FAST ITERATIVE SOLVERS FOR CONVECTION-DIFFUSION CONTROL PROBLEMS

JOHN W. PEARSON* AND ANDREW J. WATHEN†

Abstract. In this manuscript, we describe effective solvers for the optimal control of stabilized convection-diffusion problems. We employ the local projection stabilization, which we show to give the same matrix system whether the discretize-then-optimize or optimize-then-discretize approach for this problem is used. We then derive two effective preconditioners for this problem, the first to be used with MINRES and the second to be used with the Bramble-Pasciak Conjugate Gradient method. The key components of both preconditioners are an accurate mass matrix approximation, a good approximation of the Schur complement, and an appropriate multigrid process to enact this latter approximation. We present numerical results to demonstrate that these preconditioners result in convergence in a small number of iterations, which is robust with respect to the mesh size h, and the regularization parameter β, for a range of problems.

Key words. PDE-constrained optimization, convection-diffusion control, preconditioning, local projection stabilization, Schur complement.

AMS subject classifications. 49M25, 65F08, 65F10, 65N30.

1. Introduction.

Convection-diffusion problems descibe important physical processes such as contaminant transport. The numerical solution of such problems, in particular in the case of dominating convection, has attracted much attention, and it is now widely appreciated what role stabilization techniques have to play. In this manuscript, we consider not the solution of single convection-diffusion problems (we will call this the solution of the forward problem), but the control of such problems. That is to say we consider solution methods for control problems involving the convection-diffusion equation, together with suitable boundary conditions. In particular we will describe two preconditioned iterative solution methods for the fast solution of such control problems.

Control problems, or PDE-constrained optimization problems, for various partial differential equations have been the subject of much research (see for example the excellent book by Tröltzsch [20]), and there has been significant recent interest in preconditioning and iterative solvers for such problems: see for example [16, 18]. In all such problems there arises the issue of whether to firstly perform discretization before optimization of the resulting discrete problem, or to construct continuous optimality conditions and then discretize. For many PDE problems, in particular those which are self-adjoint, the two possible approaches of discretize-then-optimize and optimize-then-discretize generally give rise to the same discrete equations — that is the two steps commute.

For the convection-diffusion control problem, Heinkenschloss and co-workers [6, 9] have considered the popular SUPG stabilized finite element method of Hughes and Brooks [10], and have shown the significant extra difficulty in the case of the control problem as opposed to the forward problem. A key issue is consistency not just of the forward problem, but also of the adjoint problem. The SUPG method does not satisfy such adjoint-consistency in general, though for the forward problem it yields $\mathcal{O}(h^{3/2})$ accuracy when using bilinear finite elements for instance (see Theorem 3.6 of [7]). For

*Numerical Analysis Group, Mathematical Institute, University of Oxford, 24–29 St Giles', Oxford, OX1 3LB, UK (john.pearson@worc.ox.ac.uk),

†Numerical Analysis Group, Mathematical Institute, University of Oxford, 24–29 St Giles', Oxford, OX1 3LB, UK (wathen@maths.ox.ac.uk)

the control problem this lead to the issue that the discretize-then-optimize approach gives rise to symmetric discrete equations in which the discrete adjoint problem is not a consistent discretization of the continuous adjoint problem, and the optimize-then-discetize approach gives rise to different and non-symmetric discrete equations which do not therefore have the structure of a discrete optimization problem.

Here we employ the adjoint-consistent local projection stabilization approach described by [1, 2, 5] which ensures that the discretize and optimize steps commute. For this approach we are able to establish preconditioned iterative solvers for the control problem which have the attractive feature of giving convergence in a number of steps independent of the parameters of the problem (including the mesh size). With an appropriate multigrid process for the advection-diffusion problem which we describe, this leads to solvers of optimal computational complexity for the PDE-constrained optimzation problems involving the convection-diffusion problem.

2. Background. In this Section, we summarize theory that we will use when solving the convection-diffusion control problem. Firstly, we will detail a method for solving the forward problem, that is the convection-diffusion equation with no optimization. We will exploit aspects of this method when we wish to solve the control problem. Secondly, we will detail some properties of ideal preconditioners for saddle point systems. The convection-diffusion control problem has saddle point structure, as we will show in Section 3, so we will need to use the theory of saddle point systems in order to develop preconditioners for this problem as in Section 4.

2.1. Solution of the Convection-Diffusion Equation. We first consider the finite element solution of the convection-diffusion equation with Dirichlet boundary conditions

$$-\epsilon\nabla^2 y + \mathbf{w}\cdot\nabla y = g, \quad \text{in } \Omega, \tag{2.1}$$

$$y = f, \quad \text{on } \partial\Omega, \tag{2.2}$$

where the domain $\Omega \subset \mathbb{R}^d$, $d = 2$ or 3, has boundary $\partial\Omega$, $\epsilon > 0$, and $\mathbf{w}$ is a divergence-free vector (i.e. $\nabla\cdot\mathbf{w} = 0$).

The $-\epsilon\nabla^2 y$ term in the above equation denotes the diffusive element, and the $\mathbf{w}\cdot\nabla y$ term represents convection. As pointed out, for example in Chapter 3 of [7], convection typically plays a more significant physical role than diffusion, so $\epsilon \ll |\mathbf{w}|$ for many practical problems. However, this in turn makes the problem more difficult to solve [7, 14], as the solution procedure will need to be robust with respect to the direction of the wind $\mathbf{w}$ and any boundary or internal layers that form.

The finite element representation of the equations (2.1)–(2.2) is given by

$$\bar{K}\mathbf{y} = \mathbf{f}, \tag{2.3}$$

where $\mathbf{y} = \{Y_i\}_{i=1,...,n}$, with Y_i denoting the coefficients of the finite element solution $y_h = \sum_{i=1}^{n+n_\partial} Y_i\phi_i$ with interior finite element basis functions $\phi_1, ..., \phi_n$, and boundary basis functions $\phi_{n+1}, ..., \phi_{n+n_\partial}$. $\bar{K}$ and $\mathbf{f}$, as stated in (2.3), are defined by

$$\begin{aligned}
&\bar{K} = \epsilon K + N + T, \\
&K = \{k_{ij}\}_{i,j=1,...,n}, \quad k_{ij} = \int_\Omega \nabla\phi_i\cdot\nabla\phi_j \; \mathrm{d}\Omega, \\
&N = \{\tilde{n}_{ij}\}_{i,j=1,...,n}, \quad \tilde{n}_{ij} = \int_\Omega (\mathbf{w}\cdot\nabla\phi_j)\phi_i \; \mathrm{d}\Omega.
\end{aligned}$$

Here, T is a matrix corresponding to the stabilization strategy used (which depends on the step size h, a stabilization parameter δ and an orthogonal projection operator π_h), and $\mathbf{f}$ is a vector corresponding to the function f (and sometimes the stabilization as well). Note that K is a *stiffness matrix*, a commonly occuring matrix in finite element problems. We discuss the definitions of T and $\mathbf{f}$ for two different stabilization methods in Section 3.1 (and note that $T = 0$ if no stabilization is used).

For the remainder of this Section, we briefly detail a method described in [7] for solving the problem (2.3), as we will use aspects of this method in our solvers for the convection-diffusion control problem in Section 4.

The method discussed in [7] for solving (2.1)–(2.2) is a preconditioned GMRES method with preconditioner that is the geometric multigrid process described in [14]. The multigrid process contains standard prolongation and restriction operators, but there are two major differences between it and a more typical multigrid process:

- **Construction of the coarse grid operator** – In most geometric multigrid algorithms, the construction of a coarse grid operator is carried out using the scaled Galerkin coarse grid operator (that is $\bar{K}_{\text{coarse}} = R\bar{K}_{\text{fine}}P$, where P is the projection operator and R the restriction operator). However, here, the coarse grid operator is *explicitly constructed* on all grids for which it is required. This will involve constructing the matrices K, N and T on each sub-grid, and will involve different stabilization parameters δ for each grid.
- **Pre- and post- smoothing** – The smoothing strategy we employ is *block Gauss-Seidel smoothing*, applied in each direction to take account of all possible wind directions, that is to say we employ 4 (2 pre- and 2 post-) smoothing steps for a two dimensional problem, and 6 smoothing steps for a three dimensional problem. This strategy is shown to be effective for a wide range of problems with our formulation, as demonstrated in Chapter 4 of [7], and [14].

2.2. Saddle Point Systems. The convection-diffusion control problem that we introduce in Section 3 is of *saddle point* structure; that is, it is of the form

$$\underbrace{\begin{bmatrix} A & B^T \\ B & C \end{bmatrix}}_{\mathcal{A}} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}, \tag{2.4}$$

where $A \in \mathbb{R}^{m\times m}$, $B \in \mathbb{R}^{q\times m}$ and $C \in \mathbb{R}^{q\times q}$, with $m \geq q$. For an overview of properties and solution methods for such systems, we refer the reader to [3].

In [11], it is demonstrated that two effective preconditioners for $\mathcal{A}$ are given by

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

where S is the (negative) *Schur complement* defined by $S = -C + BA^{-1}B^T$. The reason these preconditioners are so potent is that the spectra of $\mathcal{P}_1^{-1}\mathcal{A}$ and $\mathcal{P}_2^{-1}\mathcal{A}$ are given by

$$\lambda(\mathcal{P}_1^{-1}\mathcal{A}) = \{\frac{1}{2}(1-\sqrt{5}), 1, \frac{1}{2}(1+\sqrt{5})\}, \quad \lambda(\mathcal{P}_2^{-1}\mathcal{A}) = \{1\},$$

so long as $\mathcal{P}_1^{-1}\mathcal{A}$ and $\mathcal{P}_2^{-1}\mathcal{A}$ are nonsingular [11, 13].

Now, $\mathcal{P}_1^{-1}\mathcal{A}$ constructed in this way is diagonalizable but $\mathcal{P}_2^{-1}\mathcal{A}$ is not, so if we apply a Krylov subspace method with $\mathcal{A}$ preconditioned by $\mathcal{P}_1$ or $\mathcal{P}_2$, we will obtain termination in 3 and 2 iterations respectively [11]. Of course, the preconditioners $\mathcal{P}_1$ and $\mathcal{P}_2$ are not practical preconditioners, as the exact inverses of A and S will need to be enforced in each case (which is particularly problematic as even when A and B are sparse, S is generally dense).

However, if we were able to construct effective approximations to A and S, $\hat{A}$ and $\hat{S}$ say, and employ the preconditioners

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

we would be likely to obtain convergence of the appropriate Krylov subspace method in few iterations. In Section 4, we derive two preconditioners based on $\hat{\mathcal{P}}_1$ and $\hat{\mathcal{P}}_2$ for the convection-diffusion control problem.

Clearly, these preconditioners will have to be incorporated into different Krylov subspace methods. The block diagonal preconditioner $\hat{P}_1$ is symmetric positive definite, and so a natural choice is the MINRES algorithm [12, 16]. By contrast, the block triangular preconditioner $\hat{P}_2$ is nonsymmetric positive definite, and so the same algorithm cannot be used. However, as described in [4, 17, 19] for example, $\hat{\mathcal{P}}_2^{-1}\mathcal{A}$ is symmetric positive definite in the inner product $\mathcal{H}$ defined by $\langle \mathbf{u}, \mathbf{v} \rangle_{\mathcal{H}} = \mathbf{u}^T \mathcal{H} \mathbf{v}$, where

$$\mathcal{H} = \begin{bmatrix} A - \gamma \hat{A} & 0 \\ 0 & \hat{S} \end{bmatrix},$$

where γ is a scaling constant applied to ensure that $A - \gamma\hat{A}$ is positive definite. Hence it is possible to use a non-standard Conjugate Gradient method with the $\mathcal{H}$-inner product; this is often referred to as the *Bramble-Pasciak Conjugate Gradient* method.

3. The Convection-Diffusion Control Problem. For the remainder of this paper, we will be considering the distributed convection-diffusion control problem

$$\begin{aligned} \min_{y,u} \quad & \frac{1}{2}\|y - \hat{y}\|^2_{L_2(\Omega)} + \frac{\beta}{2}\|u\|^2_{L_2(\Omega)} \\ \text{s.t.} \quad & -\epsilon\nabla^2 y + \mathbf{w}\cdot\nabla y = u, \quad \text{in } \Omega, \\ & y = f, \quad \text{on } \partial\Omega, \end{aligned} \tag{3.1}$$

where y denotes the *state* with $\hat{y}$ some desired state, u denotes the *control*, and $\beta > 0$ is a regularization parameter (sometimes known as the *Tikhonov parameter*).

We employ a finite element method to solve the problem, that is we write

$$y_h = \sum_{i=1}^{n+n_\partial} Y_i\phi_i, \quad u_h = \sum_{i=1}^{n} U_i\phi_i, \quad p_h = \sum_{i=1}^{n} P_i\phi_i,$$

where p denotes the Lagrange multiplier we use. Note that we discretize the state y, the control u and the Lagrange multiplier p using the same basis functions here. Note also that the coefficients $Y_{n+1}, ..., Y_{n+n_\partial}$ are trivially obtained by considering the specified Dirichlet boundary condition $y = f$.

For the rest of this Section, we define $\mathbf{y}$, $\mathbf{u}$ and $\mathbf{p}$ as follows:

$$\mathbf{y} = \{Y_i\}_{i=1,...,n+n_\partial}, \quad \mathbf{u} = \{U_i\}_{i=1,...,n}, \quad \mathbf{p} = \{P_i\}_{i=1,...,n}.$$

3.1. Stabilization of the Control Problem. One important consideration when solving the convection-diffusion control problem (or indeed the convection-diffusion problem itself) is that of stabilizing the problem. It is well-known that without any form of stabilization, accurate solution of the convection-diffusion equation [7, 14] and the convection-diffusion control problem [2, 9] is compromised due to the formation of layers in the approximate solution, potentially leading to large errors for small ϵ.

One popular method of avoiding this problem is by using the *Streamline Upwind Petrov-Galerkin* (SUPG) stabilization, which was introduced in [10] and discussed further in literature such as [7, 9, 15]. For the forward problem, using this stabilization would result in a system of the form (2.3), with K and N as above, and

$$\begin{aligned} T = \{\tau^{\delta}_{h,ij}\}_{i,j=1,\ldots,n}, \quad \tau^{\delta}_{h,ij} &= \delta \int_{\Omega} (\mathbf{w}\cdot\nabla\phi_i)(\mathbf{w}\cdot\nabla\phi_j)\ \mathrm{d}\Omega \\ &\qquad - \epsilon\delta \sum_k \int_{\Delta_k} (\nabla^2\phi_i)(\mathbf{w}\cdot\nabla\phi_j)\ \mathrm{d}\Omega, \\ \mathbf{f} = \{f_i\}_{i=1,\ldots,n}, \quad f_i &= \int_{\Omega} f\phi_i\ \mathrm{d}\Omega + \delta\int_{\Omega} f\mathbf{w}\cdot\nabla\phi_i\ \mathrm{d}\Omega, \end{aligned}$$

with stabilization parameter δ, and Δ_k denoting the k-th element in our finite element discretization. It is well-recognised that this method is effective for solving the forward problem (see Chapters 3 and 4 of [7] for instance). However, for the convection-diffusion control problem, problems arise – the matrix systems that we obtain when we use the *discretize-then-optimize* and *optimize-then-discretize* formulations of Sections 3.2 and 3.3 do not commute [15]. This is a difficulty as we would then have to choose between solving the discretize-then-optimize matrix system, which would not be strongly consistent (meaning the solutions to the optimization problem would not satify all the optimality conditions), or the optimize-then-discretize system, which is non-symmetric and so is not the optimality system for any finite dimensional problem. Further, the non-symmetry of the matrix system that arises using the optimize-then-discretize approach means that we cannot apply the methodology detailed in Section 4 to solve it, as our methods depend on the matrix being symmetric. It is also believed that applying SUPG to the optimal control problem will guarantee at most first-order accuracy in the solution [9].

To deal with these two problems, we now introduce the *local projection stabilization* (LPS) method, which is discussed in [2, 8] for example. Applying this stabilization for the forward problem again yields a matrix system of the form (2.3), with K and N as above and

$$\begin{aligned} T = \{\tau^{\delta}_{h,ij}\}_{i,j=1,\ldots,n}, \quad \tau^{\delta}_{h,ij} &= \delta \int_{\Omega} \big(\mathbf{w}\cdot\nabla\phi_i - \pi_h(\mathbf{w}\cdot\nabla\phi_i)\big) \\ &\qquad \times \big(\mathbf{w}\cdot\nabla\phi_j - \pi_h(\mathbf{w}\cdot\nabla\phi_j)\big)\ \mathrm{d}\Omega, \\ \mathbf{f} = \{f_i\}_{i=1,\ldots,n}, \quad f_i &= \int_{\Omega} f\phi_i\ \mathrm{d}\Omega, \end{aligned} \tag{3.2}$$

with δ again a stabilization parameter and π_h an orthogonal projection operator. Furthermore, as we will demonstrate in Sections 3.2 and 3.3, when this stabilization is applied in the optimal control setting, the discretize-then-optimize and optimize-then-discretize systems are consistent and self-adjoint; that is the discretization and optimization steps commute.

There are a number of considerations which need to be made when applying this method in the control setting with a uniform grid and bilinear basis functions, as we will do in Section 5:

- **Stabilization parameter** δ – We take δ to be the following, as in [2]:

$$\delta = \begin{cases} 0 & \text{if Pe} < 1, \\ \frac{h}{\|\mathbf{w}\|_2} & \text{if Pe} \geq 1, \end{cases}$$

  where the mesh Péclet number Pe is defined on each element as

$$\text{Pe} = \frac{h\,\|\mathbf{w}\|_2}{\epsilon}.$$

  Clearly this means that the stabilization depends on the mesh size, and if the step-size h is less than $\frac{\epsilon}{\|\mathbf{w}\|_2}$, then no stabilization procedure will be applied.
- **Orthogonal projection operator** π_h – We require an L_2-orthogonal projection operator defined on patches of the domain, that satisfies L_2-norm properties specified on p.4 of [2]. We will proceed by working with **Q1** elements with equally-spaced nodes, and divide the domain into patches consisting of 2 elements in each dimension. From this, we will take $\pi_h(v)$ (where v has support solely on that patch) to be equal to the integral of v over the patch divided by the area of the patch (in 2D this will be $4h^2$). This definition will satisfy the required properties in our formulation.
- **Error of LPS method** – In [2], it is shown that the LPS stabilization gives $\mathcal{O}(h^{3/2})$ convergence for problems of the form (3.1) for bilinear finite elements. This further motivates the use of the LPS stabilization method for the remainder of this manuscript.

3.2. Derivation of Matrix System: Discretize-then-Optimize. We now seek to show that, when using the LPS method described in Section 3.1, the matrix systems obtained when using the discretize-then-optimize and optimize-then-discretize approaches are the same. The derivation of the matrix system when using the discretize-then-optimize approach is straightforward. We first note that the discretized version of the PDE constraint is given by

$$\bar{K}\mathbf{y} - M\mathbf{u} = \mathbf{d},$$

where $\mathbf{d}$ is given below.

We also note that we may write the functional that we are trying to minimize, $\frac{1}{2}\|y-\hat{y}\|^2_{L_2(\Omega)} + \frac{\beta}{2}\|u\|^2_{L_2(\Omega)}$, as

$$\frac{1}{2}\|y-\hat{y}\|^2_{L_2(\Omega)} + \frac{\beta}{2}\|u\|^2_{L_2(\Omega)} = \frac{1}{2}\mathbf{y}^T M\mathbf{y} - \mathbf{b}^T\mathbf{y} + C + \frac{\beta}{2}\mathbf{u}^T M\mathbf{u},$$

where C is a constant independent of $\mathbf{y}$, M denotes the *mass matrix* defined by

$$M = \{m_{ij}\}_{i,j=1,\ldots,n}, \quad m_{ij} = \int_\Omega \phi_i\phi_j \ \mathrm{d}\Omega,$$

and $\mathbf{b}$ is given by

$$\mathbf{b} = \{b_i\}_{i=1,\ldots,n}, \quad b_i = \int_\Omega \hat{y}\phi_i \ \mathrm{d}\Omega.$$

We therefore deduce that the Lagrangian we wish to find the stationary point of is given by

$$\mathcal{L}(\mathbf{y},\mathbf{u},\mathbf{p}) = \frac{1}{2}\mathbf{y}^T M\mathbf{y} - \mathbf{y}^T M\hat{\mathbf{y}} + C + \frac{\beta}{2}\mathbf{u}^T M\mathbf{u} + \mathbf{p}^T(\bar{K}\mathbf{y} - M\mathbf{u} - \mathbf{d}). \quad (3.3)$$

Differentiating (3.3) with respect to $\mathbf{y}$, $\mathbf{u}$ and $\mathbf{p}$ yields the following 3 systems of equations:

$$\begin{aligned} M\mathbf{y} + \bar{K}^T\mathbf{p} &= \mathbf{b}, \\ \beta M\mathbf{u} - M\mathbf{p} &= \mathbf{0}, \\ \bar{K}\mathbf{y} - M\mathbf{u} &= \mathbf{d}, \end{aligned}$$

where

$$\mathbf{d} = \{d_i\}_{i=1,\ldots,n}, \quad d_i = \int_\Omega f\phi_i \ \mathrm{d}\Omega - \sum_{j=n+1}^{n+n_\partial} Y_j \int_\Omega \nabla\phi_i \cdot \nabla\phi_j \ \mathrm{d}\Omega.$$

We therefore obtain the 3×3 block matrix system of

$$\begin{bmatrix} M & 0 & \bar{K}^T \\ 0 & \beta M & -M \\ \bar{K} & -M & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \\ \mathbf{d} \end{bmatrix}, \quad (3.4)$$

which is of the saddle point form discussed in Section 2.2.

3.3. Derivation of Matrix System: Optimize-then-Discretize. To derive the optimize-then-discretize formulation, we need to consider the Lagrangian of the form

$$\begin{aligned} \tilde{\mathcal{L}}(y,u,p,\tilde{p}) = {} & \frac{1}{2}\left\|y-\hat{y}\right\|^2_{L_2(\Omega)} + \frac{\beta}{2}\left\|u\right\|^2_{L_2(\Omega)} \\ & + \int_\Omega (-\epsilon\nabla^2 y + \mathbf{w}\cdot\nabla y - u)p \ \mathrm{d}\Omega + \int_{\partial\Omega}(y-f)\tilde{p} \ \mathrm{d}s. \end{aligned}$$

Note that the second Lagrange multiplier $\tilde{p}$ is included in this case as we are not guaranteed to satisfy the boundary conditions as with the discretize-then-optimize approach.

As in [15] for example, we differentiate $\tilde{\mathcal{L}}$ with respect to the state y, the control u and the Lagrange multipliers p and $\tilde{p}$, and study the resulting equations. Calculating the Fréchet derivative with respect to y, and applying the Divergence Theorem and the Fundamental Lemma of Calculus of Variations, as in [15], yields

$$\begin{aligned} -\epsilon\nabla^2 p - (\mathbf{w}\cdot\nabla)p - (\nabla\cdot\mathbf{w})p &= \hat{y} - y, \quad \text{in } \Omega, \\ p &= 0, \qquad \text{on } \partial\Omega, \end{aligned}$$

from which we use the assumption $\nabla\cdot\mathbf{w} = 0$ to give the *adjoint equation*

$$\begin{aligned} -\epsilon\nabla^2 p - \mathbf{w}\cdot\nabla p &= \hat{y} - y, \quad \text{in } \Omega, \\ p &= 0, \qquad \text{on } \partial\Omega. \end{aligned}$$

Further, differentiating with respect to u generates the *gradient equation*

$$\beta u - p = 0,$$

and differentiating with respect to the Lagrange multipliers p and $\tilde{p}$ yields the *state equation*

$$\begin{aligned} -\epsilon\nabla^2 y + \mathbf{w}\cdot\nabla y = u, &\quad \text{in } \Omega, \\ y = f, &\quad \text{on } \partial\Omega. \end{aligned}$$

Discretizing the adjoint, gradient and state equations using the stabilization (3.2) yields the matrix system

$$\begin{bmatrix} M & 0 & \bar{K}^T \\ 0 & \beta M & -M \\ \bar{K} & -M & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \\ \mathbf{d} \end{bmatrix},$$

which is the same saddle point system as that derived using the discretize-then-optimize approach. We therefore consider the solution of this system for the remainder of this manuscript.

4. Preconditioning the Matrix System. In this Section, we consider how one might precondition the matrix system (3.4) for solving the convection-diffusion control problem with local projection stabilization. We will use the saddle point theory of Section 2.2 in this Section.

We first note that we may write (3.4) as a sparse saddle point system of the form (2.4), with $A = \begin{bmatrix} M & 0 \\ 0 & \beta M \end{bmatrix}$, $B = \begin{bmatrix} \bar{K} & -M \end{bmatrix}$ and $C = 0 \in \mathbb{R}^{n\times n}$. By the theory of Section 2.2, we see that we may obtain an effective solver if we have a good approximation to the matrix $\begin{bmatrix} M & 0 \\ 0 & \beta M \end{bmatrix}$, as well as the Schur complement of the matrix system, which is given by

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

The sparsity structure of the saddle point system is illustrated in Figure 4.1.

We therefore start by considering an accurate approximation of these two matrices. As discussed in [21], the Chebyshev semi-iterative method is effective for approximating mass matrices, so in our preconditioners we approximate A by $\hat{A}$, where

$$\hat{A} = \begin{bmatrix} \hat{M} & 0 \\ 0 & \beta\hat{M} \end{bmatrix},$$

and $\hat{M}$ denotes 20 steps of Chebyshev semi-iteration applied to M.

To find an accurate approximation of the Schur complement, we apply the result of Theorem 1, using the fact that the symmetric part of $\bar{K}$, $H := \frac{1}{2}(\bar{K} + \bar{K}^T)$, is positive semi-definite in our formulation. This is due to the positive-definiteness of the symmetric matrix $\epsilon K + T$ and the skew-symmetry of the matrix N in our formulation (see Chapter 3 of [7] for more details). We note that Theorem 1 is an extension of the result proved in [13], which applied to symmetric operators rather than the non-symmetric operator $\bar{K}$ we are considering in this manuscript.

THEOREM 1. *Suppose that the symmetric part of $\bar{K}$, $H := \frac{1}{2}(\bar{K}+\bar{K}^T)$, is positive*

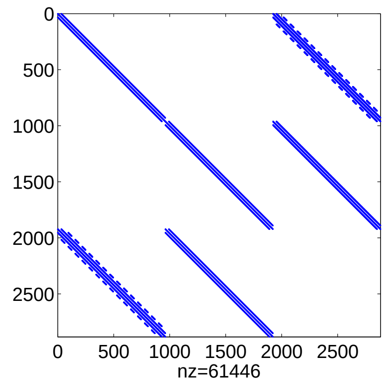


FIG. 4.1. *Sparsity pattern of the system* (3.4) *for Problem 1 as stated in Section 5 for* $h = 2^{-5}$.

semi-definite. Then, if we approximate the Schur complement S *by*

$$\begin{aligned}\hat{S} &= \left(\bar{K} + \frac{1}{\sqrt{\beta}}M\right) M^{-1} \left(\bar{K}^T + \frac{1}{\sqrt{\beta}}M\right) \\ &= \left(\bar{K} + \frac{1}{\sqrt{\beta}}M\right) M^{-1} \left(\bar{K} + \frac{1}{\sqrt{\beta}}M\right)^T,\end{aligned}$$

we can bound the eigenvalues of $\hat{S}^{-1}S$ *as follows:*

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

Proof. We have that

$$\begin{aligned} & \hat{S}^{-1}S\mathbf{x} = \mu\mathbf{x} \\ \Leftrightarrow \quad & \left(\bar{K}M^{-1}\bar{K}^T + \frac{1}{\beta}M\right)\mathbf{x} = \mu\left[\bar{K}M^{-1}\bar{K}^T + \frac{1}{\beta}M + \frac{1}{\sqrt{\beta}}(\bar{K}+\bar{K}^T)\right]\mathbf{x} \\ \Leftrightarrow \quad & \left(\beta\bar{K}M^{-1}\bar{K}^T + M\right)\mathbf{x} = \mu\left[\beta\bar{K}M^{-1}\bar{K}^T + M + \sqrt{\beta}(\bar{K}+\bar{K}^T)\right]\mathbf{x}.\end{aligned}$$

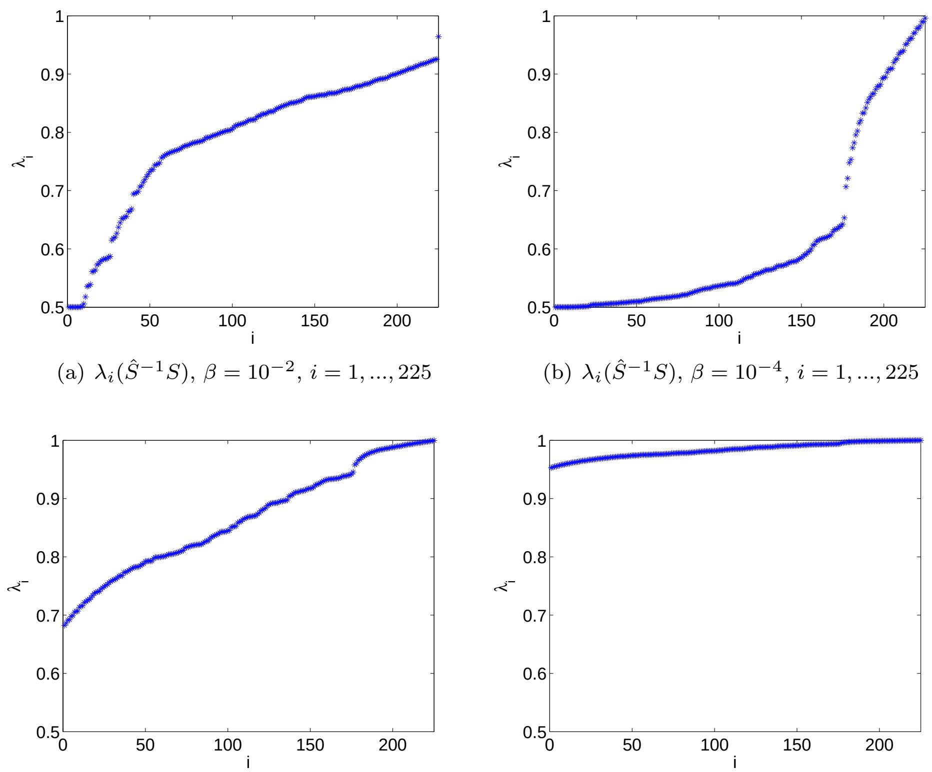
It is therefore sufficient to show that the Rayleigh quotient $R := \frac{\mathbf{v}^T S\mathbf{v}}{\mathbf{v}^T\hat{S}\mathbf{v}}$ satisfies

$$R = \frac{\mathbf{v}^T\left[\beta\bar{K}M^{-1}\bar{K}^T + M\right]\mathbf{v}}{\mathbf{v}^T\left[\beta\bar{K}M^{-1}\bar{K}^T + M + \sqrt{\beta}(\bar{K}+\bar{K}^T)\right]\mathbf{v}} \in [\frac{1}{2}, 1].$$

Now, we may write that $\beta\bar{K}M^{-1}\bar{K}^T = (\sqrt{\beta}\bar{K}M^{-1/2})(\sqrt{\beta}\bar{K}M^{-1/2})^T$ and $M = (M^{1/2})(M^{1/2})^T$. Therefore, with $\mathbf{a} = (\sqrt{\beta}\bar{K}M^{-1/2})^T\mathbf{v}$, $\mathbf{b} = (M^{1/2})^T\mathbf{v}$, we have that

$$R = \frac{\mathbf{a}^T\mathbf{a} + \mathbf{b}^T\mathbf{b}}{(\mathbf{a}+\mathbf{b})^T(\mathbf{a}+\mathbf{b})} = \frac{\mathbf{a}^T\mathbf{a} + \mathbf{b}^T\mathbf{b}}{\mathbf{a}^T\mathbf{a} + \mathbf{b}^T\mathbf{b} + \mathbf{a}^T\mathbf{b} + \mathbf{b}^T\mathbf{a}}.$$

We can see that $\mathbf{a}^T\mathbf{b} + \mathbf{b}^T\mathbf{a} = \sqrt{\beta}\mathbf{v}^T(\bar{K}+\bar{K}^T)\mathbf{v} = 2\sqrt{\beta}\mathbf{v}^T H\mathbf{v} \geq 0$ by the positive semi-definiteness of H, and hence that $R \leq \frac{\mathbf{a}^T\mathbf{a}+\mathbf{b}^T\mathbf{b}}{\mathbf{a}^T\mathbf{a}+\mathbf{b}^T\mathbf{b}} = 1$, as $\mathbf{b} \neq \mathbf{0}$ for any $\mathbf{v}$. To



(a) $\lambda_i(\hat{S}^{-1}S)$, $\beta = 10^{-2}$, $i = 1, ..., 225$

(b) $\lambda_i(\hat{S}^{-1}S)$, $\beta = 10^{-4}$, $i = 1, ..., 225$

(c) $\lambda_i(\hat{S}^{-1}S)$, $\beta = 10^{-6}$, $i = 1, ..., 225$

(d) $\lambda_i(\hat{S}^{-1}S)$, $\beta = 10^{-8}$, $i = 1, ..., 225$

FIG. 4.2. *Spectra of $\hat{S}^{-1}S$ for $\beta = 10^{-2}$, $\beta = 10^{-4}$, $\beta = 10^{-6}$ and $\beta = 10^{-8}$, for an evenly-spaced grid on $\Omega = [0,1]^2$ with $h = 2^{-4}$, $\epsilon = \frac{1}{100}$ and $\mathbf{w} = \left[\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right]^T$.*

show that $R \geq \frac{1}{2}$, we proceed as follows, noting again that $\mathbf{b} \neq \mathbf{0}$:

$$\begin{aligned}
R \geq \frac{1}{2} &\Leftrightarrow \frac{\mathbf{a}^T\mathbf{a} + \mathbf{b}^T\mathbf{b}}{(\mathbf{a}+\mathbf{b})^T(\mathbf{a}+\mathbf{b})} \geq \frac{1}{2} \\
&\Leftrightarrow \mathbf{a}^T\mathbf{a} + \mathbf{b}^T\mathbf{b} \geq \frac{1}{2}\left[\mathbf{a}^T\mathbf{a} + \mathbf{b}^T\mathbf{b} + \mathbf{a}^T\mathbf{b} + \mathbf{b}^T\mathbf{a}\right] \\
&\Leftrightarrow \frac{1}{2}\left[\mathbf{a}^T\mathbf{a} + \mathbf{b}^T\mathbf{b} - \mathbf{a}^T\mathbf{b} - \mathbf{b}^T\mathbf{a}\right] \geq 0 \\
&\Leftrightarrow (\mathbf{a}-\mathbf{b})^T(\mathbf{a}-\mathbf{b}) \geq 0.
\end{aligned}$$

As $(\mathbf{a}-\mathbf{b})^T(\mathbf{a}-\mathbf{b}) \geq 0$ is clearly satisfied by basic vector properties, the result is proved. □

Demonstrations of the eigenvalue distribution of $\hat{S}^{-1}S$ for a variety of values of β in a particular practical case are shown in Figure 4.2.

Therefore, using Theorem 1, we may obtain an effective Schur complement approximation if we can find a good way of approximating the matrices $\bar{K} + \frac{1}{\sqrt{\beta}}M$ and $\left(\bar{K} + \frac{1}{\sqrt{\beta}}M\right)^T$. The method we use for approximating these matrices is the geometric multigrid process described for the forward problem in Section 2.1, with the coarse-grid matrices formed explicitly rather than by the use of prolongation and restriction operators, and with block Gauss-Seidel smoothing.

So, as we now have effective approximations of the matrices $\hat{A}$ and $\hat{S}$, we can propose two effective preconditioners of the form

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

as described in Section 2.2.

Now, unlike the forward problem, the convection-diffusion control problem is symmetric with our (symmetric) stabilization, so as $\hat{\mathcal{P}}_1$ is symmetric positive-definite, one possible method for solving the matrix system (3.4) would be to apply a MINRES method with preconditioner $\hat{\mathcal{P}}_1$. Another possible method would be to apply the Bramble-Pasciak Conjugate Gradient method as described in Section 2.2, with preconditioner $\hat{\mathcal{P}}_2$ and inner product given by

$$\mathcal{H} = \begin{bmatrix} M - \gamma\hat{M} & 0 & 0 \\ 0 & \beta(M - \gamma\hat{M}) & 0 \\ 0 & 0 & \hat{S} \end{bmatrix},$$

where γ is a constant which can be chosen a priori to ensure that $M - \gamma\hat{M}$ is positive definite; results for a 2D **Q1** mass matrix which may be applied for the test problem of Section 5 are provided in [17].

We therefore conclude that we have two effective methods for solving the convection-diffusion control problem:

- A MINRES approach with preconditioner given by
$$\hat{\mathcal{P}}_1 = \begin{bmatrix} \hat{M} & 0 & 0 \\ 0 & \beta\hat{M} & 0 \\ 0 & 0 & \hat{S} \end{bmatrix}, \tag{4.1}$$
where $\hat{M}$ denotes 20 steps of Chebyshev semi-iteration to approximate the mass matrix M, and $\hat{S}$ denotes the approximation to the Schur complement discussed above.
- A Bramble-Pasciak CG solver with preconditioner given by
$$\hat{\mathcal{P}}_2 = \begin{bmatrix} \gamma\hat{M} & 0 & 0 \\ 0 & \beta\gamma\hat{M} & 0 \\ \bar{K} & -M & -\hat{S} \end{bmatrix}, \tag{4.2}$$
where $\hat{M}$ and $\hat{S}$ are defined as above. The inner product on which the CG algorithm should be used, as well as how to choose the constant γ, are described above.

At this point, we make two notes about our preconditioning strategy and its applicability:

1. The matrix system (3.4) for the distributed convection-diffusion control problem could potentially be reduced to the following system of equations by elimination of the discretized gradient equation:
$$\begin{bmatrix} M & \bar{K}^T \\ \bar{K} & -\frac{1}{\beta}M \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{d} \end{bmatrix}, \quad \mathbf{p} = \beta\mathbf{u}.$$

We note that our preconditioning strategy is equally valid for this problem, as we still obtain a saddle point system of the structure discussed in Section 2.2, so we will still need to implement a Chebyshev semi-iteration process to approximate M, and enact the approximation of the Schur complement S, which remains the same as for the system (3.4).

2. We believe that other similar methods could be devised to solve the convection-diffusion control problem based on the framework discussed in this Section. For instance, we see no reason why a preconditioner of the form

$$\hat{\mathcal{P}}_3 = \begin{bmatrix} \hat{A} & B^T \\ B & B\hat{A}^{-1}B^T - \hat{S} \end{bmatrix} = \begin{bmatrix} I & 0 \\ B\hat{A}^{-1} & I \end{bmatrix} \begin{bmatrix} \hat{A} & B^T \\ 0 & -\hat{S} \end{bmatrix},$$

which was discussed in the context of the Poisson control problem in [13, 18], could not be applied for this problem using our approximations $\hat{A}$ and $\hat{S}$.

5. Numerical Results. In this Section, we provide numerical results to demonstrate the effectiveness of our method.

The two problems that we consider are stated below, with plots of solutions to these problems shown in Figures 5.1 and 5.2 respectively.

- **Problem 1:** We wish to solve the following distributed convection-diffusion control problem on $\Omega = [0,1]^2$:

$$\begin{aligned} \min_{y,u} \quad & \frac{1}{2}\|y\|^2_{L_2(\Omega)} + \frac{\beta}{2}\|u\|^2_{L_2(\Omega)} \\ \text{s.t.} \quad & -\epsilon\nabla^2 y + \mathbf{w}\cdot\nabla y = u, \quad \text{in } \Omega, \\ & y = \begin{cases} 1 & \text{on } \partial\Omega_1 := ([0,1]\times\{1\}) \cup (\{0\}\times[\frac{1}{2},1]), \\ 0 & \text{on } \partial\Omega_2 := \partial\Omega\backslash\partial\Omega_1, \end{cases} \end{aligned}$$

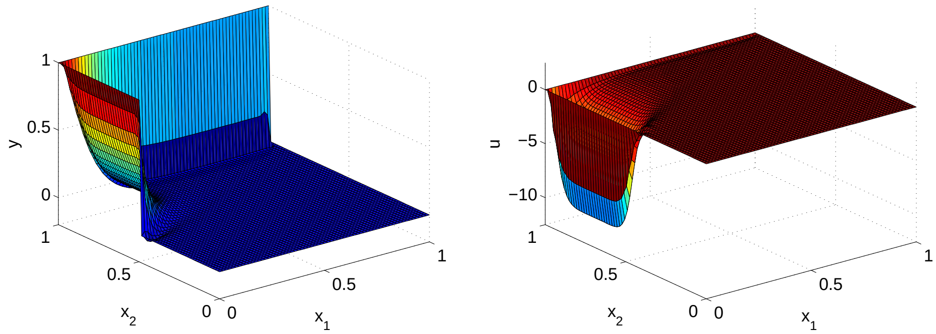
  where $\mathbf{w} = \left[\cos\frac{\pi}{6}, \sin\frac{\pi}{6}\right]^T$. This is an optimal control problem involving a constant wind $\mathbf{w}$; forward problems of this form have previously been considered in literature such as [7, 15].
- **Problem 2:** We wish to solve the following distributed convection-diffusion control problem on $\Omega = [-1,1]^2$:

$$\begin{aligned} \min_{y,u} \quad & \frac{1}{2}\|y\|^2_{L_2(\Omega)} + \frac{\beta}{2}\|u\|^2_{L_2(\Omega)} \\ \text{s.t.} \quad & -\epsilon\nabla^2 y + \mathbf{w}\cdot\nabla y = u, \quad \text{in } \Omega, \\ & y = \begin{cases} 1 & \text{on } \partial\Omega_1 := \{1\}\times[-1,1], \\ 0 & \text{on } \partial\Omega_2 := \partial\Omega\backslash\partial\Omega_1, \end{cases} \end{aligned}$$

  where $\mathbf{w} = \left[2y(1-x^2), -2x(1-y^2)\right]^T$. This is an optimal control formulation of the *double-glazing problem* discussed on p.119. of [7]: a model of the temperature in a cavity with recirculating wind $\mathbf{w}$. Note that there are discontinuities at the points $(x,y) = (-1,1)$ and $(x,y) = (1,1)$ due to the boundary conditions specified.

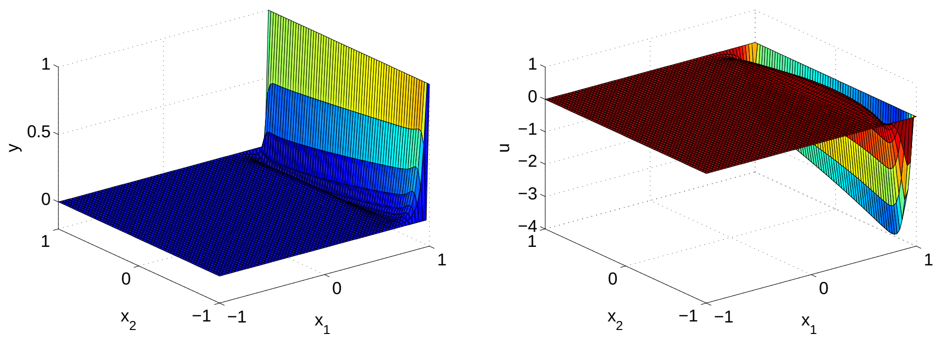
In our numerical tests, we discretize the state y, control u and adjoint p using **Q1** finite element basis functions. We use the local projection stabilization as described in Section 3.1.

In Table 5.1, we present the number of MINRES iterations with preconditioner $\hat{\mathcal{P}}_1$ required to solve Problem 1 with $\epsilon = \frac{1}{200}$ to a tolerance of 10^{-6}, and in Table 5.2 we



(a) State y (b) Control u

FIG. 5.1. *Solutions of state and control for Problem 1 using* ***Q1*** *basis functions with* $\epsilon = \frac{1}{200}$ *and* $\beta = 10^{-2}$.



(a) State y (b) Control u

FIG. 5.2. *Solutions of state and control for Problem 2 using* ***Q1*** *basis functions with* $\epsilon = \frac{1}{50}$ *and* $\beta = 10^{-2}$.

show how many Bramble-Pasciak CG iterations are required to solve the same problem to the same tolerance with preconditioner $\hat{\mathcal{P}}_2$ and with $\gamma = 0.95$. We observe that both our solvers generate convergence in a small number of iterations. The convergence rate actually improves as β decreases, probably because our Schur complement becomes better for smaller β, as illustrated by Figure 4.2. Although we took the wind $\mathbf{w} = \left[\cos\frac{\pi}{6}, \sin\frac{\pi}{6}\right]^T$ and the value $\epsilon = \frac{1}{200}$, we found that the results for $\beta = 10^{-2} - 10^{-8}$ were similar for any constant wind with vector 2-norm equal to 1, and $\epsilon \gtrsim 10^{-6}$. We note that altering the boundary conditions or target function $\hat{y}$ would not change the matrix system involved (as $\mathbf{w}$ is a constant vector), so our solvers seem to be very robust for constant winds and values β which are of computational interest.

In Table 5.3, we present the number of preconditioned MINRES iterations required to solve Problem 2, a harder problem, to the same tolerance and with $\epsilon = \frac{1}{50}$, with the number of preconditioned Bramble-Pasciak CG iterations required to solve this problem shown in Table 5.4. Once more, for this problem and a wide range of values of β, our solvers are effective, with convergence achieved in a very small number of iterations. However, for this more difficult problem, our solvers do not perform as well for larger β as ϵ is decreased. The reason for this is that the effects of the non-constant

		β							
		10^{-2}		10^{-4}		10^{-6}		10^{-8}	
h	Size	Iter.	Time	Iter.	Time	Iter.	Time	Iter.	Time
2^{-2}	27	11	0.0437	11	0.0448	5	0.0362	3	0.0339
2^{-3}	147	15	0.0903	13	0.0874	7	0.0764	5	0.0713
2^{-4}	675	15	0.1446	13	0.1390	9	0.1286	5	0.1152
2^{-5}	2883	15	0.2881	13	0.2704	9	0.2334	5	0.1957
2^{-6}	11907	17	1.2661	15	1.1594	13	1.0253	7	0.6744

Table 5.1

Number of minres *iterations with block diagonal preconditioner* (4.1) *needed to solve Problem 1 with $\epsilon = \frac{1}{200}$, and computation times needed to do so (in seconds), using* **Q1** *basis functions to approximate the state, control and adjoint, for a range of values of h (and hence problem size) and β.*

		β							
		10^{-2}		10^{-4}		10^{-6}		10^{-8}	
h	Size	Iter.	Time	Iter.	Time	Iter.	Time	Iter.	Time
2^{-2}	27	11	0.0513	10	0.0509	9	0.0573	9	0.0471
2^{-3}	147	12	0.0925	13	0.0977	9	0.0873	9	0.0876
2^{-4}	675	13	0.1540	14	0.1582	11	0.1459	10	0.1404
2^{-5}	2883	14	0.3167	14	0.3273	14	0.3156	10	0.2679
2^{-6}	11907	15	1.3456	15	1.3394	16	1.4196	11	1.0663

Table 5.2

Number of Bramble-Pasciak cg *iterations with block triangular preconditioner* (4.2) *needed to solve Problem 1 with $\epsilon = \frac{1}{200}$, and computation times taken to do so (in seconds), using* **Q1** *basis functions to approximate the state, control and adjoint, for a range of values of h (and hence problem size) and β.*

wind $\mathbf{w}$ become more important when the parameters are adjusted in this way, and so the performance of our multigrid solver degrades in these more difficult cases.

However, in general, the results in this Section have demonstrated that the solvers we have proposed are potent ones for a number of convection-diffusion control problems, a class of problems which, as for the convection-diffusion equation itself, is fraught with numerical difficulties. The number of iterations required to solve these problems is small, and the convergence of the solvers improve rather than degrade as β is decreased. As is observable in the computation times shown in Tables 5.1–5.4, the convergence is close to linear with respect to the size of the matrix system; the only barrier to complete mesh-independent convergence is the nonlinearity of the time taken to effect each block Gauss-Seidel smoothing steps in the multigrid cycle, which is also problematic for the gmres solution to the forward problem. The effectiveness of our solvers would therefore be improved further if there were devised a more robust algebraic or geometric multigrid solver for the appropriate matrices that guarantee h-independent convergence – this would also advance the solvers for the convection-diffusion equation itself.

		β							
		10^{-2}		10^{-4}		10^{-6}		10^{-8}	
h	Size	Iter.	Time	Iter.	Time	Iter.	Time	Iter.	Time
2^{-1}	27	9	0.0487	5	0.0436	3	0.0410	3	0.0415
2^{-2}	147	13	0.1025	7	0.0919	5	0.0884	3	0.0849
2^{-3}	675	15	0.1681	9	0.1510	5	0.1398	3	0.1341
2^{-4}	2883	15	0.3119	13	0.2942	7	0.2444	5	0.2281
2^{-5}	11907	15	1.1631	15	1.1702	9	0.8498	5	0.6421

Table 5.3

Number of MINRES *iterations with block diagonal preconditioner* (4.1) *needed to solve Problem 2 with $\epsilon = \frac{1}{50}$, and computation times needed to do so (in seconds), using **Q1** basis functions to approximate the state, control and adjoint, for a range of values of h (and hence problem size) and β.*

		β							
		10^{-2}		10^{-4}		10^{-6}		10^{-8}	
h	Size	Iter.	Time	Iter.	Time	Iter.	Time	Iter.	Time
2^{-1}	27	8	0.0561	7	0.0505	7	0.0507	7	0.0501
2^{-2}	147	11	0.1074	8	0.0997	8	0.0986	8	0.0998
2^{-3}	675	13	0.1774	10	0.1651	9	0.1609	9	0.1619
2^{-4}	2883	13	0.3276	13	0.3271	10	0.2946	9	0.2837
2^{-5}	11907	13	1.1988	15	1.3204	13	1.1983	10	1.0050

Table 5.4

Number of Bramble-Pasciak CG *iterations with block triangular preconditioner* (4.2) *needed to solve Problem 2 with $\epsilon = \frac{1}{50}$, and computation times taken to do so (in seconds), using **Q1** basis functions to approximate the state, control and adjoint, for a range of values of h (and hence problem size) and β.*

6. Conclusions. In this manuscript, we have first given an overview of a GMRES approach for solving the convection-diffusion equation, as well as summarizing some general properties of saddle point systems and some possible solution methods for such systems.

We then introduced the convection-diffusion control problem, and demonstrated that, with a suitable stabilization technique (we used the local projection stabilization in this paper), the discretize-then-optimize and optimize-then-discretize operations commute: that is the same saddle point systems arises from both approaches.

We then proposed two effective solvers for solving the convection-diffusion control problem: one involving a MINRES solver with a block diagonal preconditioner, and one involving a Bramble-Pasciak Conjugate Gradient approach with a block triangular preconditioner. The key components of each of these preconditioners are a good approximation to the mass matrix, a powerful approximation to the Schur complement of the matrix system, and a geometric multigrid process which enables us to enact that approximation.

Numerical results given in Section 5 demonstrate that both our proposed solvers are effective ones for a variety of problems, yielding fast, close to linear convergence as the problem size is increased; this rate of convergence improves as the regularization parameter β is decreased. That the convergence rate cannot worsen as β is increased is proved theoretically. We have observed in results not presented here that our solution method also works well with the discretize-then-optimize formulation of the control problem with SUPG stabilization (which generates a symmetric system). The method also works well with no stabilization at all when such an approach is reasonable; for such diffusion-dominated problems, more standard methods (including multigrid) could likely be used. If new stabilization methods are discovered for this problem, we might predict that our proposed preconditioners will again prove to be potentially useful for solving these problems. The major piece of further work arising from our research is the need to develop a more robust multigrid procedure for approximating the relevant matrices, so that our method becomes valid for a wider range of values of ϵ and divergence-free winds $\mathbf{w}$, though we note that this is research that would also be useful when solving the simpler forward problem.

Acknowledgements. The first author was supported for this work by the Engineering and Physical Sciences Research Council (UK), Grant EP/P505216/1.

REFERENCES

[1] R. Becker, and M. Braack, *A Finite Element Pressure Gradient Stabilization for the Stokes Equations Based on Local Projections*, Calcolo, 38(4), pp.173–199, 2001.
[2] R. Becker, and B. Vexler, *Optimal Control of the Convection-Diffusion Equation Using Stabilized Finite Element Methods*, Numerische Mathematik, 106(3), pp.349–367, 2007.
[3] M. Benzi, G. H. Golub, and J. Liesen, *Numerical Solution of Saddle Point Problems*, Acta Numerica, 14, pp.1–137, 2005.
[4] J. H. Bramble, and J. E. Pasciak, *A Preconditioning Technique for Indefinite Systems Resulting from Mixed Approximations of Elliptic Problems*, Mathematics of Computation, 50(181), pp.1–17, 1988.
[5] M. Braack, and E. Burman, *Local Projection Stabilization for the Oseen Problem and its Interpretation as a Variational Multiscale Method*, SIAM Journal on Numerical Analysis, 43(6), pp.2544–2566, 2006.
[6] S. S. Collis, and M. Heinkenschloss, *Analysis of the Streamline Upwind/Petrov Galerkin Method Applied to the Solution of Optimal Control Problems*, Technical Report CAAM TR02-01, Rice University, 2002.
[7] H. C. Elman, D. J. Silvester, and A. J. Wathen, *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics*, Numerical Mathematics and Scientific Computation Series, Oxford University Press: New York, 2005.
[8] J.-L. Guermond, *Stabilization of Galerkin Approximations of Transport Equations by Subgrid Modeling*, Modélisation Mathématique at Analyse Numérique, 33(6), pp.1293–1316, 1999.
[9] M. Heinkenschloss, and D. Leykekhman, *Local Error Estimates for SUPG Solutions of Advection-Dominated Elliptic Linear-Quadratic Optimal Control Problems*, SIAM Journal on Numerical Analysis, 47(6), pp.4607–4638, 2010.
[10] T. J. R. Hughes, and A. Brooks. *Finite Element Methods for Convection Dominated Flows*, pp.120–131 of "A Multidimensional Upwind Scheme with no Crosswind Diffusion", Vol. 34 of AMD Series, ASME, 1979.
[11] M. F. Murphy, G. H. Golub, and A. J. Wathen, *A Note on Preconditioning for Indefinite Linear Systems*, SIAM Journal on Scientific Computing, 21(6), pp.1969–1972, 2000.
[12] C. C. Paige, and M. A. Saunders, *Solutions of Sparse Indefinite Systems of Linear Equations*, SIAM Journal on Numerical Analysis, 12(4), pp.617–629, 1975.
[13] J. W. Pearson, and A. J. Wathen, *A New Approximation of the Schur Complement in Preconditioners for PDE Constrained Optimization*, to appear in Numerical Linear Algebra with Applications, 2010.
[14] A. Ramage, *A Multigrid Preconditioner for Stabilised Discretisations of Advection-Diffusion Problems*, Journal of Computational and Applied Mathematics, 110 (1999), pp.187–203.
[15] T. Rees, *Preconditioning Iterative Methods for PDE Constrained Optimization*, Doctor of Phi-

losophy Thesis, University of Oxford, 2010.
[16] T. Rees, H. S. Dollar, and A. J. Wathen, *Optimal Solvers for PDE-constrained Optimization*, SIAM Journal on Scientific Computing, 32(1), pp.271–298, 2010.
[17] T. Rees, and M. Stoll, *Block Triangular Preconditioners for PDE-constrained Optimization*, to appear in Numerical Linear Algebra with Applications, Numerical Linear Algebra with Applications, 17, pp. 977–996, 2010.
[18] J. Schöberl, and W. Zulehner, *Symmetric Indefinite Preconditioners for Saddle Point Problems with Applications to PDE-constrained Optimization Problems*, SIAM Journal on Matrix Analysis and Applications, 29(3), pp.752–773, 2007.
[19] M. Stoll, and A. Wathen, *Combination Preconditioning and the Bramble-Pasciak$^+$ Preconditioner*, SIAM Journal on Matrix Analysis and Applications, 30(2), pp.582–608, 2008.
[20] F. Tröltzsch, *Optimal Control of Partial Differential Equations: Theory, Methods and Applications*, Graduate Series in Mathematics, American Mathematical Society, 2010.
[21] A. J. Wathen, and T. Rees, *Chebyshev Semi-iteration in Preconditioning for Problems Including the Mass Matrix*, Electronic Transactions on Numerical Analysis, 34, pp.125–135, 2009.