Q. J. R. Meteorol. Soc. (1997), **123**, pp. 1097–1130

A contour-advective semi-Lagrangian numerical algorithm for simulating fine-scale conservative dynamical fields

By DAVID G. DRITSCHEL[1]* and MAARTEN H. P. AMBAUM[2]
[1] *University of Cambridge, U.K.*
[2] *Royal Netherlands Meteorological Institute, The Netherlands*

(Received 12 March 1996; revised 17 July 1996)

SUMMARY

This paper describes a novel numerical algorithm for simulating the evolution of fine-scale conservative fields in layer-wise two-dimensional flows, the most important examples of which are the earth's atmosphere and oceans. The algorithm combines two radically different algorithms, one Lagrangian and the other Eulerian, to achieve an unexpected gain in computational efficiency.

The algorithm is demonstrated for multi-layer quasi-geostrophic flow, and results are presented for a simulation of a tilted stratospheric polar vortex and of nearly-inviscid quasi-geostrophic turbulence. The turbulence results contradict previous arguments and simulation results that have suggested an ultimate two-dimensional, vertically-coherent character of the flow. Ongoing extensions of the algorithm to the generally ageostrophic flows characteristic of planetary fluid dynamics are outlined.

KEYWORDS: Contour surgery Fast algorithm Potential vorticity Turbulence Vortex dynamics

1. INTRODUCTION

Motivated by the recently recognized significance of fine-scale structure in the dynamical–chemical processes leading to stratospheric ozone depletion (McIntyre 1995), atmospheric scientists have developed a powerful analysis tool called 'Contour Advection' (Waugh and Plumb 1994; Norton 1994). To visualize the likely behaviour of nearly-conservative fields down to scales well below computational grid-scale, Contour Advection (CA) uses the 'Contour Surgery' (CS) algorithm (Dritschel 1988, 1989) to represent the field as contours (field isolevels) on a given surface (normally a surface of constant potential temperature or entropy). Each contour is in turn represented by a variable number of computational points (nodes) spaced according to a function of contour curvature. Each node is moved (advected) by a velocity which, in CA, is interpolated from gridded data, and in CS, is calculated from integrals over the advected contours (something possible only for idealized systems in linear balance—see below). Surgery limits the complexity of the contours by removing filamentary structures smaller than a prescribed scale and topologically reconnecting contours of the same field level or parts of the same contour when they get closer than this prescribed scale. Crucially, the scale of surgery is much smaller—an order of magnitude smaller—than the typical grid-scale of fields obtained by analyses. As a result, CA permits one to 'see' features that are invisible in an instantaneous grid-generated picture of the field.

CA appears to get something out of nothing. First of all, it must start with a coarse representation of the field, whereas the true field is believed to contain many filamentary features, like those revealed by a CA simulation over several days. Evidently, the general stretching and thinning of filaments means that the missing filaments would largely be seen as a hairy filamentary microstructure superimposed on a newer, fatter filamentary structure developing from the original coarse representation of the field. Secondly, CA ignores variations in the velocity field below the analysis grid-scale, while it retains variations in the advected field well below the grid-scale (Waugh and Plumb 1994 demonstrate the

* Corresponding author: Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Silver Street, Cambridge CB3 9EW, U.K. e-mail: dgd@amtp.cam.ac.uk.

robustness of this approach). In fact, observed velocity spectra do decay rapidly with diminishing scale (Gage and Nastrom 1985), but it is critical that this decay be more rapid than that for the advected field, and it is for most fields of interest, in particular for the potential vorticity (PV) field, the field which, through the 'invertability principle', largely determines the velocity and temperature fields (Hoskins *et al.* 1985). The mathematical justification for this assumption, reviewed below, is related to the PV's dependence on spatial derivatives of the velocity field. That CA works, and works surprisingly well, is proven on real data in Waugh *et al.* (1994) and in Plumb *et al.* (1994).

The algorithm described in this paper takes CA one step further by internally calculating the advecting velocity field from a coarse-grained picture of the PV, which is otherwise held in a contour representation. Thus, the assumption is made that the fine-scale PV contributes negligibly to the advecting velocity field. This assumption may, however, be justified along the lines just given. We do not require that the PV entirely determines the velocity field (as it must in CS), and hence the idea may be more widely applicable to realistic atmospheric and ocean models. In the present algorithm for quasi-geostrophic flow though, it does.

The essential new step is to convert, rapidly, the fine-scale Lagrangian representation of the PV field into a coarser-grained Eulerian representation, which may be manipulated (perhaps along with other fields) to obtain the velocity field again in the Eulerian representation. The latter can be used as in CA to move the PV field in Lagrangian form.

This hybrid algorithm is called the Contour-Advective Semi-Lagrangian (CASL) algorithm. It is not the semi-Lagrangian scheme now commonly used in weather-forecasting models (see Staniforth and Coté 1991 or Gravel 1996 and refs.); that scheme carries out a space and time interpolation to approximate the advective derivative ($\partial/\partial t + \mathbf{u} \cdot \nabla$) permitting models to use a larger time-step. Fields are not held in a contour representation; rather, the object is to determine the trajectory of a fluid particle that arrives at a given grid point after one time-step. This gives rise to a nonlinear implicit problem, whose solution can be costly for large time-steps. In the present CASL algorithm, the time-stepping is fully explicit, since the underlying grid is used only to provide the advecting velocity field. As a result of this stability, the time-step is relatively large, being on the advective time-scale, and it is chosen only for accuracy.

The CASL algorithm has some remarkable properties. There is no lateral diffusion of PV; this is because such diffusion, often *ad hoc*, is here unnecessary for numerical stability. Rather, the CASL algorithm introduces another model for sub-grid processes in which filamentary structures are retained down to a scale much smaller than the grid-scale. In other words, sub-grid processes are treated explicitly. Consequently, flow diagnostics are sensible down to the scale of surgery. The only question is how to choose the scale separation between the grid-scale and the scale of surgery. One can appeal to the rapid decay of the energy spectrum (compared to that of the PV) as well as to numerical efficiency considerations (see section 3(*c*) below), but it is perhaps more reliable to actually examine the effect of this scale separation in realistic flows. This was done originally by Waugh and Plumb (1994) in the context of stratospheric dynamics and recently by Methven (1996) in the context of baroclinic life-cycles. It was concluded that it is appropriate to retain scales of PV up to ten times finer than the smallest scale of the advecting velocity field. In fact, this scale-separation factor is consistent with spectral and efficiency considerations.

The hybrid character of the CASL algorithm is similar in spirit to the so-called particle-in-cell (PIC) algorithm (Christiansen 1973). In fact the two methods differ essentially only in their Lagrangian parts: in the PIC algorithm, the vorticity field is represented by a cloud of point vortices (delta functions of the vorticity), instead of by contours. This singular representation, however, has several disadvantages: there is no canonical way of describing the field in terms of delta functions; there is no analogy of area conservation or topology conservation (no notion of symplectic structure); and there is no natural way of adapting resolution to maintain an accurate representation for the vorticity field. Furthermore the CASL algorithm appears to be more general in the way it can be extended to non-balanced systems. The price one has to pay, to represent the field in terms of contours, is a much less trivial conversion of the Lagrangian field to the Eulerian field. But though the conversion is less trivial, it is still not very costly.

The following section describes the CASL algorithm in detail for multi-layer quasi-geostrophic (QG) flow. Section 3 gives the results of various numerical tests which indicate how best to choose the numerical parameters for a desired level of accuracy. Section 3 also gives a break-down of the computational costs for a typical application, here to the polar stratospheric vortex. Section 4 presents the results of a simulation of nearly-inviscid QG turbulence, revealing not only complexity as never seen before in a numerical simulation, but also the emergence and persistance of previously unexpected vortical structures. Section 5 discusses a proposed extension of the algorithm to ageostrophic flows, in particular to the shallow-water equations. Section 6 presents the conclusions, including remarks on the addition of weakly non-conservative effects.

2. The algorithm for quasi-geostrophic flow

Quasi-geostrophy (Pedlosky 1979; Houghton 1986) is a useful, commonly employed approximation valid for a rapidly rotating, stratified fluid (Stegner and Zeitlin 1995). It leads to a system with rich dynamics (see, e.g. McWilliams 1989, 1990; Dritschel and de la Torre Juárez 1996), much of which carries over to the more complete, ageostrophic systems governing atmospheric and oceanic dynamics (Vallis 1996). The CASL algorithm is not restricted to the QG system, but it is a useful starting point to illustrate the method. Furthermore, the CS algorithm may also be employed (Dritschel and Saravanan 1994) in this case, allowing one to establish the CASL algorithm's conservation properties. Parts of the algorithm which require a more general treatment for ageostrophic systems are indicated and returned to in section 5.

The attractive, simplifying feature of quasi-geostrophy is the omission of high-speed gravity waves, which, in ageostrophic models, severely erode numerical efficiency by forcing a small time-step. To combat this problem, general circulation models used in weather forecasting artificially slow down gravity waves by using a semi-implicit time-stepping scheme (Simmons *et al.* 1978) coupled with a larger time-step. The fastest gravity waves have the shortest wavelengths and are considered numerical noise, something to be eliminated as far as possible. This is not an easy task, and the present strategy is to employ semi-implicit time-stepping, rather than an expensive-to-implement 'balanced' scheme, in which the gravity waves are diagnosed, not evolved (M. E. McIntyre and W. A. Norton, personal communication). The QG system employs linear balance, which is efficient, but not universally accurate (Stegner and Seitlin 1995). Linear balance means that there exists a linear-operator relationship between the QG potential vorticity (PV) and the streamfunction, whose spatial derivatives give the velocity and the perturbation (potential) temperature or density fields.

The equations governing QG flow (Hoskins *et al.* 1985) are given by

$$\frac{\mathrm{D}q}{\mathrm{D}t} \equiv \frac{\partial q}{\partial t} + \mathbf{u}\cdot\nabla q = 0 \qquad (1a)$$

$$f + \nabla^2\psi + \frac{1}{\rho_0}\frac{\partial}{\partial z}\left(\rho_0\frac{f_0^2}{N^2}\frac{\partial\psi}{\partial z}\right) = q \qquad (1b)$$

$$\frac{\partial\psi}{\partial z} = 0 \quad \text{at } z = 0 \text{ and } H \tag{1c}$$

$$u = -\frac{\partial\psi}{\partial y} \quad \text{and} \quad v = \frac{\partial\psi}{\partial x} \tag{1d}$$

where ψ is the streamfunction, u and v are the horizontal velocity components in the x and y directions, q is the QG PV, a scalar function of x, y, z and time t, $\rho_0(z)$ is the basic-state density profile with height z (actually, z is log-pressure, the fluid being in hydrostatic equilibrium), f is the Coriolis parameter (twice the planetary angular velocity projected normal to the planet's surface), and $N(z)$ is the buoyancy frequency of a neutral particle undergoing vertical oscillations. A β-plane (Cartesian) geometry is adopted for simplicity, in which $f = f_0 + \beta y$ (the notation is the same as adopted in a previous work, Dritschel and Saravanan 1994).

Following Hoskins *et al.* (1985), we have replaced the generally inhomogeneous boundary conditions at the rigid surfaces $z = 0$ and $z = H$ that arise from variable surface temperature distributions and topography by homogeneous ones and equivalent sheet distributions of PV, γ_B and γ_T, at these surfaces. That is, q includes, generally, $\gamma_B(x, y, t)\delta(z) + \gamma_T(x, y, t)\delta(z - H)$, and, like q, $D\gamma_B/Dt = 0$ and $D\gamma_T/Dt = 0$. Topography, $z = h(x, y) \ll H$ (for QG theory to be valid) corresponds to a fixed distribution of surface PV, $-fh(x, y)/H$, and, often, a free distribution of surface PV, initially equal to $+fh(x, y)/H$, to compensate for the topographic PV, but thereafter different as a result of advection. Similarly, one can specify the flow (specify ψ) at either surface through an equivalent double-sheet distribution of PV along with the boundary condition $\psi = 0$ for the homogeneous problem. In practice, the fluid is divided into a finite number of layers or levels, and these singular PV distributions are spread over the layers adjacent to the surfaces, with an error proportional to the square of the layer thickness (see, e.g. Dritschel and Saravanan 1994).

One can also consider a free surface at $z = H + \eta(x, y, t)$, $\eta \ll H$, which changes the boundary condition at $z = H$ to $\partial\psi/\partial z + N^2\psi/g = 0$, where g is the acceleration due to gravity (or reduced gravity in the oceanic context).

The horizontal boundary conditions to be adopted are doubly-periodic. A spherical surface may seem more natural, but QG theory breaks down in equatorial regions, where $f \to 0$, because one cannot guarantee $(q - f)/f \ll 1$ there. Previous studies using CS (Dritschel and Saravanan 1994, Dritschel and de la Torre Juárez 1996) adopted an unbounded f-plane ($\beta = 0$), to focus on the dynamics of the polar vortex (McIntyre 1995) and other smaller, intermediate-scale vortices found in mid-latitudes. Here, because the algorithm relies in part on an underlying grid, it is necessary (without coordinate stretching) to use a finite geometry, and a box is the simplest, most common one employed. Some applications, such as flow in an idealized ocean basin, might call for rigid horizontal boundaries, but such boundary conditions turn out to be easier to handle than doubly-periodic ones for the CASL algorithm. The modifications that would be necessary as a result of changing the boundary conditions are indicated below.

Without loss of generality, we consider the square periodic box, $-\pi < x \leqslant \pi$ and $-\pi < y \leqslant \pi$. This presumes a coordinate scaling. Normally, x and y are scaled on the 'Rossby radius of deformation', L_R, equal to $N_T H_\rho/f_0$, where $N_T \equiv N(z = H)$, and z is scaled on H_ρ. Here H_ρ is the 'density-scale height', defined from $\rho_0(z) = \rho_{00}\exp(-z/H_\rho)$ for a 'compressible' fluid; otherwise, for a 'Boussinesq' fluid (constant ρ_0), $H_\rho = H$. It is sometimes useful to further scale all three coordinates by a constant c; this is equivalent to putting $L_R = c$.

The steps in the CASL algorithm are described in the following subsections.

(*a*) *Initialization*

Before the simulation begins, a script is executed to choose the vertical structure of the fluid (ρ_0 and N), its total depth H, the number of discrete vertical levels or layers n_v (see Dritschel and Saravanan 1994), the number of vertical modes $M \leqslant n_v$ (see section 2(*d*) below), various contour-surgery parameters (μ, L) controlling the node distribution along contours and surgery (Dritschel 1989), the time-step Δt, the horizontal grid resolution n_h, the PV inversion/conversion grid ratio m_g (see section 2(*c*) below), the planetary vorticity gradient β, the bottom topography $h(x, y)$, and, of course, the PV distribution itself. The latter is specified as it normally is in CS, that is, in contour form, in each horizontal layer or level. One may alternatively contour a given continuous distribution of PV. Contour levels are assumed to be separated by a uniform step increment, Δq. This simplifies the graphical visualization but is not required by the algorithm. Between contours, the PV is uniform, so the contours represent jumps in PV, as in CS. This is not a limitation in practice: piecewise-uniform PV gives rise to a velocity field scarcely distinguishable from that arising from a continuous distribution, as demonstrated in an earlier study (Legras and Dritschel 1993). Indeed, the unimportance of fine-scale PV is at the heart of the present algorithm.

The simulation begins by reading the input data, initializing frequently used constants (e.g. FFT cosine and sine tables, see below), calculating the average PV in each layer directly from the contours (this average does not change in time), and then entering the main loop over time-steps. A fourth-order Runge–Kutta time-stepping scheme is used, so every time-step requires four evaluations of the velocity field. The latter is obtained by executing the following four steps (*b*)–(*e*).

(*b*) *PV contour-to-grid conversion*

The CASL algorithm hinges on the existence of a fast method for converting a contoured field into a gridded field. Normally, the reverse is done for producing pictures from the results of conventional simulations, where the fields are retained in a Eulerian representation. The method developed requires an O(n) number of operations, where n is the total number of nodes on the PV contours, and it is highly vectorizable.

Each layer or level in the domain is divided into $n_h \times n_h$ equal squares of side length $\Delta = 2\pi/n_h$. The lower-left corner at $\mathbf{x} = \{-\pi, -\pi\}$ is indexed (1, 1), and the upper-right corner at $\mathbf{x} = \{\pi, \pi\}$ is indexed $(n_h + 1, n_h + 1)$. Of course, due to periodicity, only the field values indexed between 1 and n_h can be distinct.

Let i, j and l denote the grid point $\bar{x}_i$, $\bar{y}_j$ and $\bar{z}_l$, where l denotes the layer or level and runs from 1 to n_v. $\bar{z}_l$ is either the mid-layer height or the level height (Dritschel and Saravanan 1994). Henceforth, these are just called 'layers'.

First, from contours passing through the left face of the domain, that is through $x = -\pi$, the PV difference, $\Delta_y q_{j,l}$, between grid points $(j, 1)$ and $(j + 1, 1)$ is calculated for all layers l. The procedure uses the fact that the PV-jump, Δq, across a contour is always equal to the PV to the left of the contour minus the PV to the right of it. This directionality rule is in fact preserved during the contour reconnections that might take place as a result of surgery. Hence, a contour which passes through the left face between j and $j + 1$ adds $s\Delta q$ to $\Delta_y q_{j,l}$, where $s = +1$ for a left to right crossing and -1 for a right to left crossing.

Next, a loop over layers is entered and is not exited until procedure (*d*) below. The gridded PV, $\bar{q}_{j,i}$, is determined to within a constant in each layer along the left wall ($i = 1$) by summing these PV differences. (Here the index l is suppressed; it indicates where it may also be suppressed in the numerical algorithm.) Starting with $\bar{q}_{1,1} = 0$, we obtain $\bar{q}$ recursively: $\bar{q}_{j+1,1} = \bar{q}_{j,1} + \Delta_y q_{j,l}$.

A similar procedure is then followed to obtain the interior PV. Adjacent nodes $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$ on a contour are examined to see if they cross at least one grid line, $y = \bar{y}_j$. This happens if the integer value of y_k/Δ differs from that of y_{k+1}/Δ. Using linear interpolation between nodes, i.e. $\mathbf{x} = \mathbf{x}_k + p\Delta\mathbf{x}_k$, where $\Delta\mathbf{x}_k = (\mathbf{x}_{k+1} - \mathbf{x}_k)$ modulo π, the crossing j closest to y_k is found, and the following quantities are stored: j, x (the first x crossing), Δx (the increment in x between crossings), the number of crossings, and the signed PV jump, $s\Delta q$, with $s = -\mathrm{sign}\Delta y_k$. In the new loop over k, these quantities are used to construct the horizontal PV jump array $\Delta_x q_{j,i}$, the jump in PV along a grid line $y = y_j$ between grid points i and $i+1$. Starting with the stored j and x values, i is obtained by the integer value of $(x+\pi)/\Delta$, and $s\Delta q$ is added to $\Delta_x q_{j,i}$. If there is more than one crossing, s is added to j and Δx is added to x, and the procedure for finding i and $\Delta_x q_{j,i}$ is repeated; otherwise, the next k is considered.

Finally, the PV at each grid point is obtained by summation, $\bar{q}_{j,i+1} = \bar{q}_{j,i} + \Delta_x q_{j,i}$, and, if $\beta \neq 0$, βy is subtracted. The PV so obtained will normally be correct only to within a constant, so the average value must be calculated and corrected using the initially calculated average value. When using spectral techniques, the procedure is even simpler than this, as described in the next step.

For flow in a channel, periodic in x, the above procedure is essentially unchanged. It is slightly simpler, in fact, since one does not need to use modulo arithmetic in the y direction. For flow in a box with rigid boundaries (and free slip—there can be no generation of PV), the procedure is much simpler, since PV contours must close within the domain. Hence, one can build the PV along the left wall (actually, just inside it) simply by summing the jumps of all contours running along this wall. The rest of the procedure is the same, apart from not having to do modulo arithmetic.

The procedure can be adapted, with minor modification, to irregular flow domains by overlaying a rectangular grid.

(*c*) *PV averaging*

At this stage, it is possible to pass directly to the next step and obtain the gridded velocity field. However, it turns out to be much more economical to average first the gridded PV field (perhaps several times) onto a grid m_g-times coarser, of grid length $\bar{\Delta} = m_g\Delta$, where m_g is a power of 2. Each iteration of this procedure doubles the grid scale-and halves the number of grid points in each direction by using a standard nine-point weighted average of the grid points surrounding and including each target grid point. The averaging can be done in a minimum of operations and results in substantial overall savings, since the PV inversion (see next step), the most costly step in the whole algorithm, can be done on a coarse grid of horizontal dimension $\bar{n}_h \times \bar{n}_h$, where $\bar{n}_h \equiv n_h/m_g$. Remarkably, this averaging leads to surprisingly little loss of accuracy, as shown below in section 3(*a*). Evidently, the averaging preserves the characteristics of the PV distribution most important to inversion.

(*d*) *Inversion*

In QG flow, the PV completely determines the velocity field. This is called the 'invertability principle' (Hoskins *et al.* 1985) or 'inversion' for short. Inversion is particularly simple in QG flow since a linear operator links the PV with the streamfunction (see Eq. (1b)). For other, ageostrophic systems, the procedure below would have to be replaced, as discussed in section 5.

The inversion is done as follows. The grid-averaged PV, $\bar{q}_{j,i}$, is passed through a two-dimensional Fast Fourier Transform (FFT) to obtain the spectral coefficients for the PV in the lth layer, $\check{q}_{kl}$, $k = 0, 1, \ldots, \bar{n}_{\mathrm{h}} - 1$, $l = 0, 1, \ldots, \bar{n}_{\mathrm{h}} - 1$. Note that in the previous steps the gridded PV was constructed only to within a constant: the actual value of $\bar{q}_{1,1}$ was unknown (it turns out that this value is very hard to determine for a doubly-periodic domain). The coefficient $\check{q}_{00}$, proportional to the mean value of the PV in the lth layer, is therefore set equal to the correct mean value calculated in the programme initialization, step (a) above. Any fixed topographic PV in the lowest layer is transformed only once initially (unless it was already provided in spectral form) and added to the free transformed PV there.

A vertical transform is then effected by projecting these coefficients onto the vertical modes for the operator in (1.2) (see Dritschel and Saravanan (1994) for full details). The vertical modes, $\{\widehat{\varphi}_1, \widehat{\varphi}_2, \ldots, \widehat{\varphi}_M\}$, are the discrete solution of the Sturm–Liouville eigenproblem,

$$\frac{1}{\rho_0}\frac{\mathrm{d}}{\mathrm{d}z}\left(\rho_0 \frac{f_0^2}{N^2}\frac{\mathrm{d}\varphi_m}{\mathrm{d}z}\right) + \alpha_m^2 \varphi_m = 0 \tag{2a}$$

$$\frac{\mathrm{d}\varphi_m}{\mathrm{d}z} = 0 \quad \text{at } z = 0 \text{ and } H \tag{2b}$$

where the α_m, $m = 1, 2, \ldots, M$, are the eigenvalues, often referred to as the 'inverse radii of deformation', corresponding to the eigenmodes φ_m. By virtue of Eq. (2b), $\alpha_1 = 0$ and φ_1 = constant—this is called the 'barotropic mode'. The lth element of the *discrete* eigenvector $\widehat{\varphi}_m$ is approximately equal to $\varphi_m(\bar{z}_l)$. Both the eigenvectors $\widehat{\varphi}_m$ and their orthonormal counterparts $\widehat{\varphi}'_m$ (obtained from the transposed inverse of the matrix of eigenvectors) are calculated upon initialization. These are used to carry out the (not-so-fast) vertical transforms. In particular, the contribution of the lth level to the fully-transformed PV, $\widehat{q}_{klm}$, is given by $\check{q}_{kl}\varphi'_{lm}$, for $m = 1, 2, \ldots, M$.

Note that the number of modes retained, M, may be less than n_{v}. Fine vertical scales, like fine horizontal scales, contribute little to the advecting velocity field. Differential horizontal advection will produce much fine-scale vertical structure, which may be explicitly resolved using a large n_{v}. The dynamical insignificance of these fine-scale structures permits choosing M smaller than n_{v}.

The transformed streamfunction is then simply obtained by division,

$$\widehat{\psi}_{klm} = -\widehat{q}_{klm}/(k^2 + l^2 + \alpha_m^2) \tag{3}$$

which represents the principle advantage over CS, which must use costly contour integrations. Furthermore the spectral inversion is done mode-by-mode, whereas in CS the inversion involves contour-integrals over all layers. At this point, another loop over layers is entered to recover the physical velocity field in each layer. This loop is not exited until the completion of the CA step which follows.

The transformed streamfunction $\check{\psi}_{kl}$ in layer l is first obtained by projection onto the vertical eigenvector:

$$\check{\psi}_{kl} = \sum_{m=1}^{M} \widehat{\psi}_{klm}\varphi_{lm}. \tag{4}$$

This is then multiplied by $-il$ and ik to get the corresponding transformed velocity field:

$$\check{u}_{kl} = -il\check{\psi}_{kl} \quad ; \quad \check{v}_{kl} = ik\check{\psi}_{kl}. \tag{5}$$

Both of these fields are passed through a two-dimensional FFT to obtain the gridded velocity field in layer l, $\bar{\mathbf{u}}_{j,i}$, $i = 1, 2, \ldots, \bar{n}_{\mathrm{h}}$, $j = 1, 2, \ldots, \bar{n}_{\mathrm{h}}$. This completes the inversion step.

(*e*) *Contour advection*

The next step is to interpolate the gridded velocity field just obtained to get the velocity at all nodes on all PV contours. As in previous work (see, e.g. Waugh and Plumb 1994), bi-linear interpolation is used within a single grid-box. This is simple and, moreover, it is accurate (see, in particular, section 3(*a*) below). The same interpolation is widely used in semi-Lagrangian schemes where accuracy is essential (Staniforth and Coté 1991, Gravel 1996).

At this point, the loop over layers is exited. The velocity on the PV contours is known. The contours are moved slightly, and it is necessary then to repeat steps (*b*) through (*e*) until all four stages of the Runge–Kutta integration are completed.

(*f*) *Surgery*

Every other time-step, or about every twentieth of the 'rotation' period T=$4\pi/|q - f|_{\max}$ associated with the maximum PV anomaly, the contours in each layer are examined for possible topological reconnections, or 'surgery' (Dritschel 1988, 1989). Surgery is performed when the distance between two contours containing the same PV or two parts of the same contour becomes closer than the 'cut-off scale', δ, equal to $\frac{1}{4}\mu^2 L$, where μL is approximately the maximum spacing between adjacent nodes on a contour, and L is a prescribed large-scale length characterizing the PV structures in the flow. The two basic operations are illustrated in Fig. 1 (these differ slightly from the original operations described in the above references to improve efficiency). The one in Fig. 1(a) results in either two contours from one (fission) or one contour from two (fusion), while the one in Fig. 1(b) shortens the ends of filaments.

Surgery involves a search for sufficiently close nodes, and this search can be costly (proportional to the square of the number of nodes in each layer) unless care is taken to minimize the number of contour pairs that need to be examined. This is done in the present algorithm by pre-sorting the contours by layer and by PV level; only contours having the same PV level (same PV on their left and same PV on their right) may possibly reconnect.

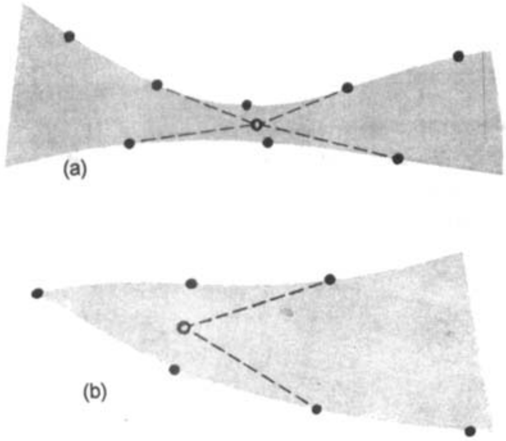

Figure 1. Illustration of how surgery is performed. (a) Surgery between two close contours or contour parts. (b) Surgery near a corner. The solid contours and nodes are the pre-surgery ones and the dashed contours and open circles are the post-surgery ones. A new, double node midway between the nodes on the top and bottom contours or contour parts replaces the original two nodes. This double node is (momentarily) shared by the left and right contour sections. The contour sections in (a) may subsequently separate by the corner surgery illustrated in (b), in which nodes near the corner are removed.

The new surgical procedure preserves this ordering and minimizes array shifts. In addition, for contours in the same layer and of the same PV level, a contour-based test is applied to eliminate contour pairs that could not possibly reconnect. Before surgery, the minimum and maximum x and y coordinates of each contour are calculated and used to compute the centre of each contour's bounding rectangle (X, Y), its half-width w and its half-height h (taking into account periodicity). Then, when considering the potential fusion of two contours C and C', a point-wise search is conducted only if both

$$|X' - X| < h + h' + \delta$$

and

$$|Y' - Y| < w + w' + \delta$$

are satisfied (again taking into account periodicity). Such tests even work for 'periodic' contours (such as those representing a planetary vorticity gradient), whose half-widths equal π. In practice, this is found to greatly improve numerical efficiency, since relatively few contour pairs satisfy the above conditions.

Surgery is absolutely necessary to control the exponential build-up of fine-scale PV. Any numerical simulation of significant duration requires a regularization at small scales. Surgery appears to be a relatively benign procedure in comparison with, for instance, hyperviscosity (Mariotti *et al.* 1994; Yao *et al.* 1995; Macaskill and Bewick 1995), which is widely used, however, in atmospheric and oceanic numerical models. Moreover, surgery in the present CASL algorithm typically operates at a tenth of the (inversion) grid-scale, and consequently one may expect that features in CASL are reliable down to the grid-scale, something which is certainly not the case in conventional grid-based models.

(*g*) *Node redistribution*

Following surgery, nodes on each contour are redistributed to account for increasing or decreasing contour curvature and length (Dritschel 1988, 1989). The way in which the number of nodes per unit length is computed in the present algorithm differs significantly from that described in the above references, so it is given here.

The mean curvature value κ_k between nodes k and $k+1$ is provided for each k by the local cubic-spline interpolation routine (as in the above references). From this, the algorithm calculates $\widetilde{\kappa}_k = \sqrt{\kappa_k^2 + 1/L^2}$, which is the curvature the contour would have if it were lying on a sphere of radius L (recall that L is a pre-set characteristic length-scale of the PV distribution). This is done to keep the spacing of successive nodes smaller than μL. From $d_k \equiv |\mathbf{x}_{k+1} - \mathbf{x}_k|$, a weight $w_k = d_k/(d_k^2 + 4\delta^2)$ is computed and used to get an intermediate averaged curvature value $\check{\kappa}_k$ at node k, namely

$$\check{\kappa}_k = \frac{w_{\kappa-1}\widetilde{\kappa}_{k-1} + w_k\widetilde{\kappa}_k}{w_{k-1} + w_k},$$

which itself is averaged with the value at node $k+1$ to get the final interval curvature value $\bar{\kappa}_k$ (effectively, the curvature at four nodes determine $\bar{\kappa}_k$). The number of new nodes to be placed per unit length is then computed from

$$\frac{(\bar{\kappa}_k L)^{1/2}}{\mu L} + \bar{\kappa}_k$$

or $2/\delta$, whichever is smaller (nodes cannot get closer than $\delta/2$).

A summary of the CASL algorithm is given in Table 1.

TABLE 1. A. FLOW CHART OF THE CASL ALGORITHM

1. Initialization
   a. Choose vertical structure (ρ_0, N), total depth (H), number of vertical levels (n_v), number of vertical modes (M), horizontal resolution ($\bar{n}_h$), the averaging factor (m_g), contour resolution parameters (μ, L), time step (Δt), planetary vorticity gradient (β), and bottom topography ($h(x, y)$).
   b. Compile the programme, putting in the required dimensions.
   c. Read in the PV contours at each level.
   d. Calculate the average PV at each level.
   e. Sort contours in each layer by PV level.
   f. Redistribute the nodes.
2. Advection
   a. Calculate the velocity field (see flow chart B).
   b. Advect the nodes as prescribed by the Runge–Kutta scheme.
   c. Perform steps a and b three more times to complete a Runge–Kutta time-step.
3. Surgery
   a. Every other time-step, perform surgery (section 2(*f*)).
4. Post-surgery
   a. Redistribute the nodes.
   b. Periodically save data for post-processing.
   c. Return to item 2, unless the desired number of time-steps have been taken.

B. FLOW CHART OF THE VELOCITY CALCULATION

1. PVCGC
   a. PV contour-to-grid conversion (section 2(*b*)), to obtain the PV field on a grid with resolution $(m_g\bar{n}_h)^2$ at each level.
   b. If desired, perform diagnostics on this PV field.
   c. Subtract βy from the PV field if $\beta \neq 0$.
   d. Perform $\log_2 m_g$ averaging iterates to obtain the PV field (minus βy) on a grid with resolution $\bar{n}_h^2$ at each level.
2. Inversion
   a. FFT of PV field at each level to obtain transformed PV field at each level.
   b. Make (0,0) component of transformed PV field equal to the pre-computed averages (flow chart A, item 1d).
   c. If present, add the topographic contribution to the lowest level.
   d. Project the transformed PV field onto the vertical modes (Eqs. 2(a, b)).
   e. Perform the spectral inversion (Eq. 3).
   f. Project the transformed streamfunction field on the vertical levels (Eq. 4).
   g. Get the transformed velocity field from the transformed streamfunction field at each level (Eq. 5).
   h. Inverse FFT to get the velocity field on the grid.
3. Interpolation
   a. Bi-linear interpolation to obtain the velocity field at each node on each contour.

3. ACCURACY TESTS AND TIMING

First in this section, the CASL algorithm is compared with a doubly-periodic two-dimensional CS algorithm. In particular, the sensitivity to the horizontal grid resolution n_h and the averaging factor m_g is examined visually and quantified. Next, the sensitivity to vertical resolution is addressed for a three-dimensional QG flow simulation by varying the number of layers n_v and the number of modes M independently within the CASL algorithm. Finally, the performance of various parts of the algorithm on a Fujistu VPX-240 vector processor is examined for a typical application.

TABLE 2. BAROTROPIC CASL AND CS SIMULATIONS

	m_g			
$\bar{n}_h$	1	2	4	8
120	34.6	35.0	49.4	96.4
240	93.3	105.2	153.4	340.6
480	351.3	403.8	576.9	
960	1399.6	1574.6		
1920	5770.2			
CS-lin.	926.6			
CS-cub.	1231.4			

Shown is total CPU time (in seconds) taken on a single processor of a Cray J90 supercomputer. All results were obtained using the CASL algorithm except for the last two; those were obtained using the CS algorithm with linear and cubic-spline interpolation in the computation of the velocity field.

(*a*) *CASL-CS comparison; the effect of horizontal resolution*

A version of the CS algorithm for a single-layer (two-dimensional) flow in doubly-periodic geometry permits one to test the essentially new aspect of the CASL algorithm, namely the use of a coarse horizontal grid. The vertical discretization in CASL is not new—it is used in CS and in conventional numerical algorithms—but its effect too is re-examined below.

The CS algorithm performs contour integrations over the Green function, which, for doubly-periodic geometry, is a complicated function (see Legras and Dritschel 1993). The Green function is split into a singular part (just the logarithm of distance) and a non-singular remainder, which is calculated by table look-up (using 1000 divisions in x and in y) to 7-decimal accuracy (the CS results below are unchanged, to within plotted line widths, when 2000 divisions are used).

The initial condition used in all the simulations performed consists of four vortices, two positive ($q' \equiv q - f = 4\pi$) and two negative ($q' = -4\pi$), of equal radius $R = 0.792665\ldots$ (chosen so that 5% of the domain is occupied by the vortices initially). The positive ones are centred at $\mathbf{X} = (0.24091, -0.05168)$ and $(1.71815, -3.05818)$, and the negative ones are centred at $\mathbf{X} = (-0.77690, -2.25197)$ and $(-2.35313, 0.07133)$. All the calculations are run with $\Delta t = 0.025$, $\mu = 0.08$ and $L = 2R$, so that there are 63 nodes per contour initially and a surgical scale of $\delta = 0.0025365\ldots \approx 2\pi/2500$.

The CS evolution is depicted in Fig. 2; 0.5 units of time separate each frame, advancing to the right and downwards, with positive vorticity rendered a darker shade of grey than negative vorticity. The vortices merge with their partners, the loosely combined positive vortex then splits momentarily, and all the interactions generate, characteristically, much fine-scale structure (see Dritschel 1995, and refs.).

The CASL algorithm was run with a variety of inversion grid resolutions $\bar{n}_h$, from 120 to 1920, and inversion/conversion grid ratios m_g, from 1 to 8. Table 2 lists the calculations performed and the CPU time required on a single processor of a Cray J90 vector processor.

All of the CASL simulations are barely distinguishable from the CS evolution in Fig. 2. This is illustrated in line-rendered comparisons in Figs. 3(a)–(d). In Fig. 3(a), the CASL simulation with the highest $\bar{n}_h$ (bold contours) is compared with two CS simulations, one using linear segments between nodes (as in the CASL algorithm) for computing

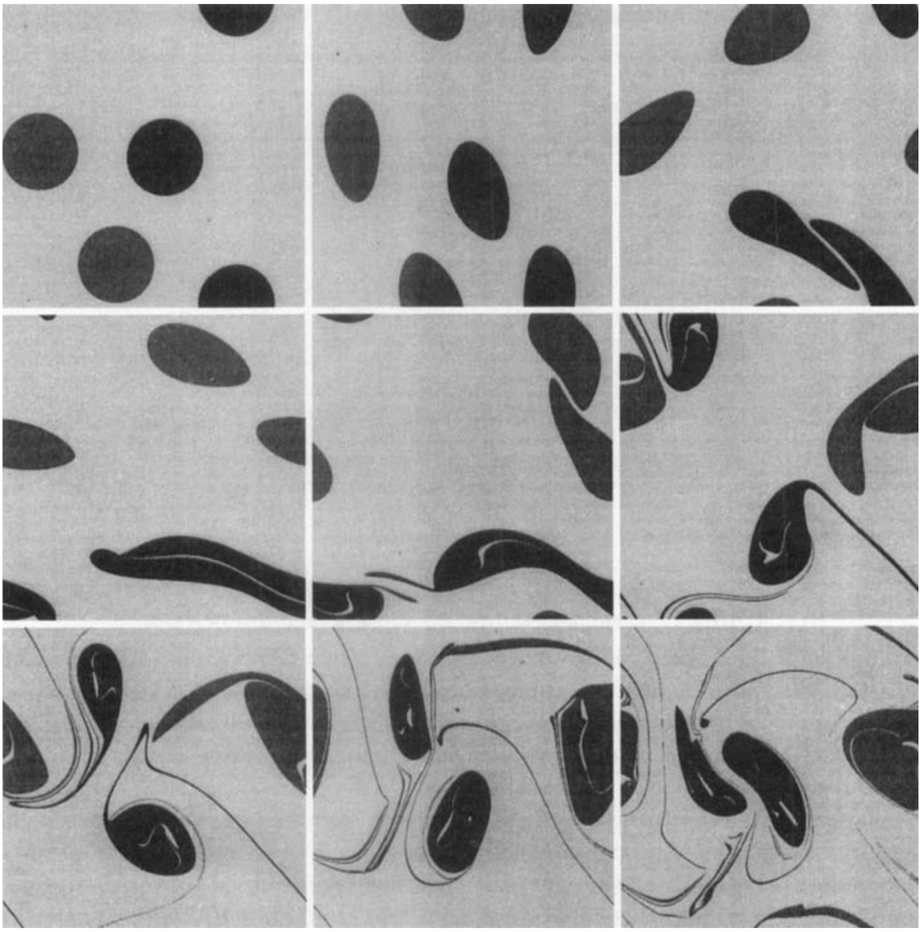

Figure 2. The evolution of four vortices in a doubly-periodic domain as simulated by contour surgery. Time advances to the right and downwards in increments of 0.5. Positive vorticity is rendered dark grey, and negative vorticity is rendered lighter grey.

the velocity field (thin contours, but indistinguisable from the bold ones), and the other using cubic splines as normally done in CS (dashed contours). The use of cubic splines causes visible differences, principally because the splined contours generally have slightly greater circulation. These differences may be reduced by using additional points in the PV contour-to-grid conversion or by adjusting the contour nodes to eliminate local circulation differences. This is work in progress.

In Fig. 3(b), $m_{\mathrm{g}} = 2$ for all simulations while $\bar{n}_{\mathrm{h}}$ is varied from 960 to 120 (bold, thin, dashed and dotted contours). The two highest-resolution cases are indistinguishable, and the errors appear to scale with the grid-scale squared, as would be expected from the level of accuracy to which the circulation is calculated. Contrast this now with Fig. 3(c) in which $\bar{n}_{\mathrm{h}} = 240$ while $m_{\mathrm{g}} = 8$, 4, 2 and 1 (bold, thin, dashed and dotted contours). The convergence is much more rapid, though the expected level of error (proportional to the PV conversion grid-scale squared) is the same. Finally, in Fig. 3(d), $n_{\mathrm{h}} = m_{\mathrm{g}}\bar{n}_{\mathrm{h}} = 960$ while both m_{g} and $\bar{n}_{\mathrm{h}}$ are varied, with m_{g} increasing and $\bar{n}_{\mathrm{h}}$ decreasing from bold to dotted contours. Again, the convergence is much more rapid than displayed in Fig. 3(b). Evidently, a fine PV conversion grid is the most important factor in getting an accurate result. A fine inversion grid is much less important.

This is quantified next in Figs. 4(a)–(c). Figure 4(a) shows the total area occupied by the vortices divided by the domain area as a function of time for the 16 simulations conducted, with distinct symbols used to differentiate $\bar{n}_{\mathrm{h}}$ (or the CS simulations) and distinct line styles used to differentiate m_{g} (consult figure). Area is not conserved even in CS due to surgery, node redistribution and time-stepping; in the CASL algorithm, moreover, the use of a finite grid and bi-linear interpolation give additional sources of error. Nonetheless, one can see that the results are clustered together apart from the coarsest-resolution CASL simulations, notably the ones with $m_{\mathrm{g}} = 1$ (note that doubling m_{g} gives better results than doubling $\bar{n}_{\mathrm{h}}$, as can be seen in the upper three curves, for instance).

Figure 4(b) shows the percentage difference in the PV fields, computed by comparing the PV values on a 1920 by 1920 grid, between the CASL simulation with $\bar{n}_{\mathrm{h}} = 1920$ ($m_{\mathrm{g}} = 1$) and all others (same line and symbol styles as above; consult figure). This shows even more clearly than the previous figure the great improvement brought about by the PV averaging.

A more dynamically-relevant measure of the differences is shown in Fig. 4(c), which displays the normalized velocity-spectrum difference between the CASL simulation with $\bar{n}_{\mathrm{h}} = 1920$ ($m_{\mathrm{g}} = 1$) and all others (as in Fig. 4(b)), $\sum_k (v_k' - v_k)^2 / \sum_k v_k^2$, where $\mathscr{E}_k = v_k^2$ and $\mathscr{E}_k' = v_k'^2$ are the energy spectra for the highest-resolution CASL simulation and any other ($\sum_k \mathscr{E}_k = E$, the total energy). In this measure, which accounts for the unimportance of fine-scale PV in inversion, the differences between the simulations are practically negligible. As in the previous figures, PV averaging is more effective in reducing the differences than halving the inversion grid length.

(*b*) *The effect of vertical resolution*

We next compare three CASL simulations differing only in the vertical discretization. Initially, the flow consists of a single slanting column of uniform anomalous PV ($q' \equiv q - f = 4\pi$) of unit radius and of slope $\mathrm{d}X/\mathrm{d}z = N/f$ (which appears as a unit slope in the plots) in a domain of depth $\pi N/f$. Each simulation uses $n_{\mathrm{h}} = 256$, $m_{\mathrm{g}} = 1$, $\mu = 0.1$, $L = 1$ (implying 80 nodes per contour initially and a surgical-scale of $\delta = 0.0025 \approx 2\pi/2500 \approx \Delta/10$), and $\Delta t = 0.025$. In the first simulation, depicted in Fig. 5, 60 levels and vertical modes are used ($n_{\mathrm{v}} = M = 60$). This is the 'high-resolution' case. The second simulation also uses 60 levels but only 20 modes, while the third uses 20 levels and 20 modes. A factor of three in levels is necessary to compare the flow fields at the same heights (the middle of the first, second, third . . . layer of a 20-layer simulation is at the same height as the middle of the second, fifth, eighth . . . layer of a 60-layer simulation).

First of all, let us briefly examine the flow evolution in Fig. 5. The column readily develops a perpendicular connection to the lower and upper surfaces while deforming horizontally and strongly shearing vertically at middle levels. The vortex just avoids being broken vertically (it does break for a vortex half as wide) and returns nearly to its initial shape. The sequence essentially repeats itself thereafter (at least up to $t = 31$, the final time of the simulation).

The second simulation, for 60 levels but only 20 modes, cannot be distinguished from that shown in Fig. 5. Again, one has to superpose contours to see differences. This superposition is done, for all three simulations, in Fig. 6(a) for the joint level closest to the bottom of the domain and in Fig. 6(b) for the joint level closest to the middle. The two 60-level simulations hardly differ up to $t = 8$ (fluid particles have then revolved eight times within the vortex core), and even well beyond this time; the 20-level simulation does

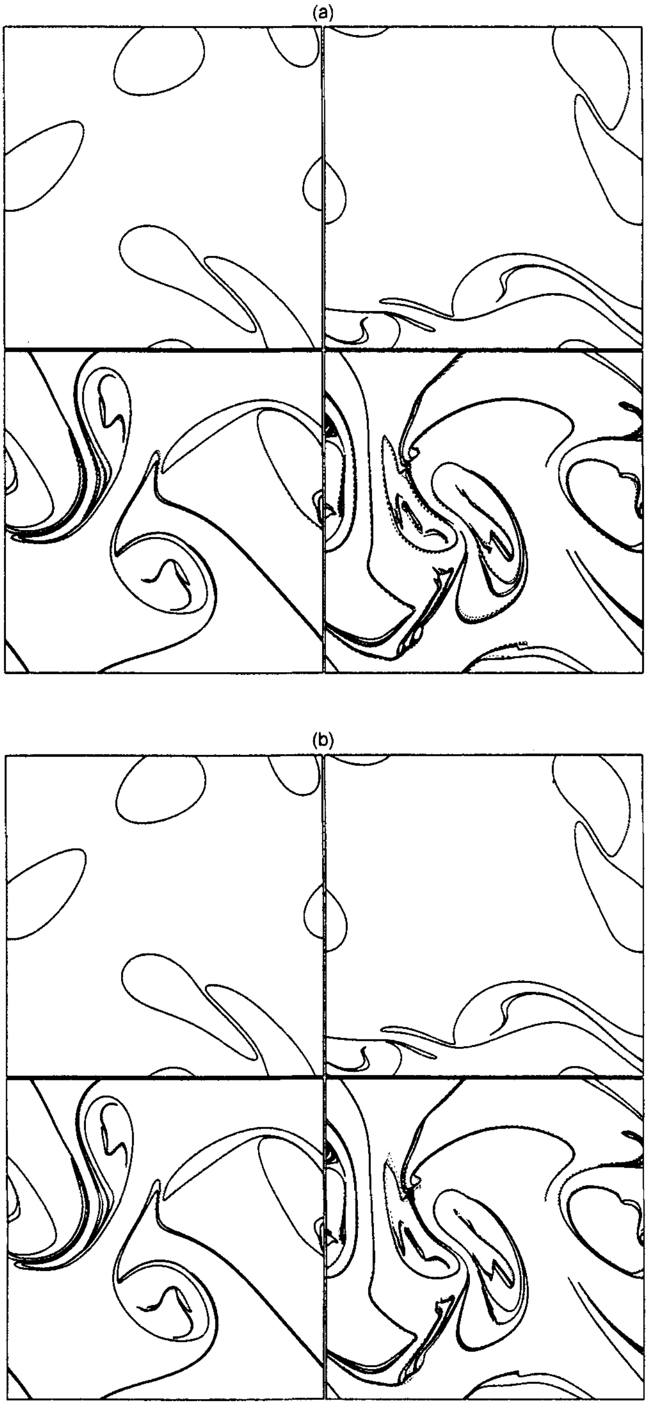

Figure 3. Vorticity contour comparison between various barotropic simulations at times $t = 1, 2, 3$ and 4: (a) CS with linear interpolation (bold contours), CASL with $\bar{n}_h = 1920$ (thin contours), and CS with cubic interpolation (dashed contours); (b) CASL simulations all having $m_g = 2$ but $\bar{n}_h = 960$ (bold), 480 (thin), 240 (dashed) and 120 (dotted); (c) CASL simulations all having $\bar{n}_h = 240$ but $m_g = 8$ (bold), 4 (thin), 2 (dashed) and 1 (dotted); (d) CASL simulations all having $n_h = m_g \bar{n}_h = 960$ but $m_g = 8$ (bold), 4 (thin), 2 (dashed) and 1 (dotted).

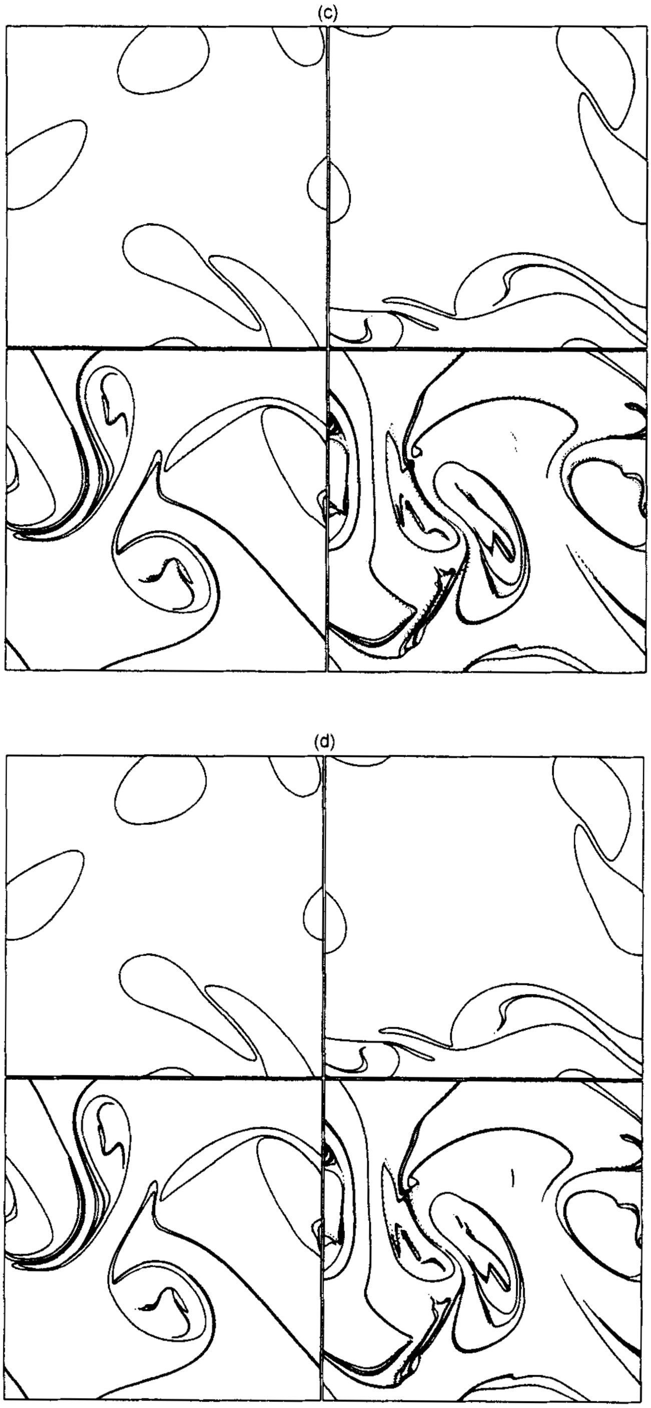

Figure 3. Continued.

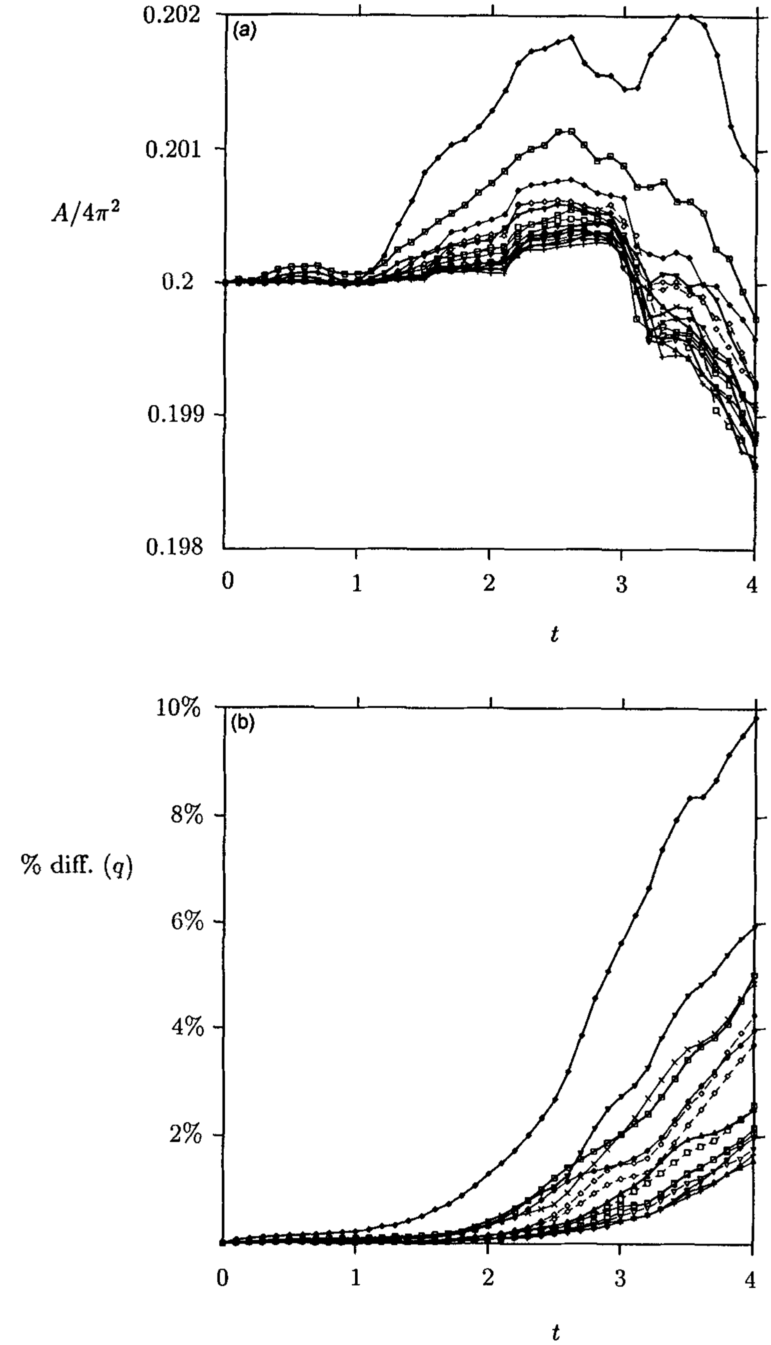

Figure 4. Diagnostic differences between the various barotropic simulations as a function of time: (a) The total area of non-zero vorticity divided by the domain area for the 16 simulations performed, using line styles to differentiate m_g (1 is bold, 2 is thin, 4 is dashed and 8 is dotted) and symbols to differentiate $\bar{n}_h$ (120 is marked with diamonds, 240 with squares, 480 with inverted triangles, 960 with triangles and 1920 with ×s). The CS simulations are marked with +s, with bold lines for linear interpolation and thin lines for cubic interpolation. (b) The percentage difference in the PV fields (difference from the CASL simulation with $\bar{n}_h = 1920$). Line styles and symbols are the same as in the previous figure, except now the ×s mark the CS calculation with cubic interpolation (+s mark the one with linear interpolation) and thin lines are used for both. (c) The velocity spectrum difference. Line styles and symbols are the same as in (b).

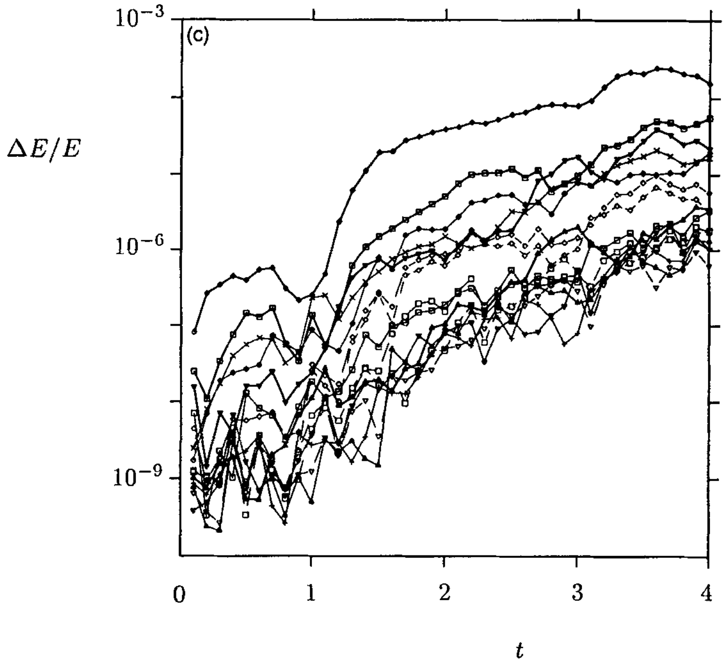

Figure 4. Continued.

differ more, but not excessively so. Figure 7 gives a quantitative measure of the agreement, namely the r.m.s. centroid difference, $\{n_v^{-1}\sum_{l=1}^{n_v}|\mathbf{X}_l' - \mathbf{X}_l|^2\}^{1/2}$ between the two 60-level simulations (bold line) and between the 60-level, 60-mode simulation and the 20-level, 20-mode simulation (thin line) versus time.

The excellent 60- and 20-mode comparison proves an important point: fine vertical structure does not contribute significantly to the advecting velocity field—this is of course the principle which underpins the CASL algorithm. The comparison with 20 levels shows that many important characteristics can be captured using moderately-coarse vertical resolution. Furthermore, there is no indication that moderately-coarse resolution leads to spurious behaviour. Indeed, following Legras and Dritschel (1993), one can estimate that the velocity difference between that corresponding to a vertically-continuous PV distribution and that corresponding to a layered PV distribution scales with $1/n_v^2$. This follows from the nature of the inversion operator.

(*c*) *Timing*

The speed of the various parts of the CASL algorithm is examined next. A flow of significant complexity was chosen for illustration, namely the evolution of the polar stratospheric vortex. A conservative QG model had been found to give meaningful results in a previous work (Dritschel and Saravanan 1994), and it is again used here. The simulation was conducted on a Fujitsu VPX-240 vector processor, whose peak theoretical speed is 2.6 Gflops.

The initial condition is a slanted column of non-uniform PV,

$$q(x, y, z', 0) - f_0 = 4\pi\left(1 - \frac{(x - X(z'))^2 + y^2}{R^2}\right)$$

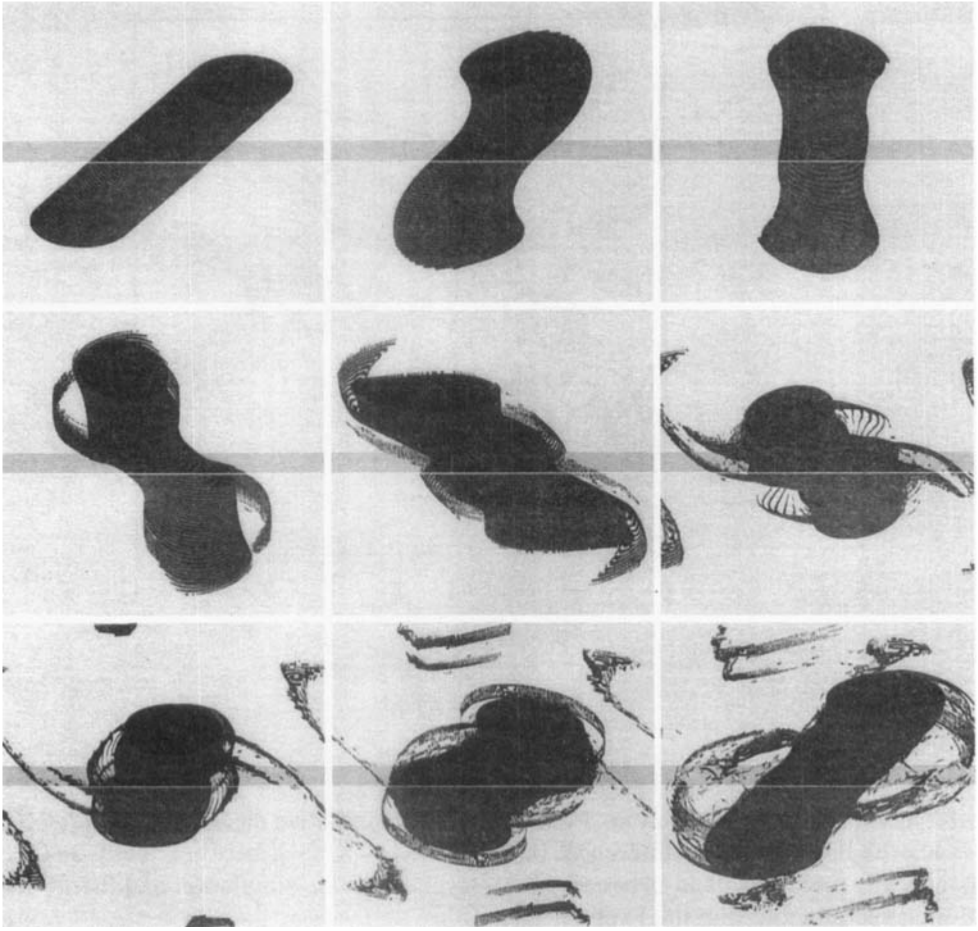

Figure 5. The evolution of an initially slanting column of uniform PV in a box of scaled-height π as simulated using the CASL algorithm, with $n_v = M = 60$, $\bar{n}_h = 256$ and $m_g = 1$. The flow is viewed head on from a 60° angle from the z axis. Positive PV is dark and (in general) negative PV (which is here absent) is light, using a lighter shade for the sides of the structures. Structures seen through the front face are further lightened (a horizontal white line shows the top part of the front face; a second white line shows the bottom of the back face, where it is not obscured). The visible bottom of the domain is darkened relative to the sides. Time advances to the right and downwards in increments of 1 unit. The evolution continues onto the next page.

where $z' = Nz/f_0$, $X(z') = 0.1L_R(z' - z'_{\text{mid}})$, $L_R = 1/3$ is the Rossby radius of deformation, $z'_{\text{mid}} = NH/2f_0$, $H = 10H_\rho$ (the fluid depth is ten density-scale heights), and $R = 3L_\text{R} = 1$ is the cross-sectional radius of the vortex.

A slanting polar vortex was chosen because it has been observed in real data (Waugh 1997). The initial slope is comparable to those observations. We have chosen a particularly deep fluid, ten scale heights, to demonstrate the known instability of the upper part of the vortex to vertically propagating waves and the long-term consequences (this is in fact related to a true instability of tall vortex columns, as demonstrated by Dritschel and de la Torre Juárez 1996).

A simulation was performed for a 50-layer QG fluid, using 50 modes as well, and a horizontal grid resolution of $\bar{n}_h = 240$ with $m_g = 2$. (A high vertical resolution was chosen to resolve the small-scale vertical structures that develop in this example.) The

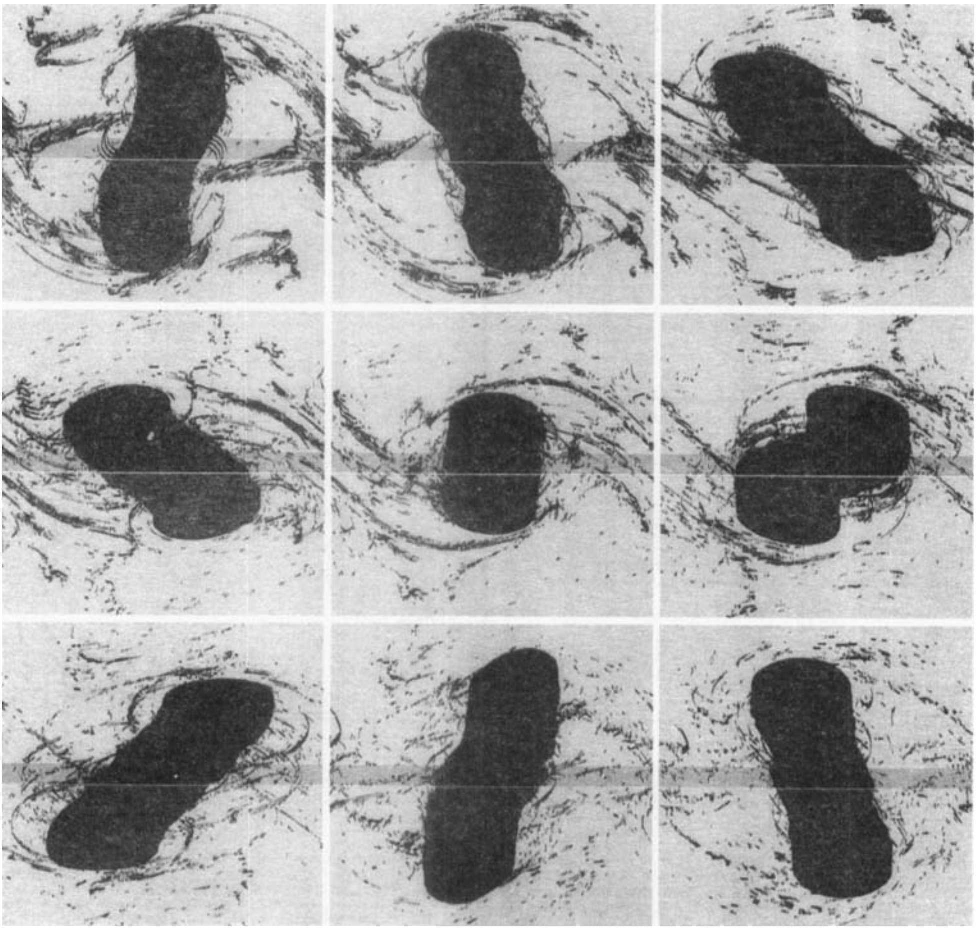

Figure 5. Continued.

other numerical parameters used were $L = 1(= R)$ and $\mu = 0.12$, giving a surgical-scale $\delta = 0.0036 \approx \bar{\Delta}/7.3$), and a time-step $\Delta t = 0.025$. The initially parabolic profile of PV is discretized initially into 20 equal steps, except for the outermost, which has half the step size (this is the best least-squares approximation to the continuous profile, see Legras and Dritschel 1993).

A few snapshots of the flow evolution are shown in Fig. 8 (time, in increments of eight units, proceeds left to right and then downwards). The highest PV values are darkest, and the sides of the layers are rendered an intermediate shade to distinguish them from the interior PV and the surrounding fluid. Initially, a wave (or a series of waves) travels up the vortex and amplifies due to the exponentially decreasing density with height. This amplification results in the upper part of the vortex being torn away; this occurs at approximately eight scale heights, a little higher than the normally observed height of approximately six scale heights (maintained bottom forcing, as in Dritschel and Saravanan 1994, would likely have a stronger influence and clip the top of the vortex lower down; this has been verified for a column of uniform PV).

Next, we turn to the computational cost. The entire simulation required 5 hours of CPU time. Initially, the contours (1000 in total) were represented by about $n = 50000$ nodes. This

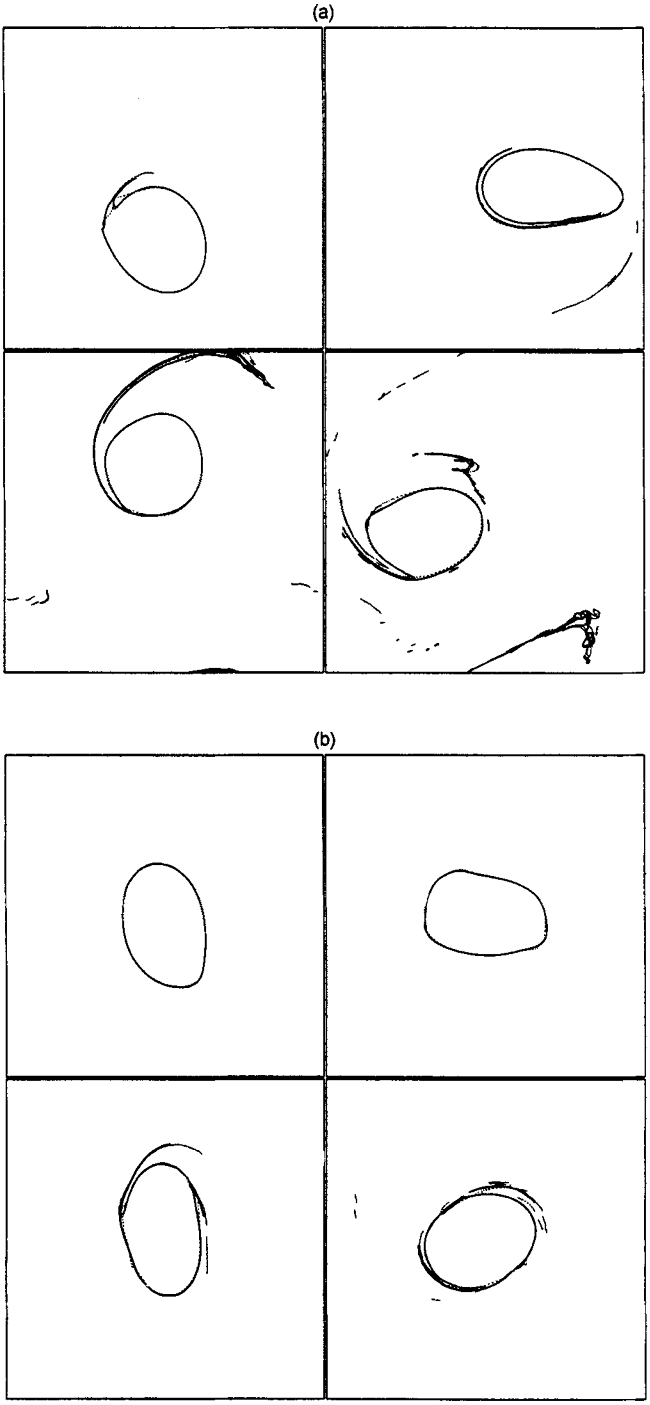

Figure 6. Vertical resolution CASL inter-comparison (a) at the joint layer nearest the bottom and (b) at the joint layer nearest the middle. Bold contours show the simulation with $n_v = M = 60$, thin solid contours the one with $n_v = 60$ and $M = 20$, and dashed contours the one with $n_v = M = 20$. Times 2, 4, 6 and 8 are compared.

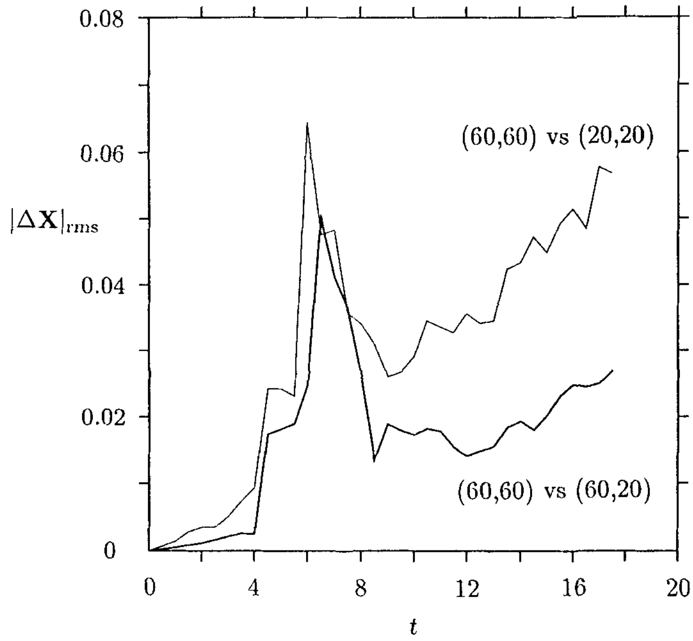

Figure 7. Time evolution of the root mean square centroid difference between the 60-level simulations (bold line) and between the 60- and 20-level simulations (thin line) versus time.

climbs to and levels off at a value of 300000 nodes by the end of the simulation. Needless to say, the flow is then much more complex than Fig. 8 makes it appear. Computing with this level of complexity is a strong test of the algorithm's efficiency.

Figure 9 shows the rate of execution, the average CPU time taken per time-step, versus time t for the various parts of the algorithm: the FFTs (triangles), the inversion (inverted triangles) excluding the FFT and PV contour-to-grid conversion (PVCGC) costs but including PV averaging, the PVCGC (+s), surgery (×s), and all other costs (diamonds) including advection, node redistribution, and the bi-linear interpolation. Also shown is n versus t (bold; see scale on the right). Note that the cost of surgery rises steeply with increasing n, briefly surpassing the total inversion costs (now including the FFTs) and then decays again despite the continued growth in n. The decay is due to the elimination of many contours and the reduced search costs as the flow settles down. Normally a flow does not sustain this level of complexity for very long; a longer simulation would show n itself decreasing a little later in time. The cost of the PVCGC also becomes comparable to the inversion costs for large n (recall that the conversion is being done on a grid twice as fine (480 × 480) as the inversion grid). Part of the PVCGC costs includes grid operations (e.g. sweeping to get the PV from PV jump values), and so there is an overhead proportional to $n_{\rm h}^2$. The remainder of the cost is proportional to n; in fact, a very good fit of the PVCGC cost is obtained by $4.0 \times 10^{-8} n_{\rm v} n_{\rm h}^2 + 1.0 \times 10^{-5} n$. The larger prefactor for n is due to the partial lack of vectorization of the PVCGC.

The inversion, surgery and PVCGC costs are not independent, but depend on how the various parameters are chosen. We recommend, after much experience running the CASL algorithm, the following parameter relationships. Having chosen the horizontal inversion grid resolution $\bar{n}_{\rm h}$ and a characteristic scale for the PV distribution L, choose

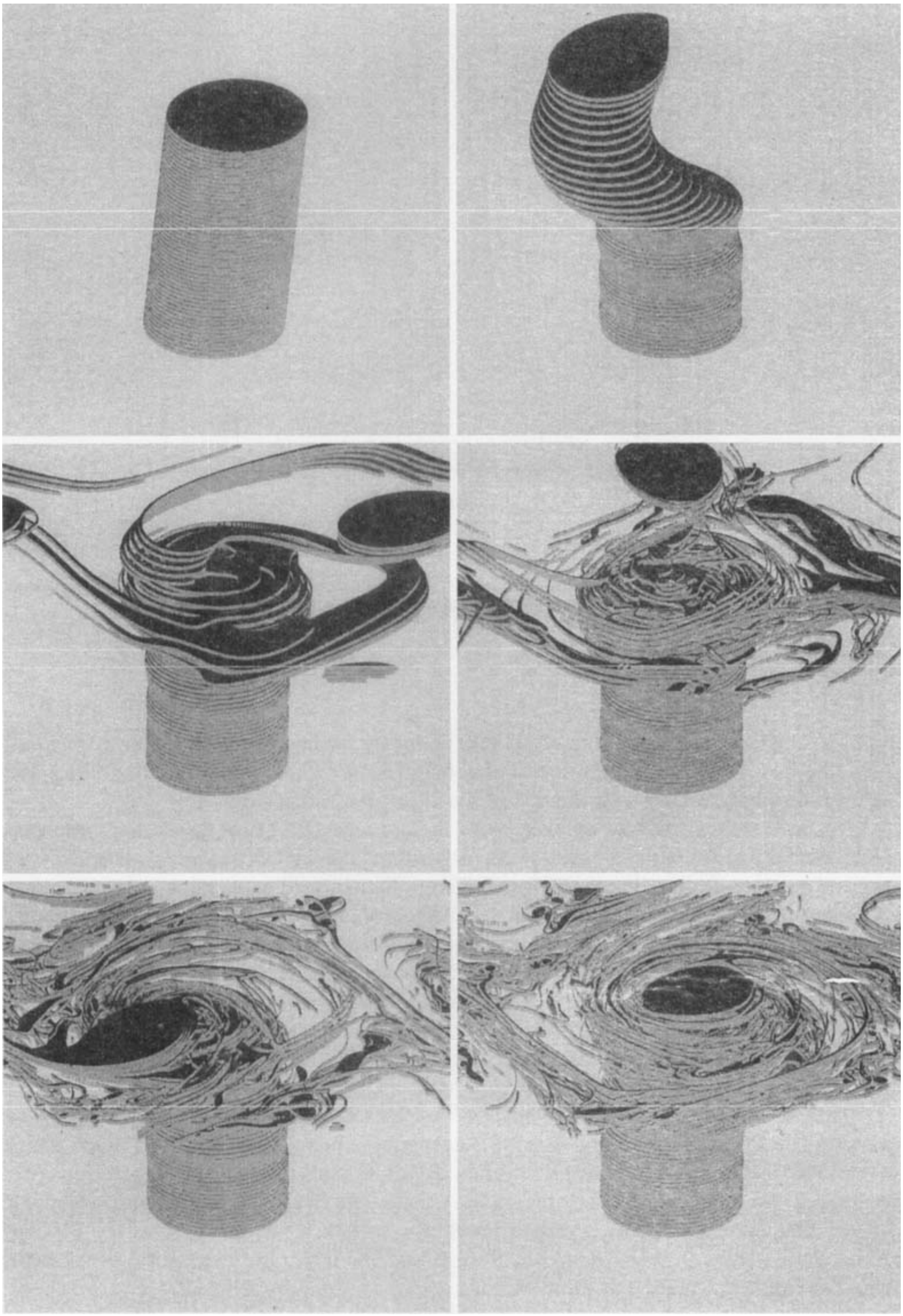

Figure 8. Evolution of an initially slanting stratospheric polar vortex; times shown are $t = 0$, 8, 16, 24, 32 and 38.6 (the tinal time). The view angle is 60° from the x axis, and the rendering is the same as in Fig. 5.

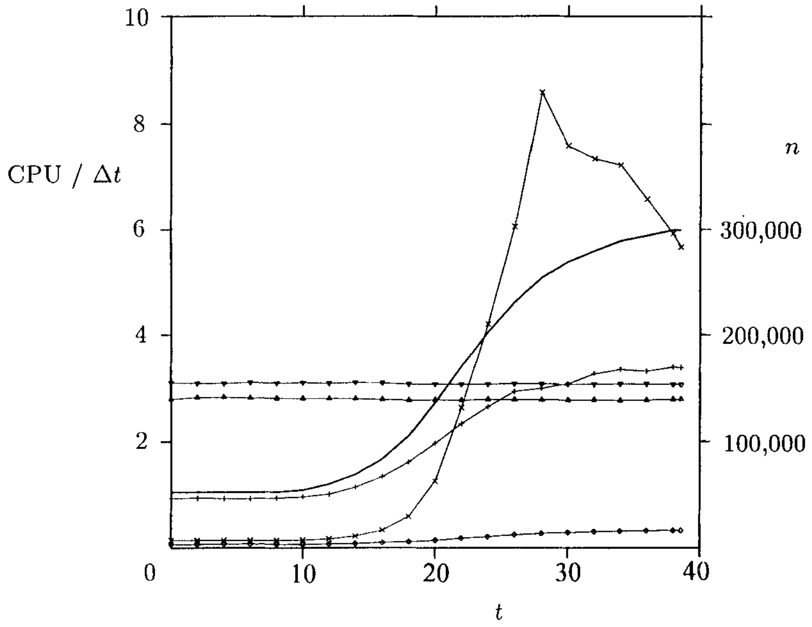

Figure 9. Rate of execution, defined as the average CPU time in seconds per time-step, of the component parts of the CASL algorithm, the FFTs (triangles), the inversion (inverted triangles) excluding the FFT and PVCGC costs but including PV averaging, the PVCGC (+s), surgery (×s), and all other costs (diamonds). Also shown (in bold) is n versus t.

the surgical-scale δ to be an eighth of the inversion grid-scale, i.e. $\delta = \bar{\Delta}/8$. Since $\delta = \mu^2 L/4$, this determines the dimensionless maximum node separation μ. Next, choose the PV inversion/conversion grid ratio m_g to be 2 or 4, depending on the accuracy required (consult section 3(a)). Choose the number of layers n_v and modes M to be 20 or greater to properly resolve a continuously stratified flow. The relative costs of the inversion, surgery and PVCGC depend on the behaviour of the total number of nodes n, which, for the relationships just outlined, depends on $\bar{n}_h$ and n_v. Normally, one can expect $n \propto n_v$. One can also expect n to be inversely related to μ and L. From experience, most nodes lie on filaments, with a node spacing of about μL (this is particularly true during the most agitated periods of evolution). The maximum total contour length is bounded by the domain area divided by the surgical-scale δ. Dividing the point spacing into this estimate for the contour length gives an estimate of the maximum number of nodes $n_{\max} < c n_v A_{\text{dom}}/\mu^3 L^2$, where A_{dom} is the domain area, or $4\pi^2$ here, and c is a dimensionless constant, which from experience, lies in the range 1/20 to 1/2. From this relationship, and $\delta = \bar{\Delta}/8$, we can see that $n_{\max} \propto n_v \bar{n}_h^{3/2}$. The PVCGC cost is thus at most comparable to the inversion and FFT costs, proportional to $n_v \bar{n}_h^2$ and $n_v \bar{n}_h^2 \log \bar{n}_h$, respectively. The surgery cost is the greatest unknown, since it can only crudely be estimated as $O(n^2)$ and thus $O(\bar{n}_h^3)$. Fortunately, though, surgery is expensive only during short periods of extreme complexity.

The overall speed of execution is greatly enhanced in CASL by using an advective time-step: fewer inversions are necessary in CASL to simulate for a fixed time interval than in conventional numerical models. A standard, pseudo-spectral simulation of the above flow would have required a time-step of 0.00125 to ensure numerical stability (Yao *et al.* 1995), a figure which is one-twentieth of that used in the CASL simulation. Moreover the effective resolution of the CASL simulation is ten times greater than the pseudo-spectral one.

4. Quasi-geostrophic turbulence

The long-time behaviour of 'freely-decaying' (or perhaps should it be 'freely evolving'?) QG turbulence (QGT) has attracted considerable interest (Rhines 1979; Hua and Haidvogel 1986; McWilliams 1989, 1990; McWilliams *et al.* 1994; Dritschel and de la Torre Juárez 1996). In part, this is because the study of QGT may indicate fundamental vortex dynamical processes which may be found in realistic, atmospheric and oceanic flows, and may thus help us interpret what we see, or what we should look for, in such flows.

Like two-dimensional turbulence (2DT), QGT develops from random initial conditions into a set of widely-spaced coherent regions of PV—'vortices' for short—embedded within a disorganized field of stretching and twisting filaments. All studies to date have considered a Boussinesq fluid (ρ_0 constant) and, except for Hua and Haidvogel (1986), constant buoyancy frequency N, constant Coriolis parameter f, and homogeneous boundary conditions (i.e. no surface temperature gradients or topography). In this case, the ratio N/f can be absorbed into the z coordinate in (1b), rendering the inversion operator isotropic. This led Charney (1971) to propose a scaling theory similar to that proposed by Kolmogorov (see Kraichnan 1974) for a 3D unstratified, non-rotating fluid. In particular, Charney argued that QGT should have an isotropic energy spectrum (in spite of the fact that the vertical velocity component is zero, a clear anisotropy in the governing equations; see Hua and Haidvogel 1986 for further remarks).

To date, simulations of QGT do not fully support Charney's theory, in particular the assumption of isotropy. Anisotropy is manifested by vortices of height-to-width aspect ratios significantly greater than f/N, and by vortices predominantly located near the top and bottom boundaries of the domain.

Recently, it has even been suggested, on the basis of a higher-resolution, longer-duration simulation of QGT (McWilliams *et al.* 1994), that QGT becomes progressively two-dimensional through the vertical alignment of PV structures (Polvani 1991; Viera 1995). In the simulation presented in McWilliams *et al.* (1994), two columns of opposite PV (q') are seen at the end of the simulation. This conclusion supports an alternative theory (Rhines 1979) that baroclinic components die out in QGT, leaving only the barotropic (z-independent) component, i.e. 2DT, a highly-anisotropic flow state.

The present motivation for re-examining QGT has come from a recent study (Dritschel and de la Torre Juárez 1996), which found that sufficiently tall PV columns in a 2D straining flow break down into surface-trapped vortices, remarkably similar to those seen in earlier pseudo-spectral simulations of QGT (McWilliams 1989). Is, then, the 2D theory robust? Might the results of McWilliams *et al.* (1994) be a consequence of the accumulation of *ad hoc* hyperdiffusive effects over the long duration of the simulation (as suggested in Dritschel and de la Torre Juárez 1996)? Does QGT retain a significant 3D character in the absence of down-gradient diffusion, as suggested by Charney (1971)?

To begin to answer these questions, a simulation of QGT was conducted using the new CASL algorithm. The simulation *started* from a 2D basic state superimposed with small 3D disturbances. Twenty equal-sized uniform PV columns, ten having $q' \equiv q - f = +Q$ and ten having $q' = -Q$, were randomly placed within the domain so that 5% of the volume had non-zero q' and no two vortices were closer than $6R_v$, R_v being the vortex radius. The height of the domain was chosen to be $H = \frac{\pi}{2} f/N$, giving a vortex full height-to-width aspect ratio of $4.431 f/N$.

The numerical parameters were set as follows: $\mu = 0.15$, $L = 2R$ (together implying a resolution of 44 nodes per contour initially), $\delta = 0.001994$, $n_v = M = 20$, $n_h = 480$, $m_g = 1$ (giving a grid length $\Delta = 0.01309$ or 6.565δ) and $\Delta t = T_v/40$, where $T_v = 4\pi/Q$ is the rotation period of an undisturbed circular column. For presentation, time is scaled on T_v, and z is scaled on f/N. The visualization is the same as for Fig. 5.

The flow remains approximately two-dimensional until $t = 9$, just after two like-signed columns have undergone a grazing collision. Figure 10 shows the evolution for a short period after this time. The two columns, seen at the front of the domain just left of centre, are the first to become strongly 3D. They first tilt and subsequently merge at the lower surface, as was found to occur for two columns in isolation (Dritschel and de la Torre Juárez 1996). A vortex of enlarged cross-section forms at the bottom of the domain while the remnants of the two columns helically intertwine above.

Further, more widespread 3D development occurs shortly thereafter. Figure 11 shows the time period from $t = 22$ to 29. In the front-right corner, two opposite-signed vortices lean over and split off a briefly-coupled baroclinic dipole at the upper surface, one may see this also in Dritschel and de la Torre Juárez (1996). In the back-right corner, two pairs of like-signed columns merge at the top and shed their helically intertwining lower parts, again as in Dritschel and de la Torre Juárez (1996). In the back-middle part of the domain, a column is split in two by the surrounding straining field. Overall, there is a breakdown of the original 2D flow.

As the evolution proceeds, the 2D character of the flow completely vanishes; after $t = 30$, no vortex extends from the top to the bottom of the domain, as illustrated in the evolution sequence shown in Fig. 12. The flow is then characterized by a set of vortex domes attached to the top and bottom boundaries, as predicted in Dritschel and de la Torre Juárez (1996). Not a single coherent PV structure exists in the centre of the domain. In time, some of the vortex domes widen by merging with other domes, but they do not deepen (this has recently been re-confirmed by a similar simulation run to $t = 200$). For deepening to occur, a surface vortex must trap and pool together previously disorganized PV floating around the domain interior. If this is not done fast enough, the disorganized PV, which is probably thinning at an exponential rate (like in 2DT, Dritschel 1993a), will mix too efficiently for trapping ever to occur. Trapping requires vertical alignment of PV (Polvani 1991; Viera 1995) (a process similar in essence to merging in 2D), and this may occur only between structures of comparable q'. A well-mixed patch will, however, appear to have a much smaller value of q' than a coherent vortex dome on account of the inversion operator's insensitivity to fine-scale PV, and hence the two structures may not align. Indeed, one may expect the well-mixed patch to behave more like a passive-tracer, than a self-organizing vortex.

This single example demonstrates that QGT is richer than had been expected. There remains much to be understood. In particular, the breakdown of two-dimensionality requires tall vortex columns, but even if the columns are initially shallow (of height to width ratio $\lesssim f/N$), interactions between them inevitably produce smaller (narrower) vortices (Dritschel 1993b, 1995), and these will be prone to breakdown (Dritschel and de la Torre Juárez 1996). The larger vortices produced presumably remain columnar. In a simulation conducted using half of the domain height used above, however, all the vortices were again observed to breakdown into baroclinic vortex domes attached to the surfaces. A further simulation was conducted using a quarter of the the original domain height, and still a great deal of baroclinic activity developed, though this time the larger vortices remain vertically coherent (though baroclinically disturbed) up to at least 66 vortex-rotation periods, see Fig. 13. Clearly, two-dimensionality is not as robust as was previously thought.

One may also ask what happens when some of the present idealizations are abandoned,

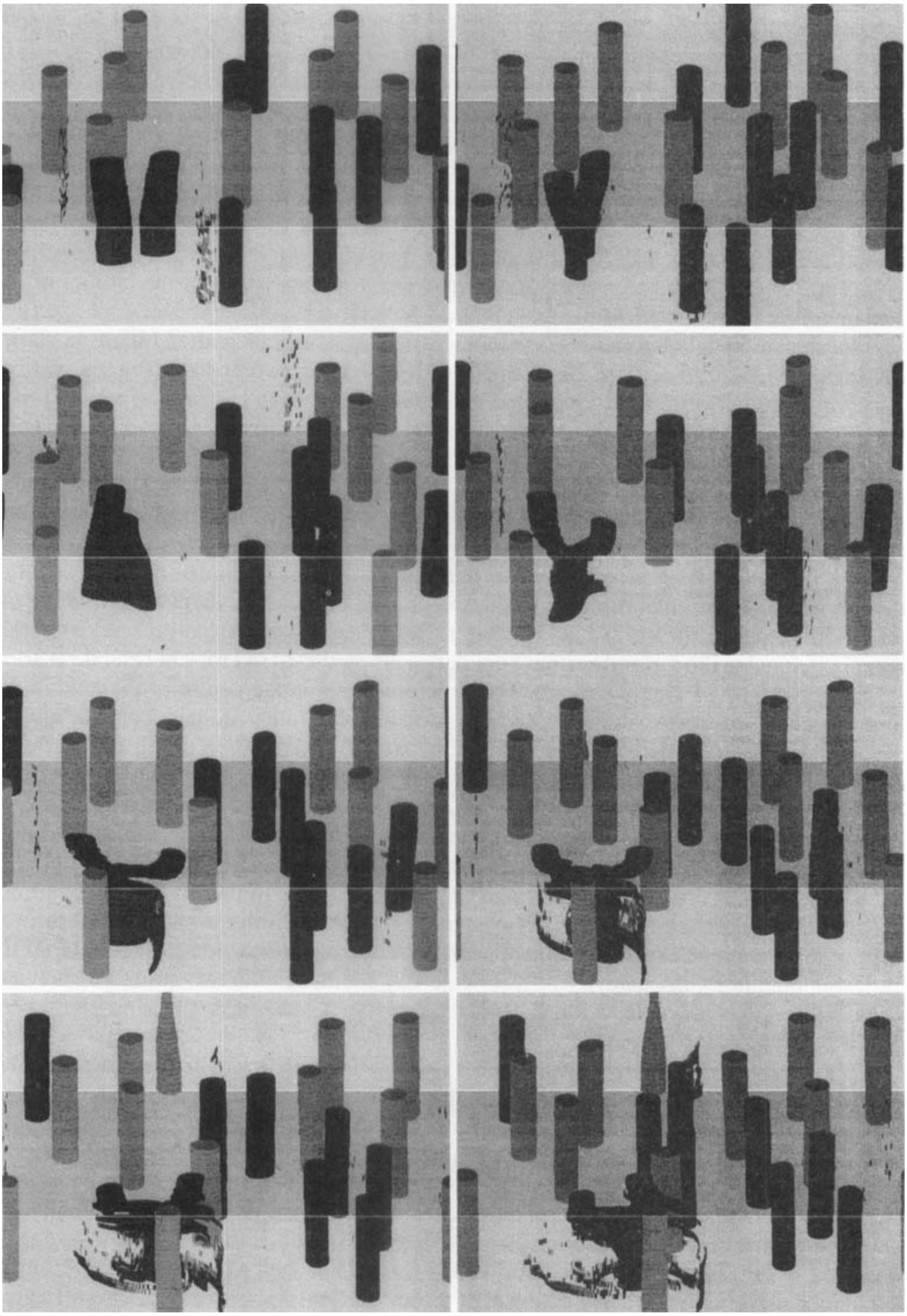

Figure 10. The PV from $t = 10$ to 17 (in unit intervals) in a 20-layer CASL simulation of QG turbulence in a domain of scaled-height $\pi/2$ (see the caption of Fig. 5 for the mode of visualization). The initial flow consists of columns of uniform PV upon which are superposed small 3D disturbances.

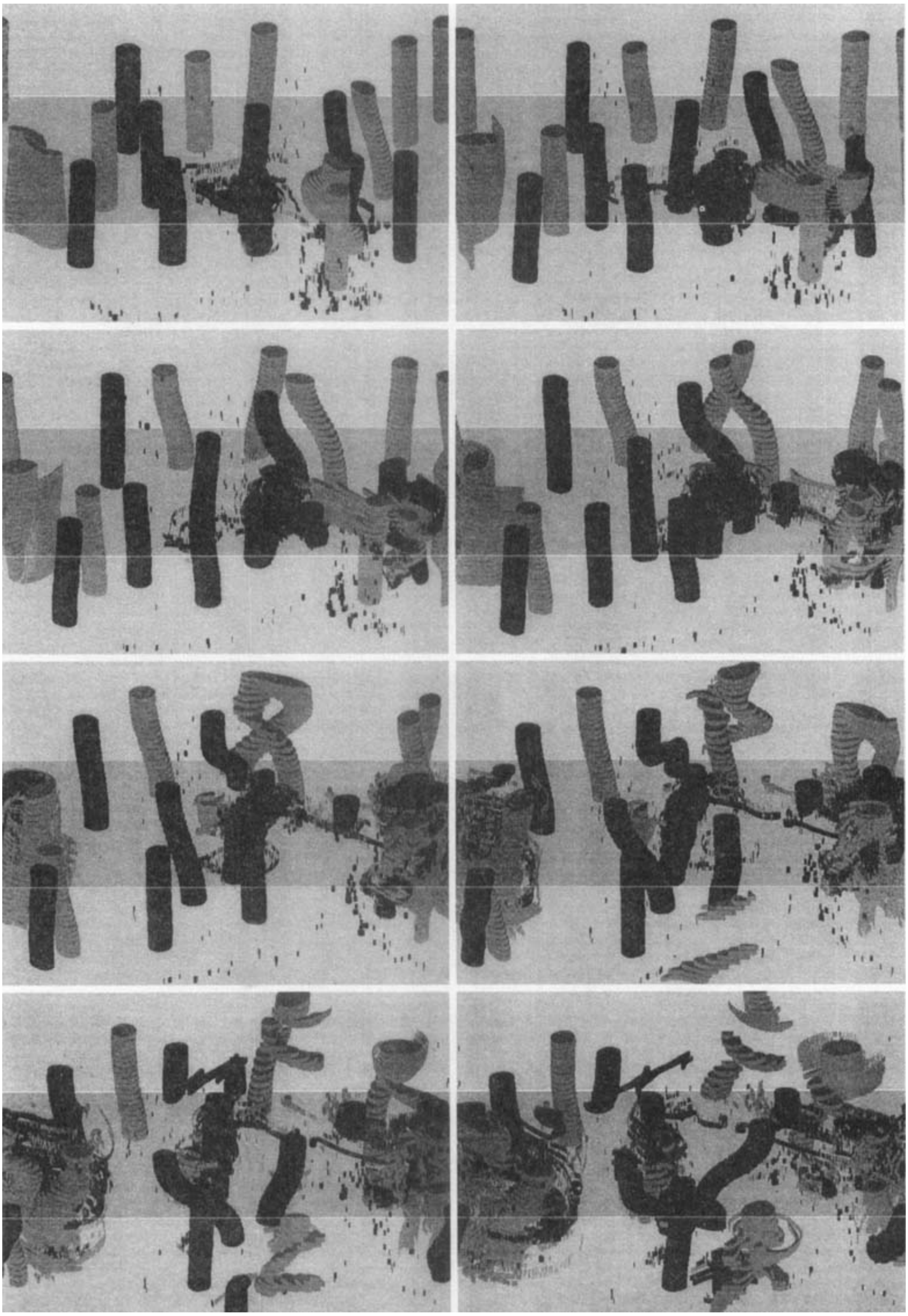

Figure 11. Same as Fig. 10 except $t = 22$ to 29 in the QG turbulence simulation.

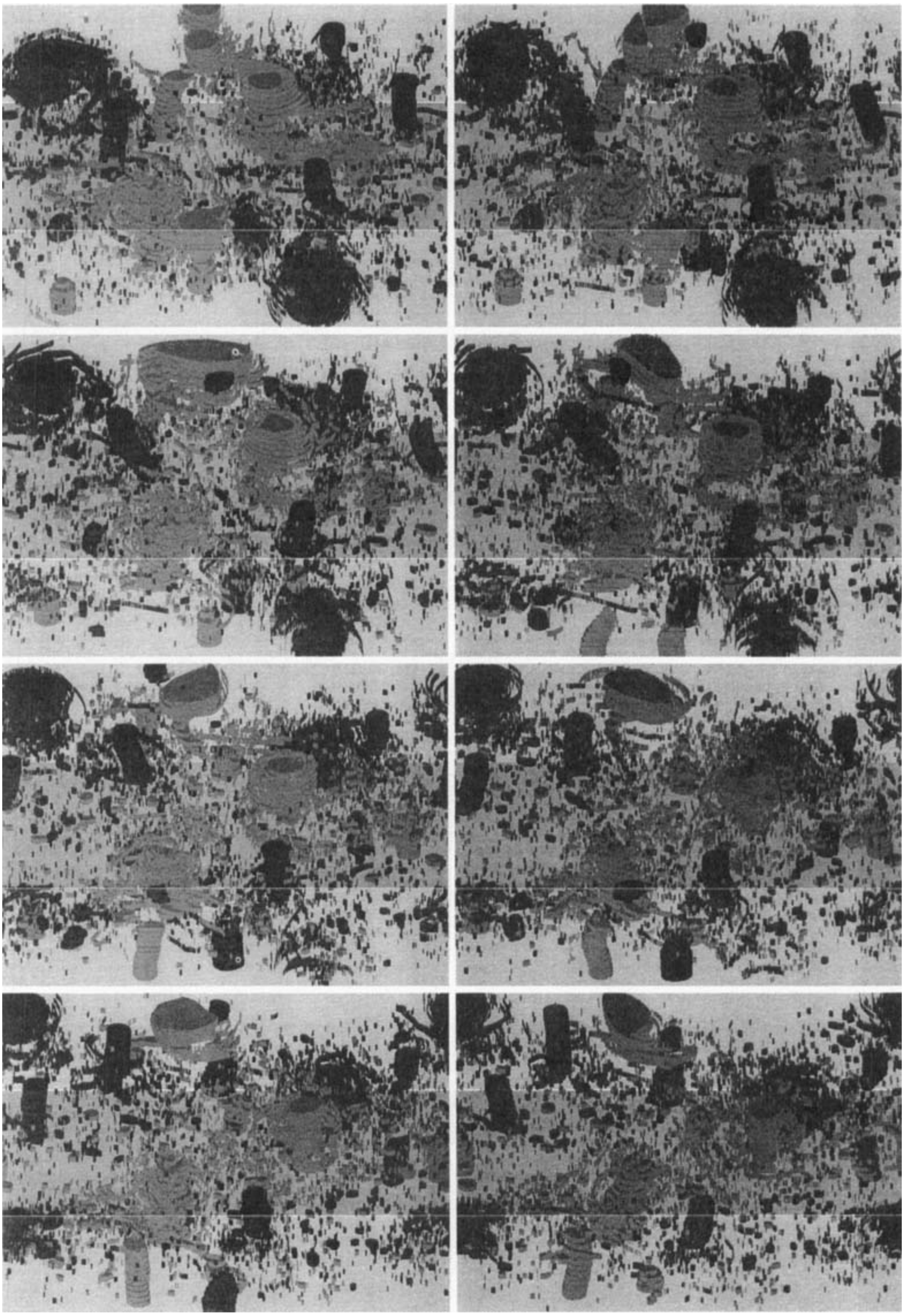

Figure 12. Same as Fig. 10 except $t = 47$ to 54 in the QG turbulence simulation.

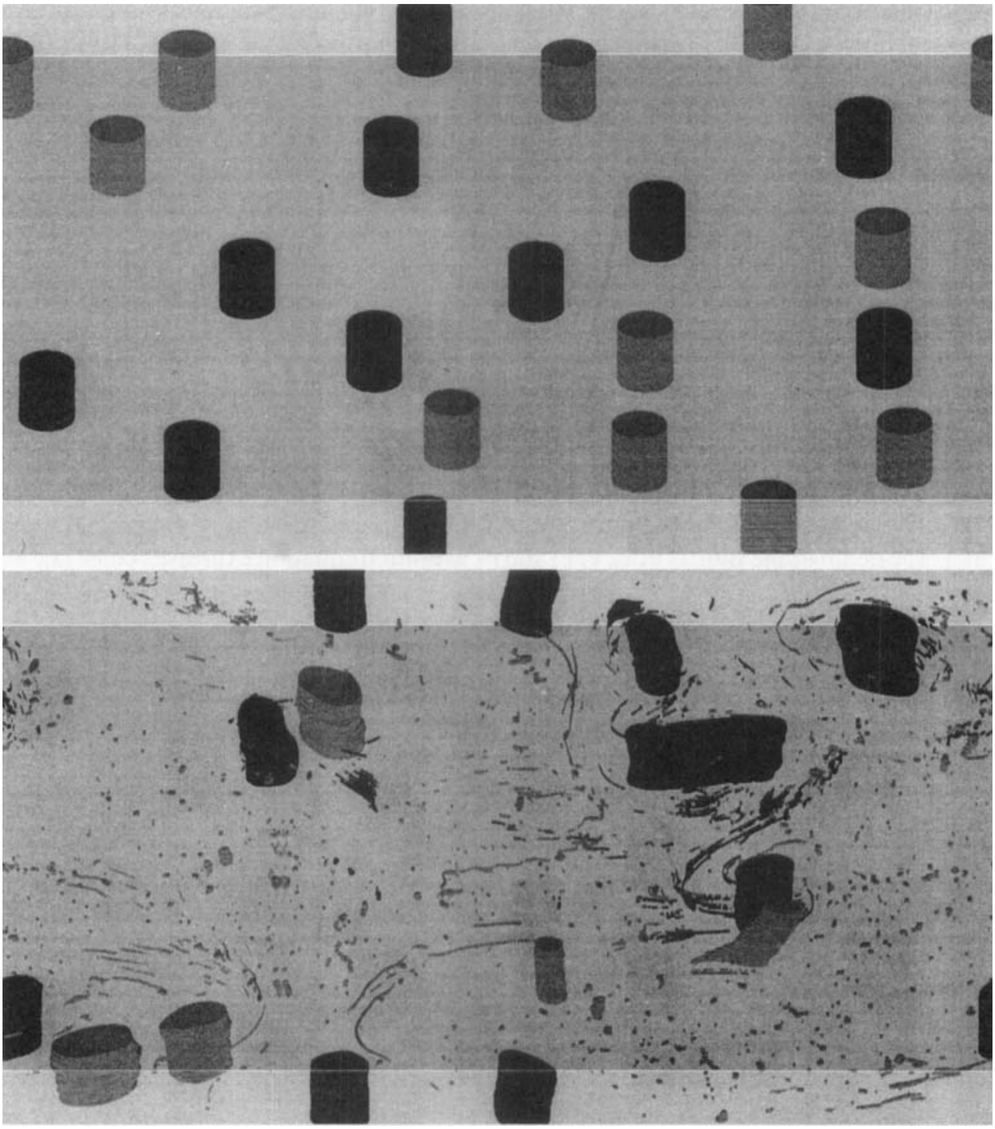

Figure 13. Initial and final times ($t = 66.6$) in a QG turbulence simulation conducted in a domain one quarter as deep as previously illustrated.

i.e. $\beta \neq 0$, bottom topography, surface temperature gradients (as a source for 'baroclinic instability', see Pierrehumbert and Swanson 1995), non-uniform N, an exponential density profile $\rho_0(z) \propto \exp(-z/H_\rho)$, etc. All of these effects can be studied efficiently, more comprehensively, and in greater detail than ever before, using the CASL algorithm.

5. Extensions to ageostrophic systems

The basic idea underpinning the CASL algorithm is to hold conservative (advected) fields in contour form to avoid excessive dissipation, as is common in sub-grid models. Such fields typically possess a shallow spectrum, decaying only gradually with diminishing scale. This decay is generally not rapid enough to maintain an accurate grid or spectral representation of the field, particularly when at least a third of the spectral tail must be sacrificed to some form of damping and possibly de-aliasing as well (Legras and Dritschel 1993; Macaskill and Bewick 1995). Even when such damping is not explicit (such as in the semi-Lagrangian scheme), implicit damping does occur and the small scales are not reliable (Gravel 1996). The advantage of the CASL algorithm is that scales are carried well below the grid-scale, providing in effect an explicit internal model for 'eddy viscosity', and thereby leaving grid-scale features relatively uncontaminated.

In more complex atmospheric and oceanic models, there is more than the PV to worry about (and there are also non-conservative effects, see section 6). In general, the velocity field must be computed dynamically—it cannot be expressed as a functional of the PV. But these other fields normally have a much steeper spectrum than the PV, so it is appropriate to hold them in a spectral or grid representation. Hence, one can use the normal model machinery for such fields and the CASL algorithm for the PV.

This idea is illustrated for the 'shallow-water' system, a frequently used model in atmosphere and ocean dynamics (see, e.g. Vallis 1996 and refs). The system describes the vertically averaged motion of a thin layer of fluid (thin compared to its horizontal variations). The depth of the fluid is $h(x, y, t)$ and its evolution equation expresses conservation of mass:

$$\frac{\partial h}{\partial t} + \nabla \cdot (h\mathbf{u}) = 0 \tag{6a}$$

where $\mathbf{u} = (u, v)$ is the vertically averaged horizontal velocity. Cartesian geometry is used here to simplify the presentation; spherical geometry is widely used in practice, as the system is perfectly valid in equatorial regions. The PV is given by

$$q = \frac{\zeta + f}{h}$$

where $\zeta = \partial v/\partial x - \partial u/\partial y$ is the relative vorticity and $f(y)$ is the Coriolis parameter. The PV satisfies

$$\frac{\mathrm{D}q}{\mathrm{D}t} = 0. \tag{6b}$$

A third and final dynamical equation can be obtained by taking the divergence of the momentum equations. Denoting $\delta = \nabla \cdot \mathbf{u}$, one may write

$$\frac{\partial \delta}{\partial t} = 2J(u, v) - \nabla \cdot (\mathbf{u}\delta) + \frac{\partial}{\partial x}\left(fv - g'\frac{\partial h}{\partial x}\right) - \frac{\partial}{\partial y}\left(fu + g'\frac{\partial h}{\partial y}\right). \tag{6c}$$

Here $J(.,.)$ is the Jacobian operator and g' denotes gravity in the atmospheric context or 'reduced gravity', $g\Delta\rho/\rho$, in the oceanic context.

The velocity field is obtained from all three dynamic fields as follows. First, from q and h, one calculates $\zeta = hq - f$. Then, after solving for ψ and ξ from

$$\nabla^2\psi = \zeta \quad \text{and} \quad \nabla^2\xi = \delta, \tag{6d}$$

we obtain $\mathbf{u}$ from

$$u = \frac{\partial \xi}{\partial x} - \frac{\partial \psi}{\partial y} \quad \text{and} \quad v = \frac{\partial \xi}{\partial y} + \frac{\partial \psi}{\partial x}. \tag{6e}$$

In terrestrial applications, when the fluid motion is nearly 'balanced', it appears that both h and δ remain smooth, and hence decay rapidly with wave number (like k^{-3} or steeper, k being the magnitude of the wave number vector). Such a decay would permit one to evolve, accurately, h and δ in spectral space or on a grid using moderate resolution, unlike the PV, which would be evolved as outlined in this paper. It is expected that explicit dissipation will not be required to control the small-scale behaviour of h and δ, since balance means that this behaviour is controlled by q, and any unbalanced motions appear to have extremely weak amplitudes at high wave numbers.

Unfortunately, the best justification for this argument—a picture of the divergence spectrum—has not been found in the literature. The best evidence for a steep divergence spectrum has been provided by Norton (1988), who performed numerous polar vortex simulations spanning a wide range of Froude numbers for small Rossby number, by Ford (1994), who performed simulations of PV strip instabilities in a channel, and by Polvani *et al.* (1994), who performed turbulence simulations in doubly-periodic geometry. The divergence fields from the small Rossby- and Froude-number experiments do suggest that δ is as smooth as h, at least below the scale of the Rossby radius of deformation, $L_R = \sqrt{g'H}/f_0$, where H is the mean depth. One can appeal also to theoretical arguments, based on small Rossby-number expansions (e.g. see Vallis, 1996). The leading-order balance equation for δ reads

$$(\nabla^2 - 1/L_R^2)\delta = \frac{1}{f_0 L_R^2}(J(\psi, \zeta) + \beta v).$$

The bracketed term on the right hand side, to leading order in Rossby number, is just $-\partial\zeta/\partial t$, and so assuming the spectrum of $\partial\zeta/\partial t$ is similar to that of ζ at small scales, i.e. like k^{-1} or steeper, then the spectrum of δ at scales smaller than L_R should be at least as steep as k^{-3}. It is not expected that higher-order corrections in balance give rise to a shallower spectrum, since the basic controlling operator $(\nabla^2 - 1/L_R^2)$ remains the same at each level of iteration (assuming such an iteration converges, of course). As for the unbalanced motion, there should be extremely little amplitude at high wave numbers, since the basic (nearly balanced) vortical motion—the potential source of free gravity waves—predominantly excites low frequencies, $\omega \lesssim f_0$, whereas free gravity waves, particularly the small-scale ones, have very much greater frequencies, $\omega \approx \sqrt{f_0^2 + g'Hk^2}$ (where k is total wave number, as above). This suggests that the projection on small-scale, high-frequency gravity waves will be exponentially weak.

Whether or not numerical noise spoils this picture for δ is not known. It may be essential to use particularly accurate numerical methods, for example a semi-implicit semi-Lagrangian scheme (Staniforth and Coté 1991, Gravel 1996), which suppresses, as much as possible, high-frequency noise. The PV, q, would be evolved using the CASL algorithm; that is, q would be converted from a contour to a grid representation (and then perhaps to a spectral one) for use in obtaining the velocity field from Eqs. (6d) and (6e), then this velocity field would be used to advect the PV contours as in Contour Advection.

6. Conclusions

A new numerical algorithm has been presented which may prove useful for atmospheric and oceanic modelling. The algorithm essentially blends the Eulerian and Lagrangian representations in order to describe the dynamics of conservative, advected fields more accurately. Such fields often exhibit much fine-scale structure and are characterized by a shallow spectrum at large wave numbers. It is argued that such fields are better handled in a Lagrangian representation, as a series of contours lying in two-dimensional layers or surfaces. This representation is generally much less dissipative, principally because the field is retained to scales well below the computational grid-scale. This sub grid-scale field does not contribute significantly to the velocity field, but it is advected by it. Dissipation does not occur at the grid-scale, but well below it.

The algorithm has been described in detail for quasi-geostrophic flow, tested against contour surgery (which can be used in this case), and illustrated in the simulation of quasi-geostrophic turbulence. That progress has been made can be judged from the gain in efficiency the new algorithm has brought about: the test simulation in section 3(*c*) required five hours on a Fujistu VPX-240 vector processor; it would have required nearly four years for the contour-surgery algorithm; a pseudo-spectral simulation of equivalent accuracy (Yao *et al.* 1995) would probably require even greater resources.

The extension of the algorithm to ageostrophic systems appears feasible, and the prototypical case of the shallow-water system has been outlined. The basic principle is to hold fields having a steep spectral decay—fields which possess little fine-scale structure—in the (conventional) Eulerian representation, and to hold the PV and other conservative tracer fields having a shallow spectral decay in the Lagrangian representation. It is anticipated that it will no longer be necessary to employ hyperviscosity or any other *ad hoc* eddy viscosity, since the fields whose fine-scale structure needs to be controlled are dissipated ultimately by surgery, well below the grid-scale, and since the fields with a steep spectral decay produce an insignificant cascade to small scales.

One remaining hurdle is the incorporation of weakly-nonconservative effects. The PV is not exactly conserved in the atmosphere and the oceans. In particular, diabatic forcing, from the heating and cooling of chemical tracers (McIntyre 1995), dilutes and concentrates atmospheric PV (Haynes and McIntyre 1987). While the use of an associated grid in the new algorithm permits one to obtain the detailed spatial structure of the forcing, one cannot directly apply this to the contours representing the PV. Where a gradient of PV is well-defined, such forcing can be implemented simply by displacing contours. A piecewise-constant PV distribution is not suitable, and some sort of filtering or smearing is necessary to give a sensible gradient. But, then there is an additional difficulty near where the PV gradient vanishes, for there it may be necessary to create new contours.

Many of these difficulties may be overcome by splitting the PV into a conservative Lagrangian part and a non-conservative Eulerian part, evolving them using CASL and conventional methods, respectively, and recombining them only when the non-conservative part begins to lose significant enstrophy. The recombination would take place on the inversion grid, to which the fine-scale conservative part would be brought through successive averagings. New PV contours would be found by contouring this combined field. Though sub grid-scale PV structure would be smeared out in this procedure, its contribution to inversion would remain largely unaltered, as the results of section 3(*a*) have demonstrated. The advantage that this method would have over conventional methods would be a sharp reduction in unwanted numerical dissipation.

ACKNOWLEDGEMENTS

DGD is supported by the UK Natural Environment Research Council. Additional support was provided by the Isaac Newton Trust. MHPA is supported by the Netherlands Geosciences Foundation (GOA).

REFERENCES

Charney, J. G. 1971 Geostrophic turbulence. *J. Atmos. Sci.*, **28**, 1087–1095

Christiansen, J. P. 1973 Numerical simulation of hydrodynamics by the method of point vortices. *J. Comput. Phys.*, **13,** 363–379

Dritschel, D. G. 1988 Contour surgery: a topological reconnection scheme for extended integrations using contour dynamics. *J. Comput. Phys.*, **77,** 240–266

1989 Contour dynamics and contour surgery: numerical algorithms for extended, high-resolution modelling of vortex dynamics in two-dimensional, inviscid, incompressible flows. *Comput. Phys. Rep.*, **10,** 77–146

1993a A fast contour dynamics method for many-vortex calculations in two-dimensional flows. *Phys. Fluids A*, **5(1),** 173–186

1993b Vortex properties of two-dimensional turbulence. *Phys. Fluids A*, **5(4),** 984–997

1995 A general theory for two-dimensional vortex interactions. *J. Fluid Mech.*, **293,** 269–303

Dritschel, D. G. and Saravanan, R. 1994 Three-dimensional quasi-geostrophic contour dynamics, with an application to stratospheric vortex dynamics. *Q. J. R. Meteorol. Soc.*, **120,** 1267–1297

Dritschel, D. G. and de la Torre Juárez, M. 1996 The instability and breakdown of tall columnar vortices in a quasi-geostrophic fluid. *J. Fluid Mech.* **328,** 129–160

Ford, R. 1994 Gravity-wave radiation from vortex trains in rotating shallow water. *J. Fluid Mech.*, **281,** 81–118

Gage, K. S. and Nastrom, G. D. 1985 On the spectrum of atmospheric velocity fluctuations seen by MST/ST radar and their interpretation. *Radio Science*, **20,** 1339–1347

Gravel, S. 1996 The semi-Lagrangian method. In Centre de Recherches Mathematiques Proceedings. Nov. 7–22 1995, Montréal

Haynes, P. H. and McIntyre, M. E. 1987 On the evolution of vorticity and potential vorticity in the presence of diabatic heating and frictional or other forces. *J. Atmos. Sci.*, **44,** 828–841

Hoskins, B. J., McIntyre, M. E. and Robertson, A. W. 1985 On the use and significance of isentropic potential-vorticity maps. *Q. J. R. Meteorol. Soc.*, **111,** 877–946

Houghton, J. T. 1986 *The Physics of Atmospheres.* 2nd Edition, Cambridge University Press

Hua, B. L. and Haidvogel, D. B. 1986 Numerical simulations of the vertical structure of quasi-geostrophic turbulence. *J. Atmos. Sci.*, **43,** 2923–2936

Kraichnan, R. H. 1974 On Kolmogorov's inertial-range theories. *J. Fluid Mech.*, **62,** 305–330

Legras, B. and Dritschel, D. G. 1993 A comparison of the contour surgery and pseudo-spectral methods. *J. Comput. Phys.*, **104,** 287–302

Macaskill, C. and Bewick, B. M. 1995 The use of hyperviscosity in simulations of geophysical fluid flows. Pp. 415–418 in Proceedings of the 12th Australasian Fluid Mechanics Conference. Sydney, Australia, December 10–15

Mariotti, A., Legras, B. and Dritschel, D. G. 1994 Vortex stripping and the erosion of coherent structures in two-dimensional flows. *Phys. Fluids*, **6,** 3954–3962

McIntyre, M. E. 1995 The stratospheric polar vortex and sub-vortex: fluid dynamics and midlatitude ozone loss. *Phil. Trans. R. Soc. London,* **32,** 227–240

McWilliams, J. C. 1989 Statistical properties of decaying geostrophic turbulence. *J. Fluid Mech.*, **198,** 199–230

1990 The vortices of geostrophic turbulence. *J. Fluid Mech.*, **219,** 387–404

McWilliams, J. C., Weiss, J. B. and Yavneh, I. 1994 Anisotropy and coherent vortex structures in planetary turbulence. *Science*, **264,** 410–413

Methven, J. 1996 'Tracer behaviour in baroclinic waves'. Ph.D. thesis. University of Reading, England

Norton, W. A. 1988 'Balance and potential vorticity inversion in atmospheric dynamics.' Ph.D. thesis. University of Cambridge, England

1994 Breaking Rossby waves in a model stratosphere diagnozed by a vortex-following coordinate system and a technique for advecting material contours. *J. Atmos. Sci.*, **51,** 654–673

Pedlosky, J. 1979 *Geophysical Fluid Dynamics.* 2nd Edition, Springer Verlag, New York

Pierrehumbert, R. T. and Swanson, K. L. 1995 Baroclinic Instability. *Ann. Rev. Fluid Mech.*, **27,** 419–467

Plumb, R. A., Waugh, D. W., Atkinson, R. J., Schoeberl, M. R., Lait, L. R., Newman, P. A., Browell, E. V., Simmons, A., Loewenstein, M. and Toohey, D. W. 1994 Intrusions into the lower stratospheric arctic vortex during the winter of 1991/1992. *J. Geophys. Res.*, **99**, 1098–1106

Polvani, L. M. 1991 Two-layer geostrophic vortex dynamics. Part 2. Alignment and two-layer V-states. *J. Fluid Mech.*, **225**, 241–170

Polvani, L. M., McWilliams, J. C., Spall, M. A. and Ford, R. 1994 The coherent structures of shallow-water turbulence: deformation-radius effects, cyclone/anticyclone asymmetry and gravity-wave generation. *Chaos*, **4**, 177–187

Rhines, P. B. 1979 Geostrophic turbulence. *Ann. Rev. Fluid Mech.*, **11**, 401–441

Simmons, A. J., Hoskins, B. J. and Burridge, D. M. 1978 Stability of the semi-implicit method of time integration. *Mon. Weather Rev.*, **106**, 405–412

Staniforth, A. and Coté, J. 1991 Semi-Lagrangian integration schemes for atmospheric models. *Mon. Weather Rev.*, **119**, 2206–2223

Stegner, A. and Zeitlin, V. 1995 What can asymptotic expansions tell us about large-scale quasi-geostrophic anticyclonic vortices? *Nonlin. Proc. in Geophys.*, **2**, 186

Vallis, G. K. 1996 Potential vorticity inversion and balanced equations of motion for rotating and stratified flows. *Q. J. R. Meteorol. Soc.*, **122**, 291–322

Viera, F. 1995 On the alignment and axisymmetrization of a vertically tilted geostrophic vortex. *J. Fluid Mech.*, **289**, 29–50

Waugh, D. W. and Plumb, R. A. 1994 Contour advection with surgery: a technique for investigating finescale structure in tracer transport. *J. Atmos. Sci.*, **51**, 530–540

Waugh, D. W., Plumb, R. A., Atkinson, R. J., Schoeberl, M. R., Lait, L. R., Newman, P. A., Loewenstein, M., Toohey, D. W., Avallone, L. M., Webster, C. R. and May, R. D. 1994 Transport of material out of the stratospheric Arctic vortex by Rossby-wave breaking. *J. Geophys. Res.*, **99**, 1071–1088

Waugh, D. W. 1997 Elliptical diagnostics of stratospheric polar vortices. *Q. J. R. Meteorol. Soc.* (in press)

Yao, H. B., Dritschel, D. G. and Zabusky, N. J. 1995 High-gradient phenomena in 2D vortex interactions. *Phys. Fluids*, **7**, 539–548