NASA-CR-201051

RIACS

Research Institute for Advanced Computer Science
NASA Ames Research Center

A fast Poisson solver for unsteady incompressible Navier-Stokes Equations on the half-staggered grid

G. H. Golub, L. C. Huang, H. Simon and W.-P. Tang

RIACS Technical Report 95.07 April 1995

A fast Poisson solver for unsteady incompressible Navier-Stokes Equations on the half-staggered grid

G. H. Golub, L. C. Huang, H. Simon and W.-P. Tang

The Research Institute for Advanced Computer Science is operated by Universities Space Research Association, The American City Building, Suite 212, Columbia, MD 21044 (410)730-2656

Work reported herein was supported in part by NASA under contract NAS 2-13721 between NASA and the Universities Space Research Association (USRA).

A fast Poisson solver for unsteady incompressible Navier-Stokes Equations on the half-staggered grid

Abstract. In this paper, a fast Poisson solver for unsteady, incompressible Navier-Stokes equations with finite difference methods on the non-uniform, half-staggered grid is presented. To achieve this, new algorithms for diagonalizing a semi-definite pair are developed. Our fast solver can also be extended to the three dimensional case. The motivation and related issues in using this second kind of staggered grid are also discussed. Numerical testing has indicated the effectiveness of this algorithm.

Key words. fast solver, generalized eigenvalues, incompressible Navier-Stokes equations.

AMS(MOS) subject classifications. 15A22, 35Q30, 65F15, 65M06, 76D05

1. Introduction. Consider the unsteady incompressible Navier-Stokes equations (INSE)

$$\frac{\partial \mathbf{w}}{\partial t} + u\frac{\partial \mathbf{w}}{\partial x} + v\frac{\partial \mathbf{w}}{\partial y} + grad\ p = \alpha\ div\ grad\ \mathbf{w} \tag{1}$$

$$div\ \mathbf{w} = 0 \tag{2}$$

where $\mathbf{w} = (u, v)'$, in a two-dimensional region Ω with initial condition

$$\mathbf{w}(x, y, 0) = \mathbf{w}^0(x, y) \qquad \text{in } \Omega$$

satisfying the constraint condition (2), and boundary condition

$$\mathbf{w}(x, y, t) = \mathbf{w}_b(x, y, t) \qquad \text{on } \partial\Omega.$$

The boundary condition on $\partial\Omega$ satisfies the consistency condition

$$\oint_{\partial\Omega} \mathbf{w}_n ds = 0. \tag{3}$$

The main difficulty in the solution of this problem is that the unsteady INSE is not entirely evolutionary; it is subject to the divergence-free constraint (2). The projection method, proposed by Chorin [3] and Temam [20], proved to be very effective and its variants are widely used in many applications. With this method, some auxiliary velocity is found and then projected into the divergence-free space via the solution of a Poisson equation for some form of the pressure. For example, let the discretization of (1) and (2), after linearization of nonlinear terms, be

$$\frac{\dot{\Delta}\mathbf{w}}{\Delta t} + A\dot{\Delta}\mathbf{w} + \nabla\phi = E \tag{4}$$

$$\nabla \cdot \mathbf{w} = 0 \tag{5}$$

where $\dot{\Delta}\mathbf{w} = \mathbf{w}^{n+1} - \mathbf{w}^n$, A represents the discretization of convection and diffusion, ∇ and $\nabla\cdot$ the discretization of *grad* and *div* respectively, $\phi = p^{n+1} - p^n$ the pressure increment, and E includes all known terms at the nth level. With an approximate factorization, Eq. (4) is put into the following fractional steps

$$\frac{\tilde{\Delta}\mathbf{w}}{\Delta t} + A\tilde{\Delta}\mathbf{w} = E \tag{6}$$

where $\tilde{\Delta}\mathbf{w} = \tilde{\mathbf{w}} - \mathbf{w}^n$ ($\tilde{\mathbf{w}}$ is called the auxiliary velocity) and

$$\frac{\mathbf{w}^{n+1} - \tilde{\mathbf{w}}}{\Delta t} + \nabla\phi = 0. \tag{7}$$

Now, applying the discrete divergence to (7), we obtain the discrete Poisson equation for ϕ:

$$\nabla^2\phi = \frac{\nabla \cdot \tilde{\mathbf{w}}}{\Delta t} \tag{8}$$

in which the discrete divergence free condition (5), $\nabla \cdot \mathbf{w}^{n+1} = 0$ has been enforced. The solution process is then:

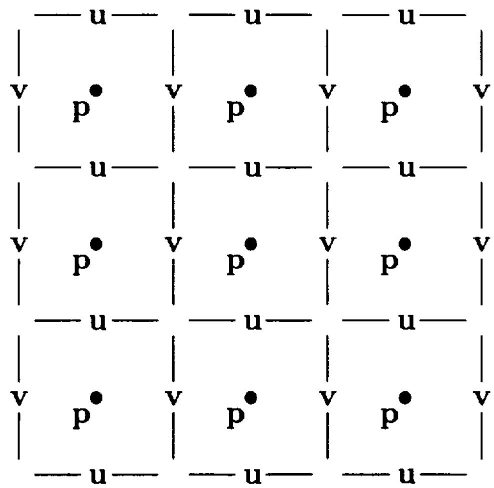

FIG. 1. *Staggered Grid*

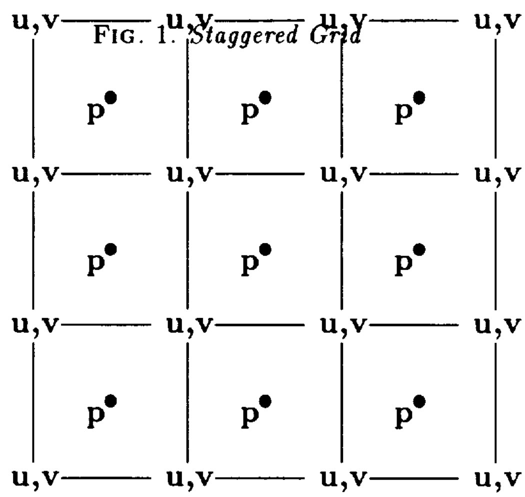

FIG. 2. *Half-Staggered Grid*

- Find $\tilde{\mathbf{w}}$ by (6).
- Solve the discrete Poisson equation (8) for ϕ.
- Update $\mathbf{w}^{n+1}$ by (7).

For a more detailed account with the boundary condition treatment , see [10].

The constraint (5) causes difficulties in the choice of the mesh, the coordinate system and the related discrete Poisson equation, etc., of the solution method. For a rectangular region Ω, the usual staggered mesh (u, v and p staggered mesh as shown in Fig. 1) is most frequently used. With this mesh, no pressure boundary condition is needed, which is mathematically correct. Let the discretized linear system of equations of (8) be denoted as

$$L\Phi = R, \tag{9}$$

where Φ is the unknown vector of size, say, M. With this mesh, L is of rank $M-1$; the constraint on the right hand side R reduces to an approximation of (3), and the solution Φ is unique up to a constant. For a rectangular region with uniform or nonuniform mesh intervals, direct solvers from the FISHPACK have proved to be most efficient, see [2, 16] for example. Also for the usual staggered mesh, the finite difference approximation of (1) is the same as the finite volume approximation. In addition, ∇p and $\nabla \cdot \mathbf{w}$ are straightforward without any interpolation. However, u and v involve different finite volumes and near the boundary half interval differencing is necessary, which are inconveniences. More serious is the problem on curvilinear meshes in arbitrary regions; there the extension of the usual staggered mesh will lose many of the above advantages, see [17] for instance. Hence, the half-staggered mesh ($\mathbf{w}$, p staggered as shown in Fig. 2 was studied in [10, 11]. This mesh was first proposed in [5] for the unsteady INSE; it was recently used in [1] with a different discretization of projection, for example; but it is not used in practical computation in general. For steady-state INSE, this mesh was used and investigated for FEM in [15] and for the Galerkin formulation in [18].

The advantages of this mesh for the unsteady INSE are: no pressure boundary condition is required; the finite difference approximation of (1) is the same as the finite volume method with the same finite volume for u and v; ∇p and $\nabla\cdot\mathbf{w}$ involve only simple averages, and there is no half interval differencing near the boundary. Most important is the extension of this staggering to curvilinear meshes in arbitrary regions. As the velocity components are at the same point, it is easily transformed to, say, contravariant components for finite volume formation, for approximation of $div\ \mathbf{w}$, and for boundary conditions, etc.. The disadvantage of this mesh is the nature of the discrete Poisson equation (8), where L is of rank $M-2$. There is an additional constraint and the solution Φ may have oscillations; see the next section.

As the first step to this problem, we consider rectangular regions with nonuniform meshes. Since (9) is to be solved at every time step for the unsteady INSE, efficient solvers are necessary and this is the subject of this paper.

2. Discretization. Consider a rectangular region with a nonuniform mesh, schematically show in Fig. 3. We generate the nonuniform mesh by some smooth transformation $x(\zeta)$, $y(\eta)$, with a uniform mesh in the computational $\zeta\eta$ region, and approximate

$$\frac{\partial}{\partial x}=\frac{d\zeta}{dx}\frac{\partial}{\partial\zeta} \qquad \text{by} \qquad \frac{\delta}{\delta x}=\frac{\Delta\zeta}{\delta x}\frac{\delta}{\Delta\zeta}$$

where δ on the right hand side denotes centered differencing, and similarly for $\partial/\partial y$. Then for every interior $\mathbf{w}$ point

$$\nabla\,\phi|_{j+\frac{1}{2},k+\frac{1}{2}}=\begin{bmatrix}\frac{\partial\phi}{\partial x}\\ \frac{\partial\phi}{\partial y}\end{bmatrix}_{j+\frac{1}{2},k+\frac{1}{2}}=\frac{1}{2}\begin{bmatrix}\frac{\phi_{j+1,k+1}-\phi_{j,k+1}}{x_{j+1}-x_j}+\frac{\phi_{j+1,k}-\phi_{j,k}}{x_{j+1}-x_j}\\ \frac{\phi_{j+1,k+1}-\phi_{j+1,k}}{y_{k+1}-y_k}+\frac{\phi_{j,k+1}-\phi_{j,k}}{y_{k+1}-x_k}\end{bmatrix} \tag{10}$$

and (7) on all the interior $\mathbf{w}$ point can be written as

$$\frac{W^{n+1}-\widetilde{W}}{\Delta t}+G\Phi=0 \tag{11}$$

where G is the gradient matrix for the solution vector Φ. For every interior ϕ point, or rather every (j,k) cell,

$$\nabla\cdot\mathbf{w}\,|_{j,k}\;=\;\frac{1}{2}\frac{1}{x_{j+\frac{1}{2}}-x_{j-\frac{1}{2}}}\left[(u_{j+\frac{1}{2},k+\frac{1}{2}}+u_{j+\frac{1}{2},k-\frac{1}{2}})-(u_{j-\frac{1}{2},k+\frac{1}{2}}+u_{j-\frac{1}{2},k-\frac{1}{2}})\right]$$

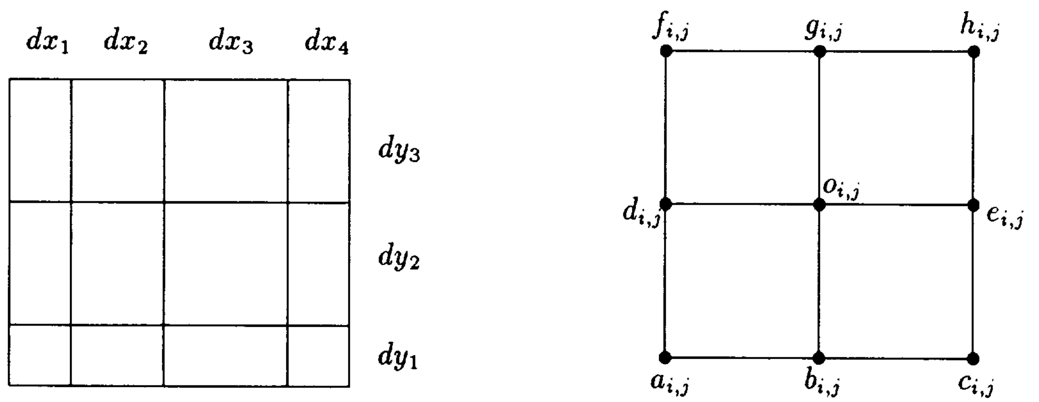

FIG. 3. *Non uniform mesh and the stencil*

$$+\frac{1}{2}\frac{1}{y_{k+\frac{1}{2}}-y_{k-\frac{1}{2}}}\left[(v_{j+\frac{1}{2},k+\frac{1}{2}}+v_{j-\frac{1}{2},k+\frac{1}{2}})-(v_{j+\frac{1}{2},k-\frac{1}{2}}+v_{j-\frac{1}{2},k-\frac{1}{2}})\right] \tag{12}$$

which results in a matrix form

$$DW=-B. \tag{13}$$

Here D is a matrix form of $\nabla\cdot$ with boundary modifications, i.e. for cells adjacent to the boundary, some of the elements in $\mathbf{w}$ in (13) are known and put into B. Hence, the final discrete system of equations is:

$$L\Phi = DG\Phi = R = \frac{1}{\Delta t}\left(D\widetilde{W}+B\right) \tag{14}$$

After some tedious derivation, the matrix L can written as

$$L = DG = \left[C_n \otimes D_x A_m + D_y B_n \otimes C_m\right],$$

which is a nine-diagonal unsymmetric matrix. Let $dx_i = x_{i+1} - x_i$, $dy_i = y_{i+1} - y_i$ and $dx_{i,i+1} = (dx_{i+1} + dx_i)/2$ and $dy_{i,i+1} = (dy_{i+1} + dy_i)/2$. Then the matrices in the expression of L can be written as

$$D_x = \begin{bmatrix} \frac{1}{dx_1} & & & \\ & \frac{1}{dx_2} & & \\ & & \ddots & \\ & & & \frac{1}{dx_n} \end{bmatrix}_{m\times m}$$

$$D_y = \begin{bmatrix} \frac{1}{dy_1} & & & \\ & \frac{1}{dy_2} & & \\ & & \ddots & \\ & & & \frac{1}{dy_n} \end{bmatrix}_{n\times n}$$

$$C_l = \begin{bmatrix} 1 & 1 & & & & \\ 1 & 2 & 1 & & & \\ & \ddots & & \ddots & & \\ & & & 1 & 2 & 1 \\ & & & & 1 & 1 \end{bmatrix}_{l\times l}$$

$$A_m = \begin{bmatrix} -\frac{1}{dx_{12}} & \frac{1}{dx_{12}} & & & \\ \frac{1}{dx_{12}} & -\left(\frac{1}{dx_{12}}+\frac{1}{dx_{23}}\right) & \frac{1}{dx_{23}} & & \\ & \ddots & \ddots & & \\ & & \ddots & \ddots & \\ & & \frac{1}{dx_{m-2,m-1}} & -\left(\frac{1}{dx_{m-2,m-1}}+\frac{1}{dx_{m-1,m}}\right) & \frac{1}{dx_{m-1,m}} \\ & & & \frac{1}{dx_{m-1,m}} & -\frac{1}{dx_{m-1,m}} \end{bmatrix};$$

$$B_n = \begin{bmatrix} -\frac{1}{dy_{12}} & \frac{1}{dy_{12}} & & & \\ \frac{1}{dy_{12}} & -\left(\frac{1}{dy_{12}}+\frac{1}{dy_{23}}\right) & \frac{1}{dy_{23}} & & \\ & \ddots & \ddots & & \\ & & \ddots & \ddots & \\ & & \frac{1}{dy_{n-2,n-1}} & -\left(\frac{1}{dy_{n-2,n-1}}+\frac{1}{dy_{n-1,n}}\right) & \frac{1}{dy_{n-1,n}} \\ & & & \frac{1}{dy_{n-1,n}} & -\frac{1}{dy_{n-1,n}} \end{bmatrix}_{n\times n}.$$

Here an $m \times n$ rectangular grid is assumed (Note that the n used here is different from that used in the time stepping process.) For the interior nodes, a nine-point stencil is presented in Fig. 3. The corresponding coefficients for a node (i, j) in that figure are:

$$\begin{aligned}
a_{i,j} &= \frac{1}{dx_i}\frac{1}{dx_{i-1,i}} + \frac{1}{dy_j}\frac{1}{dy_{j-1,j}} \\
b_{i,j} &= -\frac{1}{dx_i}\left(\frac{1}{dx_{i-1,i}} + \frac{1}{dx_{i,i+1}}\right) + 2\frac{1}{dy_j}\frac{1}{dy_{j-1,j}} \\
c_{i,j} &= \frac{1}{dx_i}\frac{1}{dx_{i,i+1}} + \frac{1}{dy_j}\frac{1}{dy_{j-1,j}} \\
d_{i,j} &= 2\frac{1}{dx_i}\frac{1}{dx_{i-1,i}} - \frac{1}{dy_j}\left(\frac{1}{dy_{j-1,j}} + \frac{1}{dy_{j,j+1}}\right) \\
e_{i,j} &= 2\frac{1}{dx_i}\frac{1}{dx_{i,i+1}} - \frac{1}{dy_j}\left(\frac{1}{dy_{j-1,j}} + \frac{1}{dy_{j,j+1}}\right) \\
f_{i,j} &= \frac{1}{dx_i}\frac{1}{dx_{i-1,i}} + \frac{1}{dy_j}\frac{1}{dy_{j,j+1}} \\
g_{i,j} &= -\frac{1}{dx_i}\left(\frac{1}{dx_{i-1,i}} + \frac{1}{dx_{i,i+1}}\right) + 2\frac{1}{dy_j}\frac{1}{dy_{j,j+1}}
\end{aligned}$$

$$
\begin{aligned}
h_{i,j} &= \frac{1}{dx_i}\frac{1}{dx_{i,i+1}} + \frac{1}{dy_j}\frac{1}{dy_{j,j+1}} \\
o_{i,j} &= -\left(a_{i,j} + c_{i,j} + f_{i,j} + h_{i,j}\right)
\end{aligned}
$$

It is not difficult to see that L is singular and in the next section we will show it is of rank $M-2$. The two base vectors of the null space $N(L)$ are

$$
\begin{aligned}
\phi_{01} &= (1,\ \ 1,1\cdots,\ \ 1)^T \otimes (\ 1,1,\ \ 1\cdots,1)^T \\
&= \mathbf{e}_n \otimes \mathbf{e}_m \\
\phi_{02} &= (1,-1,1\cdots,-1^n)^T \otimes (-1,1,-1\cdots,-1^{m-1})^T \\
&= \mathbf{e'}_n \otimes -\mathbf{e'}_m
\end{aligned}
$$

where

$$
\begin{aligned}
\mathbf{e}_l &= \underbrace{(1,1,1,\cdots,1)^T}_{l} \\
\mathbf{e'}_l &= \underbrace{(1,-1,1,\cdots,-1^{l-1})^T}_{l}.
\end{aligned}
$$

since $A_m\mathbf{e}_m = 0$, $B_n\mathbf{e}_n = 0$ and $C_l\mathbf{e'}_l = 0$.

We note from ϕ_{01} that the solution ϕ can differ by a constant which is irrelevant as our interest is only in $\nabla\phi$. From ϕ_{02}, we see that ϕ can have oscillations, called the checkerboard effect [15], and from (10), we also see that in the present finite difference method the oscillations in ϕ do not affect $\nabla\phi$ and hence the solution $\mathbf{w}$.

To ensure the solution of (14) exists, the right hand side R must satisfy

$$
R \in N(L^T)^{\perp}.
$$

Thus, R has to be orthogonal with the two base vectors of $N(L^T)$. For a nonuniform mesh, the two base vectors of the null space of the discrete operator L^T are

$$
\begin{aligned}
\psi_{01} &= \mathbf{dy}_n \otimes \mathbf{dx}_m \\
\psi_{02} &= \mathbf{e'}_n \otimes -\mathbf{e'}_m
\end{aligned}
$$

where

$$
\mathbf{dx}_m = \underbrace{(dx_1, dx_2, \cdots, dx_m)^T}_{m}.
$$

and

$$
\mathbf{dy}_n = \underbrace{(dy_1, dy_2, \cdots, dy_n)^T}_{n}.
$$

This can be verified by multiplying directly with the matrix L^T.

$$\begin{aligned}
L^T\psi_{01} &= [C_n \otimes A_m D_x + B_n D_y \otimes C_m]\,\mathbf{dy}_n \otimes \mathbf{dx}_m \\
&= C_n \mathbf{dy}_n \otimes A_m D_x \mathbf{dx}_m + B_n D_y \mathbf{dy}_n \otimes C_m \mathbf{dx}_m \\
&= 0 \\
L^T\psi_{02} &= [C_n \otimes A_m D_x + B_n D_y \otimes C_m]\,\mathbf{e'}_n \otimes -\mathbf{e'}_m \\
&= -C_n \mathbf{e'}_n \otimes A_m D_x \mathbf{e'}_m - B_n D_y \mathbf{e'}_n \otimes C_m \mathbf{e'}_m \\
&= 0
\end{aligned}$$

since $A_m D_x \mathbf{dx}_m = 0$, $B_n D_y \mathbf{dy}_n = 0$ and $C_l \mathbf{e'}_l = 0$.

In [15], Sani investigated the solution of (13) and gave the base vectors of $N(D^T)$ to be ψ_{01} and ψ_{02}. He showed that the constraint $\psi_{01}^T B = 0$ reduces to a discrete form of (3), and $\psi_{02}^T B = 0$ yields a constraint for the tangential velocity component on the boundary.

For the unsteady INSE, the interest is in the solution of (14). Due to the divergence form of R, $\psi_{01}^T D\widetilde{W}$ will cancel out, leaving $\psi_{01}^T R = (\psi_{01}^T B)/\Delta t$. Hence, the constraint $\psi_{01}^T R = 0$ is a discrete form of (3). $\psi_{02}^T R = 0$ will also hold if (5) holds for some velocity field with the given boundary condition, i.e. (13) holds for some W, for example, the initial velocity field with time independent boundary condition.

When $dx_i \equiv \Delta x$ and $dy_i \equiv \Delta y$ are constant, the matrix L is symmetric and has a simpler form

$$L = \left[\frac{1}{\Delta x^2} C_n \otimes A'_m + \frac{1}{\Delta y^2} A'_n \otimes C_m\right]$$

where

$$A'_l = \begin{bmatrix} -1 & 1 & & & & \\ 1 & -2 & 1 & & & \\ & \ddots & & \ddots & & \\ & & & 1 & -2 & 1 \\ & & & & 1 & -1 \end{bmatrix}_{l\times l}.$$

In particular, when $\Delta x_i = \Delta y_i = h$, the discretization is a well known skewed five point stencil (see Fig. 4).

3. Fast solver. As we mentioned in the introduction, since the matrix equation (9) must be solved every time step, an efficient solver is essential. Fast solvers for the sum of two matrix tensor products were discussed in the first author's paper[2]. Later, L. Kaufman and D. Warner [13] provided a generalization to the higher order discretization case. In particular, the eigen-decomposition applused in [2] is replaced by a generalized eigenvalue and eigenvector problem. If we examine the formula of the matrix L, it is clear that generalized eigen-decomposition is needed to develop our fast solver. For a matrix of the form

$$A \otimes B + C \otimes E,$$

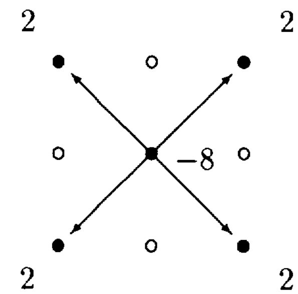

FIG. 4. *Skewed five point stencil*

the key step in developing a fast algorithm in [2, 13] was to simultaneously diagonalize the matrices B and E by a matrix Z for which

$$Z^T B Z = I, \qquad Z^T E Z = D,$$

where B and E are symmetric and positive definite. There are two obvious problems which cause these algorithms not to be directly applicable here. The first problem is that the matrices L, $D_x A_m$ and $D_y B_n$ are not symmetric. The second problem is the singularity of the matrices A_m, C_l and B_n. Many of the effective generalized eigenvalue algorithms [4, 8, 12, 21] for diagonalizing two banded matrices simultaneously need the matrix pair to be nonsingular and symmetric. Unfortunately, both A_m and C_l are singular and $D_x A_m$ is unsymmetric and thus a new approach is needed.

3.1. Equal-spaced in one direction. Assume the mesh is equal-spaced in one direction, say the x direction.

Let $\Delta x_i = \Delta x$, $i = 1, \cdots, m$. Then the matrix L can be written as

$$L = \left[\frac{1}{\Delta x^2} C_n \otimes A'_m + D_y B_n \otimes C_m \right]. \tag{15}$$

Though the matrix L is not symmetric, both $-A'_m$ and C_m are symmetric, semi-definite matrices. We can show the following result.

THEOREM 3.1. *There exists a matrix Z such that*

$$\begin{aligned} Z^T C^2 Z &= -A'_m \\ Z^T S^2 Z &= C_m \end{aligned}$$

where the diagonal elements c_i, s_i of C and S satisfy

$$c_i^2 + s_i^2 = 1, \qquad 0 \le c_i, s_i \le 1 \qquad i = 1, \cdots m.$$

Proof. The proof is a natural generalization of the result presented in [21], where a pair of symmetric and definite matrices is considered.

There exist two orthogonal matrices T_1 and T_2 such that

$$\begin{aligned} -A'_m &= T_1 D_1^2 T_1^T, \\ C_m &= T_2 D_2^2 T_2^T, \end{aligned}$$

where D_i, $i = 1, 2$ are diagonal matrices with non-negative diagonal elements. Now, construct

$$W = \begin{bmatrix} D_1 T_1^T \\ D_2 T_2^T \end{bmatrix} = QR$$

where Q, R make up the QR decomposition of the matrix W. Then partition matrix Q as follows:

$$Q = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} \tag{16}$$

Applying Stewart's theorem[19], we have from the Singular Value Decomposition(SVD)[7]

$$\begin{aligned} Q_1 &= U_1 C V_1 \\ Q_2 &= U_2 S V_2 \end{aligned}$$

where U_1, U_2, V_1 and V_2 are all orthogonal matrices and C, S are diagonal matrices $\mathrm{diag}(c_i)$ and $\mathrm{diag}(s_i)$, respectively. Then from (16) and the orthogonality of Q,

$$V_1 = V_2 = V$$

and

$$c_i^2 + s_i^2 = 1.$$

Therefore, we have

$$\begin{aligned} -A'_m &= T_1 D_1^2 T_1^T = R^T V^T C^2 V R = Z^T C^2 Z, \\ C_m &= T_2 D_2^2 T_2^T = R^T V^T S^2 V R = Z^T S^2 Z, \end{aligned}$$

where

$$Z = VR.$$

□

After the simultaneous diagonalization of the matrices A_m and C_m is achieved, the fast solver in this case is similar to the Buzbee-Golub-Nilson or Kaufman-Warner algorithm. First, the matrix equation (15) can be transformed into a tridiagonal matrix equation as follows:

$$\begin{aligned} (I \otimes Z^{-T})\, L\, (I \otimes Z^{-1}) &= I \otimes Z^{-T} \left[\frac{1}{\Delta x^2} C_n \otimes A'_m + D_y B_n \otimes C_m\right] I \otimes Z^{-1} \\ &= \frac{1}{\Delta x^2} C_n \otimes Z^{-T} A'_m Z^{-1} + D_y B_n \otimes Z^{-T} C_m Z^{-1} \\ &= \frac{-1}{\Delta x^2} C_n \otimes C^2 + D_y B_n \otimes S^2 \\ &= L'', \end{aligned}$$

where L'' is a tridiagonal matrix. A simple permutation matrix transforms L'' into a block diagonal matrix. The diagonal elements are $n \times n$ tridiagonal matrices. The solution of these tridiagonal matrices is trivial. In particular, they can be implemented on a vector or parallel computer efficiently.

For a regular mesh, $\Delta x = \Delta y = h$, this block diagonal matrix has a very simple form. We show the following result.

THEOREM 3.2. *The rank of the matrix L for regular mesh is of $mn - 2$.*

Proof. We know that both A'_m and C_m have only one zero eigenvalue and $N(A'_m) \cap N(C_m) = \emptyset$. Therefore, the two zero diagonal elements c_{i^*} and s_{j^*} in $\mathcal{C}$ and $\mathcal{S}$ will not be in the same location. It is easy to arrange them such that $c_1 = 0$ and $s_m = 0$. After permuting the matrix L'' to a block diagonal matrix, the first and last diagonal block matrix will be:

$$T_1 = s_1^2 A'_n, \qquad T_n = c_n^2 C_n,$$

respectively. Each has only one zero eigenvalue. The rest of the diagonal block matrices are of the form

$$T_i = -c_i^2 C_n + s_i^2 A'_n, \qquad i = 2, \cdots n-1, \tag{17}$$

where $c_i^2 + s_i^2 = 1$. When $c_i = s_i$, (17) reduces to a diagonal matrix

$$diag\{1, 2, \cdots, 2, 1\},$$

and it is not singular. If $c_i \neq s_i$, T_i is a tridiagonal matrix, viz

$$T_i = (-c_i^2 + s_i^2) \begin{bmatrix} -\alpha & 1 & & & \\ 1 & -2\alpha & 1 & & \\ & \ddots & & \ddots & \\ & & 1 & -2\alpha & 1 \\ & & & 1 & -\alpha \end{bmatrix}_{l \times l}$$

where $|\alpha| = \left| \frac{1}{s_i^2 - c_i^2} \right| > 1$, and hence T_i is non-singular. $\square$

We conjecture that this is true in general. Numerical computations support our conjecture.

As we discussed in the previous section, the solutions of the matrix equation (14) can differ by a linear combination of the two base vectors in $N(L)$. Some special attention needs to be paid to the solution of linear systems involving the first and last tridiagonal submatrices T_1 and T_m.

3.2. General case. For a general non-uniform grid, the simultaneous diagonalization of both matrix $D_x A_m$ and C_m by a single matrix can not be achieved. However, the generalized eigenvectors[1] can still be used to achieve the same purpose with two

[1] It is not difficult to see that the generalized eigenvalue problem:

$$D_x A_m x = \lambda C_m x$$

different matrices. Recall the matrix equation:

$$L\Phi = R \tag{19}$$

where

$$L = DG = [C_n \otimes D_x A_m + D_y B_n \otimes C_m]\,. \tag{20}$$

The QZ algorithm [7, 14] is used to compute the generalized eigenvalues (α_i, β_i) and its corresponding eigenvectors v_i such that

$$\beta_i D_x A_m v_i = \alpha_i C_m v_i.$$

Let $\mathcal{D}_a$ and $\mathcal{D}_c$ be two diagonal matrices whose diagonal elements are α_i and β_i, respectively. We have

$$D_x A_m V \mathcal{D}_c = C_m V \mathcal{D}_a$$

where matrix $V = [v_1, v_1, \cdots, v_m]$. Since we know that each matrix A_m and C_m has one zero eigenvalue, it is clear that each of $\mathcal{D}_a$ and $\mathcal{D}_c$ has one zero diagonal element. Due to the orthogonality of the null spaces of $N(A_m)$ and $N(C_m)$, the two zero diagonal elements will not be in the same location. We arrange the generalized eigenvectors such that

$$\alpha_n = 0; \qquad \beta_1 = 0.$$

Then we construct a matrix

$$U = [D_x A_m [v_1, v_2, \cdots .v_{m-1}], C_m v_m]. \tag{21}$$

It is not difficult to see that

$$U^{-1}(D_x A_m)V = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & & 0 \end{bmatrix} = \tilde{I}_m$$

and

$$U^{-1} C V = \begin{bmatrix} 0 & & & & \\ & \beta_2/\alpha_2 & & & \\ & & \ddots & & \\ & & & \beta_{n-1}/\alpha_{n-1} & \\ & & & & 1 \end{bmatrix} = \mathcal{D}_x.$$

---

is a discrete approximation of the eigenvalue problem

$$-y'' = \lambda y, \qquad y'(0) = y'(1) = 0 \tag{18}$$

on a non-uniform grid. Therefore, this pencil is diagonalizable. Our extensive numerical testings with randomly non-uniform grid have verified this observation. The eigenvectors are approximations of the eigenfunctions of (18).

Thus, the tridiagonalization of (20) can be achieved as follows:

$$\begin{aligned}
(I_n \otimes U^{-1})L(I_n \otimes V) &= (I_n \otimes U^{-1})\left[C_n \otimes D_x A_m + D_y B_n \otimes C_m\right](I_n \otimes V) \\
&= C_n \otimes U^{-1} D_x A_m V + D_y B_n \otimes U^{-1} C V \\
&= C_n \otimes \tilde{I}_m + D_y B_n \otimes \mathcal{D}_x \\
&= \tilde{L}.
\end{aligned}$$

Here, the matrix $\tilde{L}$ has three non-zero diagonals with a bandwidth of m. The resulting matrix equation will be

$$\begin{aligned}
(I_n \otimes U^{-1})L(I_n \otimes V)(I_n \otimes V^{-1})\Phi &= (I_n \otimes U^{-1})R \\
\tilde{L}\tilde{\Phi} &= \tilde{R},
\end{aligned}$$

where

$$\begin{aligned}
\tilde{\Phi} &= (I_n \otimes V^{-1})\Phi \\
\tilde{R} &= (I_n \otimes U^{-1})R.
\end{aligned}$$

We start with an ordering of the grid nodes first in the x direction, then in the y direction. Let P be the permutation matrix, which reorders the grid nodes first in the y direction, then in the x direction. We have

$$\begin{aligned}
(P\tilde{L}P)(P\tilde{\Phi}) &= P\tilde{R} \\
\hat{L}\hat{\Phi} &= \hat{R},
\end{aligned}$$

where $\hat{L} = P\tilde{L}P$, $\hat{\Phi} = P\tilde{\Phi}$ and $\hat{R} = P\tilde{R}$. The matrix $\hat{L}$ is a block diagonal matrix in which each diagonal element T_i is an $n \times n$ tridiagonal matrix:

$$\begin{aligned}
T_1 &= C_n \\
T_i &= C_n + \frac{\beta_i}{\alpha_i} D_y B_n \qquad (i = 2, 3, \cdots, m-1) \\
T_m &= D_y B_n
\end{aligned}$$

and

$$\hat{L} = \begin{bmatrix} T_1 & & & & \\ & T_2 & & & \\ & & \ddots & & \\ & & & T_{m-1} & \\ & & & & T_m \end{bmatrix}.$$

The solutions of the T_i's are trivial, but, some attention should be paid to the first one and last one, since both T_1 and T_m are rank $n-1$ and the top left $(n-1) \times (n-1)$ submatrices of T_1 and T_m are not singular. We can solve the two non-singular submatrices and assume the last element of the solution is zero. Using this approach, the contribution of the two null space base vectors of L is set to be zero and this will not affect the final solution of the INSE. The final solution of (19) is

$$\Phi = (I_n \otimes V)P\hat{\Phi}.$$

The fast solver can be summarized as follows:

1. Preprocess.
   (a) Compute the generalized eigen-decomposition $(\mathcal{D}_a, \mathcal{D}_c, V)$ of the pencil $(D_x A_m, C_m)$.
   (b) Compute the LU decomposition of the matrix U given in (21).
   (c) Form the LU decompositions of the matrices T_i.
   Note that this step is needed once at the beginning of the solution process of the INSE. The cost of it can be amortized over many time steps. The complexity of step (a) and (b) is $O(m^3)$, and of (c) is $O(nm)$.
2. Solve
$$(I \otimes U)\tilde{R} = R.$$
   The complexity is $O(nm^2)$.
3. Solve
$$\hat{L}\hat{\Phi} = \hat{R} = P\tilde{R}.$$
   The complexity of this step is $O(nm)$.
4. Compute
$$\Phi = (I \otimes V)P\hat{\Phi}$$
   The complexity of this step is $O(nm^2)$.

4. Three dimensional case. The fast solver can be extended to three dimensional problems without difficulty. Assume an $m \times n \times l$ non-uniform grid for the solution domain. The coefficient matrix of the three dimensional case in (19) will be

$$L = C_l \otimes C_n \otimes D_x A_m + C_l \otimes D_y B_n \otimes C_m + D_z E_l \otimes C_n \otimes C_m.$$

where

$$D_z = \begin{bmatrix} \frac{1}{dz_1} & & & \\ & \frac{1}{dz_2} & & \\ & & \ddots & \\ & & & \frac{1}{dz_l} \end{bmatrix}_{l \times l}$$

and

$$E_l = \begin{bmatrix} -\frac{1}{dz_{12}} & \frac{1}{dz_{12}} & & & & \\ \frac{1}{dz_{12}} & -\left(\frac{1}{dz_{12}} + \frac{1}{dz_{23}}\right) & \frac{1}{dz_{23}} & & & \\ & \ddots & \ddots & & & \\ & & \ddots & \ddots & & \\ & & & \frac{1}{dz_{l-2,l-1}} & -\left(\frac{1}{dz_{l-2,l-1}} + \frac{1}{dz_{l-1,l}}\right) & \frac{1}{dz_{l-1,l}} \\ & & & & \frac{1}{dz_{l-1,l}} & -\frac{1}{dz_{l-1,l}} \end{bmatrix}_{l\times l}.$$

There are two generalized eigenvalue problems

$$\begin{aligned} D_x A_m x &= \lambda C_m x, \\ D_y B_n y &= \tau C_n y \end{aligned}$$

which need to be solved. Let V and W be the eigenvector matrices for the two problems, respectively. Similarly, form

$$\begin{aligned} U &= [D_x A_m[v_1, v_2, \cdots . v_{m-1}], C_m v_m], \\ Q &= [D_y B_n[w_1, w_2, \cdots . w_{n-1}], C_n w_n] \end{aligned}$$

and we have

$$\begin{aligned} U^{-1} D_x A_m V = \tilde{I}_m, \qquad & U^{-1} C_m V = \mathcal{D}_x, \\ Q^{-1} D_y B_n W = \tilde{I}_n, \qquad & Q^{-1} C_n W = \mathcal{D}_y. \end{aligned}$$

The tridiagonalization of L can be achieved as follows

$$\begin{aligned} (I_l \otimes Q^{-1} \otimes I_m)(I_l \otimes I_n \otimes U^{-1})\ L\ (I_l \otimes W \otimes I_m)(I_l \otimes I_n \otimes V) = \\ C_l \otimes \mathcal{D}_y \otimes \tilde{I}_m + C_l \otimes \tilde{I}_n \otimes \mathcal{D}_x + D_z E_l \otimes \mathcal{D}_y \otimes \mathcal{D}_x \end{aligned}$$

5. Numerical Experiment. Consider INSE with exact solution

$$\begin{aligned} u &= e^t \sin x \cos y \\ v &= -e^t \cos x \sin y \\ p &= e^t \sin x \sin y \end{aligned} \tag{22}$$

with corresponding nonhomogeneous terms in the momentum equations, on a square $0 \le x \le \pi$ and $0 \le y \le \pi$. The initial and boundary values of u and v are taken from (22). The conservative version of the centered-difference Crank-Nicolson scheme was used for (6). For the discrete Poisson equation, the first constraint $\phi_{01}^T R = 0$ is obviously satisfied, allowing for discretization error. The second constraint $\phi_{02}^T R = 0$ reduces to

$$\sum_{red} \nabla \cdot \mathbf{w} - \sum_{black} \nabla \cdot \mathbf{w} = 0 \tag{23}$$

corresponding to the checkerboard effect, and hence is also satisfied (except for the discretization error).

On a 64×64 nonuniform mesh with $\min_i(dx_i) = \min_i(dy_i) \approx 3.3 \times 10^{-2}$ and $\max_i(dx_i) = \max_i(dy_i) \approx 7.6 \times 10^{-2}$, the fast solver developed herein took 0.13 *sec* CPU time on the 100MHz Indigo R4000 workstation for each solution of (14), with residual $\approx 0.4 \cdot 10^{-3}$ and $\max_{ij}(\nabla \cdot \mathbf{w}) \approx 0.1 \cdot 10^{-3}$. There was no apparent oscillation in pressure, but adding ± 1, say, to the initial pressure at alternate cells (the red and black cells) will cause an oscillating pressure distribution as shown in Fig 6 for $t = 1$. As we indicated earlier, the oscillation in p has no effect on the solution of u and v shown in Fig. 5, nor any effect on the solution process.

In contrast, solving the same problem using an adaptive multigrid method [10] took six times more CPU time. This version of the multigrid is not very effective in this application, presumably due mainly to the treatment for the second constraint.

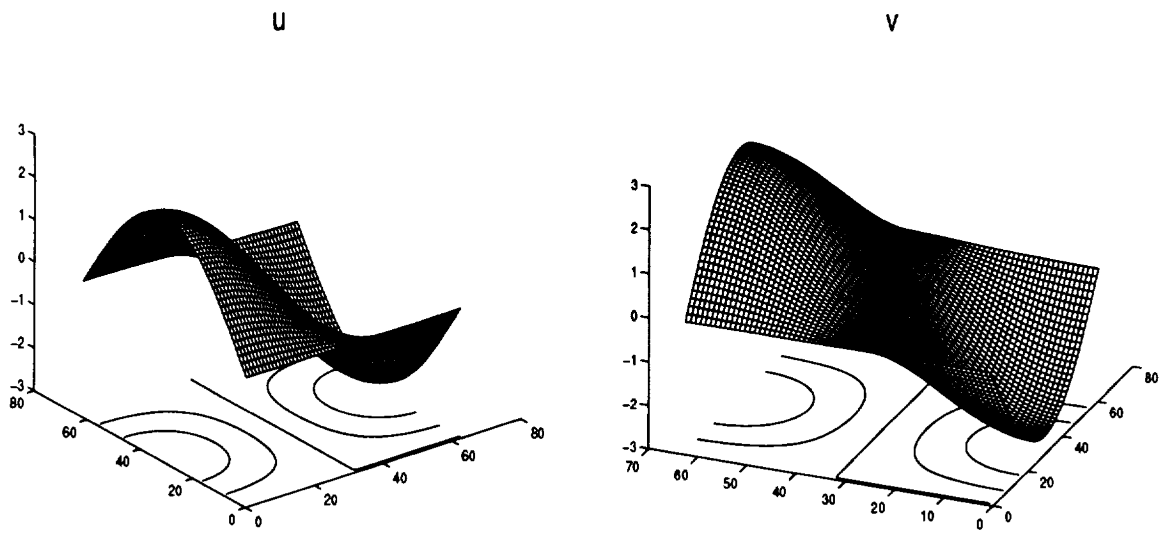

FIG. 5. *Solution u and v at $t = 1$*

6. Conclusion. We have developed a fast Poisson solver for the unsteady incompressible Navier-Stokes Equation with finite difference methods on the non-uniform, half-staggered grid. Due to the efficiency of this method, it is possible to solve the unsteady flow with $Re = 10,000$ [9]. Although this method can only be applied to an orthogonal rectangular grid, it can be used as a preconditioner or in a domain decomposition scheme for general applications. Our next project will be development of effective solvers for unsteady INSE on an irregular domain with curvilinear grid.

REFERENCES

[1] J. B. BELL, P. COLELLA, AND H. M. GLAZ, *A second-order projection method for the incompressible Navier-Stokes equations*, Journal of Computational Physics, 85 (1989), pp. 257–283.

[2] B. BUZBEE, G. GOLUB, AND C. NILSON, *On direct methods for solving Poisson's equation*, SIAM Journal on Numerical Analysis, 7 (1970), pp. 627–656.

[3] A. J. CHORIN, *Numerical solution of Navier-Stokes equations*, Mathematics of Computation, 22 (1968), pp. 745–762.

[4] C. R. CRAWFORD, *Reduction of aa band-symmetric generalized eigenvalue problem*, Comm. ACM., 16 (1973), pp. 41–44.

[5] M. FORTIN, R. PEYRET, AND R. TEMAM, *Calcul des ecoulements d'un fluide visqueux incompressible*, J. Mec, 10 (1971), pp. 357–390.

Oscillating P

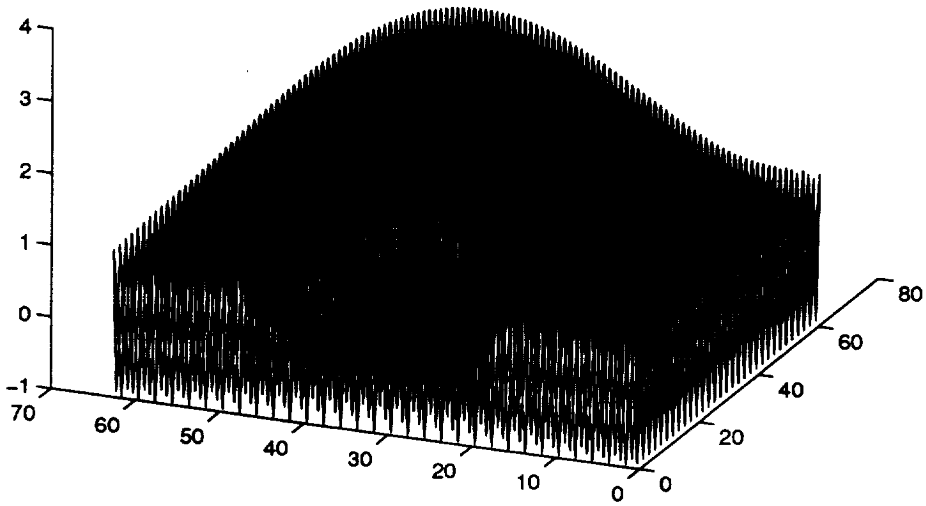

FIG. 6. *Oscillating pressure at* $t = 1$

[6] G. GOLUB AND C. V. LOAN, *Matrix Computations, second edition*, The Johns Hopkins University Press, Baltimore, MD, 1989.

[7] G. GOLUB AND V. E. LANGLOIS, *Direct solution of the equation for the Stokes stream function*, Computer Methods in Applied Mechanics and Engineering, 19 (1979), pp. 391–399.

[8] R. G. GRIMES, J. G. LEWIS, AND H. D. SIMON, *A shifted block Lanczos algorithm for solving sparse symmetric generalized eigenproblems*, SIAM Journal on Matrix Analysis and Applications, 15 (1994), pp. 228–272.

[9] L. C. HUANG, J. OLIGER, W.-P. TANG AND Y. D. WU, *Toward efficient and robust finite difference schemes for unsteady incompressible Navier-Stokes equation on the half staggered mesh*, Sixth International Symposium on CFD, (1995).

[10] L. C. HUANG, *On boundary treatment for the numerical solution of the incompressible Navier-Stokes equations with finite difference methods*, J. Comput. Math, (1994).

[11] ——, *On the staggered mesh II for the finite difference solution of the unsteady incompressible Navier-Stokes equations*, J. Comput. Math, (1994). Submitted.

[12] L. KAUFMAN, *An algorithm for the banded symmetric generalized matrix eigenvalue problem*, SIAM Journal on Matrix Analysis and Applications, 14 (1993), pp. 372–389.

[13] L. KAUFMAN AND D. D. WARNER, *High-order, fast-direct methods for separable elliptic equations*, SIAM Journal on Numerical Analysis, 21 (1984), pp. 672–694.

[14] C. B. MOLER AND G. W. STEWART, *An algorithm for generalized matrix eigenvalue problems*, SIAM Journal on Numerical Analysis, 10 (1973), pp. 241–256.

[15] R. L. SANI, *The cause and cure of the spurious pressure generated by certain FEM solution of the incompressible Navier-Stokes equations:Part I*, Int. J. for Num. Meth. in Fluids, 1 (1981), pp. 17–43.

[16] U. SCHMANN AND R. SWEET, *A direct method for the solution of Poisson's equation with Neumann boundary conditions on a staggered grid of arbitrary sizes*, Journal of Computational Physics, 20 (1976), pp. 171–182.

[17] SHYY AND VU, *On the adoption of velocity variable and grid system for fluid flow computation*

in curvilinear coordinates, Journal of Computational Physics, 92 (1991), pp. 82–105.

[18] A. B. STEPHEN, *A finite difference Galerkin formulation for the incompressible Navier-Stokes equations*, Journal of Computational Physics, 53 (1984), pp. 152–172.

[19] G. W. STEWART, *On the perturbation of pseudo-inverses, projections, and linear least squares problems*, SIAM Review, 19 (1977), pp. 634–662.

[20] R. TEMAM, *Navier-Stokes Equations, Theory and Numerical Analysis*, Elsevier Science Pub. B. V., New York, 1984.

[21] S. WANG AND S. ZHAO, *An algorithm for $Ax = \lambda Bx$ with symmetric and positive definite a and b*, SIAM Journal on Matrix Analysis and Applications, 12 (1991), pp. 654–660.

RIACS
Mail Stop T041-5
NASA Ames Research Center
Moffett Field, CA 94035