PRECONDITIONERS FOR TWO-PHASE INCOMPRESSIBLE NAVIER–STOKES FLOW[*]

NIALL BOOTLAND[†], ALISTAIR BENTLEY[‡], CHRISTOPHER KEES[§], AND ANDREW WATHEN[†]

Abstract. We consider iterative methods for solving the linearized Navier–Stokes equations arising from two-phase flow problems and the efficient preconditioning of such systems when using mixed finite element methods. Our target application is simulation within the Proteus toolkit; in particular, we will give results for a dynamic dam-break problem in two dimensions. We focus on a preconditioner motivated by approximate commutators which has proved effective, displaying mesh-independent convergence for the constant coefficient single-phase Navier–Stokes equations. This approach is known as the "pressure convection–diffusion" (PCD) preconditioner [H. C. Elman, D. J. Silvester, and A. J. Wathen, *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics*, 2nd ed., Oxford University Press, Oxford, 2014]. However, the original technique fails to give comparable performance in its given form when applied to variable coefficient Navier–Stokes systems such as those arising in two-phase flow models. Here we develop a generalization of this preconditioner appropriate for two-phase flow, requiring a new form for PCD. We omit considerations of boundary conditions to focus on the key features of two-phase flow. Before considering our target application, we present numerical results within the controlled setting of a simplified problem using a variety of different mixed elements. We compare these results with those for a straightforward extension to another commutator-based method known as the "least-squares commutator" (LSC) preconditioner, a technique also discussed in the aforementioned reference. We demonstrate that favorable properties of the original PCD and LSC preconditioners (without boundary adjustments) are retained with the new preconditioners in the two-phase situation.

Key words. preconditioner, two-phase flow, Navier–Stokes, Schur complement, finite elements

AMS subject classifications. 65F08, 65F10, 65N22, 76D05, 76D07, 76T10

DOI. 10.1137/17M1153674

1. Introduction. The motivation for this work stems from the computational challenges involved in simulating two-phase flows, in particular, air–water flows. The primary costs incurred in such simulations are in the solution of the linear systems arising after linearization and discretization of the governing equations. As such, the efficient solution of these linear systems is crucial and motivates the development of preconditioners appropriate for the variable coefficient nature of two-phase flow.

We consider incompressible flow of two immiscible Newtonian phases. Suppose the two-phase problem is defined on an open bounded domain $\Omega \subset \mathbb{R}^d$ ($d = 2, 3$) with one phase occupying Ω_1 and the second Ω_2 such that $\overline{\Omega} = \overline{\Omega}_1 \cup \overline{\Omega}_2$ and $\Omega_1 \cap \Omega_2 = \emptyset$.

[†]Mathematical Institute, University of Oxford, Radcliffe Observatory Quarter, Woodstock Road, Oxford, OX2 6GG, UK (bootland@maths.ox.ac.uk, wathen@maths.ox.ac.uk).

[‡]Department of Mathematical Sciences, Clemson University, Martin Hall, Clemson, SC 29634 (abentle@clemson.edu).

[§]Coastal and Hydraulics Laboratory, U.S. Army Engineer Research and Development Center, 3909 Halls Ferry Road, Vicksburg, MS 39180-6133 (christopher.e.kees@usace.army.mil).

Let the interface between the two phases be denoted $\Gamma = \partial\Omega_1 \cap \partial\Omega_2$. Note that Ω_1 and Ω_2 may vary over time and that neither set is required to be connected.

The fluid flow is modeled by the incompressible Navier–Stokes equations

$$(1) \qquad \alpha\rho\frac{\partial \boldsymbol{u}}{\partial t} + \rho\,\boldsymbol{u}\cdot\nabla\,\boldsymbol{u} + \nabla\,p - \nabla\cdot(2\mu\,\boldsymbol{Du}) = \rho\boldsymbol{f} \qquad \text{in } \Omega_i,$$

$$(2) \qquad \nabla\cdot\boldsymbol{u} = 0 \qquad \text{in } \Omega_i,\ i = 1, 2.$$

Here we have fluid velocity $\boldsymbol{u}$, pressure p, density ρ, dynamic viscosity μ, and body force per unit mass $\boldsymbol{f}$. We use the notation $\boldsymbol{Du} = \frac{1}{2}(\nabla\,\boldsymbol{u} + (\nabla\,\boldsymbol{u})^T)$ for the rate of deformation tensor and $\boldsymbol{\sigma}(\boldsymbol{u}, p) = -p\,\boldsymbol{I} + 2\mu\,\boldsymbol{Du}$ for the stress tensor. The value $\alpha = 0$ is taken for steady flow whilst $\alpha = 1$ for time-dependent flow, in which case the dependent variables may vary over time.

After suitable scaling we assume the variables are dimensionless with the piecewise constant density and viscosity being given by

$$(3) \qquad \rho = \begin{cases} 1 & \text{in } \Omega_1, \\[2mm] \dfrac{\rho_2}{\rho_1} & \text{in } \Omega_2, \end{cases} \qquad\qquad \mu = \begin{cases} \dfrac{1}{Re} & \text{in } \Omega_1, \\[2mm] \dfrac{\mu_2}{\mu_1 Re} & \text{in } \Omega_2, \end{cases}$$

where ρ_i and μ_i are, respectively, the (dimensional) density and viscosity of the fluid in Ω_i, $i = 1, 2$, and Re is the Reynolds number of the first fluid. We are interested in the behavior for increasing Reynolds numbers and so assume the Reynolds number of the second fluid is smaller than that of the first so that $\rho_2/\rho_1 \le \mu_2/\mu_1$. The difficulty in solving the problem typically increases for larger Reynolds numbers and so here Re characterizes the Reynolds number dependence.

In the time-dependent case we require a technique to keep track of the interface as it moves in time, typically, this might be a volume-of-fluid (VOF) method or a level set method (see, e.g., [46, 23, 31, 1]). Often, for numerical solution, the tracking of the fluids can be decoupled from solving the Navier–Stokes system so that, when we are required to solve this system, the interface position is known. Since our focus is on the efficient solution of the Navier–Stokes system we suppose that the interface position is known and thus the density and viscosity functions are fully specified. Note that, more generally, these need not be piecewise constant functions, for instance, when using a model which incorporates a narrow transition region around the interface.

To complete our model we require appropriate conditions on the boundary $\partial\Omega$ and interface Γ. Here, for simplicity, we consider enclosed flow in two dimensions. We suppose that the boundary is split into two parts $\partial\Omega_D$ and $\partial\Omega_{FS}$ with $\partial\Omega = \partial\Omega_D \cup \partial\Omega_{FS}$ and $\partial\Omega_D \cap \partial\Omega_{FS} = \emptyset$. We mpose the conditions

$$(4) \qquad \boldsymbol{u} = \boldsymbol{u}_D \qquad \text{on } \partial\Omega_D,$$

$$(5) \qquad \boldsymbol{u}\cdot\boldsymbol{n} = 0,\ (\boldsymbol{\sigma}(\boldsymbol{u}, p)\boldsymbol{n})\cdot\boldsymbol{t} = 0 \qquad \text{on } \partial\Omega_{FS},$$

$$(6) \qquad [\boldsymbol{u}] = \boldsymbol{0},\ [\boldsymbol{\sigma}(\boldsymbol{u}, p)\boldsymbol{n}] = \boldsymbol{g} \qquad \text{on } \Gamma,$$

where $\boldsymbol{n}$ and $\boldsymbol{t}$ are unit normal and tangent vectors to the corresponding boundary, $[a]_\Gamma = (a|_{\Omega_1} - a|_{\Omega_2})\,|_\Gamma$, and $\boldsymbol{g}$ is a localized force term, for instance corresponding to interface tension in the continuum surface force model [7, 23]. The boundary conditions are interpreted as fully specifying the velocity $\boldsymbol{u}_D$ on the Dirichlet boundary $\partial\Omega_D$ and applying free-slip on the wall boundary $\partial\Omega_{FS}$. Note that for enclosed flow an additional constraint is required to fix the level of the pressure such as specifying

the average pressure (see (9)). Across the interface we enforce that the velocity is continuous and if $\boldsymbol{g} = \boldsymbol{0}$, so that we neglect surface tension (this will be true in our case), then the second condition ensures that the normal stress is continuous. Finally, in the time-dependent case we also suppose that we have suitable initial conditions.

To numerically solve the problem we apply a nonlinear iteration method to treat the nonlinearity $\rho\,\boldsymbol{u}\cdot\nabla\,\boldsymbol{u}$ in the momentum equation (1); in our exposition we consider Picard iteration, though Newton iteration is possible too. In the time-dependent case we suppose that implicit time integration is applied, for instance, using the backward Euler scheme. See, for example, [48] for further details in the case of single-phase Navier–Stokes flow. The linearized equations that follow are known as the generalized Oseen problem: given a divergence-free vector field $\boldsymbol{w}$, find $\boldsymbol{u}$ and p such that

$$(7) \qquad \frac{\alpha\rho}{\Delta t}\boldsymbol{u} + \rho\,\boldsymbol{w}\cdot\nabla\,\boldsymbol{u} + \nabla\,p - \nabla\cdot(2\mu\,\boldsymbol{Du}) = \boldsymbol{r} \qquad\qquad \text{in } \Omega_i,$$

$$(8) \qquad\qquad\qquad\qquad\qquad \nabla\cdot\boldsymbol{u} = 0 \qquad\qquad \text{in } \Omega_i,\ i = 1, 2,$$

along with the boundary conditions (4)–(6). In Picard iteration the wind $\boldsymbol{w}$ is the approximation of $\boldsymbol{u}$ from the previous iteration, or else zero in the starting iteration, in which case the corresponding Stokes problem is to be solved. Here $\boldsymbol{r}$ collects the forcing term $\rho\boldsymbol{f}$ and all other known terms such as those arising from the time integration. Solution of the linearized system (7)–(8) along with appropriate boundary conditions is key to the overall efficiency of the computational modeling and is the focus for our development of preconditioners.

To discretize in space we consider finite element methodology, nonetheless, the preconditioning techniques we will describe could equally be applied to alternative discretizations such as the MAC scheme [25]. The details of the weak formulation and finite element discretization are known and are found in, for example, [23]; we summarize the resulting finite element problem here. We assume appropriate, though not necessarily inf–sup stable, finite element spaces for velocity and pressure given by

$$(9) \qquad \mathbb{V}^h \subset H^1(\Omega)^2 \qquad\qquad \text{and} \qquad\qquad \mathbb{Q}^h \subset \left\{ p \in L^2(\Omega)\ :\ \int_\Omega p = 0 \right\},$$

respectively. Then we wish to find $\boldsymbol{u}_h \in \mathbb{V}^h$ and $p_h \in \mathbb{Q}^h$ such that

$$(10) \qquad \frac{\alpha}{\Delta t}m(\boldsymbol{u}_h, \boldsymbol{v}_h) + n(\boldsymbol{w}_h; \boldsymbol{u}_h, \boldsymbol{v}_h) + b(p_h, \boldsymbol{v}_h) + a(\boldsymbol{u}_h, \boldsymbol{v}_h) = (\boldsymbol{r}, \boldsymbol{v}_h),$$

$$(11) \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad b(q_h, \boldsymbol{u}_h) = c(p_h, q_h)$$

for all $\boldsymbol{v}_h \in \mathbb{V}^h_0$ and $q_h \in \mathbb{Q}^h$. Here $\boldsymbol{w}_h$ is the known previous iterate for $\boldsymbol{u}_h$ as part of the nonlinear iteration and $\mathbb{V}^h_0$ is the finite element space with homogeneous essential (Dirichlet) boundary conditions applied. Further, $(\cdot, \cdot)$ denotes the standard L^2 inner product while

$$(12) \qquad m(\boldsymbol{u}, \boldsymbol{v}) = \int_\Omega \rho\,\boldsymbol{u}\cdot\boldsymbol{v}, \qquad\qquad n(\boldsymbol{z}; \boldsymbol{u}, \boldsymbol{v}) = \int_\Omega \rho\,(\boldsymbol{z}\cdot\nabla\,\boldsymbol{u})\cdot\boldsymbol{v},$$

$$(13) \qquad a(\boldsymbol{u}, \boldsymbol{v}) = \int_\Omega 2\mu\,\mathrm{Tr}\,(\boldsymbol{Du}\,\boldsymbol{Dv}), \qquad\qquad b(p, \boldsymbol{u}) = -\int_\Omega p\,\nabla\cdot\boldsymbol{u},$$

and $c(\cdot, \cdot)$ is a stabilization term, needed for inf–sup unstable spaces, or else is zero. Note that the trilinear form $n(\cdot; \cdot, \cdot)$ may additionally require some stabilization in the case of dominating advection, for instance, by using a streamline-diffusion method (see [48] for discussion in the single-phase case).

To obtain a linear system we let $\{\boldsymbol{\varphi}_j\}$ be a set of velocity basis functions and $\{\psi_j\}$ be a set of pressure basis functions. Defining the matrices

(14)
$$M_{i,j}^{(\rho)} = m(\boldsymbol{\varphi}_j, \boldsymbol{\varphi}_i), \quad N_{i,j}^{(\rho)} = n(\boldsymbol{w}_h; \boldsymbol{\varphi}_j, \boldsymbol{\varphi}_i),$$

$$A_{i,j}^{(\mu)} = a(\boldsymbol{\varphi}_j, \boldsymbol{\varphi}_i), \quad B_{i,j} = b(\psi_i, \boldsymbol{\varphi}_j), \quad C_{i,j} = c(\psi_j, \psi_i),$$

then the generic form of the linear systems to be solved is

(15)
$$\begin{pmatrix} \frac{\alpha}{\Delta t} M^{(\rho)} + N^{(\rho)} + A^{(\mu)} & B^T \\ B & -C \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{p} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix}.$$

Here the right-hand side vectors $\mathbf{f}$ and $\mathbf{g}$ collect together the terms involving $\boldsymbol{r}$ and known boundary terms. For clarity, we will use the notation of bracketed exponents when we wish to explicitly illustrate the scaling by density and viscosity within the matrices. The stabilization matrix C will depend on the choice of finite elements used; as such we defer precise details until discussing such choices in subsection 4.2. Note that C might also incorporate some scaling, typically with the inverse of viscosity.

The block system (15) is of (generalized) saddle-point form for the coefficient vectors $\mathbf{u}$ and $\mathbf{p}$. For large problems, this system must be solved using iterative methods, typically a Krylov subspace method. Since the presence of $N^{(\rho)}$ makes the system nonsymmetric, a common choice of the Krylov method is GMRES [47]; this is what we shall use for our numerical results. The performance of an iterative method depends on the conditioning of the linear system and thus preconditioners are essential for efficient computation. The preconditioners we build on and develop here are block preconditioners which show mesh-independent convergence and only mild dependence on the Reynolds number. Before describing block preconditioning techniques, we discuss related work in preconditioning of the Navier–Stokes equations and similar problems in two-phase flow.

In addition to block preconditioners, a wide variety of preconditioners have been proposed for saddle-point systems originating from incompressible flow and related problems. These include domain decomposition methods (see [45]), however, standard approaches may not give robust and scalable results as detailed in [13]. Nonetheless, by incorporating a multilevel approach along with aggressive coarsening based on graph partitioning, as explored in [36], scalability is seen in the tests of [13]. Multigrid techniques are often used for subproblems in the solution process, however, specialist multigrid methods have been proposed for the full Navier–Stokes system going back to Vanka [54]. While the approach of Vanka uses a coupled smoother, more recently uncoupled smoothers have been seen to offer advantages in efficiency; for example, a method for the Stokes equations using a distributive Gauss–Seidel relaxation based on the least-squares commutator is introduced in [55]. Another approach is to use incomplete LU factorization (ILU), a technique developed in the saddle-point ILU preconditioner of [52] and the ILU preconditioner for nonsymmetric saddle-point matrices of [33]. Augmented Lagrangian techniques can also yield effective methods for the Navier–Stokes equations; in particular, see the modified augmented Lagrangian preconditioners of [5].

In this work our focus is on block preconditioners, which have seen considerable attention in recent years. Here, a matrix factorization of the block system is utilized and appropriate approximations of the factors are needed to devise preconditioners which are efficient. The principal challenge is an effective approximation to the Schur complement arising in the factorization. A taxonomy of these approximate block factorization methods for incompressible Navier–Stokes flow is given in [16]. Though

not originally envisaged within this framework, this includes SIMPLE-type methods [53]. Other popular block preconditioners for Navier–Stokes flow are based on approximate commutators, primarily these are the pressure convection–diffusion (PCD) and least-squares commutator (LSC) preconditioners, which are discussed in detail in [20]. Though most often described through an approximate commutation relationship, the PCD approach was initially devised in [30] by considering Green's tensors. The PCD preconditioner requires a convection–diffusion operator projected onto the discrete pressure space and this additional operator must be constructed. In search of a more automatic approach, the LSC preconditioner described in [15], but developed from [18], is based solely on algebraic considerations of minimizing the norm of the commutator in a least-squares sense. Both of these commutator-based preconditioners show scalable results in the tests of [20] and [13].

The preconditioners we have described above are primarily considered in the case of constant density and viscosity flows and, to the authors' knowledge, little has been explored for preconditioning variable density and viscosity Navier–Stokes problems. Nonetheless, recently, approaches based on augmented Lagrangian techniques [5] were developed for variable viscosity in [26] and extended to incompressible non-Newtonian flows in [27]; see also [2] for a simpler approach incorporating variable density in the time-dependent case. In the case of variable coefficient problems, such as two-phase flow, most work is devoted to the Stokes problem. Here a pressure mass matrix scaled by the inverse of the viscosity is a good choice of Schur complement preconditioner for the stationary problem, as shown in the case of two-phase flow in [43] and more generally investigated for variable viscosity in [22]; see also [8, 38]. By considering an abstract parameter dependent saddle-point system, this is extended to two-phase non-stationary Stokes flow in [41] and can be seen as generalizing the Cahouet–Chabard preconditioner [9]. Similar block preconditioners are constructed for the variable density and viscosity Stokes problem when using finite volumes in [10], and applied to uniform staggered grids, showing promising results. This work brings together finite element literature on solving the coupled velocity–pressure system making use of Schur complements and applies such techniques within the finite volume setting, where the dominant paradigm has been splitting or projection methods [24]. A variant of LSC which takes into account viscosity contrast is given in [39] and used for their studies in computational geodynamics. Another approach is the Schur method described in [51]; this paper also compares what is effectively the same LSC variant of [39] and labels this LSC_D, a name we shall adopt here too. While these methods hold potential to be used for variable coefficient Navier–Stokes flow we have found no reference to such an application. We consider the adaptation of such block preconditioners to two-phase Navier–Stokes flow, in particular, the PCD as well as LSC approaches.

The outline for the remainder of this document is as follows. In section 2 we detail block preconditioning of the saddle-point system before outlining the PCD approach for single-phase flow. Our main contribution is given in section 3 where we extend this preconditioner to two-phase flow. We also describe the LSC approach and provide a simple extension that improves performance in the case of two-phase flow. Numerical results for a simplified problem are given in section 4 for the choice of $\boldsymbol{Q}_2$–$\boldsymbol{Q}_1$, $\boldsymbol{Q}_1$–$\boldsymbol{Q}_1$, and $\boldsymbol{Q}_2$–$\boldsymbol{P}_{-1}$ elements. For this we use the software package IFISS [49, 19] within MATLAB, which we have adapted to incorporate varying density and viscosity. This allows us to describe dependence on the parameters in a controlled manner; we will observe that the favorable properties of the preconditioners (without boundary adjustments) are retained. In section 5 we focus on our target application of a dam-break problem simulated within Proteus, a computational methods and sim-

ulation toolkit (http://proteustoolkit.org). Since Proteus uses unstructured meshes and $\boldsymbol{P}_1$–$\boldsymbol{P}_1$ elements, we first provide results for our simplified problem in Proteus, verifying similar behavior to our computations in IFISS, before moving to consider the performance of PCD for the dam-break simulation. Finally, section 6 draws together our conclusions.

2. Block preconditioning techniques. The saddle-point system (15) has a coefficient matrix of the form

$$(16) \qquad K = \left(\begin{array}{cc} F & B^T \\ B & -C \end{array} \right),$$

where $F = A^{(\mu)} + N^{(\rho)} + \frac{\alpha}{\Delta t} M^{(\rho)}$. A standard approach of [40, 29] for preconditioning, based on a block LU-decomposition of K, is to right precondition using a matrix of the block triangular form

$$(17) \qquad P = \left(\begin{array}{cc} \widehat{F} & B^T \\ 0 & -\widehat{S} \end{array} \right),$$

where $\widehat{F}$ and $\widehat{S}$ approximate F and the Schur complement $S = BF^{-1}B^T + C$, arising in the decomposition, respectively. If the approximations were exact then only two iterations would be required by the Krylov method [40]. However, F and, in particular, S are impractical to work with. Since in application of the preconditioner (17) we require the actions of $\widehat{F}^{-1}$ and $\widehat{S}^{-1}$, we seek approximations where these actions are efficient yet remain faithful to the operators they approximate. Typically for incompressible fluid flow problems, the difficulty in using the preconditioner (17) is in finding a good approximation to S; see, for instance, [20] for further details.

As noted in [30], for non-self-adjoint problems which yield nonsymmetric linear systems such as (16), the development of preconditioners typically relies on heuristic arguments since eigenvalues alone are not generally descriptive of the convergence behavior for nonsymmetric Krylov methods (though see [32, 37]). We will focus on the approach of the PCD preconditioner [30, 48] for approximating the Schur complement. This is comprehensively described in [20]; here we briefly outline the key idea and provide the most basic form. In the remainder of this section we suppose that the density and viscosity are everywhere constant (and so $\rho \equiv 1$) so that we consider single-phase flow. However, to present these ideas in a transparent way for extension to variable coefficient problems we retain the density and viscosity scaling. Note that, due to our nondimensionalization, μ here is identified with the kinematic viscosity.

2.1. The pressure-convection diffusion preconditioner. The approach of the PCD preconditioner is most often motivated by considering the commutator $\mathcal{E}$ between the divergence operator and the convection–diffusion operator[1]

$$(18) \qquad \mathcal{F} = -\mu \nabla \cdot \nabla + \rho \boldsymbol{w}_h \cdot \nabla .$$

We suppose that we can have an analogous operator to $\mathcal{F}$ on the pressure space, denoted $\mathcal{F}_p$, so that

$$(19) \qquad \mathcal{E} = \nabla \cdot \mathcal{F} - \mathcal{F}_p \nabla \cdot .$$

---

[1]For a constant viscosity μ the viscous term, $-\nabla \cdot (2\mu \boldsymbol{D})$, reduces to the vector Laplacian using the incompressibility constraint; this is the form most often written for exposition of the PCD approach.

Though $\mathcal{F}_p$ is in general not rigorously defined, we suppose the commutator is small in some sense. It is noted in [20] that $\mathcal{E}$ would be zero if $\boldsymbol{w}_h$ were a constant and the operators were defined on an unbounded domain. The idea is then to use a discrete version of $\mathcal{E}$ and within it to isolate the Schur complement S. Upon equating the discrete commutator to be zero we obtain an approximation to S in terms of finite element matrices. When time stepping, we can define $\mathcal{F}$ to also include the term $\frac{\alpha\rho}{\Delta t}$ from (7). Since this additional term is a scaling of the identity operator it cancels in the commutator. Thus the same approach can be applied in the time-dependent case.

For the discrete commutator, to correctly scale the discrete operators we require the finite element mass matrices on the pressure and velocity spaces given by

$$(20) \qquad M_{p;i,j} = \int_\Omega \psi_j \cdot \psi_i, \qquad\qquad M_{i,j} = \int_\Omega \boldsymbol{\varphi}_j \cdot \boldsymbol{\varphi}_i.$$

The discrete commutator is then

$$(21) \qquad \mathcal{E}_h = \left(M_p^{-1}B\right)\left(M^{-1}F\right) - \left(M_p^{-1}F_p\right)\left(M_p^{-1}B\right),$$

where F_p represents the discrete form of $\mathcal{F}_p$. On equating $\mathcal{E}_h$ to be zero this can be rearranged, after postmultiplication by $F^{-1}B^T$, to give the approximation

$$(22) \qquad BF^{-1}B^T \approx M_p F_p^{-1} B M^{-1} B^T.$$

For simplicity we will now assume that the pressure approximation is continuous, though extensions can be made to discontinuous pressure approximation [20]. We will provide results in section 4 for $\boldsymbol{Q}_2$–$\boldsymbol{P}_{-1}$ elements, a pairing with discontinuous pressure. In the continuous case we can define F_p as

$$(23) \qquad F_{p;i,j} = \int_\Omega \mu \nabla \psi_j \cdot \nabla \psi_i + \int_\Omega \rho \left(\boldsymbol{w}_h \cdot \nabla \psi_j\right) \cdot \psi_i + \frac{\alpha}{\Delta t} \int_\Omega \rho \psi_j \cdot \psi_i.$$

These terms can be written as $F_p = A_p^{(\mu)} + N_p^{(\rho)} + \frac{\alpha}{\Delta t} M_p^{(\rho)}$, respectively. Finally, the scaled Laplacian term $BM^{-1}B^T$ is replaced by the sparse pressure Laplacian A_p, where

$$(24) \qquad A_{p;i,j} = \int_\Omega \nabla \psi_j \cdot \nabla \psi_i.$$

This yields the PCD approximation to the inverse Schur complement

$$(25) \qquad \widehat{S}^{-1} = A_p^{-1} F_p M_p^{-1},$$

using the matrices defined in (24), (23), and (20), respectively. The choice (25) for $\widehat{S}^{-1}$ is also applicable when a stabilization matrix $C \neq 0$ is necessary [20, section 9.2.1]. Key features of (25) are that we need only multiply by F_p and do not have to invert such a matrix, nevertheless the nonnormality of the problem is included within the preconditioner through F_p. For a practical implementation, multigrid methods can be used to effect the action of A_p^{-1} and M_p can be replaced by the spectrally equivalent matrix $\mathrm{diag}(M_p)$ and thus easily inverted [56]. Alternatively, the action of M_p^{-1} on a vector is well approximated using Chebyshev semi-iteration with only a small number of iterations required [57].

We remark that the commutator can also be taken with the gradient instead of the divergence, as originally done in [48]. The effect of this is that the ordering of the operators in (25), our approximate inverse Schur complement $\widehat{S}^{-1}$, is reversed. It was seen when considering boundary conditions that the choice of the divergence was favorable; see [20, Remark 9.3].

2.2. The relation of PCD to the Cahouet–Chabard preconditioner. We conclude this section by noting a relation between the PCD preconditioner (25) and the Cahouet–Chabard preconditioner [9] for the generalized Stokes problem; this will provide a useful viewpoint on how a two-phase PCD preconditioner should behave.

In Stokes flow the convective term is omitted so that in (16) for single-phase flow $F = \mu A + \frac{1}{\Delta t} M$, where A is the discrete velocity Laplacian and M is the velocity mass matrix (since $\rho \equiv 1$). Cahouet and Chabard show that the appropriate Schur complement which balances these two terms is

$$(26) \qquad \widehat{S}^{-1} = \mu M_p^{-1} + \frac{1}{\Delta t} A_p^{-1},$$

where A_p is a discrete pressure Laplacian operator, for instance, given by (24) for continuous pressure approximation.

Now consider the PCD preconditioner in this case. Since the convective term is not present we have $F_p = \mu A_p + \frac{1}{\Delta t} M_p$ and thus the Schur complement approximation

$$(27) \qquad \widehat{S}^{-1} = A_p^{-1} F_p M_p^{-1} = A_p^{-1} \left(\mu A_p + \frac{1}{\Delta t} M_p \right) M_p^{-1} = \mu M_p^{-1} + \frac{1}{\Delta t} A_p^{-1}.$$

Hence we see that the PCD preconditioner precisely reduces to the Cahouet–Chabard preconditioner in the case of generalized Stokes flow. This provides the viewpoint that PCD extends the Cahouet–Chabard preconditioner to the case of Navier–Stokes flow.

3. Preconditioners for two-phase flow. We now move to the focus of our work, namely, preconditioning for two-phase flow. The primary new feature in the problem is the variable density and viscosity which gives additional scaling in the equations. The key to adapting the preconditioners detailed in section 2 will be to incorporate this scaling appropriately into their formulation.

3.1. Two-phase pressure-convection diffusion preconditioning. On its first introduction, the PCD preconditioner for steady flow was derived using Fourier techniques and Green's tensors [30]. Due to the variable coefficient nature of two-phase flow, this approach cannot apply here. However, in the original approach, in order that the preconditioner defaults to the optimal choice in the Stokes limit when the convective term tends to zero, a mass matrix is included to give the correct scaling. The same philosophy applies here.

Before addressing the appropriate choice in the Stokes case, we must be clear on the form of the viscous term used. In the literature different forms are considered, including $-\nabla \cdot (\mu \nabla \boldsymbol{u})$ in [43], $-\nabla \cdot (\mu \boldsymbol{Du})$ in [22, 41], and $-\nabla \cdot (2\mu \boldsymbol{Du})$ in [23, 10]. The overall scaling of the latter term contains an additional factor of two compared with the former choices, as such an appropriate Schur complement approximation for the Stokes problem differs by a factor of a half. The distinction between different forms of the viscous term used (including incorporating bulk viscosity) and the effect on the Schur complement is discussed further in [10]. Since we use the latter choice above, we present results for this case, including a factor of two where appropriate as compared with the cited references.

For two-phase flow, the appropriate pressure mass matrix arising in the Stokes case is given in [43] and scales inversely with the viscosity as

$$(28) \qquad M_{p;i,j}^{(1/\mu)} = \int_\Omega (2\mu)^{-1} \psi_j \cdot \psi_i.$$

This mass matrix is also suitable for general variable viscosity Stokes flows [22].

Looking at the construction of the PCD preconditioner in subsection 2.1, F_p is given as in (23), but now with the density and viscosity being piecewise constant. If we assume that PCD still takes the form in (25) but with additional scaling, then, so that the viscosity scaling within the Schur complement is commensurate, when we use the scaled mass matrix $M_p^{(1/\mu)}$ we require the pressure Laplacian-type term

$$(29) \qquad A_{p;i,j}^{(\mu)} = \int_\Omega \mu \nabla \psi_j \cdot \nabla \psi_i.$$

Note this is already constructed as part of F_p just as in the single-phase case since for constant viscosity $A_p^{(\mu)} = \mu A_p$. These choices give a Schur complement approximation

$$(30) \qquad \widehat{S}^{-1} = \left(A_p^{(\mu)}\right)^{-1} F_p \left(M_p^{(1/\mu)}\right)^{-1}.$$

However, from our numerical experience, it is apparent that the performance of this preconditioner depends poorly on the density ratio of the two fluids. In particular, (30) does not work so well in the time-dependent case. To understand this we go back to the relationship with the Cahouet–Chabard preconditioner detailed in subsection 2.2.

By considering an abstract parameter dependent saddle-point system, the authors of [41] derive a preconditioner for two-phase time-dependent Stokes flow which can be seen as generalizing the Cahouet–Chabard preconditioner. This preconditioner uses an approximation to the Schur complement given by

$$(31) \qquad \widehat{S}^{-1} = \left(M_p^{(1/\mu)}\right)^{-1} + \frac{1}{\Delta t}\left(A_p^{(1/\rho)}\right)^{-1},$$

where the pressure Laplacian-type term is now inversely proportional to the density,

$$(32) \qquad A_{p;i,j}^{(1/\rho)} = \int_\Omega \rho^{-1} \nabla \psi_j \cdot \nabla \psi_i.$$

The new feature here is the dependence on the density ρ. This can be understood heuristically by considering the single-phase case as follows. First, we note that the viscosity μ in (26) is in fact identified with the kinematic viscosity. In our case, where μ is the dynamic viscosity, the inverse Schur complement is multiplied by the density (in addition to there being a factor of two arising from the choice of viscous term) and so the approximation should read

$$(33) \qquad \widehat{S}^{-1} = 2\mu M_p^{-1} + \frac{\rho}{\Delta t} A_p^{-1} = \left((2\mu)^{-1} M_p\right)^{-1} + \frac{1}{\Delta t}\left(\rho^{-1} A_p\right)^{-1}.$$

In the two-phase case, the scaling with density and viscosity must be incorporated within the integral definition of the matrices and so we obtain (31). An inverse Schur complement approximation very similar to (31), appropriate for finite volume Stokes solvers, is given in [10]. Here the marker and cell (MAC) discretization on staggered grids is employed. The primary difference is that for finite volumes a diagonal matrix of viscosities at each pressure degree of freedom can be used for the viscous term, whereas in the finite element setting a mass matrix is appropriate.

It is now clear, from the viewpoint of the relation between PCD and the Cahouet–Chabard preconditioner, that the approximation (30) does not default to the correct choice of the preconditioner (31) in the time-dependent Stokes case. To give a more robust generalization of PCD to two-phase flow with appropriate scaling we must split the matrix F_p and treat the separate terms accordingly.

The matrix F_p in the two-phase case consists of the three parts

$$(34) \qquad F_p = A_p^{(\mu)} + N_p^{(\rho)} + \frac{\alpha}{\Delta t} M_p^{(\rho)}.$$

To treat the viscous part $A_p^{(\mu)}$ of F_p we use the scaling from (28) and (29) but to treat the remaining part, depending on the density, we scale using the mass matrix

$$(35) \qquad M_{p;i,j}^{(\rho)} = \int_\Omega \rho \, \psi_j \cdot \psi_i$$

and use the density scaled Laplacian-type term (32). This yields the approximation

$$(36) \qquad \widehat{S}^{-1} = \left(M_p^{(1/\mu)} \right)^{-1} + \left(A_p^{(1/\rho)} \right)^{-1} \left(N_p^{(\rho)} + \frac{\alpha}{\Delta t} M_p^{(\rho)} \right) \left(M_p^{(\rho)} \right)^{-1}.$$

In practice, however, we have found it beneficial to further cancel the scaling with the density ρ in the final two bracketed terms of (36) to give two-phase PCD as

$$(37) \qquad \widehat{S}^{-1} = \left(M_p^{(1/\mu)} \right)^{-1} + \left(A_p^{(1/\rho)} \right)^{-1} \left(N_p^{(1)} + \frac{\alpha}{\Delta t} M_p^{(1)} \right) \left(M_p^{(1)} \right)^{-1}.$$

Here the bracketed exponent is given as 1 to be clear that these are unscaled terms whose definition does not include ρ. Namely we have

$$(38) \qquad N_{p;i,j}^{(1)} = \int_\Omega (\boldsymbol{w}_h \cdot \nabla \psi_j) \cdot \psi_i,$$

while $M_p^{(1)}$ is the standard mass matrix. While this gives a different approximation, the overall scaling of the two separate terms in the sum remain the same and, although it is feasible to cancel the density as in (37), it remains important that $A_p^{(1/\rho)}$ keeps the correct scaling. Further, note that $M_p^{(1)}$ remains the same at each time step while $M_p^{(\rho)}$ would change due to the moving phases and so would need recomputing. From the arguments presented here, it is not clear that (37) should be preferable to (36), however, we are actively investigating this further.

As with single-phase PCD, in the two-phase PCD approximation of (37) only one pressure Laplacian-type solve is needed; this is because the appearance of $A_p^{(\mu)}$ cancels out. We note that in the time-dependent Stokes case, when $N_p^{(1)}$ is zero, we have

$$(39) \qquad \begin{aligned} \widehat{S}^{-1} &= \left(M_p^{(1/\mu)} \right)^{-1} + \left(A_p^{(1/\rho)} \right)^{-1} \left(\frac{1}{\Delta t} M_p^{(1)} \right) \left(M_p^{(1)} \right)^{-1} \\ &= \left(M_p^{(1/\mu)} \right)^{-1} + \frac{1}{\Delta t} \left(A_p^{(1/\rho)} \right)^{-1}, \end{aligned}$$

and so we return to the generalized Cahouet–Chabard preconditioner (31). Further, despite the new form of the preconditioner (37), it is a generalization of the original PCD preconditioner (25) since for everywhere constant viscosity μ and density $\rho \equiv 1$ (and thus using the Laplacian for the viscous term, removing the factor two in (28))

$$(40) \qquad \begin{aligned} \widehat{S}^{-1} &= \left(\mu^{-1} M_p \right)^{-1} + A_p^{-1} \left(N_p + \frac{\alpha}{\Delta t} M_p \right) M_p^{-1} \\ &= A_p^{-1} \left(\mu A_p \right) M_p^{-1} + A_p^{-1} \left(N_p + \frac{\alpha}{\Delta t} M_p \right) M_p^{-1} \\ &= A_p^{-1} F_p M_p^{-1}. \end{aligned}$$

We remark that the generalization of PCD to two-phase flow in (37) cannot be written in the form of the original PCD in (25) with scaling in A_p and M_p. The scaling here comes from the mass matrices on the velocity and pressure spaces which depend on the corresponding norms (see [41] for the choice of norms in the generalized Stokes case). If simple scaled norms are used in the mass matrix scaling of the terms in the discrete commutator (21), one for the pressure space and one for the velocity space, then it is not possible to construct a preconditioner which defaults to the appropriate generalized Cahouet–Chabard preconditioner. We are currently exploring whether commutator-based approaches might still be extensible to derive an appropriate two-phase preconditioner. Nonetheless, it appears that such a form as (25) is a simplification which can only be made for an everywhere constant density and viscosity flow.

While we have considered discretization with finite elements, we envisage that the approach can also be utilized with finite volumes or finite differences; for instance, using the MAC discretization [25] and building on the work in [10]. The primary requirement is an appropriate discretization of the pressure space convection term.

Finally, we note that the features of PCD described in subsection 2.1 are retained in that we do not have to invert a pressure space convection–diffusion operator and only one discrete Laplacian-type subproblem must be solved in applying (37). In practice, each mass matrix can be replaced by a diagonal approximation or a Chebyshev semi-iteration can be used to approximate the inverse action of each mass matrix on a vector. Multigrid methods are again applicable for the solution of the scaled pressure Laplacian-type subproblem.

Before moving on to discuss our numerical results, we first briefly detail another commutator-based preconditioner often considered alongside PCD: the LSC method.

3.2. The least-squares commutator preconditioner. While still using the commutator (19), the LSC preconditioner uses a purely algebraic approach to defining F_p by choosing it so that the discrete commutator (21) is small in a least-squares sense. It turns out that it is more convenient to consider the adjoint of the commutator and choose F_p by minimizing each individual vector norm of the columns of F_p^T; for further details see [20]. This yields the weighted least-squares problem

$$(41) \qquad \min \left\| \left[M^{-1} F^T M^{-1} B^T \right]_j - M^{-1} B^T M_p^{-1} \left[F_p^T \right]_j \right\|_M$$

for each column j. Using the normal equations, F_p can be given as

$$(42) \qquad F_p = \left(B M^{-1} F M^{-1} B^T \right) \left(B M^{-1} B^T \right)^{-1} M_p$$

and, hence, substituting this expression into (22), we arrive at the Schur complement approximation

$$(43) \qquad B F^{-1} B^T \approx \left(B M^{-1} B^T \right) \left(B M^{-1} F M^{-1} B^T \right)^{-1} \left(B M^{-1} B^T \right).$$

In practice we replace M with its diagonal $T = \mathrm{diag}(M)$, and so construct the sparse scaled Laplacian $B T^{-1} B^T$. The LSC inverse Schur complement approximation then takes the form

$$(44) \qquad \widehat{S}^{-1} = \left(B T^{-1} B^T \right)^{-1} \left(B T^{-1} F T^{-1} B^T \right) \left(B T^{-1} B^T \right)^{-1}.$$

This approximation is not immediately applicable to stabilized elements where $C \neq 0$, however, appropriate modifications can be made in such an instance [17]. As with

the PCD method, the action of the inverse of the discrete Laplacian term can be effectively applied using multigrid methods; however, note that we now need to solve two discrete Laplacian subproblems in the application of $\widehat{S}^{-1}$. Multiplication by F captures the nonnormality of the problem within the action of $\widehat{S}^{-1}$.

While the LSC approach is purely algebraic, incorporating boundary adjustments is important in order to obtain scalable performance for enclosed flow problems such as those we consider; see [20]. We do not pursue this topic here as we focus on the key aspects of generalizing the original preconditioners to be more effective in the case of two-phase flow. Since we expect poorer scalability from LSC in this situation, for brevity we choose to forgo discussion on the stabilization terms required when using inf–sup unstable elements, which follow in a straightforward manner from [17], and we will omit the LSC approach when discussing our target dam-break simulation.

The scaling within LSC is known to be an important factor. In [18], a precursor to LSC is derived which has the same form as (44) except that the scaling matrix T is not present. That so-called $BFBt$ method was seen to have poorer performance than the later LSC method [15]. In general, the choice of scaling is essential for the efficiency of such a preconditioner. In an effort to use a scaling which incorporates changes in viscosity for the Stokes problem, [39] use the form

$$(45) \qquad \widehat{S}^{-1} = \left(BD^{-1}B^T\right)^{-1} \left(BD^{-1}FD^{-1}B^T\right) \left(BD^{-1}B^T\right)^{-1},$$

where $D_{i,i} = \max_j |F_{i,j}|$ is a diagonal scaling matrix. In [51] the authors consider $D = \mathrm{diag}(F)$, which in practice is essentially the same scaling, and label this method $\mathrm{LSC_D}$. Our experience shows that, for the Navier–Stokes case, this preconditioner does not scale well with the problem size, with it exhibiting a poor h-dependence. To improve performance over both $\mathrm{LSC_D}$ and the original (single-phase) approach we consider a scaled mass matrix coming from a viscosity weighted least-squares problem.

The choice of norms on the velocity and pressure spaces determines the form of the mass matrices used in the discrete commutator (21) and weighted least-squares problem (41). However, the resulting LSC approximation is independent of the norm on the pressure space. In view of this, we introduce the viscosity scaled norm on the velocity space and the corresponding velocity mass matrix

$$(46) \qquad \|\boldsymbol{u}\|_\mu^2 = \int_\Omega \mu\,\boldsymbol{u}\cdot\boldsymbol{u}, \qquad\qquad M_{i,j}^{(\mu)} = \int_\Omega \mu\,\boldsymbol{\varphi}_j\cdot\boldsymbol{\varphi}_i,$$

and keep the standard L^2 norm on the pressure space. Then the construction of the discrete commutator and weighed least-squares problem is the same as in (21) and (41), except with M replaced by $M^{(\mu)}$; as such we omit the details here. Letting $T^{(\mu)} = \mathrm{diag}(M^{(\mu)})$, the resulting two-phase LSC Schur complement approximation is

$$(47) \qquad \widehat{S}^{-1} = \left(BT^{(\mu)^{-1}}B^T\right)^{-1} \left(BT^{(\mu)^{-1}}FT^{(\mu)^{-1}}B^T\right) \left(BT^{(\mu)^{-1}}B^T\right)^{-1}.$$

Unlike with PCD, it is not necessary to split the convection–diffusion term and scale differently. Our experience is that such a splitting of F does not gain any improvement in the preconditioner. We note that when the viscosity μ is everywhere constant this can be factored out and canceled to give the original form (44). As with the original LSC preconditioner, (47) requires the solution of two scaled discrete Laplacian-type subproblems which, in practice, can be achieved using multigrid techniques.

Unlike the PCD method, the LSC approach does not simplify to the Cahouet–Chabard preconditioner in the Stokes case and so we cannot use this viewpoint to

guide our choice of scaling. We will see that our choice of viscosity scaling in (47) can significantly improve the robustness of the LSC approach to contrasts in the material parameters, however, it remains unclear whether this choice is optimal. We are currently investigating the importance of such scaling and the impact of boundary adjustments when LSC-type methods are applied to two-phase flow problems.

3.3. A comparison of PCD and LSC. Detailed comparisons between the original PCD and LSC preconditioners are given in [44] and [20]. Our two-phase strategies compare similarly. To summarize the key differences: PCD requires the construction of additional matrices on the pressure space while LSC requires only matrices which are readily available; this means that the LSC method directly applies when a discontinuous pressure approximation is used while the formulation of PCD is further complicated. On the other hand, PCD naturally extends to stabilized elements while LSC does not immediately apply; here, stabilization terms are required [17]. Further, only one solution of a discrete Laplacian-type subproblem is needed by PCD while LSC requires two. We also note that we have not considered boundary conditions for A_p and F_p (or N_p) within PCD. A more in depth look at the commutator can shed light on this area but is beyond the scope of this work; see [20]. Such considerations do not explicitly arise for the LSC preconditioner but can nonetheless be important.

In the single-phase case, mesh-independent convergence rates have been observed for both preconditioners. Eigenvalue bounds can be found in [44] and while for PCD they are h-independent, for LSC the known bounds depend on h. Norm equivalence of the PCD preconditioner is shown in [37] and, further, rigorous GMRES convergence bounds based on the field of values are given for a mass matrix Schur complement approximation; see also bounds in [4]. These field of values bounds are h-independent. For PCD and LSC, all known bounds depend on the Reynolds number. Numerical experiments in [20] show iteration counts for both methods that are mildly dependent on Re. Our results in the next section will demonstrate that the favorable properties of these preconditioners for the single-phase case are retained in the two-phase case.

4. Numerical results for a simplified problem. In this section we present results exhibiting the behavior of our new preconditioners for a benchmark 2-dimensional (2D) test problem, that of lid driven cavity flow. We suppose $\Omega = (-1,1)^2$ with $\Omega_2 = (-\frac{1}{2}, \frac{1}{2})^2$, and thus $\Omega_1 = \Omega \setminus \overline{\Omega}_2$, and consider a regularized cavity with the prescribed flow

$$(48) \qquad \boldsymbol{u} = \left(\begin{array}{c} 1 - x^4 \\ 0 \end{array} \right) \quad \text{for} \quad y = 1, \ -1 \le x \le 1,$$

along the lid while no flow boundary conditions are imposed on the remaining three edges. Note that this configuration of fluids is similar to that in the SINKER problem used in [39, 51]. We suppose that no external body forces are acting and neglect surface tension. In the governing linearized equations of (7)–(8) we suppose for simplicity that $\boldsymbol{r} = \boldsymbol{0}$ so that in the time-dependent case we start from an initially nonmoving flow. Further, we solve the Stokes problem (with $\boldsymbol{w} = \boldsymbol{0}$) using the generalized Cahouet–Chabard preconditioner (31) to provide the initial guess for the Picard iteration.

The computations are implemented within MATLAB using the software package IFISS [49, 19], which we have adapted to incorporate spatially varying density and viscosity, though note there is no capability to update the phases in time. As such we consider solving for a single time step or else the more challenging case of a steady problem. Results for a fully dynamic two-phase problem are presented in section 5.

Numerical results provided in this section tabulate the number of iterations that GMRES requires for solving the linear system (15) to a given tolerance. The GMRES Krylov method is used (without restarting) with the preconditioner (17) in which the inverse Schur complement approximations of two-phase PCD (37) and two-phase LSC (47) are used. We present average GMRES iteration counts over the course of the Picard iteration, omitting the initial Stokes solve. Since our focus is on the quality of the Schur complement approximation, we suppose that the approximation of F is exact, that is, $\widehat{F}^{-1} = F^{-1}$ as given by a direct solver ("backslash" in MATLAB).

4.1. Convergence and termination. To measure the convergence we look at the residual vector for the linear system given in (15) to solve for the $(k+1)$th Picard iterate. The linear residual is

$$(49) \qquad \mathbf{r}^{(\ell)} = \begin{pmatrix} \frac{\alpha}{\Delta t} M^{(\rho)} + N^{(\rho)}(\boldsymbol{w}_h^{(k)}) + A^{(\mu)} & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u}^{(\ell)} \\ \mathbf{p}^{(\ell)} \end{pmatrix} - \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix},$$

where $\mathbf{u}^{(\ell)}$ and $\mathbf{p}^{(\ell)}$ are the approximate solutions to the system after ℓ iterations of preconditioned GMRES. The notation of $N^{(\rho)}(\boldsymbol{w}_h^{(k)})$ is used here to recall that the convective part depends on the solution $\boldsymbol{w}_h^{(k)}$ of the previous Picard iteration. Denoting the discrete velocity and pressure vectors of the kth Picard iterate by $\mathbf{w}^{(k)}$ and $\mathbf{q}^{(k)}$, respectively, we also define the nonlinear residual as

$$(50) \qquad \mathbf{s}^{(k)} = \begin{pmatrix} \frac{\alpha}{\Delta t} M^{(\rho)} + N^{(\rho)}(\boldsymbol{w}_h^{(k)}) + A^{(\mu)} & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \mathbf{w}^{(k)} \\ \mathbf{q}^{(k)} \end{pmatrix} - \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix}.$$

After the kth Picard iteration, we terminate the GMRES iteration for solving the linearized system given in (15) to give the $(k + 1)$th Picard iterate once the relative residual norm decreases below a prescribed tolerance ε, that is, we terminate once

$$(51) \qquad\qquad\qquad \|\mathbf{r}^{(\ell)}\|_2 \le \varepsilon \|\mathbf{s}^{(k)}\|_2.$$

For each inner application of GMRES we use an initial guess of the past solution $\mathbf{u}^{(0)} = \mathbf{w}^{(k)}$, $\mathbf{p}^{(0)} = \mathbf{q}^{(k)}$ and use a tolerance of $\varepsilon = 10^{-6}$ for termination. This method of tracking the convergence is the default used by IFISS. The Picard iteration is terminated upon a relative reduction of 10^{-5} for the nonlinear residual, as in [20]. Further, at each Picard iteration, the implementation is such that the linear system is solved for a correction to the current iterate (see [20, section 8.3]).

4.2. Choice of mixed finite elements. For our computations we consider three choices of mixed finite element pairings. In each case, we will utilize a regular grid of square elements having side length h, thus $4h^{-2}$ elements in total. First, we use the inf–sup stable pairing of $\boldsymbol{Q}_2$–$\boldsymbol{Q}_1$ elements. In this case no stabilization is needed ($C = 0$) and since the pressure approximation is continuous we use (32) and (38) in the formulation of two-phase PCD.

Second, we employ equal order $\boldsymbol{Q}_1$–$\boldsymbol{Q}_1$ elements. The pressure space remains the same but now stabilization is required and we follow the approach developed in [14]. As noted in [17], the lack of stability of $\boldsymbol{Q}_1$–$\boldsymbol{Q}_1$ approximation stems from the mismatch between the discrete divergence of the velocity field and the discrete pressure space $\boldsymbol{Q}_1$. The approach of [14] is to project the pressure into the appropriate space. This requires the L^2 projection from the pressure space into $\boldsymbol{P}_0$, which we denote by Π_0. It is defined locally on each element $\square$ through a local averaging:

$$(52) \qquad\qquad\qquad \Pi_0 p_h|_\square = \frac{1}{|\square|} \int_\square p_h \qquad \text{for all } \square \in \mathcal{T}_h,$$

where $\mathcal{T}_h$ is the set of $\boldsymbol{Q}_1$ elements used and $|\square|$ denotes the area of element $\square$. For the Navier–Stokes problem, to ensure commensurate scaling, the resulting stabilization matrix is scaled by the inverse of viscosity, as in [17]. With variable viscosity the appropriate stabilization term is then given by

$$(53) \qquad c(p_h, q_h) = \sum_{\square \in \mathcal{T}_h} \int_\square \mu^{-1}(p_h - \Pi_0 p_h)|_\square (q_h - \Pi_0 q_h)|_\square.$$

from which the associated stabilization matrix C in (15) can be constructed. This matrix can be assembled from element contributions in the standard manner.

Finally, we consider a finite element pairing which uses a discontinuous pressure approximation. For this we choose $\boldsymbol{Q}_2-\boldsymbol{P}_{-1}$ elements, which are naturally inf–sup stable and conserve mass elementwise. Two-phase PCD requires the construction of both a Laplacian-type matrix $A_p^{(1/\rho)}$ and a convection matrix $N_p^{(1)}$ on the pressure space but now definitions (32) and (38) no longer apply. To construct appropriate operators when the pressure is discontinuous across elements, we do so locally on elements and utilize edge contributions. The construction is more involved with the main ideas being detailed in [20, section 9.2.1] and a proof of concept implementation being found within IFISS [49]. As such, we do not provide a full description here but note that the only difference in our case is the inverse density scaling within the term $A_p^{(1/\rho)}$. For this we have found it beneficial to take an arithmetic average of the two density values in the edge contributions when density differs across an element edge. Hence, for the pressure degree of freedom p_0 at the centroid of a square element $\square$, the corresponding approximation to the Laplacian-type term $A_p^{(1/\rho)}$ is given by (cf. [20, equations (9.29)–(9.30)])

$$(54) \qquad \begin{aligned} \nabla \cdot \left(\rho^{-1} \nabla p \right)|_\square &\approx \frac{1}{h} \left(\frac{2}{\rho_{+x} + \rho_0} \frac{p_{+x} - p_0}{h} - \frac{2}{\rho_0 + \rho_{-x}} \frac{p_0 - p_{-x}}{h} \right) \\ &+ \frac{1}{h} \left(\frac{2}{\rho_{+y} + \rho_0} \frac{p_{+y} - p_0}{h} - \frac{2}{\rho_0 + \rho_{-y}} \frac{p_0 - p_{-y}}{h} \right), \end{aligned}$$

where the subscripts denote values from the neighboring elements: for example, p_{+x} refers to the pressure degree of freedom at the centroid of the neighboring element in the positive x direction while ρ_{+x} refers to the density in this element. Here, we have assumed for simplicity that the interface lies along edges of the elements used.

4.3. Preconditioner performance results. We first note that the numerical experiments presented in this section have also been conducted using ideal variants of the preconditioners, that is, using direct solvers for any Laplacian-type and mass matrix subproblems. However, since the resulting iteration counts are very similar, differing by at most a small number of iterations, we omit them here and justify the following choices. When requiring the action of $\widehat{S}^{-1}$, the solutions of the scaled Laplacian-type subproblems are given approximately using one algebraic multigrid (AMG) V-cycle using a Ruge and Stüben implementation, specifically the HSL_MI20 code [6] built into IFISS. For the PCD method (37), the scaled mass matrix solutions are approximated using three steps of Chebyshev semi-iteration [57]. We present the majority of our results using $\boldsymbol{Q}_2-\boldsymbol{Q}_1$ elements to demonstrate the behavior of the preconditioners over differing parameters. We only consider this discretization for two-phase LSC as additional stabilization is required when using $\boldsymbol{Q}_1-\boldsymbol{Q}_1$ elements and, further, we note that qualitatively similar results were found for $\boldsymbol{Q}_2-\boldsymbol{P}_{-1}$ elements and thus these results are omitted for brevity.

TABLE 1

Preconditioned GMRES iterations using two-phase PCD (37)/two-phase LSC (47) for the Q_2–Q_1 solution to the steady lid driven cavity problem with $Re = 100$, $h = 1/128$, and varying density ratio $\widehat{\rho} = \rho_2/\rho_1$ and viscosity ratio $\widehat{\mu} = \mu_2/\mu_1$. Values marked $\star$ are omitted since here the dominating Reynolds number changes, being associated with the second phase.

$\widehat{\mu}$	$\widehat{\rho}$						
	10^{-3}	10^{-2}	10^{-1}	1	10	100	1000
10^{-3}	30 / 59	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$
10^{-2}	29 / 58	30 / 58	$\star$	$\star$	$\star$	$\star$	$\star$
10^{-1}	24 / 54	24 / 54	25 / 54	$\star$	$\star$	$\star$	$\star$
1	19 / 38	19 / 38	19 / 38	20 / 40	$\star$	$\star$	$\star$
10	24 / 44	24 / 44	24 / 44	24 / 44	27 / 44	$\star$	$\star$
100	26 / 38	26 / 38	26 / 38	27 / 38	27 / 38	29 / 37	$\star$
1000	26 / 36	26 / 36	26 / 36	27 / 36	27 / 36	27 / 34	28 / 33

TABLE 2

Preconditioned GMRES iterations using original (single-phase) PCD (25)/LSC (44) for the Q_2–Q_1 solution to the steady lid driven cavity problem with $Re = 100$, $h = 1/128$, and varying density ratio $\widehat{\rho} = \rho_2/\rho_1$ and viscosity ratio $\widehat{\mu} = \mu_2/\mu_1$. Values marked $\star$ are omitted since here the dominating Reynolds number changes, being associated with the second phase.

$\widehat{\mu}$	$\widehat{\rho}$						
	10^{-3}	10^{-2}	10^{-1}	1	10	100	1000
10^{-3}	173 / 261	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$
10^{-2}	104 / 74	101 / 74	$\star$	$\star$	$\star$	$\star$	$\star$
10^{-1}	46 / 38	46 / 38	45 / 38	$\star$	$\star$	$\star$	$\star$
1	29 / 37	29 / 37	29 / 37	30 / 39	$\star$	$\star$	$\star$
10	36 / 36	36 / 36	36 / 36	36 / 36	43 / 36	$\star$	$\star$
100	76 / 81	76 / 81	76 / 81	76 / 81	76 / 80	88 / 71	$\star$
1000	125 / 320	125 / 320	125 / 320	125 / 320	123 / 322	120 / 310	140 / 296

To start, we consider steady flow ($\alpha = 0$) and show how the performance of each preconditioner depends on density and viscosity. Table 1 displays results for two-phase PCD/two-phase LSC for a variety of density ratios $\widehat{\rho} = \rho_2/\rho_1$ and viscosity ratios $\widehat{\mu} = \mu_2/\mu_1$ with given Reynolds number $Re = 100$ and grid size $h = 1/128$. Note that we omit values marked "$\star$" since here the dominating Reynolds number changes—dependence on Re is given in Table 4. We see that, while there is some variation with the density and, in particular, viscosity ratios, the performance is fairly robust across the range of ratios tested. This is not true of the original PCD or LSC preconditioners for this problem; these methods give results which depend significantly on the density and viscosity ratios, and poor mesh dependence away from ratios close to unity. Equivalent values to those in Table 1 reach into several hundreds of iterations for the original methods, as exhibited in Table 2. We remark that the discrepancy between the two-phase PCD and original PCD methods for the case of $\widehat{\rho} = 1$ and $\widehat{\mu} = 1$ is due to the additional factor of two present in the two-phase PCD method, stemming from the choice of viscous term used, as discussed in subsection 3.1.

TABLE 3

The total number of degrees of freedom (DOFs) in the lid driven cavity problem for the differing grid sizes h and choice of mixed finite elements. For $\boldsymbol{Q}_2$–$\boldsymbol{Q}_1$ elements this is given by the formula $2(4h^{-1}-1)^2 + (2h^{-1}+1)^2$. For $\boldsymbol{Q}_2$–$\boldsymbol{P}_{-1}$ elements it is given by $2(4h^{-1}-1)^2 + 3(2h^{-1})^2$ while for $\boldsymbol{Q}_1$–$\boldsymbol{Q}_1$ elements it is $2(2h^{-1}-1)^2 + (2h^{-1}+1)^2$. Note that in each row of the table the number of velocity degrees of freedom is the same.

h	$\boldsymbol{Q}_2$–$\boldsymbol{Q}_1$	$\boldsymbol{Q}_2$–$\boldsymbol{P}_{-1}$	h	$\boldsymbol{Q}_1$–$\boldsymbol{Q}_1$
$\frac{1}{16}$	9,027	11,010	$\frac{1}{32}$	12,163
$\frac{1}{32}$	36,483	44,546	$\frac{1}{64}$	48,899
$\frac{1}{64}$	146,691	179,202	$\frac{1}{128}$	196,099
$\frac{1}{128}$	588,291	718,850	$\frac{1}{256}$	785,411
$\frac{1}{256}$	2,356,227	2,879,490	$\frac{1}{512}$	3,143,683

We note here that some variation is also seen in the results for the generalized Cahouet–Chabard preconditioner in [41]; in particular, for large ratios in opposite directions for density and viscosity the method becomes much less efficient. Results provided in [26], for an augmented Lagrangian approach to solve the Navier–Stokes equations with variable viscosity, also depend on the viscosity through the minimum and maximum values taken in the domain.

Before continuing to give results for varying grid sizes we detail the scope of the overall problem sizes that we use for the comparisons in this work. For each choice of mixed finite elements we consider a sequence of five grids. The total number of degrees of freedom (unknowns in the problem) for each linear system solve are given in Table 3, including a one-dimensional null space related to a constant pressure mode, thus our constraint on the pressure in (9) to specify it uniquely. We note that on the finest grids the number of degrees of freedom reaches into several million unknowns.

The dependence of the iteration counts on Reynolds number and grid size is given in Table 4 for the example of $\widehat{\rho} = 1.2 \times 10^{-3}$ and $\widehat{\mu} = 1.8 \times 10^{-2}$, corresponding to ratios in an air–water system. As for the original PCD and LSC preconditioners, there is a mild dependence on Re, with higher Reynolds number flows requiring more iterations. These results are comparable with [20, Table 9.5], where the grid sizes used in Table 4 correspond to grid parameters $l = 6, \ldots, 10$ of [20]. (Note that in [20] "old" refers to the original preconditioners without adjustments for boundary conditions, a topic not pursued here as we choose to focus on the key aspects of generalizing PCD and LSC to two-phase flow. Further, the operators in PCD act in reverse order; see [20, Remark 9.3].) Our results for two-phase PCD display nearly mesh-independent behavior akin to the original method applied to a single fluid, with only a very small increase in iterations required for finer grids. For the highest Reynolds number flows, we note a slight jump in iterations required for the finest grid and this is attributed to the fact that new features of these flows are resolved using this grid; this will also be seen in further results. Two-phase LSC shows more dependence on h but the increase per refinement is not too great and again performance is similar to that of the original method (without boundary adjustments) applied to single-phase flow. We remark that the trends shown in Table 4 are also seen for other choices of density and viscosity ratios. We further note that when a larger number of iterations is required, such as with high Reynolds number flows, the variation seen in iteration counts for different density and viscosity ratios also increases relatively, as might be expected.

TABLE 4

Preconditioned GMRES iterations using two-phase PCD (37)/two-phase LSC (47)/LSC$_D$ (45) for the $\boldsymbol{Q}_2$–$\boldsymbol{Q}_1$ solution to the steady lid driven cavity problem with density ratio $\widehat{\rho} = 1.2 \times 10^{-3}$, viscosity ratio $\widehat{\mu} = 1.8 \times 10^{-2}$ (air–water values), and varying Reynolds number Re and grid size h.

h	Re				
	10	$10^{1.5}$	100	$10^{2.5}$	1000
$\frac{1}{16}$	17 / 15 / 28	20 / 19 / 30	24 / 23 / 32	28 / 27 / 37	37 / 37 / 51
$\frac{1}{32}$	19 / 18 / 54	21 / 20 / 57	25 / 28 / 63	29 / 34 / 73	35 / 30 / 89
$\frac{1}{64}$	19 / 24 / 91	22 / 26 / 95	27 / 41 / 103	32 / 33 / 117	36 / 46 / 148
$\frac{1}{128}$	20 / 48 / 145	23 / 51 / 154	28 / 56 / 166	33 / 62 / 192	38 / 67 / 244
$\frac{1}{256}$	20 / 69 / 243	24 / 71 / 250	29 / 81 / 267	36 / 91 / 308	44 / 99 / 396

TABLE 5

Preconditioned GMRES iterations using two-phase PCD (37) for the $\boldsymbol{Q}_1$–$\boldsymbol{Q}_1$ and $\boldsymbol{Q}_2$–$\boldsymbol{P}_{-1}$ solution to the steady lid driven cavity problem with density ratio $\widehat{\rho} = 1.2 \times 10^{-3}$, viscosity ratio $\widehat{\mu} = 1.8 \times 10^{-2}$ (air–water values), and varying Reynolds number Re and grid size h.

$\boldsymbol{Q}_1$–$\boldsymbol{Q}_1$ elements

h	Re				
	10	$10^{1.5}$	100	$10^{2.5}$	1000
$\frac{1}{32}$	16	18	22	27	44
$\frac{1}{64}$	16	19	24	29	34
$\frac{1}{128}$	17	20	25	30	35
$\frac{1}{256}$	17	20	25	32	37
$\frac{1}{512}$	17	20	26	33	43

$\boldsymbol{Q}_2$–$\boldsymbol{P}_{-1}$ elements

h	Re				
	10	$10^{1.5}$	100	$10^{2.5}$	1000
$\frac{1}{16}$	17	21	27	38	73
$\frac{1}{32}$	18	21	27	34	55
$\frac{1}{64}$	18	21	27	35	40
$\frac{1}{128}$	18	21	27	34	42
$\frac{1}{256}$	18	21	27	36	49

We mention here that our experience with the LSC variant known as LSC$_D$ [39, 51] shows a significant dependence on the grid size h. This is demonstrated in Table 4, which allows a direct comparison between LSC$_D$ and our two-phase LSC approach. In particular, on the finest grid, LSC$_D$ requires more than triple the number of iterations required by our two-phase LSC approach. On finer meshes we expect this difference only to increase due to the poor scalability of LSC$_D$. We also note here that, while our two-phase PCD and, to a much lesser extent, LSC preconditioners show nearly mesh-independent behavior, it may be that boundary adjustments must be made to achieve full mesh independence, as is necessary for the original preconditioners [20].

Before moving to unsteady flow, we provide analogous results for two-phase PCD when utilizing our other choices of finite element pairings. Table 5 provides results for both (stabilized) $\boldsymbol{Q}_1$–$\boldsymbol{Q}_1$ elements and $\boldsymbol{Q}_2$–$\boldsymbol{P}_{-1}$ elements. We see similar behavior with respect to mesh refinement and Reynolds number as with $\boldsymbol{Q}_2$–$\boldsymbol{Q}_1$ (similar trends are also found in other studies not shown). The performance of two-phase PCD for $\boldsymbol{Q}_2$–$\boldsymbol{P}_{-1}$ is particularly notable for being mesh independent, aside from the large Reynolds number cases. We note that, in such large Reynolds number cases, the iteration counts initially decrease; this is attributed to the fact that on coarse meshes the flow is not well resolved or captured by the discrete operators used.

We now consider the time-dependent case ($\alpha = 1$). We remark that our numerical tests give similar dependence on the density and viscosity ratios, as well as on the

TABLE 6

Preconditioned GMRES iterations using two-phase PCD (37)/two-phase LSC (47) for the Q_2-Q_1 solution to the lid driven cavity problem with density ratio $\hat{\rho} = 1.2 \times 10^{-3}$, viscosity ratio $\hat{\mu} = 1.8 \times 10^{-2}$ (air–water values), $h = 1/128$, and varying Reynolds number Re and time step Δt.

Δt	Re				
	10	$10^{1.5}$	100	$10^{2.5}$	1000
10	20 / 47	23 / 51	27 / 55	32 / 59	36 / 63
1	19 / 46	21 / 47	23 / 45	24 / 45	25 / 41
$\frac{1}{10}$	16 / 39	16 / 37	16 / 35	15 / 32	16 / 28

grid size, to the steady case. The performance with Reynolds number is linked to the time step, as illustrated in Table 6 using Q_2-Q_1 elements. For a large time step the behavior is similar to that of the steady case. However, once the time step is small enough, we see that the trend in Reynolds number flattens and then reverses so that higher Reynolds number flows require similar or fewer iterations for convergence. As anticipated, when solving with a smaller time step fewer iterations are required since the time-stepping term becomes dominant, pushing the eigenvalues of the linear system away from zero and thus allowing faster convergence of the iterative solver.

The results of this section for a simplified two-phase test problem suggest that our new preconditioners perform reasonably well across a wide range of parameters and extend the utility of the PCD and LSC methodologies to two-phase flow. For the case of enclosed flow (without considering boundary adjustments) two-phase PCD appears favorable, showing nearly mesh-independent behavior and only requiring a single Laplacian-type subproblem to be solved per application of the Schur complement preconditioner. We now focus on the two-phase PCD approach. To gain further insight we consider a more challenging and realistic test problem within a framework which incorporates additional features pertinent to simulations of two-phase flow.

5. Numerical results for a dynamic dam-break problem. Free-surface models that accurately describe complicated air–water flow dynamics are an important application of the two-phase Navier–Stokes equations. For example, level set and VOF methods can be combined with two-phase Navier–Stokes equations to simulate intricate hydraulic processes such as waves crashing into coastal structures. For these multiphysics models, the Navier–Stokes equations are just one component in a larger system of equations [31]. In this section, we apply the two-phase PCD preconditioner in a free-surface model to gauge its effectiveness for dynamic multiphysics problems. In nearly all cases, solving the discrete linearized Navier–Stokes equations is the most time consuming part of a multiphysics simulation, so superior preconditioners can dramatically improve computational run times. As a result, the effectiveness of a preconditioner in a dynamic setting is very relevant to practitioners.

In the following study, we use the RANS2P module of the Proteus computational methods and simulation toolkit (http://proteustoolkit.org). RANS2P is a fluid dynamics software module developed at the U.S. Army Corps of Engineers for solving large air–water free-surface problems that arise in coastal and hydraulic applications. For a detailed description of the two-phase Navier–Stokes model within the RANS2P module, see [3].

As with other dynamic free-surface models, the two-phase Navier–Stokes model in RANS2P has several key characteristics that affect preconditioner performance. First,

the dynamic nature of the model problems involved typically requires taking short time steps. Consequently, the large temporal component of the problem diminishes the effect of the advective terms at each time step, making the system easier to solve than a steady-state problem. A second feature of such free-surface models is that they often simulate high Reynolds number flows. As such, in order to produce meaningful results, we must modify the Navier–Stokes equations with numerical stabilization terms. RANS2P uses a variation of algebraic sub-grid scale (ASGS) stabilization for the advection and pressure. Further details are available in [3, 28, 11, 50].

Before moving forward, we say a few words about applying $\widehat{F}^{-1}$ (recall (17)). For large problems, it is important that the action of $\widehat{F}^{-1}$ is scalable and provides a reliable approximation to F^{-1}. Thus, developing effective methods to approximate F^{-1} is an active area of our research. For advection dominated flows, however, finding an effective and robust $\widehat{F}^{-1}$ can be difficult [42]. Further complicating matters is the ASGS stabilization used and the dynamic two-phase nature of the simulations. We have implemented an approach which treats $\widehat{F}^{-1}$ as a block preconditioned GMRES iteration where the subblocks are approximated using multigrid methods and this has shown signs of being effective both in our work and others [26]; see also results in [2]. Unfortunately, maintaining solver robustness throughout an entire multiphysics simulation remains an issue. As such, we defer discussion of this important topic to future work and focus our efforts here on the Schur complement approximation, taking $\widehat{F}^{-1}$ to be given by the sparse direct solver SuperLU_DIST [35, 34].

We now comment on some implementation details of the PCD approximation used in RANS2P. First, one component of the Navier–Stokes stabilization is a shock capturing diffusion term. Thus, to better reflect the simulation viscosity, the viscosity used within the definition of two-phase PCD is the sum of the shock capturing term and the physical viscosity. Second, we use diagonal lumped mass matrices within two-phase PCD. Finally, even though the RANS2P module uses Newton iteration to solve the Navier–Stokes equations, rather than Picard iteration, the form of the PCD approximation does not change (see [20]). For additional details see [3].

As a basis for comparison, we consider the SIMPLE method (see [20, 53]) where

$$(55) \qquad \widehat{S}^{-1} = (B \operatorname{diag}(F)^{-1} B^T + C)^{-1}.$$

For steady problems, SIMPLE is not a competitive alternative to two-phase PCD since the $\operatorname{diag}(F)$ term does not capture well the advective features of the flow. However, for dynamic problems with short time steps, the method is effective in many cases.

Neither SIMPLE nor the two-phase PCD method offers a clear implementation advantage over the other. Indeed, at each time step, three (sparse) discrete operators must be constructed for two-phase PCD. In contrast, SIMPLE (55) is algebraic and constructed using blocks taken directly from the global linear system. For sufficiently large problems, however, the matrix–matrix product used in the construction of (55) is expensive. In terms of applying the approximations, two-phase PCD requires one V-cycle to solve the pressure Laplacian-type term $A_p^{(1/\rho)}$, a vector–scalar product, two diagonal matrix–vector products, a sparse matrix–vector product, and two vector sums. By comparison, applying SIMPLE only requires a single multigrid V-cycle. Our experience, however, suggests that SIMPLE requires a more accurate multigrid solver than two-phase PCD to serve as a reliable preconditioner in the RANS2P module. In each approach we make use of BoomerAMG from HYPRE [21] for the V-cycle.

In the following examples we use right preconditioned GMRES to compare the SIMPLE and two-phase PCD methods. This ensures the convergence of the linear

TABLE 7

Preconditioned GMRES iterations using two-phase PCD (37) / SIMPLE (55) for the $\boldsymbol{P}_1$–$\boldsymbol{P}_1$ solution to the steady lid driven cavity problem with density ratio $\widehat{\rho} = 1.2 \times 10^{-3}$, viscosity ratio $\widehat{\mu} = 1.8 \times 10^{-2}$ (air–water values), and varying Reynolds number Re and mesh size h. The total number of degrees of freedom (DOFs) for each problem size is also listed for comparison with Table 3.

h	DOFs	Re				
		10	$10^{1.5}$	100	$10^{2.5}$	1000
$\frac{1}{16}$	5,115	25 / 18	27 / 18	29 / 23	36 / 26	49 / 25
$\frac{1}{32}$	20,085	24 / 38	26 / 23	29 / 32	35 / 36	47 / 39
$\frac{1}{64}$	79,152	25 / 56	27 / 55	29 / 56	35 / 45	46 / 56
$\frac{1}{128}$	308,613	24 / 53	27 / 85	29 / 77	34 / 83	46 / 84
$\frac{1}{256}$	1,191,810	24 / 116	27 / 116	30 / 105	35 / 112	47 / 115

solver is measured using nonpreconditioned residuals. In RANS2P, our experience suggests that nonpreconditioned residuals produce more stable results and provide a reliable basis for comparing different preconditioners.

For dynamic free-surface simulations, the linear solver termination criterion used depends on the nonlinear solver tolerance. We always use a relative residual tolerance of 10^{-6} for the linear solver. However, we also include an absolute linear solver tolerance that is one order of magnitude smaller than the absolute nonlinear solver tolerance. For example, if the absolute tolerance of the nonlinear solver is 10^{-7}, then the linear solver will include an absolute tolerance of 10^{-8} for termination. This approach maintains nonlinear solver convergence while limiting unnecessary work in the linear solver.

Before examining preconditioner performance for a complete multiphysics model, we first present results using (unstructured) $\boldsymbol{P}_1$–$\boldsymbol{P}_1$ elements for the steady lid driven cavity problem using the RANS2P module. Note that we measure the mesh refinement here using $h = \sqrt{2T_A}$, where T_A is the area of the largest triangle. Table 7 shows analogous results to Table 4 for the two-phase PCD and SIMPLE methods. We see that the SIMPLE method fails to produce scalable results, with iteration counts increasing as we refine the mesh. In contrast, the two-phase PCD approach exhibits mesh-independent behavior similar to the $\boldsymbol{Q}_1$–$\boldsymbol{Q}_1$ element results from IFISS in Table 5. It is also worth mentioning that the iteration level of the $\boldsymbol{P}_1$–$\boldsymbol{P}_1$ elements is higher than the $\boldsymbol{Q}_1$–$\boldsymbol{Q}_1$ elements because different stabilization methods are used.

Next we consider the 2D dam-break benchmark problem described in [12, 58], which requires using the complete multiphysics RANS2P model. The domain is rectangular with $\Omega = (0, 3.22) \times (0, 1.8)$, and the free-slip condition (5) is applied everywhere on the boundary $\partial\Omega$. Initially, there is a standing column of water in $\Omega_1 = (0, 1.2) \times (0, 0.6)$ with the remaining space being air. The simulation runs for a two second time interval and begins as the column of water collapses under gravity and proceeds to collide with the right-hand wall of the tank. This collision creates a wave and ultimately topological changes in the phases. Figure 1 displays several snapshots of the simulation. The dam-break problem provides a good benchmark for testing the two-phase PCD and SIMPLE preconditioners because its features are typical of many dynamic, multiphysics problems of practical interest.

To compare the scaling performance of two-phase PCD and SIMPLE, we consider two simulations. In the first, time steps are selected to ensure that the CFL number
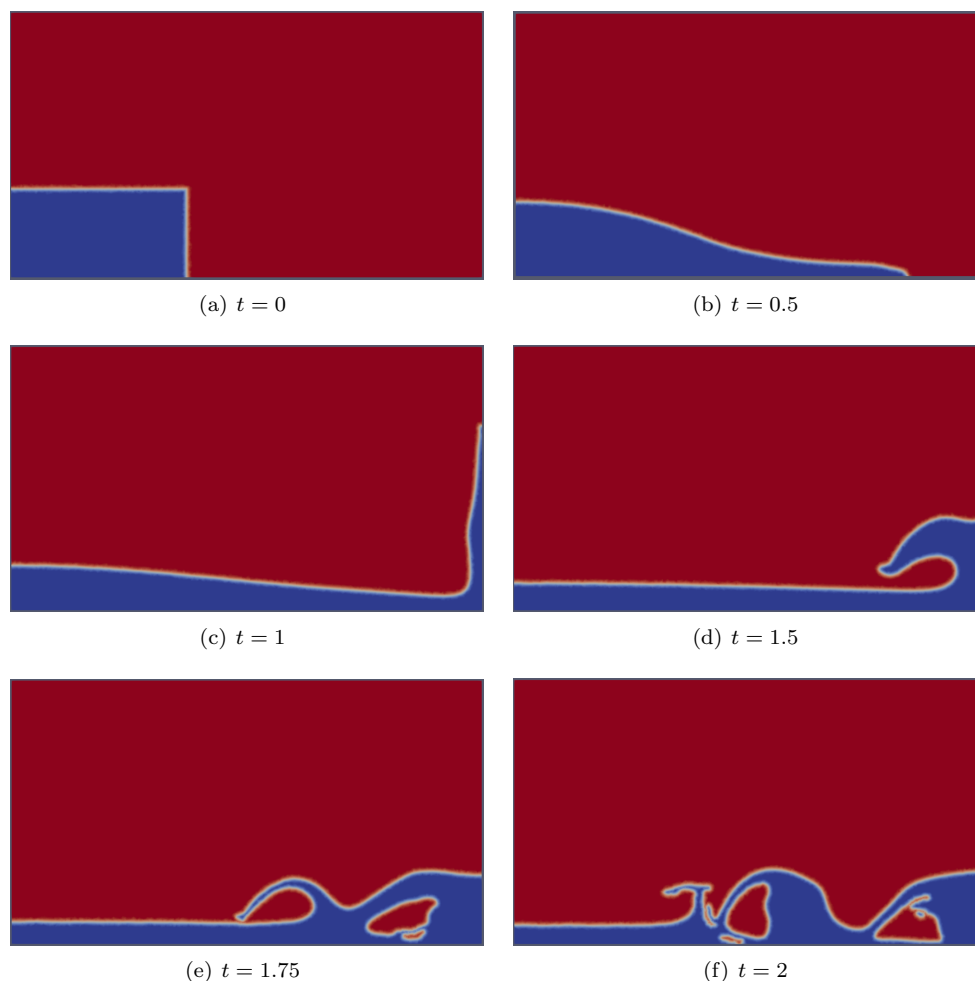
(a) $t = 0$

(b) $t = 0.5$

(c) $t = 1$

(d) $t = 1.5$

(e) $t = 1.75$

(f) $t = 2$

Fig. 1. *Evolution of the dam-break simulation in Proteus at selected points in time. The VOF is plotted with blue representing the water phase and red being the air.*

is less than or equal to 0.9. Such restrictions are often necessary for nonlinear solver convergence and solution accuracy. However, in some important cases this restriction on the CFL number is not strictly necessary.[2] Thus, in the second simulation, we use a fixed time step of $\Delta t = 0.01$. In this case, the CFL number is larger than one for much of the simulation, reaching a maximum of 20.5 and typically being above 2.5. Nonetheless, the time step is still small enough to achieve nonlinear solver convergence and solution accuracy. For both simulations, we analyze the average and maximum number of GMRES iterations required at five different levels of mesh refinement.

---

[2] Accurate computation of relevant quantities of interest, such as drag force, for fixed hydraulic structures or vessels, frequently results in quasi-steady flows. In particular, the free surface may tend towards a steady wake structure or standing wave pattern, and this structure dominates the force on the given structure. In these cases, it is frequently desirable to use a fixed time step that results in CFL numbers significantly larger than one. Time stepping is then carried out until the quasi-steady hydrodynamic conditions are reached or the quantity of interest has reached a constant or steady periodic value.

Table 8

The average/maximum number of GMRES iterations and simulation run times (in minutes) required across different meshes when running the dam-break problem ensuring the CFL number is less than or equal to 0.9.

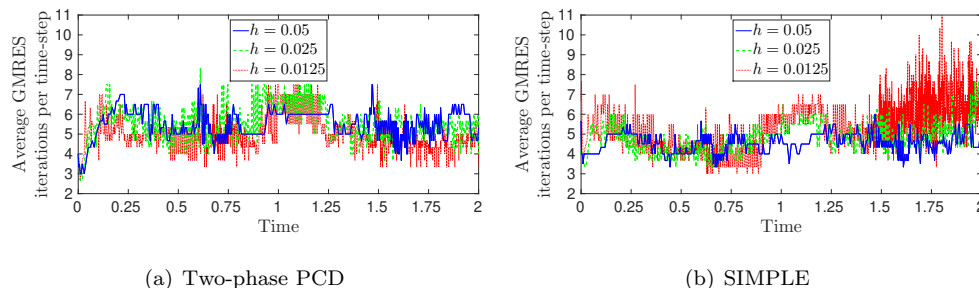|	$h = 0.2$	$h = 0.1$	$h = 0.05$	$h = 0.025$	$h = 0.0125$
Two-phase PCD	5 / 8 (0.5)	5 / 9 (0.6)	5 / 10 (2.6)	5 / 11 (14.7)	5 / 10 (126.0)
SIMPLE	4 / 10 (0.4)	4 / 10 (0.6)	4 / 10 (2.4)	5 / 13 (15.5)	5 / 17 (140.1)

(a) Two-phase PCD   (b) SIMPLE

Fig. 2. *Average preconditioned GMRES iterations per time step for the two-phase PCD and SIMPLE preconditioners when the CFL number is less than or equal to* 0.9.

These mesh refinement levels are selected so that, by the final refinement, the physics of the simulation is sufficiently resolved to perform relevant engineering analysis. The dam-break timings were collected using 8 cores of a dedicated 2.3-GHz Intel Xeon Haswell processor with 128 GBytes of DDR4 memory on the Topaz supercomputer in the Department of Defense High Performance Computing Modernization Program.

Table 8 presents the average and maximum number of GMRES iterations taken during the first simulation with a restricted CFL number. These results suggest that, on average, the SIMPLE and two-phase PCD preconditioners both scale well with the mesh size. However, Table 8 also reveals that the maximum number of iterations required by the SIMPLE method increases as the mesh is refined. As seen in Figure 2, the increase in maximum iterations of the SIMPLE preconditioner occurs as the air and water phases begin to undergo topological changes around one and a half seconds into the simulation. Indeed, as the water phase reconnects with itself, it generates a pressure that causes the air phase to accelerate and increase the advective features of the simulation. Since SIMPLE only uses the diagonal elements of the matrix F, it appears unable to fully capture these additional advection dynamics. In contrast, the two-phase PCD preconditioner scales well during this mixing phase of the simulation.

Table 8 also reveals that, on the most refined meshes, the two-phase PCD method is faster than the SIMPLE approach. One reason for this is that the two-phase PCD preconditioner requires fewer GMRES iterations than the SIMPLE preconditioner during the mixing phase of the simulation. A second reason is that the AMG method applied within the SIMPLE preconditioner requires more computational effort than the AMG method used in the two-phase PCD preconditioner. Finally, the two-phase PCD method tends to exceed the linear solver threshold by a larger margin than the SIMPLE approach, leading to slightly smaller residual norms in the nonlinear solver. Interestingly, this difference slightly reduces the computational effort needed to solve other components in the full RANS2P model.

TABLE 9

The average/maximum number of GMRES iterations and simulation run times (in minutes) required across different meshes when running the dam-break problem with fixed $\Delta t = 0.01$. Note that $\times$ indicates the simulation stopped due to failure in the solver convergence.

	$h = 0.2$	$h = 0.1$	$h = 0.05$	$h = 0.025$	$h = 0.0125$
Two-phase PCD	4 / 8 (0.5)	5 / 9 (0.6)	8 / 14 (1.6)	11 / 25 (5.8)	14 / 34 (26.5)
SIMPLE	4 / 10 (0.4)	4 / 10 (0.6)	5 / 10 (1.5)	10 / 32 (6.6)	$\times$

Results for the second simulation, using a fixed time step $\Delta t = 0.01$, are shown in Table 9 and suggest that, on coarse meshes, SIMPLE and the two-phase PCD preconditioner are competitive with one another. In contrast to the first simulation, however, as the mesh is refined, the performance of the SIMPLE method rapidly deteriorates while the two-phase PCD preconditioner remains relatively stable, with iteration counts increasing only modestly. Again, two-phase PCD is preferable based on the timing results. These results are consistent with the steady-state performance observed above in Table 7. As the mesh is refined for a fixed time step, the advective features of the system become more pronounced and the CFL number increases. As observed for the steady lid driven cavity problem, the SIMPLE approach does not capture the features of an advection dominated flow well enough to provide a robust preconditioner. The two-phase PCD preconditioner, however, does account for such features and thus remains capable of producing stable, reliable results in this setting.

Overall, our results for the two-phase PCD preconditioner in a free-surface, multiphysics setting are encouraging. When a restricted CFL number is used, the two-phase PCD preconditioner slightly outperforms the SIMPLE method both in terms of reducing the number of GMRES iterations required as well as delivering faster run times. As the CFL number of the flow increases, two-phase PCD demonstrates a significant improvement over the SIMPLE method due to its superior steady-state performance. Together, these results suggest that the two-phase PCD approach can be effectively used as an approximation to the inverse Schur complement in coupled free-surface problems.

6. Conclusions. The application of PCD and LSC preconditioning techniques for the Navier–Stokes equations has proved to be an effective approach for computing flows of a single Newtonian phase. In this work we have generalized the formulation of these preconditioners to the case of the variable coefficient Navier–Stokes equations arising in models of two-phase incompressible flow. For the PCD method this requires a new form of the preconditioner, a point made clear through the relation of PCD with the Cahouet–Chabard preconditioner. While a variant of LSC has previously been proposed for variable viscosity Stokes flow, we present a superior method for Navier–Stokes flow which directly appeals to the relevant commutator. Our new two-phase PCD and LSC preconditioners retain the favorable features of the original preconditioners and exhibit similar performance in our tests on a simplified enclosed flow problem. As for the original methods, additional consideration of boundary conditions will be necessary to gain fully mesh-independent behavior, in particular for the case of flow problems with inflow and outflow boundary conditions when using the two-phase PCD preconditioner [20]. We intend to investigate this in future work.

Our results for a fully dynamic dam-break test problem showed that the two-phase PCD approach can be effectively used in coupled free-surface problems. Moreover, since it captures sufficient information about the advective term, it provides a stable

preconditioner when larger time steps are permissible. Nonetheless, further work is needed to investigate whether the additional terms added to the discrete equations can be incorporated within the two-phase PCD preconditioner to improve performance. Moreover, an important part of the block preconditioners which we consider here is a scalable approximation to the velocity block F. This topic requires further attention for variable coefficient Navier–Stokes equations and is an active area of investigation. Additionally, in practical applications, simulations of two-phase flow run in a high performance computing environment with a parallel implementation; it remains to be seen how efficient the preconditioners we have proposed are within this framework. We are currently exploring the parallel scalability of our approach on larger problems of interest.

Finally, we note that we did not require the two-phase nature explicitly in the construction of the preconditioners, thus the techniques proposed here might also be investigated for their utility when more general forms of variable density and variable viscosity flows are proposed.

REFERENCES

[1] I. AKKERMAN, Y. BAZILEVS, C. E. KEES, AND M. W. FARTHING, *Isogeometric analysis of free-surface flow*, J. Comput. Phys., 230 (2011), pp. 4137–4152.

[2] O. AXELSSON, X. HE, AND M. NEYTCHEVA, *Numerical solution of the time-dependent Navier–Stokes equation for variable density–variable viscosity. Part I*, Math. Model. Anal., 20 (2015), pp. 232–260.

[3] A. BENTLEY, N. BOOTLAND, A. WATHEN, AND C. KEES, *Implementation Details of the Level Set Two-Phase Navier–Stokes Equations in Proteus*, Technical report TR2017-10, Department of Mathematical Sciences, Clemson University, 2017, http://www.clemson.edu/science/departments/math-stat/about/technical-reports.html.

[4] 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 (2011), pp. 770–788.

[5] M. BENZI, M. A. OLSHANSKII, AND Z. WANG, *Modified augmented Lagrangian preconditioners for the incompressible Navier–Stokes equations*, Internat. J. Numer. Methods Fluids, 66 (2011), pp. 486–508.

[6] J. BOYLE, M. MIHAJLOVIĆ, AND J. SCOTT, *HSL_MI20: An efficient AMG preconditioner for finite element problems in 3D*, Internat. J. Numer. Methods Engrg., 82 (2010), pp. 64–98.

[7] J. U. BRACKBILL, D. B. KOTHE, AND C. ZEMACH, *A continuum method for modeling surface tension*, J. Comput. Phys., 100 (1992), pp. 335–354.

[8] C. BURSTEDDE, O. GHATTAS, G. STADLER, T. TU, AND L. C. WILCOX, *Parallel scalable adjoint-based adaptive solution of variable-viscosity Stokes flow problems*, Comput. Methods Appl. Mech. Engrg., 198 (2009), pp. 1691–1700.

[9] J. CAHOUET AND J.-P. CHABARD, *Some fast 3D finite element solvers for the generalized Stokes problem*, Internat. J. Numer. Methods Fluids, 8 (1988), pp. 869–895.

[10] M. CAI, A. NONAKA, J. B. BELL, B. E. GRIFFITH, AND A. DONEV, *Efficient variable-coefficient finite-volume Stokes solvers*, Commun. Comput. Phys., 16 (2014), pp. 1263–1297.

[11] R. CODINA, *A stabilized finite element method for generalized stationary incompressible flows*, Comput. Methods Appl. Mech. Engrg., 190 (2001), pp. 2681–2706.

[12] A. COLLAGROSSI AND M. LANDRINI, *Numerical simulation of interfacial flows by smoothed particle hydrodynamics*, J. Comput. Phys., 191 (2003), pp. 448–475.

[13] E. C. CYR, J. N. SHADID, AND R. S. TUMINARO, *Stabilization and scalable block preconditioning for the Navier–Stokes equations*, J. Comput. Phys., 231 (2012), pp. 345–363.

[14] C. R. DOHRMANN AND P. B. BOCHEV, *A stabilized finite element method for the Stokes problem based on polynomial pressure projections*, Internat. J. Numer. Methods Fluids, 46 (2004), pp. 183–201.

[15] H. ELMAN, V. E. HOWLE, J. SHADID, R. SHUTTLEWORTH, AND R. TUMINARO, *Block preconditioners based on approximate commutators*, SIAM J. Sci. Comput., 27 (2006), pp. 1651–1668.

[16] H. ELMAN, V. E. HOWLE, J. SHADID, R. SHUTTLEWORTH, AND R. TUMINARO, *A taxonomy and comparison of parallel block multi-level preconditioners for the incompressible Navier–Stokes equations*, J. Comput. Phys., 227 (2008), pp. 1790–1808.

[17] H. ELMAN, V. E. HOWLE, J. SHADID, D. SILVESTER, AND R. TUMINARO, *Least squares preconditioners for stabilized discretizations of the Navier–Stokes equations*, SIAM J. Sci. Comput., 30 (2007), pp. 290–311.

[18] H. C. ELMAN, *Preconditioning for the steady-state Navier–Stokes equations with low viscosity*, SIAM J. Sci. Comput., 20 (1999), pp. 1299–1316.

[19] H. C. ELMAN, A. RAMAGE, AND D. J. SILVESTER, *Algorithm* 866: *IFISS, a Matlab toolbox for modelling incompressible flow*, ACM Trans. Math. Software, 33 (2007), 14.

[20] H. C. ELMAN, D. J. SILVESTER, AND A. J. WATHEN, *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics*, 2nd ed., Oxford University Press, Oxford, 2014.

[21] R. D. FALGOUT AND U. M. YANG, *hypre: A library of high performance preconditioners*, in Computational Science – ICCS 2002, Lecture Notes in Comput. Sci. 2331, P. M. A. Sloot, A. G. Hoekstra, C. J. K. Tan, and J. J. Dongarra, eds., Springer, Berlin, 2002, pp. 632–641.

[22] P. P. GRINEVICH AND M. A. OLSHANSKII, *An iterative method for the Stokes-type problem with variable viscosity*, SIAM J. Sci. Comput., 31 (2009), pp. 3959–3978.

[23] S. GROSS, V. REICHELT, AND A. REUSKEN, *A finite element based level set method for two-phase incompressible flows*, Comput. Vis. Sci., 9 (2006), pp. 239–257.

[24] J. L. GUERMOND, P. MINEV, AND J. SHEN, *An overview of projection methods for incompressible flows*, Comput. Methods Appl. Mech. Engrg., 195 (2006), pp. 6011–6045.

[25] F. H. HARLOW AND J. E. WELCH, *Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface*, Phys. Fluids, 8 (1965), pp. 2182–2189.

[26] X. HE AND M. NEYTCHEVA, *Preconditioning the incompressible Navier–Stokes equations with variable viscosity*, J. Comput. Math., 30 (2012), pp. 461–482.

[27] X. HE, M. NEYTCHEVA, AND C. VUIK, *On Preconditioning Incompressible Non-Newtonian Flow Problems*, Technical report 13-07, Reports of the Delft Institute of Applied Mathematics, Delft University of Technology, 2013.

[28] T. J. R. HUGHES, *Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods*, Comput. Methods Appl. Mech. Engrg., 127 (1995), pp. 387–401.

[29] I. C. F. IPSEN, *A note on preconditioning nonsymmetric matrices*, SIAM J. Sci. Comput., 23 (2001), pp. 1050–1051.

[30] D. KAY, D. LOGHIN, AND A. WATHEN, *A preconditioner for the steady-state Navier–Stokes equations*, SIAM J. Sci. Comput., 24 (2002), pp. 237–256.

[31] C. E. KEES, I. AKKERMAN, M. W. FARTHING, AND Y. BAZILEVS, *A conservative level set method suitable for variable-order approximations and unstructured meshes*, J. Comput. Phys., 230 (2011), pp. 4536–4558.

[32] A. KLAWONN AND G. STARKE, *Block triangular preconditioners for nonsymmetric saddle point problems: Field-of-values analysis*, Numer. Math., 81 (1999), pp. 577–594.

[33] I. N. KONSHIN, M. A. OLSHANSKII, AND Y. V. VASSILEVSKI, *ILU preconditioners for nonsymmetric saddle-point matrices with application to the incompressible Navier–Stokes equations*, SIAM J. Sci. Comput., 37 (2015), pp. A2171–A2197.

[34] X. S. LI AND J. W. DEMMEL, *SuperLU_DIST: A scalable distributed-memory sparse direct solver for unsymmetric linear systems*, ACM Trans. Math. Software, 29 (2003), pp. 110–140.

[35] X. S. LI, J. W. DEMMEL, J. R. GILBERT, L. GRIGORI, M. SHAO, AND I. YAMAZAKI, *SuperLU Users' Guide*, Technical report LBNL-44289, Lawrence Berkeley National Laboratory, Berkeley, CA, 1999. http://crd.lbl.gov/~xiaoye/SuperLU/.

[36] P. T. LIN, M. SALA, J. N. SHADID, AND R. S. TUMINARO, *Performance of fully coupled algebraic multilevel domain decomposition preconditioners for incompressible flow and transport*, Internat. J. Numer. Methods Engrg., 67 (2006), pp. 208–225.

[37] D. LOGHIN AND A. J. WATHEN, *Analysis of preconditioners for saddle-point problems*, SIAM J. Sci. Comput., 25 (2004), pp. 2029–2049.

[38] D. A. MAY, J. BROWN, AND L. LE POURHIET, *A scalable, matrix-free multigrid preconditioner for finite element discretizations of heterogeneous Stokes flow*, Comput. Methods Appl. Mech. Engrg., 290 (2015), pp. 492–523.

[39] D. A. MAY AND L. MORESI, *Preconditioned iterative methods for Stokes flow problems arising in computational geodynamics*, Phys. Earth Planet. Interiors, 171 (2008), pp. 33–47.

[40] M. F. MURPHY, G. H. GOLUB, AND A. J. WATHEN, *A note on preconditioning for indefinite linear systems*, SIAM J. Sci. Comput., 21 (2000), pp. 1969–1972.

[41] M. A. OLSHANSKII, J. PETERS, AND A. REUSKEN, *Uniform preconditioners for a parameter dependent saddle point problem with application to generalized Stokes interface equations*, Numer. Math., 105 (2006), pp. 159–191.

[42] M. A. OLSHANSKII AND A. REUSKEN, *Convergence analysis of a multigrid method for a convection-dominated model problem*, SIAM J. Numer. Anal., 42 (2004), pp. 1261–1291.

[43] M. A. OLSHANSKII AND A. REUSKEN, *Analysis of a Stokes interface problem*, Numer. Math., 103 (2006), pp. 129–149.

[44] M. A. OLSHANSKII AND Y. V. VASSILEVSKI, *Pressure Schur complement preconditioners for the discrete Oseen problem*, SIAM J. Sci. Comput., 29 (2007), pp. 2686–2704.

[45] A. QUARTERONI AND A. VALLI, *Domain Decomposition Methods for Partial Differential Equations*, Oxford University Press, Oxford, 1999.

[46] M. RUDMAN, *Volume-tracking methods for interfacial flow calculations*, Internat. J. Numer. Methods Fluids, 24 (1997), pp. 671–691.

[47] Y. SAAD AND M. H. SCHULTZ, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 856–869.

[48] D. SILVESTER, H. ELMAN, D. KAY, AND A. WATHEN, *Efficient preconditioning of the linearized Navier–Stokes equations for incompressible flow*, J. Comput. Appl. Math., 128 (2001), pp. 261–279.

[49] D. J. SILVESTER, H. C. ELMAN, AND A. RAMAGE, *Incompressible Flow and Iterative Solver Software (IFISS), Version* 3.3, 2013, http://www.manchester.ac.uk/ifiss/.

[50] T. E. TEZDUYAR, *Stabilized finite element formulations for incompressible flow computations*, Adv. Appl. Math., 28 (1992), pp. 1–44.

[51] M. UR REHMAN, T. GEENEN, C. VUIK, G. SEGAL, AND S. P. MACLACHLAN, *On iterative methods for the incompressible Stokes problem*, Internat. J. Numer. Methods Fluids, 65 (2011), pp. 1180–1200.

[52] M. UR REHMAN, C. VUIK, AND G. SEGAL, *A comparison of preconditioners for incompressible Navier–Stokes solvers*, Internat. J. Numer. Methods Fluids, 57 (2008), pp. 1731–1751.

[53] M. UR REHMAN, C. VUIK, AND G. SEGAL, *SIMPLE-type preconditioners for the Oseen problem*, Internat. J. Numer. Methods Fluids, 61 (2009), pp. 432–452.

[54] S. P. VANKA, *Block-implicit multigrid solution of Navier–Stokes equations in primative variables*, J. Comput. Phys., 65 (1986), pp. 138–158.

[55] M. WANG AND L. CHEN, *Multigrid methods for the Stokes equations using distributive Gauss–Seidel relaxations based on the least squares commutator*, J. Sci. Comput., 56 (2013), pp. 409–431.

[56] A. J. WATHEN, *Realistic eigenvalue bounds for the Galerkin mass matrix*, IMA J. Numer. Anal., 7 (1987), pp. 449–457.

[57] A. J. WATHEN AND T. REES, *Chebyshev semi-iteration in preconditioning for problems including the mass matrix*, Electron. Trans. Numer. Anal., 34 (2009), pp. 125–135.

[58] Z. Q. ZHOU, J. O. DE KAT, AND B. BUCHNER, *A nonlinear 3-D approach to simulate green water dynamics on deck*, in Proceedings of the 7th International Conference on Numerical Ship Hydrodyn, J. Piquet, ed., Nantes, Ft. Belvoir OTIC, Fairfax County, VA, 1999, 5.