Journal of Computational Physics 228 (2009) 6411–6425

Contents lists available at ScienceDirect

Journal of Computational Physics

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

The HyperCASL algorithm: A new approach to the numerical simulation of geophysical flows

Jérôme Fontane *, David G. Dritschel

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

ARTICLE INFO

Article history:
Received 19 January 2009
Received in revised form 14 May 2009
Accepted 14 May 2009
Available online 21 May 2009

Keywords:
Contour dynamics
Vortex methods
Contour advection
Geophysical flows

ABSTRACT

We describe a major extension to the Contour-Advective Semi-Lagrangian (CASL) algorithm [D.G. Dritschel, M.H.P. Ambaum, A contour-advective semi-Lagrangian numerical algorithm for simulating fine-scale conservative dynamical fields, Quart. J. Roy. Meteorol. Soc. 123 (1997) 1097–1130; D.G. Dritschel, M.H.P. Ambaum, The diabatic contour advective semi-Lagrangian algorithm, Mon. Weather Rev. 134 (9) (2006) 2503–2514]. The extension, called 'HyperCASL' (HCASL), uses Lagrangian advection of material potential vorticity contours like CASL, but a Vortex-In-Cell (VIC) method for the treatment of diabatic forcing or damping. In this way, HyperCASL is fully Lagrangian regarding advection. A grid is used as in CASL to deal with 'inversion' (computing the velocity field from the potential vorticity field).

First, the novel aspects of the algorithm are described including several improvements to the underlying CASL algorithm. All numerical parameters are chosen so as to minimise the computational cost while improving conservation properties. Finally, a thorough intercode comparison is conducted using a two-dimensional inviscid unforced turbulence test-case. This enables us to point out the advantages of this new algorithm in terms of resolution, computational cost and numerical diffusion compared to other existing methods, namely CASL, VIC and Pseudo-Spectral (PS) methods.

1. Introduction

Computational efficiency and the resolution of small-scale turbulent fluid motions are two conflicting goals in the numerical simulation of geophysical flows. This is a critical issue for many applications such as climate modelling, weather forecasting and tracking pollutants where precise predictions (particularly of concentration fluctuations) need to be obtained over large domains in a short period of time. Developed over the last decade, the CASL algorithm [6,7,9,10,14] is well-suited for many of these applications. The CASL algorithm is a hybrid contour dynamics method [24] designed specifically to achieve high resolution at reasonable computational costs. The use of an underlying ultra-fine 'contouring' grid together with the advection of contours enables one to resolve scales far below those associated with the main coarser grid (or inversion grid) where the velocity field is determined with standard Eulerian methods. This algorithm also maintains potential vorticity (PV) gradients much steeper than those the inversion grid would allow with standard Eulerian methods. Contour regularisation or 'surgery' [4,5] accounts for essential numerical dissipation. Errors arising from surgery are kept at a subgrid level by transferring them to a residual potential vorticity field which is evolved by standard spectral (PS) or grid point methods on the main grid and included in the total PV field for inversion [7].

Dritschel and Fontane [8] have recently announced a novel *fully Lagrangian* way of handling these regularisation errors in an extension of CASL. The HyperCASL (HCASL) algorithm combines Contour Dynamics and Vortex-In-Cell (VIC) methods [2].

* Corresponding author.
E-mail address: jerome@mcs.st-and.ac.uk (J. Fontane).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.05.025

Here the residual PV field is transferred to a set of point vortices whose circulation is dynamically adjusted to compensate for these errors. Not only does this new approach keep diffusion at a small level below the grid scale, but it also allows one to introduce non-conservative effects naturally using the point vortices in a similar way.

In this paper, the HyperCASL algorithm is presented in detail and compared to other existing methods. In Section 2, its structure is described and the values of all numerical parameters are determined by minimising computational cost while improving conservation properties. Using as a generic example the standard barotropic case of complex two-dimensional decaying turbulence, in Section 3 we compare the HyperCASL algorithm to three other methods, namely the CASL, VIC and PS methods. The comparison of the methods is done in two steps. First we examine how various known flow properties are captured by each algorithm. Then we turn to a comparison in terms of numerical properties (convergence, accuracy and cost). The paper concludes in Section 4 with a discussion of potential applications and further extensions.

2. The hyperCASL algorithm

The scalar PV field q is decomposed into its 'adiabatic' part q_a which is materially conserved and its 'diabatic' part q_d which changes on fluid particles in response of any kind of source term. This yields two evolution equations for PV:

$$\frac{\mathrm{D}q_a}{\mathrm{D}t} = 0; \quad \frac{\mathrm{D}q_d}{\mathrm{D}t} = S, \tag{1}$$

where $\mathrm{D}/\mathrm{D}t \equiv \partial/\partial t + \boldsymbol{u}\cdot\boldsymbol{\nabla}$ is the material derivative and $S(\boldsymbol{x},t)$ represents any source (forcing or damping) term. Note that combining these equations together gives the exact equation satisfied by PV, i.e. $\mathrm{D}q/\mathrm{D}t = S$. These two equations are in fact coupled since the velocity $\boldsymbol{u}$ is obtained from the inversion of the total PV $q = q_a + q_d$ (and in general other fields representing non-advective processes like gravity wave propagation, see e.g. [9].

As in the CASL algorithm [6], the adiabatic PV field q_a is represented by contours which are well-suited to resolve extraordinary details of the flow at a small fraction of the computational cost required by standard spectral methods [9,10]. In CASL, numerical dissipation is principally due to the regularisation of contours (surgery) and occurs well below the main (inversion) grid level.

However, the non-conservative PV field q_d is not easily treated by contour advection. This is why a spectral approach with explicit but weak numerical diffusion is used in CASL [7]. In the HyperCASL algorithm, the novelty is that the evolution of the diabatic PV is performed in a fully Lagrangian manner. We use an array of point vortices whose circulations are dynamically adjusted to exactly solve the equation $\mathrm{D}q_d/\mathrm{D}t = S$ on the inversion grid. Thus dissipation takes place at a subgrid level and there is no numerical diffusion all the way down to the inversion grid scale.

2.1. A test-case: two-dimensional inviscid unforced turbulence

In order to thoroughly analyse all numerical aspects of the HyperCASL algorithm, we need to consider a demanding test-case that exhibits generic features of geophysical flows. The extensively studied case of two-dimensional freely-decaying turbulence is well-suited for this purpose. Not only does it produce significant complexity like many geophysical flows via a direct enstrophy cascade associated with the strong filamentation of vorticity, but it also has conservation properties useful for assessing the overall numerical errors of the method [10,11,13,22]. This flow will be used as a benchmark throughout the paper to monitor the sensitivity to numerical parameters and to compare the HyperCASL algorithm with other methods. We use the standard quasi-geostrophic equations governing the motion of a mid-latitude shallow-water flow in near hydrostatic and geostrophic equilibrium (see e.g. [23]):

$$\frac{\mathrm{D}q}{\mathrm{D}t} = \frac{\psi - \psi_e}{\tau_{\mathrm{th}} L_D^2} - \frac{\zeta}{\tau_{\mathrm{ek}}}, \tag{2a}$$

$$(\boldsymbol{\nabla}^2 - L_D^{-2})\psi = q + q_{\mathrm{bot}} - \beta y; \quad \boldsymbol{u} = (u,v) = \left(-\frac{\partial\psi}{\partial y}, \frac{\partial\psi}{\partial x}\right). \tag{2b}$$

Here, ψ denotes the streamfunction and τ_{th} is the thermal damping time for the flow to relax to a prescribed thermal equilibrium streamfunction ψ_e. The Rossby deformation length is $L_D = \sqrt{gH}/f_0$ where g is the acceleration due to gravity, H is the mean fluid depth, and f_0 is the mean Coriolis frequency. $\zeta = \nabla^2\psi$ is the relative vertical vorticity, τ_{ek} is the Ekman damping time, q_{bot} is the PV associated with bottom topography at $z = b(\boldsymbol{x})(q_{\mathrm{bot}} = -fb/H)$, and β is the y-gradient of the Coriolis frequency. The flow is doubly-periodic in a domain of (scaled) dimensions $2\pi \times 2\pi$.

For the specific 2D turbulence case, the deformation radius is infinite, the Coriolis frequency is constant and neither damping nor topography are active[1], i.e. $L_D = \infty, \beta = 0, \tau_{\mathrm{th}} = \infty, \tau_{\mathrm{ek}} = \infty$ and $q_{\mathrm{bot}} = 0$. The simulation starts from a random-phased PV distribution with an energy spectrum of the form

$$E(k) = ck^{2p-3}e^{-(p-1)(k/k_0)^2}, \tag{3}$$

[1] An example of a thermally forced jet with active damping and topography can be found in [8].

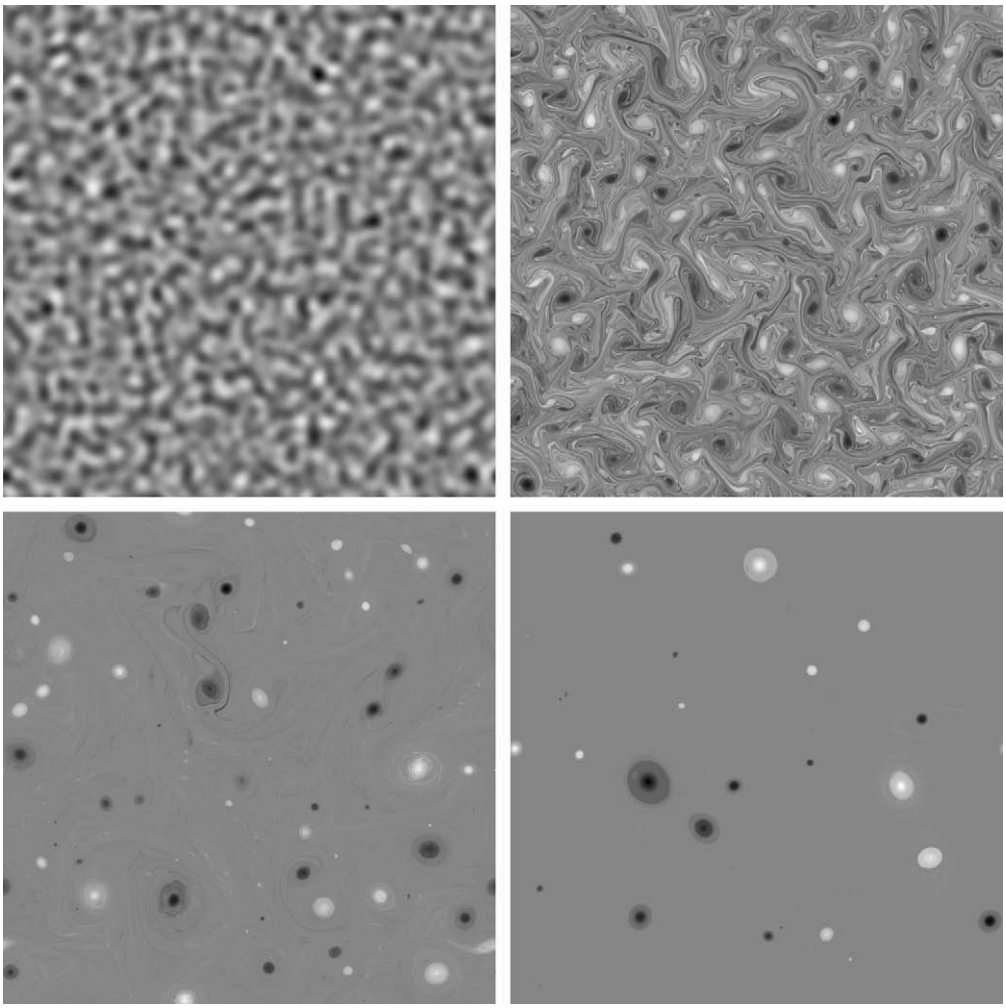


Fig. 1. The PV field q at times $t = 0, 10, 100$ and 500 from left to right and top to bottom. A linear grey scale is used with white being the highest level of PV value and black being the lowest. The maximum value of $|q|$ at these times is 12.97, 13.12, 13.15 and 11.36.

where k_0 is the enstrophy-energy centroid, p any integer greater than 1 and c a constant adjusted to set the maximum PV at some preset value. For the current calculations we use $k_0 = 16, p = 3$ and a maximum PV of 4π. Since this flow will be used repeatedly throughout the paper, we briefly present its evolution here from a simulation performed with the HyperCASL algorithm. The flow is computed on an inversion grid of dimensions 256×256 and we use an average of 2×2 point vortices per grid box. The PV jump is $\Delta q = 0.59$ across all contours (see Subsection 2.3). Fig. 1 exhibits the evolution of the PV field at four different times of the flow evolution. The initial random-phased PV field is structured on small scales and looks blurry as a consequence of the use of the random distribution (3) in spectral space. In time, the flow gets progressively organised into coherent vortices of both signs. They continuously interact and their number decreases through merging or filamentation processes [15,16] until there are only two vortices left (one of each sign) at very late times as found originally by Matthaeus et al. [18] and Montgomery et al. [19].

2.2. Transfer of a gridded field to point vortices

The algorithm used to convert a given source field[2] S into changes in the strengths of the point vortices was first described in detail in [8]. Here, we recall the main idea and then present a thorough investigation aimed to make the procedure to work most efficiently. The algorithm consists of an iterative process which inverts

[2] The source field can either be a prescribed damping/forcing field such as thermal damping, Ekman pumping or stochastic forcing, and includes the residual PV field coming from surgery or recontouring.

$$S_{ij} = \sum_{k \in G_{ij}} w_{i,j,k} \frac{\mathrm{d}\tilde{q}_k}{\mathrm{d}t}, \tag{4}$$

for $\mathrm{d}\tilde{q}_k/\mathrm{d}t$ (the change in the normalised strengths of the point vortices), where S_{ij} is the source term at grid point (i,j), G_{ij} is the domain consisting of the grid boxes[3] surrounding (i,j), and $w_{i,j,k}$ are the interpolation weights. The vortices lying within G_{ij} are found simply by integer arithmetic. The normalised circulations of the point vortices, $\tilde{q}_k = \Gamma_k \Delta x \Delta y$ with $\Delta x, \Delta y$ the grid lengths in the x and y directions, have units of PV. Note that removing the time derivative in the sum in (4) gives the diabatic PV field q_{dij} on the left hand side. The solution to (4) is sought as the interpolation of an unknown gridded field F divided by the local vortex density ρ_k obtained by two interpolations. Since the interpolation scheme used in the algorithm is bi-linear, only four grid points are necessary:

$$\rho_k = w_{i,j,k} D_{i,j} + w_{i+1,j,k} D_{i+1,j} + w_{i,j+1,k} D_{i,j+1} + w_{i+1,j+1,k} D_{i+1,j+1}, \tag{5}$$

where the bi-linear weights are explicitly given by

$$w_{i,j,k} = \left(1 - \frac{|x_k - \bar{x}_i|}{\Delta x}\right)\left(1 - \frac{|y_k - \bar{y}_j|}{\Delta y}\right), \tag{6}$$

(x_k, y_k) being the position of vortex k and $(\bar{x}_i, \bar{y}_j)$ the coordinates of grid point (i,j), and $D_{i,j}$ is the vortex density at each grid point, i.e.

$$D_{i,j} = \sum_{k \in G_{ij}} w_{i,j,k}. \tag{7}$$

The solution to (4) is obtained via the unknown field F using

$$\rho_k \frac{\mathrm{d}\tilde{q}_k}{\mathrm{d}t} = w_{i,j,k} F_{i,j} + w_{i+1,j,k} F_{i+1,j} + w_{i,j+1,k} F_{i,j+1} + w_{i+1,j+1,k} F_{i+1,j+1}. \tag{8}$$

Starting from a first guess for F, a residual matrix R is set equal to the source term S. Then, the right-hand side of (4) is subtracted from the residual R, thereby redefining it, and the right-hand side of (8) with $F = R$ divided by the local vortex density ρ_k is added to $\mathrm{D}\tilde{q}_k/\mathrm{d}t$. This procedure is carried out iteratively until the L_∞-norm of the residual is less than some preset precision ε. When S is the residual PV field coming from the regularisation of contours, the precision is based on the PV jump across contours Δq, i.e. $\varepsilon = 10^{-n}\Delta q$ with n a positive integer; otherwise the L_2-norm of the source field is used, i.e. $\varepsilon = 10^{-n}||S||_2$. We varied the exponent n in the tolerance parameter between 6 and 12. The use of $n = 12$ did not have a significant impact on the flow properties and solution accuracy compared to $n = 6$, whereas the efficiency of the algorithm was significantly reduced by the use of higher precision. Hence, we recommend choosing $n = 6$.

All the difficulty now lies in choosing a good first guess for $\mathrm{d}\tilde{q}_k/\mathrm{d}t$ so that the convergence of the procedure is obtained in a small number of iterations. Most of the time, we start from the known values at the previous time step, but when these values are not available or when the vortices are re-arranged (see Subsection 2.3), the first guess is obtained simply from (8) with $F = S$. We tested more refined types of initialisation but none of them sufficiently improved the first guess to significantly reduce the number of iterations. For instance, we tested a smoother initialisation built on a Laplacian-based expression of the form $F = \alpha S + \beta \Delta S + \gamma \Delta^2 S + \delta(\mathrm{d}^4 S/\mathrm{d}x^2\mathrm{d}y^2)$ where $\alpha, \beta, \gamma, \delta$ are constants obtained for a uniform distribution of vortices. In any case the crude starting guess used here does not prevent the method from converging rapidly as explained next.

We use $m \times m$ point vortices on average per grid box that are initially placed on a regular array. In time, they are displaced by the flow. This progressive disorganization of the array influences the convergence of the iterative procedure. This is examined by monitoring the number of iterations needed for convergence. For this purpose, we displace randomly the point vortices from their original regular position by a fraction ϕ of the grid length Δx. We take the source term S to be a random-phased field with a spectrum $\Omega(k) = Ak/(k^2 + k_0^2)^2$. Here, k_0 is the wavenumber centroid and A is a constant chosen to ensure $||S||_2 = 1$. For this analysis we used $k_0 = 5$. The corresponding source field is illustrated in Fig. 2 (left). This rough random field structured on a wide range of scales is well-suited to test the robustness of the iterative method as it gives one of the worst situations to be interpolated that one may find in practical situations. The first guess of the procedure is (8) with $F = S$. Fig. 2 (right) shows the dependance of the average number of iterations (from 10,000 calculations with differing random S) as a function of the fractional displacement ϕ for m = 2, 3 and 4. First, the number of iterations is smallest when the array is regular ($\phi = 0$). Convergence is obtained for only 29, 44, 52 iterations when m = 2, 3, 4, respectively, indicating that the iterative inversion of (4) is efficient for a regular array. Secondly, there is a noticeable increase in the number of iterations as ϕ increases, especially for m = 2. This could be detrimental for the efficiency of the HyperCASL algorithm. To avoid this, the point vortices must be re-arranged on a regular array often enough to prevent the cost of the iterative procedure from increasing beyond reasonable limits. The regularisation of the array of vortices is discussed in the next subsection together with the regularisation of contours.

[3] The number of grid boxes comprising the domain G_{ij} depends on the interpolation scheme chosen to transfer the gridded field to point vortices and vice versa. Here, we use a bi-linear interpolation so G_{ij} only consists of four grid boxes. But we also tested a third-order interpolation scheme, namely the third-order interpolation scheme M'_4 of Cottet et al. [3], using 16 points and in this case G_{ij} consists of 16 grid boxes. Third-order schemes are known to be more accurate and are commonly used in VIC methods (the M'_4 scheme is used with the VIC algorithm in Section 3). However, tests performed revealed that its use in HCASL produces more numerical errors for a higher computational cost. This is why the bilinear scheme is preferred here.

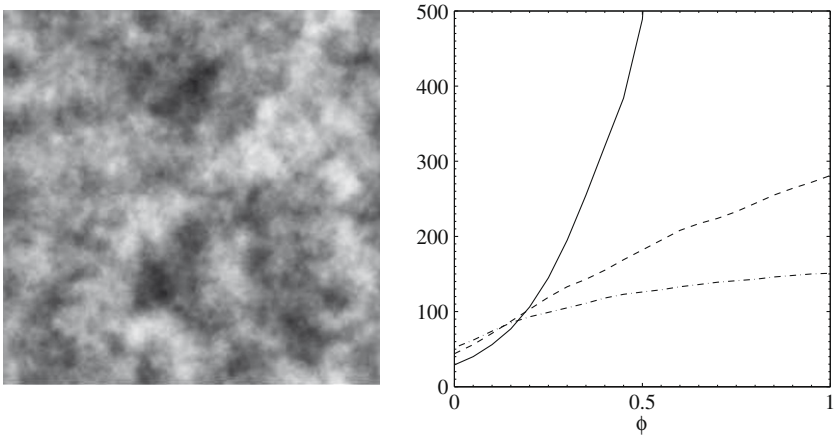


Fig. 2. Random-phased source field S (left) used for testing the convergence of the iterative procedure. Number of iterations (right) needed for convergence of the iterative procedure as a function of the initial displacement of the point vortices with $m = 2$ (solid), $m = 3$ (dashed) and $m = 4$ (dash-dotted). The displacement ϕ is in units of a grid length.

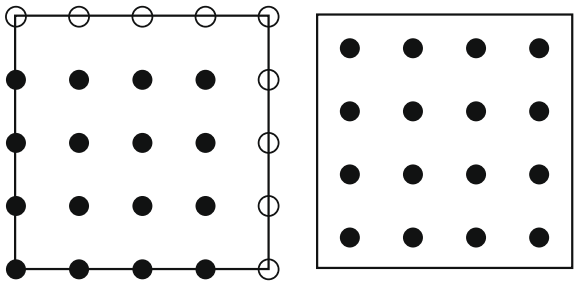


Fig. 3. Two distinct point vortex placements within a grid box for $m = 4$. The black square denotes the grid box and the filled circles correspond to point vortices within the grid box while the unfilled ones represent periodic point vortices belonging to the neighbouring grid boxes.

The last point to be discussed concerns the placement of the regular array within the grid box. Fig. 3 shows two distinct placements of the point vortices within a grid box: the left one involves placing a point vortex at each grid point, while the right one centres the array such that no point vortex lies at a grid point. Both placements where used in the algorithm and the centred placement (right) was found to be detrimental for two reasons. First, convergence is much slower: it requires 116 iterations for the case presented above with $m = 2$ while only 29 iterations are needed for the off-centred placement on the left. Second, using the centred array generates a progressive increase in the rms value of point vortex circulations, yielding a rapid blow-up of the simulation. This occurs because the inversion of (4) with the centred array allows the combined increase in the strengths of point vortices equally distant from a grid point. A variation in their circulation is initially transparent in the inversion process as their contribution at the grid points cancel each other and the source field is correctly reproduced. But as time advances, point vortices are displaced and the initial cancellation no longer holds, leading to a non-physical gridded PV field. This erroneous behaviour is repeated every time the array is regularised, and the accumulation of these errors becomes untenable for the numerical procedure. This erroneous behaviour is illustrated in Fig. 4 (left) with the temporal evolution of the L_2-norm of the difference in circulation between two consecutive point vortices along the x direction. While this difference remains at a low level with the off-centred placement, it rapidly increases with the centred array and prevents the simulation from continuing beyond $t = 7$. This numerical divergence is characterised by the appearance of noise in the PV field at the scale of the point vortices, leading to a grainy aspect of the flow as displayed in Fig. 4 (right). The use of the off-centred array with one point vortex at each grid point prevents this erroneous behaviour. This is now standard in the HyperCASL algorithm.

2.3. Numerical settings

All other numerical parameters have been chosen as a compromise between accuracy and efficiency. These differ from their original settings in the CASL algorithm [6,7]. An exhaustive and careful investigation based on more than one hundred calculations has been conducted over six months to set all these values. Only a brief summary is provided here.[4]

[4] Full details of the tests carried out to justify these parameter choices may be found on the web at www-vortex.mcs.st-and.ac.uk/HyperCASL.pdf.

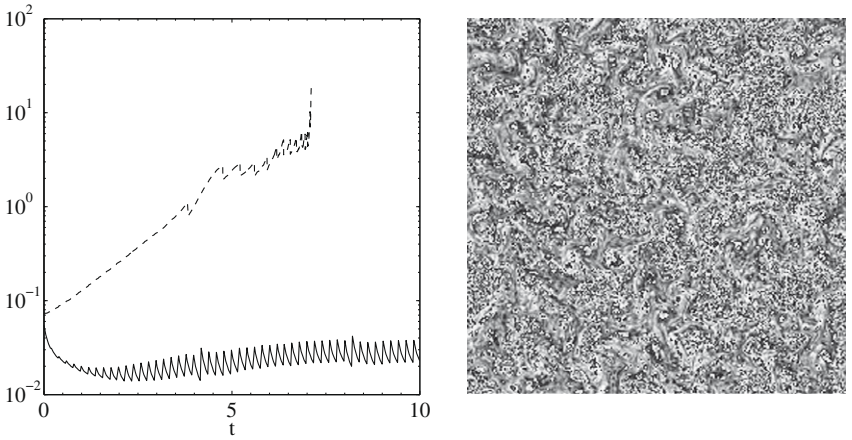


Fig. 4. Evolution of the rms value of the circulation difference between two consecutive point vortices along the x direction (left) with an off-centred (solid line) and a centred (dashed line) placement of the array within the grid box. Gridded PV field at $t = 7$ (right) obtained for the simulation using the centred placement.

We first discuss regularisation as it affects both the precision and the efficiency of the method by working on contours and point vortices. Surgery [4,5] is a necessary regularisation process of the contours needed to control the exponential build-up of fine-scale PV. It is performed when the distance between two contours of the same PV level becomes closer than the surgical scale, $\delta = \frac{1}{4}\mu^2 L$ where $l_{\rm sep} = \mu L$ corresponds to the maximum node separation, μ is the dimensionless node spacing and L is a characteristic large-scale length of the PV structures in the flow. As a compromise between numerical efficiency and accuracy, we choose $\delta = \Delta x/16$ and $l_{\rm sep} = 1.25\Delta x$, which gives a dimensionless node spacing $\mu = 0.2$. Surgery however has a non-negligible computational cost and its application must be kept to a minimum. Contour regularisation is required just before contours get strongly deformed by the flow, so we measured the stretching/compression of adjacent nodes at each time step to determine the adequate period of time between two consecutive applications of surgery. This resulted in a new generic criterion : surgery is performed when the "twist parameter" $\tau_{\rm con} \approx 2.5$, where $\tau_{\rm con}$ is the time integral of the maximum relative vorticity $|\zeta|_{\rm max}$ between the last surgery time t_0 and the current time t, i.e. $\tau_{\rm con} = \int_{t_0}^{t} |\zeta|_{\rm max} {\rm d}t$.

By modifying the contours, surgery also changes the gridded PV field. The PV difference before and after surgery or "PV residual" is transferred to the point vortices by adjusting their circulations so that the gridded PV *is not altered by surgery*. In CASL, a recontouring process transferring the residual or diabatic PV field q_d to the contours is performed regularly to prevent excessive diffusion associated with the use of spectral methods [7]. Here, there is no diffusion at the grid level since a vortex-in-cell method is used to uptake the PV residual. Nonetheless, surgery is replaced by recontouring every twentieth regularisation so as to transfer the PV associated with the point vortices to the contours. This also keeps contours from becoming tangled or from crossing [21].

New contours are built on an ultra-fine "recontouring" grid 16 times finer than the inversion grid[5] from the PV field obtained from merging the contour and vortex PV (q_a and q_d). The use of a much finer grid is motivated by a significant improvement in conservation properties. Here again, the recontouring process induces differences in the gridded PV field associated with the new contours, and the residual PV is taken up by an array of point vortices to ensure no change in the PV on the inversion grid.

As stated in the previous subsection, vortices need to be re-arranged every so often on a regular array in order to reduce the computational cost. Every time the point vortices are regularised, their circulations are modified to ensure that this operation has no effect on the PV induced at each grid point. For consistency, a similar parameter $\tau_{\rm vor}$ based on the time integral of the relative vorticity is used. The optimal value is here $\tau_{\rm vor} \lesssim 1$ since it maximises the numerical efficiency while preventing erroneous energy increase due to numerical errors.

The PV jump Δq is the last numerical parameter to be set. Its value has to be chosen according to the topology of the flow. As done in CASL [6,9], the PV jump is computed from a fixed number of intervals n_q and the extremum values of all non-zero PV fields available at the beginning of the simulation (*i.e.* the initial PV field, the thermal equilibrium PV field and the PV field associated with bottom topography if they are present):

$$\Delta q = \frac{q_{\rm max} - q_{\rm min}}{n_q}. \tag{9}$$

If Δq is found to be zero or greater than the maximum PV jump $\Delta q_{max} = 2q_{\rm eddy}/n_q$ based on a characteristic eddy PV $q_{\rm eddy}$, we set $\Delta q = \Delta q_{max}$. Here, the number of contours n_q is kept constant for two reasons. First, the spatial resolution associated with

[5] The ultra-fine grid has a grid length equal to the surgical scale, i.e. $\delta = \Delta x/16$.

Table 1
Numerical parameters used in the HyperCASL algorithm together with their default values.

Parameter	Default value
Inversion grid resolution	$N = 256$
Inversion grid length	$\Delta x = \Delta y = 2\pi/N$
Ultra-fine grid resolution	$N_f = 16N$
Point vortices per grid box	$m \times m = 2 \times 2$
Number of contour intervals	$n_q = 40$
Maximum PV jump allowed	$\Delta q_{max} = q_{\text{eddy}}/n_q$
PV jump across all contours	$\Delta q = \min(\Delta q_{\max}, (q_{\max} - q_{\min})/n_q)$
Surgical scale	$\delta = \frac{1}{4}\mu^2 L = \Delta x/16$
Maximum node separation	$l_{\text{sep}} = \mu L = 1.25\Delta x$
Dimensionless node spacing	$\mu = 4\delta/l_{\text{sep}} = 0.2$
Large-scale length	$L = l_{\text{sep}}^2/4\delta = 6.25\Delta x = 100\delta$

contours is *already* a function of the inversion grid resolution N since both the surgical scale and the node separation are proportional to the inversion grid length $\Delta x = 2\pi/N$, *i.e.* $\delta = \Delta x/16$ and $l_{\text{sep}} = 1.25\Delta x$. Second, the mean length of the contours increases with N so that steep PV gradients are still well resolved when resolution increases. This result together with the node separation being proportional to the grid length gives a total number of nodes and associated CPU costs proportional to N^2. This is consistent numerically as the contour costs now balance the grid costs (i.e. FFT and other operations performed at the inversion grid level). So a constant value for n_q is adequate both in terms of accuracy and efficiency. After many tests, the value $n_q = 40$ has been found to be a good compromise between accuracy and cost and is now recommended as a default value. As a matter of course this choice is not immutable; one can either reduce it for a quick numerical test or increase it to obtain higher accuracy simulations as done by Dritschel et al. [11,12] using CASL with $n_q = 50$.

We summarise in Table 1 all numerical parameters of the HyperCASL algorithm with their default values. Note that the grid resolution N entirely controls the node resolution on contours, and that Δq is in general fully determined by the initial PV field. The characteristic eddy PV q_{eddy} must be specified by the user but it is only used if a suitable PV jump is not available from the initial, thermal equilibrium, or topographic PV distributions. All numerical parameters have now been carefully examined and adjusted to values that lead, approximately, to minimal numerical errors and maximal computational efficiency. We can now turn to a comparative study of the numerical algorithms for simulating these geophysical flows (i.e. HCASL, CASL, VIC and PS).

3. Inter-code comparison

In this section, we compare simulations of the 2D turbulence test-case using four different methods HCASL, CASL, PS and VIC. A detailed list of all the simulations performed is presented in Table 2, giving the algorithm type, the inversion grid resolution N, the characteristic wavenumber k_0 of the initial random PV field, the number of runs and the average cpu cost per unit of time. Unless stated, the results presented in this section are obtained by ensemble averaging data over sets of ten simulations similar to the one presented in Section 2.1 (differing only in the initial random PV field – first four lines of Table 2). The comparison of the methods is done in two steps. First we examine how various known flow properties are captured by each algorithm. Then we turn to a comparison in terms of numerical properties: convergence, accuracy and cost.

3.1. Comparison based on flow properties

Fig. 5 presents the PV evolution within a sixteenth of the domain in one of the ten simulations for each algorithm. All simulations were carried out until time $t = 500$, but here we do not show the PV field corresponding to the final time. At that time, few vortices remain in the domain and none are present in the subdomain selected for the illustration, leading to mostly uniformly grey images (at least for one of the algorithms). We show images at time $t = 100$ instead, when vortical activity is still roughly equally distributed throughout the domain. Although identical at the initial time, the four calculations become progressively different in time. In the initial stage of the flow evolution ($0 \leqslant t \lesssim 10$), the dynamics are not yet dominated by coherent vortices and the four simulations present similar patterns as clearly shown in the second row of images corresponding to $t = 5$. Beyond that stage, coherent structures which spontaneously emerge from the random initial PV field drive the flow dynamics. The simulations for the different algorithms are no longer identical and none of the PV fields resemble each other by $t = 20$. This point is crucial if one wants to compare properly different numerical methods: comparisons based on flow fields are only relevant for early times because they become too uncorrelated and hardly comparable beyond that early stage. For this reason we next examine other methods of comparison. Nonetheless one thing is obvious in the evolution of the PV fields: one can observe a great difference in the spatial resolution between the two first and the two last columns. While tiny filaments of vorticity field are observed in the images of the two first columns, they are not present in the images of the last two columns. The lower spatial resolution is also discernible by the pixelisation of these images. The presence of extraordinary detail in both HCASL and CASL simulations comes from the use of a lengthscale for contours

Table 2
Detailed list of the simulations performed for the inter-code comparison. For each case, we give the algorithm type, the inversion grid resolution N, the characteristic wavenumber k_0 of the initial random PV field, the number of runs and the ensemble-averaged cpu cost per unit of time (given in cpu seconds for an Intel 2.4 Ghz processor).

Algorithm	N	k_0	Runs	cpu cost per unit of time (s)
HCASL	256	16	10	38.3
CASL	256	16	10	32.1
PS	256	16	10	3.8
VIC	256	16	10	11.9
PS	4096	16	1	5178
HCASL	32	4	1	0.36
CASL	32	4	1	0.2
PS	32	4	1	0.01
VIC	32	4	1	0.08
HCASL	64	4	1	1.3
CASL	64	4	1	1.1
PS	64	4	1	0.1
VIC	64	4	1	0.7
HCASL	128	4	1	6.1
CASL	128	4	1	6.3
PS	128	4	1	1.1
VIC	128	4	1	4.2
HCASL	256	4	1	35
CASL	256	4	1	54.5
PS	256	4	1	11.6
VIC	256	4	1	18.2
HCASL	512	4	1	182.4
CASL	512	4	1	610.2
PS	512	4	1	163.3
VIC	512	4	1	89.1
PS	1024	4	1	1273.7
VIC	1024	4	1	257.5

16 times finer than the inversion grid scale used for the PS and VIC algorithms. Note that most of the data presented here for HCASL and CASL are post-processed (including imaging of the PV field) on the recontouring grid.

The evolution of energy E for all algorithms is displayed in Fig. 6(a). Varying the initial random-phased PV field induces variations in the initial energy. Thus energy has been normalised by its initial value E_0 so that all data can be compared and averaged. VIC, and to a lesser extent PS, is better at conserving energy though they clearly exhibit strongly diffusive behaviour in Fig. 5. In enstrophy (and in higher moments), the situation is reversed, favouring HCASL, see Fig. 6(b). The energy variations in HCASL and CASL arise from small random inaccuracies within the contour to grid conversion step. This generates a kind of Brownian motion which only appears to affect the energy. These small-scale energy changes prevent the methods from giving a reproducible evolution for energy. Nonetheless, the differences in energy conservation between algorithms remain small and in the first stage of the flow evolution, when all algorithms produce comparable vorticity fields (i.e. $t \lesssim 10$) energy conservation obtained with HCASL and CASL is as good as that obtained with VIC. The enstrophy by contrast is not sensitive to these inaccuracies.

Now, if we focus only on HCASL and CASL, one can see that HCASL produces better results. Not only does energy decrease slower at initial times (see Fig. 6(a)) but it also does not show an unphysical growth at late times like in CASL. The enstrophy decrease is also in favour of HCASL, particularly at late times when HCASL gives the best conservation overall (see Fig. 6(b)).

A suitable way of measuring the numerical diffusion associated with the methods is to consider the diffusion coefficient of vorticity ν_{vort}. As previously argued, numerical models always introduce diffusion in inviscid flows. As a result enstrophy is decaying with time at a rate that can be estimated from the enstrophy equation for a viscous fluid:

$$\frac{\mathrm{D}Z}{\mathrm{D}t} = -2\nu_{\text{vort}}P, \tag{10}$$

where $P = \frac{1}{2}\langle|\nabla q|^2\rangle$ is the palinstrophy. The evolution of P thus indicates the rate at which vorticity gradients (and thereby vortex filaments) are numerically diffused. From its evolution plotted in Fig. 7(a), one can see that its early growth is closely similar for the four algorithms but later P decreases faster for PS and VIC compared to HCASL and CASL. The latter point illustrates the greater ability of HCASL and CASL to retain steep vorticity gradients. The corresponding diffusion coefficient of vorticity is displayed in Fig. 7(b). As a consequence of the better conservation of palinstrophy, ν_{vort} is smaller for HCASL and CASL past the palinstrophy peak. It should be mentioned that both P and ν_{vort} have been computed on the inversion grid for HCASL and CASL. When the finer recontouring grid is used, one obtains much smaller (two orders of magnitude lower) values of ν_{vort}, see the two lower curves in Fig. 7(b). This clearly shows that both HCASL and CASL present a much lower numerical diffusion than the PS and VIC methods.

We now examine energy $E(k)$ and enstrophy $Z(k)$ spectra in Fig. 8 to see to what extent the algorithms are able to describe the transfers between wavenumbers. Due to the difference of the effective resolution between HCASL and CASL on the one

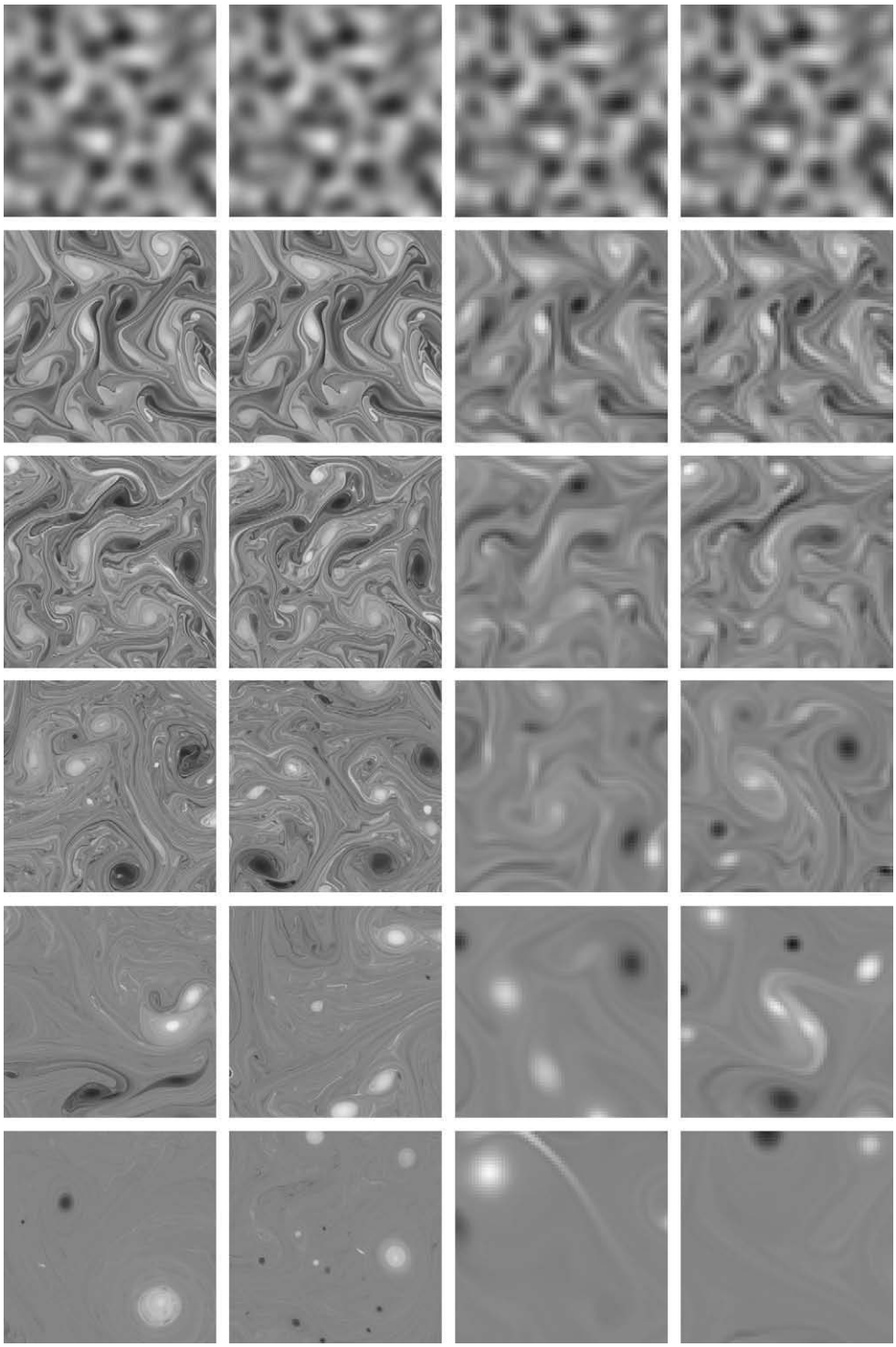


Fig. 5. The PV field q at times $t = 0, 5, 10, 20, 50$ and 100 from top to bottom with HCASL (first column), CASL (second column), PS (third column) and VIC (fourth column). Only a sixteenth of the domain is represented. A linear grey scale is used with white being the highest level of PV and black being the lowest.

hand and PS and VIC on the other, the range of resolved wavenumbers is much larger for HCASL and CASL ($k_{\max} = 2048$) than for PS and VIC ($k_{\max} = 128$). Note the presence of a bump peaking around $k = 500$ for the initial enstrophy spectrum produced by both HCASL and CASL. This bump is a numerical artefact of the postprocessing coming from the interpolation of

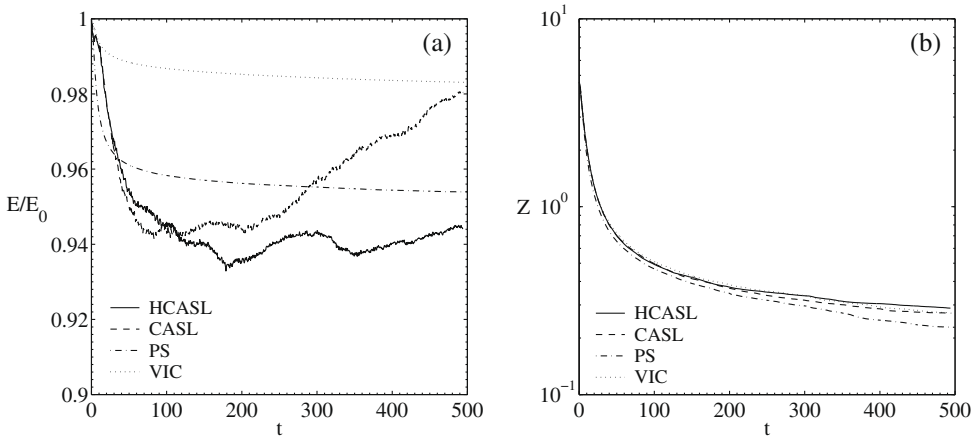


Fig. 6. Compared evolution of the ensemble-averaged (a) energy and (b) enstrophy for the four algorithms on the entire simulation period $t \in [0, 500]$.

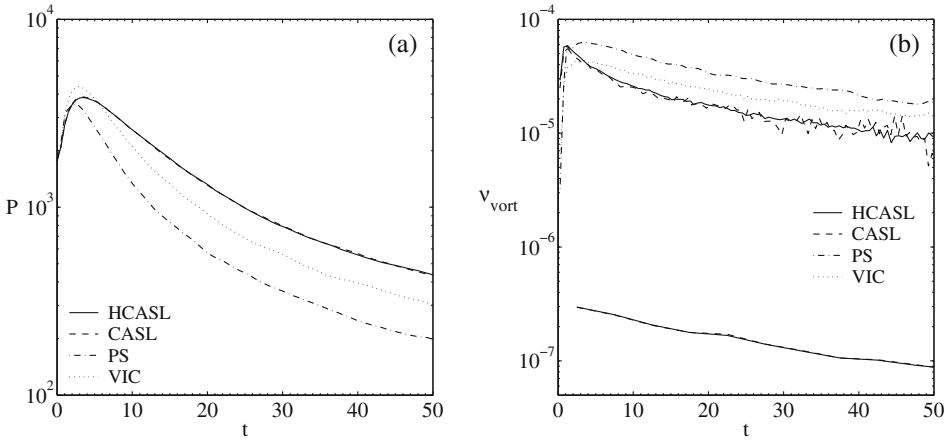
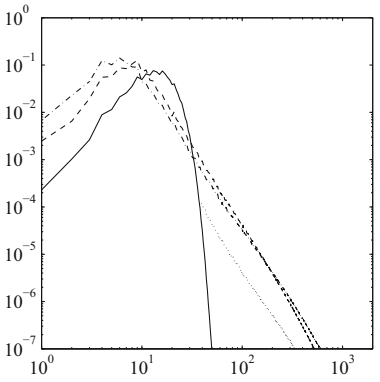


Fig. 7. Early evolution of (a) the palinstrophy P and (b) the vorticity diffusion coefficient ν_{vort}. In (b) ν_{vort} has been computed on the inversion grid (upper curves) for the four algorithms, and on the recontouring grid (lower curves) for HCASL and CASL.

the initial random PV field which was defined on the inversion grid to the recontouring grid. It affects only the initial time and the spectra at later times are free of such erroneous behaviour at large k. This bump is also present for the initial energy spectrum but it does not appear on the figure within the range of scales chosen for the plots.

The upscale energy and the downscale enstrophy transfers are characterised in spectral space by the existence of an inertial range which scales as a power law, i.e. $E(k) \sim k^{-3}$ and $Z(k) \sim k^{-1}$ as theoretically demonstrated by [1,17]. These theoretical scalings are shown in Fig. 8. While HCASL and CASL correctly capture the inertial range, PS and VIC are clearly unable to do so. The theoretical scalings were derived using a local spectral approach and are only valid when the turbulence is fully developed [11,12]. Indeed, the Batchelor–Kraichnan model does not take into account the effect of coherent vortices which are local in physical space and therefore nonlocal in spectral space. The turbulence is fully developed when the palinstrophy reaches its maximum [13], i.e. $t \approx 3$ as can be seen in Fig. 7(a), and the influence of the coherent vortices on the flow is significant from $t \approx 10$. Thus the theoretical scalings are here strictly valid only for $t = 5$ and $t = 10$, as can be seen most clearly in the enstrophy spectra.

Finally, we consider how the population of coherent vortices is represented by each algorithm. Based on a self-similar distribution of vortices of different areas A, Dritschel et al. [11] showed theoretically that the vortex number density $n(A)$ scales as $n(A) \sim A^{-1}$. We computed $n(A)$ as a function of the vortex area A for each algorithm and the results are displayed in Fig. 9. The vortex number density $n(A)$, normalised by the total number of vortices $N_v = \int n(A)dA$, is obtained by a temporal averaging over two distinct intervals, namely $t \in [10, 100]$ and $t \in [100, 500]$. Clearly, both PS and VIC fail to represent the correct vortex distribution. Due to the low resolution used, vortices of size tinier than 0.02 are not present in the flow and the theoretical scaling law is poorly matched for $t \in [10, 100]$ and not at all for $t \in [100, 500]$. Conversely, for both HCASL and CASL vortices with areas as small as 10^{-5} are present in the flow due to the much higher resolution available. While the measured vortex number density closely fits the theoretical scaling law on $t \in [100, 500]$, the measured slope is a little steeper



than A^{-1} on the interval $t \in [10, 100]$. This difference comes from the fact that for early times the distribution is not completely self-similar in size. Indeed, large vortices are appearing only at late times as a result of successive mergers of smaller

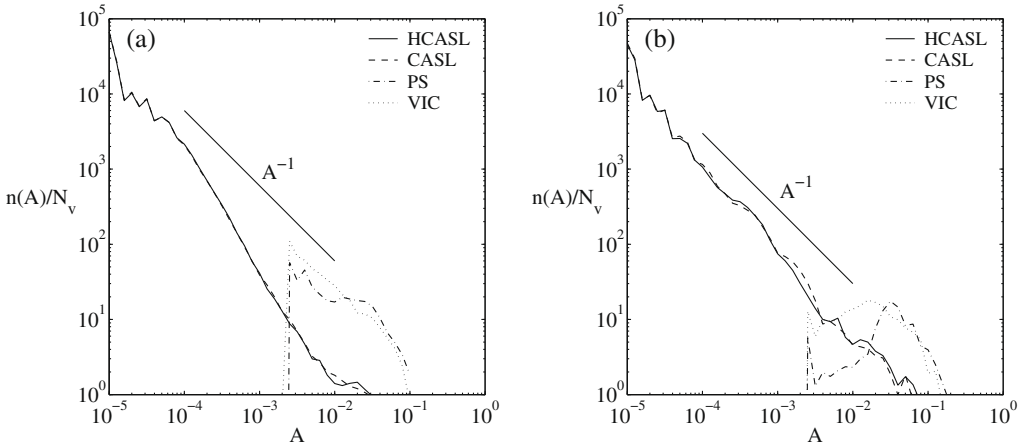


Fig. 9. Time-averaged vortex number density $n(A)$ normalised by the total number of vortices N_v versus area A for the four algorithms. Curves correspond to data collected on the period (a) $t \in [10, 100]$, and (b) $t \in [100, 500]$.

vortices. Comparison of Fig. 9(a) and (b) shows that the probability of finding large vortices increases in time. Furthermore, the largest vortex size is $A_{\max} = 0.03$ by $t = 100$ and grows to $A_{\max} = 0.07$ by $t = 500$.

The above study of various properties of decaying 2D turbulence clearly shows that, for a given resolution, both HCASL and CASL are superior to PS and VIC methods in capturing the physical features of this flow. In the next section, we perform a comparison in terms of numerical properties, i.e. convergence, accuracy and cost of the different algorithms.

3.2. Comparison based on numerical properties

In computations of practical interest, only modest resolutions are used because a large portion of the computational resources must be dedicated to other physical processes, e.g. chemical reactions or radiation schemes. As a result, an efficient numerical algorithm, or "dynamical core" in atmospheric models, is designed to produce the best flow representation for a given resolution, or equivalently, to result in the fastest convergence as the resolution is increased. We present a direct, visual comparison of the convergence rate of each method in Fig. 10 showing PV contours in a sixteenth of the domain for increasing resolution, i.e. $N = 32,\ 64, 128, 256$ and 512 from top to bottom. The corresponding simulations (rows 5–24 of Table 2) start from an initial random PV field structured on larger scales, i.e. smaller k_0, so that the flow can be properly represented even for the smallest resolution used, $N = 32$. The fields are plotted at $t = 5$ when all simulations are still comparable as discussed in the previous section. While essential features of the flow are already captured by HCASL and CASL at $N = 64$, convergence seems to be reached only at $N = 512$ for the PS and VIC methods. The ripples observed in the upper part of the PV field in the $N = 512$ PS simulation are a sign that convergence is not totally achieved. The key difference when comparing the four images across a row is the steepness of the PV gradients. The PV fields obtained with HCASL and CASL at resolution $N = 64$ show PV gradients as steep as those produced by the PS and VIC methods at resolution $N = 512$. From these figures it is obvious that both algorithms based on contour advection converge much faster than PS and VIC methods.

An appropriate way of assessing the accuracy of the methods consists of examining their ability to conserve area between iso-levels of potential vorticity (this is a direct consequence of Kelvin's circulation theorem in the absence of any source field). The area error is computed in a similar way to that in [9], who computed the mass error for a perturbed unstable zonal jet. The initial PV field is divided into regions $\mathcal{R}_j$, with integer $j \in [-M, M]$, each corresponding to a potential vorticity level q_j. Ideally, the area A_j of each region does not change in time, but in practice numerical errors inevitably lead to changes in A_j. The PV increment $\delta q = q_{j+1} - q_j$ between two consecutive levels is here taken to be $\delta q = q_{\text{eddy}}/20 = 4\pi/20$. The number of regions is computed automatically from the extremum values of the PV field, i.e. $M = (q_{\max} - q_{\min})/2\delta q$. For each level, the total area A_j occupied by PV level q_j is computed on the recontouring grid to improve the accuracy of the calculation. Thus, the PV field q must be first interpolated onto a 16 times finer grid (in each direction) for the PS and VIC simulations. For each grid point the region $\mathcal{R}_j$ to which q belongs is determined by finding the integer j for which $(j - \frac{1}{2})\delta q < q \leqslant (j + \frac{1}{2})\delta q$. The area error is then defined as the rms difference between $A_j(t)$ and $A_j(0)$ normalised by the initial area, that is:

$$\varepsilon_A(t) = \sqrt{\frac{\sum_{j=-M}^{M} \big(A_j(t) - A_j(0)\big)^2}{\sum_{j=-M}^{M} A_j^2(0)}}. \tag{11}$$

Fig. 11 displays the evolution of the area error over the time interval $t \in [0, 10]$ for the four algorithms. Both HCASL and CASL (note that the corresponding curves are indistinguishable on the figure) clearly provide a better conservation of the area. At $t = 5$, $\varepsilon_A(t)$ is about 10 times greater for PS and VIC (0.33 and 0.29, respectively) than for HCASL and CASL, which only

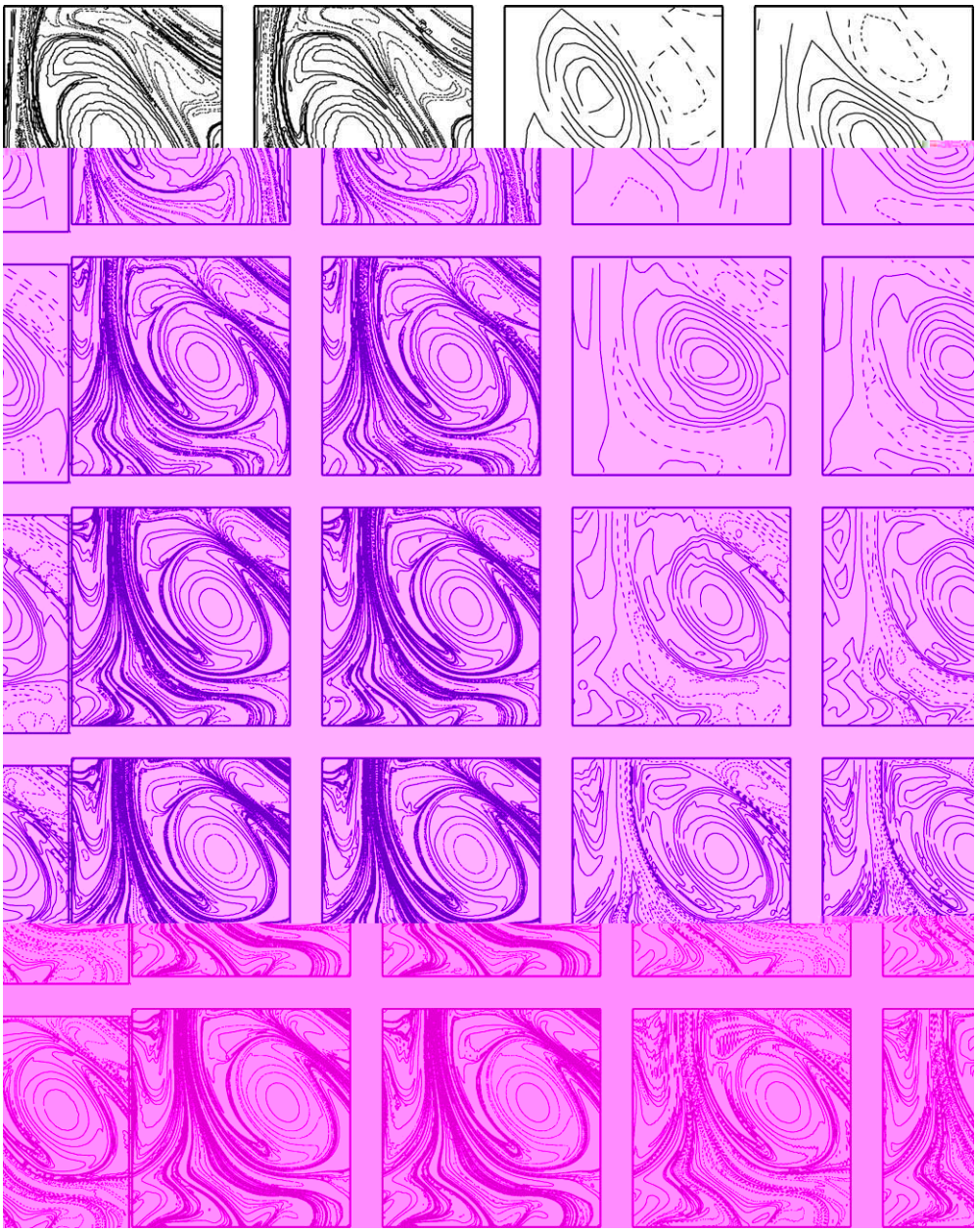


Fig. 10. Contour representation of the PV field q at $t = 5$ in one sixteenth of the domain for increasing resolution from top to bottom, i.e. $N = 32, 64, 128, 256$ and 512, with HCASL (first column), CASL (second column), PS (third column) and VIC (fourth column). The contour interval used is $\Delta q = 4\pi/10 = 1.2566$ and dashed contours correspond to negative values.

produce an error of 3% in area conservation. By $t = 10$, the error difference between methods is reduced but is still large, i.e. $\varepsilon_A(t)$ is three times greater for PS and VIC than for HCASL and CASL. These results clearly favour both HCASL and CASL.

We now consider the efficiency of the algorithms. It can be measured by the cpu cost per unit of time as given in Table 2 for each of the simulations carried out. Focusing on the series of 10 runs (four first rows), spectral methods give the best performance and VIC, CASL and HCASL are, respectively 3.2, 8.6 and 10.2 times slower. If we consider only these computational costs, one may be tempted to use the PS method since it is much faster than the other methods. But one must keep in mind the associated spatial resolution when comparing timings. For a fair comparison, a simulation using spectral methods has been carried out on a grid as fine as the recontouring grid, i.e. $N = 4096$. The timing obtained now strongly disfavours the PS method since the entire simulation took almost 24 days which is 135 times the duration of the simulation using HCASL. A comparison of the efficiency based on these timings only, however, does not take into account the advantages of Lagrangian methods over Eulerian ones. Indeed for these cases (lines 1–4 of Table 2), the constraint on the time step is not limited by the CFL condition in CASL and PS due to a flow structured on small scales ($k_0 = 16$) and a spatial resolution not high enough

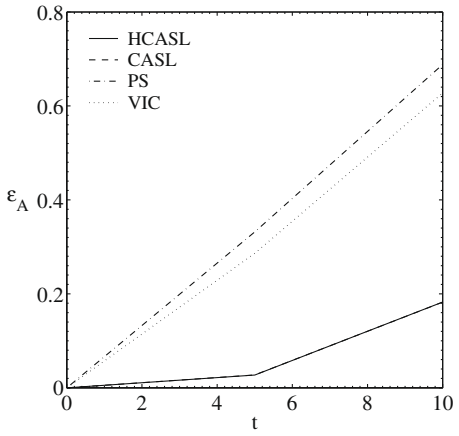


Fig. 11. Early times evolution of the area error ε_A for the four algorithms.

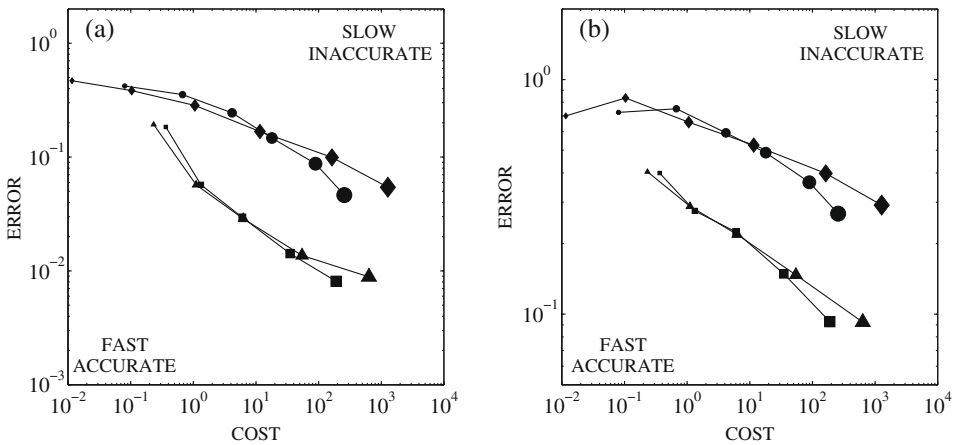


Fig. 12. The area error ε_A at (a) $t = 5$ and (b) $t = 10$ vs the cost (in cpu seconds) per unit of time for all algorithms. The symbols indicate different methods: squares for HCASL, triangles for CASL, diamonds for PS and circles for VIC. Their size increases with resolution, from $N = 32$ to $N = 512$ for HCASL and CASL and from $N = 32$ to $N = 1024$ for PS and VIC.

($N = 256$). For a flow structured on larger scales ($k_0 = 4$), increasing the resolution disfavours Eulerian methods because the CFL condition limits the time step. While CASL (resp. PS) is faster than HCASL (resp. VIC) for $N = 32$, the comparison is reversed in favour of HCASL (respectively, VIC) for $N = 512$.

As everyone would agree, efficiency and accuracy of the algorithms must not be compared separately but together because the best method is not simply the most accurate or the fastest but the one combining precision and efficiency. To this end, we plot in Fig. 12 the area error ε_A at $t = 5$ and $t = 10$ as a function of the cost per unit of time for the four algorithms. Each symbol represents an algorithm: squares for HCASL, triangles for CASL, diamonds for PS and circles for VIC. The size of the symbols indicates the resolution used: from $N = 32$ to $N = 512$ for HCASL and CASL and from $N = 32$ to $N = 1024$ for PS and VIC. From this figure it is readily apparent that both methods using contours advection outperform PS and VIC methods. A spatial resolution as high as $N = 1024$ is needed for PS and VIC to be at least as accurate as HCASL and CASL for a resolution of $N = 64$. And reaching this precision takes 205 (respectively, 42) more cpu time for PS (resp. VIC) than required by HCASL and CASL. A comparison of the latter two is also interesting. If both give roughly the same level of accuracy for a given resolution, HCASL becomes faster than CASL when N increases due to the CFL limitation on the time step in the Eulerian part of CASL. The same remark holds for the direct comparison of PS and VIC although VIC is always a little more accurate than PS.

Based on numerical arguments, the two algorithms based on contour advection strongly outperform both PS and VIC methods, and HCASL outperforms CASL at high resolutions due to its fully Lagrangian nature.

4. Conclusions

We have presented the HyperCASL algorithm, a new fully Lagrangian method for the simulation of geophysical flows combining Vortex-In-Cell and Contour Dynamic methods. The potential vorticity, the fundamental dynamically active tracer,

is split into two parts. As in CASL the adiabatic part is represented by contours, while the diabatic part (changing on fluid particles in response of any physical or numerical source term) is handled by an array of point vortices whose circulations are adjusted dynamically so as to represent exactly the effect of the forcing at the inversion grid level. This novel technique ensures that the system is non-dissipative down the inversion grid length. The numerical errors act only on scales that are smaller that the inversion grid length. This results in accurate computations with much coarser grids than standard Pseudo-Spectral or Vortex-In-Cell methods would need to employ to give comparable accuracy.

A thorough investigation of all aspects of the numerical method enabled us to set the values of the numerical parameters so as to minimise the computational cost while improving conservation properties. Then using the 2D decaying turbulence test-case, a comparison with three other algorithms showed the advantages of HCASL over CASL, PS and VIC methods. Focusing on the description of the flow's physical properties, HCASL results are similar to those of CASL while outperforming PS and VIC methods. Considering numerical aspects, HCASL was shown to converge faster than PS and VIC, and for a given spatial resolution, HCASL was found to be more accurate. Conversely, the computational cost required for HCASL to reach a preset precision for the solution is much less than that needed by PS and VIC methods. Compared to CASL, HCASL gives roughly the same accuracy and the same convergence rate but HCASL is more efficient at high resolutions owing to its fully Lagrangian nature: there is no CFL constraint on the time step.

In the light of these results, the HyperCASL algorithm represents a new opportunity for modelling complex geophysical flows efficiently. Extensions to more complete equation sets are straightforward [14], and for these an accurate treatment of PV is also vital for an accurate representation of the complete dynamics. There are also some future promising opportunities such as treating spherical geometry (the atmosphere) or complex domains (the oceans). HyperCASL is not limited to PV, but can be applied to other tracers (like chemical species in the atmosphere), including moisture. In particular, modelling water vapour by contours and subgrid-scale convection by Lagrangian particles may be a timely opportunity to improve the forecasting of precipitation [20].

Acknowledgments

Jérôme Fontane is supported by the European Community in the framework of the CONVECT project under Grant No. PIEF-GA-2008-221003.

References

[1] G.K. Batchelor, Computation of the energy spectrum in homogeneous two-dimensional turbulence, Phys. Fluid 12 (Suppl. II) (1969) 233–239.
[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 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, 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 (1989) 77–146.
[6] D.G. Dritschel, M.H.P. Ambaum, A contour-advective semi-Lagrangian numerical algorithm for simulating fine-scale conservative dynamical fields, Quart. J. Roy. Meteorol. Soc. 123 (1997) 1097–1130.
[7] D.G. Dritschel, M.H.P. Ambaum, The diabatic contour advective semi-Lagrangian algorithm, Mon. Weather Rev. 134 (9) (2006) 2503–2514.
[8] D.G. Dritschel, J. Fontane, The HyperCASL algorithm, in: Turbulence in the Atmosphere and Oceans Proceedings IUTAM/INI Workshop Held 8–12 December 2008, Springer-Verlag, 2009.
[9] D.G. Dritschel, L.M. Polvani, A.R. Mohebalhojeh, The contour-advective semi-Lagrangian algorithm for the shallow water equations, Mon. Weather Rev. 127 (7) (1999) 1551–1565.
[10] D.G. Dritschel, R.K. Scott, On the simulation of nearly inviscid two-dimensional turbulence, J. Comput. Phys. 228 (2009) 2707–2711.
[11] D.G. Dritschel, R.K. Scott, C. Macaskill, G.A. Gottwald, C.V. Tran, A unifying scaling theory for vortex dynamics in two-dimensional turbulence, Phys. Rev. Lett. 101 (2008) 094501.
[12] 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., submitted for publication.
[13] D.G. Dritschel, C.V. Tran, R.K. Scott, Revisiting Batchelor theory of two-dimensional turbulence, J. Fluid Mech. 591 (2007) 379–391.
[14] D.G. Dritschel, Á. Viúdez, A balanced approach to modelling rotating stably-stratified geophysical flows, J. Fluid Mech. 488 (2003) 123–150.
[15] D.G. Dritschel, D.W. Waugh, Quantification of inelastic interaction of unequal vortices in two-dimensional vortex dynamics, Phys. Fluid A 4 (1992) 1737–1744.
[16] D.G. Dritschel, N.J. Zabusky, On the nature of vortex interactions and models in unforced nearly-inviscid two-dimensional turbulence, Phys. Fluid 8 (1996) 1252–1256.
[17] R.H. Kraichnan, Inertial ranges in two-dimensional turbulence, Phys. Fluid 10 (1967) 1417–1423.
[18] W.H. Matthaeus, W.T. Stribling, D. Matinez, S. Oughton, D. Montgomery, Selective decay and coherent vortices in two-dimensional incompressible flows, Phys. Rev. Lett. 66 (1991) 2731–2734.
[19] D. Montgomery, W.H. Matthaeus, W.T. Stribling, D. Matinez, S. Oughton, Relaxation in two dimensions and the "sinh-Poisson" equation, Phys. Fluid A 4 (1992) 3–6.
[20] R.T. Pierrehumbert, The hydrologic cycle in deep time climate problems, Nature 419 (2002) 191–198.
[21] T.M. Schaerf, On Contour Crossings in Contour-advective Simulations of Geophysical Fluid Flows, Ph.D. Thesis, University of Sydney, 2006, 197 pp.
[22] C.V. Tran, D.G. Dritschel, Vanishing enstrophy dissipation in two-dimensional Navier–Stokes turbulence in the inviscid limit, J. Fluid Mech. 559 (2006) 107–116.
[23] G.K. Vallis, Atmospheric and Oceanic Fluid Dynamics: Fundamentals and Large-scale Circulation, Cambridge University Press, 2006. 745pp.
[24] N.J. Zabusky, M. Hughesand, K.V. Roberts, Contour dynamics for the Euler equations in two dimensions, J. Comput. Phys. 30 (1979) 96–106.