FAST VORTEX METHODS

John A. Strain

Mathematics Department and
Lawrence Berkeley National Laboratory
University of California
Berkeley, California 94720

ABSTRACT

We present three fast adaptive vortex methods for the 2D Euler equations. All obtain long-time accuracy at almost optimal cost by using four tools: adaptive quadrature, free-Lagrangian formulation, the fast multipole method and a nonstandard error analysis. Our error analysis halves the differentiability required of the flow, suggests an efficient new balance of smoothing parameters, and combines naturally with fast summation schemes. Numerical experiments with our methods confirm our theoretical predictions and display excellent long-time accuracy.

INTRODUCTION

Vortex methods solve the 2D incompressible Euler equations in the vorticity formulation by discretizing the Biot-Savart law with the aid of the flow map. They have been extensively studied, widely generalized and applied to complex high-Reynolds-number flows: See (Gustafson and Sethian, 1991) for a survey.

Vortex methods involve several components; velocity evaluation, vortex motion, diffusion, boundary conditions and regridding. In this paper, we improve the speed, accuracy and robustness of the velocity evaluation. We eliminate the flow map, improve the quadrature used for the Biot-Savart law, and analyze the error in velocity evaluation in a nonstandard way, requiring less differentiability of the flow and obtaining efficient new parameter balances. We employ standard techniques for the vortex motion and consider inviscid free-space flow to eliminate diffusion and boundary conditions. Our approach combines naturally with regridding and fast multipole methods.

Lagrangian vortex methods move the nodes of a fixed quadrature rule with the computed fluid velocity, preserving the weights of the rule by incompressibility. This procedure loses accuracy when the flow becomes disorganized, motivating many regridding techniques. Even before the flow becomes disorganized, however, obtaining high-order accuracy with fixed quadrature weights requires smoothing of the singular Biot-Savart kernel. Smoothing gives high-order accuracy for short times but slows down the fast multipole method and halves the order of accuracy relative to the differentiability of the flow.

In this paper, we discuss three fast adaptive vortex methods. We briefly review the triangulated vortex method of (Russo and Strain, 1994) and the quadrature-based method of (Strain, 1996a), then present the smoothed method of (Strain, 1996b). Triangulated vortex methods are robust, accurate and efficient but limited to second-order accuracy. Quadrature-based methods compute adaptive quadratures tailored to the Biot-Savart kernel at each time step, yielding free-Lagrangian methods which maintain long-time high-order accuracy at asymptotically optimal cost. The smoothed method couples kernel smoothing with adaptive quadrature rules *not* tailored to the Biot-Savart kernel, producing long-time high-order accuracy. The asymptotic slowdown produced by kernel smoothing is almost eliminated by a careful choice of smoothing functions and parameters, based on a new error analysis of the velocity evaluation.

The structure of these methods is standard: At each time step, the smoothed velocity is evaluated once and the vortices are moved with an explicit multistep method. The velocity evaluation is nonstandard, and different for each method. We have implemented and tested all of these new methods; the error is small on standard test problems and the theoretical predictions are fully verified. More complex flows are also computed.

EQUATIONS OF MOTION

The 2D incompressible Euler equations

$$\begin{aligned} \dot{u} + uu_x + vu_y + p_x/\rho &= 0 \\ \dot{v} + uv_x + vv_y + p_y/\rho &= 0 \\ u_x + v_y &= 0, \end{aligned}$$

involve the fluid velocity $u(z,t) = (u,v)$, where $z = (x,y)$, the pressure $p(z,t)$ and the constant density ρ. Taking the 2D curl eliminates the pressure, giving the vorticity equation

$$\dot{\omega} + u\omega_x + v\omega_y = 0$$

for the vorticity $\omega = v_x - u_y$. Let $z \mapsto \Phi(z,t)$ be the flow map, defined by

$$\dot{\Phi}(z,t) = u(\Phi(z,t),t). \tag{1}$$

Then vorticity is conserved along particle paths:

$$\omega(\Phi(z,t),t) = \omega(z,0); \tag{2}$$

When ω has compact support, the velocity is given by the Biot-Savart law

$$u(z,t) = \int K(z-z')\omega(z')dx'dy' \tag{3}$$

where K is the Biot-Savart kernel

$$K(z) = \frac{z^\perp}{2\pi r^2}, \quad z^\perp = (-y,x), \quad r^2 = x^2 + y^2. \tag{4}$$

Thus we have a closed system for Φ and ω, the "free-Lagrangian" equations of motion consisting of Eq. (2) coupled with

$$\dot{\Phi}(z,t) = \int K(\Phi(z,t) - z')\omega(z',t)dx'dy'. \tag{5}$$

The Lagrangian equation of motion is derived by changing variables $z' \leftarrow \Phi(z',t)$. The Jacobian is unity because the flow is incompressible, so this gives a closed system for Φ alone:

$$\dot{\Phi}(z,t) = \int K(\Phi(z,t) - \Phi(z',t))\omega(z',0)dx'dy'. \tag{6}$$

This requires values of ω only at time $t = 0$, and is the usual starting point for vortex methods.

VORTEX METHODS

Lagrangian vortex methods now discretize Eq. (6), tracking N points $z_j(t) \approx \Phi(z_j,t)$ moving with the fluid velocity, starting at $t = 0$ from the nodes z_j of a quadrature formula with weights w_j. Suppose we use a quadrature formula

$$\int g(z)dxdy = \sum_{j=1}^{N} w_j g(z_j) + E_N(g)$$

with a qth-order error bound

$$|E_N(g)| \leq Ch^q \|g\|_q \tag{7}$$

for $g \in C^q$. Here h is the mesh size of the rule and the C^q norm is defined by

$$\|g\|_0 = \max_z |g(z)|, \qquad \|g\|_q = \|g\|_0 + \sum_{\alpha+\beta=q} \|\partial_x^\alpha \partial_y^\beta g\|_0 .$$

Applying this quadrature to the Lagrangian equation of motion (6) gives a system of N ordinary differential equations:

$$\dot{z}_i(t) = \sum_{j\neq i} w_j K(z_i(t) - z_j(t))\omega(z_j,0).$$

The quadrature error bound Eq. (7) is infinite since K is unbounded, so we replace K by the smoothed kernel

$$K_\delta(z) = \varphi_\delta * K(z) = f(r/\delta)K(z)$$

where $*$ denotes convolution,

$$\varphi_\delta(z) = \delta^{-2}\varphi(r/\delta)$$

φ is an appropriate radial "core function," and the "shape factor" f is given by

$$f(r) = 2\pi \int_0^r s\varphi(s)ds.$$

CONVERGENCE THEORY

Almost all modern vortex methods use smoothing, often with φ and the "core radius" δ chosen to give high-order convergence as the mesh size h vanishes (Chorin, 1989; Hald, 1991). This can be guaranteed by the following conditions on φ and ω:

$$\begin{aligned} \int \varphi &= 1, \\ \int x^\alpha y^\beta \varphi &= 0, \qquad 1 \leq \alpha + \beta \leq m-1, \\ \int |z|^m |\varphi| &< \infty \end{aligned} \tag{8}$$

$$\varphi \in C^L \quad \text{and} \quad \varphi(z) = 0 \quad \text{for} \quad |z| \geq 1, \tag{9}$$

$$\omega \in C^M \qquad \text{has compact support.} \tag{10}$$

High-order accuracy requires smooth solutions, so condition (10) on ω is natural. Compact support in condition (9) can be weakened, but it is important for efficiency. Given these conditions, a typical convergence theorem follows.

Theorem 1 (Anderson and Greengard, 1985) *Assume conditions (8) through (10) are satisfied with $L \geq 3$, $M \geq \max(L+1, m+2)$ and $m \geq 4$. Let $\delta = ch^a$ where $0 < a < 1$. Suppose L is large enough to satisfy*

$$L > \frac{(m-1)a}{1-a}.$$

Then the computed flow map $\Phi_{h,\delta}$ satisfies

$$\|\Phi - \Phi_{h,\delta}\|_h \leq O(h^{ma})$$

as h and δ go to zero.

Here the discrete 2-norm is given by

$$\|g\|_h = \left(h^2 \sum_i |g(z_i)|^2 \right)^{1/2}$$

where z_i are the initial vortex positions, and similar bounds hold for the computed velocity and vorticity.

This theorem allows a close to 1 and δ close to $O(h)$ only for very smooth flows, where L and M are large. For general flows, Hald (1987) shows that $\delta = O(\sqrt{h})$ is a good choice. Then $2m$ derivatives of ω guarantee only $O(h^m)$ accuracy. Later, we reduce this to $m+1$ derivatives at the cost of redefining convergence.

COST AND ACCURACY

Convergence theory must be augmented by practical considerations of cost and accuracy. Since each velocity value is a sum

$$u_{h,\delta}(z_i) = \sum_{j=1}^{N} K_\delta(z_i - z_j) w_j \omega(z_j, 0),$$

direct evaluation costs $O(N^2)$ work. This is prohibitively expensive if the flow is complex, since many vortices are required. The expense has been reduced by fast summation schemes (Anderson, 1986; Carrier, Greengard and Rokhlin, 1988; Strain, 1992) which evaluate unsmoothed sums like

$$u(z_i) = \sum_{j=1}^{N} K(z_i - z_j) w_j$$

to accuracy ϵ in about $O(N \log \epsilon)$ work, by separating local from global interactions and separating the variables. These schemes are much faster than direct evaluation for large N.

However, this does not completely resolve the difficulty. Fast methods cannot evaluate the smoothed interaction $K_\delta(z_i - z_j)$ between vortices z_i and z_j closer than δ, because $K_\delta \neq K$. Asymptotically, there are $O(N\delta^2)$ vortices in a circle of radius δ, so if $\delta = O(\sqrt{h})$ there are a total of $O(N^2\delta^2) = O(N^2 h) = O(N^{3/2})$ local interactions to be evaluated directly. Thus fast summation schemes slow down from $O(N)$ to $O(N^{3/2})$ when K is smoothed inside a radius $\delta = O(\sqrt{h})$.

Hence there is a conflict between smoothing and fast summation. If we try to make δ close to $O(h)$ to speed up fast summation, we need many derivatives of the flow for a modest order of convergence. Larger δ is more accurate for rougher flows, but hampers fast summation schemes. Our error analysis resolves this conflict by allowing another $O(\epsilon)$ in the error.

THE PERLMAN EFFECT

A completely different obstacle to accurate calculations with vortex methods is the "Perlman effect." The error bound for numerical quadrature in Eq. (7) depends on order-q derivatives of the integrand

$$g(z') = K_\delta(\Phi(z,t) - \Phi(z',t))\omega(z',0),$$

so the derivatives of the flow map will affect the error bound. The flow map moves fluid particles far apart, developing large derivatives when the flow becomes disorganized. Thus vortex methods lose high-order accuracy in long-time calculations (Perlman 1985; Beale and Majda 1985). For example, Figure 1 plots the number of correct bits in the computed velocity of a standard test case for a fourth-order vortex method. Fourth-order accuracy—evidenced by the gain of one tick mark per line in the figure—is attained only during a very short initial time period.

The Perlman effect has motivated much research on regridding, the idea being to avoid large derivatives of the flow map by restarting before the flow becomes disorganized (Nordmark, 1991). Similarly, Beale (1988) has developed an iterative reweighting scheme. The Perlamn effect also motivated the free-Lagrangian vortex methods reviewed here, which eliminate the flow map from the Biot-Savart integral. Thus

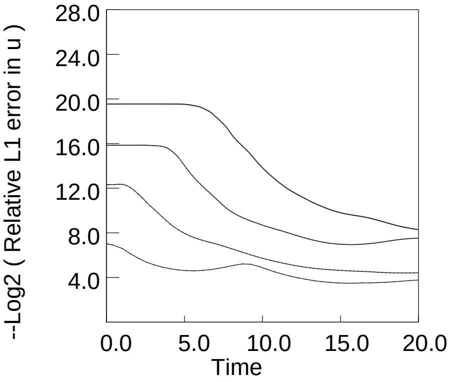

Figure 1: Correct bits in u for the Lagrangian vortex method.

$$\dot{\Phi} = \int K(\Phi - z')\omega(z',t)dx'dy',$$

replaces the Lagrangian equation of motion Eq. (6). Since ω values are known only at the moving points $z_j(t)$, each velocity evaluation requires adaptive quadratures with new weights adapted to the current vortex positions. We now present three such methods.

TRIANGULATED METHODS

Triangulated vortex methods evolve points $z_j(t)$ by

$$\dot{z}_j(t) = \int K(z_j(t) - z')\omega_h(z',t)dx'dy', \qquad (11)$$

where ω_h is a piecewise linear interpolant to the vorticity values

$$\omega_h(z_j(t),t) = \omega_h(z_j,0) = \omega(z_j,0)$$

and the nodes $z_j(t)$ form the vertices of a triangulation of $\mathbf{R}^2$.

Given any piecewise linear function ω_h on a triangulation of $\mathbf{R}^2$, one can evaluate the corresponding velocity u_h exactly, with results depending very strongly on the triangulation. In (Chacon Rebollo and Hou, 1990), this observation was combined with a fixed triangulation carried by the flow. While convergent, the resulting scheme costs $O(N^2)$ work per time step with a large constant, and loses accuracy very quickly because the triangulation degenerates.

We developed practical triangulated vortex methods in (Russo and Strain, 1994); a fast summation scheme brought the cost down to $O(N^{4/3})$ and a fast Delaunay triangulation scheme gave excellent long-time accuracy. An adaptive initial triangulation technique made the method robust enough to compute even discontinuous patches of vorticity. Figure 2 plots errors for the test case used in Fig. 1. The error displays no Perlman effect; second-order accuracy (one tick per line) is maintained uniformly in time. The triangulated approach is now being applied to flows in three dimensions with viscosity and boundaries (Huyer and Grant, 1994). Difficulties obtaining higher-order accuracy motivated the next approach.

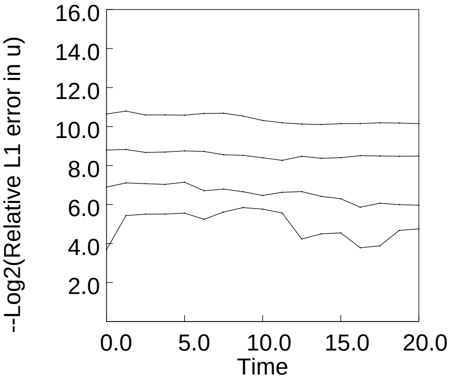

Figure 2: Correct bits in u in 1-norm, computed by the triangulated vortex method.

QUADRATURE-BASED METHODS

Higher-order free-Lagrangian methods were developed in (Strain, 1996a). They use time-dependent quadrature weights $w_{ij}(t)$ giving high-order accuracy in the Biot-Savart law:

$$\begin{aligned} u(z_i,t) &= \int K(z_i - z')\omega(z',t)dx'dy' \\ &\approx \sum_{j=1}^{N} w_{ij}(t)K(z_i - z_j)\omega(z_j,t). \end{aligned}$$

For example, high-order product integration weights make smoothing unnecessary, but the i-dependence of $w_{ij}(t)$ precludes fast summation methods. Thus we construct weights with the "locally-corrected property" that $w_{ij} = w_j$ for almost all j, where w_j are the weights of some "smooth" quadrature rule with points z_j. Such rules can be built and the velocity evaluated in $O(N \log^2 N)$ work. The price for efficiency is a redefinition of convergence. The error bound for these quadratures is $O(\epsilon + h^q)$, where ϵ is an arbitrary user-specified error tolerance and the constant in the $O(N)$ cost depends weakly on ϵ. Thus one gets order-q convergence only down to $O(\epsilon)$. This is sufficient for three reasons: computer arithmetic has finite precision, practical computations can afford only rather low accuracy, and fast summation methods introduce an $O(\epsilon)$ error as well. High-order accuracy can be maintained for long times, though these rules are somewhat expensive to implement.

A FAST ADAPTIVE METHOD

We now describe a high-order fast adaptive vortex method which aims to avoid obstacles both to speed and to accuracy. The key ingredients are

- A free-Lagrangian formulation to avoid the Perlman effect.
- Adaptive quadrature rules *not* tailored to the Biot-Savart kernel.
- New error bounds requiring fewer derivatives of the vorticity and leading to an efficient new smoothing strategy.

These ingredients combine to give a method with almost optimal efficiency and long-time high-order accuracy, without excessive differentiability requirements on the flow.

OVERVIEW

We begin with quadrature. Given N nodes $z_j \in \mathbf{R}^2$, we compute the weights of a quadrature rule with order-q accuracy on C^q functions if the nodes are well distributed. We partition the nodes into rectangular cells and build order-q rules on each cell. The union of these rules is globally accurate of order q (Strain, 1995). After quadrature, we discuss smoothing. We state a standard smoothing error bound and construct a family of arbitrary-order accurate core functions. Next we present a new error analysis which leads to an efficient new balance between quadrature and smoothing. We conclude with some numerical results.

A TREE STRUCTURE

Let $B = [a,b] \times [c,d]$ be a rectangle containing the nodes z_j. Our quadratures partition B into rectangular cells B_i, each containing enough nodes to construct an order-q quadrature. There are $q(q+1)/2$ monomials $x^\alpha y^\beta$ of degree $\alpha + \beta \le q-1$, so we will need at least $p \ge q(q+1)/2$ nodes per cell. Thus we build a tree structure to partition B into cells containing p or $p+1$ nodes each.

Let $B = B_1$ be the level-0 root of the tree. Divide B_1 in half along its longest edge, with the dividing plane located so that each half of B_1 contains either $\lfloor N/2 \rfloor$ or $\lfloor N/2 \rfloor + 1$ nodes. This gives the level-1 cells B_2 and B_3. Recursively, split B_2 and B_3 along their longest edges to get B_4 through B_7, each containing $\lfloor N/4 \rfloor$ or $\lfloor N/4 \rfloor + 1$ nodes z_j. Repeat this procedure L times to get $M = 2^L$ cells B_i on the finest level L, numbered from $i = M$ to $i = 2M-1$, each containing $p = \lfloor N/M \rfloor$ or $p+1$ nodes z_j. The union of all the cells on any given level is B. The tree structure is stored by listing the boundaries of each cell $B_i = [a_i,b_i] \times [c_i,d_i]$ from $i = 1$ to $i = 2M-1$, a total of $4 \cdot 2M$ numbers, and indexing the nodes into a list so that the nodes $z_j \in B_i$ are given by $j = j(s)$ for $s = b(i), \ldots, e(i)$ and three integer functions j, b and e. This can be done in $O(N \log N)$, but the simplest method requires sorting each cell before each subdivision, giving a total cost $O(N \log^2 N)$ for the tree construction with an $O(N \log N)$ sorting method such as Heapsort.

QUADRATURE RULES

We now construct qth-order quadrature rules on B with N quadrature nodes z_j given. Assume $N \ge m := q(q+1)/2$, and choose $L \ge 0$ with $p := \lfloor N/2^L \rfloor \ge m$. The tree structure divides B into $M = 2^L$ rectangular cells B_i, each containing either p or $p+1$ nodes z_j. On each B_i, we construct local weights W_j^i for

$z_j \in B_i$ by solving the following system of m linear equations in at least p unknowns:

$$\sum_{z_j \in B_i} P_\alpha(x_j) P_\beta(y_j) W_j^i = \int_{B_i} P_\alpha(x) P_\beta(y) dx dy$$

for $0 \leq \alpha + \beta \leq q - 1$. Here $P_\alpha(x) P_\beta(y)$ are the two-variable Legendre polynomials on B_i. Since $p \geq m$, this system of m equations in at least p unknowns generically has solutions. We compute the solution W_j^i of least 2-norm, using a complete orthogonal factorization from LAPACK (Anderson et al.,1992). The weights of the rule W are then defined to be $W_j = W_j^i$ where $z_j \in B_i$. This rule integrates all polynomials of degree less than q exactly over all level-L cells B_i. This property implies order-q accuracy:

Theorem 2 (Strain, 1995) *Suppose $B = \cup_{i=1}^M B_i$ and W integrates $x^\alpha y^\beta$ exactly over each cell B_i for $0 \leq \alpha + \beta \leq q - 1$. Then for any C^q function g on B, the quadrature error*

$$E = \int_B g(z) dx dy - \sum_{j=1}^N W_j g(z_j)$$

satisfies the bound

$$|E| \leq \Omega |B| \frac{h^q}{q!} \|g\|_q$$

where h is the longest cell edge, $\Omega = 1 + \frac{1}{|B|} \sum_{j=1}^N |W_j|$ and $|B| = (b - a)(d - c)$ is the area of B.

The condition number Ω cannot be bounded a priori for arbitrary points, but we can easily compute it a posteriori, yielding an excellent diagnostic for the quality of the rule.

SMOOTHING

The following theorem is a standard error bound for smoothing with a core function satisfying moment conditions of order m:

Theorem 3 (Raviart, 1985) *Assume the compactly supported core function φ satisfies the moment conditions*

$$\int \varphi = 1, \qquad \int x^\alpha y^\beta \varphi = 0, \quad 1 \leq \alpha + \beta \leq m - 1,$$

$$M = \frac{1}{m!} \int |z|^m |\varphi| < \infty.$$

Suppose u belongs to the Sobolev space $W^{m,p}$ of functions with m derivatives in L^p, where $1 \leq p \leq \infty$. Then

$$\|\varphi_\delta * u - u\|_{L^p} \leq M \delta^m \sum_{\alpha+\beta=m} \|\partial_x^\alpha \partial_y^\beta u\|_{L^p}.$$

Suppose φ is a continuous radial function and write $\varphi(z) = \varphi(r)$ where $r^2 = |z|^2 = x^2 + y^2$. Then $\int x^\alpha y^\beta \varphi(z) dx dy = 0$ if α or β is odd, so the moment conditions simplify to

$$\int_0^1 \varphi(r) r dr = 1/2\pi, \qquad \int_0^1 \varphi(r) r^{2j+1} dr = 0, \quad j = 1, \ldots, n$$

where $m = 2n + 2$ is even. A standard calculation gives

$$K_\delta(z) = \varphi_\delta * K(z) = f\left(\frac{r}{\delta}\right) K(z)$$

where the "shape factor" f is given by

$$f(r) = 2\pi \int_0^r \varphi(s) s ds.$$

Since $\varphi(r) = 0$ for $r > 1$, we have $f(r) = 1$ for $r > 1$ so $K_\delta(z)$ is identical to $K(z)$ for $r > \delta$. This facilitates the application of fast summation methods.

A convenient ansatz for the shape factor is

$$f(r) = \varrho^p \left[a_d \varrho^d + \cdots + a_0 \right] + 1 \tag{12}$$

where $\varrho = (1 - r^2)_+ = \max(0, 1 - r^2)$ and $\varphi(r) = \frac{1}{2\pi r} f'(r)$. The $d + 1$ coefficients a_i must be chosen so that φ satisfies $n + 1$ moment conditions, so we cannot expect a solution unless $d \geq n$. If $d > n$, the linear system of moment conditions is underdetermined, and we use a complete orthogonal factorization routine to find the solution with smallest 2-norm. Given the coefficients a_i, we have

$$K_\delta(z) = \frac{z^\perp}{2\pi r^2} \left[(1 - r^2/\delta^2)_+^p (a_d (1 - r^2/\delta^2)_+^d + \cdots + a_0) + 1 \right],$$

where $z^\perp = (-y, x)$. Since $f(0) = 0$, we can reduce roundoff problems for small r by finding a polynomial g such that

$$f(r) = r^2 g(\varrho) = r^2 \left[b_{p+d-1} \varrho^{p+d-1} + \cdots + b_0 \right].$$

In terms of g, we have a convenient formula

$$K_\delta(z) = \frac{z^\perp}{2\pi \max(r^2, \delta^2)} g((1 - r^2/\delta^2)_+).$$

Figure 3 shows several of the shape factors f; the increasing oscillation as n increases follows naturally from the vanishing of more moments.

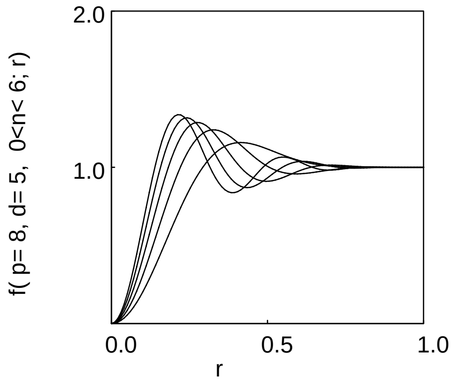

Figure 3: Piecewise polynomial shape factors f.

TIME STEPPING

Since the Euler equations are not stiff and we are constructing high-order vortex methods, we discretize time with explicit s-step Adams methods. These methods require an accurate procedure for computing the s starting values. Suppose we use an explicit s-step Adams method with a fixed time step Δ_f. We begin with a tiny time step $\Delta_i << \Delta_f$ and 1-step Adams, giving error $O(\Delta_i^2)$. Since our final method is order-s accurate, we should choose $\Delta_i = O(\Delta_f^{s/2})$. We now increase the order of the Adams method by 1 at each step until order s is reached, simultaneously increasing Δ_i by a factor $R \leq 2$ until Δ_f is reached. The final non-equidistant step is adjusted to land precisely at $t = \Delta_f$.

BALANCE OF ERROR

We now balance the errors due to smoothing and quadrature. The error in velocity evaluation splits naturally into two parts

$$\begin{aligned} E &= |u(z) - \sum_{j=1}^{N} w_j K_\delta(z - z_j)\omega(z_j)| \\ &\leq |K * \omega(z) - K_\delta * \omega(z)| \\ &\quad + |K_\delta * \omega(z) - \sum_{j=1}^{N} w_j K_\delta(z - z_j)\omega(z_j)| \\ &= E_\delta + E_{N,\delta}. \end{aligned}$$

The first term is the smoothing error, which satisfies

$$E_\delta \leq M\delta^m \|u\|_m$$

if φ satisfies moment conditions of order m and $u \in C^m$. The second term is the quadrature error, which satisfies

$$E_{N,\delta} \leq \Omega\,|B|\,\frac{h^q}{q!}\,\|g\|_q$$

for each fixed z. Here $g(z') = K_\delta(z - z')\omega(z')$, so by a standard inequality for the C^q norm of a product

$$\|g\|_q \leq C(\|K_\delta\|_q\|\omega\|_0 + \|K_\delta\|_0\|\omega\|_q).$$

Scaling gives

$$K_\delta(z) = \delta^{-1}\int \varphi\left(\frac{z}{\delta} - z'\right)\frac{z'^{\perp}}{2\pi|z'|^2}dx'dy',$$

so there is some constant C depending only on φ, such that

$$\|\partial_x^\alpha \partial_y^\beta K_\delta\|_0 \leq C\delta^{\alpha+\beta-1}$$

if $\varphi \in C^{\alpha+\beta}$. Thus if $\varphi \in C^q$, we have

$$\|g\|_q \leq \frac{1}{\delta}C(\delta^{-q}\|\omega\|_0 + \|\omega\|_q)$$

so the error in velocity evaluation satisfies

$$E \leq C(\delta^m\|u\|_m + \delta^{-1}\left(\frac{h}{\delta}\right)^q\|\omega\|_0 + \delta^{-1}h^q\|\omega\|_q).$$

where q is the order of quadrature and $\varphi \in C^q$ satisfies moment conditions of order m.

We choose δ as a function of h to make

$$\delta^{-1}\left(\frac{h}{\delta}\right)^q \leq \epsilon,$$

where ϵ is a user-specified error tolerance, fixed as h vanishes. This implies

$$\delta = O(\epsilon^{-1/(q+1)}h^{q/(q+1)}) = O(h^a), \qquad a = 1 - \frac{1}{q+1},$$

and our error bound becomes

$$E \leq C(\epsilon\|\omega\|_0 + h^{\frac{mq}{q+1}}\|u\|_m + h^{\frac{q^2}{q+1}}\|\omega\|_q).$$

The choice $m = q$ balances the two remaining terms, so

$$E \leq C\left[\epsilon\|\omega\|_0 + h^k(\|\omega\|_q + \|u\|_q)\right] = O(\epsilon + h^k)$$

where $k = q^2/(q+1) = q - 1 + \frac{1}{q+1} > q - 1$. For quadrature of orders $q = 2, 4, 6, 8, 10$, the exponent a in $\delta = O(h^a)$ is 0.66, 0.80, 0.86, 0.89, 0.91 respectively, with order of accuracy k equal to 1.33, 3.20, 5.14, 7.11, 9.09 rapidly approaching $q - 1$ from above as q increases. Thus δ is very close to $O(h)$ for methods of high order k, with only q derivatives of ω required. This allows us to use fast summation methods with excellent efficiency: the fast multipole method with this δ costs $O(N^b)$ with $b = 1 + \frac{1}{q+1} = 1.33, 1.20, 1.14, 1.11, 1.09$, very close to 1.

We combine this order-k velocity evaluation with an Adams method of order $s = q > k$, because the first-order Euler equations imply that the velocity should have the same order of differentiability in time as in space, with particle positions one order smoother by the flow map equation Eq. (1). An $O(\epsilon + h^k)$ error in the velocity u at each time step does not accumulate in the multistep solution of

$$\dot{\Phi}(z,t) = u(\Phi(z,t),t)$$

so we expect to obtain an error

$$O(\epsilon + \Delta_f^s + h^k)\|\omega\|_q$$

in Φ as h and Δ_f vanish. This would imply similar estimates for the velocity and vorticity by standard arguments.

RESULTS AND DISCUSSION

We implemented the algorithms described above and studied the performance of the fast adaptive method. First, we measured the accuracy and efficiency of the velocity evaluation scheme in isolation. Then we measured the error in long-time calculations with the full method. Finally, we studied the interaction of several smooth patches of vorticity.

We studied the accuracy of the velocity evaluation of orders $k = 1.33$, 3.20, 5.14 and 7.11 corresponding to $m = q = 2$, 4, 6, and 8, using the Perlman vorticity (Perlman, 1985)

$$\omega_P(z) = \left(\max(0, 1 - r^2)\right)^P$$

where $P = 10$. The vorticity ω_P is a C^{P-1} function on $\mathbf{R}^2$, while the corresponding velocity fields are C^P:

$$u(z) = (1 - \omega_{P+1}(z))\frac{z^\perp}{(2P+2)r^2}.$$

This is a stationary radial solution of the Euler equations with shear and a popular test case for vortex methods.

We tested our method with a random initial grid. Given N and n with $n^2 \leq N$, the grid has n^2 vortices uniformly distributed over a rectangle. and the remaining $N - n^2$ vortices distributed in regions where the vorticity is large, providing some degree of random adaptivity.

We generated $N = 500, 1000, 2000, \ldots, 64000$ vortices in such a grid with $n^2 \approx N/10$ and evaluated the velocity at each of the vortices, using core functions and quadratures of orders $m = q = 2, 4, 6, 8$. The number of correct bits

$$B_l = \max\left(0, -\log_2\left[\frac{\|u - u_{h,\delta}\|_l}{\|u\|_l}\right]\right)$$

in the computed velocity $u_{h,\delta}$ in L^1 and L^∞ norms, the CPU times T (in seconds on a Sparc-2 workstation) and other statistics are reported in Table 1. The velocity evaluation produces error $O(\epsilon + N^{-k/2})$ with $k/2 = 0.67$, 1.60, 2.57 and 3.55 in $O(N^b \log \epsilon)$ CPU time with $b = 1.33$, 1.20, 1.14 and 1.11 and a constant of proportionality depending very weakly on the order q. Note that when N doubles, the average cell size h decreases by a factor $\sqrt{2}$, so we expect to gain $k/2$ bits per line in each table until $O(\epsilon)$ is reached.

For first-order methods, the $O(N^{-2/3})$ errors dominate so the $O(\epsilon)$ limit on accuracy never appears. For higher-order methods, we get higher-order convergence in the region where the smoothed kernel is resolved. After the $O(\epsilon)$ limit is reached, convergence continues slowly.

We also tested the long-time accuracy of the method on several Perlman-type test cases for $0 \leq t \leq 20$, a final time at which the fastest fluid particles have completed 1.6 revolutions while the slowest have completed only 0.2. This strong shear is usually considered a severe test for a vortex method. We started with an adaptive random grid with $n^2 \approx 0.8N$, and used core functions, quadratures and Adams methods of orders $m = q = s = 2$, 4 and 6, yielding adaptive vortex methods of orders $k = 1.33$, 3.20 and 5.14. We tested each method on a Perlman patch of minimal smoothness, with $P = q + 1 = 3$, 5 and 7. In particular, the errors at different orders are unrelated. The correct bits in L^1 in the velocity are plotted in Fig. 4. The plots are individually scaled and ticked in such a way that the number of correct bits should increase by half a tick mark at each line. These results clearly confirm the long-time high-order accuracy of the method; they do not show the loss of accuracy observed in Lagrangian vortex methods (for example in Fig. 1). The errors are highly oscillatory on a small scale, because a new quadrature rule is built from scratch at each step.

As a more complex example, we used the order-3.20 method to compute 20 interacting smooth patches of vorticity. Thus the initial vorticity is given by

$$\omega(z, 0) = \sum_{j=1}^{Q} \Omega_j (1 - |z - z_j|^2)^P$$

where $Q = 20$, $P = 5$ and z_j and Ω_j are random. Some sample vorticity contours are shown in Fig. 5.

$m = q = 2,\ p = 4,\ d = 1,\ k = 1.33$					
N	h	δ	B_1	B_∞	T
500	0.497	0.631	1.95	1.42	4.83
1000	0.328	0.479	2.48	2.03	13.8
2000	0.205	0.351	3.28	2.79	43.6
4000	0.142	0.275	3.91	3.41	142.7
8000	0.089	0.203	4.74	4.26	336.2
16000	0.064	0.163	5.38	4.88	1155
32000	0.039	0.118	6.29	5.79	2051
64000	0.028	0.095	6.93	6.41	6493

$m = q = 4,\ p = 6,\ d = 2,\ k = 3.20$					
N	h	δ	B_1	B_∞	T
500	1.300	1.481	1.02	0.52	6.87
1000	0.807	1.011	2.26	1.79	22.2
2000	0.443	0.625	4.49	3.71	75
4000	0.300	0.457	6.00	5.21	209
8000	0.180	0.305	8.16	7.22	632
16000	0.128	0.232	9.71	9.00	1485
32000	0.078	0.156	11.9	10.3	4498
64000	0.057	0.121	13.3	12.0	8111

$m = q = 6,\ p = 8,\ d = 3,\ k = 5.14$					
N	h	δ	B_1	B_∞	T
500	1.760	1.948	0.0	0.0	8.66
1000	1.170	1.374	0.31	0.0	26.9
2000	0.721	0.905	4.49	2.67	90.6
4000	0.386	0.529	8.21	5.51	281
8000	0.263	0.381	10.2	7.18	774
16000	0.158	0.245	12.0	6.74	1574
32000	0.114	0.185	14.0	9.30	4988
64000	0.068	0.118	15.0	10.1	7634

$m = q = 8,\ p = 10,\ d = 4,\ k = 7.11$					
N	h	δ	B_1	B_∞	T
500	1.810	2.033	1.34	0.48	9.22
1000	1.690	1.912	0.0	0.0	34.9
2000	1.100	1.304	3.34	2.23	111
4000	0.677	0.848	7.79	6.29	358
8000	0.362	0.486	10.4	6.89	960
16000	0.247	0.346	11.9	8.00	2923
32000	0.146	0.217	12.9	8.15	5828
64000	0.106	0.162	15.1	9.99	14650

Table 1: Velocity evaluation results for a Perlman-type vorticity field ω_{10} with N adaptive random points: Correct bits B_1 and B_∞ in u, CPU times T, cell size h and core radius δ. Here q is the quadrature order, m is the moment order, $p - 2$ the order of smoothness and d the degree of the core function. The final results have order k.

ACKNOWLEDGMENTS

This research was supported by a NSF Young Investigator Award, Air Force Office of Scientific Research Grant FDF49620-93-1-0053, and the Applied Mathematical Sciences Subprogram of the Office of Energy Research, U.S. Department of Energy under Contract DE-AC03-76SF00098.

REFERENCES

C. R. Anderson, 1986, "A method of local corrections for computing the velocity field due to a collection of vortex blobs," *Journal of Computational Physics*, Vol. 62, pp. 111–127.

C. Anderson and C. Greengard, 1985, "On vortex methods," *SIAM Journal of Mathematical Analysis*, Vol. 22, pp. 413–440.

E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen, 1992, *LAPACK Users' Guide*, SIAM, Philadelphia.

J. T. Beale, 1985, "On the accuracy of vortex methods at large times," in B. Engquist, M. Luskin and A. Majda, editors, *Computational fluid dynamics and reacting gas flow*, vol. 12 of *IMA Volumes in Mathematics and Applications*, Springer-Verlag.

J. T. Beale and A. Majda, 1985, "High order accurate vortex methods with explicit velocity kernels," *Journal of Computational Physics*, Vol. 58, pp. 188–208.

J. Carrier, L. Greengard, and V. Rokhlin, 1988, "A fast adaptive multipole method for particle simulations," *SIAM Journal of Scientific and Statistical Computing*, Vol. 9, pp. 669–686.

A. J. Chorin, 1989, *ComputationalFluid Mechanics: Selected Papers*, Academic Press.

T. Chacon Rebollo and T. Y. Hou, 1990, "A Lagrangian finite element method for the 2-D Euler equations," *Communications in Pure and Applied Mathematics*, Vol. XLIII, pp.735–767.

K. E. Gustafson and J. A. Sethian, editors, 1991, *Vortex methods and vortex motion*, SIAM, Philadelphia.

O. H. Hald, 1987, "Convergence of vortex methods for Euler's equations III," *SIAM Journal of Numerical Analysis*, Vol. 24, pp. 538–582.

O. H. Hald, 1991, "Convergence of vortex methods," in Gustafson and Sethian (1991), pp. 33–58.

S. A. Huyer and J. R. Grant, 1994, "Incorporation of boundaries for 2D triangular vorticity element methods," Technical Report, NUWC.

H. O. Nordmark, 1991, "Rezoning for higher-order vortex methods," *Journal of Computational Physics*, Vol. 97, pp. 366–387.

M. Perlman, 1985, "On the accuracy of vortex methods," *Journal of Computational Physics*, Vol. 59, pp.200–223.

P. A. Raviart, 1985, "An analysis of particle methods," in F. Brezzi, editor, *Numerical methods in fluid dynamics (Lecture notes in mathematics; 1127) Fondazione C.I.M.E., Firenze*, Springer-Verlag, 1985.

G. Russo and J. A. Strain, 1994, "Fast triangulated vortex methods for the 2D Euler equations," *Journal of Computational Physics*, Vol. 111, pp. 291–323.

J. A. Strain, 1992, "Fast potential theory II: Layer potentials and discrete sums," *Journal of Computational Physics*, Vol. 99, pp. 251–270.

J. A. Strain, 1995, "Locally-corrected multidimensional quadrature rules for singular functions," *SIAM Journal of Scientific and Statistical Computing*, Vol. 16, pp. 1–26.

J. A. Strain, 1996a, "2D vortex methods and singular quadrature rules," *Journal of Computational Physics*, Vol. 124, pp. 1–23.

J. A. Strain, 1996b, "Fast Adaptive 2D Vortex Methods," *Journal of Computational Physics*, 32 pp., submitted.

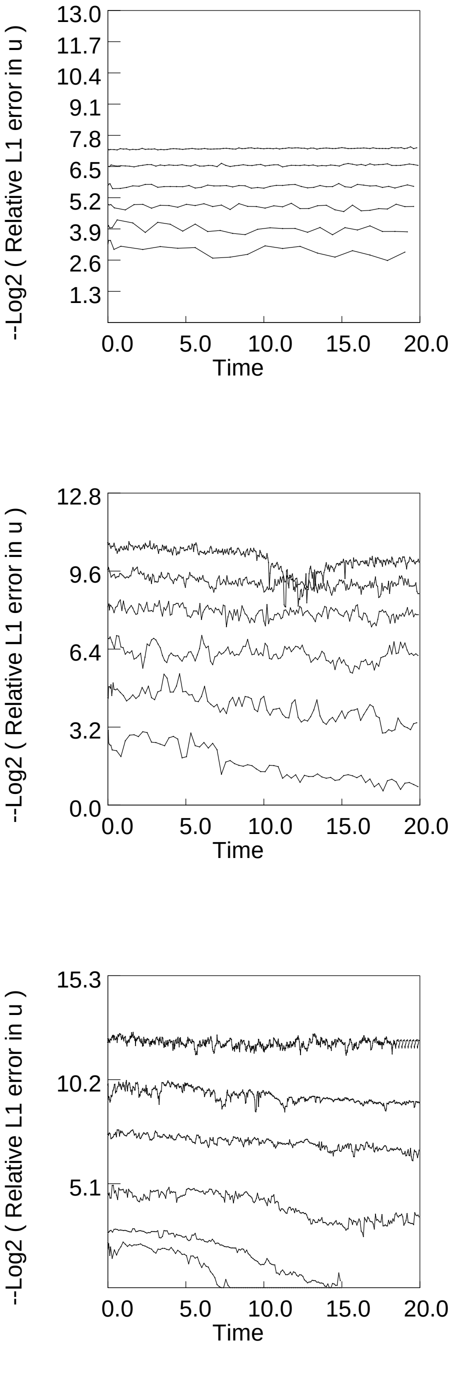

Figure 4: Correct bits in u for adaptive vortex methods of orders 1.33, 3.20 and 5.14.

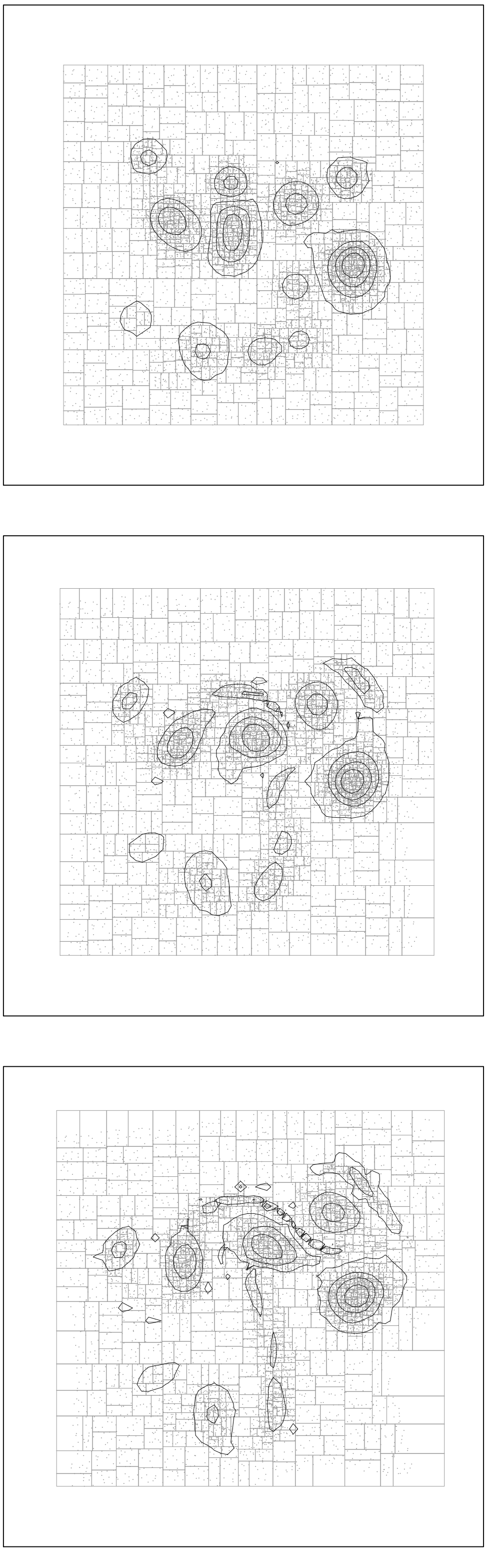

Figure 5: Vorticity contours for 20 Perlman patches ω_5 at $t = 0$, 8, and 16.