JOURNAL OF COMPUTATIONAL PHYSICS 108, 51–58 (1993)

An Analysis of the Fractional Step Method

J. BLAIR PEROT

Mechanical Engineering Department, Stanford University, Stanford, California 94305

Received March 16, 1992; revised October 28, 1992

The fractional step method for solving the incompressible Navier–Stokes equations in primitive variables is analyzed as a block LU decomposition. In this formulation the issues involving boundary conditions for the intermediate velocity variables and the pressure are clearly resolved. In addition, it is shown that poor temporal accuracy (first-order) is not due to boundary conditions, but due to the method itself. A generalized block LU decomposition that overcomes this difficulty is presented, allowing arbitrarily high temporal order of accuracy. The generalized decomposition is shown to be useful for a wide range of problems including *steady* problems. Technical issues, such as stability and the appropriate pressure update scheme, are also addressed. Numerical simulations of the unsteady, incompressible Navier–Stokes equations in a square domain confirm the theoretical results.

1. INTRODUCTION

This paper is concerned with techniques for the solution of the unsteady, incompressible Navier–Stokes equations in primative variables. In particular, it is concerned with a formulation first introduced independently by Chorin [1] and Temam [2] and now referred to as the fractional step method. Despite many advantages and extensive use in the past by numerous researchers [3–5], the fractional step method has a few major drawbacks. In general, the method is first-order accurate in time, and serious confusion and/or disagreement concerning boundary conditions and the details of the methods implementation exists. This paper shows that these problems can be resolved by resisting the urge to view the fractional step method as a time splitting and, instead, viewing it as an approximate block LU factorization of the fully discretized equations.

Consider the non-dimensionalized unsteady incompressible Navier–Stokes equations

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} = -\nabla p + \frac{1}{\mathrm{Re}}\nabla^2\mathbf{u}, \tag{1a}$$

$$\nabla\cdot\mathbf{u} = 0. \tag{1b}$$

Straightforward discretization of these equations will produce a system of equations of the form

$$\begin{pmatrix} A & G \\ D & 0 \end{pmatrix}\begin{pmatrix} \mathbf{v}^{n+1} \\ p^{n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{r} \\ 0 \end{pmatrix} + \begin{pmatrix} b\cdot c's \\ b\cdot c's \end{pmatrix}, \tag{2}$$

where A, G, and D are submatrices, $\mathbf{v}^{n+1}$ and p^{n+1} are the unknown discrete velocity and pressure vectors, and the right-hand side vector, $\mathbf{r}$, contains all those quantities that are already known.

For example, if the diffusive terms are updated using the trapezoidal or Crank–Nicholson method, the convective terms are updated using the second-order Adams–Bashforth method, and the spatial discretization is accomplished with a finite volume discretization or with lumped finite elements, then the Navier–Stokes equations become

$$\frac{(\mathbf{v}^{n+1}-\mathbf{v}^n)}{\Delta t} + \left[\frac{3}{2}\mathbf{H}(\mathbf{v}^n) - \frac{1}{2}\mathbf{H}(\mathbf{v}^{n-1})\right] = -Gp^{n+1} + \frac{1}{2\,\mathrm{Re}}L(\mathbf{v}^{n+1}+\mathbf{v}^n) + (b\cdot c's), \tag{3a}$$

$$D\mathbf{v}^{n+1} = 0 + (b\cdot c's), \tag{3b}$$

where $\mathbf{H}(\mathbf{v}^n)$ is the discrete convective operator (presumably evaluated in some conservative fashion), L is the discrete Laplacian operator, D is the discrete divergence operator, and G is the discrete gradient operator.

It is important to note that the boundary conditions have already been incorporated at this point. The exact forms of the matrices L, D, and G are dependent on the boundary conditions. The unknowns, $\mathbf{v}^{n+1}$ and p^{n+1} refer to only those nodes in the interior of the domain, not to boundary nodes. If it is assumed that all pressure nodes are interior to the domain (which is almost always the case), then boundary conditions on the pressure are not required [3]. This is most easily seen by expanding (3a) for a node near the boundary. In addition, for periodic, homogeneous

0021-9991/93 $5.00

no-slip, or no stress boundary conditions on the velocity, the boundary condition vectors are identically zero. They are retained throughout this analysis for generality.

Rearranging the relations (3a) and (3b) in the form given by Eq. (2), it is found that

$$A=\frac{1}{\Delta t}\left[I-\frac{\Delta t}{2\,\mathrm{Re}}L\right], \tag{4a}$$

$$\mathbf{r}=\frac{1}{\Delta t}\left[I+\frac{\Delta t}{2\,\mathrm{Re}}L\right]\mathbf{v}^n-\left[\frac{3}{2}\mathbf{H}(\mathbf{v}^n)-\frac{1}{2}\mathbf{H}(\mathbf{v}^{n-1})\right]. \tag{4b}$$

The exact expressions for A and $\mathbf{r}$ are dependent on the specific temporal and spatial discretizations that were chosen, but the previous example is fairly typical, implicit on the diffusive terms for stability and explicit on the convective terms for simplicity. For a fully explicit method, the matrix A is simply $1/\Delta t$ times the identity matrix. For a fully implicit method, the matrix A would contain elements from the linearized convective terms. An unlumped finite element method might have $A=(1/\Delta t)[M-(\Delta t/2\,\mathrm{Re})L]$, where M is a sparse "mass matrix." In any case, the resulting equations can be put in the general form given by (2).

It is possible to create primative variable formulations that avoid the solution for p^{n+1} altogether and do not have the form given by Eq. (2). These methods avoid most of the problems inherent in solving for the pressure. This is accomplished by employing a judicious manipulation of the equations [6, 7] or a clever choice of basis functions [8, 9]. Unfortunately, these methods inevitably require spatial operators with at least fourth-order accuracy or greater, and often they require simple geometries or boundary conditions. For these reasons discretizations of the form given by (2) are often desirable or inevitable.

Having committed ourselves to a formulation of the form given by Eq. (2), we must note that this system is not easy to solve. Although the system is typically sparse, it is also large and indefinite. The submatrices D and G are not square, and only the submatrix A is invertable. Ideally, we would like to simplify the problem and take advantage of some of the properties of the submatrices that might make inversion easier. For instance, usually $D=G^T$, and often A is symmetric and positive definite. On Cartesian finite difference or finite volume grids, A is block diagonal and can be further factored into a series of tridiagonals. The next section shows how these properties of the submatrices can be fully utilized.

2. BLOCK LU DECOMPOSITION

Given a system of equations, with a form such as Eq. (2), the most obvious way to remove the indefiniteness and solve the problem is to approximate the divergence equation and solve the system

$$\begin{pmatrix}A & G\\ D & \lambda I\end{pmatrix}\begin{pmatrix}\mathbf{v}^{n+1}\\ p^{n+1}\end{pmatrix}=\begin{pmatrix}\mathbf{r}\\ \lambda p^n\end{pmatrix}+\begin{pmatrix}b\cdot c's\\ b\cdot c's\end{pmatrix}, \tag{5}$$

with λ as small as numerically possible. This is typically referred to as an artificial compressibility method or a penalty method [10], and although it is commonly used, it has a number of major drawbacks. For unsteady problems the accuracy is $O(\lambda)$. The appropriate value of the parameter λ is not obvious (overly small values of λ make the system ill-conditioned), and the size of the system has not been reduced. Finally, the discrete continuity equation is violated.

Instead, let us consider another approximation to Eq. (2),

$$\begin{pmatrix}A & (\Delta t\,A)G\\ D & 0\end{pmatrix}\begin{pmatrix}\mathbf{v}^{n+1}\\ p^{n+1}\end{pmatrix}=\begin{pmatrix}\mathbf{r}\\ 0\end{pmatrix}+\begin{pmatrix}b\cdot c's\\ b\cdot c's\end{pmatrix}. \tag{6}$$

In this case the pressure gradient term in the momentum equations has been altered. For our previous example $(\Delta t\,A)=[I-(\Delta t/2\,\mathrm{Re})L]$, so the relation (6) is a first-order temporal approximation of Eq. (2) with an error term $(\Delta t/2\,\mathrm{Re})\,LGp^{n+1}$. Fortunately, the error term is diffusive and tends to aid stability. More complicated expressions for A are easily covered by the generalized method to be presented later. Note that only the momentum equations are approximated. The discrete continuity equation remains unaltered, since it is important to satisfy the discrete continuity equations (and not create mass), if a realistic solution is desired.

The approximate system given by (6) can be factored into the block LU decomposition

$$\begin{pmatrix}A & 0\\ D & -\Delta t\,DG\end{pmatrix}\begin{pmatrix}\mathbf{v}^*\\ p^{n+1}\end{pmatrix}=\begin{pmatrix}\mathbf{r}\\ 0\end{pmatrix}+\begin{pmatrix}b\cdot c's\\ b\cdot c's\end{pmatrix} \tag{7a}$$

and

$$\begin{pmatrix}I & \Delta t\,G\\ 0 & I\end{pmatrix}\begin{pmatrix}\mathbf{v}^{n+1}\\ p^{n+1}\end{pmatrix}=\begin{pmatrix}\mathbf{v}^*\\ p^{n+1}\end{pmatrix}. \tag{7b}$$

Further simplified this results in the series of operations

$$A\mathbf{v}^*=\mathbf{r}+b\cdot c's, \tag{8a}$$

$$\Delta t\,DGp^{n+1}=D\mathbf{v}^*-b\cdot c's, \tag{8b}$$

$$\mathbf{v}^{n+1}=\mathbf{v}^*-\Delta t\,Gp^{n+1}. \tag{8c}$$

Note that $\mathbf{v}^*$ is simply an intermediate variable, defined by either (7a) or (8a). The boundary condition vectors are the same as those defined in (2). There is no ambiguity in the

operations given by the relations (8a)–(8c). In particular, boundary conditions on $\mathbf{v}^*$ and p^{n+1} are not required.

In addition to preserving discrete incompressibility, the block LU factorization embodied by (8a)–(8c) has decomposed a large indefinite system into two smaller, far better behaved problems, a matrix inversion for $\mathbf{v}^*$ and a discrete Poisson-like equation for p^{n+1}. The disadvantage of first-order temporal accuracy will soon be remedied by the generalized block LU decomposition.

3. SIMILARITY WITH THE FRACTIONAL STEP METHOD

This section shows that the block LU decomposition is essentially a fully discrete version of the fractional step method, which is usually presented as a semi-discrete time splitting. The subtle differences in approach, however, are enough to enable the block LU decomposition to overcome most of the problems encountered by the traditional fractional step method.

The fractional step method begins from a time discretization of the Navier–Stokes equations. For simplicity we use the previous example of Adams–Bashforth for the convective terms and trapezoidal for the diffusive terms. This results in

$$\frac{(\mathbf{u}^{n+1}-\mathbf{u}^n)}{\Delta t}+\left[\frac{3}{2}(\mathbf{u}^n\cdot\nabla)\,\mathbf{u}^n-\frac{1}{2}(\mathbf{u}^{n-1}\cdot\nabla)\,\mathbf{u}^{n-1}\right]$$

$$=-\nabla p^{n+1}+\frac{1}{2\,\mathrm{Re}}\nabla^2(\mathbf{u}^{n+1}+\mathbf{u}^n), \tag{9a}$$

$$\nabla\cdot\mathbf{u}^{n+1}=0. \tag{9b}$$

The idea is to approximate (9a) by calculating a tentative velocity, $\mathbf{u}^*$, using the momentum equations without the pressure, and then use the pressure to project the tentative velocity into the space of discretely incompressible functions thereby finding the final velocity. Mathematically this looks like a simple time splitting of (9a),

$$\frac{\mathbf{u}^*-\mathbf{u}^n}{\Delta t}+\left[\frac{3}{2}(\mathbf{u}^n\cdot\nabla)\,\mathbf{u}^n-\frac{1}{2}(\mathbf{u}^{n-1}\cdot\nabla)\,\mathbf{u}^{n-1}\right]$$

$$=\frac{1}{2\,\mathrm{Re}}\nabla^2(\mathbf{u}^*+\mathbf{u}^n). \tag{10a}$$

$$\frac{\mathbf{u}^{n+1}-\mathbf{u}^*}{\Delta t}=-\nabla p^{n+1}. \tag{10b}$$

The pressure in (10b) is found by taking the divergence of Eq. (10b) and by invoking the incompressibility condition, Eq. (9b). This results in the Poisson equation for the pressure,

$$(\nabla\cdot\nabla)\,p^{n+1}=\frac{1}{\Delta t}\nabla\cdot\mathbf{u}^*. \tag{10c}$$

So the preceding equations are solved in the order (10a), (10c), (10b), at every time step. Rearranging both the equations and their order gives

$$\frac{1}{\Delta t}\left[I-\frac{\Delta t}{2\,\mathrm{Re}}\nabla^2\right]\mathbf{u}^*$$

$$=\frac{1}{\Delta t}\left[I+\frac{\Delta t}{2\,\mathrm{Re}}\nabla^2\right]\mathbf{u}^n$$

$$-\left[\frac{3}{2}(\mathbf{u}^n\cdot\nabla)\,\mathbf{u}^n-\frac{1}{2}(\mathbf{u}^{n-1}\cdot\nabla)\,\mathbf{u}^{n-1}\right], \tag{11a}$$

$$\Delta t(\nabla\cdot\nabla)\,p^{n+1}=\nabla\cdot\mathbf{u}^*, \tag{11b}$$

$$\mathbf{u}^{n+1}=\mathbf{u}^*-\Delta t\,\nabla p^{n+1}. \tag{11c}$$

The relations (11a)–(11c) bear a striking resemblance to the block LU decomposition given by (8a)–(8c). In fact, if the definitions of the matrix A and the vector $\mathbf{r}$ are reviewed, it is clear that the fractional step method is simply the continuous analog of the block LU decomposition. This analysis is reinforced by the fact that the error term, found by adding (10a) and (10b) together and comparing to (9a), is $(\Delta t/2\,\mathrm{Re})\,\nabla^2\nabla p^{n+1}$, which is also just the continuous version of what was found for the block LU decomposition.

Traditionally, it is only at this point in the fractional step method that (11a)–(11c) are spatially discretized. This makes the method independent of any particular spatial discretization scheme. It also causes two fundamental problems. Boundary conditions for the tentative velocity and the pressure are now required (a task which has caused considerable debate [3, 10, 11, 12]), and improving on the first-order accuracy of the method becomes very difficult [13, 14].

The subtle differences between the traditional fractional step method and the block LU decomposition are due to the point at which spatial discretization and boundary conditions are implemented. By implementing boundary conditions before any splitting or decomposition takes place, the block LU factorization does not require any boundary conditions on intermediate variables or on the pressure. All boundary condition information is already incorporated in the submatrices and the boundary condition vector.

If the traditional fractional step formulation is still preferred, or if one desires to analyze the appropriate boundary conditions for such a formulation, this can be accomplished by working backwards. That is, "correct" boundary conditions are those which, when spatially discretized, will cause the system to recover the form given by the block LU formulation. For example, for a no-slip boundary, correct conditions are $\mathbf{u}^*=0$ and $\partial p^{n+1}/\partial n=0$ on the boundary. This is because when discretized, $\nabla\cdot$ with Dirichlet boundary conditions is equivalent to the operator D, and the discretization of ∇ with Neumann boundary conditions is equivalent to the operator G.

Attempts to improve the time order of accuracy of the fractional step method through improved boundary conditions on intermediate variables and pressure appear to be misled. The block LU decomposition shows that boundary conditions are not an issue and that first-order accuracy in time is a fundamental result of the method itself.

4. THE GENERALIZED BLOCK LU DECOMPOSITION

Since we have an expression for the error term, it is possible to use this information to create a second-order method. This is done by approximating (2) by

$$\begin{pmatrix} A & (\Delta t\, A)G \\ D & 0 \end{pmatrix}\begin{pmatrix} \mathbf{v}^{n+1} \\ p^{n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{r} - (\Delta t/2\ \mathrm{Re})\, LGp^n \\ 0 \end{pmatrix} + \begin{pmatrix} b \cdot c's \\ b \cdot c's \end{pmatrix}, \tag{12}$$

where the right-hand side vector, $\mathbf{r}$, has been augmented by an extra term. The error term now becomes $((\Delta t)^2/2\ \mathrm{Re})\, LG((p^{n+1} - p^n)/\Delta t)$. Although previously proposed [13], this method has not to the author's knowledge been implemented. It has suspect stability properties due to the ad hoc modification of the right-hand side vector.

Instead, consider the following generalization of the block LU factorization method. Introduce the matrix B, and then approximate (2) with

$$\begin{pmatrix} A & (AB)G \\ D & 0 \end{pmatrix}\begin{pmatrix} \mathbf{v}^{n+1} \\ p^{n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{r} \\ 0 \end{pmatrix} + \begin{pmatrix} b \cdot c's \\ b \cdot c's \end{pmatrix}. \tag{13}$$

This can be factored into

$$\begin{pmatrix} A & 0 \\ D & -DBG \end{pmatrix}\begin{pmatrix} \mathbf{v}^{*} \\ p^{n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{r} \\ 0 \end{pmatrix} + \begin{pmatrix} b \cdot c's \\ b \cdot c's \end{pmatrix}, \tag{14a}$$

and

$$\begin{pmatrix} I & BG \\ 0 & I \end{pmatrix}\begin{pmatrix} \mathbf{v}^{n+1} \\ p^{n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{v}^{*} \\ p^{n+1} \end{pmatrix}, \tag{14b}$$

which reduces to the following sequence of operations:

$$A\mathbf{v}^{*} = \mathbf{r} + b \cdot c's, \tag{15a}$$

$$DBGp^{n+1} = D\mathbf{v}^{*} - b \cdot c's, \tag{15b}$$

$$\mathbf{v}^{n+1} = \mathbf{v}^{*} - BGp^{n+1}. \tag{15c}$$

If B is chosen equal to Δt times the identity matrix, then the generalization reverts to the original first-order block LU decomposition. However, if B is chosen to be an approximate inverse of A, then higher order accuracy can be achieved. For example, if $A = (1/\Delta t)[I - (\Delta t/2\ \mathrm{Re})L]$ as in our previous examples, then choosing $B = \Delta t\,[I + (\Delta t/2\ \mathrm{Re})L]$ results in a second-order error term $(\Delta t/2\ \mathrm{Re})^2\, L^2Gp^{n+1}$. Choosing $B = \Delta t[I + (\Delta t/2\ \mathrm{Re})L + (\Delta t/2\ \mathrm{Re})^2\, L^2]$ results in a third-order error term, and so on. Of course, third-order and higher approximations are a waste of effort in this case, since the original discretization is only second-order accurate to begin with. But higher order decompositions are certainly possible. As an aside, if $B = A^{-1}$ is chosen, then the generalized block LU decomposition is equivalent to the Uzawa method [15]. Uzawa methods involve nested iteration and have traditionally been avoided for that reason. The generalized block LU decomposition is a more viable solution, providing additional accuracy at the price of only slightly increased computational work.

5. APPLICATION TO THE STEADY STOKES EQUATIONS

It may not be apparent that the generalized block LU decomposition technique applies to a far wider class of problems than the traditional time split fractional step method. To illustrate this fact we outline how the method could be used to solve the *steady* Stokes equations. These equations are

$$\nabla^2\mathbf{v} - \nabla p + \mathbf{f} = 0, \tag{16a}$$

$$\nabla \cdot \mathbf{v} = 0. \tag{16b}$$

Full discretization of these equations gives

$$\begin{pmatrix} -L & G \\ D & 0 \end{pmatrix}\begin{pmatrix} \mathbf{v} \\ p \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ 0 \end{pmatrix} + \begin{pmatrix} b \cdot c's \\ b \cdot c's \end{pmatrix}. \tag{17}$$

This matrix problem has the same form as (2), with $A = -L$ and $\mathbf{r} = \mathbf{f}$, and so it can be solved in the same manner. If an approximate inverse, B, exists such that the product AB does not ruin the overall accuracy, then the generalized block LU factorization will not alter the overall accuracy of the discretization. It is quite likely that B will be available. If A is being inverted via an iterative method, then it makes sense to precondition A with an approximate inverse [16]. If the preconditioner is sufficiently good, we can use it for B as well. The incomplete Cholesky factorization of A is a good choice for B, since B will retain the same sparsity structure as A.

6. PRESSURE ISSUES

The pressure is a very interesting variable in the context of numerical discretizations of the incompressible Navier–Stokes equations. As has already been demonstrated, boundary conditions on the discrete pressure are usually unnecessary. A number of other complicated issues

surrounding the pressure also exist and have been put off until this point so that they could be addressed in full.

The role of the discrete pressure is to act as a Lagrange multiplier. It implicitly guarantees that discrete continuity is satisfied at every time step. This role of the discrete pressure has two important consequences. First, the order of accuracy of the pressure update does not effect the order of accuracy of the velocity field. This is discussed in a paper by Temam [17]. And second, no matter what scheme is used, the discrete pressure will always be first-order accurate in time. This is because the actual pressure is a solution of $\nabla^2 p = \nabla \cdot (\mathbf{u} \cdot \nabla \mathbf{u})$, whereas the discrete pressure is a solution of $\nabla^2 p^{n+1} = (1/\Delta t)\nabla \cdot \mathbf{v}^*$ and the two source terms are only identical to first-order in time. These conclusions are confirmed by the numerical results presented in Section 7.

Let us consider the consequences of using a more complicated pressure update scheme. A generalized scheme for the pressure can be written as $\alpha G p^{n+1} + (1-\alpha) G p^n$, where α is a scaler varying from 0 to 1. Under these conditions Eq. (2) becomes

$$\begin{pmatrix} A & \alpha G \\ D & 0 \end{pmatrix} \begin{pmatrix} \mathbf{v}^{n+1} \\ p^{n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{r} - (1-\alpha)\,\Delta t\, G p^n \\ 0 \end{pmatrix} + \begin{pmatrix} b \cdot c's \\ b \cdot c's \end{pmatrix}, \quad (18)$$

and the standard block LU decomposition is

$$A\mathbf{v}^* = \mathbf{r} - (1-\alpha)\,\Delta t\, G p^n + b \cdot c's, \quad (19a)$$

$$\alpha\,\Delta t\, D G p^{n+1} = D\mathbf{v}^* - b \cdot c's, \quad (19b)$$

$$\mathbf{v}^{n+1} = \mathbf{v}^* - \alpha\,\Delta t\, G p^{n+1}. \quad (19c)$$

The error term now becomes $(\alpha\,\Delta t/2\,\mathrm{Re})\,LGp^{n+1}$.

When $\alpha = 1$ the previous implicit Euler method is recovered. When α is decreased to $\frac{1}{2}$, the pressure is updated with the trapezoidal method. The error in the momentum equations is accordingly reduced by a factor of $\alpha = \frac{1}{2}$, but remains first-order. The error term is the result of the LU factorization (or time splitting) and is not a result of the pressure update scheme. It is therefore not fundamentally altered by a change to a trapezoidal pressure update.

In addition, the trapezoidal pressure update scheme is only neutrally stable. The pressure term must account for the entire interval in time from time step n to time step $n+1$. If p^n is slightly underestimated then p^{n+1} must make up for this deficiency by a slight overestimation. In the next interval this will result in p^{n+1} being too large, and then p^{n+2} being too small. Because the pressure is defined only up to an arbitrary additive function of time, these oscillations do not propagate to the velocity field, but they eventually cause numerical difficulties. Choices of α less than one-half can significantly reduce the splitting error, but should be avoided since they are strongly unstable.

Finally, the discrete pressure has been used to develop an incorrect argument, suggesting that the fractional step method is second-order accurate in time. Using the traditional time discretized formulation, it is claimed that the error term can be absorbed into the pressure term by defining a new pressure variable ϕ^{n+1} such that

$$\nabla\phi^{n+1} = \nabla p^{n+1} - \frac{\alpha\,\Delta t}{2\,\mathrm{Re}} \nabla^2 \nabla p^{n+1}. \quad (20)$$

However, it is legal to define ϕ^{n+1} only if the right-hand side of (20) can indeed be represented as the gradient of a function. In the continuous case this is certainly true, since the operators ∇ and ∇^2 commute. However, our operators are actually discrete (even in the traditional fractional step method the operators must eventually be spatially discretized) and the question is, do those discrete operators commute? That is, the error term can be absorbed only if it can be written as the discrete gradient of a function. If $LG = GQ$, where Q is some matrix, then the fractional step method will indeed be second-order accurate. In general, the previous condition is *not* satisfied. A notable exception is when periodic boundary conditions are in effect. Then, even finite difference versions of the operator, G, satisfy the above requirement. This may explain why previous researchers thought it was their boundary conditions that were destroying the accuracy. Another reason may be that the error term contains a Laplacian operator, which tends to be larger near boundaries.

7. NUMERICAL SIMULATIONS OF THE NAVIER–STOKES EQUATIONS

To confirm the theoretical results presented in this paper, numerical simulations of the unsteady Navier–Stokes equations, in a square domain of unit length, were performed. The equations were solved using a second-order finite volume scheme on a uniform mesh of 64×64 cells. This spatial resolution was found to be more than adequate. The diffusive terms were updated using the trapezoidal method, and the pressure term was updated using either the implicit Euler or the trapezoidal method. Convective terms were updated explicitly using the second-order Adams–Bashforth method. The initial condition was a vortex flow given by

$$u^0 = (1 - \cos(2\pi x))\sin(2\pi y) \quad (21a)$$

$$v^0 = (\cos(2\pi y) - 1)\sin(2\pi x). \quad (21b)$$

Either no-slip or periodic boundary conditions were used at the domain boundaries.

Simulations were performed on a CM-2 [18] at a Reynolds number of 5000. Measurements were made at a time ($t = 0.1$) which corresponds to at least 10 iterations at the largest time step. Since an exact solution for this flow

does not exist, data were taken for several Δt, and the error was determined from the limiting value. Pressure error was measured in the L^2 norm, but velocity error was measured at a single point near the boundary (0.0625, 0.0625). This was to make sure that no "boundary layer" with a decreased order of convergence existed [12].

In the first series of tests we used no-slip boundary conditions and an implicit Euler pressure update. The generalized block LU decomposition was used with three different approximate inverses: case (a), $B = \Delta t\, I$, case (b), $B = \Delta t[I + (\Delta t/2\ \mathrm{Re})L]$, and case (c), $B = \Delta t[I + (\Delta t/2\ \mathrm{Re})L + (\Delta t/2\ \mathrm{Re})^2 L^2]$. The convergence of the velocity error for these three cases is shown in Fig. 1. Figure 1 shows that case (a), the standard block LU decomposition (or equivalently, the traditional fractional step method) is very definitely first order and that the first-order convergence can be remedied by using the approximate inverse given in case (b). Improving the approximate inverse to case (c) does not improve the order of accuracy, since the original discretization was only second-order accurate to begin with.

It is interesting to note that in case (b) the matrix B is positive definite only for small Δt. This means that the matrix to be inverted in the Poisson equation $DBG(\Delta t\, p^{n+1}) = D\mathbf{v}^* - (b \cdot c's)$ is also not positive definite and that a conjugate gradient method will not converge for the above equation. The limit on Δt such that B is guaranteed to be positive definite goes like $(\Delta x)^2$, which is very restrictive. We could use a solution technique for indefinite matrices, but that defeats much of the purpose of the block LU decomposition. The best solution is to use the approximate inverse defined in case (c), which does not improve the order of accuracy, but is always positive definite.

Also note that the family of approximate inverses that logically extend from cases (a)–(c) always results in an error term that has a positive coefficient and a diffusive nature. Therefore, the error term will always aid in the stability of the overall method. This does not imply that the method will be stable, but it does imply that the generalized block LU decomposition will not make an otherwise stable method unstable.

The next series of tests, shown in Fig. 2, examines the effect of boundary conditions and pressure update schemes on the velocity error. The approximate inverse was fixed at $B = \Delta t\, I$. Case (a) was a benchmark case with no-slip boundary conditions and an implicit Euler pressure update. As noted in Fig. 1 it is first-order. Case (b) is identical to the benchmark case, except that periodic boundary conditions were used. As predicted in the previous section, under these conditions the method is now second-order accurate in time. Case (c) uses no-slip boundary conditions, but employs a trapezoidal update for the pressure instead of an implicit Euler. For large Δt the error approaches a factor of one-half of the benchmark case. For smaller Δt the error is dominated by an error in the convective terms rather than the error in the factorization, and it is therefore identical to the benchmark case.

To remove the complication of error in the convective terms, Fig. 3 shows similar results from solutions of the unsteady Stokes equations, where the convective terms are

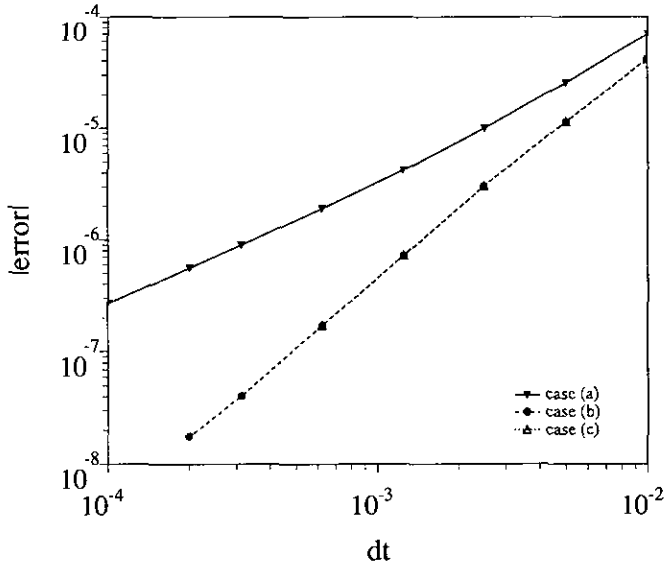

FIG. 1. Temporal convergence of the velocity error when the generalized block LU decomposition method is used. Case (a) uses the "standard" first-order approximate inverse, no-slip boundary conditions, and an implicit Euler update for the pressure. Cases (b) and (c) use second- and third-order approximate inverses, respectively.

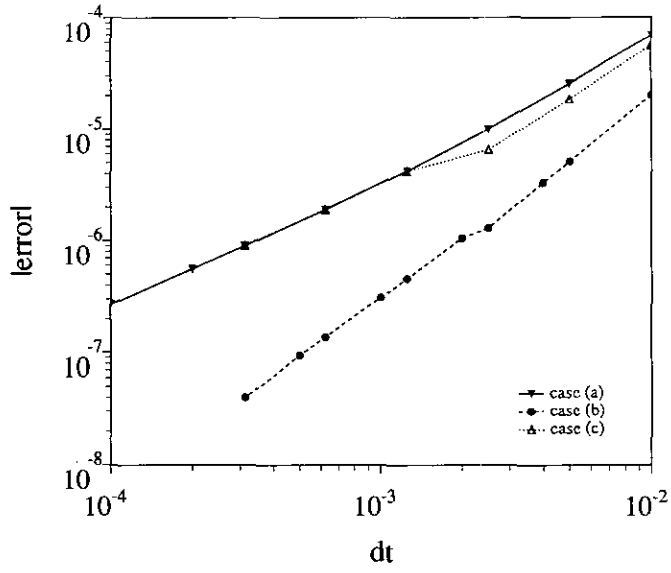

FIG. 2. Temporal convergence of the velocity error. Case (a) is the "standard" method. Case (b) is the "standard" method, but using periodic boundary conditions instead of no-slip boundary conditions. Case (c) is the "standard" method but uses a second-order trapezoidal update for the pressure, instead of implicit Euler.

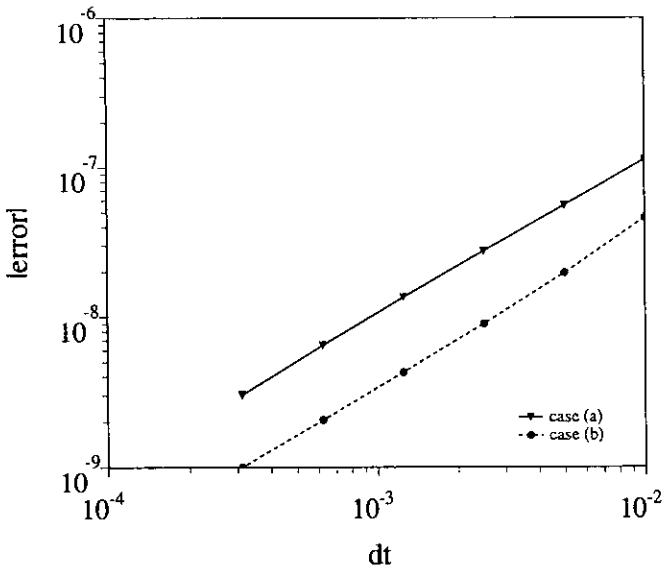

FIG. 3. Temporal convergence of the velocity error for the Stokes equations. Case (a) uses an implicit Euler pressure update. Case (b) uses trapezoidal pressure update.

identically zero. Case (a) uses an implicit Euler pressure update, and case (b) uses a trapezoidal pressure update. This demonstrates more clearly that the trapezoidal pressure update reduces the factorization error by roughly one-half but does not change the first-order convergence behavior.

The final series of tests is concerned with the convergence of the pressure and is presented in Fig. 4. Case (a) uses

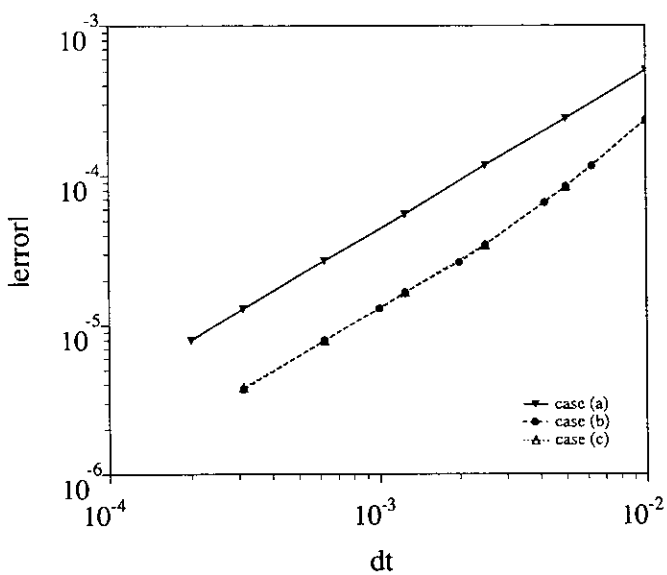

FIG. 4. Temporal convergence of the pressure error (L_2). Case (a) uses a second-order approximate inverse, no-slip boundary conditions, and implicit Euler update for the pressure. Case (b) uses the second-order trapezoidal update for the pressure. Case (c) also uses the second-order trapezoidal update for the pressure, but only a first-order approximate inverse.

no-slip boundary conditions, the second-order approximate inverse, and an implicit Euler for the pressure update. Case (b) switches to a trapezoidal pressure update. Case (c) uses the first-order approximate inverse, with the second-order trapezoidal pressure update. In all cases the pressure error is only first-order. However, the trapezoidal method does reduce the error by approximately a factor of one-half over the implicit Euler method.

The numerical results confirm a number of important facts. The traditional fractional step method is only second-order accurate in time for periodic boundary conditions (or special forms of the operator, G). The first-order accuracy can be improved by using the generalized block LU decomposition and is irrespective of the order of accuracy of the pressure update. This is fortunate since the pressure is always first-order accurate in time.

8. SUMMARY

By viewing the fractional step method as an approximate block LU decomposition it has been shown that boundary condition issues can be trivially solved and time accuracy increased. It may be possible that ever more complicated expressions for the boundary conditions on intermediate variables and the pressure could improve the accuracy of the traditional fractional step method [14]. Certainly, boundary conditions (for example, periodic boundary conditions) can intimately affect the temporal order of convergence. However, it is hard to argue against the fact that the simplest solution is to avoid the issue of boundary conditions on temporary variables altogether, as the block LU decomposition does.

The elegance of the generalized block LU decomposition is further witnessed by its ability to solve any system of equations of the form given by Eq. (2), not just those arising from time-dependent problems. In addition, the formulation is conducive to analysis and has led to the prediction and explanation of many curious phenomena associated with the fractional step method. Numerical experiments have confirmed the efficacy of that analysis.

ACKNOWLEDGMENTS

This work was funded by a grant from the National Science Foundation. Computations were carried out at the NAS Parallel Systems Facility of NASA-Ames Research Center. In addition, I am greatly indebted to Sandy Wilson for her thoughtful comments and to Haecheon Choi for many enlightening discussions.

REFERENCES

1. A. J. Chorin, *Math. Comput.* **22**, 745 (1968).
2. R. Temam, *Arch. Rat. Mech. Anal.* **32**, 377 (1969).

3. J. Kim and P. Moin, *J. Comput. Phys.* **59**, 308 (1985).
4. T. Zang and M. Hussaini, *Appl. Math. Comput.* **19**, 359 (1986).
5. N. N. Yanenko, *The Method of Fractional Steps* (Springer-Verlag, New York, 1971).
6. J. Kim, P. Moin, and R. Moser, *J. Comput. Phys.* **177**, 133 (1987).
7. C. Canuto, M. Hussaini, A. Quarteroni, and T. Zang, *Spectral Methods in Fluid Dynamics* (Springer-Verlag, New York, 1988).
8. P. R. Spalart, NASA TM 88222 (1986).
9. V. Girault and P. Raviart, *Finite Element Approximation of the Navier–Stokes Equations* (Springer-Verlag, Berlin, 1986).
10. R. Peyret and D. Taylor, *Computational Methods for Fluid Flow* (Springer-Verlag, Berlin, 1983).
11. S. Orszag, M. Israeli, and M. Deville, *J. Sci. Comput.* **1**, 75 (1986).
12. A. Tomboulides, M. Israeli, and G. Karniadakis, *J. Sci. Comput.* **4**, 291 (1989).
13. Y. Maday, A. Patera, and E. Ronquist, *J. Sci. Comput.* **5**, 263 (1990) (note conclusion).
14. G. Karniadakis, M. Israeli, and S. Orszag, *J. Comput. Phys.* **97**, 414 (1991).
15. R. Temam, *Navier–Stokes Equations: Theory and Numerical Analysis* (North-Holland, New York, 1979).
16. G. Golub and C. Van Loan, *Matrix Computations* (Johns Hopkins Univ. Press, Baltimore, 1983).
17. R. Temam, *Theor. Comput. Fluid Dynamics*, personal communication.
18. J. B. Perot, *Proceedings, Conference on Parallel Computational Fluid Dynamics*, edited by J. Hauser (Elsevier Science, Amsterdam, 1992).