Received: 12 February 2019 | Revised: 29 October 2019 | Accepted: 5 November 2019

DOI: 10.1002/nla.2276

RESEARCH ARTICLE

WILEY

Interior-point methods and preconditioning for PDE-constrained optimization problems involving sparsity terms

John W. Pearson[1] | **Margherita Porcelli**[2] | **Martin Stoll**[3]

[1]School of Mathematics, The University of Edinburgh, Edinburgh, United Kingdom

[2]Dipartimento di Matematica, Università di Bologna, Bologna, Italy

[3]Professorship of Scientific Computing, Faculty of Mathematics, Technische Universität Chemnitz, Chemnitz, Germany

Correspondence
Margherita Porcelli, Dipartimento di Matematica, Università di Bologna, Piazza di Porta San Donato 5, 40126 Bologna, Italy.
Email: margherita.porcelli@unibo.it

Funding information
Engineering and Physical Sciences Research Council Fellowship, Grant/Award Number: EP/M018857/2; The Alan Turing Institute; DAAD-MIUR Joint Mobility Program 2018-2020, Grant/Award Number: 57396654; National Group of Computing Science (GNCS-INDAM)

Summary

Partial differential equation (PDE)–constrained optimization problems with control or state constraints are challenging from an analytical and numerical perspective. The combination of these constraints with a sparsity-promoting L^1 term within the objective function requires sophisticated optimization methods. We propose the use of an interior-point scheme applied to a smoothed reformulation of the discretized problem and illustrate that such a scheme exhibits robust performance with respect to parameter changes. To increase the potency of this method, we introduce fast and efficient preconditioners that enable us to solve problems from a number of PDE applications in low iteration numbers and CPU times, even when the parameters involved are altered dramatically.

KEYWORDS

box constraints, interior-point methods, PDE-constrained optimization, preconditioning, saddle-point systems, sparsity

JEL CLASSIFICATION

65F08, 65F10, 65K05, 76D55, 90C20, 93C20

1 | INTRODUCTION

In this paper, we address the challenge of solving matrix systems arising from PDE-constrained optimization problems.[1–3] Such formulations arise in a multitude of applications, ranging from the control of fluid flows[4] to image processing contexts.[5] The particular question considered in this paper is how to efficiently handle sparsity-promoting cost terms within the objective function, as well as additional constraints imposed on the control variable and even the state variable. In fact, seeking optimal control functions that are both contained within a range of function values and zero on large parts of the domain has become extremely relevant in practical applications.[6]

In detail, we commence by studying the problem of finding $(\mathrm{y},\mathrm{u}) \in H^1(\Omega) \times L^2(\Omega)$ such that the functional

$$\mathcal{F}(\mathrm{y},\mathrm{u}) = \frac{1}{2}\|\mathrm{y}-\mathrm{y}_\mathrm{d}\|^2_{L^2(\Omega)} + \frac{\alpha}{2}\|\mathrm{u}\|^2_{L^2(\Omega)} + \beta\|\mathrm{u}\|_{L^1(\Omega)} \tag{1}$$

https://doi.org/10.1002/nla.2276

is minimized subject to the partial differential equation (PDE) constraint

$$-\Delta \mathrm{y} = \mathrm{u} + \mathrm{f} \quad \text{in } \Omega, \tag{2}$$

$$\mathrm{y} = \mathrm{g} \qquad \text{on } \Gamma, \tag{3}$$

where we assume that Equation (2) is understood in the weak sense.[3] Here, $\Omega \subset \mathbb{R}^2$ or $\mathbb{R}^3$ denotes a spatial domain with boundary Γ. Additionally, we allow for box constraints on the control

$$\mathrm{u_a} \leq \mathrm{u} \leq \mathrm{u_b} \quad \text{a.e. in } \Omega \tag{4}$$

and, for the sake of generality, consider the possibility that there are also box constraints on the state

$$\mathrm{y_a} \leq \mathrm{y} \leq \mathrm{y_b} \quad \text{a.e. in } \Omega. \tag{5}$$

We follow the convention of recent numerical studies[7–10] and investigate the case where the lower (upper) bounds of the box constraints are nonpositive (nonnegative). Here, the functions $\mathrm{y_d}, \mathrm{f}, \mathrm{g}, \mathrm{u_a}, \mathrm{u_b}, \mathrm{y_a}, \mathrm{y_b} \in L^2(\Omega)$ are provided in the problem statement, with $\alpha, \beta > 0$ given problem-specific *regularization parameters*. The functions $\mathrm{y}, \mathrm{y_d}$, and u denote the state, the desired state, and the control, respectively. The state y and the control u are then linked via a state equation (the PDE). In this work, we examine several representative state equations, including Poisson's equation (2) as well as the convection–diffusion equation and the heat equation. Furthermore, we consider the case where the difference between state y and desired state $\mathrm{y_d}$ is only observed on a certain part of the domain, that is, over $\Omega_1 \subset \Omega$, with the first quadratic term in (1) then having the form $\frac{1}{2}\|\mathrm{y} - \mathrm{y_d}\|^2_{L^2(\Omega_1)}$. We refer to this case as the "partial observation" case.

There are many difficulties associated with problems (1)–(5), such as selecting a suitable discretization and choosing an efficient approach for handling the box constraints and the sparsity term. In particular, the state-constrained problem itself, not even including the L^1-norm term, leads to a problem formulation where the regularity of the Lagrange multiplier is reduced (see the work of Casas[11] for details). Additionally, the simultaneous treatment of control and state constraints is a complex task. For this, Günther et al.[12] propose the use of Moreau–Yosida regularization in order to add the state constraints as a penalty to the objective function. Other approaches are based on a semismooth Newton method (see, e.g., the works of Herzog and Sachs[13] and Porcelli et al.[14]). In fact, the inclusion of control/state constraints leads to a semismooth nonlinear formulation of the first-order optimality conditions.[15–17] Interestingly, the structure of the resulting nonlinear system is preserved if the L^1-norm penalization is added.[6,13,14] Therefore, its solution also generally relies on semismooth Newton approaches, and an infinite-dimensional formulation is commonly utilized to derive the first-order optimality system. Stadler[6] was the first to study PDE-constrained optimization problems that include an L^1 term by applying a semismooth Newton approach, and many contributions have been made to the study of these problems in recent years (cf. the works of Herzog et al.,[18,19] among others). Our objective is to tackle the coupled problem of both box constraints combined with the sparsity-promoting term, using the interior-point method (IPM).

The paper by Porcelli et al.[14] provides a complete analysis of a globally convergent semismooth Newton method proposed for problem (1)–(4). Theoretical and practical aspects are investigated for both the linear algebra phase and the convergence behavior of the nonlinear method. The numerical experiments carried out revealed a drawback of the method, as it exhibited poor convergence behavior for limiting values of the regularization parameter α.

The aim of this paper is to propose a new framework for the solution of (1)–(5) for a wider class of state equations and boundary conditions and, at the same time, attempt to overcome the numerical limitations of the global semismooth approach.

To pursue this issue, we utilize IPMs, which have shown great applicability for nonlinear programming problems[20,21] and have also found effective use within the PDE-constrained optimization framework.[22,23] In particular, IPMs for linear and (convex) quadratic programming problems display several features that make them particularly attractive for very large-scale optimization (see, e.g., the recent survey paper by Gondzio[24]). Their main advantages are undoubtedly their low-degree polynomial worst-case complexity and their ability to deliver optimal solutions in an almost-constant number of iterations that depends very little, if at all, on the problem dimension. This feature makes IPMs perfect candidates for huge-scale discretized PDE-constrained optimal control problems.

Recently, in the work of Pearson and Gondzio,[22] an interior-point approach has been successfully applied to the solution of problem (1)–(5), with $\beta = 0$. In this case, the discretization of the optimization problem leads to a convex quadratic programming problem, so IPMs may naturally be applied and indeed demonstrated very good convergence properties. Furthermore, the rich structure of the linear systems arising in this framework allows one to design efficient and robust preconditioners, based on those originally developed for the Poisson control problem without box constraints.[25]

In this work, we extend the approach proposed in the work of Pearson and Gondzio[22] to the more difficult and general case with $\beta > 0$ and apply it to several typical PDE-constrained optimal control problems. To implement an interior-point scheme for this problem, we utilize two key ingredients that will be described in detail in Section 3: an appropriate discretization of the L^1-norm that allows us to write the discretized problem in a matrix–vector form and a suitable smoothing of the resulting vector ℓ_1-norm that yields a final quadratic programming form of the discretized problem. The first ingredient is based on the discretization described in the work of Wachsmuth and Wachsmuth[10] and has recently been applied to problem (1)–(4) in the works of Song et al.,[7–9] where block-coordinate–like methods are then introduced. The second ingredient has been widely used for solving the ubiquitous L^1-norm–regularized quadratic problem as, for example, when computing sparse solutions in wavelet-based deconvolution problems and compressed sensing.[26] This strategy has been used to solve an ordinary differential equation (ODE)–constrained optimization problem from robotics in the work of Vossen and Maurer.[27] There, the authors tackle a problem arising from the discretization of the system of ODEs, solving this using the all-at-once interior-point solver IPOPT.[28] Given the moderate dimensions of the matrix systems arising from ODE problems, it is possible to apply direct solvers, although this approach is infeasible for the PDE setting we consider here. On the other hand, to our knowledge, the use of the smoothing technique for the L^1 term is new within the PDE applications considered here, and a careful derivation of solvers for the underlying matrix systems is necessary in this framework.

In order for this method to be computationally tractable for high-dimensional PDE applications, it is essential to devise potent numerical solvers for the sequence of saddle-point systems generated by the IPM, so we propose new preconditioners that may be embedded within suitable Krylov subspace methods, based on approximations of the $(1, 1)$-block and the Schur complement. In contrast to previous results for the case $\beta = 0$ in the work of Pearson and Gondzio,[22] the structure of the resulting systems is more complex due to the L^1-norm contribution within the objective function. In particular, as a result of the smoothing of the resulting vector ℓ_1-norm term, the $(1, 1)$-block is of larger dimension and is also close to singular, which must be carefully addressed when devising potent preconditioners. This also has implications for the structure of the Schur complement, which is again more complex than for L^2-regularized problems and for which suitable approximations must also be devised. In this paper, we derive new preconditioners for the matrix systems arising from (1) and analyze the spectral properties of the preconditioned $(1, 1)$-block and Schur complement, to guide us as to their effectiveness. We also demonstrate that our approach is applicable when tackling problems involving partial observations, meaning that the $(1, 1)$-blocks of the saddle-point systems are singular, or time-dependent problems.

We structure this paper as follows. The discretization of the continuous problem is discussed in Section 2, whereupon an interior-point scheme is introduced in Section 3 together with the description of the linear algebra considerations. Hence, Section 4 is devoted to introducing preconditioning strategies to improve the convergence behavior of the linear iterative solver. We highlight a “matching approach” that introduces robust approximations to the Schur complement of the linear system. Additionally, we propose a preconditioning strategy for problems involving partial observations in Section 4.3 and time-dependent PDE-constrained optimization in Section 4.4. Section 5 illustrates the performance of our scheme for a variety of different parameter regimes, discretization levels, and PDE constraints.

Notation. The L^1-norm of a function u is denoted by $\|\mathrm{u}\|_{L^1}$, whereas the ℓ_1-norm of a vector u is denoted by $\|u\|_1$. Components of a vector x are denoted by x_j or by $x_{a,j}$ for a vector x_a. The matrix I_n denotes the $n \times n$ identity matrix, and 1_n is the column vector of ones of dimension n.

2 | PROBLEM DISCRETIZATION AND QUADRATIC PROGRAMMING FORMULATION

We here apply a discretize-then-optimize approach to (1)–(5) and use a finite element discretization that retains a favorable property of the vector ℓ_1-norm, specifically that it is separable with respect to the vector components. This key step allows us to state the discretized problem as a convex quadratic program that may be tackled using an IPM.

Let n denote the dimension of the discretized space, for both state and control variables, and let h be the corresponding mesh size. Let matrix L represent a discretization of the Laplacian operator (the *stiffness matrix*) when Poisson's equation is considered or, more generally, the discretization of a non–self-adjoint elliptic differential operator, and let matrix M be the finite element Gram matrix or *mass matrix*. Finally, we denote by $y, u, y_d, f, u_a, u_b, y_a$, and y_b the discrete counterparts of the functions $\mathrm{y}, \mathrm{u}, \mathrm{y_d}, \mathrm{f}, \mathrm{u_a}, \mathrm{u_b}, \mathrm{y_a}$, and $\mathrm{y_b}$, respectively.

The discretization without the additional sparsity term follows a standard Galerkin approach.[3,13,29] For the discretization of the L^1 term, we here follow[7–10] and apply the nodal quadrature rule, as follows:

$$\|\mathrm{u}\|_{\mathrm{L}^1(\Omega)} \approx \sum_{i=1}^{n} |u_i| \int_\Omega \phi_i(x)\,\mathrm{d}x,$$

where $\{\phi_i\}$ are the finite element basis functions used and u_i are the components of u. It is shown in the work of Wachsmuth and Wachsmuth[10] that first-order convergence with respect to mesh size may be achieved using this approximation with piecewise linear discretizations of the control. We define a lumped mass matrix D as

$$D := \operatorname{diag}\left(\int_\Omega \phi_i(x)\,\mathrm{d}x\right)_{i=1}^{n},$$

so that the discretized L^1-norm can be written in matrix–vector form as $\|Du\|_1$. As a result, the overall finite element discretization of problem (1)–(5) may be stated as

$$\begin{aligned} \min_{y\in\mathbb{R}^n, u\in\mathbb{R}^n} \ & \frac{1}{2}(y-y_d)^T M(y-y_d) + \frac{\alpha}{2}u^T M u + \beta\|Du\|_1 \\ \text{s.t.}\ \ & Ly - Mu = f, \end{aligned} \tag{6}$$

while additionally being in the presence of control and state constraints, as follows:

$$u_a \le u \le u_b, \qquad y_a \le y \le y_b. \tag{7}$$

The problems we consider will always have control constraints present and will sometimes also involve state constraints.

Problem (6)–(7) is a linearly constrained quadratic problem with bound constraints on the state and control variables (y, u) and with an additional nonsmooth weighted ℓ_1-norm term of the variable u. A possible approach to handle the nonsmoothness in the problem consists of using smoothing techniques for the ℓ_1-norm term (see, e.g., the works of Figueiredo et al.,[26] Fountoulakis and Gondzio,[30] and Fountoulakis et al.[31]). We here consider a classical strategy proposed in the work of Figueiredo et al.,[26] which linearizes the ℓ_1-norm by splitting the variable u as follows. Let $w, v \in \mathbb{R}^n$ be such that

$$|u_i| = w_i + v_i, \quad i = 1, \ldots, n,$$

where $w_i = \max(u_i, 0)$ and $v_i = \max(-u_i, 0)$. Therefore, we have

$$\|u\|_1 = 1_n^T w + 1_n^T v,$$

with $w, v \ge 0$. In the weighted case, which we are interested in when approximating the discretized version of $\|\mathrm{u}\|_{\mathrm{L}^1(\Omega)}$ by $\|Du\|_1$, we obtain

$$\|Du\|_1 = 1_n^T D w + 1_n^T D v.$$

By using the relationship

$$u = w - v, \tag{8}$$

one may now rewrite problem (6) in terms of variables (y, z), with

$$z = \begin{bmatrix} w \\ v \end{bmatrix}.$$

Note that bounds for u, that is,

$$u_a \le u \le u_b,$$

now have to be replaced by the following bounds for z:

$$z_a \le z \le z_b,$$

with

$$z_a = \begin{bmatrix} \max\{u_a, 0\} \\ -\min\{u_b, 0\} \end{bmatrix}, \qquad z_b = \begin{bmatrix} \max\{u_b, 0\} \\ -\min\{u_a, 0\} \end{bmatrix}.$$

We note that these bounds automatically satisfy the constraint $z \geq 0$. Overall, we have the desired quadratic programming formulation, as follows:

$$\begin{aligned} \min_{y \in \mathbb{R}^n, z \in \mathbb{R}^{2n}} \; & Q(y, z) := \frac{1}{2}(y - y_d)^T M (y - y_d) + \frac{\alpha}{2} z^T \widetilde{M} z + \beta 1_{2n}^T \bar{D} z \\ \text{s.t.} \; & Ly - \bar{M} z = f, \\ & z_a \leq z \leq z_b, \\ & y_a \leq y \leq y_b, \end{aligned} \tag{9}$$

where

$$\widetilde{M} = \begin{bmatrix} M & -M \\ -M & M \end{bmatrix}, \qquad \bar{D} = [\, D \;\, D \,], \qquad \bar{M} = [\, M \;\, -M \,].$$

In the next section, we derive an interior-point scheme for the solution of the above problem. Clearly, once optimal values of variable z, and therefore of w and v, are found, the control u of the initial problem is retrieved by (8). We observe that we gain smoothness in the problem at the expense of increasing the number of variables by 50% within the problem statement. Fortunately, this increase will not have a significant impact in the linear algebra solution phase of our method, as we only require additional sparse matrix–vector multiplications and the storage of the additional control vectors.

3 | INTERIOR-POINT FRAMEWORK AND NEWTON EQUATIONS

The three key steps to set up an IPM are the following. First, the bound constraints are "eliminated" by using a logarithmic barrier function. For problem (9), the barrier function takes the form

$$\begin{aligned} \Psi_\mu(y, z, p) = Q(y, z) + p^T(Ly - \bar{M}z - f) - \mu \sum \log(y_j - y_{a,j}) - \mu \sum \log(y_{b,j} - y_j) \\ - \mu \sum \log(z_j - z_{a,j}) - \mu \sum \log(z_{b,j} - z_j), \end{aligned}$$

where $p \in \mathbb{R}^n$ is the Lagrange multiplier (or adjoint variable) associated with the state equation, and $\mu > 0$ is the barrier parameter that controls the relation between the barrier term and the original objective $Q(y, z)$. As the IPM progresses, μ is decreased toward zero.

The second step involves applying the duality theory and deriving the first-order optimality conditions to obtain a nonlinear system parameterized by μ. Differentiating Ψ_μ with respect to (y, z, p) gives the nonlinear system

$$My - My_d + L^T p - \lambda_{y,a} + \lambda_{y,b} = 0, \tag{10}$$

$$\alpha \widetilde{M} z + \beta \bar{D}^T 1_n - \bar{M}^T p - \lambda_{z,a} + \lambda_{z,b} = 0, \tag{11}$$

$$Ly - \bar{M} z - f = 0, \tag{12}$$

where the jth entries of the Lagrange multipliers $\lambda_{y,a}$, $\lambda_{y,b}$, $\lambda_{z,a}$, $\lambda_{z,b}$ are defined as follows:

$$(\lambda_{y,a})_j = \frac{\mu}{y_j - y_{a,j}}, \qquad (\lambda_{y,b})_j = \frac{\mu}{y_{b,j} - y_j}, \qquad (\lambda_{z,a})_j = \frac{\mu}{z_j - z_{a,j}}, \qquad (\lambda_{z,b})_j = \frac{\mu}{z_{b,j} - z_j}. \tag{13}$$

Also, the following bound constraints enforce the constraints on y and z via

$$\lambda_{y,a} \geq 0, \qquad \lambda_{y,b} \geq 0, \qquad \lambda_{z,a} \geq 0, \qquad \lambda_{z,b} \geq 0.$$

The third crucial step of the IPM is the application of Newton's method to the nonlinear system given by the seven equalities in (10)–(13). We now derive the Newton equations, following the description in the work of Pearson and Gondzio.[22] Letting $y, z, p, \lambda_{y,a}, \lambda_{y,b}, \lambda_{z,a}, \lambda_{z,b}$ denote the most recent Newton iterates, these quantities are updated at each iteration by computing the corresponding Newton steps $\Delta y, \Delta z, \Delta p, \Delta\lambda_{y,a}, \Delta\lambda_{y,b}, \Delta\lambda_{z,a}, \Delta\lambda_{z,b}$, through the solution of the following Newton system:

$$\begin{bmatrix} M & 0 & L^T & -I_n & I_n & 0 & 0 \\ 0 & \alpha\widetilde{M} & -\bar{M}^T & 0 & 0 & -I_{2n} & I_{2n} \\ L & -\bar{M} & 0 & 0 & 0 & 0 & 0 \\ \Lambda_{y,a} & 0 & 0 & Y-Y_a & 0 & 0 & 0 \\ -\Lambda_{y,b} & 0 & 0 & 0 & Y_b-Y & 0 & 0 \\ 0 & \Lambda_{z,a} & 0 & 0 & 0 & Z-Z_a & 0 \\ 0 & -\Lambda_{z,b} & 0 & 0 & 0 & 0 & Z_b-Z \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta z \\ \Delta p \\ \Delta\lambda_{y,a} \\ \Delta\lambda_{y,b} \\ \Delta\lambda_{z,a} \\ \Delta\lambda_{z,b} \end{bmatrix}$$
$$= -\begin{bmatrix} My - My_d + L^Tp - \lambda_{y,a} + \lambda_{y,b} \\ \alpha\widetilde{M}z + \beta\bar{D}^T 1_n - \bar{M}^T p - \lambda_{z,a} + \lambda_{z,b} \\ Ly - \bar{M}z - f \\ (y-y_a).*\lambda_{y,a} - \mu 1_n \\ (y_b - y).*\lambda_{y,b} - \mu 1_n \\ (z-z_a).*\lambda_{z,a} - \mu 1_{2n} \\ (z_b - z).*\lambda_{z,a} - \mu 1_{2n} \end{bmatrix}, \tag{14}$$

where $Y, Z, \Lambda_{y,a}, \Lambda_{y,b}, \Lambda_{z,a}, \Lambda_{z,b}$ are diagonal matrices, with the most recent iterates $y, z, p, \lambda_{y,a}, \lambda_{y,b}, \lambda_{z,a}, \lambda_{z,b}$ appearing on their diagonal entries. Similarly, the matrices Y_a, Y_b, Z_a, Z_b are diagonal matrices corresponding to the bounds y_a, y_b, z_a, z_b. Here, we utilize the MATLAB notation "$.*$" to denote the componentwise product. We observe that the contribution of the ℓ_1-norm term only arises on the right-hand side, that is, β does not appear within the matrix we need to solve for.

Eliminating $\Delta\lambda_{y,a}, \Delta\lambda_{y,b}, \Delta\lambda_{z,a}, \Delta\lambda_{z,b}$ from (14), we obtain the following reduced linear system:

$$\begin{bmatrix} M+\Theta_y & 0 & L^T \\ 0 & \alpha\widetilde{M}+\Theta_z & -\bar{M}^T \\ L & -\bar{M} & 0 \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta z \\ \Delta p \end{bmatrix} = -\begin{bmatrix} My - My_d + L^Tp - \mu(Y-Y_a)^{-1}1_n + \mu(Y_b-Y)^{-1}1_n \\ \alpha\widetilde{M}z + \beta\bar{D}^T 1_n - \bar{M}^T p - \mu(Z-Z_a)^{-1}1_{2n} + \mu(Z_b-Z)^{-1}1_{2n} \\ Ly - \bar{M}z - f \end{bmatrix}, \tag{15}$$

with

$$\Theta_y = (Y-Y_a)^{-1}\Lambda_{y,a} + (Y_b-Y)^{-1}\Lambda_{y,b}, \qquad \Theta_z = (Z-Z_a)^{-1}\Lambda_{z,a} + (Z_b-Z)^{-1}\Lambda_{z,b}$$

being diagonal and positive definite matrices, which are typically very ill-conditioned. In particular, in our implementation, as is standard within IPM codes, we set a maximum value for the diagonal entries of Θ_y and Θ_z (of the order of the inverse of machine precision) to combat the possibility of a diagonal entry being infinite numerically. Once the above system is solved, one can compute the steps for the Lagrange multipliers, as follows:

$$\Delta\lambda_{y,a} = -(Y-Y_a)^{-1}\Lambda_{y,a}\Delta y - \Lambda_{y,a} + \mu(Y-Y_a)^{-1}1_n, \tag{16}$$

$$\Delta\lambda_{y,b} = (Y_b-Y)^{-1}\Lambda_{y,b}\Delta y - \Lambda_{y,b} + \mu(Y_b-Y)^{-1}1_n, \tag{17}$$

$$\Delta\lambda_{z,a} = -(Z-Z_a)^{-1}\Lambda_{z,a}\Delta z - \Lambda_{z,a} + \mu(Z-Z_a)^{-1}1_{2n}, \tag{18}$$

$$\Delta\lambda_{z,b} = (Z_b-Z)^{-1}\Lambda_{z,b}\Delta z - \Lambda_{z,b} + \mu(Z_b-Z)^{-1}1_{2n}. \tag{19}$$

After updating the iterates and ensuring that they remain feasible, the barrier μ is reduced, and a new Newton step is performed.

For the sake of completeness, the structure of the overall interior-point algorithm is reported in the Appendix and follows the standard infeasible interior-point path-following scheme outlined in the work of Gondzio.[24] We report the formulas for the primal and dual feasibilities, given by

$$\xi_p^k = Ly^k - \bar{M}z^k - f, \qquad \xi_d^k = \begin{bmatrix} My^k - My_d + L^Tp^k - \lambda_{y,a}^k + \lambda_{y,b}^k \\ \alpha\widetilde{M}z^k + \beta\bar{D}^T 1_n - \bar{M}^T p^k - \lambda_{z,a}^k + \lambda_{z,b}^k \end{bmatrix}, \tag{20}$$

respectively, and the complementarity gap

$$\xi_c^k = \begin{bmatrix} (y^k - y_a).*\lambda_{y,a}^k - \mu^k 1_n \\ (y_b - y^k).*\lambda_{y,b}^k - \mu^k 1_n \\ (z^k - z_a).*\lambda_{z,a}^k - \mu^k 1_{2n} \\ (z_b - z^k).*\lambda_{z,a}^k - \mu^k 1_{2n} \end{bmatrix}, \tag{21}$$

for problem (9). Here, k denotes the iteration counter for the IPM, with $y^k, z^k, p^k, \lambda_{y,a}^k, \lambda_{y,b}^k, \lambda_{u,a}^k, \lambda_{u,b}^k, \mu^k$ being the values of $y, z, p, \lambda_{y,a}, \lambda_{y,b}, \lambda_{u,a}, \lambda_{u,b}, \mu$ at the kth iteration.

The measure of the change in the norm of $\xi_p^k, \xi_d^k, \xi_c^k$ allows us to monitor the convergence of the entire process. Computationally, the main bottleneck of the algorithm is the linear algebra phase, that is, the efficient solution of the Newton system (15). This is the focus of the forthcoming section.

4 | PRECONDITIONING

Having arrived at the Newton system (15), the main task at this stage is to construct fast and effective methods for the solution of such systems. In this work, we elect to apply iterative (Krylov subspace) solvers, both the minimal residual (MINRES) method[32] for symmetric matrix systems and the generalized minimal residual (GMRES) algorithm[33] that may also be applied to nonsymmetric matrices. We wish to accelerate these methods using carefully chosen preconditioners.

To develop these preconditioners, we observe that (15) is a *saddle-point system* (see the work of Benzi et al.[34] for a review of such systems) of the form

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

with

$$A = \begin{bmatrix} M + \Theta_y & 0 \\ 0 & \alpha\widetilde{M} + \Theta_z \end{bmatrix}, \qquad B = [\, L \;\; -\bar{M} \,], \qquad C = [\, 0 \,].$$

Provided A is nonsingular, it is well known that two *ideal preconditioners* for the saddle-point matrix $\mathcal{A}$ are given by

$$\mathcal{P}_1 = \begin{bmatrix} A & 0 \\ 0 & S \end{bmatrix}, \qquad \mathcal{P}_2 = \begin{bmatrix} A & 0 \\ B & -S \end{bmatrix},$$

where the (negative) *Schur complement* $S := -C + BA^{-1}B^T$. In particular, provided the preconditioned system is nonsingular, it can be shown that[35–37]

$$\lambda\left(\mathcal{P}_1^{-1}\mathcal{A}\right) \in \left\{1, \frac{1}{2}\left(1 \pm \sqrt{5}\right)\right\}, \qquad \lambda\left(\mathcal{P}_2^{-1}\mathcal{A}\right) \in \{1\},$$

and, hence, that a suitable Krylov method preconditioned by $\mathcal{P}_1$ or $\mathcal{P}_2$ will converge in three or two iterations, respectively.

Of course, we would not wish to work with preconditioner $\mathcal{P}_1$ or $\mathcal{P}_2$ in practice, as they would be prohibitively expensive to invert. We therefore wish to develop analogous preconditioners of the form

$$\mathcal{P}_D = \begin{bmatrix} \widehat{A} & 0 \\ 0 & \widehat{S} \end{bmatrix}, \qquad \mathcal{P}_T = \begin{bmatrix} \widehat{A} & 0 \\ B & -\widehat{S} \end{bmatrix},$$

where $\widehat{A}$ and $\widehat{S}$ are suitable and computationally cheap approximations of the $(1, 1)$-block A and the Schur complement S. Provided $\widehat{A}$ and $\widehat{S}$ are symmetric positive definite, preconditioner $\mathcal{P}_D$ may be applied within the MINRES algorithm, and $\mathcal{P}_T$ is applied within a nonsymmetric solver such as GMRES.

Our focus is therefore to develop such approximations for the corresponding matrices for the Newton system (15), as follows:

$$A = \begin{bmatrix} M + \Theta_y & 0 \\ 0 & \alpha\widetilde{M} + \Theta_z \end{bmatrix}, \qquad S = [\, L \;\; -\bar{M} \,] \begin{bmatrix} M + \Theta_y & 0 \\ 0 & \alpha\widetilde{M} + \Theta_z \end{bmatrix}^{-1} \begin{bmatrix} L^T \\ -\bar{M}^T \end{bmatrix}.$$

4.1 | Approximation of $(1,1)$-block

An effective approximation of the $(1,1)$-block A will require cheap and accurate approximations of the matrices $M+\Theta_y$ and $\alpha\widetilde{M}+\Theta_z$. The key property that we make use of when devising such approximations is that a mass matrix M may be effectively approximated by its diagonal D_M within a preconditioner, for a range of (nodal) finite element bases.[38] For instance, when using $Q1$ basis functions, which we later employ within our numerical experiments, it can be shown that $\lambda(D_M^{-1}M)\in\left[\frac{1}{4},\frac{9}{4}\right]$ for a two-dimensional problem, with $\lambda(D_M^{-1}M)\in\left[\frac{1}{8},\frac{27}{8}\right]$ in three dimensions.

This valuable property of mass matrices can be exploited and enhanced by applying the *Chebyshev semi-iteration* method,[39–41] which utilizes the effectiveness of the diagonal approximation and accelerates it. Now, it may be easily shown that

$$\begin{aligned}&\left[\lambda_{\min}((D_M+\Theta_y)^{-1}(M+\Theta_y)),\lambda_{\max}((D_M+\Theta_y)^{-1}(M+\Theta_y))\right]\\ &\subset\left[\min\left\{\lambda_{\min}\left(D_M^{-1}M\right),1\right\},\max\left\{\lambda_{\max}\left(D_M^{-1}M\right),1\right\}\right],\end{aligned}$$

due to the positivity of the diagonal matrix Θ_y. Here, $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ denote the smallest and largest eigenvalues of a matrix, respectively. In other words, the diagonal of $M+\Theta_y$ also clusters the eigenvalues within a preconditioner. The same argument may therefore be used to apply Chebyshev semi-iteration to $M+\Theta_y$ within a preconditioner, and so we elect to use this approach.

We now turn our attention to the matrix $\alpha\widetilde{M}+\Theta_z$, first decomposing $\Theta_z=\text{blkdiag}(\Theta_w,\Theta_v)$, where Θ_w and Θ_v denote the components of Θ_z corresponding to w and v, respectively. Therefore, in this notation, we have

$$\alpha\widetilde{M}+\Theta_z=\begin{bmatrix}\alpha M+\Theta_w & -\alpha M\\ -\alpha M & \alpha M+\Theta_v\end{bmatrix}.$$

Note that $\widetilde{M}$ is positive semidefinite but $\alpha\widetilde{M}+\Theta_z$ is positive definite since the diagonal Θ_z is positive definite (the control and state bounds are enforced as strict inequalities at each Newton step).

A result we apply is that of theorems 2.1(i) and 2.2(i) in the work of Lu and Shiou,[42] which gives us the following statements about the inverse of 2×2 block matrices.

Theorem 1. *Consider the inverse of the block matrix*

$$\begin{bmatrix}A & B_1\\ B_2 & C\end{bmatrix}. \tag{22}$$

If A is nonsingular and $C-B_2A^{-1}B_1$ is invertible, then (22) is invertible, with

$$\begin{bmatrix}A & B_1\\ B_2 & C\end{bmatrix}^{-1}=\begin{bmatrix}A^{-1}+A^{-1}B_1(C-B_2A^{-1}B_1)^{-1}B_2A^{-1} & -A^{-1}B_1(C-B_2A^{-1}B_1)^{-1}\\ -(C-B_2A^{-1}B_1)^{-1}B_2A^{-1} & (C-B_2A^{-1}B_1)^{-1}\end{bmatrix}. \tag{23}$$

Alternatively, if B_1 is nonsingular and $B_2-CB_1^{-1}A$ is invertible, then (22) is invertible, with

$$\begin{bmatrix}A & B_1\\ B_2 & C\end{bmatrix}^{-1}=\begin{bmatrix}-\left(B_2-CB_1^{-1}A\right)^{-1}CB_1^{-1} & \left(B_2-CB_1^{-1}A\right)^{-1}\\ B_1^{-1}+B_1^{-1}A\left(B_2-CB_1^{-1}A\right)^{-1}CB_1^{-1} & -B_1^{-1}A\left(B_2-CB_1^{-1}A\right)^{-1}\end{bmatrix}. \tag{24}$$

For the purposes of this working, we may therefore consider the matrix $\alpha\widetilde{M}+\Theta_z$ itself as a block matrix (22), with $A=\alpha M+\Theta_w$, $B_1=B_2=-\alpha M$, $C=\alpha M+\Theta_v$. It may easily be verified that A, $C-B_2A^{-1}B_1$, B_1, $B_2-CB_1^{-1}A$ are then invertible matrices, and so, the results (23) and (24) both hold in this setting.

We now consider approximating $\alpha\widetilde{M}+\Theta_z$ within a preconditioner by replacing all mass matrices with their diagonals, that is, writing

$$\alpha\widetilde{D}_M+\Theta_z:=\begin{bmatrix}\alpha D_M+\Theta_w & -\alpha D_M\\ -\alpha D_M & \alpha D_M+\Theta_v\end{bmatrix}.$$

This would give us a practical approximation, by using expression (23) to apply $(\alpha\widetilde{D}_M+\Theta_z)^{-1}$, provided it can be demonstrated that $\alpha\widetilde{D}_M+\Theta_z$ well approximates $\alpha\widetilde{M}+\Theta_z$. This is indeed the case, as demonstrated using the result below.

Theorem 2. *The eigenvalues λ of the matrix*

$$\begin{bmatrix} \alpha D_M+\Theta_w & -\alpha D_M \\ -\alpha D_M & \alpha D_M+\Theta_v \end{bmatrix}^{-1} \begin{bmatrix} \alpha M+\Theta_w & -\alpha M \\ -\alpha M & \alpha M+\Theta_v \end{bmatrix} \tag{25}$$

are all contained within the interval

$$\lambda \in \left[\min\left\{\lambda_{\min}\left(D_M^{-1}M\right),1\right\},\max\left\{\lambda_{\max}\left(D_M^{-1}M\right),1\right\}\right].$$

Proof. The eigenvalues of (25) satisfy

$$\begin{bmatrix} \alpha M+\Theta_w & -\alpha M \\ -\alpha M & \alpha M+\Theta_v \end{bmatrix}\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \lambda \begin{bmatrix} \alpha D_M+\Theta_w & -\alpha D_M \\ -\alpha D_M & \alpha D_M+\Theta_v \end{bmatrix}\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix},$$

with $\mathbf{x}_1$ and $\mathbf{x}_2$ not both equal to $\mathbf{0}$, which may be decomposed to write

$$(\alpha M+\Theta_w)\mathbf{x}_1-\alpha M\mathbf{x}_2=\lambda(\alpha D_M+\Theta_w)\mathbf{x}_1-\lambda\alpha D_M\mathbf{x}_2, \tag{26}$$

$$-\alpha M\mathbf{x}_1+(\alpha M+\Theta_v)\mathbf{x}_2=-\lambda\alpha D_M\mathbf{x}_1+\lambda(\alpha D_M+\Theta_v)\mathbf{x}_2. \tag{27}$$

Summing (26) and (27) gives

$$\Theta_w\mathbf{x}_1+\Theta_v\mathbf{x}_2=\lambda\Theta_w\mathbf{x}_1+\lambda\Theta_v\mathbf{x}_2=\lambda(\Theta_w\mathbf{x}_1+\Theta_v\mathbf{x}_2),$$

which tells us that either $\lambda=1$ or $\Theta_w\mathbf{x}_1+\Theta_v\mathbf{x}_2=\mathbf{0}$. In the latter case, we substitute $\mathbf{x}_1=-\Theta_w^{-1}\Theta_v\mathbf{x}_2$ into (26) to give

$$\begin{aligned} -(\alpha M+\Theta_w)\Theta_w^{-1}\Theta_v\mathbf{x}_2-\alpha M\mathbf{x}_2 &= -\lambda(\alpha D_M+\Theta_w)\Theta_w^{-1}\Theta_v\mathbf{x}_2-\lambda\alpha D_M\mathbf{x}_2 \\ \Rightarrow \qquad \left[\alpha M\left(\Theta_w^{-1}\Theta_v+I\right)+\Theta_v\right]\mathbf{x}_2 &= \lambda\left[\alpha D_M\left(\Theta_w^{-1}\Theta_v+I\right)+\Theta_v\right]\mathbf{x}_2, \end{aligned}$$

which, in turn, tells us that

$$\left[\alpha M\left(\Theta_w^{-1}\Theta_v+I\right)^{1/2}+\Theta_v\left(\Theta_w^{-1}\Theta_v+I\right)^{-1/2}\right]\mathbf{x}_3=\lambda\left[\alpha D_M\left(\Theta_w^{-1}\Theta_v+I\right)^{1/2}+\Theta_v\left(\Theta_w^{-1}\Theta_v+I\right)^{-1/2}\right]\mathbf{x}_3,$$

where $\mathbf{x}_3=(\Theta_w^{-1}\Theta_v+I)^{1/2}\mathbf{x}_2\neq\mathbf{0}$. Premultiplying both sides of the equation by $(\Theta_w^{-1}\Theta_v+I)^{1/2}$ then gives

$$\left[\alpha\left(\Theta_w^{-1}\Theta_v+I\right)^{1/2}M\left(\Theta_w^{-1}\Theta_v+I\right)^{1/2}+\Theta_v\right]\mathbf{x}_3=\lambda\left[\alpha\left(\Theta_w^{-1}\Theta_v+I\right)^{1/2}D_M\left(\Theta_w^{-1}\Theta_v+I\right)^{1/2}+\Theta_v\right]\mathbf{x}_3$$

and therefore that the bounds on eigenvalues may be described by the Rayleigh quotient

$$\frac{\mathbf{x}_3^T\left[\alpha\left(\Theta_w^{-1}\Theta_v+I\right)^{1/2}M\left(\Theta_w^{-1}\Theta_v+I\right)^{1/2}+\Theta_v\right]\mathbf{x}_3}{\mathbf{x}_3^T\left[\alpha\left(\Theta_w^{-1}\Theta_v+I\right)^{1/2}D_M\left(\Theta_w^{-1}\Theta_v+I\right)^{1/2}+\Theta_v\right]\mathbf{x}_3}.$$

Now, as $\mathbf{x}_3^T\Theta_v\mathbf{x}_3$ is a positive number, λ may be bounded within the range of the following Rayleigh quotient:

$$
\begin{aligned}
\lambda \in \Bigg[& \min\left\{ \min_{\mathbf{x}_3} \frac{\mathbf{x}_3^T\left[\alpha\left(\Theta_w^{-1}\Theta_v+I\right)^{1/2}M\left(\Theta_w^{-1}\Theta_v+I\right)^{1/2}\right]\mathbf{x}_3}{\mathbf{x}_3^T\left[\alpha\left(\Theta_w^{-1}\Theta_v+I\right)^{1/2}D_M\left(\Theta_w^{-1}\Theta_v+I\right)^{1/2}\right]\mathbf{x}_3}, 1\right\}, \\
& \max\left\{ \max_{\mathbf{x}_3} \frac{\mathbf{x}_3^T\left[\alpha\left(\Theta_w^{-1}\Theta_v+I\right)^{1/2}M\left(\Theta_w^{-1}\Theta_v+I\right)^{1/2}\right]\mathbf{x}_3}{\mathbf{x}_3^T\left[\alpha\left(\Theta_w^{-1}\Theta_v+I\right)^{1/2}D_M\left(\Theta_w^{-1}\Theta_v+I\right)^{1/2}\right]\mathbf{x}_3}, 1\right\}\Bigg] \\
= & \left[\min\left\{\min_{\mathbf{x}_4}\frac{\mathbf{x}_4^TM\mathbf{x}_4}{\mathbf{x}_4^TD_M\mathbf{x}_4}, 1\right\}, \max\left\{\max_{\mathbf{x}_4}\frac{\mathbf{x}_4^TM\mathbf{x}_4}{\mathbf{x}_4^TD_M\mathbf{x}_4}, 1\right\}\right] \\
\subset & \left[\min\left\{\lambda_{\min}\left(D_M^{-1}M\right), 1\right\}, \max\left\{\lambda_{\max}\left(D_M^{-1}M\right), 1\right\}\right],
\end{aligned}
$$

where, in the above derivation, $\mathbf{x}_4 = (\Theta_w^{-1}\Theta_v + I)^{1/2}\mathbf{x}_3 \neq \mathbf{0}$. This gives the stated result. □

Remark 1. Theorem 2 is a positive result, due to diagonal preconditioning of a mass matrix giving tight eigenvalue bounds for a range of nodal basis functions.[38] We have now obtained a cheap approximation of the $(1,1)$-block of our saddle-point system, with eigenvalues of the preconditioned matrix provably contained within a tight interval. We emphasize the fact that the interval boundaries, and thus the region of interest where the eigenvalues lie, are independent of all system parameters, such as penalization, regularization, mesh, and time-step parameters.

4.2 | Approximation of Schur complement

The Schur complement of the Newton system (15) under consideration is given by

$$
S = L(M+\Theta_y)^{-1}L^T + [\,-M \;\; M\,]\begin{bmatrix} \alpha M+\Theta_w & -\alpha M \\ -\alpha M & \alpha M + \Theta_v \end{bmatrix}^{-1}\begin{bmatrix} -M \\ M \end{bmatrix}.
$$

For the matrix inverse in the above expression, we again consider the matrix $\alpha\widetilde{M} + \Theta_z$ as a block matrix of the form (22), with $A = \alpha M + \Theta_w$, $B_1 = B_2 = B = -\alpha M$, $C = \alpha M + \Theta_v$. Using (24) then gives

$$
\begin{aligned}
[\,-M \;\; M\,]\begin{bmatrix} A & B \\ B & C \end{bmatrix}^{-1}\begin{bmatrix} -M \\ M \end{bmatrix} &= [\,-M \;\; M\,]\begin{bmatrix} (B-CB^{-1}A)^{-1}CB^{-1}M + (B-CB^{-1}A)^{-1}M \\ -B^{-1}M - B^{-1}A(B-CB^{-1}A)^{-1}CB^{-1}M - B^{-1}A(B-CB^{-1}A)^{-1}M \end{bmatrix} \\
&= -M\left[B^{-1} + (B^{-1}A+I)(B-CB^{-1}A)^{-1}(CB^{-1}+I)\right]M,
\end{aligned}
$$

whereupon substituting in the relevant A, B, and C gives that this expression can be written as follows:

$$
\begin{aligned}
& \frac{1}{\alpha}M - \left(-\frac{1}{\alpha}A + M\right)\left(-\alpha M + \frac{1}{\alpha}CM^{-1}A\right)^{-1}\left(-\frac{1}{\alpha}C + M\right) \\
= & \frac{1}{\alpha}M + \left(\frac{1}{\alpha}\Theta_w\right)\left(\alpha M - \left(\alpha M + \Theta_w + \Theta_v + \frac{1}{\alpha}\Theta_v M^{-1}\Theta_w\right)\right)^{-1}\left(\frac{1}{\alpha}\Theta_v\right) \\
= & \frac{1}{\alpha}M - \frac{1}{\alpha^2}\left(\Theta_w^{-1} + \Theta_v^{-1} + \frac{1}{\alpha}M^{-1}\right)^{-1}.
\end{aligned}
$$

Therefore, S may be written as

$$
S = L(M+\Theta_y)^{-1}L^T + \frac{1}{\alpha}M - \frac{1}{\alpha^2}\left(\Theta_w^{-1} + \Theta_v^{-1} + \frac{1}{\alpha}M^{-1}\right)^{-1}. \tag{28}
$$

It can be shown that S consists of a sum of two symmetric positive semidefinite matrices. The matrix $L(M+\Theta_y)^{-1}L^T$ clearly satisfies this property due to the positive definiteness of $M + \Theta_y$, and $\frac{1}{\alpha}M - \frac{1}{\alpha^2}(\Theta_w^{-1} + \Theta_v^{-1} + \frac{1}{\alpha}M^{-1})^{-1}$ is, in fact, positive definite by the following argument:

$$\begin{aligned}\frac{1}{\alpha}M-\frac{1}{\alpha^2}\left(\frac{1}{\alpha}M^{-1}+\Theta_w^{-1}+\Theta_v^{-1}\right)^{-1}\succ 0 \quad &\Leftrightarrow \quad \frac{1}{\alpha^2}\left(\frac{1}{\alpha}M^{-1}+\Theta_w^{-1}+\Theta_v^{-1}\right)^{-1}\prec\frac{1}{\alpha}M\\ &\Leftrightarrow \quad \alpha^2\left(\frac{1}{\alpha}M^{-1}+\Theta_w^{-1}+\Theta_v^{-1}\right)\succ\alpha M^{-1}\\ &\Leftrightarrow \quad M^{-1}+\alpha\Theta_w^{-1}+\alpha\Theta_v^{-1}\succ M^{-1}.\end{aligned}$$

On the basis of this observation, we apply a *matching strategy*, previously derived in the works of Pearson and Wathen[25] and Pearson et al.[43] for simpler PDE-constrained optimization problems, which relies on a Schur complement being written in this form. In more detail, we approximate the Schur complement S by

$$\widehat{S}=\left(L+\widehat{M}\right)(M+\Theta_y)^{-1}\left(L+\widehat{M}\right)^T, \tag{29}$$

where $\widehat{M}$ is chosen such that the "outer" term of $\widehat{S}$ in (29) approximates the second and third terms of S in (28), that is,

$$\widehat{M}(M+\Theta_y)^{-1}\widehat{M}^T\approx\frac{1}{\alpha}M-\frac{1}{\alpha^2}\left(\Theta_w^{-1}+\Theta_v^{-1}+\frac{1}{\alpha}M^{-1}\right)^{-1}.$$

This may be achieved if

$$\widehat{M}\approx\left[\frac{1}{\alpha}M-\frac{1}{\alpha^2}\left(\Theta_w^{-1}+\Theta_v^{-1}+\frac{1}{\alpha}M^{-1}\right)^{-1}\right]^{1/2}(M+\Theta_y)^{1/2}.$$

A natural choice, which may be readily worked with on a computer, therefore involves replacing mass matrices with their diagonals, making the square roots of matrices practical to work with, and therefore setting

$$\widehat{M}=\left[\frac{1}{\alpha}D_M-\frac{1}{\alpha^2}\left(\Theta_w^{-1}+\Theta_v^{-1}+\frac{1}{\alpha}D_M^{-1}\right)^{-1}\right]^{1/2}(D_M+\Theta_y)^{1/2}.$$

We therefore have a Schur complement approximation $\widehat{S}$ that may be approximately inverted by applying a multigrid method to the matrix $L+\widehat{M}$ and its transpose, along with a matrix–vector multiplication for $M+\Theta_y$.

Below, we present a result concerning the lower bounds of the eigenvalues of the preconditioned Schur complement.

Theorem 3. *In the case of lumped (diagonal) mass matrices, the eigenvalues of the preconditioned Schur complement all satisfy*

$$\lambda(\widehat{S}^{-1}S)\geq\frac{1}{2}.$$

Proof. Bounds for the eigenvalues of $\widehat{S}^{-1}S$ are determined by the extrema of the Rayleigh quotient

$$R:=\frac{\mathbf{v}^TS\mathbf{v}}{\mathbf{v}^T\widehat{S}\mathbf{v}}=\frac{\chi^T\chi+\boldsymbol{\omega}^T\boldsymbol{\omega}}{(\chi+\gamma)^T(\chi+\gamma)},$$

where

$$\begin{aligned}\chi&=(M+\Theta_y)^{-1/2}L^T\mathbf{v},\\ \boldsymbol{\omega}&=\left[\frac{1}{\alpha}M-\frac{1}{\alpha^2}\left(\Theta_w^{-1}+\Theta_v^{-1}+\frac{1}{\alpha}M^{-1}\right)^{-1}\right]^{1/2}\mathbf{v},\\ \gamma&=(M+\Theta_y)^{-1/2}(D_M+\Theta_y)^{1/2}\left[\frac{1}{\alpha}D_M-\frac{1}{\alpha^2}\left(\Theta_w^{-1}+\Theta_v^{-1}+\frac{1}{\alpha}D_M^{-1}\right)^{-1}\right]^{1/2}\mathbf{v}.\end{aligned}$$

Following the argument used in lemma 2 in the work of Pearson and Gondzio,[22] we may bound R as follows:

$$R = \frac{\chi^T\chi + \frac{\boldsymbol{\omega}^T\boldsymbol{\omega}}{\gamma^T\gamma}\gamma^T\gamma}{(\chi+\gamma)^T(\chi+\gamma)} \geq \min\left\{\frac{\boldsymbol{\omega}^T\boldsymbol{\omega}}{\gamma^T\gamma}, 1\right\}\cdot\frac{\chi^T\chi+\gamma^T\gamma}{(\chi+\gamma)^T(\chi+\gamma)} \geq \frac{1}{2}\cdot\min\left\{\frac{\boldsymbol{\omega}^T\boldsymbol{\omega}}{\gamma^T\gamma}, 1\right\}, \tag{30}$$

using the argument

$$\begin{aligned} \frac{1}{2}(\chi-\gamma)^T(\chi-\gamma) \geq 0 \quad &\Leftrightarrow \quad \chi^T\chi+\gamma^T\gamma \geq \frac{1}{2}(\chi+\gamma)^T(\chi+\gamma) \\ &\Leftrightarrow \quad \frac{\chi^T\chi+\gamma^T\gamma}{(\chi+\gamma)^T(\chi+\gamma)} \geq \frac{1}{2}. \end{aligned}$$

We now turn our attention to the product $\frac{\boldsymbol{\omega}^T\boldsymbol{\omega}}{\gamma^T\gamma}$. Straightforward calculation tells us that

$$\frac{\boldsymbol{\omega}^T\boldsymbol{\omega}}{\gamma^T\gamma} = \underbrace{\frac{\mathbf{v}^T[M-(\Theta+M^{-1})^{-1}]\mathbf{v}}{\mathbf{v}^T\left[D_M-\left(\Theta+D_M^{-1}\right)^{-1}\right]\mathbf{v}}}_{=:R_\Theta}\cdot\frac{\mathbf{w}^T(D_M+\Theta_y)^{-1}\mathbf{w}}{\mathbf{w}^T(M+\Theta_y)^{-1}\mathbf{w}},$$

where $\Theta := \alpha\Theta_w^{-1}+\alpha\Theta_v^{-1}$ and $\mathbf{w} := (D_M+\Theta_y)^{1/2}\left[\frac{1}{\alpha}D_M-\frac{1}{\alpha^2}\left(\Theta_w^{-1}+\Theta_v^{-1}+\frac{1}{\alpha}D_M^{-1}\right)^{-1}\right]^{1/2}\mathbf{v}$. It may be observed that

$$\frac{\mathbf{w}^T(D_M+\Theta_y)^{-1}\mathbf{w}}{\mathbf{w}^T(M+\Theta_y)^{-1}\mathbf{w}} \geq \lambda_{\min}\left((D_M+\Theta_y)^{-1}(M+\Theta_y)\right) \geq \min\left\{\lambda_{\min}\left(D_M^{-1}M\right), 1\right\}$$

and, hence, that

$$\frac{\boldsymbol{\omega}^T\boldsymbol{\omega}}{\gamma^T\gamma} \geq R_\Theta\cdot\min\left\{\lambda_{\min}\left(D_M^{-1}M\right), 1\right\}. \tag{31}$$

Finally, we observe that $R_\Theta = 1$ for lumped mass matrices, as $D_M = M$. Inserting (31) into (30) then gives the required result. □

Remark 2. For consistent mass matrices, the working above still holds, except R_Θ and $\lambda_{\min}(D_M^{-1}M)$ are not equal to 1. Therefore, the bound reads

$$\lambda(\widehat{S}^{-1}S) \geq \frac{1}{2}\cdot\min\left\{\min R_\Theta\cdot\min\left\{\lambda_{\min}\left(D_M^{-1}M\right), 1\right\}, 1\right\}$$

and depends on the matrix $[D_M-(\Theta+D_M^{-1})^{-1}]^{-1}[M-(\Theta+M^{-1})^{-1}]$, which does not have uniformly bounded eigenvalues. This is, however, a weak bound, and in practice, we find that the (smallest and largest) eigenvalues of the preconditioned Schur complement are moderate in size.

Furthermore, in numerical experiments, we find the vast majority of the eigenvalues of $\widehat{S}^{-1}S$ to be clustered in the interval $\left[\frac{1}{2}, 1\right]$, particularly as the IPM approaches convergence, for the following reasons. In theorem 4.1 in the work of Pearson and Wathen,[44] it is shown that

$$\lambda\left(\left[\left(L+\frac{1}{\sqrt{\alpha}}M\right)M^{-1}\left(L+\frac{1}{\sqrt{\alpha}}M\right)^T\right]^{-1}\left[LM^{-1}L^T+\frac{1}{\alpha}M\right]\right) \in \left[\frac{1}{2}, 1\right], \tag{32}$$

for any (positive) value of α and any mesh size, provided $L+L^T$ is positive semidefinite, which is the case for Poisson and convection–diffusion problems for instance. For the Schur complement (28) and Schur complement approximation (29), as the IPM approaches convergence, two cases can arise: (a) Some entries of $\Theta_w^{-1}+\Theta_v^{-1}$ can approach zero, whereupon substituting these values into (28) and (29) gives that S and $\widehat{S}$ are both approximately $L(M+\Theta_y^{-1})^{-1}L^T$,

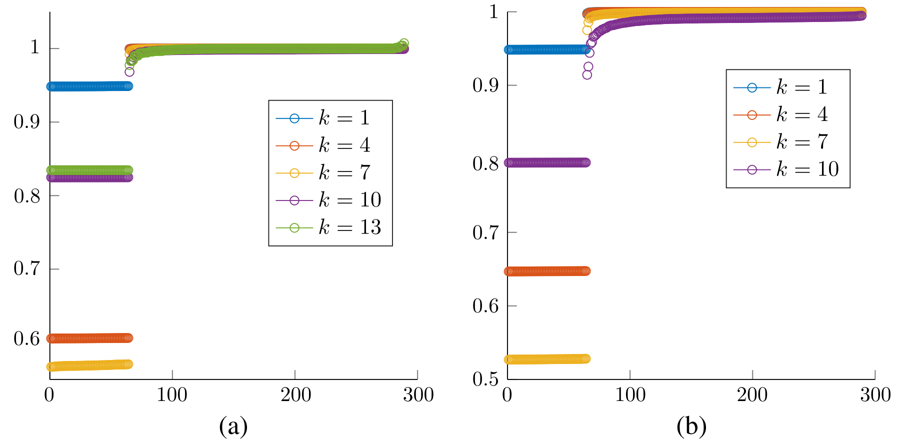


FIGURE 1 Eigenvalue distribution of $\widehat{S}^{-1}S$ at interior-point iterations (1, 4, 7, 10, and 13) for the test problem with Poisson's equation and at interior-point iterations (1, 4, 7, and 10) for the test problem with the convection–diffusion equation (with mesh size $h = 2^{-4}$). (a) Poisson eigenvalues. (b) Convection–diffusion eigenvalues

so the eigenvalues of $\widehat{S}^{-1}S$ should be roughly 1; (ii) some entries of $\Theta_w^{-1} + \Theta_v^{-1}$ approach infinity (with many entries of Θ_y correspondingly approaching zero), so S is approximately $LM^{-1}L^T + \frac{1}{\alpha}M$, with $\widehat{S}$ as an approximation of $(L+\frac{1}{\sqrt{\alpha}}M)M^{-1}(L+\frac{1}{\sqrt{\alpha}}M)^T$, giving clustered eigenvalues as predicted by (32). The numerical evidence of the described behavior, for consistent mass matrices, is shown in Figure 1.

We note that the (1, 1)-block and Schur complement approximations that we have derived are both symmetric positive definite, so we may apply the MINRES algorithm with a block diagonal preconditioner of the form

$$\mathcal{P}_D = \begin{bmatrix} M+\Theta_y & 0 & 0 & 0 \\ 0 & \alpha D_M + \Theta_w & -\alpha D_M & 0 \\ 0 & -\alpha D_M & \alpha D_M + \Theta_v & 0 \\ 0 & 0 & 0 & \widehat{S} \end{bmatrix},$$

with $\widehat{S}$ defined as above.

It is also possible to exploit the often faster convergence achieved by block triangular preconditioners within GMRES and utilize the block triangular preconditioner

$$\mathcal{P}_T = \begin{bmatrix} M+\Theta_y & 0 & 0 & 0 \\ 0 & \alpha D_M + \Theta_w & -\alpha D_M & 0 \\ 0 & -\alpha D_M & \alpha D_M + \Theta_v & 0 \\ L & -M & M & -\widehat{S} \end{bmatrix}.$$

4.3 | Preconditioner for partial observations

In practice, the quantity of importance from a practical point of view is the difference between the state variable and the desired state on a certain region of the domain, that is, $\Omega_1 \subset \Omega$, in which case one would instead consider the term $\frac{1}{2}\|\mathrm{y} - \mathrm{y}_d\|^2_{L^2(\Omega_1)}$ within the cost functional (1). We now briefly outline how to tackle the resulting matrix systems. One may follow developments in the works of Benner et al.[45] and Herzog et al.[46] to obtain the preconditioner

$$\mathcal{P}_\Pi^{-1} = \begin{bmatrix} 0 & L^{-1}\bar{M}(\alpha\widetilde{M}+\Theta_z)^{-1} & L^{-1} \\ 0 & (\alpha\widetilde{M}+\Theta_z)^{-1} & 0 \\ -\widehat{S}_\Pi^{-1} & \widehat{S}_\Pi^{-1}(M+\Theta_y)L^{-1}\bar{M}(\alpha\widetilde{M}+\Theta_z)^{-1} & \widehat{S}_\Pi^{-1}(M+\Theta_y)L^{-1} \end{bmatrix}. \tag{33}$$

The matrix $\widehat{S}_\Pi$ is designed to approximate the Schur complement S_Π of the *permuted matrix system*, that is, the Schur complement of

$$\begin{bmatrix} M+\Theta_y & 0 & L^T \\ 0 & \alpha\widetilde{M}+\Theta_z & -\bar{M}^T \\ L & -\bar{M} & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & I \\ 0 & I & 0 \\ I & 0 & 0 \end{bmatrix},$$

which is given by

$$\widehat{S}_\Pi \approx S_\Pi = L^T + (M+\Theta_y)L^{-1}\bar{M}(\alpha\widetilde{M}+\Theta_z)^{-1}\bar{M}^T.$$

Applying the preconditioner is, in fact, more straightforward than it currently appears. To compute a vector $\mathbf{v} = \mathcal{P}_\Pi^{-1}\mathbf{w}$, where $\mathbf{v} := [\mathbf{v}_1^T, \quad \mathbf{v}_2^T, \quad \mathbf{v}_3^T]^T$, $\mathbf{w} := [\mathbf{w}_1^T, \quad \mathbf{w}_2^T, \quad \mathbf{w}_3^T]^T$, we first observe from the second block of $\mathcal{P}_\Pi^{-1}$ that

$$(\alpha\widetilde{M}+\Theta_z)^{-1}\mathbf{w}_2 = \mathbf{v}_2.$$

The first equation derived from (33) then gives

$$\begin{aligned} & L^{-1}\bar{M}(\alpha\widetilde{M}+\Theta_z)^{-1}\mathbf{w}_2 + L^{-1}\mathbf{w}_3 = \mathbf{v}_1 \\ \Rightarrow \quad & L^{-1}(\bar{M}\mathbf{v}_2 + \mathbf{w}_3) = \mathbf{v}_1, \end{aligned}$$

and applying this within the last equation in (33) gives

$$\begin{aligned} & -\widehat{S}_\Pi^{-1}\mathbf{w}_1 + \widehat{S}_\Pi^{-1}(M+\Theta_y)L^{-1}\bar{M}(\alpha\widetilde{M}+\Theta_z)^{-1}\mathbf{w}_2 + \widehat{S}_\Pi^{-1}(M+\Theta_y)L^{-1}\mathbf{w}_3 = \mathbf{v}_3 \\ \Rightarrow \quad & -\widehat{S}_\Pi^{-1}\mathbf{w}_1 + \widehat{S}_\Pi^{-1}(M+\Theta_y)\left(L^{-1}\bar{M}(\alpha\widetilde{M}+\Theta_z)^{-1}\mathbf{w}_2 + L^{-1}\mathbf{w}_3\right) = \mathbf{v}_3 \\ \Rightarrow \quad & \widehat{S}_\Pi^{-1}((M+\Theta_y)\mathbf{v}_1 - \mathbf{w}_1) = \mathbf{v}_3. \end{aligned}$$

Thus, we need to approximately solve with $\widehat{S}_\Pi$, L, and $\alpha\widetilde{M}+\Theta_z$, which are all invertible matrices, to apply the preconditioner. We now briefly discuss our choice of $\widehat{S}_\Pi$. We suggest a matching strategy as above, to write

$$S_\Pi = L^T + (M+\Theta_y)L^{-1}\bar{M}(\alpha\widetilde{M}+\Theta_z)^{-1}\bar{M}^T \approx (L^T+M_l)L^{-1}(L+M_r) = \widehat{S}_\Pi,$$

where

$$M_l L^{-1} M_r \approx (M+\Theta_y)L^{-1}\bar{M}(\alpha\widetilde{M}+\Theta_z)^{-1}\bar{M}^T.$$

Such an approximation may be achieved if, for example,

$$M_l = M+\Theta_y, \qquad M_r \approx \bar{M}(\alpha\widetilde{M}+\Theta_z)^{-1}\bar{M}^T.$$

We take a matrix based on the approximation $\widehat{M}$ from the previous section to approximate M_r.

4.4 | Time-dependent problems

We may also apply our methodology to design preconditioners for time-dependent problems. For instance, consider minimizing the cost functional

$$\mathcal{F}(\mathrm{y},\mathrm{u}) = \frac{1}{2}\|\mathrm{y}-\mathrm{y_d}\|^2_{L^2(\Omega\times(0,T))} + \frac{\alpha}{2}\|\mathrm{u}\|^2_{L^2(\Omega\times(0,T))} + \beta\|\mathrm{u}\|_{L^1(\Omega\times(0,T))},$$

subject to the PDE $\mathrm{y}_t - \Delta\mathrm{y} = \mathrm{u}+\mathrm{f}$ on the space–time interval $\Omega\times(0,T)$, along with suitable boundary and initial conditions.

With the backward Euler method used to handle the time derivative, the matrix within the system to be solved is of the form

$$\mathcal{A} = \begin{bmatrix} \tau\mathcal{M}_c+\Theta_y & 0 & \mathcal{L}^T \\ 0 & \alpha\tau\widetilde{\mathcal{M}}_c+\Theta_z & -\tau\bar{\mathcal{M}}^T \\ \mathcal{L} & -\tau\bar{\mathcal{M}} & 0 \end{bmatrix}, \tag{34}$$

with τ as the time step used.

The matrix $\mathcal{M}_c$ is a block diagonal matrix consisting of multiples of mass matrices on each block diagonal corresponding to each time step, depending on the quadrature rule used to approximate the cost functional in the time domain. For example, if a trapezoidal rule is used, then $\mathcal{M}_c = \mathrm{blkdiag}(\frac{1}{2}M, M, \ldots, M, \frac{1}{2}M)$, and if a rectangle rule is used, then

$\mathcal{M}_c = \mathcal{M} := \text{blkdiag}(M, M, \ldots, M, M)$. The matrix $\mathcal{L}$ is a block-lower triangular matrix representing the all-at-once Euler discretization, with $M + \tau L$ appearing on each block diagonal and $-M$ on each block subdiagonal. Furthermore, we have

$$\widetilde{\mathcal{M}}_c = \begin{bmatrix} \mathcal{M}_c & -\mathcal{M}_c \\ -\mathcal{M}_c & \mathcal{M}_c \end{bmatrix}, \qquad \bar{\mathcal{M}} = [\, \mathcal{M} \;\; -\mathcal{M} \,].$$

We now consider saddle-point preconditioners for matrix (34). We may apply a block triangular preconditioner of the form

$$\mathcal{P}_T = \begin{bmatrix} \tau\mathcal{M}_c + \Theta_y & 0 & 0 & 0 \\ 0 & \alpha\tau\mathcal{D}_{M_c} + \Theta_w & -\alpha\tau\mathcal{D}_{M_c} & 0 \\ 0 & -\alpha\tau\mathcal{D}_{M_c} & \alpha\tau\mathcal{D}_{M_c} + \Theta_v & 0 \\ \mathcal{L} & -\tau\mathcal{M} & \tau\mathcal{M} & -\widehat{\mathcal{S}} \end{bmatrix}$$

or an analogous block diagonal preconditioner, where $\mathcal{D}_{M_c} := \text{diag}(\mathcal{M}_c)$, the matrix $\tau\mathcal{M}_c + \Theta_y$ can be approximately inverted by applying Chebyshev semi-iteration to the matrices arising at each time step, and $\widehat{\mathcal{S}}$ is an approximation of the Schur complement

$$\mathcal{S} = \mathcal{L}(\tau\mathcal{M}_c + \Theta_y)^{-1}\mathcal{L}^T + \frac{\tau}{\alpha}\mathcal{M}\mathcal{M}_c^{-1}\mathcal{M} - \frac{1}{\alpha^2}\mathcal{M}\mathcal{M}_c^{-1}\left(\Theta_w^{-1} + \Theta_v^{-1} + \frac{1}{\alpha\tau}\mathcal{M}_c^{-1}\right)\mathcal{M}_c^{-1}\mathcal{M}.$$

We select the approximation

$$\widehat{\mathcal{S}} = (\mathcal{L} + \widehat{\mathcal{M}})(\tau\mathcal{M}_c + \Theta_y)^{-1}(\mathcal{L} + \widehat{\mathcal{M}})^T,$$

using the same reasoning as in Section 4.2, where

$$\widehat{\mathcal{M}} = \left[\frac{\tau}{\alpha}\mathcal{D}_M^2\mathcal{D}_{M_c}^{-1} - \frac{1}{\alpha^2}\mathcal{D}_M^2\mathcal{D}_{M_c}^{-2}\left(\Theta_w^{-1} + \Theta_v^{-1} + \frac{1}{\alpha\tau}\mathcal{D}_{M_c}^{-1}\right)\right]^{1/2}(\tau\mathcal{D}_{M_c} + \Theta_y)^{1/2},$$

with $\mathcal{D}_M := \text{diag}(\mathcal{M})$. Within the numerical experiments of the forthcoming section, we apply the preconditioning strategy that arises from the working above.

5 | NUMERICAL EXPERIMENTS

We now implement the interior-point algorithm described in the Appendix, using MATLAB R2017b on an Intel Xeon computer with a 2.40-GHz processor and 250-GB RAM. Within the algorithm, we employ the preconditioned MINRES[32] and GMRES[33] methods with the following preconditioners:

- IPM-GMRES-$\mathcal{P}_T$: GMRES and block triangular preconditioner $\mathcal{P}_T$,
- IPM-MINRES-$\mathcal{P}_D$: MINRES with block diagonal preconditioner $\mathcal{P}_D$, and
- IPM-GMRES-$\mathcal{P}_\Pi$: GMRES and nonsymmetric preconditioner $\mathcal{P}_\Pi$.

Regarding the parameters listed in the Appendix, we use $\alpha_0 = 0.995$ and $\epsilon_p = \epsilon_{\text{d}} = \epsilon_c = 10^{-6}$. For the barrier reduction parameter σ, we consider for each class of problems tested a value that ensures a smooth decrease in the complementarity measure ξ_c^k in (21), that is, $\|\xi_c^k\| = \mathcal{O}(\mu^k)$. This way, the number of nonlinear (interior-point) iterations typically depends only on σ. We solve the linear matrix systems to a (relative unpreconditioned residual norm) tolerance of 10^{-10}.

We apply the IFISS software package[47,48] to build the relevant finite element matrices for the two-dimensional examples shown in this section and use the DEAL.II library[49] in the three-dimensional case. In each case, we utilize $Q1$ finite elements for the state, control, and adjoint variables.

We apply 20 steps of Chebyshev semi-iteration to approximate the inverse of mass matrices, as well as mass matrices plus positive diagonal matrices, whenever they arise within the preconditioners. Applying the approximate inverses of the Schur complement approximations derived for each of our preconditioners requires solving for matrices of the form $L + \widehat{M}$ and its transpose. For this, we typically utilize three V-cycles of the algebraic multigrid routine HSL-MI20,[50] with a Gauss–Seidel coarse solver, and apply five steps of pre- and post-smoothing. For tests on the simpler Poisson problem in Section 5.1, we use two V-cycles and three steps of pre- and post-smoothing. For time-dependent problems, we also use Chebyshev semi-iteration and algebraic multigrid within the preconditioner, but are required to apply the methods to matrices arising from each time step.

In the forthcoming tables of results, we report the average number of linear (MINRES or GMRES) iterations AV-LI and the average CPU time AV-CPU. The overall number of nonlinear (interior-point) iterations NLI is specified in the table captions. We believe these demonstrate the effectiveness of our proposed interior-point and preconditioning approaches, as well as the robustness of the overall method, for a range of PDEs, matrix dimensions, and parameters involved in the problem setup.

5.1 | A Poisson problem

We first examine an optimization problem involving Poisson's equation, investigating the behavior of the IPM and our proposed preconditioners.

5.1.1 | Two-dimensional case

We focus initially on the performance of our solvers for the two-dimensional Poisson problem, employing both IPM-GMRES-$\mathcal{P}_T$ and IPM-MINRES-$\mathcal{P}_D$ methods, as well as considering some sparsity issues. We set the box constraints for the control to be $\mathrm{u_a} = -2$, $\mathrm{u_b} = 1.5$, and the desired state $\mathrm{y_d} = \sin(\pi \mathrm{x}_1)\sin(\pi \mathrm{x}_2)$, with x_i denoting the ith spatial variable. Figure 2 displays the computed optimal controls for this problem for a particular setup on the domain $\Omega = (0,1)^2$, for both $\beta = 5 \times 10^{-2}$ and $\beta = 5 \times 10^{-3}$ as well as $\alpha = 10^{-2}$. Table 1 reports the level of sparsity in the computed solution, as well as its ℓ_1-norm, when varying the regularization parameters α and β. The value of SPARSITY in the table is computed by measuring the percentage of components of u, which are below a certain threshold (10^{-2} in our case; see, e.g., the work of Wen et al.[51]). We observe that our algorithm reliably computes sparse controls, and as expected, the sparsity of the solution increases when β is correspondingly increased.

In Table 2, we compare the performance of preconditioners $\mathcal{P}_T$ and $\mathcal{P}_D$ within the IPM, varying the spatial mesh size $h = 2^{-i}, i = 6, 7, 8, 9$, corresponding to $n = 4225, 16641, 66049, 263169$ degrees of freedom, as well as the regularization parameter α, while fixing the value $\beta = 10^{-2}$. (Table 1 indicates that this value of β gives rise to a computationally interesting case.) We set $\sigma = 0.2$ and take nine interior-point iterations with a final value $\mu^k = 5 \times 10^{-7}$. Figure 3 provides a representation of the typical convergence behavior for the feasibilities ξ_p^k, ξ_d^k and complementarity ξ_c^k, together with the decrease of μ^k with this value of σ. The reported results demonstrate good robustness of both preconditioners with respect to both h and α in terms of linear iterations and CPU time, with IPM-GMRES-$\mathcal{P}_T$ outperforming IPM-MINRES-$\mathcal{P}_D$ in each measure. Despite the fact that the value of AV-LI is constant in both implementations, we observe that when using IPM-MINRES-$\mathcal{P}_D$, the number of preconditioned MINRES iterations slightly increases as $\mu^k \to 0$, as many entries of Θ_z tend to zero. On the contrary, the number of preconditioned GMRES iterations hardly varies with k.

We also investigate the performance of our IPMs on purely sparse problems. We test the solver for very small values of α and find that both the IPM and the preconditioner technique work well. In fact, the number of nonlinear IP steps NLI does not vary with α, and the preconditioner is still robust with respect to nearly zero values of α. For the discretization

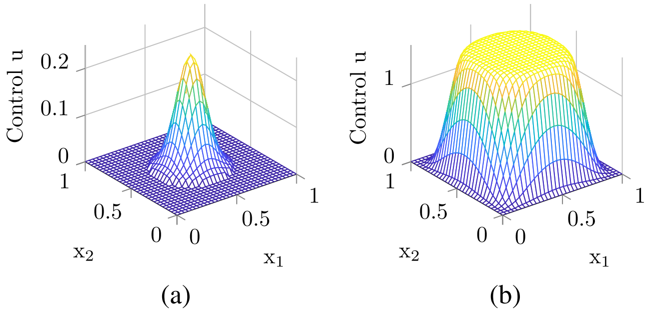


FIGURE 2 Poisson problem: computed solutions for the control u, for two values of β. (a) Control u, $\beta = 5 \times 10^{-2}$. (b) Control u, $\beta = 5 \times 10^{-3}$

TABLE 1 Poisson problem: sparsity features of the computed optimal control, for a range of α and β, and mesh size $h = 2^{-5}$

	$\beta = 10^{-1}$		$\beta = 10^{-2}$		$\beta = 10^{-3}$	
	SPARSITY	$\|\|\boldsymbol{u}\|\|_1$	SPARSITY	$\|\|\boldsymbol{u}\|\|_1$	SPARSITY	$\|\|\boldsymbol{u}\|\|_1$
$\alpha = 10^{-2}$	99%	3	15%	7×10^2	12%	1×10^3
$\alpha = 10^{-4}$	100%	2	38%	9×10^2	12%	1×10^3
$\alpha = 10^{-6}$	100%	2	39%	9×10^2	12%	1×10^3

TABLE 2 Poisson problem: average Krylov iterations and CPU times for the problem with control constraints, for a range of h and α, $\beta = 10^{-2}$, $\sigma = 0.25$, NLI = 12

		IPM-GMRES-$\mathcal{P}_T$		IPM-MINRES-$\mathcal{P}_D$	
$h = 2^{-l}$	$\log_{10}\alpha$	AV-LI	AV-CPU	AV-LI	AV-CPU
6	−2	9.4	0.1	20.9	0.3
	−4	7.9	0.1	16.1	0.2
	−6	7.9	0.1	15.6	0.2
7	−2	8.9	0.4	19.8	0.9
	−4	8.3	0.4	16.8	0.8
	−6	8.3	0.4	16.3	0.8
8	−2	9.1	1.6	19.8	3.5
	−4	8.7	1.5	17.7	3.3
	−6	8.8	1.6	17.3	3.2
9	−2	9.6	8.0	20.6	16.7
	−4	9.3	7.7	18.7	15.5
	−6	9.3	7.6	18.0	14.6

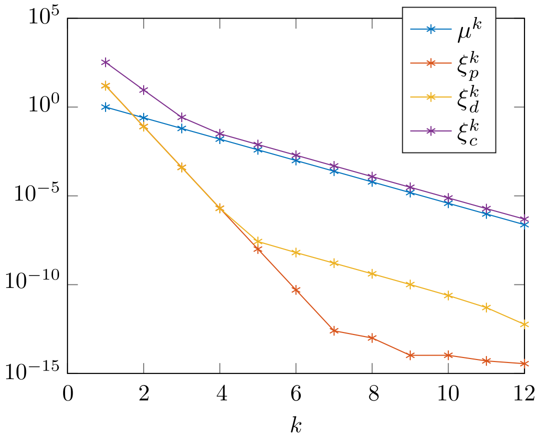


FIGURE 3 Typical convergence history of the relevant quantities $\mu^k, \xi_p^k, \xi_d^k, \xi_c^k$

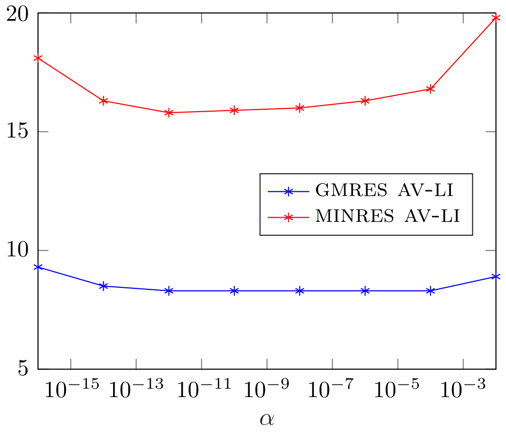


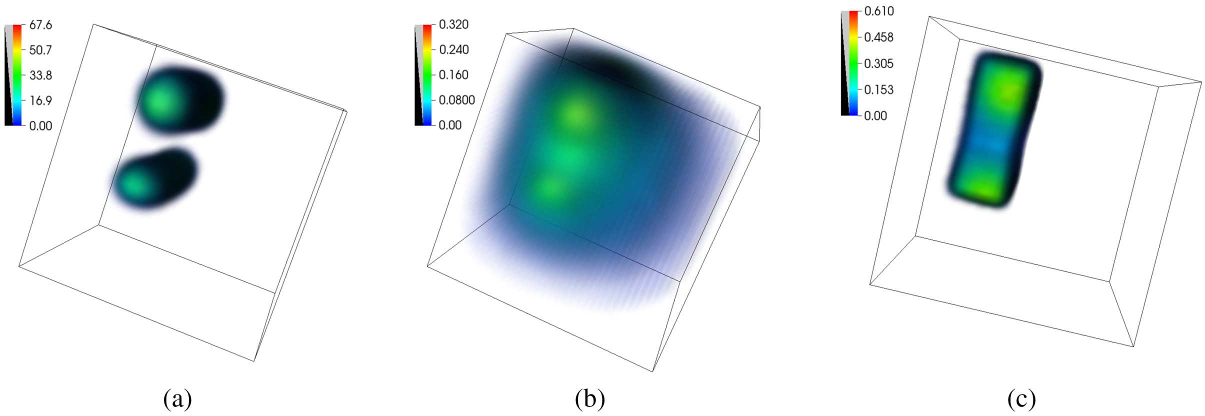
FIGURE 4 Poisson problem: average Krylov iterations for a range of α, with $h = 2^{-7}$, $\beta = 10^{-2}$, $\sigma = 0.25$ (NLI = 12). GMRES AV-LI = average number of linear iterations for generalized minimal residual; MINRES AV-LI = average number of linear iterations for minimal residual

level $l = 7$, we report in Figure 4 the average number of linear iterations AV-LI (of both GMRES and MINRES) versus α with values $\alpha = 10^{-2i}$, $i = 1, 2, \ldots, 8$.

As a final validation of the general framework outlined, we report in Table 3 results obtained when imposing both control and state constraints within the Poisson setting described above. In particular, we set $\mathrm{y_a} = -0.1$, $\mathrm{y_b} = 0.8$, $\mathrm{u_a} = -1$, $\mathrm{u_b} = 15$ and test the most promising implementation of the IPM, that is, the IPM-GMRES-$\mathcal{P}_T$ routine, while varying h and α. The reported values of AV-LI confirm the robustness of the preconditioning strategy proposed.

TABLE 3 (Left) Poisson problem: average Krylov iterations and CPU times for the problem with both control and state constraints, for a range of h and α, $\beta = 10^{-2}$, $\sigma = 0.3$ (NLI = 17). (Right) Three-dimensional Poisson problem with partial observations: average Krylov iterations and CPU times for the problem, for a range of numbers of degrees of freedom in each variable n and α, $\beta = 10^{-3}$, $\sigma = 0.25$ (NLI = 12)

		IPM-GMRES-$\mathcal{P}_T$				IPM-GMRES-$\mathcal{P}_\Pi$	
$h = 2^{-l}$	$\log_{10}\alpha$	AV-LI	AV-CPU	n	$\log_{10}\alpha$	AV-LI	AV-CPU
6	−2	15.5	0.2	729	−2	11.9	0.1
	−4	12.3	0.2		−4	13.1	0.1
	−6	10.6	0.1		−6	13.1	0.1
7	−2	14.6	0.7	4913	−2	11.8	0.3
	−4	12.3	0.6		−4	12.1	0.3
	−6	10.4	0.5		−6	12.1	0.3
8	−2	14.4	2.5	35937	−2	11.9	2.3
	−4	12.2	2.2		−4	11.9	2.3
	−6	10.6	1.9		−6	11.9	2.3
9	−2	13.8	10.9	274625	−2	13.1	21.1
	−4	11.6	9.4		−4	13.1	21.5
	−6	10.7	8.7		−6	13.1	21.3



(a) (b) (c)

FIGURE 5 Three-dimensional Poisson problem with partial observations: computed solutions for the control, state, and desired state. (a) Computed control u. (b) Computed state y. (c) Desired state y_d

5.1.2 | Three-dimensional case with partial observations

We also wish to present results for the case of partial observations, paired with a three-dimensional example involving Poisson's equation on $\Omega = (0,1)^3$. The desired state is illustrated in Figure 5. We use the preconditioner $\mathcal{P}_\Pi$, as the observation domain Ω_1 is given by $0.2 < x_1 < 0.4$, $0.4 < x_2 < 0.9$, $0 \le x_3 \le 1$, and therefore, the (1, 1)-block of matrix (15) can be singular. The results for the computation with $\alpha = 10^{-5}$, $\beta = 10^{-3}$, and without additional box constraints, are also presented in Figure 5, with the discretization involving 35937 degrees of freedom. To illustrate the performance of the proposed preconditioner $\mathcal{P}_\Pi$ with respect to changes in the parameter regimes, in Table 3, we provide results for a computation involving sparsity constraints applied to the control, as well as partial observation of the state, and set $u_a = -2$, $u_b = 1.5$. Again, the results are very promising, and a large degree of robustness is achieved.

5.2 | A convection–diffusion problem

We next consider the optimal control of the convection–diffusion equation given by $-\varepsilon\Delta y + \vec{w}\cdot\nabla y = u$ on the domain $\Omega = (0,1)^2$, with the wind vector $\vec{w}$ given by $\vec{w} = [2x_2(1-x_1^2), -2x_1(1-x_2^2)]^T$ and the bounds on the control given by $u_a = -2$ and $u_b = 1.5$. The desired state is here defined by $y_d = \exp(-64(x_1-0.5)^2 + (x_2-0.5)^2)$. Figure 6 displays the computed optimal controls for this problem for two values of β as well as $\alpha = 10^{-2}$.

The discretization is again performed using $Q1$ finite elements, while also employing the streamline-upwind Petrov–Galerkin[52] upwinding scheme as implemented in IFISS. The results of our scheme are given in Table 4, which again exhibit robustness with respect to h and α, while also performing well for both values of ε tested.

We now provide a numerical insight on the comparison between the proposed IPM approach and the commonly used semismooth Newton approach.[16] We therefore compare IPM-GMRES-$\mathcal{P}_T$ and the implementation SSN-GMRES-IPF of the global semismooth Newton method proposed for PDE-constrained optimization problems with sparsity-promoting terms in the work of Porcelli et al.[14] When using the SSN-GMRES-IPF approach, global convergence is attained using a

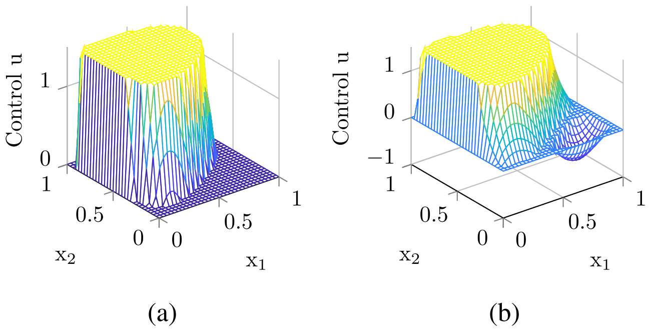


FIGURE 6 Convection–diffusion problem: computed solutions for the control u, for two values of β. (a) Control u, $\beta = 10^{-2}$. (b) Control u, $\beta = 10^{-3}$

TABLE 4 Convection–diffusion problem: average Krylov iterations and CPU times for the problem with control constraints, for a range of h and α, $\beta = 10^{-3}$, $\sigma = 0.25$ (NLI = 11) with $\varepsilon = 10^{-1}$, and $\sigma = 0.4$ (NLI = 16) with $\varepsilon = 10^{-2}$

		$\varepsilon = 10^{-1}$				$\varepsilon = 10^{-2}$			
		IPM-GMRES-$\mathcal{P}_T$		IPM-MINRES-$\mathcal{P}_D$		IPM-GMRES-$\mathcal{P}_T$		IPM-MINRES-$\mathcal{P}_D$	
$h = 2^{-l}$	$\log_{10}\alpha$	AV-LI	AV-CPU	AV-LI	AV-CPU	AV-LI	AV-CPU	AV-LI	AV-CPU
6	−2	9.4	0.2	21.1	0.5	11.2	0.5	25.8	1.1
	−4	8.3	0.2	18.2	0.4	10.5	0.5	23.2	1.0
	−6	8.2	0.2	17.8	0.4	10.5	0.5	23.5	1.0
7	−2	8.2	0.8	18.0	1.7	9.2	1.6	20.6	3.4
	−4	7.5	0.7	16.3	1.5	8.7	1.5	19.0	3.1
	−6	7.5	0.7	16.1	1.5	8.7	1.5	19.4	3.1
8	−2	7.5	2.7	16.3	5.6	8.0	3.8	17.1	7.9
	−4	7.0	2.5	15.1	5.2	7.7	3.7	16.4	7.5
	−6	7.0	2.5	14.8	5.1	7.7	3.7	16.4	7.5
9	−2	7.0	11.2	14.9	23.0	7.3	13.1	15.1	26.3
	−4	6.7	11.0	14.2	22.4	6.8	12.5	14.4	25.5
	−6	6.7	11.0	13.9	21.7	6.8	12.5	14.5	25.5

TABLE 5 Convection–diffusion problem: comparison between IPM-GMRES-$\mathcal{P}_T$ and SSN-GMRES-IPF in terms of the number of nonlinear iterations (NLI) and total CPU times (TCPU) for the problem with control constraints, for a range of h and α, $\beta = 10^{-3}$, $\varepsilon = 10^{-1}$

		IPM-GMRES-$\mathcal{P}_T$		SSN-GMRES-IPF	
$h = 2^{-l}$	$\log_{10}\alpha$	NLI	TCPU	NLI	TCPU
6	−2	11	2.8	5	4.2
	−4	11	2.5	19	27.9
	−6	11	2.4	>100	
	−8	11	2.4	> 100	
7	−2	11	9.4	5	14.0
	−4	11	8.7	18	101.9
	−6	11	8.7	> 100	
	−8	11	9.1	> 100	
8	−2	11	36.6	5	43.4
	−4	11	34.4	20	345.3
	−6	11	33.9	> 100	
	−8	11	33.8	> 100	
9	−2	11	155.9	5	147.3
	−4	11	149.8	21	1265.4
	−6	11	148.9	> 100	
	−8	11	149.6	> 100	

nonsmooth line-search strategy, and the linear systems arising in the linear algebra phase are solved by using preconditioned GMRES. We consider the 2 × 2 block formulation and an indefinite preconditioner available in factorized form.[14,17] Since the semismooth approach requires a diagonal mass matrix in the discretization of the complementarity conditions, in the experiments with SSN-GMRES-IPF, we use a lumped mass matrix. Table 5 collects results concerning the nonlinear behavior of the two methods: the number of nonlinear iterations (NLI) and the total CPU time (TCPU).

TABLE 6 Heat equation problem: average Krylov iterations and CPU times for the problem with control constraints, for a range of h, α, and τ, $\beta = 10^{-2}$, $\sigma = 0.25$ (NLI = 13)

		IPM-GMRES-$\mathcal{P}_T$					
		$\tau = 0.04$		$\tau = 0.02$		$\tau = 0.01$	
$h = 2^{-l}$	$\log_{10}\alpha$	AV-LI	AV-CPU	AV-LI	AV-CPU	AV-LI	AV-CPU
4	−2	13.9	0.6	13.1	1.0	13.1	2.2
	−4	13.3	0.5	12.2	1.0	12.3	2.0
	−6	12.8	0.5	12.0	1.0	12.0	2.0
5	−2	14.6	1.6	14.0	3.1	14.7	6.6
	−4	13.9	1.5	13.3	2.9	13.3	5.8
	−6	13.6	1.5	12.8	2.8	13.0	5.7
6	−2	15.5	5.9	14.6	11.4	15.4	23.7
	−4	14.8	5.8	14.0	10.6	14.0	21.7
	−6	14.6	5.5	13.8	10.6	13.9	21.5

We again highlight that the number of nonlinear interior-point iterations does not vary with α. In fact, the mildly aggressive choice of barrier reduction factor σ yields a low number of nonlinear iterations, even for limiting values of α. By contrast, SSN-GMRES-IPF struggles as $\alpha \to 0$. Furthermore, overall, the interior-point strategy outperforms the semismooth method in terms of total CPU time.

5.3 | A heat equation problem

To demonstrate the applicability of our methodology to time-dependent problems, we now perform experiments on an optimization problem with the heat equation acting as a constraint. We utilize the implicit Euler scheme on a time interval up to $T = 1$, for varying values of time step τ, and set a time-independent desired state to be $\mathrm{y_d} = \sin(\pi \mathrm{x}_1)\sin(\pi \mathrm{x}_2)$. We consider a control problem with full observations, with Table 6 illustrating the performance of the IPM and preconditioner $\mathcal{P}_T$ for varying mesh sizes and values of α, with fixed $\beta = 10^{-2}$. Considerable robustness is again achieved, particularly with respect to changes in the time step.

Remark 3. We emphasize that the robustness with respect to α, in terms of the number of nonlinear interior-point iterations, is a result of the suitable choices made for the barrier reduction factor σ. In particular, in all the test cases discussed, the choice of σ is mildly aggressive (from 0.2 to 0.4 in the most difficult cases), yielding a low number of nonlinear iterations, even for limiting values of α. By contrast, a semismooth Newton approach globalized with a line-search strategy may perform poorly as $\alpha \to 0$, as observed above.

6 | CONCLUSIONS

We have presented a new IPM for PDE-constrained optimization problems that include additional box constraints on the control variable, as well as possibly the state variable, and a sparsity-promoting L^1-norm term for the control within the cost functional. We incorporated a splitting of the control into positive and negative parts, as well as a suitable nodal quadrature rule, to linearize the L^1-norm, and considered preconditioned iterative solvers for the Newton systems arising at each interior-point iteration. Through theoretical justification for our approximations of the $(1, 1)$-block and Schur complement of the Newton systems, as well as numerical experiments, we have demonstrated the effectiveness and robustness of our approach, which may be applied within symmetric and nonsymmetric Krylov methods, for a range of steady and time-dependent PDE-constrained optimization problems. As an outlook, the implementation of our algorithms within faster (compiled) code would allow the applicability of our robust preconditioning schemes to even larger matrix systems.

ACKNOWLEDGEMENTS

The authors are grateful to two anonymous referees for their careful reading of the manuscript and their helpful comments. J. W. Pearson gratefully acknowledges support from the Engineering and Physical Sciences Research Council Fellowship EP/M018857/2 and a Fellowship from The Alan Turing Institute in London. The works of M. Porcelli and M. Stoll were partially supported by the DAAD-MIUR Joint Mobility Program 2018–2020 (Grant 57396654). The work of M. Porcelli was also partially supported by the National Group of Computing Science (GNCS-INDAM).

CONFLICT OF INTEREST

This work does not have any conflicts of interest.

ORCID

Margherita Porcelli https://orcid.org/0000-0003-0183-1204
Martin Stoll https://orcid.org/0000-0003-0951-4756

REFERENCES

1. Hinze M, Pinnau R, Ulbrich M, Ulbrich S. Optimization with PDE constraints. Dordrecht, The Netherlands: Springer Netherlands; 2009. (Mathematical modelling: theory and applications).
2. Ito K, Kunisch K. Lagrange multiplier approach to variational problems and applications. Philadelphia, PA: Society for Industrial and Applied Mathematics; 2008. (Advances in design and control, society for industrial and applied mathematics; No. 15).
3. Tröltzsch F. Optimal control of partial differential equations: Theory, methods and applications. Providence, RI: American Mathematical Society; 2010.
4. Hinze M. Optimal and instantaneous control of the instationary Navier–Stokes equations [Dissertation]. Berlin, Germany: Technische Universität Dresden; 2000.
5. De los Reyes JC, Schönlieb C-B. Image denoising: Learning the noise model via nonsmooth PDE-constrained optimization. Inverse Probl Imaging. 2013;7(4):1183–1214.
6. Stadler G. Elliptic optimal control problems with L^1-control cost and applications for the placement of control devices. Comput Optim Appl. 2009;44(2):159–181.
7. Song X, Chen B, Yu B. Error estimates for sparse optimal control problems by piecewise linear finite element approximation. arXiv preprint arXiv:1709.09539; 2017.
8. Song X, Chen B, Yu B. Mesh independence of an accelerated block coordinate descent method for sparse optimal control problems. arXiv preprint arXiv:1709.00005; 2017.
9. Song X, Chen B, Yu B. An efficient duality-based approach for PDE-constrained sparse optimization. Comput Optim Appl. 2018;69(2):461–500.
10. Wachsmuth G, Wachsmuth D. Convergence and regularization results for optimal control problems with sparsity functional. ESAIM: Control Optim Calc Var. 2011;17(3):858–886.
11. Casas E. Control of an elliptic problem with pointwise state constraints. SIAM J Control Optim. 1986;24(6):1309–1318.
12. Günther A, Hinze M, Tber MH. A posteriori error representations for elliptic optimal control problems with control and state constraints. In: Constrained optimization and optimal control for partial differential equations. Basel, Switzerland: Springer; 2012. p. 303–317.
13. Herzog R, Sachs E. Preconditioned conjugate gradient method for optimal control problems with control and state constraints. SIAM J Matrix Anal Appl. 2010;31(5):2291–2317.
14. Porcelli M, Simoncini V, Stoll M. Preconditioning PDE-constrained optimization with L^1-sparsity and control constraints. Comput Math Appl. 2017;74(5):1059–1075.
15. Bergounioux M, Ito K, Kunisch K. Primal-dual strategy for constrained optimal control problems. SIAM J Control Optim. 1999;37(4):1176–1194.
16. Hintermüller M, Ito K, Kunisch K. The primal-dual active set strategy as a semismooth Newton method. SIAM J Optim. 2003;13(3):865–888.
17. Porcelli M, Simoncini V, Tani M. Preconditioning of active-set Newton methods for PDE-constrained optimal control problems. SIAM J Sci Comput. 2015;37(5):S472–S502.
18. Herzog R, Obermeier J, Wachsmuth G. Annular and sectorial sparsity in optimal control of elliptic equations. Comput Optim Appl. 2015;62(1):157–180.
19. Herzog R, Stadler G, Wachsmuth G. Directional sparsity in optimal control of partial differential equations. SIAM J Control Optim. 2012;50(2):943–963.
20. Nocedal J, Wright SJ. Numerical optimization. New York, NY: Springer-Verlag; 2006. (Springer series in operations research and financial engineering).
21. Wright SJ. Primal-dual interior-point methods. Philadelphia, PA: Society for Industrial and Applied Mathematics; 1997.
22. Pearson JW, Gondzio J. Fast interior point solution of quadratic programming problems arising from PDE-constrained optimization. Numerische Mathematik. 2017;137(4):959–999.
23. Ulbrich S. Primal-dual interior-point methods for PDE-constrained optimization. Mathematical Programming. 2009;117(1-2):435–485.
24. Gondzio J. Interior point methods 25 years later. Eur J Oper Res. 2012;218(3):587–601.
25. Pearson JW, Wathen AJ. A new approximation of the Schur complement in preconditioners for PDE-constrained optimization. Numer Linear Algebra Appl. 2012;19:816–829.
26. Figueiredo MAT, Nowak RD, Wright SJ. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. IEEE J Sel Top Signal Process. 2007;1(4):586–597.
27. Vossen G, Maurer H. On L^1-minimization in optimal control and applications to robotics. Optim Control Appl Methods. 2006;27(6):301–321.
28. Wächter A, Biegler LT. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Mathematical Programming. 2006;106(1):25–57.

29. Rees T, Stoll M, Wathen A. All-at-once preconditioning in PDE-constrained optimization. Kybernetika. 2010;46:341–360.
30. Fountoulakis K, Gondzio J. A second-order method for strongly convex ℓ_1-regularization problems. Mathematical Programming. 2016;156(1-2):189–219.
31. Fountoulakis K, Gondzio J, Zhlobich P. Matrix-free interior point method for compressed sensing problems. Mathematical Programming Computation. 2014;6(1):1–31.
32. Paige CC, Saunders MA. Solution of sparse indefinite systems of linear equations. SIAM J Numer Anal. 1975;12(4):617–629.
33. Saad Y, Schultz MH. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J Sci Stat Comput. 1986;7(3):856–869.
34. Benzi M, Golub GH, Liesen J. Numerical solution of saddle point problems. Acta Numerica. 2005;14:1–137.
35. Ipsen ICF. A note on preconditioning non-symmetric matrices. SIAM J Sci Comput. 2001;23(2):1050–1051.
36. Kuznetsov YA. Efficient iterative solvers for elliptic finite element problems on nonmatching grids. Russ J Numer Anal Math Model. 1995;10:187–211.
37. Murphy MF, Golub GH, Wathen AJ. A note on preconditioning for indefinite linear systems. SIAM J Sci Comput. 2000;21(6):1969–1972.
38. Wathen AJ. Realistic eigenvalue bounds for the Galerkin mass matrix. IMA J Numer Anal. 1987;7(4):449–457.
39. Golub GH, Varga RS. Chebyshev semi-iterative methods, successive over-relaxation iterative methods, and second order Richardson iterative methods. I. Numerische Mathematik. 1961;3:147–156.
40. Golub GH, Varga RS. Chebyshev semi-iterative methods, successive over-relaxation iterative methods, and second order Richardson iterative methods. II. Numerische Mathematik. 1961;3:157–168.
41. Wathen A, Rees T. Chebyshev semi-iteration in preconditioning for problems including the mass matrix. Electron Trans Numer Anal. 2009;34:125–135.
42. Lu T-T, Shiou S-H. Inverses of 2 × 2 block matrices. Comput Math Appl. 2002;43(1-2):119–129.
43. Pearson JW, Stoll M, Wathen AJ. Regularization-robust preconditioners for time-dependent PDE-constrained optimization problems. SIAM J Matrix Anal Appl. 2012;33(4):1126–1152.
44. Pearson JW, Wathen AJ. Fast iterative solvers for convection-diffusion control problems. Electron Trans Numer Anal. 2013;40:294–310.
45. Benner P, Dolgov S, Onwunta A, Stoll M. Low-rank solvers for unsteady Stokes–Brinkman optimal control problem with random data. Comput Methods Appl Mech Eng. 2016;304:26–54.
46. Herzog R, Pearson JW, Stoll M. Fast iterative solvers for an optimal transport problem. Adv Comput Math. 2019;45:495–517.
47. Elman HC, Ramage A, Silvester DJ. Algorithm 866: IFISS, a Matlab toolbox for modelling incompressible flow. ACM Trans Math Softw. 2007;33(14). Article No. 14.
48. Elman HC, Ramage A, Silvester DJ. Incompressible flow and iterative solver software (IFISS). Version 3.5. 2018. https://personalpages.manchester.ac.uk/staff/david.silvester/ifiss/
49. Bangerth W, Hartmann R, Kanschat G. deal.II—A general-purpose object-oriented finite element library. ACM Trans Math Softw. 2007;33(4). Article No. 24.
50. Boyle J, Mihajlović MD, Scott JA. HSL_MI20: an efficient AMG preconditioner for finite element problems in 3D. Int J Numer Methods Eng. 2010;82(1):64–98.
51. Wen Z, Yin W, Goldfarb D, Zhang Y. A fast algorithm for sparse reconstruction based on shrinkage, subspace optimization, and continuation. SIAM J Sci Comput. 2010;32(4):1832–1857.
52. Brooks AN, Hughes TJR. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations. Comput Methods Appl Mech Eng. 1982;32(1):199–259.

How to cite this article: Pearson JW, Porcelli M, Stoll M. Interior-point methods and preconditioning for PDE-constrained optimization problems involving sparsity terms. *Numer Linear Algebra Appl*. 2019;e2276. https://doi.org/10.1002/nla.2276

APPENDIX

INTERIOR-POINT ALGORITHM FOR QUADRATIC PROGRAMMING

In the algorithm below, we present the structure of the IPM that we apply within our numerical experiments, following the interior-point path-following scheme described in the work of Gondzio.[24] It is clear that the main computational effort arises from solving the Newton system (15) at each iteration.

Algorithm A.1: Interior Point Algorithm for Quadratic Programming

```
Parameters
    $\alpha_0 \in (0,1)$,   step-size factor to boundary
    $\sigma \in (0,1)$,   barrier reduction parameter
    $\epsilon_p,\ \epsilon_d,\ \epsilon_c$,   stopping tolerances
    Interior point method stops when $\left\|\xi_p^k\right\| \le \epsilon_p,\ \left\|\xi_d^k\right\| \le \epsilon_d,\ \left\|\xi_c^k\right\| \le \epsilon_c$
Initialize IPM
    Set the initial guesses for $y^0,\ z^0,\ p^0,\ \lambda_{y,a}^0,\ \lambda_{y,b}^0,\ \lambda_{z,a}^0,\ \lambda_{z,b}^0$
    Set the initial barrier parameter $\mu^0$
    Compute primal infeasibility $\xi_p^0$, dual infeasibility $\xi_d^0$, and complementarity gap $\xi_c^0$,
        as in (20)– (21) with $k = 0$
Interior Point Method
    while $\left(\left\|\xi_p^k\right\| > \epsilon_p$ or $\left\|\xi_d^k\right\| > \epsilon_d$ or $\left\|\xi_c^k\right\| > \epsilon_c\right)$
        Reduce barrier parameter $\mu^{k+1} = \sigma\mu^k$
        Solve Newton system (15) for primal-dual Newton direction $\Delta y,\ \Delta z,\ \Delta p$
        Use (16)– (19) to find $\Delta\lambda_{y,a},\ \Delta\lambda_{y,b},\ \Delta\lambda_{z,a},\ \Delta\lambda_{z,b}$
        Find $\alpha_P,\ \alpha_D$ s.t. bound constraints on primal and dual variables hold
        Set $\alpha_P = \alpha_0\alpha_P,\ \alpha_D = \alpha_0\alpha_D$
        Make step: $y^{k+1} = y^k + \alpha_P\Delta y,\ z^{k+1} = z^k + \alpha_P\Delta z,\ p^{k+1} = p^k + \alpha_D\Delta p$
            $\lambda_{y,a}^{k+1} = \lambda_{y,a}^k + \alpha_D\Delta\lambda_{y,a},\ \lambda_{y,b}^{k+1} = \lambda_{y,b}^k + \alpha_D\Delta\lambda_{y,b}$
            $\lambda_{z,a}^{k+1} = \lambda_{z,a}^k + \alpha_D\Delta\lambda_{z,a},\ \lambda_{z,b}^{k+1} = \lambda_{z,b}^k + \alpha_D\Delta\lambda_{z,b}$
        Update infeasibilities $\xi_p^{k+1},\ \xi_d^{k+1}$, and compute the complementarity gap $\xi_c^{k+1}$
            as in (20)– (21)
        Set iteration number $k = k + 1$
    end
```