JOURNAL OF COMPUTATIONAL PHYSICS 95, 400–435 (1991)

Diffusing-Vortex Numerical Scheme for Solving Incompressible Navier–Stokes Equations

ZHI YUN LU

Department of Mechanical, Industrial, and Aerospace Engineering, New York Institute of Technology, Old Westbury, New York, 11568

AND

TIMOTHY J. ROSS

Department of Civil Engineering, University of New Mexico, Albuquerque, New Mexico 87131

Received July 24, 1989; revised February 23, 1990

A new numerical algorithm, the diffusing-vortex method for time-dependent two-dimensional Navier–Stokes equations, which was previously presented and applied to the incompressible viscous flow past a circular cylinder with high Reynolds number by Lu and Shen [1], is further developed for extension to general two-dimensional initial value problems and boundary value problems. The new algorithm consists of two time steps is a simulation cycle: a Lagrangian convection simulation for the first time step and a diffusion simulation through the use of new vortex points at dixed Eulerian mesh points for the second time step. The mathematical mechanisms of computation behind this algorithm, and its characteristics of convergence and accuracy, are analyzed in applications for the following problems: (1) an initial value problem involving the decay of a single vortex of finite size and the decay and interaction of a vortex-pair of two finite-core regions; (2) a boundary value problem: the unsteady flow field around a rotating cylinder with high Reynolds number up to $\mathrm{Re}_d = U \cdot D/\nu < 10{,}000$. Numerical results are compared with either exact solutions or other numerical methods. The numerical advantages of the diffusing-vortex scheme over other conventional vortex methods, Cloud-in-Cell methods, particle methods, and some finite difference schemes are evaluated in terms of reducing total CPU time, avoiding cutoff procedures, and sidestepping various interpolations. © 1991 Academic Press, Inc.

1. INTRODUCTION

Two-dimensional and three-dimensional viscous flow problems of practical importance remain today beyond the scope of prevailing methods for the solution of the governing full Navier–Stokes equations. Great strides have been made in recent years in the computational treatment of the Navier–Stokes equations as evidenced by many research papers on the subject, especially in the development of various finite difference and finite element methods. It is well recognized that the non-linear con-

0021-9991/91 $3.00

vection terms are primarily responsible for many of the complex flow phenomena, and are the major cause of numerical difficulties. At high Reynolds numbers, strong non-linearity effects produce details characterized by scales which are widely apart, and the resolution of their interaction has been a major challenge. In other words, the length scale for the viscous region adjacent to the wall is inversely proportional to the square root of the Reynolds number, therefore much smaller than that for the inviscid region. In discretizing the entire domain, a very fine mesh is required to provide sufficient accuracy within the very thin boundary layer. To properly capture the complicated interaction behavior, very high mesh resolution is required. Without a very efficient and accurate numerical algorithm and an effective data structure, it is extremely difficult to obtain reliable numerical results.

Much success in solving Navier–Stokes equations has been achieved by using the finite-difference technique, especially for application to internal flow at low Reynolds numbers. Extensive bibliographies are given by Harlow [2], Roache [3], and Wilkes *et al.* [4]. These references survey a great variety of schemes including the so-called methods: MAC, PIC, CEL, LINC, SIMPLE, SIMPLER, PISO, ICCG, QUICK, REMIXCS, and others. Only a few computational examples using the above algorithms have involved Reynolds number larger than 1000. Thoman and Szewczyk [5] use the finite difference scheme (originally suggested by Lelevier as reported by Richtmyer [6]) to calculate the flow past a circular cylinder. A method developed by Collins and Dennis [7] for solving high Reynolds number flow is based on expansions of the Fourier type and was successful under the assumption of the normalized time being less than one. Wei and Giiceri [8] present the results for two-dimensional separated flows past blunt bodies of arbitrary shape using the modified strongly implicit (MSI) method for Reynolds numbers less than 100. Kwak and Chakravarthy [9] developed an implicit three-dimensional finite difference code, using primitive variables and pseudocompressibility approach. Mane and Ta Phuoc Loc [10] solve the vorticity transport equation for unsteady flow past an airfoil with Reynolds numbers up to 10^5 by using fourth-order accurate alternating directional implicit (ADI) schemes. Although some quantitative results like the instantaneous vorticitiy distribution, the separation point, drag coefficient, lift coefficient, and wake length as a function of time were not presented in their paper, their final results are in very good agreement with a flow visualization of streamlines.

Typically, the "vortex method" of simulating a real flow field is to discretize the regions of nonzero vorticity into point vortices and track their motion in a Lagrangian coordinate system. The local velocities can be computed by the Biot–Savart law, or from a Poisson equation according to classical hydrodynamics. Calculations using this method to study inviscid fluid motion involving point vortices have a long history—from Rosenhead's hand calculation in 1931 [11] using a few vortices to recent computer calculations involving thousands of vortices. General reviews of vortex methods have been given, e.g., by Saffman and Baker [12], Leonard [13, 14], Aref [15], Anderson and Greengard [16]. The details of the high order vortex method are described in Beale and Majda's papers [17–19].

Perlman [20] and analyzed the effects of various parameters on the accuracy of vortex methods. Chorin in a number of papers [21–23] pioneered a random walk treatment of the two-dimensional vorticity equation. The main advanatage of such vortex methods is the elimination of the need of a fixed mesh grid. Another advanrage is that the computational domain excludes the irrotational region and becomes much smaller in many cases. However, the computational work is proportional to the square of the number of vortices, N^2.

To reduce the operation count of basic vortex interaction equations and the singular behavior of point vortices, the "vortex-in-cell" method was developed. As a special version of the "Cloud-in-Cell (CIC)" method that has arisen in several fields, the vortex-in-cell method was described for vortices by Christiansen [24]. In this method, the velocity field over a fixed grid was calculated from the stream function, which is obtained from the vorticity by inversion of the Poisson equation. However, two additional steps are introduced: (1) the generation of mesh-point values for the vorticity field from the discrete vortices and (2) the interpolation of mesh point values of the velocity field back to the Lagrangian vortex points. Unfortunately, the uncertain numerical errors arising from the anisotropic interpolation of velocities in step (2) could often be significant. In numerical results presented by Baker [25], while the total circulation was conserved by the CIC method, the angular impulse was not conserved.

Recently, Greengard and Rokhlin [26] proposed a multipole algorithm that would significantly reduce the velocity calculation in potential flow for initial value problems if the distribution of vorticity is reasonably uniform in a square-shaped region of interest.

An elegant random vortex blob method for simulating high Reynolds number flow was proposed by Chorin [21–23] for application to two-dimensional cases. According to Chorin, the vorticity field is discretized with a vortex blob of small but finite support. The vorticity transport equation is solved by tracking a collection of vortex blobs in the interior and by tracking vortex sheets near the solid boundary through the fluid with an random motion added to model the viscous diffusion. A carefully conducted application so far is due to Cheer [27, 28], who recently used Chorin's algorithm and studied a chellenging subject—the unsteady separated wake development around an impulsively started circular cylinder. Cheer's newest results [28] were in very good agreement with experimental data of Bouard and Coutanceau [29] and have evidently shown that the fractional time step associated with the random walk algorithm is a valid tool for problems involving a solid boundary in flow with high Reynolds numbers. However, experience has shown (Shestakov [30]; Lu [31]; Shen and Lu [32]) that convergence is often slow and computation is time-consuming since the total number of operations per time step for this algorithm is proportional to N^2.

With regard to the method by Raviart and his co-workers [33], as outlined and analyzed in Cottet and Gallic [34, 35], like Chorin they track the discretized vortices strictly in Lagrangian fashion without a fixed grid. Unlike Chorin they allow the strength (or "weight") of each vortex to change after time step Δt, by evaluating the contributions from the Gaussian spreading of all the vortices. The random-walk displacements are eliminated, but other difficulties associated with the usual Lagrangian vortex methods remain—e.g., the dependence of CPU on N^2 for velocity and the cutoff effects. Later, Choquin and Lucquin [36] used a deterministic particle method to calculate the vorticity field induced by a single circular vortex. They found that the choice of the cutoff parameter in the particle method does play a very important role in the accuracy of the method. No numerical examples calculated by this method are known to us where *a solid boundary exists.*

A new numerical algorithm, the diffusing-vortex method as previously used by Lu and Shen [1] to study the two-dimensional incompressible flow past a cricular cylinder at Reynolds number 9500. The results showed excellent agreement with the experimental visualization obtained by Bouard and Coutanceay [29].

This paper describes the extension of the work by Lu [37] and Lu and Shen [1] to formulate a new numerical algorithm and device a numerical scheme for solving the general two-dimensional time-dependent incompressible Navier–Stokes equations at *high Reynolds number.* The advantages of tis method in reducing CPU time, avoiding the difficulties in cutoff properties of the vortices, and circumventing various anisotropic interpolations will be shown through applications to general initial value problems and to boundary value problems. In Section 2, we present the basic approach which is based on operator-splitting and discuss the features that distinguish our method from other closely related ones. Section 2 also gives the details of our algorithm for initial value problems and boundary value problems and consideration of truncation error, mesh size, and time step. Section 3 deals with numerical examples and results. Section 4 summarizes our conclusions.

2. The New Algorithm

Two-dimensional incompressible viscous flow described by the Navier–Stokes equations are considered. Let (x, y) be the Cartesian coordinates, (u, v) be the velocity components, t be the time and ζ be the vorticity. Suppose now that L_c is the characteristic length and U is the characteristic velocity of the flow. Then the following nondimensional quantities, without the "bar," can be defined:

$$u=\bar{u}/U, \quad v=\bar{v}/U, \quad x=\bar{x}/L_c, \quad y=\bar{y}/L_c, \quad t=\bar{t}U/L_c, \quad \zeta=\bar{\zeta}L_c/U. \tag{1}$$

Consequently, the vorticity transport equation can be written as

$$\frac{\partial\zeta}{\partial t}+u\frac{\partial\zeta}{\partial x}+v\frac{\partial\zeta}{\partial y}=\frac{1}{\mathrm{Re}}\nabla^2\zeta, \tag{2}$$

where $\mathrm{Re}=UL_c/v$ is the Reynolds number. The flow is fully described, of course, when the vorticity ζ is known at each instant.

Equation (2) is composed of the two mechanisms of vorticity diffusion and vorticity convection. Following Yanenko [38] and Chorin [21–23], the time integration may be broken into two fractional steps: pure diffusion and inviscid convection. Thus, we solve separately during time step Δt, the pure diffusion equation

$$\frac{\partial \zeta}{\partial t}=\frac{1}{\mathrm{Re}}\nabla^2\zeta \tag{3}$$

and the inviscid convection equation

$$\frac{\partial \zeta}{\partial t}+u\frac{\partial \zeta}{\partial x}+v\frac{\partial \zeta}{\partial y}=0. \tag{4}$$

This technique was previously adopted by Chorin [21–23] in his random-walk vortex method. Because of Eq. (3), Chorin proposed a random-walk simulation of the diffusion process, based on its well-known mathematical analogy with Brownian motion. Equation (4) was handled by straightforward tracking. To validate his algorithm, the classical unsteady boundary layer over an impulsively started flat plate was solved as an example. With relatively few vortices and moderate time steps, the steady state Blasius solution was indeed reproduced by taking the average of a large number of profiles. However in this case, to obtain accurate instantaneous unsteady velocity profiles, Chorin's computations need further processing (see Shen and Lu [32]). Lu [31] also showed that the statistical error could become serious if the total number of vortices was not large enough. Since the statistical error is inversely proportional to the square root of the total number of samples N, but the amount of computing time increases like N^2, the improvement of accuracy would rapidly become prohibitively costly.

A study was initiated to sidestep the random-walk aspects in Chorin's algorithm. Actually, the task is only to be able to update the vorticity field, accurately, for small Δt in accordance with Eq. (3). Gaussian spreading of each vortex after Δt was used, before moving the vortices according to Eq. (4). For numerical implementation, simpler programming and less CPU could be achieved by retaining a fixed grid for the vortical region. As a result, a scheme has been developed in which for each time step the discretized vortices always start from the mesh points of a fixed grid. To simulate diffusion, each vortex is first shattered during Δt to spread its vorticity, in a Gaussian distribution, among all mesh points. The total vorticity at each mesh point then defines the new vortices, as well as the local velocity to be used for the vortex displacement during Δt. *In contrast with the usual Lagrangian vortex methods, the vortices move only one time step, then dissolve to create new vortices at fixed mesh points.* The details of the algorithm are described below.

2.1. *Simulation of Diffusion for Initial Value Problems*

For the initial value problem in an unbounded domain without a solid boundary, if the vorticity distribution at moment t is known, the vorticity field after a lapse of time Δt, i.e., the solution to Eq. (3), can be expressed in terms of the Green's function

$$\zeta(x, y, t+\Delta t) = \iint_{R^2} \zeta(x', y', t)\, G(x, y, x', y', \Delta t)\, dx'\, dy', \tag{5}$$

where G is the Green's function for an initial value problem, which is the Gaussian

$$G(x, y, x', y', \Delta t) = \frac{\text{Re}}{4\pi\, \Delta t} \exp\left(-\frac{r^2\, \text{Re}}{4\, \Delta t} \right), \tag{6}$$

where $r^2 = [(x-x')^2 + (y-y')^2]^{1/2}$ and all quantities are normalized. The discretized form of Eq. (5) can be written as

$$\zeta(x, y, t+\Delta t) = \sum_i \sum_j \gamma_{i,j} G(x, y, x_{i,j}, y_{i,j}, \Delta t), \tag{7}$$

where the subscripts i and j are integers referring to x and y at the mesh points, $\gamma_{i,j} = \zeta_{i,j} \Delta x\, \Delta y$, and

$$r_{i,j} = [(x - x_{i,j})^2 + (y - y_{i,j})^2]^{1/2}.$$

The diffusion process of a continuous vorticity field is thus approximated by the decay of many discretized vortices, located at the mesh points $(x_{i,j}, y_{i,j})$ with strength $\gamma_{i,j}$, each of which spreads with a Gaussian distribution, with variance $\delta^2 = 2\, \Delta t/\text{Re}$, over the unbounded domain. From a statistical point of view, the probability that the vortex at (i, j) moves to a region of area $dx \cdot dy$ centered at the point (x, y) would be $G(x, y; x_i, y_j)\, dx \cdot dy$, and G is given by the Gaussian in Eq. (6). Recalling the mathematical relationship between Brownian motion and the diffusion process, this interpretation is also the cornerstone on which Chorin's random-walk algorithm rests.

Let the spatial ragion with vorticity be divided into uniform square mesh cells, of length $\Delta x = \Delta y = h$, and each small area h^2 is centered at $(x_{i,j}, y_{i,j})$ as shown by Fig. 1. The integer symbol (m, n) indicates the position of calculated point which is at $x = m \cdot h$ and $y = n \cdot h$. They will be used frequetly as x, y indices at next time level $\Delta + 1$.

The vorticity after Δt at each nodal point (m, n) is from Eq. (7)

$$\zeta(x_m, y_n, t+\Delta t) = \frac{h^2}{\pi \lambda^2} \sum_i \sum_j \zeta(x_{i,j}, y_{i,j}, t) \exp\left(-\frac{r_{i,j}^2}{\lambda^2} \right), \tag{8}$$

where

$$\lambda = (4\, \Delta t/\text{Re})^{1/2}, \tag{9}$$

$$\begin{aligned} x_{i,j} &= x_i = i \cdot h \qquad (i = 1, 2, ..., i_{\max}) \\ y_{i,j} &= y_j = j \cdot h \qquad (j = 1, 2, ..., j_{\max}). \end{aligned} \tag{10}$$

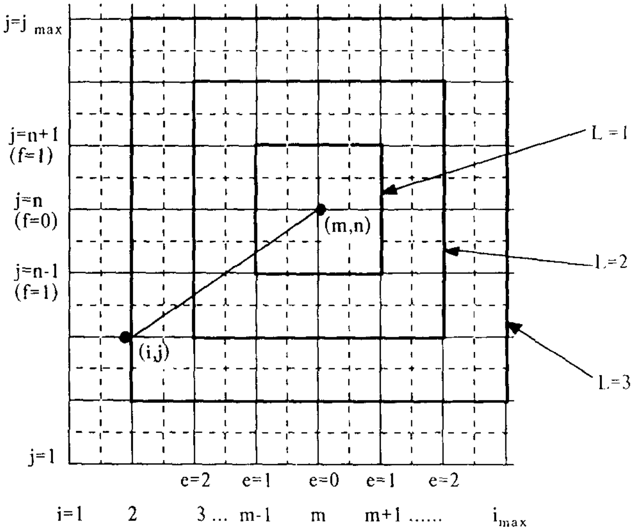

FIG. 1. Mesh scheme for initial value problem.

Let a mesh length parameter c be introduced,

$$c = h/\lambda, \tag{11}$$

and also the time level index A, which denotes the discretized time t after A time steps Δt; i.e., $t = A \cdot \Delta t$. Then Eq. (8) may be rewritten as

$$\zeta_{m,n}^{(A+1)} = \frac{c^2}{\pi} \sum_i \sum_j \zeta_{i,j}^{(A)} \exp\{-[(i-m)^2 + (j-n)^2]c^2\}. \tag{12}$$

Given a mesh point $P(m, n)$, the nearest neighbors consist of eight mesh points on the solid line $L = 1$ (the first layer, see Fig. 1). Similarly, on the second line $L = 2$, there are $2 * 8 = 16$ mesh points. By induction the terms in Eq. (12) may be regrouped to give

$$\zeta_{m,n}^{(A+1)} = \frac{c^2}{\pi} \left\{ \sigma_{0,0}^{(A)} + \sum_{k=1}^{2} \sigma_{1,2-k}^{(A)} \exp\{-[1^2 + (2-k)^2]\} \right. \\ \left. + \cdots + \sum_{k=1}^{2L} \sigma_{L,|L-k+1|}^{(A)} \exp\{-[L^2 + (L-k+1)^2]c^2\} + \cdots \right\}, \tag{13}$$

where the symbol $\sigma_{e,f}^{(A)}$ is defined by

$$\sigma_{e,f}^{(A)} = \sigma_{0,0}^{(A)} = \zeta_{m,n}^{(\zeta)} \qquad \text{for} \quad e = f = 0 \tag{14a}$$

$$\sigma_{e,f}^{(A)} = \zeta_{m+e,n+f}^{(A)} + \zeta_{m+e,n-f}^{(A)} + \zeta_{m-e,n+f}^{(A)} + \zeta_{m-e,n-f}^{(A)} \qquad \text{otherwise.} \tag{14b}$$

In Eqs. (14a) and (14b) the subscript e is an integer indicating the horizontal distance between the fixed reference point (m, n) and the calculated point in terms of the number of mesh cells and the subscript f is similarly the integer referring to the vertical distance (see Fig. 1) between the same points.

It is well known that the maximum value of the solution of the transient diffusion equation in unbounded space can only decay with time. Supposing the maximum value is at the point (m, n), then

$$\zeta_{m,n}^{(\Delta+1)} \leqslant \zeta_{m,n}^{(\Delta)} \tag{15}$$

Because of Eq. (13), the above inequality becomes

$$\begin{aligned}\zeta_{m,n}^{(\Delta+1)} = \frac{c^2}{\pi}\Bigg\{&\sigma_{0,0}^{(\Delta)} + \sum_{k=1}^{2} \sigma_{1,2-k}^{(\Delta)} \exp\{-[1^2+(2-k)^2]\} + \cdots \\ &+ \sum_{k=1}^{2L} \sigma_{L,|L-k+1|}^{(\Delta)} \exp\{-[L^2+(L-k+1)^2]c^2\} + \cdots\Bigg\} \leqslant \zeta_{m,n}^{(\Delta)}.\end{aligned} \tag{16}$$

In the limiting case where the vorticity is uniformly distributed over the whole domain,

$$\zeta_{m,n}^{(\Delta+1)} = \zeta_{m,n}^{(\Delta)} = \cdots = \zeta_0. \tag{17}$$

Equations (14a) and (14b) simply become one equation,

$$\sigma_{e,f}^{(\Delta)} = 4\zeta_0, \tag{18}$$

and Eq. (16) is reduced to

$$\begin{aligned}\frac{c^2}{\pi}\{&1+4[\exp(-2c^2)+\exp(-c^2)]+4[\exp(-8c^2)+2\exp(-5c^2)+\exp(-4c^2)] \\ &+4[\exp(-18c^2)+2\exp(-13c^2)+2\exp(-10c^2)+\exp(-9c^2)]+\cdots\}=1.\end{aligned} \tag{19}$$

Increasing the total number of computing layers L to larger than 11 has little effect (an accuracy of 10^{-8}) on the solution for c, since the extra terms on the left-hand side of Eq. (19) become negligibly small. From a computational point of view, however, the value of c controls the difference between the exact solution $\zeta(x_m, y_n, t+\Delta t) = \zeta_0$ and the discretized approximate solution $\zeta_{m,n}^{(\Delta+1)}$. We consider it imperative that any algorithm reproduce correctly the equilibrium state $\zeta = \zeta_0$. In this case, the relative discretization error for vorticity can be measured in the terms of the norm

$$E = \sum_m \sum_n \left\{\frac{\zeta_{m,n}^{(\Delta+1)} - \zeta_0}{\zeta_0}\right\}^2. \tag{20}$$

The effect of the mesh length parameter c on the discretized solution for this limiting case is shown by Fig. 2. For $c \approx 1$, the ratio of the calculated vorticity over

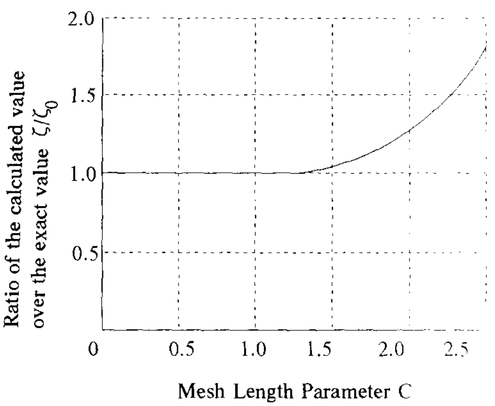

FIG. 2. Effect of the mesh length parameter c on discretized solution.

the exact value is equal to 1 approximately. When $c > 1.3$, error increases rapidly. For the given value of $E \approx O(10^{-5})$, the value of c must satisfy

$$c < 0.878. \tag{21}$$

The inequality (21) *will be regarded as a necessary condition for the discretization of the diffusion equation. It relates the mesh length with the diffusion distance for a given time step and Reynolds number.*

For an arbitrary vorticity field at time t, the inequality (21) *is till true at the point where the vorticity is a maximum* $\zeta_{\max} = \zeta^{(A)}_{m,n}$. Since all other vortices are weaker, the upper bound of $\zeta^{(A+1)}_{m,n}$ is less than that for the uniform state $\zeta = \zeta_0$. Hence Eq. (21) must be equally valid, in fact somewhat conservative. At any rate, the correct representation of the uniform state and the decay of maximum vorticity are thus both ensured.

Cottet and Gallic [34, 35] give a convergence proof for their particle method. This proof states that, in order for the method to converge to a solution of the Navier–Stokes equations, it is necessary to have

$$h \leqslant C_c(\Delta t/\mathrm{Re})1/2 \tag{21b}$$

for some constant C_c. Using our notation with λ (see Eq. (9)), the inequality (21b) can be rewritten as

$$c \leqslant 2C_c. \tag{21c}$$

Our inequality (21) has shown numerically that the unknown constant in Cottet and Gallic's formula [35] is a function of the global relative error E, as shown in Fig. 2. For the special value of $E \approx O(10^{-5})$, Cottet and Gallic's constant C_c should be less than 0.439.

2.2. *Simulation of Diffusion for Boundary Value Problems*

With a given value of the vorticity along the boundary, $\zeta_b(t)$, and the initial distribution of the vorticity, $\zeta_0(x, y, t)$, at time t, the solution of the diffusion equation Eq. (3) at $t+\Delta t$ is (see Morse and Feshbach [39])

$$\begin{aligned}\zeta(x, y, t+\Delta t) &= \iint_D G|_{\tau=t}\, \zeta_0(x', y', t)\, dx'\, dy' - \frac{1}{\text{Re}} \int_t^{t+\Delta t} d\tau \int_b \zeta_b \left[\frac{\partial G}{\partial n}\right]_b ds \\ &= \zeta_{\text{I}}(x, y, t+\Delta t) + \zeta_{\text{II}}(x, y, t+\Delta t),\end{aligned} \tag{22}$$

where G is the appropriate Green function, defined by

$$\begin{aligned}\frac{\partial G}{\partial t} &= \frac{1}{\text{Re}} \nabla^2 G + \delta(x-\bar{x})\, \delta(y-\bar{y})\, \delta(t-\tau) \\ G|_b &= 0 \qquad \text{for} \quad t > \tau\end{aligned} \tag{23}$$

and the term $(\partial G/\partial n)_b$ indicates the derivative in the direction normal to the surface b. The first term ζ_{I} represents the diffusion contribution from the vortices inside domain D and ζ_{II} is the contribution from the vortices on the solid boundary.

The solution of Eqs. (22) and (23) consists of answering three critically fundamental questions concerned with the validity of approximations to Green's function. These questions are:

- How can Green's function be used to calculate the diffusion solution for an arbitrary boundary?
- Is the total number of operations for the diffusion simulation still of the order of N^2?
- Does mesh distortion exist with the new scheme?

Answers to these questions are provided in the subsequent sections of this paper.

2.2.1. *Construction of the Green's Function for an Arbitrary Boundary*

It is difficult to solve the time-dependent Green's function analytically for arbitrary 2D boundary conditions. Fortunately, *for high Reynolds number cases* $\text{Re} \gg 1$, *we can use the solution of the flat plate as an approximation to the solution for an arbitrary boundary.* Suppose the domain of interest is the half-space $y > 0$, bounded by the line $y = 0$ (see Fig. 3a). In this case the required Green's function is obviously obtained by adding to the diffusing δ-function, i.e., the Gaussian, its image in the lower half-plane. In fact, it satisfies the no slip condition, including the zero normal velocity along $y = 0$, automatically.

As shown in Fig. 3a, for a vortex near an infinite flat plate, the Green's function can be written

$$G(x_m, y_n, x_i, y_j, t, \tau) = \frac{\text{Re}}{4\pi(t-\tau)} \left\{\exp\left(-\frac{r_{i,j}^2\, \text{Re}}{4(t-\tau)}\right) - \exp\left(-\frac{(r_{i,j})_{im}^2\, \text{Re}}{4(t-\tau)}\right)\right\}, \tag{24}$$

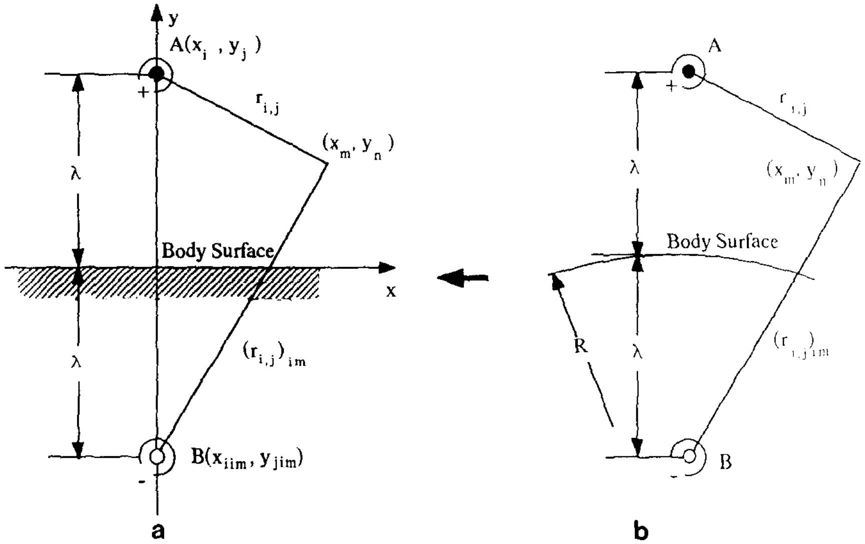

FIG. 3. (a) Vortex point $A(x_i, y_j)$ and its image $B(x_{iim}, y_{jim})$; (b) vortex A outside curve surface (simplified as Fig. 3a if $\lambda \ll 1$

where $r_{i,j}$ is the distance between the vortex point at (x_i, y_j) and the calculated mesh point (x_m, y_n),

$$r_{i,j} = [(x_i - x_m)^2 + (y_j - y_n)^2]^{1/2} \tag{25a}$$

and $(r_{i,j})_{im}$ is the distance between the point (x_{iim}, y_{jim}) inside the solid body (the image point of $x_i, y_j)$ and the point (x_m, y_n),

$$(r_{i,j})_{im} = [(x_{iim} - x_m)^2 + (y_{jim} - y_n)^2]^{1/2}. \tag{25b}$$

Using $\lambda = (4\,\Delta t/\mathrm{Re})^{1/2}$, changing the form of Eq. (24) to fit equation (22), i.e.,

$$G(x_m, y_n, x_i, y_j, \Delta t) = \frac{1}{\pi\lambda^2}\left\{\exp\left(-\frac{r_{i,j}^2}{\lambda^2}\right) - \exp\left(-\frac{(r_{i,j})_{im}^2}{\lambda^2}\right)\right\} \tag{26}$$

and substituting Eq. (26) into the discretized form of the integral, the first term of Eq. (22) becomes

$$\zeta_1(x_m, y_n, t+\Delta t) = \frac{1}{\pi\lambda^2}\sum_i\sum_j \zeta(x_i, y_j, t)\left\{\exp\left(-\frac{r_{i,j}^2}{\lambda^2}\right) - \exp\left(-\frac{(r_{i,j})_{im}^2}{\lambda^2}\right)\right\}\Delta x\,\Delta y \tag{27a}$$

or

$$\zeta_1(x_m, y_n, t+\Delta t) = \sum_i\sum_j \gamma_{i,j}\cdot G(x_m, y_n, x_i, y_j, \Delta t). \tag{27b}$$

A similar expression can be obtained for the second term of Eq. (22).

We have noticed that the form of Eq. (27b) is similar to that of Eq. (7). The Green's function expressed in Eq. (26), however, is different from the Green's function expressed in Eq. (6).

2.2.2. *Criteria for Simplification of the Green's Function and Operation Count*

The operation count per time step to calculate Eq. (27a) or Eq. (27b), in principle, should be $O(N^2)$, where N is the number of mesh points. It looks as if the same troublesome problem arises here as in point vortex methods that employ the Biot–Savart law to determine the velocity from the vortices. Fortunately, there is an important and essential difference: namely, for Eq. (27a), the influence of the neighboring vortices falls away exponentially with distance-squared, instead of only algebraically in the case of the Biot–Savart law (more detail will be given in the next two subsections). The first term in Eq. (22) represents the diffusion contribution from all the vortices $\gamma_{i,j}$, outside of the flat plate, and the second term is that from all the image points of $\gamma_{i,j}$ inside the solid body (in the lower half-plane).

Setting $\Delta x \approx \Delta y \approx \lambda$ and then considering the only contribution to the points (x_m, y_n) by the vortex point at (x_i, y_j) with the strength $\gamma_{i,j}$, where $\gamma_{i,j} = \zeta(x_i, y_j, t)\lambda^2$ in the first term of Eq. (27), the fraction of $\gamma_{i,j}$ reaching point (x_m, y_n) after Δt is

$$\varepsilon = \frac{\zeta(x_m, y_n, t+\Delta t)}{\zeta(x_i, y_j, t)} = \frac{1}{\pi}\exp\left(-\frac{r_{i,j}^2}{\lambda^2}\right). \tag{28}$$

The value of ε decreases very rapidly as $r_{i,j}/\lambda$ increases. For example, if the vorticity is truncated at $3.9\times 10^{-5}\zeta(x_i, y_j, t)$ the value of $r_{i,j}/\lambda$ is equal to 3, i.e., the diffusion from the vortex blob at (x_i, y_j) can only reach the points which are located inside the circle with radius 3λ and centered at (x_i, y_j). In other words, if there are many vortex blobs coexisting on the (x, y) plane the vorticity at point (x_m, y_n) induced by diffusion is only related to those vortex blobs which are within the circle of radius 3λ and centered at point (x_m, y_n).

Therefore, for the general 2D case at high Reynolds numbers, the whole domain for diffusion simulation outside a solid body now can be divided into two regions: Region 1 is the "layer" with thickness of order $O(\lambda)$ (say, 3λ) surrounding the solid body (λ must satisfy $\lambda \ll R$, R being the radius of curvature of the solid surface) and Region 2 is the flow region outside the "layer," extending to infinity, as shown by Figs. 3a and 4.

Hence, the Green's function in Region 1 can be approximated by the superposition of the Green's function for a single vortex point outside a flat plate and its image point as long as $\lambda \ll R$, as shown by Figs. 3b and 4. The solid surface can be considered as a flat plate to those vortex points within the "layer" (e.g., typically, for $\Delta t = 0.01$ and $\mathrm{Re} = 10{,}000$, $\lambda = (4\,\Delta t/\mathrm{Re})^{1/2} = 2\cdot 10^{-3}$ which is indeed much smaller than $R \sim 1$). Therefore, the expression Eq. (26) will be modified further and used as the basis for the approximation of the Green's function in general 2D cases.

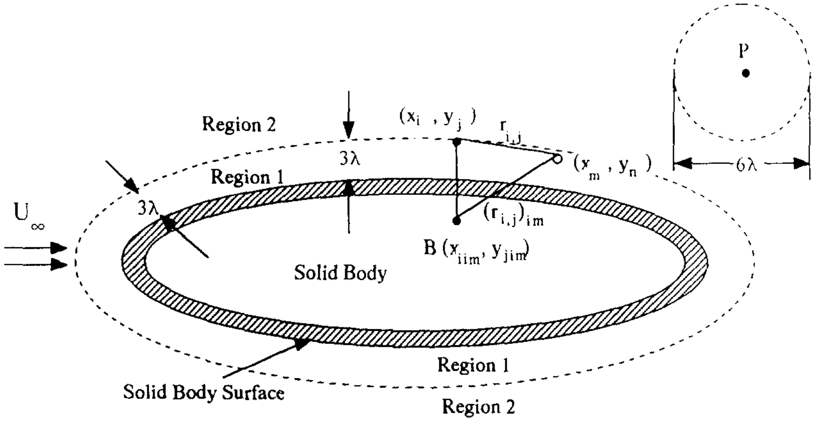

FIG. 4. Vortex point (x_i, y_j) within Region 1 and its image B inside the boundary. (The vortex point P with a diffusion circle in Region 2 for error $\varepsilon \leqslant 3.9 * 10^{-5}$.)

For the vortex points within Region 2, the Green's function is simply

$$G(x_m, y_n, x_i, y_j, t, \tau) = \frac{\text{Re}}{4\pi(t-\tau)} \exp\left(-\frac{r_{i,j}^2 \text{Re}}{4(t-\tau)}\right). \tag{29}$$

These points are far away (beyond $\sim 3\lambda$) from the solid boundary like point "p" on Fig. 4 and, thus, would not interact with a solid boundary in diffusion simulation and would behave like inviscid vortices. However, they can still have their diffusion contribution to their neighboring points within a radius $R \sim 3\lambda$, by behaving like a *viscous core.*

For small Δt and large Reynolds number Re, the vortices within the thin boundary layer are well represented, while those farther away would behave like inviscid vortices, thus also well represented. To verify the validity of the Green's function constructed above, the imulsively rotating cylinder problem has been chosen as a benchmark example for which an exact solution exists (results of this benchmark example are given below in Section 3.3).

2.3. *Simulation of the Convection Process*

To discretize Eq. (4) for convection, the velocity field is needed. After the vorticity diffusion at each mesh point (x_m, y_n) at time $t + \Delta t$ is calculated by Eq. (13), the velocity at each mesh point is determined via the stream function, by solving the discretized Poisson equation together with the proper boundary conditions. This choice over the Biot–Savart law is known to be computationally advantageous for large values of N. The usual three-point central difference approximation is written for time-level A,

$$\frac{\psi_{i-1,j}^{(A)} - 2\psi_{i,j}^{(A)} + \psi_{i+1,j}^{(A)}}{(\Delta x)^2} + \frac{\psi_{i,j-1}^{(A)} - 2\psi_{i,j}^{(A)} + \psi_{i,j+1}^{(A)}}{(\Delta y)^2} = -\zeta_{i,j}^{(A)}, \tag{30}$$

and the velocity field is evaluated similarly by a centered difference for u and v,

$$u_{i,j}^{(\Lambda)}=\frac{\psi_{i,j+1}^{(\Lambda)}-\psi_{i,j-1}^{(\Lambda)}}{2\,\Delta y} \tag{31a}$$

$$v_{i,j}^{(\Lambda)}=-\frac{\psi_{i+1,j}^{(\Lambda)}-\psi_{i-1,j}^{(\Lambda)}}{2\,\Delta x}. \tag{31b}$$

According to Eq. (4), the discretized vortex at (i, j) is to be displaced with the velocity, given by Eqs. (31a) and (31b), and by the time step Δt to reach time level $\Lambda+1$. The new position of the vortex at (i, j) at new step $\Lambda+1$ after convection is simply

$$x_{i,j}^{(\Lambda+1)}=x_{i,j}^{(\Lambda)}+u_{i,j}^{(\Lambda)}\cdot\Delta t \tag{32a}$$

$$y_{i,j}^{(\Lambda+1)}=y_{i,j}^{(\Lambda+1)}+v_{i,j}^{(\Lambda)}\cdot\Delta t. \tag{32b}$$

The points $(x_{i,j}^{(\Lambda+1)}, y_{i,j}^{(\Lambda+1)})$ are, of course, not at the mesh points of the fixed grid.

2.4. *Reinitiation of the Vorticity Field*

A common disadvantage of the pure Lagrangian tracking scheme is that the computational element would suffer from extreme distortion after a certain number of time steps, as pointed out by Van Dommenlem and Shen [40]. The distortion is even worse when the scheme is applied to the three-dimensional motion of the vortex filaments. To avoid this kind of distortion, a *smoothing procedure after each Lagrangian step* is introduced as follows.

The continuous vorticity at time level $\Lambda+1$ after convection is represented by point vortices, each of which possesses strength $\gamma_{i,j}=\zeta(x_{i,j}, y_{i,j}, t)h^2$ but is displaced from $(x_{i,j}^{(\Lambda)}, y_{i,j}^{(\Lambda)})$ to its new position $(x_{i,j}^{(\Lambda+1)}, y_{i,j}^{(\Lambda+1)})$ as shown by points A and A', respectively, in Fig. 5. The points $(x_{i,j}^{(\Lambda+1)}, y_{i,j}^{(\Lambda+1)})$ are, of course, not at the mesh points of the fixed grid. The distance AA' is obviously

$$AA'=[(x_{i,j}^{(\Lambda+1)}-x_{i,j}^{(\Lambda)})^2+(y_{i,j}^{(\Lambda+1)}-y_{i,j}^{(\Lambda)})^2]. \tag{33}$$

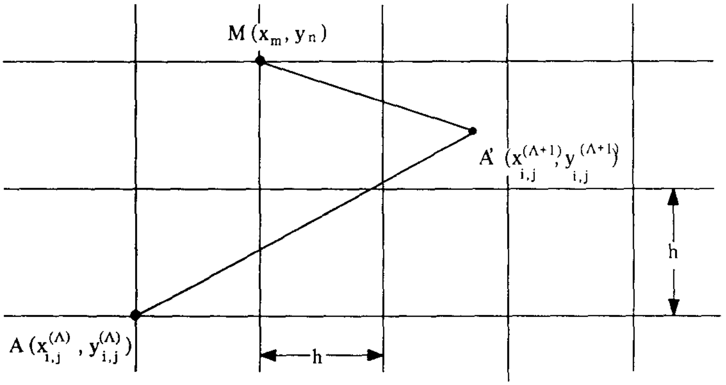
$M(x_m, y_n)$
A' $(x^{(\Lambda+1)}_{i,j}, y^{(\Lambda+1)}_{i,j})$
h
h
$A(x^{(\Lambda)}_{i,j}, y^{(\Lambda)}_{i,j})$

FIG. 5. Change of position of vortex point $A(x_{i,j}^{(\Lambda)}, y_{i,j}^{(\Lambda)})$ after convection.

We now calculate the diffusion simulation at time step $\Lambda+2$. The new vorticity at each mesh point (x_m, y_n) is the summation of the diffusion contribution from all the vortices, effectively those in a certain neighborhood of the point (x_m, y_n); e.g., expression Eq. (8) now becomes

$$(\zeta_{\mathrm{I}})^{(\Lambda+2)}_{m,n}=\frac{h^2}{\pi\lambda^2}\sum_i\sum_j\zeta^{(\Lambda+1)}_{i,j}\left\{\exp\left(-\frac{r^2_{i,j}}{\lambda^2}\right)-\exp\left(-\frac{(r_{i,j})^2_{im}}{\lambda^2}\right)\right\}, \tag{34}$$

where $r_{i,j}$ is no longer the distance between the regular mesh points, but is now represented by the line $A'M$, as shown in Fig. 5, i.e.,

$$r^2_{i,j}=[(x^{(\Lambda+1)}_{i,j}-x_m)^2+(y^{(\Lambda+1)}_{i,j}-y_n)^2]. \tag{35}$$

In this way, the vorticity value at each fixed nodal point at time step $\Lambda+2$ has been recovered.

It should be noted that the above procedure for recovering all the nodal values of vorticity is completely different from the Vortex-In-Cell method. For a typical Vortex-In-Cell method (see [24, 25]), the pointwise vortex A' in Fig. 6 is assumed to possess uniform vorticity within this cell. It contributes incremental vorticity $\zeta_{i,j}$ to each to the four mesh points at (i, j), $(i+1, j)$, $(i, j+1)$, and $(i+1, j+1)$ according to a certain interpolation scheme, as shown for one cell in Fig. 6,

$$\zeta_{i,j}=A_k\frac{\gamma_{A'}}{h^2} \qquad (k=1, 2, 3, 4), \tag{36}$$

where h is the mesh spacing and A_k's are the areas. The stength of point vortex A' is to be credited to four mesh nodes according to Eq. (36). After all the vorticity has been distributed among all the mesh points a finite difference form of the Poisson equation is solved. After velocity components are obtained, the velocity of the point vortex A' must be determined again by a bilinear interpolation in terms of an area-weighting technique in order to move point vortex A' for next time step. These two bilinear interpolation procedures could often introduce significant accumulative

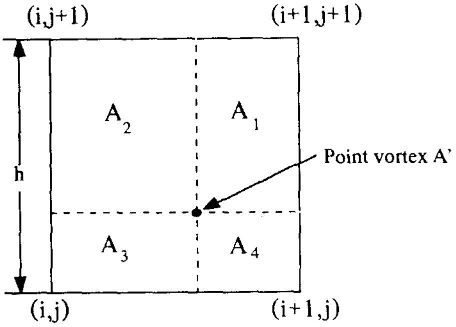

FIG. 6. Area weighting scheme for the vortex-in-cell method.

error into numerical results. Baker [25] calculated the roll-up of the vortex sheet and concluded that the angular momentum was not conserved, while the total circulation and linear impulse are conserved by the Vortex-in-Cell method. A similar conclusion was also drawn by Leonard [13] in his review paper. *In the new algorithm, however, the above two interpolation procedures are completely avoided.*

2.5. *Accuracy, Stability, and Efficiency Considerations*

The error due to diffusion simulation, i.e., the truncation error of using Eq. (8), to approximate Eq. (5), is of order $O(h^3)$ at each step. The truncation error for the scheme of the first-order accuracy, Eqs. (32a) and (32b), is $O(\Delta t^2)$ for each step. The total error for the two fractional steps Eqs. (3) and (4), following Chorin [21–23] and Marchuck [40], can be estimated approximately as the summation of the two errors. Thus, for the present square mesh scheme the cumulative error is

$$E_{\text{total}} \sim O(h^2) + O(\Delta t). \tag{37}$$

The validity of Eq. (37) will be confirmed in the subsequent numerical examples of this paper.

When the constraint Eq. (21) is followed, it requires $h/(4\,\Delta t/\mathrm{Re})^{1/2} < c_E$; c_E being a constant of order 1, depending on the error E. For a chosen value of h, the following value for Δt is produced:

$$\Delta t > (h/C_E)^2 \cdot \mathrm{Re}/4. \tag{38}$$

For example, if $\mathrm{Re} = 1000$, $h = 0.001$, and $C_E = 0.878$, we obtain $\Delta t > 0.324 \times 10^{-3}$ as the lower limit of Δt for proper diffusion simulation. There is no restriction on the maximum time step length Δt due to stability requirements. Thus, given the mesh length h, the value of Δt should be chosen relatively large and the estimated total truncation error of Eq. (37) becomes of the order $O(\Delta t^2)$.

The operation count per time step to calculate using either Eq. (8) or Eq. (27a), in principle, should be of the order $O(N^2)$, where N is the number of mesh points. However, as was mentioned before, according to Eq. (28), $r_{i,j}/\lambda$ is a function only of ε:

$r_{i,j}/\lambda$

	1.00	2.00	3.00	4.00
ε	1.2×10^{-1}	5.8×10^{-3}	3.9×10^{-5}	3.6×10^{-8}

This is simply the manifestation that the diffusion of each concentrated vortex is limited to within the diffusion distance $O(\lambda)$, as in Eq. (9).

Consequently, for a grid point (m, n), it is only necessary to take contributions from vortices whithin a circle of r_ε if ε is specified. For $\varepsilon = 3.9 \times 10^{-5}$, $r_{i,j}/\lambda = 3$ and the number of vortices contained in the circle may be estimated as

$$n_\varepsilon \sim \frac{\pi r_\varepsilon^2}{\lambda^2} = 28.2 < 29 \qquad \text{for} \quad \varepsilon = 3.9 \times 10^{-5}. \tag{39}$$

As an application for the square mesh scheme shown in Fig. 1, omitting all the mesh points beyond the layer $L = 4$ in the calculation of Eq. (12) would give a truncation error of the order of $O(3.6 \times 10^{-8} \cdot [32\zeta^*_{i,j}])$, where the factor 32 indicates that there are 32 mesh points along the layer $L = 4$ and $\zeta^*_{i,j}$ is the maximum vorticity at this time step.

Therefore, the total number of operations for diffusion simulation of Eq. (8) *or Eq.* (27a) *has nothing to do with the total number of vortex points* (*or mesh points*) *N. Instead, it is a fixed value n_ε which is a function of accuracy ε.*

Also note that Eq. (28) indicates that the total number of operations at each time step does not depend on the Reynolds number for a fixed value of the truncation error. On the other hand, the mesh length h should be kept within the order of the diffusion distance $\lambda = (4\,\Delta t/\mathrm{Re})^{1/2}$.

3. Computational Examples

3.1. *Decay of a Single Vortex with Finite Circular Core*

In order to test the performance of the present method, it has been applied to the problem of the decay of a single vortex with a finite circular cure. Milinazzo and Saffman [41] earlier used this problem in a critical examination of Chorin's random-vortex method. Subsequently, the same problem was chosen by Roberts [42] to reexamine the effect of non-smooth initial conditions on the accuracy of the random-vortex method. An important task is to solve the vorticity transport equation, Eq. (2), with the initial condition in an unbounded domain,

$$\begin{aligned} \zeta(x, y, t)|_{t=0} &= 1 \qquad \text{if} \quad x^2 + y^2 \leqslant 1 \\ \zeta(x, y, t)|_{t=0} &= 0 \qquad \text{otherwise.} \end{aligned} \tag{40}$$

The convection terms actually disappear for this particular problem because of symmetry. To test the validity of the present algorithm, however, it is instructive to avoid taking advantage of the special radial symmetry of the vorticity distribution, but use the complete vorticity transport equation, Eq. (2) instead of the pure diffusion equation, Eq. (3). For this particular example, for convenience, the nondimensional time expressed by Eq. (1) is redefined as

$$t^* = t/\mathrm{Re} \qquad \text{or} \qquad t = t^* \cdot \mathrm{Re}. \tag{41}$$

Substituting Eq. (41) into Eqs. (2) and (3) and then dropping the asterisk, a new nondimensional vorticity diffusion is obtained which does not contain the Reynolds number explicitly. The circular domain R is then divided by n concentric circles as shown in Fig. 7. Each circular ring is of the same thickness $\lambda = (4/\Delta t)^{1/2}$ and the total number of circular rings is $n = 1/\lambda$. The area of the pth ring is $S = (2p-1)\pi\lambda^2$ $(p = 1, 2, 3, ..., n)$. The coordinate of the center of the qth small area in the pth circular ring is obviously

$$x_{p,q} = r_p \cos(2\pi q/M_p); \qquad y_{p,q} = r_p \sin(2\pi q/M_p) \qquad (p = 1, 2, ..., n;\ q = 1, 2, ..., M_p),$$

where M_p indicates the integer part of $(2p-1)\pi$.

It must be stressed that the foregoing circular mesh structure is only used for the first step because the initial domain happens to be a circle. The actual mesh structure permanently used for subsequent computation is the simple square grid centered at the origin with spacing h in each coordinate direction, as in Fig. 1. Therefore there are two mesh systems overlapping each other in the first step. After the first step the ring mesh system is discarded.

The computational domain is defined by two finite constants x_∞ and y_∞. The total number of mesh units is $N_x \times N_y$, $N_x = 2x_\infty/h$ and $N_y = 2y_\infty/h$. The coordinates of mesh point (i, j) are $x_i = -x_\infty + i \cdot h$ and $y_j = -y_\infty + j \cdot h$, where i, j are integers $(1 \leqslant i \leqslant N_x$ and $1 \leqslant j \leqslant N_y)$. The initial stength of the vortex at (x_p, y_q) is $\gamma^{(0)}_{p,q} = (\lambda \cdot \lambda_p))\, \zeta^{(0)}_{p,q} = \lambda \cdot \lambda_p$. The vorticity at mesh point (x_i, y_j) after diffusion at the first time step $A = 1$ is

$$\zeta^{(1)}_{i,j} = \frac{1}{\pi\lambda} \sum_{p=1}^{n} \lambda_p \sum_{q=1}^{M_p} \exp\left(-\frac{r^2_{p,q}}{\lambda^2}\right), \tag{42}$$

where $r_{p,q} = [(x_p - x_i)^2 + (y_q - y_j)^2]^{1/2}$. As shown by Eq. (39), most of the operations in Eq. (42) are not needed. Thus Eq. (42) is rewritten as

$$\zeta^{(1)}_{i,j} = \frac{1}{\pi\lambda} \sum_{p=1}^{n_\varepsilon} \lambda_p \sum_{q=1}^{n_\varepsilon} \exp\left(-\frac{r^2_{p,q}}{\lambda^2}\right), \tag{43}$$

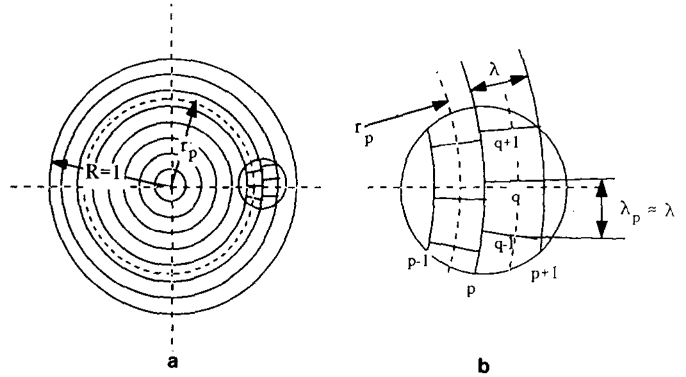

FIG. 7. Initial vorticity field for a circular vortex $R = 1$: (a) Domain divided by n concentric circles; (b) Enlarged mesh element.

where n_ε is defined by Eq. (39). The Poisson equation (30) can now be solved to obtain the velocity field for the updated vorticity distribution Eq. (43). The values of x_∞ and y_∞, however, have to be specified prior to solving the Poisson equation.

3.1.1. *Initial and Boundary Condition for the Stream Function*

The initial conditions consist of the solutions of the Poisson and the Laplace equation, inside and outside the circle, respectively, and both are equal to zero at $R = 1$,

$$\psi^{(0)} = (1/4)(1-r^2) \text{ for } r \leqslant 1; \qquad \psi^{(0)} = (-1/2)\log r \text{ for } r \geqslant 1. \tag{44a}$$

The fluid velocity is everywhere azimuthal,

$$u^{(0)} = -y/2; \quad v^{(0)} = x/2 \text{ for } r \leqslant 1; \quad u^{(0)} = -y/(2r^2); \quad v^{(0)} = x/(2r^2) \text{ for } r \geqslant 1 \tag{44b}$$

To solve the velocity field at step $\Lambda = 1$, the infinite domain need to be truncated at a certain large distance r_∞. The computation is to be carried our within the domain $x \leqslant r_\infty$ and $y \leqslant r_\infty$. The boundary value of the stream function is then determined as

$$\psi^{(\Lambda)} = \psi^{(0)} = -\tfrac{1}{2}\log r_\infty. \tag{45}$$

The Poisson equation with the initial vorticity field Eqs. (44a), (44b) and boundary condition Eq. (45) is solved to obtain velocity components $u_{i,j}^{(1)}$, $v_{i,j}^{(1)}$. Substituting $u_{i,j}^{(1)}$, and $v_{i,j}^{(1)}$ into Eqs. (32a) and (32b), the new position $(x_{i,j}^{(2)}, y_{i,j}^{(2)})$ is obtained, of the point vortex which was situated at nodal point $(x_{i,j}^{(1)}, y_{i,j}^{(1)})$ originally.

The next step is to solve for the vorticity field at step $\Lambda + 1 = 3$ according to the diffusion method expressed by the following expression,

$$\zeta_{m,n}^{(\Lambda+1)} = \frac{c^2}{\pi} \sum_i \sum_j \zeta_{i,j}^{(\Lambda)} \cdot \exp\left(-\frac{r_{i,j}^2}{\lambda^2}\right), \tag{46}$$

where c is given by Eq. (11).

The foregoing steps are repeated until the desired time step is reached.

3.1.2. *Computational Results and Conservativity*

The well-known integral invariants and a decay law for an arbitrary two-dimensional vorticity distribution are

$$\iint_D \xi \, dx \, dy = \Gamma_0 \tag{47a}$$

$$\frac{d}{dt} \iint_D \zeta r^2 \, dx \, dy = \frac{4}{\mathrm{Re}} \iint_D \zeta \, dx \, dy, \tag{47b}$$

where Γ_0 is the initial total circulation and D designates the entire flow field. To check the conservativity of the present method, we calculated the total circulation Γ at each time step for a single vortex with a circular core. A typical result for

Re = 5000 plotted in Fig. 8a has shown that the present method does satisfy the vorticity specified by Eq. (47a). Furthermore, using conservation relations (47a) and (47b), we can obtain the following angular-momentum relation,

$$A_e(t) = \frac{\int_{R^2} (x^2 + y^2)\, \zeta(x, y, t)\, dx\, dy}{\int_{R^2} \zeta(x, y, t)\, dx\, dy} = A_e(0) + \frac{4t}{\text{Re}}, \tag{48}$$

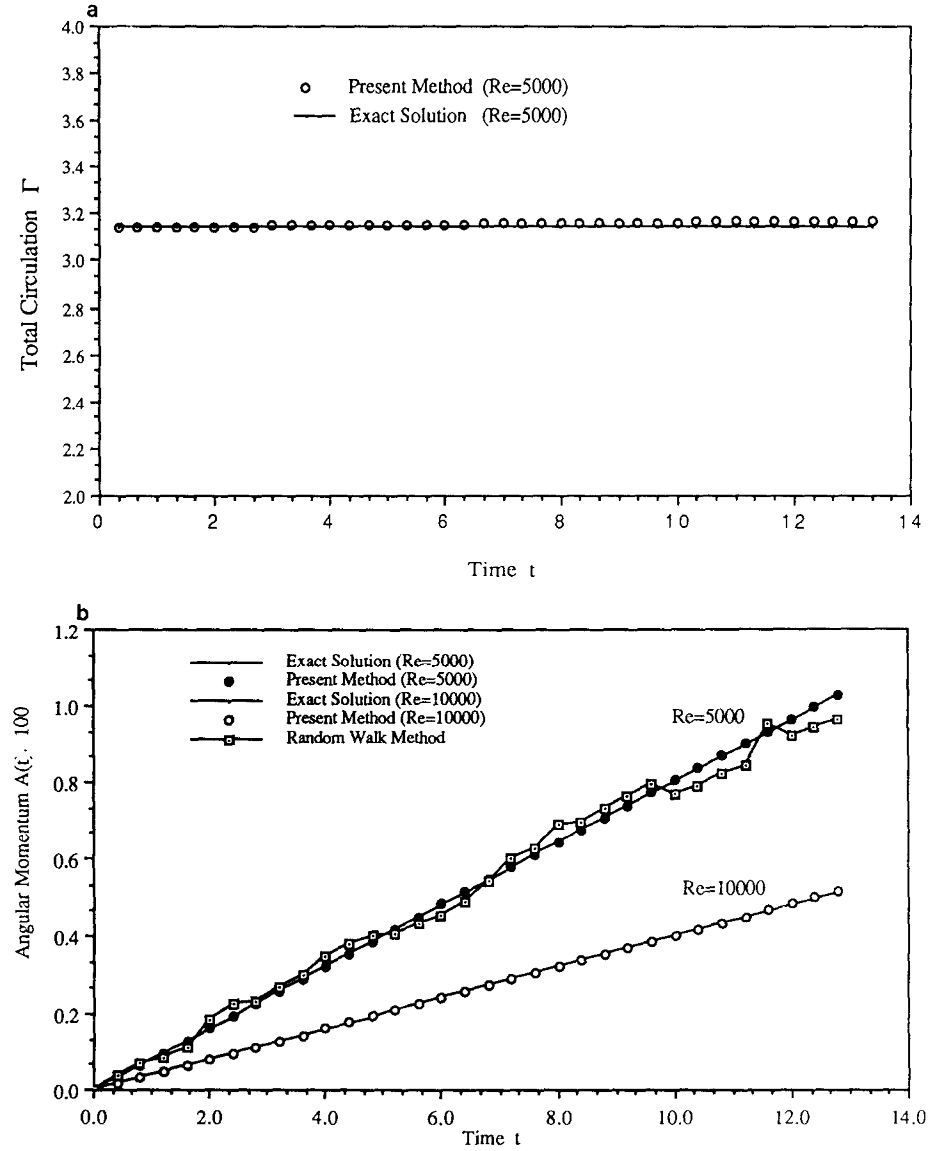

FIG. 8. (a) Evolution of total circulation Γ of single vortex with finite core ($R = 1$); (b) evolution of angular momentum of single vortex with finite core ($R = 1$).

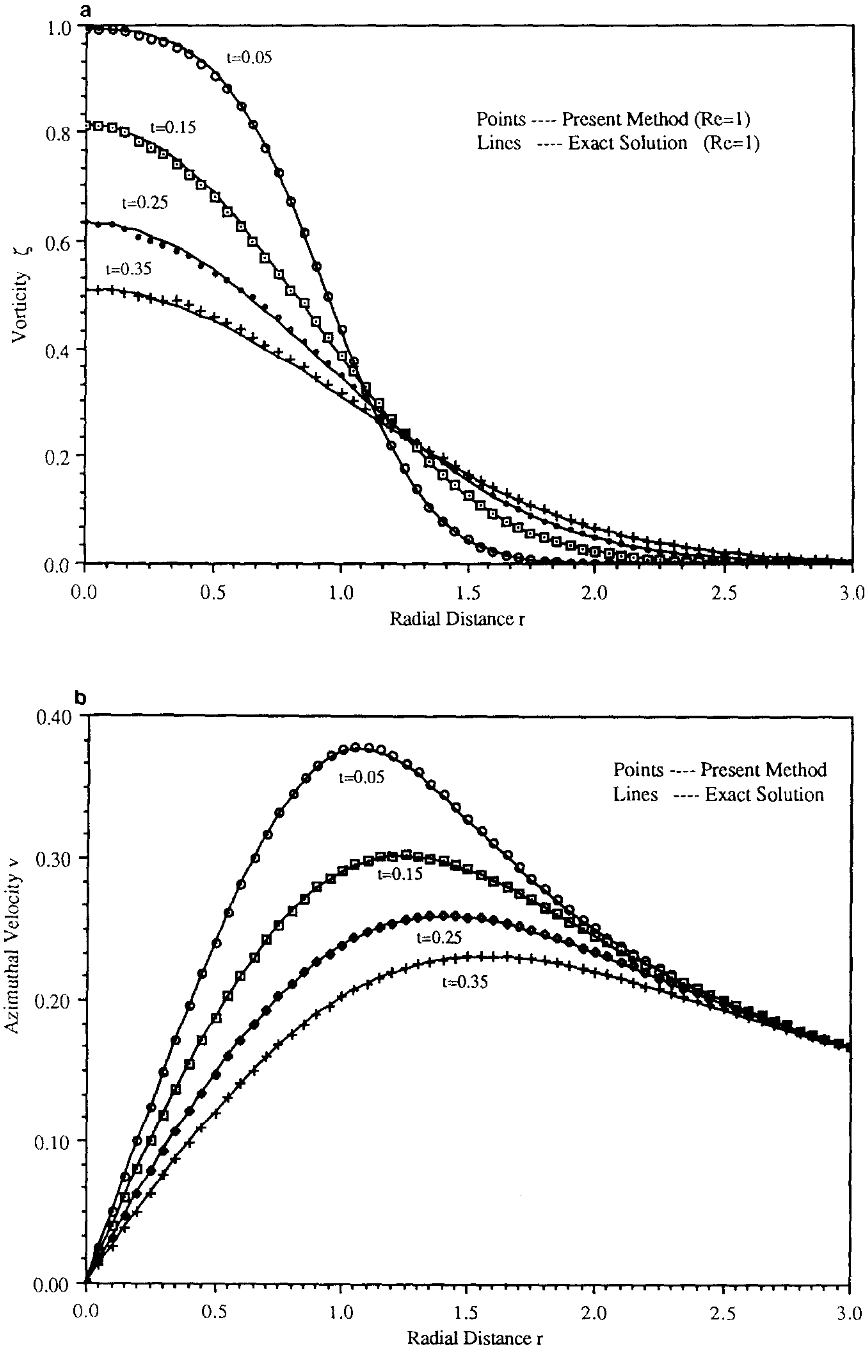

FIG. 9. (a) Vorticity field for single vortex of finite core $R = 1$ (non-dimensional time is scaled as $t = tU/(L \cdot \mathrm{Re}) = t \cdot \nu/L^2$); (b) azimuthal velocity field for single vortex of finite core $R = 1$.

where $A_e(0)$ is the initial value of $A_e(t)$, in this case equal to $\frac{1}{2}$, and R^2 denotes the non-zero vorticity region. *The conservativity relations* (47a), (47b), *and* (48) *are used to check the accuracy of the diffusing-vortex mthod.*

Figures 8b, 9a, and 9b display the computed value of the angular-momentum integral $A(t)$ (as was defined by [42]), the vorticity field, and the velocity field, respectively. The computation is carried out at $\Delta t = 0.05$, $h = 0.1$, for $t < 1.5$, $h = 0.2$ for $t > 1.5$ and $x_\infty = y_\infty = 10$ for Reynolds number $\mathrm{Re} = 1$, and $\Delta t = 0.05$, $h = 0.006$, $x_\infty = y_\infty = 3$, for $\mathrm{Re} = 5000$. For the discretized version, the angular-momentum integral is

$$A(t) = \frac{\sum_i \sum_j \zeta_{i,j}^{(A)}(x_i^2 + y_j^2)}{\sum_i \sum_j \zeta_{i,j}^{(A)}}. \tag{49}$$

The relative error is defined as

$$e(T) = \frac{|A(T) - A_e(T)|}{A_e(T)}. \tag{50}$$

Figure 8 compares the angular momentum calculated by different methods and the exact solution. The calculation was carried until $T = 4\pi$, which is the initial rotation time of the vortex.

In the random-vortex method Roberts [43] obtained his solution with $e(T) = 0.453\%$ for Reynolds number $\mathrm{Re}_d = 10{,}000$ ($\mathrm{Re} = \mathrm{Re}_r = 5000$). The value of $e(T)$ obtained by the present method in 0.22% for $\mathrm{Re} = 5000$. The random vortex calculation was also performed. The advantage of the new algorithm over the random vortex method is very obvious as shown by Fig. 8 and Table I. The CPU time is for the same computer (IBM3090-600E), so the comparison is valid. For the random vortex method, the nonzero vorticity core was discretized into 10,000 vortex points to maintain the comparable relative error $e(T)$ with the diffusing-vortex method.

However, comparison in integral quantities such as the angular momentum is not conclusive in assessing accuracy. *It is more convincing when the instantaneous vorticity and velocity computed by the new diffusing-votex method are found to match very accurately with the exact solution, as shown in Figs.* 9a *and* 9b, *even for low*

TABLE I

Comparison of Angular Momentum for a Circular Vortex

Method	$e(T)$ (%)	CPU (min)
Random vortex	0.52	47
New diffusing-vortex	0.22	21

Note. 10, 201 mesh points for the diffusing-vortex method; 10,000 vortices for the random vortex method.

Reynolds number Re = 1. There were neither instantaneous velocity distributions nor vorticity distributions published by both Roberts [43] and Milinazzo and Saffman [42].

3.2. *Decay of Vortex Pairs of Two Finite-Core Regions in Proximity*

The decay of the vortex pair of two finite cirular cores with both positive and negative sign have been calculated to further confirm the validity of our algorithm. This calculation is based on the initial conditions and boundary conditions for the stream function. Results of these calculations are described here.

3.2.1. *Initial and Boundary Conditions for the Stream Function*

In order to examine further the new method with non-trivial convective terms, we next solve the evolution of an initial vortex pair, each of them having a finite core.

The nondimensional initial condition is

$$\zeta(x, y, t)|_{t=0} = 1 \quad \text{for} \quad (x-x_c)^2 + y^2 \leqslant R_0^2 \text{ or } (x+x_c)^2 + y^2 \leqslant R_0^2 \qquad (51a)$$

$$\zeta(x, y, t)|_{t=0} = 0 \quad \text{otherwise.} \qquad (51b)$$

There is no analytic solution for the instantaneous velocity and vorticity for this problem; The square mesh structure used in example A is again used here. Following example A, the initial two circular domains are divided into n concentric circular rings, each of which possesses the same thickness $\lambda = (4\,\Delta t/\text{Re})^{1/2}$. The procedures here are similar to that for example A. The schematic is shown by Fig. 10. To solve the Poisson equation, the boundary condition for the stream function at the farfield, which can be described approximately as

$$\psi(x, y) = \frac{1}{[illegible]} \iint \log[(x-x')^2 + (y-y')^2]^{1/2}\, dx'\, dy' \qquad (52)$$

3.2.2. *Computational Results and Conservativity*

The calculation is carried out for Eqs. (2) and (3) without using the transformation of Eq. (41). The angular momentum for each time step is calculated using Eq. (49). In Table II, the relative error defined by Eq. (50) and the computing time for each case are listed as a function of the mesh length h for Re = 5000, $\Delta t = 0.08$, $R_0 = 4h$, $x_c = 8h$, and fixed farfield boundaries $x_\infty = y_\infty = 0.4$.

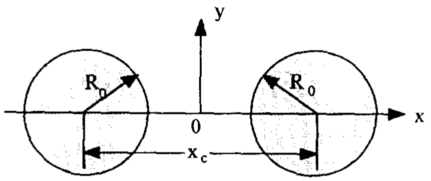

FIG. 10. Initial vortex pair at $t = 0$.

TABLE II

Effect of Mesh Length on Relative Error of Angular Momentum for the Vortex Pair Problem at a Fixed Value of Δt

h	0.889×10^{-2}	0.800×10^{-2}	0.635×10^{-2}	0.494×10^{-2}	0.400×10^{-2}
$c = h/\lambda$	1.111	1.000	0.794	0.618	0.500
$N_x \times N_y$	91×91	101×101	126×126	$162 = \times 162$	201×201
$e(T)$	2.88%	0.41%	0.26%	0.18%	0.15%
CPU	17.1 min	23.1 min	60.5 min	138.6 min	238.5 min

Note. $\mathrm{Re} = 5000$, $\Delta t = 0.08$, $\lambda = 0.00775$, $x_\infty = y_\infty = 0.4000$.

We notice that the conservativity relation (48) for a single vortex is also valid for the present example. The evolution of $A(t) - A_0$ with time t for $\mathrm{Re} = 1000$, 3000, 5000, and 7000 is also shown in Fig. 11. The results were obtained with $h = 0.00635$, $\Delta t = 0.08$, $N_x \times N_y = 101 \times 101$, $x_\infty = y_\infty = 0.4000$ and each calculation required about 24 min of CPU time. The computation was done with an IBM3090-600E (a 64-bit supercomputer at Cornell University).

The results in Table II show that the mesh length parameter c plays an important role for both the computing time and, especially, the relative error. The value of the relative error $e(T)$ for $c > 1.1$ is much higher than that for $c < 1$ as shown in Fig. 12a. However, it is interesting to note that the value of $e(T)$ does not decrease very much while c decreases further from $c = 0.7940$ to 0.5000 (correspondingly, h

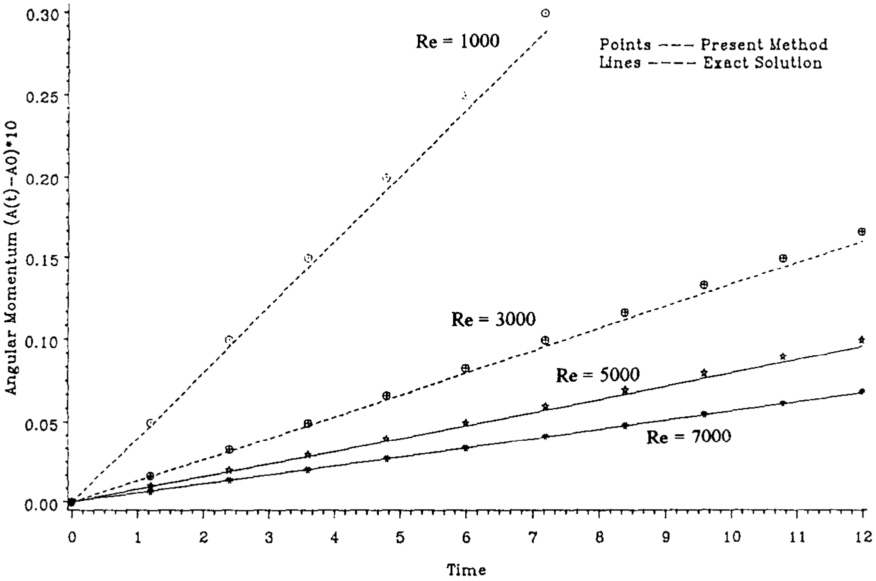

FIG. 11. Angular momentum vs time for different Reynolds numbers (a vortex pair with same positive sign).

decreases from 0.00635 to 0.004 in this interval). The reason is that the truncation error of the convection simulation is of the order $O(\Delta t^2) = O(0.08^2) = O(6 \times 10^{-3})$ and dominates over the error of the diffusion simulation, which is of the order $O(h^3) = O(0.005^3) = O(10^{-7})$. In order to chack the estimate of Eq. (37), $e(T)$ is computed for different values of h under the condition $\mathrm{Re} = 5000$ and $c = 1$, i.e., $h = \lambda = (4\,\Delta t/\mathrm{Re})^{1/2}$. According to Eq. (37), the error estimate is $O(\Delta t)$. The behavior in Fig. 12b shows that the calculated results do follow the trend.

Figures 13a–d show the evolution of the vorticity field using three-dimensional graphics at Re = 5000. The instantaneous stream line patterns are plotted in Figs. 14a–d. Inirially the vortex pair was situated on the x-axis with a separation distance $x_c = 8h$. As time increases two cores with both positive vorticity gradually approach each other and finally merge into one core and the long axis of the core is no longer coincident with the x-axis after a few time steps. Lo and Ting [43] published their research results for low Reynolds number (less than 100) for this problem, using asymptotic expansion methods, but they did not give the instantaneous flow field.

3.3. *Unsteady Flow Field outside a Rotating Cylinder*

The flow field induced by an impulsively rotating cylinder with radius $R = 1$ and constant angular velocity ω is now studied. The definition of the nondimensional quantity in Eq. (1) is still used, setting characteristic length $L_c = R$ and characteristic velocity $U = \omega R$, then

$$\begin{gathered} u = \omega R\bar{u}, \qquad v = \omega R\bar{v}, \qquad x = R\bar{x} \\ y = R\bar{y}, \qquad r = R\bar{r}, \qquad t = \frac{\bar{t}}{\omega}, \qquad \zeta = \omega\bar{\zeta}. \end{gathered} \tag{53}$$

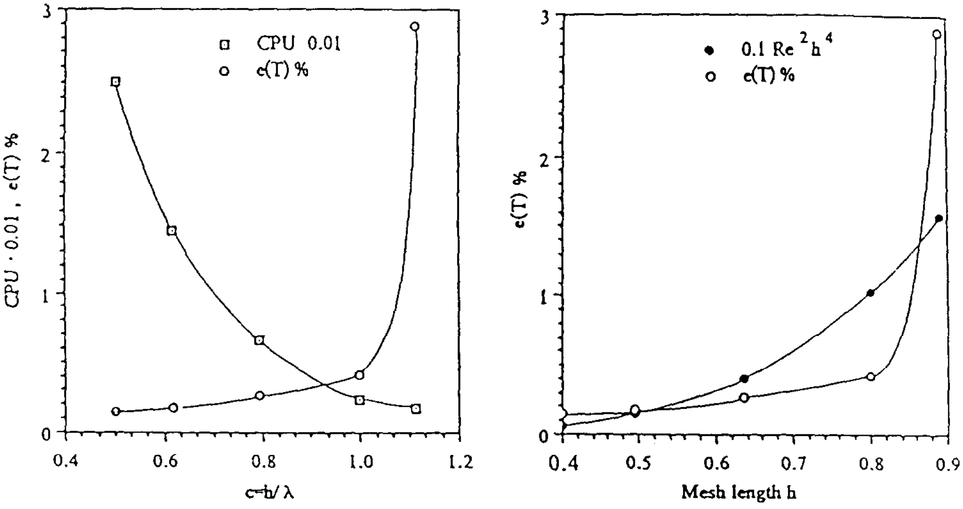

FIG. 12. (a) Effect of mesh length on relative error of angular momentum and CPU for a vortex pair with both positive sign; (b) Comparison between $e(T)$ and a curve $y = 0.1\,\mathrm{Re}^2 h^4$.

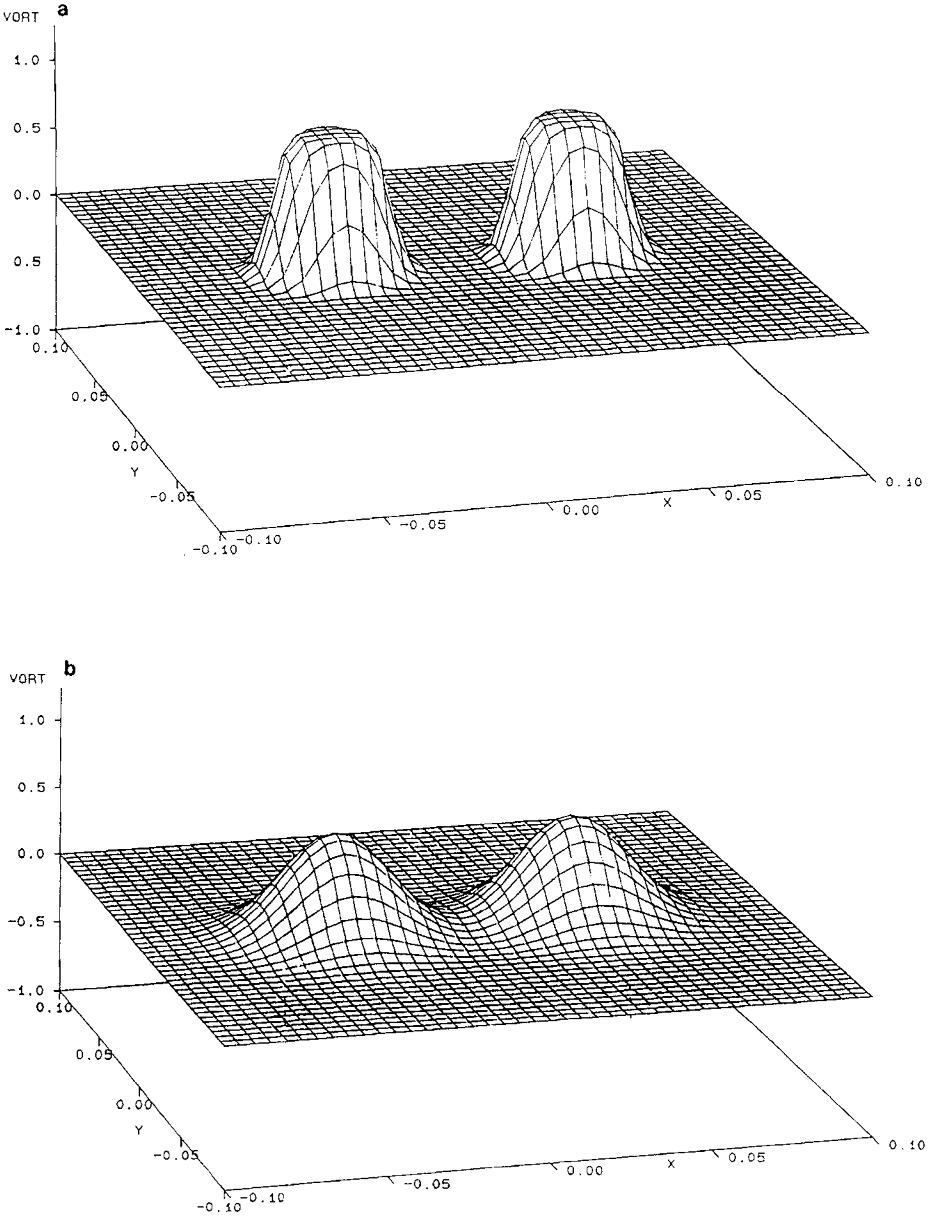

FIG. 13. Vorticity field of a vortex pair with same sign (Re = 5000): (a) time = 0.04; (b) time = 0.40; (c) time = 1.40; (d) time = 2.40.

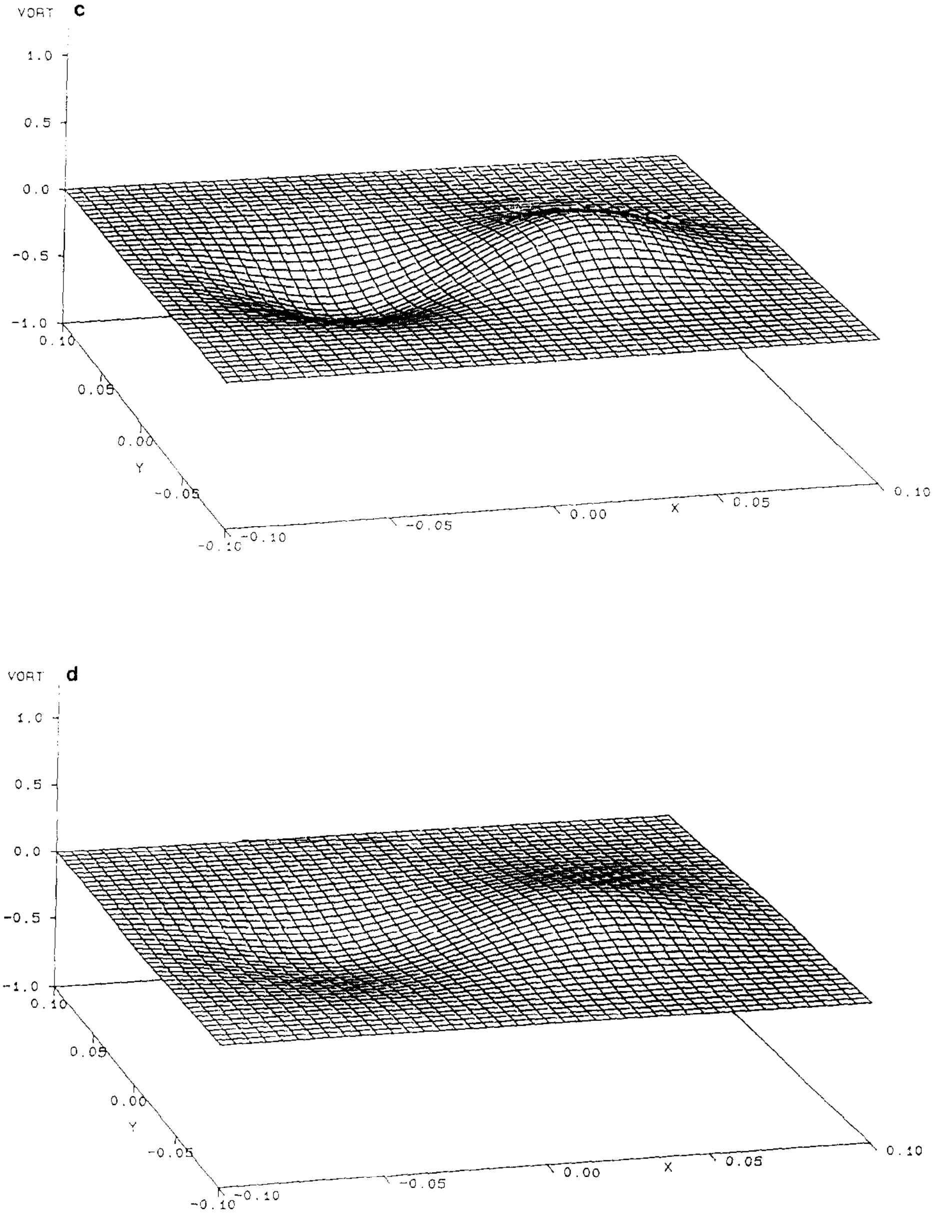

FIG. 13—*Continued*

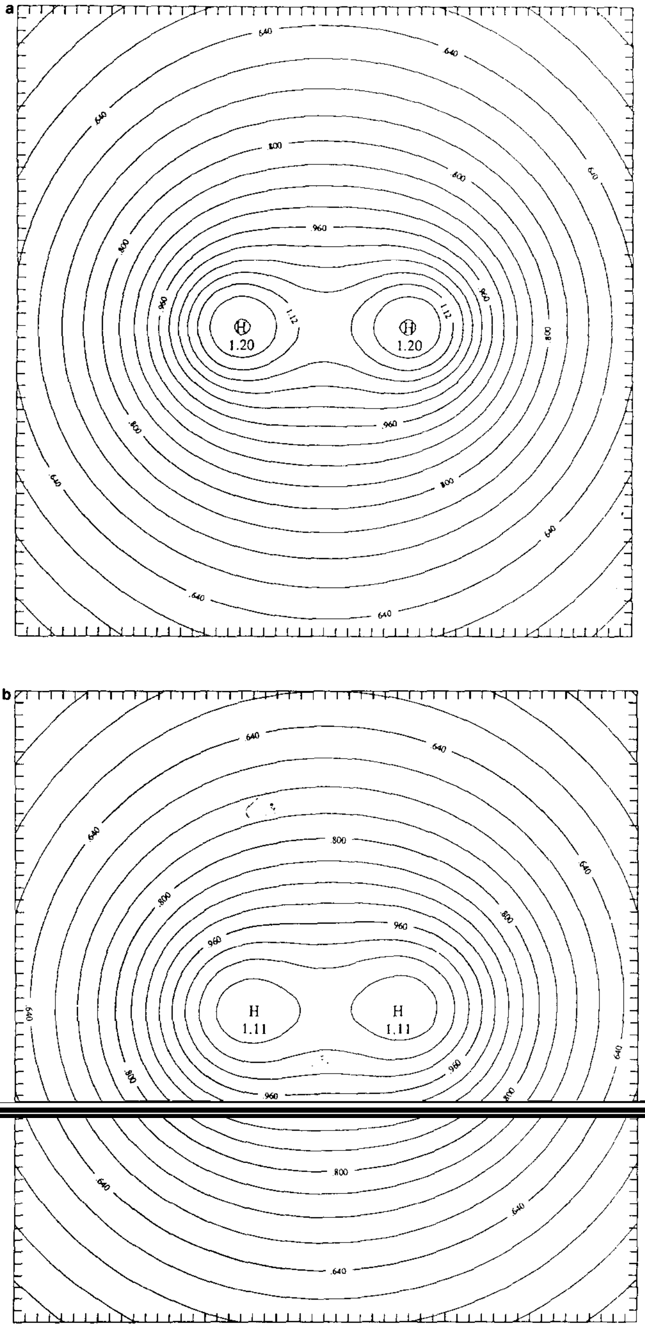

FIG. 14. Instantaneous stream function for a vortex pair with both positive sign: (a) time = 0.04; (b) time = 0.40; (c) time = 1.40; (d) time = 2.40.

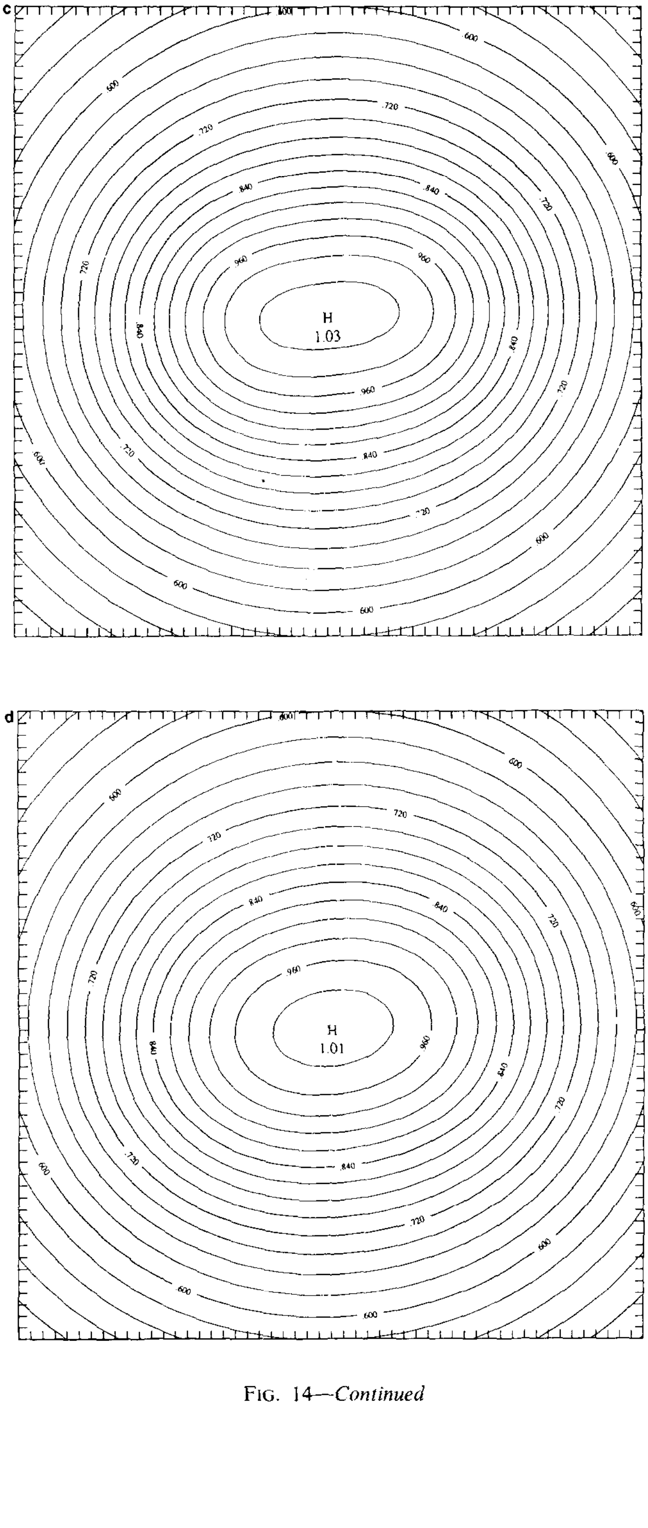

FIG. 14—*Continued*

A solution is needed for the vorticitiy transport equation, Eq. (2), under the boundary condition, dropping the bar, and using the polar coordinate system (r, θ),

$$v_\theta = 1 \text{ for } r = 1; \qquad v_\theta = 0 \text{ for } r = \infty. \tag{54}$$

3.3.1. *Initial Condition*

One of the difficulties in this problem is the specification of the initial conditions. For small time, the vorticity is initially infinite and confined to an infinitesimally thin region surrounding the cylindrical surface. Thus the direct application of a finite difference approximation to the equation cannot give the initial flow field correctly. But the Rayleigh solution is a good approximation. The velocity and vorticity at each mesh point (r_m, θ_n) are

$$v_\theta(r_m, \theta_n, t_0) = 1 - \frac{2}{(\pi)^{1/2}} \int_0^{\Delta_m} \exp(-x^2)\, dx \tag{55}$$

$$\zeta(r_m, \theta_n, t_0) = -\left(\frac{\text{Re}}{\pi t_0}\right)^{1/2} \exp(-\Delta_m^2), \tag{56}$$

where $\Delta_m = (\text{Re}/4t_0)^{1/2}(r_m - 1)$.

3.3.2. *Diffusion Simulation*

The counterpart of Eq. (27) under the polar coordinate system is

$$\zeta_1(r_m, \theta_n, t + \Delta t) = \frac{1}{\pi\lambda^2} \sum_i \sum_j \zeta(r_i, \theta_j, t) \left[\exp\left(-\frac{r_{i,j}^2}{\lambda^2}\right) - \exp\left(-\frac{(r_{i,j})_{\text{img}}^2}{\lambda^2}\right)\right] \Delta\sigma \tag{57a}$$

or

$$\zeta_1(r_m, \theta_n, t + \Delta t) = \sum_i \sum_j \gamma_{i,j} \left[\frac{1}{\pi\lambda^2} \exp\left(-\frac{r_{i,j}^2}{\lambda^2}\right) - \frac{1}{\pi\lambda^2} \exp\left(-\frac{(r_{i,j})_{\text{img}}^2}{\lambda^2}\right)\right], \tag{57b}$$

where $\gamma_{i,j}$ is the strength or small circulation of the vortex at (r_i, θ_j),

$$\gamma_{i,j} = \zeta(r_i, \theta_j, t)\, \Delta\sigma; \qquad \Delta\sigma = r_{i,j}(\Delta r)_i(\Delta\theta)_j, \tag{58}$$

where $r_{i,j}$ is the distance between the vortex point at (r_i, θ_j) and the calculated mesh point (r_m, θ_n),

$$r_{i,j} = [(r_i \cos\theta_j - r_m \cos\theta_n)^2 + (r_i \sin\theta_j - r_m \sin\theta_n)^2]^{1/2} \tag{59}$$

and $(r_{i,j})_{\text{img}}$ is the distance between the point (r_m, θ_n) and the point $(1/\gamma_i, \theta_j)$ inside the circular cylinder which is the image point of (r_i, θ_j),

$$(r_{i,j})_{\text{img}} = \{[\gamma_i^{-1} \cos\theta_j - r_m \cos\theta_n]^2 + [\gamma_i^{-1} \sin\theta_j - r_m \sin\theta_n]^2\}^{1/2}. \tag{60}$$

For those vortices far away from a solid surface (for instance, beyond $r \sim 3\lambda$), the vortex $\gamma_{i,j}$ would only carry a finite core of the Gaussian distribution, i.e., Eq. (57b) would degenerate to

$$\zeta_{\mathrm{I}}(r_m, \theta_n, t+\Delta t) = \sum_i \sum_j \gamma_{i,j} \cdot \left[\frac{1}{\pi\lambda^2} \exp\left(-\frac{r_{i,j}^2}{\lambda^2} \right) \right]. \tag{61a}$$

The second term of Eq. (22) represents the diffusion contribution from the vorticity on the cylindrical surface and can be integrated. For a solid boundary of arbitrary shape, the factor in the second term of Eq. (22) can be approximated as

$$\left(\frac{\partial G}{\partial n} \right)_b = \frac{G - G_{\mathrm{img}}}{2\Delta}, \tag{61b}$$

where $\Delta = r - R$ is the normal distance of the calculated point to the solid boundary. If the surface of the solid boundary is divided into m equal pieces and the length of each of the pieces is Δs, the second term of Eq. (22) can be written as

$$\zeta_{\mathrm{II}}(r_m, \theta_n, t+\Delta t) = -\frac{1}{\mathrm{Re}} \zeta_b \left(\frac{G - G_{\mathrm{img}}}{2\Delta} \right) \Delta s. \tag{61c}$$

The Green's function in Eq. (61b) is determined by Eq. (26) and the value of G_{img} is taken from point A_{img} which is the image point of point A. As analyzed before, the number of operations surrounding point A is very small since the value of G decreases exponentially with the distance squared. Substituting Eqs. (61a) and (61c) into Eq. (22), the vorticity field at $t + \Delta t$ is then

$$\zeta(r_m, \theta_n, t+\Delta t) = \sum_i \sum_j \gamma_{i,j} G(r_m, \theta_n, r_i, \theta_j) - \frac{1}{\mathrm{Re}} \Delta t \sum_{k=1}^{M} \zeta_b \left(\frac{G - G_{\mathrm{img}}}{2\Delta} \right) \Delta s, \tag{62}$$

where Green's function in the cylindrical coordinate system is

$$G(r_m, \theta_n, r_i, \theta_j) = \frac{1}{\pi\lambda^2} \left[\exp\left(\frac{(r_{i,j})^2}{\lambda^2} \right) - \exp\left(-\frac{(r_{i,j})^2_{\mathrm{img}}}{\lambda^2} \right) \right] \tag{63}$$

and $\gamma_{i,j}$, $r_{i,j}$, and $(r_{i,j})_{\mathrm{img}}$ are given by Eqs. (58), (59), and (60).

3.3.3. *Boundary Condition–Vorticity Value on Cylinder Surface for the Next Time Step*

It has been well recognized that the most challenging task for directly solving the vorticity transport equation lies in how to determine the vorticity value on solid boundary surfaces. In practice the boundary condition is usually given in terms of the velocity (no slip condition) and not vorticity. This is because the vorticity value on the solid surfaces is often very difficult to calculate. With our diffusing-votex algorithm, fortunately, this difficulty can be overcome. In our rotating cylinder problem, we do not need to use the known solution on the cylinder surface as the boundary valued of the vorticity at each time step. Instead, the vorticity value on the cylinder surface is obtained by directly substituting $r_m = 1$ into Eq. (62) for the next time step. We should note that our method does not easily deal with velocity boundary conditions, since our Navier–Stokes equations are written in vorticity form and not velocity form.

3.3.4. *Convection Simulation*

The new position of the vortex (i, j) after the convection step under the polar coordinate system is then defined by

$$r_{i,j}^{(A+1)} = r_{i,j}^{(A)} + (v_\xi)_{i,j}^{(A)} \cdot \Delta t \tag{64a}$$

$$\theta_{i,j}^{(A+1)} = \theta_{i,j}^{(A)} + (v_\theta)_{i,j}^{(A)} \cdot \frac{\Delta t}{r_{i,j}}, \tag{64b}$$

where the superscript A indicates the time step, i.e., $t = A \cdot \Delta t$. While the calculations with Eqs. (64a) and (64b) are performed, the continuous vorticity field is simply replaced by many point vortices, each of which possesses circulation $\zeta(r_i, \theta_j, t)\, \Delta\sigma$, and then each point vortex at $[r_{i,j}^{(A)}, \theta_{i,j}^{(A)}]$ is allowed to move to its new position $[r_{i,j}^{(A+1}, \theta_{i,j}^{(A+1)}]$ which is not at nodal points, in general. Note that the single subscript, as in r_i, θ_j, designates the mesh point, and the coordinates with double subscripts and single superscript A, as in $r_{i,j}^{(A)}$, $\theta_{i,j}^{(A)}$, represent the position of each discretized vortex point under the pollar system.

Following the whole procedure for reinitiation of the vorticity field, the diffusion contribution to mesh point $M(r_m, \theta_n)$ from the vortex $A'[r_{i,j}^{(A+1)}, \theta_{i,j}^{(A+1)}]$ is only related to the distance between them, i.e., $A'M$ as plotted on Fig. 15:

$$A'M = [(r_{i,j}^{(A+1)} \cos \theta_{i,j}^{(A+1)} - r_m \cos \theta_n)^2 + (r_{i,j}^{(A+1)} \sin \theta_{i,j}^{(A+1)} - r_m \sin \theta_n)^2]^{1/2}. \tag{65}$$

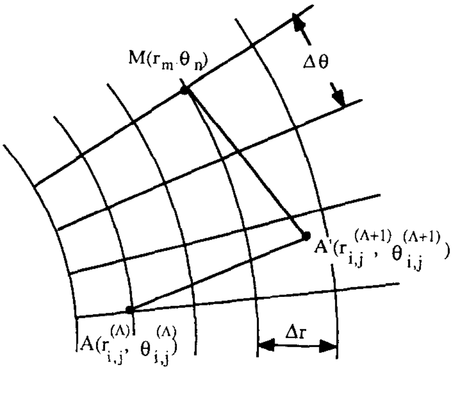
$M(r_m, \theta_n)$
$\Delta\theta$
$A'(r_{i,j}^{(A+1)}, \theta_{i,j}^{(A+1)})$
$A(r_{i,j}^{(A)}, \theta_{i,j}^{(A)})$
Δr

FIG. 15. Point vortex A at time t moves to its new position A' at time $t + \Delta t$.

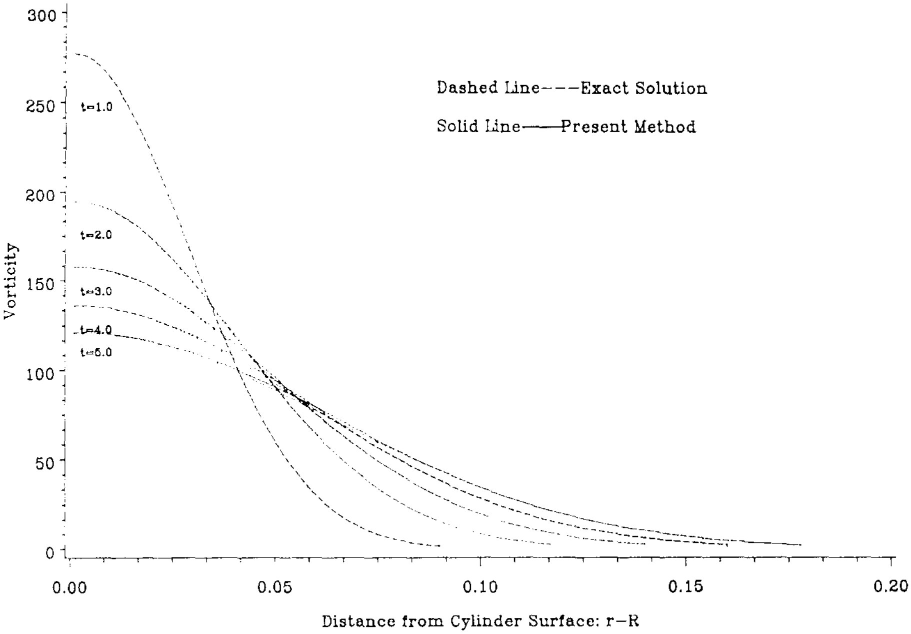

FIG. 16. Unsteady vorticy field induced by a rotating cylinder at Re = 5000.

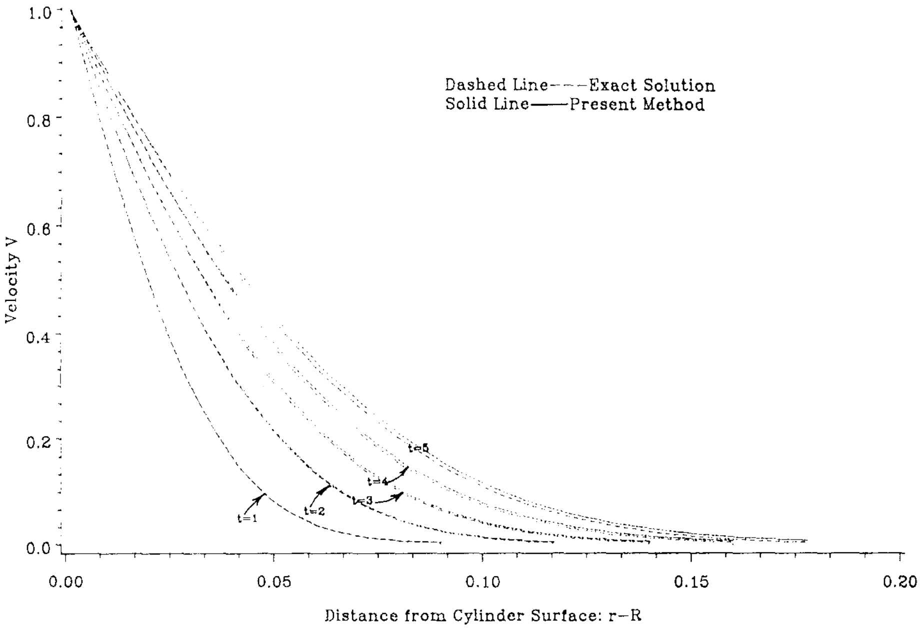

FIG. 17. Unsteady velocity field induced by a rotating cylinder at Re = 5000.

The new vorticity at each mesh point (r_m, θ_n) is the summation of the diffusion contribution from all the vortices, effectively those in a certain neighborhood of the point (r_m, θ_n). The procedure for recovering the vorticity value at each fixed nodal point at time step $\Lambda + 2$ is the same as that described before.

The calculation for this problem started at $t_0 = 0.01$, using the Rayleigh solution. The Reynolds numbers chosen were $\mathrm{Re}_d = UD/\nu = 10{,}000$ (i.e., $\mathrm{Re} = \mathrm{Re}_r = 5000$). The computational parameters were $\Delta t = 0.04$ and $\lambda = 0.00566$, and $\Delta\theta = \Delta\xi = 0.005$ for the transformed polar system $\xi = \log r$. The unsteady vorticity field and velocity field are shown in Figs. 16 and 17. The difference between the new method and the exact solution increases with time. These results have demonstrated that the new diffusing-vortex algorithm procedures solutions in excellent agreement with the exact solutions.

4. Conclusions and Discussion

A new numerical solution algorithm, the diffusing-votex method, has been developed to study two-dimensional incompressible flow at high Reynolds number. The algorithm discretizes the vorticity field and splits the transport equation into two fractional steps: diffusion and convection. It may be natural to compare this method with Chorin's random-vortex method. The essential differences are: (1) the diffusion process for each time step is simulated by the exact solution of the diffusion equation in an unbounded domain, instead of by a random walk approximation. The new algorithm uses the local value of the vorticity field at fixed mesh points for the discretized vortex strength. The discretized vortices do not maintain the same vortex stength as required by a random walk algorithm. The number of operations for the diffusion calculation for our diffusing-vortex method is no more than n_ε (although the total number of operations for the random vortex method is N^2, it is considered one of the most efficient methods for diffusion simulation among the conventional vortex methods); (2) the velocity field is determined for the same fixed mesh points so that it is possible to use a high-order convection scheme without interpolation. The convection term, which is troublesome for typical finite difference schemes at high Reynolds number, is simulated by tracking the motion of each discretized vortex point. But *the new scheme is Lagrangian for only one time step, through the use of new particles at fixed mesh points for the next time step.*

The new diffusing-vortex method has demonstrated great potential for extension to three-dimensional (3D) calculations, based on the following two issues. First, the computational examples have shown that the CPU time is of the order of $N(\log N)$ instead of N^2 for typical vortex methods (N is the total number of vortices). Second, there is no restriction on the maximum value of the time step length imposed by stability. Instead, there is a lower liimit of Δt for a proper diffusion simulation, i.e., $\Delta t > (h/C_E)^2 \cdot \mathrm{Re}/4$. In terms of efficiency, our method may be limited in 3D when compared to grid-free methods for some problems. This issue is currently under investigation.

ACKNOWLEDGMENTS

This work was fully supported under National Science Foundation Award ISI-8760931 and is gratefully acknowledged. The authors are also grateful to the Cornell University Theory Center for providing computational resources and advice during the research period. Finally, the authors would like to express their gratitude to Professor S. F. Shen for his advice and his earlier contributions in this field.

REFERENCES

1. Z. Y. LU AND S. F. SHEN, *Numerical Methods in Laminar and Turbulent Flow, Vol. 5*, edited by C. Taylor, *et al.* (Pineridge Press, Swansea, UK, 1987), Part 1, p. 619.
2. F. H. HARLOW, Los Alamos Scientific Laboratory Report No. La-4281, 1969 (unpublished).
3. P. J. ROACHE, *Computational Fluid Dynamics* (Hermosa, Albuquerque, NM 1972), p. 351.
4. N. S. WILKES, C. P. THOMPSON, J. R. KIGHTLEY, I. P. JONES, AND A. D. BURNS, *Numerical Methods in Laminar and Turbulent Flow*, 1984, Vol. 4.
5. D. C. THOMAN AND A. A. SZEWCZYK, *Phys. Fluids Suppl.* **12(II)**, 76 (1969).
6. R. D. RICHTMYER AND K. W. MORTON, *Difference Methods for Initial Value Problems*, 2nd ed. (Wiley, New York, 1967).
7. W. M. COLLINS AND S. C. R. DENNIS, *J. Fluid Mech.* **60**, Part 1, 105 (1973b).
8. S. S. WEI AND S. I. GHCERI, *Numerical Methods in Laminar and Turbulent Flow, Vol. 5*, edited by C. Taylor, *et al.* (Pineridge Press, Swansea, UK, 1987), Part 1, p. 693.
9. D. KWAK AND S. R. CHAKRAVARTHY, *AIAA J.* **24**, No. 3, 390 (1986).
10. L. MANE AND TA PHUOC LOC, in *Numerical Methods in Laminar and Turbulent Flow, Vol.* 5, edited by C. Taylore *et al.* (Pineridge Press, Swansea, UK, 1987), Part 1, p. 867.
11. L. ROSENHEAD, *Laminar Boundary Layers* (Oxford Univ. Press, Oxford, 1963).
12. P. G. SAFFMAN AND G. R. BAKER, *Ann. Rev. Fluid Mech.* **111**, 95 (1979).
13. A. LEONARD, *J. Comput. Phys.* **37**, 289 (1980).
14. A. LEONARD, *Ann. Rev. Fluid Mech.* **17**, 523 (1985).
15. H. AREF, *Ann. Rev. Fluid Mech.* **15**, 345 (1983).
16. A. ANDERSON AND C. GREENGARD, *SIAM J. Numer. Anal.* **22**, No. 3, 413 (1985).
17. J. T. BEALE AND A. MAJDA, *J. Comput. Phys.* **58**, 188 (1985).
18. J. T. BEALE AND A. MAJDA, *Math. Comput.* **39**, 1 (1982).
19. J. T. BEALE AND A. MAJDA, *Math. Comput.* **39**, 28 (1982).
20. M. PERLMAN, *J. Comput. Phys.* **59**, 200 (1985).
21. A. J. CHORIN, *J. Fluid Mech.* **57** No. 4, 785 (1973).
22. A. J. CHORIN, *J. Comput. Phys.* **27**, 428 (1978).
23. A. J. CHORIN, *SIAM J. Sci. Statist. Comput.* **1**, 1 (1980).
24. J. P. CHRISTIANSEN, J. *Comput. Phys.* **13**, 363 (1973).
25. G. R. BAKER, *J. Comput. Phys.* **31**, 76 (1979).
26. L. GREENGARD AND V. A. ROKHLIN, Yale University Research Report No. YALEU/DCS/RR-459, 1986 (unpublished).
27. A. Y. CHEER, *SIAM J. Sci. Statist. Comput.* **4**, 685 (1983).
28. A. Y. CHEER, *J. Fluid Mech.* **201**, 485 (1989).
29. R. BOUARD AND M. COUTANCEAU, *J. Fluid Mech.* **101**, No. 3, 583 (1980).
30. A. I. SHESTAKOV, *J. Comput. Phys.* **31**, 313 (1979).
31. Z. Y. LU, M. S. thesis, Cornell University, 1983 (unpublished).
32. S. F. SHEN AND Z. Y. LU, *Numerical Methods in Heat Transfer, III*, edited by R. W. Lewis (Wiley, New York, 1985), p. 79.
33. P. A. RAVIART, An analysis of perticle methods, *Numerical Methods in Fluid Dynamics*, CIME course, Como, 1983 (unpublished).

34. G. H. COTTET AND S. GALLIC, Centre de Mathématiques Appliquées, Ecole Poly-technique Rapport Interne No. 115, 1985 (unpublished).
35. G. H. COTTET AND S. GALLIC, Centre de Mathématiques Appliquées, Ecole Poly-technique Rapport Interne No. 158, 1987 (unpublished).
36. J. P. CHOQUIN AND B. LUCQUIN, *Int. J. Numer. Methods Fluids* **8**, 1439 (1988).
37. Z. Y. LU, Ph. D. thesis (Aerospace Engineering), Cornell University, 1987 (unpublished).
38. N. N. YANENKO, *The Method of Fractional Steps*, English transl., edited by M. Holt (Springer-Verlag, Berlin, 1971).
39. P. M. MORSE AND H. FESHBACH, *Methods of Theoretical Physics* (McGraw–Hill, New York, 1953), p. 857.
40. L. L. VAN DOMMENLEM AND S. F. SHEN, in *Proceedings, Thirteenth Biennial Fluid Mechanics Symposium* (Olsztyn-Kortowo, Poland, 1977).
41. G. I. MARCHUK, *Dokl. Acad. Sci. USSR* **155**, Nos. 1–6, 10 (1965).
42. F. MILINAZZO AND P. G. SAFFMAN, *J. Comput. Phys.* **23**, 380 (1977).
43. S. ROBERTS, *J. Comput. Phys.* **58**, 29 (1985).
44. R. K. C. LO AND L. TING, *Phys. Fluids* **19**, No. 6, 912 (1976).