JOURNAL OF COMPUTATIONAL PHYSICS **127,** 88–109 (1996)
ARTICLE NO. 0160

A New Diffusion Procedure for Vortex Methods

S. SHANKAR AND L. VAN DOMMELEN

Department of Mechanical Engineering, FAMU/FSU College of Engineering, 2525 Pottsdamer Road, Tallahassee, Florida 32310

Received September 19, 1994; revised February 27, 1996

A new method is proposed for simulating diffusion in vortex methods for two-dimensional incompressible flows. The method resolves length scales up to the spacing of the vortices. The grid-free nature of vortex methods is fully retained and the distribution of the vortices can be irregular. It is shown for the Stokes equations that in principle, the method can have any order of accuracy. It also conserves circulation, linear, and angular momentum. The method is based on exchanging a conserved quantity between arbitrary computational points. This suggests that extensions to more general flows may be possible. For the two-dimensional incompressible flows studied, circulation is exchanged between vortices to simulate diffusion. The amounts of circulation exchanged must satisfy a linear system of equations. Based on stability considerations, the exchanged amounts should further be positive. A procedure to find a solution to this problem is formulated using linear programming techniques. To test the method, the decay processes of a single point vortex and of a counterrotating pair of point vortices are computed. Current limitations of the method are discussed.

1. INTRODUCTION

The evolution of the vorticity in two-dimensional incompressible flows is due to convection and diffusion. Vortex methods approximate the vorticity distribution by discrete vortices [38, 39, 52, 3]. This allows the convection of vorticity to be represented by motion of the individual vortices.

The velocity of the vortices can be computed efficiently using the fast adaptive algorithm developed by Van Dommelen and Rundensteiner [60]. Similar fast algorithms have been developed by Greengard and Rokhlin [31], Carrier, Greengard, and Rokhlin [12], Anderson [4], and Almgren, Buttke, and Colella [2]. However, this paper is concerned with the more difficult problem of representing the diffusion processes.

Chorin proposed the random walk method [17] in which the vortex positions are given random displacements to simulate diffusion. His procedure conserves the total circulation but it does not conserve the mean position of the vorticity exactly. Milinazzo and Saffman [45] tested the random walk method for the case of an initially finite region of vorticity in an unbounded domain. They corrected for the mean position error but found that the error in mean size of the vortex system is proportional to the inverse square root of the number of vortices N. If the initial vorticity inside the region is constant, the number of vortices must be increased with the Reynolds number to keep the relative error in change in size constant at finite times. Roberts [50] showed that if the relative error in size itself is of importance, higher Reynolds numbers do not require additional vortices. In fact, the number of vortices can be reduced if the initial data represent the initial mean size accurately.

Marchioro and Pulvirenti [41], Goodman [28], and Long [40] have shown the convergence of the method. Chang [13] discusses how to incorporate the random walk in Runge–Kutta time-stepping schemes.

Ghoniem and Sherman [27] studied ways of handling boundary conditions. Sethian and Ghoniem [53] studied convergence for a backward-facing step numerically. Cheer [14] has implemented the random vortex-sheet/blob method for the flow due to an impulsively started cylinder at Re = 9500.

Fogelson and Dillon [24] have used a simplified one-dimensional version of the problem to study the question of how much smoothing should be applied to the random walk results. They found that convergence occurs when the random walk solution is smoothed over a distance that is large compared to the point spacing. Their results show that still a very large number of vortices is needed to improve the accuracy of the random walk method.

Degond and Mustieles [22] formulated a deterministic displacement that simulates diffusion. This is accomplished by defining an "equivalent convection velocity" for the diffusion process. They do point out that this method may be less accurate and more expensive than other methods for the Navier–Stokes and the heat equations. However, it may be suited to problems in kinetic theory of plasma physics.

Another procedure is the core expansion method proposed by Leonard [38]. In this method the core of each vortex is allowed to expand to simulate diffusion. Such an expansion correctly approximates the Stokes equation. However, Greengard [30] has shown that it cannot model convection correctly when applied to the Navier–Stokes equations.

0021-9991/96 $18.00

A deterministic method to simulate diffusion has been developed by Raviart [48], Choquin and Huberson [15], and Cottet and Mas-Gallic [20]. They use viscous/inviscid splitting of the vorticity equation and then solve the diffusion equation exactly using the fundamental solution of the heat equation. Presently the ''deterministic particle (or vortex) method'' has been developed along different lines by Degond and Mas-Gallic [21] and Mas-Gallic and Raviart [43]. Two basic ingredients in this approach are (a) to consider the strength (circulation) of each particle as an unknown coefficient that changes with time due to diffusion effects and (b) to approximate parts of the governing equations by integral operators that are discretized using the particle positions as quadrature points. Winckelmans and Leonard [66] refer to this method as ''particle strength exchange'' (PSE) scheme and have used it in their study of the fusion of vortex rings.

Mas-Gallic [44] has extended this deterministic particle method to problems with boundary conditions. Koumoutsakos, Leonard, and Pépin [35] describe the vorticity flux at solid wall boundary conditions based on the fundamental solution of the heat equation. Koumoutsakos and Leonard [36] have applied the scheme to impulsively started-and-stopped flow around a translating and rotating cylinder for Reynolds numbers from 40 to 9500.

In practice, such methods start out with a fixed number of particles, distributed uniformly over the domain, each with a prescribed initial strength [47, 34]. The changes in the particle strengths simulate the diffusion effects through a system of ordinary differential equations. The vortex size must be sufficiently large that it encloses enough vortices to resolve it.

A method with properties similar to the particle strength exchange scheme was derived by Fishelov [23]. She convolves the spatial derivatives in the vorticity equation with a smoothing function and then transfers the derivatives on to that function. This scheme is stable in the L_2 norm, at least for the heat equation, and it readily extends to a higher order of accuracy. With proper discretization, it can be made to conserve vorticity exactly.

Another deterministic vortex method is the free Lagrangian method [25, 11, 49 51]. The basic idea is to construct a finite difference scheme for the derivatives using the Voronoi diagram of the vortices. This method does conserve vorticity and angular momentum [51], but it is only weakly first-order consistent and it requires a uniformity condition for the distribution of the points [11].

We propose an alternate method to handle diffusion, following procedures used by Van Dommelen [61, 62] to solve the heat equation in one dimension. It is similar to the methods above in that it changes the strengths of the vortices to simulate diffusion: fractions of the circulation of each vortex are moved to neighboring vortices in order to produce the correct amount of diffusion. However, unlike the deterministic particle methods [21, 23], the maximum distance that the circulation of a vortex can move during a time-step is restricted to a chosen distance of the order of the point spacing. This allows scales up to the point spacing to be resolved. Unlike free Lagrangian methods, no partitioning of the domain is attempted; instead, all vortices within the allowed distance are included in the discretization.

The key question is to choose the fraction of the circulation of each vortex that is moved (redistributed) to each neighboring vortex. This choice determines the accuracy of the approximation, its stability, and its conservation properties. We will formulate a system of equations, (16) and the following, from which the redistribution fractions can be found. This system can be extended to any order of accuracy. A uniform distribution of vortices is not required; however, for uniformly distributed points the method can be equivalent to a finite difference scheme. Positivity of the solution of the system is enforced to ensure stability. We use a solution procedure that is guaranteed to find a positive solution to our system of equations, if one exists. If there is no acceptable solution, we add new vortices until there is one.

Fundamentally, our procedure differs from the usual particle methods by separating the computation of the vorticity into two distinct steps; (a) determination of particle values from equations involving localized versions of conservation laws; (b) reconstruction of the vorticity field by convolution. This separation allows us to achieve any chosen order of accuracy regardless of the geometry of the particle distribution. However, unlike other particle methods and other numerical methods, in our scheme an individual particle value has no identifiable meaning. It is the combination of nearby particle values and positions that determines the local solution. In Section 9, we discuss the practical implications of these differences.

2. GOVERNING EQUATIONS

The 2D Navier–Stokes equations for an incompressible fluid imply a vorticity convection–diffusion equation [5] of the form

$$\frac{\partial \omega}{\partial t} + \mathbf{u}\cdot\nabla\omega = \nu\,\nabla^2\omega, \tag{1}$$

where $\mathbf{u} = (u, v)$ is the velocity field, ω is the vorticity, defined as the curl of the velocity, t is the time, and ν is the kinematic viscosity of the fluid. Properly nondimensionalized, ν is the inverse of the Reynolds number. The velocity follows from the vorticity by means of the Biot–Savart law [5]. The flow region is assumed to be un-

bounded; we are currently working on computations including boundaries.

The vorticity distribution $\omega(\mathbf{x}, t)$ was approximated by a collection of "vortex blobs,"

$$\omega \approx \sum_{i=1}^{N} \Gamma_i(t)\phi_\delta(\mathbf{x} - \mathbf{x}_i(t)), \tag{2}$$

where Γ_i is the circulation of the vortex located at $\mathbf{x}_i$. In a point vortex method, the function ϕ_δ would be a delta function; however, this introduces the possibility of blowup of the velocities when vortices approach each other. Hence, we followed Chorin's [17] proposal to use vortex blobs with a finite core size. We chose the core of the usual form

$$\phi_\delta(\mathbf{x}) = \frac{1}{\delta^2}\,\phi\left(\frac{\mathbf{x}}{\delta}\right), \tag{3}$$

in which ϕ is a smooth axisymmetric function that integrates to one and δ is a small typical width.

The order of accuracy of the approximation depends on how many moments of ϕ vanish. For the Euler equations various convergence and accuracy proofs have been given, starting with Hald [32] and Beale and Majda [7] in higher order approximation. These assume that δ is asymptotically large compared to the point spacing; however, Goodman, Hou, and Lowengrub [29] show that point vortices also converge for all finite times if they are initially uniformly spaced.

Based on our practical experience with random walk computations [59, 57, 64], we chose a low-order algebraic blob of shape

$$\phi(\mathbf{x}) = \frac{1}{\pi(1 + |\mathbf{x}|^2)^2}, \tag{4}$$

which falls in a class discussed by Hald [33] for the Euler equations. Figures 10 and 11 suggest that for the present flows there is no significant change in the solution if the algebraic core is replaced by a more commonly used second-order exponential core. Higher-order applications would require a more sophisticated blob shape. We used a relatively small vortex diameter $\delta = \sqrt{0.5\nu\,\Delta t}$, as is common in practical applications [29].

The convection–diffusion equation (1) for the vorticity was solved by a fractional step, or viscous splitting, algorithm [67, 18, 6, 9] as follows:

Convection step:

$$\frac{d\mathbf{x}_i}{dt} = \mathbf{u}_i(\mathbf{x}_1, ..., \mathbf{x}_j, ..., \mathbf{x}_N, t) \tag{5}$$

$$\frac{d\Gamma_i}{dt} = 0. \tag{6}$$

Diffusion step:

$$\frac{d\mathbf{x}_i}{dt} = 0 \tag{7}$$

$$\frac{\partial\omega}{\partial t} = \nu\nabla^2\omega. \tag{8}$$

The order in which the diffusion and convection time-steps were performed was inverted every time-step to improve the rate of convergence.

The convection time-step was integrated using a fourth-order Runge–Kutta scheme, with the velocity found from the fast summation algorithm [60]. The diffusion step was solved by the redistribution technique discussed in the next section.

3. VORTICITY REDISTRIBUTION

The purpose of the vorticity redistribution technique is to simulate the diffusion of each vortex i during a time-step. As sketched in Fig. 1, this is done by distributing fractions of its circulation Γ_i^n to its neighboring vortices. The question is how to select the neighborbood and the fractions so that the correct diffusion is approximated. The next subsection will answer that question.

3.1. *Formulation*

Vortices will be considered to be within the neighborhood of a vortex i if they are within a predetermined distance from the vortex. We take this distance to be of the order of the typical diffusion distance during a time-step. To be precise, the typical diffusion distance h_v will

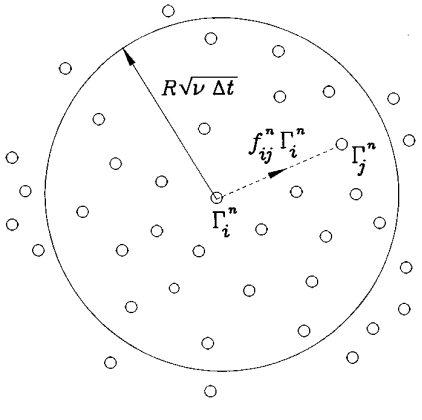

FIG. 1. Redistribution of the circulation of a vortex Γ_i^n.

be defined as

$$h_v \equiv \sqrt{\nu\,\Delta t}, \tag{9}$$

and a vortex j is part of the neighborhood of vortex i if

$$|\mathbf{x}_j - \mathbf{x}_i| \le R h_v. \tag{10}$$

We used a maximum distance $\sqrt{12}h_v$ in all computations presented in this paper. Some guidelines for choosing this distance will be given in Section 7.

Within this neighborhood of vortex i, its diffusion will be approximated by moving fractions of its circulation towards the other vortices. We will indicate the fraction that moves from vortex i to a vortex j by f_{ij}^n. Implementation of the redistribution method is in principle merely a matter of determining fractions f_{ij}^n that approximate the correct diffusion over a time-step accurately and stably.

Yet, we choose not to identify the vortex strengths with any particular smooth interpolated vorticity distribution. The reason is that due to straining effects, the vortex locations can become very irregular. In the absence of a continuous vorticity field, the question arises how a meaningful representation of the diffusion process can still be achieved.

The redistribution method changes a vorticity distribution

$$\omega^n = \sum_i \Gamma_i^n \phi_\delta(\mathbf{x} - \mathbf{x}_i) \tag{11}$$

into

$$\omega^{n+1} = \sum_i \sum_j f_{ij}^n \Gamma_i^n \phi_\delta(\mathbf{x} - \mathbf{x}_j). \tag{12}$$

We would like this change to approximate the true diffusion over the time step in some way. Our approach will be to demand that all finite wave numbers of the Fourier transform are correctly damped. This is similar to a weak formulation in which Fourier modes are used as weighting functions.

The Fourier transform of the new vorticity distribution is

$$\hat{\omega}^{n+1} = \hat{\phi}(k\delta) \sum_i \Gamma_i^n e^{-i\mathbf{k}\cdot\mathbf{x}_i} \sum_j f_{ij}^n e^{-i\mathbf{k}\cdot(\mathbf{x}_j - \mathbf{x}_i)}. \tag{13}$$

This is to be compared with the Fourier transform of the exactly diffused vorticity:

$$\hat{\omega}_e^{n+1} = \hat{\phi}(k\delta) \sum_i \Gamma_i^n e^{-i\mathbf{k}\cdot\mathbf{x}_i} e^{-k^2 \nu\,\Delta t}. \tag{14}$$

The two Fourier transforms cannot be equal for all values of k using only a finite number of vortices. However, within the neighborhood of vortex i, the distance $|\mathbf{x}_j - \mathbf{x}_i|$ is a small quantity of order $O(\sqrt{\Delta t})$; compare (9) and (10). This makes it possible to approximate the trailing exponentials in the two Fourier transforms by a truncated Taylor series. It does turn out to be possible to equate the Fourier transforms with these truncated Taylor series. The detailed derivation is given in Appendix A.

The resulting equations are the redistribution equations we were looking for. They involve scaled relative vortex positions defined as

$$\boldsymbol{\xi}_{ij} \equiv \frac{\mathbf{x}_j - \mathbf{x}_i}{h_v}, \tag{15}$$

that are bounded by the neighborhood radius $\xi_{ij} \le R$.

In terms of the scaled coordinates, the final redistribution equations are

$$O(1): \quad \sum_j f_{ij}^n = 1; \tag{16}$$

$$O(\Delta t)^{1/2}: \quad \sum_j f_{ij}^n \xi_{1ij} = 0; \qquad \sum_j f_{ij}^n \xi_{2ij} = 0; \tag{17}$$

$$O(\Delta t): \quad \sum_j f_{ij}^n \xi_{1ij}^2 = 2; \qquad \sum_j f_{ij}^n \xi_{1ij}\xi_{2ij} = 0; \qquad \sum_j f_{ij}^n \xi_{2ij}^2 = 2; \tag{18}$$

$$O(\Delta t)^{3/2}: \quad \sum_j f_{ij}^n \xi_{1ij}^3 = 0; \qquad \sum_j f_{ij}^n \xi_{1ij}^2\xi_{2ij} = 0; \qquad \sum_j f_{ij}^n \xi_{1ij}\xi_{2ij}^2 = 0; \quad \sum_j f_{ij}^n \xi_{2ij}^3 = 0; \tag{19}$$

$$O(\Delta t)^{m/2}: \quad \text{Higher-order moment equations, } m = 4, \ldots, M+1. \tag{20}$$

From these equations, the redistribution fractions f_{ij}^n are to be found.

Consistency requires that the numerical solution approximates the $O(\Delta t)$ diffusive changes in the exact solution: the redistribution fractions must at least satisfy (16) through (18). This results in a truncation error of order $O(h_v)$. Subsequent equations, (19), (20), can be included to achieve a higher order of accuracy $O(h_v^M)$.

Thus, in principle, the accuracy can be increased arbitrarily, although for the Navier–Stokes equations the splitting error also has to be considered. The conditioning of the above system of equations also needs to be taken into account in a practical application. The equations could be recast in terms of orthogonal polynomials such as Legendre polynomials to improve the conditioning. On the other hand, the conditioning of the system may not be very important; the requirement is not to find a particular solu-

tion for the fractions f_{ij}^n, but to satisfy the equations accurately. In the numerical results in this paper, we simply solved (16) through (18) in the form shown.

The redistribution equations are similar to the equations obtained when a Taylor series expansion of the exact solution is substituted into a finite difference formula, or to the moment conditions in the particle methods. In fact, consistency of a finite difference scheme requires the same agreement for finite wave numbers (e.g., [58, (10.1.3)]). For uniform point spacing and redistribution fractions, the redistribution method is equivalent to an explicit finite difference scheme. The redistribution equations do not involve the smoothing function. This allows us to choose this funciton after the actual computation has already been completed.

In implementing the redistribution scheme, it is important to realize that not all solutions f_{ij}^n to (16) and the following will lead to a convergent approximation. For example, a consistent but unstable explicit finite difference scheme would satisfy the equations. Some form of stability condition needs to be imposed; following Van Dommelen [62], we will demand that all fractions are positive:

$$f_{ij}^n \geq 0. \tag{21}$$

This ensures that the l_1 norm $\Gamma^n = \sum_i |\Gamma_i^n|$ of the circulation cannot grow.

In the next two subsections we will further justify the above conditions using physical and mathematical arguments. However, the truly relevant questions are clearly whether the equations are solvable, whether they can be solved using only a finite number of neighboring points within a finite scaled distance R, and whether the numerical solution approaches the exact solution with the expected rate of convergence. In the following, Sections 4 through 7, we will prove that the answer to all these questions is affirmative for the linear Stokes equations. To verify that our method also works for the nonlinear Navier–Stokes equations, we will present example computations with nontrivial convection effects in Section 8.

3.2. *Physical Meaning of the Equations*

Equations (16) through (21), derived in the previous subsection are the core of the redistribution method. While they were derived using mathematical arguments, some have a clear physical meaning. For example, the lowest order Eq. (16) conserves circulation for each vortex.

Next, (17) conserves the center of vorticity; and (18) implies the correct expansion of the mean diameter. These conservation laws are expressions of the physical laws of conservation of linear and angular momentum [37].

The positivity condition expresses the physical fact that reverse vorticity cannot form spontaneously in the middle of a flow field.

The size (10) of the redistribution region corresponds to the typical distance of order $O(\sqrt{\nu\,\Delta t})$ over which the vorticity of a vortex diffuses during a time-step. It ensures that numerically the vorticity diffuses out over a distance of the same order.

Together, these properties imply that even if numerical resolution is poor, the possible effects of the errors remain quite limited. No false circulation, linear or angular momentum, or reversed vorticity an be created by the numerical errors. The center of vorticity is unaffected and the root mean square size of the vortex system expands at the correct rate. The vorticity does not expand over a region much larger than the physical one. The long range errors in velocity, which are determined by the vorticity moments, vanish. Disjoint sets of vortices more than $O(\sqrt{\nu\,\Delta t})$ apart satisfy the conservation laws individually.

3.3. *General Justification*

We would certainly not suggest that our redistribution method, and its detailed implementation, is the only possible approach to diffusion in vortex computations. We merely want to explain the reasoning that led us to formulate this particular procedure. Hopefully, this will explain why our method does have a number of advantages that may be of importance.

We wanted a scheme to replace the random walk method in our computations. Like this method, it should not require ordered vortices; the method should not be based on associations between individual vortices such as a uniform or regular distribution of the vortices, a numerical quadrature rule using the vortices, or any partitioning of the domain. Our motivation for this demand was that in a Lagrangian computation convection effects eventually decouple vortices initially associated with each other. Any order introduces complications; it needs to be decided how long the computation can proceed without restoring a new order and how to restore it. It causes uncertainty about the possible errors introduced by each of those decisions. A scheme that merely identifies neighboring vortices avoids these difficulties. It also simplifies the computation of flows about complex geometries.

We did not want a partitioning of the domain as in ''unstructured'' computations. Such a partitioning is still a form of structure that must be regenerated. It brings in complicating aspects such as the geometry of triangles that are not found in the physical flow. Our scheme uses all available vortices within some reasonable distance, rather than a selected subset, to find a suitable discretization for the diffusion of each vortex.

Yet, variations on our scheme remain possible. For example, instead of attempting to describe the diffusion of each individual vortex separately as we do, it would be possible to divide the domain into small square or hexago-

nal regions and demand only that the net diffusion of all vortices within each region is correctly represented to some order.

However, the work involved in diffusing the individual vortices does not seem to be prohibitive. This is certainly true theoretically, since the work for the redistribution process is asymptotically negligible compared to the work needed to find the velocity field. (We do not consider the machine precision finite as other authors, since this does not allow convergence to occur). In our experimental results for a limited number of vortices, the actual work is acceptable but still significant. As explained in Section 5, we believe that this is due to our brute force approach to finding the redistribution fractions.

Our requirement that the redistribution weights be positive was motivated in part by the standard five-point explicit finite difference scheme for the diffusion process. For that finite difference scheme, the transition from a stable to an unstable scheme occurs when one of the fractions becomes negative. Therefore, at least when the vortices are located on a uniform mesh and the redistribution radius includes five vortices at a time, the positivity constraint needs to be satisfied. Furthermore, the positivity condition is sufficient; in Section 4 we will prove for the linear Stokes equation that it ensures convergence of the method for any arbitrary point distribution.

On the other hand, there are certainly stable finite difference schemes with negative fractions that will be excluded by the positivity constraint. Yet this does not appear to be an unacceptable loss; suitable positive solutions can always be found. As discussed in Section 7, for any order of approximation our redistribution equations can be solved using only a finite number of points within a finite scaled radius R.

3.4. *Implementation*

Basically, the redistribution method is a matter of finding the fractions f_{ij}^n, from (16) and the following, and redistributing the circulation of each vortex according to these fractions. Assuming that valid fractions f_{ij}^n exist, they can be found using linear programming techniques, as explained in Section 5. However, it is possible that no valid fractions exist using the available vortices in a neighborhood. In that case, we create new vortices until there is a solution, as discussed in Section 7.

To evaluate the velocity field or the vorticity field, a smoothing function is used. Note that the redistribution process itself is independent of the vortex core shape ϕ_δ, since (16) and the following are.

In our actual computations we used two different vortex cores; the low-order algebraic vortex core mentioned in the previous section was used during the actual computation to integrate the convection processes. Our reason was that our interests tend to be in unsteady separating flows at large Reynolds numbers. Such flows involve short scale vorticity features that can be lost for large core sizes, making a small core size desirable regardless of the order of accuracy of the core. For a small core size, a relatively low order of accuracy can be sufficient; and other considerations may be more important. In particular, positive second-order cores such as the one we used have the advantage that they cannot introduce false vorticity of opposite sign.

The other core is used to evaluate the vorticity field for output purposes. In this case the considerations for the choice of the core are somewhat different. For maximum visual smoothness, a large core is desirable, since a large core gives the greatest reduction in short wave errors. These short wave errors are, further, also much more pronounced in the vorticity field than in the velocity. On the other hand, there is much less risk that a larger core would smooth small features, since the actual solution is now known, and the effect of the core size can be determined experimentally without repeating any of the computation. Further, even if there would be some loss of information about the shorter wave lengths, this loss does not affect the further computation, since the second core is used for output only. Such considerations suggest the use of a core with a high order of accuracy, since these can be larger for a given accuracy. Thus we chose a relatively large but infinite-order core,

$$\phi(\mathbf{x}) = \frac{J_1(|\mathbf{x}|)}{2\pi|\mathbf{x}|}, \tag{22}$$

of the form proposed by Leonard [38]. According to Figs. 10 and 11, our results are not sensitive to the precise form of both these core shapes.

4. CONVERGENCE FOR THE STOKES EQUATIONS

In this section, we will prove convergence of the redistribution algorithm for the Stokes or heat equations. The redistribution fractions are assumed to satisfy the redistribution equations (16) and the following, to satisfy the positivity constraint (21) and to be restricted to vortices within a mutual distance (10), with $R > 1$. Since we are enforcing consistency in the L_2 norm, using the Fourier transform, while we have stability in the l_1 norm, and the redistribution fractions are only partly determined, the conventional convergence arguments need some modifications.

We will show convergence in the L_2 norm by showing convergence of the Fourier transform of the numerical solution,

$$\hat{\omega}^n = \hat{\phi}(k\delta) \sum_i \Gamma_i^n e^{-i\mathbf{k}\cdot\mathbf{x}_i} \tag{23}$$

to the Fourier transform of the exact solution,

$$\hat{\omega}(t) = \hat{\omega}_0 e^{-k^2 \nu t} \tag{24}$$

in the L_2 norm. Here ω_0 is the given initial vorticity and we do not explicitly show the dependence on $\mathbf{k}$.

The total error consists of the error induced by discretizing the initial data and the error induced by the redistribution method itself:

$$\begin{aligned}\|\hat{\omega}_0 e^{-k^2 \nu t} - \hat{\omega}^n\| \le \|(\hat{\omega}_0 - \hat{\omega}^0) e^{-k^2 \nu t}\| \\ + \|\hat{\omega}^0 e^{-k^2 \nu t} - \hat{\omega}^n\|.\end{aligned} \tag{25}$$

The first error due to discretizing the initial data can be important if the initial data have only limited smoothness or if a low-order smoothing function is used. It depends on how the initial discretization is performed. Typically the initial vortices are given a uniform spacing $h = O(h_v)$ and the initial vortex strength is taken as $\Gamma_i^0 = h^2 \omega_0(\mathbf{x}_i)$. Since the initial vorticity field is evaluated only at the vortices, some information is lost; aliasing makes ω_0 indistinguishable from the Fourier interpolant ω_h through the vorticity values. The total error due to discretization of the initial data can be written:

$$\begin{aligned}\|(\hat{\omega}_0 - \hat{\omega}^0) e^{-k^2 \nu t}\| \le \|(\hat{\omega}_0 - \hat{\omega}_h) e^{-k^2 \nu t}\| \\ + \|(\hat{\omega}_h - \hat{\omega}^0) e^{-k^2 \nu t}\|.\end{aligned} \tag{26}$$

The magnitude of the first of these two errors depends on the number of square integrable derivatives of the initial vorticity. It may be shown that if σ derivatives are square integrable, this error is of order h^σ ([58, pp. 198–206]. In two dimensions σ has to be greater than one, but fractional values are allowed.

The second error is due to the vortex core. Assuming $\hat{\phi}_\delta$ to be bounded, for nonzero times the order of this error is simply the order of accuracy of the vortex core. Thus, if the core is accurate $O(h_v^M)$, the overall accuracy of the computation is not affected by the core.

It follows that for a sufficiently accurate smoothing function and smooth initial data, the only important error will be that due to the redistribution process. To estimate this error, we first define the local error in the Fourier transform at time-level n to be the difference between the redistribution solution and the exactly diffused solution from the previous time-step:

$$\hat{\varepsilon}^n \equiv \hat{\omega}^{n+1} - \hat{\omega}^n e^{-k^2 \nu \Delta t}. \tag{27}$$

By repeated application of this definition, the error in the Fourier transform due to redistribution can be bounded by

$$|\hat{\omega}^n - \hat{\omega}^0 e^{-k^2 \nu t}| \le \frac{\hat{\varepsilon}}{k^2 h_v^2}(1 + k^2 h_v^2), \tag{28}$$

where $h_v = \sqrt{\nu \Delta t}$ and $\hat{\varepsilon} = \max_{i=0}^{n-1}\{|\hat{\varepsilon}^i|\}$. A similar estimate could be derived for particle methods [66].

To estimate $\hat{\varepsilon}$, recall that the redistribution equations (16) and the following ensure vanishing of the first few powers of Δt in the error in (27). Therefore, the Taylor series remainder theorem can be used to express the remaining difference $\hat{\varepsilon}^n$. That expression is shown in Appendix A; it can be bounded as

$$\hat{\varepsilon} \le |\hat{\phi}(k\delta)| \max_n (\Gamma^n F^n) R^{M+2} (k h_v)^{M+2} (1 + k^2 h_v^2), \tag{29}$$

$$F^n = \max_i \sum_j |f_{ij}^n|, \quad \Gamma^n = \sum_i |\Gamma_i^n|. \tag{30}$$

For the assumed positivity of the redistribution fractions (21), $F^n = 1$ and Γ^n cannot increase. Thus the total error in the Fourier transform is bounded by

$$|\hat{\omega}^n - \hat{\omega}^0 e^{-k^2 \nu t}| \le |\hat{\phi}(k\delta)| 4\Gamma^0 R^{M+2} \min\{(k h_v)^M, 1\}, \tag{31}$$

where the second bound comes from the bound $|\phi_\delta| \Gamma^n$ to (23).

In this paper, we will assume that Γ^0, the absolute circulation of the discretized initial data, is finite. Note that this is a restriction on the l_1 norm of the initial discrete vortex strengths, rather than on the L_2 norm of the initial vorticity distribution. However, the Cauchy inequality applied to

$$\sum_i h|\omega_i|(1 + x_i^2 + y_i^2)^{(1+\alpha)/2} \cdot h(1 + x_i^2 + y_i^2)^{-(1+\alpha)/2}, \tag{32}$$

with α an arbitrary positive constant, readily shows that Γ can be bounded in terms of the L_2 norms of the initial vorticity and aliasing error, provided that the initial vorticity is restricted to a finite region or at least decays sufficiently rapidly at large distances. For example, it would suffice that $\omega_0 = O(x^{-2-\alpha})$ for $x \to \infty$ for some $\alpha > 0$.

The final L_2 error in the vorticity is found from square integration of (31) over all wavenumbers. Thus the error due to redistribution is found to be:

$$\begin{aligned}\|\hat{\omega}^n - \hat{\omega}^0 e^{-k^2 \nu t}\| \le C\Gamma^0 R^{M+2} \frac{h_v^M}{\delta^{M+1}}, \\ C = 4\left(2\pi \int_0^\infty |\hat{\phi}(k)|^2 k^{2M+1}\, dk\right)^{1/2}\end{aligned} \tag{33}$$

To minimize this error, a relatively large core size is desirable. If we take the core size proprotional to some small power α of h_v, the error will be $O(h_v^{M-\alpha(M+1)})$. Since

we can take α as any positive number, we can obtain any order of accuracy arbitrarily close to $O(h_v^M)$. Note, however, that for a core with a finite order of accuracy, the first error in (25), due to discretizing the initial data, limits the maximum size of δ. Our core (22) is infinite-order accurate.

This completes the discussion of convergence for the Stokes equations. It is interesting to note that the true stability conditions are that F^n and Γ^n are bounded. Next we will address how to find the redistribution fractions f_{ij}^n in an actual application of the scheme.

5. FINDING THE REDISTRIBUTION AMOUNTS

As explained in Section 3, the key to the redistribution method is to find a positive solution to the system (16) and the following for the redistribution fractions f_{ij}^n. The system is linear, but usually not square; the number of unknown fractions f_{ij}^n is the number of vortices in the neighborhood, while the number of equations is determined by the order of accuracy M desired. In this section we will discuss our strategy for obtaining a positive solution for the f_{ij}^n, assuming that one exists. The question what to do if no positive solution exists will be addressed in Section 7.

The problem of finding a nonnegative solution to an underdetermined system of equations is the standard ''phase I'' problem in linear programming that can be solved by slack variables. However, following Van Dommelen [62] we will use a different approach. First, we note that the fractions f_{ij}^n must be in the range [0, 1]. We may shift the origin to the center of that range, by defining

$$w_j \equiv f_{ij}^n - \tfrac{1}{2}, \tag{34}$$

where the additional dependence of w_j on the vortex i and the time-step n is to be understood. In terms of the w_j, a solution is acceptable if the maximum norm of the solution vector, $\|\mathbf{w}\|_\infty \equiv \max_j\{|w_j|\}$ is less than or equal to $\frac{1}{2}$.

Our approach is to find the solution for $\mathbf{w}$ with the least maximum norm. If the maximum norm is less than or equal to $\frac{1}{2}$, an acceptable solution has been found. On the other hand, if the maximum norm exceeds $\frac{1}{2}$, it must mean that no acceptable solution exists. In that case we create more vortices as described in Section 7.

The least maximum solution algorithm that we used is described in the next section. We did do some comparative testing of this algorithm against a standard library routine (IMSL) for the phase I linear programming problem. We found that the number of iterations in the methods was about equal, but that the library routine ran about two times more slowly, possibly due to the extensive safeguards in its implementation. It appears that computational speed is not an important consideration in selecting the method.

However, the least maximum procedure will create a strictly positive solution if one exists, while the linear programming method for the phase I problem will select the minimum number of vortices for the redistribution. As a result, the least maximum procedure tends to spread out the vorticity somewhat better.

In this study we solved the least maximum problem for each vortex at each time-step from scratch (even for the Stokes flow in which a single solution could have been used for all time-steps). This is a very inefficient approach, since the systems are almost unchanged from one time-step to the next. The relative locations of the vortices in the neighborhood change only by an amount $O(h_v^3)$ during a time-step. This means that a single solution can be used over an asymptotically large number of time-steps. Additionally, in our time splitting we perform two diffusion steps back to back. We solve each from scratch although they are identical.

The disadvantage is that some information has to be stored from one time-step to the next. For example, the fractions f_{ij}^n could be stored and up-dated at each time-step until one turns negative, at which time the system could again be solved from scratch. Alternately, we could merely store the information which of the fractions have magnitude less than the maximum norm. This is sufficient information to solve a least maximum problem quickly. In any case, our computational times for the redistribution method can presumably still be improved significantly.

We did use one shortcut in our procedure. As a preconditioning to finding the least maximum solution, we performed a Gram–Schmidt orthogonalization on the rows of the system. This orthogonalization directly determines the least length solution, and we found that in about 60% of the cases, the least length solution was positive. Thus we could skip the determination of the least maximum solution in the majority of cases.

There are also tests that could be performed to decide a priori that a system has no acceptable solution: according to estimates given in Appendix A, there must be at least one neighborhood vortex at a distance of more that $\sqrt{4\nu\,\Delta t}$, the maximum horizontal and vertical distances should be at least $\sqrt{2\nu\,\Delta t}$, and at least $4\sqrt{\nu\,\Delta t}/(R + \sqrt{R^2+8})$ in any direction. For third-order accuracy or higher, there should be at least one vortex within a distance $\sqrt{8\nu\,\Delta t}$.

6. LEAST MAXIMUM SOLUTION PROCEDURE

The strategy for finding a redistribution solution was formulated in the previous section; it reduces to the standard mathematical problem of finding the least maximum norm solution to a linear system of equations. Van Dommelen [62] used an ad hoc procedure to solve this problem [63]. However, in the current study we have adopted a

scheme developed by Abdelmalek [1]. This choice was based on some numerical experiments that showed that the method below usually takes less computational time. The procedure of Van Dommelen tends to be somewhat more robust on poorly conditioned systems, but we have adopted a Gram–Schmidt orthogonalization of the rows of the matrix as a standard preconditioning.

We do point out that the procedure of Van Dommelen has the following advantages: (a) it allows the iterations to be terminated early; even without convergence, the solution might satisfy the positivity condition; (b) the method provides a lower bound on the least maximum that might be used to predict early that a system does not have an acceptable solution; and (c) it might be extended to allow the solution at the previous time-step to be used as a starting point of the iterations. More research is needed, but the procedure below was found to be reliable and converged well.

Our starting point is the linear system of equations

$$A\mathbf{w} = \mathbf{b} \tag{35}$$

obtained from the redistribution system (16) and the following by shifting the unknown fractions according to (34) and orthogonalizing the rows. We now want to find the least maximum solution to this problem.

The problem of finding a least maximum solution to a general linear system may be formulated as a linear programming problem. This is achieved by considering the maximum norm $\|\mathbf{w}\|_\infty$ as another unknown. Casting equalities as the two inequalities $\geq$ and $\leq$, this yields

$$\begin{pmatrix} A & \mathbf{0} \\ I & \mathbf{1} \\ -A & \mathbf{0} \\ -I & \mathbf{1} \end{pmatrix} \begin{pmatrix} \mathbf{w} \\ \|\mathbf{w}\|_\infty \end{pmatrix} \geq \begin{pmatrix} \mathbf{b} \\ \mathbf{0} \\ -\mathbf{b} \\ \mathbf{0} \end{pmatrix}, \tag{36}$$

where $\mathbf{0}$ and $\mathbf{1}$ indicate vectors of zeros and ones. The objective function to minimize is the maximum norm $\|\mathbf{w}\|_\infty$.

Abdelmalek [1] points out that there are advantages to solving the dual problem. The dual maximizes

$$(\mathbf{b}^T\mathbf{0}^T \quad -\mathbf{b}^T\mathbf{0}^T)\mathbf{y}, \tag{37}$$

subject to the constraints

$$\begin{pmatrix} A^T & I & -A^T & -I \\ \mathbf{0}^T & \mathbf{1}^T & \mathbf{0}^T & \mathbf{1}^T \end{pmatrix} \mathbf{y} = \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix}, \tag{38}$$

$$\mathbf{y} \geq 0. \tag{39}$$

The advantage is that an initial feasible solution is easy to find, so that no slack variables are needed. Further, due to the special structure of the matrix, the only storage needed is for the original matrix and a few vectors. This also reduces the work required to find the optimal solution.

The simplex method [26] requires a number of different tolerances to be specified a priori. We followed the recommendations of Clasen [19]. Convergence occurred typically within about 12 vertex interchanges in the simplex method.

7. ADDING VORTICES

The numerical technique of the previous two sections will find a positive solution to the redistribution equations as long as one exists. A solution does not necessarily exist, however. In that case, new vortices with zero circulation are added until a positive solution does become possible.

There are various reasons why a solution may not exist. For example, the number of vortices in the neighborhood may be less than the chosen number of redistribution equations. First-order accuracy requires at least six vortices, and this number increases for higher orders.

Further, the neighborhood radius may be too small for the desired order of accuracy M. According to an estimate derived in Appendix A, the scaled neighborhood radius R must be at least $\sqrt{2M_e}$, with M_e the even integer M or $M + 1$. For first- or second-order accuracy, this requires a minimum value $R = 2$. For third-order accuracy or higher, the vortices should also not be spaced too far apart; the scaled spacing cannot exceed $\sqrt{8}$.

At the outer edge of the region containing the vortices, a solution always requires new vortices. According to an estimate derived in Appendix A, the vortex region must expand by a finite, scaled distance in each direction.

On the other hand, under reasonable conditions positive solutions to the redistribution equations do exist. For example, a standard five-point explicit finite difference formula with $\Delta x = \Delta y = R$ gives a second-order positive solution as long as R is at least the minimum value 2 as mentioned above. Similarly a fourth-order solution exists if R is at least $\sqrt{8}$; see Appendix A.

More generally, there is always a finite scaled neighborhood radius R for which the existence of a positive solution is assured, provided only that there are no ''holes'' in the distribution of the vortices that exceed some finite, scaled size d. This is shown in Appendix A; however, it does not give values for R and d.

In our first-order computations, we choose the redistribution radius $R = \sqrt{12}$, which is well above the minimum value 2 for which a positive solution becomes possible. The reason is that the minimum value requires vortices placed at optimum positions. For a larger radius, a positive solution may be found for more general vortex placings.

One question of concern is where to place newly created vortices. Van Dommelen [62] showed in one dimension that if the computation starts with a single vortex and new vortices are added at scaled distances $\sqrt{6}$, the fourth-order accurate finite difference scheme is obtained. Based on this observation, we adopted the strategy that if no positive solution can be found for a certain vortex, a new vortex is added at a scaled distance $\sqrt{6}$ from the considered vortex. The angular location of the new vortex is chosen among 12 possible positions spaced 30° apart, by maximizing the distance between the new vortex and the existing vortices.

This procedure worked well in practice, but it is certainly not unique. For example, Van Dommelen showed that the new vortices may also be placed at random positions without apparent ill effects. However, our procedure has some advantages. It will always succeed; a positive solution is assured as soon as the points of a five-point finite difference stencil have been filled. It also tends to fill up the holes in the distribution of the vortices. Since the newly added vortices are located away from the edge of the redistribution region, it takes a finite time before they can convect out ot it.

Figure 2 shows the increase in the number of vortices for an example computation. The computation is the Stokes flow starting from two concentrated, counterrotating vortices. It is found that our strategy of placing new vortices increases the vortex density initially until it fills up the "holes" in the distribution. When a certain vortex density is reached, the distribution becomes steady. For example, the region shown for the final time in Fig. 2 is unchanged at double that time.

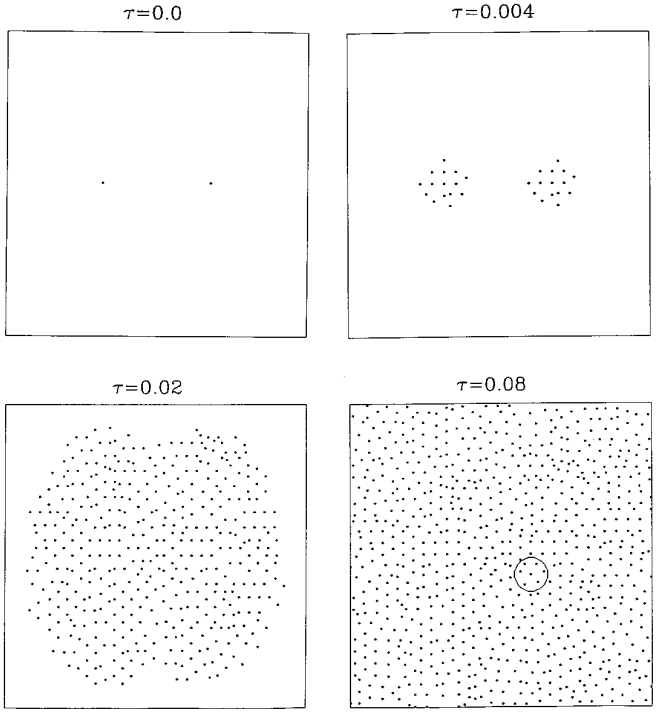

FIG. 2. Growth in number of vortices for the Stokes flow starting from a pair of counter rotating point vortices. The small circle indicates the size of the neighborhood.

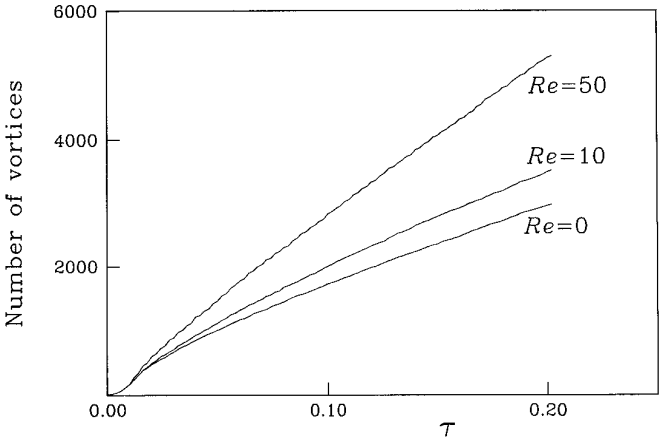

FIG. 3. Total number of computational vortices versus time.

As shown in Fig. 3, the total number of vortices in the computation does continue to grow. The reason is that new vortices continue to be added near the boundaries of the domain. In fact, since the region containing vorticity continues to grow linearly with time, ideally the number of vortices should also grow linearly.

However, it was noted above that redistribution must expand the region containing the vortices by a scaled distance that does not depend on time. This would lead to a number of vortices that grows quadratically in time. It would include large amounts of vortices with exponentially small strengths at large distances. To prevent this growth, we do not redistribute a vortex if its strength is below the machine epsilon. Using this restriction, Fig. 3 shows that the growth is indeed quite linear.

Convection introduces a further complication. Even if a vortex can be redistributed at a given time, after a finite time convection can move the vortices to locations for which a positive solution may no longer exist. In that case new vortices must be added. This can happen even though incompressibility ensures that the average vortex density does not change. The reason is that vortices might approach closely, which allows holes in the vortex distribution to form, even though in principle there are enough vortices to fill those.

As an example, Fig. 4 shows the evolution of the vortex distribution at Reynolds number 50, when there are very strong convection effects. While we always add new vortices in the biggest hole we can find locally, it is seen that convection has caused some vortices to approach closely.

As a result, the number of vortices in a typical redistribution radius, shown as a circle, has increased, compared to the case of no convection in Fig. 2.

The additional vortices require increased computational resources. It also raises the more fundamental question whether the number of vortices within a redistribution distance remains finite. This requires a total number of vortices that increases linearly when the time-step is refined. Table I verifies this requirement at Reynolds number 50.

For many practical applications, high Reynolds numbers are of most interest. For such applications it would also be desirable that the number of vortices within a redistribution radius remains finite in the limit of an infinite Reynolds number. However, it is evident from Figs. 3 and 4 that the number of vortices increases without apparent bound when the Reynolds number is increased. One reason is the use of a scaled viscous time in Fig. 3; for a constant physical time the total number of vortices decays with the Reynolds number.

Yet even at a constant time, the average number of vortices in a redistribution radius still increases with the Reynolds number. The reason seems to be that fluid straining is particularly strong for this flow; there is no bound on the magnitude of the velocity at any given time when the Reynolds number increases. It would, however, be desirable to prevent a significant increase in the number of vortices under all circumstances, since it results in the loss of numerical efficiency. This requires that the positions of the vortices be periodically restored to a more even distribution.

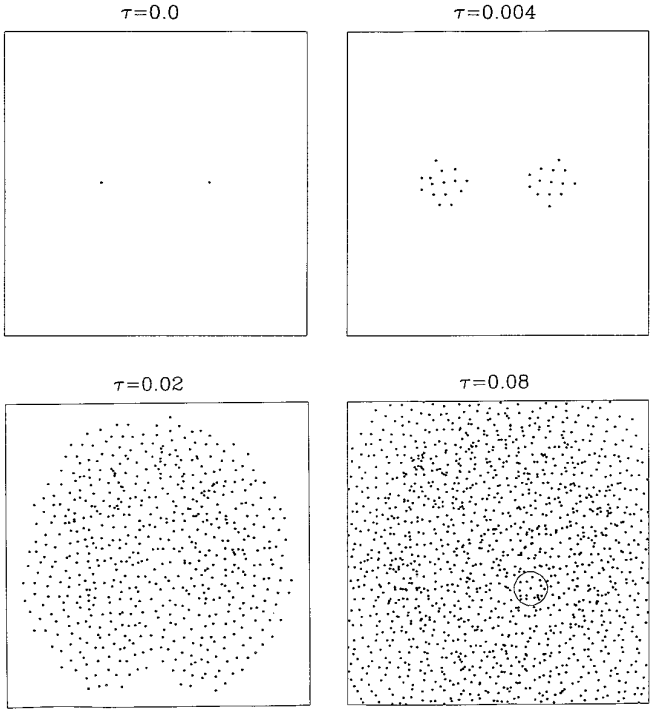

FIG. 4. Growth in number of vortices for a flow starting from a pair of counterrotating point vortices at Re = 50. The small circle indicates the size of the neighborhood.

TABLE I

Number of Vortices at Reynolds Number 50

Decay of a vortex pair	
Time step	Number of vortices
0.004	2669
0.002	5307
0.001	10414

In circular cylinder computations reported elsewhere [54–56], we simply replaced vortices that moved very close by a single combined vortex. In those computations we did not experience a significant increase in scaled vortex density with the Reynolds number.

It should be emphasized that condensing nearby vortices into single vortices is not the same as the need to regenerate the mesh in particle methods. First, the only purpose here is merely to increase the numerical efficiency, not to maintain accuracy. Our computation can continue without it, although at lower efficiency. Second, there is no need to produce a new ordering, or association, of the computational points; there is no repartitioning of the domain; there is no quadrature rule to update. We simply give one vortex the combined strength and location and we drop the other vortex from the further computation.

It is even possible to incorporate this condensation directly into the redistribution process itself. For a vortex located close to another vortex, we might simply try to find a solution to the redistribution equations that does not involve the vortex itself. The vortex then loses all its circulation and can be removed. The redistribution fractions could be required to be positive as before, or a less restrictive condition might be imposed to remove even more vortices. In particular, the convergence analysis for Stokes flow in Section 4 would not be affected if the fractions were merely bounded in the l_1 norm and the circulation was allowed to grow by a relative amount $O(\Delta t)$. More research is needed to settle these points.

8. NUMERICAL RESULTS

This section discusses actual computations of two diffusing flows using the redistribution method. The first is the diffusion of a point vortex in time, and the second is the diffusion of a pair of counterrotating point vortices. Addi-

tional physical discussions of the results below, including the long time behavior, can be found in [65]. Results of computations of flows about rotating and translating cylinders using our scheme can be found in [54–56].

8.1 *Point Vortex*

The problem of a single point vortex diffusing out in an infinite domain is a good test case since an exact solution exists; its form is that of a diffusing point heat source governed by the heat equation. Oseen [37] pointed out that this provides a solution not just to the linear Stokes equations, but also to the nonlinear Navier–Stokes equations. Further, while convection is trivial for the exact solution, it is not for the discretized solution: the discretization produces noncircular streamlines as well as numerical errors.

The Reynolds number will be defined as $\Gamma/2\pi\nu$ and the flow is normalized to have a nondimensional circulation $\Gamma = 2\pi$. Numerical results will be presented for Reynolds numbers of 0, 10, and 50. We will use a scaled viscous time $\tau \equiv \nu t$. In terms of this scaled time, the exact solution is independent of the Reynolds number. We computed the range $0 \le \tau \le 0.3$ using a numerical time-step $\Delta\tau = 0.002$. We also repeated all computations at $\Delta\tau = 0.004$ to verify their accuracy. As explained in earlier sections, we performed redistribution over a neighborhood of radius $\sqrt{12\,\Delta\tau}$. Where no solution to the redistribution problem could be found, we added new vortices at a distance $\sqrt{6\,\Delta\tau}$. We used the vortex core (4) in integrating the convection processes, and the core (22) to evaluate pointwise values of the vorticity distribution. All computations were carried out in 32-bit precision on a VAX4000-300 computer running VMS V6.0.

Other investigations of similar flows have focussed on the average square radius of the vortex, $\overline{r^2} = \sum_i \Gamma_i(x_i^2 + y_i^2)/\sum_i \Gamma_i$, which grows as 4τ. The redistribution scheme reproduces this growth exactly due to (18), while the vortex blob method preserves it during the convection step. Indeed Fig. 5 shows excellent agreement, indicating that the solution of the redistribution equations and the numerical integration of convection are accurate. The relative error is of order 10^{-6}. It is clear that such accuracy could not be achieved using the random walk method with the same number of vortices [45, 50, 24, 23].

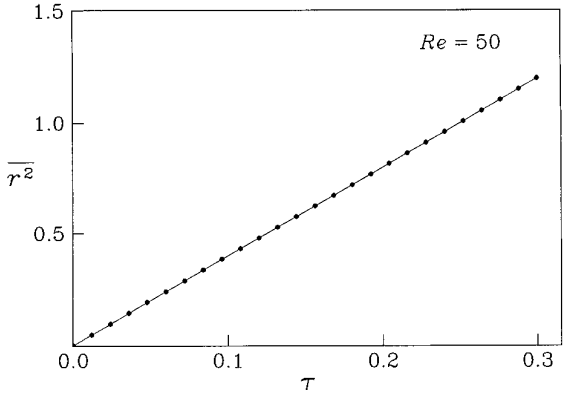

FIG. 5. Growth in mean square radius of a single diffusing vortex for Re = 50. The solid line is exact and circles are computed.

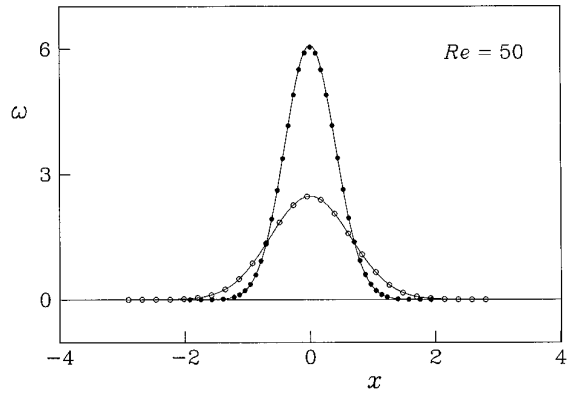

FIG. 6. Vorticity distribution of a diffusing point vortex for Re = 50 along the horizontal symmetry axis at times $\tau = 0.082$ and 0.202. The solid line is exact and circles are computed.

Net circulation is conserved to six digits accuracy. For Stokes flow the mean vortex position remains at the origin to an accuracy of order 10^{-7}. This error increases to the order of 10^{-5} for higher Reynolds numbers in agreement with the chosen truncation error in the fast velocity summation scheme [60].

Despite the arbitrary locations of the vortices, it is possible to obtain accurate pointwise vorticity values. For example, Fig. 6 compares the numerical vorticity along a horizontal line with the exact solution. A close examination shows that since the infinite-order smoothing function is not entirely positive, it produces a very small negative vorticity at the tail end of the distribution. The maximum errors at $\tau = 0.082$ and 0.202 are 0.046 and 0.016, respectively, which amounts to 0.75% and 0.65% of the maximum vorticity.

Table II shows the numer of vortices and computational times for the fast summation and redistribution parts of the computation. It should be noted that the time needed for convection is increased due to subdivision of the convective time-step. Since the flow starts from a concentrated vortex, during the first few time-steps the vortices rotate rapidly about each other. To limit the angle that the vortices can move, the early convection steps were subdivided further. For example, at Reynolds number 50, the first convection step was subdivided into 50 equal parts. The subdivision was then decreased inversely proportional to the time-step number. Since the number of vortices at early times is very small, the additional work is limited.

TABLE II

Computational Times for a Point Vortex at $\tau = 0.202$ and $\Delta\tau = 0.002$

Decay of a point vortex			
Reynolds number	Number of vortices	Convection CPU seconds	Diffusion CPU seconds
0	2482	0	2363
10	2863	1723	2724
50	3896	3320	4216

As discussed at the end of Section 5, the diffusion time may be greatly reduced for Stokes flow and most likely also for other Reynolds numbers, by not solving the redistribution equations from scratch each time. step. As discussed at the end of Section 7, the number of vortices at high Reynolds number may be reduced by some form of regeneration of the vortex distribution.

8.2. *Counterrotating Vortex Pair*

The second flow studied starts from two counterrotating vortices spaced a unit distance apart. Such a vortex system will drift in the direction normal to the line connecting the vortex centers. Using the same normalizations as for the single vortex, the initial drift velocity is unity (which explains our choice of normalizations). This flow presents a case in which convection is not trivial even for the exact solution. It also involves mutual cancellation of negative and positive vorticity.

For Stokes flow, the exact solution is simply the superposition of two single vortices. Figure 7 shows the vorticity distribution along the connecting line. There is excellent agreement between the exact solution and the computed vorticity. This is also evident from the computed vorticity lines in Fig. 8. A standard random walk approach would experience considerable difficulty with this flow, since vanishing vorticity is represented by roughly equal amounts of negative and positive vortices of the initial strength [62]. The computed decay of the circulation in a half plane is compared to the analytic solution in Fig. 9. A physical

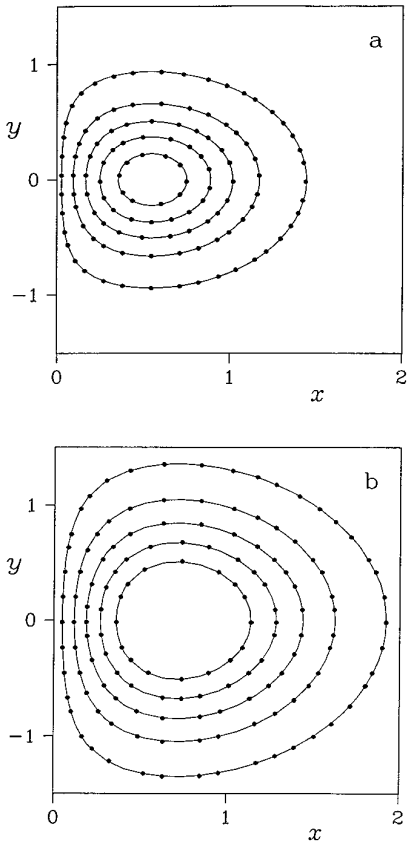

FIG. 8. Isovorticity contours (a) at time $\tau = 0.082$: $\omega = 5.00, 3.85, 2.70, 1.55, 0.40$; (b) at time $\tau = 0.202$: $\omega = 1.40, 1.10, 0.80, 0.50, 0.20$. The solid line is exact and circles are computed.

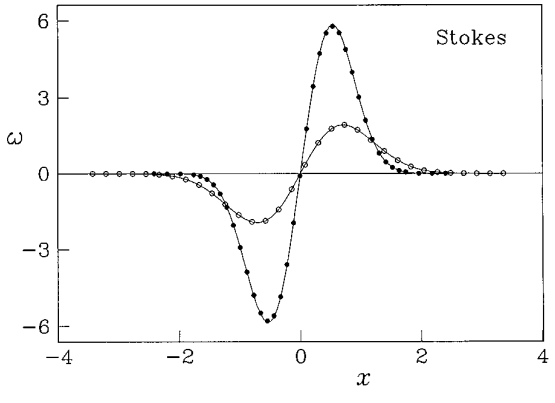

FIG. 7. Stokes flow solution for a pair of counterrotating vortices: vorticity along the connecting line at times $\tau = 0.082$ and 0.202. The solid line is exact and circles are computed.

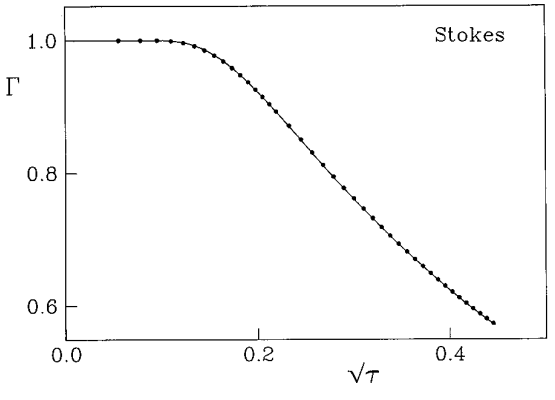

FIG. 9. Decay of the circulation in a half plane. The solid line is exact and circles are computed.

discussion of the results and comparison with related work may be found in [65].

Unfortunately, for a nonzero Reynolds number no exact analytic solution is available to show how well the nontrivial convection effects are represented. Instead, we use the expansion derived in [65] which is valid for sufficiently small times. To obtain high resolution for small times, we repeated the computation for Reynolds number 50 at $\Delta\tau = 0.00025$.

Figure 10 shows vorticity contours at two early times. The dashed curves in this graph represent the analytical solution given by simple superposition of single vortex solutions, while the solid curves include the next order in the small time expansion. The difference between the curves represents nontrivial convection effects. Since the two curves are close together in the first graph, we expect the exact solution to be close to the solid curve. The computations do indeed reproduce this curve closely. At the later time, the small time expansion is probably no longer very accurate, since it is based on the approximation that the size of the vortices is small compared to their distance.

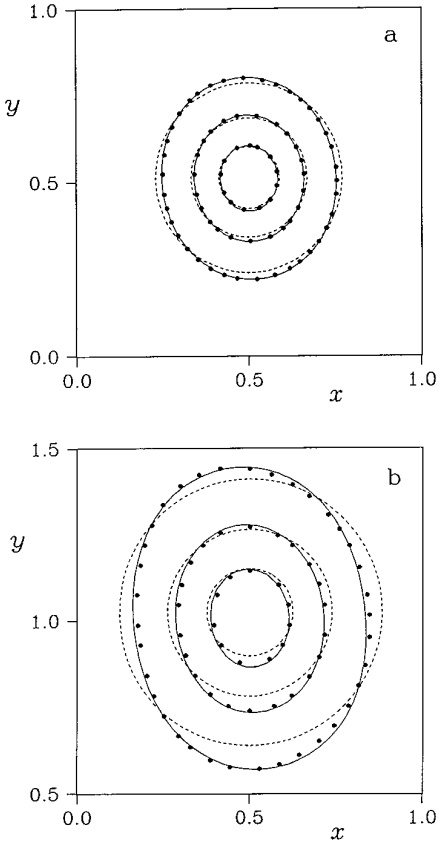

FIG. 10. Isovorticity contours for a counterrotating vortex pair for Re = 50: (a) at time $\tau = 0.01025$: $\omega = 40, 24, 8$; (b) at time $\tau = 0.02050$: $\omega = 20, 12, 4$. The dashed and solid lines represent orders of approximation in the analytic expansion. Circles are the computed.

TABLE III

Computational Times for Counterrotating Vortices at $\tau = 0.202$ and $\Delta\tau = 0.002$

Decay of a vortex pair			
Reynolds number	Number of vortices	Convection CPU seconds	Diffusion CPU seconds
0	2981	0	2796
10	3525	2593	3524
50	5307	4303	5631

The inaccuracy is reflected in sizeable differences between the solid and dashed curves. While the exact solution is now no longer certain, we still assume that it is accurately represented by the numerical solution. One reason is that the computed solution is close to the solid curve; second, we expect the next higher-order term in the small time expansion to have three periods along a contour, which seems to agree with the number of curve crossings in Fig. 10. Table III shows the number of vortices and computational times at $\tau = 0.202$.

This study used algebraically decaying vortex cores, rather than the somewhat more usual exponentially decaying ones. To check the effect, we repeated the computation of Fig. 10, using exponentially decaying cores. We used a second-order Gaussian core [8] for convection, while at the end of the computation, the vorticity was evaluated using a fourth-order Gaussian core [8]. The results in Fig. 11 show that the effect is negligible, although the Gaussian results seem slightly less accurate, based on the comparison with the small time expansion at the earlier time.

9. DISCUSSION

In this section we want to compare the redistribution method with other vortex methods, and in particular with the particle strength exchange method and the related method by Fishelov. Although in other studies we have found the random walk method to be robust, simple to apply, and quite reliable, it is often hard to distinguish physical changes in the solution from the numerical errors. The difficulty is that the errors, which can be quite noticeable, are random. In fact, this was our motivation to develop the redistribution method as a more accurate alternative that still avoids the need for order in the vortex distribution or associations between the vortices.

Like the particle methods, the redistribution method simulates the diffusion of a vortex by moving circulation to neighboring vortices. This similarity is rather superficial, however, since Lagrangian finite difference, finite element, and spectral representations of the diffusion process would all do this. By construction, the redistribution method is

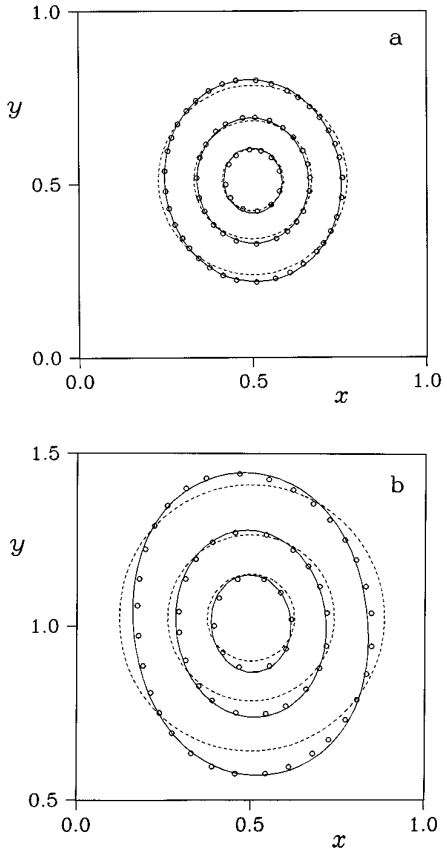

FIG. 11. The effect of using exponentially decaying core shapes instead of algebraically decaying ones.

closest to a finite difference method, rather than a particle method, and reduces to one when the distribution of the vortices is uniform. For that reason, it might be considered to be a computed finite difference formula.

The various numerical methods mentioned above differ in the way the amounts of circulation to be moved to neighboring vortices are determined, and in the number of neighboring vortices involved. In particular, the particle methods transfer circulation between vortices proportional to the local value of a diffusion kernel. In contrast, the redistribution method computes the amounts to be transferred from procedures similar to ones used to construct finite difference formulas. This allows the redistribution method to satisfy the necessary equations using only a finite number of vortices, similar to a finite difference method.

Particle methods cannot do this. For these methods to converge, the diffusion kernel must have a size δ that is asymptotically large compared to the point spacing h; it must be integrated correctly [21, 23]. As a result, in particle methods the diffusion of a vortex involves an unbounded number of neighboring vortices, as in a spectral method.

In practice, $\delta = O(h^p)$, $0 < p < 1$, and p may be as small as 0.08 or 0.1 [16]. Other computations have used much smaller cores, but remeshed frequently to uniformly distributed vortices. For example, Pépin [47] uses $\delta/h = 1.8$ to compute the flow around a cylinder at Reynolds number 9500, but he reports remeshing every six or seven time-steps.

9.1. *Resolution of Short Scales*

The requirement of particle methods that the diffusion core size must be asymptotically large compared to the vortex spacing can be a disadvantage. It is not the vortex spacing, but the larger core size that limits the smallest scales that can be resolved during the computation.

This is especially so since the core size must be chosen before the computation can be conducted, at a time when little precise information about the flow to be computed is likely to exist. When widely differing straining rates cannot be excluded beforehand, it may be tempting to make the core sufficiently large to ensure that it will remain well resolved during the computation. A choice that optimizes both the errors in small scales and the errors in discretizing the core may not be very easy to make.

In contrast, the redistribution method proceeds without a core. The smallest scales for which the computation is meaningful are limited by the redistribution radius, which is of the order of the point spacing, not asymptotically large compared to it.

If the vorticity field itself is desired at some given time, we still need to evaluate it using an evaluation core, but this is a different problem. At the evaluation stage, all information about the solution is known, and the core can be selected, based on the actual solution properties at the given time. In practice, we reduce the core size until short wave errors start to show up. In principle, it would even be possible to select a core size based on the local solution properties, but so far we have always used a spatially constant core.

Furthermore, the evaluation core does not affect the actual computation; all information about the short waves remains available for the convection algorithm to use.

In practical applications, significant short scales might be due to rapid changes in boundary conditions or due to strong straining during the vortex ejection from boundary layers that follows the unsteady separation process discovered by Van Dommelen and Shen [56]. As a model example we will address the case of a diffusing point vortex. This is the fundamental solution of the diffusion equation and presents the limiting case where the entire initial vorticity distribution has such a small scale that it computationally appears to be a point. According to the exact solution for a diffusing vortex, the vorticity diffuses out over a typical distance $\sqrt{\nu t}$. The particle methods are inaccurate for times for which this diffusion distance is still small or finite compared to the kernel size δ.

The method of Degond and Mas-Gallic [21] leaves the vortex largely undiffused during early times; it diffuses only a small fraction of the vortex over an area of typical size δ. Instead, it should diffuse all of the vortex over the distance $\sqrt{\nu t}$. The method of Fishelov [23] initially also leaves the vortex undiffused; it simulates the diffusion by the creation of negative and positive vorticity within a region of size δ.

While the smallest vortex size that the particle methods can resolve is determined by the size of the kernel, asymptotically the point spacing must be much smaller. As a result, there is a range of times for which the vortex is already large compared to the point spacing, but small or finite compared to the kernel. For those times, the particle methods give inaccurate results.

On the other hand, the redistribution method gives a valid approximation to the exact solution as soon as the size of the vortex $l = \sqrt{\nu t}$ becomes large compared to the typical point spacing h_v. We simply take the size of our smoothing function $\delta = h_v^\alpha l^{1-\alpha}$ for some $\alpha < \frac{1}{2}$. According to the error bounds (25) and (33), for α close to zero this produces an L_2 relative error of almost $O(h_v/l)^M$, which is the best accuracy that can be expected. This means that we are using a smoothing function with a variable core size. However, this has no consequences; the redistribution process is independent of the smoothing function. The smoothing function is merely used in the final evaluation of the solution and can be optimized for the instantaneous properties of the computed solution.

For still earlier times, after only a finite number of time-steps, the size of the delta function is of the order of the point spacing, and an accurate representation of the vorticity is not possible. This is not a shortcoming of the redistribution method; it cannot distinguish whether the initial condition is a true point vortex or a spike of a size smaller than the spacing of the numerical points. Thus the solution is truly indeterminate as long as the vortex distribution is of the order of the point spacing. The best that can be hoped for during these times is that the numerical solution gives the correct typical size of the vortex distribution. Since the redistribution method uses a finite scaled redistribution radius R, it restricts diffusion to a region of the correct order of magnitude. Moreover, except for the uncertainty in the initial data, the correct root mean square radius of the vorticity distribution is achieved.

How important the difference is for practical computations remains to be decided. As mentioned in the beginning of this section, results for the ratio of point spacing to core size vary. Some computations have used a ratio quite close to one. These computations have maintained a highly efficient vortex distribution by frequent remeshing. However, this does reintroduce concerns with regard to regeneration times, mesh generation, interpolation errors, quadrature errors, etc. that a truly Lagrangian computation attempts to avoid.

In any case, in practical applications we did find that our method works well at relatively low numbers of vortices. For a circular cylinder at a Reynolds number of 9500, we found that our method at about 60,000 vortices [54] gave results that remain unchanged when the number of vortices is increased further and a separated vorticity distribution that is in excellent agreement with the particle computations of Koumoutsakos and Leonard [36], which use much more vortices. For the number of vortices we used, the smallest scales, such as the boundary layer thickness, are not much larger than the typical point spacing.

We do want to point out a concern about our method when it is applied to vortices arranged according to a smoothly varying mesh distribution. Our method was not designed for such a purpose; we were interested in truly Lagrangian computations when convection has thoroughly mixed the vortices. Yet our method can be used on a uniform vortex distribution as well, since it will simply generate an explicit finite difference formula with good conservation and positivity properties. However, when our method is applied on a smoothly varying mesh of vortices, instead of a uniform one, our choice to solve the redistribution equations using a least maximum procedure is probably not the best one. The resulting redistribution weights do not always depend smoothly on the vortex positions. This will produce unnecessary short wave errors from the long wave components. But since our scheme seems, at present, quite inefficient for smoothly varying vortex distributions, there may not be much point in attempting to formulate an alternative to the least maximum procedure.

9.2. *Automatic Remeshing*

One of the main difficulties in a Lagrangian determination of diffusion processes is that convection can cause a severe deterioration in the spatial distribution of the computational points. For the particle methods, the most important consequence is a loss of accuracy in the quadratures [42, 46]. To maintain accuracy, such methods require a careful monitoring of the particle overlap at all times [34, 47]. Solutions for extended times require periodic reinitialization or remeshing of the particle distribution [47, 34, 10].

The influence of the remeshing process and the time period between remeshings are additional sources of errors and uncertainties. Some particle computations have reported remeshing every six or seven time-steps [47]. This suggests the question at which time a Lagrangian computation stops being mesh-free or truly Lagrangian.

The redistribution method does away with these difficulties. Its only constraint on the point distribution is that a positive solution to the redistribution equation exists. If

there is none, a new vortex is added to create one. As a result, the point distribution is implicitly checked at each vortex at each time-step, and restored before it can affect our error estimates.

It is important to reiterate that our computation is truly mesh-free. We simply add a new vortex when we need one. We do not create a new partitioning of the domain; we do not create new quadrature rules based on the new vortex and its neighbors; we do not make any associations between the new vortex and its neighbors.

In computations reported elsewhere [56], we have also searched the existing vortices for any ones that are no longer truly useful and simply have thrown them out, after distributing their vorticity over the neighboring vortices. No other steps were needed.

In our computations, we do not even bother creating a mesh of vortices around our solid bodies. For example, Shankar [56] computed vortices bouncing off a circular cylinder by merely placing the incoming vortices at the desired initial position. The vortices introduce slip at the surface of the cylinder, and new vortices are created at the wall to cancel the slip. Our method automatically takes care of extending this vorticity distribution at the wall out into the field. This would work the same way, regardless of the number and complexity of the solid bodies present.

The mesh-free nature of our computation also allows us to restrict vortices to exactly the regions where we need them. For example, we do not include vortices of zero strength in our computation as particle computations have done [47].

9.3. *Computational Speed*

The computational effort required by our method is an area of considerable concern, but more as a practical, rather than as a theoretical, issue. From a rigorous theoretical point of view, the redistribution method is in fact superior to the particle methods with respect to computational time. After all, in the limiting process in which convergence is achieved, the particle methods must transfer their vorticity to infinitely many neighbors, requiring infinitely many computational operations. Although the redistribution method must elaborately compute the fractions to transfer to the neighbors, only a finite number of neighbors is involved, making the work asymptotically finite.

However, the situation is much less clear than this argument might suggest. The comparison above assumes that the particle and redistribution methods use the same number of vortices and time-steps. Yet, a particle method such as Fishelov's [23] can be exponentially accurate. While the redistribution method can have any fixed order of accuracy, at least for the Stokes equations, it cannot be exponentially accurate using a finite number of points. For infinitely smooth initial data, an exponentially accurate particle method would asymptotically need much less points than a fixed-order redistribution method, making the above comparison of times meaningless.

Furthermore, under realistic conditions the number of neighboring vortices affected in a particle method is not likely to be very large. Since it is significantly less work to transfer a vorticity fraction onto a vortex than to compute that fraction from a linear programming problem, finite or not, the asymptotic estimate is clearly misleading for practical applications. This is particularly so for the particle methods that remesh every few time steps, e.g., [47, 34]. These may involve as little as on the order of 200 neighboring vortices.

In any case, from a practical point of view the real question is whether the computational time for the redistribution step leads to an unacceptable increase in the total computational time. If the time for redistribution would be much larger than the time needed to find the velocity field, it would significantly reduce the problems that could be addressed with the method. Our computational examples in Section 8 showed that this is not the case. To put it in perspective, we may note that in order to resolve length scales only twice as small, a computation would need 16 times the computational effort in two dimensions and 32 times in three.

Furthermore, as discussed in Section 5, we have not yet made serious attempts to reduce the time required for our method. Since there seems theoretically no limit to the reduction in computational effort that might be achievable, this seems a promising area for further research.

A true saving of computational time compared to particle methods can occur if the initial vorticity is sparse. The redistribution method, with its capability to deal with randomly distributed, independent vortices, need use only vortices in regions in which vorticity exists. New vortices are automatically added when the region with vorticity expands. For example, for the diffusion of a point vortex we started with a single vortex and we let our method add vortices automatically. Particle methods typically start out with a large number of vortices, most of which are inactive at those early times. (An improvement suggested by Pépin [47] is to allow the number of particles to be increased during remeshing, thus allowing fewer particles to be used during the first stages.)

9.4. *Simplicity*

It has been argued that the redistribution method introduces significant additional complexity in a vortex computation. We cannot agree with this sentiment, at least not if a fast summation algorithm is used to find the velocity field in the convection step. In its simplest implementation, followed in this paper, the redistribution method needs to do two things for each vortex: identify the neighboring vortices and solve the redistribution equations.

Neighboring vortices are already identified by the fast summation procedure used to find the velocities. In our program, that part of the fast summation process was simply repeated in the redistribution stage to account for the different neighborhood sizes in fast summation and redistribution. Thus there is no significant further complexity with regard to this requirement.

While our method also requires the solution of the redistribution equations, this does not truly add complexity to our procedure either. Solution of a linear system of equations is a standard mathematical problem, not a problem specific to the redistribution method. Ideally, the redistribution equations are merely handed to a ''canned'' library routine for solution. Actually, we wrote our own subroutine based on an algorithm found in literature [1].

If there is no solution to the redistribution equations, a vortex needs to be added. This too, is a very simple process. We simply try a few locations for this vortex and stick it in the largest hole we find.

It is important to note that none of the above requirements depends on what flow is being computed. The complexity of the flow does not affect them. The length scales and the strength of the convective processes do not affect them. No parameters need to be chosen based on the flow properties. In other words, these issues need to be addressed only once.

While admittedly the random walk method is even simpler to apply than our method, in our opinion particle methods are not. First, while our method extends the vortex distribution automatically from the solid surfaces into the field, a particle method faces a separate mesh generation problem: it needs to create an effective partitioning and quadrature procedure. Such problems can become difficult for complex configurations. Yet, in order to resolve the diffusion cores with the minimum number of vortices, an effective vortex distribution is highly desirable.

Further, a particle method needs to monitor its vortex distribution. It needs to formulate criteria that determine whether a given distribution, with widely varying local properties, needs to be updated globally. Or it needs to address the even more complex issue of local updating. It needs to update the vortex distribution without introducing artificial diffusion or smearing steep gradients. The best choices for the time interval between updating, the new point distribution, the transfer of vorticity between meshes, etc., all depend on the actual flow being computed. Optimal choices will require trial and error.

Furthermore, particle methods face the need to select a smoothing function to perform diffusion. This function must be selected a priori in order to be able to perform the computation of the diffusion process. Since at that time not much may be known about the flow to be computed, an optimal choice will not always be a simple task.

The redistribution method has it much easier since the computation of the diffusion processes does not depend on a smoothing function. A smoothing function is only used in the final evaluation of the results. At that time, much more information is available since the strength of the vortices has already been determined. It also makes it possible to optimize the size or shape of the smoothing function based on the computed properties without repeating the complete computation.

Admittedly, in an actual computation at nonzero Reynolds number, a smoothing function must still be used to find the velocity field. However, experience indicates that the choice of this smoothing function is often not very critical. For example, Milinazzo and Saffman [45] obtained meaningful random walk results with a very small smoothing function. Goodman, Hou, and Lowengrub [29] show that no smoothing function is necessary if the vortices are initially uniformly spaced.

Thus the redistribution method is simple to apply and flexible. On the other hand, additional complications can certainly arise if its performance is to be optimized. For example, while extended convection is not a problem for accuracy as in particle methods, it can certainly reduce numerical efficiency. As discussed in Section 7, it would be desirable to combine vortices that approach very closely. While there are no associations between vortices that need to be maintained, it would still have to be shown that this process does not introduce instability or inaccuracy. We note, however, that we have used it without difficulty elsewhere [54–56].

Similarly, solving the redistribution equations from scratch at every time-step seems wasteful; as pointed out in section 5, these equations change little from one time-step to the next. Yet, to use the solution of one time-step during the next one would clearly add complexity, such as what information to save from one time-step to the next and how to update the old solution.

For higher order of accuracy, the conditioning of the redistribution equations needs to be considered. It would need to be determined whether it might be advantageous to recast the equations in other equivalent forms. The effects of numerical errors in the solution process would need consideration. Again, such considerations would be independent of the flow being considered.

9.5. *Conservation Laws and Positivity*

The fact that the redistribution method computes the individual redistribution fractions makes it easier to obtain certain desirable properties. In particular, it allows the conservation laws to be satisfied exactly. Even when resolution is very poor, such as initially for a diffusing point vortex, at least no false circulation or linear and angular momentum will be created.

A random walk procedure conserves only circulation exactly. While corrections are possible that conserve the center of vorticity [45, 13], subgroups of vortices can still perform an appreciable net motion without a physical mechanism causing it. The particle scheme proposed by Fishelov [23] is not conservative unless a corrected rule is used to perform the integrations in her convolution, but the potential high-order of accuracy may make this unimportant. The particle methods do not satisfy conservation of center of vorticity exactly when the particle distribution becomes nonuniform.

Another advantage of the redistribution equations is that they tend to localize the errors in velocity that result from the numerical diffusion. To be precise, the redistribution equations ensure that the leading order decay terms of the error in velocity vanish exactly.

Another desirable property is the positivity of the redistribution fractions; it assures that regardless of the numerical inaccuracy, no false reversed vorticity is created. Whether particle methods satisfy this constraint depends on factors such as the choice of smoothing function and of the time discretization. For example, Fishelov's method does not satisfy positivity and can generate reversed vorticity, although the amount should be very small if the vorticity distribution is sufficiently smooth. The integral constraints given by Degond and Mas-Gallic [21] show that third-order accurate particle schemes do not satisfy positivity.

It may seem surprising that for the Stokes equations the redistribution method can achieve any order of accuracy with positive fractions, while the particle methods cannot. The reason is that the redistribution method discretizes convection for a finite time-step, rather than an infinitesimal one. In particular, if we let the time-step tend to zero in the redistribution method, while keeping the location of the vortices fixed, the scaled spacing between the vortices would tend to infinity. In Appendix A it is shown that the redistribution equations do not have a positive third-order solution if the scaled spacing is more than a finite value. Thus a high order of accuracy can only be achieved for a finite time-step.

For the Navier–Stokes equations a finite time-step is a mixed blessing; the splitting into viscous and inviscid steps should not introduce an error larger than the spatial order of accuracy. Note, however, that the time-step is an order smaller than the spatial resolution. In an unbounded domain, Strang splitting with reversal of the order of the steps [6] would be fourth-order accurate with respect to space.

10. CONCLUSIONS

By construction, the redistribution method might be considered to be a computed finite difference formula for vortices with disordered locations. The primary intent of this paper was to verify that the basic concepts of the method are sound. We wanted to show that it is possible to use disorganized vortex locations, that it is allowable to be vague about what the individual vortex strengths really mean, beyond the statement that a meaningful vorticity field can be *constructed* from them by some convolution if desired, and that flow computations with very strong convective effects can be continued without any reorganization of the vortex locations and still not result in additional errors or a blowup of the vortex density. These properties were verified using the linear convergence analysis and the computed examples with analytical solutions. We refer elsewhere for actual physical applications using the redistribution method; see [54–56, 65].

From all the results, the redistribution method emerges as a viable alternative to other procedures such as random walk and particle methods. It shares with the random walk method the property that the vortices are truly independent; there is no need for regularity in their positions, no partitioning of the domain, no numerical quadrature rule. These are very desirable properties for flows with strong convective straining effects, or when complex geometries are involved. Yet, our results with the redistribution method have been much better than our own earlier results with the random walk method; for example, we refer to our computations of the separation process from a circular cylinder elsewhere, [64, 54] for the random walk and redistribution methods, respectively. In our opinion, these differences are well worth the additional computational time required to perform the redistribution step.

Such improved accuracy might also be obtained using particle methods, but particle methods do not allow independent vortices. Our method has some other advantages over particle methods, such as the fact that new vortices are added automatically, when the vorticity diffuses toward new locations or straining depletes the regions of vortices. Such new vortices are created only when and where they are needed. Our method can also show a better performance for small scale phenomena; compare the discussion about a diffusing point vortex in Subsection 9.1. Further advantages are that the conservation laws are satisfied exactly and that our method can preserve the sign of the vorticity exactly even at higher orders of accuracy.

However, in view of the initial nature of our investigations, it should not be surprising that there are some significant areas in which our method cannot yet compete with particle methods. For example, Fishelov's method [23] readily provides a high order of accuracy. In theory, this is also possible with our method, but it remains unknown how well this will work out in practice.

Further, there is a disadvantage of speed. The work that is required for the redistribution method is of the order of the time required for convection. Particle methods are much faster; the computational time needed for diffusion

in typical particle computations is negligible compared to the time needed for convection. Yet, we must point out that so far, we have not made any serious attempt to reduce our computational time. We certainly have no doubt that the brute force approach we have followed so far is unnecessarily inefficient. Interestingly, in theory the reduction in computational effort could still be arbitrarily large. Future studies will have to determine how much of that is really possible.

APPENDIX A: DETAILED DERIVATIONS

This appendix gives some derivations mentioned in the paper. First is the derivation of the redistribution equations (16) and the following. We write the difference between the Fourier transform of the redistributed vorticity (13) and the exactly diffused vorticity (14) in terms of the viscous scale $h_v = \sqrt{\nu\,\Delta t}$ and the scaled relative positions (15):

$$\hat{\omega}^{n+1} - \hat{\omega}_e^{n+1} = \hat{\phi}(k\delta) \sum_i \Gamma_i^n e^{i\mathbf{k}\cdot\mathbf{x}_i} \left(\sum_j f_{ij}^n e^{-ih_v \mathbf{k}\cdot\boldsymbol{\xi}_{ij}} - e^{-h_v^2 k^2} \right). \tag{40}$$

In the redistribution method, this error is made small, of order $O(h_v k)^{M+2}$, or $O(\Delta t)^{(M+2)/2}$ for any finite wavenumber k, by equating the Taylor series expansions of the two terms within the parentheses to that order. This produces, for any $m \le M + 1$,

$$\begin{aligned} \sum_j f_{ij}^n (k_1 \xi_{1ij} + k_2 \xi_{2ij})^m &= 0 && (m \text{ odd}), \\ &= \frac{m!}{(\frac{1}{2}m)!} (k_1^2 + k_2^2)^{(1/2)m} && (m \text{ even}). \end{aligned} \tag{41}$$

Expanding using the binomial theorem, the individual equations become, for $m_1 + m_2 \le M + 1$,

$$\begin{aligned} \sum_j f_{ij}^n \xi_{1ij}^{m_1} \xi_{2ij}^{m_2} &= 0 && (m_1 \text{ or } m_2 \text{ odd}) \\ &= \frac{m_1! m_2!}{(\frac{1}{2}m_1)!(\frac{1}{2}m_2)!} && (m_1 \text{ and } m_2 \text{ even}). \end{aligned} \tag{42}$$

These are the redistribution equations written out in (16) and the following.

The remaining error in the Fourier transform, needed in Section 4, is according to the Taylor series remainder theorem,

$$\begin{aligned} \hat{\omega}^{n+1} - \hat{\omega}_e^{n+1} = \hat{\phi}(k\delta) \sum_i \Gamma_i^n e^{-i\mathbf{k}\cdot\mathbf{x}_i} \\ \times \left\{ \frac{(kh_v R)^{M+2}}{(M+2)!} \sum_j f_{ij}^n (\cos\alpha_{ij\mathbf{k}} + i\cos\beta_{ij\mathbf{k}}) \left(\frac{\mathbf{k}\cdot\boldsymbol{\xi}_{ij}}{kR} \right)^{M+2} \right. \\ \left. - \frac{(-k^2 h_v^2)^{(1/2)M_e+1}}{(\frac{1}{2}M_e + 1)!} e^{-\gamma_k^2} \right\}, \end{aligned} \tag{43}$$

where the values of $\alpha_{ij\mathbf{k}}$, $\beta_{ij\mathbf{k}}$, and γ_k represent the undetermined midpoints in the remainder theorem, and M_e is the even integer M or $M + 1$.

To find the lower bound to the redistribution radius mentioned in Section 7, we integrate (41) over the unit circle to produce

$$\sum_j f_{ij}^n \xi_{ij}^m = 2^m (\tfrac{1}{2}m)! \quad (m \le M + 1; m \text{ even}). \tag{44}$$

Hence, in terms of the even integer $M_e = M$ or $M + 1$,

$$\max_j \xi_{ij}^2 \ge \sum_j f_{ij}^n \xi_{ij}^{M_e} \Big/ \sum_j f_{ij}^n \xi_{ij}^{M_e - 2} = 2M_e, \tag{45}$$

which implies the lower bound for the redistribution radius.

Up to fourth-order accuracy, this estimate for the minimum radius is precise. It may be verified by direct substitution into the redistribution equations that a positive second-order solution is obtained by spreading the fractions evenly over the circle with scaled radius $R = 2$. Similarly, a positive fourth-order solution is obtained by giving the vortex being redistributed a fraction 0.5 and spreading the other half evenly over the circle $R = \sqrt{8}$.

Next we verify an assertion made in Section 7: as long as all vortices are redistributed, the region containing the vortices must expand a finite scaled amount in each direction. To do so, we derive a lower bound $\xi_{1\max}$ to $\max_j(\xi_{1ij})$, using the cases $m_2 = 0$ and $m_1 = 0$, 1, and 2 of (42),

$$\begin{aligned} 8 &= \sum_j f_{ij}^n (|\xi_{1ij}| + \xi_{1ij})^2 + \sum_j f_{ij}^n (|\xi_{1ij}| - \xi_{1ij})^2 \\ &\le 4\xi_{1\max}^2 + \sum_j f_{ij}^n (|\xi_{1ij}| - \xi_{1ij}) 2R \\ &= 4\xi_{1\max}^2 + \sum_j f_{ij}^n (|\xi_{1ij}| + \xi_{1ij}) 2R \\ &\le 4\xi_{1\max}^2 + 4\xi_{1\max} R. \end{aligned} \tag{46}$$

The solution of the quadratic shows that

$$\xi_{1\max} \ge 4/(R + \sqrt{R^2 + 8}). \tag{47}$$

Applied to the vortex at the largest value of x, this value describes how much the vortex distribution needs to expand in the x-direction in order for the redistribution equations to be solvable at that vortex. Since the redistribution problem is independent of the angular position of the coordinate system, this minimum expansion applies in any direction.

Next we verify an assertion made in Section 7 and Subsection 9.5: for third-order accuracy or higher, the scaled spacing between the vortices cannot be arbitrarily large. Defining $\xi_{\min} = \min_{j\neq i}\{\xi_{ij}\}$ and using (44) for $m = 4$ and $m = 2$, it is seen that

$$32 = \sum_j f_{ij}^n \xi_{ij}^4 \geq \sum_j f_{ij}^n \xi_{ij}^2 \xi_{\min}^2 = 4\xi_{\min}^2. \tag{48}$$

Hence, the vortices cannot be spaced further apart than a scaled distance $\sqrt{8}$.

Finally we verify an assertion made in Section 7: there are finite values R and d so that a positive solution to the redistribution equations exists within the circle with scaled radius R, provided that there are no square holes exceeding a scaled size d in the distribution of the vortices. To do so, we first note that the diffusing delta function

$$f(\xi_{1i}, \xi_{2i}) = \frac{1}{4\pi} e^{-(\xi_{1i}^2+\xi_{2i}^2)/4} \tag{49}$$

gives an exact continuous solution to our redistribution equations, replacing $\sum_j$ by $\iint d\xi_{1i}\, d\xi_{2i}$. We now discretize this continuous solution using a cutoff at radius R and a subdivision of the remaining domain into squares of size d. We then select one vortex in each square and give it a fraction

$$f_{ij}^n = d^2 f(\xi_{1ij}, \xi_{2ij}) + d^2 \sum_{m,n} c_{mn} p_m(\xi_{1ij}) p_n(\xi_{2ij}), \tag{50}$$

where the $p_n(\xi)$ are for $|\xi| \leq 1$ polynomials of degree $M + 1$ satisfying

$$\int_{-1}^{1} p_n(\xi)\xi^m\, d\xi = \delta_{mn} \quad (m = 0, ..., M + 1) \tag{51}$$

and zero elsewhere. In the above expression for f_{ij}^n, the first term provides a positive approximate solution to the redistribution equations, since the sums over j become straightforward numerical approximations to the corresponding integrals of the continuous function f. The second term gives corrections that make this approximation exact for suitable values of the constants c_{mn}. When R is large enough and d is small enough, these corrections do not change the positivity of the first term. This can be seen as follows: since the polynomials are bounded, there is a finite maximum value for the constants $|c_{mn}|$ below which the correction terms cannot change the sign of the first term to f_{ij}^n. Further, since the redistribution equations give a system of equations for the c_{mn} that tends to a unit matrix, there is a value of d below which the maximum $|c_{mn}|$ can be bounded by a multiple of the maximum error due to the first term in (50). That error can be reduced to any finite amount by selecting a large enough R and a small enough d to make the numerical integrals sufficiently accurate. Hence the required positive total solution (50) can always be assured for some finite R and d.

At least for the case of first-order accuracy, $M = 1$, for any R greater than the minimum value $R = 2$, a finite hole size d exists that ensures a positive solution. This can be seen by selecting nine vortices to satisfy the redistribution equations. Eight of these are chosen as closely as possible to eight equally spaced points on the outside circle and given a nominal fraction $f_{ij}^n = 1/2R^2$, and the last point is chosen to be the vortex being redistributed and given a nominal weight $1 - 4/R^2$. This satisfies the redistribution equations approximately, and it is readily seen that for these nominal positions, the needed corrections in the weights to make the approximation exact can be bounded by the errors. Thus, similar to the derivation above, the corrections do not change the sign of the weights when d is small enough. The actual value of d is unknown, but clearly d must tend to zero when $R \to 2$; the allowed hole size must be small enough to ensure that there are vortices outside the circle $R = 2$ within which no solution exists.

ACKNOWLEDGMENTS

We thank the referees for a number of helpful comments that led us to improve discussions in Sections 7 and 9. This research was supported by AFOSR F49620-93-I-0013 and by NASA NAG-W-2930. Facilities were provided by FMRL and SCRI at FSU.

REFERENCES

1. N. N. Abdelmalek, *J. Approx. Theory* **20,** 57 (1977).
2. A. S. Almgren, T. Buttke, and P. Colella, *J. Comput. Phys.* **113,** 177 (1994).
3. C. R. Anderson and C. Greengard (Eds.), *Proceedings, AMS Seminar on Vortex Dynamics and Vortex Methods, Seattle, Washington, 1990,* Lectures in Applied Mathematics, Vol. 28 (Am. Math. Soc., Providence, RI, 1991).
4. C. R. Anderson, *SIAM J. Sci. Stat. Comput.* **13,** 923 (1992).
5. G. K. Batchelor, *An Introduction to Fluid Dynamics* (Cambridge Univ. Press, Cambridge, 1987).
6. J. T. Beale and A. Majda, *Math. Comput.* **37,** 243 (1981).
7. J. T. Beale and A. Majda, *Math. Comput.* **39,** 29 (1982).
8. J. T. Beale and A. Majda, *J. Comput. Phys.* **58,** 188 (1985).
9. J. T. Beale and C. Greengard, *Commun. Pure Appl. Math.* **47,** 1083 (1994).
10. P. S. Bernard, *J. Comput. Phys.* **117,** 132 (1995).
11. C. Börgers and C. S. Peskin, *J. Comput. Phys.* **70,** 397 (1987).

12. J. Carrier, L. Greengard, and V. Rokhlin, *SIAM J. Sci. Stat. Comput.* **9,** 669 (1988).
13. C.-C. Chang, *J. Comput. Phys.* **76,** 281 (1988).
14. A. Y. Cheer, *J. Fluid Mech.* **201,** 485 (1989).
15. J. P. Choquin and S. Huberson, *Comput. Fluids* **17,** 397 (1989).
16. J. P. Choquin and B. Lucquin-Desreux, *Int. J. Numer. Methods Fluids* **8,** 1439 (1988).
17. A. J. Chorin, *J. Fluid Mech.* **57,** 785 (1973).
18. A. J. Chorin, T. J. R. Hughes, M. F. McCracken, and J. E. Marsden, *Commun. Pure Appl. Math.* **31,** 205 (1978).
19. R. J. Clasen, *Commun. ACM* **9,** 802 (1966).
20. G. H. Cottet and S. Mas-Gallic, *Numer. Math.* **57,** 805 (1990).
21. P. Degond and S. Mas-Gallic, *Math. Comput.* **53,** 485 (1989).
22. P. Degond and F.-J. Mustieles, *SIAM J. Sci. Stat. Comput.* **11,** 293 (1990).
23. D. Fishelov, *J. Comput. Phys.* **86,** 211 (1990).
24. A. L. Fogelson and R. H. Dillon, *J. Comput Phys.* **109,** 155 (1993).
25. M. J. Fritts, W. P. Crowley, and H. Trease (Eds.), *The Free Lagrange Method,* Lecture Notes in Physics, Vol. 238 (Springer-Verlag, Berlin, 1985).
26. S. I. Gass, *Linear Programming* (McGraw-Hill, New York, 1985).
27. A. F. Ghoniem and F. S. Sherman, *J. Comput. Phys.* **61,** 1 (1985).
28. J. Goodman, *Commun. Pure Appl. Math.* **40,** 189 (1987).
29. J. Goodman, T. Y. Hou, and J. Lowengrub, *Commun. Pure Appl. Math.* **43,** 415 (1990).
30. C. Greengard, *J. Comput. Phys.* **61,** 345 (1985).
31. L. Greengard and V. Rokhlin, Report YALEU/DCS/RR-602, Department of Computer Science, Yale University, 1988 (unpublished).
32. O. H. Hald, *SIAM J. Numer. Anal.* **16,** 726 (1979).
33. O. H. Hald, *SIAM J. Numer. Anal.* **24,** 538 (1987).
34. P. D. Koumoutsakos, Ph.D. thesis, California Institute of Technology, 1993 (unpublished).
35. P. Koumoutsakos, A. Leonard, and F. Pépin, *J. Comput. Phys.* **113,** 52 (1994).
36. P. Koumoutsakos and A. Leonard, *J. Fluid Mech.* **296,** 1 (1995).
37. Sir H. Lamb, *Hydrodynamics* (Dover, New York, 1945).
38. A. Leonard, *J. Comput. Phys.* **37,** 289 (1980).
39. A. Leonard, *Annu. Rev. Fluid Mech.* **17,** 523 (1985).
40. D.-G. Long, *J. Amer. Math. Soc.* **1,** 779 (1988).
41. C. Marchioro and M. Pulvirenti, *Commun. Math. Phys.* **84,** 483 (1982).
42. J. S. Marshall and J. R. Grant, "A Lagrangian Collocation Method for Vorticity Transport in Viscous Fluid Flows," in *Forum on Vortex Methods for Engineering Applications, Albuquerque, New Mexico, 1995* (Sandia National Laboratory, Albuquerque, N.M., 1995), p. 173.
43. S. Mas-Gallic and P. A. Raviart, Internal Report R86013, Lab. Anal. Num., Université Pierre et Marie Curie, Paris, Francis, 1986; *C.R. Acad. Sci. Paris Sér. I* **305,** 431 (1987).
44. S. Mas-Gallic, "Deterministic Particle Method: Diffusion and Boundary Conditions," in *Proceedings, AMS Seminar on Vortex dynamics and vortex methods, Seattle, Washington, 1990,* Lectures in Applied Mathematics, Vol. 28, edited by C. R. Anderson and C. Greengard (Am. Math. Soc., Providence, RI, 1991), p. 433.
45. F. Milinazzo and P. G. Saffman, *J. Comput. Phys.* **23,** 380 (1977).
46. R. B. Pelz and Y. Gulak, *Bull. Am. Phys. Soc.* **39,** 1892 (1994).
47. F. M. Pépin, Ph.D. thesis, California Institute of Technology, 1990 (unpublished).
48. P. A. Raviart, "An Analysis of Particle Methods," in *Numerical Methods in Fluid Dynamics,* Lecture Notes in Math., Vol. 1127, edited by F. Brezzi, (Springer-Verlag, New York/Berlin, 1985), p. 243.
49. M. D. Rees and K. W. Morton, *SIAM J. Sci. Stat. Comput.* **12,** 547 (1991).
50. S. Roberts, *J. Comput. Phys.* **58,** 29 (1985).
51. G. Russo, *J. Comput. Phys.* **108,** 84 (1993).
52. T. Sarpkaya, *J. Fluids Eng.* **111,** 5 (1989).
53. J. A. Sethian and A. F. Ghoniem, *J. Comput. Phys.* **74,** 283 (1988).
54. S. Shankar, S.-C. Wang, and L. L. van Dommelen, "Simulating Diffusion in Vortex Methods Using a Vorticity Redistribution Technique," in *Forum on Vortex Methods for Engineering Applications, Albuquerque, New Mexico, 1995* (Sandia National Laboratory, Albuquerque, N.M., 1995), p. 105.
55. S. Shankar and L. L. van Dommelen, "A New Diffusion Scheme in Vortex Methods for Three-Dimensional Incompressible Flows," in *Second International Workshop on Vortex Flows and Related Numerical Methods, Montréal, Canada, August* 20–24, 1995, European Series in Applied and Industrial Mathematics, Société de Mathematiques et Industrielles (SMAI), to appear; http://www.emath.fr/Maths/Proc/procEng.html. [Also to appear as CD-ROM]
56. S. Shankar, Ph.D. thesis, Florida State University, in preparation.
57. C. Shih, L. Lourenco, L. L. van Dommelen, and A. Krothapalli, *AIAA J.* **30,** 1153 (1992).
58. J. C. Strikwerda, *Finite Difference Schemes and Partial Differential Equations* (Wadsworth & Brooks/Cole, Belmont, CA, 1989).
59. L. L. van Dommelen, "Unsteady Separation from a Lagrangian Point of View," in *ASME Forum on Unsteady Flow Separation, Cincinatti, Ohio,* 1987, FED 52, edited by K. Ghia, p. 81.
60. L. L. van Dommelen and E. A. Rundensteiner, *J. Comput. Phys.* **83,** 126 (1989).
61. L. L. van Dommelen, "Some Experiments on a Vortex Redistribution Method," in *American Mathematical Society regional meeting, Hoboken, NJ,* Oct 21–22, 1989.
62. L. L. van Dommelen, FMRL Report TR-3, Department of Mechanical Engineering, Florida State University, 1989 (unpublished).
63. L. L. van Dommelen, FMRL Report TR-4, Department of Mechanical Engineering, Florida State University, 1989 (unpublished).
64. L. L. van Dommelen and S.-C. Wang, "Determining Unsteady 2D and 3D Boundary Layer Separation," in *Symposium on Aerodynamics & Aeroacoustics,* edited by K.-Y. Fung (World Scientific, Singapore, 1994), p. 187.
65. L. van Dommelen and S. Shankar, *Phys. Fluids A.* **7,** 808 (1995).
66. G. S. Winckelmans and A. Leonard, *J. Comput. Phys.* **109,** 247 (1993).
67. N. N. Yanenko, *The Method of Fractional Steps,* English transl., edited by M. Holt (Springer-Verlag, Berlin, 1971).