

# AN AUGMENTED LAGRANGIAN PRECONDITIONER FOR THE CONTROL OF THE NAVIER–STOKES EQUATIONS

SANTOLO LEVEQUE\*, MICHELE BENZI†, AND PATRICK E. FARRELL‡

**Abstract.** We address the solution of the distributed control problem for the steady, incompressible Navier–Stokes equations. We propose an inexact Newton linearization of the optimality conditions. Upon discretization by a finite element scheme, we obtain a sequence of large symmetric linear systems of saddle-point type. We use an augmented Lagrangian-based block triangular preconditioner in combination with the flexible GMRES method at each Newton step. The preconditioner is applied inexactly via a suitable multigrid solver. Numerical experiments indicate that the resulting method appears to be fairly robust with respect to viscosity, mesh size, and the choice of regularization parameter when applied to 2D problems.

**Key words.** distributed control, incompressible Navier–Stokes equations, KKT conditions, finite elements, inexact Newton, augmented Lagrangian, preconditioning, multigrid

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

**1. Introduction.** In this work, we consider the distributed control of the flow of a Newtonian viscous fluid, subject to the incompressibility constraint. We restrict ourselves to steady problems; in this case, the physics is described by the stationary, incompressible Navier–Stokes equations.

The distributed control of the stationary incompressible Navier–Stokes equations consists of the minimization of a cost functional subject to the Navier–Stokes equations with the introduction of a control variable in the system. The control is supposed to act on the whole spatial domain. The formulation of the problem is as follows: given a bounded Lipschitz domain  $\Omega \subset \mathbb{R}^d$ ,  $d \in \{2, 3\}$ , the Navier–Stokes control problem considered in this work is defined as

$$\min_{\vec{v}, \vec{u}} J(\vec{v}, \vec{u}) = \frac{1}{2} \int_{\Omega} |\vec{v}(\mathbf{x}) - \vec{v}_d(\mathbf{x})|^2 \, d\Omega + \frac{\beta}{2} \int_{\Omega} |\vec{u}(\mathbf{x})|^2 \, d\Omega \quad (1.1)$$

subject to

$$\begin{cases} -\nu \nabla^2 \vec{v} + \vec{v} \cdot \nabla \vec{v} + \nabla p = \vec{u} + \vec{f}(\mathbf{x}) & \text{in } \Omega, \\ -\nabla \cdot \vec{v}(\mathbf{x}) = 0 & \text{in } \Omega, \\ \vec{v}(\mathbf{x}) = \vec{g}(\mathbf{x}) & \text{on } \partial\Omega, \end{cases} \quad (1.2)$$

where  $\vec{v}$  and  $p$  are the *state velocity* and *state pressure*, respectively, and  $\vec{u}$  is the *control* (the precise spaces in which solutions are sought will be specified later). The functions  $\vec{f}$  and  $\vec{g}$  represent the forcing term acting on the system and suitable boundary conditions, respectively. Further,  $\vec{v}_d$  is the desired (velocity) state, and the parameter  $\beta > 0$  is a regularization parameter. Finally, the parameter  $\nu > 0$  represents the kinematic viscosity of the fluid. The type of flow described by the Navier–Stokes equations is influenced by the viscosity  $\nu$ . Specifically, denoting by  $L$  and  $V$  the characteristic length and velocity scales of the flow, respectively, the nature of the flow

---

\*CRM Ennio De Giorgi, Scuola Normale Superiore, Piazza dei Cavalieri 3, 56126, Pisa, Italy ([santolo.leveque@sns.it](mailto:santolo.leveque@sns.it)).

†Scuola Normale Superiore, Piazza dei Cavalieri 7, 56126 Pisa, Italy ([michele.benzi@sns.it](mailto:michele.benzi@sns.it)).

‡Mathematical Institute, University of Oxford, Radcliffe Observatory Quarter, Woodstock Road, Oxford OX2 6GG, UK ([patrick.farrell@maths.ox.ac.uk](mailto:patrick.farrell@maths.ox.ac.uk)).

is determined by the Reynolds number, defined as  $Re = \frac{LV}{\nu}$ : from small to moderate Reynolds number, the flow is laminar, while at high Reynolds number the flow may become turbulent. From a point of view of applications, the distributed control of incompressible viscous fluid can be realized, for example, in polarizable fluids by applying an electromagnetic force, see [57]. Other applications can be found in, e.g., [22] and references therein.

The non-linearity of the PDEs (1.2) obviously requires one to employ a non-linear iteration in order to obtain an approximation of the solution of a Navier–Stokes control problem. The aim of this work is to derive effective preconditioners for the linear systems arising from the non-linear iteration employed. In [32], Leveque and Pearson considered a Picard iteration applied to the control of the (stationary and instationary) Navier–Stokes equations. The Picard linearization of the first-order optimality conditions of the Navier–Stokes control problem results in two coupled *Oseen equations*, to be solved simultaneously. By making use of a block-commutator argument, the authors in [32] were able to derive a robust preconditioner for the resulting discretized Oseen equations. The preconditioner showed a mild dependence on the viscosity of the fluid, while being robust with respect to the mesh-size. With this approach, the authors of [32] were able to solve problems with a rather small viscosity. However, as the viscosity decreased, an increase in the number of Picard iterations required for reaching a prescribed reduction on the non-linear relative residual was observed. This is not surprising, as the contraction factor governing the (linear) convergence of the Picard iteration applied to the stationary Navier–Stokes equations is known to approach 1 as the Reynolds number approaches its critical value  $Re_*$  [28, 23]. One way to obtain faster convergence of the non-linear iteration is to employ second-order methods, such as Newton’s method, which, however, necessitates a good initial guess; the smaller the viscosity, the closer the initial guess must be to the solution in order for Newton’s method to converge; the convergence is, eventually, quadratic [23, 28]. While these considerations apply to the forward problem, one can expect them to hold also for the control problem.

In the context of optimization problems, Newton’s method makes use of second derivative information and, generally speaking, the linear systems arising from Newton’s method are more complicated, and harder to solve, than the ones arising from Picard’s iteration. For this reason, but also in order to avoid computing the full Hessian matrix, use is often made of suitable approximations. One class of methods obtained in this way are the so called *inexact Newton methods*, see [37]. In this framework, one can express the derivatives of the cost functional in terms of the Jacobian of the residual. From here, a new approximation of a solution of the problem can be obtained by neglecting certain second-order terms arising from the Hessian. Additionally, the linear systems involved in the computation of the Newton steps need not be solved to high accuracy.

For the inexact Newton iteration applied to the stationary Navier–Stokes control problem to be viable, efficient and robust preconditioners are needed in the solution of the resulting linear systems. Despite the excellent performance of the preconditioner derived in [32] for the discretized Picard linearization of the first-order optimality conditions, one cannot expect this preconditioner to be efficient when applied in the context of an inexact Newton method. In fact, the *Newton matrix* arising in a Newton step is not defined on the pressure space, thus it cannot be included in the commutator argument. From here, one can expect the block-commutator approach to perform poorly for problems with moderate to very small viscosity (a fact that is already

known for the forward problem, see, for instance, [16, Section 9.3.3]).

A remarkably robust approach for solving for the Newton step applied to the (forward) Navier–Stokes equations is augmented Lagrangian-based preconditioning. This type of preconditioner was developed in [7] for the solution of the linear systems arising from a Picard linearization of the stationary 2D Navier–Stokes equations. The crucial ingredient in this approach is a specialized multigrid solver used to approximately solve the augmented momentum equation. The preconditioner was shown to be robust with respect to the mesh-size and, more importantly, to the viscosity of the fluid. In [17] this strategy was extended to the solution of the linear systems arising from a Newton linearization of the 3D Navier–Stokes equations. Again, the robustness of the preconditioner was confirmed, allowing the authors to solve problems with very small viscosity. In the present work, we employ an augmented Lagrangian preconditioner in the solution of linear systems of saddle-point type arising in the inexact Newton iteration applied to the stationary incompressible Navier–Stokes control problem. We emphasize that the structure of these linear systems is quite different from the one considered in [7, 17].

The paper is structured as follows. In Section 2, we introduce the non-linear iteration employed in this work for the solution of the stationary Navier–Stokes control problem (1.1)–(1.2). In Section 3, we introduce the augmented Lagrangian preconditioner employed in this work for solving the linear systems arising from the linearization of the problem adopted. In Section 4, we present numerical experiments showing the robustness of the proposed methodology. Finally, in Section 5 we draw our conclusions and present future developments of this work.

**2. Inexact Newton iteration.** In this section, we describe the non-linear iteration that we employ in order to solve the stationary incompressible Navier–Stokes control problem (1.1)–(1.2).

In order to introduce the non-linear iteration, one has to derive the first-order optimality conditions (also called *Karush–Kuhn–Tucker (KKT) conditions*). This can be done in two ways. On one hand, one can first derive the optimality conditions in the infinite-dimensional Hilbert space where the solutions are sought, then discretize the conditions so obtained; this approach is known as *optimize-then-discretize*. On the other hand, one can first discretize the cost functional and the constraints in (1.1)–(1.2), then derive the optimality conditions by mean of classical optimization theorems in the finite-dimensional setting; this approach is known as *discretize-then-optimize*.

In the following, we derive the Newton iteration when employing either the optimize-then-discretize or the discretize-then-optimize approach. We opt for this choice for two reasons. First, we show the flexibility of the augmented Lagrangian preconditioner when either one of the above strategies is employed. Second, numerical examples demonstrate the effectiveness and robustness of the approach, for a wide range of problem parameters.

The Newton iteration produces a sequence of approximations  $\vec{v}^{(k)}(\mathbf{x}) \approx \vec{v}(\mathbf{x})$  and  $\vec{u}^{(k)}(\mathbf{x}) \approx \vec{u}(\mathbf{x})$  of the state velocity and control respectively. The pressures  $p(\mathbf{x})$  and  $\mu(\mathbf{x})$  arise in the optimality conditions as the Lagrange multipliers enforcing the adjoint and forward incompressibility constraints. We employ inf–sup stable finite element pairs to discretize the velocity–pressure pair. Letting  $\{\vec{\phi}_i\}_{i=1}^{n_v}$  and  $\{\psi_i\}_{i=1}^{n_p}$  be the basis functions for an inf–sup stable pair of finite elements for velocity and

pressure respectively, we introduce the following matrices:

$$\mathbf{K} = \left\{ \int_{\Omega} \nabla \vec{\phi}_i : \nabla \vec{\phi}_j \, d\Omega \right\}_{i,j=1}^{n_v}, \quad \mathbf{N}^{(k)} = \left\{ \int_{\Omega} (\vec{v}^{(k)} \cdot \nabla \vec{\phi}_j) \cdot \vec{\phi}_i \, d\Omega \right\}_{i,j=1}^{n_v},$$

$$\mathbf{M} = \left\{ \int_{\Omega} \vec{\phi}_i \cdot \vec{\phi}_j \, d\Omega \right\}_{i,j=1}^{n_v}, \quad B = \left\{ - \int_{\Omega} \psi_i \nabla \cdot \vec{\phi}_j \, d\Omega \right\}_{i=1, \dots, n_p}^{j=1, \dots, n_v}.$$

The matrix  $\mathbf{K}$  is generally referred to as a (vector-)stiffness matrix, and the matrix  $\mathbf{M}$  is referred to as a (vector-)mass matrix; both of these matrices are symmetric positive definite (s.p.d.). The matrix  $\mathbf{N}^{(k)}$  is referred to as a (vector-)convection matrix, and is skew-symmetric (i.e.  $\mathbf{N}^{(k)} + (\mathbf{N}^{(k)})^\top = 0$ ) if the “wind”  $\vec{v}^{(k)}$  is incompressible; finally, the matrix  $B$  is the (negative) divergence matrix. In addition to the previous matrices, we also introduce the following:

$$\mathbf{H}^{(k)} = \left\{ \int_{\Omega} (\vec{\phi}_j \cdot \nabla \vec{v}^{(k)}) \cdot \vec{\phi}_i \, d\Omega \right\}_{i,j=1}^{n_v}, \quad (2.1)$$

which is the matrix arising in Newton’s method from linearizing the convection term.

At high Reynolds number, the problem is convection-dominated. This requires us to make use of a stabilization procedure. In the following,  $\mathbf{W}^{(k)}$  denotes a possible stabilization matrix for the convection operator. It should be mentioned that the matrix  $\mathbf{W}^{(k)}$  represents a differential operator that is not related to the physics, and is introduced solely to enhance coercivity (that is, increase the positivity of the real part of the eigenvalues) of the discretization, thereby allowing it to be stable. Following the discussions in [31, 43], in some of the numerical experiments presented in this work we employ the Local Projection Stabilization (LPS) approach described in [3, 4, 11]. For an analysis of the order of convergence of the LPS applied to the forward Oseen problem, we refer to [35]. For other possible stabilizations applied to the forward Oseen problem, see, for instance, [12, 19, 27, 54].

We would like to mention here that the choice of employing the LPS stabilization has been made for an important reason. When solving the control of the convection–diffusion equation, the discretize-then-optimize and the optimize-then-discretize approaches do not commute if a different stabilization is employed, see [14]. However, as shown in [4], when employing the LPS approach to the control of the stationary convection–diffusion equation, the discretization and optimization steps commute. Since at each Newton iteration one has to solve a convection–diffusion control problem, the choice of the LPS as stabilization is very natural.

The LPS formulation is defined as follows. Given  $\pi_h$ , an  $L^2$ -orthogonal (discontinuous) projection operator onto the finite dimensional space  $\bar{V}$  defined on patches of  $\Omega$ , where by a patch we mean a union of cells sharing a vertex, we consider the *fluctuation operator*  $\kappa_h = \text{Id} - \pi_h$ . Here,  $\text{Id}$  denotes the identity operator. Then, the matrix  $\mathbf{W}^{(k)}$  is defined as

$$\mathbf{W}^{(k)} = \left\{ \delta^{(k)} \int_{\Omega} \kappa_h(\vec{v}^{(k)} \cdot \nabla \vec{\phi}_i) \cdot \kappa_h(\vec{v}^{(k)} \cdot \nabla \vec{\phi}_j) \, d\Omega \right\}_{i,j=1}^{n_v}, \quad (2.2)$$

where  $\delta^{(k)} \geq 0$  is a stabilization parameter. As in [3], we define the projection by means of

$$\pi_h(q)|_{\mathbf{P}_m} = \frac{1}{|\mathbf{P}_m|} \int_{\mathbf{P}_m} q \, d\mathbf{P}_m, \quad \forall q \in L^2(\Omega),$$

where  $|\mathbf{P}_m|$  is the (Lebesgue) measure of the patch  $\mathbf{P}_m$ . We refer the reader to [35] for the motivation of the choice of this projection and for the possible choices of the finite dimensional space  $\bar{V}$ . Finally, as in [16, p.253] the stabilization parameter is taken to be

$$\delta_m^{(k)} = \begin{cases} \frac{h_m}{2\|\vec{v}_m^{(k)}\|} \left(1 - \frac{1}{Pe_m}\right) & \text{if } Pe_m > 1, \\ 0 & \text{if } Pe_m \leq 1, \end{cases}$$

where  $\|\vec{v}_m^{(k)}\|$  is the Euclidean norm of  $\vec{v}^{(k)}$  at the patch centroid,  $h_m$  is a measure of the patch length in the direction of the wind, and  $Pe_m = \|\vec{v}_m^{(k)}\|h_m/(2\nu)$  is the patch Péclet number. For the discretization to be stable, we expect  $Pe_m$  to be less than 1. We would like to mention that the choice we made of the stabilization parameter is only a heuristic.

In the following, we set  $\mathbf{D}^{(k)} = \nu\mathbf{K} + \mathbf{N}^{(k)} + \mathbf{W}^{(k)}$ . Further, we define

$$V := \{\vec{v} \in \mathcal{H}^1(\Omega)^d \mid \vec{v} = \vec{g} \text{ on } \partial\Omega\}, \quad V_0 := \{\vec{v} \in \mathcal{H}^1(\Omega)^d \mid \vec{v} = \vec{0} \text{ on } \partial\Omega\}, \quad Q := L_0^2(\Omega),$$

with  $\mathcal{H}^1(\Omega)^d$  the usual Sobolev space of square-integrable functions in  $\mathbb{R}^d$  with square-integrable weak derivatives.

**2.1. Optimize-then-Discretize.** We begin by considering the optimize-then-discretize approach, which makes use of the so called formal Lagrangian method, see [55, Section 2.10].

First, we derive the continuous KKT conditions. Introducing the adjoint velocity  $\vec{\zeta}$  and the adjoint pressure  $\mu$ , one may consider the Lagrangian associated with (1.1)–(1.2), as in [44]. Then, one can write the KKT conditions as

$$\left\{ \begin{array}{l} \left. \begin{array}{l} -\nu\nabla^2\vec{\zeta} - \vec{v} \cdot \nabla\vec{\zeta} + (\nabla\vec{v})^\top\vec{\zeta} + \nabla\mu = \vec{v}_d - \vec{v} \quad \text{in } \Omega \\ -\nabla \cdot \vec{\zeta}(x) = 0 \quad \text{in } \Omega \\ \vec{\zeta}(x) = \vec{0} \quad \text{on } \partial\Omega \end{array} \right\} \begin{array}{l} \text{adjoint} \\ \text{equations} \end{array} \\ \\ \left. \begin{array}{l} \beta\vec{u} - \vec{\zeta} = 0 \quad \text{in } \Omega \\ -\nu\nabla^2\vec{v} + \vec{v} \cdot \nabla\vec{v} + \nabla p = \vec{u} + \vec{f} \quad \text{in } \Omega \\ -\nabla \cdot \vec{v}(x) = 0 \quad \text{in } \Omega \\ \vec{v}(x) = \vec{g}(x) \quad \text{on } \partial\Omega \end{array} \right\} \begin{array}{l} \text{gradient} \\ \text{equation} \\ \text{state} \\ \text{equations} \end{array} \end{array} \right. \quad (2.3)$$

The expression of the gradient equation motivates us to take  $\vec{v}$  (thus,  $\vec{\zeta}$ ) and  $\vec{u}$  in the same space, so that we can eliminate  $\vec{u}$  from the system.

Problem (2.3) consists of a coupled system of non-linear PDEs. We employ an inexact Newton method in order to derive an approximation of the solutions. The goal is to write the Newton step for solving the KKT conditions in (2.3), and then neglect the *curvature term* of the full Hessian. In order to write the Newton system, we follow the work in [25].

Given  $\vec{v}^{(k)} \in V$ ,  $p^{(k)} \in Q$ ,  $\vec{\zeta}^{(k)} \in V_0$ ,  $\mu^{(k)} \in Q$  the current approximations to  $\vec{v}$ ,  $p$ ,  $\vec{\zeta}$ , and  $\mu$ , respectively, one can write the Newton iterates as

$$\begin{aligned} \vec{v}^{(k+1)} &= \vec{v}^{(k)} + \delta\vec{v}^{(k)}, & p^{(k+1)} &= p^{(k)} + \delta p^{(k)}, \\ \vec{\zeta}^{(k+1)} &= \vec{\zeta}^{(k)} + \delta\vec{\zeta}^{(k)}, & \mu^{(k+1)} &= \mu^{(k)} + \delta\mu^{(k)}, \end{aligned}$$

where the Newton corrections are solutions of the following system of PDEs (written in weak form):

$$\left\{ \begin{array}{l} \nu(\nabla \delta \vec{\zeta}^{(k)}, \nabla \vec{w}_1) - (\vec{v}^{(k)} \cdot \nabla \delta \vec{\zeta}^{(k)}, \vec{w}_1) + ((\nabla \vec{v}^{(k)})^\top \delta \vec{\zeta}^{(k)}, \vec{w}_1) \\ \quad - (\delta \mu^{(k)}, \nabla \cdot \vec{w}_1) + (\delta \vec{v}^{(k)}, \vec{w}_1) + (\vec{\zeta}^{(k)} \cdot \nabla \delta \vec{v}^{(k)}, \vec{w}_1) \\ \quad \quad \quad + (\delta \vec{v}^{(k)} \cdot \nabla \vec{\zeta}^{(k)}, \vec{w}_1) = \vec{R}_1^{(k)}, \\ \quad \quad \quad - (q_2, \nabla \cdot \delta \vec{\zeta}^{(k)}) = r_2^{(k)}, \\ \nu(\nabla \delta \vec{v}^{(k)}, \nabla \vec{w}_2) + (\vec{v}^{(k)} \cdot \nabla \delta \vec{v}^{(k)}, \vec{w}_2) + (\delta \vec{v}^{(k)} \cdot \nabla \vec{v}^{(k)}, \vec{w}_2) \\ \quad \quad \quad - (\delta p^{(k)}, \nabla \cdot \vec{w}_2) - \frac{1}{\beta} (\delta \vec{\zeta}^{(k)}, \vec{w}_2) = \vec{R}_2^{(k)}, \\ \quad \quad \quad - (q_1, \nabla \cdot \delta \vec{v}^{(k)}) = r_1^{(k)}, \end{array} \right. \quad (2.4)$$

for any  $(\vec{w}_1, \vec{w}_2, q_1, q_2) \in V_0 \times V_0 \times Q \times Q$ . The residuals  $\vec{R}_1^{(k)}$ ,  $r_1^{(k)}$ ,  $\vec{R}_2^{(k)}$ ,  $r_2^{(k)}$  are given by

$$\left\{ \begin{array}{l} \vec{R}_1^{(k)} = (\vec{v}_d, \vec{w}_1) - (\vec{v}^{(k)}, \vec{w}_1) - \nu(\nabla \vec{\zeta}^{(k)}, \nabla \vec{w}_1) + (\vec{v}^{(k)} \cdot \nabla \vec{\zeta}^{(k)}, \vec{w}_1) \\ \quad - ((\nabla \vec{v}^{(k)})^\top \vec{\zeta}^{(k)}, \vec{w}_1) + (\mu^{(k)}, \nabla \cdot \vec{w}_1), \\ r_2^{(k)} = (q_2, \nabla \cdot \vec{\zeta}^{(k)}), \\ \vec{R}_2^{(k)} = (\vec{f}, \vec{w}_2) - \nu(\nabla \vec{v}^{(k)}, \nabla \vec{w}_2) - (\vec{v}^{(k)} \cdot \nabla \vec{v}^{(k)}, \vec{w}_2) + (p^{(k)}, \nabla \cdot \vec{w}_2) \\ \quad + \frac{1}{\beta} (\vec{\zeta}^{(k)}, \vec{w}_2), \\ r_1^{(k)} = (q_1, \nabla \cdot \vec{v}^{(k)}). \end{array} \right. \quad (2.5)$$

We have now to derive the discrete system to solve at each non-linear iteration. Before doing this, we derive an inexact Newton method by neglecting the coupling terms  $(\vec{\zeta}^{(k)} \cdot \nabla \delta \vec{v}^{(k)}, \vec{w}_1)$  and  $(\delta \vec{v}^{(k)} \cdot \nabla \vec{\zeta}^{(k)}, \vec{w}_1)$  in (2.4), as done in [9], for instance. As we will show below, dropping these terms allows us to derive effective and robust preconditioners for the linearized inexact Newton system, while still maintaining second-order convergence of the method. The inexact Newton iteration we consider is given by the following system of PDEs:

$$\left\{ \begin{array}{l} (\delta \vec{v}^{(k)}, \vec{w}_1) + \nu(\nabla \delta \vec{\zeta}^{(k)}, \nabla \vec{w}_1) - (\vec{v}^{(k)} \cdot \nabla \delta \vec{\zeta}^{(k)}, \vec{w}_1) \\ \quad + ((\nabla \vec{v}^{(k)})^\top \delta \vec{\zeta}^{(k)}, \vec{w}_1) - (\delta \mu^{(k)}, \nabla \cdot \vec{w}_1) = \vec{R}_1^{(k)}, \\ \quad \quad \quad - (q_2, \nabla \cdot \delta \vec{\zeta}^{(k)}) = r_2^{(k)}, \\ \nu(\nabla \delta \vec{v}^{(k)}, \nabla \vec{w}_2) + (\vec{v}^{(k)} \cdot \nabla \delta \vec{v}^{(k)}, \vec{w}_2) + (\delta \vec{v}^{(k)} \cdot \nabla \vec{v}^{(k)}, \vec{w}_2) \\ \quad \quad \quad - (\delta p^{(k)}, \nabla \cdot \vec{w}_2) - \frac{1}{\beta} (\delta \vec{\zeta}^{(k)}, \vec{w}_2) = \vec{R}_2^{(k)}, \\ \quad \quad \quad - (q_1, \nabla \cdot \delta \vec{v}^{(k)}) = r_1^{(k)}, \end{array} \right. \quad (2.6)$$

for any  $(\vec{w}_1, \vec{w}_2, q_1, q_2) \in V_0 \times V_0 \times Q \times Q$ .

Next, assuming that an inf-sup stable finite element discretization is applied to (2.6), we can write the (finite-dimensional) inexact Newton step in the form

$$\begin{array}{ll} \mathbf{v}^{(k+1)} = \mathbf{v}^{(k)} + \delta \mathbf{v}^{(k)}, & \mathbf{p}^{(k+1)} = \mathbf{p}^{(k)} + \delta \mathbf{p}^{(k)}, \\ \boldsymbol{\zeta}^{(k+1)} = \boldsymbol{\zeta}^{(k)} + \delta \boldsymbol{\zeta}^{(k)}, & \boldsymbol{\mu}^{(k+1)} = \boldsymbol{\mu}^{(k)} + \delta \boldsymbol{\mu}^{(k)}, \end{array}$$

where the inexact Newton corrections satisfy the following linear algebraic system:

$$\underbrace{\begin{bmatrix} \mathcal{F}_{\text{OD}}^{(k)} & \mathcal{B}^\top \\ \mathcal{B} & \mathcal{O} \end{bmatrix}}_{\mathcal{A}_{\text{OD}}^{(k)}} \begin{bmatrix} \delta \mathbf{v}^{(k)} \\ \delta \boldsymbol{\zeta}^{(k)} \\ \delta \boldsymbol{\mu}^{(k)} \\ \delta \mathbf{p}^{(k)} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_1^{(k)} \\ \mathbf{R}_2^{(k)} \\ \mathbf{r}_1^{(k)} \\ \mathbf{r}_2^{(k)} \end{bmatrix}, \quad (2.7)$$

where, by setting  $\mathbf{M}_\beta = \frac{1}{\beta} \mathbf{M}$  and  $\mathbf{D}_{\text{adj}}^{(k)} = \nu \mathbf{K} - \mathbf{N}^{(k)} + \mathbf{W}^{(k)}$ , we have

$$\mathcal{F}_{\text{OD}}^{(k)} = \begin{bmatrix} \mathbf{M} & \mathbf{D}_{\text{adj}}^{(k)} + (\mathbf{H}^{(k)})^\top \\ \mathbf{D}^{(k)} + \mathbf{H}^{(k)} & -\mathbf{M}_\beta \end{bmatrix}, \quad \mathcal{B} = \begin{bmatrix} B & 0 \\ 0 & B \end{bmatrix}, \quad \mathcal{O} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}. \quad (2.8)$$

The discrete residuals in (2.7) are given by

$$\begin{cases} \mathbf{R}_1^{(k)} = \mathbf{M} \mathbf{v}_d - \mathbf{M} \mathbf{v}^{(k)} - \mathbf{D}_{\text{adj}}^{(k)} \boldsymbol{\zeta}^{(k)} - B^\top \boldsymbol{\mu}^{(k)} - \boldsymbol{\omega}^{(k)}, \\ \mathbf{R}_2^{(k)} = \mathbf{f} - \mathbf{D}^{(k)} \mathbf{v}^{(k)} - B^\top \mathbf{p}^{(k)} + \mathbf{M}_\beta \boldsymbol{\zeta}^{(k)}, \\ \mathbf{r}_1^{(k)} = -B \mathbf{v}^{(k)}, \\ \mathbf{r}_2^{(k)} = -B \boldsymbol{\zeta}^{(k)}, \end{cases} \quad (2.9)$$

where  $\mathbf{v}_d$  is the vector corresponding to the discretized desired state  $\vec{v}_d$ ,  $\mathbf{f}$  is the vector corresponding to the discretized force function  $f$ , and  $\boldsymbol{\omega}^{(k)} = \{((\nabla \vec{v}^{(k)})^\top \vec{\zeta}^{(k)}, \phi_i)\}_{i=1}^{n_v}$ .

**2.2. Discretize-then-Optimize.** We now present the inexact Newton iteration for the discretize-then-optimize approach. We suppose that we have approximations  $\vec{v}^{(k)}$  and  $\vec{\zeta}^{(k)}$  of the state velocity  $\vec{v}$  and of the adjoint velocity  $\vec{\zeta}$ , respectively.

We first discretize the cost functional in (1.1) and the constraints in (1.2). In order to derive a second-order method, we need to employ a Newton discretization of the convection operator. Specifically, we have

$$\min_{\mathbf{v}, \mathbf{u}} \mathbf{J}(\mathbf{v}, \mathbf{u}) = \frac{1}{2} (\mathbf{v} - \mathbf{v}_d)^\top \mathbf{M} (\mathbf{v} - \mathbf{v}_d) + \frac{\beta}{2} \mathbf{u}^\top \mathbf{M} \mathbf{u},$$

subject to

$$\begin{cases} (\mathbf{D}^{(k)} + \mathbf{H}^{(k)}) \mathbf{v} + B^\top \mathbf{p} - \mathbf{M} \mathbf{u} = \mathbf{f}, \\ B \mathbf{v} = \mathbf{0}, \\ \mathbf{v}_b = \mathbf{g}, \end{cases}$$

where  $\mathbf{v}_b$  are the components of the vector  $\mathbf{v}$  related to the boundary nodes. By employing classical constrained optimization theory, one can derive the first-order optimality conditions for the problem above. Specifically, the KKT conditions are given by the following system of equations:

$$\begin{cases} (\mathbf{D}^{(k)} + \mathbf{H}^{(k)}) \mathbf{v} + B^\top \mathbf{p} - \frac{1}{\beta} \mathbf{M} \boldsymbol{\zeta} = \mathbf{f}, \\ B \mathbf{v} = \mathbf{0}, \\ \mathbf{M} \mathbf{v} + (\mathbf{D}^{(k)} + \mathbf{H}^{(k)})^\top \boldsymbol{\zeta} + B^\top \boldsymbol{\mu} = \mathbf{M} \mathbf{v}_d, \\ B \boldsymbol{\zeta} = \mathbf{0}, \end{cases}$$

where we have substituted the gradient equation  $\beta \mathbf{u} = \boldsymbol{\zeta}$  into the state equation.

As a last step, we derive the inexact Newton iteration. This is given by

$$\begin{aligned} \mathbf{v}^{(k+1)} &= \mathbf{v}^{(k)} + \delta \mathbf{v}^{(k)}, & \mathbf{p}^{(k+1)} &= \mathbf{p}^{(k)} + \delta \mathbf{p}^{(k)}, \\ \boldsymbol{\zeta}^{(k+1)} &= \boldsymbol{\zeta}^{(k)} + \delta \boldsymbol{\zeta}^{(k)}, & \boldsymbol{\mu}^{(k+1)} &= \boldsymbol{\mu}^{(k)} + \delta \boldsymbol{\mu}^{(k)}, \end{aligned}$$

where the inexact Newton corrections satisfy the following linear system:

$$\underbrace{\begin{bmatrix} \mathcal{F}_{\text{DO}}^{(k)} & \mathcal{B}^\top \\ \mathcal{B} & \mathcal{O} \end{bmatrix}}_{\mathcal{A}_{\text{DO}}^{(k)}} \begin{bmatrix} \delta \mathbf{v}^{(k)} \\ \delta \boldsymbol{\zeta}^{(k)} \\ \delta \boldsymbol{\mu}^{(k)} \\ \delta \mathbf{p}^{(k)} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_1^{(k)} \\ \mathbf{R}_2^{(k)} \\ \mathbf{r}_1^{(k)} \\ \mathbf{r}_2^{(k)} \end{bmatrix}, \quad (2.10)$$

with the discrete non-linear residual given as in (2.9), the matrices  $\mathcal{B}$  and  $\mathcal{O}$  defined as in (2.8), and

$$\mathcal{F}_{\text{DO}}^{(k)} = \begin{bmatrix} \mathbf{M} & (\mathbf{D}^{(k)} + \mathbf{H}^{(k)})^\top \\ \mathbf{D}^{(k)} + \mathbf{H}^{(k)} & -\mathbf{M}_\beta \end{bmatrix}. \quad (2.11)$$

Note that the linear system given in (2.10) does not correspond to the full Newton iteration, as in the (1,1)-block we do not include second-order information. In fact, introducing the function

$$\mathcal{G}(\mathbf{v}, \boldsymbol{\zeta}) = \int_{\Omega} \left( \sum_{i=1}^{n_v} \mathbf{v}_i \vec{\phi}_i \right) \cdot \left( \sum_{i=1}^{n_v} \mathbf{v}_i \nabla \vec{\phi}_i \right) \cdot \left( \sum_{i=1}^{n_v} \zeta_i \vec{\phi}_i \right) d\Omega,$$

the full Newton system would be given by (2.10) with the (1,1)-block of the matrix  $\mathcal{F}_{\text{DO}}^{(k)}$  given by  $\mathbf{M} + \nabla_{\mathbf{v}\mathbf{v}} \mathcal{G}(\mathbf{v}^{(k)}, \boldsymbol{\zeta}^{(k)})$ . After some manipulations, one can write

$$\nabla_{\mathbf{v}\mathbf{v}} \mathcal{G}(\mathbf{v}^{(k)}, \boldsymbol{\zeta}^{(k)}) = \mathbf{N}_\zeta^{(k)} + \mathbf{H}_\zeta^{(k)},$$

with

$$\mathbf{N}_\zeta^{(k)} = \left\{ \int_{\Omega} (\zeta^{(k)} \cdot \nabla \vec{\phi}_j) \cdot \vec{\phi}_i d\Omega \right\}_{i,j=1}^{n_v}, \quad \mathbf{H}_\zeta^{(k)} = \left\{ \int_{\Omega} (\vec{\phi}_j \cdot \nabla \zeta^{(k)}) \cdot \vec{\phi}_i d\Omega \right\}_{i,j=1}^{n_v}.$$

**3. Preconditioning approach.** In this section, we present the preconditioning strategy that we adopt for solving the sequence of linear systems arising in the non-linear iterations.

We are interested in the solution of linear systems of the form

$$\underbrace{\begin{bmatrix} \Phi & \Psi_1^\top \\ \Psi_2 & -\Theta \end{bmatrix}}_{\mathcal{A}} \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}, \quad (3.1)$$

with invertible  $\Phi$  and a possibly non-zero  $\Theta$ . Systems of this form are referred to as generalized saddle-point problems in [5]. Note that, for the problems we are interested in, both the whole linear system (see (2.7) and (2.10)) and the subsystems associated with its (1,1)-block (see (2.8) and (2.11)) are generalized saddle-point problems. We also observe that the discretize-then-optimize strategy leads to symmetric systems ( $\Phi = \Phi^\top$  and  $\Psi_1 = \Psi_2$ ), while the optimize-then-discretize one does not since, generally speaking,  $\mathbf{D}_{\text{adj}}^{(k)} \neq (\mathbf{D}^{(k)})^\top$ , e.g. due to lack of exact incompressibility of the state velocity approximation.

As is well-known, an “ideal” optimal preconditioner for the system in (3.1) is given by either of the following block-triangular matrices:

$$\mathcal{P}_1 = \begin{bmatrix} \Phi & 0 \\ \Psi_2 & -S \end{bmatrix}, \quad \mathcal{P}_2 = \begin{bmatrix} \Phi & \Psi_1^\top \\ 0 & -S \end{bmatrix}, \quad (3.2)$$

where the (1,1)-block of the preconditioner  $\mathcal{P}_i$ ,  $i = 1, 2$ , is the (1,1)-block of the matrix  $\mathcal{A}$  considered, with the (2,2)-block  $S = \Theta + \Psi_2 \Phi^{-1} \Psi_1^\top$  being the (negative) Schur complement. The optimality of the preconditioners derives from the spectral properties of the preconditioned matrices. Indeed, assuming that the Schur complement  $S$  is invertible, it can be proved that  $\lambda(\mathcal{P}_i^{-1} \mathcal{A}) = \{1\}$ , for  $i = 1, 2$ , with minimal polynomial of degree two; see, for instance, [26, 36]. Therefore, a suitable minimum residual method would converge in at most two iterations, in exact arithmetic.

In practical applications, the linear system in (3.1) can be very large and  $\Phi^{-1}$  full, thus, even forming the Schur complement  $S$  is often unfeasible. For this reason, many of the most effective preconditioners are based on approximations of  $\Phi$  and  $S$ . Specifically, given invertible approximations  $\tilde{\Phi}$  and  $\tilde{S}$  of  $\Phi$  and  $S$ , respectively, rather than considering the ideal preconditioners  $\mathcal{P}_1$  and  $\mathcal{P}_2$ , one instead employs approximations of the form

$$\tilde{\mathcal{P}}_1 = \begin{bmatrix} \tilde{\Phi} & 0 \\ \Psi_2 & -\tilde{S} \end{bmatrix}, \quad \tilde{\mathcal{P}}_2 = \begin{bmatrix} \tilde{\Phi} & \Psi_1^\top \\ 0 & -\tilde{S} \end{bmatrix}. \quad (3.3)$$

Obviously, the approximations  $\tilde{\Phi}$  and  $\tilde{S}$  are problem-dependent. In some cases they are defined implicitly through some inexact solution procedure for linear systems involving the matrices  $\Phi$  and  $S$ .

As mentioned, for incompressible Navier–Stokes control problems the (1,1)-block  $\Phi$  of the linear system arising in the non-linear iteration is itself a (generalized) saddle-point system. Linear systems with a similar “nested” structure arise frequently in the context of PDE-constrained optimization problems, and many researchers have devoted their efforts in devising robust solvers for such linear systems. We refer the interested reader to [1, 24, 45, 47, 51, 53, 59] and the references therein.

The most delicate task in the derivation of the preconditioners (3.3) for the (sequence of) linear system(s) (2.7) and (2.10) is finding an approximation for the Schur complement  $S$ . In [32], the authors have derived a (heuristic) approximation of the Schur complement of the linear systems arising in the Picard iteration for the solution of (stationary and instationary) Navier–Stokes control problems, based on a commutator argument. This strategy has been shown to be very effective and robust for a wide range of parameters and problems, see [15, 30, 32]. However, as we will see below, when applied to a Newton iteration for the Navier–Stokes control problem, the commutator argument presents some limitations. We also mention here the work [58], where the authors derived a rotated block-diagonal preconditioner for the solution of the Picard linearization of the stationary Navier–Stokes control problem, when employing a discretize-then-optimize approach.

In the following sections, we describe the strategies employed for approximating the main blocks of the preconditioners we use. In what follows, we denote with  $I_n \in \mathbb{R}^{n \times n}$  the identity matrix. Further,  $K_p$  and  $M_p$  denote stiffness and mass matrices on the pressure space, respectively.

**3.1. Matching strategy.** We first introduce the approximation of the (1,1)-block of the (whole) linear systems arising from the non-linear iteration. The preconditioner we adopt employs the *matching strategy* [42] for approximating the (inner)

Schur complement of the (1,1)-block, which couples the state and adjoint velocities and is of saddle-point type, as in (3.1). The strategy employed for approximating the outer Schur complement (the one of the whole system, where also the incompressibility constraints are considered) will be discussed below. Although the matching strategy was devised for symmetric indefinite systems, in our derivation we consider a matrix  $\mathcal{A}$  of the form given in (3.1), where the (1,2)-block may not be the transpose of the (2,1)-block. This is because the linear system (2.8) arising from the optimize-then-discretize strategy, as already observed, may not be symmetric. Further, we will assume that the (1,1)-block  $\Phi$  is s.p.d., and that the (2,2)-block is a positive multiple of the (1,1)-block, namely  $\Theta = \frac{1}{\beta}\Phi$ , for some  $\beta > 0$ .

As we discussed above, an ideal preconditioner for the system considered is given by the block-triangular matrix  $\mathcal{P}_1$  given in (3.2). Again, we are interested in finding approximations of  $\Phi$  and  $S$ . For the problems considered here, the matrix  $\Phi$  is a mass matrix. An efficient way of approximating it is given by a fixed number of steps of the Chebyshev semi-iteration [20, 21, 56], preconditioned with a Jacobi splitting.

We now introduce the matching strategy for approximating the inner Schur complement  $S = \frac{1}{\beta}\Phi + \Psi_2\Phi^{-1}\Psi_1^\top$ . We seek an easily invertible approximation that “captures” the two s.p.d. terms in  $S$ , namely,  $\frac{1}{\beta}\Phi$  and  $\Psi_2\Phi^{-1}\Psi_1^\top$ . The approximation we seek is of the form

$$\tilde{S} := (\Psi_2 + \Lambda)\Phi^{-1}(\Psi_1 + \Lambda)^\top \approx S, \quad (3.4)$$

with the matrix  $\Lambda$  such that  $\Lambda\Phi^{-1}\Lambda^\top = \frac{1}{\beta}\Phi$ . It is easy to check that, under our assumptions, this relation holds if we choose

$$\Lambda = \frac{1}{\sqrt{\beta}}\Phi.$$

When the incompressibility constraints are satisfied in the strong sense, the approximation given in (3.4) for the Picard iteration is optimal. In fact, in this case, the system is symmetric, and each non-linear iteration can be considered as the optimality conditions of an incompressible convection–diffusion control problem. From here, it can be proved that  $\lambda(\tilde{S}^{-1}S) \subseteq [\frac{1}{2}, 1]$ ; see, for instance, [43].

Remarkably, the approximation given in (3.4) can also be employed when the incompressibility constraints are satisfied only inexactly. In fact, one can observe the robustness of this approach when employing it as an inner solver for the linear system arising in a Picard iteration for the Navier–Stokes control problem; see [32]. We note that in this case the system is non-symmetric, thus the derivation of bounds for the spectrum of the preconditioned matrix is non-trivial. Similarly, one can apply the matching strategy when solving the linear systems arising from an inexact Newton iteration. However, one cannot prove the optimality of the strategy. In fact, the upper bound of the spectrum of the preconditioned matrix holds if and only if the “mixed term”  $\Psi_2 + \Psi_1^\top$  is s.p.d.; see, for example, [43]. Nonetheless, we expect this approach to result in a robust preconditioner.

**3.2. Block Pressure Convection–Diffusion preconditioner.** We present here the block pressure convection–diffusion preconditioner for the outer Schur complement. This heuristic approach has been derived in [32] for the Navier–Stokes control problem. We would like to mention that the authors in [15] independently derived a similar approach for a parallel-in-time solver for the all-at-once linear system arising from a Picard iteration applied to the forward instationary incompressible Navier–Stokes equations. We follow the derivation in [32].

The block pressure convection–diffusion preconditioner is a generalization of the pressure convection–diffusion preconditioner derived in [29]. Suppose we have a differential operator  $\mathcal{D}$  of the form

$$\mathcal{D} = \begin{bmatrix} \mathcal{D}^{1,1} & \dots & \mathcal{D}^{1,n} \\ \vdots & \ddots & \vdots \\ \mathcal{D}^{n,1} & \dots & \mathcal{D}^{n,n} \end{bmatrix},$$

for some  $n \in \mathbb{N}$ . Here, each  $\mathcal{D}^{i,j}$  is a differential operator on the velocity space, for  $i, j = 1, 2, \dots, n$ . Suppose that for each  $\mathcal{D}^{i,j}$  the corresponding differential operator on the pressure space  $\mathcal{D}_p^{i,j}$  is well defined, for  $i, j = 1, 2, \dots, n$ . Suppose also that the “commutator” operator

$$\mathcal{E}_n = \mathcal{D}\nabla_n - \nabla_n\mathcal{D}_p,$$

is small in some sense, where  $\nabla_n = I_n \otimes \nabla$ . After discretizing with stable finite elements we obtain

$$(\mathcal{M}^{-1}\mathbf{D})\mathcal{M}^{-1}\vec{B}^\top - \mathcal{M}^{-1}\vec{B}^\top(\mathcal{M}_p^{-1}D_p) \approx 0, \quad (3.5)$$

where  $\mathcal{M} = I_n \otimes \mathbf{M}$ ,  $\mathcal{M}_p = I_n \otimes M_p$ ,  $\vec{B} = I_n \otimes B$ , and

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}^{1,1} & \dots & \mathbf{D}^{1,n} \\ \vdots & \ddots & \vdots \\ \mathbf{D}^{n,1} & \dots & \mathbf{D}^{n,n} \end{bmatrix}, \quad D_p = \begin{bmatrix} D_p^{1,1} & \dots & D_p^{1,n} \\ \vdots & \ddots & \vdots \\ D_p^{n,1} & \dots & D_p^{n,n} \end{bmatrix}.$$

Here,  $\mathbf{M}^{-1}\mathbf{D}^{i,j}$  and  $M_p^{-1}D_p^{i,j}$  are the corresponding discretizations of  $\mathcal{D}^{i,j}$  and  $\mathcal{D}_p^{i,j}$ , respectively. Assuming invertibility of  $\mathbf{D}$  and of  $D_p$ , by pre-multiplying (3.5) by  $\vec{B}\mathbf{D}^{-1}\mathcal{M}$  and post-multiplying by  $D_p^{-1}\mathcal{M}_p$  we obtain

$$\vec{B}\mathcal{M}^{-1}\vec{B}^\top D_p^{-1}\mathcal{M}_p \approx \vec{B}\mathbf{D}^{-1}\vec{B}^\top.$$

We note that  $\vec{B}\mathcal{M}^{-1}\vec{B}^\top = I_n \otimes (\mathbf{B}\mathbf{M}^{-1}B^\top)$ . Further, it can be proved that  $K_p \approx \mathbf{B}\mathbf{M}^{-1}B^\top$  for enclosed flow, see [16, pp.176–177]. From here, we can derive the following approximation:

$$\mathcal{K}_p D_p^{-1}\mathcal{M}_p \approx \vec{B}\mathbf{D}^{-1}\vec{B}^\top, \quad (3.6)$$

with  $\mathcal{K}_p = I_n \otimes K_p$ . For the problems we consider in this work, we have  $n = 2$ , and the matrix  $\mathbf{D}$  is the (1,1)-block of the discretized optimality conditions derived in section 2.

The block pressure convection–diffusion preconditioner has been proved to be an efficient and robust approximation of the Schur complement of linear systems arising from incompressible flow problems. In fact, despite the complex structures of the problems considered in [15, 30, 32], the numerical results obtained showed only a mild dependence of the preconditioner with respect to the parameters involved. However, this approximation can be applied only to problems involving the incompressible Stokes equations or to the Picard approximation of problems involving the incompressible Navier–Stokes equations. In fact, it is not possible to define a differential operator on the pressure space corresponding to the differential operator

represented by the matrix in (2.1) containing second-order information of the convection term. To be specific, the term  $\nabla \vec{v}^{(k)}$  is a tensor, while the finite element basis functions on the pressure space are scalar functions, thus, if one replaces the finite element basis functions on the velocity space in (2.1) with the finite element basis functions on the pressure space, one would obtain an operator that is not well defined over the pressure space. For this reason, one should not expect this approach to result in a robust solver when applied to a Newton iteration. In the following section, we introduce an approach that obviates this issue.

**3.3. Augmented Lagrangian preconditioner.** In this section we develop an augmented Lagrangian preconditioner for the outer Schur complement, as an alternative to the block pressure convection–diffusion approach of the previous section. As mentioned above, the most delicate task when deriving a robust preconditioner of the form given in (3.3) is finding a good approximation  $\tilde{S}$  of the Schur complement  $S$ . A powerful approach that circumvents this issue is given by the augmented Lagrangian formulation [18]. An additional benefit is that, when applied to incompressible flow problems, this approach can be regarded as a form of grad-div stabilization, in which an extra term is added to the momentum equation in order to better enforce the incompressibility of the fluid, see [39, 40]. The extra term allows for improved stability and accuracy of some discretizations, and leads to the construction of robust preconditioning techniques

Augmented Lagrangian-based preconditioning has been shown to be very effective for solving linearizations of incompressible fluid flow problems. One such technique was developed in [7] for the solution of the Picard iteration applied to the (forward) incompressible Navier–Stokes equations in 2D, and has been successfully extended to the Newton iteration of the 3D incompressible Navier–Stokes equations [17], to problems with indefinite (1, 1)-block [6, 38], as well as to other problems of saddle-point type. We also mention the recent work [34], where the authors derived an augmented Lagrangian preconditioner for the all-at-once linear system arising upon discretization of the instationary incompressible Navier–Stokes equations, when employing a Picard linearization.

We begin presenting the idea of the augmented Lagrangian preconditioner for general saddle-point systems, giving as an example the linearization of the stationary incompressible Navier–Stokes equations. Then, we specialize the framework to the case of linearizations of the stationary Navier–Stokes control problem considered in this work.

Given the linear system in (3.1), with  $\Theta = 0$  and  $\Psi_1 = \Psi_2 = \Psi$ , the idea of the augmented Lagrangian preconditioner is to consider an equivalent system of the form

$$\underbrace{\begin{bmatrix} \Phi + \gamma \Psi^\top \mathcal{W}^{-1} \Psi & \Psi^\top \\ \Psi & 0 \end{bmatrix}}_{\mathcal{A}_\gamma} \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{b}}_1 \\ \mathbf{b}_2 \end{bmatrix}, \quad (3.7)$$

for suitable  $\gamma > 0$  and  $\mathcal{W}$ , where we set  $\hat{\mathbf{b}}_1 = \mathbf{b}_1 + \gamma \Psi_1^\top \mathcal{W}^{-1} \mathbf{b}_2$ . For example, in the setting of the incompressible Navier–Stokes equations, for which we have  $\Psi = B$ , the matrix  $\mathcal{W}$  is chosen to be the pressure mass matrix  $M_p$  or its diagonal. On the other hand, the choice of  $\gamma$  is more delicate, as we describe below.

In order to solve the system in (3.7), we employ the block-triangular preconditioner  $\mathcal{P}_2$  given in (3.2). Given the structure of the (1, 1)-block of  $\mathcal{A}_\gamma$  in (3.7), with a

small change in notation<sup>1</sup>, we consider the ideal preconditioner

$$\mathcal{P}_\gamma = \begin{bmatrix} \Phi + \gamma\Psi^\top\mathcal{W}^{-1}\Psi & \Psi^\top \\ 0 & -S_\gamma \end{bmatrix}, \quad (3.8)$$

where  $S_\gamma = \Psi(\Phi + \gamma\Psi^\top\mathcal{W}^{-1}\Psi)^{-1}\Psi^\top$ . Assuming that the matrix  $\Psi\Phi^{-1}\Psi^\top$  is invertible and employing the Sherman–Morrison–Woodbury formula, one can show that

$$S_\gamma^{-1} = (\Psi\Phi^{-1}\Psi^\top)^{-1} + \gamma\mathcal{W}^{-1}. \quad (3.9)$$

In fact, in this case the Sherman–Morrison–Woodbury formula writes as

$$(\Phi + \gamma\Psi^\top\mathcal{W}^{-1}\Psi)^{-1} = \Phi^{-1} - \Phi^{-1}\Psi^\top (\gamma^{-1}\mathcal{W} + \Psi\Phi^{-1}\Psi^\top)^{-1} \Psi\Phi^{-1},$$

which, together with the invertibility of  $\Psi\Phi^{-1}\Psi^\top$ , implies that

$$\begin{aligned} S_\gamma &= \Psi\Phi^{-1}\Psi^\top - \Psi\Phi^{-1}\Psi^\top (\gamma^{-1}\mathcal{W} + \Psi\Phi^{-1}\Psi^\top)^{-1} \Psi\Phi^{-1}\Psi^\top \\ &= \Psi\Phi^{-1}\Psi^\top (\gamma^{-1}\mathcal{W} + \Psi\Phi^{-1}\Psi^\top)^{-1} (\gamma^{-1}\mathcal{W} + \Psi\Phi^{-1}\Psi^\top - \Psi\Phi^{-1}\Psi^\top) \\ &= [(\Psi\Phi^{-1}\Psi^\top)^{-1} + \gamma\mathcal{W}^{-1}]^{-1}. \end{aligned}$$

From the derivation above, we understand that [an approximate inverse of  \$S\_\gamma\$  can be obtained given a “rough” approximation of  \$\(\Psi\Phi^{-1}\Psi^\top\)^{-1}\$](#) . However, above all the choice of the parameter  $\gamma$  is important in order to obtain a fast solver. [For example, in the case of saddle-point systems arising from the linearization and discretization of the stationary Navier–Stokes equations, the term  \$\Psi\Phi^{-1}\Psi^\top\$  is replaced by a multiple of the mass matrix  \$M\_p\$  \(or its diagonal\)](#). In particular, the following approximation is employed:

$$\tilde{S}_\gamma^{-1} = \nu M_p^{-1} + \gamma M_p^{-1}, \quad \text{or} \quad \tilde{S}_\gamma = \frac{1}{\nu + \gamma} M_p.$$

In [8], the authors proved that the choice of  $\gamma \sim \nu^{-1}$  results in an  $h$ - and  $\nu$ -independent preconditioner, but numerical experiments show that even smaller values of  $\gamma$  result in a robust solver, see [7].

We now move to the derivation of the preconditioner for the Navier–Stokes control problem. In order to simplify the notations, we consider the linear system (2.7) arising from the discretization of the inexact Newton iteration by employing an optimize-then-discretize approach, noting that the derivation easily generalizes to the discretize-then-optimize one.

As mentioned above, the first step consists of considering an equivalent system of the form

$$\underbrace{\begin{bmatrix} \mathcal{F}_{\text{OD}}^{(k)} + \gamma\mathcal{B}^\top\mathcal{W}^{-1}\mathcal{B} & \mathcal{B}^\top \\ \mathcal{B} & 0 \end{bmatrix}}_{\mathcal{A}^{(k)}} \begin{bmatrix} \delta\mathbf{v}^{(k)} \\ \delta\boldsymbol{\zeta}^{(k)} \\ \delta\boldsymbol{\mu}^{(k)} \\ \delta\mathbf{p}^{(k)} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{R}}_1^{(k)} \\ \hat{\mathbf{R}}_2^{(k)} \\ \mathbf{r}_1^{(k)} \\ \mathbf{r}_2^{(k)} \end{bmatrix},$$

for  $\gamma > 0$  and a suitable matrix  $\mathcal{W}$ . Here, the vectors  $\hat{\mathbf{R}}_1^{(k)}$  and  $\hat{\mathbf{R}}_2^{(k)}$  are given by

$$\begin{bmatrix} \hat{\mathbf{R}}_1^{(k)} \\ \hat{\mathbf{R}}_2^{(k)} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_1^{(k)} \\ \mathbf{R}_2^{(k)} \end{bmatrix} + \gamma\Psi^\top\mathcal{W}^{-1} \begin{bmatrix} \mathbf{r}_1^{(k)} \\ \mathbf{r}_2^{(k)} \end{bmatrix}.$$

<sup>1</sup>Note that the preconditioner  $\mathcal{P}_2$  specializes to  $\mathcal{P}_\gamma$  for the augmented Lagrangian approach.

We employ as matrix  $\mathcal{W}$  the following matrix:

$$\mathcal{W} = \begin{bmatrix} 0 & W \\ W & 0 \end{bmatrix}, \quad (3.10)$$

with  $W$  being the pressure mass matrix  $M_p$  or its diagonal. With this choice of  $\mathcal{W}$ , the augmented  $(1, 1)$ -block becomes

$$\mathcal{F}_{\text{OD}}^{(k)} + \gamma \mathcal{B}^\top \mathcal{W}^{-1} \mathcal{B} = \begin{bmatrix} \mathbf{M} & \mathbf{D}_{\text{adj}}^{(k)} + (\mathbf{H}^{(k)})^\top + \gamma B^\top W^{-1} B \\ \mathbf{D}^{(k)} + \mathbf{H}^{(k)} + \gamma B^\top W^{-1} B & -\mathbf{M}_\beta \end{bmatrix}.$$

We note that this choice of  $\mathcal{W}$  can be regarded as a grad-div stabilization of both the state and adjoint equations. This stabilization can be derived in a formal way starting from the control problem (1.1)–(1.2), adding the grad-div stabilization term in the constraints (1.2), and finally deriving the optimality conditions for this problem, as done in [13].

We end this section by finding a suitable approximation of the Schur complement  $S_\gamma$  in (3.8). By applying the Sherman–Morrison–Woodbury formula, one obtains that the inverse of  $S_\gamma$  can be written as in (3.9). In our case, the matrix  $\mathcal{F}_{\text{OD}}^{(k)}$  depends on the regularization parameter  $\beta$ . In numerical experiments, we observed that the term  $(\mathcal{B}(\mathcal{F}_{\text{OD}}^{(k)})^{-1} \mathcal{B}^\top)^{-1}$  in (3.9) is not negligible for small  $\beta$ . Therefore, in order to obtain a robust solver, we need to find a suitable approximation of it. Thus, in order to improve the robustness of the solver, we consider the following approximation of  $S_\gamma^{-1}$ :

$$\tilde{S}_\gamma^{-1} = \begin{bmatrix} K_p^{-1} & \gamma W^{-1} \\ \gamma W^{-1} & -\frac{1}{\beta} K_p^{-1} \end{bmatrix}. \quad (3.11)$$

We made this choice as, for small  $\beta$ , the matrix  $\mathcal{F}_{\text{OD}}^{(k)}$  is well approximated by a multiple of a block-diagonal mass matrix. Specifically, for  $\beta$  small, we have

$$\mathcal{F}_{\text{OD}}^{(k)} \approx \begin{bmatrix} \mathbf{M} & 0 \\ 0 & -\frac{1}{\beta} \mathbf{M} \end{bmatrix}.$$

Then, using the fact that  $K_p \approx B \mathbf{M}^{-1} B^\top$ , we obtain the proposed approximation. Finally, the observation above gives us a heuristic choice of the parameter  $\gamma$ . In fact, the matrix  $\mathcal{F}_{\text{OD}}^{(k)} + \gamma \Psi^\top \mathcal{W}^{-1} \Psi$  is similar to the following matrix:

$$\underbrace{\begin{bmatrix} \mathbf{M} & \sqrt{\beta}(\mathbf{D}_{\text{adj}}^{(k)} + (\mathbf{H}^{(k)})^\top) \\ \sqrt{\beta}(\mathbf{D}^{(k)} + \mathbf{H}^{(k)}) & -\mathbf{M} \end{bmatrix}}_{\hat{\mathcal{F}}_{\text{OD}}^{(k)}} + \bar{\gamma} \begin{bmatrix} 0 & B^\top W^{-1} B \\ B^\top W^{-1} B & 0 \end{bmatrix},$$

with  $\bar{\gamma} = \sqrt{\beta} \gamma$ . Following a similar argument as in [8], one can expect that for large values of  $\bar{\gamma}$  the augmented Lagrangian preconditioner is able to cluster the majority of the eigenvalues of the preconditioned matrix around 1. This is because the term  $\bar{\gamma} B^\top W^{-1} B$  is becoming dominant with respect to the matrix  $\hat{\mathcal{F}}_{\text{OD}}^{(k)}$ , thus the way one approximates [the inverse of the matrix  \$\Psi \Phi^{-1} \Psi^\top\$  in \(3.9\)](#) becomes less critical for large

$\bar{\gamma}$ . Finally, the previous observation leads to the heuristic choice of  $\gamma > \frac{1}{\sqrt{\beta}}$ . A more formal eigenvalue analysis is beyond the scope of this work, and will be the topic of future research.

Finally, as we mentioned above, practical preconditioners usually require inexact solves for the main blocks. For this reason, the preconditioner we employ in the numerical tests is an approximation  $\tilde{\mathcal{P}}_\gamma$  of the ideal preconditioner  $\mathcal{P}_\gamma$  given in (3.8). Specifically, we approximate the (1,1)-block  $\mathcal{F}_{\text{OD}}^{(k)} + \gamma \mathcal{B}^\top \mathcal{W}^{-1} \mathcal{B}$  with a fixed number of iterations of the FGMRES solver preconditioned by a suitable preconditioner, and employ the matrix  $\tilde{S}_\gamma$  in (3.11) as an approximation of the Schur complement  $S_\gamma$ .

**4. Numerical results.** In this section, we present numerical evidence of the robustness of the augmented Lagrangian approach.

In all our tests, we employ as a preconditioner the block-triangular matrix  $\tilde{\mathcal{P}}_2$  given in (3.3). Note that, as we mentioned above, in the augmented Lagrangian framework the preconditioner  $\tilde{\mathcal{P}}_2$  specializes to  $\tilde{\mathcal{P}}_\gamma$  as described at the end of Section 3.3. The inverse of the (1,1)-block that couples the two velocities is approximated by 5 iterations of preconditioned GMRES [49], employing the block-triangular matrix  $\tilde{\mathcal{P}}_1$  defined in (3.3) as a preconditioner. In the inner solver, we approximate the inverse of the (1,1)-block with 20 steps of Chebyshev semi-iteration, while the Schur complement is approximated by the matrix  $\tilde{S}$  defined in (3.4). Since we need an inner Krylov solver and our systems are non-symmetric, we employ flexible GMRES [48] as outer solver, with restart every 10 iterations, up to a tolerance  $10^{-6}$  on the relative residual. All CPU times below are reported in seconds. We summarise the preconditioner for the augmented Lagrangian method in Figure 4.1. Following our observation at the end of Section 3.3, for a given  $\beta$  we set  $\gamma = 10/\sqrt{\beta}$ .

In all our tests, we report the value of the discrete cost functional  $J_h(\mathbf{v}^*, \mathbf{u}^*)$  at the optimal solution  $(\mathbf{v}^*, \mathbf{u}^*)$  found at the end of the non-linear process. The discrete cost functional is defined as follows:

$$J_h^* := J_h(\mathbf{v}^*, \mathbf{u}^*) = \frac{1}{2}(\mathbf{v}^* - \mathbf{v}_d)^\top \mathbf{M}(\mathbf{v}^* - \mathbf{v}_d) + \frac{\beta}{2}(\mathbf{u}^*)^\top \mathbf{M}\mathbf{u}^*.$$

**4.1. Lid-Driven Cavity.** For the first test we consider here, we set  $d = 2$  (that is,  $\mathbf{x} = (x_1, x_1)$ ) and  $\Omega = (-1, 1)^2$ . The Navier–Stokes control problem is given by (1.1)–(1.2), with  $\vec{f} = \vec{v}_d = \vec{0}$ , and

$$\vec{g} = \begin{cases} [1, 0]^\top & \text{on } \partial\Omega_1 := (-1, 1) \times \{1\}, \\ [0, 0]^\top & \text{on } \partial\Omega \setminus \partial\Omega_1. \end{cases}$$

We employ both the optimize-then-discretize and the discretize-then-optimize strategy, showing that the augmented Lagrangian approach can obtain fast convergence for solving the Navier–Stokes control problem, exhibiting a robust behaviour with respect to problem parameters. In the first setting, the optimize-then-discretize one, we compare the augmented Lagrangian preconditioner with the block pressure convection–diffusion preconditioner; this is done by applying the inverses of the velocity blocks in (3.4) exactly, since for an inexact solver the augmented velocity blocks require a specific geometric multigrid not easily available in MATLAB. Then, in the discretize-then-optimize setting, we give as a proof of concept that the whole augmented Lagrangian preconditioner can be run in an inexact framework, by applying

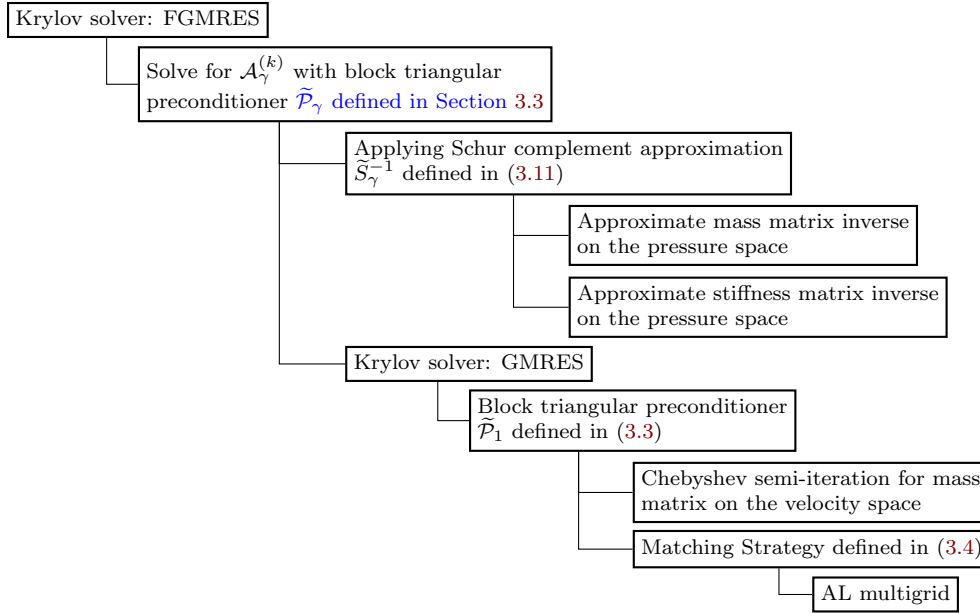


Fig. 4.1: Solver diagram for the augmented Lagrangian preconditioner applied to stationary Navier–Stokes control problems.

a fixed number of cycles of a specific geometric multigrid on the augmented velocity blocks built into Firedrake. The results obtained from the second test allow us to observe the strength of the augmented Lagrangian preconditioner, that allows one to obtain a prescribed reduction on the relative linear residual in very few iterations and, more importantly, independently of the problem parameters even if the preconditioner is applied inexactly.

We would like to note here that for this problem the desired state does not solve the stationary incompressible Navier–Stokes equations problem with the imposed boundary conditions. This is due to a mismatch of the boundary conditions on the “lid” of the square domain. For this reason, we cannot expect the cost functional to be driven to 0 as the mesh is refined.

**4.1.1. Optimize-then-Discretize.** We first compare the augmented Lagrangian strategy with the block pressure convection–diffusion preconditioner, in an optimize-then-discretize framework. Specifically, we are interested in comparing the two Schur complement approximations. The tests presented here are run on MATLAB R2018b, using a 1.70GHz Intel quad-core i5 processor and 8 GB RAM on an Ubuntu 18.04.1 LTS operating system.

The outer solver is based on the flexible GMRES routine in the TT-Toolbox [41], while we employ the GMRES routine implemented in MATLAB as inner solver. The inverse of the stiffness matrix on the pressure space is approximated by 2 V-cycles (with 2 symmetric Gauss–Seidel iterations for pre-/post-smoothing) of the HSL\_MI20 solver [10]. We apply 20 steps of Chebyshev semi-iteration as an approximate inverse of the mass matrix on the pressure space. Since we are interested in understanding the behaviour of the Schur complement approximations, we solve for the approximation

given in (3.4) exactly, that is, the blocks  $\Psi_2 + \Lambda$  and  $(\Psi_1 + \Lambda)^\top$  are solved by MATLAB backslash. We employ the diagonal of the mass matrix on the pressure space as matrix  $W$  in (3.10). Finally, when applying the block pressure convection–diffusion preconditioner, we approximate the Schur complement with the matrix defined in (3.6) (the differential operator on the pressure space does not contain the Newton matrices), while we employ the approximation given in (3.11) for the augmented Lagrangian approach.

Regarding the non-linear iteration, we solve each problem up to a reduction of  $10^{-5}$  on the non-linear relative residual. We allow 10 iterations for inexact Newton. The first non-linear iteration is employed for solving the corresponding Stokes control problem.

We use inf–sup stable Taylor–Hood  $[Q_2]^2$ – $Q_1$  finite elements in the spatial dimensions, with level of refinement  $l$  representing a (spatial) uniform grid of mesh-size  $h = 2^{1-l}$  for  $Q_1$  basis functions, and  $h = 2^{-l}$  for  $Q_2$  elements, in each dimension. For this test case, we employ the LPS approach for stabilizing the linearization of the Navier–Stokes control problem at each Newton iteration.

Table 4.1: Average GMRES iterations with the block pressure convection–diffusion preconditioner, for  $\nu = \frac{1}{100}$ ,  $\frac{1}{250}$ , and  $\frac{1}{500}$ , and a range of  $l$ ,  $\beta$ .

$l$	$\nu = \frac{1}{100}$					$\nu = \frac{1}{250}$					$\nu = \frac{1}{500}$				
	$\beta$					$\beta$					$\beta$				
	$10^{-1}$	$10^{-2}$	$10^{-3}$	$10^{-4}$	$10^{-5}$	$10^{-1}$	$10^{-2}$	$10^{-3}$	$10^{-4}$	$10^{-5}$	$10^{-1}$	$10^{-2}$	$10^{-3}$	$10^{-4}$	$10^{-5}$
3	30	16	11	9	7	33	15	10	9	7	31	15	10	9	7
4	36	21	13	10	8	64	22	12	10	8	75	22	12	10	8
5	39	21	16	12	9	75	31	16	11	9	150	35	16	11	9
6	41	22	17	14	11	73	29	19	14	10	131	41	22	13	10
7	38	25	19	16	13	64	31	21	16	13	140† <sup>2</sup>	24	25	17	13

In Table 4.1, we report the average number of flexible GMRES (FGMRES) iterations (rounded to the nearest integer) required for solving the inexact Newton iteration, when employing the block pressure convection–diffusion preconditioner for approximating the Schur complement. Further, in Tables 4.2–4.4 we report the average number of FGMRES iterations required for solving the corresponding problem, when employing the augmented Lagrangian preconditioner, for our heuristic choice of  $\gamma$ , together with the value of the discrete cost functional  $J_h^*$ . We report the value of the cost functional  $J_h^*$  only for the augmented Lagrangian approach, as this coincides up to the second significant digit with the one obtained by applying the block pressure convection–diffusion preconditioner. Finally, in Table 4.5 we report the total number of inexact Newton iterations required for obtaining convergence (when applying the augmented Lagrangian preconditioner), together with the total degrees of freedom (DoF) of the system solved at each non-linear iteration. Again, all the linear solves involved in the application of the two preconditioners are performed exactly by backslash.

From Table 4.1, we observe the  $h$ -robustness of the block pressure convection–diffusion preconditioner for small values of  $\beta$ . For larger values of  $\beta$ , however, the preconditioner is strongly dependent on the viscosity  $\nu$  and lacks robustness. In fact,

<sup>2</sup>† means that the outer (inexact Newton) iteration did not converge in 10 iterations. The average number of FGMRES iterations is evaluated over the first 5 inexact Newton iterations.

the linear solver is not able to converge for small values of the viscosity and larger  $\beta$ , resulting in a non-convergent inexact Newton method. On the other hand, as observed from Tables 4.2–4.4 the augmented Lagrangian preconditioner is able to reach convergence in an average number of linear iterations that is roughly constant with respect to all the parameters involved in the problems, especially for sufficiently fine grids. In particular, we observe that the augmented Lagrangian preconditioner requires at most 9 iterations in average for reaching the prescribed tolerance on the relative residual. The efficiency of the approach is not lost even if one drives the regularization parameter towards 0, converging in at most 5 iterations for  $\beta = 10^{-6}$  and  $10^{-8}$ . Further, from Tables 4.2–4.4 we observe that the discrete cost functional  $J_h^*$  is decreasing as the mesh size  $h$  goes to 0. Finally, from Table 4.5 we note that the number of inexact Newton iterations required for reaching a prescribed non-linear tolerance is roughly constant with respect to the regularization parameter and the viscosity, for fine grids, with a small dependence on the viscosity for coarse grids.

Table 4.2: Average GMRES iterations and value of the discrete cost functional  $J_h^*$  with the augmented Lagrangian preconditioner with  $\gamma = 10/\sqrt{\beta}$ , for  $\nu = \frac{1}{100}$  and a range of  $l, \beta$ .

$l$	$\beta$													
	$10^{-1}$		$10^{-2}$		$10^{-3}$		$10^{-4}$		$10^{-5}$		$10^{-6}$		$10^{-8}$	
	it	$J_h^*$	it	$J_h^*$	it	$J_h^*$	it	$J_h^*$	it	$J_h^*$	it	$J_h^*$	it	$J_h^*$
3	6	4.2e-02	5	3.3e-02	4	3.1e-02	3	3.1e-02	3	3.0e-02	3	3.0e-02	2	3.0e-02
4	7	3.7e-02	6	2.3e-02	5	1.7e-02	4	1.5e-02	4	1.5e-02	3	1.5e-02	3	1.5e-02
5	7	3.6e-02	5	2.1e-02	5	1.3e-02	6	8.8e-03	4	7.7e-03	4	7.5e-03	3	7.5e-03
6	6	3.6e-02	6	2.1e-02	5	1.2e-02	8	7.0e-03	5	4.6e-03	5	3.9e-03	3	3.7e-03
7	6	3.6e-02	5	2.1e-02	5	1.2e-02	6	6.9e-03	7	3.9e-03	5	2.5e-03	4	1.9e-03

Table 4.3: Average GMRES iterations and value of the discrete cost functional  $J_h^*$  with the augmented Lagrangian preconditioner with  $\gamma = 10/\sqrt{\beta}$ , for  $\nu = \frac{1}{250}$  and a range of  $l, \beta$ .

$l$	$\beta$													
	$10^{-1}$		$10^{-2}$		$10^{-3}$		$10^{-4}$		$10^{-5}$		$10^{-6}$		$10^{-8}$	
	it	$J_h^*$	it	$J_h^*$	it	$J_h^*$	it	$J_h^*$	it	$J_h^*$	it	$J_h^*$	it	$J_h^*$
3	6	3.5e-02	5	3.1e-02	4	3.1e-02	4	3.0e-02	3	3.0e-02	3	3.0e-02	2	3.0e-02
4	7	2.4e-02	6	1.8e-02	4	1.5e-02	4	1.5e-02	3	1.5e-02	3	1.5e-02	3	1.5e-02
5	7	2.4e-02	7	1.4e-02	5	9.3e-03	5	7.8e-03	3	7.5e-03	3	7.5e-03	3	7.5e-03
6	6	2.4e-02	6	1.4e-02	6	7.8e-03	5	4.9e-03	5	3.9e-03	4	3.7e-03	4	3.7e-03
7	7	2.4e-02	6	1.4e-02	6	7.8e-03	6	4.4e-03	5	2.7e-03	5	2.0e-03	4	1.9e-03

**4.1.2. Discretize-then-Optimize.** In this section, we report the results of our augmented Lagrangian preconditioner when employing an inexact solver on the augmented block, when considering a discretize-then-optimize approach. We employ the Firedrake system [46] for the construction of the finite element discretizations of the inexact Newton iteration and of the linear and non-linear solvers employed. We employ the exactly incompressible Scott–Vogelius  $[P_4]^2$ – $DP_3$  finite element pair [50]. We mention that for this problem no stabilization for the linearized convection term is employed in the discretization. Since we employ discontinuous finite elements on

Table 4.4: Average GMRES iterations and value of the discrete cost functional  $J_h^*$  with the augmented Lagrangian preconditioner with  $\gamma = 10/\sqrt{\beta}$ , for  $\nu = \frac{1}{500}$  and a range of  $l, \beta$ .

$l$	$\beta$													
	$10^{-1}$		$10^{-2}$		$10^{-3}$		$10^{-4}$		$10^{-5}$		$10^{-6}$		$10^{-8}$	
	it	$J_h^*$	it	$J_h^*$	it	$J_h^*$	it	$J_h^*$	it	$J_h^*$	it	$J_h^*$	it	$J_h^*$
3	7	3.2e-02	5	3.1e-02	4	3.0e-02	4	3.0e-02	3	3.0e-02	3	3.0e-02	2	3.0e-02
4	9	1.9e-02	6	1.6e-02	4	1.5e-02	4	1.5e-02	3	1.5e-02	3	1.5e-02	3	1.5e-02
5	8	1.7e-02	6	1.0e-02	5	8.1e-03	5	7.6e-03	3	7.5e-03	3	7.5e-03	3	7.5e-03
6	8	1.7e-02	6	9.7e-03	5	5.7e-03	5	4.2e-03	5	3.8e-03	4	3.7e-03	4	3.7e-03
7	9	1.7e-02	6	9.8e-03	6	5.5e-03	7	3.2e-03	5	2.2e-03	5	1.9e-03	4	1.9e-03

Table 4.5: Degrees of freedom (DoF) and number of inexact Newton iterations required for stationary Navier–Stokes control problem. In each cell are the inexact Newton iterations for the given  $l, \nu$ , and  $\beta = 10^{-j}$ ,  $j = 1, \dots, 5$ .

$l$	DoF	$\nu = \frac{1}{100}$					$\nu = \frac{1}{250}$					$\nu = \frac{1}{500}$				
3	1062	5	4	3	3	3	7	5	4	3	3	8	5	4	3	3
4	4422	4	4	3	3	3	6	4	3	3	3	8	5	5	4	3
5	18,054	4	3	3	3	3	4	4	3	3	3	6	4	3	3	3
6	72,966	4	3	3	4	3	4	3	3	3	3	4	3	3	3	3
7	293,382	3	3	3	3	3	3	3	3	3	3	4	3	3	3	3

the pressure space, the mass matrix is block-diagonal and is solved cellwise. The stiffness matrix is solved with a two-level  $p$ -multigrid solver applied to an auxiliary-space  $P_3$  discretization, i.e.  $DP_3 \rightarrow P_3 \rightarrow P_1$ , with Chebyshev-accelerated symmetric Gauss–Seidel pointwise relaxation on the degree-3 levels and a geometric multigrid cycle applied on the  $P_1$  problem. Finally, the momentum blocks are inexactly solved with one cycle of the multigrid derived in [7] and implemented in Firedrake in [17]. Ten iterations of FGMRES preconditioned by an additive vertex-star solve are employed as relaxation on each fine level, while the coarse problem is solved with SuperLU\_DIST [33]. The vertex-star solve is performed using PETSc’s PCASM [2, 52]. For the coarsest level ( $l = 1$ ), multigrid has been replaced by an exact solver based on the LU factorization of each block. We employ a Intel Core Ultra 7 165U processor and 64 GB RAM on an Ubuntu 22.04.5 LTS operating system.

We allow for 10 inexact Newton iterations, specifying as a stopping criterion a reduction of  $10^{-5}$  on either the relative or the absolute non-linear residual. We run our preconditioned iterative solver up to a tolerance of  $10^{-6}$  on either the relative or the absolute residual is achieved. We report in Tables 4.6–4.7 the average number of FGMRES iterations required for reaching convergence, together with the value of the cost functional  $J_h^*$ . Further, we report in Tables 4.9–4.11 the total CPU times required for the solution of the problem, with the dimension of the systems solved showed and of the average block-smoother per multigrid level in Table 4.9. Note that the CPU times reported include the evaluation of the non-linear residual, the discretization of the operators, and the construction of the geometric multigrid on the augmented velocity blocks, in addition to the times required for the linear solvers. Further, note that  $l = 1$  represents the coarsest mesh, thus no smoother has been employed.

From Tables 4.6–4.7, we can observe again the overall robustness of the augmented

Lagrangian approach, even when employing inexact solvers throughout. Only for larger values of  $\beta$  on the finest grid do we see some growth in the number of FGMRES iterations; nevertheless, even in this setting the solver requires at most 15 iterations on average to reach convergence. Further, we observe that the discrete cost functional  $J_h^*$  tends to a minimum as the mesh is refined. Finally, from Tables 4.9–4.11 we observe that for the finest meshes the CPU time scales linearly with respect to the problem size, while for coarsest meshes we observe the small problem scaling effect.

Table 4.6: Average FGMRES iterations and value of the discrete cost functional  $J_h^*$  with the augmented Lagrangian preconditioner with  $\gamma = 10/\sqrt{\beta}$ , for  $\nu = \frac{1}{100}$ , and a range of  $l, \beta$ .

$l$	$\beta$													
	$10^{-1}$		$10^{-2}$		$10^{-3}$		$10^{-4}$		$10^{-5}$		$10^{-6}$		$10^{-8}$	
	it	$J_h^*$	it	$J_h^*$	it	$J_h^*$	it	$J_h^*$	it	$J_h^*$	it	$J_h^*$	it	$J_h^*$
1	3	4.5e-02	3	4.3e-02	3	4.3e-02	3	4.3e-02	3	4.3e-02	3	4.3e-02	2	4.2e-02
2	3	3.2e-02	3	2.3e-02	3	2.1e-02	2	2.1e-02	3	2.1e-02	3	2.1e-02	2	2.1e-02
3	3	3.4e-02	3	2.0e-02	3	1.2e-02	4	1.1e-02	3	0.1e-02	3	1.0e-02	2	1.0e-02
4	5	3.5e-02	4	2.1e-02	3	1.1e-02	3	6.8e-03	4	5.4e-03	3	5.2e-03	2	5.1e-03
5	6	3.6e-02	7	2.1e-02	3	1.2e-02	3	6.8e-03	3	3.8e-03	3	2.8e-03	3	2.6e-03
6	9	3.6e-02	5	2.1e-02	3	1.2e-02	3	6.9e-03	3	3.9e-03	3	2.1e-03	4	1.3e-03

Table 4.7: Average FGMRES iterations and value of the discrete cost functional  $J_h^*$  with the augmented Lagrangian preconditioner with  $\gamma = 10/\sqrt{\beta}$ , for  $\nu = \frac{1}{250}$ , and a range of  $l, \beta$ .

$l$	$\beta$													
	$10^{-1}$		$10^{-2}$		$10^{-3}$		$10^{-4}$		$10^{-5}$		$10^{-6}$		$10^{-8}$	
	it	$J_h^*$	it	$J_h^*$	it	$J_h^*$	it	$J_h^*$	it	$J_h^*$	it	$J_h^*$	it	$J_h^*$
1	3	4.4e-02	3	4.3e-02	3	4.3e-02	3	4.3e-02	3	4.3e-02	3	4.3e-02	2	4.2e-02
2	3	2.4e-02	3	2.2e-02	3	2.1e-02	3	2.1e-02	3	2.1e-02	3	2.1e-02	2	2.1e-02
3	4	2.2e-02	3	1.3e-02	4	1.1e-02	3	1.0e-02	3	1.0e-02	2	1.0e-02	2	1.0e-02
4	4	2.3e-02	4	1.3e-02	3	7.5e-03	3	5.5e-03	4	5.2e-03	4	5.1e-03	1	5.0e-03
5	6	2.3e-02	5	1.3e-02	3	7.6e-03	3	4.2e-03	3	2.9e-03	4	2.6e-03	4	2.6e-03
6	13	2.3e-02	8	1.4e-02	3	7.7e-03	3	4.4e-03	3	2.3e-03	4	1.5e-03	4	1.3e-03

Table 4.8: Average FGMRES iterations and value of the discrete cost functional  $J_h^*$  with the augmented Lagrangian preconditioner with  $\gamma = 10/\sqrt{\beta}$ , for  $\nu = \frac{1}{500}$ , and a range of  $l, \beta$ .

$l$	$\beta$													
	$10^{-1}$		$10^{-2}$		$10^{-3}$		$10^{-4}$		$10^{-5}$		$10^{-6}$		$10^{-8}$	
	it	$J_h^*$	it	$J_h^*$	it	$J_h^*$	it	$J_h^*$	it	$J_h^*$	it	$J_h^*$	it	$J_h^*$
1	3	4.3e-02	3	4.3e-02	3	4.3e-02	3	4.3e-02	3	4.3e-02	3	4.3e-02	2	4.2e-02
2	3	2.2e-02	3	2.1e-02	2	2.1e-02	3	2.1e-02	3	2.1e-02	3	2.1e-02	2	2.1e-02
3	4	1.6e-02	4	1.1e-02	4	1.0e-02	3	1.0e-02	3	1.0e-02	2	1.0e-02	2	1.0e-02
4	4	1.6e-02	3	9.2e-03	3	6.0e-03	4	5.2e-03	4	5.2e-03	4	5.1e-03	4	5.1e-03
5	8	1.7e-02	5	9.6e-03	3	5.3e-03	3	3.2e-03	4	2.7e-03	4	2.6e-03	2	2.5e-03
6	15	1.7e-02	7	9.7e-03	3	5.5e-03	3	3.1e-03	3	1.7e-03	4	1.4e-03	4	1.3e-03

Table 4.9: Degrees of freedom (DoF), average size of the block-smoothers (BlkSm), and total CPU times, for the augmented Lagrangian preconditioner with  $\gamma = 10/\sqrt{\beta}$ , for  $\nu = \frac{1}{100}$  and a range of  $l, \beta$ .

$l$	DoF	BlkSm	$\beta$						
			$10^{-1}$	$10^{-2}$	$10^{-3}$	$10^{-4}$	$10^{-5}$	$10^{-6}$	$10^{-8}$
1	484	\	16.7	15.6	16.1	13.4	12.8	12.5	12.3
2	1796	61	25.2	26.1	26.6	21.1	19.6	22.1	19.4
3	6916	67	34.8	27.7	20.5	21.6	21.8	22.5	19.9
4	27,140	70	35.7	33.9	24.4	24.6	25.6	24.3	23.4
5	107,524	72	61.5	67.9	33.1	32.8	33.5	33.9	32.5
6	428,036	73	218	136	61.6	61.9	63.3	62.8	73.1

Table 4.10: Total CPU times, for the augmented Lagrangian preconditioner with  $\gamma = 10/\sqrt{\beta}$ , for  $\nu = \frac{1}{250}$  and a range of  $l, \beta$ .

$l$	$\beta$						
	$10^{-1}$	$10^{-2}$	$10^{-3}$	$10^{-4}$	$10^{-5}$	$10^{-6}$	$10^{-8}$
1	16.2	16.5	16.4	12.9	12.9	13.2	12.9
2	30.6	26.8	26.6	19.6	20.5	18.9	19.7
3	28.5	21.5	23.4	21.2	21.7	20.8	21.3
4	47.1	35.1	24.1	25.1	26.3	25.5	22.2
5	65.8	56.8	32.4	33.3	34.3	36.1	36.8
6	294	195	61.6	62.7	61.9	73.5	73.6

Table 4.11: Total CPU times, for the augmented Lagrangian preconditioner with  $\gamma = 10/\sqrt{\beta}$ , for  $\nu = \frac{1}{500}$  and a range of  $l, \beta$ .

$l$	$\beta$						
	$10^{-1}$	$10^{-2}$	$10^{-3}$	$10^{-4}$	$10^{-5}$	$10^{-6}$	$10^{-8}$
1	15.9	16.3	16.7	12.7	12.8	12.9	12.7
2	26.4	28.1	26.9	19.4	19.8	19.7	19.9
3	30.4	30.7	22.7	21.9	21.3	21.5	22.4
4	45.5	25.4	26.1	26.3	26.4	26.2	25.1
5	76.2	57.9	33.5	33.7	38.3	36.5	31.3
6	338	185	62.3	64.7	63.4	74.2	74.3

**4.2. Backward-Facing Step.** As last test, we solve the Navier–Stokes control problem for the backward-facing step. We apply a Poiseuille inflow on the left end and “no-slip” (zero boundary) conditions on the top and bottom of the step. Natural boundary conditions are imposed on the right end of the step. We set  $\vec{f} = \vec{0}$ , and seek as desired state  $\vec{v}_d$  the solution of the stationary Stokes problem obtained when imposing this forcing and boundary conditions.

As in Section 4.1.2, we apply the discretize-then-optimize strategy, employing the Firedrake system for the derivation of the optimality conditions of the linearized problem and for the construction of the linear and non-linear solvers. Further, as above we employ the Scott–Vogelius  $[P_4]^2$ – $DP_3$  finite element pair. Finally, we allow for 10 inexact Newton iterations, specifying as a stopping criterion a reduction of  $10^{-5}$

on either the relative or the absolute non-linear residual, running the linear solver up to a tolerance  $10^{-6}$  on either the relative or the absolute linear residual. For this test, we set  $\gamma = 10^3$  and  $\gamma = 10^4$ . We opt for this choice of  $\gamma$  as the step problem is more difficult to solve, due to the singularity of the pressure in the corner of the step. We report in Table 4.12 the results for  $\gamma = 10^3$  and in Table 4.13 the results for  $\gamma = 10^4$ , respectively. Further, in Tables 4.14–4.15 we report the total CPU times required for solving the problem, with the dimension of each linear system solved and of the average block-smoother per multigrid level reported in Table 4.14.

Table 4.12: Inexact Newton iteration: average FGMRES iterations and value of the discrete cost functional  $J_h^*$  with the augmented Lagrangian preconditioner with  $\gamma = 10^3$ , for  $\nu = \frac{1}{100}$ ,  $\frac{1}{250}$ , and  $\frac{1}{500}$ , and a range of  $l$ ,  $\beta$ .

$\nu$	$l$	$\beta$													
		$10^{-1}$		$10^{-2}$		$10^{-3}$		$10^{-4}$		$10^{-5}$		$10^{-6}$		$10^{-8}$	
		it	$J_h^*$	it	$J_h^*$	it	$J_h^*$	it	$J_h^*$	it	$J_h^*$	it	$J_h^*$	it	$J_h^*$
$\frac{1}{100}$	1	3	1.3e00	3	1.3e00	3	2.6e00	3	1.2e00	5	6.9e-01	5	5.3e-01	13	3.0e-01
	2	4	1.4e00	3	1.3e00	5	1.6e00	4	1.4e00	8	8.4e-01	20	1.6e00	12	3.2e-01
	3	3	1.6e00	3	1.5e00	3	2.5e00	3	1.2e00	6	7.1e-01	11	6.0e-01	12	2.7e-01
$\frac{1}{250}$	1	3	1.3e00	3	1.3e00	3	2.6e00	3	1.2e00	5	6.9e-01	5	5.3e-01	13	3.0e-01
	2	4	1.4e00	3	1.4e00	5	1.6e00	10	1.4e00	16	1.9e00	22	2.2e00	12	3.2e-01
	3	3	1.7e00	3	1.5e00	4	2.3e00	5	1.8e00	9	9.6e-01	20	8.9e-01	12	3.2e-01
$\frac{1}{500}$	1	3	1.3e00	3	1.3e00	3	2.6e00	3	1.2e00	5	6.9e-01	5	5.3e-01	13	3.0e-01
	2	3	1.7e00	4	1.3e00	6	1.4e00	9	1.5e00	17	1.7e00	17	6.1e-01	12	3.1e-01
	3	3	1.7e00	3	1.5e00	5	1.8e00	8	1.8e00	14	1.1e00	22	2.4e00	11	3.2e-01

Table 4.13: Inexact Newton iteration: average FGMRES iterations and value of the discrete cost functional  $J_h^*$  with the augmented Lagrangian preconditioner with  $\gamma = 10^4$ , for  $\nu = \frac{1}{100}$ ,  $\frac{1}{250}$ , and  $\frac{1}{500}$ , and a range of  $l$ ,  $\beta$ .

$\nu$	$l$	$\beta$													
		$10^{-1}$		$10^{-2}$		$10^{-3}$		$10^{-4}$		$10^{-5}$		$10^{-6}$		$10^{-8}$	
		it	$J_h^*$	it	$J_h^*$	it	$J_h^*$	it	$J_h^*$	it	$J_h^*$	it	$J_h^*$	it	$J_h^*$
$\frac{1}{100}$	1	2	3.5e00	3	1.3e00	2	3.3e00	2	3.2e00	3	2.5e00	2	1.3e00	4	5.3e-01
	2	2	2.4e00	2	2.4e00	3	1.7e00	4	1.5e00	7	1.2e00	10	1.4e00	22	3.4e00
	3	3	1.8e00	3	1.7e00	3	1.7e00	3	1.3e00	5	1.7e00	10	1.3e00	22	3.1e00
$\frac{1}{250}$	1	2	3.4e00	3	1.3e00	2	3.3e00	2	3.2e00	3	2.5e00	2	1.3e00	4	5.2e-01
	2	2	2.3e00	2	2.3e00	4	1.3e00	4	1.3e00	8	1.3e00	10	1.4e00	21	2.7e00
	3	2	2.4e00	3	1.7e00	3	1.3e00	4	1.5e00	7	1.3e00	14	1.3e00	16	6.1e-01
$\frac{1}{500}$	1	2	3.4e00	3	1.3e00	2	3.3e00	2	3.2e00	3	2.5e00	2	1.3e00	4	5.2e-01
	2	3	1.7e00	3	1.7e00	3	1.7e00	6	1.3e00	7	1.3e00	7	2.1e00	22	1.9e00
	3	3	1.7e00	3	1.8e00	3	1.7e00	5	1.3e00	7	1.2e00	10	1.4e00	19	1.9e00

From Tables 4.12–4.13, we observe satisfactory robustness of the solver for the larger values of  $\beta$ , with a small increase in the number of iterations for  $\beta = 10^{-6}$  and  $\beta = 10^{-8}$ . Nonetheless, the solver is able to obtain convergence in at most 22 iterations on average. Further, from Tables 4.14–4.15 we observe that the CPU times scale nearly linearly with respect to the problem size.

**5. Conclusions.** In this work, an augmented Lagrangian preconditioner has been proposed for use within an inexact Newton iteration applied to the control of

Table 4.14: Inexact Newton iteration: degrees of freedom (DoF), average size of the block-smoothers (BlkSm), and total CPU times of the augmented Lagrangian preconditioner with  $\gamma = 10^3$ , for  $\nu = \frac{1}{100}$ ,  $\frac{1}{250}$ , and  $\frac{1}{500}$ , and a range of  $l$ ,  $\beta$ .

$\nu$	$l$	DoF	BlkSm	$\beta$						
				$10^{-1}$	$10^{-2}$	$10^{-3}$	$10^{-4}$	$10^{-5}$	$10^{-6}$	$10^{-8}$
$\frac{1}{100}$	1	168,184	\	11.5	3.8	3.7	4.0	4.5	4.7	7.1
	2	669,476	72	81.0	58.9	81.1	70.2	116	258	164
	3	2,671,396	73	364	358	360	358	555	884	951
$\frac{1}{250}$	1	168,184	\	3.6	3.7	3.7	4.0	4.6	4.8	7.2
	2	669,476	72	69.5	58.5	81.7	140	212	283	165
	3	2,671,396	73	360	359	426	492	757	1481	948
$\frac{1}{500}$	1	168,184	\	3.5	3.8	3.8	3.9	4.8	4.6	7.2
	2	669,476	72	58.8	70.7	93.6	129	224	224	166
	3	2,671,396	73	361	359	491	684	1080	1604	884

Table 4.15: Inexact Newton iteration: total CPU times of the augmented Lagrangian preconditioner with  $\gamma = 10^4$ , for  $\nu = \frac{1}{100}$ ,  $\frac{1}{250}$ , and  $\frac{1}{500}$ , and a range of  $l$ ,  $\beta$ .

$\nu$	$l$	$\beta$						
		$10^{-1}$	$10^{-2}$	$10^{-3}$	$10^{-4}$	$10^{-5}$	$10^{-6}$	$10^{-8}$
$\frac{1}{100}$	1	11.1	3.6	3.3	3.4	3.8	3.5	4.2
	2	54.1	46.4	58.4	70.5	106	141	282
	3	361	358	359	359	488	820	1616
$\frac{1}{250}$	1	3.3	3.5	3.5	3.5	3.9	3.6	4.3
	2	46.8	46.9	70.4	70.9	118	141	272
	3	295	360	362	430	629	1090	1219
$\frac{1}{500}$	1	3.3	3.6	3.4	3.6	3.8	3.6	4.2
	2	59.7	59	59	94.1	106	106	284
	3	362	363	362	493	625	825	1416

the stationary Navier–Stokes equations. We compared the proposed preconditioner with a block pressure–convection–diffusion preconditioner, which was derived for the Picard linearization of the Navier–Stokes control problem. Numerical results showed the good robustness of the augmented Lagrangian approach with respect to the mesh size, the regularization parameter, and the viscosity of the fluid, when employing an exact solver for the momentum blocks. The augmented Lagrangian preconditioner has been shown to be more robust than the (exact) block pressure convection–diffusion preconditioner when solving an inexact Newton linearization of the control problem considered here. Most importantly, the robustness of the approach is preserved even if the momentum block is solved inexactly, through the action of a suitable multigrid approach. Future research will focus on an eigenvalue analysis of the proposed strategy, its extension to the control of time-dependent Navier–Stokes equations, and the application of the augmented Lagrangian preconditioner to more complicated PDE-constrained optimization problems.

**Acknowledgements.** The authors gratefully acknowledge two anonymous referees for their valuable comments. SL and PEF acknowledge Pablo Brubeck Martinez for useful discussions. PEF was funded by the Engineering and Physical Sciences Research Council [grant numbers EP/R029423/1 and EP/W026163/1]. SL and MB are members of Gruppo Nazionale di Calcolo Scientifico (GNCS) of the Istituto Nazionale

## REFERENCES

- [1] O. Axelsson, S. Farouq, and M. Neytcheva, *A preconditioner for optimal control problems, constrained by Stokes equation with a time-harmonic control*, J. Comp. Appl. Math., 310, pp. 5–18, 2017.
- [2] S. Balay et al., *PETSc/TAO Users Manual*, Tech. Report ANL-21/39 - Revision 3.21, Argonne National Laboratory, 2024.
- [3] R. Becker and M. Braack, *A finite element pressure gradient stabilization for the Stokes equations based on local projections*, Calcolo, 38, pp. 173–199, 2001.
- [4] R. Becker and B. Vexler, *Optimal control of the convection–diffusion equation using stabilized finite element methods*, Numer. Math., 106, pp. 349–367, 2007.
- [5] M. Benzi, G. H. Golub, and J. Liesen, *Numerical solution of saddle point problems*, Acta Numerica, 14, pp. 1–137, 2005.
- [6] M. Benzi and J. Liu, *Block preconditioning for saddle point systems with indefinite (1, 1) block*, Intern. J. Comput. Math., 84, pp. 1117–1129, 2007.
- [7] M. Benzi and M. A. Olshanskii, *An augmented Lagrangian-based approach to the Oseen problem*, SIAM J. Sci. Comput., 28, pp. 2095–2113, 2006.
- [8] M. Benzi and M. A. Olshanskii, *Field-of-values convergence analysis of augmented Lagrangian preconditioners for the linearized Navier–Stokes problem*, SIAM J. Numer. Anal., 49, pp. 770–788, 2011.
- [9] M. Benzi, E. Haber, and L. Taralli, *A preconditioning technique for a class of PDE-constrained optimization problems*, Adv. Comput. Math., 35, pp. 149–173, 2011.
- [10] J. Boyle, M. Mihajlović, and J. Scott, *HSLMI20: an efficient AMG preconditioner for finite element problems in 3D*, Int. J. Numer. Meth. Eng., 82, pp. 64–98, 2010.
- [11] M. Braack and E. Burman, *Local projection stabilization for the Oseen problem and its interpretation as a variational multiscale method*, SIAM J. Numer. Anal., 43, pp. 2544–2566, 2006.
- [12] A. N. Brooks and T. J. R. Hughes, *Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations*, Comput. Methods Appl. Mech. Eng., 32, pp. 199–259, 1982.
- [13] A. Çibik, *The effect of a sparse grad–div stabilization on control of stationary Navier–Stokes equations*, J. Math. Anal. Appl., 437, 613–628, 2016.
- [14] 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, Dept. of Computational and Applied Math., Rice University, Houston, 2002.
- [15] F. Danieli, B. S. Southworth, and A. J. Wathen, *Space-time block preconditioning for incompressible flow*, SIAM J. Sci. Comput., 44, pp. A337–A363, 2022.
- [16] H. C. Elman, D. J. Silvester, and A. J. Wathen, *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics*, Oxford University Press, 2nd Edition, 2014.
- [17] P. E. Farrell, L. Mitchell, and F. Wechsung, *An augmented Lagrangian preconditioner for the 3D stationary incompressible Navier–Stokes equations at high Reynolds numbers*, SIAM J. Sci. Comput., 41, pp. A3075–A3096, 2019.
- [18] M. Fortin and R. Glowinski, *Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems*, Stud. Math. Appl. 15, North-Holland, Amsterdam/New York/Oxford, 1983.
- [19] L. P. Franca and S. L. Frey, *Stabilized finite element methods: II. The incompressible Navier–Stokes equations*, Comput. Methods Appl. Mech. Eng., 99, pp. 209–233, 1992.
- [20] G. H. Golub and R. S. Varga, *Chebyshev semi-iterative methods, successive over-relaxation iterative methods, and second order Richardson iterative methods, part I*, Numer. Math., 3, pp. 147–156, 1961.
- [21] G. H. Golub and R. S. Varga, *Chebyshev semi-iterative methods, successive over-relaxation iterative methods, and second order Richardson iterative methods, part II*, Numer. Math., 3, pp. 157–168, 1961.
- [22] M. D. Gunzburger, *Perspectives in Flow Control and Optimization*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2003.
- [23] M. D. Gunzburger and J. S. Peterson, *On conforming finite element methods for the inhomogeneous stationary Navier–Stokes equations*, Numer. Math., 42, pp. 173–194, 1983.
- [24] G. Heidel and A. Wathen, *Preconditioning for boundary control problems in incompressible fluid dynamics*, Numer. Linear Algebra Appl., 26, e2218, 2019.

- [25] M. Hinze and K. Kunisch, *Second order methods for optimal control of time-dependent fluid flow*, SIAM J. Optim. and Control, 40, pp. 925–946, 2001.
- [26] I. C. F. Ipsen, *A note on preconditioning nonsymmetric matrices*, SIAM J. Sci. Comput., 23, pp. 1050–1051, 2001.
- [27] C. Johnson and J. Saranen, *Streamline diffusion methods for the incompressible Euler and Navier–Stokes equations*, Math. Comp., 47, pp. 1–18, 1986.
- [28] O. Karakashian, *On a Galerkin–Lagrange multiplier method for the stationary Navier–Stokes equations*, SIAM J. Numer. Anal., 19, pp. 909–923, 1982.
- [29] D. Kay, D. Loghin, and A. Wathen, *A preconditioner for the steady-state Navier–Stokes equations*, SIAM J. Sci. Comput., 24, pp. 237–256, 2002.
- [30] S. Leveque, L. Bergamaschi, Á. Martínez, and J. W. Pearson, *Parallel-in-time solver for the all-at-once Runge–Kutta discretization*, arXiv preprint arXiv:2303.02090, 2023.
- [31] S. Leveque and J. W. Pearson, *Fast iterative solver for the optimal control of time-dependent PDEs with Crank–Nicolson discretization in time*, Numer. Linear Algebra Appl., 29, e2419, 2022.
- [32] S. Leveque and J. W. Pearson, *Parameter-robust preconditioning for Oseen iteration applied to stationary and instationary Navier–Stokes control*, SIAM J. Sci. Comput., 44, pp. B694–B722, 2022.
- [33] X. S. Li and J. W. Demmel, *SuperLU\_DIST: A scalable distributed-memory sparse direct solver for unsymmetric linear systems*, ACM Trans. Math. Softw., 29, pp. 110–140, 2003.
- [34] C. Lohmann C. and S. Turek, *Augmented Lagrangian acceleration of global-in-time pressure Schur complement solvers for incompressible Oseen equations*, J. Math. Fluid Mech., 26, Art. 27, 2024.
- [35] G. Matthies and L. Tobiska, *Local projection type stabilization applied to inf–sup stable discretizations of the Oseen problem*, IMA J. Numer. Anal., 35, pp. 239–269, 2015.
- [36] M. F. Murphy, G. H. Golub, and A. J. Wathen, *A note on preconditioning for indefinite linear systems*, SIAM J. Sci. Comput., 21, pp. 1969–1972, 2000.
- [37] J. Nocedal and S. Wright: *Numerical Optimization*, Springer, New York, 2nd Edition, 2006.
- [38] M. A. Olshanskii and M. Benzi: *An augmented Lagrangian approach to linearized problems in hydrodynamic stability*, SIAM J. Sci. Comput., 30, pp. 1459–147, 2008.
- [39] M. Olshanskii, G. Lube, T. Heister, and J. Löwe, *Grad-div stabilization and subgrid pressure models for the incompressible Navier–Stokes equations*, Comput. Meth. Appl. Mech. Eng., 198, pp. 3975–3988, 2009.
- [40] M. A. Olshanskii and A. Reusken, *Grad-div stabilization for Stokes equations*, Math. Comp., 73, pp. 1699–171, 2004.
- [41] I. V. Oseledets *et al.*, *TT-Toolbox software*; see <https://github.com/oseledets/TT-Toolbox>.
- [42] J. W. Pearson, A. J. Wathen, *A new approximation of the Schur complement in preconditioners for PDE-constrained optimization*, Numer. Linear Algebra Appl., 19, 816–829, 2012.
- [43] J. W. Pearson and A. J. Wathen, *Fast iterative solvers for convection–diffusion control problems*, Electron. Trans. Numer. Anal., 40, pp. 294–310, 2013.
- [44] M. Pošta and T. Roubíček, *Optimal control of Navier–Stokes equations by Oseen approximation*, Comp. Math. Appl., 53, pp. 569–581, 2007.
- [45] Y. Qiu, M. B. van Gijzen, J.-W. van Wingerden, M. Verhaegen, and C. Vuik, *Preconditioning Navier–Stokes control using multilevel sequentially semiseparable matrix computations*, Numer. Linear Algebra Appl., 28, e2349, 2021.
- [46] F. Rathgeber, D. A. Ham, L. Mitchell, M. Lange, F. Luporini, A. T. T. Mcrae, G.-T. Bercea, G. R. Markall, and P. H. J. Kelly, *Firedrake: automating the finite element method by composing abstractions*, ACM Trans. Math. Softw., 43, Art. 24, 2016.
- [47] T. Rees, H. S. Dollar, and A. J. Wathen, *Optimal solvers for PDE-constrained optimization*, SIAM J. Sci. Comput., 32, pp. 271–298, 2010.
- [48] Y. Saad, *A flexible inner–outer preconditioned GMRES algorithm*, SIAM J. Sci. Comput., 14, pp. 461–469, 1993.
- [49] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7, pp. 856–869, 1986.
- [50] L. R. Scott and M. Vogelius, *Norm estimates for a maximal right inverse of the divergence operator in spaces of piecewise polynomials*, ESAIM: Math. Mod. Num. Anal., 19, pp. 111–143, 1985.
- [51] J. Schöberl and W. Zulehner, *Symmetric indefinite preconditioners for saddle point problems with applications to PDE-constrained optimization problems*, SIAM J. Matrix Anal. Appl., 29, pp. 752–773, 2007.
- [52] B. F. Smith, P. E. Björstad and W. D. Gropp, *Domain decomposition: parallel multilevel*

- methods for elliptic partial differential equations*, Cambridge University Press, 1996.
- [53] M. Stoll and T. Breiten, *A low-rank in time approach to PDE-constrained optimization*, SIAM J. Sci. Comput., 37, pp. B1–B29, 2014.
  - [54] L. Tobiska and G. Lube, *A modified streamline diffusion method for solving the stationary Navier–Stokes equations*, Numer. Math., 59, pp. 13–29, 1991.
  - [55] F. Tröltzsch, *Optimal Control of Partial Differential Equations: Theory, Methods and Applications*, Graduate Series in Mathematics, American Mathematical Society, 2010.
  - [56] A. Wathen and T. Rees, *Chebyshev semi-iteration in preconditioning for problems including the mass matrix*, Electron. Trans. Numer. Anal., 34, pp. 125–135, 2009.
  - [57] T. Weier and G. Gerbeth, *Control of separated flows by time periodic Lorentz forces*, European J. Mech. B Fluids, 23, pp. 835–849, 2004.
  - [58] H. Xu and Z.-Q. Wang, *Rotated block diagonal preconditioners for Navier-Stokes control problems*, Comput. Math. Appl., 154, pp. 1–11, 2024.
  - [59] W. Zulehner, *Nonstandard norms and robust estimates for saddle point problems*, SIAM J. Matrix Anal. Appl., 32, pp. 536–560, 2011.