JOURNAL OF COMPUTATIONAL PHYSICS 95, 101–116 (1991)

# Timestepping Lagrangian Particles in Two Dimensional Eulerian Flow Fields

DAVE RAMSDEN

*Daleth Research, 1255 Oscar Street,*
*Victoria, British Columbia, Canada V8V 2X6*

AND

GREG HOLLOWAY

*Institute of Ocean Sciences, Victoria, British Columbia, Canada V8L 4B2*

Received September 7, 1989; revised March 9, 1990

The speed and accuracy of different methods for the interpolation and timestepping of Lagrangian particles in Eulerian velocity fields are examined. Two circumstances are considered: (1) a steady flow field in which particles trace closed orbits along constant streamfunction values; and (2) decaying two-dimensional turbulence in which vorticity is conserved within a correction for dissipation. A fourth-order Runge–Kutta timestepping is shown to give best results for the steady flow field. Tracking an invariant along particle paths in a fully eddy active case is shown to be relatively insensitive with regard to choice of interpolation or timestepping method. The cause of this insensitivity is shown to be a resolution problem of the Eulerian field dynamics and methods are outlined to ameliorate the problem. © 1991 Academic Press, Inc.

## INTRODUCTION

The extensive use of drifters in oceanic experiments creates large volumes of data which can be compared to drift tracks derived from numerical ocean circulation models. Flow parameters such as eddy diffusion coefficients and eddy kinetic energy per unit mass can be estimated from drifter statistics (e.g., Rossby *et al.* [1] and Freeland *et al.* [2]). If practical inferences are to be drawn from statistics derived from particles in a numerical model, it is important to estimate the efficiencies and accuracies of various interpolation and stepping methods.

This paper examines these issues in the context of two-dimensional turbulence modelled by the dealiased spectral transform method after Orszag [3]. The vorticity evolution equation is solved on a periodic $(x, y)$ rectangular domain:

$$\frac{\partial \zeta}{\partial t} + J(\psi, \zeta) = -D_\zeta \qquad (1)$$

0021-9991/91 $3.00

Here $\zeta$ is the vertical component of vorticity defined as $\zeta = \nabla^2\psi$, $\psi$ is streamfunction such that $U = -\partial\psi/\partial y$, $V = \partial\psi/\partial x$, and $\nabla^2$ is the horizontal Laplacian in $x, y$. $D_\zeta$ is dissipation and $J$ denotes the Jacobian determinant $J(A, B) = \partial_x A \cdot \partial_y B - \partial_x B \cdot \partial_y A$.

Spatial derivatives are calculated by a spectral transform method in which $\zeta$ is expressed as a truncated Fourier expansion, giving periodic boundary conditions in $x$ and $y$. The field is expanded on a set of wavenumbers $\mathbf{k} = (k_x, k_y)$, $k^2 = k_x^2 + k_y^2$. Timestepping of the Eulerian field is accomplished by a leapfrog scheme except for dissipation which is calculated exactly. For operator $D_\zeta$, conventional Laplacian dissipation $\nu\nabla^2\zeta$ is taken, $\nu$ being a numeric constant. The flow field is expressed on $N \times M$ grid points, with $N$ and $M$ being powers of 2 (not necessarily the same).

Dynamics associated with Eq. (1) conserve vorticity along particle paths in the absence of dissipation. For the gridded non-continuous case, kinetic energy and mean square vorticity are conserved by the spectral method implementation of Eq. (1). It is not guaranteed that vorticity will necessarily be conserved along particle paths but we will use this as our test invariant. Given this limitation the questions to be answered are:

1. What is the best way to interpolate flow properties (primarily velocity) from grid points to the particle positions?
2. What is the best way to timestep the particle positions?

Related issues have been examined by Haidvogel [4] in the context of a forced, dissipated system on a $\beta$-plane in which the nonconservation of vorticity was separated into dissipation, forcing and aliasing terms. He concluded that following Lagrangian particles would be very difficult due to aliasing errors. These errors were found to be dependent on model resolution and kinetic energy spectral rolloff rates. The more important issue of contamination of particle *statistics* was also examined. This was done by calculating averaged particle separations as functions of model resolution and Haidvogel concluded that particle statistics will be dependent on model resolution.

Given these fundamental limitations, the aim here is to explore differing methods under controlled conditions to find the best ways of implementing Lagrangian particles.

## Spatial Interpolation

Consider an Eulerian field $\Phi(x_i, y_j)$ defined on a square grid $(x_i, y_j\colon i, j = 1, N)$ with corresponding Fourier representation $\phi(k_x, k_y)$ defined over the truncated spectral domain $k^2 < (N/2)^2 = k_t^2$. Corresponding intergrid values $\Phi(x, y)$ can be found by the slow Fourier transform (SFT),

$$\Phi(x, y) = \sum_{k_x, k_y} \phi(k_x, k_y) \cdot e^{\iota(k_x x + k_y y)}, \tag{2}$$

where the summation is over all Fourier modes.

$\Phi(x, y)$ can also be approximated by using a polynomial fit to nearby grid points:

$$\Phi(x, y) \simeq \sum_{i=1}^{m} \sum_{j=1}^{m} \Phi(x_i, y_j) A_{i,j} \prod_{l=1, l \neq i}^{m} \prod_{k=1, k \neq j}^{m} (x - x_l)(y - y_k), \tag{3}$$

where $m$ is the number of grid points in the fit $m = 2, 4, 6, 8, \ldots$, the order of fit $o$ being one less than $m$. $A_{i,j}$ is given by

$$A_{i,j} = \left( \prod_{l=1, l \neq i}^{m} \prod_{k=1, k \neq j}^{m} (x_i - x_l)(y_j - y_k) \right)^{-1}. \tag{4}$$

Because of our use of periodic boundary conditions, $o$ can be chosen to be as large as one wishes. We explore these cases, up through $o = 7$, in order to evaluate the dependence of timing and accuracy on $o$. In many applications where domain boundaries will be present, lower orders (as $o = 1$) might be chosen to avoid influence of points lying outside the flow domain. Higher order fits in the interior can be reduced in order (toward $o = 1$) as the particle approaches the boundary.

If it is desired to interpolate a velocity field, Eq. (3) can be used in one of two ways. $U$ or $V$ can be formed in Fourier space and interpolated by Eq. (3) (the "Fourier" method), or Eq. (3) can be differentiated with respect to $y$ or $x$ so that $U$ or $V$ can be calculated from $\psi$ (the "streamfunction" method), e.g.,

$$U(x, y) = -\partial_y \psi \simeq -\sum_{i=1}^{m} \sum_{j=1}^{m} A_{i,j} \psi(x_i, y_j) \times \sum_{l=1, l \neq j}^{m} \prod_{k=1, k \neq i}^{m} \prod_{p=1, p \neq j, l}^{m} (x - x_l)(y - y_p). \tag{5}$$

Taking the SFT as a reference, the polynomial methods can be compared directly to it. Two velocity fields were defined from a random phase Fourier initialization, one with a $k^{-3}$ power spectrum and the other with a $k^{-1}$ spectrum. Thirty-six

TABLE I

Relative Times and Accuracies of Interpolation Methods

| | Fourier | | | Streamfunction | | |
|---|---|---|---|---|---|---|
| Method | Time (s) | $E(k^{-3})$ | $E(k^{-1})$ | Time (s) | $E(k^{-3})$ | $E(k^{-1})$ |
| Poly, $m = 2$ | 0.01 | $0.996 \times 10^{-2}$ | $0.124 \times 10^{0}$ | 0.02 | $0.341 \times 10^{-1}$ | $0.205 \times 10^{0}$ |
| Poly, $m = 4$ | 0.05 | $0.401 \times 10^{-2}$ | $0.688 \times 10^{-1}$ | 0.10 | $0.646 \times 10^{-2}$ | $0.111 \times 10^{0}$ |
| Poly, $m = 6$ | 0.10 | $0.282 \times 10^{-2}$ | $0.533 \times 10^{-1}$ | 0.40 | $0.452 \times 10^{-2}$ | $0.896 \times 10^{-1}$ |
| Poly, $m = 8$ | 0.21 | $0.236 \times 10^{-2}$ | $0.464 \times 10^{-1}$ | 1.19 | $0.371 \times 10^{-2}$ | $0.722 \times 10^{-1}$ |
| FFT(1) | 0.43 | — | — | — | — | — |
| SFT(36) | 6.07 | — | — | — | — | — |

particles were released, this number being chosen to permit statistical averaging. Average relative error $E$ for 36 velocities interpolated on these fields is

$$E=\sum_{i=1}^{36}(U_i-U_s)^2\Big/\sum_{i=1}^{36}U_s^2,$$

where $U_i$ and $U_s$ are the interpolated and SFT velocities, respectively. Table I summarizes the computer times taken for the streamfunction, Fourier, and SFT methods with the corresponding accuracies. These interpolations were all performed at grid resolution $64^2$ on a VAX785 computer (single precision). The times shown for the polynomial interpolations are for the 36 interpolations only. Also shown is the time for one $64^2$ fast Fourier transform (FFT) which must be included in the total times for the interpolation method. The times for the SFT are for the 36 SFTs.

The interpolation times for the polynomial fits are roughly quadratic in order of fit for the Fourier method. One might expect from Eq. (3) that the times would be quartic in number of fit grid points $m$, but terms can be combined and the calculations become relatively more efficient at the higher orders.

Unless the number of floats is very small the SFT will be slower than any of the polynomial fits. The accuracy of the fits are dependent on the order of fit and the spectral form of the field to be interpolated. The Fourier method of velocity interpolation is more accurate than the streamfunction method in all cases. Streamfunction method interpolations are slower than the Fourier method but the overall timing comparison depends on how often the FFT must be called. One FFT will suffice to calculate $\psi$, $U$, and $V$, using the streamfunction method whereas three FFTs are needed for the Fourier method. We will use the Fourier method of polynomial interpolation from now on unless otherwise noted.

Table I also illustrates that a kinetic energy field with a geophysically plausible energy spectrum ($k^{-3}$) will yield velocities which can be interpolated to about 0.02% error with $m=8$ whilst the corresponding vorticity field ($k^{-1}$) can be interpolated to about an accuracy of 5% with the same fit parameters. These results are also applicable to semi-Lagrangian methods (e.g., Robert [5]).

## Timestepping Methods in Frozen Streamfunction Fields

The Eulerian model provides timestepped fields at regular time increments. It shall be assumed that a constant timestep interval $\delta t$ is desired for integration of the particle positions and this timestep will be the same as the Eulerian model. Consider a Lagrangian particle at time $t$ to have position $\mathbf{x}=(x,y)$ and velocity $\mathbf{u}=\mathbf{u}(\mathbf{x})=(U,V)$ calculated by one of the above methods. The Lagrangian timestepping methods considered are:

1. Euler forward step,

$$\mathbf{x}(t+\delta t)=\mathbf{x}(t)+\mathbf{u}(t)\,\delta t. \tag{6}$$

2. Huen (predictor-corrector),

$$\begin{aligned} \mathbf{x}^*(t+\delta t) &= \mathbf{x}(t) + \mathbf{u}(t)\,\delta t \\ \mathbf{x}(t+\delta t) &= \mathbf{x}(t) + 1/2(\mathbf{u}(t) + \mathbf{u}^*(t))\,\delta t, \end{aligned} \tag{7}$$

where $\mathbf{u}^*$ is calculated at intermediate advanced point $\mathbf{x}^*$ on the advanced Eulerian field.

3. Fourth-order Runge–Kutta (Press *et al.* [6]),

$$\begin{aligned} &\mathbf{x}_1 = \mathbf{x} + \mathbf{u}\,\delta t/2 \\ &\mathbf{u}_2 = \mathbf{u}(\mathbf{x}_1), \qquad \mathbf{x}_2 = \mathbf{x} + \mathbf{u}_2\,\delta t/2 \\ &\mathbf{u}_3 = \mathbf{u}(\mathbf{x}_2), \qquad \mathbf{x}_3 = \mathbf{x} + \mathbf{u}_3\,\delta t \\ &\mathbf{u}_4 = \mathbf{u}(\mathbf{x}_3) \\ &\mathbf{x}(t+\delta t) = \mathbf{x}(t) + \delta t(\mathbf{u} + 2\mathbf{u}_2 + 2\mathbf{u}_3 + \mathbf{u}_4)/6, \end{aligned} \tag{8}$$

where $\mathbf{u}_2$ and $\mathbf{u}_3$ are evaluated at time midpoint (average the two Eulerian fields) and $\mathbf{u}_4$ is calculated at the advanced Eulerian field.

4. Robert filtered leapfrog,

$$\mathbf{x}(t+\delta t) = \mathbf{x}(t-\delta t) + 2\mathbf{u}(t)\,\delta t, \tag{9}$$

where the position at time $t$ is then time filtered (Robert [7]),

$$\mathbf{x}^f(t) = \mathbf{x}(t) - R(\mathbf{x}(t+\delta t) - 2\mathbf{x}(t) + \mathbf{x}^f(t-\delta t))$$

with $R$ the Robert filter weight $0 \leqslant R < 0.25$ and superscript $f$ indicates a filtered field. The first timestep calculation is done with a Huen step.

5. Adams–Bashforth,

$$\mathbf{x}(t+\delta t) = \mathbf{x}(t) + 0.5(3\mathbf{u}(t) - \mathbf{u}(t-\delta t))\,\delta t, \tag{10}$$

where the initial value of $\mathbf{u}(-\delta t)$ is calculated by taking an Euler backward step at timestep 0 and computing the velocities at those locations.

6. Bennett–Clites [8],

$$(x^{n+1} - x^n)/\delta t = U + 0.5(\partial U/\partial x)(x^{n+1} - x^n) + 0.5(\partial U/\partial y)(y^{n+1} - y^n) \tag{11}$$

$$(y^{n+1} - y^n)/\delta t = V + 0.5(\partial V/\partial x)(x^{n+1} - x^n) + 0.5(\partial V/\partial y)(y^{n+1} - y^n), \tag{12}$$

where all derivatives are calculated at timestep $n$. Equations (11) and (12) yield $x^{n+1}$ and $y^{n+1}$.

If, instead of integrating Eq. (1), the Eulerian motion field $\psi$ is held constant, then

TABLE II

Relative Times and Polynomial Accuracies of the Particle Stepping Methods

| Method | Time (s) | $m=2$ | $m=4$ | $m=8$ | Analytic |
|---|---|---|---|---|---|
| Euler | 1.25 | — | — | 0.052 | 0.052 |
| Leapfrog | 1.25 | — | — | $2.5\times10^{-8}$ | $2.5\times10^{-8}$ |
| Adams–Bashforth | 1.25 | — | — | $1.6\times10^{-8}$ | $1.6\times10^{-8}$ |
| Huen | 2.60 | $9.0\times10^{-7}$ | $2.0\times10^{-9}$ | $2.0\times10^{-9}$ | $2.0\times10^{-9}$ |
| Bennett–Clites | 3.24 | — | — | $2.7\times10^{-9}$ | $2.3\times10^{-9}$ |
| Runge–Kutta | 5.46 | — | — | $8.5\times10^{-12}$ | $5.5\times10^{-12}$ |

particles in such a "frozen field" should trace closed orbits along lines of constant streamfunction. To test the timestepping methods a streamfunction of the form

$$\psi(x, y)=\sin(x)\sin(y), \qquad 0\leqslant x, y\leqslant 2\pi$$

was used. This choice gives us the opportunity of separating interpolation from stepping errors as all derivatives in Eqs. (6) to (12) can be evaluated analytically. Thirty-six particles were initialized at random locations and integrated until several orbits were completed. The timestep interval was chosen so that the fastest particles will advance about one-half grid point each timestep.

The left side of Table II shows the relative times taken for the timestepping methods when interpolations were used. These times reflect the amounts necessary to calculate all necessary FFTs, interpolation and timestepping of 36 particles.

Relative stream function error is defined in the same way as for the interpolation tests.

$$E(t)=\sum_{i=1}^{36}(\psi_i-\psi_{0,i})^2\Bigg/\sum_{i=1}^{36}\psi_{0,i}^2,$$

Where $\psi_{0,i}$ is the initial stream function value and $\psi_i$ is the measured stream function at time $t$. Figure 1 shows the stream function errors versus integration time for the Huen, Runge–Kutta, and Bennett–Clites timesteps. Integration time is the sum of the timesteps $\delta t$.

The top left of Fig. 1 also shows a comparison of the order of fit as the dashed line uses $m=8$ and the dotted line (barely visible) $m=4$. The top right uses Huen step with $m=2$.

The bottom left of Fig. 1 shows the streamfunction (dotted) and Fourier (dashed) methods of calculating $u$ with $m=8$ compared to the analytic method. The right side of Table II summarizes the accuracies of the polynomials, stepping methods, and interpolation. As is seen there is not much difference between the results for $m=4$ and $m=8$, but there is quite a difference between $m=2$ (bilinear) and $m=4$ (bicubic) for this case. The Fourier method of calculating $\mathbf{u}$ performs marginally

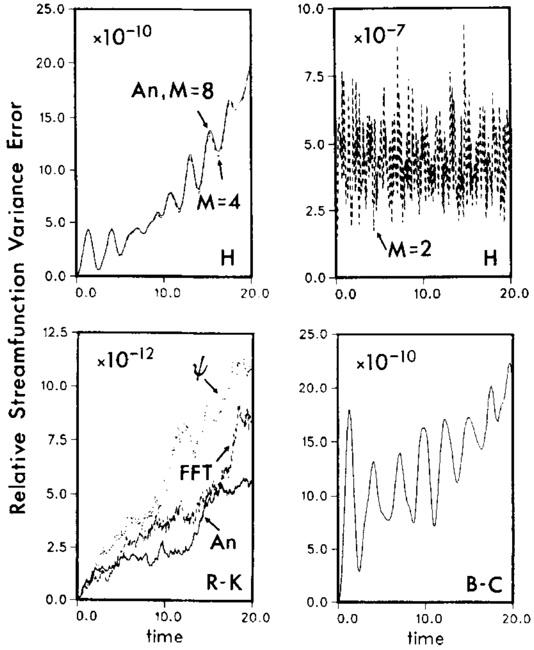

FIG. 1. The average relative streamfunction errors versus integration time for the frozen field case. All the solid curves in this and Fig. 2 use exact analytic forms of derivatives. Timestepping methods are Huen (top), fourth-order Runge–Kutta (bottom left), and Bennett–Clites (bottom right). See text for a description of the other curves.

better than the streamfunction method and interpolation errors can only be seen in the most accurate timestepping methods.

Figure 2 shows Adams–Bashforth, Euler, and leapfrog timestepping with and without a Robert filter. Also shown (top left) are the results using the SFT (dashed) and $m = 8$ polynomial (dotted, barely visible). The SFT is obviously no better than polynomial interpolation for this case.

To no surprise the accuracy of the timestepping methods goes as the theoretical order of accuracy. Runge–Kutta (fourth-order accurate) is best and Euler (first-order) is worst. The leapfrog method with no filtering has an oscillatory mode (note the thickness of the error curve) and is comparable in accuracy to the Adams–Bashforth (both are second-order accurate). The addition of a Robert filter makes the leapfrog method worse in accuracy but suppresses the oscillation.

Leapfrog timestepping is widely used for integration of the Eulerian field equations due to its efficiency and accuracy (Haltiner and Williams [9]). In part this is because the computational (oscillatory) mode is neutrally stable with respect to advection or wave propagation. One might think that the advantages of leapfrog should carry over to Lagrangian particle advection, but this is not so. Although the

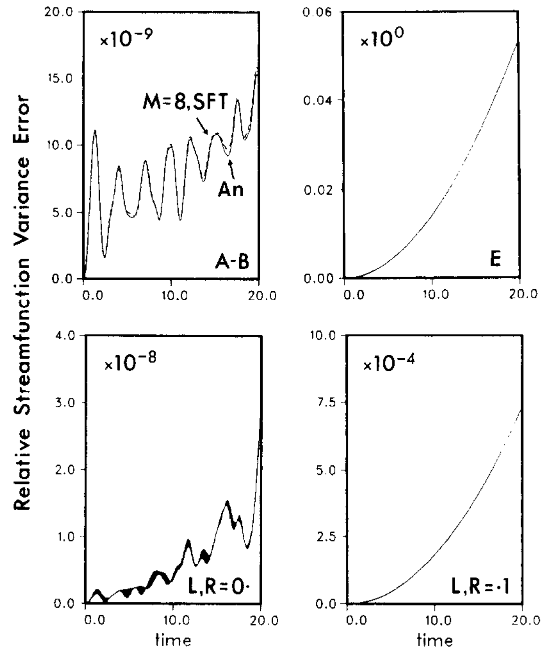

FIG. 2. Streamfunction errors as Fig. 1 for Adams–Bashforth (top left), Euler (top right), and leapfrog (bottom) with Robert filter weights of 0. (left) and 0.1 (right).

equation of motion for Lagrangian particles is only "advection," the operator is different and has very different stability properties. Consider a simple Eulerian flow:

$$\psi = xy; \qquad \text{hence } U = -x \text{ and } V = y.$$

Although the motion is a simple advection, components of particle position satisfy

$$\frac{dx}{dt} = -x; \qquad \frac{dy}{dt} = y,$$

i.e., "friction"-type equations for which leapfrog is unconditionally unstable. The particular stagnation flow illustrated here may not occur in any particular Eulerian field simulation; rather, the point is that a method which may be quite suitable for Eulerian advection can fail dramatically for Lagrangian advection.

## TIMESTEPPING METHODS IN AN EVOLVING VORTICITY FIELD

The frozen field case has both a strength and a weakness in that the motion fields are not timestepped. Errors will not be introduced into the particle calculations by

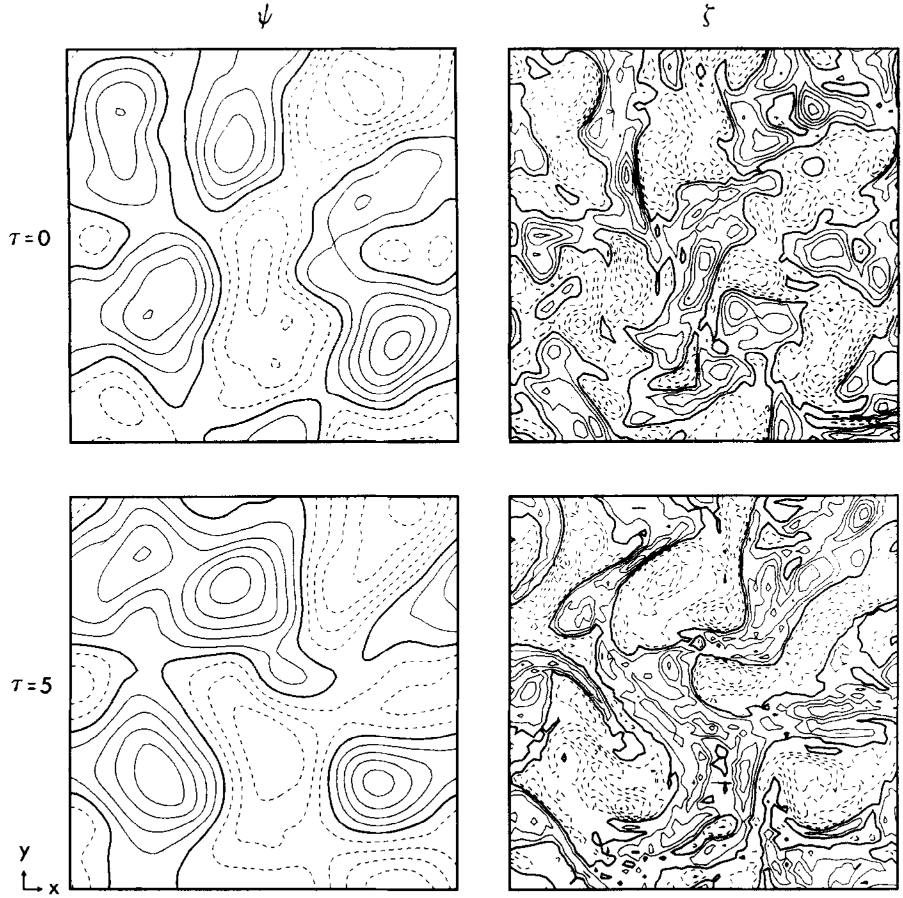

FIG. 3. The initial (top) and final (bottom) streamfunctions (left) and vorticities (right) for vorticity following experiments.

stepping the fields. In practice, models do have to be timestepped and the resultant effects on particles may be important.

In this section the conservation of vorticity on particle paths in the presence of an evolving Eulerian flow field is considered. Comparison runs were performed on an identical flow field at grid size $64^2$ with particles released at identical points and times while altering only the particle interpolation and stepping methods. Figure 3 shows the start and end streamfunction and vorticity fields for these cases; Fig. 4 shows the start and end energy spectra. In these figures and in the following text, time $t$ is non-dimensionalized by r.m.s. vorticity as $\tau = \zeta_{\text{rms}} t$.

In order to prepare a satisfactory stream function field to initialize the runs, forcing was added to the Eulerian evolution equation, Eq. (1). The form of the forcing used to bring about statistical stationarity was an isotropic, random phase, low wavenumber spectrum. A Robert filter was applied to the Eulerian leapfrog

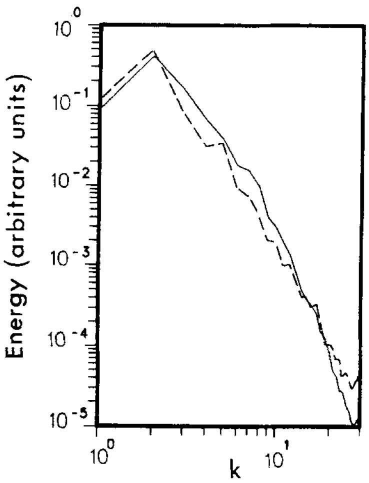

FIG. 4. The initial (solid) and final (dashed) energy spectra plotted versus total wavenumber for the cases shown in Fig. 3.

timestepping to suppress the leapfrog mode introduced by the forcing. The runs shown in Fig. 3 and 4 were initialized from that field but included no forcing or Eulerian Robert filering. They were hence viscous rundown experiments with enstrophy (mean square vorticity) decaying about 25% over the course of the experiments ($5\tau$ time units).

From the analytic description of the dissipation, values of $\nu\nabla^2\zeta$ can be calculated and interpolated to particle positions. Initial particle vorticities $\zeta(0, x_0, y_0)$ can be corrected for dissipation at each timestep yielding the theoretical decayed vorticity $\zeta'(t, x, y)$. This can be compared to the vorticity calculated at the particle $\zeta(t, x, y)$ yielding vorticity error $\varepsilon_i(t)$ at particle $i$ given by

$$\varepsilon_i(t) = \zeta_i(t, x, y) - \zeta'_i(t, x, y) \tag{13}$$

$$\zeta'_i(t, x, y) = \zeta_i(0, x_0, y_0) + \nu \int_0^t \nabla^2 \zeta_i(t', x', y')\, dt'; \tag{14}$$

$x'$ and $y'$ are at the particle position at time $t'$ and $x_0, y_0$ are initial positions. This error is then expressed as

$$E(t) = \sum_{i=1}^{36} \varepsilon_i(t)^2 \Big/ \sum_{i=1}^{36} \zeta_i^2.$$

## VORTICITY ERRORS DUE TO INTERPOLATION METHOD

Thirty-six particles were released in a regular $6 \times 6$ grid into the turbulence field and the particle parameters were calculated using either the SFT or polynomial fits.

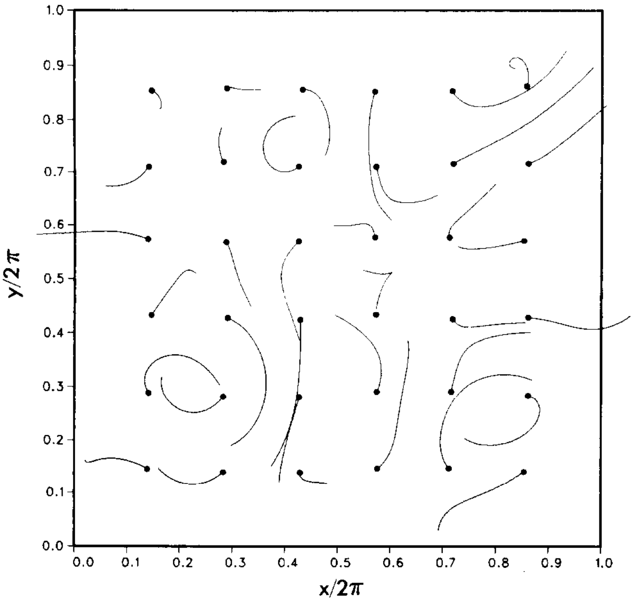

FIG. 5. The float tracks for particles placed in the evolving flows. The case shown utilized $m = 8$ and a Huen timestep. The small dots represent the release points.

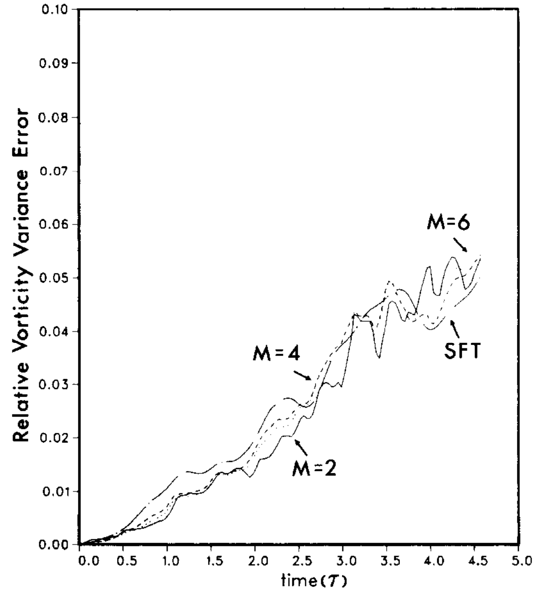

FIG. 6. The relative enstrophy error of the evolving flow field plotted versus $\tau$ for polynomial fits with $m = 2$ (solid), $m = 4$ (dashed), and $m = 6$ (dotted), and the slow Fourier transform (broken solid). All used Huen timestepping.

Figure 5 shows the particle track paths for a run using a polynomial fit parameter of 8. The timestepping method used for the case shown in Fig. 5 was the Huen.

Figure 6 illustrates the average relative vorticity errors associated with the SFT and three values of interpolation order. As is seen in Fig. 6, it is difficult to discern much difference between interpolation methods and an order of fit of 1 ($m=2$) is as good as higher orders of fit or the SFT.

## Vorticity Errors Due to Timestepping Method

Using an interpolation with $m=4$ on the same flow field as used in the previous section, Fig. 7 shows the average relative vorticity errors versus time for four of the timestepping methods. The same plot for the other stepping methods cases looks almost the same. It appears that the interpolation and timestepping methods for particles in 2D turbulence are largely irrelevant. The error growth that is independent of timestepping method exceeds any differences which may be obtained from ensemble averaging of the individual methods.

In summary we observe that using realistic Eulerian parameters from an eddy active model, an average of 1% of particle vorticity variance per $\tau$ is not followed for this case and no interpolation or timestepping method can minimize this loss.

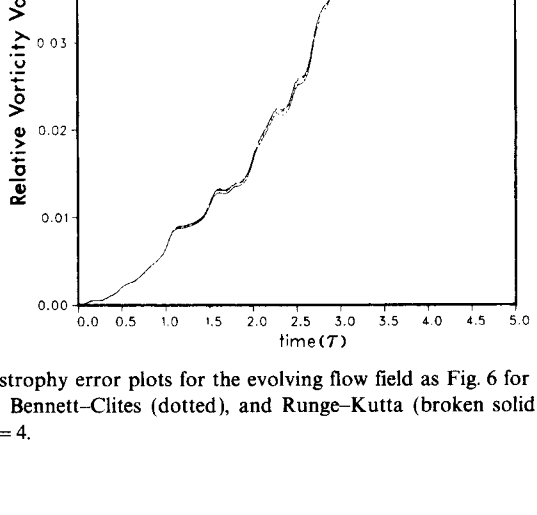

FIG. 7. Relative enstrophy error plots for the evolving flow field as Fig. 6 for the Euler (solid), leapfrog ($R=0.$) (dashed), Bennett–Clites (dotted), and Runge–Kutta (broken solid) timesteps. All used a polynomial fit with $m=4$.

## Origin of the Vorticity Error

Haidvogel [4] anticipated the difficulties in particle following experienced in the previous section. Vorticity conservation following a particle is a property of the continuum Eulerian field equations. Truncating the Fourier representation of the Eulerian model at some cutoff wavenumber should not be expected to retain

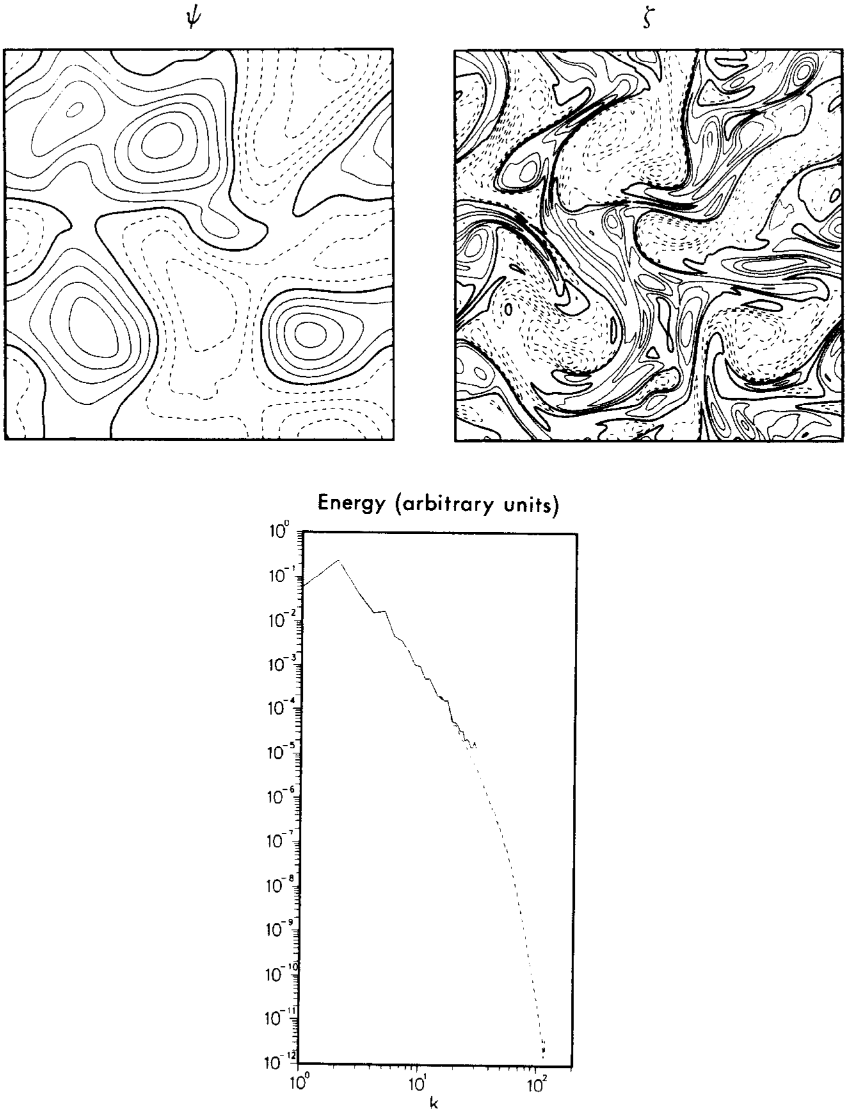

FIG. 8. The final streamfunction (left), vorticity (center), and energy spectrum (right dashed) for the $256^2$ case. The solid curve in the energy spectrum plot is for the end of the $64^2$ case.

Lagrangian information (only a subset of properties, usually energy and enstrophy, of the Eulerian equations are retained by various means). This problem arises from the Jacobian $J(\psi, \zeta)$ which, in absence of cutoff, would project to higher wavenumbers. The impact of omitting such interactions can be explored as follows:

Consider what happens when the simulations of the previous section performed at $64^2$ are repeated at resolutions $128^2$ or $256^2$. Figure 8 shows the end state streamfunction, vorticity and energy spectra for the $256^2$ case after $\tau = 4.5$. Also overlayed on the energy spectra plot is the result from the $64^2$ case. Visually, Fig. 8 and the end states of Fig. 3 are similar at the large scales but differ in the small scale behaviour.

The vorticity error plots for the higher resolution cases are shown in Fig. 9 and the improvement from the previous section is dramatic. Also shown in Fig. 9 are the vorticity errors from cases started from mature $128^2$ and $256^2$ start states. The $128^2$ case initialized from the $64^2$ field tracks the floats quite well until about $\tau = 1$, at which time the error slope becomes about equal to the other $128^2$ case. This is the time it has taken for vorticity variance to move into the upper half of Fourier space and for particle following errors to appear. Correspondingly, the $256^2$ case started from a $64^2$ mature endstate takes about $\tau = 2$ time units to fill out.

The cause of the vorticity error in the $64^2$ case is a resolution problem. Nonlinear dynamics tend to transfer vorticity variance into smaller scales. If the smaller scales do not exist, those wavenumber interactions are suppressed and *the physics is not adequately represented.* Haidvogel recommended that the upper half of wavenumber space be depleted of energy to ensure the alias error be insignificant. In practice, as

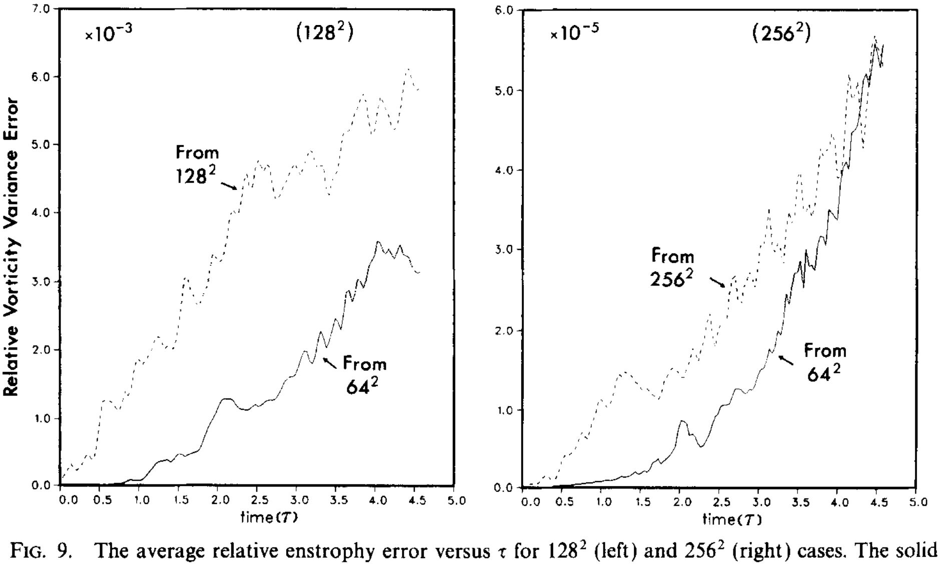

FIG. 9. The average relative enstrophy error versus $\tau$ for $128^2$ (left) and $256^2$ (right) cases. The solid lines represent cases started from the $64^2$ field and the dashed lines represent cases begun from mature states at each resolution.

time progresses, the wavenumber spectrum will fill out and similar truncation errors will then appear at the smaller scales. As seen in Fig. 9 there will still be (reduced) error growth unless there is sufficient damping.

Defining a truncation error to be $\Sigma(J_2 - J_1)^2/\Sigma J_2^2$, where $J_1$ is the Jacobian performed at resolution $64^2$ and $J_2$ is performed at twice the resolution with the summation over all real space, the value obtained is 0.03, which is of the same order as the error growth rate $(0.015/\tau)$ in the $64^2$ cases. The truncation error was calculated at the start of the simulation when the upper half of wavenumber space in the $128^2$ case was empty.

To relate vorticity nonconservation to the parameters of the Eulerian model, one may form a "truncation" (or "grid-scale") Reynolds number $Re = u_t \cdot l_t/\nu = \zeta_{\mathrm{rms}}/\nu \cdot k_t^2$, where $k_t$ is the truncation wavenumber, $l_t = 1/k_t$, and $u_t = \zeta_{\mathrm{rms}} \cdot l_t$. Then it is seen that the error variance growth rates from Fig. 6 and 9 (rates 0.015, 0.002, and 0.00002 per $\tau$) decrease with decreasing grid-scale Reynolds number.

One might regard vorticity nonconservation by 10% or more in unit nondimensional time $\tau$ (i.e., $\sqrt{0.015}$ from Fig. 6) as a disturbing loss of skill. However, this should also be interpreted from the view of how large is a typical error in particle position. For the three relative error variance growth rates given above, and given $\zeta_{\mathrm{rms}} \approx 5$ in each case, the r.m.s. vorticity errors after unit time are about 0.5, 0.2, and 0.03. For each of the cases, r.m.s. gradients of $\zeta$ fall in the range 20 to 30. Thus actual positional errors are typically 0.03, 0.01, and 0.001. The grid spacing for these three cases is $\pi/k_t$, roughly 0.1, 0.05, and 0.025, so that positional errors expressed as a fraction of grid spacing are 0.3, 0.2, and 0.04. In terms of accurate position information following the particles, the error appears to be significantly less than one grid space after unit nondimensional time even for grid-scale *Re* somewhat larger than unity. Greater skill at particle following can be achieved by executing at smaller truncation *Re*.

## Conclusions

Errors due to the interpolation of a real space field by polynomial fitting are dependent on the spectral form of the field and the order of fit. Evaluating velocities from interpolated streamfunction fields is not as accurate as forming velocities in Fourier space and then interpolating directly.

A frozen (steady) flow field in which derivatives can be evaluated analytically has been used to demonstrate that an Euler forward step is a poor choice for a timestepping method while a fourth-order Runge–Kutta scheme gives best stream function following errors. Errors due to interpolation are only observable for the most accurate timestepping methods. A Robert filter tends to degrade the performance of leapfrog timestepping for the Lagrangian particles; however, Lagrangian leapfrog timestepping may exhibit an unconditionally unstable computational mode under circumstances where the leapfrog is neutrally stable in the Eulerian field equations. For the frozen streamfunction field, an order of fit 3 gives almost identical results to one of 7, while an order of fit 1 is not as accurate.

For evolving fields in which the subgrid Reynolds number is of order unity, the choice of interpolation and timestepping is largely irrelevant due to the presence of errors introduced by finite resolution of the 2D Eulerian dynamics. These vorticity following errors can be limited by increasing model resolution or viscosity. Rough estimates of vorticity error growth predictions for an existing model can be made by calculating an untruncated Jacobian and comparing it to the truncated Jacobian.

To include particles in an eddy active flow, the modeller is faced with choices. Depending upon the intended accuracy to which positions or vorticity balances at particles are required, either (1) parameters of the Eulerian model may need to be constrained, or (2) model resolution may have to be increased for any given set of flow parameters. In many applications, the major source of error for particles will be determined by truncation of nonlinear interactions on account of finite resolution. When particle accuracy is so limited by resolution, then relatively crude (but inexpensive) interpolation and timestepping methods may be used. The errors incurred by limited resolution can be most dramatic in terms of inaccurate vorticity balances following particles while positional errors are relatively small. Improving the vorticity balances requires that small scale structures of the Eulerian flows be more strongly damped, a criterion that corresponds to setting a grid-scale Reynolds number to values rather less than unity.

## Acknowledgments

The authors thank Ms. Susan Lozier for a careful reading of a draft version of the paper. This work was supported in part under ONR Contract N000014-87-G-0262.

## References

1. H. T. Rossby, S. C. Riser, and A. J. Mariano, *Eddies in Marine Science* (Springer-Verlag, Berlin, 1983), p. 66.
2. H. J. Freeland, P. B. Rhines, and H. T. Rossby, *J. Mar. Res.* **33**, 383 (1975).
3. S. A. Orszag, *Stud. Appl. Math.* **51**, 253 (1971).
4. D. B. Haidvogel, *Ocean Model.* **45** (1982), to appear.
5. A. Robert, *Atmos. Ocean* **19**, No. 1, 35 (1981).
6. W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vettering, *Numerical Recipes* (Cambridge Univ. Press, Cambridge, UK, 1987), p. 550.
7. A. Robert, *J. Meteorol. Soc. Jpn* **44**, 237 (1966).
8. J. R. Bennett and A. H. Clites, *J. Comput. Phys.* **68**, 272 (1987).
9. G. J. Haltiner and R. J. Williams, *Numerical Prediction and Dynamic Meteorology*, 2 ed., (Wiley, New York, 1980), p. 122.