is an orthogonal matrix. Moreover, since $0 \leq \widetilde{M} < I$, it holds $\frac{1}{2}I \leq \widetilde{M}^2 + (I - \widetilde{M})^2 \leq I$. Hence, the condition number of $\mathscr{G}^2$ is bounded by 2, i.e. $\kappa(\mathscr{G}^2) < 2$, while $\mathscr{G}$ has eigenvalues in the intervals $\left(-1, -\frac{1}{\sqrt{2}}\right) \cup \left(\frac{1}{\sqrt{2}}, 1\right)$. The

number of iterations for the matrix $\mathscr{G}$ is less than twice those required for a matrix with condition number 2 but still more than the iterations needed for the method in Section 2.

4.1.3 Interpolation-based operator preconditioning

Scaling the reduced optimality system (16), we have

$$\widetilde{\mathscr{A}_R}\begin{bmatrix}\mathbf{y}\\ \boldsymbol{\lambda}\end{bmatrix} \equiv \begin{bmatrix} M & \sqrt{\beta}K \\ \sqrt{\beta}K & -M\end{bmatrix}\begin{bmatrix}\mathbf{y}\\ \frac{1}{\sqrt{\beta}}\boldsymbol{\lambda}\end{bmatrix} = \begin{bmatrix}\mathbf{b}\\ \frac{1}{\sqrt{\beta}}\mathbf{d}\end{bmatrix}. \tag{19}$$

The *ideal* preconditioners for the above system are

$$\mathscr{P}_0 = \begin{bmatrix} M & 0 \\ 0 & M+\beta K M^{-1}K\end{bmatrix} \text{ and } \mathscr{P}_1 = \begin{bmatrix} M+\beta K M^{-1}K & 0 \\ 0 & M\end{bmatrix}.$$

Define $\mathscr{P}_{1/2} = [\mathscr{P}_0, \mathscr{P}_1]_{1/2}$, namely,

$$\mathscr{P}_{1/2} = [\mathscr{P}_0, \mathscr{P}_1]_{1/2} \equiv \begin{bmatrix} [M, M+\beta K M^{-1}K]_{1/2} & 0 \\ 0 & [M+\beta K M^{-1}K, M]_{1/2}\end{bmatrix}. \tag{20}$$

Here we follow the notations from [19], $[T, R]_\theta = T^{1/2}(T^{-1/2}RT^{-1/2})^\theta T^{1/2}$ for some spd matrices T, R, and $\theta \in [0, 1]$.

Further, it can be observed, c.f. [19], that $[M, \beta K M^{-1}K]_{1/2} = \sqrt{\beta}K$. The above relations lead to the following preconditioner

$$\widehat{\mathscr{P}}_{1/2} = \begin{bmatrix} M+\sqrt{\beta}K & 0 \\ 0 & M+\sqrt{\beta}K\end{bmatrix}, \tag{21}$$

cf. [12, 20]. This preconditioner is equivalent to the preconditioner (18) obtained using standard norms.

4.1.4 Schur complement approximation

In [14, 16], for the saddle point system (15), the following preconditioners are proposed,

$$\widehat{\mathscr{P}}_{bd_1} = \begin{bmatrix} \widehat{M} & 0 & 0 \\ 0 & \beta\widehat{M} & 0 \\ 0 & 0 & \widehat{S}\end{bmatrix} \tag{22}$$

and

$$\widehat{\mathscr{P}}_{blt} = \begin{bmatrix} \widehat{M} & 0 & 0 \\ 0 & \beta\widehat{M} & 0 \\ K & -M & -\widehat{S}\end{bmatrix} \tag{23}$$

where $\widehat{M}$ is an approximation of the mass matrix M (via 20 Chebyshev iterations, see [14]) and $\widehat{S}$ is the Schur complement approximation.

The Schur complement approximation proposed in [16] is

$$\widehat{S}_1 = K M^{-1} K \tag{24}$$

and it is shown that the eigenvalues of $\widehat{S}_1^{-1} S$ are bounded as

$$\lambda\left(\widehat{S}_1^{-1} S\right) \in \left[\frac{1}{\beta} c\bar{h}^4 + 1, \frac{1}{\beta}\bar{C} + 1\right]$$

where $\bar{c}$ and $\bar{C}$ are real positive constants independent of the mesh size h. Clearly, this approximation suffers from convergence rate deterioration as β becomes smaller. In order to remedy this, an improvement to the approximation of S is proposed in [14], namely,

$$\widehat{S}_2 = \left(K + \frac{1}{\sqrt{\beta}} M\right) M^{-1} \left(K + \frac{1}{\sqrt{\beta}} M\right). \tag{25}$$

The approximation $\widehat{S}_2$ has been proved to be both h- and β-independent, satisfying

$$\lambda\left(\widehat{S}_2^{-1} S\right) \in \left[\frac{1}{2}, 1\right].$$

This follows easily since $\widehat{S}_2 = S + \frac{1}{\sqrt{\beta}}(K + K^T)$, where $S = K M^{-1} K + \frac{1}{\beta} M$. The latter implies that

$$M^{-1/2} \widehat{S}_2 M^{-1/2} = \widetilde{K}^2 + \frac{1}{\beta} I + \frac{1}{\sqrt{\beta}}(\widetilde{K} + I) \quad \text{and} \quad M^{-1/2} S M^{-1/2} = \widetilde{K}^2 + \frac{1}{\beta} I,$$

where $\widetilde{K} = M^{-1/2} K M^{-1/2}$.

For the reduced optimality system (16) the proposed preconditioner reads

$$\widehat{\mathscr{P}}_{bd_2} = \begin{bmatrix} \widehat{M} & 0 \\ 0 & \left(K + \frac{1}{\sqrt{\beta}} M\right) M^{-1} \left(K + \frac{1}{\sqrt{\beta}} M\right) \end{bmatrix}. \tag{26}$$

4.2 Preconditioners for the distributed optimal control problem constrained by the convection-diffusion equation

Following the discussion in [15], we consider in this paper only stabilization via LPS, as it results in optimality systems that have the same structure whether *optimize-then-discretize* or *discretize-then-optimize* is used. The resulting (non-compressed) system becomes

$$\mathscr{A}_F \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \lambda \end{bmatrix} \equiv \begin{bmatrix} M & 0 & F^T \\ 0 & \beta M & -M \\ F & -M & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \\ \mathbf{d} \end{bmatrix}. \tag{27}$$

The reduced optimality system for the problem is

$$\mathscr{A}_R \begin{bmatrix} \mathbf{y} \\ \lambda \end{bmatrix} \equiv \begin{bmatrix} M & F^T \\ F & -\frac{1}{\beta} M \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{d} \end{bmatrix}. \tag{28}$$

We now present an overview of different preconditioners available in literature for solving the saddle point systems (27) and (28).

4.2.1 (Negative) Schur complement approximation

In [15], to solve for the saddle point system (27), the following preconditioners are proposed,

$$\widetilde{\mathscr{P}}_{bd_1} = \begin{bmatrix} \widehat{M} & 0 & 0 \\ 0 & \beta\widehat{M} & 0 \\ 0 & 0 & \widehat{S} \end{bmatrix} \tag{29}$$

and

$$\widetilde{\mathscr{P}}_{blt} = \begin{bmatrix} \widehat{M} & 0 & 0 \\ 0 & \beta\widehat{M} & 0 \\ F & -M & -\widehat{S} \end{bmatrix} \tag{30}$$

where $\widehat{M}$ is the approximation of the mass matrix and $\widehat{S} = (F + \frac{1}{\sqrt{\beta}}M)M^{-1}(F + \frac{1}{\sqrt{\beta}}M)$ is the Schur complement approximation. The approximation $\widehat{S}$ is based on the extension of results proved in [14] with the spectral bound given by

$$\lambda\left(\widehat{S}^{-1}S\right) \in \left[\frac{1}{2}, 1\right]. \tag{31}$$

For the reduced optimality system (28), the block-diagonal preconditioner reads as follows, cf. [15],

$$\widetilde{\mathscr{P}}_{bd_2} = \begin{bmatrix} \widehat{M} & 0 \\ 0 & \left(F + \frac{1}{\sqrt{\beta}}M\right)M^{-1}\left(F + \frac{1}{\sqrt{\beta}}M\right) \end{bmatrix}, \tag{32}$$

4.2.2 Operator preconditioning with non-standard norms

Following the ideas from [19] as in (18), using non-standard norms we test also

$$\widetilde{\mathscr{P}}_{nsn} = \begin{bmatrix} M + \sqrt{\beta}F & \\ 0 & \frac{1}{\beta}\left(M + \sqrt{\beta}F\right) \end{bmatrix}. \tag{33}$$

4.3 The preconditioner from Section 2

We define our preconditioner, described in Section 2, in the current context of PDE-constrained optimization problems.

$$\widetilde{\mathscr{P}}_F = \begin{bmatrix} M & -\beta F^T \\ F & M + \sqrt{\beta}(F + F^T) \end{bmatrix}. \tag{34}$$

4.4 Block-diagonal and block counter-diagonal preconditioners involving only the mass matrix

All preconditioners presented so far involve both matrices M and K or M and F. With the aim of obtaining a cheaper preconditioner one can use a block-diagonal or block counter-diagonal preconditioner, such as $\mathscr{P}_D = \begin{bmatrix} M & 0 \\ 0 & M \end{bmatrix}$ for the matrix $\begin{bmatrix} M & -\beta K^T \\ K & M \end{bmatrix}$ and $\mathscr{P}_{BCD} = \begin{bmatrix} 0 & 0 & -M \\ 0 & M & 0 \\ -M & 0 & 0 \end{bmatrix}$ for the matrix $\mathscr{A}_* = \begin{bmatrix} \beta M & 0 & -M \\ 0 & M & K^T \\ -M & K & 0 \end{bmatrix}$, see [4] for the latter choice. The matrix $\mathscr{A}_*$ is obtained from (15) using permutations.

A computation shows that the eigenvalues of both preconditioned matrices cluster at unity when β is very small. However, the eigenvalues are complex and depend on $(\beta\mu_i)^{1/2}$ and $(\beta\mu_i)^{1/3}$, respectively, where $\{\mu_i\}$ is the set of eigenvalues of the matrix $BM^{-1}KM^{-1}K^T$. Hence, the methods are not robust with respect to the parameters h and β. Clearly β must be of order $O(h^4)$, to achieve a good clustering. To exemplify, for $h = 2^{-8}$ we would need $\beta \approx 10^{-10}$.

5 Numerical results

In this section we test and compare the numerical and computational efficiency of the various preconditioning techniques on the two benchmark PDE-constrained optimization problems. All results are obtained with C++ implemented code using the open source finite element library deal.ii [5]. Further, deal.ii provides interface to the Trilinos library [17], giving access to various data structures, iterative solution methods and preconditioners including an Algebraic Multigrid (AMG) solver. All experiments are performed on Intel(R) Core(TM) i5 CPU 750 @ 2.67GHz-2.80GHz with installed memory RAM of 4GB.

The results presented in the tables use the following conventions. For each value of β and h, we show the number of *outer* (MINRES or FGMRES) iterations in the first row. The adjacent brackets represent the average *inner* iterations for each *outer* iteration; the first number to the left shows the average number of AMG iterations and the number to the right shows the average number of Chebyshev semi-iterations. To further clarify, the presented number of Chebyshev iterations reflects the number of all occurrences of M in the corresponding preconditioner. The same holds for the average AMG iterations. The number below the first row represents the CPU times to solve the problem (in seconds).

Table 1 Problem 1, non-reduced system, preconditioner $\widehat{\mathscr{P}}_{bd_1}$, MINRES as outer solver

	β								
Size	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
3267	11(2+14)	13(2+15)	13(2+16)	13(2+18)	12(2+19)	12(2+20)	12(2+22)	10(2+24)	8(2+24)
	0.014	0.017	0.018	0.019	0.017	0.018	0.019	0.016	0.013
12675	11(2+12)	13(2+13)	13(2+14)	13(2+15)	12(2+17)	12(2+18)	12(2+19)	12(2+21)	10(2+23)
	0.046	0.054	0.056	0.059	0.057	0.057	0.059	0.06	0.054
49923	11(2+10)	13(2+11)	13(2+12)	13(2+13)	12(2+14)	11(2+16)	11(2+17)	11(2+18)	10(2+20)
	0.186	0.221	0.225	0.232	0.221	0.21	0.215	0.221	0.209
198147	11(2+9)	13(2+10)	13(2+10)	13(2+11)	12(2+12)	11(2+13)	11(2+15)	11(2+17)	9(2+18)
	0.798	0.95	0.956	0.97	0.932	0.875	0.907	0.944	0.803

5.1 Distributed optimal control problem constrained by the Poisson equation

Consider Problem 1 with $\Omega = [0, 1]^2$ and let $\widehat{y}$ be the desired state given by

$$\widehat{y} = \begin{cases} (2x_1 - 1)^2(2x_2 - 1)^2 & \text{if } \mathbf{x} \in \left[0, \frac{1}{2}\right]^2, \\ 0 & \text{otherwise.} \end{cases}$$

This problem is also considered in [8, 16]. Recall that numerically dealing with the distributed optimal control of the Poisson equation leads to the optimally system (15)

$$\begin{bmatrix} M & 0 & K^T \\ 0 & \beta M & -M \\ K & -M & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \\ \mathbf{d} \end{bmatrix} \tag{35}$$

Table 2 Problem 1: non-reduced system, preconditioner $\widehat{\mathscr{P}}_{blt_2}$, FGMRES as outer solver

	β								
Size	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
3267	13(2+10)	15(2+11)	15(2+13)	17(2+14)	15(2+14)	17(2+17)	17(2+19)	16(2+20)	13(2+21)
	0.033	0.02	0.021	0.033	0.021	0.034	0.035	0.026	0.022
12675	13(2+8)	15(2+9)	17(2+9)	18(2+11)	19(2+12)	17(2+13)	17(2+14)	18(2+16)	17(2+18)
	0.054	0.065	0.082	0.083	0.095	0.082	0.084	0.092	0.091
49923	16(2+6)	15(2+7)	16(2+8)	18(2+9)	19(2+10)	19(2+11)	17(2+11)	18(2+13)	18(2+15)
	0.269	0.255	0.278	0.319	0.357	0.352	0.318	0.346	0.362
198147	14(2+6)	16(2+6)	16(2+7)	16(2+6)	17(2+8)	17(2+8)	19(2+9)	19(2+10)	21(2+12)
	1.018	1.166	1.175	1.327	1.279	1.287	1.469	1.521	1.72

Table 3 Problem 1: reduced system, preconditioner $\widehat{\mathscr{P}}_{bd_2}$, MINRES as outer solver

	β								
Size	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
2178	11(2+8)	14(2+8)	15(2+9)	16(2+9)	17(2+11)	16(2+12)	13(2+13)	9(2+13)	5(2+14)
	0.008	0.01	0.01	0.011	0.013	0.012	0.01	0.007	0.008
8450	14(2+11)	14(2+11)	15(2+10)	16(2+10)	17(2+9)	17(2+10)	15(2+11)	13(2+12)	9(2+12)
	0.053	0.052	0.054	0.057	0.059	0.061	0.057	0.05	0.034
33282	11(2+11)	13(2+11)	15(2+10)	16(2+10)	17(2+9)	17(2+9)	15(2+9)	15(2+10)	12(2+11)
	0.167	0.191	0.217	0.228	0.239	0.235	0.21	0.217	0.18
132098	18(2+11)	15(2+11)	15(2+11)	16(2+10)	17(2+10)	17(2+9)	16(2+9)	15(2+8)	13(2+10)
	1.162	0.974	0.967	1.01	1.056	1.042	0.97	0.9	0.818

with its reduced form as in (16)

$$\begin{bmatrix} M & K^T \\ K & -\frac{1}{\beta}M \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{d} \end{bmatrix}. \tag{36}$$

The state y, the control u and the adjoint λ are all discretized using the **Q1** basis functions. We solve the problem with the *five* different preconditioners discussed earlier, i.e., $\widehat{\mathscr{P}}_{bd_1}$, $\widehat{\mathscr{P}}_{blt}$, $\widehat{\mathscr{P}}_{bd_2}$, $\widehat{\mathscr{P}}_{nsn}$, and $\widehat{\mathscr{P}}_F$. The relative convergence tolerance of the outer solver is set to 10^{-6} in the $L^2(\Omega)$ norm. The mass matrix is approximated using at most 20 Chebyshev semi-iterations with a relative convergence tolerance set to 10^{-4} in the $L^2(\Omega)$ norm. To approximate the blocks $M+\sqrt{\beta}K$ and $K+\frac{1}{\sqrt{\beta}}M$ we apply one V-cycle AMG iteration with two pre-smoothing and two post-smoothing steps by a symmetric Gauss-Seidel (smoother) method with a relative convergence tolerance set to 10^{-4} in the $L^2(\Omega)$ norm. In case of the block corresponding to the

Table 4 Problem 1: reduced system, preconditioner $\widehat{\mathscr{P}}_{nsn}$, MINRES as outer solver

	β								
Size	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
2178	12(2+0)	14(2+0)	14(2+0)	13(2+0)	12(2+0)	12(2+0)	11(2+0)	9(2+0)	7(2+0)
	0.005	0.006	0.006	0.005	0.005	0.005	0.005	0.004	0.003
8450	14(2+0)	14(2+0)	14(2+0)	14(2+0)	12(2+0)	12(2+0)	11(2+0)	11(2+0)	9(2+0)
	0.035	0.035	0.035	0.035	0.031	0.031	0.03	0.029	0.023
33282	12(2+0)	14(2+0)	14(2+0)	14(2+0)	13(2+0)	12(2+0)	12(2+0)	11(2+0)	11(2+0)
	0.125	0.144	0.144	0.145	0.134	0.125	0.125	0.115	0.114
132098	14(2+0)	14(2+0)	14(2+0)	14(2+0)	13(2+0)	12(2+0)	12(2+0)	11(2+0)	11(2+0)
	0.607	0.608	0.607	0.609	0.567	0.528	0.528	0.489	0.489

Table 5 Problem 1: reduced system, preconditioner $\widehat{\mathscr{P}}_F$, FGMRES as outer solver

	β								
Size	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
2178	6(2+0)	6(2+0)	7(2+0)	7(2+0)	7(2+0)	7(2+0)	6(2+0)	6(2+0)	4(2+0)
	0.003	0.003	0.004	0.004	0.004	0.004	0.003	0.007	0.002
8450	6(2+0)	7(2+0)	7(2+0)	7(2+0)	6(2+0)	6(2+0)	6(2+0)	6(2+0)	5(2+0)
	0.018	0.02	0.021	0.021	0.018	0.018	0.019	0.018	0.019
33282	5(2+0)	6(2+0)	6(2+0)	6(2+0)	6(2+0)	6(2+0)	6(2+0)	5(2+0)	5(2+0)
	0.061	0.072	0.072	0.072	0.072	0.072	0.072	0.061	0.061
132098	6(2+0)	6(2+0)	6(2+0)	6(2+0)	6(2+0)	6(2+0)	6(2+0)	5(2+0)	5(2+0)
	0.305	0.306	0.304	0.304	0.305	0.304	0.305	0.261	0.262

discrete differential operator $(K + \frac{1}{\sqrt{\beta}}M)M^{-1}(K + \frac{1}{\sqrt{\beta}}M)^T$, the transposed part $(K + \frac{1}{\sqrt{\beta}}M)^T$ is approximated similarly to $(K + \frac{1}{\sqrt{\beta}}M)$. The results are presented in Tables 1, 2, 3, 4 and 5.

Remark 2 The theory of the convergence of MINRES requires a fixed preconditioner, thus a fixed number of Chebyshev iterations has to be performed, otherwise the preconditioning becomes (slightly) variable. Despite of that we have used a variable number of Chebyshev iterations. Numerical tests, not reported here, show that the outer iterations remain the same or are decreased by one in a very few cases. The execution time, however exhibits a slight increase.

In Table 1, we present the results of preconditioning the saddle point system (35) with the block-diagonal preconditioner $\widehat{\mathscr{P}}_{bd_1}$ defined in (22).

In Table 2, we present the results of preconditioning (35) by $\widehat{\mathscr{P}}_{blt_2}$, defined as

$$\widehat{\mathscr{P}}_{blt_2} = \begin{bmatrix} \widehat{M} & 0 & 0 \\ 0 & \beta\widehat{M} & 0 \\ K & -M & -\left\{\left(K + \frac{1}{\sqrt{\beta}}M\right)M^{-1}\left(K + \frac{1}{\sqrt{\beta}}M\right)^T\right\} \end{bmatrix} \tag{37}$$

Table 3 shows the results of preconditioning the reduced optimality system (36) with the block-diagonal preconditioner $\widehat{\mathscr{P}}_{bd_2}$, defined in (26).

The results of preconditioning the reduced optimality system (36) by $\widehat{\mathscr{P}}_{nsn}$, defined in (18), are presented in Table 4.

The results, presented in Table 5, illustrate the performance of the preconditioner

$$\widehat{\mathscr{P}}_F = \begin{bmatrix} M & -\beta K^T \\ K & M + 2\sqrt{\beta}K \end{bmatrix}, \tag{38}$$

Table 6 Problem 1: cost functional $\mathscr{J}$ and related quantities

β	Iter	$\|u\|_2$	$\|y-\widehat{y}\|_2$	$\frac{\|y-\widehat{y}\|_2}{\|\widehat{y}\|_2}$	J	$\frac{\|b-\mathscr{A}x\|_2}{\|b\|_2}$	Time
2e−02	5	4.7e+0	3.96e−2	3.96e−1	2.25e−1	3.94e−7	0.011
2e−03	6	2.6e+1	2.87e−2	2.87e−1	6.70e−1	1.56e−6	0.012
2e−04	6	7.1e+1	1.42e−2	1.42e−1	5.01e−1	1.76e−6	0.012
2e−05	6	1.2e+2	4.55e−3	4.55e−2	1.51e−1	1.69e−6	0.012
2e−06	6	1.6e+2	1.22e−3	1.22e−2	2.49e−2	1.74e−6	0.012
2e−07	6	1.8e+2	3.09e−4	3.08e−3	3.20e−3	1.68e−6	0.012
2e−08	6	1.9e+2	8.32e−5	8.32e−4	3.68e−4	1.48e−6	0.012
2e−09	6	2.0e+2	3.95e−5	3.94e−4	4.06e−5	1.08e−6	0.012
2e−10	5	2.1e+2	3.60e−5	3.60e−4	4.36e−6	2.87e−6	0.010

applied to the transformed system $\widetilde{\mathscr{A}}_{RT} = \begin{bmatrix} M & -\beta K^T \\ K & M \end{bmatrix}$. We use Algorithm (1) to solve the system, which requires one AMG solve to approximate the block $M+\sqrt{\beta}K$ twice.

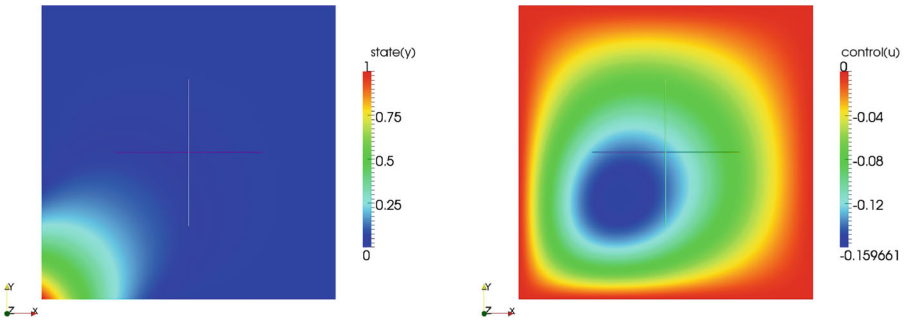
5.2 Discussion

The regularization parameter β determines how close the state y approaches the desired state $\widehat{y}$. We now illustrate the behavior of the cost functional $\mathscr{J}$ for different values of β. For this purpose, we reproduce Table 6 as in [8], with slight modifications. We observe that as β decreases, $y \to \widehat{y}$. Another observation is that $\|u\|$ increases with decreasing β. As commented in [8], otherwise the cost functional $\mathscr{J}$ becomes more insensitive to $\|u\|$ as β becomes small.

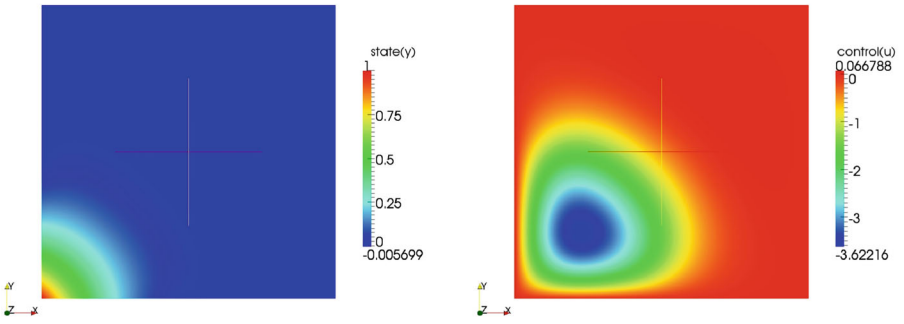
Table 6 is produced using the preconditioner $\widehat{\mathscr{P}}_F$. For mesh size 2^{-6}, we show the number of *outer* FGMRES iterations represented as "iter". $\|u\|$ represents the $L^2(\Omega)$ norm of the control u, $\|y-\widehat{y}\|$ measures how closely the state y matches the desired state $\widehat{y}$ and $\|y-\widehat{y}\|/\|\widehat{y}\|$ measures the relative error. $\mathscr{J}$ is the calculated cost functional.[1] The ratio $\|b-\mathscr{A}x\|/\|b\|$ represents the relative residual norm of the KKT system to show that the system has converged in the $L^2(\Omega)$ norm. Finally, the last column shows the time (sec) required to solve the system. Moreover, observing the iteration count presented in Tables 1–5, it is clear that all five preconditioners are robust with respect to mesh size h and the regularization parameter β.

We observe that the preconditioner $\widehat{\mathscr{P}}_F$ is the most efficient compared to the others tested. The preconditioner $\widehat{\mathscr{P}}_{nsn}$ takes the 2nd place. The results for preconditioners $\widehat{\mathscr{P}}_{bd_1}$ and $\widehat{\mathscr{P}}_{blt}$ are in line with what is obtained in [14]. The preconditioner $\widehat{\mathscr{P}}_{bd_2}$ being the reduced version of $\widehat{\mathscr{P}}_{bd_1}$, does not however appear to perform any better.

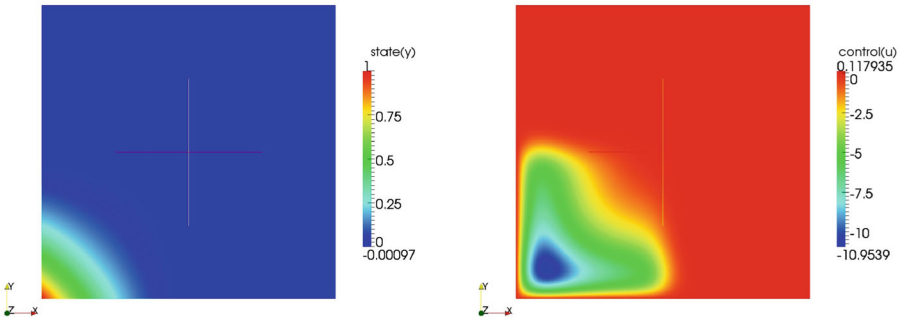
[1] The differences between the cost functionals using all other preconditioners are insignificant.



(a) $\beta = 2 \times 10^{-2}$ (b) $\mathbf{u}, \beta = 2 \times 10^{-2}$



(c) $\beta = 2 \times 10^{-4}$ (d) $\beta = 2 \times 10^{-4}$



(e) $\beta = 2 \times 10^{-6}$ (f) $\beta = 2 \times 10^{-6}$

Fig. 2 State (y) and control (u) distribution for different values of β

The CPU time (sec) required to solve the relevant saddle point systems for various values of β shows that the preconditioner $\widehat{\mathscr{P}}_F$ appears to be the most efficient, followed by the preconditioner $\widehat{\mathscr{P}}_{nsn}$. Moreover, the preconditioner $\widehat{\mathscr{P}}_{bd_2}$ for the reduced optimality system performs better compared to the preconditioners $\widehat{\mathscr{P}}_{bd_1}$ and $\widehat{\mathscr{P}}_{blt}$ for the full optimality system. Thus, reducing the optimality system shows clear advantage in decreasing the amount of time needed to solve the problem.

Table 7 Problem 2: non-reduced system, preconditioner $\widetilde{\mathscr{P}}_{bd_1}$

	β								
Size	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
$\varepsilon = 1/500$									
3267	23(2+17)	21(2+16)	24(2+20)	20(2+22)	16(2+23)	12(2+23)	9(2+23)	8(2+24)	8(2+25)
	0.031	0.026	0.033	0.028	0.022	0.016	0.012	0.011	0.011
12675	29(2+15)	27(2+14)	23(2+17)	19(2+20)	20(2+22)	14(2+22)	11(2+22)	10(2+22)	8(2+23)
	0.129	0.116	0.103	0.088	0.097	0.067	0.053	0.049	0.04
49923	37(2+13)	33(2+13)	25(2+16)	19(2+16)	19(2+20)	18(2+21)	14(2+21)	10(2+21)	9(2+22)
	0.725	0.63	0.486	0.372	0.397	0.385	0.296	0.215	0.196
198147	48(2+11)	46(2+11)	29(2+12)	19(2+14)	19(2+17)	17(2+19)	15(2+20)	12(2+21)	9(2+20)
	3.934	3.773	2.472	1.655	1.79	1.667	1.485	1.226	0.937
$\varepsilon = 1/1500$									
3267	29(2+17)	28(2+17)	28(2+20)	22(2+22)	16(2+23)	12(2+23)	10(2+24)	8(2+24)	7(2+24)
	0.04	0.038	0.04	0.03	0.022	0.016	0.014	0.011	0.01
12675	46(2+16)	42(2+16)	36(2+19)	24(2+22)	18(2+22)	12(2+22)	10(2+23)	8(2+23)	8(2+24)
	0.213	0.19	0.18	0.116	0.086	0.057	0.048	0.039	0.04
49923	54(2+14)	52(2+14)	46(2+17)	30(2+21)	19(2+21)	16(2+22)	12(2+22)	10(2+22)	8(2+23)
	0.993	0.966	0.889	0.661	0.398	0.335	0.253	0.214	0.174
198147	56(2+12)	80(2+12)	49(2+14)	25(2+16)	21(2+19)	17(2+21)	14(2+21)	11(2+22)	9(2+22)
	4.544	6.494	4.145	2.187	1.991	1.648	1.365	1.105	0.918

Finally, using the preconditioner $\widehat{\mathscr{P}_F}$ to solve the problem, we reproduce the plots for the state y and the control u for various values of β, as obtained in [8].

Figure 2a, c and e represent the state $\mathbf{y}$ of the system while Fig. 2b, d and f represent the control $\mathbf{u}$.

5.3 Distributed optimal control problem constrained by the convection-diffusion equation

Consider Problem 2, where $\Omega = [0,1]^2$, $\mathbf{w}$ represents divergence-free wind, and $\varepsilon > 0$ represents viscosity. We choose $\mathbf{w} = [cos\theta, sin\theta]$ for $\theta = \frac{\pi}{4}$ so that the maximum value of $\|w\|_2$ is equal to 1 on Ω. Further, $\widehat{y}$ is the desired state given by

$$\widehat{y} = \begin{cases} (2x_1 - 1)^2(2x_2 - 1)^2 \text{ if } x \in \left[0, \frac{1}{2}\right], \\ 0 \qquad\qquad\qquad\qquad \text{otherwise.} \end{cases} \tag{39}$$

Table 8 Problem 2: non-reduced system, preconditioner $\widetilde{\mathscr{P}}_{blt}$

	β								
Size	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
$\varepsilon = 1/500$									
3267	23(2+16)	21(2+15)	21(2+19)	20(2+22)	16(2+23)	12(2+23)	9(2+23)	8(2+23)	8(2+24)
	0.03	0.026	0.029	0.029	0.023	0.017	0.013	0.012	0.012
12675	29(2+14)	27(2+13)	23(2+17)	19(2+19)	18(2+21)	14(2+22)	11(2+22)	9(2+22)	8(2+23)
	0.133	0.116	0.107	0.091	0.089	0.07	0.055	0.046	0.042
49923	38(2+13)	33(2+11)	25(2+14)	19(2+16)	17(2+19)	16(2+20)	13(2+21)	11(2+22)	9(2+22)
	0.729	0.614	0.488	0.382	0.359	0.344	0.283	0.245	0.202
198147	48(2+10)	43(2+8)	27(2+11)	19(2+14)	17(2+16)	17(2+19)	15(2+20)	11(2+20)	11(2+21)
	3.959	3.435	2.333	1.682	1.584	1.666	1.521	1.099	1.137
$\varepsilon = 1/1500$									
3267	29(2+17)	28(2+16)	28(2+21)	20(2+22)	14(2+23)	12(2+23)	10(2+24)	8(2+24)	6(2+24)
	0.05	0.038	0.041	0.028	0.019	0.017	0.014	0.011	0.009
12675	46(2+17)	37(2+15)	34(2+19)	24(2+22)	18(2+21)	12(2+22)	10(2+23)	8(2+23)	8(2+24)
	0.207	0.172	0.175	0.12	0.088	0.059	0.05	0.041	0.041
49923	52(2+14)	52(2+12)	37(2+15)	23(2+18)	20(2+20)	16(2+21)	12(2+22)	10(2+22)	8(2+22)
	0.988	0.957	0.719	0.472	0.422	0.341	0.258	0.219	0.178
198147	56(2+12)	80(2+10)	49(2+11)	27(2+15)	21(2+20)	17(2+20)	14(2+21)	11(2+22)	9(2+22)
	4.667	6.491	4.057	2.421	2.02	1.651	1.382	1.091	0.908

This problem is also considered in (Chapter 6, [16]). As already stated, discretizing and using LPS leads to the optimality systems A_F and A_R as in (27).

$$\begin{bmatrix} M & 0 & F^T \\ 0 & \beta M & -M \\ F & -M & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \\ \mathbf{d} \end{bmatrix} \tag{40}$$

with its reduced form given by

$$\begin{bmatrix} M & F^T \\ F & -\frac{1}{\beta} M \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{d} \end{bmatrix}, \tag{41}$$

where F is a non-symmetric block. The state y, control u and the adjoint λ are discretized using **Q1** basis functions. We solve the problem with the five different preconditioners discussed earlier, i.e., $\widetilde{\mathscr{P}}_{bd_1}$, $\widetilde{\mathscr{P}}_{blt}$, $\widetilde{\mathscr{P}}_{bd_2}$, $\widetilde{\mathscr{P}}_{nsn}$, and $\widetilde{\mathscr{P}}_F$. Solving for the optimality systems (40) and (41) using the LPS scheme is discussed in Appendix A . The relative convergence tolerance of the outer solver (FGMRES) is set to 10^{-6} in the $L^2(\Omega)$ norm. Each operation on the mass matrix is approximated using at most 20 Chebyshev semi-iterations with a relative convergence tolerance set to 10^{-4} in

Table 9 Problem 2: reduced system, preconditioner $\widetilde{\mathscr{P}}_{bd_2}$

	β								
Size	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
$\varepsilon = 1/500$									
2178	24(2+7)	37(2+7)	34(2+8)	23(2+9)	14(2+10)	9(2+10)	7(2+10)	5(2+11)	5(2+11)
	0.022	0.037	0.036	0.022	0.013	0.008	0.007	0.005	0.005
8450	32(2+6)	47(2+5)	42(2+7)	27(2+8)	20(2+9)	12(2+9)	8(2+9)	7(2+10)	5(2+11)
	0.097	0.132	0.124	0.083	0.062	0.037	0.025	0.023	0.017
33282	40(2+5)	70(2+5)	51(2+5)	39(2+7)	24(2+7)	17(2+8)	11(2+8)	8(2+9)	7(2+9)
	0.534	0.921	0.678	0.544	0.338	0.245	0.158	0.118	0.106
132098	48(2+4)	105(2+4)	73(2+5)	42(2+5)	28(2+6)	21(2+7)	15(2+8)	10(2+8)	8(2+8)
	2.751	6.015	4.278	2.544	1.755	1.336	0.972	0.661	0.545
$\varepsilon = 1/1500$									
2178	34(2+8)	51(2+7)	40(2+9)	21(2+10)	12(2+10)	8(2+10)	6(2+11)	5(2+12)	5(2+12)
	0.034	0.048	0.041	0.02	0.011	0.007	0.006	0.005	0.005
8450	46(2+7)	78(2+6)	54(2+8)	28(2+8)	16(2+9)	9(2+10)	8(2+10)	5(2+10)	5(2+11)
	0.136	0.228	0.165	0.089	0.049	0.028	0.025	0.017	0.017
33282	53(2+6)	142(2+5)	105(2+6)	43(2+7)	22(2+8)	13(2+9)	8(2+9)	7(2+9)	6(2+10)
	0.719	1.913	1.427	0.607	0.313	0.19	0.119	0.106	0.094
132098	57(2+5)	200(2+5)	168(2+5)	58(2+6)	29(2+7)	18(2+8)	11(2+9)	8(2+9)	7(2+9)
	3.396	11.876	9.923	3.538	1.829	1.177	0.735	0.549	0.495

the $L^2(\Omega)$ norm. To approximate each block corresponding to higher order discrete differential operators $M + \sqrt{\beta}F$ and $F + \frac{1}{\sqrt{\beta}}M$ we use again a V-cycle AMG-preconditioned FGMRES with two pre-smoothing and two post-smoothing steps by a block Gauss-Seidel (smoother) method with a relative convergence tolerance set to 10^{-4} in the $L^2(\Omega)$ norm. In case of the block corresponding to the discrete differential operator $(F + \frac{1}{\sqrt{\beta}}M)M^{-1}(F + \frac{1}{\sqrt{\beta}}M)^T$, the transpose part $(F + \frac{1}{\sqrt{\beta}}M)^T$ is approximated analogously to $(F + \frac{1}{\sqrt{\beta}}M)$. The results[2] are presented in Tables 7, 8, 9, 10 and 11.

We present results for $\varepsilon = 1/500$ and for the particularly convection-dominated case, $\varepsilon = 1/1500$. We find that all five preconditioners are also robust to smaller values of ε. We again find $\widetilde{\mathscr{P}}_F$ to be the best in terms of parameter robustness, iteration count and run time requirements.

[2]The LPS scheme requires computing the approximate solutions $\tilde{\mathbf{y}}$ and $\tilde{\boldsymbol{\lambda}}$. In our numerical experiments, we use one outer iteration for this purpose, but it is not accounted for in the iteration count presented in Tables 7–11.

Table 10 Problem 2: reduced system, preconditioner $\widetilde{\mathscr{P}}_{nsn}$

	β								
Size	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
$\varepsilon = 1/500$									
2178	24(2+0)	22(2+0)	22(2+0)	20(2+0)	16(2+0)	10(2+0)	8(2+0)	6(2+0)	6(2+0)
	0.016	0.014	0.014	0.012	0.01	0.006	0.005	0.004	0.004
8450	31(2+0)	28(2+0)	26(2+0)	22(2+0)	18(2+0)	14(2+0)	10(2+0)	8(2+0)	6(2+0)
	0.079	0.068	0.062	0.051	0.041	0.032	0.023	0.019	0.014
33282	39(2+0)	37(2+0)	31(2+0)	23(2+0)	20(2+0)	16(2+0)	12(2+0)	8(2+0)	6(2+0)
	0.424	0.404	0.347	0.248	0.214	0.17	0.129	0.088	0.068
132098	45(2+0)	47(2+0)	35(2+0)	25(2+0)	21(2+0)	19(2+0)	16(2+0)	10(2+0)	8(2+0)
	2.157	2.251	1.705	1.197	1.023	0.906	0.762	0.484	0.395
$\varepsilon = 1/1500$									
2178	31(2+0)	26(2+0)	26(2+0)	20(2+0)	14(2+0)	10(2+0)	7(2+0)	6(2+0)	5(2+0)
	0.023	0.018	0.018	0.012	0.008	0.006	0.004	0.004	0.003
8450	44(2+0)	39(2+0)	33(2+0)	24(2+0)	18(2+0)	12(2+0)	8(2+0)	6(2+0)	6(2+0)
	0.103	0.095	0.083	0.057	0.041	0.027	0.019	0.015	0.015
33282	51(2+0)	53(2+0)	45(2+0)	28(2+0)	22(2+0)	14(2+0)	10(2+0)	8(2+0)	6(2+0)
	0.55	0.573	0.486	0.306	0.236	0.149	0.108	0.088	0.068
132098	55(2+0)	80(2+0)	62(2+0)	35(2+0)	24(2+0)	18(2+0)	12(2+0)	8(2+0)	6(2+0)
	2.641	3.843	3.016	1.705	1.151	0.857	0.575	0.395	0.308

In Table 7 we present the results of preconditioning the saddle point system (40) with the block-diagonal preconditioner $\widetilde{\mathscr{P}}_{bd_1}$, defined in (29).

In Table 8, we present the results of preconditioning the saddle point system (40) with the block lower-triangular preconditioner $\widetilde{\mathscr{P}}_{blt}$ from (30).

In Table 9 we present the results of preconditioning the reduced optimality system (41) with the block-diagonal preconditioner $\widetilde{\mathscr{P}}_{bd_2}$ from (32).

Next, in Table 10, we present the results of preconditioning the reduced optimality system (41) with the block-diagonal preconditioner $\widetilde{\mathscr{P}}_{nsn}$ from (33).

Finally, in Table 11, we show the performance of the preconditioner $\widetilde{\mathscr{P}}_F$ from (34), applied to the transformed system $\begin{bmatrix} M & -\beta F^T \\ F & M \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \tilde{\boldsymbol{\lambda}} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{d} \end{bmatrix}$. To solve systems with $\widetilde{\mathscr{P}}_F$, we use Algorithm 1.

5.4 Discussion

From the iteration counts in Tables 7–11, it is clear that all these preconditioners are robust with respect to mesh size h and the regularization parameter β.

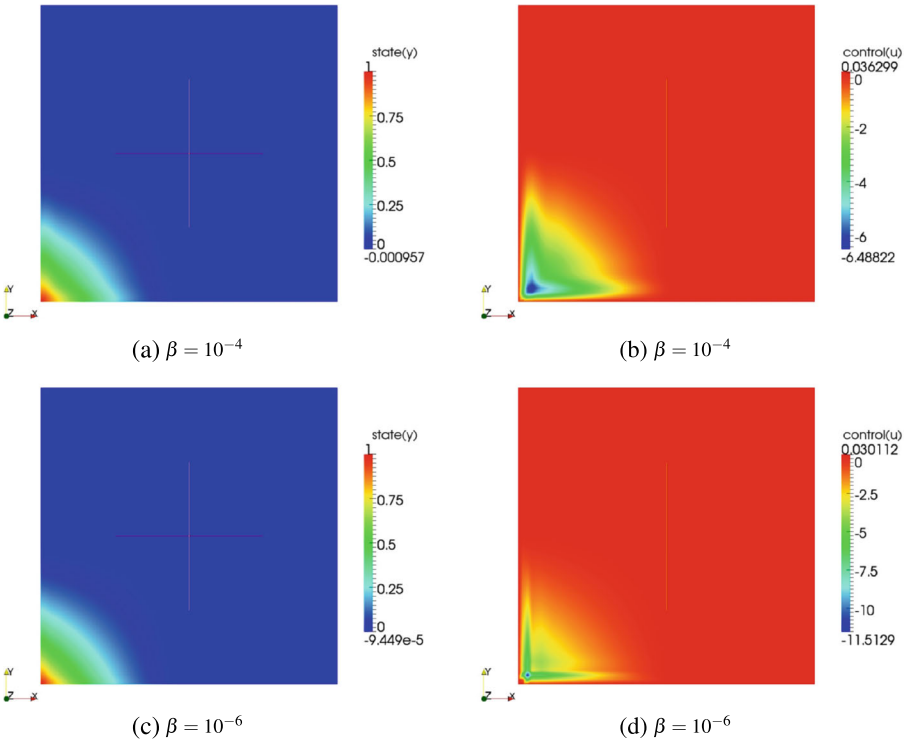
Table 11 Problem 2: reduced system, preconditioner $\widetilde{\mathscr{P}}_F$

	β								
Size	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
$\varepsilon = 1/500$									
2178	11(2+0)	11(2+0)	11(2+0)	9(2+0)	7(2+0)	5(2+0)	3(2+0)	3(2+0)	2(3+0)
	0.007	0.007	0.007	0.005	0.004	0.003	0.002	0.002	0.002
8450	14(2+0)	14(2+0)	12(2+0)	9(2+0)	8(2+0)	5(2+0)	4(2+0)	3(2+0)	2(3+0)
	0.032	0.03	0.025	0.019	0.017	0.012	0.01	0.008	0.006
33282	17(2+0)	16(2+0)	13(2+0)	10(2+0)	8(2+0)	6(2+0)	4(2+0)	3(2+0)	3(2+0)
	0.19	0.176	0.143	0.111	0.09	0.07	0.05	0.04	0.04
132098	19(2+0)	18(2+0)	15(2+0)	10(2+0)	8(2+0)	7(2+0)	5(2+0)	4(2+0)	3(2+0)
	0.946	0.888	0.742	0.504	0.42	0.364	0.272	0.227	0.182
$\varepsilon = 1/1500$									
2178	15(2+0)	14(2+0)	13(2+0)	9(2+0)	6(2+0)	4(2+0)	3(2+0)	3(2+0)	2(3+0)
	0.01	0.009	0.008	0.005	0.004	0.003	0.002	0.002	0.002
8450	19(2+0)	17(2+0)	15(2+0)	10(2+0)	7(2+0)	5(2+0)	3(2+0)	3(2+0)	2(3+0)
	0.047	0.036	0.032	0.022	0.016	0.012	0.008	0.008	0.006
33282	21(2+0)	23(2+0)	20(2+0)	12(2+0)	8(2+0)	6(2+0)	4(2+0)	3(2+0)	2(3+0)
	0.235	0.26	0.219	0.132	0.09	0.07	0.05	0.04	0.031
132098	22(2+0)	32(2+0)	27(2+0)	14(2+0)	9(2+0)	7(2+0)	5(2+0)	3(2+0)	3(2+0)
	1.097	1.645	1.343	0.693	0.455	0.363	0.274	0.182	0.182

We now illustrate the behavior of the cost functional $\mathscr{J}$ for different values of β. Recall that how closely the state y approaches the desired state $\widehat{y}$ is determined by the regularization parameter β. However we observe (across all tested preconditioners)

Table 12 Problem 2: the cost functional $\mathscr{J}$ and related quantities

β	Iter	$\|u\|_2$	$\|y-\widehat{y}\|_2$	$\frac{\|y-\widehat{y}\|_2}{\|\widehat{y}\|_2}$	J	$\frac{\|b-\mathscr{A}x\|_2}{\|b\|_2}$	Time
1e−02	14	5.79e+1	6.98e−2	6.98e−1	1.67e+1	4.40e−8	0.032
1e−03	14	4.94e+1	1.09e−2	1.09e−1	1.22e+0	7.46e−9	0.030
1e−04	12	4.94e+1	1.98e−3	1.98e−2	1.22e−1	3.69e−8	0.025
1e−05	9	5.08e+1	3.77e−4	3.77e−3	1.29e−2	4.95e−8	0.019
1e−06	8	5.19e+1	6.61e−5	6.60e−4	1.34e−3	1.51e−8	0.017
1e−07	5	5.22e+1	3.62e−5	3.62e−4	1.36e−4	5.32e−8	0.012
1e−08	4	5.22e+1	3.61e−5	3.61e−4	1.36e−5	8.63e−9	0.010
1e−09	3	5.22e+1	3.61e−5	3.61e−4	1.36e−6	6.94e−9	0.008
1e−10	2	5.22e+1	3.61e−5	3.61e−4	1.37e−7	3.58e−8	0.006



(a) $\beta = 10^{-4}$ (b) $\beta = 10^{-4}$

(c) $\beta = 10^{-6}$ (d) $\beta = 10^{-6}$

Fig. 3 State (y) and control (u) distribution for different values of β

that $\|y - \widehat{y}\|$ stops to decrease any further around $\beta \leq 10^{-6}$. Further, we observe that $\|u\|$ stops increasing further around $\beta \leq 10^{-6}$. This indicates that the optimal value of β for the problem is around 10^{-6}.

Table 12 is produced using the preconditioner $\widetilde{\mathscr{P}}_F$. For mesh size 2^{-6}, we show the number of *outer* FGMRES iterations represented as "iter". $\|u\|$ represents the $L^2(\Omega)$ norm of the control u, $\|y - \widehat{y}\|$ measures how closely the state y matches the desired state $\widehat{y}$ and $\|y - \widehat{y}\|/\|\widehat{y}\|$ measures the relative error. $\mathscr{J}$ is the calculated cost functional, $\|b - \mathscr{A}x\|/\|b\|$ represents the residual norm of the KKT system of equations to show the system converged in the $L^2(\Omega)$ norm. Finally, the last column tells us the time (sec) it took to solve the system. The differences between the cost functionals using all other preconditioners are insignificant.

The comparison of the performance of the different preconditioners in terms of iterations required to solve the relevant saddle point systems for various values of β clearly shows that the preconditioner $\widetilde{\mathscr{P}}_F$ outperforms all the other four preconditioners.

The comparison of the performance of the five preconditioners in terms of the CPU time (sec) required to solve the relevant saddle point systems for various values of β shows a clear advantage of reducing the optimality system. Again, the preconditioner $\widetilde{\mathscr{P}}_F$ performs exceptionally well.

The results of our numerical experiments clearly indicate that the preconditioner $\widetilde{\mathscr{P}}_F$ is very effective for solving convection-diffusion control problems. Further, it is shown that this preconditioner is also robust to the mesh size h and the regularization parameter β.

Finally, we produce the plots for the state y and the control u for two different values of β to give a visual clue to the solutions obtained. The mesh size is set to $h = 2^{-6}$ and we use the preconditioner $\widetilde{\mathscr{P}}_F$ to solve the problem.

Figure 3a and c represent the state y of the system while Fig. 3b and f represent the control u.

6 Conclusions

All of the tested methods have the important property of robust performance with respect to the meshsize h and the optimality control parameter β. The most efficient of the preconditioning techniques are $\mathscr{P}_{nsn}$ and $\mathscr{P}_F$. It has been proven that both lead to a condition number of the corresponding preconditioned matrix, bounded by 2. However, for $\mathscr{P}_{nsn}$ it holds for the square of the preconditioned matrix, which is indefinite. Therefore it needs about twice the number of iterations, compared to the $\mathscr{P}_F$-preconditioned method.

Acknowledgments The work of the first author is supported by the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070). The comments of the anonymous reviewers are also gratefully acknowledged.

Appendix A: Some details regarding the local projection scheme and its implementation in `deal.ii`

We include the details below in order to ensure reproducibility of the numerical experiments, presented in this paper.

We follow the discussion on the LPS scheme in [6]. Consider a two-dimensional mesh $\mathscr{T}_H$ consisting of open cells, in our case quadrilaterals.

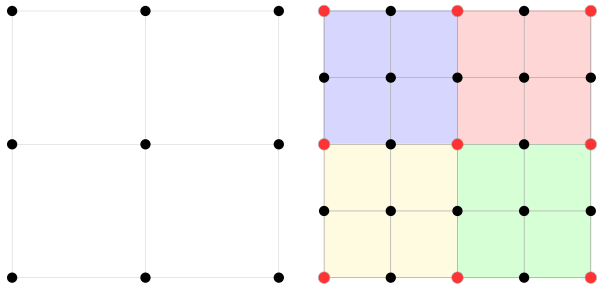


Fig. 4 $\mathscr{T}_H$ with 4 cells (*left*) and refined mesh $\mathscr{T}_h$ with 16 cells and 4 patches (each patch portrayed by a different color) with ● representing the support on the patches (*right*)

A uniform refinement of the mesh $\mathscr{T}_H$ gives us a refined mesh $\mathscr{T}_h$, where each refined cell on $\mathscr{T}_H$ creates four cells in $\mathscr{T}_h$, referred to as a patch $\mathbf{P}$; so we will have four patches in this case.

LPS uses standard finite element discretization with stabilization based on local projections. Next we define an $L^2(\mathbf{P})$ orthogonal projection operator. (Note that an orthogonal projection operator gives a good average approximation of a function, compared to an interpolation operator. However, both these operators have difficulties in approximating highly oscillatory or discontinuous functions.) Let

$$P_h : L^2(\Omega) \to V_H^{const},$$

on the patches of the domain, with V_H^{const} being a cell-wise[3] constant function on the patches. Further, this operator satisfies the following approximation and stability properties on the patches, c.f. [6]:

$$\|P_h v\|_{L^2(\mathbf{P})} \leq c\|v\|_{L^2(\mathbf{P})} \qquad \forall v \in L^2(\mathbf{P}).$$

$$\|v - P_h v\|_{L^2(\mathbf{P})} \leq ch\|\nabla v\|_{L^2(\mathbf{P})} \qquad \forall v \in H^1(\mathbf{P}).$$

We now introduce a positive stabilization parameter δ associated with a bilinear symmetric stabilization form $\tau_h^\delta : V_h \times V_h \to \mathbb{R}$ given by

$$\tau_h^\delta(u_h, v_h) = \delta(\mathbf{w} \cdot \nabla u_h - P_h(\mathbf{w} \cdot \nabla u_h) \times (\mathbf{w} \cdot \nabla v_h - P_h(\mathbf{w} \cdot \nabla v_h))).$$

In terms of finite element basis functions we write

$$T = \{\tau_{h,i,j}^\delta\}_{i,j=1,\ldots,n}, \quad \tau_{h,i,j}^\delta = \delta \int_\Omega (\mathbf{w}\cdot\nabla\phi_i - P_h(\mathbf{w}\cdot\nabla\phi_i)\times(\mathbf{w}\cdot\nabla\phi_j - P_h(\mathbf{w}\cdot\nabla\phi_j))).$$

Consider the solution of the following convection-diffusion equation

$$\begin{aligned} -\varepsilon\Delta u + (\mathbf{w} \cdot \nabla)u &= f \quad \text{in } \Omega \\ u &= g \text{ on } \partial\Omega, \end{aligned}$$

where g is continuous and $\mathbf{w} = [sin\theta, cos\theta]$.

The stabilization parameter δ_k is defined locally on individual elements and depends on the (local) Peclet number

$$P_\varepsilon^k = \frac{h_k\|\mathbf{w}\|}{\varepsilon}.$$

The parameter δ_k is given by

$$\delta_k = \begin{cases} \dfrac{h_k}{\|\mathbf{w}\|}, & \text{if } P_\varepsilon^k \geq 1, \\ 0, & \text{otherwise.} \end{cases}$$

The discretization of the convection-diffusion equation with LPS scheme leads to

$$\varepsilon(\nabla u_h, \nabla v_h) + (\mathbf{w}.\nabla u_h, v_h) + \delta_k((I - P_h)\mathbf{w}.\nabla u_h, \mathbf{w}.\nabla v_h) = f$$

or

$$\varepsilon(\nabla u_h, \nabla v_h) + (\mathbf{w}.\nabla u_h, v_h) + \delta_k(\mathbf{w}.\nabla u_h, \mathbf{w}.\nabla v_h) = f + \delta_k P_h(\mathbf{w}.\nabla u_h, \mathbf{w}.\nabla v_h).$$

[3]There are four cells in a patch in our considered example.

We now split the above equation as follows

$$\varepsilon(\nabla u_h, \nabla v_h) + \delta_k(\mathbf{w}.\nabla u_h, \mathbf{w}.\nabla v_h) + (\mathbf{w}.\nabla u_h, v_h) = f + \delta_k \xi_h$$

$$P_h(\mathbf{w}.\nabla u_h, v_h)\xi_H = \delta_k \mathbf{w}.(\nabla u_h, v_h),$$

where ξ_H is a cell-wise constant function of patches and ξ_h is the interpolation of ξ_H to $\mathscr{T}_h$.

The following algorithm solves the discrete convection-diffusion equation using the LPS scheme.

1. Set $\xi_h = 0$.
2. Assemble (42).
3. Compute $\tilde{u}_h$ solving $\varepsilon(\nabla u_h, \nabla v_h) + \delta_k(\mathbf{w}.\nabla u_h, \mathbf{w}.\nabla v_h)(\mathbf{w}.\nabla u_h, v_h) = f + \xi_h$ using one iteration of FGMRES.
4. Interpolate $\tilde{u}_h$ to $\tilde{u}_H$. This takes the nodes ● (see Fig. 4) from the fine mesh $\mathscr{T}_h$ to the coarse mesh $\mathscr{T}_H$.
5. Compute gradient of $\tilde{u}_H$, i.e., $\nabla \tilde{u}_H$.
6. Assemble (42), i.e., $M_{P_h}\xi_H = \delta(\mathbf{w}.\nabla \tilde{u}_H, v_H)$ and solve for ξ_H.
7. Interpolate ξ_H back to ξ_h. This replaces the nodes ● (see Fig. 4) on the fine mesh $\mathscr{T}_h$ with the values ξ_H computed on the coarse mesh $\mathscr{T}_H$.
8. Assemble (42) and solve using FGMRES to convergence.

References

1. Axelsson, O., Boyanova, P., Kronbichler, M., Neytcheva, M., Wu, X.: Numerical and computational efficiency of solvers for two-phase problems. Comput. Math. Appl. **65**, 301–314 (2013)
2. Axelsson, O., Kucherov, A.: Real valued iterative methods for solving complex symmetric linear systems. Numer. Linear Algebra Appl. **7**, 197–218 (2000)
3. Axelsson, O., Neytcheva, M., Bashir Ahmad, A.: Comparison of iterative methods to solve complex valued linear algebraic systems. Numer. Alg. **66**, 811–841 (2014)
4. Bai, Z.-Z.: Block preconditioners for elliptic PDE-constrained optimization problems. Computing **91**, 379–395 (2011)
5. Bangerth, W., Hartmann, R., Kanschat, G.: deal.II-a general-purpose object-oriented finite element library. ACM T. Math. Software **33** (2007). doi:10.1145/1268776.1268779. Art. 24
6. Becker, R., Vexler, B.: Optimal control of the convection-diffusion equation using stabilized finite element methods. Numer. Math. **106**, 349–367 (2007)
7. Boyanova, P., Do-Quang, M., Neytcheva, M.: Efficient preconditioners for large scale binary Cahn-Hilliard models. Comput. Methods Appl. Math. **12**, 1–22 (2012)
8. Choi, Y.: Simultaneous analysis and design in PDE-constrained optimization. Doctor of Philosophy Thesis, Stanford University (2012)
9. Greenbaum, A., Ptak, V., Strakos, Z.: Any convergence curve is possible for GMRES. SIAM Matrix Anal. Appl. **17**, 465–470 (1996)
10. Heinkenschloss, M., Leykekhman, D.: Local error estimates for SUPG solutions of advection-dominated elliptic linear-quadratic optimal control problems. SIAM J. Numer. Anal. **47**, 4607–4638 (2010)
11. Hinze, M., Pinnau, R., Ulbrich, M., Ulbrich, S.: Optimization with PDE constrains. Springer, Berlin Heidelberg New York (2009)
12. Kollmann, M.: Efficient iterative solvers for saddle point systems arising in PDE-constrained optimization problems with inequality constraints. Doctor of Philosophy Thesis, Johannes Kepler University Linz (2013)

13. Lions, J.-L.: Optimal Control of Systems Governed by Partial Differential Equations. Springer, Berlin (1971)
14. Pearson, J.W., Wathen, A.J.: A new approximation of the Schur complement in preconditioners for PDE-constrained optimization. Numer. Linear Alg. Appl. **19**, 816–829 (2012)
15. Pearson, J.W., Wathen, A.J.: Fast iterative solvers for convection-diffusion control problems. ETNA **40**, 294–310 (2013)
16. Rees, T.: Preconditioning iterative methods for PDE constrained optimization. Doctor of Philosophy Thesis, University of Oxford (2010)
17. The Trilinos Project http://trilinos.sandia.gov/
18. Tröltzsch, F.: Optimal control of partial differential equations: theory, methods and applications, AMS, Graduate Studies in Mathematics (2010)
19. Zulehner, W.: Nonstandard norms and robust estimates for saddle-point problems. SIAM J. Matrix Anal. Appl. **32**, 536–560 (2011)
20. Zulehner, W.: Efficient solvers for saddle point problems with applications to PDE-constrained optimization. Advanced Finite Element Methods and Applications, Lecture Notes in Applied and Computational Mechanics, vol. 66, pp. 197–216 (2013)