JOURNAL OF COMPUTATIONAL PHYSICS 99, 190–195 (1992)

Numerical Solution of the Steady Stokes Equations*

ERIC YU TAU†

Lawrence Berkeley Laboratory, Department of Mathematics, University of California, Berkeley, California 94720

Received January 31, 1990; revised April 8, 1991

In this paper we present a fast numerical technique for finding solutions of the steady-state Stokes equations on both two- and three-dimensional domains. We implement the method on a special staggered grid for a rectangular (cubic) domain and obtain a solution in an order of $O(N \log N)$ operations for both two- and three-dimensional cases, where N is the number of grid points in the domain. The main idea is to derive from the Stokes equations an equation for the pressure p, $Ap = b$, where the matrix A is semi-positive definite and very-well conditioned on the orthogonal complement of its null space.

1. INTRODUCTION

For simplicity, we describe the problem in the two dimensional case. The Stokes equations have the form

$$-\mu \Delta \mathbf{u} + \nabla p = \mathbf{f}, \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{2}$$

where $\mathbf{u} = (u, v)$ is the velocity field, p is the pressure, $\mathbf{f} = (f_x, f_y)$ is the body force, and μ is the viscosity of the fluid. Let Ω be a rectangular domain. We assume the following boundary condition on $\mathbf{u}$,

$$\mathbf{u}(x, y) = \mathbf{u}_b(x, y), \qquad (x, y) \in \partial\Omega. \tag{3}$$

$\mathbf{u}_b$ must satisfy

$$\int_{\partial\Omega} \mathbf{u}_b \cdot \mathbf{n}\, dS = 0, \tag{4}$$

where $\mathbf{n}$ is the outer unit normal to $\partial\Omega$. It is well known that we need not prescribe the boundary condition on p.

Most of the work done before on this problem begins with replacing Eqs. (1) and (2) by the system

$$-\mu \Delta \mathbf{u} + \nabla p = \mathbf{f}$$

$$\Delta p = \nabla \cdot \mathbf{f}$$

and then trying to solve the Laplace equation for p. The lack of boundary condition on p causes some difficulty. Much work has been done to determine the correct boundary condition for p (see, e.g., discussion in Strikwerda [7]). Work using the original system (1) and (2) has also been done, but it still requires some additional conditions for p near the boundary [4, 7].

Chorin [2] proposed a projection method for the more general Navier–Stokes equations that determines p without any artificial conditions on p. The method is based on the use of the equation $\nabla \cdot \mathbf{u} = 0$ instead of $\Delta p = \nabla \cdot \mathbf{f}$ for finding p. Here we present a method based on the evaluation of p for the Stokes equations that requires no artificial conditions on p either. The idea is to use consistent finite difference operators to discretize Eqs. (1) and (2). It will be shown later in this section that the resulting system of equations is semi-definite if the difference operators satisfy a discrete analogue of the condition $(\mathbf{u}, \nabla p) = -(\nabla \cdot \mathbf{u}, p)$ (see [3] for the discrete form of this condition). We eliminate $\mathbf{u}$ to form an equation $Ap = b$ which uniquely determines p (up to a constant, of course). Since it is computationally inefficient to express $\mathbf{u}$ in terms of p explicitly, we shall only have to evaluate Ap, i.e., given a p, we can find Ap, where A is not written out (as in the iterative implementation of Chorin's projection method [2]). It turns out that for some properly chosen operators the matrix A is semi-positive definite and very-well conditioned on the orthogonal complement of its null space, i.e., the condition number is small and bounded independently of the mesh size, so we can use the conjugate-gradient method to get the solution for p. Similar ideas were used by Maday and Patera [5] with a spectral element approach.

Assume the domain Ω is covered by some uniform grid of

* This work is supported in part by the Applied Mathematical Sciences Subprogram of the Office of Energy Research, U.S. Department of Energy, under Contract Number DE-AC03-76SF000098.

† Current address: Department of Mathematics, Duke University, Durham, North Carolina 27706.

0021-9991/92 $3.00

mesh size h. Following the notations of Anderson [1], we approximate Eqs. (1) and (2) by finite difference equations,

$$-\mu \Delta_h \mathbf{u} + (\nabla_x^h, \nabla_y^h) p = \mathbf{f}, \tag{5}$$

$$D_x^h u + D_y^h v = 0, \tag{6}$$

where Δ_h is an approximation to the Laplacian, ∇_x^h and ∇_y^h to the derivatives in the gradient operator, D_x^h and D_y^h to the derivatives in the divergence operator. Notice that near the boundary $\Delta_h \mathbf{u}$ and $D_x^h u + D_y^h v$ contain velocity values on the boundary which are known. If we move those known terms to the right-hand side of equations, we can rewrite Eqs. (5) and (6) as

$$-\mu \tilde{\Delta}_h \mathbf{u} + (\nabla_x^h, \nabla_y^h) p = \mathbf{f} + \mathbf{g}(\mathbf{u}_b) \tag{7}$$

$$\tilde{D}_x^h u + \tilde{D}_y^h v = h(\mathbf{u}_b), \tag{8}$$

where

$$\mathbf{g}(\mathbf{u}_b) = \mu \Delta_h \mathbf{u} - \mu \tilde{\Delta}_h \mathbf{u}$$

and

$$h(\mathbf{u}_b) = \tilde{D}_x^h u + \tilde{D}_y^h v - (D_x^h u + D_y^h v).$$

In other words, the operators with tilde's differ from the original operators only near the boundary, and they are the parts of the operators Δ_h, D_x^h, D_y^h that operate only on unknown values of the velocity. We will show how to calculate them once Δ_h, D_x^h, and D_y^h are defined in Section 2 (see also [1]).

Put Eqs. (7) and (8) in block matrix form. We have

$$\begin{pmatrix} -\mu \tilde{\Delta}_h & 0 & \nabla_x^h \\ 0 & -\mu \tilde{\Delta}_h & \nabla_y^h \\ \tilde{D}_x^h & \tilde{D}_y^h & 0 \end{pmatrix} \begin{pmatrix} u \\ v \\ p \end{pmatrix} = \begin{pmatrix} f_x + g_x(\mathbf{u}_b) \\ f_y + g_y(\mathbf{u}_b) \\ h(\mathbf{u}_b) \end{pmatrix}. \tag{9}$$

Consider the quadratic form

$$(u, v, p) \begin{pmatrix} -\mu \tilde{\Delta}_h & 0 & \nabla_x^h \\ 0 & -\mu \tilde{\Delta}_h & \nabla_y^h \\ \tilde{D}_x^h & \tilde{D}_y^h & 0 \end{pmatrix} \begin{pmatrix} u \\ v \\ p \end{pmatrix}$$

$$= -\mu[(u, \tilde{\Delta}_h u) + (v, \tilde{\Delta}_h v)] + (u, \nabla_x^h p) + (v, \nabla_y^h p) + (\tilde{D}_x^h u, p) + (\tilde{D}_y^h v, p).$$

It is positive semi-definite, provided

$$(u, \nabla_x^h p) + (v, \nabla_y^h p) = -((\tilde{D}_x^h u, p) + (\tilde{D}_y^h v, p)) \tag{10}$$

and

$$(u, \tilde{\Delta}_h u) \leqslant 0, \qquad (v, \tilde{\Delta}_h v) \leqslant 0. \tag{11}$$

Conditions (11) are satisfied for most reasonable difference approximations to the Laplacian while condition (10) is just a discrete analogue of the condition $(\mathbf{u}, \nabla p) = -(\nabla \cdot \mathbf{u}, p)$.

We can formally eliminate u and v in Eq. (9) to form a equation for p as follows:

From Eq. (7) we obtain

$$u = \frac{1}{\mu} \{\tilde{\Delta}_h^{-1} \nabla_x^h p - \tilde{\Delta}_h^{-1}(f_x + g_x(\mathbf{u}_b))\}$$

and

$$v = \frac{1}{\mu} \{\tilde{\Delta}_h^{-1} \nabla_y^h p - \tilde{\Delta}_h^{-1}(f_y + g_y(\mathbf{u}_b))\}.$$

Substitution of these expressions into Eq. (8) gives

$$(\tilde{D}_x^h \tilde{\Delta}_h^{-1} \nabla_x^h + \tilde{D}_y^h \tilde{\Delta}_h^{-1} \nabla_y^h) p = \tilde{D}_x^h \tilde{\Delta}_h^{-1}(f_x + g_x(\mathbf{u}_b)) + \tilde{D}_y^h \tilde{\Delta}_h^{-1}(f_y + g_y(\mathbf{u}_b)) + \mu h(\mathbf{u}_b)$$

or simply

$$Ap = F^1 + F^2, \tag{12}$$

where

$$\begin{aligned} A &= \tilde{D}_x^h \tilde{\Delta}_h^{-1} \nabla_x^h + \tilde{D}_y^h \tilde{\Delta}_h^{-1} \nabla_y^h \\ F^1 &= \tilde{D}_x^h \tilde{\Delta}_h^{-1}(f_x + g_x(\mathbf{u}_b)) + \tilde{D}_y^h \tilde{\Delta}_h^{-1}(f_y + g_y(\mathbf{u}_b)) \\ F^2 &= \mu h(\mathbf{u}_b). \end{aligned} \tag{12a}$$

If we can solve Eq. (12) for p, then we can use Eq. (7) to obtain solutions for u and v.

Of course we do not have to find $\tilde{\Delta}_h^{-1}$ explicitly, but rather given a p (and $\mathbf{u}_b$), we can solve Eq. (7) for u and v and then use Eq. (8) to form Ap. It turns out that for some properly chosen operators Δ_h, D_x^h, D_y^h, ∇_x^h, and ∇_y^h the matrix A is semi-positive definite, so we can use the conjugate-gradient method to solve Eq. (12).

Recall that $\tilde{\Delta}_h$, $\tilde{D}_x^h$, $\tilde{D}_y^h$ are different from Δ_h, D_x^h, D_y^h only near the boundary, so if we choose

$$\Delta_h = D_x^h \nabla_x^h + D_y^h \nabla_y^h \tag{13}$$

which is just a discrete analogue of the definition of Laplacian, we will have

$$\tilde{\Delta}_h = \tilde{D}_x^h \nabla_x^h + \tilde{D}_y^h \nabla_y^h$$

on all grid points except those near the boundary. By looking at the structure of matrix A in (12a), it is reasonable to believe that A is close to the identity matrix and that the conjugate-gradient method will converge in a few steps. We shall see that this is indeed the case by numerical examples.

2. DISCRETIZATION OF THE DOMAIN AND CHOICE OF D^h, ∇^h, Δ_h

We now implement the method for a special case of a staggered grid. Assume Ω is a unit square. Let $h=1/n$. Define meshes (see Fig. 1)

$$U=\{(ih,(j+\tfrac{1}{2})h)\,|\,i=0,\ldots,n;j=0,\ldots,n-1\}$$
$$V=\{((i+\tfrac{1}{2})h,jh)\,|\,i=0,\ldots,n-1;j=0,\ldots,n\}$$
$$P=\{((i+\tfrac{1}{2})h,(j+\tfrac{1}{2})h)\,|\,i,j=0,\ldots,n-1\}$$
$$U^0=U-U\cap\partial\Omega,\qquad V^0=V-V\cap\partial\Omega$$

and define u on U, v on V, and p on P.

The operators are defined as

$D_x^h: U\to P$

by $(D_x^h u)_{i+1/2,j+1/2}=(u_{i+1,j+1/2}-u_{i,j+1/2})/h$

$D_y^h: V\to P$

by $(D_y^h v)_{i+1/2,j+1/2}=(v_{i+1/2,j+1}-v_{i+1/2,j})/h$

$\nabla_x^h: P\to U^0$

by $(\nabla_x^h p)_{i,j+1/2}=(p_{i+1/2,j+1/2}-p_{i-1/2,j+1/2})/h$

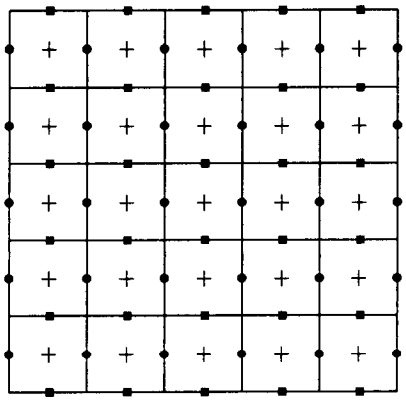
$\nabla_y^h: P\to V^0$

by $(\nabla_y^h p)_{i,j+1/2}=(p_{i+1/2,j+1/2}-p_{i+1/2,j-1/2})/h$

and we define $\Delta_h u$ on U_0, $\Delta_h v$ on V_0 by the standard five-point approximation to the Laplacian except near the boundary, where we use a first-order approximation; for instance, we put

$$\frac{\partial^2 u}{\partial y^2}\approx\frac{u_{i,3/2}-3u_{i,1/2}+2u_{1,0}}{h^2}$$

at $j=1$. Similarly for $j=n$ and $\partial^2 v/\partial x^2$. With those choices of the operators, it can be checked that relation (13) is satisfied on all grid points, except those near the boundary.



$\bullet\in U\qquad\blacksquare\in V\qquad +\in P$

FIG. 1. Discretization of domain.

Now the operators D^h etc. are defined. We can obtain the corresponding $\tilde{D}^h$ as follows: if $D_x^h u$ does not involve $\mathbf{u}_b$, put $\tilde{D}_x^h u=D_x^h u$; if it does, just put all $\mathbf{u}_b=0$ in $D_x^h u$ to obtain $\tilde{D}_x^h u$. For instance,

$$(\tilde{D}_x^h u)_{1/2,j+1/2}=\frac{u_{1,j+1/2}}{h}.$$

If we define the inner product on U^0, V^0, and P by

$$(\phi,\psi)_{U^0}=\sum_{i=1}^{n-1}\sum_{j=1}^{n}\phi_{i,j-1/2}\psi_{i,j-1/2}$$
$$(\phi,\psi)_{V^0}=\sum_{i=1}^{n}\sum_{j=1}^{n-1}\phi_{i-1/2,j}\psi_{i-1/2,j}$$

and

$$(\phi,\psi)_P=\sum_{i=1}^{n}\sum_{j=1}^{n}\phi_{i-1/2,j-1/2}\psi_{i-1/2,j-1/2},$$

then it is easy to check that

$$(\tilde{D}_x^h u+\tilde{D}_y^h v,p)_P=-(\nabla_x^h p,u)_{U^0}-(\nabla_y^h p,v)_{V^0}.$$

Moreover, since

$$-\tilde{\Delta}_h=\begin{pmatrix}5&-1&&&&-1&0&-1&&&\\-1&4&-1&&&&-1&0&-1&&\\&\ddots&\ddots&&&&&\ddots&\ddots&&\\&\ddots&\ddots&&&&&\ddots&\ddots&&\\&\ddots&\ddots&&&&&\ddots&\ddots&&\\&\ddots&\ddots&&&&&\ddots&\ddots&&\\&\ddots&\ddots&&&&&\ddots&\ddots&&\\&\ddots&\ddots&&&&&\ddots&\ddots&&\\&&-1&0&-1&&&&-1&5\end{pmatrix},$$

we have, by Gerschgorin's theorem,

$$-(u,\tilde{\Delta}_h u)_{U^0}\geqslant 0$$

and

$$-(v,\tilde{\Delta}_h v)_{V^0}\geqslant 0.$$

Thus the matrix in Eq. (9) is indeed semi-positive definite and our numerical experiment shows that the matrix A is also semi-positive definite.

With the discretization and the choice of operators made here, Eq. (12) has a non-trivial null space which is a constant pressure field. For the equation to have a solution the right-hand side must be orthogonal to the null space. In this case, this condition turns out to be

$$\sum_{i,j} (F^1_{i,j} + F^2_{i,j}) = 0.$$

It can be verified that it is always true that

$$\sum_{i,j} F^1_{i,j} = 0.$$

The second part of the sum, $\sum_{i,j} F^2_{i,j}$, is just a midpoint rule approximation to the condition $\int_{\partial\Omega} \mathbf{u}_b \cdot \mathbf{n}\, dS = 0$ (see [1]).

The above description can be naturally extended to the three-dimensional case and all the conclusions remain true.

3. SOLUTION OF THE EQUATIONS

In this section, we specify the routines we used in our numerical examples. The goal is to solve Eq. (7) for $\mathbf{u}$ quickly with a given p. In the two-dimensional case there is a fast solver for this equation, namely, the subroutine BLKTRI from the NCAR set of routines FISHPACK. In the three-dimensional case, however, there is no fast solver available. We can apply the fast fourier transform (FFT) technique to reduce the problem to a two-dimensional problem (see, e.g., [8]). For instance, consider the equation for u,

$$\mu \tilde{\Delta}_h u = \nabla^h_x p + f_x + g_x(\mathbf{u}_b).$$

The grid is defined by $x = (i-1)h$, $y = (j-0.5)h$, $z = (k-0.5)h$. We use FFT in the x-direction to reduce the problem to $(n-1)$ two-dimensional systems on the staggered grid $y = (j-0.5)h$, $z = (k-0.5)h$. Since u is specified on the boundary, only sine functions are needed in the FFT. For the resulting two-dimensional systems, we use the subroutine HSTCRT from the NCAR set of routines FISHPACK.

To solve equation $Ap = b$, we use the standard conjugate-gradient method. Since there is a non-trivial null space, care must be taken to eliminate components in the null space. Let $\mathbf{c}$ denote the null vector of the matrix (which is a constant vector in this case) and $(\cdot, \cdot)$ the inner product on p defined in Section 2. The conjugate-gradient method algorithm is as follows:

$$r^0 = b - Ap^0, \qquad p^0 \text{ initial guess.}$$

For $k = 0, 1, 2, \ldots$,

$$r^k = r^k - \frac{(r^k, \mathbf{c})}{(\mathbf{c}, \mathbf{c})}\mathbf{c}$$

$$\beta^k = \frac{(r^k, r^k)}{(r^{k-1}, r^{k-1})}, \qquad k \geqslant 1,\ \beta^0 = 0$$

$$z^k = r^k + \beta^k z^{k-1}$$

$$\alpha^k = \frac{(r^k, r^k)}{(z^k, Az^k)}$$

$$p^{k+1} = p^k + \alpha^k z^k$$

$$r^{k+1} = r^k - \alpha^k A z^k.$$

The computational work in each iteration step of the conjugate-gradient method is done mostly to form Ap, everything else is just some vector products which require labor of order $O(N)$ (where N is the number of points in the domain). The formation of Ap requires solutions to two Laplace equations in the two-dimensional case or three Laplace equations in the three-dimensional case which cost $O(N \log N)$. Our solution to the steady Stokes equations thus requires work of order $2MN \log N$ or $3MN \log N$, where M is the number of iterations in the conjugate-gradient method. We will see from our numerical examples that M is very small even for very large N in the three-dimensional case.

We first give an example in the two-dimensional case. As a test problem, we take $\mathbf{f} = (3\cos(x)\sin(y), -\sin(x)\cos(y))$, $\mu = 1$, $u_b = \cos(x)\sin(y)$, $v_b = -\sin(x)\cos(y)$, and $\Omega = \{x, y \mid 0 \leqslant x \leqslant 2\pi,\ 0 \leqslant y \leqslant \pi\}$. In this case the exact solution is known to be

$$u = \cos(x)\sin(y), \qquad v = -\sin(x)\cos(y),$$

$$\frac{\partial p}{\partial x} = \cos(x)\sin(y), \qquad \frac{\partial p}{\partial y} = \sin(x)\cos(y).$$

In Table I we display the results of the calculation. We use the residual r of the pressure equation (12) as our computation criterion, i.e., the iteration stops when $\max(r_{i,j})$ is less

TABLE I

Numerical Results in 2D Case

Grid size	20×10	40×20	80×40	160×80
Iterations	8	10	10	11
L^∞ norm of r	9.50×10^{-8}	8.84×10^{-9}	7.71×10^{-8}	9.06×10^{-8}
$e(u)$	1.21×10^{-3}	2.92×10^{-4}	7.23×10^{-5}	1.80×10^{-5}
$e(v)$	5.06×10^{-4}	1.17×10^{-4}	2.87×10^{-5}	7.13×10^{-6}
$e(p_x)$	2.07×10^{-2}	5.68×10^{-3}	1.48×10^{-3}	3.77×10^{-4}
$e(p_y)$	1.49×10^{-2}	4.58×10^{-3}	1.26×10^{-3}	3.30×10^{-4}

than 1×10^{-7}. We should mention that by our construction this residual is exactly the divergence of the velocity field; i.e., we have

$$r_{i,j} = (D_x^h u + D_y^h v)_{i,j}.$$

We denote by $e(u)$, $e(v)$, $e(p_x)$, and $e(p_y)$ the errors of u, v, p_x, and p_y, i.e., the absolute value of the differences between the exact solutions and our numerical solutions, respectively. All the errors are measured in the L^∞ norm.

For the case of three dimensions, we choose the test problem to be

$$u_b = \sin(x)\cos(y)\cos(2z)$$
$$v_b = \cos(x)\sin(y)\cos(2z)$$
$$w_b = -\cos(x)\cos(y)\sin(2z)$$
$$\mathbf{f} = (0, 0, -18\cos(x)\cos(y)\sin(2z))$$
$$\mu = 1,$$
$$\Omega = \{x, y, z \mid 0 \leqslant x \leqslant \pi, 0 \leqslant y \leqslant \pi, 0 \leqslant z \leqslant \pi/2\}.$$

We also have the exact solution as:

$$u = \sin(x)\cos(y)\cos(2z)$$
$$v = \cos(x)\sin(y)\cos(2z)$$
$$w = -\cos(x)\cos(y)\sin(2z)$$
$$p = 6\cos(x)\cos(y)\cos(2z).$$

Table II shows our numerical results for this problem.

We can see from the numerical results that we achieved second-order accuracy for the velocity in both cases despite the fact that we used a first-order approximation for the Laplacian operator near the boundary. The calculation for the pressure is less accurate, which is expected. The number of iterations grows very slightly as the mesh size increases both in the two- and three-dimensional cases. There is no substantial increase of the number of iterations from two dimensions to three dimensions. This makes our method particularly suitable for three-dimensional calculations.

TABLE II

Numerical Results in 3D Case

Grid size	$20 \times 20 \times 10$	$40 \times 40 \times 20$	$80 \times 80 \times 40$	$160 \times 160 \times 80$
Iterations	13	16	18	19
L^∞ norm of r	9.82×10^{-9}	1.48×10^{-8}	5.18×10^{-8}	8.47×10^{-8}
$e(u)$	9.47×10^{-3}	2.66×10^{-3}	7.07×10^{-4}	1.83×10^{-4}
$e(v)$	9.47×10^{-3}	2.66×10^{-3}	7.07×10^{-4}	1.83×10^{-4}
$e(w)$	1.02×10^{-2}	2.64×10^{-3}	6.66×10^{-4}	1.67×10^{-4}
$e(p_x)$	0.105	5.54×10^{-2}	2.81×10^{-2}	1.41×10^{-2}
$e(p_y)$	0.105	5.54×10^{-2}	3.72×10^{-2}	1.41×10^{-2}
$e(p_z)$	0.202	0.100	6.71×10^{-2}	2.52×10^{-2}

TABLE III

Iterations Needed for a Different Discretization

Grid size	20×10	40×20	80×40	160×80
Iterations	31	45	48	49
L^∞ norm of r	7.98×10^{-8}	9.49×10^{-8}	9.12×10^{-8}	9.25×10^{-8}

One should note that for the conjugate-gradient method to converge in a few iterations, it is very important that relation (13) in Section 2 be satisfied; i.e., the gradient and divergence operators should combine to form the discrete Laplacian used in the discretization of $\Delta_h \mathbf{u}$. To show the importance of this, we make another calculation with a different discretization of the domain and different choices of the operators. We take the same test problem in two dimensions. This time the two components of the velocity are defined on the same set of grid points defined by x, $h = ih$, $i = 0, \ldots, n$. The pressure p is defined on the same grid as before. The operators are defined as

$$(D_x^h u)_{i+1/2,j+1/2} = \frac{1}{2h}[(u_{i+1,j+1} - u_{i,j+1}) + (u_{i+1,j} - u_{i,j})]$$

$$(D_y^h v)_{i+1/2,j+1/2} = \frac{1}{2h}[(v_{i+1,j+1} - v_{i+1,j}) + (v_{i,j+1} - v_{i,j})]$$

$$(\nabla_x^h p)_{i,j} = \frac{1}{2h}[(p_{i+1/2,j+1/2} - p_{i-1/2,j+1/2}) + (p_{i+1/2,j-1/2} - p_{i-1/2,j-1/2})]$$

$$(\nabla_y^h p)_{i,j} = \frac{1}{2h}[(p_{i+1/2,j+1/2} - p_{i+1/2,j-1/2}) + (p_{i-1/2,j+1/2} - p_{i-1/2,j-1/2})],$$

and Δ_h by the standard five-point approximation to the Laplacian. These definitions satisfy (10) (see [6]) and (11), but not (13). Table III shows the results of this calculation. We see that it requires about four times the iterations needed for the previous one.

4. CONCLUSIONS

We have presented a fast numerical technique for solving the steady-state Stokes equations which requires no artificial conditions for the pressure near the boundary. Our numerical experiments show that the method is of second-

order accuracy for the velocity. The key idea of the method is to use some consistent difference operators to approximate the Stokes equations and then derive an equation $Ap = b$ for the pressure such that the matrix A is semi-positive definite and very-well conditioned on the orthogonal complement of its null space. Therefore it is very efficient to use the standard conjugate-gradient method to solve the pressure equation. We implemented our method on a staggered grid for rectangular (cubic) domains and showed that the conjugate-gradient method converged in a very few iterations even for a very large number of grid points in the three-dimensional case. The reason we chose a staggered grid is because it is the simplest way to define the gradient and divergence operators so that they combine to form the usual five-point Laplacian, resulting in the easy solution for the Poisson equation for a given p. Other discretization will also work as long as relation (13) in Section 2 is satisfied and the resulting Poisson equation is easy to solve. The method can also be extended to other boundary conditions very easily since the boundary conditions only come in the solutions of the Poisson equation. Finally, the method is certainly applicable to more general domains as long as we can solve the Laplace equation on those domains.

ACKNOWLEDGMENT

The author thanks Professor Alexandre Chorin for many fruitful suggestions and very patient help during the preparation of this paper.

REFERENCES

1. C. R. Anderson, Lawrence Berkeley Laboratory Report LBL-26353, 1988, Berkeley, CA (unpublished).
2. A. J. Chorin, *Math. Comput.* **22**, 745 (1988).
3. A. J. Chorin, *Math. Comput.* **23**, 341 (1969).
4. F. H. Harlow and J. E. Welch, *Phys. Fluids* **8**, 2182 (1965).
5. Y. Maday and A. T. Patera, "Spectral Element Method for the Incompressible Navier–Stokes Equations," in *State-of-art Surveys on Computational Mechanics*, edited by A. K. Noor and J. T. Oden (ASME, New York, 1989).
6. A. B. Stephens, J. B. Bell, J. M. Solomon, and L. B. Hackerman, *J. Comput. Phys.* **53**, 152 (1984).
7. J. Strikwerda, *SIAM J. Sci. Statist. Comput.* **5**, 56 (1984).
8. P. N. Swarztrauber, *SIAM Rev.* **19**, No. 3, 490 (1977).