

PRECONDITIONED ITERATIVE METHODS FOR NAVIER-STOKES CONTROL PROBLEMS

JOHN W. PEARSON*

Abstract. PDE-constrained optimization problems are a class of problems which have attracted much recent attention in scientific computing and applied science. In this paper, we discuss preconditioned iterative methods for a class of Navier-Stokes control problems, one of the main problems of this type in the field of fluid dynamics. Having detailed the Oseen-type iteration we use to solve the problems and derived the structure of the matrix system to be solved at each step, we utilize the theory of saddle point systems to develop efficient preconditioned iterative solution techniques for these problems. We also require theory of solving convection-diffusion control problems, as well as a commutator argument to justify one of the components of the preconditioner.

Key words. PDE-constrained optimization, Navier-Stokes control, Oseen iteration, preconditioning, Schur complement.

AMS subject classifications. 65F08, 65F10, 65F50, 76D05, 76D55, 93C20

1. Introduction. An active field of research in applied mathematics of late has been that of optimal control problems (see [25]). One of the many areas in which these problems have considerable applicability is that of fluid dynamics. In this article, we consider one important such problem within this field, that of the Navier-Stokes control problem. There has been much previous work on this problem, and we refer the interested reader to literature such as [2, 3, 6, 7, 10, 11, 13, 21, 26].

The development of effective solvers for the forward Navier-Stokes equations is a fairly recent development in numerical analysis. In [14], the authors utilized saddle point theory in combination with a commutator argument to approximate the Schur complement of the matrix system corresponding to each step of an outer iteration, to develop a preconditioner for such a problem. We now wish to apply similar strategies for the problem where the Navier-Stokes equations are not the entire problem, but are purely the constraints applied when the minimization of some cost functional is sought.

To solve the Navier-Stokes control problem, we make use of an Oseen-type iteration (as discussed in [21]) to deal with the nonlinear terms in the forward and adjoint equations. We then make use of similar saddle point theory to develop effective preconditioned iterative methods for the linear system generated by each step of the outer iteration. We exploit solvers previously developed in [19], as well as a commutator argument, to derive new block diagonal and block triangular preconditioners for the Navier-Stokes control problem. The author has previously used such strategies to good effect to solve the simpler Stokes control problem in [17] – with some modifications, we are able to extend this to the more general Navier-Stokes control problem.

This paper is structured as follows. In Section 2, we introduce the distributed Navier-Stokes control problem considered, discuss the outer iteration employed, and state the matrix system we are required to solve at each step of this outer iteration. In Section 3, we derive two preconditioners for this problem (one block diagonal and one block triangular), using ideas from saddle point theory, preconditioners for convection-diffusion control problems and commutator arguments. We discuss the

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

major operations required to apply these preconditioners in practice. In Section 4, we present numerical results to highlight the performance of our method for two test problems, and in Section 5, we make some concluding remarks and discuss possible extensions of this work.

2. The Navier-Stokes Control Problem. The problem we will consider in this article is the time-independent (distributed) Navier-Stokes control problem, given by

$$\begin{aligned} \min_{\underline{\mathbf{v}}, \underline{\mathbf{u}}} \quad & \frac{1}{2} \|\underline{\mathbf{v}} - \widehat{\underline{\mathbf{v}}}\|_{L_2(\Omega)}^2 + \frac{\beta}{2} \|\underline{\mathbf{u}}\|_{L_2(\Omega)}^2 \\ \text{s.t.} \quad & -\nu \nabla^2 \underline{\mathbf{v}} + (\underline{\mathbf{v}} \cdot \nabla) \underline{\mathbf{v}} + \nabla p = \underline{\mathbf{u}}, \quad \text{in } \Omega, \\ & -\nabla \cdot \underline{\mathbf{v}} = 0, \quad \text{in } \Omega, \\ & \underline{\mathbf{v}} = \underline{\mathbf{g}}, \quad \text{on } \partial\Omega. \end{aligned}$$

Here, we work on the domain $\Omega \subset \mathbb{R}^d$, $d \in \{2, 3\}$, with boundary $\partial\Omega$. The variables $\underline{\mathbf{v}}$ and p are the *state variables*, denoting the velocity (in d -dimensions) and pressure respectively. The variable $\underline{\mathbf{u}}$ denotes the *control variable*. The parameter β is a *regularization parameter* (or *Tikhonov parameter*), the parameter ν denotes viscosity, and $\widehat{\underline{\mathbf{v}}}$ is some *desired state*. For the purposes of the optimal control problem, we now introduce $\underline{\boldsymbol{\lambda}}$ and μ as the *adjoint variables* to $\underline{\mathbf{v}}$ and p respectively.

In order to linearize this problem, we have a number of possibilities. The approach we follow is an Oseen-type (or Picard-type) iteration, as examined in [21]. This involves choosing an initial guess for states and control, and then solving in turn the problems

$$\begin{aligned} \min_{\underline{\mathbf{v}}, \underline{\mathbf{u}}} \quad & \frac{1}{2} \|\underline{\mathbf{v}} - \widehat{\underline{\mathbf{v}}}\|_{L_2(\Omega)}^2 + \frac{\beta}{2} \|\underline{\mathbf{u}}\|_{L_2(\Omega)}^2 - \int_{\Omega} (\underline{\mathbf{v}} \cdot \nabla \bar{\underline{\mathbf{v}}}) \bar{\underline{\boldsymbol{\lambda}}} \, d\Omega \\ \text{s.t.} \quad & -\nu \nabla^2 \underline{\mathbf{v}} + (\bar{\underline{\mathbf{v}}} \cdot \nabla) \underline{\mathbf{v}} + \nabla p = \underline{\mathbf{u}}, \quad \text{in } \Omega, \\ & -\nabla \cdot \underline{\mathbf{v}} = 0, \quad \text{in } \Omega, \\ & \underline{\mathbf{v}} = \underline{\mathbf{g}}, \quad \text{on } \partial\Omega, \end{aligned}$$

where $\bar{\underline{\mathbf{v}}}$ denotes the most recent iterate of $\underline{\mathbf{v}}$ and $\bar{\underline{\boldsymbol{\lambda}}}$ the most recent iterate of $\underline{\boldsymbol{\lambda}}$, until convergence of the solution is achieved. As shown in [21], $-\int_{\Omega} (\underline{\mathbf{v}} \cdot \nabla \bar{\underline{\mathbf{v}}}) \bar{\underline{\boldsymbol{\lambda}}} \, d\Omega$ is the appropriate correction term required to prove convergence.¹

We will use a **Q2-Q1** finite element (the *Taylor-Hood element*) to solve this problem, that is we use **Q2** elements to discretize $\underline{\mathbf{v}}$ (and $\underline{\boldsymbol{\lambda}}$) and **Q1** elements for p (and μ), as well as **Q2** elements for the control $\underline{\mathbf{u}}$.

We next apply a discretize-then-optimize approach² to obtain the following matrix

¹As stated in [21], this choice of correction term makes the cost functional strictly convex on a linear manifold, giving a unique solution for any given $\bar{\underline{\mathbf{v}}}$ and $\bar{\underline{\boldsymbol{\lambda}}}$.

²The discretize-then-optimize approach, which corresponds to forming a discrete Lagrangian and then deriving optimality conditions in terms of this, results in a symmetric matrix system at each Oseen iteration. The alternative optimize-then-discretize approach involves deriving continuous optimality conditions and then discretizing them. This strategy will not necessarily result in symmetric matrix systems for this problem, and the systems obtained will depend on the precise discretization strategy used.

system at each step of the outer iteration:

$$\begin{bmatrix} M & 0 & F^T & B^T \\ 0 & 0 & B & 0 \\ F & B^T & -\frac{1}{\beta}M & 0 \\ B & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\lambda} \\ \boldsymbol{\mu} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{z}} + \mathbf{c} \\ \mathbf{0} \\ \mathbf{d} \\ \mathbf{f} \end{bmatrix} + \begin{bmatrix} \bar{\mathbf{w}} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \quad (2.1)$$

where $F = \nu K + N$, with M and K denoting $d \times d$ block matrices consisting of standard finite element mass and stiffness matrices on their block diagonals. Furthermore, N is a matrix denoting convective terms of the type $\int_{\Omega} (\bar{\mathbf{v}} \cdot \nabla \phi_j) \phi_i \, d\Omega$, and $\bar{\mathbf{w}}$ contains terms of the form $\int_{\Omega} (\phi_i \cdot \nabla \bar{\mathbf{v}}) \bar{\boldsymbol{\lambda}} \, d\Omega$. The vector $\hat{\mathbf{z}}$ contains terms arising from the target state $\hat{\mathbf{v}}$.

In this form, the bottom-left of the matrix system corresponds to the forward problem, and the top-right relates to the adjoint problem. Indeed the top two equations are referred to as the *adjoint equations* for the optimal control problem, and the bottom two the *state equations*. The system (2.1) was reached by eliminating the *gradient equation* $\beta \mathbf{u} - \boldsymbol{\lambda} = \mathbf{0}$ relating the control and an adjoint variable. The terms corresponding to the cost functional occur on the block diagonal. Equation (2.1) states the form of matrix system for which we will consider preconditioned iterative methods in Section 3.

We note that the Oseen-type iteration we have selected to deal with the non-linearity of the optimal control problem is not the only option we have to do this. Another possibility is to apply a Newton-type method (see [7, 10, 11, 26] for instance); however we discover that applying such a method causes the $(1, 1)$ -block of the resulting matrix systems to be dominated by convective terms. This is highly problematic when constructing an iterative solver for such a system using our approach, and we therefore conclude that the Oseen iteration strategy presented is the one best suited to our preconditioners.

3. Preconditioners for the Navier-Stokes Control Problem. We now wish to examine how to construct effective preconditioners for the matrix systems arising from the Navier-Stokes control problem. To do this, we exploit the theory of *saddle point systems*, that is matrix systems of the form

$$\underbrace{\begin{bmatrix} \Phi & \Psi^T \\ \Psi & -\Theta \end{bmatrix}}_{\mathcal{A}} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix},$$

where $\Phi \in \mathbb{R}^{m \times m}$, $\Psi \in \mathbb{R}^{q \times m}$ and $\Theta \in \mathbb{R}^{q \times q}$, with $m \geq q$. We refer the interested reader to [1] for a comprehensive review of numerical methods for such systems. It is well known that some effective (ideal) preconditioners for \mathcal{A} can be represented as [15, 16]

$$\mathcal{P}_1 = \begin{bmatrix} \Phi & 0 \\ 0 & S \end{bmatrix}, \quad \mathcal{P}_2 = \begin{bmatrix} \Phi & 0 \\ \Psi & S \end{bmatrix}, \quad \mathcal{P}_3 = \begin{bmatrix} \Phi & 0 \\ \Psi & -S \end{bmatrix},$$

where $S = \Theta + \Psi \Phi^{-1} \Psi^T$ is the (negative) *Schur complement* of the matrix system. A number of results are known to justify this. In more detail, as long as the preconditioned system is nonsingular

$$\lambda(\mathcal{P}_2^{-1} \mathcal{A}) = \{\pm 1\}, \quad \lambda(\mathcal{P}_3^{-1} \mathcal{A}) = \{1\},$$

for any choice of Θ , and

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

provided $\Theta = 0$. The above results for the case $\Theta = 0$ were shown in [15, 16], and the eigenvalues of $\mathcal{P}_2^{-1}\mathcal{A}$ and $\mathcal{P}_3^{-1}\mathcal{A}$ in the general case $\Theta \neq 0$ were proved in [12]. A suitable Krylov subspace method with preconditioner \mathcal{P}_1 , \mathcal{P}_2 or \mathcal{P}_3 is hence an *optimal solver*, and will converge in 3, 2 and 2 iterations in each case, subject to the appropriate conditions and provided \mathcal{A} is invertible [16].

Furthermore, as shown in [17] for instance, provided Φ and Θ are symmetric positive definite matrices, the eigenvalues of $\mathcal{P}_1^{-1}\mathcal{A}$ can be bounded as follows:

$$\lambda(\mathcal{P}_1^{-1}\mathcal{A}) \in \left[-1, \frac{1}{2}(1 - \sqrt{5})\right] \cup \left[1, \frac{1}{2}(1 + \sqrt{5})\right]. \quad (3.1)$$

In general, Φ and S are not practical preconditioners, so the main step in developing a preconditioner for \mathcal{A} of the form \mathcal{P}_1 , \mathcal{P}_2 and \mathcal{P}_3 is constructing effective approximations $\hat{\Phi}$ and \hat{S} to Φ and S , which can be inverted cheaply and feasibly.

Motivated by this saddle point theory, we may therefore rearrange the matrix system (2.1) obtained at each step of the Oseen iteration for the Navier-Stokes control problem. Specifically, we write it as

$$\underbrace{\begin{bmatrix} M & F^T & B^T & 0 \\ F & -\frac{1}{\beta}M & 0 & B^T \\ B & 0 & 0 & 0 \\ 0 & B & 0 & 0 \end{bmatrix}}_{\mathcal{A}} \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\lambda} \\ \boldsymbol{\mu} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{z}} + \mathbf{c} \\ \mathbf{d} \\ \mathbf{f} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \bar{\mathbf{w}} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \quad (3.2)$$

which is a saddle point system with

$$\Phi = \begin{bmatrix} M & F^T \\ F & -\frac{1}{\beta}M \end{bmatrix}, \quad \Psi = \begin{bmatrix} B & 0 \\ 0 & B \end{bmatrix}, \quad \Theta = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

From now on we denote as \mathcal{A} the particular matrix system corresponding to the Navier-Stokes control problem we consider. The preconditioners we derive in this section for this problem will be based upon applying the saddle point theory explained above. Such a strategy has previously been applied to the simpler Stokes control problem by the author in [17].

We spend the majority of this section motivating a block diagonal preconditioner for the matrix system (3.2). We first note that the (1,1)-block Φ of \mathcal{A} is itself a saddle point system – specifically, it relates to the matrix system corresponding to the (distributed) convection-diffusion control problem discussed in [19]. It is explained in [19] that the matrix

$$\hat{\Phi} = \begin{bmatrix} M & 0 \\ 0 & \left(F + \frac{1}{\sqrt{\beta}}M\right) M^{-1} \left(F + \frac{1}{\sqrt{\beta}}M\right)^T \end{bmatrix} =: \begin{bmatrix} M & 0 \\ 0 & \hat{S}_{CD} \end{bmatrix} \quad (3.3)$$

is an effective preconditioner for the matrix Φ .³ We therefore advocate this as a good choice for the (1,1)-block of the preconditioner for \mathcal{A} . We note that, in [17], the

³It is clear from (3.1) that the “ideal” block diagonal preconditioner should be an effective

strategy employed for solving the Stokes control problem involved re-arranging the matrix system such that the $(1, 1)$ -block corresponded to a Poisson control problem, and then approximating this block by preconditioners for Poisson control derived in [20, 29].

In order to develop a good approximation of the Schur complement

$$S = \begin{bmatrix} B & 0 \\ 0 & B \end{bmatrix} \begin{bmatrix} M & F^T \\ F & -\frac{1}{\beta}M \end{bmatrix}^{-1} \begin{bmatrix} B^T & 0 \\ 0 & B^T \end{bmatrix}$$

of \mathcal{A} , we approximate the matrix $\begin{bmatrix} M & F^T \\ F & -\frac{1}{\beta}M \end{bmatrix}$ in terms of its $(1, 1)$ -block and exact Schur complement, and write

$$\begin{aligned} S &\approx \begin{bmatrix} B & 0 \\ 0 & B \end{bmatrix} \begin{bmatrix} M & 0 \\ 0 & FM^{-1}F^T + \frac{1}{\beta}M \end{bmatrix}^{-1} \begin{bmatrix} B^T & 0 \\ 0 & B^T \end{bmatrix} \\ &= \begin{bmatrix} BM^{-1}B^T & 0 \\ 0 & B \left(FM^{-1}F^T + \frac{1}{\beta}M \right)^{-1} B^T \end{bmatrix}. \end{aligned}$$

Of course, due to the fact that B is rectangular and therefore non-invertible, the matrices $BM^{-1}B^T$ and $B \left(FM^{-1}F^T + \frac{1}{\beta}M \right)^{-1} B^T$ cannot be used directly within the approximation of the Schur complement – we must therefore introduce approximations of these matrices. It is however well established that $BM^{-1}B^T$ may be approximated effectively by K_p (see [9, Chapter 8]), so we utilize this within our block diagonal preconditioner.⁴

We now consider how best to approximate the matrix $B \left(FM^{-1}F^T + \frac{1}{\beta}M \right)^{-1} B^T$.

We do this by applying a *commutator argument* as stated below.⁵

The argument starts by examining the following commutator:

$$\mathcal{E} = (\mathcal{L})\nabla - \nabla(\mathcal{L})_p,$$

where $\mathcal{L} = (-\nu\nabla^2 + \bar{\mathbf{v}} \cdot \nabla) \cdot (-\nu\nabla^2 + \bar{\mathbf{v}} \cdot \nabla)^T + \frac{1}{\beta}I$. The key assumption we use is that \mathcal{E} is small. The operator \mathcal{L} is carefully chosen to give us a matrix that we can use to approximate $\Sigma := B \left(FM^{-1}F^T + \frac{1}{\beta}M \right)^{-1} B^T$.

We next discretize this commutator using finite elements to obtain

$$\mathcal{E}_h = (M^{-1}L)M^{-1}B^T - M^{-1}B^T(M_p^{-1}L_p) \approx 0, \quad (3.4)$$

approximation of Φ . The main step required to show that our choice of approximation $\hat{\Phi}$ is a potent one is to prove, as in [19], that the eigenvalues of the preconditioned Schur complement of Φ are given as follows:

$$\lambda \left(\left[\left(F + \frac{1}{\sqrt{\beta}}M \right) M^{-1} \left(F + \frac{1}{\sqrt{\beta}}M \right)^T \right]^{-1} \left[FM^{-1}F^T + \frac{1}{\beta}M \right] \right) \in \left[\frac{1}{2}, 1 \right],$$

and hence that $\left(F + \frac{1}{\sqrt{\beta}}M \right) M^{-1} \left(F + \frac{1}{\sqrt{\beta}}M \right)^T$ is an effective Schur complement approximation.

⁴We may argue that $BM^{-1}B^T \approx K_p$ as follows: on the continuous level, it is clear that $-\nabla \cdot \nabla = -\nabla^2$. As in the finite element space K_p corresponds to the operator $-\nabla^2$, B relates to the negative of the divergence operator, B^T represents the gradient operator and M corresponds to the identity operator, the approximation of K_p by $BM^{-1}B^T$ is a natural one.

⁵This is a very similar argument to that used by the author for the less general and simpler Stokes control problem in [17].

where $L = FM^{-1}F^T + \frac{1}{\beta}M$. Note that we have carried over to the discrete space our assumption that the commutator is small. Pre-multiplying (3.4) by $BL^{-1}M$ and post-multiplying by $L_p^{-1}M_p$, where $L_p = F_pM_p^{-1}F_p^T + \frac{1}{\beta}M_p$, then gives that $BM^{-1}B^TL_p^{-1}M_p \approx BL^{-1}B^T$. We then use that $BM^{-1}B^T \approx K_p$ and substitute in the expressions for L and L_p to give

$$\begin{aligned}\Sigma &= B \left(FM^{-1}F^T + \frac{1}{\beta}M \right)^{-1} B^T \approx K_p \left(F_pM_p^{-1}F_p^T + \frac{1}{\beta}M_p \right)^{-1} M_p \\ \Rightarrow \Sigma^{-1} &\approx M_p^{-1} \left(F_pM_p^{-1}F_p^T + \frac{1}{\beta}M_p \right) K_p^{-1} = M_p^{-1}F_pM_p^{-1}F_p^T K_p^{-1} + \frac{1}{\beta}K_p^{-1}.\end{aligned}$$

This commutator argument therefore generates a desired approximation to be used in our preconditioner for \mathcal{A} . We highlight that an argument of the above type was used by Cahouet and Chabard in [5] for the forward Stokes equations, and therefore preconditioners of this form are commonly referred to as Cahouet-Chabard preconditioners.

We may use the result of this argument to approximate the Schur complement of \mathcal{A} by

$$\hat{S} := \begin{bmatrix} K_p & 0 \\ 0 & \left(M_p^{-1}F_pM_p^{-1}F_p^T K_p^{-1} + \frac{1}{\beta}K_p^{-1} \right)^{-1} \end{bmatrix} =: \begin{bmatrix} K_p & 0 \\ 0 & \hat{S}_{NS,1}^{-1} \end{bmatrix}.$$

Therefore, putting all the pieces together, we may write a proposed preconditioner \mathcal{P}_D for \mathcal{A} as $\text{blkdiag}(\hat{\Phi}, \hat{S})$, i.e.

$$\mathcal{P}_D = \begin{bmatrix} M & 0 & 0 & 0 \\ 0 & \left(F + \frac{1}{\sqrt{\beta}}M \right) M^{-1} \left(F + \frac{1}{\sqrt{\beta}}M \right)^T & 0 & 0 \\ 0 & 0 & K_p & 0 \\ 0 & 0 & 0 & \hat{S}_{NS,1}^{-1} \end{bmatrix}.$$

We note that, due to the non-symmetry of the matrix $M_p^{-1}F_pM_p^{-1}F_p^T K_p^{-1}$, this is not a symmetric preconditioner, despite the fact that \mathcal{A} is itself symmetric. We are therefore not able to use a symmetric solver with this preconditioner, and would instead need to use a solver such as GMRES [23] with the preconditioner \mathcal{P}_D .

We note that the above commutator argument is a heuristic approach, and due to the non-symmetry of the matrix approximation generated it would be very difficult to analyze in great detail. In Figure 3.1 we provide eigenvalue plots for the matrix $\left(M_p^{-1}F_pM_p^{-1}F_p^T K_p^{-1} + \frac{1}{\beta}K_p^{-1} \right) \left[B \left(FM^{-1}F^T + \frac{1}{\beta}M \right)^{-1} B^T \right]$, for small matrix systems arising from the final Oseen iteration applied to Problem 1 as stated in Section 4. Plots are given for a range of β and Reynolds numbers Re . The matrix we consider for our plots is equal to Σ preconditioned by our approximation of Σ . In these plots, we omit the zero eigenvalue resulting from the vector of ones belonging to the nullspace of B^T , as well as the largest eigenvalue. The reason for this latter omission is that we find there is a single eigenvalue of this matrix of much larger magnitude than the others – identifying a way of isolating and removing this very large eigenvalue would improve our approximation. We find however that the remainder of the eigenvalues are well clustered for a range of parameters, and that an individual eigenvalue is unlikely to greatly delay convergence of a Krylov subspace method.

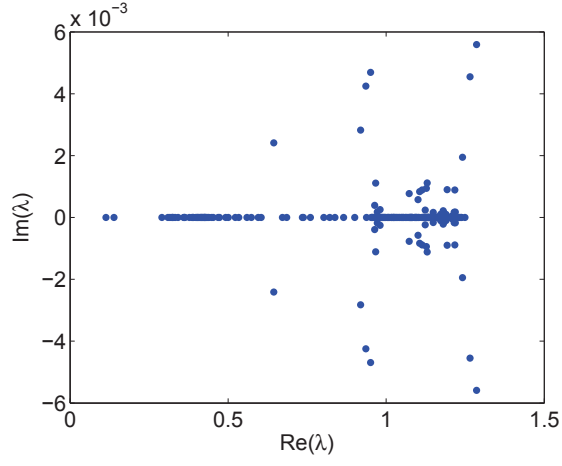
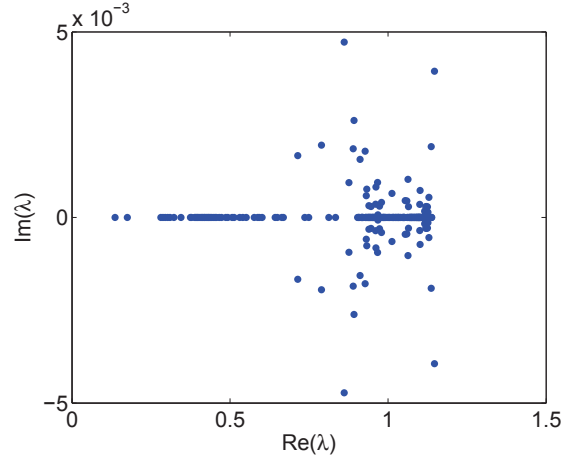
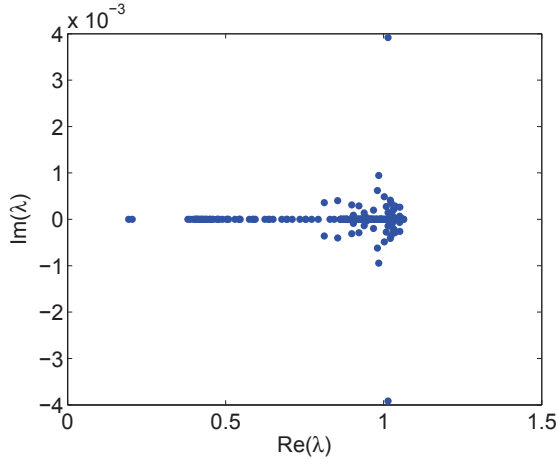
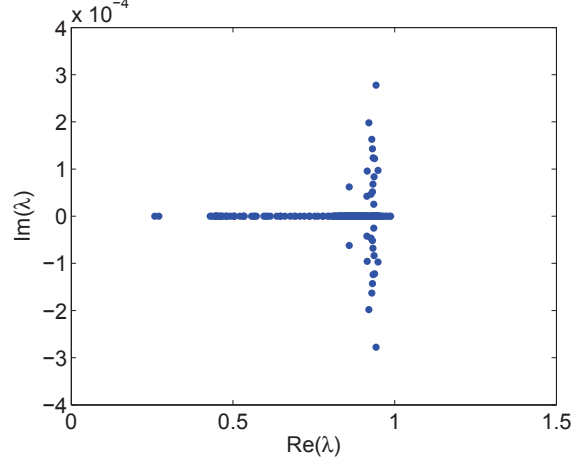
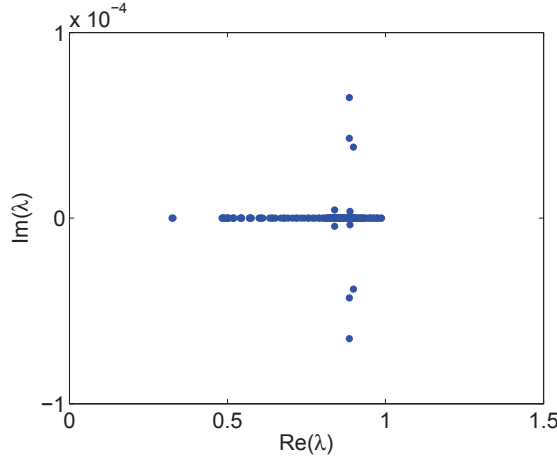
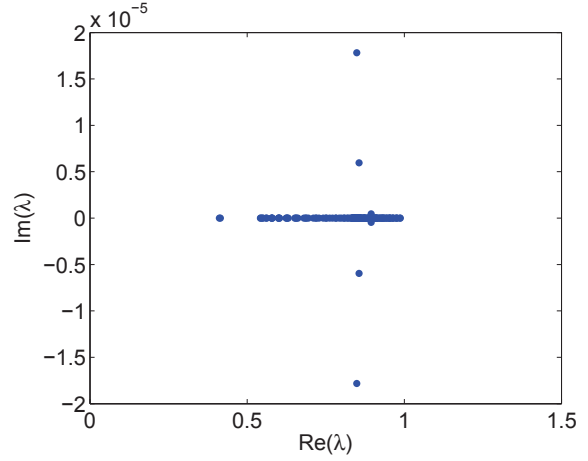
(a) $\beta = 10^{-1}$, $\text{Re} = 100$ (b) $\beta = 10^{-1}$, $\text{Re} = 200$ (c) $\beta = 10^{-2}$, $\text{Re} = 100$ (d) $\beta = 10^{-2}$, $\text{Re} = 200$ (e) $\beta = 10^{-3}$, $\text{Re} = 100$ (f) $\beta = 10^{-3}$, $\text{Re} = 200$

FIG. 3.1. Spectra of $\left(M_p^{-1}F_pM_p^{-1}F_p^TK_p^{-1} + \frac{1}{\beta}K_p^{-1}\right) \left[B \left(FM^{-1}F^T + \frac{1}{\beta}M\right)^{-1}B^T\right]$ for $\beta = 10^{-1}, 10^{-2}, 10^{-3}$ and $\text{Re} = \frac{2}{\nu} = 100$ and 200 , for the final Oseen step in the solution of Problem 1, on an evenly spaced grid on $\Omega = [-1, 1]^2$ with $h = 2^{-4}$.

This fact also means that we could create a block triangular preconditioner for the matrix \mathcal{A} , without imposing further restrictions on the solvers we could use it with. We therefore now derive such a block triangular preconditioner. We start by

approximating Φ with the non-symmetric matrix

$$\begin{bmatrix} M & 0 \\ F & -\left(F + \frac{1}{\sqrt{\beta}}M\right)M^{-1}\left(F + \frac{1}{\sqrt{\beta}}M\right)^T \end{bmatrix},$$

which is motivated by the saddle point theory as discussed previously, and is detailed in [19].

When approximating the Schur complement of \mathcal{A} for this block triangular preconditioner, we may write

$$\begin{aligned} & \begin{bmatrix} B & 0 \\ 0 & B \end{bmatrix} \begin{bmatrix} M & F^T \\ F & -\frac{1}{\beta}M \end{bmatrix}^{-1} \begin{bmatrix} B^T & 0 \\ 0 & B^T \end{bmatrix} \\ & \approx \begin{bmatrix} B & 0 \\ 0 & B \end{bmatrix} \begin{bmatrix} M & 0 \\ F & -\hat{S}_{CD} \end{bmatrix}^{-1} \begin{bmatrix} B^T & 0 \\ 0 & B^T \end{bmatrix} \\ & = \begin{bmatrix} B & 0 \\ 0 & B \end{bmatrix} \begin{bmatrix} M^{-1} & 0 \\ \hat{S}_{CD}^{-1}FM^{-1} & -\hat{S}_{CD}^{-1} \end{bmatrix} \begin{bmatrix} B^T & 0 \\ 0 & B^T \end{bmatrix} \\ & = \begin{bmatrix} BM^{-1}B^T & 0 \\ B\hat{S}_{CD}^{-1}FM^{-1}B^T & -B\hat{S}_{CD}^{-1}B^T \end{bmatrix} \\ & \approx \begin{bmatrix} K_p & 0 \\ B\hat{S}_{CD}^{-1}FM^{-1}B^T & -\hat{S}_{NS,2}^{-1} \end{bmatrix}, \end{aligned}$$

where \hat{S}_{CD} is as defined in (3.3), and

$$\hat{S}_{NS,2} := M_p^{-1}F_pM_p^{-1}F_p^TK_p^{-1} + \frac{1}{\beta}K_p^{-1} + \frac{1}{\sqrt{\beta}}M_p^{-1}(F_p + F_p^T)K_p^{-1}.$$

As for the block diagonal preconditioner, we use the fact that $BM^{-1}B^T \approx K_p$. We then take $\mathcal{L} = \left(-\nu\nabla^2 + \bar{\mathbf{v}} \cdot \nabla + \frac{1}{\sqrt{\beta}}I\right) \cdot \left(-\nu\nabla^2 + \bar{\mathbf{v}} \cdot \nabla + \frac{1}{\sqrt{\beta}}I\right)^T$ in the above commutator argument to obtain that $B\hat{S}_{CD}^{-1}B^T \approx \hat{S}_{NS,2}^{-1}$.

Putting all the pieces together, we may postulate that

$$\mathcal{P}_T = \begin{bmatrix} M & 0 & 0 & 0 \\ F & -\hat{S}_{CD} & 0 & 0 \\ B & 0 & K_p & 0 \\ 0 & B & B\hat{S}_{CD}^{-1}FM^{-1}B^T & -\hat{S}_{NS,2}^{-1} \end{bmatrix}$$

is an appropriate choice of a block triangular preconditioner for \mathcal{A} . We may incorporate this into an iterative method such as GMRES.

Having derived our two proposed preconditioners \mathcal{P}_D and \mathcal{P}_T , we now examine the dominant processes required to apply the inverses of these preconditioners – we of course do not invert any of the matrices exactly, but instead approximate them. We approximate the inverse of a mass matrix by Chebyshev semi-iteration as discussed in [28], and we deal with the matrix K_p by using an algebraic multigrid (AMG) routine `HSL_MI20` from the Harwell Subroutine Library (HSL) [4]. Whenever the matrix $F + \frac{1}{\sqrt{\beta}}M$ or its transpose appears, we also use the same AMG routine, but we note that, for flows with a very large Reynolds number (which is a harder problem numerically due to dominant convective terms within the matrix), we would need

to apply a more specialized multigrid routine, such as that described in [22] for the forward convection-diffusion problem.⁶

Below we detail the dominant operations required to approximate \mathcal{P}_D^{-1} (for this we view \mathcal{P}_D as a 4×4 block matrix, and refer to the blocks as such):

- **(1, 1):** 1 Chebyshev semi-iteration for M
- **(2, 2):** 2 multigrid operations: 1 for $F + \frac{1}{\sqrt{\beta}}M$ and 1 for its transpose
- **(3, 3):** 1 multigrid operation for K_p
- **(4, 4):** 1 multigrid operation for K_p , and 2 Chebyshev semi-iterations for M_p
- **Total:** 3 Chebyshev semi-iterations and 4 multigrids (2 dealing with terms involving convection).

We similarly detail the dominant operations required to approximate \mathcal{P}_T^{-1} :

- **(1, 1):** 1 Chebyshev semi-iteration for M
- **(2, 2):** 2 multigrid operations: 1 for $F + \frac{1}{\sqrt{\beta}}M$ and 1 for its transpose
- **(3, 3):** 1 multigrid operation for K_p
- **(4, 3):** 1 Chebyshev semi-iteration for M_p , and 2 multigrid operations: 1 for $F + \frac{1}{\sqrt{\beta}}M$ and 1 for its transpose
- **(4, 4):** 1 multigrid operation for K_p , and 3 Chebyshev semi-iterations for M_p
- **Total:** 5 Chebyshev semi-iterations and 6 multigrids (4 dealing with terms involving convection).

We note that a single application of the inverse of \mathcal{P}_T is hence more expensive than an application of the inverse of \mathcal{P}_D , and therefore a fixed number of GMRES iterations will be cheaper when used with the preconditioner \mathcal{P}_D . We also comment that one convenient feature of both preconditioners is that one never has to invert the matrices F_p or F_p^T exactly. When they appear in the preconditioners, a matrix multiply is involved rather than an inversion – this is a positive aspect of our preconditioners as these matrices may contain large convective terms, and hence applying a multigrid routine to them may be troublesome numerically.

We will demonstrate the potential effectiveness of both proposed preconditioners in the next section. We will use the GMRES method to show this, but we note that other methods, such as BICGSTAB [27], could also be applied.

4. Numerical Results. In this section, we test our proposed solvers on the following two examples:

- **Problem 1:** We consider the following optimal control variant of the lid-driven cavity problem:

$$\begin{aligned}
 \min_{\underline{\mathbf{v}}, \underline{\mathbf{u}}} \quad & \frac{1}{2} \|\underline{\mathbf{v}}\|_{L_2(\Omega)}^2 + \frac{\beta}{2} \|\underline{\mathbf{u}}\|_{L_2(\Omega)}^2 \\
 \text{s.t.} \quad & -\nu \nabla^2 \underline{\mathbf{v}} + (\underline{\mathbf{v}} \cdot \nabla) \underline{\mathbf{v}} + \nabla p = \underline{\mathbf{u}}, \quad \text{in } \Omega := [-1, 1]^2, \\
 & -\nabla \cdot \underline{\mathbf{v}} = 0, \quad \text{in } \Omega, \\
 & \underline{\mathbf{v}} = \begin{cases} [1, 0]^T & \text{on } [-1, 1] \times \{1\}, \\ [0, 0]^T & \text{on } \partial\Omega \setminus ([-1, 1] \times \{1\}). \end{cases}
 \end{aligned}$$

⁶In the case of high Reynolds number flow, the problem may require strategies such as stabilization as well, which would be taken account of within the multigrid routine. As this is a specialized subject area, with the appropriate stabilization technique highly dependent on the type of finite elements used to solve the problem, we do not investigate this in this article, but instead provide a more general picture of the strategies required to solve this problem.

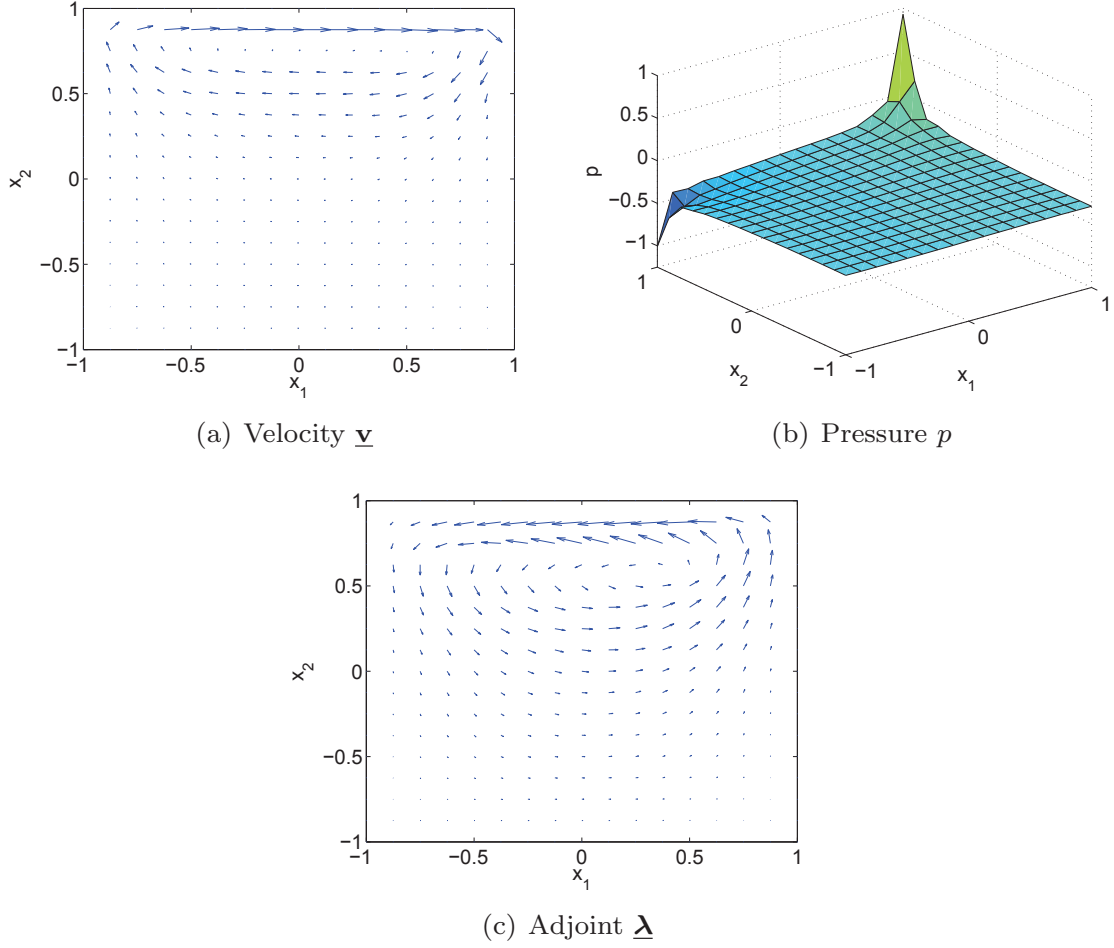


FIG. 4.1. *Solution plots for velocity $\underline{\mathbf{v}}$, pressure p and adjoint $\underline{\lambda}$ for Problem 1, with $\beta = 1$ and $Re = 100$.*

- **Problem 2:** We consider a target state involving a recirculating wind near to the boundary and zero velocity near the centre of the domain, with the problem statement as follows:

$$\begin{aligned}
 \min_{\underline{\mathbf{v}}, \underline{\mathbf{u}}} \quad & \frac{1}{2} \|\underline{\mathbf{v}} - \widehat{\underline{\mathbf{v}}}\|_{L_2(\Omega)}^2 + \frac{\beta}{2} \|\underline{\mathbf{u}}\|_{L_2(\Omega)}^2 \\
 \text{s.t.} \quad & -\nu \nabla^2 \underline{\mathbf{v}} + (\underline{\mathbf{v}} \cdot \nabla) \underline{\mathbf{v}} + \nabla p = \underline{\mathbf{u}}, \quad \text{in } \Omega := [-1, 1]^2, \\
 & -\nabla \cdot \underline{\mathbf{v}} = 0, \quad \text{in } \Omega, \\
 & \underline{\mathbf{v}} = \widehat{\underline{\mathbf{v}}}, \quad \text{on } \partial\Omega,
 \end{aligned}$$

where

$$\widehat{\underline{\mathbf{v}}} = \begin{cases} \left[\frac{1}{2}x_2(1 - x_1^2), -\frac{1}{2}x_1(1 - x_2^2) \right]^T & \text{if } x_1^2 + x_2^2 \geq \frac{1}{2}, \\ [0, 0]^T & \text{otherwise,} \end{cases}$$

and $\mathbf{x} = [x_1, x_2]^T$ denotes the spatial coordinates.

We present solution plots for these problems in Figures 4.1 and 4.2. We solve these problems iteratively with an outer iteration tolerance of 10^{-5} (we measure convergence using the ratio of the vector 2-norm of the difference between the current and previous iterates for $\underline{\mathbf{v}}$ divided by the vector 2-norm of the previous iterate for $\underline{\mathbf{v}}$) and with a GMRES tolerance of 10^{-6} . When applying our preconditioners, we use 20 steps of Chebyshev semi-iteration whenever we need to approximate the inverse of a mass

\mathcal{P}_D		β					
Re = 50		10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
h	2^{-3}	5 67	5 55	4 44	4 (34)*	3 (23)*	2 (18)*
	2^{-4}	5 83	4 70	4 58	4 48	3 (34)*	3 (24)*
	2^{-5}	5 104	4 85	3 71	3 60	3 48	3 (36)*
	2^{-6}	5 124	4 97	3 80	3 71	3 62	3 50
	2^{-7}	5 158	4 118	3 95	3 80	3 71	3 62

\mathcal{P}_D		β					
Re = 50		10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
h	2^{-3}	0.389	0.331	0.252	(0.374)*	(0.237)*	(0.181)*
	2^{-4}	1.23	1.20	1.00	0.83	(1.87)*	(1.28)*
	2^{-5}	5.93	4.90	4.80	4.50	5.88	(9.56)*
	2^{-6}	33.6	25.4	20.3	19.8	21.3	16.9
	2^{-7}	191	140	111	90.5	82.1	99.5

TABLE 4.1

Top: Number of outer iterations (in blue) and average number of GMRES iterations per outer iteration (rounded to the nearest integer) when solving Problem 1 with preconditioner \mathcal{P}_D , for a variety of h and β , and with $Re = 50$. Bottom: Average CPU times (in seconds) for the same values.

\mathcal{P}_D		β					
$h = 2^{-5}$		1	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}
Re	1	3 182	3 180	3 157	2 135	2 119	2 106
	10	4 183	3 140	3 111	3 92	3 80	3 71
	50	7 144	5 104	4 85	3 71	3 60	3 48
	100	8 122	5 93	4 74	4 61	4 50	3 39
	200	12 118	6 87	5 65	4 52	4 41	3 (30)*

TABLE 4.2

Number of outer iterations (in blue) and average number of GMRES iterations per outer iteration (rounded to the nearest integer) when solving Problem 1 with preconditioner \mathcal{P}_D , for a variety of β and Re , and with $h = 2^{-5}$.

matrix, and 2 V-cycles of the AMG routine from the Harwell Subroutine Library [4] when approximating the inverses of all other matrices. In order to construct the relevant finite element matrices, we use and modify code from the IFISS software system [8, 24], from which we also modify the version of the GMRES code for solving our problems.

We wish to test the performance of our methods for different values of h (which we define to be the mesh-size between **Q2**-nodes), regularization parameter β and viscosity ν .⁷ We define the Reynolds number of the flow we consider to be $Re := \frac{2}{\nu}$, as we are working on a domain of length scale 2. In Table 4.1, we fix the value of Re to be 50, and test our preconditioner \mathcal{P}_D on Problem 1 for a variety of h and β . We display the number of Oseen iterations (with our initial guess the solution

⁷All results are obtained using a tri-core 2.5 GHz workstation.

\mathcal{P}_T		β					
Re = 50		10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
h	2^{-3}	5 45	5 30	4 23	4 (18)*	3 (14)*	2 (12)*
	2^{-4}	5 61	4 41	4 31	4 24	3 (19)*	3 (15)*
	2^{-5}	5 72	4 59	3 42	3 32	3 24	3 (20)*
	2^{-6}	5 86	4 68	3 52	3 43	3 34	3 24
	2^{-7}	5 113	4 82	3 64	3 52	3 45	3 37

\mathcal{P}_T		β					
Re = 50		10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
h	2^{-3}	0.421	0.288	0.213	(0.348)*	(0.268)*	(0.234)*
	2^{-4}	1.46	1.17	0.905	0.710	(2.07)*	(1.58)*
	2^{-5}	6.80	5.57	4.96	4.31	5.56	(10.8)*
	2^{-6}	37.8	29.6	22.1	20.4	22.1	14.9
	2^{-7}	227	160	125	101	86.1	107

TABLE 4.3

Top: Number of outer iterations (in blue) and average number of GMRES iterations per outer iteration (rounded to the nearest integer) when solving Problem 1 with preconditioner \mathcal{P}_T , for a variety of h and β , and with $Re = 50$. Bottom: Average CPU times (in seconds) for the same values.

\mathcal{P}_T		β					
$h = 2^{-5}$		1	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}
Re	1	3 143	3 136	3 115	2 99	2 84	2 71
	10	4 130	3 102	3 77	3 67	3 50	3 40
	50	7 105	5 72	4 59	3 42	3 32	3 24
	100	8 88	5 65	4 46	4 35	4 25	3 20
	200	12 84	6 60	5 39	4 28	4 21	3 (17)*

TABLE 4.4

Number of outer iterations (in blue) and average number of GMRES iterations per outer iteration (rounded to the nearest integer) when solving Problem 1 with preconditioner \mathcal{P}_T , for a variety of β and Re , and with $h = 2^{-5}$.

of the equivalent Stokes control problem – that is the same problem but without the convective terms), the average number of GMRES iterations per Oseen iteration, and the average CPU time for each Oseen iteration. Table 4.3 displays the same results with preconditioner \mathcal{P}_T . We also wish to examine the effect of altering the Reynolds number. Therefore, in Tables 4.2 and 4.4, we present results from tests of the preconditioners \mathcal{P}_D and \mathcal{P}_T on Problem 1 for a variety of values of β and Re , with a fixed mesh-size $h = 2^{-5}$.⁸

⁸In Tables 4.1–4.6, the numbers in brackets and labelled * correspond to values where the AMG routine used does not work, due to the presence of positive off-diagonal entries. We use direct solves for such matrices in place of a multigrid routine. We note that this only applies in the case where h is large and β is small, which is not an interesting parameter regime in practice.

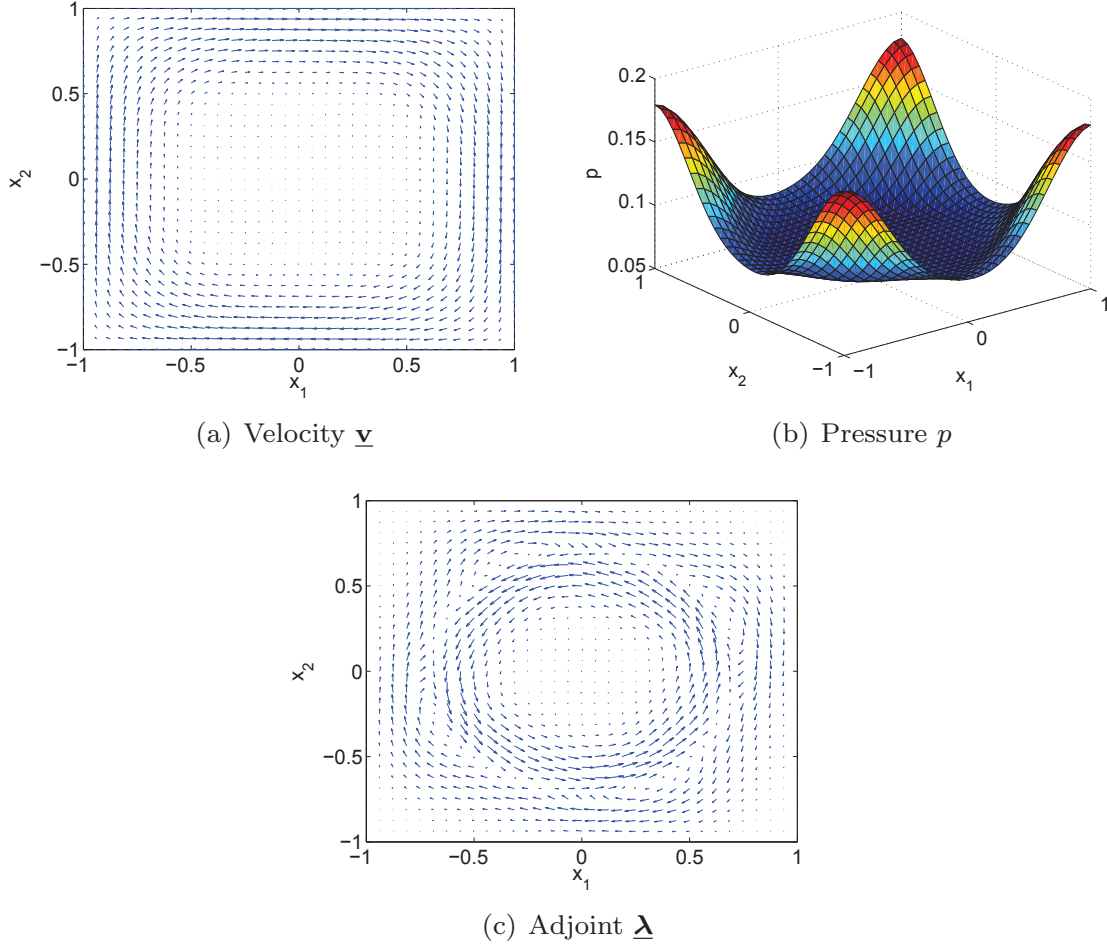


FIG. 4.2. Solution plots for velocity \underline{v} , pressure p and adjoint $\underline{\lambda}$ for Problem 2, with $\beta = 1$ and $Re = 200$.

From the tables, we first note that the number of outer (Oseen/Picard) iterations is reasonable for all parameter values tested (though the number rises as the Reynolds number is increased), so we believe our choice of this outer iteration is an appropriate one for these problems. Looking at the average number of GMRES iterations and CPU times in Tables 4.1 and 4.3, we note a benign dependence on h when using our solvers, though we believe that the increase in iteration numbers as h decreases is reasonable, as the size of the matrix system increases by roughly a factor of 4 as h is halved. Our methods also perform better as β and ν are decreased, as shown in Tables 4.2 and 4.4. The decrease in iteration numbers as ν is decreased (i.e. for higher Reynolds numbers) is in some sense surprising, though we point out that the accuracy of the finite element solution is likely to be worse in these cases for a fixed h .⁹

In Tables 4.5 and 4.6, we display the iteration numbers taken to solve Problem 2 using preconditioners \mathcal{P}_D and \mathcal{P}_T , for a range of h , β and Re . We see that for this harder problem, the iteration numbers are slightly larger, but all are still reasonable given the complexity of the problem.

From the results obtained, we observe that our two solvers involving the preconditioners \mathcal{P}_D and \mathcal{P}_T perform quite similarly. However, it appears that, although the block triangular preconditioner \mathcal{P}_T consistently solves the problem in fewer iterations, the block diagonal preconditioner \mathcal{P}_D does so in lower CPU time for the majority of

⁹We also note that beyond values of $Re \approx 200$, the AMG routine used begins to struggle due to the dominant convective terms within the relevant matrices, so a more sophisticated multigrid routine would need to be employed.

\mathcal{P}_D		β				
Re = 50		10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}
h	2^{-3}	4 88	4 61	4 44	3 (35)*	2 (30)*
	2^{-4}	4 116	3 85	3 60	3 44	3 (33)*
	2^{-5}	4 155	3 108	3 75	3 54	3 41
	2^{-6}	4 218	3 143	3 94	3 66	2 50

\mathcal{P}_D		β				
Re = 100		10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}
h	2^{-3}	6 84	5 52	4 (36)*	3 (30)*	2 (27)*
	2^{-4}	5 120	4 78	4 51	3 38	3 (29)*
	2^{-5}	5 167	3 102	3 68	3 45	3 35
	2^{-6}	5 244	3 135	3 84	3 59	3 44

TABLE 4.5

Number of outer iterations (in blue) and average number of GMRES iterations per outer iteration (rounded to the nearest integer) when solving Problem 2 with preconditioner \mathcal{P}_D , for a variety of h and β , with $Re = 50$ and $Re = 100$.

\mathcal{P}_T		β				
Re = 50		10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}
h	2^{-3}	4 56	4 35	4 24	3 (20)*	2 (16)*
	2^{-4}	4 81	3 54	3 35	3 24	3 (20)*
	2^{-5}	4 111	3 74	3 48	3 31	3 22
	2^{-6}	4 151	3 93	3 63	3 45	2 30

\mathcal{P}_T		β				
Re = 100		10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}
h	2^{-3}	6 54	5 31	4 (22)*	3 (17)*	2 (14)*
	2^{-4}	5 79	4 46	4 28	3 21	3 (17)*
	2^{-5}	5 115	3 70	3 42	3 26	3 19
	2^{-6}	5 169	3 94	3 58	3 38	3 25

TABLE 4.6

Number of outer iterations (in blue) and average number of GMRES iterations per outer iteration (rounded to the nearest integer) when solving Problem 2 with preconditioner \mathcal{P}_T , for a variety of h and β , with $Re = 50$ and $Re = 100$.

parameter values studied. This is due to the larger number of operations required to apply the preconditioner \mathcal{P}_T , as detailed in the previous section. Importantly however, we have demonstrated that solving a number of complicated Navier-Stokes control problems is feasible for a range of parameter values using either of our methods, and we believe that the iteration numbers obtained are satisfactory considering the complexity of the problem. To illustrate the importance of developing such feasible iterative methods for Navier-Stokes control, we compare direct and iterative solution strategies in Table 4.7 – here it is shown that our method gives solutions in reasonable times for matrix systems which are sufficiently large that direct methods

h	SIZE	DIRECT	ITERATIVE
2^{-3}	1,318	0.050	0.272
2^{-4}	4,934	0.321	1.11
2^{-5}	19,078	1.76	4.12
2^{-6}	75,014	16.1	23.5
2^{-7}	297,478	—	121
2^{-8}	1,184,774	—	1240

TABLE 4.7

CPU times (in seconds) for solving Problem 1, with sparse direct solves in MATLAB, and our iterative approach with preconditioner \mathcal{P}_D , for a range of h and with $\beta = 10^{-2}$, $Re = 100$. Where ‘—’ is denoted, the sparse direct method failed to give a solution.

fail when attempting to solving them.

5. Conclusions and Possible Extensions. In this article, we have discussed, derived and tested preconditioned iterative methods for the time-independent distributed Navier-Stokes control problem. From an applications point-of-view, it is highly desirable to develop solvers such as these which are not only feasible, but efficient and rapid given the complexity of the problem. We have motivated our solvers using saddle point theory, preconditioners for the simpler convection-diffusion control problem, and commutator arguments to approximate complex matrices in a computationally cheap way. Promising numerical results indicate the feasibility and effectiveness of our proposed solution strategies.

There are a number of possible extensions to this work. One is to consider solving this problem using a Newton iteration. For this problem, one would need to develop ways of dealing with convection-dominated terms within the $(1, 1)$ -block of the saddle point system, possibly using a radically different strategy to that discussed in this paper. However, if this could be done, it would be very worthwhile, due to the more reliable convergence of the outer iteration which will result. It would also be useful to adapt this strategy to problems with Neumann boundary conditions involved, in particular boundary control variants of this problem, or such problems involving additional inequality constraints on the states or control. Finally, one could consider the time-dependent analogue of the Navier-Stokes control problem, perhaps using strategies based on those applied to problems involving the optimal control of the heat equation in [18] for instance. Finding efficient solvers for such a time-dependent problem would be a development of considerable practical interest and a useful extension to this research.

Acknowledgements. The author would like to thank Andy Wathen and Martin Stoll for helpful discussions about this research. The author was supported for this work by the Engineering and Physical Sciences Research Council (UK), Grant EP/P505216/1.

REFERENCES

- [1] M. Benzi, G. H. Golub, and J. Liesen, *Numerical Solution of Saddle Point Problems*, Acta Numerica, 14, pp.1–137, 2005.
- [2] G. Biros, and O. Ghattas, *Parallel Lagrange-Newton-Krylov-Schur Methods for PDE-*

- Constrained Optimization. Part I: The Krylov-Schur Solver*, SIAM Journal on Scientific Computing, 27(2), pp.687–713, 2005.
- [3] G. Biros, and O. Ghattas, *Parallel Lagrange-Newton-Krylov-Schur Methods for PDE-Constrained Optimization. Part II: The Lagrange-Newton Solver and its Applications to Optimal Control of Steady Viscous Flows*, SIAM Journal on Scientific Computing, 27(2), pp.714–739, 2005.
 - [4] J. Boyle, M. D. Mihajlovic, and J. A. Scott JA, *HSL_MI20: An Efficient AMG Preconditioner for Finite Element Problems in 3D*, International Journal for Numerical Methods in Engineering, 82, pp.64–98, 2010.
 - [5] J. Cahouet, and J.-P. Chabard, *Some Fast 3D Finite Element Solvers for the Generalized Stokes Problem*, International Journal for Numerical Methods in Fluids, 8, pp.869–895, 1988.
 - [6] J. C. De Los Reyes, and R. Griesse, *State-Constrained Optimal Control of the Three-Dimensional Stationary Navier-Stokes Equations*, Journal of Mathematical Analysis and Applications, 343, pp.257–272, 2008.
 - [7] J. C. De Los Reyes, and F. Tröltzsch, *Optimal Control of the Stationary Navier-Stokes Equations with Mixed Control-State Constraints*, SIAM Journal on Control and Optimization, 46, pp.604–629, 2007.
 - [8] H. C. Elman, A. Ramage, and D. Silvester, *Algorithm 866: IFISS, a MATLAB Toolbox for Modelling Incompressible Flow*, ACM Transactions on Mathematical Software, 33, pp.2–14, 2007.
 - [9] 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.
 - [10] M. Hinze, M. Köster, and S. Turek, *A Space-Time Multigrid Solver for Distributed Control of the Time-Dependent Navier-Stokes System*, Priority Programme 1253, Preprint Number SPP1253-16-02, 2008.
 - [11] M. Hinze, and K. Kunisch, *Second Order Methods for Optimal Control of Time-Dependent Fluid Flow*, SIAM Journal on Control and Optimization, 40, pp.925–946, 2001.
 - [12] I. C. F. Ipsen, *A Note on Preconditioning Non-Symmetric Matrices*, SIAM Journal on Scientific Computing, 23(3), pp.1050–1051, 2001.
 - [13] C. John, and D. Wachsmuth, *Optimal Dirichlet Boundary Control of Navier-Stokes Equations with State Constraint*, Numerical Functional Analysis and Optimization, 30(11&12), pp.1309–1338, 2009.
 - [14] D. Kay, D. Loghin, and A. Wathen, *A Preconditioner for the Steady-State Navier-Stokes Equations*, SIAM Journal on Scientific Computing, 24(1), pp.237–256, 2002.
 - [15] Y. A. Kuznetsov, *Efficient Iterative Solvers for Elliptic Finite Element Problems on Nonmatching Grids*, Russian Journal of Numerical Analysis and Mathematical Modelling, 10(3), pp.187–211, 1995.
 - [16] 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.
 - [17] J. W. Pearson, *On the Role of Commutator Arguments in the Development of Parameter-Robust Preconditioners for Stokes Control Problems*, submitted to IMA Journal of Numerical Analysis, 2012.
 - [18] J. W. Pearson, M. Stoll, and A. J. Wathen, *Regularization-Robust Preconditioners for Time-Dependent PDE-Constrained Optimization Problems*, SIAM Journal on Matrix Analysis and Applications, 33(4), pp.1126–1152, 2012.
 - [19] J. W. Pearson, and A. J. Wathen, *Fast Iterative Methods for Convection-Diffusion Control Problems*, submitted to Electronic Transactions on Numerical Analysis, 2011.
 - [20] J. W. Pearson, and A. J. Wathen, *A New Schur Complement Approximation in Preconditioners for PDE-Constrained Optimization*, Numerical Linear Algebra with Applications, 19(5), pp.816–829, 2012.
 - [21] M. Pošta, and T. Roubíček, *Optimal Control of Navier-Stokes Equations by Oseen Approximation*, Computers and Mathematics with Applications, 53, pp.569–581, 2007.
 - [22] A. Ramage, *A Multigrid Preconditioner for Stabilised Discretisations of Advection-Diffusion Problems*, Journal of Computational and Applied Mathematics, 110 (1999), pp.187–203.
 - [23] Y. Saad, and M.H. Schultz, *GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems*, SIAM Journal on Scientific and Statistical Computation, 7, pp.856–869, 1986.
 - [24] D. Silvester, H. Elman, and A. Ramage, *Incompressible Flow and Iterative Solver Software (IFISS)*, version 3.1, <http://www.manchester.ac.uk/ifiss/>, 2011.
 - [25] F. Tröltzsch, *Optimal Control of Partial Differential Equations: Theory, Methods and Appli-*

- cations*, Graduate Series in Mathematics, American Mathematical Society, 2010.
- [26] M. Ulbrich, *Constrained Optimal Control of Navier-Stokes Flow by Semismooth Newton Methods*, Systems & Control Letters, 48, pp.297–311, 2003.
 - [27] H. A. Van der Vorst, *BICGSTAB: A Fast and Smoothly Converging Variant of BICG for the Solution of Nonsymmetric Linear Systems*, SIAM Journal on Scientific and Statistical Computation, 13, pp.631–644, 1992.
 - [28] 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.
 - [29] W. Zulehner, *Nonstandard Norms and Robust Estimates for Saddle Point Problems*, SIAM Journal on Matrix Analysis and Applications, 32(2), pp.536–560, 2011.