A Second-Order Projection Method for the Incompressible Navier–Stokes Equations in Arbitrary Domains*

ERIC YU TAU

Department of Medicine and Mathematics, Duke University, Durham, North Carolina 27710

Received November 19, 1993

We present a projection method for the numerical solution of the incompressible Navier–Stokes equations in an arbitrary domain that is second-order accurate in both space and time. The original projection method was developed by Chorin, in which an intermediate velocity field is calculated from the momentum equations which is then projected onto the space of divergence-free vector fields. Our method is based on the projection method developed by Bell and co-workers which is designed for problems in regular domains. We use the continuity equation to derive a pressure equation to compute the gradient part of the vector field. An integral form of the continuity equation is used to give us a natural way to define the discrete divergence operator for cells near the boundary which ensures the diagonal dominance of the resulting pressure equation. We then use the restarted version of the GMRES method to solve the pressure equation. © 1994 Academic Press, Inc.

1. INTRODUCTION

In many of the applications in engineering and mathematical modelling in medicine, it is often necessary to solve the Navier–Stokes equations in irregular domains. In this paper we present a second-order projection method for the incompressible Navier–Stokes equations in domains with irregular boundaries. The main ideas are adapted from the method developed by Bell, Colella, and Glaz [2] that solves incompressible flow problems in regular domains. The Navier–Stokes equations for incompressible flows are

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

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

on a domain Ω, where $\mathbf{u}$ is the velocity field, p is the pressure, and Re is the Reynolds number. Typical initial and boundary conditions for Eqs. (1) and (2) include specifying an initial velocity field $\mathbf{u}$ throughout the domain Ω and a boundary condition for $\mathbf{u}$ on the boundary $\partial\Omega$. No condition on p is required.

The projection method was first developed by Chorin [3]. It computes an intermediate vector field by ignoring the incompressibility condition and then projects the vector field onto a divergence-free field to obtain the velocity field. The method is based on the idea of Hodge decomposition [12, 3]. The original method was shown to be first-order accurate in time [4]. Peskin [7] adapted Chorin's method to irregular domains with moving, curved boundaries by embedding the physical domain into a larger regular domain and then solving the equations over the entire regular domain, including points outside the physical domain, with the physical boundaries replaced by some artificial forces. In a recent paper, Anjivel [9] used a different approach to adapt Chorin's method to irregular domains, in which only the physical domains were discretized and complicated finite difference approximations were used to approximate the Navier–Stokes equations near the boundaries. Here we use an approach that is similar to Anjivel's to develop a second-order projection method for irregular domains.

We start with a general description of the second-order projection method. If we rewrite Eqs. (1) and (2) as

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla p = \frac{1}{\mathrm{Re}}\Delta\mathbf{u} - (\mathbf{u}\cdot\nabla)\mathbf{u}, \tag{3}$$

$$\nabla\cdot\mathbf{u} = 0, \tag{4}$$

then by the Hodge decomposition, $\partial\mathbf{u}/\partial t$ is the divergence-free part of the right-hand side of Eq. (3). We then discretize Eqs. (3) and (4) in time, defining $\mathbf{u}^n$ to be the velocity field

* This work is supported in part by a post-doctorate fellowship from the Center for Indoor Air Research and EPA Cooperative Agreement CR820450 from the Environmental Criteria and Assessment Office.

0021-9991/94 $6.00

at $t = n\,\Delta t$ and $p^{n+1/2}$ to be the pressure at $t = (n + \frac{1}{2})\,\Delta t$, to obtain

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} + \nabla p^{n+1/2} = \frac{1}{\mathrm{Re}} \Delta \left(\frac{\mathbf{u}^{n+1} + \mathbf{u}^n}{2} \right) - [(\mathbf{u} \cdot \nabla)\mathbf{u}]^{n+1/2}, \tag{5}$$

$$\nabla \cdot \mathbf{u}^{n+1} = 0. \tag{6}$$

Given $\mathbf{u}^n$, we would like to find $\mathbf{u}^{n+1}$ and $p^{n+1/2}$ that satisfy Eqs. (5) and (6). For computational efficiency, we use the following iterative scheme (see [2] for the reason why). Assume that we have $\mathbf{u}^n$ and that we denote by $p^{n+1/2,k}$ the kth approximation to $p^{n+1/2}$. We first compute $\mathbf{u}^{*,k}$ by solving the following equation:

$$\frac{\mathbf{u}^{*,k} - \mathbf{u}^n}{\Delta t} + \nabla p^{n+1/2,k} = \frac{1}{\mathrm{Re}} \Delta \left(\frac{\mathbf{u}^{*,k} + \mathbf{u}^n}{2} \right) - [(\mathbf{u} \cdot \nabla)\mathbf{u}]^{n+1/2}. \tag{7}$$

Once $\mathbf{u}^{*,k}$ is computed, we decompose the right-hand side of Eq. (7) into

$$\frac{\mathbf{u}^{n+1,k+1} - \mathbf{u}^n}{\Delta t} + \nabla p^{n+1/2,k+1} = \frac{1}{\mathrm{Re}} \Delta \left(\frac{\mathbf{u}^{*,k} + \mathbf{u}^n}{2} \right) - [(\mathbf{u} \cdot \nabla)\mathbf{u}]^{n+1/2}, \tag{8}$$

where $\mathbf{u}^{n+1,k}$ is divergence-free. We then use $p^{n+1/2,k+1}$ to repeat the process until we achieve the convergence $p^{n+1/2,k+1} \to p^{n+1/2}$ and $\mathbf{u}^{n+1,k+1} \to \mathbf{u}^{n+1}$. The initial guess of $p^{n+1/2,0}$ can be taken as $p^{n-1/2}$ for $n > 1$ or zero for $n = 1$. It has been shown in practice (see [2]) that for small Δt iteration is not needed beyond the first time step. A simple reason for this is that we do not have a good approximation for p at the first time step.

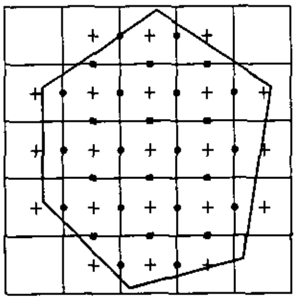
2. DISCRETIZATION OF DOMAIN

Assume that we have a domain Ω. We will assume that Ω has piecewise linear boundaries. Any other domain can be approximated by such a domain. We then cover domain Ω by a rectangle R with a Cartesian grid of mesh size h. For simplicity, we assume the same mesh size in both directions. Extension to different mesh sizes is straightforward. We use a staggered grid. We define the x-component of velocity, u, at the middle of right and left edges of each cell, the y-component of velocity, v, at the middle of top and bottom edges of each cell, and the pressure p at cell centers. So we have three sets of grid lines and grid points corresponding to u, v, and p, respectively. We also define the intersections of u-grid lines and v-grid lines with the boundaries of Ω to be u-boundary points and v-boundary points, respectively. As will be seen later, there is no need to define the p-boundary points. All the u- and v-grid points that are interior to domain Ω are marked as inside points. For pressure p, the inside points are defined as follows: for each p-grid point, if on the edges of the cell that has it as center there is at least one inside point of u or v, then the p-grid point is marked as an inside point. It is possible for an inside point of p to be physically outside of domain Ω (see Fig. 1). The reason for this definition of p is that this will give us two neighboring p-grid points for every inside u- or v-grid point, so that the appropriate component of ∇p can be easily defined (see later in this section).

Next we discretize the Navier–Stokes equations on all the inside grid points. If we denote $U = \{\text{all inside points of } u\}$, $V = \{\text{all inside points of } v\}$, and $P = \{\text{all inside points of } p\}$, then the x-component of the momentum equation is defined on U, the y-component of the momentum equation is on V, and the continuity equation is on P. We first discuss the evaluation of the terms in the momentum equations. We use the x-component as an example; the y-component is essentially the same. We would like to evaluate each term in the equation

$$\frac{u^{n+1} - u^n}{\Delta t} + \nabla_x^h p^{n+1/2} = \frac{1}{\mathrm{Re}} \Delta_h \left(\frac{u^{n+1} + u^n}{2} \right) - [N_x(\mathbf{u}^n)]^{n+1/2}, \tag{9}$$

where Δ_h is an approximation to the Laplacian, ∇_x^h (and ∇_y^h) is an approximation to the derivatives in the gradient operator, $[N_x(\mathbf{u}^n)]^{n+1/2}$ is an approximation to the x-component of the nonlinear term. We used $\mathbf{u}^n$ in $N_x(\mathbf{u}^n)$ to indicate the fact that the nonlinear term is evaluated from the knowledge of $\mathbf{u}^n$ only.



• ∈ U ▪ ∈ V + ∈ P

FIG. 1. The definition of grids.

The operator $\nabla_x^h(P \to U)$ is defined on every point of U by

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

Because of our definition of P, if $u_{i+1/2,j} \in U$, then we will have $p_{i+1,j} \in P$ and $p_{i,j} \in P$. So $\nabla_x^h p$ is well defined on every point in U. For Δ_h we use the standard five-point approximation for interior points, and for points near the boundary we can use either first-order or second-order approximations, depending on whether we can find enough grid points and boundary points. Take the x derivative, for example. If a grid point A (see Fig. 2) has only one boundary point as its neighbor in x direction, then on the other side the neighbor will be a grid point B. In this situation a fourth point D can be found on the side of B away from A, which can be a grid point (as is the case in Fig. 2) or a boundary point. We then use those four points to form a second-order approximation to $\partial^2 u/\partial x^2$. If both neighbors are boundary points (grid point E in Fig. 2) we then use the three points (the two boundary points F, G plus E itself) to form a first-order approximation. For approximation of the nonlinear term, we follow the idea in [2]. Some modifications have to be made because we use a different grid and also because of the irregular domain. We define, suppress the superscript $(n+\frac{1}{2})$,

$$[N_x(\mathbf{u}^n)]_{i+1/2,j} = \frac{u_{i+1,j} + u_{ij}}{2}\left(\frac{u_{i+1,j} - u_{ij}}{h}\right) + \frac{v_{i+1/2,j+1/2} + v_{i+1/2,j-1/2}}{2} \times \left(\frac{u_{i+1/2,j+1/2} - u_{i+1/2,j-1/2}}{h}\right),$$

where u_{ij} and v_{ij} are velocities at cell centers, and $u_{i\pm 1/2,j\pm 1/2}$ and $v_{i\pm 1/2,j\pm 1/2}$ are velocities at cell corners. Velocities are evaluated at time $t = (n+\frac{1}{2})\,\Delta t$ using the Taylor series of $\mathbf{u}^n$ from both sides. For instance,

$$u_{i+1/2,j+1/2}^{n+1/2,B} = u_{i+1/2,j}^n + \frac{h}{2} u_{y,i+1/2,j} + \frac{\Delta t}{2} u_{t,i+1/2,j},$$

$$u_{ij}^{n+1/2,R} = u_{i+1/2,j}^n - \frac{h}{2} u_{x,i+1/2,j} + \frac{\Delta t}{2} u_{t,i+1/2,j}.$$

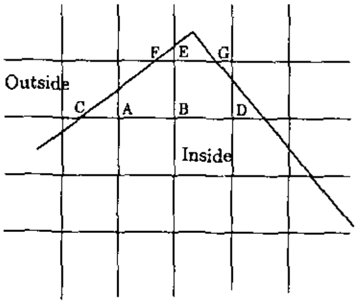


FIG. 2. Points used to approximate $\partial^2 u/\partial x^2$ for grid points A and E.

The spatial derivatives are constructed by first forming a centred difference approximation and then using a limiting process so that no new maxima or minima are introduced. For temporal derivatives, we go back to the Navier–Stokes equations to obtain an approximation for $\mathbf{u}_t$ by information at the previous time step (for more details, see [2]). For points near boundaries we use the known velocity at time $t = (n+\frac{1}{2})\,\Delta t$ so the time expansion in the Taylor series will not be needed for those points.

Now we have to eliminate the ambiguities. We use an upwind-like scheme. For velocities at centers, we define

$$u_{ce} = \begin{cases} u_{ce}^L & \text{if} \quad u_{ce}^L \geqslant 0, \quad u_{ce}^L + u_{ce}^R \geqslant 0 \\ 0 & \text{if} \quad u_{ce}^L < 0, \quad u_{ce}^R > 0 \\ u_{ce}^R & \text{otherwise.} \end{cases}$$

and similarly for v_{ce}. At the corners, we first look for u_{co}^B, u_{co}^T or v_{co}^L, v_{co}^R that have the same sign. For instance, say $u_{co}^B \geqslant 0$ and $u_{co}^T \geqslant 0$, we then define $v_{co} = c_{co}^{\mathrm{L}}$, or if $u_{co}^B \leqslant 0$ and $u_{co}^T \leqslant 0$ we define $v_{co} = v_{co}^{\mathrm{R}}$. Once v_{co} is defined, we then define

$$u_{co} = \begin{cases} u_{co}^B & \text{if} \quad v_{co} > 0 \\ u_{co}^T & \text{if} \quad v_{co} < 0 \\ \frac{1}{2}(u_{co}^B + u_{co}^T) & \text{if} \quad v_{co} = 0; \end{cases}$$

similarly when v_{co}^{L}, $v_{\mathrm{co}}^{\mathrm{R}}$ have the same sign. If they both have the same sign, either can be used. It can be easily verified that they will yield the same result. If none has the same sign, then we define

$$u_{co,ij} = \tfrac{1}{2}(u_{co}^B + u_{co}^T)$$

and

$$v_{co,ij} = \tfrac{1}{2}(v_{co}^L + v_{co}^R).$$

Since the method is explicit, we require that

$$\max\left(\frac{u_{ij}\,\Delta t}{h}, \frac{v_{ij}\,\Delta t}{h}\right) \leqslant 1$$

for stability.

The last step in each iteration is the projection itself. We want to compute $\mathbf{u}^{n+1}$ from $\mathbf{u}^*$ such that

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} + (\nabla_x, \nabla_y)\, p^{n+1/2} = \mathbf{f}(\mathbf{u}^*, \mathbf{u}^n) \qquad (10)$$

and

$$\nabla \cdot \mathbf{u}^{n+1} = 0, \qquad (11)$$

where $\quad \mathbf{f}(\mathbf{u}^*, \mathbf{u}^n) = (1/\mathrm{Re})\,\Delta_h((\mathbf{u}^* + \mathbf{u}^n)/2) - [\mathbf{N}(\mathbf{u}^n)]^{n+1/2}$.

To avoid the difficulty of defining the finite difference approximations to the divergence operator near the boundary, we use the integral form of Eq. (11),

$$\frac{1}{h^2}\oint_{\text{cell}} \mathbf{u}^{n+1}\cdot d\mathbf{r}=0, \tag{12}$$

where the equation is defined for each and every cell that has its center as an inside p-point. We use the midpoint rule for the integration. For cells that are completely inside the physical domain, we have, for cell (i, j),

$$\begin{aligned}\frac{1}{h^2}\oint_{\text{cell}} \mathbf{u}\cdot d\mathbf{r} &\approx \frac{1}{h^2}\,(u_{i+1/2,j}h-u_{i-1/2,j}h\\ &\quad+v_{i,j+1/2}h-v_{i,j-1/2}h)\\ &=\left(\frac{u_{i+1/2,j}-u_{i-1/2,j}}{h}+\frac{v_{i,j+1/2}-v_{i,j-1/2}}{h}\right)\\ &=0\end{aligned} \tag{13}$$

So Eq. (12) effectively defines a centered difference approximation to the divergence operator (see, e.g., [11]). For cells that are only partially inside, we define, e.g., for cell $ABCDE$ (see Fig. 3),

$$\begin{aligned}\frac{1}{h^2}\oint_{\text{cell}} \mathbf{u}\cdot d\mathbf{r} &\approx \frac{1}{h^2}\Big((y_B-y_A)\,u_{AB}-(x_C-x_B)\,v_{BC}\\ &\quad+(y_D-y_C)\,u_{CD}-(x_E-x_D)\,v_{DE}\\ &\quad+\frac{1}{2}(u_E+u_A)(y_A-y_E)\\ &\quad-\frac{1}{2}(v_E+v_A)(x_A-x_E)\Big)\\ &=0\end{aligned} \tag{14}$$

where x_A, y_A are x and y coordinates of point A, and u_{AB} is the x-component of velocity at the middle of segment AB, which is obtained by the interpolation of u_A and the other nearest neighbor, in this case the velocity at the inside u-grid point at the middle of BF; similarly for v_{DE}. v_{BC} and u_{CD} are the velocity components at the grid points at the centers of segments BC and CD. Equation (14) could be used to define approximations to the divergence operator near the boundary. We should point out that approximations like (14) are only first order for cells near the boundary. But as we will see from our test results, this will not affect the overall accuracy of the scheme.

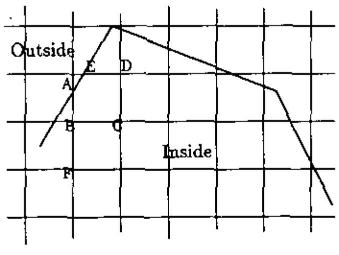


FIG. 3. The definition of a boundary cell.

With those definitions we will not have

$$(D^h\mathbf{u}, \phi)_s=-(\mathbf{u}, G^h\phi)_v, \tag{15}$$

where D^h and G^h are discrete approximations to divergence and gradient operators; $(\cdot,\cdot)_s$ and $(\cdot,\cdot)_v$ represent appropriate inner products on discrete spaces of scalars and vectors, respectively. But Eq. (15) is violated near the boundary only, so that it will not present us with any major difficulty for the projection (see Chorin [3] and Soloman and Szymczak [10]).

The other advantage of using the integral form (12) is that it will guarantee the diagonal dominance of the resulting pressure equation (17), thus the stability. So we can avoid the difficulty of choosing appropriate parameters to ensure the stability of the system as was done in [9].

If we move the terms that involve velocity values on the boundary which are known in Eq. (14) to the right-hand side, we then have

$$D\mathbf{u}^{n+1}=F(\mathbf{u}_b^{n+1}), \tag{16}$$

where D is the coefficient matrix and $\mathbf{u}_b^{n+1}$ is the known velocity value on the boundary at time $t=(n+1)\,\Delta t$. Next we use Eqs. (10) and (16) to eliminate the velocity to form an equation for the pressure p (for details see Anderson [1] and Tau [11]),

$$Ap=b. \tag{17}$$

After we solve Eq. (17) for p then we go back to Eq. (10) to find the divergence-free $\mathbf{u}^{n+1}$. This completes the projection.

3. SOLUTIONS TO THE EQUATIONS AND THE GMRES METHOD

Because of the irregularity of the domain, both systems of Eqs. (7) and (17) have nonsymmetric coefficient matrices. We use the restarted version of the generalized minimal residual method (GMRES(m)) developed by Saad and Schultz [8] to solve both systems. Equation (7) is a Helmholtz equation with a negative coefficient $-2\text{Re}/\Delta t$ and the resulting linear system is diagonal dominant and very well conditioned. The GMRES(m) algorithm will converge very fast for this equation. Equation (17) is a five-point approximation to the Laplace equation for the points that are not near the boundary (see [1]). For other points the equation can be shown to be still diagonal dominant but barely. The GMRES(m) algorithm converges much slower for this equation than for Eq. (7) and most of the work for one iteration step is done to solve this equation. The steps of the GMRES(m) algorithm are as follows:

1. *Start.* Choose x_0 and compute $r_0 = b - Ax_0$ and $v_1 = r_0/\|r_0\|$.
2. *Iterate.* For $j = 1, 2, ..., m$ do:

$$h_{i,j} = (Av_j, v_i),\ i = 1, 2, ..., j,$$

$$\hat{v}_{j+1} = Av_j - \sum_{i=1}^{j} h_{i,j} v_i,$$

$$h_{j+1,i} = \|\hat{v}_{j+1}\|, \text{ and}$$

$$v_{j+1} = \hat{v}_{j+1}/h_{j+1,i}.$$

3. *Form the approximate solution.*

   $x_m = x_0 + V_m y_m$, where y_m minimizes $\|\beta e_1 - H_m y\|$, $y \in R^m$,

   with $\beta = \|r_0\|$, $H_m = \{h_{i,j}\}_{(m+1)\times m}$ and $V_m = \{v_1, v_2, ..., v_m\}$.

4. *Restart.*

   Compute $r_m = b - Ax_m$; if satisfied then stop,
   else compute $x_0 = x_m$, $v_1 = r_m/\|r_m\|$ and go to 2.

For details on the implementation of the algorithm, we refer our readers to [8]. The second step in the algorithm is the Gram–Schmidt process for finding an l_2-orthonormal basis $\{v_1, v_2, ..., v_m\}$ of the Krylov subspace $K_m = \text{span}\{v_1, Av_1, ..., A^{m-1}v_1\}$. In practice we find out that when GMRES(m) is used to solve the pressure equation (17), reorthogonalization has to be used to guarantee the orthogonality of basis vectors $\{v_1, v_2, ..., v_m\}$. We use the Gram–Schmidt process with reorthogonalization described in [5].

4. NUMERICAL RESULTS

To test our algorithm, we apply it to a domain with one bifurcation branch (see Fig. 4) which is bounded by a $\pi \times 0.7\pi$ rectangle. We solve the following test problem with known solutions:

$$u = -\cos(x)\sin(y)\, e^{-2t/\text{Re}}, \qquad v = \sin(x)\cos(y)\, e^{-2t/\text{Re}},$$

$$\frac{\partial p}{\partial x} = \frac{1}{2}\sin(2x)\, e^{-4t/\text{Re}}, \qquad \frac{\partial p}{\partial y} = \frac{1}{2}\sin(2x)\, e^{-4t/\text{Re}}.$$

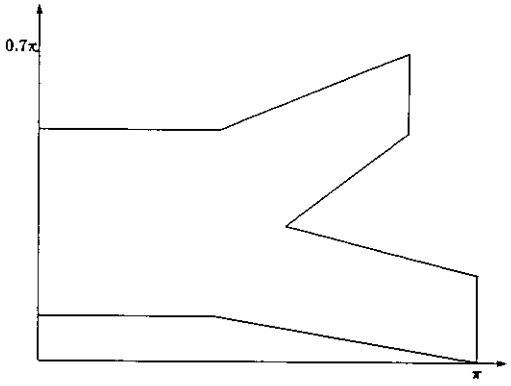


FIG. 4. A domain with one bifurcation.

Computations were performed for Reynolds numbers 100, 500, 2000, and 5000, with grid sizes of $\Delta x = \pi/2^n$, $\Delta y = 0.7\pi/2^n$ for $n = 4, 5, 6, 7$. For each grid size the time step is taken by $\Delta t = 0.25(\Delta x + \Delta y)$. In our solutions to the linear systems, we use the residual r as our computation criterion, i.e., the iteration stops when $\|r\|_2$ is less than a predetermined small positive number ε. Here $\|r\|_2$ is defined as a equivalent of L_2 norm by

$$\|r\|_2 = \left(\frac{1}{N} \sum_{\text{inside}(i,j)} r_{i,j}^2\right)^{1/2},$$

where N is the total number of points inside the domain. In all of our computations we set $\varepsilon = 1 \times 10^{-5}$. We should mention that by our construction the residual of the pressure equation is exactly the divergence of the velocity field; i.e., we have

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

for the pressure equation. This guarantees that the L_2 norm of the numerical divergence of the resulting velocity field is bounded by ε. The velocity field is computed up till time $T = 0.392699$ (a maximum of 32 time steps for $n = 7$). The error of velocity field is measured by

$$e(\mathbf{u}) = \max(e(u), e(v)),$$

where $e(u)$ and $e(v)$ are the maximum errors of u, v measured against the exact solution, respectively. The rate of convergence is defined by

$$R = \log_2 \frac{\|e\|^n}{\|e\|^{n+1}}.$$

TABLE I

Convergence Results at Time $T = 0.392699$

Re	16 × 16	Rate	32 × 32	Rate	64 × 64	Rate	128 × 128
100	3.160E − 3	1.978	8.021E − 4	2.076	1.902E − 4	2.173	4.219E − 5
500	3.361E − 3	1.877	9.086E − 4	1.990	2.287E − 4	2.046	5.536E − 5
2000	3.403E − 3	1.863	9.353E − 4	1.964	2.397E − 4	1.969	6.120E − 5
5000	3.412E − 3	1.858	9.411E − 4	1.959	2.421E − 4	1.982	6.128E − 5

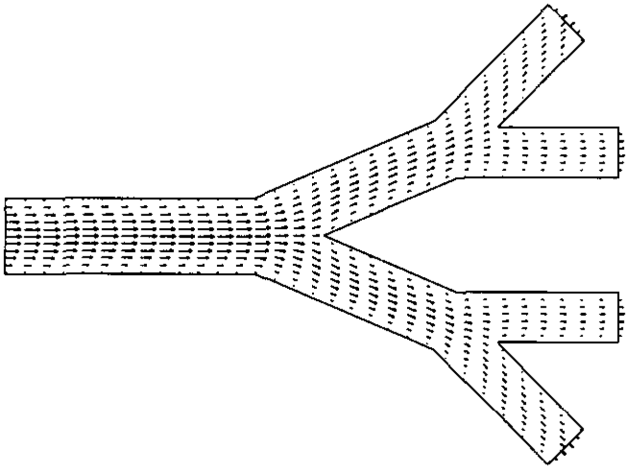


FIG. 5. The velocity field for a model of human conducting airways.

The results indicate clearly that we achieved second-order accuracy. Any deviation from second order is probably due to the fact that our approximation to the continuity equation is only first order for cells near the boundary. Two more things worth noting from the results. One is that, for a fixed Reynolds number, the convergence rate increases slightly from coarse grid to fine grid. A possible explanation for this is that as we refine the grid the number of boundary cells decreases percentage-wise so the effect of first-order approximation for the boundary cells becomes less significant. The other is that the rate decreases as the Reynolds number increases. The reason for this is probably that as the Reynolds number increases, the effect of numerical diffusion becomes more significant (see Table I).

For our second example, we use the algorithm to compute the steady velocity field of an air flow in human conducting airways. We use a model with two bifurcations (data from [6]) as shown in Fig. 5. We specify the velocity boundary conditions to be parabolic at the entrance and exits, and zero everywhere else. The maximum of the parabolic profile at the entrance is derived from the peak volume flow under normal breathing conditions and the maxima at the exits are adjusted such that the net flow into the region is zero. The velocity field is displayed in Fig. 5. A 64 × 64 grid is used and a steady state is achieve after 20 time steps. The Reynolds number of the flow is about 24.

5. CONCLUSIONS

In this paper we have presented a projection method for the incompressible Navier–Stokes equations in an arbitrary domain. Our numerical example illustrates that the method is second-order accurate in both time and space. We used the integral form of the continuity equation to derive a pressure equation that is always stable. The use of the integral form gave us a natural way to define the discrete divergence operator for cells near the boundary so we avoided the difficulty of choosing the appropriate discrete divergence operator near the boundary to ensure the stability of the resulting pressure equation.

One drawback of the method is that when applied to complicated domains the solution to the pressure equation converges very slowly. Some preconditioning to the pressure equation may improve the convergence rate. Future efforts will be addressed to this issue.

ACKNOWLEDGMENT

The author thanks Dr. S. Anjivel for many helpful suggestions during our discussions and for allowing the author to use some of the codes of his program.

REFERENCES

1. C. R. Anderson, Lawrence Berkeley Laboratory Report LBL-26353, 1988, Berkeley, CA (unpublished).
2. J. Bell, P. Colella, and H. Glaz, *J. Comput. Phys.* **85**, 257 (1989).
3. A. J. Chorin, *Math. Comput.* **22**, 745 (1968).
4. A. J. Chorin, *Math. Comput.* **23**, 341 (1969).
5. J. W. Daniel, W. B. Gragg, L. Kaufman, and G. W. Stewart, *Math. Comput.* **30**, 772 (1976).
6. D. E. Olson, G. A. Dart, and G. F. Filley, *J. Appl. Physiol.* **28** (4), 482 (1970).
7. C. S. Peskin, *J. Comput. Phys.* **10**, 252 (1972).
8. Y. Saad and M. H. Schultz, *SIAM J. Sci. Stat. Comput.* 7 (3), 856 (1986).
9. S. Anjivel, *J. Comput. Phys.*, to appear.
10. J. M. Soloman and W. G. Szymczak, "Finite Difference Solutions for the Incompressible Navier-Stokes Equations using Galerkin Technique," in *Fifth IMACS International Symposium on Computer Methods for Partial Differential Equations, Lehigh University, June 19–21, 1984.*
11. E. Y. Tau, *J. Comput. Phys.* **99**, 190 (1992).
12. R. Temam, *Navier–Stokes Equations* (Elsevier Science, Amsterdam, 1984).