NUMERICAL LINEAR ALGEBRA WITH APPLICATIONS
Numer. Linear Algebra Appl. 0000; **00**:1–1
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nla

Block preconditioners for finite element discretization of incompressible flow with thermal convection

Victoria E. Howle[1]* and Robert C. Kirby[1]

[1] *Department of Mathematics and Statistics; Texas Tech University; MS 1042; Lubbock, TX 79409-1042*

SUMMARY

We derive block preconditioners for a finite element discretization of incompressible flow coupled to heat transport by the Boussinesq approximation. Our techniques rely on effectively approximating the Schur complement obtained by eliminating the fluid variables to obtain an equation for temperature alone. Additionally, the method utilizes existing block-structured preconditioners and multilevel methods for the Navier—Stokes equations and scalar convection-diffusion. We find that the preconditioner remains robust and scalable even when the subsolves are applied quite inexactly. Copyright © 0000 John Wiley & Sons, Ltd.

Received ...

KEY WORDS: Finite element, incompressible flow, Bénard convection, multiphysics, block preconditioner.

1. INTRODUCTION

In this paper, we extend block preconditioners for inf-sup stable finite element discretizations of the incompressible Navier–Stokes equations,

$$\begin{aligned} -\nu\Delta u + u\cdot\nabla u + \nabla p &= f \\ \nabla\cdot u &= 0, \end{aligned} \tag{1}$$

posed on some domain $\Omega \subset \mathbb{R}^d$ for $d = 2, 3$ and equipped with appropriate boundary conditions, to a particular non-dimensionalization of a coupled fluid-thermal problem, Bénard convection:

$$\begin{aligned} -\Delta u + u\cdot\nabla u + \nabla p &= -\frac{Ra}{Pr}\hat{g}T \\ \nabla\cdot u &= 0 \\ -\frac{1}{Pr}\Delta T + u\cdot\nabla T &= 0, \end{aligned} \tag{2}$$

again posed on some domain Ω along with boundary conditions. The fluid velocity and pressure are u and p, respectively, in both equations, and the temperature is T. The ν in (1) is the fluid viscosity. The Rayleigh number Ra measures the ratio of energy from buoyant forces to viscous

*Correspondence to: Department of Mathematics and Statistics; Texas Tech University; MS 1042; Lubbock, TX 79409-1042. Email: victoria.howle@ttu.edu

Contract/grant sponsor: National Science Foundation; contract/grant number: 0830655

dissipation and heat conduction, the Prandtl number Pr measures the ratio of viscosity to heat conduction, and $\hat{g}$ denotes a unit vector along the axis in which gravity acts. This model employs the Boussinesq approximation, in which temperature-dependent density variations are assumed to affect the momentum balance only through a buoyant force. For more information, including the non-dimensionalization used in (2), we follow the treatment of Carey and Oden [6].

While variations of this problem are quite classical in the fluid mechanics literature, it still is attracting interest among solver research as a challenging model problem for coupled physics. For example, it is considered as an example for pseudo-transient continuation in [7] and for AMG-type preconditioners in [18].

2. DISCRETIZATION AND LINEARIZATION

We will consider inf-sup stable $P^2 - P^1$ Taylor-Hood discretizations of the fluid equations, and P^1 approximation of the temperature. We let V_h be the vector-valued P^2 velocity space and W_h the scalar-valued P^1 space. The weak formulation of the incompressible Navier–Stokes equations is to find $(u_h, p_h) \in V_h \times W_h$ such that

$$\begin{aligned} \nu\left(\nabla u_h, \nabla v_h\right) + \left(u_h \cdot \nabla u_h, v_h\right) - \left(p_h, \nabla \cdot v_h\right) &= \left(f, v_h\right) \\ \left(\nabla \cdot u_h, w_h\right) &= 0, \end{aligned} \tag{3}$$

for all $(v_h, w_h) \in V_h \times W_h$ satisfying the homogeneous boundary conditions.

The thermal convection problem admits the weak form

$$\begin{aligned} \left(\nabla u_h, \nabla v_h\right) + \left(u_h \cdot \nabla u_h, v_h\right) - \left(p_h, \nabla \cdot v_h\right) &= \frac{Ra}{Pr}\left(T_h, v_{h,g}\right) \\ \left(\nabla \cdot u_h, w_h\right) &= 0 \\ \frac{1}{Pr}\left(\nabla T_h, \nabla r_h\right) + \left(u_h \cdot \nabla T_h, r_h\right) &= 0, \end{aligned} \tag{4}$$

where $v_{h,g}$ means the component of v_h in the direction of gravity.

In fact, our numerical simulations will use a more complicated weak form based on Nitsche boundary conditions [21, 13, 1]. Our numerical calculations are performed using the Sundance library [19], but the standard row-replacement Dirichlet boundary conditions disturb the block structure. Nitsche boundary conditions leave the block structure intact. For simplicity of exposition we use standard Dirichlet boundary conditions in this paper, although the formulation naturally carries over to Nitsche-type conditions.

We will primarily be interested in Newton-type algorithms for the nonlinear system of equations, which require the Jacobian operators. These block systems have a tighter coupling between the fluid and temperature equations, giving rise to a more challenging preconditioning problem than successively iterating between the two equations. We linearize at the level of the weak form, giving rise to a linear variational problem for the Newton step for Navier–Stokes. The weak forms of these Jacobians are

$$\begin{aligned} \nu\left(\nabla u_h, \nabla v_h\right) + \left(u_h^0 \cdot \nabla u_h, v_h\right) + \left(u_h \cdot \nabla u_h^0, v_h\right) - \left(p_h, \nabla \cdot v_h\right) &= \left(f, v_h\right) \\ \left(\nabla \cdot u_h, w_h\right) &= 0, \end{aligned} \tag{5}$$

for Navier–Stokes and

$$\begin{aligned} \left(\nabla u_h, \nabla v_h\right) + \left(u_h^0 \cdot \nabla u_h, v_h\right) + \left(u_h \cdot \nabla u_h^0, v_h\right) - \left(p_h, \nabla \cdot v_h\right) &= \frac{Ra}{Pr}\left(T_h, v_{h,g}\right) \\ \left(\nabla \cdot u_h, w_h\right) &= 0 \\ \frac{1}{Pr}\left(\nabla T_h, \nabla r_h\right) + \left(u_h^0 \cdot \nabla T_h, r_h\right) + \left(u_h \cdot \nabla T_h^0, r_h\right) &= 0, \end{aligned} \tag{6}$$

for convection, where T_h^0 and u_h^0 are the temperature and velocity about which the system is linearized.

DOI: 10.1002/nla

The linear algebraic problem for a single Newton step for Navier–Stokes takes the form

$$\begin{pmatrix} F & B^t \\ -B & 0 \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}, \tag{7}$$

where B and B^t are matrices corresponding to discrete divergence and gradient operators, and F operates on the discrete velocity space. If an equal-order stabilized method is used [4], the bottom right block will be nonzero. Typically, preconditioners can be appropriately modified to account for this; see, for example, [10].

If we define the block vectors $x = \begin{pmatrix} u \\ p \end{pmatrix}$ and $g = \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}$, then we may write the block system in the simpler form

$$Nx = g, \tag{8}$$

where N represents the block 2×2 Navier–Stokes system (7). We will use this later when deriving methods for the convection problem when we wish to treat the linearized Navier–Stokes equations as a single entity.

The linear system for a Newton step for the convection problem takes the form

$$\begin{pmatrix} F & B^t & M_1 \\ -B & 0 & 0 \\ M_2 & 0 & K \end{pmatrix} \begin{pmatrix} u \\ p \\ T \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix}. \tag{9}$$

The matrices F and B are the same as in linearized Navier–Stokes. The matrix M_1 arises from the term $(\frac{Ra}{Pr}T, v_g)$, where g is the Cartesian direction in which gravity acts (y in 2d, z in 3d). In 2d, if the velocity variables in each direction are segregated, this has the form of $M_1^t = (0, M^t)$, where M is a rectangular mass matrix with rows corresponding to the P^2 basis functions and columns to P^1. The matrix M_2 arises from the Jacobian term $(u \cdot \nabla T_0, r)$, where T_0 is the temperature in the current Newton iterate. The matrix K comes from $\frac{1}{Pr}(\nabla T, \nabla r) + (u_0 \cdot \nabla T, r)$, so is a standard linear convection-diffusion operator. In our simulations, the fluid velocity has not become large enough to require additional stabilization of this term, although it would not complicate the block structure if such terms were included.

If we cluster the velocity and pressure unknowns as in (8), the convection system becomes

$$\begin{pmatrix} N & \widetilde{M_1} \\ \widetilde{M_2} & K \end{pmatrix} \begin{pmatrix} x \\ T \end{pmatrix} = \begin{pmatrix} g \\ f_3 \end{pmatrix}, \tag{10}$$

where $\widetilde{M_1} = \begin{pmatrix} M_1 \\ 0 \end{pmatrix}$ and $\widetilde{M_2} = (M_2, 0)$.

3. BLOCK PRECONDITIONERS FOR THE THERMAL-TEMPERATURE SYSTEM

As with Navier–Stokes [9] and other coupled systems [23], we begin our discussion with a block LU factorization of (10):

$$\begin{pmatrix} N & \widetilde{M_1} \\ \widetilde{M_2} & K \end{pmatrix} = \begin{pmatrix} I & 0 \\ \widetilde{M_2}N^{-1} & I \end{pmatrix} \begin{pmatrix} N & \widetilde{M_1} \\ 0 & S_T \end{pmatrix}, \tag{11}$$

where $S_T \equiv K - \widetilde{M_2}N^{-1}M_1$ is the Schur complement obtained by eliminating the fluid variables. We denoting the block matrix on the left as A and the right-hand side as LU. The matrix $L = AU^{-1}$ is unit lower triangular, where we invert U to obtain

$$U^{-1} = \begin{pmatrix} N^{-1} & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} I & -\widetilde{M_1} \\ 0 & I \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & S_T^{-1} \end{pmatrix}. \tag{12}$$

DOI: 10.1002/nla

GMRES for the preconditioned system AU^{-1} will converge in exactly two (very expensive) iterations [20]. While this iteration is typically impractical, it motivates preconditioners based on approximations to the Schur complement. To our knowledge, our work is the first to use this approach with approximate Schur complements for Bénard convection. Our insight comes from examining the operator-level theory, so we look at the weak forms giving rise to the particular blocks more closely.

N comes from the weak form of the linearized Navier–Stokes equations

$$\langle N(u,p),(v,w)\rangle = (\nabla u, \nabla v) + (u_0 \cdot \nabla u, v) + (u \cdot \nabla u_0, v) - (p, \nabla \cdot v) + (\nabla \cdot u, w).$$

To be more precise, we might write $N = N_{u_0}$, for the operator depends on the state u_0 about which the problem is linearized. This is a linear second-order elliptic operator, bounded from $(H^1)^d \times L^2_0$ into its dual. It also possesses elliptic regularity, so that its inverse maps

The block $\widetilde{M_1}$ is given by

$$\langle \widetilde{M_1}(u,p), r\rangle = (u \cdot \nabla c_0, r),$$

which gives rise to a mass-like matrix on vector-valued functions, albeit with a possibly discontinuous coefficient ∇c_0. So, with the Riesz Representation Theorem, it is a bounded linear operator on L^2.

The block $\widetilde{M_2}$ takes any L^2 function and computes

$$\langle \widetilde{M_2} c, (v,w)\rangle = -\frac{Ra}{Pr}(c\mathbf{g}, v),$$

the (scaled) L^2 inner product of one component of v with an L^2 or H^1 function c. This is likewise a bounded operator.

Finally, the block $K : H^1_D \to \left(H^1_D\right)'$ is given by

$$\langle Kc, r\rangle = \frac{1}{Pr}(\nabla c, \nabla r) + (u_0 \cdot \nabla c, r),$$

which is a linear, second-order elliptic operator possessing full elliptic regularity.

Now, we consider the Schur complement $S_T = K - \widetilde{M_1} N^{-1} \widetilde{M_2}$ as an operator on H^1 functions satisfying the temperature boundary conditions. K takes such functions into H^{-1} and $\widetilde{M_1} N^{-1} \widetilde{M_2}$ into L^2; the Schur complement is a well-defined elliptic integro-differential operator. Motivated by techniques in [12] and [17], we consider using a differential operator as a preconditioner. In particular, we consider K. As a right preconditioner, we have

$$\left(K - \widetilde{M_1} N^{-1} \widetilde{M_2}\right) K^{-1} = I - \widetilde{M_1} N^{-1} \widetilde{M_2} K^{-1}.$$

$\widetilde{M_1}$, $\widetilde{M_2}$ are bounded operators on L^2. Moreover, the regularity of K and N means that K^{-1} and N^{-1} are compact operators mapping from from L^2-type spaces into smoother ones compactly embedded in L^2. This is due to regularity (which requires basic assumptions on the smoothness of the boundary and input data) and the Rellich selection theorem [11].

This preconditioned operator, then, takes the form of a compact perturbation of the identity. From the classic theory of compact operators, we know that the eigenvalues of $S_T K^{-1}$ have finite multiplicity (nontrivial in infinite dimensions) and have 1 as the only accumulation point. Krylov methods in Hilbert space are known to converge superlinearly on such operators [5, 24]. Although we do not have rigorous estimates when we restrict to the finite-dimensional setting, we do find favorable convergence properties in practice.

So then, we propose the preconditioner $P^{-1} \approx U^{-1}$ by substituting our approximation $S_T^{-1} \approx K^{-1}$ into (12)

$$P^{-1} = \begin{pmatrix} N^{-1} & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} I & -\widetilde{M_1} \\ 0 & I \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & K^{-1} \end{pmatrix}. \tag{13}$$

This preconditioner still requires the solution of a linearized Navier–Stokes system and a scalar convection-diffusion system. Each could be applied inexactly (and we study this numerically later), or we could make an even further approximation by replacing the N^{-1} with a preconditioner for the Navier–Stokes system, to which we now turn.

DOI: 10.1002/nla

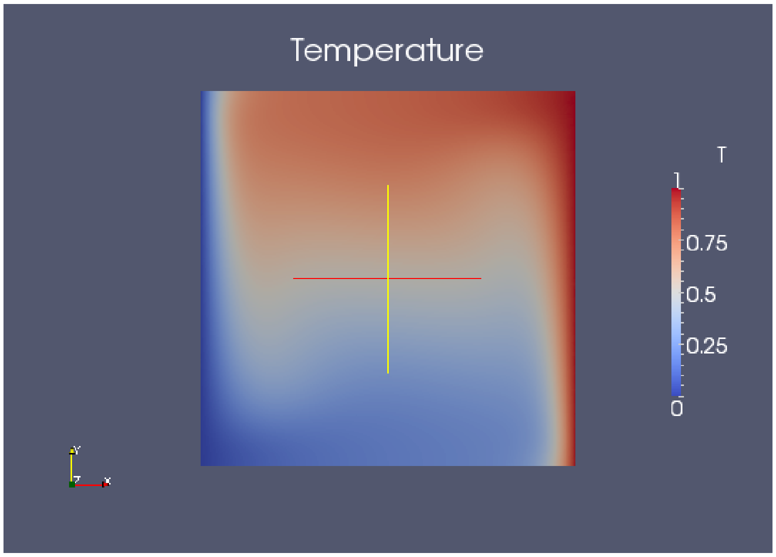

Figure 1. Temperature profile of Bénard convection problem on a 64×64 mesh using Taylor-Hood finite elements with $Ra = 2 \times 10^5$ and $Pr = 1$. No-flow velocity boundary conditions are imposed, and the temperature is set to be 1 on the right boundary and 0 on the left boundary, with insulated boundary conditions on the top and bottom. The nonlinear algebraic equations were solved to a Newton tolerance of 10^{-6}.

3.1. Further approximation: a 3×3 block preconditioner

Rather than an exact (or inexact) application of N^{-1}, it is also possible to simply use a Navier–Stokes preconditioner. If that preconditioner is itself block-structured, we obtain a 3×3 system. If we consider approximate Schur complement methods of the form in [10], the first block matrix on the right hand side of (12) expands into three matrices, and we expand the block of fluid variables into separate blocks for pressure and velocity:

$$\left[\begin{array}{cc|c} F^{-1} & 0 & 0 \\ 0 & I & 0 \\ \hline 0 & 0 & I \end{array}\right] \left[\begin{array}{cc|c} I & -B^t & 0 \\ 0 & I & 0 \\ \hline 0 & 0 & I \end{array}\right] \left[\begin{array}{cc|c} I & 0 & 0 \\ 0 & X^{-1} & 0 \\ \hline 0 & 0 & I \end{array}\right] \left[\begin{array}{cc|c} I & 0 & -M_1 \\ 0 & I & 0 \\ \hline 0 & 0 & I \end{array}\right] \left[\begin{array}{cc|c} I & 0 & 0 \\ 0 & I & 0 \\ \hline 0 & 0 & K^{-1} \end{array}\right]. \tag{14}$$

Here, X can be any of the pressure Schur complement approximations for Navier–Stokes [9]. If other block preconditioners, such as those based on augmented Lagrangian methods [2, 3], are considered, the first three matrices will be modified accordingly.

4. NUMERICAL RESULTS

In this section, we examine our various solution methodologies on a problem that is a slight modification of the classic two-dimensional problem. In that problem, fluid is in a box with no-slip boundary conditions for the fluid. A unit temperature difference is imposed in the vertical direction, and insulating boundary conditions are applied on the remaining sides. At large Rayleigh numbers, this creates an instability leading to overturning cells. In practice, this is a rather subtle problem to simulate, requiring a very good initial guess for Newton's method. The problem of imposing the temperature difference horizontally gives an easier nonlinear system. Our algorithm, leads to the temperature profile shown in Figure 1.

4.1. The 3×3 block preconditioner

We begin with the preconditioner (14), which uses the Navier–Stokes preconditioner in place of a preconditioned Navier–Stokes solve. We use the pressure convection-diffusion (PCD) preconditioner [8], where the Schur complement for Navier–Stokes is approximated with $X = \widetilde{S_p} = A_p F_p^{-1} M_p$, as the Navier–Stokes preconditioner. Here A_p is a Laplacian operator on the pressure space, F_p is a convection-diffusion operator on the pressure space, and M_p is a pressure mass matrix. Table I displays the number of Newton steps and the average number of GMRES iterations per Newton step for this method over a range Rayleigh numbers and problem sizes. We used a Newton tolerance of 1×10^{-6} with an inner GMRES tolerance of 1×10^{-8} on the relative residual. For the inner subsolves, we use GMRES preconditioned with algebraic multigrid on K and CG preconditioned with algebraic multigrid on A_p and M_p. All inner solves have a stopping tolerance on the relative residual of 1×10^{-10}. As we are using the Sundance library, which is part of the Trilinos framework [16], for our numerical calculations, we also use Trilinos packages for the linear systems that arise. Here we use the iterative solver package Belos [25] for the main GMRES iteration. For the PCD subsolves F, A_p, and M_p, and for the K subsolve, as proof of concept we use direct solves through the Trilinos package Amesos [22]. In terms of impact on the number of GMRES iterations (as opposed to runtime performance), direct subsolves are the best case scenario.

We can see that the number of Newton iterations and the average number of GMRES iterations per Newton step are essentially independent with respect to problem size. The number of Newton steps is mildly dependent on Ra, and the number of GMRES iterations per Newton step grows more strongly with Ra. This is to be expected, as increasing Ra contributes to the off-diagonal blocks and also increases the flow rate and hence asymmetry of the F matrix. We take the Rayleigh number up to 2×10^4. Much above this value, an equal-order discretization with stabilization would be needed. In our experiments with $Ra = 2 \times 10^5$, only the coarsest mesh converged.

Ra	2×10^2		2×10^3		2×10^4	
N	Newt	GMRES	Newt	GMRES	Newt	GMRES
16	3	37.3	4	49.0	7	75.4
32	3	39.3	4	51.5	7	80.3
64	2	38.5	4	53.8	6	82.8

Table I. Number of Newton steps and average GMRES iteration count per Newton step when the linear system is solved using the 3×3 block preconditioner. A Newton tolerance of 1×10^{-6} with an inner GMRES tolerance of 1×10^{-8} was used. All of the inner-most systems, F, A_p, M_p, and K were solved using direct methods. The number of Newton iterations is essentially independent, with some small dependence on Ra. The average number of GMRES iterations per Newton step is also largely independent of problem size, but grows more strongly with Ra.

For this preconditioner to be efficient, we must be able to use iterative methods on the subsolves with fairly low accuracy without significantly degrading the number of GMRES iterations needed. In Tables II and III, we use GMRES preconditioned with algebraic multigrid on the inner F and K systems. For the A_p and M_p systems, we use CG preconditioned with algebraic multigrid. We use loose tolerances on these inner-most iterations of 1×10^{-4} in Table II and 1×10^{-2} in Table III. Because we are solving the inner systems inexactly, we switch to FGMRES for the main iteration. We can see that number of GMRES (FGMRES) iterations increases as we lower the tolerance on the inner solves. With inner tolerances of 1×10^{-4} the number of FGMRES iterations remains relatively independent of problem size. With inner tolerances of 1×10^{-2}, we get a little more growth in FGMRES iterations with problem size. In both tables, the problem on the finest mesh with Rayleigh number 2×10^4 failed to converge. In both cases the failure was in the inner solve on the F system, which failed to reach the respective tolerance after 1500 iterations.

DOI: 10.1002/nla

F, A_p, M_p, K tols = 10^{-4}						
Ra	2×10^2		2×10^3		2×10^4	
N	Newt	FGMRES	Newt	FGMRES	Newt	FGMRES
16	3	41.0	4	51.0	7	77.0
32	3	44.0	4	55.3	7	81.7
64	2	45.0	4	59.3	-	-

Table II. Number of Newton steps and average FGMRES iteration count per Newton step when the linear system is solved using the 3×3 block preconditioner. A Newton tolerance of 1×10^{-6} with an inner FGMRES tolerance of 1×10^{-8} was used. The inner-most systems, F, A_p, M_p, and K have tolerances of 1×10^{-4}. The number of Newton iterations is essentially independent, with some small dependence on Ra. The average number of GMRES iterations per Newton step is also largely independent of problem size, but grows more strongly with Ra. The method failed on the finest grid with $Ra = 2 \times 10^4$. At this Rayleigh number, the F solve failed to reach a tolerance of 10^{-4} after 1500 iterations.

F, A_p, M_p, K tols = 10^{-2}						
Ra	2×10^2		2×10^3		2×10^4	
N	Newt	FGMRES	Newt	FGMRES	Newt	FGMRES
16	3	85.7	4	101.3	7	128.9
32	3	118.3	4	134.8	7	157.6
64	2	143.0	4	175.3	-	-

Table III. Number of Newton steps and average FGMRES iteration count per Newton step when the linear system is solved using the 3×3 block preconditioner. A Newton tolerance of 1×10^{-6} with an inner FGMRES tolerance of 1×10^{-8} was used. The inner-most systems, F, A_p, M_p, and K have tolerances of 1×10^{-2}. The number of Newton iterations is essentially independent, with some small dependence on Ra. The average number of GMRES iterations per Newton step is also largely independent of problem size, but grows more strongly with Ra. The method failed on the finest grid with $Ra = 2 \times 10^4$. At this Rayleigh number, the F solve failed to reach a tolerance of 10^{-2} after 1500 iterations.

4.2. The 2×2 block preconditioner

Now, we turn to the nested 2×2 preconditioner (13). (This presents some practical complications, as the linear algebra has a nested block structure with the fluid block itself having subblocks. We will address the software-related issues in a separate publication.)

In Tables IV through VII, we apply the preconditioner (13) using varying degrees of accuracy on the subsolves, starting with all tight tolerances and gradually loosening the tolerances on the subsolves. These tables show that although loosening the subsolve tolerances does lead to a larger number of GMRES (FGMRES) iterations, there is no increase in the number of Newton steps. In all cases, our linear system stopping criteria tolerances are on the relative residual.

To apply the preconditioner (13) requires solving one linear system with K and one with the Navier–Stokes block N. We use GMRES right preconditioned with algebraic multigrid for the K solve. The Navier–Stokes block is solved using FGMRES right preconditioned with the PCD preconditioner. The PCD preconditioner requires linear solves on the convection-diffusion operator F, the pressure Laplacian A_p, and the pressure mass matrix M_p. We use GMRES on the F solve and CG on the A_p and M_p solves. Each of these is preconditioned with algebraic multigrid.

As before, we use Trilinos packages for the linear systems that arise. We use Belos for the main GMRES (or FGMRES) iteration. Because we need FGMRES once the subsolves have looser tolerances, for consistency, we use FGMRES throughout. Similarly, we use FGMRES from Belos for the Navier–Stokes solve. For the K solve and the linear systems that arise from the PCD preconditioner, we use the AztecOO [15] package and precondition using the algebraic multigrid with smoothed aggregation from ML [14].

We will see in Table IX, that although the looser tolerances in the subsolves lead to a larger number of FGMRES iterations, each iteration is sufficiently cheaper that the loose tolerance runs are noticeably faster than the runs with tight inner tolerances. All timing runs were performed (in serial) on an 8-core Power Mac workstation with 32 GB of RAM.

To begin, in Table IV, we consider the case of carrying out all solves to very high tolerance. This is typically suboptimal for run-time, but provides a baseline against which to compare iteration counts when inexact solves are employed. The Navier–Stokes solve has a tolerance of 1×10^{-10}. Within the PCD preconditioner, the F, A_p, and M_p solves have tolerances of 1×10^{-10}. Similarly, the K solve has a tolerance of 1×10^{-10}. Again we display results over a range of Rayleigh numbers and problem sizes. In Table IV and subsequent tables, we take the Rayleigh number up to 2×10^4. Much above this value, the iterative subsolves struggled significantly and an equal-order discretization with stabilization would be needed. Here we experience independence in both the number of Newton steps and the average number of FGMRES iterations with respect to problem size and only mild growth in each with Ra. It is worth noting the very low iteration count – this preconditioner only differs from [20] in the approximation of $S \approx K$. This result suggests that our Schur complement approximation is a quite effective starting point for further approximations, to which we now turn.

NS tol = 10^{-10}, PCD tols = 10^{-10}, K tol = 10^{-10}						
Ra	2×10^2		2×10^3		2×10^4	
N	Newt	FGMRES	Newt	FGMRES	Newt	FGMRES
16	3	3.3	4	5.3	7	7.6
32	3	3.3	4	4.8	7	7.6
64	2	3.0	4	4.8	6	6.8

Table IV. Number of Newton steps and average FGMRES iteration count per Newton step when the linear system is solved using the 2×2 block preconditioner. A Newton tolerance of 1×10^{-6} with an inner FGMRES tolerance of 1×10^{-8} was used. The Navier–Stokes solve has a tolerance of 1×10^{-10}. Within the PCD preconditioner, the F, A_p, and M_p solves have tolerances of 1×10^{-10}. Similarly, the K solve has a tolerance of 1×10^{-10}. The number of Newton iterations and the average number of iterations per Newton step are essentially independent with some dependence on Ra.

Next, in Table V, we raise the tolerance on the iteration for applying N^{-1} to 1×10^{-2}, while keeping the tolerances in the F, A_p, and M_p subsolves tight. This leads to mildly increased iteration counts in the outer linear solve, but at a much-reduced cost in applying the preconditioner. We examine these costs through timings in Tables IX and X.

NS tol = 10^{-2}, PCD tols = 10^{-10}, K tol = 10^{-10}						
Ra	2×10^2		2×10^3		2×10^4	
N	Newt	FGMRES	Newt	FGMRES	Newt	FGMRES
16	3	6.0	4	9.0	7	17.7
32	3	5.7	4	9.0	7	18.4
64	2	5.5	4	9.0	6	17.0

Table V. Number of Newton steps and average FGMRES iteration count per Newton step when the linear system is solved using the inexact 2×2 block preconditioner. A Newton tolerance of 1×10^{-6} with an inner FGMRES tolerance of 1×10^{-8} was used. The Navier–Stokes solve has a tolerance of 1×10^{-2}. Within the PCD preconditioner, the F, A_p, and M_p solves have tolerances of 1×10^{-10}. Similarly, the K solve has a tolerance of 1×10^{-10}. The number of Newton iterations and the average number of iterations per Newton step are essentially independent with some dependence on Ra.

As a further approximation, the subsolves within the Navier–Stokes preconditioner need only be applied accurately enough to achieve the desired tolerance for the Navier–Stokes equations. An

DOI: 10.1002/nla

additional performance gain is realized then by raising the F, A_p, and M_p tolerances within the PCD preconditioner. While the A_p and M_p solves are quite efficient, the F solve is the most difficult step of the entire process, so relaxing this tolerance leads to a significant gain. We can see in Table VI that raising these inner-most tolerances has almost no negative effect on either the number of Newton steps or the average number of FGMRES iterations.

NS tol = 10^{-2}, PCD tols = 10^{-4}, K tol = 10^{-10}						
Ra	2×10^2		2×10^3		2×10^4	
N	Newt	FGMRES	Newt	FGMRES	Newt	FGMRES
16	3	6.0	4	9.5	7	17.3
32	3	6.0	4	9.0	7	17.4
64	2	5.0	4	9.0	6	17.2

Table VI. Number of Newton steps and average FGMRES iteration count per Newton step when the linear system is solved using the inexact 2×2 block preconditioner. A Newton tolerance of 1×10^{-6} with an inner FGMRES tolerance of 1×10^{-8} was used. The Navier–Stokes solve has a tolerance of 1×10^{-2}. Within the PCD preconditioner, the F, A_p, and M_p solves have tolerances of 1×10^{-4}. The K solve has a tolerance of 1×10^{-10}. The number of Newton iterations and the average number of iterations per Newton step are essentially independent with some dependence on Ra, and there has been almost no change in either as compared with those in Table V.

While the N solve dominates the cost of applying the preconditioner, it is also interesting to consider the effect of raising the tolerance on the K solve, which can also be strongly nonsymmetric and present some difficulties. Table VII indicates that increasing this tolerance has little further effect on the overall iteration count, though later we see it also has but a modest effect on overall run-time.

All of the runs with the 2×2 preconditioner take significantly fewer GMRES (or FGMRES) iterations than those with the 3×3 preconditioner. However in terms of runtime efficiency, the nested 2×2 preconditioner also includes a solve on the Navier–Stokes block instead of just an application of the preconditioner. We will examine this trade-off with some timings results in the next subsection.

NS tol = 10^{-2}, PCD tols = 10^{-4}, K tol = 10^{-2}						
Ra	2×10^2		2×10^3		2×10^4	
N	Newt	FGMRES	Newt	FGMRES	Newt	FGMRES
16	3	6.3	4	9.7	7	17.4
32	3	6.3	4	9.8	7	18.0
64	2	5.5	4	9.3	6	15.3

Table VII. Number of Newton steps and average FGMRES iteration count per Newton step when the linear system is solved using the inexact 2×2 block preconditioner. A Newton tolerance of 1×10^{-6} with an inner FGMRES tolerance of 1×10^{-8} was used. The Navier–Stokes solve has a tolerance of 1×10^{-2}. Within the PCD preconditioner, the F, A_p, and M_p solves have tolerances of 1×10^{-4}. The K solve has a tolerance of 1×10^{-2}. The number of Newton iterations and the average number of iterations per Newton step are essentially independent with some dependence on Ra.

4.3. *Some timing results*

Reporting thorough timing and inner iteration results for the large array of options considered would quickly become unwieldy. Yet, it is also important to ascertain the relative effects of the various approximations and strategies we propose. To this end, in Tables VIII and IX, we consider the 3×3 and 2×2 preconditioners, respectively, in the case of $Ra = 2 \times 10^4$ and $nx = ny = 32$ and report timings for a range of approximations in the preconditioners. We can see that there is a significant performance gain in the 2×2 case as we loosen the tolerance on the Navier–Stokes solve, and an even more significant gain, for both preconditioners, as we loosen the tolerances on the subsolves in the PCD preconditioner. We also see modest performance improvements by relaxing the tolerance on the K solve.

	PCD tol = 10^{-10} K tol = 10^{-10}	PCD tol = 10^{-6} K tol = 10^{-6}	PCD tol = 10^{-4} K tol = 10^{-4}	PCD tol = 10^{-2} K tol = 10^{-2}
Newt FGMRES	7 80.3	7 80.3	7 81.7	7 157.6
Timing (sec)	922.2	493.0	297.3	231.8

Table VIII. The number of Newton iterations, the average number of iterations per Newton step, and timings (in seconds) using the 3×3 block preconditioner of (14). A Newton tolerance of 1×10^{-6} with an inner FGMRES tolerance of 1×10^{-8} was used. We used Ra = 2×10^4, and a problem size of $nx = ny = 32$. The tightest subsolves on F, A_p, M_p, and K have a tolerance of 1×10^{-10}, then successively looser tolerances of 1×10^{-6}, 1×10^{-4}, and 1×10^{-2}.

	NS tol = 10^{-10} PCD tol = 10^{-10} K tol = 10^{-10}	NS tol = 10^{-2} PCD tol = 10^{-10} K tol = 10^{-10}	NS tol = 10^{-2} PCD tol = 10^{-4} K tol = 10^{-10}	NS tol = 10^{-2} PCD tol = 10^{-4} K tol = 10^{-2}
Newt FGMRES	7 7.1	7 18.4	7 17.4	7 18.0
Timing (sec)	674	477.6	269.1	281.2

Table IX. The number of Newton iterations, the average number of iterations per Newton step, and timings (in seconds) using the 2×2 block preconditioner of (13). A Newton tolerance of 1×10^{-6} with an inner FGMRES tolerance of 1×10^{-8} was used. We used Ra = 2×10^4, and a problem size of $nx = ny = 32$. The "tight" NS FGMRES tolerance is 1×10^{-10} and the "loose" NS FGMRES is 1×10^{-2}. "Tight" subsolves on F, A_p, and M_p have a tolerance of 1×10^{-10} and "loose" subsolves have tolerances of 1×10^{-4}. "Tight" tolerance on the K solve is 1×10^{-10}, and "loose" is 1×10^{-2}.

In comparing the 2×2 and 3×3 timings, the two methods seem to be competitive in terms of runtime efficiency. There is a trade off between a larger number of GMRES (or FGMRES) iterations in the 3×3 preconditioner verses the extra layer of a GMRES (or FGMRES) solve on the Navier-Stokes block in the 2×2 preconditioner. Based on our experience thus far, however, the 3×3 preconditioner seems to be more fragile at the higher Rayleigh numbers and more dependent on the quality of the solves on the convection-diffusion operator F.

Finally, we consider the 2×2 preconditioner in the case with fixed $Ra = 2 \times 10^4$ and all loose tolerances on the subsolves. We vary the problems size from $nx = ny = 16$ to $nx = ny = 128$.

5. CONCLUSION

We have proposed a methodology for developing block preconditioners for a nonisothermal incompressible fluid. PDE-level insight leads to a very simple Schur-complement approximation in the block factorization, and we conveniently make use of existing solvers for Navier–Stokes. This strategy seems to be new for coupled fluids problems. Moreover, inexact preconditioner

DOI: 10.1002/nla

	$N = 8$	$N = 16$	$N = 32$	$N = 64$	$N = 128$
Newt FGMRES	7 19.3	7 17.4	7 18.0	6 15.3	6 18.0
Timing (sec)	416	627.2	1660	4882	24790

Table X. The number of Newton iterations, the average number of iterations per Newton step, and timings (in seconds) using the 2×2 block preconditioner of (13) with all loose tolerances. A Newton tolerance of 1×10^{-6} with an inner FGMRES tolerance of 1×10^{-8} was used. We use Ra = 2×10^4, and vary problems size. The NS FGMRES tolerance is 1×10^{-2}, tolerances on the F, A_p, and M_p subsolves are 1×10^{-4}, and the K solve tolerance is 1×10^{-2}.

application allows us to streamline the F solve, the typical bottleneck in Navier–Stokes solvers, quite significantly.

Many of our insights are not specific to fluids, especially those regarding regularity and compactness. In the future, we hope not only to extend our techniques to three dimensions and to other fluids-based problems such as MHD, but also to consider their applicability in a more abstract framework for multiphysics.

REFERENCES

1. R. Becker, *Mesh adaptation for Dirichlet flow control via Nitsche's method*, Communications in numerical methods in engineering, 18 (2002), pp. 669–680.
2. M. Benzi and M. A. Olshanskii, *An augmented Lagrangian-based approach to the Oseen problem*, SIAM J. Scientific Computing, 28 (2006), pp. 2095–2113.
3. M. Benzi, M. A. Olshanskii, and Z. Wang, *Modified augmented Lagrangian preconditioners for the incompressible Navier–Stokes equations*, International Journal for Numerical Methods in Fluids, (publishes electronically) (2010).
4. F. Brezzi and M. Fortin, *Mixed and Hybrid Finite Element Methods*, Springer-Verlag, New York, 1991.
5. S. L. Campbell, I. C. F. Ipsen, C. T. Kelley, and C. D. Meyer, *GMRES and the minimal polynomial*, BIT Numerical Mathematics, 36 (1996), pp. 664–675.
6. G. F. Carey and J. T. Oden, *Finite elements. Vol. VI*, The Texas Finite Element Series, VI, Prentice Hall Inc., Englewood Cliffs, NJ, 1986. Fluid mechanics.
7. T. S. Coffey, C. T. Kelley, and D. E. Keyes, *Pseudotransient continuation and differential-algebraic equations*, SIAM Journal on Scientific Computing, 25 (2004), pp. 553–569.
8. H. C. 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.
9. ———, *A taxonomy and comparison of parallel block multi-level preconditioners for the incompressible Navier–Stokes equations*, Journal of Computational Physics, 227 (2008), pp. 1790–1808.
10. H. C. 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.
11. L. C. Evans, *Partial Differential Equations*, Americal Mathematical Society, Providence, RI, 1998.
12. V. Faber, T. A. Manteuffel, and S. V. Parter, *On the theory of equivalent operators and application to the numerical solution of uniformly elliptic partial differential equations* 1*, Advances in applied mathematics, 11 (1990), pp. 109–163.
13. J. Freund and R. Stenberg, *On weakly imposed boundary conditions for second order problems*, in Proceedings of the Ninth Int. Conf. Finite Elements in Fluids, Venice, 1995, pp. 327–336.
14. M. W. Gee, C. M. Siefert, J. J. Hu, R. S. Tuminaro, and M. G. Sala, *ML 5.0 smoothed aggregation user's guide*, Tech. Rep. SAND2006-2649, Sandia National Laboratories, 2006.
15. M. A. Heroux, *AztecOO user guide*, Tech. Rep. SAND2004-3796, Sandia National Laboratories, 2004.
16. M. A. Heroux, R. A. Bartlett, V. E. Howle, R. J. Hoekstra, J. J. Hu, T. G. Kolda, R. B. Lehoucq, K. R. Long, R. P. Pawlowski, E. T. Phipps, A. G. Salinger, H. K. Thornquist, R. S. Tuminaro, J. M. Willenbring, A. Williams, and K. S. Stanley, *An overview of the Trilinos project*, ACM Trans. Math. Softw., 31 (2005), pp. 397–423.
17. R. C. Kirby, *From Functional Analysis to Iterative Methods*, SIAM Review, 52 (2010), pp. 269–293.
18. P. T. Lin, J. N. Shadid, R. S. Tuminaro, M. Sala, G. L. Hennigan, and R. P. Pawlowski, *A parallel fully coupled algebraic multilevel preconditioner applied to multiphysics PDE applications: Drift-diffusion, flow/transport/reaction, resistive MHD*, International Journal for Numerical Methods in Fluids, 64 (2010), pp. 1148–1179.
19. K. Long, *Large-Scale PDE-Constrained Optimization*, Lecture Notes in Computational Science and Engineering, Springer, Heidelberg, 2003, ch. Sundance: a rapid prototyping toolkit for parallel PDE simulation and optimization.
20. 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.
21. J. Nitsche, *On Dirichlet problems using subspaces with nearly zero boundary conditions*, The Mathematical Foundations of the Finite Element Method with Applications to Partial Differential Equations, (1972), pp. 603–627.
22. M. Sala, K. Stanley, and M. Heroux, *Amesos: A set of general interfaces to sparse direct solver libraries*, in Proceedings of PARA'06 Conference, Umea, Sweden, 2006.
23. J. N. Shadid, E. C. Cyr, R. Pawlowski, R. Tuminaro, L. Chacon, and P. Lin, *Initial performance of fully-coupled AMG and approximate block factorization preconditioners for solution of implicit FE resitive MHD*, in Proceedings of ECCOMAS-CFD, 2010.
24. V. Simoncini and D. Szyld, *On the occurrence of superlinear convergence of exact and inexact krylov subspace methods*, SIAM Review, 47 (2005), pp. 247–272.
25. H. K. Thornquist, *Belos*, http://trilinos.sandia.gov/packages/belos/index.html.

DOI: 10.1002/nla