UNIVERSITY OF MARYLAND
INSTITUTE FOR ADVANCED COMPUTER STUDIES
DEPARTMENT OF COMPUTER SCIENCE

PRECONDITIONING STRATEGIES FOR MODELS OF INCOMPRESSIBLE FLOW

CS-TR #4543 / UMIACS TR #2003-111
HOWARD C. ELMAN†
NOVEMBER 2003

Abstract. We describe some new preconditioning strategies for handling the algebraic systems of equations that arise from discretization of the incompressible Navier-Stokes equations. We demonstrate how these methods adapt in a straightforward manner to decisions on implicit or explicit time discretization, explore their use on a collection of benchmark problems, and show how they relate to classical techniques such as projection methods and SIMPLE.

Key words. Navier-Stokes equations, solvers, preconditioning, incompressible fluids.

†Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742. `elman@cs.umd.edu`. This work was supported in part by the National Science Foundation under grant DMS0208015.

PRECONDITIONING STRATEGIES FOR MODELS OF INCOMPRESSIBLE FLOW

HOWARD ELMAN[†]

Abstract. We describe some new preconditioning strategies for handling the algebraic systems of equations that arise from discretization of the incompressible Navier-Stokes equations. We demonstrate how these methods adapt in a straightforward manner to decisions on implicit or explicit time discretization, explore their use on a collection of benchmark problems, and show how they relate to classical techniques such as projection methods and SIMPLE.

Key words. Navier-Stokes equations, solvers, preconditioning, incompressible fluids.

1. Introduction. In this paper, we describe a new class of computational algorithms for solving the systems of algebraic equations that arise from discretization and linearization of the incompressible Navier-Stokes equations

$$\begin{array}{rcl} \mathbf{u}_t - \nu\nabla^2\mathbf{u} + (\mathbf{u}\cdot \mathrm{grad})\,\mathbf{u} + \mathrm{grad}\, p & = & \mathbf{f} \\ -\mathrm{div}\,\mathbf{u} & = & 0 \end{array} \quad \text{in } \Omega, \tag{1.1}$$

subject to suitable boundary conditions on $\partial\Omega$. Here, Ω is an open bounded domain in $\mathbb{R}^2$ or $\mathbb{R}^3$, $\mathbf{u}$ and p are the velocity and pressure, respectively, $\mathbf{f}$ is the body force per unit mass, and ν is the kinematic viscosity. The algorithms consist of *preconditioning strategies* to be used in conjunction with Krylov subspace methods. They are applied to the primitive variable formulation of (1.1) and are designed to take advantage of the structure of the systems.

The objective in developing these solution algorithms is for them to be effective and adaptable to a variety of circumstances. In particular, they can handle both steady and evolutionary problems in a straightforward manner, and they offer the possibility of being extended to more general systems, such as those that include temperature in the model. Their implementation depends on having efficient solution algorithms for certain subsidiary problems, specifically, the Poisson equation and the convection-diffusion equation. These scalar problems are easier to solve than the Navier-Stokes equations; effective approaches include multigrid, domain decomposition, and sparse direct methods. Once such "building blocks" are available, they can be integrated into a solver for the coupled system (1.1). In this paper, we describe such solvers and demonstrate their utility.

A summary of the paper is as follows. Section 2 gives a brief overview of the original development of the algorithmic approach as designed for the steady Stokes equations and describes what modifications are needed for the Navier-Stokes equations. Section 3 shows how this general approach is related to other traditional strategies for solving (1.1), including projection methods [4, 32] and SIMPLE [25]. Section 4 presents the main ideas for constructing solvers designed for the Navier-Stokes equations, which entail devising strategies for efficiently approximating the inverse of a component of the discrete operator. Section 5 shows the results of a series of numerical experiments demonstrating the utility of this approach for evolutionary problems. Finally, Section 6 summarizes the approach and presents some ways it can be generalized to handle more complex models.

[†]Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742. elman@cs.umd.edu. This work was supported in part by the National Science Foundation under grant DMS0208015.

2. Background. By way of introduction, consider the steady-state Stokes equations

$$\begin{aligned} -\nabla^2 \mathbf{u} + \operatorname{grad} p &= \mathbf{f} \\ -\operatorname{div} \mathbf{u} &= 0\,. \end{aligned} \tag{2.1}$$

Div-stable discretization by finite elements [15] or finite differences [23] leads to a linear system of equations

$$\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix} \tag{2.2}$$

where, for problems in d dimensions, A is a block diagonal matrix consisting of a set of d uncoupled discrete Laplace operators. The coefficient matrix of (2.2) is symmetric and indefinite, and therefore the MINRES [24] variant of the conjugate gradient method is applicable. This iterative method requires a fixed amount of computational work at each step, and it is the optimal Krylov subspace method with respect to the vector Euclidian norm for solving $\mathcal{A}x = b$ where $\mathcal{A}$ is symmetric indefinite. That is, the residual $r_k = b - \mathcal{A}x_k$ of the kth iterate satisfies

$$\|r_k\|_2 = \min_{p_k(0)\in\Pi_k} \|p_k(\mathcal{A})r_0\|_2 \le \min_{p_k(0)\in\Pi_k} \max_{\lambda\in\sigma(\mathcal{A})} |p_k(\lambda)|\,\|r_0\|, \tag{2.3}$$

where Π_k denotes the set of all real polynomials p_k of degree at most k for which $p_k(0) = 1$, and $\sigma(\mathcal{A})$ is the set of eigenvalues of $\mathcal{A}$.

If $\sigma(\mathcal{A})$ is contained in two equal-sized intervals

$$[-a,-b] \cup [c,d], \qquad a,\, b,\, c,\, d\, > 0,$$

then the convergence factor $\min_{p_k(0)\in\Pi_k} \max_{\lambda\in\sigma(\mathcal{A})} |p_k(\lambda)|$ is bounded by [16]

$$2\left(\frac{1-\sqrt{(bc)/(ad)}}{1+\sqrt{(bc)/(ad)}}\right)^{1/2}.$$

The key for rapid convergence is for this quantity to be small. For (2.2), this is achieved by preconditioning. Consider a preconditioning operator of the form [27, 30, 34]

$$\mathcal{Q} = \begin{pmatrix} A & 0 \\ 0 & Q_S \end{pmatrix}. \tag{2.4}$$

This leads to the generalized eigenvalue problem

$$\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \lambda \begin{pmatrix} A & 0 \\ 0 & Q_S \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix}. \tag{2.5}$$

If $\lambda \ne 1$, then the first block of this equation gives $u = [1/(\lambda-1)]\,A^{-1}B^Tp$, and substitution into the second block yields

$$BA^{-1}B^Tp = \mu Q_S p, \quad \mu = \lambda(\lambda-1), \quad \lambda = \frac{1\pm\sqrt{1+4\mu}}{2}. \tag{2.6}$$

A good approximation Q_S to the *Schur complement* $BA^{-1}B^T$ will result in eigenvalues $\{\mu\}$ that lie in a small interval, so that the eigenvalues $\{\lambda\}$ in turn lie in two small intervals. It is shown in [33] that a good choice for Q_S is the pressure mass matrix, M_p. In particular, all μ are contained in an interval that is independent of the discretization mesh parameter h, and, therefore, all λ are also independent of h.

Use of the preconditioner (2.4) with MINRES entails the application of the action of the inverse of $\mathcal{Q}$ to a vector at each iteration; this requires the solution of a set of Poisson equations on the discrete velocity space (application of the action of A^{-1}), and application of the action of M_p^{-1} on the discrete pressure space. The pressure mass matrix is uniformly well-conditioned with respect to h, so the latter operation is inexpensive [35]. For this preconditioner to be useful, the Poisson solves must be done efficiently. A feature of this approach is that these solves can be also be *approximated* with little degradation of its effectiveness. (This contrasts with the alternative strategy of using iterative methods to solve the decoupled system $BA^{-1}B^Tp = BA^{-1}f$.) Formally, this corresponds to replacing A in (2.4) with some approximation Q_A. A good choice would be some Q_A that is spectrally equivalent to A with respect to h, obtained for example using a few steps of multigrid applied to the Poisson equation. See [9, 31] for more details on this and other aspects of solving this problem.

Now consider the Navier-Stokes equations (1.1). Fully implicit time discretization leads to the coupled nonlinear equations

$$\begin{aligned} \tfrac{1}{\alpha}\mathbf{u}^{(m+1)} - \nu\nabla^2\mathbf{u}^{(m+1)} + (\mathbf{u}^{(m+1)}\cdot \mathrm{grad})\,\mathbf{u}^{(m+1)} + \mathrm{grad}\, p^{(m+1)} &= \mathbf{f}(\mathbf{u}^{(m)}) \\ -\mathrm{div}\,\mathbf{u}^{(m+1)} &= 0, \end{aligned}$$

where $(\mathbf{u}^{(m)}, p^{(m)})^T$ is the solution at time step m, and α and $\mathbf{f}(\mathbf{u}^{(m)})$ depend on the time discretization strategy. For example, for the backward Euler method, $\alpha = \Delta t$, the time step, and $\mathbf{f}(\mathbf{u}^{(m)}) = \mathbf{f} - \frac{1}{\Delta t}\mathbf{u}^{(m)}$. At each time step, this system can then be solved using a nonlinear iteration, producing a sequence of iterates $(\mathbf{u}_j^{(m+1)}, p_j^{(m+1)})^T$. An example is the *Picard iteration*, in which the convection coefficient is lagged:

$$\begin{aligned} \tfrac{1}{\alpha}\mathbf{u}_{j+1}^{(m+1)} - \nu\nabla^2\mathbf{u}_{j+1}^{(m+1)} + (\mathbf{u}_j^{(m+1)}\cdot \mathrm{grad})\,\mathbf{u}_{j+1}^{(m+1)} + \mathrm{grad}\, p_{j+1}^{(m+1)} &= \mathbf{f}(\mathbf{u}^{(m)}) \\ -\mathrm{div}\,\mathbf{u}_{j+1}^{(m+1)} &= 0 \end{aligned} \tag{2.7}$$

Div-stable spatial discretization [15, 23] gives a linear system of equations of the form

$$\begin{pmatrix} F & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} u^{(m+1)} \\ p^{(m+1)} \end{pmatrix} = \begin{pmatrix} f^{(m)} \\ g^{(m)} \end{pmatrix}, \tag{2.8}$$

where F is now a block diagonal matrix consisting of a set of d uncoupled discrete operators arising from the time-dependent convection-diffusion equation. The blocks of F essentially have the form

$$\frac{1}{\alpha}M + \nu A + N \tag{2.9}$$

where M, A and N are a discrete mass matrix, Laplacian, and convection operator, respectively. We will discuss our results below in terms of the *Reynolds number* $Re = \frac{|\mathbf{u}|L}{\nu}$; in our examples, the length scale and velocity scales are $L = 2$, $|\mathbf{u}| = 1$, so that $Re = 2/\nu$.[1]

[1]For large Re, F may also include stabilization terms, as when the streamline diffusion discretization [2] is used. This discretization strategy is used for the experimental results discussed in Section 5.

The analogue of (2.4) for (2.8) is

$$\mathcal{Q} = \begin{pmatrix} F & 0 \\ 0 & Q_S \end{pmatrix}. \tag{2.10}$$

Preconditioning as in (2.5) leads to the eigenvalue problem

$$BF^{-1}B^T p = \mu Q_S p, \tag{2.11}$$

for the Schur complement, and once again, we seek an operator Q_S for which the eigenvalues are tightly clustered, and such that application of the inverse of Q_S to a vector in the discrete pressure space is inexpensive.

We defer a discussion of this main point, strategies for choosing Q_S, to Section 4. We conclude here by identifying an improvement in the general design of solution algorithms available for the Navier-Stokes equations. The eigenvalues of (2.11) may be complex, and this would place the eigenvalues of the preconditioned version of the Navier-Stokes equations in two regions in the complex plane, one on each side of the imaginary axis [6]. (This is analogous to the two intervals containing the eigenvalues of the preconditioned Stokes operator.) For the Stokes equations, the positive-definite block diagonal form of the preconditioner makes the preconditioned operator symmetric, which in turn allows the use of the optimal MINRES method. Now, however, (2.8) is not symmetric and there is no Krylov subspace solver that is optimal as in (2.3) and has a fixed amount of computational work per iteration [13, 14]. Since there is no symmetry to maintain, we can use a *block-triangular* variant of (2.10),

$$\mathcal{Q} = \begin{pmatrix} F & B^T \\ 0 & -Q_S \end{pmatrix}. \tag{2.12}$$

This choice leads to the generalized eigenvalue problem

$$\begin{pmatrix} F & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \mu \begin{pmatrix} F & B^T \\ 0 & -Q_S \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix},$$

for which the eigenvalues are those of (2.11) together with $\mu = 1$. A good choice of Q_S will then force all eigenvalues to be clustered on one side of the imaginary axis. Use of this preconditioner in combination with a Krylov subspace method such as GMRES [28] requires approximately half the iterations as the variant based on (2.10), with minimal extra cost per iteration.

So far we have restricted our attention to stable discretizations, for which there is a zero block in the (2,2)-entry of the coefficient matrix of (2.8). It is often convenient to use discretizations that require *stabilization*; for example, this enables the use of equal-order finite elements for velocities and pressures on a common grid [18, 21]. In this case, the system to be solved has the form

$$\begin{pmatrix} F & B^T \\ B & -C \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}, \tag{2.13}$$

where C is a stabilization operator. A second interpretation of (2.12) provides insight into what is needed in this situation. Consider the block LU-factorization

$$\begin{pmatrix} F & B^T \\ B & -C \end{pmatrix} = \begin{pmatrix} I & 0 \\ BF^{-1} & I \end{pmatrix} \begin{pmatrix} F & B^T \\ 0 & -(BF^{-1}B^T + C) \end{pmatrix}. \tag{2.14}$$

This means

$$\begin{pmatrix} F & B^T \\ B & -C \end{pmatrix} \begin{pmatrix} F & B^T \\ 0 & -(BF^{-1}B^T + C) \end{pmatrix}^{-1} = \begin{pmatrix} I & 0 \\ BF^{-1} & I \end{pmatrix}$$

is an "ideally" preconditioned system whose eigenvalues are identically 1. It suggests that the preconditioner should have the form

$$\mathcal{Q} = \begin{pmatrix} Q_F & B^T \\ 0 & -Q_S \end{pmatrix}. \tag{2.15}$$

That is, just as for stable discretizations, we require a good approximation Q_S for the Schur complement with respect to F, which for (2.13) is $BF^{-1}B^T + C$. Moreover, as discussed for the Stokes equations, in general, additional efficiencies can be achieved using $Q_F \approx F$, i.e. by using iterative methods to approximate the action of the inverse of the (convection-diffusion) operator F.

3. Relation to other methods. In this section, we show some connections between the preconditioning methods considered above and two established solution methods for the Navier-Stokes equations, projection methods and SIMPLE. This is a brief overview of a more detailed discussion that can be found in [10].

The "classical" first order projection method for evolutionary problems [4, 32] can be viewed as a two-step procedure for advancing from time step m to step $m+1$. Viewed in its semi-discrete form, with only time discretization, it is

$$\begin{aligned} &\text{Step 1: solve} \quad \frac{\mathbf{u}^{(*)} - \mathbf{u}^{(m)}}{\Delta t} - \nu \nabla^2 \mathbf{u}^{(*)} + (\mathbf{u}^{(m)} \cdot grad)\mathbf{u}^{(m)} = \mathbf{f} \text{ for } \mathbf{u}^{(*)}, \\ &\text{Step 2: solve} \quad \begin{pmatrix} \frac{1}{\Delta t} I & \nabla \\ -\nabla & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u}^{(m+1)} \\ p^{(m+1)} \end{pmatrix} = \begin{pmatrix} \frac{1}{\Delta t}\mathbf{u}^{(*)} \\ 0 \end{pmatrix}. \end{aligned} \tag{3.1}$$

In the second step, $p^{(m+1)}$ is obtained by solving a Poisson equation, and $\mathbf{u}^{(m+1)}$ is then the orthogonal projection of the intermediate quantity $\mathbf{u}^{(*)}$ into the space of incompressible vector fields. Spatial discretization gives the matrix formulation

$$\begin{aligned} &\text{Step 1: solve} \quad \left(\tfrac{1}{\Delta t} M + \nu A\right) u^{(*)} = f - \left(-\tfrac{1}{\Delta t} M + N\right) u^{(m)} \\ &\text{Step 2: solve} \quad \begin{pmatrix} \frac{1}{\Delta t} M & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} u^{(m+1)} \\ p^{(m+1)} \end{pmatrix} = \begin{pmatrix} \frac{1}{\Delta t} M u^{(*)} \\ 0 \end{pmatrix}, \end{aligned}$$

where A, N and M are as in (2.9). The updated discrete pressure is obtained by solving the discrete pressure Poisson equation

$$BM^{-1}B^T p^{(m+1)} = Bu^{(*)}.$$

Substitution of $u^{(*)}$ into Step 2 shows that the advancement in time is done by solving the algebraic system

$$\begin{pmatrix} \frac{1}{\Delta t} M + \nu A & \left(\frac{1}{\Delta t} M + \nu A\right)\left(\frac{1}{\Delta t} M\right)^{-1} B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} u^{(m+1)} \\ p^{(m+1)} \end{pmatrix} = \begin{pmatrix} f - \left(-\frac{1}{\Delta t} M + N\right) u^{(m)} \\ 0 \end{pmatrix}. \tag{3.2}$$

It was observed in [26] that the sequence of operations performed for the projection method derive from a block LU-decomposition of the coefficient matrix of this system,

$$\begin{pmatrix} \frac{1}{\Delta t}M+\nu A & \left(\frac{1}{\Delta t}M+\nu A\right)\left(\frac{1}{\Delta t}M\right)^{-1}B^T \\ B & 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{\Delta t}M+\nu A & 0 \\ B & -B\left(\frac{1}{\Delta t}M\right)^{-1}B^T \end{pmatrix} \begin{pmatrix} I & \left(\frac{1}{\Delta t}M\right)^{-1}B^T \\ 0 & I \end{pmatrix}. \tag{3.3}$$

Following [26], it is instructive to contrast this with what would be required to perform an update derived purely from linearization and discretization of the original problem (1.1). If linearization is performed in a manner analogous to (3.1), i.e., by treating convection fully explicitly, then a time step would consist of solving the system

$$\begin{pmatrix} \frac{1}{\Delta t}M+\nu A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} u^{(m+1)} \\ p^{(m+1)} \end{pmatrix} = \begin{pmatrix} f-\left(-\frac{1}{\Delta t}M+N\right)u^{(m)} \\ 0 \end{pmatrix} \tag{3.4}$$

instead of (3.2). The coefficient matrix of (3.2) can be viewed as an approximation to the coefficient matrix of (3.4), the only difference lying in the block (1,2)–entry:

$$B^T-\left(\tfrac{1}{\Delta t}M+\nu A\right)\left(\tfrac{1}{\Delta t}M\right)^{-1}B^T = -(\Delta t)\nu A M^{-1}B^T = O(\Delta t).$$

Since this is of the same order of magnitude as the time discretization error, there is no loss of accuracy associated with the projection method [17]. Thus, projection methods can be viewed as a device for avoiding having to solve the Stokes-like system of equations of (3.2). The analogue for (3.4) of the block-LU decomposition (3.3) is

$$\begin{pmatrix} \frac{1}{\Delta t}M+\nu A & B^T \\ B & 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{\Delta t}M+\nu A & 0 \\ B & -B\left(\frac{1}{\Delta t}M+\nu A\right)^{-1}B^T \end{pmatrix} \begin{pmatrix} I & \left(\frac{1}{\Delta t}M+\nu A\right)^{-1}B^T \\ 0 & I \end{pmatrix},$$

which is a factorization like that of (2.14). As we have observed, what is needed for efficient processing of this system a good approximation to the Schur complement operator, in this case $B\left(\frac{1}{\Delta t}M+\nu A\right)^{-1}B^T$. This particular (generalized Stokes) problem has been treated in [1, 3].

Remark. Viewing (3.2) as derived from (3.4) also provides a means of implicitly defining boundary conditions for projection methods. This is done for the pressures via the Schur complement operator $BM^{-1}B^T$ appearing in (3.3). No boundary conditions are needed for $u^{(*)}$ since this quantity is implicitly incorporated into (3.2). See [5, 26] for further discussion of this point; reference [5] also shows how these ideas work for higher order time discretization.

To describe a connection between the widely used SIMPLE ("Semi-Implicit Method for Pressure-Linked Equations") method [25] and the preconditioning methodology of Section 2, we follow Wesseling [36, pp. 296ff]. SIMPLE uses a block factorization

$$\begin{pmatrix} F & B^T \\ B & 0 \end{pmatrix} \approx \begin{pmatrix} Q_F & 0 \\ B & -B\hat{F}B^T \end{pmatrix} \begin{pmatrix} I & \hat{F}^{-1}B^T \\ 0 & I \end{pmatrix} \tag{3.5}$$

where both Q_F and $\hat{F}$ are approximations to F. This represents an alternative approximation to the block factorization (2.14). The first operator Q_F is determined in a manner analogous to the approach of Section 2, via an iteration that approximates the action of the inverse of F. The second approximation $\hat{F}$ is chosen so that the operator $B\hat{F}^{-1}B^T$ can be used explicitly. The standard implementation [25] uses the diagonal of F for $\hat{F}$. This means that the approximate Schur complement $B\hat{F}^{-1}B^T$ resembles a discrete Laplacian operator.

The solver for (2.8) derived from (3.5) is a stationary iteration essentially of the form

$$\begin{pmatrix} u_{j+1}^{(m+1)} \\ p_{j+1}^{(m+1)} \end{pmatrix} = \begin{pmatrix} u_{j}^{(m+1)} \\ p_{j}^{(m+1)} \end{pmatrix} +$$
$$\begin{pmatrix} I & \hat{F}^{-1}B^T \\ 0 & I \end{pmatrix}^{-1} \begin{pmatrix} Q_F & 0 \\ B & -B\hat{F}B^T \end{pmatrix}^{-1} \left[\begin{pmatrix} f^{(m)} \\ g^{(m)} \end{pmatrix} - \begin{pmatrix} F & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} u_j^{(m)} \\ p_j^{(m)} \end{pmatrix} \right].$$

This can easily be adapted to produce a preconditioned iteration. The main difference between this approach and those of the next section lies in the approximation to the Schur complement. The choice determined by $\hat{F} = diag(F)$ is a good one in the case of small time steps but is less effective when the spatial mesh size is small or when flows are convection-dominated [36].

4. Approximation to the Schur complement. In contrast to the methods discussed in the previous section, the perspective of the new approach is to treat the coupled equations directly by approximating the Schur complement associated with (2.8) or (2.13). In this section, we discuss two ways to do this.

For the first, assume that both m and j are fixed in (2.7), and let $\mathbf{w} = \mathbf{u}_j^{(m+1)}$ denote the lagged convection coefficient. Consider the translated convection-diffusion operator $\frac{1}{\alpha}I - \nu\nabla^2 + \mathbf{w}\cdot\nabla$. Suppose that the pressure space also admits a convection-diffusion operator $(-\nu\nabla^2 + \mathbf{w}\cdot\nabla)_p$, and furthermore that the commutator of the translated convection-diffusion operators with the gradient operator,

$$(\tfrac{1}{\alpha}I - \nu\nabla^2 + \mathbf{w}\cdot\nabla)\nabla - \nabla(\tfrac{1}{\alpha}I - \nu\nabla^2 + \mathbf{w}\cdot\nabla)_p\,,$$

is small in some sense. A discrete version of this assertion is that

$$(M_u^{-1}F)\,(M_u^{-1}B^T) - (M_u^{-1}B^T)\,(M_p^{-1}F_p) \tag{4.1}$$

is also small, where M_u is the mass matrix associated with the velocity discretization and F_p is a discrete approximation to the translated convection-diffusion operator; both F and F_p have the form given in (2.9). It follows that

$$BF^{-1}B^T \approx A_pF_p^{-1}M_p\,, \tag{4.2}$$

where $A_p = BM_u^{-1}B^T$ is a discrete Laplacian operator. The matrix on the right hand side here defines a preconditioning operator Q_S. More generally, any suitable discrete approximation to the Laplacian can be used for A_p; in particular, if stabilization is required, then $BM_u^{-1}B^T$ will be rank-deficient and an alternative, stable, approximation A_p to the Laplacian would be needed. The resulting operator can then be used as an approximation to $BF^{-1}B^T + C$. See [12, 20, 29] for additional discussion of these points; in particular, [20] gives an alternative derivation of Q_S using the fundamental solution tensor for the linearized Navier-Stokes operator. An important point is that although commutativity is used in the derivation above, it is not necessary that (4.1) be small (it is not small when equal order finite element methods on different grids are used [7]) for the idea to be effective.

An alternative approximation to the Schur complement is derived from a simple observation in linear algebra [8]. Suppose G and H are two rectangular matrices of dimensions $n_1 \times n_2$ with full rank $n_1 \le n_2$. The matrix

$$H^T(GH^T)^{-1}G^T$$

maps $\mathbb{R}^{n_1}$ to $range(H^T) \subset \mathbb{R}^{n_1}$, and it fixes $range(H^T)$. That is, $H^T(GH^T)^{-1}G^T = I$ on $range(H^T)$. With the choices $G = BF^{-1}$, $H = B$, this gives

$$B^T\,(BF^{-1}B^T)^{-1}BF^{-1} = I \qquad \text{on } range(B^T)$$

or, equivalently,

$$B^T\,(BF^{-1}B^T)^{-1}B = F \qquad \text{on } range(F^{-1}B^T).$$

If $range(B^T)$ were contained in $range(F^{-1}B^T)$, then this would imply that

$$(BB^T)\,(BF^{-1}B^T)^{-1}(BB^T) = BFB^T,$$

or

$$(BF^{-1}B^T)^{-1} = (BB^T)^{-1}(BFB^T)\,(BB^T)^{-1}. \tag{4.3}$$

It is generally not the case that $range(B^T) \subset range(F^{-1}B^T)$, so that the expression above is not a valid equality. However, if we view (4.3) as an approximation, we can use the expression on the right side to *define* a preconditioning operator Q_S^{-1}. Note that this approach is applicable only to div-stable discretizations; ideas to generalize it to stabilized discretizations are under development.

We refer to the operator defined by (2.15) and (4.2) as the "F_p-preconditioner," and that defined by (2.15) and (4.3) as the "BFBt-preconditioner." Both strategies were originally developed with steady problems in mind, and in this regard they have been studied in [8, 11, 12, 20, 22, 29]. Some of their properties for solving the problems that arise from low-order finite element or finite-difference discretization of steady problems are as follows:

1. With F_p-preconditioning, GMRES iteration exhibits a rate of convergence that is independent of the discretization mesh size h [12, 22].
2. With BFBt-preconditioning, convergence of GMRES iteration is mildly dependent on mesh size, with iteration counts that appear to grow in proportion to $h^{-1/2}$ [8].
3. Both methods lead to convergence rates that are mildly dependent on the Reynolds number [8, 12, 20, 29].

As we will see in Section 5, for evolutionary problems the dependence of convergence rates on mesh size and Reynolds numbers becomes negligable; similar results have also been shown in [9, 11]. The results cited here are largely experimental. The report [22] contains rigorous bounds showing that the eigenvalues of the F_p-preconditioned operator $\mathcal{A}\mathcal{Q}^{-1}$ are contained in a region that is independent of mesh size h and time step Δt; these can be used to establish bounds on the asymptotic convergence rates of GMRES.

Using the preconditioner (2.15) with an iterative method such as GMRES to solve systems (2.8) or (2.13) requires that the action of the inverse of $\mathcal{Q}$ be applied at each step. The main computational tasks required for this are to apply the action of the inverse of Q_S to a member of the discrete pressure space, and to apply the action of the inverse of Q_F to a member of the discrete velocity space. For Picard iteration (2.7), the latter operation entails solution of a set of scalar discrete convection-diffusion equations. This can be done effectively by iterative methods. For evolutionary problems especially, this is a straightforward computation because of the presence of the mass matrix in F; see [11].

For the other main task, application of the action of the inverse of Q_S, the principal cost is for solution of the Poisson equation. The F_p-preconditioner requires one Poisson solve at each step, and the BFBt-preconditioner requires two per step. Once again, this task can be handled by iterative methods, and moreover, approximate solutions to the Poisson equations are sufficient. In our experience, one or two steps of V-cycle multigrid are sufficient for good performance of the complete solver.

If we compare these two preconditioning methods, it is evident that the F_p-approach tends to have more favorable properties. However, one advantage of the BFBt method is that it is fully automated: it is defined explicitly in terms of operators constructed from the discretization, and it requires no action on the part of a potential user in order to be specified. In contrast, for the F_p-preconditioner, it is necessary that the matrices F_p and A_p be constructed. In principal this can be done using a code similar to the one that produces F, but it must be done. It is also necessary to make decisions on how boundary conditions affect the definitions of F_p and A_p.

5. Experimental results. In this section, we show some representative experimental results on performance of the preconditioners described in the previous section. We used two benchmark problems:

1. The two-dimensional driven cavity problem on the domain $\Omega = [-1,1] \times [-1,1]$. Boundary conditions are $\mathbf{u} \equiv 0$ on $\partial\Omega$ except $u_1(x,1) = 1$ at the top of Ω.
2. Flow over a backward-facing step. Ω is the L-shaped domain $[-1,0]\times[0,1] \cup [0,5]\times[-1,1]$, with parabolic inflow conditions $u_1(-1,y) = 1-y^2$, $u_2(-1,y) = 0$, natural boundary conditions $\nu\frac{\partial u_1}{\partial x} - p = 0$, $\frac{\partial u_2}{\partial y} = 0$ at the outflow boundary $x = 5$, and $\mathbf{u} \equiv 0$ otherwise.

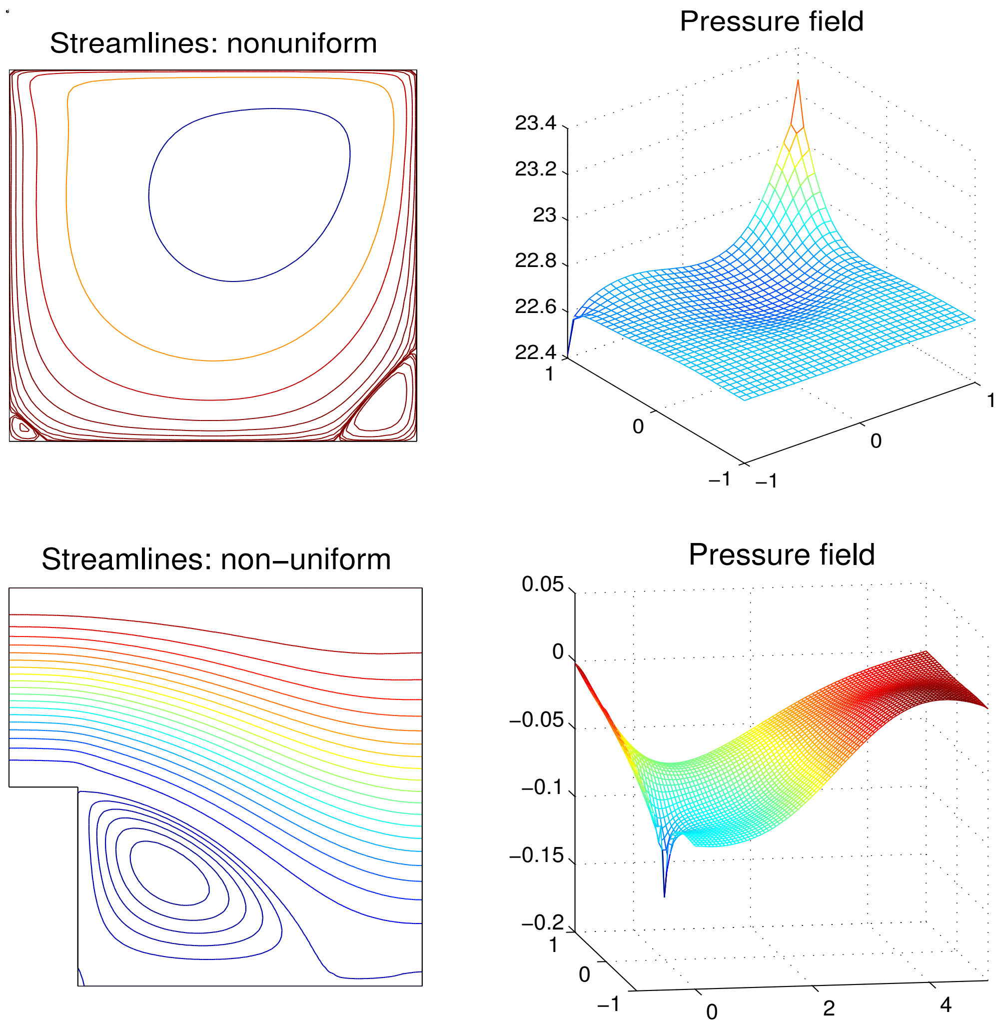
Steady solutions of these problems for Reynolds number 200 are depicted in Figure 5.1.

Details concerning the experiments are as follows. Spatial discretization was done using the stable Q_2-Q_1 finite element discretization consisting of biquadratic elements for the velocities and bilinear elements for the pressures, on a uniform grid. Streamline upwinding [2] was used in cases where the cell Reynolds number $\frac{|\mathbf{u}|h}{2\nu}$ is greater than one. Rather than perform a full transient iteration, we simulated time discretization as follows. For most of the tests, we performed a Picard iteration for the steady problem, saved the coefficient matrix J arising from the second Picard step, and then computed

$$F = \frac{1}{\Delta t} M + J \qquad (5.1)$$

where Δt is to be viewed as a pseudo-time step. For backward Euler discretization, Δt is the value of the time step. For higher order time discretizations, there are other scaling factors involved. For example, the Crank-Nicolson discretization would have time step equal to $\Delta t/2$. In these experiments, once F is defined by (5.1), we

FIG. 5.1. *Steady state solutions of the driven cavity (top) and backward step (bottom) problems*



solve solve $Fu = f$ where f is the right hand side that arises from the steady Picard iteration.

To specify the operators A_p and F_p used in the F_p-preconditioner, it is necessary to associate boundary conditions with them. In these tests, for the driven cavity (enclosed flow) problem, A_p and F_p are defined as though derived from Neumann boundary conditions. For the steady version of the backward step problem, it is necessary to use a Dirichlet condition at the inflow boundary $x = -1$. For the transient step problem, we found a Neumann condition for F_p at the inflow to be slightly more effective and this choice was used in the experiments. We note that although this issue is similar to what is often faced for projection methods [19], here it is only an aspect of the solution algorithm and it has no effect on discretization of the pressure, for which no boundary conditions are specified.

Representative results are shown in Figures 5.2 and 5.3 for the driven cavity problem, and in Figures 5.4 and 5.5 for the backward step. We show results for

FIG. 5.2. *Iterations of preconditioned* GMRES *at sample time and Picard steps, for driven cavity problem,* $Re = 200$ *and* Q_2-Q_1 *finite elements. Left:* 32×32 *grid, right:* 64×64 *grid.*

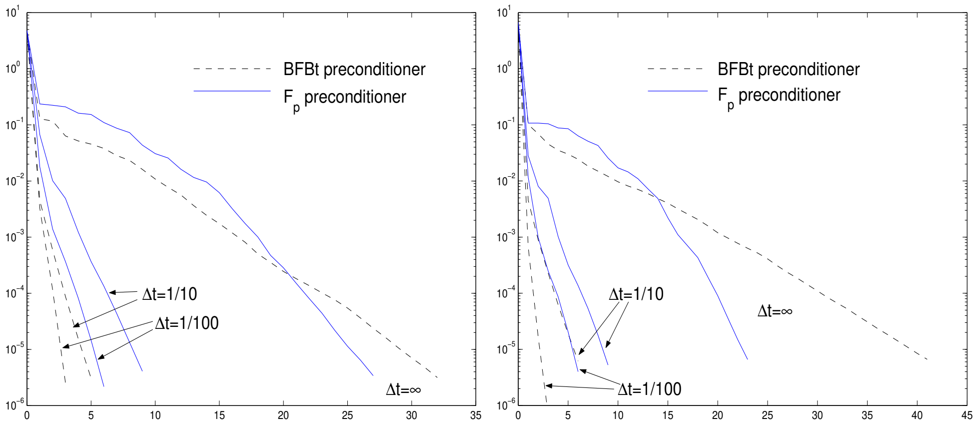


FIG. 5.3. *Iterations of preconditioned* GMRES *at sample time and Picard steps, for driven cavity problem,* $Re = 1000$ *and* Q_2-Q_1 *finite elements. Left:* 32×32 *grid, right:* 64×64 *grid, bottom:* 128×128 *grid.*

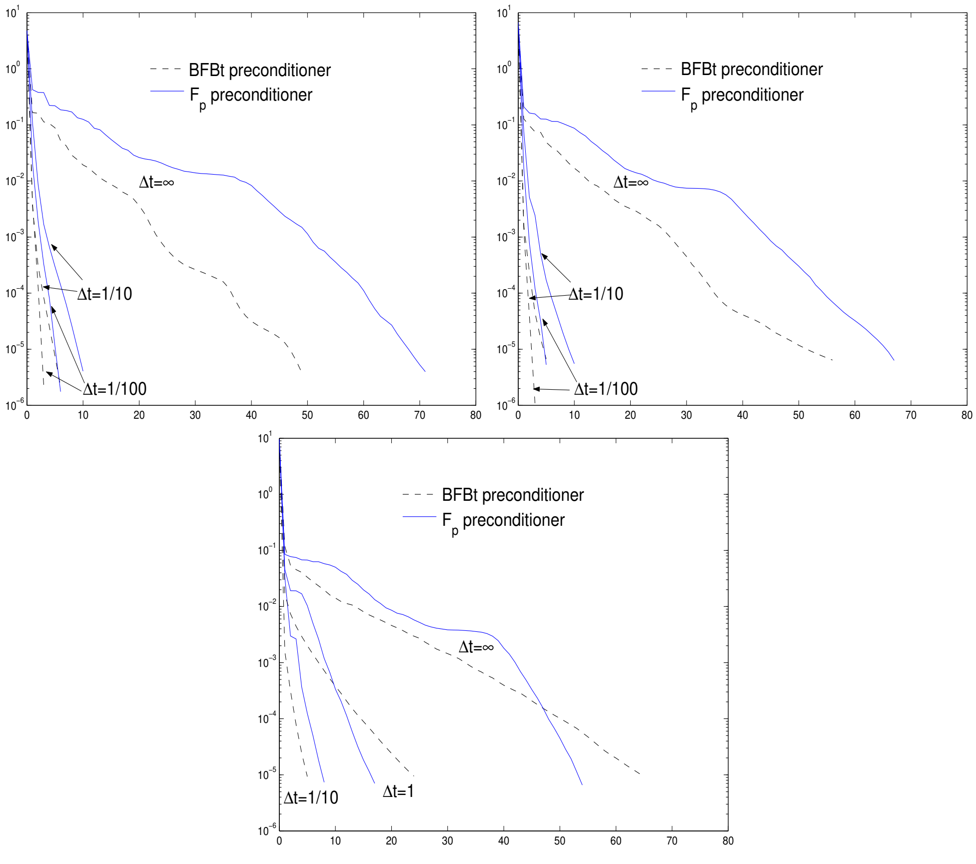
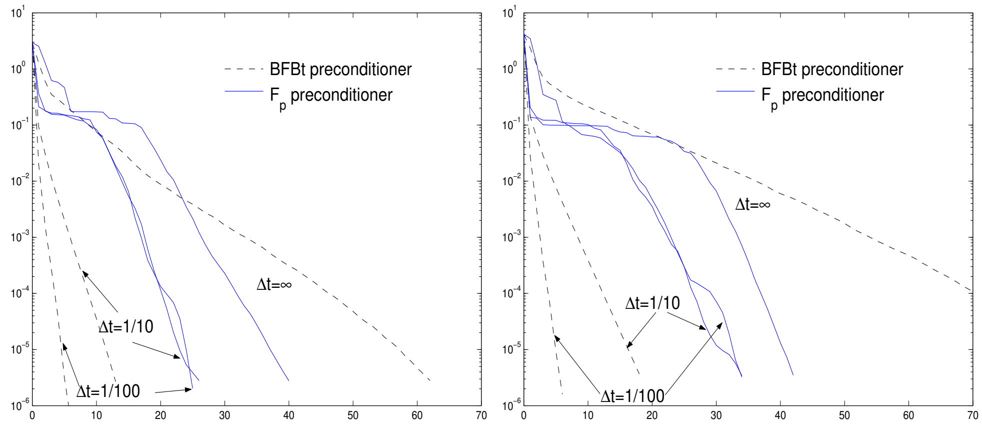


FIG. 5.4. *Iterations of preconditioned* GMRES *at sample time and Picard steps, for backward facing step, $Re = 200$ and Q_2-Q_1 finite elements. Left: 32×96 grid, right: 64×192 grid.*



$\Delta t = 1/100$, $1/10$, 1 (in one example), and ∞. The last value, which corresponds to the steady problem, gives an idea of what the maximal solution costs (per time step) would be in the case of very large CFL numbers.

The main points to observe concerning the transient problem are as follows:

- For fixed Δt, the iteration counts required for convergence are essentially independent of both the Reynolds number and the discretization mesh size.
- Iteration counts are decreasing as a function of the time step size. This is a consequence of the fact that the term $\frac{1}{\Delta t}M$ becomes more dominant in the definition of both the discrete operator and the preconditioner as $\Delta t \to 0$.
- The BFBt-preconditioned solvers require fewer iterations (typically on the order of 10 or fewer for the driven cavity problem and 20 or fewer for the backward step) than the F_p-preconditioned solvers. Although the computations for the BFBt operator are more expensive at any step (requiring two Poisson solves at each step instead of one), there is typically at least a 50% savings in iterations, which makes the BFBt-preconditioner more efficient in these examples.
- Note that the first and third assertions *do not* carry over to the steady problem, where only the F_p-preconditioner is mesh independent and the performance of both methods depends on Re.

The problems arising from the backward facing step are considerably more difficult than those arising from driven cavity flow. Although this is not necessarily unexpected, there is no obvious explanation that can be seen purely from the properties of the algebraic systems.

Table 5.1 gives estimates for the CFL numbers $\|u\|\,\Delta t/h$ for these tests, derived from the empirically observed values $\|u\| \approx 17$ (in the vector Euclidian norm) for the driven cavity problem and $\|u\| \approx 29$ for the backward step. It is evident that this approach enables the use of large CFL numbers when it is feasible, i.e., when accurate representation of short time-scale physics is not the goal.

Finally, Figure 5.6 shows a few results for the case where the nonlinear iteration is based on Newton's method instead of the Picard iteration (2.7). As above, the experiments were done for coefficient matrix $F = \frac{1}{\Delta t}M + J$ where J is now the Jacobian of the nonlinear system obtained after two Newton iterations. These coefficient

FIG. 5.5. *Iterations of preconditioned* GMRES *at sample time and Picard steps, for backward facing step, $Re = 1000$ and Q_2-Q_1 finite elements. Left: 32×96 grid, right: 64×192 grid.*

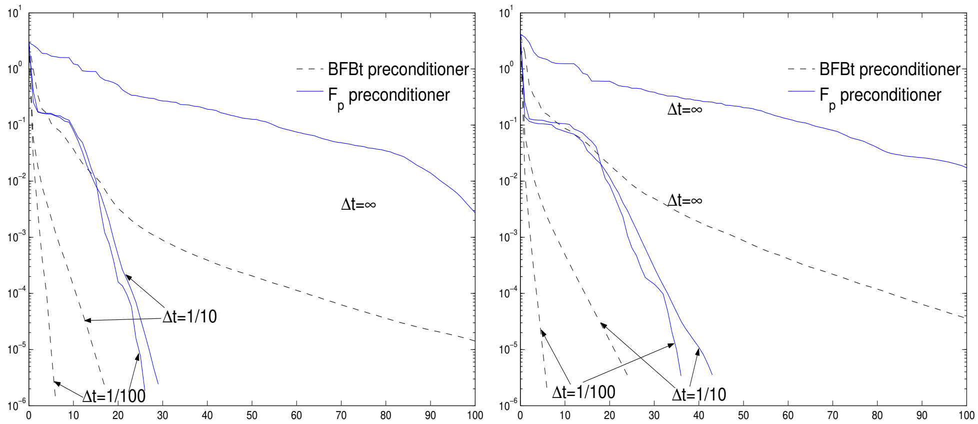


TABLE 5.1
Estimated CFL numbers for test problems.

	Driven cavity mesh			Backward step mesh	
Δt	32×32	64×64	128×128	32×96	64×192
$1/10$	27	54	109	46	93
$1/100$	2.7	5.4	10.9	4.6	9.3

matrices have a more complex structure, and in particular, F is no longer a block diagonal matrix. In this case, the F_p-preconditioner is defined using the velocity from the previous step for the convection coefficient. These graphs should be compared with the first ones from Figures 5.2 and 5.4; they show that the costs to solve these problems are roughly twice those incurred for Picard iteration.

6. Concluding remarks and generalizations. The goal of developing these approaches for preconditioning is to enable the development of flexible and easily implemented solvers for the Navier-Stokes equations. This is achieved in part by building on efforts to develop efficient solvers for simpler subsidiary problems such as the convection-diffusion and Poisson equations. The resulting algorithms can be applied directly to both evolutionary and steady problems and enable the use of time discretization with large CFL numbers.

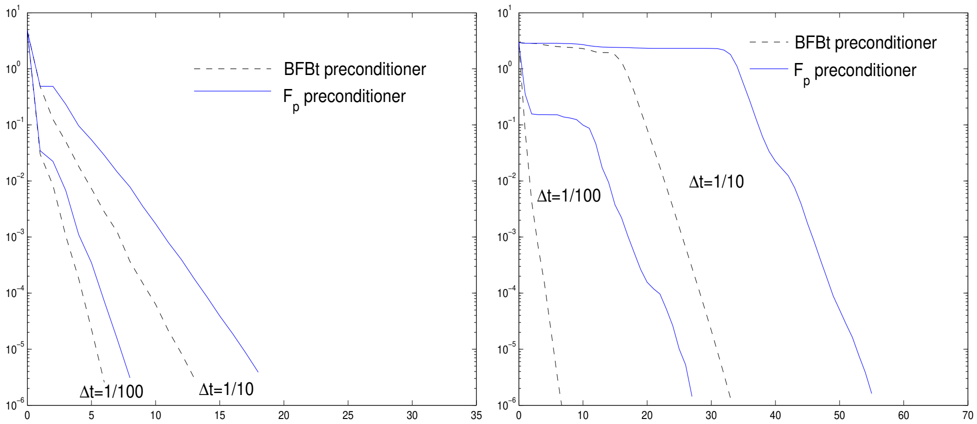
We conclude with the observation that they also offer the potential to handle more general systems. Consider the case where heat transport is combined with the Navier-Stokes equations, giving rise to the Boussinesq equations

$$\left.\begin{array}{l} \alpha \mathbf{u}_t - \nabla \cdot (\nu_u \nabla \mathbf{u}) + (\mathbf{u} \cdot \operatorname{grad})\mathbf{u} + \operatorname{grad} p = \mathbf{f}(T) \\ \alpha T_t - \nabla \cdot (\nu_T \nabla T) + (\mathbf{u} \cdot \operatorname{grad}) T = g(T) \\ -\operatorname{div} \mathbf{u} = 0 \end{array}\right. \quad \text{on } \mathcal{D} \subset \mathbf{R}^d,\ d = 2 \text{ or } 3. \tag{6.1}$$

Linearization and discretization (implicitly in time in the case of transient problems) leads to a sequence of linear systems of equations now having the form

$$\begin{pmatrix} F_u & G & B^T \\ H & F_T & 0 \\ B & 0 & 0 \end{pmatrix} \begin{pmatrix} \delta \\ \tau \\ \eta \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ g \\ 0 \end{pmatrix}. \tag{6.2}$$

FIG. 5.6. *Iterations of preconditioned* GMRES *at sample time and Newton steps, for* $Re = 200$ *and* Q_2*-*Q_1 *finite elements Left: driven cavity problem on a* 32×32 *grid. Right: backward facing step on a* 32×96 *grid.*



The precise structure of the individual blocks of the coefficient matrix depends on the strategy used to linearize, that is, on the algorithm used to perform the nonlinear iteration. If a Picard iteration is used, then both F_u and F_T are convection-diffusion operators as above, and $H = 0$. In this case, the Schur complement operator is $BF_u^{-1}B^T$, which is identical to the operator arising from the Navier-Stokes equations. Thus, we expect these ideas to be directly applicable in more general settings.

REFERENCES

[1] J. H. Bramble and J. E. Pasciak. Iterative techniques for time dependent Stokes problems. In W. Habashi, editor, *Solution Techniques for Large-Scale CFD Problems*, pages 201–216. John Wiley, New York, 1995.

[2] A. Brooks and T. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Comp. Meth. Appl. Mech. Engng.*, 32:199–259, 1982.

[3] J. Cahouet and J.-P. Chabard. Some fast 3D finite element solvers for the generalized Stokes problem. *Int. J. Numer. Meth. Fluids*, 8:869–895, 1988.

[4] A. J. Chorin. Numerical solution of the Navier-Stokes equations. *Math. Comp*, 22:745–762, 1968.

[5] J. K. Dukowicz and A. S. Dvinsky. Approximate factorization as a high order splitting for implicit incompressible flow equations. *J. Comput. Phys.*, 102:336–347, 1992.

[6] H. Elman and D. Silvester. Fast nonsymmetric iterations and preconditioning for Navier-Stokes equations. *SIAM J. Sci. Comput.*, 17:33–46, 1996.

[7] H. C. Elman. Multigrid and Krylov subspace methods for the discrete Stokes equations. *Int. J. Numer. Meth. Fluids*, 227:755–770, 1996.

[8] H. C. Elman. Preconditioning for the steady-state Navier-Stokes equations with low viscosity. *SIAM J. Sci. Comput.*, 20:1299–1316, 1999.

[9] H. C. Elman. Preconditioners for saddle point problems arising in computational fluid dynamics. *Applied Numerical Mathematics*, 43:75–89, 2002.

[10] H. C. Elman, V. E. Howles, J. Shadid, and R. Tuminaro. In preparation, 2003.

[11] H. C. Elman, V. E. Howles, J. Shadid, and R. Tuminaro. A parallel block multi-level preconditioner for the 3D incompressible Navier–Stokes equations. *J. Comput. Phys.*, 187:505–523, 2003.

[12] H. C. Elman, D. J. Silvester, and A. J. Wathen. Performance and analysis of saddle point preconditioners for the discrete steady-state Navier-Stokes equations. *Numer. Math.*, 90:665–688, 2002.

[13] V. Faber and T. A. Manteuffel. Necessary and sufficient conditions for the existence of a

conjugate gradient method. *SIAM J. Numer. Anal*, 21:352–362, 1984.

[14] V. Faber and T. A. Manteuffel. Orthogonal error methods. *SIAM J. Numer. Anal*, 24:170–187, 1987.

[15] V. Girault and P. A. Raviart. *Finite Element Approximation of the Navier-Stokes Equations*. Springer-Verlag, New York, 1986.

[16] A. Greenbaum. *Iterative Methods for Solving Linear Systems*. SIAM, Philadelphia, 1997.

[17] M. O. Henriksen and J. Holmen. Algebraic splitting for incompressible Navier-Stokes equations. *J. Comput. Phys.*, 175:438–453, 2002.

[18] T. J. Hughes and L. P. Franca. A new finite element formulation for computational fluid dynamics: VII. The Stokes problem with various well-posed boundary conditions: Symmetric formulations that converge for all velocity/pressure spaces. *Comp. Meths. Appl. Mech. Engrg.*, 65:85–96, 1987.

[19] G. E. Karniadakis, M. Israeli, and S. Orszag. High-order splitting methods for the incompressible Navier-Stokes equations. *J. Comput. Phys.*, 97:414–443, 1991.

[20] D. Kay, D. Loghin, and A. Wathen. A preconditioner for the steady-state Navier-Stokes equations. *SIAM J. Sci. Comput.*, 24:237–256, 2002.

[21] N. Kechkar and D. Silvester. Analysis of locally stabilized mixed finite element methods for the Stokes problem. *Math. Comp.*, 58:1–10, 1992.

[22] D. Loghin. Analysis of Preconditioned Picard Iterations for the Navier-Stokes Equations. Technical Report 01/10, Oxford University Computing Laboratory, 2001.

[23] R. A. Nicolaides. Analysis and convergence of the MAC scheme I. *SIAM J. Numer. Anal.*, 29:1579–1591, 1992.

[24] C. C. Paige and M. A. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM. J. Numer. Anal.*, 12:617–629, 1975.

[25] S. V. Patankar and D. B. Spalding. A calculation proedure for heat and mass transfer in three-dimensional parabolic flows. *Int. J. Heat and Mass Transfer*, 15:1787–1806, 1972.

[26] J. B. Perot. An analysis of the fractional step method. *J. Comput. Phys.*, 108:51–58, 1993.

[27] T. Rusten and R. Winther. A preconditioned iterative method for saddle point problems. *SIAM J. Matr. Anal. Appl.*, 13:887–904, 1992.

[28] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7:856–869, 1986.

[29] D. Silvester, H. Elman, D. Kay, and A. Wathen. Efficient preconditioning of the linearized Navier-Stokes equations for incompressible flow. *J. Comp. Appl. Math.*, 128:261–279, 2001.

[30] D. Silvester and A. Wathen. Fast iterative solution of stabilized Stokes systems. Part II: Using block preconditioners. *SIAM J. Numer. Anal.*, 31:1352–1367, 1994.

[31] D. J. Silvester and A. J. Wathen. Fast and robust solvers for time-discretised incompressible Navier-Stokes equations. In D. F. Griffiths and G. A. Watson, editors, *Numerical Analysis: Proceedings of the 1995 Dundee Biennial Conference*. Longman, 1996. Pitman Research Notes in Mathematics Series 344.

[32] R. Témam. Sur l'approximation de la solution des équations de Navier-Stokes par la méthod des pas fractionnaires (II). *Arch. Rational Mech. Anal.*, 33:377–385, 1969.

[33] R. Verfürth. A combined conjugate gradient-multigrid algorithm for the numerical solution of the Stokes problem. *IMA J. Numer. Anal.*, 4:441–455, 1984.

[34] A. Wathen and D. Silvester. Fast iterative solution of stabilized Stokes systems. Part I: Using simple diagonal preconditioners. *SIAM J. Numer. Anal.*, 30:630–649, 1993.

[35] A. J. Wathen. Realistic eigenvalue bounds for the Galerkin mass matrix. *IMA J. Numer. Anal.*, 7:449–457, 1987.

[36] P. Wesseling. *Principles of Computational Fluid Dynamics*. Springer-Verlag, Berlin, 2001.