JOURNAL OF COMPUTATIONAL PHYSICS **125,** 207–224 (1996)
ARTICLE NO. 0089

A High Order Explicit Method for the Computation of Flow About a Circular Cylinder

CHRISTOPHER R. ANDERSON

Department of Mathematics, University of California at Los Angeles, Los Angeles, California 90024

AND

MARC B. REIDER

Department of Mathematics, IBM T. J. Watson Research Center, Yorktown Heights, New York 10598

Received July 30, 1993; revised October 10, 1995

In this paper a finite difference method for computing the solutions of the incompressible Navier–Stokes equations for flow about a circular cylinder in two dimensions is presented. A stream function/vorticity formulation of the equation is used and the numerical method incorporates recent developments in the computation of vorticity boundary conditions as well as far field boundary conditions. Three schemes are described, one of second order accuracy, one of fourth order accuracy, and a hybrid method which is second order accurate in the computation of the vorticity transport and fourth order accurate in the determination of the stream function. Fully resolved solutions for flow past a cylinder have been computed over a range of Reynolds numbers from 1000 to 9500. Comparisons are made between the results obtained with methods of different orders of accuracy as well as of the effectiveness of the vorticity and far field boundary conditions.

1. INTRODUCTION

The purpose of this paper is to describe a finite difference method for computing two dimensional viscous incompressible flow past a circular cylinder and to present the results of computations performed with this method. The computations consist of determining fully resolved, short time solutions of an impulsively started cylinder up to Reynolds number 9500.

Flow about a circular cylinder is widely used as a test problem because it exhibits features of unsteady viscous flow past bluff bodies—especially unsteady vortex shedding. Moreover, there are experimental data [28, 5, 16] and the simple geometry makes it the problem of choice for testing numerical methods and mathematical modeling procedures. For example, flow about a circular cylinder was used as a test problem by Chorin [9] to introduce the vortex blob method and has served as test problems for refinements of the technique [8, 24, 27]. With regard to finite difference methods, early work at modest Reynolds numbers was performed by [13, 25]. More recently, through the use of advanced numerical techniques (e.g., compact high order difference stencils and exponentially stretched grids), solutions have been obtained for higher Reynolds numbers [14, 15, 7, 6]. This problem has also been the focus of mathematical and analytical modeling; for example, see Sarpkaya [23, 22].

Our interest in obtaining accurate, fully resolved solutions to the Navier–Stokes equations for flow about a circular cylinder was motivated by our work on boundary layer models [4]. While existing finite difference methods could have been used to provide such solutions, they do not incorporate some recent results having to do with the treatment of infinite boundary conditions (using essentially domain decomposition procedures) [2] and vorticity boundary conditions [3]. This paper describes a finite difference method which incorporates these results.

The resulting method is an explicit finite difference method of second order (or fourth order) accuracy in space and of fourth order accuracy in time. The vorticity boundary conditions that are presented allow one to easily obtain high accuracy in both the interior and boundary values of vorticity. This formulation of the boundary conditions also allows one to implement time dependent boundary conditions without any loss of time accuracy.

The specific problem that we consider in this paper is the computation of the viscous flow about a circular cylinder as it is accelerated from rest to unit velocity. The length of time over which we compute the solution to this problem is what might be considered short time—a time interval in which the cylinder has translated one or two diameters. However, during this time interval the solutions reveal a rather complex shedding structure (particularly at high Reynolds number) and thus provide useful solutions for investigating the process of boundary layer separation phenomena.

0021-9991/96 $18.00

In the solutions we compute the vorticity in the fluid is confined to a small region about the cylinder and we need only use a grid which extends one or two cylinder radii downstream to capture its evolution. Furthermore, the use of infinite-domain boundary conditions allows us to use the same grid for the computation of the stream function. The limited extent of the computational grid allows us the luxury of having a grid which is uniform in each direction, both normal and tangent to the cylinder. This is particularly important for the problem considered here in which the boundary layer separates and enters the fluid. In this case vorticity gradients are not aligned with the cylinder surface and a grid which has good resolution in both directions is needed.

In the first section we shall describe the specific problem we are solving and provide the relevant equations. In the second section we give the details of the numerical method. The latter will include a description of the procedures for discretization of the equations, for treating the infinite computational domain, for implementing the vorticity boundary conditions and the procedures used for time discretization. In the remaining sections we provide numerical results and discuss issues related to initial conditions and other computational aspects.

2. PROBLEM DESCRIPTION AND EQUATIONS OF MOTION

The problem we are considering is the determination of the two dimensional motion of the fluid which arises when a circular cylinder is accelerated from rest to a unit velocity. We carry out the computation in a coordinate frame fixed to the cylinder, and so we formulate the problem as one of determining the motion of the fluid about a stationary cylinder with a variation in the velocity at infinity.

The Navier–Stokes equations for an incompressible fluid of constant density are used to describe the fluid motion. When non-dimensionalized and expressed in polar coordinates, the vorticity stream function formulation of the Navier–Stokes equations becomes

$$\frac{\partial \omega}{\partial t} + \left(u, \frac{v}{r}\right) \cdot \left(\frac{\partial \omega}{\partial r}, \frac{\partial \omega}{\partial \theta}\right) = \frac{1}{\mathrm{Re}} \left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial \omega}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 \omega}{\partial \theta^2}\right) \quad (1)$$

$$u = \frac{1}{r}\frac{\partial \Psi}{\partial \theta} + b(t)u_\infty(r, \theta) \quad v = -\frac{\partial \Psi}{\partial r} + b(t)v_\infty(r, \theta) \quad (2)$$

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial \Psi}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 \Psi}{\partial \theta^2} = -\omega \quad (3)$$

for $r \geq r_a$ and $0 \leq \theta \leq 2\pi$. Here (u, v) are the radial and tangential components of the velocity, respectively, ω is the vorticity, and Ψ is the stream function. The velocity field $(u_\infty(r, \theta), v_\infty(r, \theta))$ corresponds to the potential flow associated with a uniform (Cartesian) velocity at infinity $(U_\infty, 0)$; i.e.,

$$(u_\infty(r, \theta), v_\infty(r, \theta)) = \left(U_\infty\left(1 - \frac{r_a^2}{r^2}\right)\cos(\theta), -U_\infty\left(1 + \frac{r_a^2}{r^2}\right)\sin(\theta)\right).$$

The non-dimensionalization is based on the cylinder diameter $2r_a$ and the velocity at infinity; $\mathrm{Re} = 2r_aU_\infty/\nu$. $b(t)$ is a function such that $b(t) = 0$ for $t < 0$ and $b(t) = 1$ at some time $t \geq 0$. (If $b(t) = 1$ for $t = 0$ then this is the problem of an impulsively started cylinder.)

The initial and boundary conditions for this problem are

$$\omega(r, \theta, 0) = 0 \qquad r_a \leq r \leq \infty \quad 0 \leq \theta \leq 2\pi \quad (4)$$

$$(u(r, \theta, t), v(r, \theta, t)) = b(t)(u_\infty(r, \theta), v_\infty(r, \theta)) \qquad r \to \infty \quad 0 \leq \theta \leq 2\pi \quad (5)$$

$$(u(r, \theta, t), v(r, \theta, t)) = (0, 0) \qquad r = r_a \quad 0 \leq \theta \leq 2\pi. \quad (6)$$

The boundary conditions on the velocity, (5) and (6), will be satisfied if the boundary conditions on the stream function Ψ satisfy

$$\Psi(r, \theta, 0) = 0 \qquad r \to \infty \; 0 \leq \theta \leq 2\pi \quad (7)$$

$$\Psi(r, \theta) = 0 \qquad r = r_a \; 0 \leq \theta \leq 2\pi \quad (8)$$

$$\frac{\partial \Psi}{\partial r}(r, \theta) = -b(t)v_\infty(r, \theta) \qquad r = r_a \; 0 \leq \theta \leq 2\pi. \quad (9)$$

The presence of the diffusive term in (1) implies that instantaneously after the fluid motion is started the associated vorticity has a finite non-zero value at every point in the domain. However, for short times, the vorticity is concentrated in a region close to the cylinder and is exponentially small away from this region. We therefore simplify our computational task by neglecting this exponentially small amount of vorticity and only determine the values of the vorticity in the annular region about the cylinder—the region described by $r_a \leq r \leq r_b$ and $0 \leq \theta \leq 2\pi$. The accuracy of this procedure is checked by comparing the solutions obtained with two different values of r_b. As will be seen from the computational results, for our particular choice of r_b the errors incurred are indeed negligible.

While we ignore the vorticity outside of the region $r = r_b$ we do not impose the boundary condition on the stream function (7) at $r = r_b$. Making such an approximation leads to large errors, and, in fact, a significant component of this paper is to describe how one can effectively implement the correct "infinite" boundary condition on Ψ.

3. NUMERICAL METHOD

The equations for vorticity transport (1) are of the convection–diffusion type, while the velocity field is determined by evaluating derivatives of a solution of Poisson's equation (2), (3). The basis for our discretization technique is the method of finite differences. Our choice of this type of discretization is motivated by the fact that for the simple geometry associated with this problem high accurate difference formulas for the convection and diffusion equation can be easily constructed. One can also utilize the discrete fast Fourier transform to obtain efficient solutions of the Poisson equation which determines Ψ. In order to implement a finite difference method one must

(i) Select a finite difference grid.

(ii) Select a finite difference discretization for the convection and diffusion of the vorticity (1).

(iii) Select a discretization (and corresponding solution procedure) for the Poisson equation (3). (We shall use the boundary conditions (7) and (8) to determine Ψ; the other boundary condition, (9) will be satisfied through an appropriate choice of the boundary vorticity values.)

(iv) Determine a technique such that the computation of Ψ while being performed on a finite domain, approximates the solution on the infinite domain.

(v) Translate the velocity boundary conditions (or equivalently the stream function boundary condition (9)) into a boundary condition for the vorticity. (The boundary vorticity needs to be determined in order to close the convection diffusion equations for the vorticity transport.)

(vi) Select an appropriate time-stepping strategy.

In the following sections we describe how we have addressed each of these problems.

3.1. Grid

The grid used for our discretization is a polar grid which extends from the surface of the cylinder r_a to a distance r_b. We will assume a uniform mesh spacing $\delta\theta$ in the θ direction and δr in the radial direction. We will designate the values of the vorticity on the computational grid as $\omega_{i,j} = \omega(i\delta\theta, r_a + j\delta r)$, with $i = 1 \ldots M$ and $j = 1 \ldots N + 1$, $M = \dfrac{2\pi}{\delta\theta}$ and $N = (r_b - r_a)/\delta r$. The values for the velocity and stream function are also labeled similarly.

3.2. Finite Difference Approximations

In the computation of this problem we chose to use "standard" centered difference approximations. This requires some explantion, as centered schemes can have undesirable properties, and this has meant that there is a general consensus that upwind or non-centered schemes are the methods of choice for incompressible flow problems.

There are essentially two problems which have to do with the use of centered schemes. The first problem has to do with the stability (in time) of centered difference approximations for the convection terms. Without a dissipative term in the equation of motion (i.e., when the viscosity equals zero (Re = ∞)), certain commonly used explicit time differencing procedures such as forward Euler and Huen's method (a second order Runge–Kutta method) are unstable. This problem can be overcome by using a time differencing method whose region of stability contains some portion of the imaginary axis in the complex plane. Two Runge–Kutta methods which satisfy this condition are the standard third and fourth order Runge–Kutta methods (see [1]). We chose to use fourth order Runge–Kutta—its larger stability region makes the computation more efficient.

The second problem has to do with the fact that centered difference approximations create spatially oscillating results in the presence of steep gradients. However, for viscous problems, if one increases the resolution while keeping the Reynolds number constant, then this problem disappears. For those familiar with the cell Reynolds number concept [20], the disappearance of oscillations is coincident with a decrease in the cell Reynolds number. Since we are interested in completely resolving the fluid motion, and, in particular, the viscous profiles present in the problem, we shall be using a sufficiently fine grid for which these oscillations do not occur. It is for this reason that we see no need for, and in fact would not want to use, an upwind scheme. As is revealed by a consideration of the truncation errors associated with centered versus non-centered differentiation formulas [1], for a given order of discretization, upwind schemes have a larger truncation error than centered schemes.

At the interior points we used

$$\frac{\partial f}{\partial x} \approx D_h^0 f = \frac{f(x+h) - f(x-h)}{2h}$$

for second order difference approximations to the derivatives occurring in the convective terms of (1) and

$$\frac{\partial f}{\partial x} \approx \tilde{D}_h^0 f = \frac{-f(x+2h) + 8f(x+h) - 8f(x-h) + f(x-2h)}{12h}$$

for the fourth order approximations. Here h is the mesh width, $\delta\theta$ or δr.

The Laplace operator in (1) and (3) was discretized using standard second and fourth order differencing. The second order discretization used was

$$\Delta\omega \approx D_r^+ D_r^- \omega + \frac{1}{r} D_r^0 \omega + \frac{1}{r^2} D_\theta^+ D_\theta^- \omega \tag{10}$$

where D^+ and D^- are the forward and backward divided difference operators. The differencing for the fourth order approximation consisted of replacing the second order difference operators in (10) by fourth order accurate difference operators. Also, in the fourth order approximation, fourth order ''one-sided'' difference operators were used near the boundary. (The formulas can be found in [1].)

At the cylinder surface, the values of the vorticity are explicitly determined (by a procedure described below), and so are used to close the finite difference approximations at the cylinder surface. At the outer boundary of the computational domain (the points with $r = r_b$) we used second order upwind differencing in the approximation of the convective terms in the equation. If the fluid flow is entering the computational region (thus necessitating the convection of vorticity from outside to inside the computational region) the value of the vorticity convected in from the exterior of the computational domain is taken to be identically zero. For the diffusive term at $r = r_b$, Neumann boundary conditions were used to close the difference approximation. In essence, the outflow boundary conditions we employed allowed for convection transport of the vorticity out of the domain but not diffusive transport. We used these boundary conditions for both the fourth and second order approximations. Our primary goal in selecting these boundary conditions was to ensure that the outflow conditions on the difference approximations did not give rise to oscillations which would contaminate the interior solution values. In the time over which our solutions were computed the vorticity was completely confined to the computational domain and so these boundary conditions did not influence the accuracy of the solution in the interior. (As will be seen in the computational results, these simple outflow boundary conditions worked very well.)

If we leave the time variable continuous (the method of lines approach) then the numerical approximation for the transport of vorticity given by Eq. (1) consists of solving the coupled set of ordinary differential equations

$$\frac{d\omega_{i,j}}{dt} = -\left(u_{i,j}, \frac{v_{i,j}}{r}\right) \cdot (D_r^0 \omega_{i,j}, D_\theta^0 \omega_{i,j}) + \frac{1}{\mathrm{Re}} \left(D_r^+ D_r^- + \frac{1}{r} D_r^0 + \frac{1}{r^2} D_\theta^+ D_\theta^-\right) \omega_{i,j}, \tag{11}$$

where the difference operators were modified near the boundaries as described above. The values for the velocity (11) are determined by using centered difference approximations of the stream function,

$$u_{i,j} = \frac{1}{r} D_\theta^0 \Psi_{i,j} \quad v_{i,j} = -D_r^0 \Psi_{i,j}. \tag{12}$$

As in the equation for transport of vorticity, the appropriate one-side stencils were used for the fourth order discretization next to boundaries. The values for the stream function are determined by the procedure described in the next section.

3.3. Computation of the Stream Function

At a fixed time, for a given vorticity distribution, the velocity field is computed by determining the stream function and then differentiating the result. The problem which determines the stream function from the vorticity is the following:

$$\Delta\Psi = \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial \Psi}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 \Psi}{\partial \theta^2} = -\omega \tag{13}$$

$$\Psi(r, \theta) = 0 \; r = r_a; \quad \Psi(r, \theta) = \Psi_\infty(r, \theta) \; r \to \infty \quad 0 \le \theta \le 2\pi. \tag{14}$$

Here Ψ_∞ is the boundary condition at infinity which includes a specification of the net circulation κ. For our problem the net circulation is zero and so we use $\Psi_\infty(r, \theta) = 0$. If we had included a fixed amount of circulation κ (i.e., flow about a rotating cylinder) then we would have used $\Psi_\infty(r, \theta) = (\kappa/2\pi) \log(r)$.

Thus, the problem determining Ψ is a Poisson equation in an infinite domain with Dirichlet boundary conditions.

3.3.1. *The Treatment of The Infinite Domain*

The method we employ is a slight modification of the procedure described in [2]. In order to obtain an approximation to (13) which incorporates the boundary conditions

at infinity, we use the following form for the stream function

$$\Psi \approx \begin{cases} (\Psi_g)_{i,j} & 0 \le \theta_i \le 2\pi \; r_a \le r_j \le r_b \\ \Psi_f(r,\theta) + \Psi_H(r,\theta) & 0 \le \theta \le 2\pi \; r \ge r_b. \end{cases}$$

Here $(\Psi_g)_{i,j}$ is the solution of the discrete Poisson equation $(\Delta^h \Psi_g)_{i,j} = \omega_{i,j}$ in the annulus $r_a \le r_j \le r_b$. $\Psi_H(r, \theta)$ is a specified harmonic function taking the prescribed boundary conditions at infinity (including a prescribed amount of net circulation). The function ψ_f is a finite Fourier series solution to Laplace's equation in the region $r \ge r_b$:

$$\Psi_f(r,\theta) = \sum_{k=-M/2+1}^{k=M/2} \left(\frac{r}{r_a}\right)^{-|k|} \alpha_k e^{ik\theta}.$$

A property of this approximation is that it is harmonic for $r > r_b$ and thus the representation is valid only if the vorticity is contained within the region $r < r_b$. In our computations this requirement is satisfied. Another property of this approximation is that it automatically satisfies the required boundary condition at infinity.

The computational task is to determine the grid values Ψ_g and the coefficients α_k in the Fourier series component Ψ_f so that the resulting function approximates the solution to (13), (14). We therefore require the approximation to satisfy

$$(\Delta^h \Psi_g)_{i,j} = -\omega_{i,j} \qquad i = 1 \ldots M, j = 2 \ldots N \quad (15)$$

$$(\Psi_g)_{i,1} = 0 \qquad i = 1 \ldots M \quad (16)$$

$$(\Psi_g)_{i,N+1} = \Psi_f(\theta_i, r_b) + \Psi_H(\theta_i, r_b) \qquad i = 1 \ldots M \quad (17)$$

$$D_r(\Psi_g)_{i,N+1} = \frac{\partial \Psi_f}{\partial r}(\theta_i, r_b) + \frac{\partial \Psi_H(\theta_i, r_b)}{\partial r} \qquad i = 1 \ldots M. \quad (18)$$

Here Δ^h is either a second or fourth order finite difference approximation to Δ, such as that described by (10) in Section 2.2. D_r is either a second or fourth order one sided difference approximation to the radial derivative.

The first two equations, (15) and (16), are finite difference approximations of (13) and (14) in the annulus $r_a \le r \le r_b$. For $r \ge r_b$ we need not require that $\Delta\Psi_f = 0$ since this is automatically satisfied (the boundary conditions at infinity are also satisfied). The condition (17) is a requirement that the approximate solution be continuous at $r = r_b$ and (18) is a requirement that the normal derivative of the approximate solution be continuous at $r = r_b$.

The solution to equations (15) and (16) is completely determined given any specification of the boundary values of Ψ_g and Ψ_f at the outer computational boundary. However, these values cannot be chosen arbitrarily, since they must be chosen in such a way that Eq. (18) is satisfied. In the Appendix we show how these considerations lead to a set of linear equations for the desired boundary values, and give an efficient procedure for solving these equations which utilizes the discrete Fourier Transform.

3.4. Vorticity Boundary Conditions

One of the principal difficulties which is encountered when using the vorticity formulation of the Navier–Stokes equations is the determination of the vorticity at solid boundaries in the domain. This difficulty arises because when one transforms the velocity–pressure formulation into the vorticity form, one does not obtain an explicit expression for the vorticity on the boundary. However, it is clear that the vorticity values on the boundary should be determined so that as the interior vorticity evolves using these values, the induced velocity field will satisfy the appropriate velocity boundary conditions. The main difficulty is how to achieve this goal for a particular discretization and boundary configuration. In our approach we find the set of equations for the boundary vorticity which lead to a scheme with the desired properties. This system of equations is linear and can be interpreted as the discretization of an integral equation for the vorticity boundary values—see [3]. In this section we will show how the system of equations is constructed and the solutions are obtained. The resulting method evolves the vorticity in such a way that at every time step the induced velocity field satisfies the velocity boundary conditions to the order of accuracy of the time-discretization procedure. The method we construct does not require that an implicit time-stepping scheme be used to evolve the interior vorticity.

There are other families of vorticity boundary conditions which allow one to determine boundary vorticity. In the finite difference context, perhaps the most well known are those given by Thom [26]. (See Peyret and Taylor [17] for references and a general discussion of the technique and its refinements.) This type of boundary condition has recently been analyzed by Hou and Wetton [12] and shown to yield second order accurate solutions. Since we are determining converged solutions we expect that either technique (i.e., the method presented here, or that based on Thom's approach) would work well. An advantage of the approach presented here is that it allows one to easily change the order of spatial accuracy and incorporate time-dependent boundary conditions.

Consider the ordinary differential equations which arise after discretization in space,

$$\frac{d\omega_{i,j}}{dt} = -\left(u_{i,j}, \frac{v_{i,j}}{r}\right)\cdot(D_r^0\omega_{i,j}, D_\theta^0\omega_{i,j}) + \frac{1}{\mathrm{Re}}\left(D_r^+D_r^- + \frac{1}{r}D_r^0 + \frac{1}{r^2}D_\theta^+D_\theta^-\right)\omega_{i,j}, \quad (19)$$
$$i = 1 \ldots M \quad j = 2 \ldots N.$$

There is one differential equation for each of the interior values of vorticity but there are no differential equations for the boundary vorticity (when $j = 1$). However, the boundary vorticity values are included in the set of equations (19) since they enter into the difference stencils occurring on the right hand side. We can write the system of differential equations (19) in the form

$$\frac{d\boldsymbol{\omega}_{\mathrm{int}}}{dt} = L(\boldsymbol{\omega}_{\mathrm{int}}) + L(\boldsymbol{\omega}_{\mathrm{bdry}}). \quad (20)$$

Here $\boldsymbol{\omega}_{\mathrm{int}}$ represents the interior values of vorticity and $\boldsymbol{\omega}_{\mathrm{bdry}}$ the boundary values of vorticity. L is the operator

$$L(\cdot) = -\left(u_{i,j}, \frac{v_{i,j}}{r}\right)\cdot(D_\theta^0[\cdot], D_\theta^0[\cdot]) + \frac{1}{\mathrm{Re}}\left(D_r^+D_r^- + \frac{1}{r}D_r^0 + \frac{1}{r^2}D_\theta^+D_\theta^-\right)[\cdot].$$

Due to the presence of $\boldsymbol{\omega}_{\mathrm{bdry}}$ in (20) these equations are underdetermined. We use the velocity boundary conditions to remove this indeterminacy.

Let G be the operator which computes the stream function from the vorticity distribution (i.e., the operator $\Psi = G[\boldsymbol{\omega}_{\mathrm{int}}]$ which is defined by the procedure described in Section (2.3)). Since the construction of the stream function utilizes boundary condition (8) the computed velocity field automatically satisfies the normal velocity boundary condition (5). What is not automatically satisfied is the tangential velocity boundary condition (6), or, when it is expressed in terms of the stream function (7),

$$\frac{\partial\Psi}{\partial r}(r, \theta) = -b(t)v_\infty(r, \theta) \quad r = r_\alpha\, 0 \le \theta \le 2\pi.$$

If we approximate the derivative in this boundary condition by a finite difference approximation, D_r, and employ the operator G, we obtain the following condition on the vorticity which must be satisfied:

$$(D_r\Psi^h)_{i,1} = (D_rG[\boldsymbol{\omega}_{\mathrm{int}}])_{i,1} = -b(t)v_\infty(r_a, \theta_i) \quad i = 1 \ldots M. \quad (21)$$

The computational task is to determine the boundary velocity, $\boldsymbol{\omega}_{\mathrm{bdry}}$, so that when one solves the ordinary differential equations (20) for the evolution of $\boldsymbol{\omega}_{\mathrm{int}}$ the constraint imposed by the second velocity boundary condition (21) is satisfied.

To guarantee that the constraint (21) is satisfied, we assume that it is satisfied at $t = 0$, i.e.,

$$(D_rG[\boldsymbol{\omega}_{\mathrm{int}}])_{i,1} = -b(0)v_\infty(r_a, \theta_i) \quad i = 1 \ldots M, \quad (22)$$

and require that the time-derivative of the constraint vanish:

$$\frac{d}{dt}(D_rG[\boldsymbol{\omega}_{\mathrm{int}}])_{i,1} = -b'(t)v_\infty(r_a, \theta_i) \quad i = 1 \ldots M. \quad (23)$$

We now use the condition (23) along with (20) to eliminate the boundary vorticity in Eqs. (20); i.e., we can explicitly determine the boundary vorticity. We have

$$\frac{d}{dt}(D_rG[\boldsymbol{\omega}_{\mathrm{int}}])_{i,1} = -b'(t)v_\infty(r_a, \theta_i)$$
$$\Rightarrow \quad \left(D_rG\left[\frac{d\boldsymbol{\omega}_{\mathrm{int}}}{dt}\right]\right)_{i,1} = -b'(t)v_\infty(r_a, \theta_i)$$
$$\Rightarrow (D_rG[L(\boldsymbol{\omega}_{\mathrm{int}}) + L(\boldsymbol{\omega}_{\mathrm{bdry}})])_{i,1} = -b'(t)v_\infty(r_a, \theta_i)$$
$$\Rightarrow \quad (D_rG[L(\boldsymbol{\omega}_{\mathrm{bdry}})])_{i,1} = -(D_rG[L(\boldsymbol{\omega}_{\mathrm{int}})])_{i,1} - b'(t)v_\infty(r_a, \theta_i).$$

Thus we have an equation which determines the boundary vorticity as a function of the interior vorticity:

$$(D_rG[L(\boldsymbol{\omega}_{\mathrm{bdry}})])_{i,1} = -(D_rG[L(\boldsymbol{\omega}_{\mathrm{int}})])_{i,1} - b'(t)v_\infty(r_a, \theta_i). \quad (24)$$

If we express the relationship defined by (24) as $\boldsymbol{\omega}_{\mathrm{bdry}} = H(\boldsymbol{\omega}_{\mathrm{int}}, t)$ then the ordinary differential equations (20) become

$$\frac{d\boldsymbol{\omega}_{\mathrm{int}}}{dt} = L(\boldsymbol{\omega}_{\mathrm{int}}) + L(H(\boldsymbol{\omega}_{\mathrm{int}}, t)), \quad (25)$$

i.e., a closed system of ordinary differential equations for the evolution of the interior vorticity. In the next section we will discuss our time stepping method for (25), but before doing this we discuss how, for given values of $\boldsymbol{\omega}_{\mathrm{int}}$, the operator $L(H(\boldsymbol{\omega}_{\mathrm{int}}, t))$ is evaluated.

$H(\boldsymbol{\omega}_{\mathrm{int}}, t)$ represents the determination of the boundary vorticity obtained as a solution of (24). To compute the right hand side of (24) we assume that $b'(t)$ is known so that the term involving that factor can be computed

explicitly. For the other term, we first observe that $L(\boldsymbol{\omega}_{\text{int}})$ are the values resulting from the application of the convection–diffusion difference operators to $\boldsymbol{\omega}_{\text{int}}$ using homogeneous boundary data and that the operator $D_r G$ represents the computation of the tangential velocity induced by these values. Thus, this term can be computed by applying the difference operator to $\boldsymbol{\omega}_{\text{int}}$, solving a Poisson problem, and then differentiating the result at the boundary points.

The operator on the left hand side of (24) is a linear operator from the boundary points to themselves. Thus, for any specific discretization the equation (24) can be represented by an $m \times m$ matrix equation, $B\boldsymbol{\omega}_{\text{int}} = \mathbf{f}$. First consider the form of B when second order differencing is used in the interior. The operator $L(\boldsymbol{\omega}_{\text{bdry}})$ takes the boundary vorticity values, scales them by a certain factor, and then projects these values into the interior of the domain by moving them one mesh point in (to $j = 2$ points). The operator represented by $D_r G[\,\cdot\,]$ computes the tangential velocity which is induced by a given amount of vorticity in the interior. Thus, B is the matrix which represents the computation of the tangential velocity induced by a given amount of boundary vorticity which has been scaled and placed one mesh point in the interior. Because of the symmetry inherent in the problem, the tangential velocity induced by vorticity one point inside the domain is independent of its location along the cylinder. For this reason, the matrix representing the computation of the velocity from such vorticity is circulant. The matrix B therefore takes the form $B = C * D$, where C is circulant and D is diagonal. The inversion of B is therefore a simple process. The equation $C\mathbf{x} = \mathbf{f}$ is first solved using the discrete Fourier transform (in a manner entirely analogous to the procedure described in the appendix for the stream function computation) and then $\boldsymbol{\omega}_{\text{bdry}} = D^{-1}\mathbf{x}$. To carry out this process one must determine the first column of C—but this is just the tangential velocity values induced by a unit value of vorticity at the location $i = 1$ and $j = 2$. Furthermore, the matrix C does not vary with time, and so its construction need only be done at the start of the computation.

When fourth order difference stencils are used, the problem becomes slightly more complex. Because the difference stencils are wider, boundary vorticity is convected and diffused onto both the first and second rows from the boundary. We can split our boundary operator into two parts: one representing the slip generated by vorticity moved from the boundary to the first interior row, and one representing slip induced by vorticity moved two rows in. Each part of the operator behaves in a similar way to the second order operator and can be represented by a circulant matrix multiplied by a diagonal matrix; that is,

$$B = C_1 * D_1 + C_2 * D_2. \tag{26}$$

While each of the two parts of B are easily invertible using the discrete Fourier Transform, there is no way to directly invert their sum. However, this sum obviously suggests a splitting for use with an iterative method [10]. We can write $B = B_1 + B_2$, where $B_1 = C_1 * D_1$ and $B_2 = C_2 * D_2$. This splitting is then used as a preconditioner to invert B. B is non-symmetric, so GMRES [21] is used as an iterative method. We find convergence to be quite rapid. The number of iterations is usually no more than 10, and the work per iteration is only $\mathcal{O}(M \log M)$, so the total work to solve for the boundary vorticity is small compared to the work spent in other parts of our procedure.

(The discretized version of the boundary vorticity operator H has a simpler form if the term for convection of vorticity is discretized in its conservative form $\nabla \cdot (\mathbf{u}\omega)$. In this case the matrix B is circulant for both the second and fourth order discretizations, and can be inverted directly with the discrete Fourier transform.)

3.5. Time-Stepping Strategy

The last item to be discussed is the choice of time-differencing. We have discretized in space and are thus confronted with the problem of solving a large system of coupled ordinary differential equations for the interior vorticity of the form

$$\frac{d\boldsymbol{\omega}_{\text{int}}}{dt} = L(\boldsymbol{\omega}_{\text{int}}) + L(H(\boldsymbol{\omega}_{\text{int}}, t))$$

When the fourth order time differencing is employed, the resulting time-stepping procedure is

$$\begin{aligned}
\mathbf{K}_1 &= L(\boldsymbol{\omega}_{\text{int}}^n) + L(H(\boldsymbol{\omega}_{\text{int}}^n, t^n)) \\
\boldsymbol{\omega}_{\text{int}}^{(\text{i})} &= \boldsymbol{\omega}_{\text{int}}^n + \frac{\delta t}{2}\mathbf{K}_1 \\
\mathbf{K}_2 &= L(\boldsymbol{\omega}_{\text{int}}^{(\text{i})}) + L\left(H\left(\boldsymbol{\omega}_{\text{int}}^{(\text{i})}, t^n + \frac{\delta t}{2}\right)\right) \\
\boldsymbol{\omega}_{\text{int}}^{(\text{ii})} &= \boldsymbol{\omega}_{\text{int}}^n + \frac{\delta t}{2}\mathbf{K}_2 \\
\mathbf{K}_3 &= L(\boldsymbol{\omega}_{\text{int}}^{(\text{ii})}) + L\left(H\left(\boldsymbol{\omega}_{\text{int}}^{(\text{ii})}, t^n + \frac{\delta t}{2}\right)\right) \\
\boldsymbol{\omega}_{\text{int}}^{(\text{iii})} &= \boldsymbol{\omega}_{\text{int}}^n + \delta t\,\mathbf{K}_3 \\
\mathbf{K}_4 &= L(\boldsymbol{\omega}_{\text{int}}^{(\text{iii})}) + L(H(\boldsymbol{\omega}_{\text{int}}^{(\text{iii})}, t^n + \delta t)) \\
\boldsymbol{\omega}_{\text{int}}^{n+1} &= \boldsymbol{\omega}_{\text{int}}^n + \delta t\left[\tfrac{1}{6}\mathbf{K}_1 + \tfrac{1}{3}\mathbf{K}_2 + \tfrac{1}{3}\mathbf{K}_3 + \tfrac{1}{6}\mathbf{K}_4\right].
\end{aligned}$$

For a given state $\boldsymbol{\omega}_{\text{int}}^{(*)}$ and time $t^n + \alpha\,\delta t$ the computation of the terms of the form $L(\boldsymbol{\omega}_{\text{int}}^{(*)}) + L(H(\boldsymbol{\omega}_{\text{int}}^{(*)}, t^n + \alpha\,\delta t))$

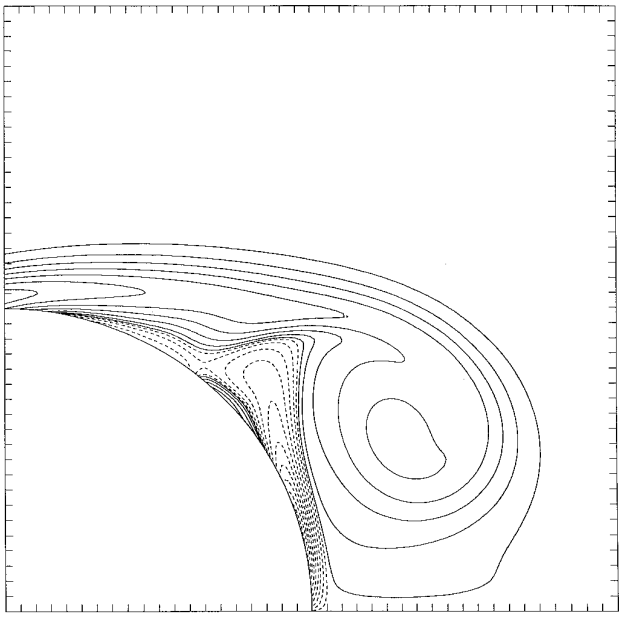

FIG. 1. Contours of vorticity for Re 1000, $t = 0.75$–1.5.

is carried out in the manner described in the previous section.

4. ALGORITHM TESTING

Before presenting the details of the solutions obtained with our method we first discuss the results of accuracy checks. In particular we were interested in determining the rates of convergence of the computed quantities, the effect which changing the outer radius of the computational domain had on the computed solution, and the effect of using an impulsive versus a finite acceleration startup procedure.

An example of our results is presented in Fig. 1. In this figure the contours of the vorticity at time 1.5 for Reynolds number 1000 are shown. The presence of large vorticity gradients near and on the surface of the cylinder are clearly seen. This aspect of the solution was presented in solutions at higher Reynolds numbers, and we were therefore interested in determining the accuracy of our vorticity on the boundary as well as in the interior. Since exact analytic solutions to the Navier–Stokes equations of time dependent flow past a cylinder are not available, we used fully converged results obtained with the fourth order method as the ''exact'' solutions. The norms used to measure the computed quantities were discrete L^2 norms:

$$\|g\|_{\text{int}} = \left(\sum_i \sum_j g_{i,j}^2 r_j \, \delta r \, \delta\theta \right)^{1/2}$$

for values $g_{i,j}$ at interior points and

$$\|g\|_{\text{boundary}} = \left(\sum_i g_{i,1}^2 r_a \, \delta\theta \right)^{1/2}$$

for values on the cylinder boundary.

For our discretization error estimates we used as a test problem the results at Reynolds number 1000. This Reynolds number was high enough so the flow structures present were quite detailed, yet still low enough to allow for full resolution on reasonably sized grids. For this Reynolds number a grid of dimensions 256 radial points $\times$ 1024 points around the cylinder was more than sufficient for full resolution. An outer computational radius of 1.0, twice that of the cylinder, was sufficient to capture the vorticity field up to time $t = 1.5$.

We considered three implementations of the method. The most accurate uses fourth order approximations in space for all of the operators. The second method uses the simpler second order approximations. We also tested a hybrid method, in which the stream function and velocity fields, including the slip on the boundary, were computed using fourth order operators while the convection and diffusion operators were discretized to second order. We conjectured that most of the errors in a complete second order computation would be due to the inaccurate calculation of vorticity boundary values. To test this conjecture we kept the convection and diffusion operators at second order and computed the boundary vorticity using a fourth order accurate approach. All of the methods used a fourth order Runge–Kutta time discretization.

Table I shows the relative errors at $t = 1.5$ as the method and grid size vary. The fourth order method does appear to be worth the extra work. Relative errors of less than 1% are attained with a 64×256 grid, one size smaller than is needed for the second order methods. The computation times for the fourth order method were slightly higher than

TABLE I

L-2 Norm of the Relative Error of Vorticity for Re 1000 Flows

Scheme	Grid size	Error in interior vorticity	Error in boundary vorticity
Fourth order	32×128	0.068	0.116
	64×256	0.0054	0.0093
	128×512	0.00048	0.00039
Second order	32×128	0.244	0.237
	64×256	0.071	0.071
	128×512	0.019	0.019
Hybrid method	32×128	0.204	0.145
	64×256	0.041	0.025
	128×512	0.0099	0.0060

TABLE II

The L-2 Norm of the Relative Error of the Stream Function for Re 1000 Flows

Scheme	Grid size	Error in stream function
Fourth order	32×128	0.015
	64×256	0.00086
	128×512	0.00011
Second order	32×128	0.063
	64×256	0.017
	128×512	0.0044
Hybrid method	32×128	0.059
	64×256	0.011
	128×512	0.0028

the second order method, but the difference is less than 20% of the run time.

The hybrid method also presents a significant improvement over the second order method, especially for the values of boundary vorticity. The hybrid method could be quite useful for flows in complicated domains or for three dimensional flows—problems for which the construction of fourth order convection and diffusion approximations is difficult.

We also considered the relative errors of the calculated stream function. In general, at a given resolution, the errors for the stream function were much smaller than the errors for the vorticity. This is to be expected since the stream function is related to the vorticity via the inverse of the Laplace operator. This operator is a smoothing operator—i.e., the operator suppresses small scale features. Moreover, we can infer that if one estimates the accuracy of the computation by the error in the stream function, then this estimate is not a good estimate for the errors in the vorticity. For example, with the second order method, a 64×256 grid is sufficient to obtain an error of less than 2% for the stream function, while a 128×512 grid is necessary for the same level of accuracy for the vorticity.

The effect of the placement of the outer computational boundary was tested by computing two fully resolved flows at Re 1000. The first of these calculations was performed with the outer boundary at $r = 1.0$, and the second with the outer boundary at $r = 1.5$. We calculated the difference between the two flows by calculating the relative L-2 norm of their difference.

The length of the wake of a flow is defined as the furthest distance along the centerline behind the cylinder in which the velocity is negative. It measures the size of the region in which there is backflow, and can be used as a measure of how far away from the cylinder the underlying potential flow has been altered by the vorticity produced. As vorticity moves further away from the cylinder, the wake length increases. Table III shows that the location of the outer boundary has a negligible influence on the computation until the wake length approaches the outer boundary. Even as the wake approaches the edge of the computational domain, errors are still small in magnitude, and they only become large if a significant amount of vorticity is exited from the domain.

TABLE III

The Relative L-2 Norm of Difference between Flow Calculated with Outer Radius 1.0 and Flow Calculated with Outer Radius 1.5

Time	L-2 norm of difference	Wake length
0.50	1.0×10^{-9}	0.09
1.00	1.6×10^{-6}	0.29
1.50	2.7×10^{-4}	0.47

Many of the computations performed on a circular cylinder involve impulsively starting the cylinder in motion. The impulsive start process should have an effect on the accuracy of the solution, but the precise nature of the effect is unclear. To investigate this we performed a set of computations in which we implemented an impulsive start calculation using three different procedures. Each of these procedures converged to the same solution as the time step was refined. The procedures differed in the rate at which they converged to this solution.

The first procedure involved solving the equations with $b(t)$ being a step function—that is, using $b(t) = 0$ for $t < 0$ and $b(t) = 1$ for $t \geq 0$. The resulting solution exhibited only first order convergence in time despite the use of a higher order time-stepping method. This is shown in Table IV, in which the errors in time are examined by looking at the relative error between solutions computed with time steps of dt and an "exact" solution computed with an excessively small time step. These differences decrease linearly as dt decreases.

The failure to attain higher order convergence results from the fact that higher order convergence is attained only if the vorticity field at the start of each step satisfies the no-slip condition. This does not hold at $t = 0$, and first order convergence results. This can be corrected in two

TABLE IV

The Relative L-2 Error in Time as Time Step Varies for Impulsive Start Calculation with No Initial Condition Modification

dt	L-2 norm of error
0.004	2.44×10^{-3}
0.002	1.22×10^{-3}
0.001	6.09×10^{-4}

Note. The error is computed at time $t = 0.1$.

TABLE V

L-2 Norm of Difference between Impulsively Started Flow and Flow with Smooth Startup

Startup time	L-2 norm of difference
0.05	1.05×10^{-1}
0.025	4.89×10^{-2}
0.0125	2.36×10^{-2}
0.00625	1.16×10^{-2}

Note. Differences are computed at time $t = 0.1$.

ways. One technique is to do a smooth startup where the velocity varies continuously from zero at $t = 0$ to unity at some startup time t_{st}. Since the velocity at the initial time is zero, the no-slip condition is satisfied at startup, and fourth order convergence in time is attained. However, the method only converges with first order accuracy with respect to the length of the startup interval. Table V shows that as t_{st} is reduced to 0, (along with dt), the solutions converge linearly to the impulsive start solutions.

Another approach is to prepare the initial data to satisfy the no-slip condition at startup. This consists of constructing a vorticity distribution in the fluid so that the velocity field at $t = 0$ satisfies the no-slip condition. The particular initial vorticity distribution we used was that which occurs when one performs one step of Euler's method with a step size ε, and then considers the vorticity in the limit as $\varepsilon \to 0$. As seen in Table VI, when this initial condition is used, fourth order convergence in time results.

In our opinion, the procedure in which one modifies the initial vorticity works best. It requires little extra work, and does not destroy the time accuracy of the computed solution. The technique can be useful in situations where the prescribed velocity varies discontinuously over time. When one encounters a discontinuity of the prescribed velocity, one adds to the vorticity distribution an amount of vorticity necessary so that the tangential velocity condition is satisfied at the beginning of the next time step. There is no need to smooth out the transitions.

TABLE VI

The Relative L-2 Error in Time as Time Step Varies for Impulsive Start Calculation with Initial Condition Modification

dt	L-2 norm of error
0.004	2.5×10^{-7}
0.002	1.4×10^{-8}
0.001	9.0×10^{-10}

Note. The error is computed at time $t = 0.1$.

TABLE VII

Computational Parameters for Flow Past a Cylinder

Re	n_r	n_θ	dt	Outer radius
1000	128	512	0.001	1.0
3000	192	1024	0.0005	0.875
9500	256	2048	0.00033	0.75

5. NUMERICAL RESULTS

In this section we present the results of solutions for flow past a cylinder of radius 0.5 over a range of Reynolds numbers. All of these flows are computed to full resolution. Our solutions were computed using an impulsive start with a modified vorticity initial condition to preserve time accuracy. Table VII presents the relevant numerical parameters for each of the runs. (For more detailed results and computations at intermediate Reynolds numbers see [19].)

In Figs. 2–4 we present plots of the contours of vorticity. Negative contours are shown as solid lines while positive contours are dashed lines. We will show only the contours for the upper right quarter of the cylinder. By symmetry the upper and lower halves of the plot will be identical (except for a sign change). The front half of the cylinder is not shown since there is little vorticity present and the flow is quite uninteresting in that region. (There is a vortex sheet near the surface, but there is no separation or other

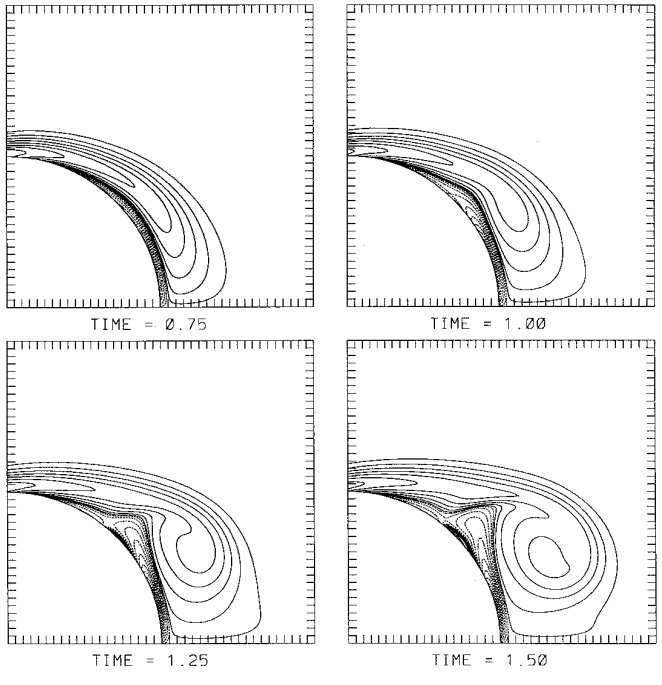

FIG. 2. Contours of vorticity for Re 1000, $t = 0.75$–1.5.

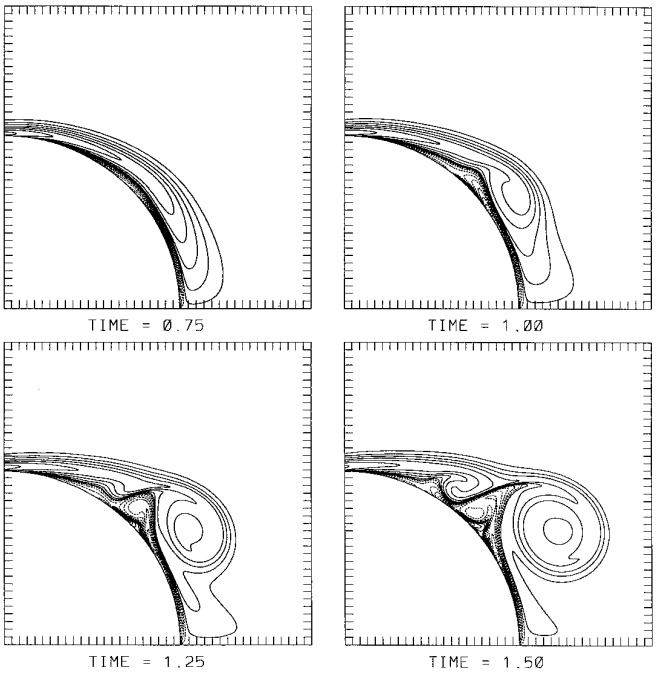

FIG. 3. Contours of vorticity for Re 3000, $t = 0.75$–1.5.

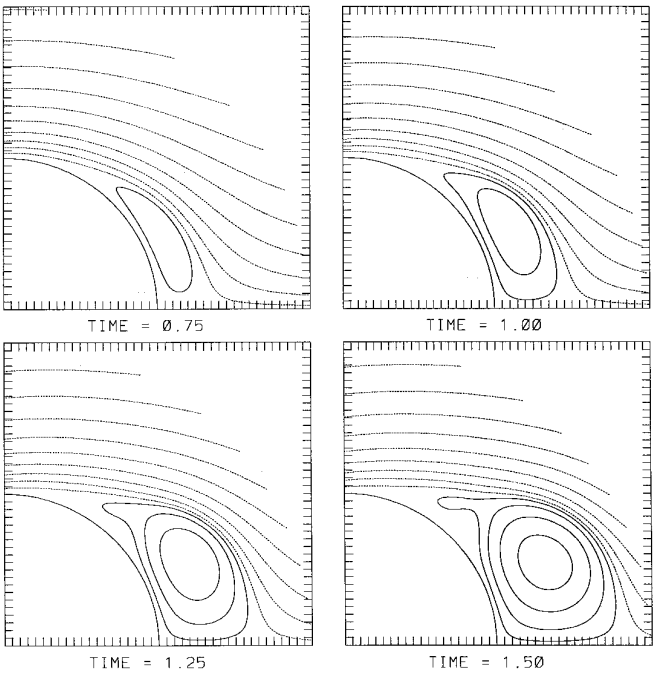

FIG. 5. Contours of stream function for Re 1000, $t = 0.75$–1.5.

significant phenomena.) The stream function for these cases is shown in Figs. 5–7. Graphs of boundary vorticity and surface pressure are given in Figs. 8 and 9. The time development of the coefficient of drag and the maximum vorticity are shown in Fig. 10. The drag is computed by evaluating along the surface of the cylinder the integral

$$D = \frac{1}{\mathrm{Re}} \int_0^{2\pi} \left(\omega - \frac{\partial \omega}{\partial n} \right) \sin\theta \, d\theta \tag{27}$$

and the drag coefficient is then found using the relationship

$$C_d = \frac{D}{2\rho U_\infty^2}. \tag{28}$$

The pressure at a point along the cylinder is given by

$$P(\theta) = \frac{1}{\mathrm{Re}} r_a \int_0^{\theta} \frac{\partial \omega}{\partial n} \, d\phi. \tag{29}$$

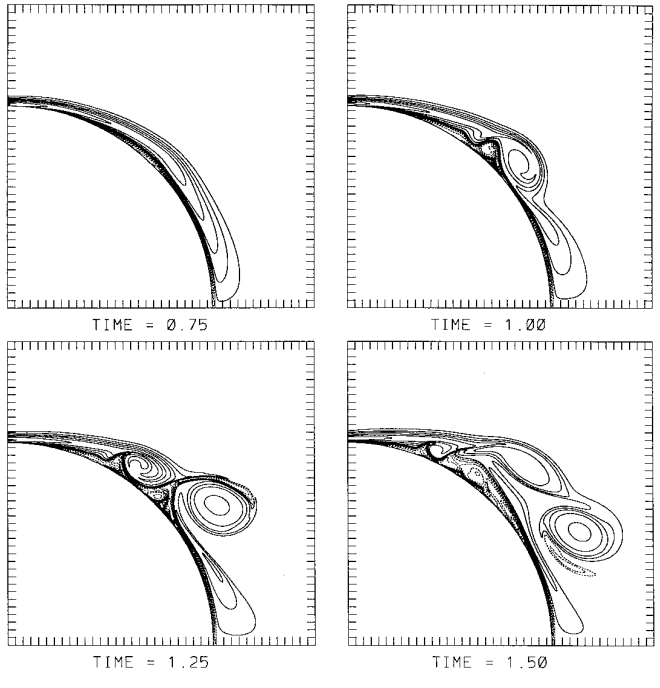

FIG. 4. Contours of vorticity for Re 9500, $t = 0.75$–1.5.

As the Reynolds number increases, a number of trends occur. The flows become more and more complicated—the decrease in diffusion allowing smaller scale features to remain. There is an increased variation in the surface pressure and boundary vorticity as one moves along the back of the cylinder. These quantities become more and more oscillatory with increased Re. The wake size decreases and the separation and the roll up of the main vortex occur earlier in time as the Reynolds number increases.

Secondary structures occur at Re 1000 (Fig. 1). At $t = 1.5$ we see that the counter-rotating flow at the rear of the cylinder has separated significantly, being pushed up by a

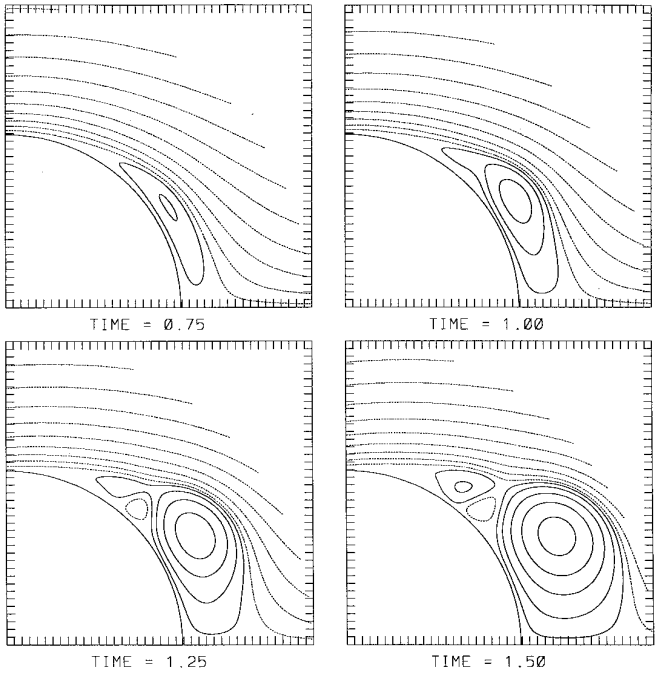

FIG. 6. Contours of stream function for Re 3000, $t = 0.75$–1.5.

second mass of positive vorticity along the cylinder. At Re 3000 (Fig. 3) a cascade of vortices of opposite signs begins to appear. This cascade forms when a mass vorticity of a given sign close to the surface separates. The separation causes vorticity of the opposite sign to form beneath it. These two vortices grow (being fed from the boundary layer vorticity) and in time, the second vortex separates, which induces the occurrence of a third vortex near the surface. This process generates structures of smaller and smaller scale near the surface. It appears that the only limit to the formation of an arbitrary large number of structures is the viscosity, which limits the size of structures which can occur. At Re 3000, we see that this process occurs about four times—i.e., we see the beginning of the development of four vortices of opposite sign. Each of these events is manifested as a change of sign of the boundary vorticity, as illustrated in Fig. 8.

The occurrence of strong vortices near the cylinder surface is reflected in the pressure and boundary vorticity. In particular, as seen in Figs. 8 and 9, these vortical structures induce large spatial fluctuations in the pressure and vorticity. Moreover, they appear to be responsible for the unexpected time dependence of the maximum of vorticity. At startup, a vortex sheet is formed, and the corresponding vorticity is of large magnitude. As expected, as time advances, the maximum of vorticity decays quite rapidly. For lower Reynolds numbers, the decay continues steadily as shedding occurs. However, at higher Reynolds numbers, the steady decay does not continue; this appears to be due to the strength of the secondary vortices that are formed.

A similar time dependent behavior is seen when the coefficient of drag is examined (Fig. 10) over our time interval of computation. The coefficient of drag is very large at the start of the computation and initially decreases with time. For very low Reynolds numbers the drag coefficient continues to decrease as it rapidly tends towards a constant. For higher Reynolds numbers, there is an initial dip in the drag coefficient, followed by an increase that again quickly approaches a constant. This behavior was seen for Reynolds numbers in the range 300–3000. However, a much more complicated behavior is seen for Re above 5000, and the drag coefficient varies greatly in time during the early shedding process. Again, this appears to be due to the formation of the strong secondary vortices which occur as the first vortex is shed.

The rate at which the vortices form also increases with Reynolds number. For Re 1000, by time $t = 1.5$ the starting vortex has just rolled up, while at Re 3000 the main vortex is well formed at $t = 1.0$, and by $t = 1.5$ a second vortex is well into the process of rolling up. The shedding process for Re 9500 occurs even more rapidly. At $t = 1.25$ the first vortex is shed and the second vortex is fully rolled up behind it. By $t = 1.5$ the second vortex is shed, and the third vortex is beginning to roll up.

We present the stream function contours of our results to facilitate comparison with other computations. However, as is clear from the figures, the complicated vortex dynamics which are occurring in the boundary layer are not reflected in the stream function.

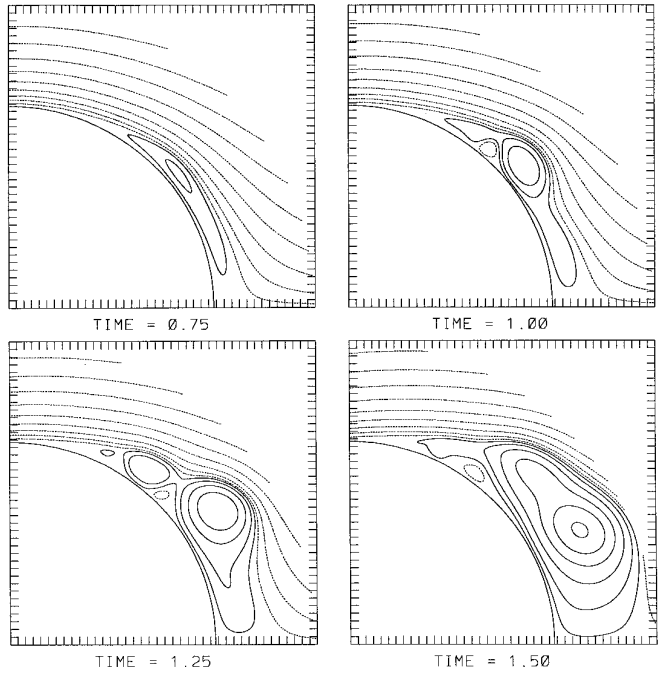

FIG. 7. Contours of stream function for Re 9500, $t = 0.75$–1.5.

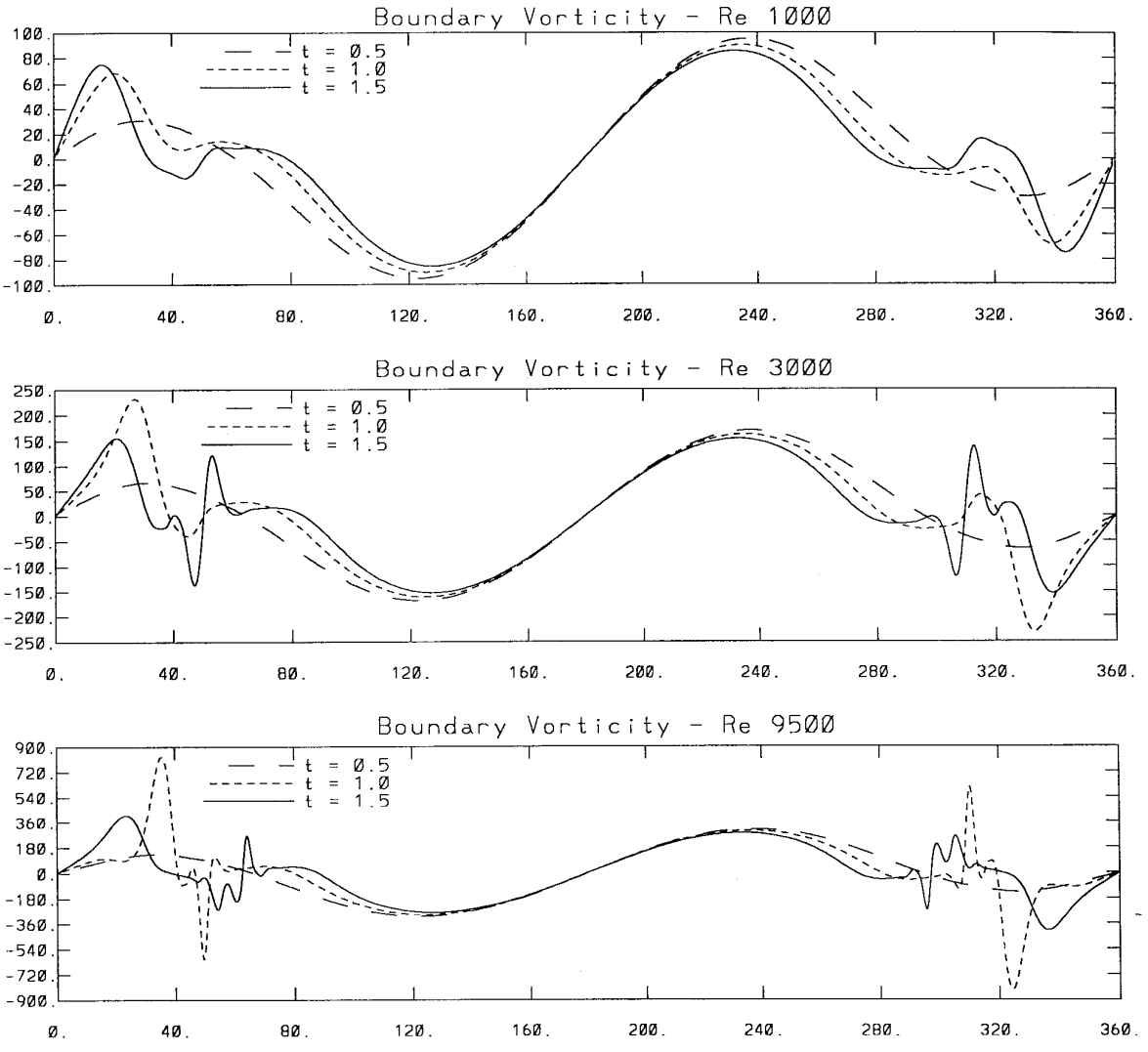

FIG. 8. Boundary vorticity along the cylinder surface. 0° is on the trailing side of the cylinder while 180° is at the front.

6. DISCUSSION AND CONCLUSIONS

In this paper we have presented an explicit finite difference method for computing the time-dependent solution of an impulsively started cylinder in two dimensions. We have reported the results obtained from this code at Reynolds numbers 1000, 3000, and 9500. These results have been obtained by solving the Navier–Stokes equations with boundary conditions appropriate for a circular cylinder in an infinite domain. The solutions presented are the results of fully resolved computations.

Our solutions clearly indicate that for this problem, where boundary layer separation is present, one must have nearly equal resolution in both the direction normal and tangential to the cylinder. From Table VII we estimate that the mesh width, Δ, necessary to resolve the flow is on the order of 1/Re. Since the magnitude of the vorticity scales as $\sqrt{\mathrm{Re}}$ (which can be inferred from the computation or from a boundary layer analysis), one finds that this mesh width is consistent with the scale estimates of Kreiss *et al.* which can be stated as

$$\Delta \approx \frac{\max|\omega|}{\sqrt{\mathrm{Re}}}.$$

The estimates in [11] are derived for periodic flows, but their agreement with our computational results indicates that they may hold more generally. The smallest necessary mesh width we have found is based on our computations using the vorticity formulation of the Navier–Stokes equations. By looking at the scale of the structures which occur in the solutions, it is clear that our estimate is a reasonable one. However, the velocity field associated with our solu-

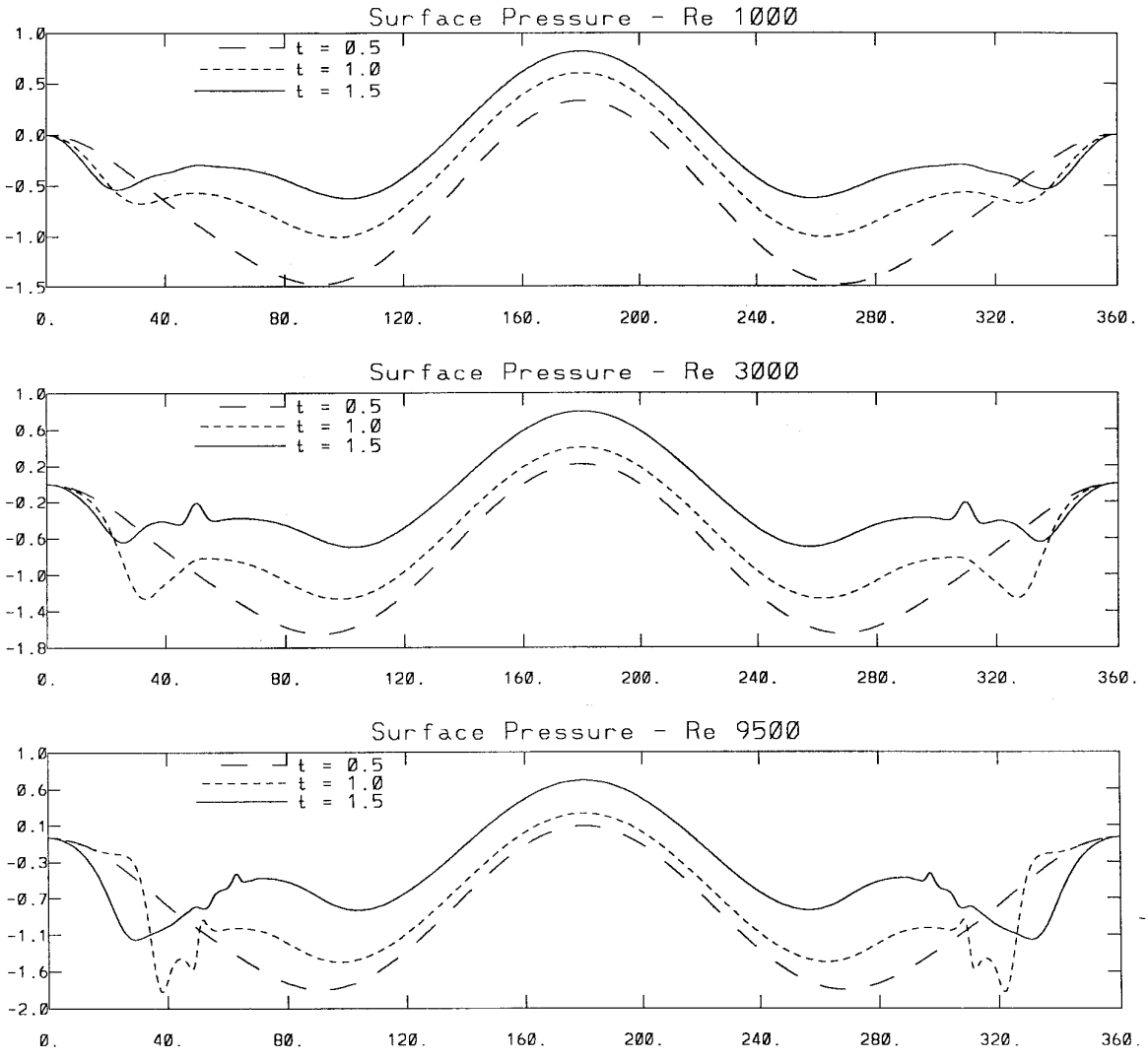

FIG. 9. Surface pressure on the cylinder. 0° is on the trailing side of the cylinder while 180° is at the front.

tions is much smoother than the vorticity distribution, and so a question remains as to whether or not our reported minimal mesh size is also the minimal mesh size needed for a computation done using the velocity/pressure formulation.

We were also interested in understanding the effect of using an impulsive start on the computation. The primary result is that the numerical method using an impulsive start converges to a single solution as the time-step tends to zero, but that the rate of convergence to the solution is only first order in time—rather than fourth order accuracy, which might be expected from the use of a fourth order Runge–Kutta time-stepping method. We demonstrated that the loss of accuracy could be circumvented by using a smooth startup procedure or by using suitably modified initial conditions.

By choosing different approximations we were able to construct a second order method, a fourth order method and a hybrid second/fourth order method. We found that in terms of computational efficiency using a fourth order approach was justified. The computational time necessary for the fourth order method was only about 20% more than that for the second order method. If one does not want to go to the trouble of implementing a fully fourth order method, then one can obtain substantial improvements using a hybrid method where only the stream function and the vorticity boundary conditions are computed using fourth order approximations.

The numerical approach presented here differs from other finite difference implementations primarily in our use of ''infinite'' boundary conditions for the stream function and our particular method for satisfying vorticity

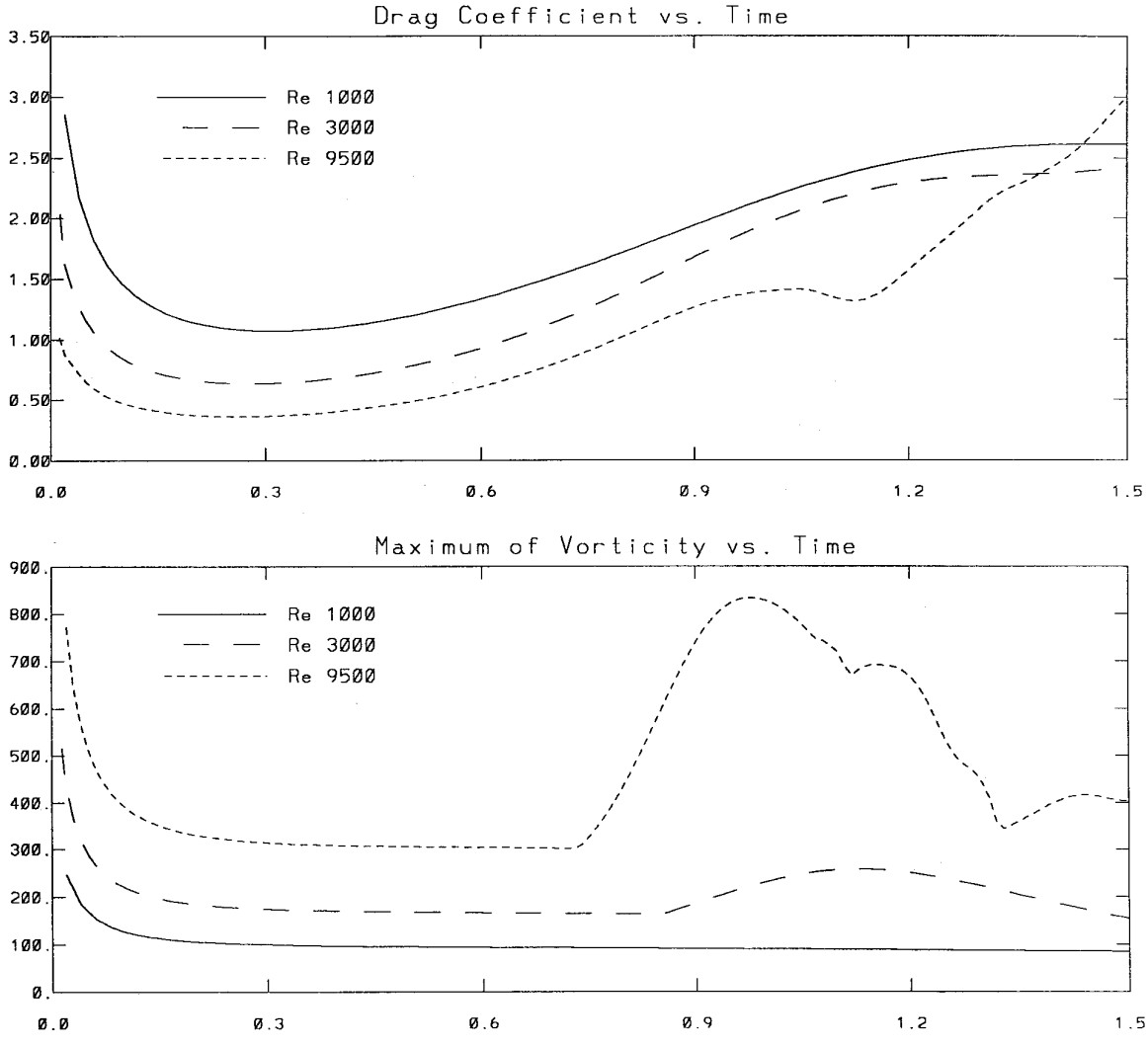

FIG. 10. Time evolution of drag coefficient and maximum value of vorticity.

boundary conditions. In both instances our procedures can be seen to result from a desire to have methods which give solutions which respect the mathematical properties of the solution to the Navier–Stokes equations. Specifically, the stream function used on the computational domain should be the restriction of the stream function associated with the solution of Laplace's equation on the infinite domain. The procedure we use provides either a second order or fourth order approximation to a stream function with this property. Furthermore, the vorticity of the computed solution should evolve in such a way that the velocity boundary conditions are satisfied. Our vorticity boundary conditions ensure that this constraint is satisfied up to the accuracy of the time-stepping method used. The fact that our computation is for a circular cylinder allows us to create efficient implementations of these techniques. Solutions for flows about more general geometries have been obtained using this approach by Reider [19]. (Conformal mapping was used.) The direct application of our procedures for general geometries is also under investigation, as well as the extension of these procedures to three dimensional computations.

APPENDIX: THE TREATMENT OF THE INFINITE DOMAIN

In this appendix we fully describe the steps necessary to solve for a stream function that satisfies the appropriate boundary conditions at infinity.

The stream function satisfies the equation

$$\Delta\Psi = \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial\Psi}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2\Psi}{\partial\theta^2} = -\omega \tag{30}$$

with boundary conditions

$$\Psi(r,\theta)=0 \quad r=r_a; \quad \Psi(r,\theta)=\Psi_\infty(r,\theta) \quad r\to\infty \qquad 0\le\theta\le 2\pi. \tag{31}$$

We seek an approximation of Ψ as a composition of functions with the form

$$\Psi\approx\begin{cases}(\Psi_g)_{i,j} & 0\le\theta_i\le 2\pi \quad r_a\le r_j\le r_b\\ \Psi_f(r,\theta)+\Psi_H(r,\theta) & 0\le\theta\le 2\pi \quad r\ge r_b.\end{cases} \tag{32}$$

Here $(\Psi_g)_{i,j}$ is the solution of the discrete Poisson equation

$$(\Delta^h\Psi_g)_{i,j}=\omega_{i,j} \qquad \Psi_g(\theta_i,r_a)=0 \quad \Psi_g(\theta_i,r_b)=c(\theta_i) \tag{33}$$

in the annulus $r_a\le r_j\le r_b$. $\Psi_H(r,\theta)$ is a specified harmonic function taking the prescribed boundary conditions at infinity (including a prescribed amount of net circulation). The function Ψ_f is a finite Fourier series solution to Laplace's equation in the region $r\ge r_b$ with boundary data $d(\theta_i)$,

$$\Psi_f(r,\theta)=\sum_{k=-M/2+1}^{k=M/2}\left(\frac{r}{r_a}\right)^{-|k|}\alpha_k e^{ik\theta}, \tag{34}$$

with the coefficients α_k chosen so that $\Psi_f(r_b,\theta_i)=d(\theta_i)$, $i=1\ldots M$.

With these definitions of Ψ_g, Ψ_f, and Ψ_H the solution Ψ will be a discrete approximation to the solution of (30) in the annulus $r_a\le r_j<r_b$ and will be an analytic solution to (30) in the region $r>r_b$. The values at $r=r_b$, $\mathbf{c}$ and $\mathbf{d}$ which are necessary to determine Ψ_g and Ψ_f are still unknown. One obtains equations for these values by requiring that the composite solution (32) be continuous and have a continuous normal derivative across the interface $r=r_b$. (The satisfaction of these conditions is necessary for the composite solution to be an accurate approximation at $r=r_b$.)

We employ the technique of collocation at the grid points with $r=r_b$ and obtain a set of discrete equations of the form

$$(\Psi_g)_{i,N+1}=\Psi_f(\theta_i,r_b)+\Psi_H(\theta_i,r_b) \qquad i=1\ldots m \tag{35}$$

$$D_r(\Psi_g)_{i,N+1}=\frac{\partial\Psi_f}{\partial r}(\theta_i,r_b)+\frac{\partial\Psi_H}{\partial r}(\theta_i,r_b) \quad i=1\ldots m \tag{36}$$

with D_r being either a second or fourth order one sided approximation to the radial derivative. (The order depends on the order of the discrete Laplacian used in the approximation of (33)).

In our approach we express the derivative of Ψ_g in the second equation, (36), as $D_r(\Psi_g)^\omega_{i,N+1}+D_r(\Psi_g)^c_{i,N+1}$. $(\Psi_g)^\omega$ is that component of Ψ_g defined as the solution of (33) with data ω in the interior and homogeneous boundary values. $(\Psi_g)^c$ is the component of Ψ_g defined as the solution of (33) with homogeneous data in the interior and boundary conditions $\Psi^c_g=0$ at $r=r_a$ and $\Psi^c_g=\mathbf{c}$ at $r=r_b$. Thus Eq. (36) becomes

$$D_r(\Psi_g)^\omega_{i,N+1}+D_r(\Psi_g)^c_{i,N+1}=\frac{\partial\Psi_f}{\partial r}(\theta_i,r_b)+\frac{\partial\Psi_H}{\partial r}(\theta_i,r_b) \qquad i=1\ldots m. \tag{37}$$

The substitution of the values $c_i=\Psi_g(\theta_i,r_b)$ and $d_i=\Psi_f(\theta_i,r_b)$ into (35) yields the equations

$$c_i=d_i+\Psi_H(\theta_i,r_b) \quad i=1\ldots m. \tag{38}$$

By definition Ψ^c_g and Ψ_f are solutions of Laplace's equations with boundary data $\mathbf{c}$ and $\mathbf{d}$ at $r=r_b$, so Eq. (37) provides a set of equations involving the boundary data which we express as

$$(B_g\mathbf{c})_i=(B_f\mathbf{d})_i+\frac{\partial\Psi_H}{\partial r}(\theta_i,r_g)-D_r(\Psi_g)^\omega_{i,N+1} \qquad i=1\ldots m. \tag{39}$$

In this equation B_g is a matrix which gives the discrete normal derivative associated with a solution of (33) with $\omega\equiv 0$ and data $\mathbf{c}$ at $r=r_b$. B_f is that matrix which gives the normal derivative of a finite Fourier series (34) which has the values $\mathbf{d}$ at $r=r_b$.

In block matrix form the equations for the unknown values $\mathbf{c}$ and $\mathbf{d}$ can be expressed as

$$\begin{pmatrix}I & -I\\ B_g & -B_f\end{pmatrix}\begin{pmatrix}\mathbf{c}\\ \mathbf{d}\end{pmatrix}=\begin{pmatrix}\boldsymbol{\Psi}_H\\ \partial\boldsymbol{\Psi}_H/\partial r-\mathbf{D}_r(\Psi_g)^\omega\end{pmatrix}. \tag{40}$$

Here we are using $\boldsymbol{\Psi}_H$ and $\partial\boldsymbol{\Psi}_H/\partial r$ to denote the values of Ψ_H and its normal derivative at the grid points with $r=r_b$.

The solution procedure we employ is one using block Gaussian elimination (or forming the Schur complement) on the system (40). The matrix equation which results from the Gaussian elimination process is a circulant matrix (i.e., it represents a discrete convolution) and we employ the discrete fast Fourier transforms to compute its solution [19]. One aspect of using the fast Fourier transform to compute solutions of circulant systems is that the matrix

need not be formed—all one needs is the first column of the matrix. Thus, while the system we must solve is composed of dense matrices, these need not be explicitly constructed to obtain solutions. (If one applies this procedure to obtain solutions with infinite boundary conditions on regions whose interface is not a circle, then the matrices which arise will no longer be circulant. However, one approach to solving such systems would be to use an iterative method and the solution procedure presented here for a circle as a pre-conditioner.)

The first step towards solving (40) consists of multiplying the equation by the matrix

$$\begin{pmatrix} I & \\ -B_g & I \end{pmatrix}.$$

This results in the system of equations

$$\begin{pmatrix} I & -I \\ & B_g - B_f \end{pmatrix}\begin{pmatrix} \mathbf{c} \\ \mathbf{d} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\Psi}_H \\ -B_g \boldsymbol{\Psi}_H + \partial \boldsymbol{\Psi}_H / \partial r - \mathbf{D}_r(\Psi_g)^\omega \end{pmatrix}. \tag{41}$$

Thus, we solve

$$(B_g - B_f)\mathbf{d} = -B_g \boldsymbol{\Psi}_H + \frac{\partial \boldsymbol{\Psi}_H}{\partial \boldsymbol{r}} - \mathbf{D}_r(\Psi_g)^\omega \tag{42}$$

for $\mathbf{d}$ and then we "back solve" to obtain

$$\mathbf{c} = \mathbf{d} + \boldsymbol{\Psi}_h.$$

With the values of $\mathbf{c}$ known we can then solve (33) for Ψ_g in the annulus and obtain a finite difference approximation to the solution in that region.

The matrices in (42) are circulant, and thus to obtain their solution using the discrete fast Fourier transform we need only know the first column of the matrices. To obtain the first column requires that we compute

$$(B_g - B_f)\mathbf{e}_1, \tag{43}$$

where $\mathbf{e}_1 = (1, 0, 0, 0, ..., 0)^t$. The computation of $B_g\mathbf{e}_1$ consists of solving (33) with $\omega \equiv 0$ and boundary data $\mathbf{e}_1$ at $r = r_b$ and then evaluating D_r of this solution at $r = r_b$. The computation of $B_f\mathbf{e}_1$ first requires determining α_k so that (34) evaluated at $r = r_b$ takes on the values $\mathbf{e}_1$. If the discrete Fourier transform of $\mathbf{e}_1$ is denoted by $\tilde{\mathbf{e}}_1$, then the α's are given by $\alpha_k = (\tilde{\mathbf{e}}_1)_k$. The second step requires evaluating the radial derivative of (34). This is efficiently accomplished by using the discrete Fourier transform to evaluate the differentiated series. The vector determined by (43) does not depend on the data ω and so need only be computed once for any particular computation.

Thus, our solution for Laplace's equation in the infinite domain is obtained by using an approximation consisting of the composition of a finite difference solution in an annulus about the cylinder and an analytic (Fourier series) solution outside the annulus. In order for this composite solution to be a solution of Laplace's equation we require that it be continuous and have continuous normal derivative at $r = r_b$. These latter conditions give rise to a set of linear equations (40) which determine the boundary data for the finite difference and the Fourier series solution at $r = r_b$. The solution of this system of equations can be reduced to solving a circulant system of equations (42)—a system which can be efficiently solved using discrete Fourier transforms. To carry out this solution procedure requires a few discrete Fourier transforms and two solutions of a discrete Poisson equation on the annulus. The latter can be accomplished using fast Poisson solvers.

ACKNOWLEDGMENTS

Dr. Anderson's research is supported in part by ARO Grant DAAL03-91-G-0162, ONR Contract N00014-86-K-0691, and NSF Grant DM586-57663. Dr. Reider's research was supported by the Department of Energy at Los Alamos National Laboratory.

REFERENCES

1. M. Abramowitz and I. Stegun (Eds.), *Handbook of Mathematical Functions.* (Dover, New York, 1965).
2. Christopher R. Anderson, Domain decomposition techniques and the solution of Poisson's equation in infinite domains, in *Domain Decomposition Methods* (SIAM, Philadelphia, 1989), p. 129.
3. Christopher R. Anderson, *J. Comput. Phys.* **80,** 72 (1989).
4. C. R. Anderson and M. B. Reider, *Phys. Fluids* **6,** 2380 (1994).
5. R. Bouard and M. Coutanceau, The early stage of development of the wake behind an impulsively started cylinder for $40 < \mathrm{Re} < 10^4$, *J. Fluid Mech.* **101,** 583 (1980).
6. M. Braza, P. Chassaing, and H. HaMinh, Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder, *J. Fluid Mech.* **165,** 79 (1986).
7. K. S. Chang and J. Y. Sa, The effect of buoyancy on vortex shedding in the near wake behind a circular cylinder, *J. Fluid Mech.* **220,** 253 (1990).
8. A. Y. Cheer, Unsteady separated wake behind an impulsively started cylinder in slightly viscous fluid, *J. Fluid Mech.* **201,** 485 (1989).
9. A. J. Chorin, Numerical study of slightly viscous flow, *J. Fluid Mech.* **57,** 785 (1973).
10. G. Golub and C. Van Loan (Eds.), *Matrix Computations* (Johns Hopkins Univ. Press, Baltimore, 1983).
11. W. D. Henshaw, H. O. Kreiss, and L. G. Reyna, On the smallest scale for the incompressible Navier–Stokes equations, *Theoret. Comput. Fluid Dynam.* **1,** 65 (1989).
12. Thomas Y. Hou and Brian T. R. Wetton, Convergence of a finite difference method using vorticity boundary conditions, in *Vortex Dynamics and Vortex Methods* (Am. Meth. Soc., Providence, 1991), p. 363.

13. D. B. Ingham, Note on the numerical solution for unsteady viscous flow past a circular cylinder, *J. Fluid Mech.* **31,** 815 (1968).
14. Y. Lecointe and J. Piquet, *Comput. Fluids* **12,** 255 (1984).
15. Ta Phuoc Loc and R. Bouard, *J. Fluid Mech.* **160,** 93 (1985).
16. A. E. Perry, M. S. Chong, and T. T. Lim, *J. Fluid Mech.* **116,** 77 (1982).
17. R. Peyret and T. D. Taylor, *Computational Methods for Fluid Flow* (Springer-Verlag, Berlin/New York, 1983), p. 184.
18. W. Press, S. Teukolsky, W. Vetterinlin, and B. Flannery, *Numerical Recipes* (Cambridge Univ. Press, Cambridge, 1992), p. 530.
19. M. Reider, *Development of Higher Order Numerical Methods for Two Dimensional Incompressible Flows with Applications to Flows around Circular Cylinders and Airfoils* Ph.D. thesis (Univ. of Calif. at Los Angeles, Los Angeles, 1992).
20. P. Roache, *Computational Fluid Dynamics* (Hermosa, Albuquerque, NM, 1976).
21. Y. Saad and H. M. Schultz, GMRES: *SIAM J. Sci. Statist. Comput.* **7,** 856 (1986).
22. T. Sarpkaya, *J. Basic Eng.* **90,** 511 (1968).
23. T. Sarpkaya, *AIAA J.* **17,** 1193 (1979).
24. P. A. Smith and P. K. Stansby, *J. Fluid Mech.* **194,** 45 (1988).
25. J. S. Son and T. J. Hanratty, *J. Fluid Mech.* **35,** 369 (1969).
26. A. Thom, *Proc. Roy. Soc. London Ser. A* **141,** 651 (1933).
27. E. Tiemroth, *The Simulation of Viscous Flow Around a Cylinder by the Random Vortex Method,* Ph.D. thesis (U.C. Berkeley Naval Arch. Dept., 1986).
28. D. J. Tritton, *J. Fluid Mech.* **6,** 547 (1959).