JOURNAL OF COMPUTATIONAL PHYSICS 77, 240–266 (1988)

Contour Surgery: A Topological Reconnection Scheme for Extended Integrations Using Contour Dynamics

DAVID G. DRITSCHEL

Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Silver Street, Cambridge CB3 9EW, United Kingdom

Received August 25, 1986; revised September 3, 1987

A numerical algorithm is described which, it is believed, can accurately model the dynamics of a two-dimensional, inviscid, incompressible fluid with unparalled spatial resolution. The fluid is assumed, however, to be divided into regions of *uniform* vorticity, conservation of vorticity ensuring that this remains true for all time. Like *contour dynamics*, the algorithm is concerned with following the evolution of the boundaries of vorticity discontinuity (contours). Unlike contour dynamics, the algorithm *automatically* removes vorticity features smaller than a predefined scale. For example, two contours enclosing the same uniform vorticity merge into one if they are close enough together. Also, the curvature along a contour is not allowed to exceed the inverse of the cutoff scale. At present, calculations with *contour surgery* resolve fluid motions extending over four to five orders of magnitude of scales (13 to 20 octaves). Such high-resolution pictures of two-dimensional vortex dynamics have been facilitated by and indeed depend critically upon a *nonlocal* adaptive node adjustment scheme, and a variety of tests quantify the accuracy of the technique.

1. INTRODUCTION

Presented below is a new and robust numerical technique for studying inviscid, incompressible, two-dimensional flow. The technique extends *contour dynamics* (CD) [2, 3, 17], an algorithm designed for a *piecewise-constant* vorticity distribution, by truncating the accessible range of spatial scales, in effect, introducing a maximum resolution. This approximation permits greatly extended integrations using a contour method at spatial resolutions far surpassing standard (continuous vorticity) methods.

The evolution of a piecewise-constant vorticity distribution depends *only* on the boundaries of vorticity discontinuity or *contours*, hence the name "contour dynamics." The purpose of a CD algorithm is to accurately track the distortions of the contours with finite spatial and temporal resolution. Calculations with CD, however, inevitably develop spatial structure of exponentially increasing complexity requiring correspondingly increasing spatial resolution. As the computer time per time step is proportional to the *square* of the spatial resolution, such calculations quickly become unaffordable.

The present paper introduces an extension to CD, called "surgery," which overcomes the difficulties just noted. In the next section, the algorithm is described in

0021-9991/88 $3.00

detail starting from the contour-dynamical core and proceeding to the surgical extension. New methods of resolution distribution and velocity determination are shown to significantly improve the accuracy of the CD part of the algorithm. The third section tests the entire *contour surgery* (CS) algorithm with direct numerical integrations. Parameters in the algorithm controlling surgery, spatial, and temporal resolution are varied for a single flow, and in one instance, a calculation is reversed to attempt to reproduce its initial conditions. Several additional calculations explore problems that have up to now been examined only by continuous vorticity techniques or pose great difficulties for such techniques. The final section outlines the present applications of the algorithm and extensions to geometries of a geophysical nature [6].

2. The Numerical Algorithm

In an unbounded, inviscid, incompressible fluid, the motion of a fluid particle depends only on the instantaneous vorticity distribution. Contour dynamics furthermore assumes that the vorticity distribution is *piecewise-constant*, conservation of vorticity ensuring that this remains true for all time. In this case, the motion of a fluid particle depends only on the instantaneous positions of the *contours* or boundaries of vorticity discontiuity and is calculated by the following sum of contour integrals (see Fig. 1 for definitions and [2, 3, 17, or 4] for a derivation):

$$\frac{d\mathbf{x}}{dt} = \mathbf{u}(\mathbf{x}) = -\frac{1}{2\pi} \sum_k \tilde{\omega}_k \oint_{C_k} \log |\mathbf{x} - \mathbf{x}_k| \, d\mathbf{x}_k. \tag{1}$$

Since fluid particles cannot cross the contours, the contours are defined by the same set of particles for all time, and this set also satisfies Eq. (1).

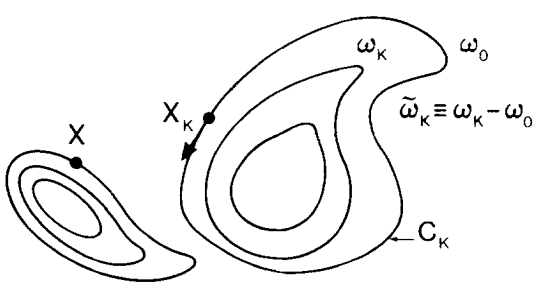

FIG. 1. The calculation of the velocity field on contours. The evaluation point is $\mathbf{x}$ and the contour integrals in Eq. (1) are taken around all contours C_k of vorticity discontinuity (with k ranging over the number of contours). $\mathbf{x}_k$ is a point on C_k where the vorticity jumps by $\tilde{\omega}_k$ crossing C_k inwards.

Contour Representation

In practice, each contour is approximated by a finite number of *nodes* connected together by interpolation functions [15, 14, 18, 4]. As the contours deform under their self-generated velocity field, additional nodes are often inserted to maintain adequate resolution, particularly in regions of developing curvature [18]. Nodes may also be removed from regions where the curvature is weakening. As computer time per time step is proportional to the square of the number of nodes, the calculation can get quite costly when a contour becomes severely deformed. It is therefore essential to redistribute the nodes at each time step in such a way as to resolve as great a range of scales with the least number of nodes for as long as possible. An approximate solution to this optimization problem is described next.

The distribution of nodes is controlled by a node density function ρ whose contour integral with respect to arc length gives the total number of nodes on a given contour. To resolve a wide range of scales, it is clear that ρ must increase with local curvature. In fact, a node density *proportional* to curvature would resolve features on all scales equally well. But it is not enough that ρ be dependent only on local curvature. Experience has shown that ρ must *also* increase with increasing velocity variation. If the node spacing exceeds or is of the same order as the scale of the velocity variation, the contour cannot react properly. This idea is implemented by making ρ a *nonlocal* function of curvature, and the following expression for the average node density between nodes j and $j+1$ emerged from a great deal of trial and error:

$$\rho_j = \frac{\hat{\kappa}_j}{1+\delta\hat{\kappa}_j/2^{1/2}} \tag{2a}$$

$$\hat{\kappa}_j = \tfrac{1}{2}(\tilde{\kappa}_j + \tilde{\kappa}_{j+1}) \tag{2b}$$

$$\tilde{\kappa}_j = (\mu L)^{-1}\,(\bar{\kappa}_j L)^a + 2^{1/2}\bar{\kappa}_j \tag{2c}$$

$$\bar{\kappa}_j = \sum_i \frac{e_i\,|\omega_i\kappa_i|}{d_{ij}^2} \Big/ \sum_i \frac{e_i\,|\omega_i|}{d_{ij}^2}, \tag{2d}$$

where δ is the surgical or cutoff scale, L is a length typical of the large-scale vorticity distribution, μ times L is the spacing of successive nodes on a *circular* vortex of radius $L \gg \delta$, a is a number between 0 and 1 that controls how quickly the node density rises with curvature, $e_i = \|\mathbf{x}_{i+1} - \mathbf{x}_i\|$ is the straight-line distance between two adjacent nodes i and $i+1$, ω_i is the jump in vorticity across the segment $(i, i+1)$ ($\omega_i = \tilde{\omega}_k$ in Eq. (1) for i on the kth contour), κ_i is the curvature at node i computed by passing a circle through the nodes $i-1$, i, and $i+1$, $d_{ij} = \|\mathbf{x}_j - \tfrac{1}{2}(\mathbf{x}_i + \mathbf{x}_{i+1})\|$ is the distance between node j and halfway between nodes i and $i+1$, and the sum in Eq. (2d) runs over all nodes.

The nonlocality of the node density function presents itself through $\bar{\kappa}_j$, Eq. (2d). The term $e_i\,|\omega_i|/d_{ij}^2$ is roughly proportional to the variation of the velocity across the segment $(j, j+1)$ contributed by the segment $(i, i+1)$. In Eq. (2c), the depen-

dence of $\tilde{\kappa}_j$ on $\bar{\kappa}_j$ touches on the question of accuracy for an algorithm with a finite range of scales ($\delta \neq 0$). The first term dominates for "large" scales, $\bar{\kappa}_j L \ll (2^{1/2}\mu)^{-1/(1-a)}$, while the second is designed to be at least equally important near the cutoff scale, $\bar{\kappa}_j = O(\delta^{-1})$. Considering just the first term, a value of a less than unity implies that the largest scales are resolved best. For example, an isolated circular vortex of sufficiently large radius R would be resolved by (approximately) $(2\pi/\mu)(R/L)^{1-a}$ nodes, a number which decreases with decreasing radius when $a < 1$. The variation in the number of nodes used to resolve different scales is consistent with the expectation that the largest scales determine the greatest part of the dynamics. The second term in Eq. (2c) is intended to become important only for scales near the cutoff scale δ and impacts on one form of surgery that takes place when curvatures exceed δ^{-1} (approximately), namely the formation of *corners*. Corners are places along a contour where three successive nodes are forced into an acute angle as a result of inadequate resolution. More discussion of corners may be found in the subsection on surgery below, but it is appropriate here to discuss how Eq. (2) behaves near the cutoff scale. The functional form of $\rho_j(\hat{\kappa}_j)$ in Eq. (2a) ($\hat{\kappa}_j$, defined in Eq. (2b), being merely the average of $\tilde{\kappa}_j$ and $\tilde{\kappa}_{j+1}$) limits the minimum distance between nodes for *any* curvature to $\delta/2^{1/2}$. However, since the curvature at a node is in fact computed by passing a circle through that node and its two neighbors, the maximum computed curvature is limited to $O(\delta^{-1})$ whence the minimum distance between nodes tends to be nearly or slightly greater than δ.

Interpolation

In the algorithm, a contour consists of its nodes and the interpolation functions between nodes. Linear interpolation is the simplest choice, but higher-order interpolation can significantly improve accuracy for a small cost in computer time. The part of the contour joining two nodes is approximated by a locally-determined cubic polynomial; between two nodes i and $i+1$ on a given contour, the contour takes the form

$$\begin{aligned} \mathbf{x}(p) &= \mathbf{x}_i + p(a_i, b_i) + \eta(p)(-b_i, a_i) \\ (a_i, b_i) &= \mathbf{x}_{i+1} - \mathbf{x}_i \\ \eta(p) &= \alpha_i p + \beta_i p^2 + \gamma_i p^3 \end{aligned} \tag{3}$$

for $0 \leqslant p \leqslant 1$ (see Fig. 2). α_i, β_i, and γ_i are determined by $\eta(1) = 0$, $\kappa(0) = \kappa_i$, and $\kappa(1) = \kappa_{i+1}$, where

$$\kappa(p) = \frac{d^2\eta/dp^2}{e_i(1 + (d\eta/dp)^2)^{3/2}}. \tag{4}$$

The distribution of nodes is such that η, the normal departure of the contour from a straight line divided by the distance between the two nodes, is small compared with unity. This allows one to neglect $(d\eta/dp)^2$ compared with 1 in the expression for

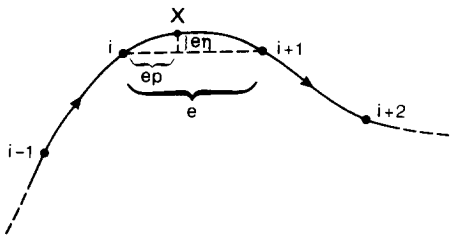

FIG. 2. The interpolation between nodes. Subscripts have been supressed (e.g., $e = e_i$).

$\kappa(p)$ above rendering the curvature a *linear* function of p between nodes and a piecewise-linear function around the contour (except at corners as noted below). Under these approximations, Eqs. (3) and (4) imply

$$\begin{aligned} \alpha_i &= -\tfrac{1}{3}e_i\kappa_i - \tfrac{1}{6}e_i\kappa_{i+1} \\ \beta_i &= \tfrac{1}{2}e_i\kappa_i \\ \gamma_i &= \tfrac{1}{6}e_i(\kappa_{i+1} - \kappa_i). \end{aligned} \tag{5}$$

One may well ask why continuity of curvature is preferred over continuity of tangent slope, as the latter appears to involve one less differentiation of the curve. Tests of a scheme that forces the contour, at each node i, to be tangent to the weighted vector $(\mathbf{x}_{i+1} - \mathbf{x}_i)/e_i^2 + (\mathbf{x}_i - \mathbf{x}_{i-1})/e_{i-1}^2$ has shown that the present curvature continuity scheme actually leads to *smaller* errors (in terms of the errors ε_A, ε_u, and ε_t discussed in the subsection on accuracy below). Indeed, in both schemes, the coefficients of the cubic polynomial $\eta(p)$ involve information at 4 successive nodes (hence the 3rd-order accuracy noted below), and so it is not surprising that one combination of the information at the 4 nodes leads to greater accuracy than another. In practice, the slight error in the continuity of tangent slopes in the curvature scheme presents no observable "side-effects."

Node Redistribution

At each time step, nodes are redistributed around the contour by way of the known variation of the contour between each pair of adjacent nodes. If a contour has corners, the node redistribution takes place *between* the corners, and the corners remain fixed. If the contour does not have corners, one node is held fixed, and the remaining nodes are adjusted relative to this fixed node. In both cases, the method for redistribution outlined below is identical.

Let 1 be the index of the first (and fixed) node and n be either the index of the node preceding the next corner or the last node on the contour. The first step is to compute the quantity

$$q = \sum_{i=1}^{n} \rho_i e_i \tag{6a}$$

and define $\tilde{n}$ to be the nearest integer to q *plus* 2. $\tilde{n}-1$ will be the number of nodes between the fixed nodes after redistribution. Next, the node densities ρ_i are multiplied by $\tilde{n}/q$ so that the sum in Eq. (6a), after this redefinition of ρ_i, equals $\tilde{n}$. Finally, the positions of the new nodes $j=2, \ldots, \tilde{n}$ are found by seeking i and p such that

$$\sum_{l=1}^{i-1} \rho_l e_l + \rho_i e_i p = j-1 \tag{6b}$$

and placing the jth new node between the old nodes i and $i+1$ at the position $\mathbf{x}(p)$ given in Eq. (3).

Velocity Determination

This new node distribution/interpolation scheme gives rise to a significant improvement in the evaluation of the velocity field from Eq. (1). Since η is small compared with unity, the velocity integral between any pair of nodes may be expanded in a perturbation series in η:

$$\dot{\mathbf{u}} = \frac{1}{2\pi} \sum_k \tilde{\omega}_k \sum_i (\Delta\mathbf{u})_i$$

$$(\Delta\mathbf{u})_i = -\int_0^1 dp \log \|\mathbf{x} - (\mathbf{x}_i + p(a_i, b_i) + \eta(-b_i, a_i))\| \left\{(a_i, b_i) + \frac{d\eta}{dp}(-b_i, a_i)\right\}$$

$$= (\Delta\mathbf{u})_{0i} + (\Delta\mathbf{u})_{1i} + \cdots$$

$$(\Delta\mathbf{u})_{0i} = -(a_i, b_i) \int_0^1 dp \log \|\mathbf{x} - (\mathbf{x}_i + p(a_i, b_i))\|$$

$$(\Delta\mathbf{u})_{1i} = (a_i, b_i)(a_i(y-y_i) - b_i(x-x_i)) \int_0^1 \frac{\eta(p)\, dp}{\|\mathbf{x} - (\mathbf{x}_i + p(a_i, b_i))\|^2}$$

$$-(-b_i, a_i) \int_0^1 dp \frac{d\eta}{dp} \log \|\mathbf{x} - (\mathbf{x}_i + p(a_i, b_i))\|. \tag{7}$$

$(\Delta\mathbf{u})_{0i}$ is the contribution to the integral from integrating along the line segment connecting the two nodes, while $(\Delta\mathbf{u})_{1i}$ represents a correction linear in η. The fact that $(\Delta\mathbf{u})_{0i}$ can be evaluated explicitly has often been used by other researchers [10, 11] to avoid ill-conditioned quadrature formulae [4]. The fact that $(\Delta\mathbf{u})_{1i}$ can *also* be evaluated explicitly has not been previously used, but, as discussed below, it allows one to use many times fewer nodes than required in linear interpolation for the same accuracy. $(\Delta\mathbf{u})_i$ to first order in η is given by (without the subscripts i)

$$\begin{aligned}\Delta\mathbf{u} = {} & (a, b)\{1 - dr - (1-d)\, s - c^2 h + c[\beta + (\tfrac{1}{2} + 2d)\, \gamma + f(s-r) + gh]\} \\ & + (-b, a)\{\alpha + (\tfrac{1}{2} + d)\, \beta + (\tfrac{1}{3} + \tfrac{1}{2}d + d^2 - c^2)\, \gamma \\ & + (s-r)[\alpha d + \beta d^2 + \gamma d^3 - c^2(\beta + 3d\gamma)] - c^2 hf\},\end{aligned} \tag{8a}$$

where

$$c = [a(y-y_i) - b(x-x_i)]/e^2 \tag{8b}$$

$$d = [a(x-x_i) + b(y-y_i)]/e^2$$

$$e = (a^2+b^2)^{1/2}$$

$$f = \frac{d\eta}{dp}(d) - c^2\gamma$$

$$g = \eta(d) - c^2(\beta + 3d\gamma)$$

$$h = \frac{1}{c}\left(\tan^{-1}\left(\frac{1-d}{c}\right) + \tan^{-1}\frac{d}{c}\right)$$

$$r = \log\|\mathbf{x}-\mathbf{x}_i\|$$

$$s = \log\|\mathbf{x}-\mathbf{x}_{i+1}\|.$$

A minor correction is made when the evaluation point $\mathbf{x}$ happens to be inside the circle whose diameter is the segment connecting two adjacent nodes, say $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$. In this case, the perturbation series given by Eq. (7) is no longer valid, particularly for $\mathbf{x}$ near the contour but not including the two nodes. However, an equally valid perturbation series results if, as illustrated in Fig. 3, the line segment connecting the two nodes is shifted parallel to itself (so the shifted segment intersects $\mathbf{x}(p^*)$, and the contour integral is expanded about this shifted segment. In other words, define $\eta^* = \eta - \eta(p^*)$ with $p^* = d_i$ and use $\mathbf{x}_i^*$, $\mathbf{x}_{i+1}^*$, and η^* in place of the corresponding unstarred quantities in Eqs. (7) and (8). In actual practice, no perceptible change is observed in computational efficiency.

Accuracy

In this subsection, quantitative measurements are presented of the kinematical accuracy of the interpolation, velocity calculation, node redistribution, and time stepping components of the algorithm when surgery is absent ($\delta = 0$). Most of the effort is directed at developing measures of error for an example flow, the Kirchoff elliptical vortex, which is steady and for which simple expressions govern the velocity and position of a fluid particle on the boundary at any time t [8]. The subsection concludes with an application to an unsteady flow.

The elliptical vortex is described by $x(\theta) = \cos\theta$, $y(\theta) = \lambda^{-1}\sin\theta$, $0 \leqslant \theta \leqslant 2\pi$ for

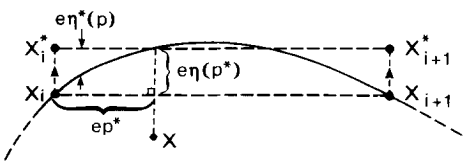

FIG. 3. Calculating the velocity when the evaluation point is close to a contour.

any aspect ratio λ, and the uniform vorticity ω within the vortex is taken to be 2π. The test algorithm replaces the exact ellipse by a fixed number of nodes n distributed according to Eq. (2) for a given curvature power a, $L=1$, and $\delta=0$. The desired distribution of nodes is obtained iteratively by starting from an equal angle (in θ) distribution and then applying Eqs. (2)–(6) five times (further iterations proving unnecessary). After each iteration, x and λy for each node are multiplied by $(x^2+\lambda^2 y^2)^{-1/2}$ to keep the nodes on the elliptical boundary.

Two measures of error associated with interpolation and velocity determination lead the discussion: the positive area difference between the exact and interpolated ellipse (the so-called L_1 area difference) scaled by the area of the ellipse,

$$\varepsilon_A = \frac{\lambda}{\pi} \sum_{i=1}^{n} e_i^2 \int_0^1 |\eta - \bar{\eta}|\, dp, \tag{9a}$$

and the RMS velocity difference (the L_2 velocity difference),

$$\varepsilon_u = \frac{(\sum_{i=1}^{n} \|\mathbf{u}_i - \bar{\mathbf{u}}_i\|^2)^{1/2}}{(\sum_{i=1}^{n} \|\bar{\mathbf{u}}_i\|^2)^{1/2}}, \tag{9b}$$

where η and $\bar{\eta}$ are the interpolated and exact departures of the contour from a straight line between nodes i and $i+1$ (see Eq. (3)), and $\mathbf{u}_i$ and $\bar{\mathbf{u}}_i$ refer to the velocity at the ith node for the interpolated and exact vortex, respectively. It is a simple exercise to show that *linear* interpolation leads to an area error of $O(n^{-2})$. Since the velocity at a node involves a sum over all nodes, one might expect the velocity error to be one order less accurate; however, cancellation effects reduce the error to $O(n^{-2})$ (a specific example is given below). *Cubic* interpolation can be expected to be *at most* two orders more accurate. In fact, the area error is $O(n^{-4})$ while the velocity error, apparently not benefitting from further cancellations, is $O(n^{-3})$. These asymptotic dependencies are expressed in terms of $\hat{\mu} \equiv 2\pi/n$ as

$$\begin{aligned} \varepsilon_A &= C_A(\lambda, a)\, \hat{\mu}^4 \\ \varepsilon_u &= C_u(\lambda, a)\, \hat{\mu}^3. \end{aligned} \tag{10}$$

Table I lists C_A and C_u versus λ for the three powers $a=\frac{1}{3}$, $\frac{2}{3}$, and 1. It has been determined to three decimal places that the choice $a=\frac{2}{3}$ minimises the velocity error for all λ. Furthermore, it appears that $\varepsilon_u = 0.029\hat{\mu}^3$ independent of λ. Such a property is ideal for the construction of an accurate numerical algorithm because errors in the computed velocity field most rapidly accumulate, being of $O(n^{-3})$. We note that the smallest area error does not coincide with the smallest velocity error. The area error is minimised at $a=\frac{1}{2}$, but $C_A(\lambda, \frac{1}{2})$ increases with λ ($C_A(10, \frac{1}{2}) = 0.515$).

The smallest area and velocity errors for *linear* interpolation of an ellipse are obtained by distributing nodes according to curvature to the *one-third* power [19], $a=\frac{1}{3}$, which corresponds to spacing the nodes at equal intervals in θ. In this case,

TABLE I

A Comparison of Accuracy between Three Different Interpolations, Curvature to the One-third, Two-thirds, and First Power

	$a=\frac{1}{3}$		$a=\frac{2}{3}$		$a=1$	
λ	C_u	C_A	C_u	C_A	C_u	C_A
1	0.0290	0.0125	0.0290	0.0125	0.0290	0.0125
2	0.0423	0.116	0.0288	0.116	0.0421	0.278
3	0.0658	0.247	0.0287	0.266	0.0658	1.48
4	0.0944	0.385	0.0288	0.427	0.0978	4.28
5	0.127	0.527	0.0291	0.584	0.142	9.10
6	0.164	0.673	0.0295	0.733	0.201	16.0
7	0.205	0.821	0.0299	0.874	0.276	25.0
8	0.250	0.971	0.0305	1.00	0.367	35.9
9	0.299	1.12	0.0311	1.13	0.471	48.4
10	0.352	1.28	0.0317	1.24	0.587	62.5

Note. Shown are $C_u = \varepsilon_u/\hat{\mu}^3$ and $C_A = \varepsilon_A/\hat{\mu}^4$, where ε_μ is the L_2 velocity error and ε_A is the L_1 area error (both defined in the text). $\hat{\mu} = 2\pi/n$, and $n = 512$ nodes are used. The comparison is shown as a function of the aspect ratio of the test ellipse, λ.

$\varepsilon_A = \hat{\mu}^2/6$ (asymptotically), independent of aspect ratio, and $\varepsilon_u = C\hat{\mu}^2$ with $C = 0.084$, 0.089, 0.105, and 0.136 for $\lambda = 1$, 2, 4, and 8, respectively.

The above errors measure only the *instantaneous* accuracy of the algorithm while, in a numerical integration, one is more concerned with the error made over one time step due to all the numerical approximations. Consider then first distributing $n = 256$ nodes as described above for a given curvature power a, next integrating the system forward in time using a typical time step, $\Delta t = 0.05$, in a 4th-order Runga–Kutta scheme, and finally redistributing the nodes (holding n fixed at 256). The interpolated ellipse is next compared with the known position of the exact ellipse by measuring the closest distance d_i between each node i and the exact elliptical boundary and computing the RMS distance error normalized by the length-scale, $\lambda^{-1/2}$:

$$\varepsilon_t = \left(\frac{\lambda}{n}\sum_{i=1}^{n} d_i^2\right)^{1/2}. \tag{11}$$

Table II lists $10^7\varepsilon_t(\lambda, a)$ for $\lambda = 1$ to 10 and three selected curvature powers. Overall, the choice $a = \frac{2}{3}$ fares best, $a = \frac{1}{3}$ being slightly worse, and $a = 1$ appearing distinctly unstable.

Finally, it is worth noting that similar accuracy results have been found for non-equilibrium vortices. The procedure to obtain ε_t wavers only slightly from that discussed above in that the exact time evolution of the vortex is unknown. Essentially, for a single vortex boundary, a calculation with n nodes carried through one time

TABLE II

The Error ε_t after One Time Step and One Node Adjustment *Multiplied by* 10^7 versus the Aspect Ratio λ of the Ellipse for Three Curvature Powers a

λ	$a=\frac{1}{3}$	$a=\frac{2}{3}$	$a=1$
1	0.8440	0.8440	0.8440
2	2.735	2.814	3.007
3	5.302	5.349	4.980
4	6.892	6.800	6.676
5	7.777	7.697	6.451
6	8.211	8.103	17.68
7	8.533	7.831	19.60
8	8.712	7.722	12.64
9	9.042	7.725	32.04
10	9.629	7.782	56.48

Note. $\Delta t = 0.05$, $n = 256$ nodes.

step Δt is compared with a calculation with $2n$ nodes carried through two time steps of size $\frac{1}{2}\Delta t$. The nodes are redistributed after the first time step on the lower resolution vortex but are not redistributed after either time step on the high resolution vortex. As an example, take $n = 256$ and $\Delta t = 0.05$ for the "squashed" ellipse, $x = \cos\theta$, $y = (1 - x/x_0)\lambda^{-1}\sin\theta$. In the limit $x_0 \to \infty$, the elliptical vortex is recovered, and the results for ε_t compare well with those in Table II; for example, when $\lambda = 1$, $\varepsilon_t = 1.007 \times 10^{-7}$ compared with 0.844×10^{-7} obtained in Table II. For initial conditions departing significantly from the elliptical vortex, such as when $x_0 = 2$ and $\lambda = 5$, the curvature power $a = \frac{2}{3}$ continues to perform better than either $a = \frac{1}{3}$ or $a = 1$: in this case $\varepsilon_t = 4.921 \times 10^{-7}$, 3.695×10^{-7}, and 8.622×10^{-7} for $a = \frac{1}{3}$, $\frac{2}{3}$, and 1, respectively.

Surgery

The development of regions of rapidly increasing curvature and ever growing contour lengths presents an insurmountable obstacle to extended calculations using CD. Calculations almost inevitably grind to a halt either from insufficient resolution or insufficient computer resources (e.g., Figs. 7 and 8 from [16] and Figs. 3, 5–10, 12–14, 16, and 17 from [4]). By the introduction of a minimum length scale, δ, CS allows the efficient computation of exceedingly complex vorticity dynamics for times unreachable by CD. Such a minimum scale represents an approximation, but it is not unlike a maximum wavenumber in a spectral model or a minimum grid spacing in a grid-point model. CS assumes that the neglected scales of motion (1) behave as a passive tracer, (2) cascade to ever finer scales, and (3) do not accumulate to a significant fraction of the total vorticity. The degree to which these assumptions are justified can only be determined from a careful examination of a wide variety of numerical experiments, some of which are presented below.

The algorithm enforces the minimum or cutoff scale δ as follows. When two contours enclosing identical vorticity find themselves closer than δ, they are joined as illustrated in Fig. 4. The node where the joining takes place becomes two differently labelled nodes, and each of these nodes receives the special status of being a *corner*. Corners are nodes which are fixed during node redistribution and make acute angles with their adjacent nodes. When a single contour tries to break into two pieces by having two distinct parts of itself closer than δ, an operation identical to that shown in Fig. 4 is performed.

Corners also develop when the curvature exceeds δ^{-1} (approximately)—an acute angle is then forced by the node density function, Eq. (2). The nodes adjacent to a corner are attached to the corner by straight line segments as a result of setting the curvature to zero at all three nodes, and this has the effect of greatly reducing the resolution near the corner once the nodes are redistributed after surgery (the number of nodes in the vicinity of the corner typically drops by 5–10). Corners are eliminated if they become obtuse by simply returning the corner to the status of a simple node.

No further surgical operations are performed except for the elimination of contours with too few nodes (4 or fewer). Even so, roughly 80 % of the computer code deals with surgery, but fortunately, only about 7 % of the computer time is spent on surgery. The actually observed timing for the non-surgical part of the algorithm in seconds per time step using 64-bit arithmetic on a two-pipe, vector-processing Cyber 205 is given by $1.37 \times 10^{-3} n + 1.01 \times 10^{-5} n^2$, where n is the total number of nodes. Thus, it takes 11.5 s to move 1000 nodes in one time step.

The flow chart presented in Table III summarizes the CS algorithm.

Blind Alleys

The development of the algorithm was not without many unsuccessful ventures. Needless to say, much was learned on the turbulent path to the present algorithm that could prove useful for further enhancements of CD/CS.

One of the most important aspects of the present algorithm is the use of a non-local node density function. Contour crossing and related spatial errors result when nodes are distributed according to a local function of curvature. The generic situation is the presence of a region of high curvature whose radius of curvature is comparable to or less than the distance between nodes on a nearby contour. The high curvature region induces a velocity field which varies too rapidly for the nearby contour to distort properly, and no matter how small the time step may be, the lack of degrees of freedom prohibits the nearby contour from "seeing" the

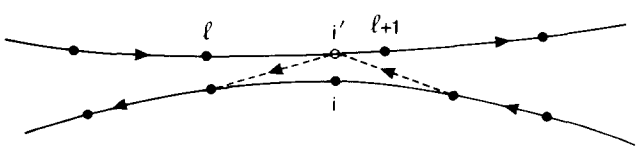

FIG. 4. The merging of two contours.

TABLE III

Flow Chart

I. Initialization
   a. Read in data specifying the vortex shape and algorithm parameters.
   b. Calculate the cubic interpolation coefficients (Eqs. (3)–(5)).
   c. Redistribute the nodes (Eqs. (2), (6)).
   d. Save the initial condition for post-processing.

II. Advection
   a. Recalculate the interpolation coefficients.
   b. Calculate the velocity field (Eq. (8)).
   c. Repeat steps a and b three more times to complete the Runga–Kutta integration, then pass to item III.

III. Surgery
   a. Search for new corners between existing corners.
   b. Search for contour merger situations (Fig. 4).
      (i) If a single contour is found to satisfy the merger condition, break the contour into two pieces. Introduce two new corners at the break (at node i' in Fig. 4).
      (ii) If two contours containing identical interior vorticity are found to satisfy the merger condition, join them into a single contour. Introduce two new corners at the place of joining (at node i' in Fig. 4).
   c. Repeat step b until all possible merger situations have been examined, then pass to item IV.

IV. Post-surgery
   a. Re-calculate the interpolation coefficients.
   b. Redistribute the nodes.
   c. Periodically save data for post-processing.
   d. Return to item II unless a specified number of time steps have been taken, in which case the calculation ends.

approaching high curvature region, and contour crossing is inevitable. A very large time step in the present algorithm would also result in spatial errors—nodes might become separated too much in one time step for accurate interpolation to take place. However, the calculations presented in the next section employ a sufficiently small time step to prevent significant interpolation errors.

Various alternative methods for interpolation between nodes have been considered. Cubic splines were not used because of their potentially wild behaviour near developing corners, particularly when the curvature varies by four or more orders of magnitude around a contour. The method discussed in [4] where the cubic between each two nodes i and $i+1$ is forced to pass through the nodes $i-1$ and $i+2$ as well and *then* calculating the curvature directly from this cubic polynomial causes the contour to buckle exponentially. Finally, higher order interpolations (e.g., quartic) require, among other things, the explicit calculation of one further term in the velocity field expansion rendering the algorithm hopelessly complex and inefficient.

Without interpolation (using straight line segments between nodes) the accumulated error, in terms of loss of conservation of circulation, etc., is much too large to perform long-time calculations. With interpolation, but using Gaussian quadrature (numerical quadrature) to perform the contour integrals in the velocity determination artificially induces Kelvin–Helmholtz instability along thin strands of vorticity (see Fig. 13 in [4]). The present method of quadrature, in which the integral along a curved segment is performed explicitly to first order in the departure of the segment from a straight line, avoids this numerical instability.

Perhaps the greatest effort was invested tackling problems that arose from improper surgery. Despite its superficial simplicity, surgery represents a logical nightmare, comprising some thousand lines of code in the algorithm. Space does not permit a discussion of each of the blind alleys encountered trying to cope with rare events, but one problem is particularly important because it constrains the choice of the various algorithm parameters. Recall that contours are joined if the distance between a node on one (part of a) contour and the *straight-line segment* connecting two nodes on another (part of a) contour is less than the cutoff scale δ. In other words, surgery does *not* look at the actual distance between the curved contours, since it has been assumed that the contour's departure from a straight line between its nodes is small. However, if the cutoff scale is smaller than this departure, surgery might not take place even though the *curved* contours are closer than the cutoff scale and, worse still, possibly crossing each other. Such errors may be avoided by choosing the cutoff scale large enough, specifically $\delta > \max(e\eta)$, where $e\eta$ is the normal departure from a straight segment of the contour between two adjacent nodes and "max" means the maximum over all nodes. To see how this requirement constrains the algorithm parameters, consider the following tractable example. For a circular vortex of radius R, it is simple to show that

$(1/8R)\,\mu^2 L^2 (R/L)^{2a}$. The expression for $\max(e\eta)$ is greatest for the largest scales, $R \approx L$, when $a > \frac{1}{2}$ (justifying the approximation used for e above) whence we obtain the estimate

$$\frac{\delta}{L} > \frac{1}{8}\mu^2. \tag{12}$$

In practice, Eq. (12) is an overestimate, and values of δ one-tenth as small as the right-hand side of Eq. (12) have been used without causing contour crossing errors.

3. Calculations with Contour Surgery

The error estimates of the previous section provide only a rough guide to the actual performance of the algorithm over an extended integration period. In the first half of this section, multiple calculations beginning with identical initial con-

TABLE IV

The Parameter Values and the Initial and Final Number of Nodes for the Seven Elliptical Vortex Calculations

Case	μ	Δt	δ	n_i	n_f
1	$\pi/100$	0.05	10^{-4}	153	2581
2	$\pi/200$	0.05	10^{-4}	294	2027
3	$\pi/50$	0.05	10^{-4}	82	1298
4	$\pi/100$	0.025	10^{-4}	154	2430
5	$\pi/100$	0.1	10^{-4}	146	1607
6	$\pi/100$	0.05	10^{-5}	153	2120
7	$\pi/100$	0.05	10^{-3}	153	2869

ditions assess the *dynamical* sensitivity of the algorithm to variations in resolution, time step, and cutoff scale, and one of these calculations is run backwards to check time-reversibility. The second half of this section is a portrait gallery exhibiting several long-time calculations that highlight a few problems of current interest in two-dimensional vortex dynamics.

Dynamical sensitivity is assessed by tracking the degree to which various quantities remain conserved over the course of the evolution. The quantities monitored are the total circulation Γ, the angular momentum J, the x- and y-centroids x_c and

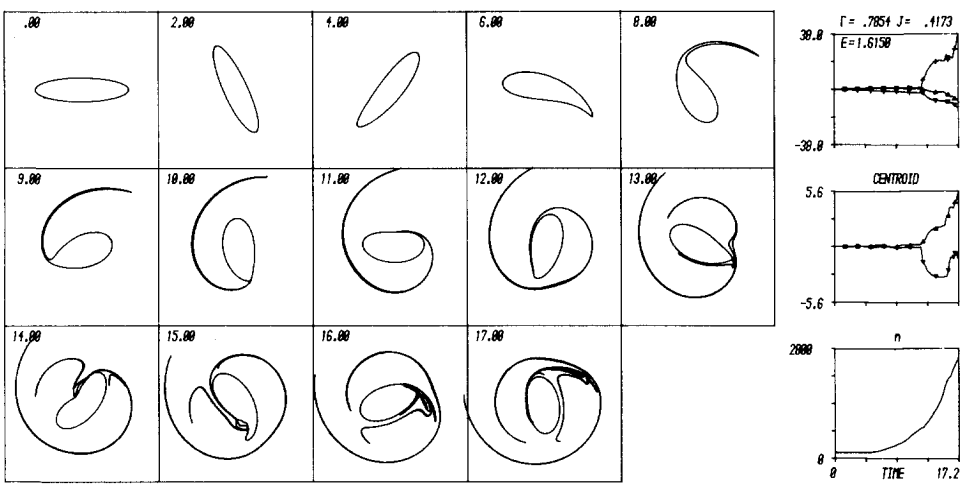

FIG. 5. The evolution of a perturbed ellipse. $\mu = \pi/100$, $\Delta t = 0.05$, $\delta = 10^{-4}$. The time is indicated in the upper left corner of each frame. The panels on the right give an indication of the accuracy of the calculation. The upper panel displays, as a function of time, departures of circulation $\Gamma(\Delta)$, angular momentum $J(\nabla)$, and energy E $(+)$ from exact conservation *multiplied* by 10^4. The middle panel displays the errors in x centroid $x_c(\Delta)$ and y centroid $y_c(\nabla)$ similarly multiplied by 10^4. The bottom panel shows the total number of nodes.

y_c, and the nondimensional excess energy E (see Eqs. (2.1.) and (2.2) of [4]). All of these quantities have contour integral representations (see Eqs. (A2), (A4), and (A8) of [4]) and are calculated as follows. For Γ, J, x_c, and y_c, the contour integration is exact to first order in the departure η of the contour from a straight line between adjacent nodes. The energy, whose calculation requires a double contour integration, relies on three-point Gaussian quadrature between adjacent nodes (for the quadrature coefficients, see [1, p. 916]). Out of convenience, all quantities are computed using $\omega/2\pi$ in place of the vorticity ω since the peak vorticity in all the calculations always equals 2π.

Seven calculations are compared that differ only in the choices of μ, Δt, and δ (see Table IV). The large-scale length L is taken to be unity throughout. The initial condition is a 4 : 1 ellipse perturbed with the superposition of the growing and decaying 3-fold symmetric linear eigemode [4]:

$$\mathbf{x}(\theta, 0) = (\cos\theta, \lambda^{-1}\sin\theta) + \frac{\varepsilon\lambda^{-1}\cos m\theta}{\sin^2\theta + \lambda^{-2}\cos^2\theta}(\lambda^{-1}\cos\theta, -\sin\theta), \qquad (13)$$

with $\lambda = 4$, $\varepsilon = 0.005$, and $m = 3$. At this aspect ratio, the 3-fold symmetric eigenmode is the only one that leads to linear instability.

The evolution of case 1 in Table IV, Fig. 5, in many ways typifies unstable elliptical vortex evolution. Initially, the eigenmode amplifies, in this case causing the vortex to become more egg-like in shape. This is followed by a strongly nonlinear stage where one or two long filaments are thrown off either or both ends (ellipses of greater eccentricities tend to break into two or more pieces—see Fig. 14 of [4] and Fig. 15 below). Next, the asymmetric velocity field arising from the distorted shape of the interior "core" of the vortex compresses and thereby enlarges one section of

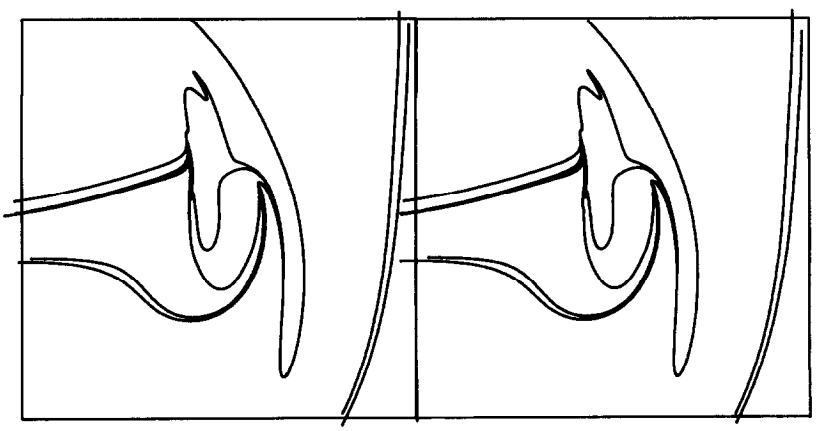

FIG. 6. A close-up comparison between cases 1 (left) and 6 (right) of Table IV at $t = 15.7$. The calculation on the right has one-tenth the cutoff scale of that on the left, otherwise the calculations are identical.

the filament(s) where the compressional strain is most extreme. The location of this enlarged region along the filament(s) propagates with nearly the same angular velocity as the interior vortex core, and the two collide. The re-attachment is short-lived however, more rotational fluid is flung off a second time, and the core of the vortex further reduces its eccentricity. In some cases, the filaments may collide a second time leaving the core yet more circular. However, once the core has sufficiently weak eccentricity, the filaments remain in the exterior field of the core. The variable strain field of the rotating core rapidly cascades the filaments to small scales, and the flow tends to an ellipse of reduced aspect ratio within a diffuse field of vorticity. An alternative explanation for the observed reduction in aspect ratio makes use of the cororating-frame streamfunction (see [9, Sect. 3]).

The seven cases in Table IV are next compared in detail. The seven calculations cannot be visibly distinguished from each other apart from the varying lengths of the very thin filaments, for example, cases 1 and 6 are compared in Fig. 6 (left and right frames), the latter having one-tenth the cutoff scale of the former ($t = 15.7$).

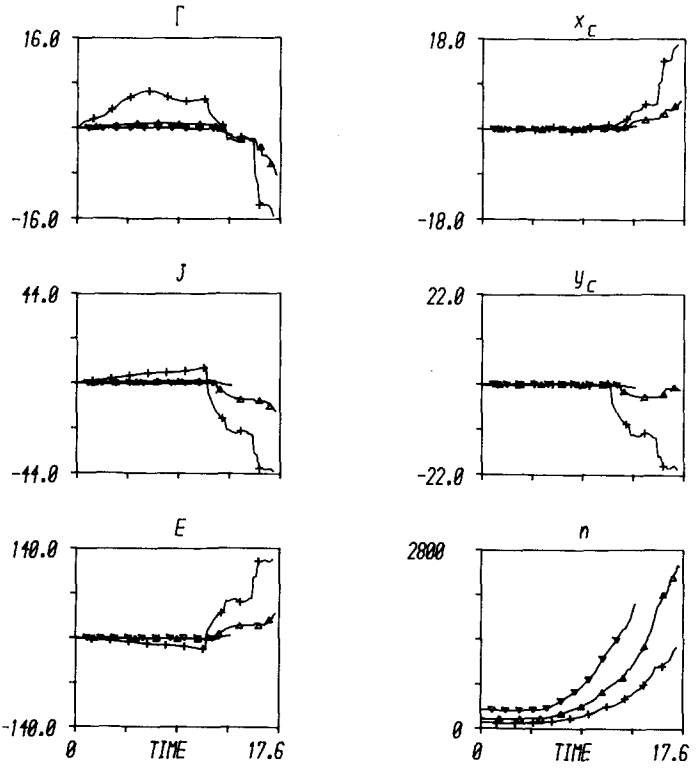

FIG. 7. Comparison across resolution, μ. The six panels illustrate the departures from conservation of Γ, J, E, x_c, and y_c multiplied by 10^4 as well as the total number of nodes for three values of μ: $\pi/200$ (∇), $\pi/100$ (Δ), and $\pi/50$ ($+$).

However, differences do emerge upon examining the departures from convervation of Γ, J, x_c, y_c, and E. Three comparisons are made: (1) across resolution (cases 1, 2, and 3), Fig. 7; (2) across time step (cases 1, 4, and 5), Fig. 8; and (3) across cutoff scale (cases 1, 6, and 7), Fig. 9. Surgical errors are noticeable from the sudden jumps in error and may either decrease *or* increase the circulation and angular momentum; for example, when a contour breaks into two pieces, Γ and J decrease (for positive vorticity) while the merger of two separate contours increases Γ and J. The energy variation tends to oppose the variations in circulation and angular momentum.

As expected, smaller values of μ, Δt, and δ lead to greater accuracy. However the relatively weak variation across time step suggests that $\Delta t = 0.05$ is sufficiently small for calculations of this length. Larger improvements occur when μ and δ are decreased. The smaller value of μ results in significantly better conservation between times when surgery takes place while smaller δ greatly reduces surgical errors. Note that halving μ nearly doubles the number of nodes making the calculation four times more expensive while reducing the cutoff scale by ten only

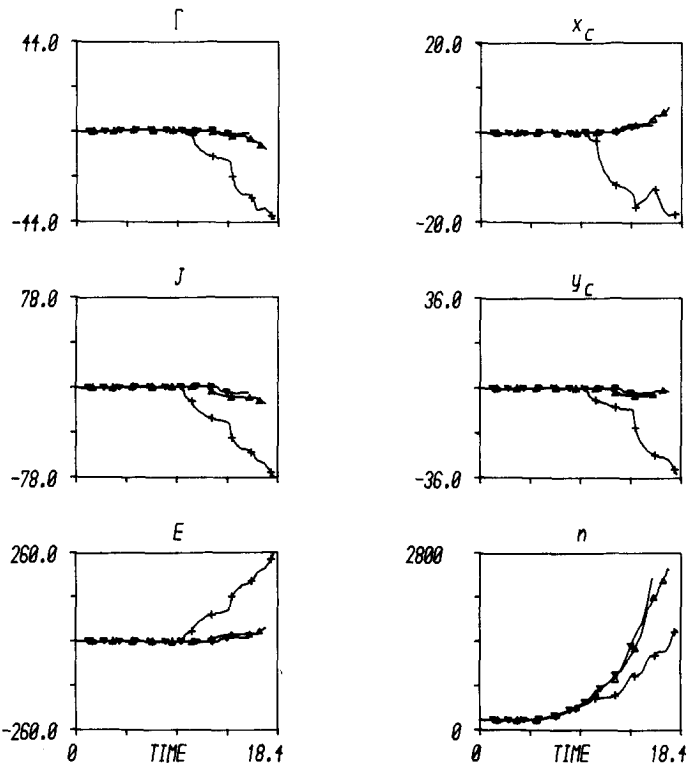

FIG. 8. Comparison across time step, Δt. Three values of Δt are compared: 0.025 (∇), 0.05 (Δ), and 0.1 (+).

slightly increases the total number of nodes. The weak dependence of the number of nodes on the cutoff scale is related to the favoritism given to the large scales by choosing the curvature power $a = \frac{2}{3}$. Few nodes are used to resolve the smallest scales, so that a dramatic increase in the range of scales does not carry with it a correspondingly dramatic increase in the number of nodes.

Next consider running a calculation back to its initial condition. Case 1 of Table IV was run back from $t = 7.5$, before surgery had taken place, to $t = 0$. Given the degree to which the vortex had been distorted by $t = 7.5$, it may be surprising that the reversed calculation succesfully reproduces the original initial conditions with very small errors in the conserved quantities (see Fig. 10). In general, however, this is not a fair test of an algorithm's accuracy. Had the ellipse been perturbed initially by numerical noise alone, no semblance of the initial conditions would have been reproduced. During the integration backwards, disturbances created by small errors in interpolation and time stepping that would normally decay in a forward integration grow on the way back to $t = 0$. However, these disturbances arise from errors which are *three* to *four* orders of magnitude smaller than the initial

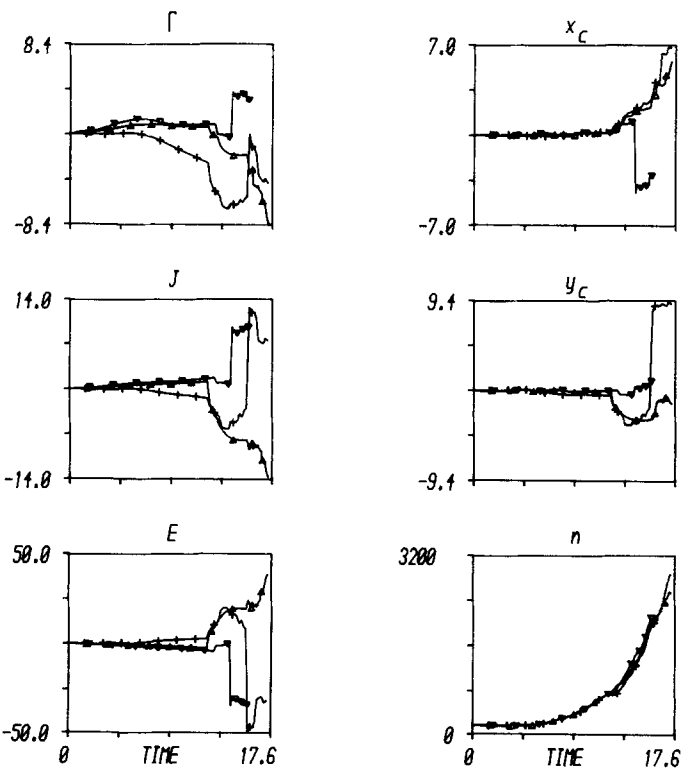

FIG. 9. Comparison across cutoff scale, δ. Three values of δ are compared: 10^{-5} (V), 10^{-4} (Δ), and 10^{-3} (+).

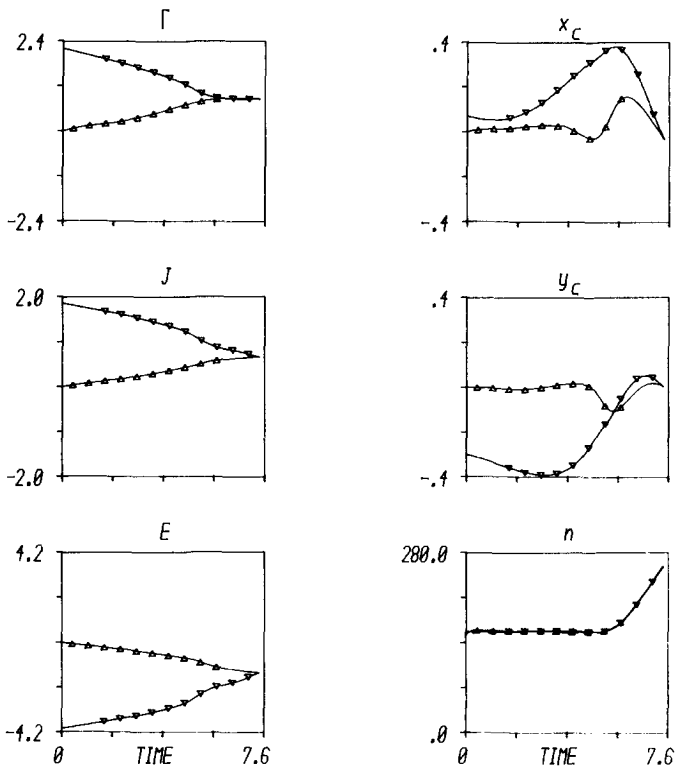

FIG. 10. The diagnostics for the outbound (Δ) and inbound (∇) evolution of case 1 in Table IV.

amplitude of the (unstable) perturbation in the example above, and consequently, over a sufficiently short integration period, little damage can be done by the time-reversed growing disturbances.

High Resolution Calculations

The following calculations not only demonstrate the ability of the algorithm to compute complex fluid motions but also bring new insight into problems that have received much attention recently using standard numerical approaches and outline several tantalizing problems in vortex dynamics. The first two calculations involve nested contours or multiple levels of vorticity and show that it is feasible to follow the dynamics of a moderate number of contours. The degree to which this represents the dynamics of a *continuous* distribution and vice versa remains an open question, however.

A recent study [9] using spectral techniques has found that continuous distributions of vorticity of initially elliptical shape evolve by repeatedly throwing off "arms" of vorticity which strip an increasing amount of vorticity off a central structure of decreasing eccentricity and growing outer vorticity gradient. The observed

rapid approach to circular symmetry has been termed "axisymmetrization." Numerous CS calculations have been performed to reproduce some of the results of [9] as well as to complement that study by examining initial conditions with relatively *small* eccentricities and *large* vorticity gradients. These initial conditions are difficult to compute for spectral models because (1) the time scale for axisymmetrization greatly increases for small eccentricities and large vorticity gradients, (2) the "arms" that are ejected tend to be very small and thin, and most importantly, (3) the author has found equilibria with small eccentricities and tight gradients that resist axisymmetrization in fully nonlinear calculations (in preparation).

A calculation is presented in which the initial condition *tends* to axisymmetrize but doubtfully does so completely. Four nested ellipses each of aspect ratio $\lambda = 2.5$ are arranged to resemble the continuous distribution $\omega(r) = 2\pi(1 - r^p)$, $p = 2$; the contour with vorticity $\omega(r)$ is the ellipse $x = r\cos\theta$, $y = \lambda^{-1} r \sin\theta$ for $0 \leqslant \theta < 2\pi$, and the discretization is such that, for N levels of vorticity, the jth contour has $r_j = (j/N)^{1/p}$, and vorticity jump $\tilde{\omega}_j = 2\pi/N$, $j = 1, ..., N$. The evolution of the vortex is shown in Fig. 11. Note also the diagnostic panels showing departures from conservation of circulation, angular momentum, energy, and centroid as well as the total number of nodes as a function of time. At $t = 7.3$, μ was increased to 0.05 in order to speed up the calculation at the risk of greater error, yet the error never exceeds 0.1 % in Γ, J, or E.

The evolution qualitatively parallels that computed spectrally [9] in several easily visible respects: (1) the initially rapid gradient intensification and loss of eccentricity of the central structure, (2) the connection of the "arms" with the cen-

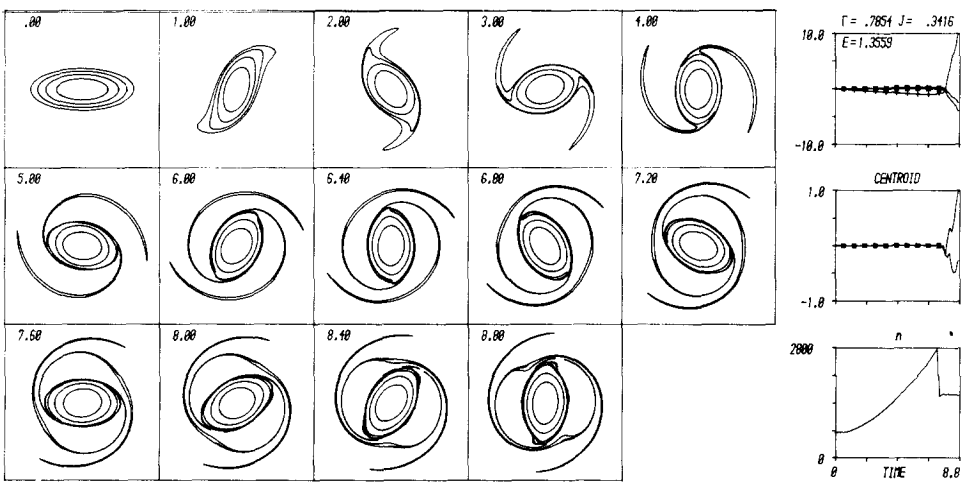

FIG. 11. The evolution of a nested, initially elliptical vortex. The algorithm parameters for this and subsequent figures are the same as used in case 1 of Table IV, and the diagnostic panels on the right are described in Fig. 5. μ was increased to 0.05 at $t = 7.3$.

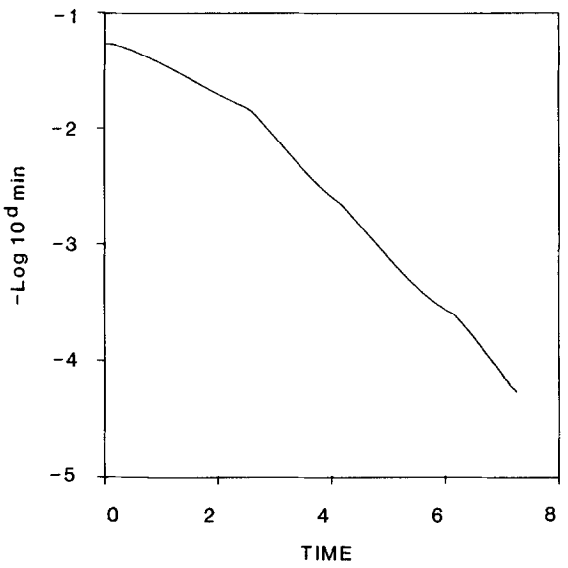

FIG. 12. The minimum distance between the two outer contours in Fig. 11 versus time.

tral structure, and (3) the generation of new "arms" resulting from (2). One aspect of axisymmetrization that lends itself ideally to CS is the (inviscid) limit of gradient intensification (in spectral calculations, gradient intensification is limited by high-order diffusion/model resolution). Consider the time evolution of the minimum distance between the two outer contours in Fig. 11 (see Fig. 12). The two contours rapidly approach each other, coming within one one-thousandth of their initial

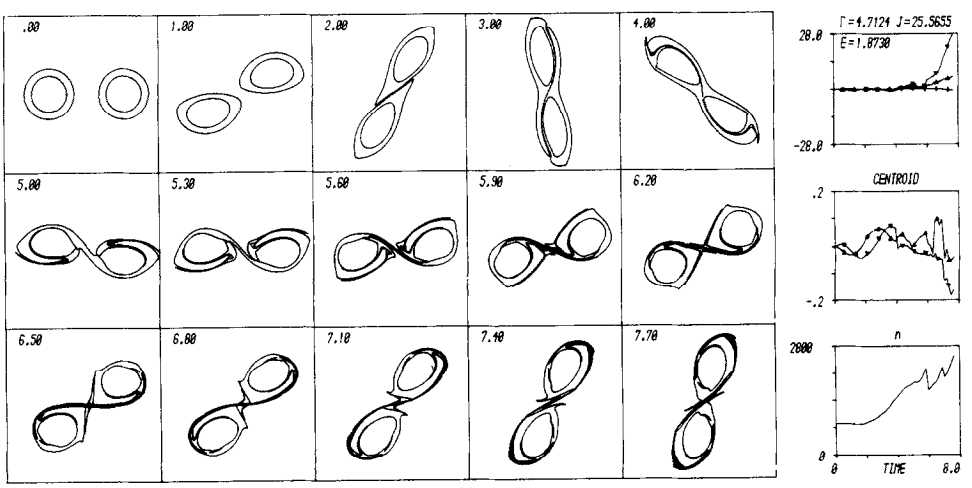

FIG. 13. Symmetric vortex merger, or partially. μ was increased to 0.05 at $t = 7$.

separation distance by $t = 7.25$. After this time, gradient intensification is largely abated, but likely begins again when the "arms" reconnect near $t = 9$ (surgery before this time prevented a simple calculation of intercontour distances.)

The second example of a nested-contour CS calculation also germinated in a recent study [11]. There, in part, spectral methods were used to examine the merger of two identical, initially circular and nonuniform vortices. Necessary and sufficient conditions for symmetric merger were obtained for the approximate system of two "elliptically desingularized" vortex patches [12] and generalised, with the support of several high-resolution spectral calculations, to nonuniform vortices. An approximate sufficient condition requiring no tendancy information at the initial time is given by

$$\sigma = \frac{\pi J}{\Gamma^2} \leqslant 11.4. \tag{14}$$

For two distant uniform circular vortices, $\sigma = (d/r)^2$ with d being the distance between vortex centres and r the radius of each vortex.

A calculation is presented which deliberately chooses σ close to the critical value for merger. The initial condition pits two identical vortices of unit radius separated by three radii from centre to centre. The vorticity distribution in each vortex is similar to that of the ellipse discussed above (a discretization of $\omega(r) = 2\pi(1 - r^2)$) with two contours. In Fig. 13, the two vortices repeatedly connect and disconnect until, at late times, a bridge persists between the two peaks of vorticity. It is suspected that the general dumbbell shape seen in the latter part of the calculation will persist indefinitely, particularly because this shape is part of a family of vortex equilibria (from unpublished work of the author). It is noted in [11] that a small

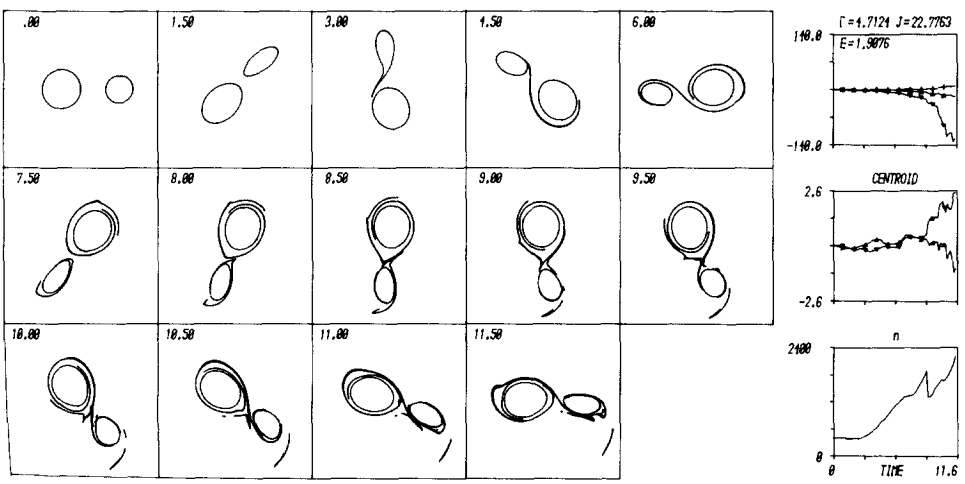

FIG. 14. Asymmetric vortex merger. μ was increased to 0.05 at $t = 8.5$.

amount of dissipation inevitably brings the centres of vorticity close enough together for "strong" (rapid) merger.

Little is known about the merger of unequal-sized vortices due to the greater variety of possibilities involved. For instance, a strong vortex might prefer to "shear out" a weaker partner rather than merge with it. Also the "strength" of the vortex depends on its concentration as well as its total circulation—a point vortex cannot be destroyed by a diffuse vortex of any strength. Consider the battle between two uniform vortices ($\omega = 2\pi$), one of unit radius and the other of radius $\sqrt{1/2}$, separated by a distance 3 from centre to centre, Fig. 14. The small vortex narrowly avoids being ripped apart and loses a small fraction of its area to the small strand of vorticity seen swimming around both vortices by $t = 6.5$. An enlarged region forms along this strand around $t = 7$ as a result of the negative strain field produced by the little-affected large vortex. Between $t = 7$ and 9.5, this enlarged region rolls up as a result of the shear outside the large vortex. Note that an enlargement on an isolated strip of vorticity would tend to roll up in the *opposite* direction, so it is clear that the roll up is dominated by shear. Between $t = 7.5$ and 10, the stagnated region between the two vortices captures and enlarges other parts of the filament. The V-shaped region at $t = 8.75$ is shortly thereafter divided between the two vortices. The part which is drawn close to the smaller vortex induces a region of high curvature on the small vortex by $t = 10.5$. Between $t = 10.5$ and 11.5, this region of high curvature amplifies significantly as it moves through a region of negatives along the contour strain and nearly "breaks" as it passes very close to the larger vortex. Several other similar events occur during the evolution, all of which appear to initiate in the regions of extreme strain and then to succomb to the strong shear around the vortices.

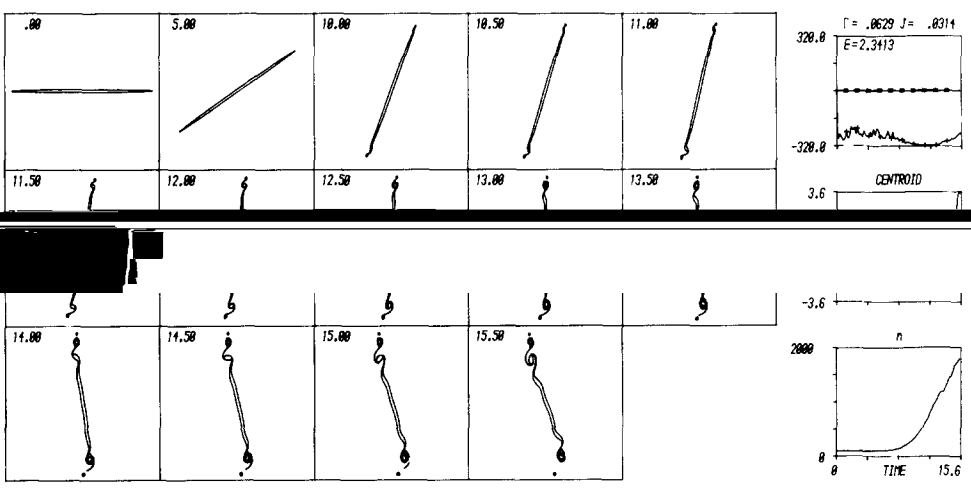

FIG. 15. The breakup of a thin strip of vorticity.

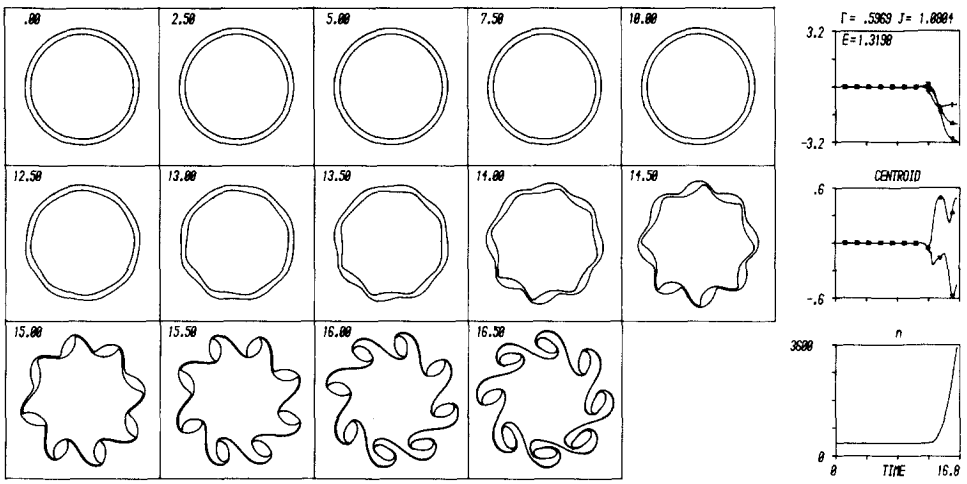

FIG. 16. The breakup of a thin annular band of vorticity.

This leads naturally to the question of the stability of thin bands of vorticity so often generated in calculations. Figures 15 and 16 show that numerical noise is sufficient to trigger the Kelvin–Helmholtz/Rayliegh instability on an isolated strip and ring of vorticity. However, despite great care (high resolution) to allow for such instabilities in calculations where the thin bands are not isolated, very little "instability" is ever seen. It is not the numerical scheme that is supressing these instabilities; one does observe the occasional "rollup," as in Figs. 5 and 14, but by and large mechanisms are operating to prevent the disintegration of thin strips. The mechanisms appear to be associated with the relatively slow growth rate of a disturbance, maximally 0.2ω (where ω is the vorticity in the strip), compared to the rate at which the large-scale flow is changing, ω. Strain is an important factor [7], but other factors such as shear and the proximity of the strip to a central structure may deter rollup as well.

4. CONCLUSIONS

Contour dynamics has been extended to systematically eliminate small scales of motion. As a CD calculation proceeds, the development of small scales requires rapidly increasing resolution requirements, and the calculation quickly becomes prohibitively expensive. Contour surgery, in effect, gives up on the small scales with the assumption that they have little dynamical importance and a negligible effect on the large-scale dynamics *for the duration of the calculation.* CS is not inherently dissipative, in the sense that inviscid invariants necessarily decrease; surgery, like spatial and temporal discretization, is simply another approximation of the dynamical equations that allows for their efficient solution. With little cost, errors

produced by surgery can be made comparable to errors in the CD-part of the algorithm.

However, CS, like other numerical algorithms, cannot be blindly driven beyond its limits of applicability. The nearly inevitable and incessant drive of an inviscid fluid to produce finer and finer scales of motion prevents any finite algorithm from accurately modelling even the largest scales of motion for arbitrarily long times. Considering just the effects of truncation, algorithms must (1) do something to remove the small scales of motion and (2) make sure that this removal process does not seriously affect the scales far separated from the truncation scale, a requirement that can only be satisfied for a finite time. In surgery, (1) vorticity features thinner than a prescribed scale are removed and (2) the surgical technique attempts to minimize its local effects on the flow.

To estimate the maximum duration of an "accurate" calculation, one must first define the error which is to be kept small. At the least, one wishes to correctly model the orientation and positions of the dominant vortex structures, and for this purpose, the accumulated error in the (non-zero) conserved quantities C (e.g., circulation, angular momentum, energy)

$$\varepsilon_C(t_m) = \Delta t \sum_{k=1}^{m} |C(t_k) - C(0)|, \tag{15}$$

where $t_k = k\,\Delta t$ and t_m is the duration of the calculation, should be kept small. As an example, with $C \equiv \Gamma^2/J$, C crudely representing the rotation rate of the flow (C

TABLE V

The Phase Error ε_C, in Degrees, for All the Calculations Presented in This Paper

Calculation	ε_C	t_m
Case from Table IV (Fig. 5)		
1	0.556	17.25
2	0.013	13.5
3	2.852	17.1
4	0.239	15.7
5	3.045	18.1
6	0.273	15.525
7	0.638	17.2
Figure		
11	0.058	8.85
13	0.015	7.75
14	0.053	11.5
15	0.525	15.65
16	0.023	16.7

Note. t_m is the duration of the calculation.

is the angular frequency in the case of a circular vortex), Table V lists the phase error ε_C (in degrees) versus t_m for all the calculations presented in this paper. In particular, note the sensitivity of ε_C to changes in the algorithm parameters for the seven cases in Table IV. In applications where the moderately fine scale structures are important, it may not be sufficient to keep ε_C small, and, in the end, one may have to resort to reproducibility at different resolutions.

In a concurrent study [5], the contour surgery algorithm is being used to examine the stability of sharp vorticity interfaces. There, in part, the smooth vortex boundaries of gently perturbed, linearly stable equilibria are shown to develop an apparently limitless degree of convolution. High resolution requirements are vital to capture this phenomenon.

The CS algorithm can be adapted to study flows in different geometries, by using a different Green's function in Eq. (1). For flow on a sphere [6], in fact the same Green's function may be used (except that the position vector x is three-dimensional), and the extension of the contour surgery algorithm is nearly trivial. It is also possible to study multi-layer quasi-geostrophic flow, and both studies aim to assess the importance of the geophysical environment in the vortex dynamics of the atmosphere.

ACKNOWLEDGMENTS

I am exceedingly grateful for the many useful discussions on contour dynamics/surgery with Professors Norman Zabusky, Ed Overman, and Mogens Melander. I am also indebted to my thesis supervisor, Dr. Ray Pierrehumbert, and to Dr. Darryl Holm for the inspiration I received from them.

REFERENCES

1. M. ABRAMOWITZ AND I. A. STEGUN, *Handbook of Mathematical Functions* (Dover, New York, 1980), p. 916.
2. G. S. DEEM AND N. J. ZABUSKY, *Phys. Rev. Lett.* **40**, 859 (1978).
3. G. S. DEEM AND N. J. ZABUSKY, in *Solitons in Action*, edited by K. Lonngren and A. Scott (Academic Press, New York, 1978), p. 277.
4. D. G. DRITSCHEL, *J. Fluid Mech.* **172**, 157 (1986).
5. D. G. DRITSCHEL, *J. Fluid Mech.*, in press.
6. D. G. DRITSCHEL, *J. Comput. Phys.*, in press.
7. D. G. DRITSCHEL, P. H. HAYNES, M. N. JUCKES, AND T. G. SHEPHERD, *J. Fluid Mech.*, in press.
8. H. LAMB, *Hydrodynamics* (Dover, New York, 1932), p. 232.
9. M. V. MELANDER, J. C. MCWILLIAMS, AND N. J. ZABUSKY, *J. Fluid Mech.* **178**, 137 (1987).
10. M. V. MELANDER, E. A. OVERMAN, AND N. J. ZABUSKY, Univ. of Pittsburgh Technical Report No. ICMA-86-93, 1986 (unpublished).
11. M. V. MELANDER, N. J. ZABUSKY, AND J. C. MCWILLIAMS, Army Research Office Report No. 86-1, 1986 (unpublished).
12. M. V. MELANDER, N. J. ZABUSKY, AND A. S. STYCZEK, *J. Fluid Mech.* **167**, 95 (1986).
13. E. A. OVERMAN AND N. J. ZABUSKY, *Phys. Fluids* **25**, 1297 (1982).

14. C. POZRIKIDIS AND J. J. L. HIGDON, *J. Fluid Mech.* **157**, 225 (1985).
15. H. M. WU, E. A. OVERMAN, AND N. J. ZABUSKY, *J. Comput. Phys.* **53**, 42 (1984).
16. N. J. ZABUSKY, *Ann. N.Y. Acad. Sci.* **373**, 160 (1981).
17. N. J. ZABUSKY, M. H. HUGHES, AND K. V. ROBERTS, *J. Comput. Phys.* **30**, 96 (1979).
18. N. J. ZABUSKY AND E. A. OVERMAN, *J. Comput. Phys.* **52**, 351 (1983).
19. Q. ZOU, E. A. OVERMAN, H. M. WU, AND N. J. ZABUSKY, University of Pittsburgh, Pittsburgh, PA, private communication (1987).