Journal of Computational Physics 229 (2010) 5408–5417

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

The combined Lagrangian advection method

David G. Dritschel *, Jérôme Fontane

School of Mathematics and Statistics, University of St Andrews, St Andrews KY16 9SS, Scotland, United Kingdom

ARTICLE INFO

Article history:
Received 14 January 2010
Received in revised form 31 March 2010
Accepted 31 March 2010
Available online 8 April 2010

Keywords:
Pseudo-spectral
Contour advection
Vortex methods
Two-dimensional turbulence

ABSTRACT

We present and test a new hybrid numerical method for simulating layerwise-two-dimensional geophysical flows. The method radically extends the original Contour-Advective Semi-Lagrangian (CASL) algorithm [5] by combining *three* computational elements for the advection of general tracers (e.g. potential vorticity, water vapor, etc.): (1) a pseudo-spectral method for large scales, (2) Lagrangian contours for intermediate to small scales, and (3) Lagrangian particles for the representation of general forcing and dissipation. The pseudo-spectral method is both efficient and highly accurate at large scales, while contour advection is efficient and accurate at small scales, allowing one to simulate extremely fine-scale structure well below the basic grid scale used to represent the velocity field. The particles allow one to efficiently incorporate general forcing and dissipation.

1. Introduction

Numerical simulations of atmospheric and oceanic fluid dynamics must deal with a vast range of active spatial and temporal scales of motion. Much of this range lies beyond the limit of numerical resolution, requiring small scales (and high frequencies) to be instead parametrised by 'eddy-diffusivity' or 'closure' schemes meant to approximate the collective effects of unresolved motions on resolved ones (cf. [17] and references). Here, we describe a new modelling advance which reduces the need for closure schemes by allowing one to efficiently extend the range of resolved scales, in particular for advected tracers. The new advance is the culmination of years of model development based on "Contour Advection" (CASL, [5]), a hybrid Lagrangian–Eulerian method stemming originally from "Contour Surgery" [4] and "Contour Dynamics" [20]. The new method, called the "Combined Lagrangian Advection Method" (CLAM), utilises three computational elements—contours, particles, and grid points (or spectral coefficients)—combined in a way to optimise performance and accuracy.

While CLAM is built for accurate conservation in the absence of forcing and dissipation, it also allows one to efficiently handle general non-conservative processes such as thermal heating, Ekman friction, and stochastic forcing [7,16,13]. Moreover, it may offer distinct advantages over commonly-used numerical methods in Geophysical Fluid Dynamics when multiple tracers (dynamical, chemical, biological) are considered.

In Section 2 below, we outline the structure of the method. It is next illustrated and tested in an example of forced two-dimensional turbulence in Section 3. Then it is applied to study an aspect of the banded circulation patterns found in planetary atmospheres in Section 4. Conclusions and ideas for further model development are offered in Section 5.

2. The method

CLAM was developed originally to better model both unforced and forced 2D turbulence at ultra-high Reynolds numbers [9–11,13]. It is an extension of the recent HyperCASL algorithm [13], which introduced the idea of using point vortices or

* Corresponding author. Tel.: +44 1334 463721; fax: +44 1334 463748.
E-mail address: dgd@mcs.st-and.ac.uk (D.G. Dritschel).

0021-9991/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2010.03.048

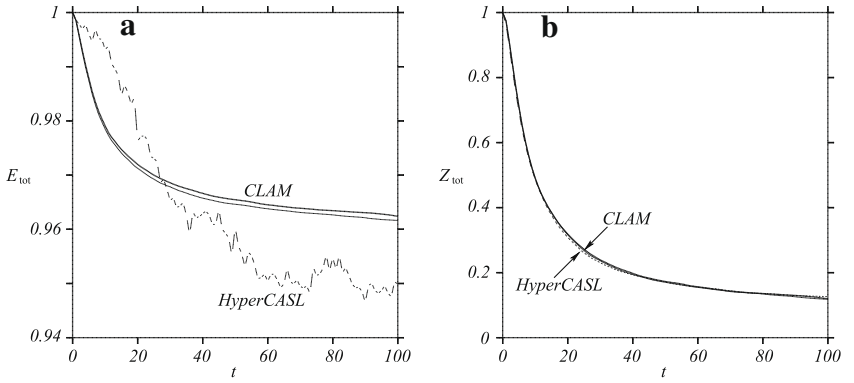

Fig. 1. Comparison of the evolution of (a) energy $E_{tot}(t)$ (normalised by $E_{tot}(0) = 0.0148588$) and (b) enstrophy $Z_{tot}(t)$ (normalised by $Z_{tot}(0) = 4.046645$) between HyperCASL (dashed) and CLAM (solid) in the case of freely-decaying two-dimensional turbulence examined in Fontane and Dritschel [13]. The two curves shown for CLAM correspond to two different ways of representing q_d, by particles (bold) or by a spectral method (thin). See text for details.

particles to represent a *residual* tracer field q_d (e.g. vorticity in 2D turbulence or potential vorticity in rotating stratified flows). The residual tracer field q_d is used as a temporary reservoir for any explicit forcing and dissipation $S(\boldsymbol{x},t)$ operating on the full tracer field q:

$$\frac{Dq}{Dt} = S(\boldsymbol{x},t) \tag{1}$$

At any instant of time t, the full tracer field q is the sum

$$q = q_a + q_d \tag{2}$$

in which q_a evolves *conservatively* by contour advection, i.e.

$$\frac{d\boldsymbol{X}_a}{dt} = \boldsymbol{u}(\boldsymbol{X}_a,t) \tag{3}$$

(equivalent to $Dq_a/Dt = \partial q_a/\partial t + \boldsymbol{u}\cdot\nabla q_a = 0$), where $\boldsymbol{X}_a$ is a point on a contour and $\boldsymbol{u}(\boldsymbol{x},t)$ is the velocity field, while q_d evolves by advecting discrete particles $\boldsymbol{X}_d$,

$$\frac{d\boldsymbol{X}_d}{dt} = \boldsymbol{u}(\boldsymbol{X}_d,t) \tag{4}$$

as well as adjusting the intensities Γ_d of individual particles to match the imposed forcing and dissipation (see Appendix A). Every 4 eddy-turnaround times T_{eddy} (determined from the maximum vorticity integrated over time), q_d is transferred to a set of contours representing the *primary* tracer field q_a through an efficient contouring procedure [7]. Standard bi-linear interpolation is used to create a corresponding gridded field of q_d from the particles when needed (see Appendix A).

Weak numerical dissipation occurs during the regularization of contours by "surgery" [4], and when resetting the particles on a regular array. Both are done approximately every $0.2T_{eddy}$. Notably, in simulations of freely-decaying two-dimensional turbulence, it has been shown that this dissipation is comparable to that needed in a conventional spectral method using a grid 10–20 times finer than used in HyperCASL and CLAM to represent the velocity field $\boldsymbol{u}$ [11].

An unwanted feature of HyperCASL is the introduction of a small level of numerical error or stochastic noise in q_a by the contour-to-grid conversion procedure and, to a much lesser extent, by contour surgery [13]. Unfortunately, this noise is statistically uniform across Fourier modes and it generates a growing k^1 tracer variance spectrum at small k. In simulations of 2D turbulence [13], this gives rise to a growing k^{-1} energy spectrum at small k (the actual energy spectrum normally decays rapidly as $k \to 0$, see e.g. [10]).[1] As a consequence, this noise primarily affects the energy while enstrophy (vorticity variance) is more robust, see Figs. 1 and 2. And in long-time simulations, it can eventually lead to significant erroneous energy loss.

CLAM removes this unwanted feature (see Fig. 1(a)) by using a pseudo-spectral (PS) method to model large scales, specifically wavenumbers $k \leqslant k_c$, where k_c is the 'filter cutoff wavenumber'. The PS method is well-designed for this purpose and, moreover, is both accurate and efficient. In CLAM, the tracer field computed this way, denoted q_s, is blended with the primary tracer field q_a (represented by contours). The full tracer field is obtained now from

$$\hat{q} = F\hat{q}_s + (1-F)\hat{q}_a + \hat{q}_d \tag{5}$$

where a hat denotes a spectral transform, and $F(k)$ is a low-pass filter (see below). The only difference between HyperCASL and CLAM is the replacement of q_a in (2) by a weighted sum of q_s and q_a. The form of the filter $F(k)$ was fixed after extensive numerical tests (see below), and it takes the form

[1] See Fig. 8 in the online notes at www-vortex.mcs.st-and.ac.uk/HyperCASL.pdf accompanying Fontane and Dritschel [13].

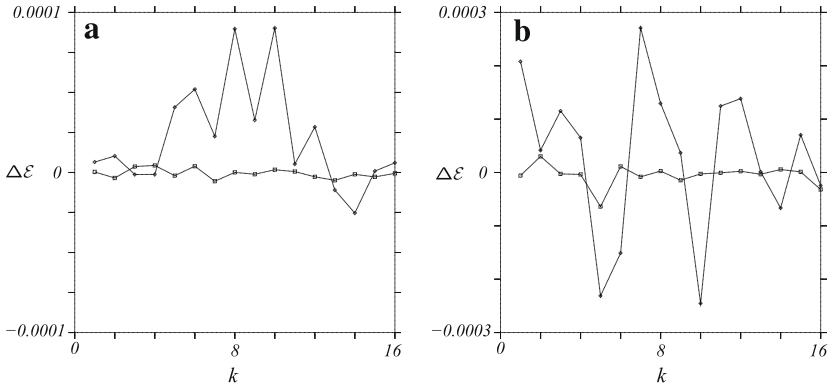

Fig. 2. Difference energy spectra $\Delta\mathscr{E}(k,t)$ as a function of wavenumber k, at low k, between HyperCASL and CLAM with particles (thin solid line with diamonds) and between CLAM with spectral q_d and CLAM with particles (bold solid line with squares), at (a) $t = 2$ and (b) $t = 5$. Note, $\mathscr{E}(k,t)$ is sum of the energy $|\boldsymbol{u}|^2/2$ in spectral components with wavevectors $\boldsymbol{k}$ satisfying $|\boldsymbol{k}| = k = 1,2,\ldots$ (nearest integer). The results were obtained from the turbulence simulations examined in Fig. 1.

$$F(k) = \frac{1}{1 + (k/k_c)^4}. \tag{6}$$

This is known as the second-order Butterworth filter [1], and it has a maximal flatness property for $k \leqslant k_c$. Additionally, there is a smooth transition between $\hat{q}_s$ and $\hat{q}_a$ around the filter cutoff wavenumber k_c.

The introduction of q_s requires another dynamical evolution equation. Here, we evolve q_s *conservatively*, i.e.

$$\frac{\mathrm{D}q_s}{\mathrm{D}t} = 0 \tag{7}$$

with no dissipation whatsoever. This is of course unsustainable in general, as $\hat{q}_s$ will cascade to high k by advection, and the solution will become unphysical if these high k components of $\hat{q}_s$ are not removed, e.g. by numerical diffusion. Instead, in CLAM, q_s and q_d are *reinitialised* at the beginning of every time-step. The field q_s is set to the *entire* q field obtained at the end of the previous time-step. Meanwhile, the field q_d is chosen to ensure that q is unchanged when q_s, q_a and q_d are next recombined. Mathematically, this means taking

$$\hat{q}_s = \hat{q} \quad \text{and} \quad \hat{q}_d = (1 - F)(\hat{q} - \hat{q}_a) \tag{8}$$

where $\hat{q}$ is the (spectral) field at the end of the previous time-step. When they are next recombined, we find

$$F\hat{q}_s + (1 - F)\hat{q}_a + \hat{q}_d = F\hat{q} + (1 - F)\hat{q}_a + (1 - F)(\hat{q} - \hat{q}_a) = \hat{q} \tag{9}$$

as required. A fourth-order Runge–Kutta procedure is convenient for the subsequent integration over a single time-step.

This reinitialisation procedure avoids the erroneous random variations of $\hat{q}_a$ across all k and moreover allows an accurate estimate of the advection term $\boldsymbol{u} \cdot \nabla q_s$ needed for evolving $\hat{q}_s$. Furthermore, any forcing or dissipation in $\hat{q}_d$ in wavenumbers $k \lesssim k_c$ is transferred every time-step to $\hat{q}_s$, thereby minimizing numerical diffusion in $\hat{q}_d$. This is especially important for forced flows, as we shall see in Section 3. Regarding efficiency, in forced flows CLAM is only a few percent slower than HyperCASL and significantly more accurate (a comparison is provided below). The extra cost associated with the spectral method, at the resolution required for its use in CLAM, is minimal due to the spectral method's particularly high efficiency compared to the contour and particle methods.

The reinitialisation of q_d does however require a transfer of the gridded q_d to the particle intensities Γ_d. This is not an efficient procedure but is required every time-step in forced flows. In unforced flows, such as considered in Fontane and Dritschel [13] for HyperCASL, this transfer is required only when the particle array is reset—in practice only after many time-steps. As a result, HyperCASL is relatively efficient, approximately 60% faster than CLAM in the simulation of unforced 2D turbulence illustrated in Fig. 1 (discussed in [13]). We can gain back this loss of efficiency, and more, if we replace the particle method for q_d by a spectral method (as in DCASL, see [7,16]) when simulating unforced flows, or flows forced or damped predominantly at large scales (e.g. thermal forcing in geophysical flows). But the use of a spectral method for q_d requires some sort of numerical diffusion for stability. The simplest approach is hyperdiffusion, i.e.

$$\frac{\mathrm{D}q_d}{\mathrm{D}t} = \nu_d(-1)^{p+1}\nabla^{2p}q_d, \tag{10}$$

and after extensive tests varying both the power p and the coefficient ν_d, we have adopted the choice $p = 2$ (bi-harmonic hyperviscosity) together with $\nu_d = 5\zeta_{\mathrm{rms}}(t)/k_m^4$, in which k_m is the maximum wavenumber associated with the inversion grid and ζ_{rms} is the r.m.s. (vertical) vorticity.[2] Besides maintaining numerical stability and greatly reducing Gibbs fringes near high

[2] Note: hyperdiffusion is applied *only* to the residual field q_d, *not* to q_s.

Table 1
Comparison of algorithm efficiency (using a 2.8 GHz Intel processor) in a simulation of freely-decaying 2D turbulence, cf. Fig. 1 and [13]. Here, HyperCASL is compared with two versions of CLAM, the first using a pseudo-spectral (PS) method for q_d and the second using a particle-in-cell (PIC) method. Here, CLAM with PS q_d is 8% more efficient than HyperCASL, while CLAM with PIC q_d is 60% less efficient.

Numerical algorithm	Cost (CPU seconds)
HyperCASL	2669
CLAM with PS q_d	2459
CLAM with PIC q_d	4285

gradients in q_d, this choice of hyperdiffusion gives the closest comparison to the results we find when using particles for q_d. Using a spectral method for q_d rather than particles is guaranteed to be more efficient. In the unforced simulation illustrated in Fig. 1, this spectral version of CLAM results in a nearly 10% gain in efficiency over HyperCASL—see Table 1—and achieves significantly greater accuracy. Although CLAM with PS q_d is slightly less accurate than CLAM with PIC q_d, it is nearly twice as efficient.

A small modification of the recontouring procedure in HyperCASL [13] was made in CLAM, following extensive tests, to improve the accuracy of this procedure. In recontouring, the difference field $q' = q - q_a$ at the end of a simulation period is extrapolated from the basic 'inversion' grid to an ultra-fine contouring grid (16 times finer in each direction) and added to the contour-associated field q_a on this grid (whose smallest scale is the scale of surgery). This combined field is then recontoured, and the error in contouring with a finite contour interval Δq present on the inversion grid is given to q_d to start the next simulation period. There is then no change to the total q on the inversion grid following recontouring. In HyperCASL, we used bi-linear interpolation to extrapolate q' to the contouring grid, whereas in CLAM (and now in HyperCASL too) we use spectral interpolation (zero padding) for greater accuracy. In this way we retain all information in q'. Spectral interpolation results in no significant loss in efficiency.

A brief history of the key developments leading to CLAM is given in Fig. 3.

3. A test case

To justify our choice of the filter $F(k)$ and the filter cutoff wavenumber k_c, we next examine a set of numerical simulations of forced 2D turbulence in a square doubly-periodic domain of width 2π. Note, here q is the 2D (barotropic) vorticity. We have carried out simulations at three different resolutions and for widely varying forcing wavenumbers k_f. Following many previous studies, we have used narrow-band forcing in which a random-phased top-hat enstrophy spectrum, constant over $k_f - \Delta k \leqslant k \leqslant k_f + \Delta k$, is added every time-step to ensure a constant rate of enstrophy growth, here 7.8957, in the absence of

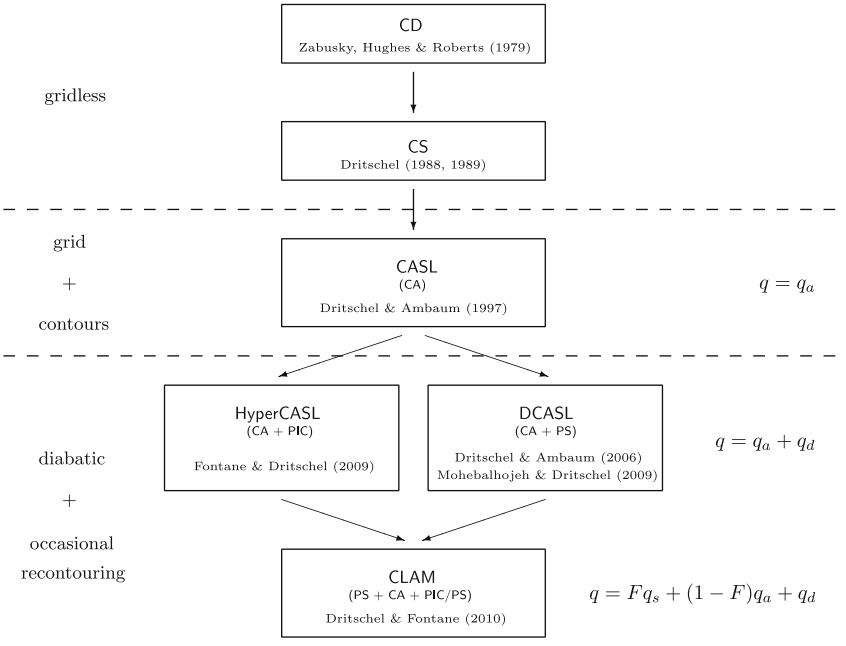

Fig. 3. A genealogical tree for numerical methods based wholly or in part on contour advection. Here CD = contour dynamics, CS = contour surgery, DCASL = diabatic CASL, PS = pseudo-spectral and PIC = particle-in-cell.

dissipation. This forcing results in a nearly constant energy growth, at the rate $\varepsilon = 0.0077409$, over the times considered. Note, the flow starts from a state of rest.

We focus on one simulation with a basic 'inversion grid' of 256×256 (with maximum wavenumber $k_m = 128$), forced at $k = k_f = 32$ (with $\Delta k = 2$), and using all of the standard parameter settings recommended in Fontane and Dritschel [13]. In particular, we use a contour interval $\Delta q = \pi/5 \approx 0.6283$ for representing q_a (using an estimated unit eddy-turnaround time). The evolution is simulated over a moderate time period, $0 \leqslant t \leqslant 20$, corresponding to nearly $80T_{\text{eddy}}$ based on the time integrated maximum vorticity, or $10.5T_{\text{eddy}}$ based on the time integrated r.m.s. vorticity. A few stages in the evolution (at $t = 2$, 5, 10 and 20) are shown in Fig. 4 (note, only 1/16th of the domain is shown). In the earliest stage shown (upper left), the direct effect of the forcing is most evident; the flow has not yet had time to evolve significantly. By the next stage, one can already see the formation of vortices surrounded by cascading filamentary debris. The later stages are closely similar, albeit with more fine-scale structure and marginally more organised large-scale vortices (the energy grows by cascading slowly to large scales from the forcing wavenumber).

In spectral space (see Fig. 5), energy $\mathscr{E}(k,t)$ (shell averaged over wavevectors $\boldsymbol{k}$ of magnitude $k = |\boldsymbol{k}|$) spreads from its initial forcing region near $k = k_f = 32$ and rapidly becomes nearly time-independent except at small k. There, energy grows as it cascades to large scales, resulting in a growth of the total energy $E_{\text{tot}}(t) = \int \mathscr{E}(k,t)\mathrm{d}k$. This energy growth is approximately linear in time, as expected statistically from narrow-band random forcing at large k_f. Fig. 6 (left panel) compares the observed growth for various simulations with the theoretical linear growth (dotted line). Two CLAM simulations are illustrated, one using particles for q_d (bold line) and the other using a spectral method for q_d (thin line). The HyperCASL simulation (lower dashed line) shows poor conservation properties. The CLAM simulations, by contrast, closely match the theoretical growth apart from a slight deficit at early times, during the first period of integration from rest without contours ($0 \leqslant t \leqslant 2.846$). These tendencies are more clearly seen in the right panel, which shows the *difference* between the actual and theoretical energy, normalised by the final energy at $t = t_f = 20$.

Fig. 7 compares the fractional energy error in CLAM simulations using (a) different filter cutoff wavenumbers k_c for the same second-order Butterworth filter (6), and (b) different orders of the Butterworth filter for the same cutoff wavenumber, $k_c = 42$ ($\approx k_m/3$). We see in (a) that $k_c = 42$ gives the best fit to the theoretical prediction, while in (b) that the second-order Butterworth filter gives the best fit. Poorer results (not shown) were found for the Gaussian filter $F(k) = e^{-(k/k_c)^2}$ and the sharp filter $F(k) = 1$ for $k \leqslant k_c$ and $F(k) = 0$ otherwise. The dependence on k_c can be explained by noting that, for small k_c, most of the dynamics is controlled by the contours in q_a, and we know from Fontane and Dritschel [13] that contour advection alone can lead to an erroneous growth of energy at low wavenumbers, resulting in an enhanced growth of E_{tot}. Conversely, for large k_c, most of the dynamics is controlled by the spectral evolution which is more diffusive than contour advection, resulting in a

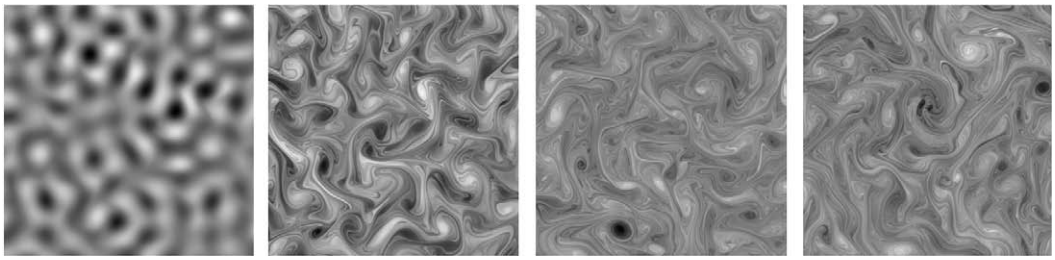

Fig. 4. Evolution of the vorticity field in a CLAM simulation of forced 2D turbulence using a basic 256×256 grid (contours are retained to scales 16 times smaller). Times $t = 2$,5, 10 and 20 are shown from left to right. Only 1/16th of the domain is shown. A linear grey scale is used between $-|q|_{\max} < q < |q|_{\max}$ (black being most negative).

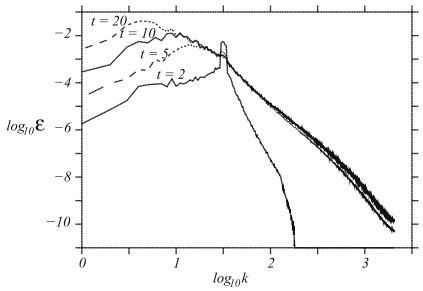

Fig. 5. Energy spectra $\mathscr{E}(k,t)$ at the times corresponding to the images in Fig. 4. Note that the spectral tails extend well beyond the maximum wavenumber $k_m = 128$ associated with the inversion grid. These tails have slopes between k^{-4} and k^{-3}, and do not change significantly after $t = 10$. Instead, the spectra build at low wavenumbers (large scales), consistent with a linear growth in total energy (see text).

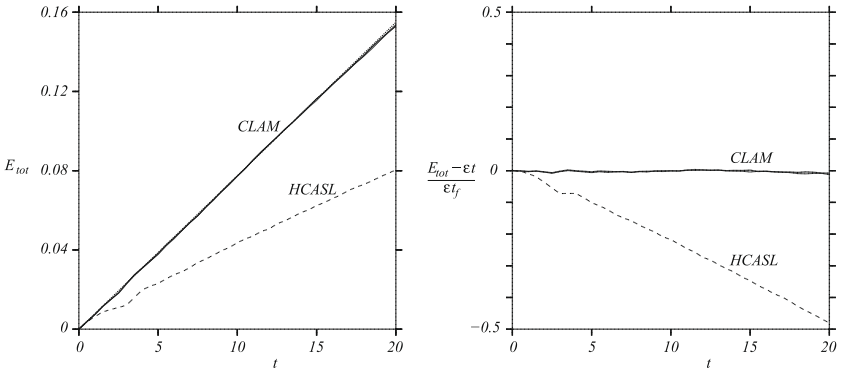

Fig. 6. Total energy $E_{\rm tot}(t)$ (left) versus time in several simulations of forced 2D turbulence, with forcing centred on wavenumber $k = k_f = 32$. The bold, thin and dashed curves correspond to CLAM using particles for q_d, CLAM using a spectral method for q_d, and HyperCASL, respectively, while the dotted line gives the theoretical growth. The figure on the right shows the fractional energy error, $(E_{\rm tot} - \varepsilon t)/\varepsilon t_f$, where $\varepsilon = 0.0077409$ is the theoretical energy growth rate and $t_f = 20$ is the final time. The three simulations use all the standard numerical parameters set out in Fontane and Dritschel [13]. In addition, the two CLAM simulations here use the second-order Butterworth filter (6) with cutoff wavenumber $k_c = k_m/3 = 42$.

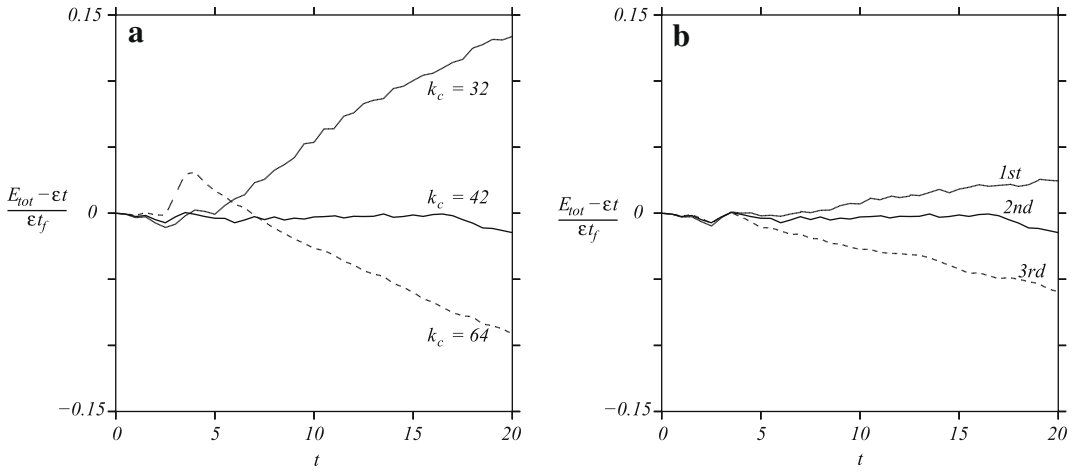

Fig. 7. (a) Fractional energy error $(E_{\rm tot} - \varepsilon t)/\varepsilon t_f$ versus time for three CLAM simulations using the second-order Butterworth filter (6) with cutoff wavenumbers $k_c = 32$ (thin line), 42 (bold) and 64 (dashed). A least-squares fit of $E_{\rm tot}(t)/\varepsilon$ to t gives a fractional slope error of 12.5%, −0.335% and −7.27%, respectively. (b) As in (a) except now for $k_c = 42$ and for the following filters $F(k)$: first-order Butterworth $[1 + (k/k_c)^2]^{-1}$ (thin line), second-order Butterworth $[1 + (k/k_c)^4]^{-1}$ (bold), and third-order Butterworth $[1 + (k/k_c)^6]^{-1}$ (dashed). The fractional slope errors are 2.17%, −0.335% and −5.45%, respectively.

retarded growth of $E_{\rm tot}$. From these results, and many other simulations for different ratios of k_f/k_m, we have found that the second-order Butterworth filter with a filter cutoff wavenumber $k_c = k_m/3$ is most accurate.

There is a limit, however, to how large one can make the ratio k_f/k_m. Above, we examined $k_f/k_m = 1/4$, but when using a larger value like $k_f/k_m = 1/2$ we observed an early retarded growth of $E_{\rm tot}$ followed by a later growth having a slope slightly lower than expected (not shown). Every effort was made to correct this growth, from modifying the numerical parameter settings in CLAM, to using a smaller contour interval Δq, to changing the bi-linear interpolation of the velocity field on contour points and particles to the third-order M'_4 scheme of Cottet et al. [3], or the incompressible quadratic-spline method of Handscomb [14], and even to replacing CLAM with a full pseudo-spectral simulation (on a four times finer inversion grid) for the early simulation period. None of these improved the growth of $E_{\rm tot}$, and all were more costly.

This limitation on the size of k_f/k_m is, however, not surprising. When k_f is only half k_m, there is little room for resolving the nonlinear spread of the forcing through q_d, and this results in numerical dissipation. Relative to the maximum effective wavenumber associated with the ultra-fine grid, here $16k_m$, our tests show that k_f should be at least 64 times smaller. Independently, using a pure pseudo-spectral method, Scott [18] has found that the maximum wavenumber must be at least 64 times the forcing wavenumber to properly model the inverse cascade in two-dimensional turbulence. It is therefore reasonable to limit k_f/k_m to 1/4 in CLAM when narrow-band spectral forcing is used. Other common types of forcing, such as thermal forcing in geophysical flows, tend to be broad-band with significant input at large scales. In general, these types of forcing can be more easily represented in numerical models, and CLAM is no exception. Indeed, CLAM like its predecessor

Table 2
Comparison of algorithm efficiency in a simulation of forced 2D turbulence, cf. Fig. 4. In this case, both versions of CLAM are less efficient (28% and 41%, respectively) than HyperCASL, but HyperCASL is unable to capture the correct linear energy growth.

Numerical algorithm	Cost (CPU seconds)
HyperCASL	7698
CLAM with PS q_d	9856
CLAM with PIC q_d	10,864

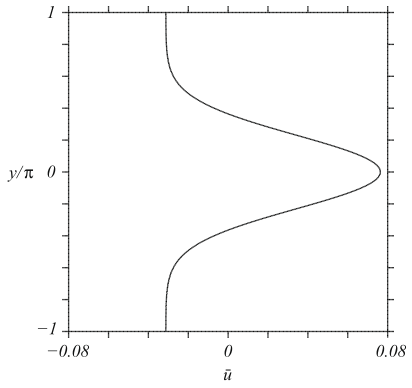

Fig. 8. Basic-state velocity profile $\bar{u}$ used in the jet sharpening experiment.

HyperCASL offers new possibilities for forcing, e.g. by introducing new particles carrying circulation and other attributes [13]. Such forcing may be particularly effective in representing subgrid-scale convection in a model of the atmosphere.

As regards efficiency, CLAM is *less* efficient than its predecessor HyperCASL as indicated in Table 2. However, the extra work in CLAM is necessary, in the case of forced 2D turbulence analysed above, to accurately capture the correct growth in energy (see Fig. 6). The retarded energy growth seen in HyperCASL is a consequence of retaining the forcing wholly in particles, which diffuse vorticity every time they are reset on a regular array. By contrast, CLAM transfers most of the forcing, every time-step, to a spectral method (that used for q_s), thereby preserving nearly all of the added vorticity. This, we emphasize, is an especially important characteristic of CLAM. Hence, CLAM not only represents the large-scale dynamics optimally (i.e. with a spectral method), but also represents complex forcing accurately.

4. An application to jet sharpening

As an illustration of the method in a geophysical context, we next examine a simulation of "jet sharpening" in a rapidly-rotating, stably-stratified flow (see [8] and references). We consider the unforced single-layer shallow-water quasi-geostrophic model

$$\frac{Dq}{Dt} = 0$$
$$\left(\nabla^2 - L_D^{-2}\right)\psi = q - f; \boldsymbol{u} = (u, v) = \left(-\frac{\partial\psi}{\partial y}, \frac{\partial\psi}{\partial x}\right) \tag{11}$$

where q is the potential vorticity (PV), $f = f_0 + \beta y$ is the background planetary vorticity or Coriolis frequency, and ψ is the streamfunction. The length scale $L_D = \sqrt{gH}/f_0$ is the Rossby deformation length, where g is gravity, H is the mean fluid depth, and f_0 is the mean Coriolis frequency.

We consider just one example, with $L_D = 1/8$, $\beta = 2$, and a broad basic-state PV distribution $\bar{q}(y) = \beta\pi\mathrm{erf}(y)$ in a square doubly-periodic domain of width 2π. (Here, f_0 has been absorbed into the definition of q, by replacing $q - f$ by $q - \beta y$ in (11).) This PV distribution induces a broad jet $\bar{u}(y)$ with maximum speed 0.07635 at $y = 0$ and zero mean, see Fig. 8. This basic-state flow is steady and stable due to the monotonic variation of PV. To induce jet sharpening—a tightening of PV gradients—a random isotropic PV field with energy spectrum $\mathscr{E}(k) = ck^3\exp(-2k^2/k_0^2)$ and $k_0 = 16$ is superimposed on the basic-state PV. The constant c is determined by specifying the r.m.s. amplitude of this field, here four times the PV contrast across the jet $\Delta q = 2\pi\beta$. A few stages of the evolution are shown in Fig. 9. The initial turbulence strongly disrupts the jet, but soon the flow re-organises into a more concentrated jet (actually a double jet, see below) together with a number of persistent vortices. At late times, the jet is clearly visible as the wavy contrast in PV in the middle of the domain. Notably, at these times the vortices remain on either side of the jet and move together with the jet meanders.[3]

[3] A gif movie of this simulation is available at www-vortex.mcs.st-and.ac.uk.

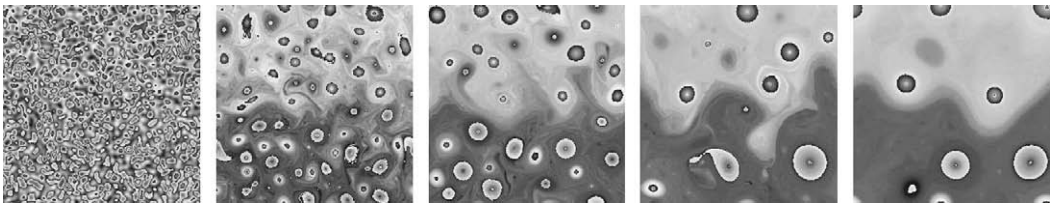

Fig. 9. Evolution of the PV field in an example of jet sharpening in a single-layer quasi-geostrophic flow. The simulation uses CLAM on a basic 256×256 grid. Times $t = 0$, 25, 100, 400 and 1000 are shown from left to right. A linear grey scale is used between $-2\pi\beta < q_{\rm mod} < 2\pi\beta$, where $q_{\rm mod} = q$ modulo $2\pi\beta$.

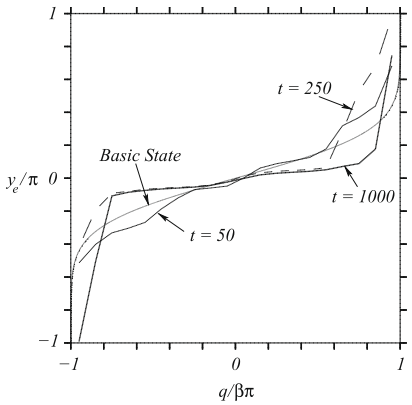

Fig. 10. The mean y location of periodically-wrapping contours of constant PV q at various times indicated. The basic-state profile is shown for comparison.

The effect of the turbulence is to accelerate the initial jet, here by more than a factor of 2.5 (along-jet speeds reach approximately 0.2). This acceleration is directly associated with tightening PV gradients [8], seen in the profiles of $y_e(q)$ displayed in Fig. 10 (notice the double-jet structure—there are two plateaux in $y_e(q)$). The function $y_e(q)$ gives the mean y location of the PV contours q = constant which wrap the domain periodically. These contours belong to the jet, and are distinct from the non-wrapping contours mainly found around the vortices. Fig. 10 demonstrates how turbulence can greatly steepen PV gradients, nearly to the point of a discontinuity.

5. Conclusions

We have described and examined a new hybrid computational method, CLAM, combining three numerical elements: pseudo-spectral, contour advection and particle-in-cell. By using each element selectively across spatial scales, CLAM achieves a substantially greater efficiency and accuracy than is possible when using any one element on its own. This has been explicitly demonstrated for HyperCASL in Fontane and Dritschel [13], and CLAM is by design more accurate for little extra cost. The use of a pseudo-spectral (PS) method allows one to accurately represent the effects of forcing, and moreover avoids the slow, erroneous growth of large-scale modes caused by errors in the contour-to-grid conversion procedure and in contour surgery. The PS method is relatively efficient compared to contour advection and particle-in-cell, and therefore numerical efficiency is not significantly degraded. While the PS method is stable only for a time-step satisfying the CFL constraint $\Delta t < \Delta x/|\boldsymbol{u}|_{\rm max}$, in realistically complex geophysical flows the time-step is nearly always limited by the maximum vertical vorticity: $\Delta t < \pi/10|\zeta|_{\rm max}$ [13]. The CFL constraint applies only to flows dominated by large-scale motions.

As the example in Section 4 indicates, CLAM is not limited to the study of idealised flows like two-dimensional turbulence. It has been widely extended to both single and multi-layer quasi-geostrophic flows subject to general forcing and dissipation mechanisms (cf. [11,12]). CLAM has also been adapted to a periodic channel geometry, and work is in progress on a spherical extension. In different contexts, CLAM has been implemented to study two-dimensional rotating shallow-water flows, density-stratified flows in a vertical cross-section using density as the tracer q [15], and idealised magneto-hydrodynamics where the (potential) vorticity q is subject to the Lorenz force [19].

Further extensions, e.g. to three-dimensional rotating stratified flows are feasible, requiring straightforward adaptions of existing CASL codes [6]. CLAM also presents a promising opportunity for accurately modelling more realistic atmospheric and oceanic flows. CLAM can carry more than one tracer q. Additional chemical and biological tracers (those that are advection dominated) may be represented by additional sets of contours (with natural parallelisation possibilities). But, *only one set of particles* is required for *all* tracers. Each particle would contain a list of attributes, related to the residue q_d belonging to each tracer. Any forcing or dissipation (including chemical reactions, changes of state, etc.) would only modify the attributes

of a particle, not its position, which is determined by simple advection, i.e. by (4). This appears to be a particularly efficient procedure for representing a collection of interacting tracers.

Acknowledgments

The authors thank C. Macaskill for his comments and suggestions during the preparation of the manuscript. This research was supported by a Marie Curie Intra European Fellowship within the 7th European Community Framework Programme (Grant No. PIEF-GA-2008-221003).

Appendix A. Particle-in-cell in CLAM

The particles in CLAM used to represent the residual field q_d are handled as in a conventional Particle-in-Cell (PIC) method, e.g. as in Christiansen and Zabusky [2]. The velocity at each particle, say at $\boldsymbol{x} = \boldsymbol{X}_d$, is found using bi-linear interpolation (fractional area weighting):

$$\boldsymbol{u}(\boldsymbol{X}_d) = w_{i,j}\boldsymbol{u}_{i,j} + w_{i+1,j}\boldsymbol{u}_{i+1,j} + w_{i,j+1}\boldsymbol{u}_{i,j+1} + w_{i+1,j+1}\boldsymbol{u}_{i+1,j+1} \tag{A.1}$$

where $\boldsymbol{u}_{i,j}$ is the velocity at the grid point $(\bar{x}_i, \bar{y}_j)$, while the $w_{i,j}$ are the interpolation weights

$$w_{i,j}(\boldsymbol{X}_d) = \left(1 - \frac{|X_d - \bar{x}_i|}{\Delta x}\right)\left(1 - \frac{|Y_d - \bar{y}_j|}{\Delta y}\right) \tag{A.2}$$

for fixed grid spacing Δx and Δy in the x and y directions. We note in passing that the same interpolation method is used for the nodes $\boldsymbol{x} = \boldsymbol{X}_a$ representing the contours.

To obtain gridded values of q_d when needed (for instance when computing $\boldsymbol{u}$ from q), an analogous procedure is used in reverse. Each particle's intensity Γ_d is divided amongst the corners of the grid box containing $\boldsymbol{X}_d$, incrementing the value of q_d there:

$$q_{di+m,j+n} \leftarrow w_{i+m,j+n}\Gamma_d \quad (m = 0, 1 \quad \text{and} \quad n = 0, 1). \tag{A.3}$$

In effect, q_d at each grid point is found by summing all particles (now indexed by k) in the adjacent grid boxes:

$$q_{di,j} = \sum_{k \in B_{i,j}} w_{i,j}(\boldsymbol{X}_{dk})\Gamma_{dk} \tag{A.4}$$

where $B_{i,j}$ denotes the four grid boxes surrounding the grid point.

Any explicit forcing and/or dissipation $S(\boldsymbol{x}, t)$ of the tracer q, see (1), is fully accounted for by setting $\mathrm{D}q_d/\mathrm{D}t = S(\boldsymbol{x}, t)$. This is naturally solved by a set of particles moving with the fluid. To do this, the variations in the particle intensities Γ_{dk} are found by the analogue of (A.4),

$$S_{i,j} = \sum_{k \in B_{i,j}} w_{i,j}(\boldsymbol{X}_{dk}) \frac{\mathrm{d}\Gamma_{dk}}{\mathrm{d}t}, \tag{A.5}$$

except now we seek $\mathrm{d}\Gamma_{dk}/\mathrm{d}t$, given $S = S_{i,j}$ over the grid. This results in a coupled system which requires numerical iteration, but in practice it is rapidly convergent (see Fontane and Dritschel [13] for further details).

References

[1] S. Butterworth, On the theory of filter amplifiers, Experimental Wireless and the Wireless Engineer 7 (1930) 536–541.
[2] J.P. Christiansen, N.J. Zabusky, Instability, coalescence and fission of finite-area vortex structures, J. Fluid Mech. 61 (1973) 219–243.
[3] G.H. Cottet, M.L. Ould Sahili, M. El Hamraoui, Multi-purpose regridding in vortex methods, in: ESAIM Proceedings of the Third International Workshop on Vortex Flows and Related Numerical Methods, vol. 7, 1999, pp. 94–103.
[4] D.G. Dritschel, Contour surgery: a topological reconnection scheme for extended integrations using contour dynamics, J. Comput. Phys. 77 (1988) 240–266.
[5] D.G. Dritschel, M.H.P. Ambaum, A contour-advective semi-Lagrangian numerical algorithm for simulating fine-scale conservative dynamical fields, Quart. J. Royal Meteorol. Soc. 123 (1997) 1097–1130.
[6] D.G. Dritschel, Á. Viúdez, A balanced approach to modelling rotating stably-stratified geophysical flows, J. Fluid Mech. 488 (2003) 123–150.
[7] D.G. Dritschel, M.H.P. Ambaum, The diabatic contour advective semi-Lagrangian algorithm, Mon. Weather Rev. 134 (2006) 2503–2514.
[8] D.G. Dritschel, M.E. McIntyre, Multiple jets as PV staircases: the Phillips effect and the resilience of eddy-transport barriers, J. Atmos. Sci. 65 (3) (2008) 855–874.
[9] D.G. Dritschel, R.K. Scott, C. Macaskill, G.A. Gottwald, C.V. Tran, Unifying scaling theory for vortex dynamics in two-dimensional turbulence, Phys. Rev. Lett. 101 (2008) 094501.
[10] D.G. Dritschel, R.K. Scott, C. Macaskill, G.A. Gottwald, C.V. Tran, Late time evolution of unforced inviscid two-dimensional turbulence, J. Fluid Mech. 640 (2009) 217–235.
[11] D.G. Dritschel, R.K. Scott, On the simulation of nearly inviscid two-dimensional turbulence, J. Comput. Phys. 228 (2009) 2707–2711.
[12] D.G. Dritschel, J. Fontane, The HyperCASL Algorithm, in: D.G. Dritschel (Ed.), Turbulence in the Atmosphere and Oceans, Springer, 2010.
[13] J. Fontane, D.G. Dritschel, The HyperCASL algorithm: a new approach to the numerical simulation of geophysical flows, J. Comput. Phys. 228 (2009) 6411–6425.
[14] D.C. Handscomb, Spline representation of incompressible flow, IMA J. Numer. Anal. 4 (1984) 491–502.
[15] S. King, M. Carr, D.G. Dritschel, On the steady state form of large amplitude internal solitary waves, J. Fluid Mech., under review.

[16] A.R. Mohebalhojeh, D.G. Dritschel, The diabatic contour advective semi-Lagrangian algorithms for the spherical shallow water equations, Mon. Weather Rev. 137 (2009) 2979–2994.
[17] B.T. Nadiga, Orientation of eddy fluxes in geostrophic turbulence, Philos. Trans. Royal Soc. A 366 (2008) 2491–2510.
[18] R.K. Scott, Non-robustness of the two-dimensional turbulent inverse cascade, Phys. Rev. E 75 (2007) 046301.
[19] S.M. Tobias, P.H. Diamond, D.G. Dritschel, The formation of MHD jets on a magnetised β-plane at finite deformation radius, Geophys. Astrophys. Fluid Dynam., under review.
[20] N.J. Zabusky, M. Hughes, K.V. Roberts, Contour dynamics for the Euler equations in two dimensions, J. Comput. Phys. 30 (1979) 96–106.