Particle-in-cell simulations of a lens on an f-plane

A. D. Kirwan, Jr., C. E. Grosch, J. J. Holdzkom Ii

Nonlinear Processes in Geophysics (1997) 4: 71–91

Nonlinear Processes in Geophysics

Particle-in-cell simulations of a lens on an f-plane

A. D. Kirwan, Jr.[1], C. E. Grosch[1,2], and J. J. Holdzkom II[1]

[1]Center for Coastal Physical Oceanography, Old Dominion University, Norfolk, Virginia, 23529 USA
[2]Department of Computer Science, Old Dominion University, Norfolk, Virginia, 23508 USA

Received 15 July 1997 - Accepted 20 October 1997

Abstract. A particle-in-cell ansatz for solving the Euler equations in a rotating frame is described. The approach is ideally suited for "layered" models of flows with sharp density and velocity fronts. The material and Coriolis accelerations in the Euler equations are solved at each particle while the gradient accelerations are evaluated on a grid and interpolated at each time step to the particles. The height of each particle is fixed and, depending on the application may be constant for all particles or may vary from particle to particle. The approach is used here to predict the evolution of a lens in a layered model with lower layer outcropping. The integral invariant of volume is conserved exactly and total energy and total angular momentum are conserved to within 3% throughout a 30 day simulation. Exceptional resolution of the density and velocity fronts is maintained during the simulation without imposing any numerical viscosity. The model also reproduces essential characteristics of analytic solutions to a parabolic shaped lens. This algorithm is well suited to parallel implementation; all of the calculations reported here were done on an IBM SP2. Performance speedup and execution time as a function of the number of processors is given.

1 Introduction

Most simulations of classical fields (electricity and magnetism, elasticity, fluid dynamics) use Eulerian or fixed grid/element methods. But as noted by Olim (1994), time steps utilized by these schemes may be restricted by stability or damping so as to be considerably smaller than the time interval appropriate for the system evolutionary time scale. Since stability usually is not a critical issue with Lagrangian or particle methods, they have become fashionable for problems with large velocities.

Correspondence to: A. D. Kirwan, Jr.

Particle methods encompass a variety of approaches including particle-in-cell (PIC) methods, point vortex methods and semi-Lagrangian methods. For a brief discussion of the differences between some of these techniques, see Pavia and Cushman-Roisin (1988). Here we are only concerned with PIC schemes, first introduced by Harlow (see Harlow (1964) and earlier references therein). With the PIC approach all gradient-type terms in the conservation of mass and momentum equations are computed at fixed grid points while the material derivatives are computed at particles. These particles generally retain their identity throughout the calculation although the particle distribution could be reinitialized at any stage of the calculation if, for example, the particle density were to become low in some region. Much of the calculation with the PIC approach is devoted to interpolating particle properties to the fixed grid, calculating gradients, and then interpolating gradient values from the grid back to the particles. As with vortex methods there is a trade-off between resolution and number of particles. Since the computational load for large particle numbers increases only linearly with the number of particles for PIC methods these can be more efficient than purely Lagrangian methods. An excellent general review of PIC methods is given by Hockney and Eastwood (1992).

Our interest is the application of PIC methods to oceanographic flows. The purpose of the PIC method described here is not to replace the standard general circulation models but to calculate the evolution of fronts and submesoscale phenomena such as eddies, squirts and mushrooms which cannot be resolved by such models. The formation and evolution of many types of atmospheric or oceanographic fronts have no analog in fluid mechanics other than multi-fluid systems. As Brackbill and Ruppel (1986) noted PIC methods, which allow for conservation of a material property at particles, can be very effective for tracking material interfaces and modeling highly distorted flows. We believe that a PIC

method, embedded in a general circulation model, may be an effective method for modeling submesoscale phenomena. Despite these potential advantages, the only applications of PIC methods to oceanographic problems we are aware of is that of Pavia and Cushman-Roisin (1988, 1990), Pavia (1989) and Mathias (1992) who used PIC methods to study ocean fronts and merging of ocean eddies.

These latter PIC applications have been used in conjunction with layered models. The model equations result from a vertical integration of the Euler equations along with an appeal to quasi-hydrostatic equilibrium. If the height of individual particles is specified either *a priori* or by some pseudo-equation of state then there is no need to calculate the horizontal divergence of the velocity field and solve the conservation of mass equation at each particle. This is considerably simpler than the standard Eulerian approach of solving a Poisson problem for the pressure at each time step. However, there is still the issue of interpolating particle heights to the grid and the gradients back to the particles in a self-consistent manner. A second issue is the number of particles to use. To date, this has been empirically set as an ad hoc balance between accuracy and computational feasibility.

Like the previous applications the approach used here employs a layered model and specifies the height of each particle. There are, however, several significant differences in our approach. First, we use an interpolation routine that exactly cancels particle self-motion caused by fictitious pressure gradients associated with the distribution of the particle heights on the grid. Moreover, our approach does not require smoothing or damping other than that resulting from the interpolation routines. Finally, there is a natural way to "parallelize" the computation and all of our computations (apart from a few benchmark runs) are carried out using a parallel form of the algorithm.

Although PIC methods are used regularly in a number of disciplines, there is some skepticism about their utility in fluid problems. One of the chief concerns is "multistreaming"; *i.e.*, adjacent particles having greatly different velocities. The fear is that strong shear may excite artificial instabilities. A second concern is the fidelity of the evolution of fronts and smoothing out of frontal features by interpolation. To demonstrate that these concerns are baseless, at least for our approach, we first apply the PIC method to a reduced gravity lens on an f-plane. Although conceptually simple, this is a difficult computational problem for oceanographers using conventional methods since the velocity field and gradient of the height field are discontinuous at the lens boundary. We then apply the method to a two-layer problem with prescribed flow in the lower layer.

This paper is organized as follows: Section 2 contains the dynamical equations for the model; Section 3 is a description of the PIC paradigm for this layer flow; the applications, exact solutions and the invariants of these flows are described in Section 4; the initialization of the height and velocity fields is given in Section 5; Section 6 contains the results of calculations for the reduced gravity case while Section 7 discusses a case in which flow in the lower layer is prescribed. Section 8 contains timing and speedup results for the parallel implementation of this PIC scheme and, finally, conclusions are presented in Section 9.

2 Dynamical equations for the two-layer model

A particularly difficult situation to model by conventional gridded methods is when there are density fronts with transverse scales less then the grid spacing. Thus, this situation seems particularly appropriate for PIC methods. The simplest physical setting is a two-layer fluid in a steadily rotating coordinate system.

The hydrodynamic equations for this setting are well known to be

$$\frac{d\mathbf{v_1}}{dt} + \mathbf{k} \times f\mathbf{v_1} = -g\nabla(h_1 + h_2), \qquad (1A)$$

$$\frac{d\mathbf{v_2}}{dt} + \mathbf{k} \times f\mathbf{v_2} = -g\nabla(h_1 + h_2) + g_*\nabla h_1, \qquad (1B)$$

$$\frac{dh_1}{dt} + h_1\nabla \cdot \mathbf{v_1} = 0, \qquad (2A)$$

$$\frac{dh_2}{dt} + h_2\nabla \cdot \mathbf{v_2} = 0, \qquad (2B)$$

$$\frac{d}{dt} = \frac{\partial}{\partial t} + \mathbf{v} \cdot \nabla. \qquad (3)$$

In these equations, $\mathbf{v_i}$ is the horizontal velocity vector in the i^{th} layer; $\mathbf{k}$ is the unit vertical vector; f is the Coriolis parameter; h_i are the instantaneous layer thicknesses; and $g_* = g(\rho_2 - \rho_1)/\rho_2$ is "reduced" gravity.

Our approach is best explained by first considering the "reduced gravity" case, which results from requiring that the lower layer be at rest, *i.e.*, $\mathbf{v_2} \equiv 0$. Since the lower layer is at rest in this case the ∇h_2 dependence can be eliminated between (1A) and (1B) to express the pressure gradient acceleration in the former in terms of h_1. Also, the scheme used by Cushman-Roisin et al. (1985) to nondimensionalize (1) and (2) is appropriate. This scales h by H (representative layer thickness), ∇ by $f/\sqrt{g_*H}$, time by f^{-1}, and $\mathbf{v}$ by $\sqrt{g_*H}$. This scaling removes all explicit parameter dependence. The resulting equations are

$$\frac{d\mathbf{v}}{dt} + \mathbf{k} \times \mathbf{v} + \nabla h = 0, \qquad (4)$$

$$\frac{dh}{dt} + h\nabla \cdot \mathbf{v} = 0. \qquad (5)$$

Since the upper layer is the only active layer subscripts are suppressed. These equations must be augmented by two path equations for the particles:

$$\frac{d\mathbf{x}}{dt} - \mathbf{v} = 0. \tag{6}$$

3 PIC Paradigm

The basis of PIC methods is to solve the d/dt terms of (4)–(6) and compute body accelerations, such as Coriolis, at the particles while calculating field accelerations, such as pressure gradients, on a discrete grid. Interpolation between the particles and the grid occurs at each time step. The central assumption of the PIC method is that the heights and volumes of individual particles are invariant with position and time. Thus there is no need to solve (5); the height and hence the volume is conserved exactly. As suggested in Brackbill and Ruppel (1986), keeping each particle height fixed provides a means for accurate tracking of the interface between the active and inert layers.

The height gradient appearing in (4) is found by computing (interpolating) the "h" field to the grid using the positions and heights of the particles, computing ∇h on the grid, and finally computing (interpolating) ∇h back to each particle. In order to clarify these ideas we outline the algorithm. Assume that the position, velocity and height of each particle is known at time t. The steps in the algorithm to advance these data to time $t+\delta t$ are:

1. interpolate the heights of the particles to the grid,
2. calculate the finite difference approximation to ∇h on the grid,
3. interpolate ∇h from the grid to the particles,
4. using an appropriate time integrator, simultaneously integrate (4) and (6) from t to $t + \delta t$. This yields the position and velocity of each particle at the new time.

For oceanographic applications the critical issue is the calculation of the height or pressure gradients. In the standard approach used in Pavia (1989), Pavia and Cushman-Roisin (1988, 1990) and Mathias (1992), the height field is specified as the interpolated sum of heights over all particles within a grid cell. Individual particle heights are usually fixed and their base areas are no larger than a computation cell.

As in prior approaches, each particle is assigned a finite volume where the total volume of the particles is the same as the initial volume of the active layer and the height field on the grid is the sum of the interpolated particle heights. At this point we differ from many other investigators. First, we require the base of each particle to be a square with sides of length 2Δ, where Δ is the fixed distance between cell centers. Thus, a typical particle base has the area of four cells and generally will overlap into nine cells. Its location is specified by the coordinates of its center of mass or apex. In contrast to others using PIC methods for oceanographic flows, the interpolation weights are explicitly given as the fraction of the particle overlapping each cell.

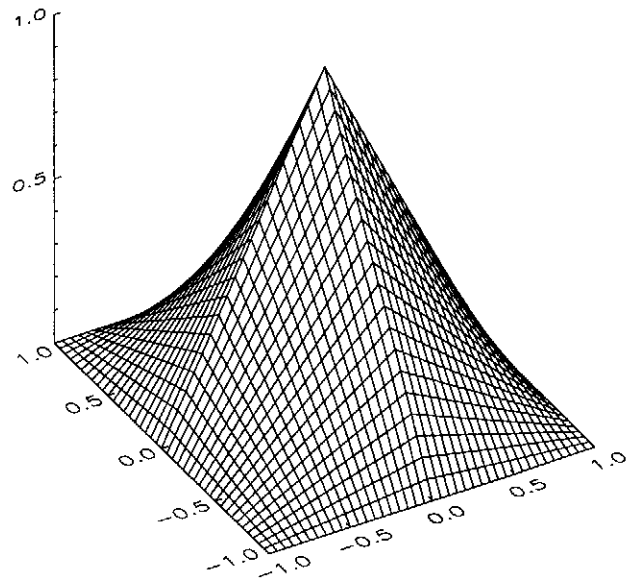

Fig. 1. Perspective picture of the shape of a particle.

Second, the particle shape used here is different from that of the finite volume particles used by many others. The height equation for a particle is

$$z_p = h_p(1 - |\mu_x|/\Delta)(1 - |\mu_y|/\Delta), \tag{7}$$

where (μ_x, μ_y) are the component distances from the apex. Note that $|\mu_x|, |\mu_y| \leq \Delta$. The apex height of a particle is h_p, which may vary from particle to particle. As can be seen from (7), the height decays to zero at the particle boundaries. This shape was utilized by Hockney and Eastwood (1992) who termed it a "triangular-shaped cloud." Figure 1 is a three-dimensional perspective of this shape. Within any one grid cell there will be considerable overlap of particle bases.

The particles are not to be thought of as solid particles that cannot be interpenetrated but as ethereal clouds used for a computational representation of fluid particles. If several particles were to be located at exactly the same (x, y) location, we interpret this as meaning that the height, h, at that location is the sum of the height of the individual particles. These particles do not lose their identity as they interpenetrate and pass through each other, so the number of particles is invariant. This approach was first suggested by Hockney (1966) and used successfully by Birdsall and Fuss (1969) in plasma simulations. Unlike the plasma applications there are no particle-particle interactions other

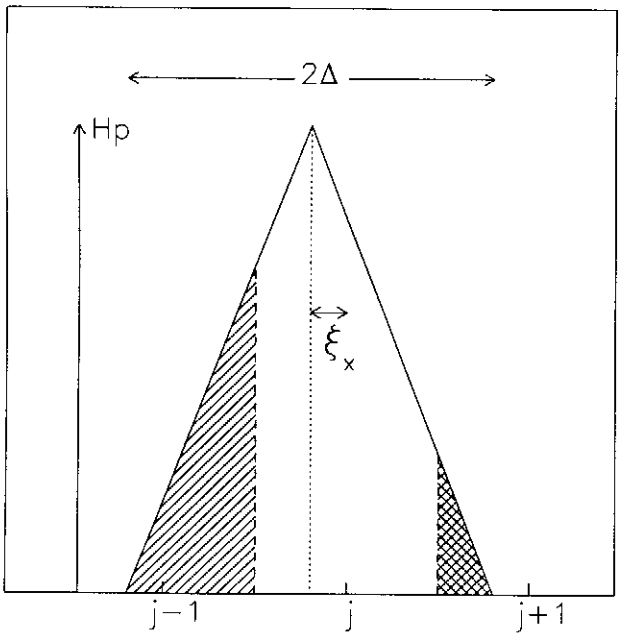

Fig. 2. Cross-section of a particle along the x-axis. The hatched, open, and cross-hatched areas divided by Δ give the weights specified in (8). The two vertical dashed lines are the cell boundaries.

than what arise through the gradient of the height field.

Other differences in our approach from previous studies are the interpolation routine used to move the gradients back to the particles and the absence of explicit smoothing. Our experience has been that the differentiation and interpolation operations are not independent but must be done in an internally consistent fashion. The principle used here is to require that gradients interpolated to a particular particle be independent of the height of that particle. This insures no self-generated motion.

Figure 2 illustrates a typical cross-section of a particle along the x axis. From (7), the normalized cross-sectional areas of the particle in each of the enveloped cells along this axis are

$$\begin{aligned} w_{j-1} &= (1/2 - \xi_x/\Delta)^2/2, \\ w_j &= \{1 - (1/2)[1/2 + 2(\xi_x/\Delta)^2]\}, \\ w_{j+1} &= (1/2 + \xi_x/\Delta)^2/2, \end{aligned} \tag{8}$$

where the normalization factor is the cross-sectional area of the particle and (ξ_x, ξ_y) are the component distances from the apex of a particular particle. The interpolation of the height of this particle to the overlapped cell centers is then given by $h_p w_{j-1}$, $h_p w_j$, $h_p w_{j+1}$ for the $j-1$, j, and $j+1$ cells, respectively. The hatched, open, and cross-hatched areas in figure 2 show how the height is partitioned at the three grid points.

The above results apply to a typical cross-section for a single particle. The two-dimensional case for a generic particle p is achieved by prescribing the weights as

$$w_p(\xi_x^p, \xi_y^p) = w_i(\xi_x^p) w_j(\xi_y^p). \tag{9}$$

To summarize, the height at the ith, jth cell center is the weighted average of all particles enveloping the cell. This is given by

$$h_{i,j} = \sum_{p=1}^{N_{i,j}} w_p h_p, \tag{10}$$

where p is the particle number in the cell and $N_{i,j}$ is the number of particles that overlap into the cell i, j. After the heights are accumulated at all grid points, second order accurate gradients are calculated and then interpolated back to the particles using the same weights. From (10), the values of the gradients at the nine relevant cell centers are readily expressed as

$$D_{i+\alpha,\, j+\beta} = [h_{i+\alpha+1,\, j+\beta} - h_{i+\alpha-1,\, j+\beta}]/2\Delta, \tag{11A}$$

for the x component of the gradient and

$$= [h_{i+\alpha,\, j+\beta+1} - h_{i+\alpha,\, j+\beta-1}]/2\Delta, \quad \alpha, \beta = -1, 0, 1, \tag{11B}$$

for the y component.

Interpolation of these gradients back to a particular particle P in cell i, j is independent of h_P. To see this, focus on the x component, (11A). The interpolation of this component back to P is

$$\begin{aligned} G_{i,j}^P = &\sum_{\alpha=-1}^{1} w_{P(i+\alpha,\, j+1)} D_{i+\alpha,\, j+1} + \\ &\sum_{\alpha=-1}^{1} w_{P(i+\alpha, j)} D_{i+\alpha,\, j} + \sum_{\alpha=-1}^{1} w_{P(i+\alpha,\, j-1)} D_{i+\alpha,\, j-1}. \end{aligned} \tag{12}$$

Using (10) and (11A) the first sum in (12) is

$$\begin{aligned} 2\Delta \sum_{\alpha=-1}^{1} w_{p(i+\alpha,\, j+1)} D_{i+\alpha,\, j+1} = \\ w_{P(i-1,\, j+1)} \Big[\sum_{p=1}^{N_{i,\, j+1}} w_p h_p - \sum_{p=1}^{N_{i-2,\, j+1}} w_p h_p \Big] \\ + w_{P(i,\, j+1)} \Big[\sum_{p=1}^{N_{i+1,\, j+1}} w_p h_p - \sum_{p=1}^{N_{i-1,\, j+1}} w_p h_p \Big] \\ + w_{P(i+1,,\, j+1)} \Big[\sum_{p=1}^{N_{i+2,\, j+1}} w_p h_p - \sum_{p=1}^{N_{i,\, j+1}} w_p h_p \Big]. \end{aligned}$$

Now, h_P does not appear in the $N_{i-2,\, j+1}$ and $N_{i+2,\, j+1}$ sums since particle P is not in either cell. It appears once in each of the other four sums; however, the terms exactly cancel. It is readily seen that this is true for each of the other sums in (12) as well as the y component of the gradient.

The cancellation of particle height in the interpolated gradient to a particle is analogous to the condition that

the second order accurate centered finite difference approximation to the gradient operator at grid points does not depend on the value of the function at those grid points. The importance of this to the present application is that there is no self-induced motion of a particle. This result arises for two reasons. First, the same weights are used for the interpolation from particles to the grid and then back to the particles and not because of a specific form such as (8). Second, the order of accuracy of the derivative (second order in our case) is not higher than that of the interpolation routine. Fourth order accurate derivatives used in conjunction with a lower order interpolation scheme may produce self-generated motion and thus would not be as internally consistent as the second order schemes.

This method has several other attributes. First, the weights are computationally efficient since they are not difficult to calculate and are used both for interpolation from particles to grid points and back to the particles. Second, as mentioned above, (5) is automatically satisfied since the heights and volumes of the particles are assigned initially and are fixed. Moreover, it is not necessary to interpolate particle velocities as long as there is no viscosity in the problem. Finally, any stable time integrator can be used to advance the position of each particle.

The major tradeoff in choosing the time integrator is that between accuracy and computational work. For example, a two step integrator, such as a second order Runge-Kutta scheme, has an accuracy of $O((\delta t)^2)$ but requires two evaluations of ∇h at each particle. Thus this integrator requires two interpolations of the particle heights to the grid, two applications of the finite difference form of ∇ to h on the grid and two interpolations of ∇h to the particles for each position advancement. On the other hand a first order accurate integrator could provide the same degree of accuracy with less computation per time step but at the cost of a much smaller time step. In all of our computations we used a second order accurate Runge-Kutta method because the computational work per unit time is substantially less than for a first order method. Any other second order time stepping scheme could have been used.

4 Applications

The problems studied here consist of an isolated fluid lens of density ρ_1 atop a fluid layer of density ρ_2 with finite thickness and of infinite horizontal extent. Figure 3 is a cartoon of the geometry.

Lens models have been the subject of considerable study since the pioneering work of Cushman-Roisin et al. (1985); see also Brickman and Ruddick (1990), Ruddick (1987), Cushman-Roisin (1987), Kirwan and Lipphardt (1993), Kirwan et al. (1992, 1994), Kirwan and Liu (1991), Ripa (1987), Rogers (1989), Young (1986) and Holdzkom et al. (1995). In these studies the lens velocity and thickness were specified as

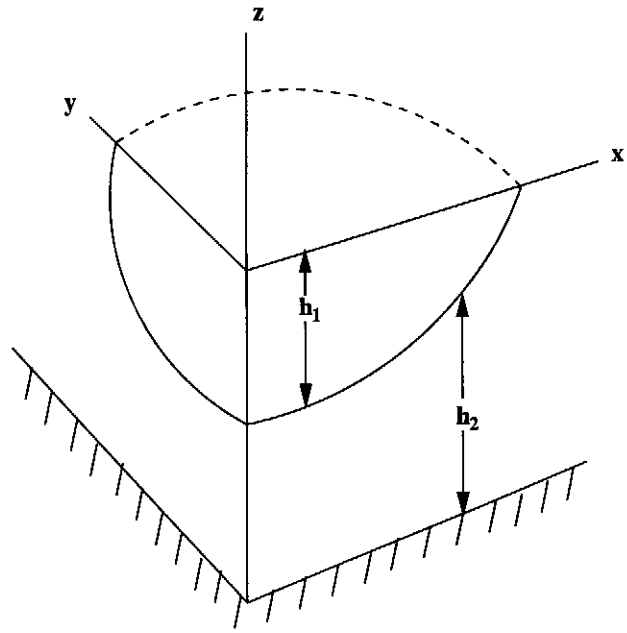

Fig. 3. Schematic of the two layer model.

$$h = h_o(t) + B_{ij}(t)x_i x_j, \quad \text{for } h \geq 0$$

$$\text{where } B_{ij} = B_{ji}$$

$$v_i = G_{ij}(t)x_j, \quad \text{for } h \geq 0$$

$$v_i = 0, \quad \text{elsewhere} \tag{13}$$

Here the summation convention is used for repeated indices and the spatial coordinate x_i is measured from the center of mass of the lens. Note that the velocity field is discontinuous at $h = 0$ as is ∇h. Moreover, the discontinuous frontal boundary, $h = 0$ must be calculated as part of the solution. Conventional gridded methods are not well suited for problems of this sort.

Substitution of (13) into (4) and (5) yields 8 coupled nonlinear ordinary differential equations for $G_{ij}(t)$, $h_o(t)$ and $B_{ij}(t)$. As shown in Young (1986), the resulting solutions can be represented as a nonlinear superposition of rotational, deformational and horizontal divergence modes. The lens frontal boundary generally may oscillate between a circle and an ellipse which tends to rotate anticyclonically.

4.1 Circular lens

For the lens model Ball (1965) obtained an analytic solution for h_o, B_{ij} and G_{ij}, which is called here the pulson solution. The initial velocity field is given by

$$v_1(0) = (G/2)x_1 - (G_R)x_2,$$

$$v_2(0) = (G_R)x_1 + (G/2)x_2, \tag{14}$$

where $G = G_{11}(0) + G_{22}(0)$ is the initial horizontal divergence and $G_R = [G_{21}(0) - G_{12}(0)]/2$ is the initial spin.

The nondiagonal components of B_{ij} and the symmetric component of G_{ij} are zero for all time while the remaining components as well as h_o oscillate at the Coriolis frequency f. Unlike most nonlinear problems the oscillation frequency of the pulson is independent of the initial conditions.

Of particular interest here are the analytic expressions for the centerline height, h_o, and $R(t)$, the lens boundary. These are found to be

$$h_o(t) = H\Gamma(t),$$
$$R(t) = -h_o(t)[B_{11}(t) + B_{22}(t)]^{-1}, \quad (15)$$

where H is the layer thickness scale at $t = 0$, and

$$\Gamma(t) = [A + \gamma \sin ft]^{-1}, \quad A > |\gamma|.$$

Also, A and γ are given by initial conditions of G_{ij}.

4.2 Elliptical Lens

The pulson flow field has vorticity and horizontal divergence but no deformation. To test the efficacy of the PIC model in the presence of a time dependent deformation component of the velocity field, the initial velocity (14) was replaced by

$$v_1(0) = (G/2 + G_N)x_1 + (G_S - G_R)x_2,$$

$$v_2(0) = (G_S + G_R)x_1 + (G/2 - G_N)x_2. \quad (16)$$

Here $G_N = [G_{11}(0) - G_{22}(0)]/2$ is the initial normal deformation and $G_S = [G_{12}(0) + G_{21}(0)]/2$ is the initial shear deformation.

The solution for the elliptical lens case, here called the deformation solution, was obtained by numerically integrating the eight equations for h_o, B_{ij} and G_{ij}. See Kirwan and Liu (1991) and Kirwan et al. (1992) for details regarding the solution procedure. This solution shows that the lens boundary starts as a circle, quickly deforms into an ellipse which rotates anticyclonically for a brief period then abruptly deforms back to a circle. This cycle is repeated; however, each new repetition of the circular or elliptical phase may not have the same geometric characteristics as the previous phase.

4.3 Integral invariants of the lens solutions

By integration over the lens, Ball (1963, 1965) showed that there are five global invariants for this motion which do not depend on specific solution forms. Four of these have succinct physical interpretation: volume, total angular momentum, total energy, and potential vorticity. The respective non-dimensional versions are:

$$V_T = \int h dA \quad (17)$$

$$L = \int [xv - yu + (x^2 + y^2)/2] h dA \quad (18)$$

$$E = \int (h/2)[u^2 + v^2 + h] dA \quad (19)$$

$$Q = \int hF(q) dA, \quad q = (1 + \partial v/\partial x - \partial u/\partial y)/h \quad (20)$$

In (20) F is an arbitrary function of the potential vorticity q. The fifth invariant is a nonlinear function of the moment of inertia of the upper layer which also involves L and E; however it is not used in this study.

With our PIC paradigm, the number and heights of the particles are constant, so V_T is conserved exactly. Furthermore, potential vorticity could be assigned to each particle. As long as the number of particles remains invariant, (20) is also conserved exactly.

5 Initialization for the PIC solution

A general scheme for initializing the height and velocity fields is described below. Consider the upper layer; it is first divided into regions bounded by isothickness contours. The area and volume of those regions is then calculated. Dividing this volume by the volume of a particle gives the number of particles for the region. Finally, the particles are equally spaced within each region. In order to keep the number of particles per cell constant between regions of different layer thickness, the height (and thus volume) of the particles may be varied from region to region. Once the initial positions of the particles are determined, each particle is assigned an initial velocity.

For the experiments discussed below, the initialization is a crucial part of the calculation in that it determines which solution of the dynamical equations will occur. Furthermore, if the initial height and velocity fields are not self-consistent, then the lens solution can not be accurately reproduced. In order to show how the general scheme outlined above is applied to the lens problem, the initialization procedure for the height and velocity fields for the two problems are given in detail.

For both problems described above the initial shape of the lens is a circular plan form with a parabolic height distribution. However, the initial velocity distribution is different for each of these problems.

The nondimensional equation for the initial lens thickness for both problems is

$$h = h_o[1 - (r/R)^2], \quad (21)$$

where h_o is the centerline thickness, r is the radial coordinate from the lens center and R is now the initial

radius of the lens. For both problems the volume occupied by the lens is

$$V_L = 2\pi \int_0^R rhdr = (\pi/2)h_o R^2. \quad (22)$$

Integration of (7) shows the volume of a particular particle to be $V_p = \Delta^2 h_p$ so the total volume occupied by the particles is

$$V_T = \Delta^2 \sum_{p=1}^{N} h_p, \quad (23)$$

where N is the total number of particles. Clearly, $V_T = V_L$ must be required.

Consider first the case where h_p is the same for all particles. Then, the number of particles is

$$N = V_T/V_p = (\pi/2)(h_o/h_p)(R/\Delta)^2. \quad (24)$$

To estimate N, divide the lens into concentric circular annuli surrounding an axial cylinder centered at the lens center. The radial width of an annulus is $2\Delta/d$ with the parameter $d \geq 2$, and the height of the lens along the axis of an annulus is given by (21). In the limiting case, $d = 2$, the annulus width is the same as the computational cell discussed in the previous section. Since the annuli are used for the initial distribution of particles in the lens it is appropriate to have the annuli spacing finer than that of the computational cells.

The central radius of the outermost annulus is at radius $r = R - \Delta/d$. Then the height at this radius is

$$h_L = h_o(\Delta/Rd)(2 - \Delta/Rd). \quad (25)$$

Obviously, $h_p \leq h_L$ with the equality holding when requiring only one particle in each cell on the lens boundary. Using this limiting case for h_p in (24) gives

$$N = (\pi/2)(R/\Delta)^3 d/(2 - \Delta/Rd). \quad (26)$$

All having the same height, these particles must be distributed nonuniformly in the lens. We shall use $R = 10^{-1}$, $\Delta = 4 \times 10^{-3}$ and $d = 16$ so (26) suggests $N \sim O(2.5 \times 10^5)$ for $h_p \sim h_L$. However, requiring ten particles in each cell in the outer boundary increases N by an order of magnitude.

The number of particles can be reduced by decreasing the horizontal spatial resolution; i.e., increasing Δ/R. From (26) it is seen that changes in resolution can cause dramatic changes in the particle count. An order of magnitude change in resolution produces three orders of magnitude change in the number of particles.

One concern with a constant h_p is the number of particles in each cell will vary with radius resulting in fewer particles in the outermost cells. This would imply some loss of accuracy in these cells. To overcome this h_p is allowed to vary with radius so that the number of particles is approximately the same in each cell. To achieve an equal number of particles in each cell first note that the volume of an annulus centered at r is

$$V_A = 2\pi \int_{r-\Delta/d}^{r+\Delta/d} rhdr$$
$$= 4\pi h_o r(\Delta/d)[1 - (r/R)^2 - (\Delta/Rd)^2]. \quad (27)$$

The number of particles required to fill this volume is

$$N_A = V_A/V_p$$
$$= 4\pi[h_o/h_p(r)](r/\Delta d)[1 - (r/R)^2 - (\Delta/Rd)^2]. \quad (28)$$

The number of cells in this annulus is the ratio of the annulus area to cell area. A simple calculation gives this number as $N_c = 4\pi r/\Delta d$. If the number of particles in each cell, $N_\# = N_A/N_c$, is to be constant the particles heights must be distributed as

$$h_p(r) = (h_o/N_\#)[1 - (r/R)^2 - (\Delta/Rd)^2]. \quad (29)$$

Adjusting the height of the particles to keep the number of particles in a cell constant also changes the requirements on the total number of particles. The cross-sectional area of the lens annulus is $\pi[R^2 - (\Delta/d)^2]$. Thus, the number of cells needed to cover the lens annulus is

$$N_{\text{cell}} = \pi[R^2 - (\Delta/d)^2]/\Delta^2, \quad (30)$$

and the total number of particles in the lens annulus is

$$N_{TA} = N_\# \cdot N_{\text{cell}}. \quad (31)$$

To complete this analysis it is necessary to fill the inner cylindrical core. Its volume is

$$V_I = 2\pi \int_0^{\Delta/d} rhdr = \pi h_o(\Delta/d)^2[1 - (1/2)(\Delta/Rd)^2]. \quad (32)$$

The number of particles required to fill this void is

$$N_I = \pi[h_o/h_{p(I)}d^2][1 - (1/2)(\Delta/Rd)^2]. \quad (33)$$

This is determined by an arbitrary choice of $h_{p(I)}$. Here we use $N_I \sim 10^2$.

A critical issue with this approach is an appropriate number of particles for each cell $N_\#$. Using different particle geometry than we do, other investigators (Pavia, 1989, Pavia and Cushman-Roisin, 1988, 1990 and Mathias, 1992) used about 15 particles per cell. It should be noted that these investigators used smoothing at every time step. With our geometry and elimination of smoothing we use $N_\# \sim 5 \times 10^2$. The total number of particles with this value of $N_\#$ is

$$N_T = N_{TA} + N_I \sim 10^6. \quad (34)$$

Once the initial position of each particle is assigned we use the lens model equations to compute the initial velocity appropriate to each particle, which depends only on the initial particle position.

It should be noted that we use a polar coordinate system here *only* to initialize the distribution of particles. The calculations presented in the next section are done with a rectangular grid even for those problems which exhibit radial symmetry.

In experiments not reported here we have initially randomized particle positions and found that the solution evolves to different states depending on the randomization. The reason is that this randomization introduces small perturbations in the height field which impacts the evolution of particle trajectories. This finding differs from Mathias (1992). However, in his calculation explicit smoothing was applied at every time step. In effect, this dissipated the perturbations. In our calculations we do not use explicit smoothing. One important consequence of this is that we are able to preserve the integral invariants to a high degree.

The description above has focused on the initialization of a confined, circularly shaped upper layer. Extensions to other geometries and to the lower layer are done in an analogous way.

6 Results of the reduced gravity calculations

In both cases, pulson and deformation, the computational domain was $-1/2 \leq x \leq 1/2$ and $-1/2 \leq y \leq 1/2$, the grid was 256×256 and the initial radius of the lens was 0.1. This means that the lens initially occupied only 3% of the computational domain. At the edges of the computational domain periodic boundary conditions were applied. However, for results reported here these conditions were never invoked since no particle ever approached the edge of the computational domain. In fact, most of the 256×256 cells are empty during these simulation. In other cases we examined, to be reported separately, appropriate boundary conditions are important.

The timestep for all simulations reported here was 0.01 inertial day.

6.1 Circular Lens

The first simulation discussed here was designed to test the ability of the PIC ansatz to reproduce a single frequency in a nonlinear flow and to provide high resolution of the associated oscillating front. Here and in the next problem we shall use the variable height formulation described in the previous section. It is stressed that there is no explicit smoothing of the results presented below.

As a first test of the PIC simulation, comparisons are made with the solutions of the circular lens (pulson) model. In this model the height field is parabolic and the velocity field linear with respect to the distance from the lens center, out to the edge of the lens where $h = 0$, for all time. The radius of the lens is time-varying. Note that the velocity field is discontinuous at the edge of the lens. These constraints are imposed only initially on the PIC model but are explicit in the lens model. Initial conditions for this experiment are given in table 1.

Table 1. Initial conditions

	Pulson Case	Deformation Case
h_o	4.875×10^{-4}	4.875×10^{-4}
$(B_{11} + B_{22})/2$	-9.75×10^{-2}	-9.75×10^{-2}
B_{12}, B_{21}	0.0	0.0
G_R	-0.25	-0.25
G_N	0.0	0.10
G_S	0.0	0.05
G	0.60	0.60

The first issue to be considered is multi-streaming. Figure 4 shows the velocity vectors for a representative cell at day 10 of the calculation. The velocity vectors are well aligned everywhere in the cell. Similar results have been obtained for other regions in this simulation as well as for other simulations. There is no indication that multi-streaming is a problem.

This figure also shows the particle spacing in the cell that develops after 10 days. (Remember, however, that each particle's base is two grid cells in length and width, and its position is represented here only by its center). In the initial field the particle spacing was uniform. We have not determined why this somewhat non-uniform spacing develops. As discussed below it has negligible effect on the calculation for as long as 30 days.

Figure 5 compares the PIC solution for the centerline height h_o (solid curves) with the analytic solution (dashed curve) given by (15). This figure shows that the PIC solution accurately reproduces both the amplitude and phase of the analytic solution. The nonlinear nature of the motion is clearly shown by the differences between the narrow crests and broad troughs. Note that beginning about day 2 small deviations between the computed and analytic solution can be seen in this figure. Although they persist for the rest of the record they do not grow. Instead, they are actually decreasing by the end of the computation.

Figure 6 compares the PIC height field at 10 days (solid curve) with the initial height field (dashed). Again, the comparison is excellent although there is some indication that near the lens center the PIC simulation is slightly thinner than the exact solution. This is also reflected in the slight increase in the simulation thickness at mid-radii. It is noted that the PIC solution retains a circular shape throughout the simulation. This indicates that the velocity field does not develop any significant deformation during the calculation.

The edge of the lens with discontinuities in the velocity and height gradient fields is the most difficult region

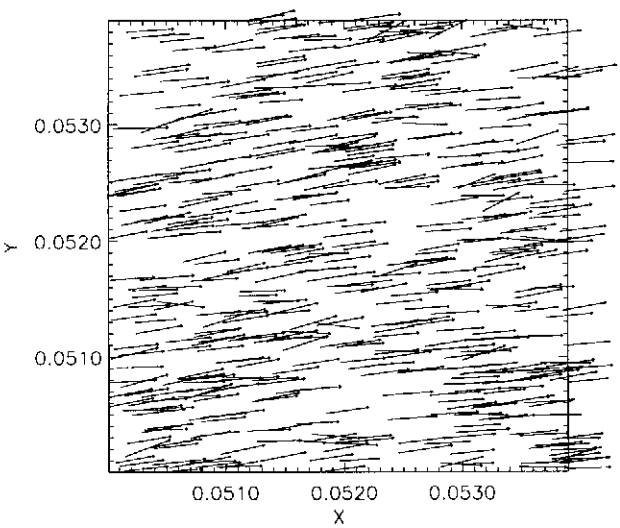

Fig. 4. Velocity vectors for each particle in one cell within the lens after 10 inertial days of integration. This is from the pulson case with 10^6 particles.

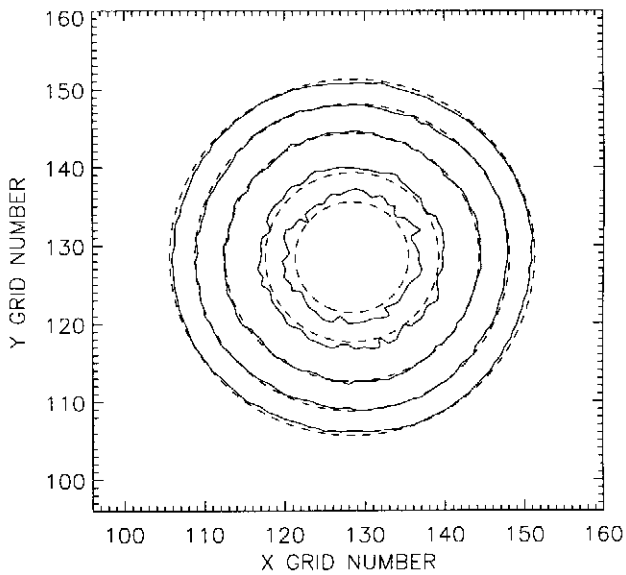

Fig. 6. Comparison of h at $t = 10$ inertial days between the PIC model (solid contours) and the initial h (dashed contours) for the pulson case.

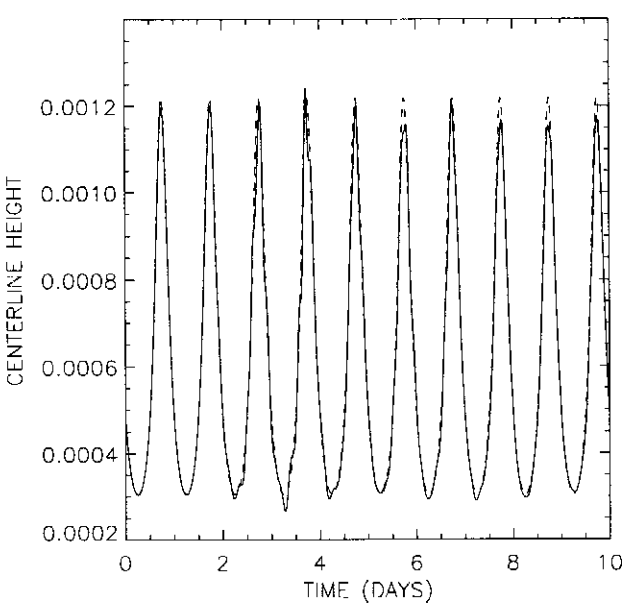

Fig. 5. Comparison of the PIC centerline height with the analytic solution for the pulson case. The PIC result is the solid curve and the analytic solution is the dashed curve. Time is in inertial days.

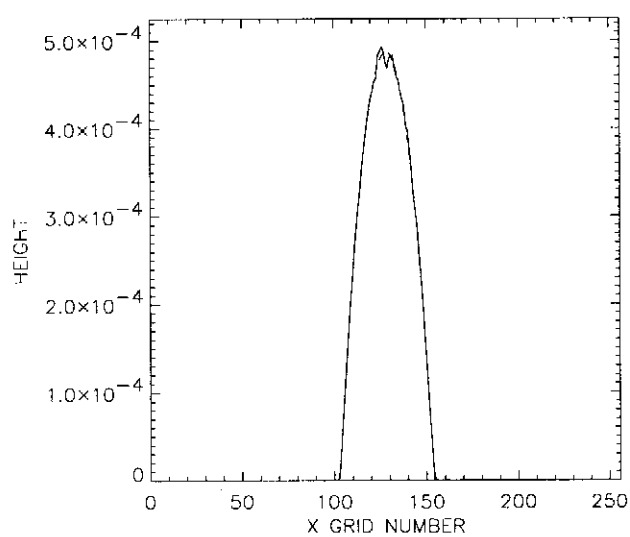

Fig. 7. Comparison of the numerical height profile (solid curve) after 10 inertial days with the analytic profile (dashed curve), using 10^6 particles for the pulson case.

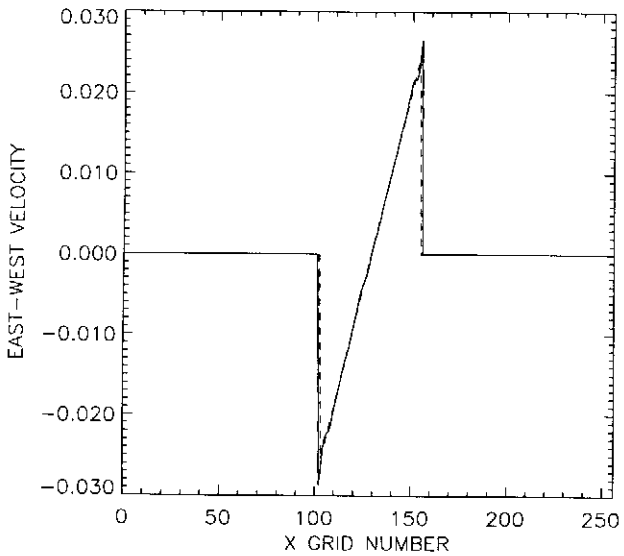

Fig. 8. Comparison between the numerical east-west velocity profile (solid curve) and the analytical solution (dashed curve) after 10 inertial days, using 10^6 particles for the pulson case.

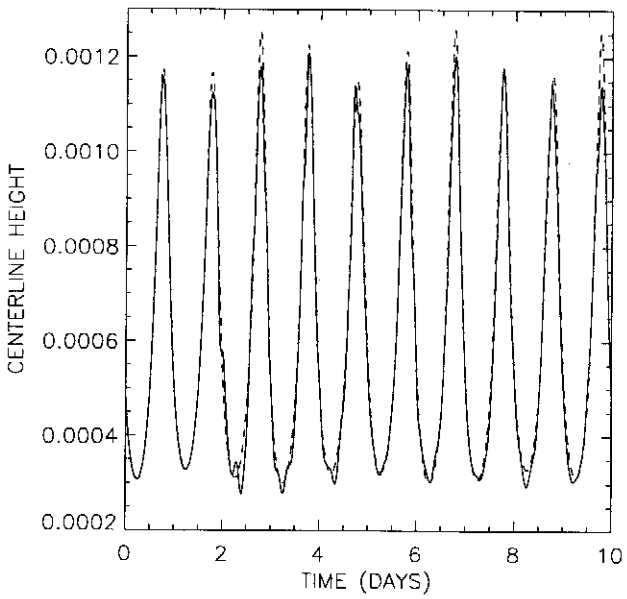

Fig. 9. Comparison of the PIC centerline height with the numerical solution to the lens model for the deformation case over 10 inertial days. The PIC solution is the solid curve while the lens solution is the dashed curve.

to simulate with conventional models. It is appropriate then to examine the performance of the PIC model in this region. Figure 7 compares the height field from the model with the analytic solution after 10 days. At the edges the numerical solution shows a slight "blur". This is the result of the "footprint" of the particle which extends over two cells. The lens center shows a slightly larger error; however, the maximum percent error is only 4%.

Because of a discontinuity at the edge of the lens, the velocity profile is the most stringent test of the model. Figure 8 compares the analytic velocity solution after 10 days with the numerical solution. The model velocity tracks the analytic solution extremely well in the lens interior. The model results also show a sharp discontinuity at the edge of the lens; however, it has been only slightly displaced outwards by the particle footprint. This is in contrast to the usually poor performance of conventional gridded models at discontinuous boundaries.

6.2 Elliptic Lens

The results given in the previous section were for the case of a circular lens with no deformation in the velocity field. Here we present results of simulations when deformation has been introduced into the flow. In the analytic case, this causes an initially circular lens to deform into an ellipse and then to oscillate between these two shapes. The choice of the amount of deformation is arbitrary but important in the evolution of the lens since it is widely believed that deformation can produce an instability that may cause the lens to break up. We have selected a level of deformation that seems to be consistent with observations of long lived ocean vortices. Holdzkom et al. (1995) analyzed remote sensing and hydrographic data from an anticyclonic vortex in the North Atlantic and showed that it underwent significant oscillations in eccentricity yet remained a viable vortex for about 6 months, after which it was re-absorbed by the Gulf Stream. The parameters for this simulation (see table 1) were chosen so as to approximate the structure of that eddy.

Figure 9 compares the PIC centerline height (solid curve) with the numerical solution to the deformation model (dashed curve). As seen in this figure, the solutions agree quite well through day 2. After this time the solutions agree well in phase but the amplitude of the peaks of the PIC solution tends to be slightly smaller than the lens solution. Note that, again, deviations between the PIC calculation and the analytic solution begin about day 2.

Figures 10 and 11 compare the evolution of the lens boundary determined by the PIC simulation with the lens solution. The detailed evolution for the first day as plots of the boundary curve at an interval of 0.2 days is shown in figure 10. The initially circular boundary quickly evolves to an elongated ellipse which pulsates and rotates anticyclonically. The cross-sectional area of the ellipse at day 0.2 is larger than the original circular area, and, as can be seen in figure 9, the corresponding height at the center is lower. At day 0.8 the lens has a circular shape once again with a smaller cross-sectional area and greater central height than at day 0.0. At day 1.0 the lens has once more taken an elliptical shape. This

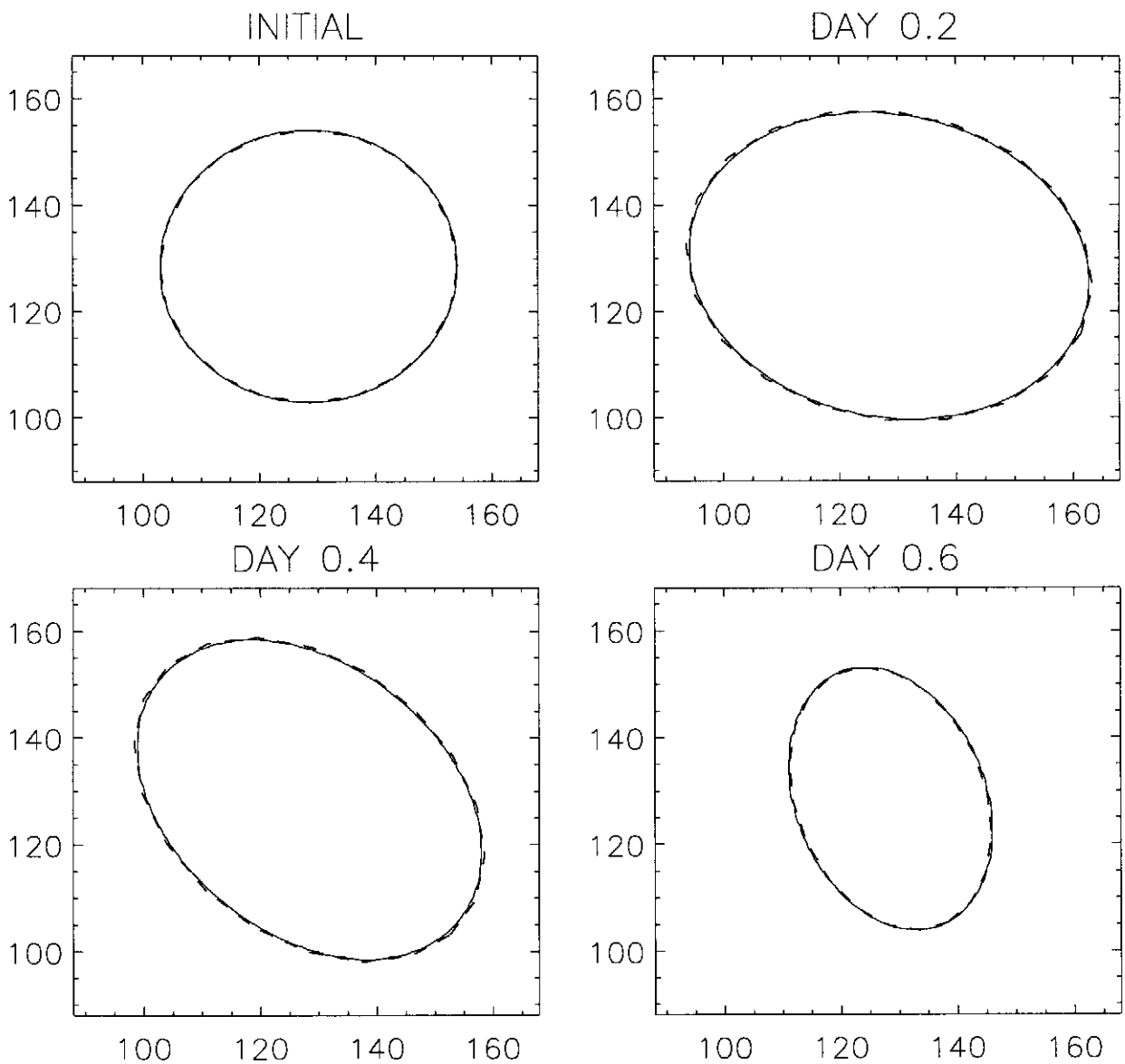

Fig. 10. Comparison of the boundary of the lens for the deformation case, as determined from the PIC solution and the deformation model over 1 inertial day. The PIC solution is the solid curve while the deformation solution is the dashed curve. The axes labels are the grid points in each direction.

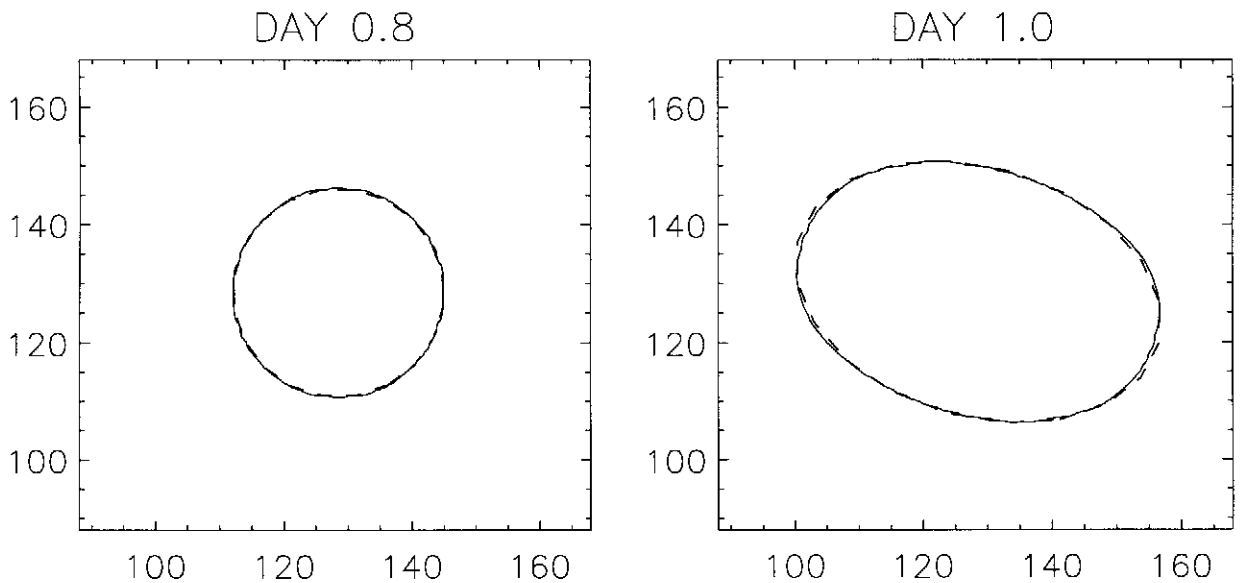

Fig. 10. cont'd.

cycle continues at the inertial frequency throughout the simulation. It is seen in figure 10 that for the first day the PIC and lens solutions are in close agreement in both amplitude and phase.

Figure 11 shows snapshots of the boundary over a 10 day period. It is clear from these results that there is good agreement between the PIC and lens solutions up to day 2.5. Beginning at day 4.0 a phase difference between the two solutions becomes apparent. A difference in cross-sectional area is also seen by day 6.5. By day 10.0 the phase difference is substantial but the difference in cross-sectional area is about the same as at day 6.5. It is important to remember that there there is a strong inertial signal in the solution. For example, between days 5.0 and 6.5 the lens does not merely rotate and deform a modest amount, but instead repeats the cycle shown in figure 10 approximately one and a half times.

The discrepancies between the PIC simulation and and lens solutions begin around day 2. There is no indication of numerical instability in either case and no indication of a hydrodynamic instability in the PIC simulations. The analytic lens model solution cannot exhibit instabilities since the height and velocity profiles are required for all time to be parabolic and linear respectively. The exact solutions are much more restricted than the PIC solution which is essentially a primitive equation solution. Although either solution could be representative of real oceanic lenses, we question whether the behavior of exact solutions which require linear velocity and quadratic height fields at all times is indicative of real fluid lens behavior. The exact solutions, however, are useful as benchmarks for qualitative aspects of the PIC simulation.

The PIC calculations described above used approximately 10^6 particles. In order to illustrate the deterioration in the solution when less particles are used, the deformation case was repeated with only about 10^5 particles. Figure 12 shows the evolution of the centerline height for the two cases. Note that for plotting purposes the 10^6 particle case is shown as the dotted curve while the 10^5 particle case is shown as the solid curve. It is seen that the peaks and troughs become ragged for the calculation with a reduced number of particles although the amplitude and phase are still resolved quite well. The maximum percent error of the height field is now 9%.

6.3 Integral Invariants

Comparison of the PIC solutions with the lens model is quite encouraging but inconclusive as to other applications of the method. A better test of the efficacy of the approach is to determine how well the PIC values conserve the integral invariants (17)–(20), as the latter are independent of specific solution forms. As discussed above the method is volume and potential vorticity preserving. However, the remaining invariants are not identically conserved so they can provide independent insight into the overall performance of the PIC model. In order emphasize their fundamental importance, the time series of the invariants is shown for thirty days.

First consider the total angular momentum, L, (18). This is composed of the sum of the local angular momentum of the individual particles and that induced by the rotating frame on the lens. Both terms are readily computed at the particles and then summed over all particles. Total energy, E, (19) is composed of kinetic energy which is readily evaluated at the particles and

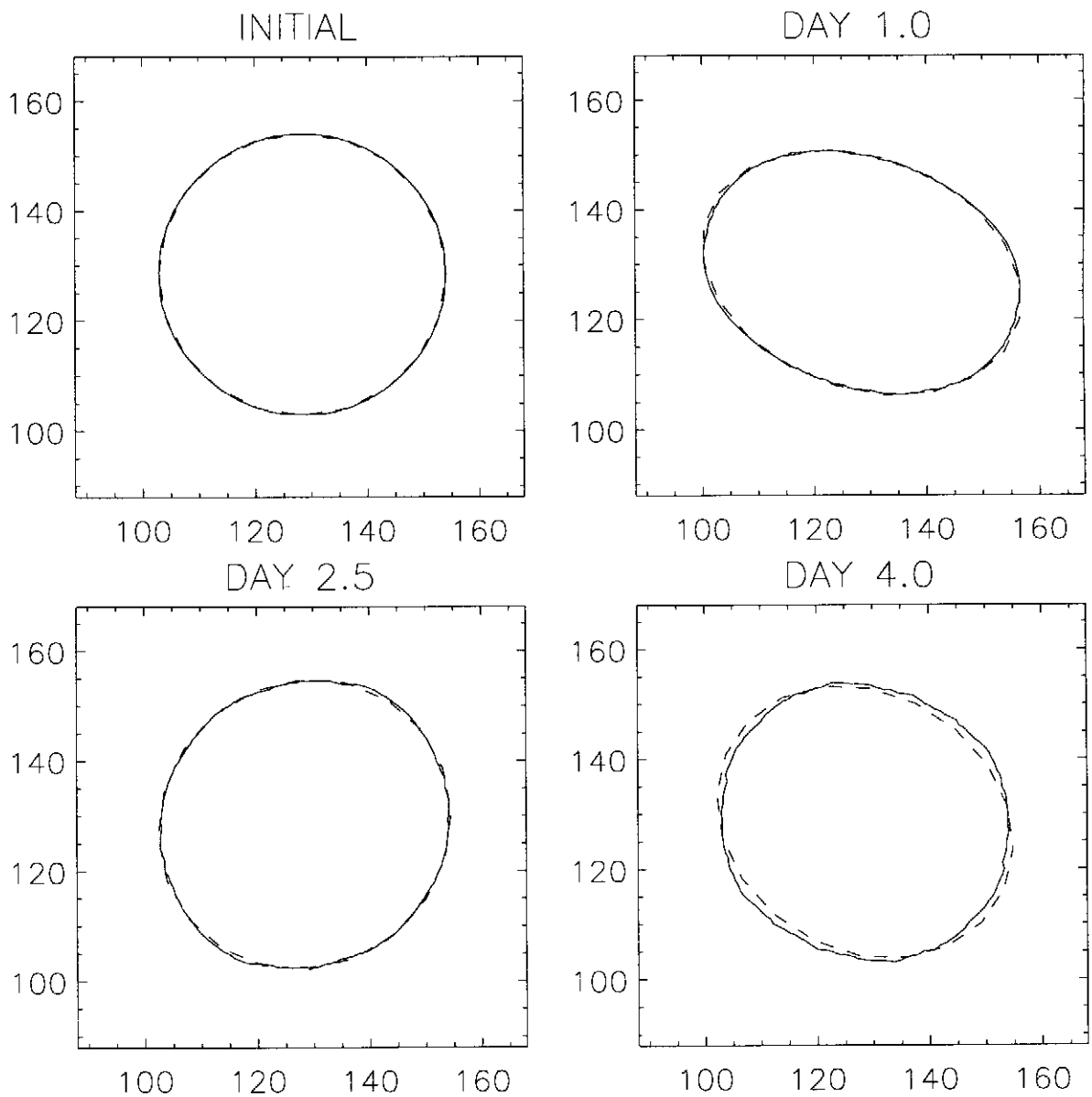

Fig. 11. Comparison of the boundary of the lens for the deformation case, as determined from the PIC solution and the deformation model over 10 inertial days. The PIC solution is the solid curve while the deformation solution is the dashed curve. The axes labels are the grid points in each direction.

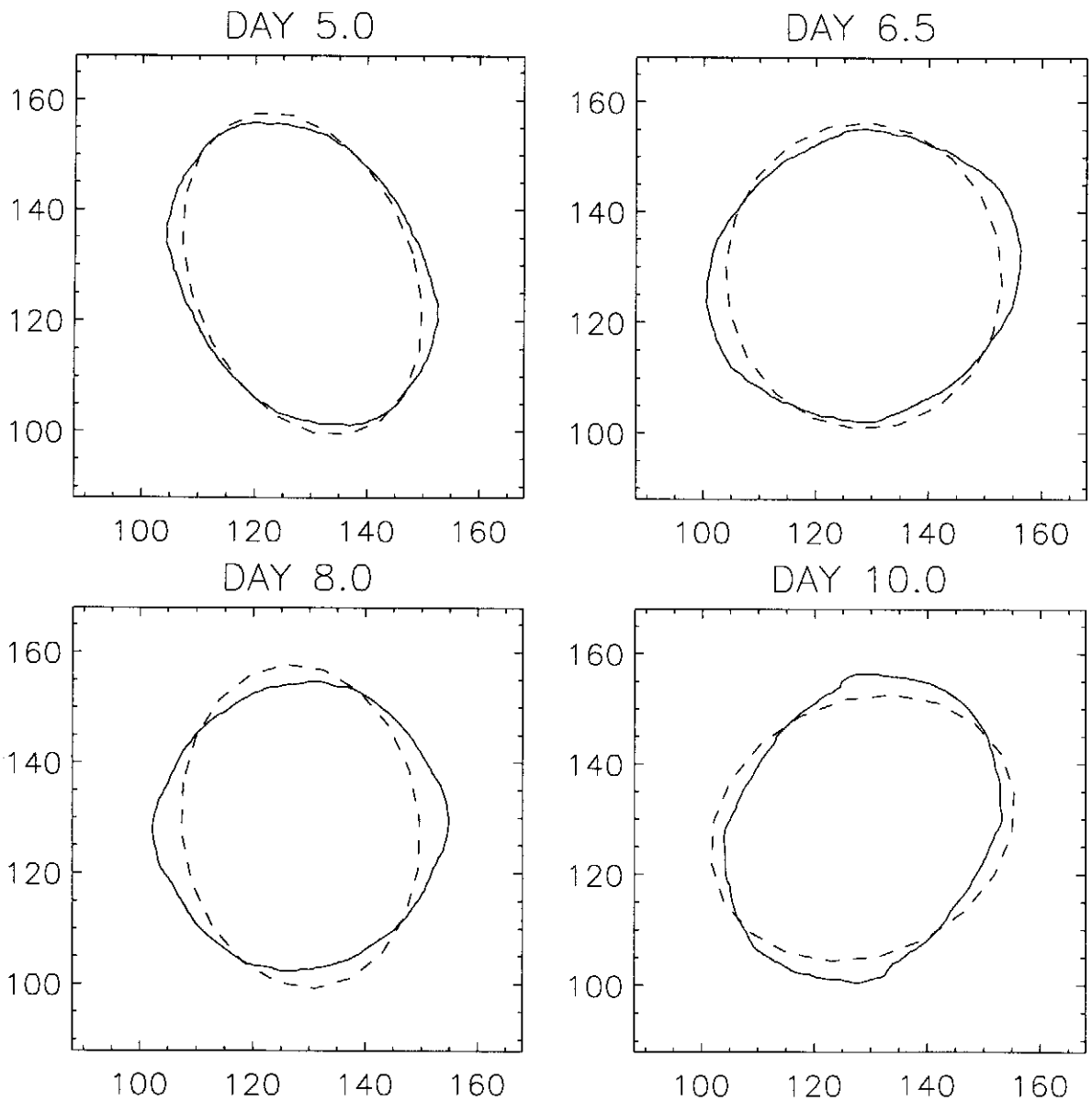

Fig. 11. cont'd.

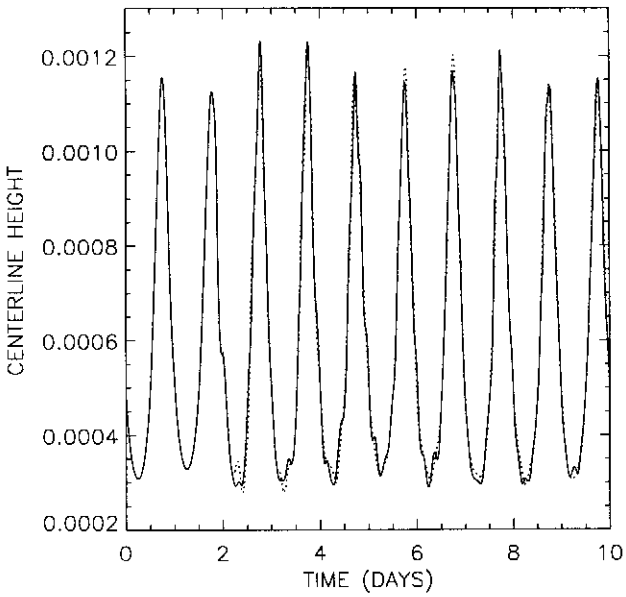

Fig. 12. Comparison, for the deformation case, of the evolution of the centerline height for PIC simulations with approximately 10^6 particles (dotted curve) and 10^5 particles (solid curve). Time is in inertial days.

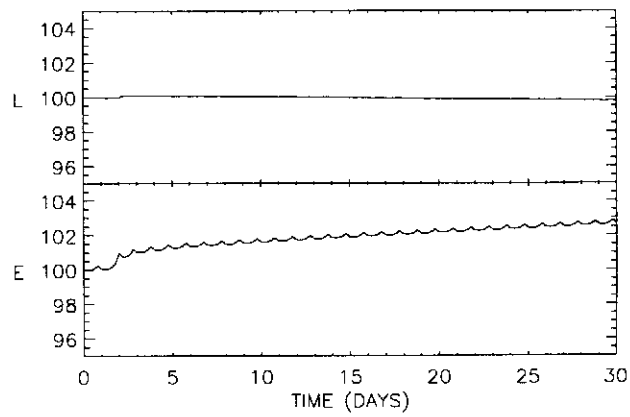

Fig. 13. Time series of total angular momentum (L) and total energy (E) as a percentage of the initial value for the pulson case. Note that only the values at every fifth time step are plotted in this figure. Time is in inertial days.

potential energy which is evaluated on the grid. This is consistent with the PIC algorithm wherein the velocity is only computed at the particles and the height field is only computed on the grid.

Figure 13 shows the time series of L and E for the pulson case, as a percentage of the values at $t = 0$, for a simulation period of 30 inertial days. The angular momentum shows a jump of a little less than 0.2% at about day 2. The jump at day 2 coincides with the offset seen in figures 5 and 9 and a "randomization" of the particle positions. Overall the value of angular momentum after 30 days is within about 0.3% of the initial values. The time series of the energy shows a jump of slightly more than 1% around day 2 and then increases very slowly with increasing time. The total energy has increased by only a maximum of 3.0% after 30 days.

Figure 14 depicts the time series for the scaled kinetic, potential and total energy for the pulson case, plotted every fifth time step. For plotting purposes the energy was scaled in order to give values between zero and one. As expected the kinetic and potential energies oscillate in time but are 90 degrees out of phase and mirror reflections of each other. The amplitude of the potential energy oscillation (solid curve) is very nearly constant. A very small increase in the amplitude of the kinetic energy oscillation (dashed curve) over the 30 days can be seen. This is the major contributor to the small overall increase in total energy.

Figure 15 shows the time series of L and E for the deformation case, again as a percentage of the values at $t = 0$, for a simulation period of 30 inertial days.

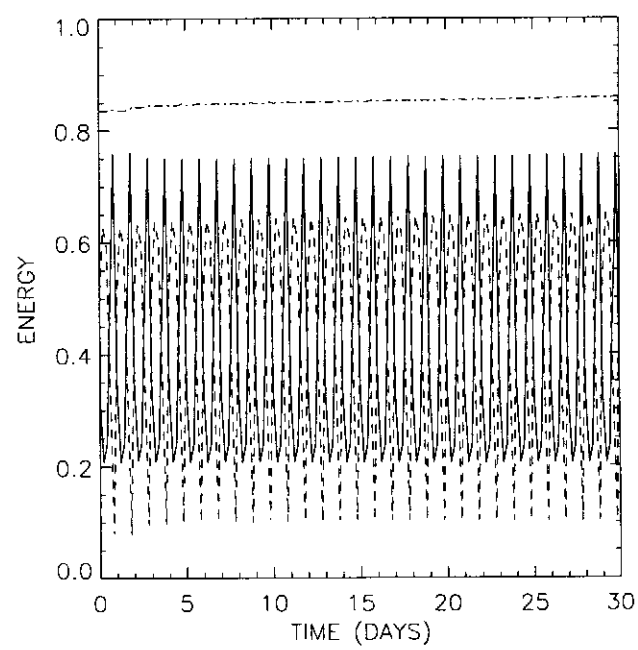

Fig. 14. Time series of kinetic (dashed), potential (solid) and total energy (dash-dot) for the pulson case. Note that only the values at every fifth time step are plotted in this figure. The total energy curve is the same as that shown in figure 13, except that it has been rescaled. Time is in inertial days.

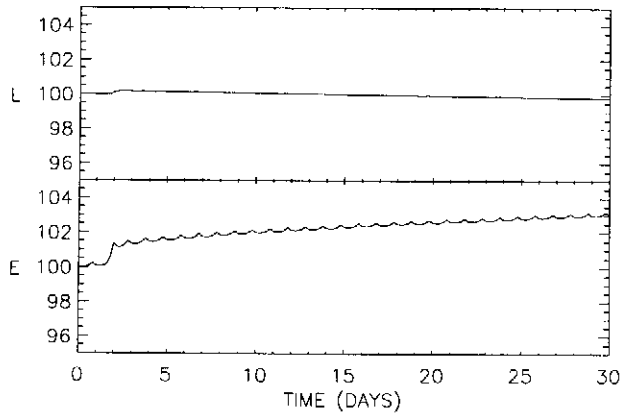

Fig. 15. Time series of total angular momentum (L) and total energy (E) as a percentage of the initial value for the deformation case. Note that only the values at every fifth time step are plotted in this figure. Time is in inertial days.

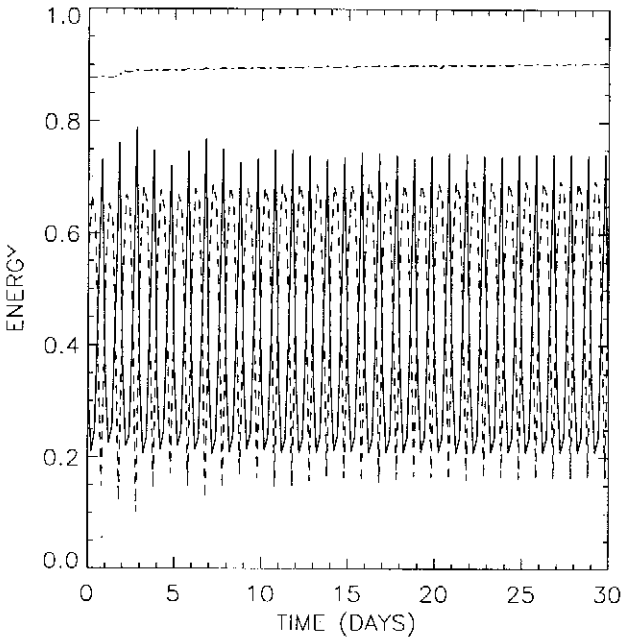

Fig. 16. Time series of kinetic (dashed), potential (solid) and total energy (dash-dot) for the deformation case. Note that only the values at every fifth time step are plotted in this figure. The total energy curve is the same as that shown in figure 15, except that it has been rescaled. Time is in inertial days.

The behavior of L and E for this case is quite similar to that of the pulson case. The angular momentum shows a jump of about 0.2% at day 2. After 30 days the angular momentum is, as in the previous case, within about 0.3% of the initial values. The energy shows a somewhat larger jump around day 2 than in the pulson case. For this case the jump is nearly 1.5%. As in the other case, there is a slow increase in E thereafter. After 30 days the total energy has increased by slightly more than 3.0% in total.

Figure 16 shows the time series for the scaled kinetic, potential and total energy for the deformation case, plotted every fifth time step. Again, scaling was done in order to give values between zero and one for plotting. As for the pulson, the kinetic and potential energies oscillate in time and are almost exactly out of phase. The amplitude of both the potential energy (solid curve) and the kinetic energy (dashed curve) are more variable in this case than in the pulson case. However, the time variation of the total energy in this case is quite similar to that of the pulson.

7 Results with prescribed flow in the lower layer

In this section we discuss one simulation where the lens layer is accelerated by prescribed flow in the lower layer which is very deep relative to the lens thickness. In this case $\mathbf{v_2} = (Sy, Sx)$ where S is the shear deformation rate. From (1B) it is readily shown that,

$$g\frac{\partial(h_1+h_2)}{\partial x} = g_*\frac{\partial h_1}{\partial x} + Sx(S-f),$$

$$g\frac{\partial(h_1+h_2)}{\partial y} = g_*\frac{\partial h_1}{\partial y} + Sy(S+f). \qquad (35)$$

Scaling the shear S by f gives the same nondimensional equations as (4) except for the environmental forcing. Note in this case only total volume and potential vorticity should be conserved. Initialization of the upper layer is done as described previously. With prescribed flow in the lower layer, no initialization is necessary in that layer.

Since the first study by Ruddick (1987) there have been a number of applications of this analytic model. See for example Brickman and Ruddick (1990), Kirwan and Lipphardt (1993), Kirwan et al. (1992, 1994), and Holdzkom et al. (1995). The simulation reported here started with an initially circular lens in an analytic steady state. The nondimensional value of S was 0.01 which is characteristic of mid-ocean values.

Figure 17 shows the time series for h_o for both the PIC model and the analytic lens model started under the same conditions. The agreement is quite good in both phase and in the low frequency modulation of the amplitude.

Figure 18 shows the evolution of the PIC lens boundary over a 25 day period. It is seen that the lens quickly deforms to an ellipsoid and rotates at about 7^o per day. This is quite close to observations of a Gulf Stream ring reported by Holdzkom et al. (1995).

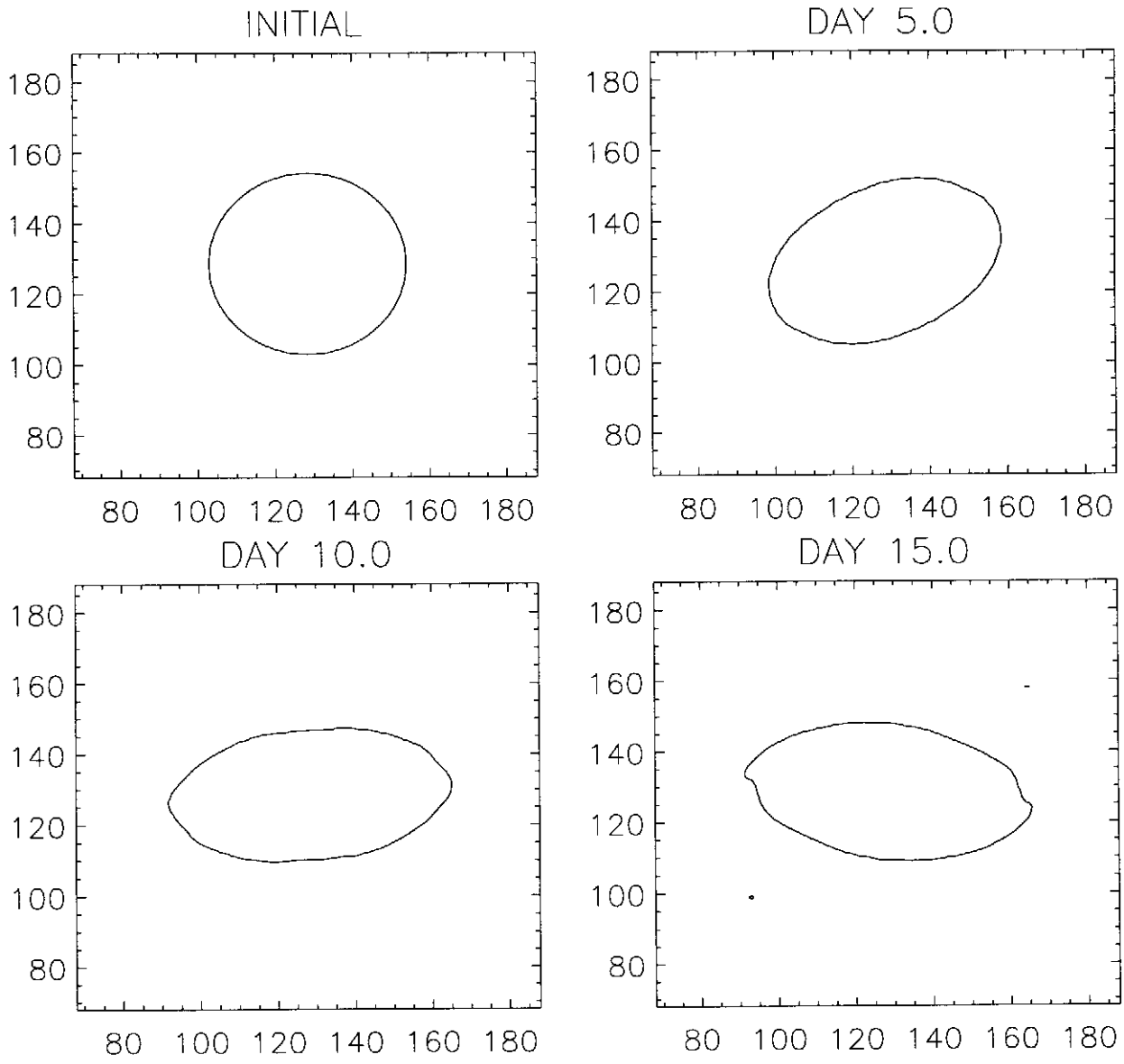

Fig. 18. Evolution of the boundary of the PIC solution over 25 inertial days for the case with prescribed shear in the lower layer. The axes labels are the grid points in each direction.

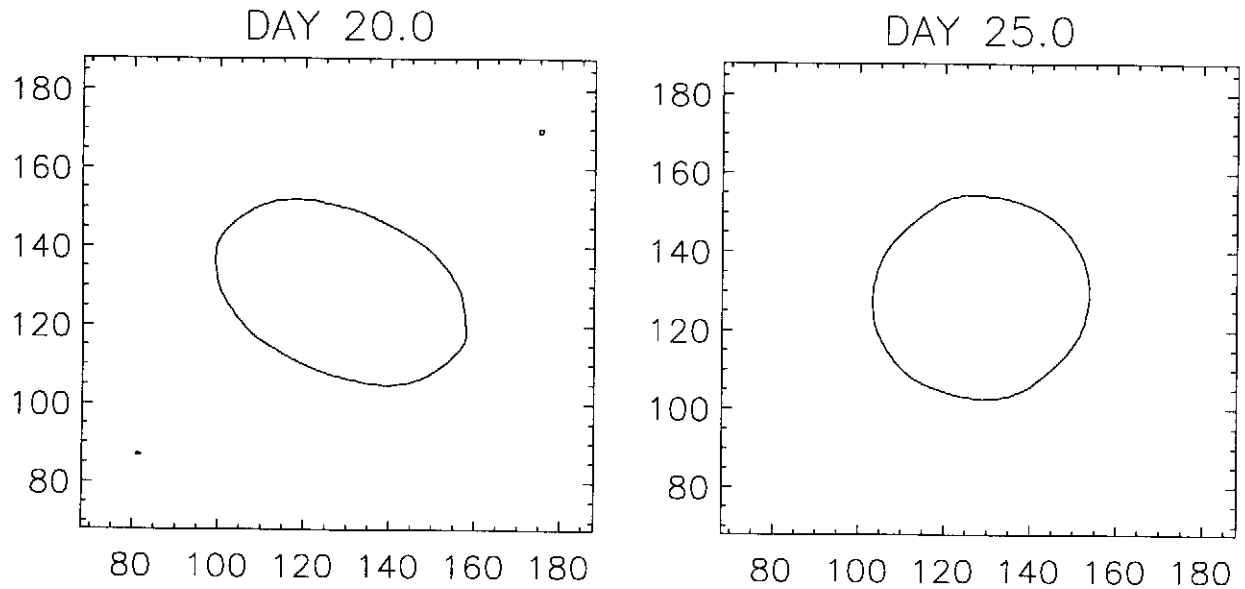

Fig. 18. cont'd.

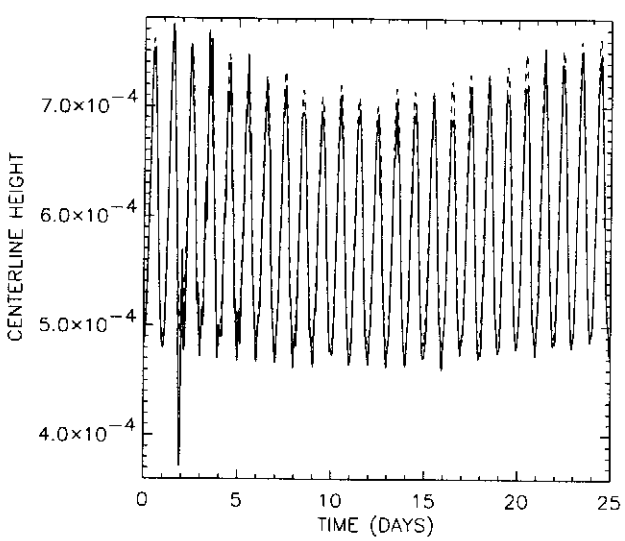

Fig. 17. Comparison of the PIC centerline height with the numerical solution to the lens model for the prescribed lower layer shear case over 25 inertial days. The PIC solution is the solid curve while the lens solution is the dashed curve.

8 Parallel implementation

A significant computational advantage of this method is that it is easily and efficiently parallelized. The PIC solution algorithm can be broken down in the following manner. Each processor is made responsible for only a subset of the particles while retaining a copy of the entire grid. Each processor interpolates the heights of its particles to its copy of the grid. When all processors have finished this, a gather–add–scatter command is issued so that each processor's grid contains the sum of heights from all processors and thus the sum of the interpolated heights from all particles. After computing gradients on the grid, each processor interpolates those gradients back to its subset of particles and updates their position and velocity. When using a few processors, the number of particles ($\approx 10^6$) is generally much greater than the number of gridpoints (256×256), thus there is little advantage to parallelizing the grid operations, as we have confirmed with experiments.

The above scheme has been implemented using the Message Passing Interface (MPI) on the IBM SP2 located at the Maui High Performance Computing Center. MPI is straightforward to implement and is portable and efficient. Timing results are shown in table 2 and in figure 19. These timings are for problems which do not include input/output. We have timed runs which include I/O and found that in most cases I/O adds very little to the run time. However, in a few cases the presence of many other users, all doing I/O, caused a noticeable increase in run time.

We define speedup as the ratio of the runtime with one processor to the runtime with n processors. The results of table 2 and figure 19 show a nearly linear speedup with the number of processors for up to four

Table 2. Timings of PIC model on mutliple nodes of an IBM SP2 for 10^6 particles

Number of Nodes	Length of Simulation (Days)	Time of Run hh:mm:ss	Speedup
1	10.0	2:50:00	$\equiv 1.00$
2	10.0	1:27:30	1.94
4	10.0	0:50:10	3.38
8	10.0	0:31:53	5.33
16	10.0	0:23:45	7.16

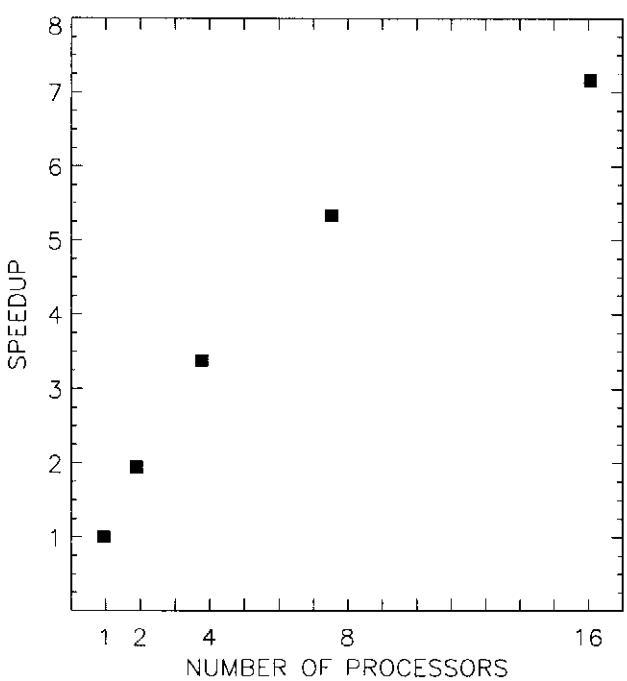

Fig. 19. Speedup of the algorithm as a function of the number of processors.

processors. With more processors, the rate of speedup decreases because of increased communication between nodes. Nevertheless, when using up to sixteen processors, the speedup still increases and there is a reduction in the total execution time. It should be noted that the speedup as we have defined it is not scaled speedup wherein the problem size is increased as the number of processors is increased. Rather, the problem size is the same in all cases. For this reason, larger problems (e.g. multiple layers and/or features) will likely utilize more nodes quite efficiently.

9 Conclusions

The results presented in section 6 showed that the PIC method used here was able to reproduce both the amplitude and phase of the analytic pulson solution with negligible distortion. The case of the deforming lens provided a good test of the method for handling typical evolutions of fronts in which time dependent local horizontal divergence, vorticity and deformation are all important. The time and space scales of the front oscillations would be computationally costly to treat by conventional gridded primitive equation models. Moreover, the PIC method also conserves angular momentum and energy.

The results presented in section 7 demonstrate the robustness of the PIC approach in the presence of environmental shear. The circular lens evolved to an ellipse and rotated anticyclonically at a rate characteristic of observations of oceanic rings. There were ultra low frequency oscillations in the hydrodynamic fields but no evidence of instability.

The overall good agreement between the PIC simulation and the lens model was surprising. To our knowledge there has been no prior comparison of the lens model with a primitive equations models and so there has been some concern about its robustness and applicability to ocean eddies. For the scenarios studied here these concerns are unjustified.

The PIC method is ideally suited for parallel architectures. In fact all of the simulations reported here were done on the IBM SP2 at the Maui High Performance Computing Center. The results for speedup and execution time as a function of number of processors indicate that increasing the number of particles, as will be required for an active two layer model, will utilize parallelism even more efficiently.

The formulation used here assigns to the particles a zone of influence, so that any given Eulerian point is influenced by many particles. Thus, with this formulation it is not appropriate to consider material properties as characteristic of a single particle occupying a particular position at a particular time. Rather, material properties should be viewed as a suitable weighted average of all the particles influencing that position at that partic-

ular time.

It is noteworthy that the results achieved here did not require any explicit smoothing. Of course, the interpolation operation performs an implicit smoothing.

The "event" occurring about day two in both simulations is worthy of some discussion. Up to day two the solutions are very close to those calculated for the analytic lens model, and the particles remain on the annulus on which they were initially distributed. Around day two the particles undergo a rapid "randomization" of their positions. The effect of this is seen in figure 4 as well as in the plots of the centerline height (figures 5 and 9) and in the time series of the integral invariants (figures 13 and 15). Thereafter the particle positions remains randomized. This effect has been noted by others (Pavia and Cushman-Roisin, 1988). After day 2 the time series of the integral invariants are quite similar to those prior to the randomization event. We do not know why the randomization occurs after about two days and not earlier or later. The effect does not occur after a fixed number of time steps; it occurs at about the same *time* regardless of the size of the timestep. It is important to note that the randomization does not appear to affect the stability or quantitative aspects of the simulation such as the invariants.

There are other unresolved issues regarding the utility of this approach. No one has yet introduced a fully dynamically interacting lower layer. By prescribing a quasigeostrophic response in the lower layer Mathias (1992) was able to use conventional gridded methods for obtaining the solution in that region while using PIC methods in the upper layer. This is more realistic than the reduced gravity model but precludes full interaction between the layers. Using PIC methods in both layers would permit general layer interaction. We are currently pursuing this.

Another issue is streaming flows where the particles are advected out of one part of the computational domain and must be replaced by inflow from another part. There is no experience with PIC methods in an oceanographic setting with problems of this sort. This problem is also under investigation.

Finally, it will be important to consider surface stress-driven flow and flow over topography. In these cases it will be necessary to incorporate viscosity, which will require some modifications in the approach used here. If these issues can be addressed successfully, this methodology may allow application to a wide spectrum of important problems that heretofore have been treated only approximately.

Acknowledgements. This study was supported in part by the Office of Naval Research under contract N00014-91-J-1560 and ONR-AASERT contract N00014-93-1-0842. A.D. Kirwan, Jr. and C.E. Grosch acknowledge the Samuel L. and Fay M. Slover endowment to Old Dominion University. The particle geometry used here was suggested to us by R. Hockney. We benefited greatly from discussions with him in September 1994. We also acknowledge the provision of computing time on the IBM SP2 at the Maui High Performance Computing Center. Finally, the authors gratefully acknowledge the technical assistance of K. Gregory.

Research sponsored in part by the Phillips Laboratory, Air Force Material Command, USAF, through the use of the MHPCC under cooperative agreement number F29601-93-2-0001. The U.S. Government is authorized to reproduce and distribute reprints notwithstanding any copyright notation thereon. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Phillips Laboratory or the U.S. Government.

We also wish to thank an anonymous reviewer for a number of constructive criticisms and suggestions.

References

Ball, F. K., Some general theorems concerning the finite motion of a shallow rotating liquid lying on a paraboloid, *J. Fluid Mech.*, *19*, 240–256, 1963.

Ball, F. K., The effect of rotation on the simpler modes of motion of a liquid in an elliptic paraboloid, *J. Fluid Mech.*, *22*(3), 529–545, 1965.

Brackbill, J. U. and Ruppel, H. M., Flip: A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions, *J. Comput. Phys.*, *65*, 314–343, 1986.

Brickman, D. and Ruddick, B., The behavior and stability of a lens in a strain field, *J. Geophys. Res.*, *95*, 9657–9670, 1990.

Cushman-Roisin, B., Exact analytical solutions for elliptical vortices of the shallow-water equations, *Tellus*, *39(A)*, 235–244, 1987.

Cushman-Roisin, B., Heil, W. H., and Nof, D., Oscillations and rotations of elliptical warm-core rings, *J. Geophys. Res.*, *90*(C6), 11,756–11,764, 1985.

Goldsbrough, G. R., The tidal oscillations in an elliptic basin of variable depth, in *Proc. R. Soc. London*, A30, pp. 157–167, 1930.

Harlow, F. H., The particle-in-cell computing method for fluid dynamics, *Meth. Comput. Phys.*, *3*, 319–343, 1964.

Hockney, R. W. and Eastwood, J. W., *Computer Simulations Using Particles*, Institute of Physics, 1992.

Holdzkom, II, J. J., Hooker, S. B., and Kirwan, Jr., A. D., A comparison of a hydrodynamic lens model to observations of a warm core ring, *J. Geophys. Res.*, *100*(C8), 15,889–15,897, 1995.

Kirwan, Jr., A. D. and Lipphardt, Jr., B. L., Coherent flows with near zero potential vorticity, *J. Mar. Sys.*, *4*(2-3), 95–115, 1993.

Kirwan, Jr., A. D. and Liu, J., Shallow water equations on an f-plane, in *Nonlinear Topics in Ocean Physics*, edited by A. R. Osborne, Course CIX, pp. 99–132, International School of Physics, 'Enrico Fermi', North Holland, Amsterdam, pp. 996, 1991.

Kirwan, Jr., A. D., Indest, A. W., Liu, J., and Clark, N., Ring evolution in general circulation models from path analysis, *J. Geophys. Res.*, *95*, 18,057–18,073, 1990.

Kirwan, Jr., A. D., Lipphardt, Jr., B. L., and Liu, J., Negative potential vorticity lenses, *Int. J. Engng. Sci.*, *30*(10), 1361–1378, 1992.

Kirwan, Jr., A. D., Lipphardt, Jr., B. L., and Gregory, K., Nonlinear ocean dynamics, in *The Oceans: Physical-Chemical Dynamics and Human Impact*, edited by S. Majumdar, E. Miller, G. Forbes, R. Schmalz, and A. Panah, The Pennsylvania Academy of Sciences, 1994.

Mathias, B. J., *Simulations of Vortex Evolution and Interaction*

in a Two-Layer Ocean with a Particle Method, Master's thesis, Thayer School of Engineering, Dartmouth College, New Hampshire, 1992.

Olim, M., A truly noninterpolating semi-Lagrangian Lax-Wendroff method, *J. Comput. Phys.*, *112*, 253–266, 1994.

Pavia, E. G., *A Numerical Study of Merging and Axisymmetrization of Oceanic Eddies*, Ph.D. thesis, Florida State University, Gainesville, 1989.

Pavia, E. G. and Cushman-Roisin, B., Modeling of oceanic fronts using a particle method, *J. Geophys. Res.*, *93*, 3554–3562, 1988.

Pavia, E. G. and Cushman-Roisin, B., Merging of frontal eddies, *J. Phys. Oceanogr.*, *20*, 1886–1906, 1990.

Ripa, P., On the stability of elliptical vortex solutions of the shallow-water equations, *J. Fluid Mech.*, *183*, 343–363, 1987.

Rogers, C., Elliptic warm-core theory: The pulsrodon, *Phys. Lett. A*, *138*, 267–273, 1989.

Ruddick, B. R., Anticyclonic lenses in large-scale strain and shear, *J. Phys. Oceanogr.*, *17*, 741–749, 1987.

Young, W. R., Elliptical vortices in shallow water, *J. Fluid Mech.*, *171*, 101–119, 1986.