ON THE ACCURACY OF VORTEX METHODS AT LARGE TIMES

J. Thomas Beale
Department of Mathematics
Duke University
Durham, NC 27706

Vortex methods simulate incompressible flow, without viscosity or at high Reynolds number, by a collection of computational elements of vorticity which are transported along computed particle paths. The velocity field can be computed from the vorticity in order to move the elements forward in time. Here we will survey the formulation and convergence theory of such methods, primarily for inviscid flow without boundaries in two or three dimensions. We also discuss a modification of the basic method intended to improve accuracy at later times and illustrate its performance with a simple test problem. It is found that the error is significantly reduced in this case. This modified method can be shown to converge, and details of the proof will be given elsewhere. Very similar ideas have been experimented with by Chris Anderson, and it is a pleasure to thank him for his helpful comments and suggestions.

For the important case of incompressible flow past boundaries at high Reynolds number, methods such as those described here can be used as one part of a general method developed by Chorin and others [12-14,31]. In this setting additional elements of vortex sheets are generated at the boundary at each time step to satisfy the no-slip condition, and a random walk simulates the effect of viscosity. Calculations illustrating the ability of this method to capture important features of the flow with good accuracy have been done by a number of workers, most recently by Ghoneim, Sethian et al. [20,21]. T. Leonard and his coworkers have applied vortex methods to a number of problems in inviscid dynamics. Leonard has given a comprehensive survey of these and related methods in [31,32].

1. Formulation and Theory in Two Dimensions.

The Euler equations in two dimensions

$$u_t + (u\cdot\nabla)u + \nabla p = 0$$

$$\nabla\cdot u = 0$$

lead to the equation for the vorticity $\omega = u_{2,1} - u_{1,2}$

$$\omega_t + (u\cdot\nabla)\omega = 0 \ , \qquad (1)$$

which shows that the vorticity is conserved along particle paths. If the vorticity $\omega(x,t)$ is known, the velocity $u(x,t)$ is determined by the convolution

$$u(x,t) = \int K(x - x')\omega(x',t)dx' \qquad (2)$$

where

$$K(x) = \frac{(-x_2,x_1)}{2\pi|x|^2} ,$$

assuming $u \to 0$ at infinity. The particle paths are of course the solutions of the ordinary differential equation

$$\frac{dx}{dt} = u(x,t) . \qquad (3)$$

The simplest form of the numerical method is obtained by combining (1), (2), (3) and discretizing. We assume that the initial vorticity $\omega_0(x)$ is smooth and zero outside a bounded set. We cover the support of ω_0 by a square grid and introduce a particle at the center ih of the ith square, where i is a pair of integers and h is the grid size. The ith particle is assigned a vorticity $\omega_i = \omega_0(ih)$. We formulate a system of ordinary differential equations for the paths $x_i(t)$ reached by these markers at later times. At each time we have available the positions $x_i(t)$ and the values $\omega(x_i(t),t) = \omega_i$, according to (1). We need to interpolate a vorticity field from which a velocity expression can be formed using (2). To do this we use "blobs" of vorticity, i.e., a combination of vortices with finite core of prescribed shape.

To define the "blob" vorticity we choose some smooth approximation φ to the delta function of total weight 1 and scale it by a parameter δ, setting $\varphi_\delta(x) = \delta^{-2}\varphi(x/\delta)$. The choice of φ and δ is discussed below. The blob approximation is then

$$\omega^{blob}(x,t) = \sum_j \varphi_\delta(x - x_j(t))\omega_j h^2 . \qquad (4)$$

(The factor of h^2 results from the fact that the flow preserves the area of the initial grid squares.) It can be thought of as a discretization of the convolution $\varphi_\delta * \omega$. It approximates the actual vorticity well provided $\delta >> h$

as both go to zero. Corresponding to (4) we have for the velocity, using (2), the expression

$$u(x,t) \sim K * \omega^{\text{blob}} = \sum_j K_\delta(x - x_j(t))\omega_j h^2 , \qquad (5)$$

where $K_\delta = K * \varphi_\delta$. This is a discretized version of (2), with K replaced by the smooth kernel K_δ. Finally we arrive at the system of ODE's for the particle positions

$$\frac{dx_i}{dt} = \sum_j K_\delta(x_i - x_j)\omega_j h^2, \qquad x_i(0) = ih . \qquad (6)$$

Once the x_i's are known the velocity field can be obtained from (5). The system (6) is of course the same as the classical equations of a system of point vortices except for the regularization of the kernel. We find in simple test problems [7] that a system of point vortices gives reasonable values for x_i, u_i for short times, but the velocities at other locations are very inaccurate. The blob system (6), in contrast, represents the velocity well independent of location.

There is in principle a serious disadvantage to this approach: with N particles, there are $O(N^2)$ operations necessary to calculate the N velocities in (6). For this reason simulations have long been done with "cloud in cell" or "particle in cell" methods, in which the velocity is calculated on a grid using a fast Poisson solver and interpolation is used (see e.g. [31]). This reduces the operation count to $O(N \log N)$. Corrections are necessary to represent the local interactions accurately, as in [30]. C. Anderson [3] has developed an improved version of this method and demonstrated with computational examples that it can be as accurate as the direct method. However, it is also possible to do the direct summation (6) rapidly by using far field approximations, e.g. as in [38]. Recently it has been shown [25] that this can be done within an arbitrary tolerance ϵ in $C_\epsilon N$ operations. These developments make large-scale simulations appear more practical.

We now discuss the choice of the smooth kernel and the convergence theory. The function φ should be fairly smooth and rapidly decreasing, have integral 1, and satisfy moment conditions

$$\int x_1^\alpha x_2^\beta \, \varphi(x_1,x_2)dx_1dx_2 = 0 \quad \text{for} \quad 0 < \alpha + \beta \leq m - 1 \qquad (7)$$

for some m. We may as well take $\varphi = \varphi(|x|)$, and in that case m is even with $m \geq 2$. For example, the choice $\varphi(r) = e^{-r^2}/\pi$ leads to $K_\delta = K\cdot(1 - e^{-r^2/\delta^2})$. In [7] the author and A. Majda developed a systematic way of generating higher order kernels from lower ones. Thus the fourth order kernel corresponding to the Gaussian is

$$K_\delta = K\cdot\{1 + (r^2/\delta^2 - 1)e^{-r^2/\delta^2}\} . \tag{8}$$

The essential aspects of the convergence of the method just described are contained in the theorem below.

Convergence Theorem. Assume $\omega_0(x)$ is smooth and has bounded support. With φ as described, take $\delta = c_0 h^q$ with $0 < q < 1$. Then for $p < \infty$ and $0 \leq t \leq T$, the error in the discrete L^p norm of the particle positions $x_i(t)$, computed from (6), is $O(\delta^m) = O(h^{mq})$. The same is true for the L^p error in a finite set for the velocity field computed from (5). Here the constants depend on T, the choice of parameters, and bounds for the initial data.

The first convergence proof of this type was given by Hald and del Prete [26] and more definitively by Hald [27]. Most subsequent treaatments have used the same outline as in Hald's work, although a recent argument of Cottet [17] has a quite different structure. The author and A. Majda [5,6] proved convergence of one method for three-dimensional flow, described below, and showed higher order convergence. A number of further improvements have been made [1,15,24,28,37]. Smoothness assumptions are not a severe restriction; Hald showed in [28] that the 2-D method converges if the vorticity is merely Hölder continuous. Convergence of the full random method for viscous flow with boundaries seems difficult, especially as the viscosity goes to zero, but there are results covering various aspects. Convergence has been shown for the random method without boundaries [34,22,33]. Hald [29] has proved convergence for a random method with vorticity generation at the boundary for a linear problem of heat transfer. Benfatto and Pulvirenti [10] have justified the splitting of viscous flow into inviscid flow and diffusion in the presence of boundaries. Cottet and Mas-Gallic [18] have shown convergence of one deterministic means of representing viscosity. Convergence proofs have also been given for a cloud-in-cell method, [16] variable density flow [2], and a finite element method version of the vortex method [4].

The convergence argument in two dimensions reduces to an estimation of the errors committed in estimating the velocity, and these are of three types. The error due to the smoothing,

$$\int [K_\delta(x - x') - K(x - x')]\omega(x',t)dx'$$

is easily seen to be $O(\delta^m)$, with the order determined by the moment conditions (7). As shown by Cottet and Raviart [15,37], the discretization error

$$\sum_j K_\delta(x - x_j)\omega_j h^2 - \int K_\delta(x - x')\omega(x',t)dx'$$

can be treated as a quadrature error in Lagrangian coordinates and bounded by $c_\ell \delta(h/\delta)^\ell$ with ℓ depending on the degree of smoothness. Their argument was simpler and more direct than the earlier versions. Their proof of the appropriate quadrature lemma was based on the Bramble-Hilbert Lemma; an even simpler proof was given by C. Greengard in [1]. The author showed in [9] that the estimate could be improved by another factor of δ by taking into account the fact that the kernel has average value zero. Finally the errors in the particle locations at previous times contribute an error to the velocity approximation of the form

$$\sum_j [K_\delta(\tilde{x}_i - \tilde{x}_j) - K_\delta(x_i - x_j)]\omega_j h^2.$$

Under mild assumptions this quantity, as a function of i, can be bounded in a discrete L^p norm by $\|\tilde{x}_i - x_i\|_{L^p}$ provided $\delta \geq c_0 h$. The essential reason is that convolution with ∇K is a bounded operator on L^p, according to the Calderon-Zygmund inequality. This point of view was introduced by Hald [27] and developed more systematically by the author and A. Majda in [5,6]. The estimation of time discretization has been treated by Anderson and Greengard [1] and by Hald [28]. The stability estimate can be thought of as a Lipschitz condition for the ODE's (6). In effect it says that the system is not "stiff". This fact is related to a special property of the incompressible Euler equations: In Lagrangian coordinates the Euler equations are like an ordinary differential equation, in the sense that the right-hand side is a bounded nonlinear operator without loss of derivatives. This point of view was developed by Arnold and others.

To assess the performance of the method we have carried out tests with a simple class of problems with exact solutions. We choose a vorticity $\omega(r)$, $r^2 = x_1^2 + x_2^2$, with $\omega = 0$ for $r > 1$. The result is a steady flow in which the particle paths are concentric circles, with shearing of the inside layers with respect to the outside. Since the angular velocity varies with the radius, the original rectangular configuration of the computational particles is greatly distorted at later time. For this reason we believe that this class of problems provides a significant test of the method. In Table 1 we give errors in the particle velocities for $\omega = (1 - r^2)^3$, $r < 1$. In this case the inside turns faster than the outside, with one inside rotation in time 4π. We have covered the disk with a square grid with $h = .1$ and introduced particles in squares with nonzero vorticity, resulting in 316 particles. The ODE's are solved with the usual fourth order Runge-Kutta method, with $\Delta t = 1$. We display the percentage L^2 error in particle velocities for $t \leq 40$, normalized by the L^2 norm of the velocity field inside the disk. Calculations are presented with $m = 4$ and several choices of δ, using the smooth kernel (8), and also one case with $m = 8$ for comparison. The trends are easily related to the theory above: For one or two rotation times the accuracy is very good, and the dominant error comes from the smoothing. (In fact, if we chose δ smaller we would have better accuracy at short times but worse at larger times.) Later on the error is considerably larger; the discretization error has grown because of the distorted geometry of the original Lagrangian configuration. Significantly, the error does not seem to grow steadily worse as time goes on. For more detailed test results, see [7]. Similar tests of circular shear layers were reported in [26,35,36].

Table 1. $\omega = (1 - r^2)^3$, **Original Method. % Error in Particle Velocities**

	$m = 4$	$m = 4$	$m = 4$	$m = 8$
time	$\delta = 2h$	$\delta = 2.5h$	$\delta = 3h$	$\delta = 4h$
0	.6	1.2	2.3	.5
8	.6	1.2	2.3	.5
16	1.2	1.4	2.3	.6
24	2.0	1.7	2.4	1.2
32	6.6	5.0	4.2	3.8
40	4.7	4.5	3.5	3.6
max	6.6	5.0	4.3	4.0

2. Improved Accuracy at Larger Times.

Although the resolution used in the above test problem is modest, it is natural to look for ways of obtaining better accuracy at larger times. Some possible approaches are (i) occasional rezoning; (ii) adding new points in stretched regions; (iii) more sophisticated quadrature rules for the velocity integral; (iv) interpolation of the vorticity using some alternative to (4). The first two have commonly been used in practice for Lagrangian methods; in fact, in [7] we found that rezoning would give much better accuracy in the test problem described above, at the expense of adding more points when the rezoning takes place. The third approach has been explored by Chiu and Nicolaides [11]. Here we will describe a simple alternative of the fourth type which is quite effective, at least for the circular shear layers.

At the current time we have available the particle locations $x_i(t)$ and the values $\omega_i = \omega(x_i(t),t)$ of the vorticity function carried along the particle paths. Our problem is to construct an interpolated vorticity field, and therefore a velocity field, using (2), from these known values. The choice of interpolating function (4) could be modified in many ways; we prefer a way which is simple and does not depend explicitly on the Lagrangian coordinates since these are quite distorted at larger times. One natural modification is to replace the coefficients ω_j in the sum (4) with coefficients γ_j to be chosen:

$$\omega(x,t) \sim \sum_j \varphi_\delta(x - x_j)\gamma_j h^2 \ . \tag{9}$$

We will choose the γ_j so that the interpolating function matches the known values at $x_i(t)$:

$$\omega_i \sim \sum_j \varphi_\delta(x_i - x_j)\gamma_j h^2 \ . \tag{10}$$

This amounts to a system of equations for the n coefficients of the form $A\gamma = \omega$. However, it may not be desirable to solve the system exactly, since the matrix $A_{ij} = \varphi_\delta(x_i - x_j)h^2$ is ill-conditioned. Regarded as an operator on functions of a continuous variable, A is a discretization of the convolution with φ_δ; according to the quadrature estimates, it is close to the identity, but it cuts out high wave numbers. Our approach is to iterate to reduce the residual $\omega - A\gamma$, using the approximation $A \sim I$. Thus we start with the initial guess $\gamma^{(0)} = \omega$ and define

$$\gamma^{(n)} = \gamma^{(n-1)} + (\omega - A\gamma^{(n-1)}) \ . \tag{11}$$

It is easy to see that the residual is then

$$\omega - A\gamma^{(n)} = (I - A)^{n+1}\omega .$$

In our experiments we fix the number of iterations n and solve ODE's as before, with (6) replaced by

$$\frac{dx_i}{dt} = \sum_j K_\delta(x_i - x_j)\gamma_j^{(n)}h^2, \quad x_i(0) = ih , \tag{12}$$

and similarly for (5). We have found that 3 or 4 iterations reduce the residual significantly but further reduction is very slow.

In Table 2 we record results for the same test problem as in Table 1. The modified method (2.4) was used with 8 iterations, but otherwise the parameters were not changed. The first line shows the particle velocity errors for times up to $t = 100$. The improvement is dramatic, with a maximum error of 1.3%. Again the errors at other locations are very similar. For comparison the second line shows the velocity error measured along a ray of radius 2, normalized by the L^2-norm of the velocity in the disk $r \leq 2$, and the third line shows the relative L^2 error in particle vorticities.

Table 2. $\omega = (1 - r^2)^3$, **8 Iterations**

% error	$t = 0$	$t < 40$	$t < 100$
particle velocity	.06	1.0	1.3
velocity, $r \leq 2$	.72	1.7	2.0
particle vorticity	.44	2.1	3.4

Although great improvement was evident using this modification, there is still a difficulty, as will be evident in the next example. In this case we take

$$\omega(r) = (1 - r)^2(1 - 2r)(1 + 4r), \quad r < 1.$$

With the same choice of parameters as before, the original method (6) leads to velocity errors of about 10% for this problem. If we use the modified method with 8 iterations, we find a velocity error inside the disk $r \leq 1$ of .7%. However, for $r \leq 2$ the velocity error, measured as before, is 5.5%. The error has been reduced inside, but not so much outside. The reason appears to be that, while (10) enforces accuracy of the interpolation inside the support of the vorticity, no control is imposed outside. One remedy is to use particles covering not only the support of the voriticity but also a border around the support. We have repeated the above experiments with particles introduced initially inside the disk $r \leq 1.2$. The number of particles (and coefficients γ_j) is increased from 316 to 448. As shown in Table 3, the velocity field for the second example is now within 1.2% of the correct value. (In the first line we have measured the velocity errors for particles only within radius 1, normalized as before.) For comparison, similar results are presented in Table 4 for the first example. From these experiments it appears that good accuracy is obtained at long time by using the modified method, provided a neighborhood of the set of nonzero vorticity is covered by computational particles.

Table 3. $\omega = (1-r)^2(1-2r)(1+4r)$ **Extra Border, 8 Interations**

% error	$t = 0$	$t < 40$	$t < 100$
particle velocity	.49	.78	1.1
velocity, $r \leq 2$	.94	1.0	1.2
particle vorticity	2.2	2.5	2.7

Table 4. $\omega = (1-r^2)^3$ **Extra Border, 8 Iterations**

% error	$t = 0$	$t < 40$	$t < 100$
particle velocity	.04	1.0	1.2
velocity, $r \leq 2$	.14	1.0	1.2
particle vorticity	.27	2.0	3.3

The method just described can be shown to converge using techniques similar to those of earlier convegence proofs. An informal statement of the convergence theorem is given below. Details of the proof will appear elsewhere.

Convergence of the Modified Method. Assume ω_0 has bounded support and is sufficiently smooth. Introduce particles on a square grid of size h covering a fixed neighborhood of the support of ω_0. Choose a core function φ of order m, and let $\delta = c_0 h^q$ with $\frac{1}{4} < q \leq \frac{1}{2}$. Compute particle positions $x_i(t)$ according to (12) with n iterations of (11). Then for fixed p and x, $1 \leq p < \infty$, $4 \leq s \leq mn$, and fixed time $0 \leq t \leq T$, the computed particle positions $x_i(t)$ and the corresponding computed velocity field converge in L^p norms with errors $O(\delta^s) = O(h^{qs})$. The assumed degree of smoothness of the underlying flow depends on s but not on n.

Other iterations could be used in place of (11) to satisfy (10). It is unclear whether a more sophisticated algebraic technique would preserve the stability of the method. In our experiments we have not found another iteration which leads to significantly better results. As pointed out by C. Anderson, the approximation (10) is in the spirit of "scattered data interpolation"; see [19] for a survey. A related optimization was used for initial blob configurations in [35]. The modification described here has recently been tested in a viscous version of the vortex method with similar results [39].

3. Convergent Methods in Three Dimension.

In three dimensional flow regions of vorticity can undergo rpaid stretching and contracting, and a numerical method must take this process into account. Unlike the 2-D case there does not seem to be a most natural version of the method. Three dimensional vortex methods have been used by Leonard et al., Rehbach, Chorin, Ashurst, and C. Greengard among others, as well as a 3-D cloud-in-cell method by Couët et al. (See Leonard's articles [31,32] for descriptions and references.) Here we will discuss two distinct versions for which convergence has been proved. All are based on the Biot-Savart Law relating the velocity $u(x,t)$ to the vorticity $\omega = \nabla \times u$,

$$u(x,t) = -\frac{1}{4\pi}\int \frac{x - x'}{|x - x'|^3} \times \omega(x',t)dx' \ ; \tag{13}$$

as in (2) we have the form $u = K * \omega$, where K is now a matrix. We may compute approximate particle paths as in (6), but the value of the vorticity must be computed as well. The evolution of vorticity is expressed in the equation

$$\frac{d\omega}{dt} + (u\cdot\nabla)\omega = (\omega\cdot\nabla)u \tag{14}$$

or alternatively by its integrated form

$$\omega(x,t) = \nabla\Phi^t(\alpha) \cdot \omega_0(\alpha), \quad x = \Phi^t(\alpha) \tag{15}$$

where $\Phi^t(\alpha)$ is the position at time t reached by a fluid particle initially located at α. Thus the vorticity is transported with the flow like a tangent vector.

Convergent vortex methods can be based on either of the expressions (14), (15). In either case we compute the locations of a representative collection of particles, introduced initially on a rectangular grid, according to the ODE's

$$\frac{dx_i}{dt} = \sum_j K_\delta(x_i - x_j)\omega_j(t)h^3, \quad x_i(0) = ih \tag{16}$$

where again $K_\delta = K * \varphi_\delta$ is a modified kernel; see [7,9] for choices of φ. In the first method we compute the ω_j's using (15), and to do this we need to approximate the Jacobian matrix $\nabla\Phi^t(\alpha)$. Since $x_j(t)$ approximates $\Phi^t(jh)$, we can approximate $\nabla\Phi^t$ by a difference operator ∇^h applied to the x_j's as functions of the Lagrangian coordinates jh, so that $\omega(x_j(t),t) \sim \nabla^h x_j \cdot \omega_{0j}$, with $\omega_{0j} = \omega_0(jh)$. Then (16) is completed by setting

$$\omega_j(t) = \nabla^h x_j(t) \cdot \omega_{0j} . \tag{17}$$

It was shown in [5,6] (see also [1]) that the computed particle paths and velocity field obtained by solving (16), (17) converge to the actual quantities under approximate conditions; the order of accuracy depends on the order of the difference operator ∇^h as well as the choice of φ. (Actually the vorticity was expressed in a more indirect way in [5]; C. Greengard [1] has pointed out that this simpler formulation is equivalent to the earlier one.) A method in a similar spirit was used by Chorin in [14].

In introducing particles initially on a rectangular grid we are thinking very generally, but in a given problem other choices may be more naturally suited to the geometry. C. Greengard [23,24] has proved convergence of a version of (16), (17) allowing a more general choice of coordinates, and used it to simulate the merger of two vortex rings. In this case the initial vortex lines are circles. Particles are introduced along the circles, and the differences in (17) amount to differences along the vortex lines. From this point of view, this method is similar to earlier treatments based on moving filaments (see Leonard [31,32]), as well as the version of Chorin [14].

The alternative method uses the ODE's (16) for the particle locations $x_i(t)$ coupled with ODE's for the vorticities $\omega_i(t)$ obtained by discretizing (14). Since the velocity field is approximated by

$$u(x,t) \sim \sum_j K_\delta(x - x_j(t))\omega_j(t)h^3$$

it is natural to use an expression for ∇u obtained by differentiating this explicitly:

$$\nabla u(x,t) \sim \sum_j \nabla K_\delta(x - x_j)\omega_j h^3 \ .$$

Substituting this in (17), we have the equations for ω_i,

$$\frac{d\omega_i}{dt} = \omega_i \cdot \sum_j \nabla K_\delta(x_i - x_j)\omega_j h^3 \ . \tag{18}$$

Then (16), (18) form a system of ODE's for the unknowns x_i, w_i. This method was introduced by C. Anderson [1] who used similar expressions in developing vortex methods for variable density [2]. Convergence of this method has been proved by the author [9] and by Cottet [17], using a very different approach.

The two methods described have somewhat different advantages. The first requires ODE's only for the positions, but depends on differences with respect to the Lagrangian coordinates, and thus might be inaccurate after the geometry becomes highly distorted. The second method probably has greater flexibility, since the Lagrangian grid is not used explicitly. However, additional ODE's are solved using more singular integrals, and there might be a loss of accuracy for this reason. For an insightful discussion of the two methods see [1]. A modified version of the first method can be formulated in analogy to that of Section 2 and has similar convergence properties. Further numerical tests are needed to assess the strengths and weaknesses of the various 3-D methods.

Acknowledgement

This work was partially supported by N.S.F. Grant No. NSF-DMS-84-08393, and also by the Institute for Mathematics and its Applications at the University of Minnesota with funds provided by the N.S.F.

References

1. C. Anderson and C. Greengard, On vortex methods, SIAM J. Numer. Anal. 22 (1985), 413-40.

2. C. Anderson, A vortex method for flows with slight density variations, J. Comp. Phys. 61 (1985), 417-32.

3. C. Anderson, A method of local corrections for computing the velocity field due to a distribution of vortex blobs, J. Comp. Phys. 62 (1986), 111-23.

4. C. Bardos, M. Bercovier, and O. Pironneau, The vortex method with finite elements, Math. Comp. 36 (1981), 119-36.

5. J. T. Beale and A. Majda, Vortex methods, I: Convergence in three dimensions, Math. Comp. 39 (1982), 1-27.

6. J. T. Beale and A. Majda, Vortex methods, II: Higher order accuracy in two and three dimensions, Math. Comp. 39 (1982), 29-52.

7. J. T. Beale and A. Majda, High order accurate vortex methods with explicit velocity kernels, J. Comput. Phys. 58 (1985), 188-208.

8. J. T. Beale and A. Majda, Vortex methods for fluid flow in two or three dimensions, Contemp. Math. 28 (1984), 221-229.

9. J. T. Beale, A convergent 3-D vortex method with grid-free stretching, Math. Comp. 46 (1986), 401-24 and S15-S20.

10. Benfatto and Pulvirenti, Convergence of Chorin-Marsden product formula in the half-plane, Commun. Math. Phys. 106 (1986), 427-58.

11. C. Chiu and R. A. Nicolaides, Convergence of a higher order vortex method for the two and three dimensional incompressible Euler equations, preprint.

12. A. J. Chorin, Numerical study of slightly viscous flow, J. Fluid Mech. 57 (1973), 785-96.

13. A. J. Chorin and J. E. Marsden, A Mathematical Introduction to Fluid Mechanics, Springer-Verlag, New York, 1979.

14. A. J. Chorin, Vortex models and boundary layer instability, SIAM J. Sci. Statist. Comput 1 (1980), 1-21.

15. G. H. Cottet, Méthodes particulaires pour l'équation d'Euler dans le plan, These de 3e cycle, Université P. et M. Curie, Paris, 1982.

16. G. H. Cottet, Convergence of a vortex in cell method for the two dimensional Euler equations, to appear in Math. Comp.

17. G. H. Cottet, On the convergence of vortex methods in two and three dimensions, preprint, 1985.

18. G. H. Cottet and S. Mas-Gallic, A particle method to solve transport-diffusion equations, Part 1: the linear case, preprint, 1985.

19. R. Franke, Scattered data interpolation: tests of some methods, Math. Comp. 38 (1982), 181-200.

20. A. F. Ghoniem and J. A. Sethian, Dynamics of turbulent structure in recirculating flow; a computational study, preprint, 1984.

21. A. F. Ghoniem and Y. Gagnon, Vortex simulation of laminar recirculating flow, preprint, 1985.

22. J. Goodman, Convergence of the random vortex method, preprint.

23. C. Greengard, Three-Dimensional Vortex Methods, Ph.D. thesis, Univ. of Calif., Berkeley, 1984.

24. C. Greengard, Convergence of the vortex filament method, Math. Comp. 47 (1986), 387-98.

25. L. Greengard and V. Rokhlin, A fast algorithm for particle simulations, preprint, 1986.

26. O. Hald and V. Del Prete, Convergence of vortex methods for Euler's equations, Math. Comp. 32 (1978), 791-809.

27. O. Hald, The convergence of vortex methods, II, SIAM J. Numer. Anal. 16 (1979), 726-55.

28. O. Hald, Convergence of vortex methods for Euler's equations III, to appear in SIAM J. Numer. Anal.

29. O. Hald, Convergence of a random method with creation of vorticity, preprint, 1984.

30. R. Hockney and J. Eastwood, Computer Simulation Using Particles, McGraw-Hill, New York, 1981.

31. A. Leonard, Vortex methods for flow simulations, J. Comput. Phys. 37 (1980), 289-335.

32. A. Leonard, Computing three-dimensional incompressible flows with vortex elements, Ann. Rev. of Fluid Mech. 17 (1985), 523-59.

33. D.-G. Long, Convergence of the random vortex method in one and two dimensions, Ph.D. thesis, Univ. of Calif., Berkeley, 1986.

34. C. Marchioro and M. Pulvirenti, Hydrodynamics in two dimensions and vortex theory, Comm. Math. Phys. 84 (1982), 483-503.

35. Y. Nakamura, A. Leonard, and P. Spalart, Vortex simulation of an inviscid shear layer, AIAA/ASME Third Joint Thermophysics, Fluids, Plasma and Heat Transfer Conference, 1982.

36. M. Perlman, On the accuracy of vortex methods, J. Comput. Phys. 59 (1985), 200-23.

37. P. A. Raviart, An Analysis of particle methods, CIME Course, Como, Italy, 1983.

38. P. R. Spalart and A. Leonard, Computation of separated flows by a vortex-tracing algorithm, AIAA 14th Fluid and Plasma Dynamics Conference, 1981.

39. J. P. Choquin and B. Lucquin, Accuracy of the deterministic particle method for Navier-Stokes equations, preprint.