Rezoning for Higher Order Vortex Methods

HENRIK O. NORDMARK

Department of Mathematics,
Center of Research and Advanced Studies (CINVESTAV), Mexico City, Mexico

Received May 16, 1989; revised August 29, 1990

The vortex method is a numerical method for approximating the flow of an incompressible, inviscid fluid. We consider the two-dimensional case. The accuracy depends on the choice of the cutoff function which approximates the delta function, on the cutoff parameter δ and on the smoothness of the initial data. We present a class of infinite-order cutoff functions with arbitrarily high rates of decay at infinity. We also present an eighth order cutoff function with compact support. We test two versions of rezoning. Version 1 has been suggested and tested by Beale and Majda, while version 2 is new. Using rezoning, we test the eighth-order cutoff function and one infinite-order cutoff function on three test problems for which the solution of Euler's equation is known analytically. The accuracy of the infinite-order cutoff function is greater than that of the eighth-order cutoff function when the flow is very smooth. We also compute the evolution of two circular vorticity patches and the evolution of one square vorticity patch over long time intervals. Finally, we make a comparison between the direct method of velocity evaluation and the Rokhlin–Greengard algorithm. The numerical experiments indicate that for smooth flows, high-order cutoffs combined with rezoning give high accuracy for long time integrations. © 1991 Academic Press, Inc.

INTRODUCTION

The vortex method for Euler's equation is a numerical method for approximating the flow of an incompressible fluid without viscosity. The idea is to approximate a vorticity distribution by a finite set of "vortex blobs" which are multiples and translates of a certain function known as the cutoff function. The cutoff function is scaled by a parameter δ and approximates the delta function as δ tends to 0. The vortex blobs induce a velocity field, which in turn moves the vortex blobs. The evolution of the vortex blobs is computed by solving a system of ordinary differential equations by standard numerical methods. In this form, the vortex method was introduced by Chorin [9] in 1973. There have been many applications of vortex methods, including the simulation of turbulent combustion in open and closed vessels, Sethian [23], the computation of unstable boundary layers, Chorin [10], aerodynamic computations, Cheer [8], Spalart [25], Leonard and Spalart [18], and flow of variable density, Anderson [2].

The vortex method can be extended to simulate viscous flow by letting each vortex take a random step after each timestep, see Chorin [9]. Sethian and

0021-9991/91 $3.00

Ghoniem [24] made an elaborate test of this procedure on viscous flow through a channel over a backwards-facing step.

A list of papers containing convergence proofs for vortex methods includes del Prete and Hald [16], Hald [14, 15], Beale and Majda [5, 6], Cottet [13], and Anderson and Greengard [3]. Only [3, 15] include convergence for the time discretization.

The accuracy of the vortex method depends on how the delta function is approximated, which in turn depends both on the choice of cutoff function and on the choice of the paramater δ. Beale and Majda [7] introduced a family of smooth cutoff functions, with unbounded support, but decaying very rapidly at infinity and Hald [15] presented several infinite-order cutoff functions. The rates of convergence for the infinite-order cutoffs are only limited by the smoothness of the flow. In this paper, we test the practical accuracy of one of these cutoff functions for flows of different degrees of smoothness. Our numerical results show that we *do* obtain orders of accuracy slightly exceeding the ones predicted by Hald's theory, but only for short time integrations. The deterioration in accuracy at later times was already observed by Perlman [21] and is even more pronounced for infinite-order cutoff functions. A natural way to overcome this difficulty is to use the rezoning technique. It was suggested and tested by Beale and Majda [7]. In this paper, we present two versions of rezoning. The first version is that of Beale and Majda [7], but with the added feature of a "built-in" criterion for determining at which times we introduce a new grid. The second version also has this feature. It is more accurate because it uses more vortices, but it costs more. A different method of improving the accuracy for large time integrations was introduced by Beale [4].

It follows from Hald's theory [15], that we should take δ proportional to $\sqrt{h}$ when using infinite order methods. However, it is not clear what the optimal proportionality constant is. That depends on a number of factors. First of all, in vortex methods without rezoning we always need to use a larger proportionality constant, for large integration times. Second the choice of proportionality constant depends strongly on the cutoff function and especially on the value of the cutoff function at the origin. For example, for the eighth-order cutoff function presented in this paper, we have to take a proportionality constant that is about 5.5 times larger than for Hald's infinite-order cutoff function. This is due to the fact that the value of the first cutoff at the origin is about 30 times larger than for the second cutoff. Finally, the choice depends on the flow itself, at least if it is not radially symmetric. A partial list of numerical experiments that test the accuracy of vortex methods includes del Prete and Hald [16], Beale and Majda [7], Beale [4], Perlman [21], and Nakamura, Leonard, and Spalart [19].

Besides the accuracy of vortex methods, the computational speed is also important. The direct method of computation requires $O(N^2)$ flops, where N is the number of vortices. A fast algorithm known as the method of multipole expansions, is given by Rokhlin and Greengard [22]. This method requires $O(N)$ flops when δ is proportional to h, and when it is applicable, it is essentially as accurate as the direct method. Another fast method is given by Anderson [1]. Both of these methods are only strictly applicable when using cutoff functions with compact support. For this reason, we present an eight-order cutoff function with compact support. We test the Rokhlin–Greengard algorithm [22] using this cutoff function.

This paper is divided into four sections. In Section 1 we describe the vortex method in two dimensions. In Section 2 we present the eighth-order cutoff function mentioned above and compare it with Hald's infinite-order cutoff. In Section 3, two versions of the method of rezoning are described, and finally, in Section 4 we present our test problems and numerical results.

1. The Basic Equations

The vorticity-stream function form of Euler's equations in two dimensions is

$$\omega_t + u \cdot \nabla\omega = 0, \tag{1.1}$$

$$\Delta\psi = -\omega, \tag{1.2}$$

$$u = \psi_y, \qquad v = -\psi_x, \tag{1.3}$$

where $\mathbf{u} = (u, v)$ is the velocity vector, $\mathbf{x} = (x, y)$ is the position vector, ω is the vorticity, and ψ is the stream function.

Solving the Poisson equation (1.2) for ψ, see [17], and differentiating as in (1.3), we obtain the velocity as

$$\mathbf{u}(\mathbf{x}, t) = \int_{\Omega(t)} K(\mathbf{x} - \mathbf{x}')\, \omega(\mathbf{x}', t)\, d\mathbf{x}', \tag{1.4}$$

where $K(\mathbf{x}) = (-y, x)^{\mathrm{T}}/2\pi\, |\mathbf{x}|^2$, $|\mathbf{x}|^2 = x^2 + y^2$, $d\mathbf{x}' = dx'\, dy'$, and $\Omega(t)$ denotes the support of ω in R^2 at time t. Since the flow is incompressible and the vorticity is preserved along particle paths, see Chorin and Marsden [11, p. 34], we can rewrite (1.4) as

$$\mathbf{u}(\mathbf{x}(\alpha, t), t) = \int_{\Omega(0)} K(\mathbf{x}(\alpha, t) - \mathbf{x}(\beta, t))\, \omega(\beta, 0)\, d\beta. \tag{1.5}$$

Here $\mathbf{x}(\alpha, t)$ is the position at time t of the particle which starts at α at time $t = 0$.

To calculate the velocity, we replace the kernel K by a kernel K_δ which is bounded at $\mathbf{x} = 0$. K_δ is the convolution of K with a smooth cutoff function Ψ_δ; i.e.,

$$K_\delta(\mathbf{x}) = \int K(\mathbf{x} - \mathbf{x}')\, \Psi_\delta(\mathbf{x}')\, d\mathbf{x}'.$$

Here Ψ_δ is defined by $\Psi_\delta(\mathbf{x}) = \delta^{-2}\Psi(\mathbf{x}/\delta)$, where Ψ is a smooth radially symmetric function satisfying

$$\int_{R^2} \Psi(\mathbf{x})\, d\mathbf{x} = 1.$$

Hence, Ψ_δ approximates the Dirac delta function as $\delta \to 0$. To complete the discretization of (1.5) we replace the variables α and β by the double indices $\mathbf{i} = (i_1, i_2)$ and $\mathbf{j}$, and replace the integral by a sum. This gives us the following system of ordinary differential equations

$$\frac{d\tilde{\mathbf{x}}_{\mathbf{i}}(t)}{dt} = \tilde{\mathbf{u}}_{\mathbf{i}}(t), \qquad \tilde{\mathbf{x}}_{\mathbf{i}}(0) = \mathbf{i}h, \tag{1.6}$$

where

$$\tilde{\mathbf{u}}_{\mathbf{i}}(t) = \sum_{\mathbf{j} \in \mathbf{J}, \mathbf{j} \neq \mathbf{i}} K_\delta(\tilde{\mathbf{x}}_{\mathbf{i}}(t) - \tilde{\mathbf{x}}_{\mathbf{j}}(t))\, c_{\mathbf{j}}. \tag{1.7}$$

Here h is the initial distance between adjacent grid-points, $\mathbf{J}$ is the set of all indices $\mathbf{j}$ such that $\mathbf{j}h \in \Omega(0)$, and the "vorticity coefficients" $c_{\mathbf{j}}$ are defined as

$$c_{\mathbf{j}} = \omega(\mathbf{j}h)h^2.$$

The numerical solution of this system is known as the vortex method. By imposing additional conditions on the cutoff function Ψ one can obtain high rates of convergence for this method.

2. Choice of Cutoff Functions

In this paper we shall focus our attention on two high-order cutoff functions Ψ. The first one is the following infinite-order cutoff function due to Hald [15],

$$\Psi(r) = \frac{4}{45\pi r^3}(16J_3(4r) - 10J_3(2r) + J_3(r)), \tag{2.1}$$

where J_3 is the Bessel function of order 3. The corresponding velocity kernel is

$$K_\delta(x, y) = \frac{(-y, x)^{\mathrm{T}}}{2\pi r^2}\left(1 - \frac{8}{45(r/\delta)^2}(4J_2(4r/\delta) - 5J_2(2r/\delta) + J_2(r/\delta))\right). \tag{2.2}$$

We used this cutoff function on all the test problems in this paper, since we wanted to check the performance of infinite-order cutoff functions in practice. The cutoff function (2.1) belongs to a larger class of infinite-order cutoff functions of the form

$$\Psi(r) = n!\left(\frac{-2}{r}\right)^{n+1} \sum_{i=0}^{n} C_i k_i^{n+1} J_{n+1}(k_i r). \tag{2.3}$$

Here $k_0 = 1$, $k_0 < k_1 < \cdots < k_n$, and $C_i = (-4\pi \prod_{j=0, j\neq i}^{n} (k_j^2 - k_i^2))^{-1}$ for $i = 0, ..., n$. The velocity kernel corresponding to (2.3) is

$$K_\delta(x, y) = \frac{(-y, x)^{\mathrm{T}}}{2\pi r^2}\left(1 + \left(\frac{r}{\delta}\right)^{-n}\left(\sum_{j=0}^{n} \gamma_j J_n(k_j r/\delta)\right)\right), \tag{2.4}$$

where $\gamma_i = 4\pi n!(-2k_i)^n C_i$. The derivation of this class of infinite-order cutoff functions can be found in [20]. Hald's cutoff function (2.1) is obtained from (2.3) by setting $n=2$, $k_1=2$, and $k_2=4$. Error estimates for this class of cutoff functions are given in Hald [15].

The second cutoff function used in this paper, derived in [20], is of eighth order and has compact support:

$$\Psi(r) = \begin{cases} -52(1-r^2)^9(140r^6 - 105r^4 + 21r^2 - 1)/\pi & \text{for} \quad 0 \leqslant r \leqslant 1 \\ 0 & \text{for} \quad r \geqslant 1 \end{cases}$$

$$K_\delta(x, y) = \begin{cases} \dfrac{(-y, x)^{\mathrm{T}}}{2\pi r^2}\left(1 + \left(1 - \dfrac{r^2}{\delta^2}\right)^{10}\left(286 - 1092\left(1 - \dfrac{r^2}{\delta^2}\right)\right.\right. & \\ \qquad \left.\left. + 1365\left(1 - \dfrac{r^2}{\delta^2}\right)^2 - 560\left(1 - \dfrac{r^2}{\delta^2}\right)^3\right)\right) & \text{for} \quad r \leqslant \delta \\ \dfrac{(-y, x)^{\mathrm{T}}}{2\pi r^2} & \text{for} \quad r > \delta. \end{cases}$$

The reason for introducing this cutoff function is that cutoff functions with compact support are the only ones that are mathematically compatible with "fast" vortex methods such as the Rokhlin–Greengard algorithm [22] or Anderson's method of local corrections [1]. Note that the higher-order cutoff functions of Beale and Majda [7] and Hald's infinite-order cutoff function (2.1) do not have compact support. We observe that the speedup for a fast algorithm is limited by the size of the cutoff parameter δ. For high-order cutoff functions, δ must be proportional to $\sqrt{h}$ in order to maintain high accuracy for long time integrations. Thus, the amount of computational labor due to "local" interactions is $O(N^{1.5})$.

The eighth-order cutoff function was tried out on test problems 1–3, and in Section 4 the results are compared to those obtained using Hald's infinite-order cutoff function. We found that the optimal value of δ for this cutoff function is about $1.7\sqrt{h}$ for test problems 1–3. The Fourier transform of Ψ is given by

$$\hat{\Psi}(t) = -6656 \cdot 11! \cdot \pi^{-2} \\ \times (t^{-10}J_{10}(t) - 84t^{-11}J_{11}(t) + 2520t^{-12}J_{12}(t) - 26880t^{-13}J_{13}(t)).$$

Therefore $|D_{\mathbf{k}}^{\alpha}\hat{\Psi}(\mathbf{k})|$ decays like const $\cdot\, (1 + |\mathbf{k}|)^{-10.5 - |\alpha|}$ for any double index α.

According to the convergence theorem of Beale and Majda [6], we then get the following error estimates for particle positions and velocities, provided that we take $\delta = Ch^q$ with $q < 9.5/18.5$, and the flow is "sufficiently" smooth:

$$\max_{0 \leqslant t \leqslant T} \|\mathbf{x}_{\mathbf{j}}(t) - \tilde{\mathbf{x}}_{\mathbf{j}}(t)\| \leqslant Ch^{8q},$$

$$\max_{0 \leqslant t \leqslant T} \|\mathbf{u}_{\mathbf{j}}(t) - \tilde{\mathbf{u}}_{\mathbf{j}}(t)\| \leqslant Ch^{8q}.$$

Here $\mathbf{x}_\mathbf{j}(t)$ and $\mathbf{u}_\mathbf{j}(t)$ are the exact position and velocity at time t of a particle which starts at $\mathbf{j}$ at time $t=0$. The norm is the discrte L^p norm with $1<p<\infty$. In particular, we may choose $q=0.5$, which would give us fourth-order convergence if the flow is smooth enough.

We would now like to compare this cutoff function Ψ with Hald's infinite order cutoff function $\tilde{\Psi}$. First we note that $\Psi(0)=52/\pi$ while $\tilde{\Psi}(0)=1.75/\pi$. Therefore, rather than comparing Ψ and $\tilde{\Psi}$, we compare Ψ_α and $\tilde{\Psi}$, where $\alpha=\sqrt{52/1.75}$ and $\Psi_\alpha(r)=\alpha^{-2}\Psi(r/\alpha)$. Then, $\Psi_\alpha(0)=\tilde{\Psi}(0)$, and by plotting $\Psi_\alpha(r)$ and $\tilde{\Psi}(r)$ on the same graph, we see that $\Psi_\alpha(r)\approx\tilde{\Psi}(r)$ for *any* r (see Fig. 2.1). It is also interesting to compare the Fourier transforms of these two cutoff functions (see Fig. 2.2). Once again, we get close agreement. We conclude that if we use $\delta=Ch^q$ with the eighth-order cutoff function Ψ, and $\delta'=C'h^q$ with Hald's infinite-order cutoff, we should have $C/C'=\sqrt{(52/1.75)}\approx 5.45$. Indeed, in test problems 1–3 we found by experiments that $\delta'=0.3\sqrt{h}$ was the best choice for Hald's infinite-order cutoff function while the best value of δ for the eighth-order cutoff function was about $1.7\sqrt{h}$. This gives the ratio $1.7/0.3\approx 5.67$! This analysis suggests that if we have found the relationship between δ and h experimentally for a particular cutoff func-

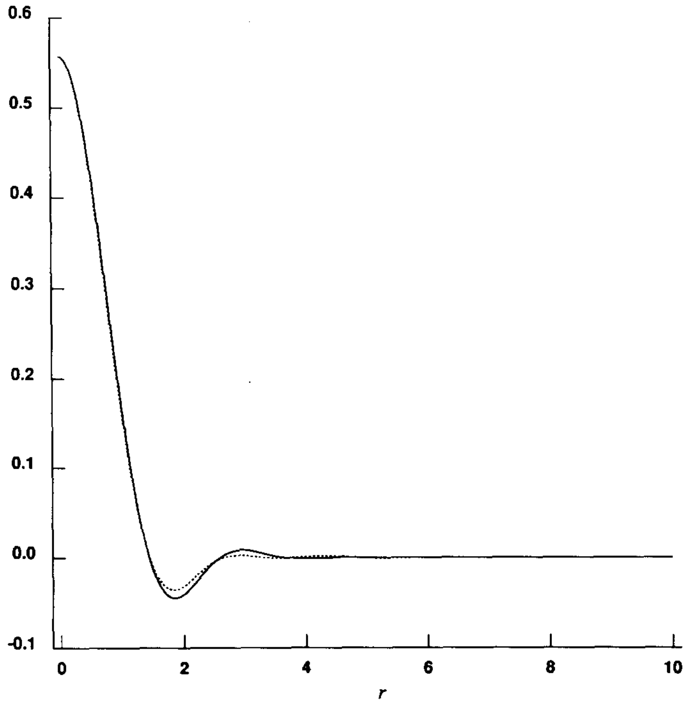

FIG. 2.1. Infinite order cutoff vs scaled 8th order cutoff. Solid curve = $\Psi_\alpha(r)$, dotted curve = $\tilde{\Psi}(r)$, $\alpha=\sqrt{(52/1.75)}$.

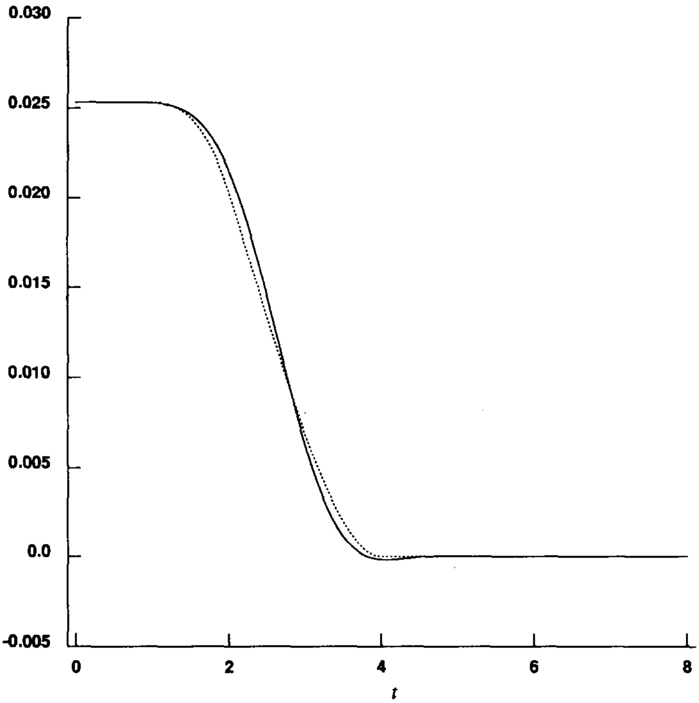

FIG. 2.2. Fourier transforms of Hald's infinite order cutoff and the scaled 8th order cutoff. Solid curve = $\hat{\Psi}(\alpha t)$, dotted curve = $\hat{\hat{\Psi}}(t)$, $\alpha = \sqrt{(52/1.75)}$.

tion Ψ_1, then we can determine the best value of δ as a function of h for any other cutoff function Ψ. Take

$$\delta_{\text{optimal}} = (\delta_1)_{\text{optimal}} \left(\frac{\Psi(0)}{\Psi_1(0)} \right)^{1/2}.$$

3. REZONING

Numerical experiments with infinite-order cutoff functions showed that for smooth flows these cutoffs give extremely accurate values of velocity and vorticity for short integration times. Unfortunately, this high accuracy is lost as time increases, so that for long integration times, these cutoffs are not significantly more accurate than lower-order cutoffs. We shall show that one way to overcome this problem is the rezoning strategy suggested by Beale and Majda [7]. We will present a version of rezoning similar to theirs, which we call Version 1, and a new method, which we call Version 2.

First we note that since Ψ_δ approximates the delta function as $\delta \to 0$, we have at time $t=0$ that

$$\omega(\mathbf{z}, 0) \approx \sum_{\mathbf{j} \in \mathbf{J}} \Psi_\delta(\mathbf{z} - \mathbf{x}_\mathbf{j}(0)) c_\mathbf{j}.$$

This holds for all $\mathbf{z}$. Here $\mathbf{J}$ is the set of all double indices $\mathbf{j} = (j_1, j_2)$ such that $\mathbf{j}h \in \Omega(0)$, the support of the initial vorticity distribution. Since vorticity is preserved along particle paths, we also expect that at later times t,

$$\omega(\mathbf{z}, t) \approx \sum_{\mathbf{j} \in \mathbf{J}} \Psi_\delta(\mathbf{z} - \mathbf{x}_\mathbf{j}(t)) c_\mathbf{j}. \tag{3.1}$$

In particular, letting $\mathbf{z} = \mathbf{x}_\mathbf{i}(t)$ in (3.1) gives

$$\omega(\mathbf{x}_\mathbf{i}(t), t) \approx \sum_{\mathbf{j} \in \mathbf{J}} \Psi_\delta(\mathbf{x}_\mathbf{i}(t) - \mathbf{x}_\mathbf{j}(t)) c_\mathbf{j}. \tag{3.2}$$

Multiplying both sides of (3.2) by h^2 and recalling that $c_\mathbf{i} = h^2 \omega(\mathbf{x}_\mathbf{i}(t), t)$ for any t gives

$$c_\mathbf{i} \approx h^2 \sum_{\mathbf{j} \in \mathbf{J}} \Psi_\delta(\mathbf{x}_\mathbf{i}(t) - \mathbf{x}_\mathbf{j}(t)) c_\mathbf{j}. \tag{3.3}$$

Therefore we define

$$c_\mathbf{i}^h(t) = h^2 \sum_{\mathbf{j} \in \mathbf{J}} \Psi_\delta(\mathbf{x}_\mathbf{i}(t) - \mathbf{x}_\mathbf{j}(t)) c_\mathbf{j} \tag{3.4}$$

and

$$E_\omega(t) = \left(h^2 \sum_{\mathbf{j} \in \mathbf{J}} (c_\mathbf{j}^h(t) - c_\mathbf{j})^2 \right)^{1/2}. \tag{3.5}$$

Here $E_\omega(t)$ stands for "the average vorticity error along vortex paths." With these preliminaries out of the way, we can now present the first version of rezoning.

Version 1

Step 1. First compute $E_\omega(0)$. Then, as in the standard vortex method, solve the system of ordinary differential equations,

$$\frac{d\tilde{\mathbf{x}}_\mathbf{i}(t)}{dt} = \tilde{\mathbf{u}}_\mathbf{i}(t), \qquad \tilde{\mathbf{x}}_\mathbf{i}(0) = \mathbf{i}h, \tag{3.6}$$

where

$$\tilde{\mathbf{u}}_\mathbf{i}(t) = \sum_{\mathbf{j} \in \mathbf{J}} K_\delta(\tilde{\mathbf{x}}_\mathbf{i}(t) - \tilde{\mathbf{x}}_\mathbf{j}(t)) c_\mathbf{j}. \tag{3.7}$$

After each time step Δt, calculate $E_\omega(t)$ and $E_\omega(t)/E_\omega(0)$. Continue to solve (3.6), (3.7) until

$$\frac{E_\omega(t)}{E_\omega(0)} > \eta,$$

where η is a parameter we have to specify. In our numerical experiments we have used $\eta = 1.1$, $\eta = 1.25$, or $\eta = 1.5$. When $E_\omega(t)/E_\omega(0) > \eta$ we no longer solve (3.6), (3.7) but go on to the next step.

Step 2. Suppose $t = T_1$ when we leave Step 1. Now we set $\tilde{\mathbf{z}}_\mathbf{j}(T_1) = \tilde{\mathbf{x}}_\mathbf{j}(T_1)$ for every $\mathbf{j} \in \mathbf{J}$. Then we introduce a new square grid, occupying a region $A \subset R^2$, which is somewhat larger than what is needed to cover all point vortices at time $t = T_1$. Let $\mathbf{J}_A$ denote the set of double indices $\mathbf{j}$ such that $\mathbf{j}h \in A$. For every $\mathbf{j} \in \mathbf{J}_A$ introduce a new vortex at every grid-point $\mathbf{j}h$. We use the old vortices one last time to compute a new "initial" vorticity distribution. To be more precise, we let

$$(c_\mathbf{i})_{\text{new}} \equiv h^2 \sum_{\mathbf{j} \in \mathbf{J}} \Psi_\delta(\mathbf{i}h - \tilde{\mathbf{x}}_\mathbf{j}(T_1))(c_\mathbf{j})_{\text{old}}, \qquad \text{for every} \quad \mathbf{i} \in \mathbf{J}_A. \tag{3.8}$$

Now we "throw away" all the old vortices $\tilde{\mathbf{x}}_\mathbf{j}$ and denote the *new* vortices as $\tilde{\mathbf{x}}_\mathbf{j}$ with $\tilde{\mathbf{x}}_\mathbf{j}(T_1) = \mathbf{j}h$. We then delete all the new vortices $\tilde{\mathbf{x}}_\mathbf{i}$ for which $|(c_\mathbf{i})_{\text{new}}| < \varepsilon$, where ε is a certain tolerance. Let $\mathbf{J}_1$ be the subset of $\mathbf{J}_A$ such that $|(c_\mathbf{i})_{\text{new}}| \geqslant \varepsilon$ for every $\mathbf{i} \in \mathbf{J}_1$. Now for $t \geqslant T_1$ we solve the following larger system of ordinary differential equations:

$$\tilde{\mathbf{u}}(\mathbf{x}, t) = \sum_{\mathbf{j} \in \mathbf{J}_1} K_\delta(\mathbf{x} - \tilde{\mathbf{x}}_\mathbf{j}(t))(c_\mathbf{j})_{\text{new}}, \tag{3.9}$$

$$\frac{d\tilde{\mathbf{x}}_\mathbf{i}(t)}{dt} = \tilde{\mathbf{u}}(\tilde{\mathbf{x}}_\mathbf{i}, t), \qquad \tilde{\mathbf{x}}_\mathbf{i}(T_1) = \mathbf{i}h, \qquad \text{for every} \quad \mathbf{i} \in \mathbf{J}_1,$$

$$\frac{d\tilde{\mathbf{z}}_\mathbf{i}(t)}{dt} = \tilde{\mathbf{u}}(\tilde{\mathbf{z}}_\mathbf{i}, t) \tag{3.10}$$

Remark. Here the $\tilde{\mathbf{z}}_\mathbf{i}$'s denote the original vortices. If we are not interested in the paths of the original vortices, but the paths of some other particles, we should let the $\tilde{\mathbf{z}}_\mathbf{i}$'s denote these particles. In that case, we do not set $\tilde{\mathbf{z}}_\mathbf{j}(T_1) = \tilde{\mathbf{x}}_\mathbf{j}(T_1)$ at the beginning of Step 2.

Now again we compute $E_\omega(t)$ and $E_\omega(t)/E_\omega(T_1)$ after every time step Δt, but now using the new vorticity coefficients in (3.4) and (3.5). Continue solving (3.9), (3.10) until $E_\omega(t)/E_\omega(T_1) > \eta$.

Step 3. Suppose $t = T_2$ when we leave Step 2. Now repeat Step 2 but replacing T_1 by T_2 in all the equations. Also in this step, *do not* set $\tilde{\mathbf{z}}_\mathbf{j}(T_2) = \tilde{\mathbf{x}}_\mathbf{j}(T_2)$, for $\mathbf{j} \in \mathbf{J}_1$. Continue this process until we reach $t = T_{\max}$.

Numerical experiments using this technique have indicated a great reduction in velocity errors for long integration times compared to the corresponding errors without rezoning, but with the same grid-spacing h. In vortex methods without rezoning, we are forced to pick a considerably larger value of δ for long integration times. With rezoning, however, we can usually use values of δ close to the optimal value of δ at time $t=0$. Nevertheless, we have also noted that when we use this form of rezoning, the velocity error takes a jump every time a new vorticity distribution is computed using (3.8). These jumps are small compared to the sharp increase in error experienced when no rezoning is applied, but still significant compared to the marginal increase in velocity error at intermediate times. To reduce this effect and the effect of "numerical viscosity," we propose the following scheme.

Version 2

In Version 2 we introduce a finer grid than in Version 1, for the purpose of calculating the new vorticity distributions on the new grids. The velocity evaluations are however done on the coarser grid. This method has some similarities with multi-grid methods, but it does not fall into the framework of such methods. The details of Version 2 are as follows:

Step 1. Let Q_0 be the set of double indices $\mathbf{q}=(q_1, q_2)$ such that $\mathbf{q}h \in \Omega(0)$, where q_1 and q_2 are *integers or half-integers*. For every $\mathbf{q} \in Q_0$ introduce a vortex $\mathbf{x}_\mathbf{q}$ with strength $c_\mathbf{q} = \omega(\mathbf{q}h)h^2$. Let $\mathbf{J}$ be the integer-pair subset of Q_0. Then solve the following system of ordinary differential equations, for every $\mathbf{q} \in Q_0$,

$$\frac{d\tilde{\mathbf{x}}_\mathbf{q}(t)}{dt} = \tilde{\mathbf{u}}_\mathbf{q}(t), \qquad \tilde{\mathbf{x}}_\mathbf{q}(0) = \mathbf{q}h, \tag{3.11}$$

where

$$\tilde{\mathbf{u}}_\mathbf{q}(t) = \sum_{\mathbf{j} \in \mathbf{J}} K_\delta(\tilde{\mathbf{x}}_\mathbf{q}(t) - \tilde{\mathbf{x}}_\mathbf{j}(t))c_\mathbf{j}. \tag{3.12}$$

After each time-step Δt, calculate $E_\omega(t)$ and $E_\omega(t)/E_\omega(0)$ using (3.4), (3.5). Note that in calculating $E_\omega(t)$ and $E_\omega(t)/E_\omega(0)$ we use *only* the vortices and vorticity coefficients with integer indices. Continue to solve (3.11), (3.12) until

$$\frac{E_\omega(t)}{E_\omega(0)} > \eta.$$

Step 2. Suppose $t = T_1$ when we leave Step 1. Set $\tilde{\mathbf{z}}_\mathbf{j}(T_1) = \tilde{\mathbf{x}}_\mathbf{j}(T_1)$ for every $\mathbf{j} \in \mathbf{J}$. As in Version 1, we introduce a new square grid, occupying a region $A \subset R^2$. Let Q_A be the set of double indices $\mathbf{q} = (q_1, q_2)$ such that $\mathbf{q}h \in A$, with q_1 and q_2 assuming both integer and half-integer values. Let $\mathbf{J}_A$ be the integer-pair subset of

Q_A. For every $\mathbf{q} \in Q_A$ introduce a new vortex at every grid-point $\mathbf{q}h$. Define the new vorticity distribution by

$$(c_{\mathbf{q}})_{\text{new}} \equiv h^2/4 \sum_{\mathbf{r} \in Q_0} \Psi_{\delta'}(\mathbf{q}h - \tilde{\mathbf{x}}_{\mathbf{r}}(T_1))(c_{\mathbf{r}})_{\text{old}}, \qquad \text{for every} \quad \mathbf{q} \in Q_A. \tag{3.13}$$

Note that the effective grid-spacing in (3.13) is $h/2$ rather than h. That is why we have a factor of $h^2/4$ in front of the summation sign instead of h^2. We must also replace Ψ_δ by $\Psi_{\delta'}$, where δ' is the cutoff parameter corresponding to a grid-spacing of $h/2$. If for any value of h we pick $\delta = \text{const} \cdot h^q$, then $\delta' = 2^{-q}\delta$. The purpose of using (3.13) instead of (3.8) is to make the error in the computed new vorticity distribution small compared to the error in the velocity evaluations, thereby reducing "numerical viscosity." As in version 1, we now "throw away" the old vortices $\tilde{\mathbf{x}}_{\mathbf{q}}$ and use this notation for the new vortices such that $\tilde{\mathbf{x}}_{\mathbf{q}}(T_1) = \mathbf{q}h$. Then let Q_1 be the subset of Q_A such that $|(c_{\mathbf{q}})_{\text{new}}| \geqslant \varepsilon$ for every $\mathbf{q} \in Q_1$, where ε is a certain tolerance. Let $\mathbf{J}_1$ be the integer-pair subset of Q_1. For $t \geqslant T_1$ solve the system of ordinary differential equations,

$$\tilde{\mathbf{u}}(\mathbf{x}, t) = \sum_{\mathbf{j} \in \mathbf{J}_1} K_\delta(\mathbf{x} - \tilde{\mathbf{x}}_{\mathbf{j}}(t))(c_{\mathbf{j}})_{\text{new}}, \tag{3.14}$$

$$\frac{d\tilde{\mathbf{x}}_{\mathbf{q}}(t)}{dt} = \tilde{\mathbf{u}}(\tilde{\mathbf{x}}_{\mathbf{q}}, t), \qquad \tilde{\mathbf{x}}_{\mathbf{q}}(T_1) = \mathbf{q}h, \qquad \text{for every} \quad \mathbf{q} \in Q_1,$$

$$\frac{d\tilde{\mathbf{z}}_{\mathbf{i}}(t)}{dt} = \tilde{\mathbf{u}}(\tilde{\mathbf{z}}_{\mathbf{i}}, t). \tag{3.15}$$

After every timestep Δt compute $E_\omega(t)$ and $E_\omega(t)/E_\omega(T_1)$, using the new vorticity coefficients with integer indices and continue solving (3.14), (3.15) until $E_\omega(t)/E_\omega(T_1) > \eta$.

Step 3. Suppose $t = T_2$ when we leave Step 2. Repeat Step 2 replacing T_1 by T_2 where applicable. As in Version 1, do not set $\tilde{\mathbf{z}}_{\mathbf{j}}(T_2) = \tilde{\mathbf{x}}_{\mathbf{j}}(T_2)$, for $\mathbf{j} \in \mathbf{J}_1$ at this point. Continue in the same manner until $t = T_{\max}$.

4. Numerical Results

We present five test problems. In the first three test problems, the solution is known analytically since the vorticity distribution is radially symmetric: The flow is circular, and the velocity at any point and at any time is given by

$$\mathbf{u}(x, y) = (u, v) = \mu(r)(-y, x)^{\mathrm{T}}, \tag{4.1}$$

where $\mu(r) = (1/r^2) \int_0^r s\omega(s)\, ds$ is the angular velocity of the flow.

The other two test problems do not have a known analytical solution. Here we show the numerical solutions graphically, and the rate of convergence is estimated by using Richardsons's extrapolation. In the first three test problems, two different cutoff functions are used, namely Hald's infinite-order cutoff and our eighth-order cutoff. Unless specified otherwise, we have used Version 2 rezoning. In all cases we used the classical fourth-order Runge–Kutta method for time integration.

We now look at the first three test problems in detail. In each of these, the vorticity distribution has the form

$$\omega(r)=\begin{cases}(1-r^2)^k & \text{for} \quad r\leqslant 1\\ 0 & \text{for} \quad r>1,\end{cases} \tag{4.2}$$

where $k=3$ in test problem 1, $k=7$ in test problem 2, and $k=14$ in test problem 3. The case $k=3$ has been tested numerically by Beale and Majda [7] and Beale [4], and the case $k=7$ by Perlman [21].

It can be shown that the Fourier transform of a vorticity distribution of this form is $C_k J_{k+1}(t)/t^{k+1}$, where C_k is a constant. Thus, $\hat{\omega}(t)$ is of order $O(t^{-(k+1.5)})$ as $t\to\infty$, although $\omega(r)$ has only k bounded derivatives. In general, a vorticity distribution ω with compact support and k bounded derivatives only guarantees that $\hat{\omega}(t)$ is of order $O(t^{-k})$ as $t\to\infty$. This means that for test problems 1–3, Hald's estimate [15, p. 568] of the moment error for infinite-order cutoffs can be improved to $O(\delta^{k+0.5})$ rather than $O(\delta^{k-1})$.

The solutions of the Euler equation for these vorticity distributions are given by (4.1) with

$$\mu(r)=\begin{cases}\dfrac{1-(1-r^2)^{k+1}}{2(k+1)r^2} & \text{for} \quad r\leqslant 1\\[2ex] \dfrac{1}{2(k+1)r^2} & \text{for} \quad r>1.\end{cases} \tag{4.3}$$

We measure the velocity error in the discrete L^2 norm,

$$E_u=\left(h^2\sum_{\mathbf{j}\in\mathbf{J}}|\mathbf{u}_\mathbf{j}(t)-\tilde{\mathbf{u}}_\mathbf{j}(t)|^2\right)^{1/2}. \tag{4.4}$$

The rate of convergence is estimated by using two successive values of h,

$$\text{rate of convergence}=\frac{\ln(E_u(h_1)/E_u(h_2))}{\ln(h_1/h_2)}. \tag{4.5}$$

Tables Ia and Ib give the velocity errors in test problem 1 at different times up to time $t=50$, for different values of h, and for the two different cutoff functions. We choose $\delta=\text{const}\cdot\sqrt{h}$, so that the moment error and the discretization error will be of approximately the same order in h. The proportionality constant has been chosen so as to minimize the velocity error at time $t=0$ when $h=0.100$. Comparing these two tables, we see that the errors are between 5 and 29% larger when the

TABLE Ia

Velocity Errors for Test Problem 1

		E_u		
t	$h = 0.125$	$h = 0.100$	$h = 0.0625$	$h = 0.05$
0.0	0.1665×10^{-4}	0.9000×10^{-5}	0.3142×10^{-5}	0.1886×10^{-5}
10.0	0.5340×10^{-4}	0.2562×10^{-4}	0.6775×10^{-5}	0.3829×10^{-5}
20.0	0.1201×10^{-3}	0.5473×10^{-4}	0.1377×10^{-4}	0.7372×10^{-5}
30.0	0.2011×10^{-3}	0.9110×10^{-4}	0.2268×10^{-4}	0.1200×10^{-4}
40.0	0.2982×10^{-3}	0.1358×10^{-3}	0.3321×10^{-4}	0.1727×10^{-4}
50.0	0.4141×10^{-3}	0.1880×10^{-3}	0.4559×10^{-4}	0.2327×10^{-4}

Note. $\omega(r) = (\max(0, 1 - r^2))^3$. Cutoff function, *Hald's infinite-order*, $\eta = 1.1$, $\delta = 0.3\sqrt{h}$, $\Delta t = 4.0\,h$, $T_{\max} = 50.0$.

eighth-order cutoff is used. The rates of convergence for the two cutoffs are also approximately the same for corresponding times t, as seen in Tables IVa and IVb. Since we take δ proportional to $\sqrt{h}$, we can expect the moment error to be of order $O(h^{1.75})$ and the discretization of order $O(h^2)$, with the infinite-order cutoff function. We could therefore only expect a rate of convergence of 1.75. However, the observed rate of convergence is greater than 2. In particular, at time $t = 0$, the computed rate of convergence is 2.3, both using infinite-order and eighth-order cutoffs. Since Perlman [21] has shown numerically that at time $t = 0$ the moment error is much larger than the discretization error, this seems to indicate that the moment error is actually of order $O(\delta^{4.6})$, rather than the best theoretical estimate of $O(\delta^{3.5})$. Beale and Majda [7] observed a rate of convergence of 3.6 at time $t = 0$ for this problem, using an eighth-order Gaussian cutoff function, but with δ proportional to $h^{0.75}$, which would correspond to a moment error of order $O(\delta^{4.8})$

TABLE Ib

Velocity Errors for Test Problem 1

		E_u		
t	$h = 0.125$	$h = 0.100$	$h = 0.0625$	$h = 0.05$
0.0	0.1751×10^{-4}	0.9473×10^{-5}	0.3329×10^{-5}	0.1998×10^{-5}
10.0	0.6122×10^{-4}	0.2945×10^{-4}	0.7661×10^{-5}	0.4233×10^{-5}
20.0	0.1519×10^{-3}	0.6172×10^{-4}	0.1497×10^{-4}	0.8168×10^{-5}
30.0	0.2757×10^{-3}	0.1062×10^{-3}	0.2429×10^{-4}	0.1339×10^{-4}
40.0	0.4207×10^{-3}	0.1594×10^{-3}	0.3609×10^{-4}	0.1974×10^{-4}
50.0	0.5799×10^{-3}	0.2225×10^{-3}	0.5000×10^{-4}	0.2681×10^{-4}

Note. $\omega(r) = (\max(0, 1 - r^2))^3$. Cutoff function, *8th order with compact support*, $\eta = 1.1$, $\delta = 1.7\sqrt{h}$, $\Delta t = 4.0\,h$, $T_{\max} = 50.0$.

TABLE IIa

Velocity Errors for Test Problem 2

	E_u			
t	$h=0.125$	$h=0.100$	$h=0.0625$	$h=0.05$
0.0	0.3242×10^{-5}	0.1293×10^{-5}	0.1587×10^{-6}	0.6384×10^{-7}
10.0	0.1634×10^{-4}	0.5244×10^{-5}	0.8657×10^{-6}	0.3588×10^{-6}
20.0	0.3291×10^{-4}	0.1108×10^{-4}	0.1791×10^{-5}	0.7356×10^{-6}
30.0	0.5286×10^{-4}	0.1815×10^{-4}	0.2810×10^{-5}	0.1143×10^{-5}
40.0	0.7567×10^{-4}	0.2639×10^{-4}	0.3910×10^{-5}	0.1580×10^{-5}
50.0	0.1027×10^{-3}	0.3600×10^{-4}	0.5098×10^{-5}	0.2046×10^{-5}

Note. $\omega(r)=(\max(0,1-r^2))^7$. Cutoff function, *Hald's infinite-order*, $\eta=1.1$, $\delta=0.3\sqrt{h}$, $\Delta t=4.0\,h$, $T_{\max}=50.0$.

which is quite close to what we observed. Beale and Majda [7] also applied rezoning to this problem, with $h=0.125$ and $\delta=0.25$. In comparing our results with [illegible]

$$\mathbf{U}^2=\iint_{r\leqslant 1}|\mathbf{u}|^2\,dx\,dy.$$

For test problem 1, $\mathbf{U}\approx0.2668$, so if we divide our values of E_u in Tables Ia and Ib by 0.2668, we obtain relative velocity errors ranging from 0.006% at time $t=0$ to 0.09% at time $t=35$, using infinite-order cutoff with $h=0.125$ and $\delta=0.3\sqrt{h}\approx0.1061$. With the eighth-order cutoff and Version 2 of the rezoning, the corresponding relative errors are 0.007% at time $t=0$ and 0.13% at time $t=35$. Beale and Majda [7] reported relative errors of 0.055% at time $t=0$ and 0.30%

TABLE IIb

Velocity Errors for Test Problem 2

	E_u			
t	$h=0.125$	$h=0.100$	$h=0.0625$	$h=0.05$
0.0	0.1352×10^{-4}	0.4651×10^{-5}	0.8460×10^{-6}	0.3576×10^{-6}
10.0	0.6466×10^{-4}	0.2441×10^{-4}	0.3822×10^{-5}	0.1518×10^{-5}
20.0	0.1521×10^{-3}	0.5617×10^{-4}	0.8522×10^{-5}	0.3209×10^{-5}
30.0	0.2642×10^{-3}	0.9726×10^{-4}	0.1414×10^{-4}	0.5246×10^{-5}
40.0	0.3964×10^{-3}	0.1460×10^{-3}	0.2057×10^{-4}	0.7588×10^{-5}
50.0	0.5477×10^{-3}	0.2034×10^{-3}	0.2794×10^{-4}	0.1023×10^{-4}

Note. $\omega(r)=(\max(0,1-r^2))^7$. Cutoff function, *8th order with compact support*, $\eta=1.1$, $\delta=1.7\sqrt{h}$, $\Delta t=4.0h$, $T_{\max}=50.0$.

TABLE IIIa

Velocity Errors for Test Problem 3

	E_u			
t	$h=0.125$	$h=0.100$	$h=0.0625$	$h=0.05$
0.0	0.4859×10^{-4}	0.1782×10^{-4}	0.1304×10^{-5}	0.2337×10^{-6}
10.0	0.2033×10^{-3}	0.8081×10^{-4}	0.4243×10^{-5}	0.7356×10^{-6}
20.0	0.4347×10^{-3}	0.1502×10^{-3}	0.8455×10^{-5}	0.1445×10^{-5}
30.0	0.7052×10^{-3}	0.2375×10^{-3}	0.1321×10^{-4}	0.2184×10^{-5}
40.0	0.1000×10^{-2}	0.3369×10^{-3}	0.1804×10^{-4}	0.2940×10^{-5}
50.0	0.1339×10^{-2}	0.4475×10^{-3}	0.2357×10^{-4}	0.3711×10^{-5}

Note. $\omega(r)=\max(0,1-r^2))^{14}$. Cutoff function, *Hald's infinite-order*, $\eta=1.1$, $\delta=0.3\sqrt{h}$, $\Delta t=4.0\,h$, $T_{\max}=50.0$.

at time $t=36$, using rezoning version 1. Beale [4] introduced a different approach to reducing the error for large times. Using the method in [4] with a fourth-order cutoff function, and on the same test problem, he obtained a relative error of 0.06% at time $t=0$ and a maximum relative error of 1.0% for $t<40$. While this is considerably better than using the original method, it is not as accurate as either of the two rezoning methods with the higher-order cutoff functions.

The results of test problem 2 are summarized in Tables IIa, IIb. As in test problem 1, we pick $\delta=1.7\sqrt{h}$, for the eighth-order cutoff and $\delta=0.3\sqrt{h}$ for the infinite-order cutoff. Numerical tests have shown that these values of δ are close to the optimal ones at time $t=0$, with $h=0.100$, even for this vorticity distribution. Comparing Table IIa to Table IIb, we see that the errors using the infinite-order cutoff are smaller than the corresponding errors for the eighth-order cutoff by a factor ranging from about 3 to 6. Nevertheless, the rate of convergence is around 4 for both methods at all times. Theoretically, the moment error for infinite-order

TABLE IIIb

Velocity Errors for Test Problem 3

	E_u			
t	$h=0.125$	$h=0.100$	$h=0.0625$	$h=0.05$
0.0	0.5584×10^{-4}	0.2759×10^{-4}	0.5139×10^{-5}	0.2264×10^{-5}
10.0	0.2795×10^{-3}	0.1289×10^{-3}	0.2214×10^{-4}	0.9603×10^{-5}
20.0	0.7324×10^{-3}	0.2955×10^{-3}	0.4790×10^{-4}	0.2050×10^{-4}
30.0	0.1238×10^{-2}	0.7755×10^{-3}	0.7981×10^{-4}	0.3306×10^{-4}
40.0	0.1807×10^{-2}	0.1819×10^{-2}	0.1149×10^{-3}	0.4732×10^{-4}
50.0	0.2522×10^{-2}	0.3071×10^{-2}	0.1554×10^{-3}	0.6337×10^{-4}

Note. $\omega(r)=(\max(0,1-r^2))^{14}$. Cutoff function, *8th order with compact support*, $\eta=1.1$, $\delta=1.7\sqrt{h}$, $\Delta t=4.0\,h$, $T_{\max}=50.0$.

TABLE IVa

Rate of Convergence of the Velocity Approximations in Test Problems 1–3, Using Hald's Infinite Order Cutoff

	Rate of Convergence		
t	$\omega(r)=(1.0-r^2)^3$	$\omega(r)=(1.0-r^2)^7$	$\omega(r)=(1.0-r^2)^{14}$
0.0	2.3	4.1	7.7
10.0	2.6	4.0	7.8
20.0	2.8	4.0	7.9
30.0	2.8	4.0	8.1
40.0	2.9	4.1	8.1
50.0	3.0	4.1	8.3

Note. $\delta=0.3\sqrt{h}$.

cutoffs is $O(\delta^{7.5})$ for this vorticity distribution; so, since δ is proportional to $\sqrt{h}$, we would expect a rate of convergence of 3.75 in this case. Hence, the observed rate of convergence is slightly higher than the theoretical rate as in test problem 1. Perlman [21] tested Gaussian cutoff functions of different orders on this vorticity distribution, but without rezoning. Using an eighth-order Gaussian cutoff, she had to take $\delta=h^{0.7}$ to minimize the velocity consistency error at time $t=10$ and $\delta=h^{0.6}$ to minimize the consistency error at time $t=20$. With these parameters and $h=0.05$, she obtained a minimum consistency error of 7.42×10^{-5} at time $t=10$ and 5.77×10^{-4} at time $t=20$. We should point out that the total error is larger than the consistency error. Our smallest (total) errors at time $t=10$ and $t=20$, with $h=0.05$ are 3.59×10^{-7} and 7.36×10^{-7}, respectively, so the rezoning procedure seems to pay off, at least when the flow is this smooth.

In test problem 3, the difference in velocity errors between the two cutoff functions is small for $h=0.125$, but it increases as h gets smaller (Tables IIIa, IIIb). For

TABLE IVb

Rate of Convergence of the Velocity Approximations in Test Problems 1–3, Using the Eighth-Order Cuttoff

	Rate of convergence		
t	$\omega(r)=(1.0-r^2)^3$	$\omega(r)=(1.0-r^2)^7$	$\omega(r)=(1.0-r^2)^{14}$
0.0	2.3	3.9	3.7
10.0	2.7	4.1	3.7
20.0	2.7	4.4	3.8
30.0	2.7	4.4	4.0
40.0	2.7	4.5	4.0
50.0	2.8	4.5	4.0

Note. $\delta=1.7\sqrt{h}$.

$h=0.05$ the error is 9.7 times smaller at time $t=0$ and 17.1 times smaller at time $t=50$ for the infinite-order cutoff compared to the eight-order cutoff. The rate of convergence is close to 8 for the infinite-order cutoff, but as expected around 4 for the eighth-order cutoff. The theoretical rate of convergence for the infinite-order cutoff is 7.25 in this case, since the moment error is $O(\delta^{14.5})$, so once again the observed rate of convergence is higher than the theoretical rate. We also made a comparison of rezoning version 1 vs rezoning version 2 using test problems 1–3 with the infinite-order cutoff. The results are shown graphically in Figs. 4.1–4.3. We see that Version 2 gives a significantly lower error and that the gap between the two versions increases with increasing smoothness of the flow. The sharp peaks in the graphs are due to the fact that sometimes the velocity error increases faster than the vorticity error. Then, after rezoning, the velocity error decreases again. In practice, these peaks do not matter, since the error at any time is much smaller than that obtained without rezoning, as we see in Fig. 4.4.

In the fourth test problem, we distribute the vorticity on two circles according to

$$\omega(x, y) = (\max(0, (1-4(|x|-0.5)^2-4y^2)))^7. \tag{4.6}$$

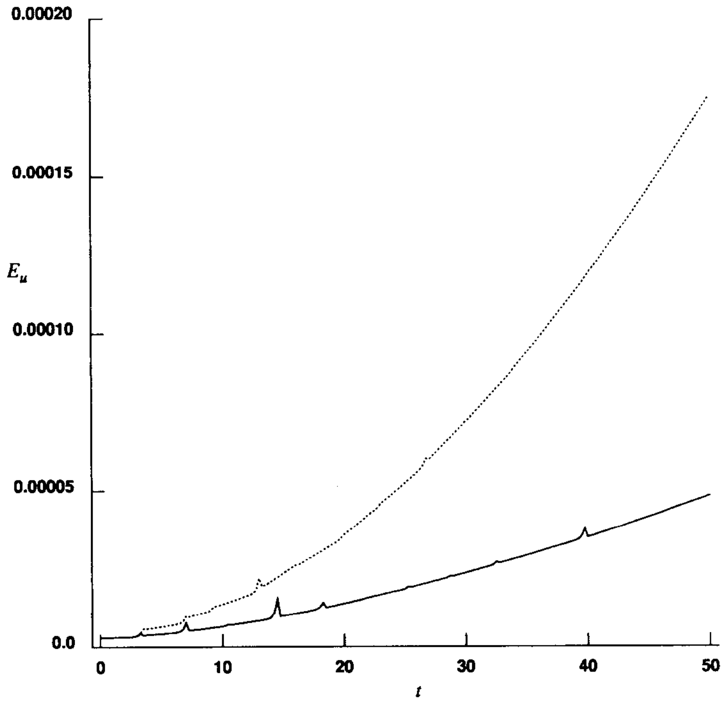

FIG. 4.1. Rezoning, Version 2 vs Version 1; $\omega(r) = (\max(0, 1-r^2))^3$, solid curve = Version 2, dotted curve = Version 1, $h=0.0625$, $\delta=0.3\sqrt{h}$, $\eta=1.25$, $\Delta t=4.0\,h$.

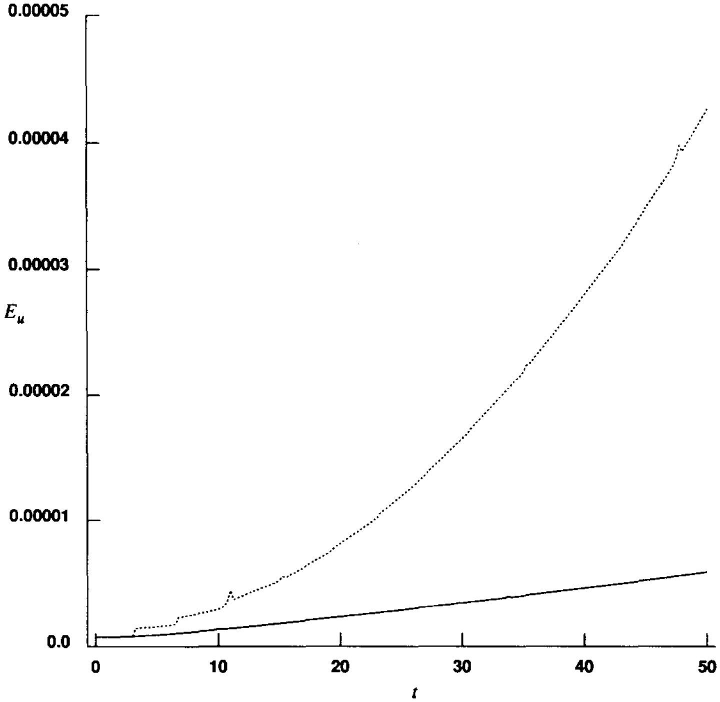

FIG. 4.2. Rezoning, Version 2 vs Version 1; $\omega(r) = (\max(0, 1 - r^2))^7$, solid curve = Version 2, dotted curve = Version 1, $h = 0.0625$, $\delta = 0.355\sqrt{h}$, $\eta = 1.25$, $\Delta t = 4.0\,h$.

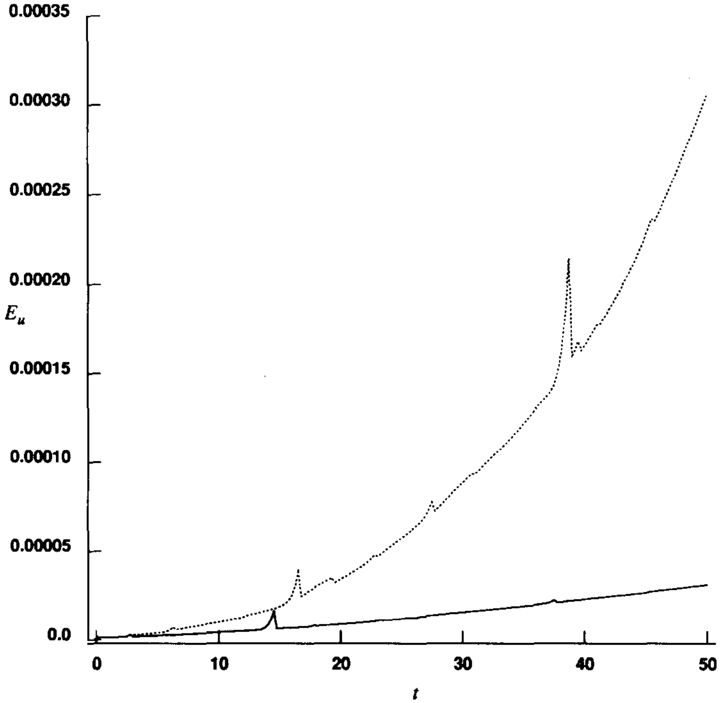

FIG. 4.3. Rezoning, Version 2 vs Version 1; $\omega(r) = (\max(0, 1 - r^2))^{14}$, solid curve = Version 2, dotted curve = Version 1, $h = 0.0625$, $\delta = 0.3\sqrt{h}$, $\eta = 1.25$, $\Delta t = 4.0\,h$.

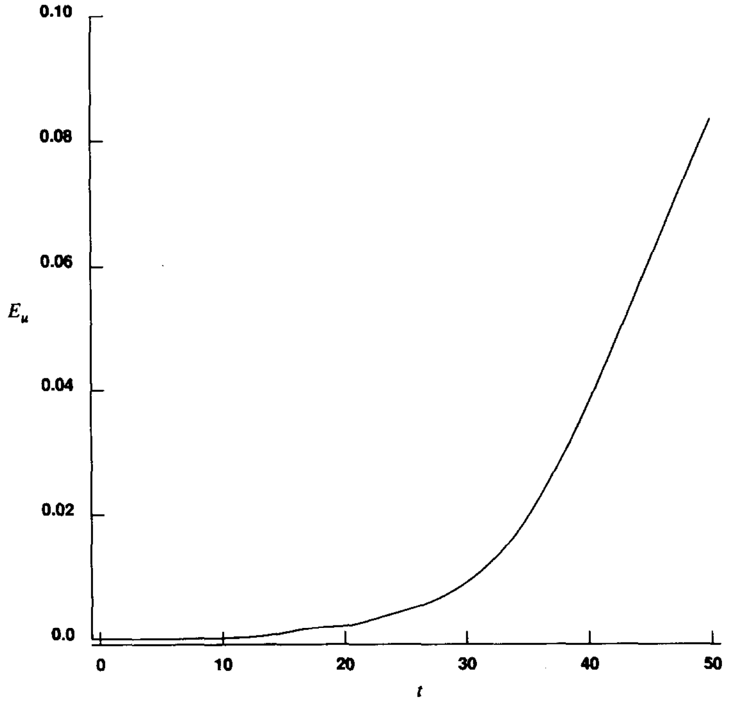

FIG. 4.4. Original vortex method; $\omega(r)=(\max(0,1-r^2))^7$, $h=0.0625$, $\delta=0.5\sqrt{h}$, $\Delta t=4.0\,h$.

Thus, we have two vorticity patches with the vorticity distributed as in test problem 2. Note that this test problem differs from the famous test problem considered by Christiansen [12]. Christiansen [12] used uniform vorticity distribution within the two circles. However, in our test problem the vorticity is concentrated at the centers of the circles and decays to 0 in a smooth fashion as we approach the boundary. The numerical solution using Hald's infinite-order cutoff with rezoning is shown in Figs. 4.5–4.15. The graphs represent vorticity level sets at different times from time $t=0$ to time $t=100$. To estimate the rate of convergence, we have used Richardson's extrapolation with three different gridsizes h, $2h/3$, and $h/2$. Assuming the rate of convergence is q, we can write $\tilde{\mathbf{u}}_i=\mathbf{u}_i+h^q e(x,y,t)+$ (higher order terms). Then

$$\frac{\|\tilde{\mathbf{u}}_i^h-\tilde{\mathbf{u}}_i^{2h/3}\|}{\|\tilde{\mathbf{u}}_i^{2h/3}-\tilde{\mathbf{u}}_i^{h/2}\|}\sim\frac{h^q-(2h/3)^q}{(2h/3)^q-(h/2)^q}=\frac{1-(2/3)^q}{(2/3)^q-(1/2)^q}. \tag{4.7}$$

The norm is taken to be the discrete L_2 norm of the differences in the computed velocities for vortices with the same initial positions. Once we have computed the first quotient in (4.7), we set the third quotient equal to this value and solve for q

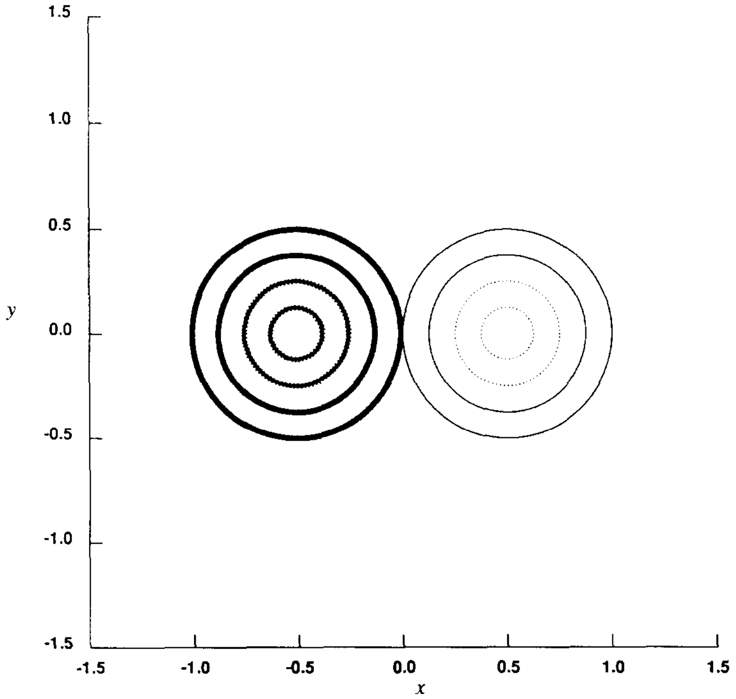

FIG. 4.5. Vorticity level sets; time = 0.0, $\omega(x, y, 0) = (\max(0, (1-4(|x|-0.5)^2-4y^2)))^7$, $h = 0.0625$, $\delta = 0.28\sqrt{h}$, $\eta = 1.25$, $\varepsilon = 0.00004\,h^2$, $\Delta t = 5.0\,h$.

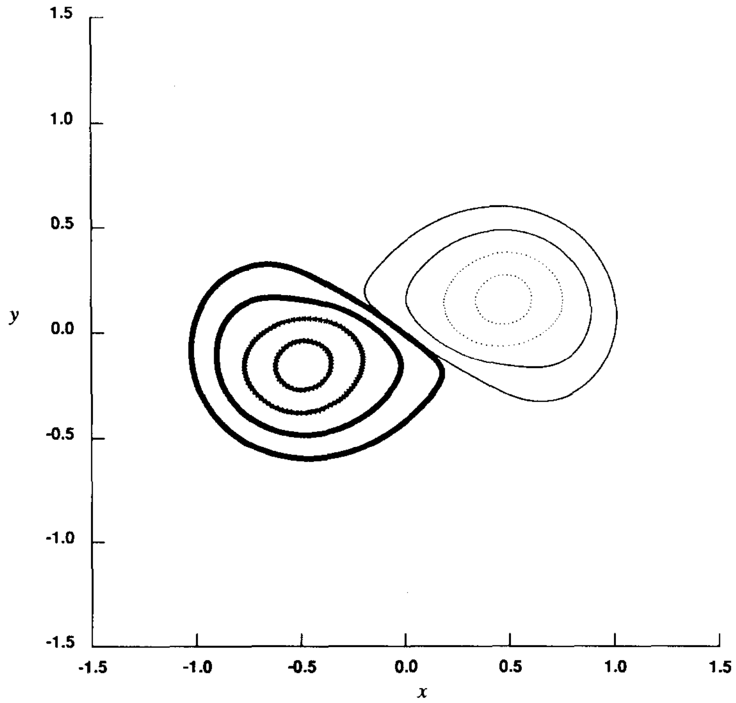

FIG. 4.6. Vorticity level sets; time = 10.0, $\omega(x, y, 0) = (\max(0, (1-4(|x|-0.5)^2-4y^2)))^7$, $h = 0.0625$, $\delta = 0.28\sqrt{h}$, $\eta = 1.25$, $\varepsilon = 0.00004\,h^2$, $\Delta t = 5.0\,h$.

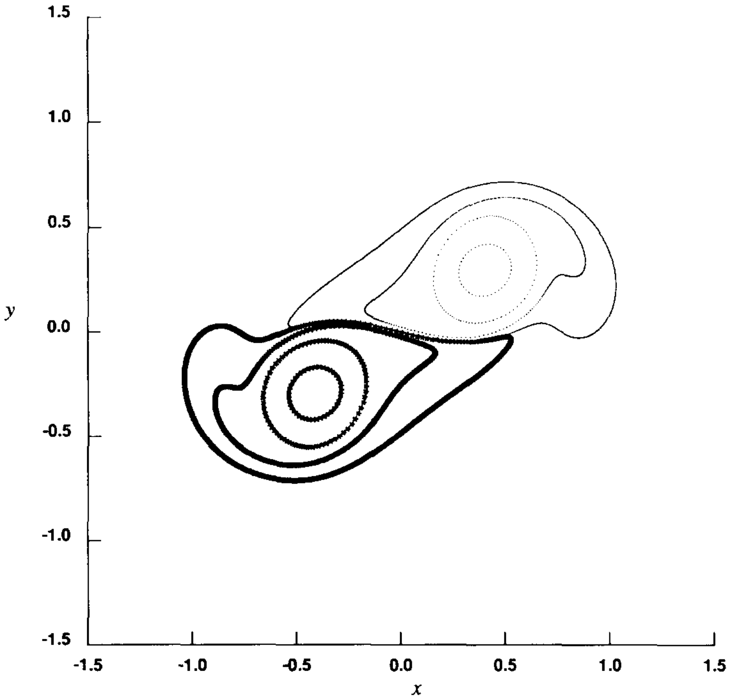

FIG. 4.7. Vorticity level sets; time = 20.0, $\omega(x, y, 0) = (\max(0, (1-4(|x|-0.5)^2-4y^2)))^7$, $h = 0.0625$, $\delta = 0.28\sqrt{h}$, $\eta = 1.25$, $\varepsilon = 0.00004\, h^2$, $\Delta t = 5.0\, h$.

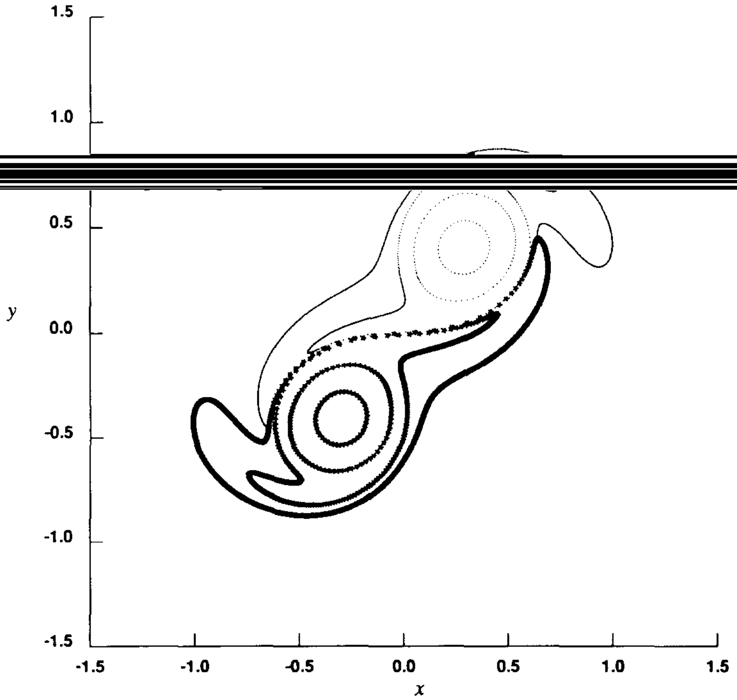

FIG. 4.8. Vorticity level sets; time = 30.0, $\omega(x, y, 0) = (\max(0, (1-4(|x|-0.5)^2-4y^2)))^7$, $h = 0.0625$, $\delta = 0.28\sqrt{h}$, $\eta = 1.25$, $\varepsilon = 0.00004\, h^2$, $\Delta t = 5.0\, h$.

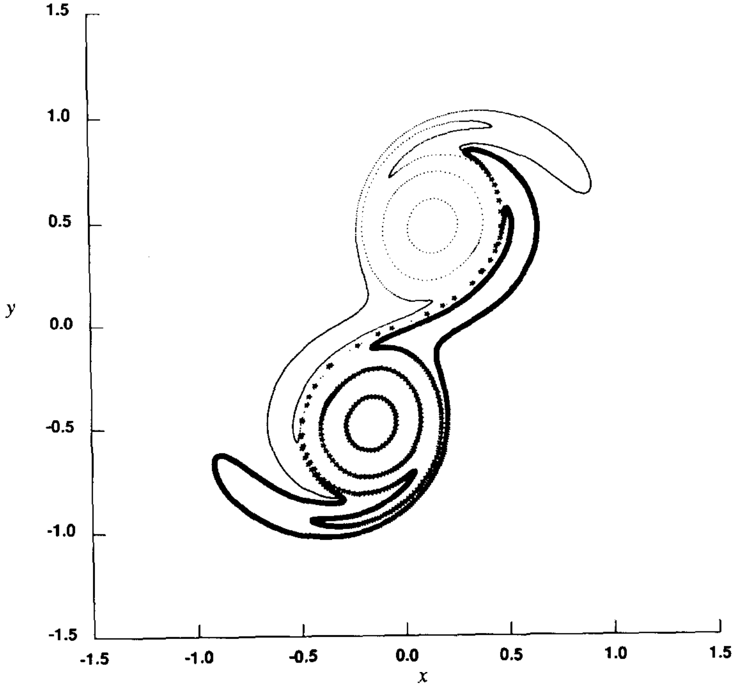

FIG. 4.9. Vorticity level sets; time = 40.0, $\omega(x, y, 0) = (\max(0, (1 - 4(|x| - 0.5)^2 - 4y^2)))^7$, $h = 0.0625$, $\delta = 0.28\sqrt{h}$, $\eta = 1.25$, $\varepsilon = 0.00004\,h^2$, $\Delta t = 5.0\,h$.

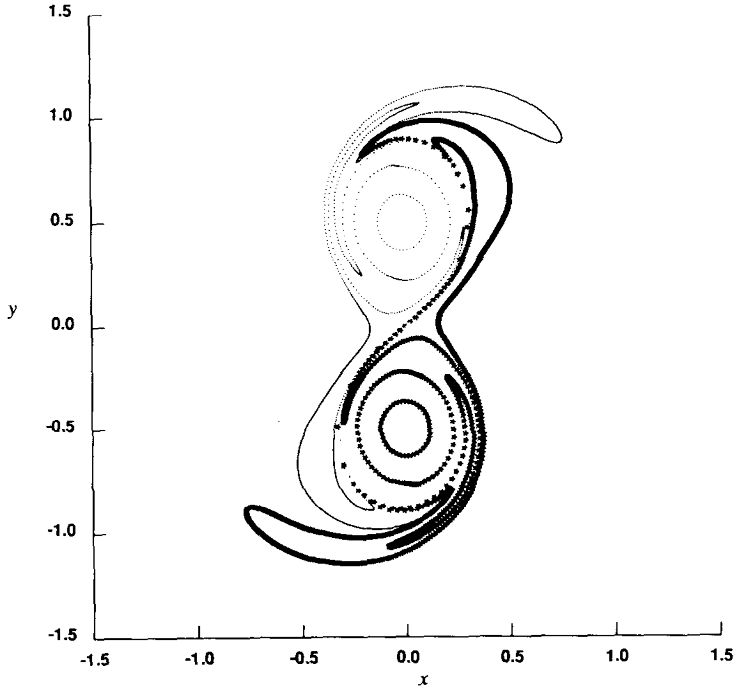

FIG. 4.10. Vorticity level sets; time = 50.0, $\omega(x, y, 0) = (\max(0, (1 - 4(|x| - 0.5)^2 - 4y^2))^7$, $h = 0.0625$, $\delta = 0.28\sqrt{h}$, $\eta = 1.25$, $\varepsilon = 0.00004\,h^2$, $\Delta t = 5.0\,h$.

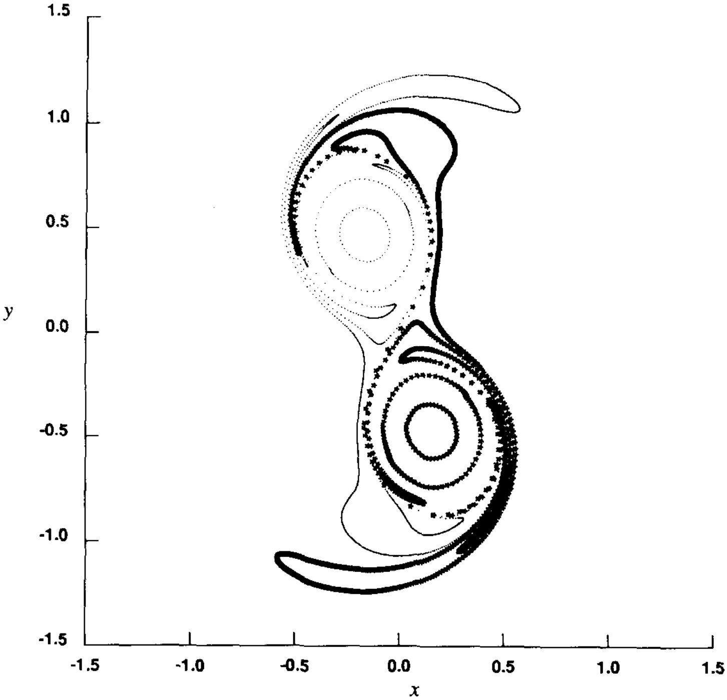

FIG. 4.11. Vorticity level sets; time = 60.0, $\omega(x, y, 0) = (\max(0, (1-4(|x|-0.5)^2-4y^2)))^7$, $h = 0.0625$, $\delta = 0.28\sqrt{h}$, $\eta = 1.25$, $\varepsilon = 0.00004\,h^2$, $\Delta t = 5.0\,h$.

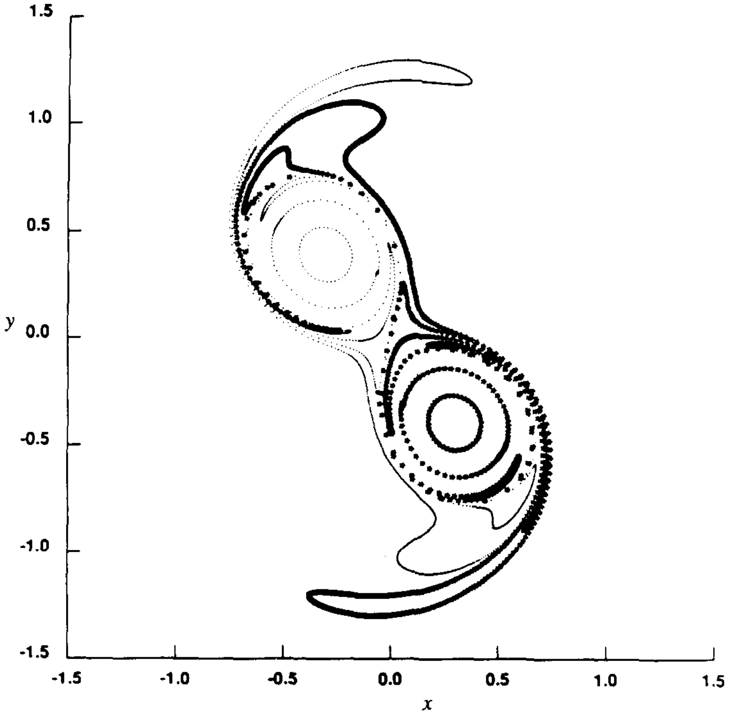

FIG. 4.12. Vorticity level sets; time = 70.0, $\omega(x, y, 0) = (\max(0, (1-4(|x|-0.5)^2-4y^2)))^7$, $h = 0.0625$, $\delta = 0.28\sqrt{h}$, $\eta = 1.25$, $\varepsilon = 0.00004\,h^2$, $\Delta t = 5.0\,h$.

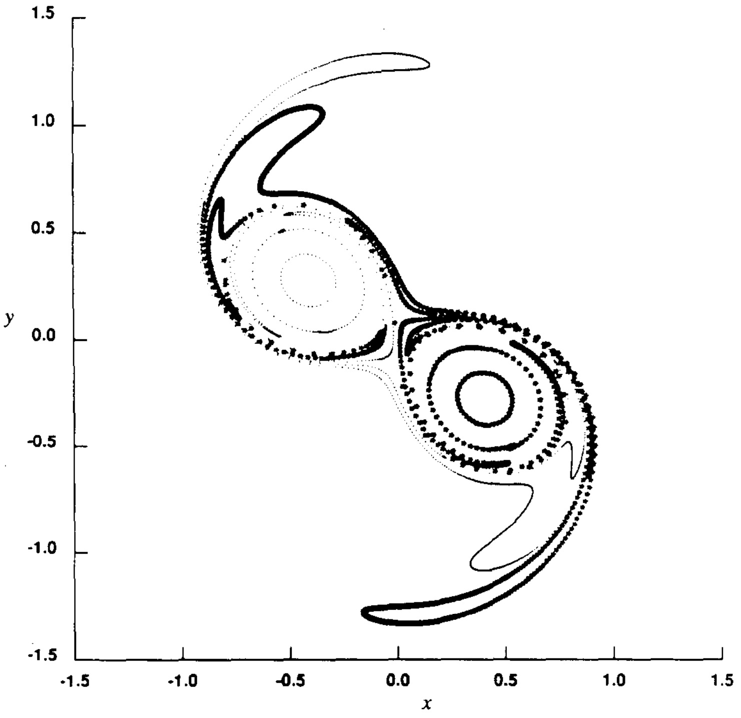

FIG. 4.13. Vorticity level sets; time = 80.0, $\omega(x, y, 0) = (\max(0, (1-4(|x|-0.5)^2-4y^2)))^7$, $h = 0.0625$, $\delta = 0.28\sqrt{h}$, $\eta = 1.25$, $\varepsilon = 0.00004\, h^2$, $\Delta t = 5.0\, h$.

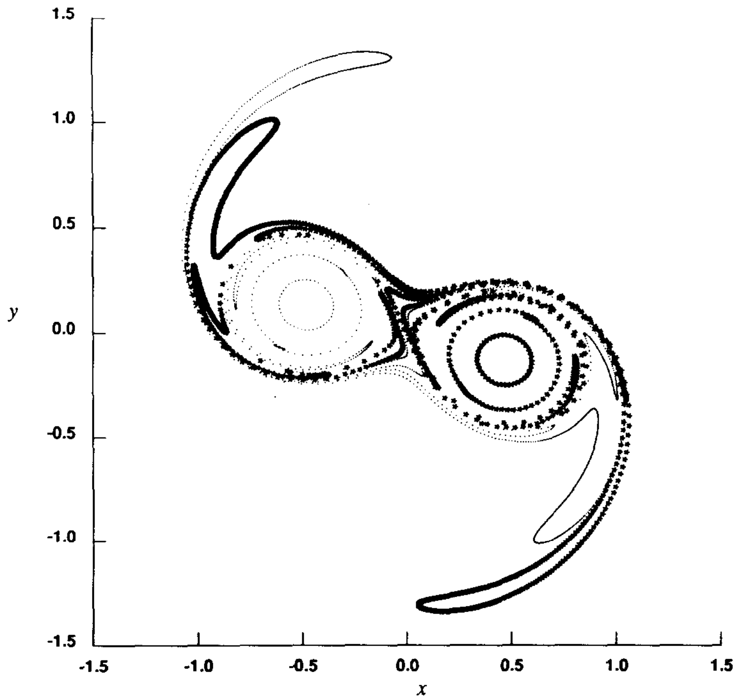

FIG. 4.14. Vorticity level sets; time = 90.0, $\omega(x, y, 0) = (\max(0, (1-4(|x|-0.5)^2-4y^2)))^7$, $h = 0.0625$, $\delta = 0.28\sqrt{h}$, $\eta = 1.25$, $\varepsilon = 0.00004\, h^2$, $\Delta t = 5.0\, h$.

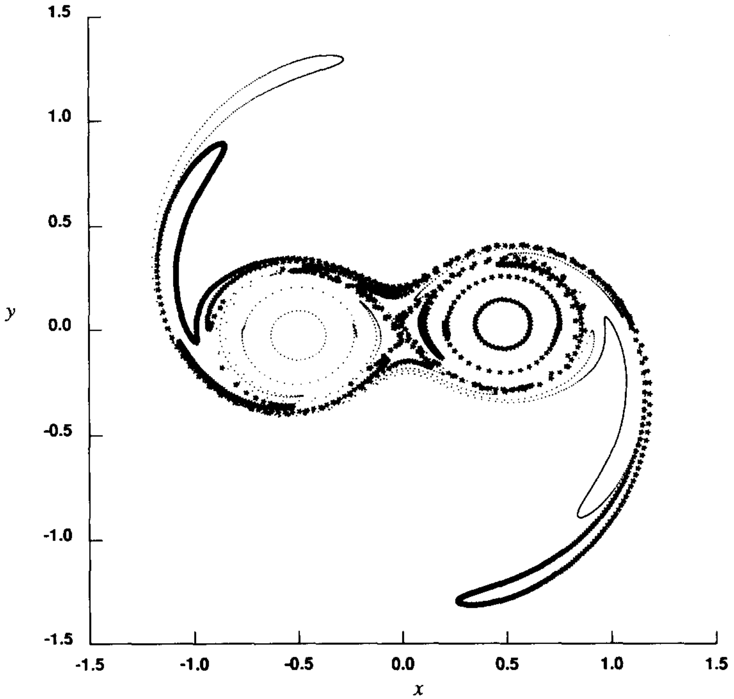

FIG. 4.15. Vorticity level sets; time = 100.0, $\omega(x, y, 0) = (\max(0, (1-4(|x|-0.5)^2-4y^2)))^7$, $h = 0.0625$, $\delta = 0.28\sqrt{h}$, $\eta = 1.25$, $\varepsilon = 0.00004\, h^2$, $\Delta t = 5.0\, h$.

numerically. Using the three grid-sizes $h = \frac{1}{10}$, $h = \frac{1}{15}$, and $h = \frac{1}{20}$, we obtain the rates of convergence in Table V. We see that the rates of convergence for this problem are similar to the rates observed in Problem 2.

In the fifth test problem, the initial vorticity is distributed on a square according to

$$\omega(x, y) = ((\max(0, 1 - x^2))(\max(0, 1 - y^2)))^7. \tag{4.8}$$

The rates of convergence at different times up to time $t = 50$ are estimated in the same way as in test problem 4, using the same three grid-sizes. Here, we had to take a larger value of δ in order to maintain a high rate of convergence up to time $t = 50$. The observed rates of convergence are lower than in problems 2 and 4, although the initial vorticity distribution has the same smoothness in this case. It is possible that the Fourier transform of the vorticity distribution has a lower rate of decay in this case, causing a lower rate of convergence. Figures 4.16–4.22 show the computed vorticity level sets at times $t = 0$, 10, 20, 30, 40, 50, and 100.

Finally, we made a comparison between the direct method of evaluating the sum in (1.7) and the Rokhlin–Greengard algorithm [22]. For this, we used the vorticity distribution of test problem 2, the eighth-order cutoff function, and $\delta = 1.7\sqrt{h}$. The

TABLE V

Rate of Convergence of the Velocity Approximations in Test Problem 4, Using Hald's Infinite Order Cutoff

t	Rate of convergence
0.0	3.7
10.0	4.1
20.0	4.3
30.0	4.1
40.0	4.4
50.0	4.4
60.0	4.2
70.0	4.6
80.0	4.5
90.0	4.5
100.0	4.9

Note. $\omega(x, y, 0) = (\max(0, (1-4(|x|-0.5)^2-4y^2)))^7$; $h=0.0625$, $\delta=0.28\sqrt{h}$, $\eta=1.25$, $\varepsilon=0.00004\,h^2$, $\Delta t=5.0\,h$.

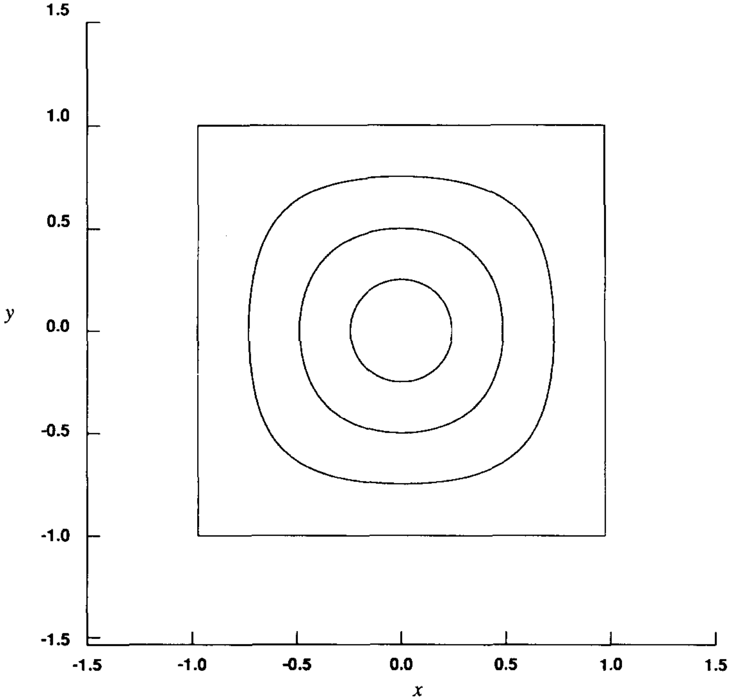

FIG. 4.16. Vorticity level sets; time $=0.0$, $\omega(x, y, 0) = [[\max(0, 1-x^2))(\max(0, 1-y^2)))^7$, $h=0.0625$, $\delta=0.355\sqrt{h}$, $\eta=1.5$, $\Delta t=4.0\,h$.

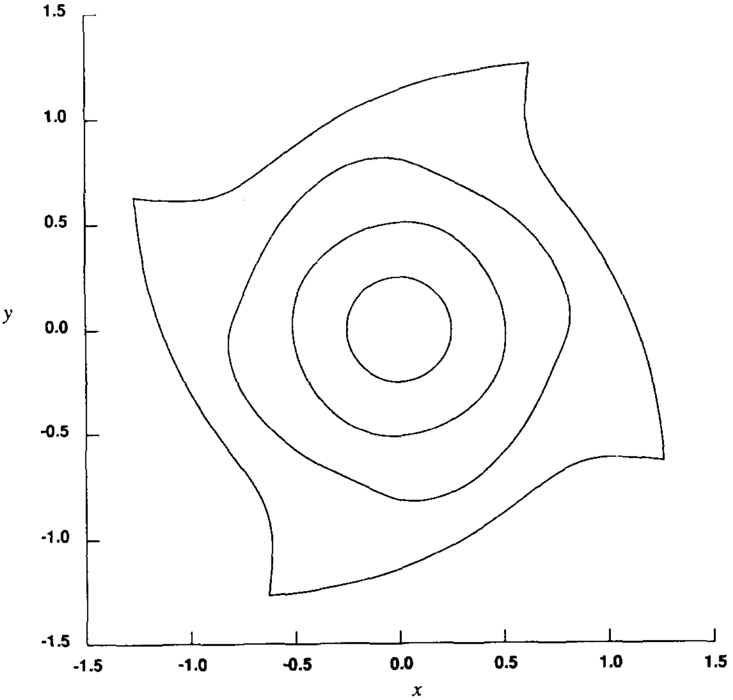

FIG. 4.17. Vorticity level sets; time = 10.0, $\omega(x, y, 0) = ((\max(0, 1-x^2))(\max(0, 1-y^2)))^7$, $h = 0.0625$, $\delta = 0.355\sqrt{h}$, $\eta = 1.5$, $\Delta t = 4.0\,h$.

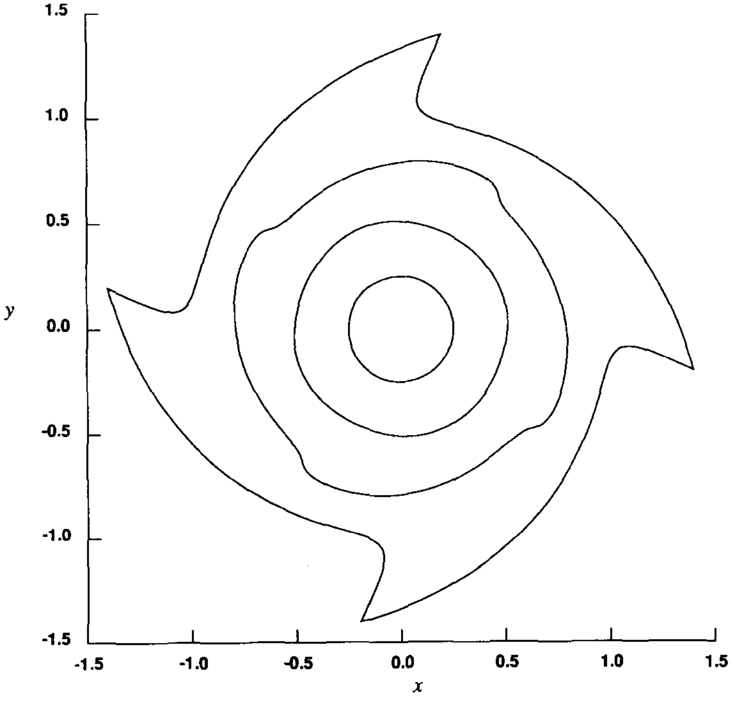

FIG. 4.18. Vorticity level sets; time = 20.0, $\omega(x, y, 0) = ((\max(0, 1-x^2))(\max(0, 1-y^2)))^7$, $h = 0.0625$, $\delta = 0.355\sqrt{h}$, $\eta = 1.5$, $\Delta t = 4.0\,h$.

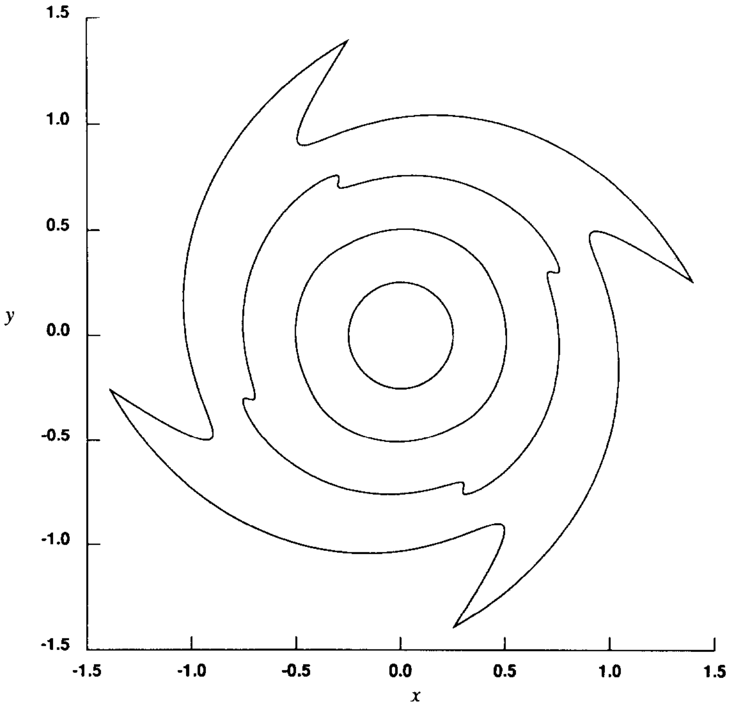

FIG. 4.19. Vorticity level sets; time = 30.0, $\omega(x, y, 0) = ((\max(0, 1-x^2))(\max(0, 1-y^2)))^7$, $h = 0.0625$, $\delta = 0.355\sqrt{h}$, $\eta = 1.5$, $\Delta t = 4.0\,h$.

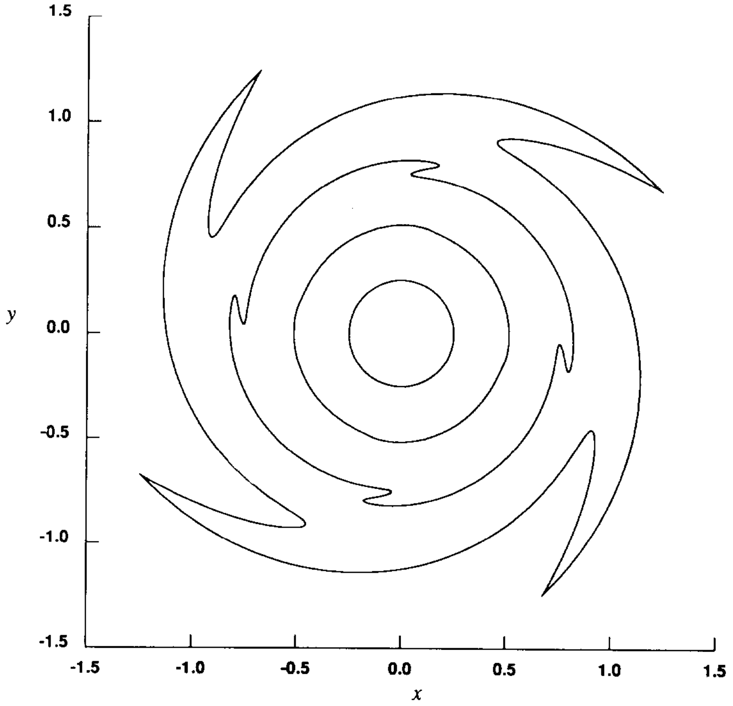

FIG. 4.20. Vorticity level sets; time = 40.0, $\omega(x, y, 0) = ((\max(0, 1-x^2))(\max(0, 1-y^2)))^7$, $h = 0.0625$, $\delta = 0.355\sqrt{h}$, $\eta = 1.5$, $\Delta t = 4.0\,h$.

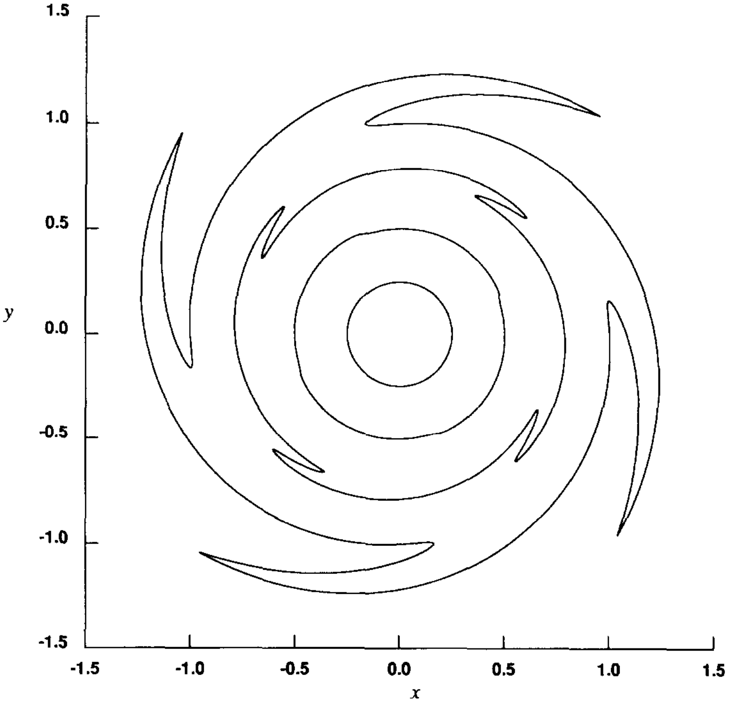

FIG. 4.21. Vorticity level sets; time = 50.0, $\omega(x, y, 0) = ((\max(0, 1-x^2))(\max(0, 1-y^2)))^7$, $h = 0.0625$, $\delta = 0.355\sqrt{h}$, $\eta = 1.5$, $\Delta t = 4.0\ h$.

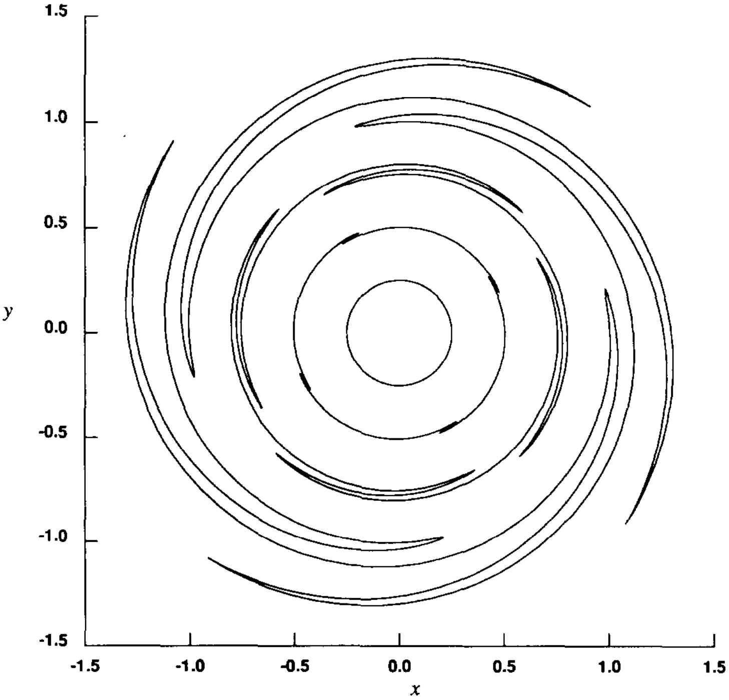

FIG. 4.22. Vorticity level sets; time = 100.0, $\omega(x, y, 0) = ((\max(0, 1-x^2))(\max(0, 1-y^2)))^7$, $h = 0.0625$, $\delta = 0.355\sqrt{h}$, $\eta = 1.5$, $\Delta t = 4.0\ h$.

TABLE VI

Rate of Convergence of the Velocity Approximations in Test Problem 5, Using Hald's Infinite Order Cutoff

t	Rate of convergence
0.0	3.2
10.0	3.2
20.0	3.2
30.0	3.3
40.0	3.3
50.0	3.3

Note. $\omega(x, y, 0) = ((\max(0, 1 - x^2))(\max(0, 1 - y^2)))^7$; $h = 0.0625$, $\delta = 0.6\sqrt{h}$, $\eta = 1.5$, $\Delta t = 4.0\,h$.

TABLE VII

A Comparison between the Direct Method and the Rokhlin–Greengard Method

		Direct method		Rokhlin–Greengard method	
h	N	E_u	CPU time	E_u	CPU time
1/16	797	0.8460×10^{-6}	1.00	0.8459×10^{-6}	0.87
1/20	1257	0.3576×10^{-6}	2.40	0.3575×10^{-6}	1.46
1/32	3209	0.5927×10^{-6}	14.7	0.5927×10^{-6}	5.46
1/40	5025	0.2507×10^{-7}	33.3	0.2508×10^{-7}	10.1
1/64	12853	0.4021×10^{-8}	218.2	0.4034×10^{-8}	40.7

Note. $\delta = 1.7\sqrt{h}$.

TABLE VIII

A Comparison between the Direct Method and the Rokhlin–Greengard Method

		Direct method		Rokhlin–Greengard method	
h	N	E_u	CPU time	E_u	CPU time
1/16	797	0.4889×10^{-6}	0.96	0.4889×10^{-6}	0.76
1/20	1257	0.1109×10^{-6}	2.32	0.1108×10^{-6}	1.28
1/32	3209	0.1622×10^{-7}	14.4	0.1613×10^{-7}	4.3
1/40	5025	0.6553×10^{-8}	32.8	0.6548×10^{-8}	8.3
1/64	12853	0.9789×10^{-9}	215.0	0.1095×10^{-8}	30.4

Note. $\delta = 1.42\sqrt{h}$.

number of terms in the multipole expansion, see [22], was set to 16. The results are summarized in Table VII. Here, N stands for the number of vortices, E_u for the velocity error at time $t=0$, and the CPU time is given in minutes for one velocity evaluation at time $t=0$ on a Gould computer. We have to emphasize that the speed of the Rokhlin–Greengard algorithm applied to vortex methods is limited by the size of the cutoff parameter δ. In fact, we set the length of a side of each box at the highest level of refinement equal to $\delta + 0.01$. The number of levels of refinements is then taken to be the smallest integer l such that $\delta \cdot 2^{l-1} \geqslant 1.0$. This is exactly what we need to cover the support of the initial vorticity distribution in test problem 2. The Rokhlin–Greengard algorithm runs faster if we choose δ smaller, since this decreases the number of "local" interactions. To demonstrate this, we repeated the calculations using the same cutoff function but with $\delta = 1.42\sqrt{h}$. See Table VIII. The errors are also smaller for the smaller δ, but unless we apply rezoning, this is only true for time $t=0$. It is clear that the rezoning frequency must increase as δ decreases. If δ is taken still much smaller, we will see an increase in the error even for time $t=0$. Therefore, to be able to take δ so small that the algorithm will run with optimal speed, we would have to choose a lower order cutoff function, which we do not believe is such a good idea for smooth flows. However, if the flow is not very smooth, we may very well use a lower order cutoff function, a smaller δ, and use the Rokhlin–Greengard algorithm with maximum efficiency.

It is also important to calculate the "local" interactions in real variables, as in the direct method, even though the non-local interactions are calculated in terms of complex variables. Otherwise the computational speed of the Rokhlin–Greengard algorithm decreases.

Acknowledgments

The author thanks Scott Baden, Paul Concus, Ole Hald, Gerry Puckett, and James Sethian for helpful discussions. The calculations presented in this paper were carried out at the Lawrence Berkeley Laboratory, and in part at the Center of Research and Advanced Studies in Mexico City, Mexico.

References

1. C. R. Anderson, *J. Comput. Phys.* **62**, 111 (1986).
2. C. R. Anderson, *J. Comput. Phys.* **61**, 417 (1985).
3. C. R. Anderson and C. Greengard, *SIAM J. Numer. Anal.* **22**, 413 (1985).
4. J. T. Beale, in *Proceedings of the Workshop on Computational Fluid Dynamics and Reacting Gas Flows, I.M.A., University of Minnesota, 1986.*
5. J. T. Beale and A. Majda, *Math. Comput.* **39**, 1 (1982).
6. J. T. Beale and A. Majda, *Math. Comput.* **39**, 29 (1982).
7. J. T. Beale and A. Majda, *J. Comput. Phys.* **58**, 188 (1985).
8. A. Y. Cheer, *SIAM J. Sci. Stat. Comput.* **4**, 685 (1983).
9. A. J. Chorin, *J. Fluid Mech.* **57**, 785 (1973).
10. A. J. Chorin, *SIAM J. Sci. Stat. Comput.* **1**, 1 (1980).

11. A. J. Chorin and J. Marsden, *A Mathematical Introduction to Fluid Mechanics* (Springer-Verlag, New York, 1979).
12. J. P. Christiansen, *J. Comput. Phys.* **13**, 363 (1973).
13. G. H. Cottet, Thèse de 3ème cycle, l'Université Pierre et Marie Curie, Paris 6, 1982 (unpublished).
14. O. H. Hald, *SIAM J. Numer. Anal.* **16**, 726 (1979).
15. O. H. Hald, *SIAM J. Numer. Anal.* **24**, 538 (1987).
16. O. H. Hald and V. M. Del Prete, *Math. Comput.* **32**, 791 (1978).
17. F. John, *Partial Differential Equations* (Springer-Verlag, New York, 1982).
18. A. Leonard and P. R. Spalart, in *AIAA 14th Fluid and Plasma Dynamics Conference, AIAA-81-1246, 1981.*
19. Y. Nakamura, A. Leonard, and P. R. Spalart, in *AIAA/ASME 3rd Joint Thermophysics, Fluids, Plasma and Heat Transfer Conference, AIAA-82-0948, 1982.*
20. H. O. Nordmark, Ph.D. thesis, Lawrence Berkeley Livermore Report No. 25259, 1988 (unpublished).
21. M. Perlman, *J. Comput. Phys.* **59**, 200 (1985).
22. V. Rokhlin and L. Greengard, *J. Comput. Phys.* **73**, 325 (1987).
23. J. A. Sethian, *J. Comput. Phys.* **54**, 425 (1984).
24. J. A. Sethian and A. F. Ghoniem, *J. Comput. Phys.* **74**, 283 (1988).
25. P. R. Spalart, Ph.D. thesis, Department of Aeronautics and Astronautics, Stanford University, 1982 (unpublished).