The diabatic contour advective
semi-Lagrangian algorithms for the spherical
shallow water equations

Ali R. Mohebalhojeh *

Institute of Geophysics, University of Tehran, Tehran, Iran

David G. Dritschel

School of Mathematics and Statistics, University of St Andrews,
St Andrews, Scotland

*Corresponding author address: Ali R. Mohebalhojeh, Institute of Geophysics, University of Tehran,
P. O. Box 14155-6466, Tehran 14359, Iran
E-mail: amoheb@ut.ac.ir
ABSTRACT

The diabatic contour-advective semi-Lagrangian (DCASL) algorithm is extended to the thermally-forced shallow water equations on the sphere. DCASL rests on the partitioning of potential vorticity (PV) to adiabatic and diabatic parts solved, respectively, by contour advection and a grid-based conventional algorithm. The presence of PV in the source term for diabatic PV makes the shallow water equations distinct from the quasigeostrophic model previously studied. To address the more rapid generation of fine-scale structures in diabatic PV, two new features are added to DCASL: (i) the use of multiple sets of contours with successively finer contour intervals and (ii) the application of the underlying method of DCASL at a higher level to diabatic PV. That is, the diabatic PV is allowed to have both contour and grid parts. The added features make it possible to make the grid part of diabatic PV arbitrarily small and thus pave the way for a fully-Lagrangian DCASL in the presence of forcing.

The DCASL algorithms are constructed using a standard semi-Lagrangian (SL) algorithm to solve for the grid-based part of diabatic PV. The 25-day time evolution of an unstable midlatitude jet triggered by the action of thermal forcing is used as a test case to examine and compare the properties of the DCASL algorithms with a pure SL algorithm for PV. Diagnostic measures on vortical and unbalanced activity as well as on the relative strength of the grid and contour parts of the solution for PV indicate that the superiority of contour advection can be maintained in the presence of even strong, non-smooth forcing.
1. Introduction

The simultaneous use of “grid” and “contour” representations for the materially conserved field of potential vorticity (PV), or some approximation to PV like quasi-geostrophic PV, is the main novel feature in the contour-advective semi-Lagrangian (CASL) algorithms developed since the original work by Dritschel and Ambaum (1997). The PV field is assumed to have a discrete distribution, i.e. a number of level sets divided by contours or PV jumps. Mathematically, the discrete distribution for PV is a piecewise uniform, discontinuous, contour representation, to be distinguished from the common use of contours to illustrate the distribution of geophysical fields. A contour is represented by a set of nodes, which are distributed nonuniformly according to the curvature of the contour. The contour representation removes any need for differentiation/interpolation in advecting PV and thus makes it possible to capture sharp gradients generated by nonlinear advection. The Lagrangian complexity generated by the cascade of (potential) enstrophy to small scales is handled by contour surgery (CS), documented in Dritschel (1988, 1989) and also in Dritschel and Ambaum (1997). CS serves as a kind of dissipation acting below grid-scale to increase the range of applicability of the contour representation for turbulent, high-Reynolds number flows.

At the heart of the CASL algorithm developed in Dritschel and Ambaum (1997) is a novel contour-to-grid transform. For the Lagrangian representation in contours, the contour-to-grid transform finds the Eulerian, grid representation. The transform is carried out on a grid finer than the main inversion grid on which the Eulerian fields are represented. In the diabatic CASL algorithm capable of handling non-conservative forcing, Dritschel and Ambaum (2006) introduced the hybrid use of the CASL algorithm of Dritschel and Ambaum
(1997) and a conventional algorithm by partitioning the PV field into adiabatic and diabatic parts. For brevity, we use the term DCASL to refer to diabatic CASL. The condition under which the partitioning is expected to work ideally is the smoothness of diabatic solution generated by the combined effects of forcing and nonlinear advection. The smoothness of diabatic PV is ensured as long as the forcing is sufficiently smooth and nonlinear advection is kept small. The radiative forcing in the quasi-geostrophic setting studied in Dritschel and Ambaum (2006) is a case of smooth forcing. The dominance of nonlinear advection in solutions for diabatic PV is avoided in DCASL of Dritschel and Ambaum (2006) by periodically transferring the diabatic PV into adiabatic PV. This is made possible by merging adiabatic and diabatic solutions periodically and “recontouring” the merged solution to reconstruct the contour representation for PV. In realistic circumstances, however, diabatic effects may lead to non-smooth forcing, as in the case of the thermally-forced spherical shallow water equations studied in this paper. The presence of PV in the source term may lead to a more rapid generation of fine-scale structures in the diabatic PV. The provision of the above-mentioned transfer mechanism in DCASL may not be adequate for highly accurate representation of diabatic PV.

The main tool in reconstructing the contour representation is a grid-to-contour transform. From a given Eulerian representation of PV on the inversion grid, the grid-to-contour transform finds a Lagrangian representation in the form of a set of contours for the PV levels at $Q = (\pm 1/2, \pm 3/2, \pm 5/2, \ldots)\Delta Q$, where $\Delta Q$ represents the contour interval. While the grid-to-contour transform assumes a continuous distribution for $Q$, the contour-to-grid transform treats the contours as piecewise uniform regions of PV divided discontinuously by contours. For any finite $\Delta Q$, the two transforms are thus not inverses. The successive
application of the grid-to-contour and contour-to-grid transforms will produce a residual containing both a debris of fine-scale structures and a large-scale contribution not captured by the contour representation. It is possible, in principle, to redefine and construct the contour-to-grid transform in such a way to make it the inverse of the grid-to-contour transform. Instead of constructing the inverse transform\(^1\), in this paper we approach the problem with the residual in a way that is mathematically motivated and avoids abandoning the use of piecewise uniform, discontinuous representations.

The mathematical question leading us to a new approach is as follows. For a continuous function \(f\), what is the best way of representing the function by a series \(f = \sum_{n=0}^{N} f_n\) of piecewise uniform, discontinuous functions \(f_n\) each determined by a unique jump \(\Delta_n f\)? An interesting case to consider is to set \(\Delta_n f = \varepsilon^n \Delta_0 f\), using a basic jump \(\Delta_0 f\) and a small number \(\varepsilon\). This paper does not attempt to solve this fundamental mathematical question. Instead, the idea of using multiple sets of contours with successively finer jumps is applied to construct various types of DCASL algorithms, from that described by Dritschel and Ambaum (2006) to a fully Lagrangian algorithm dispensing with the conventional algorithm for diabatic PV.

Our main objectives are (1) to elucidate the enhanced capability to handle arbitrary forcing, both localized and distributed, and (2) to assess accuracy in the face of challenges introduced by the addition of forcing to the primitive equations.

The paper is organized as follows. Section 2 introduces the thermally forced shallow water equations. The PV-based numerical algorithms spanning a spectrum of algorithms

\(^1\)The inverse transform is expected to be computationally expensive and lead to a re-emergence of the main problem with interpolation, that is, excessive diffusion.
from pure semi-Lagrangian to fully Lagrangian are presented in Section 3. The test case of Galewsky et al. (2004), referred to as GSP04, is extended to the thermally-forced shallow water equations in Section 4 where the solutions obtained by a standard semi-Lagrangian algorithm and various contour-advective algorithms are compared in detail. Finally a few conclusions are drawn in Section 5.

2. Formulation

Except for the changes due to the introduction of thermal forcing, the formulation follows that presented in Mohebalhojeh and Dritschel (2007), where the reader can find details not presented here for brevity. The thermally-forced shallow water equations in a momentum-mass representation are written as

\[
\begin{align*}
\frac{D\boldsymbol{v}}{Dt} + f \hat{k} \times \boldsymbol{v} &= -g \nabla h \\
\frac{\partial h}{\partial t} + \nabla \cdot (h \boldsymbol{v}) &= S_h = \frac{h_e - h}{\tau}
\end{align*}
\]

where \(\frac{D/\partial t}{D} = \partial/\partial t + \boldsymbol{v} \cdot \nabla\) is the material derivative, \(\boldsymbol{v}\) the velocity vector, \(h\) the depth field, \(h_e\) the equilibrium depth field, \(\tau\) the relaxation time, \(S_h\) the source term for \(h\), \(g\) the acceleration due to gravity, and \(f\) the Coriolis parameter. On the sphere, \(f = 2\Omega \sin \phi\) where \(\Omega_E\) is the rotation rate of the sphere and \(\phi\) is latitude. Denoting the global mean depth at \(t = 0\) by \(H_0\) and the global mean equilibrium depth by \(H_e\), we write \(h = H_0(1 + \tilde{h})\), \(h_e = H_e(1 + \tilde{h}_e)\), scale the mass continuity equation (2.2) by \(H_0\), introduce a gravity wave
speed \( c = \sqrt{gH_0} \), and rewrite the momentum-mass equations as

\[
\frac{D}{Dt} + f \mathbf{k} \times \mathbf{v} = -c^2 \nabla \bar{h} \tag{2.3}
\]

\[
\frac{\partial \bar{h}}{\partial t} + \nabla \cdot [(1 + \bar{h}) \mathbf{v}] = \frac{S_h}{\tau} = \frac{(H_e/H_0) - 1}{\tau} + \frac{(H_e/H_0)\bar{h}_e - \bar{h}}{\tau}, \tag{2.4}
\]

where \( \bar{h} \) is the (dimensionless) perturbation depth field and \( \bar{h}_e \) the perturbation equilibrium depth field. For brevity, from now on, the “perturbation” is dropped when referring to either of the two depth fields. In (2.4) we have accommodated the general case where \( H_e \neq H_0 \) (Polvani et al. 1995; Rong and Waugh 2004) such that on average mass can be introduced to, or removed from, the system by thermal forcing. The Rossby–Ertel PV is defined using the same form as in the unforced equations, namely by \( Q = (f + \zeta)/(1 + \bar{h}) \), where \( Q \) denotes PV and \( \zeta \) the relative vorticity. The presence of thermal forcing turns the material conservation of PV into

\[
\frac{DQ}{Dt} = S_Q = -\frac{Q}{1 + \bar{h}} S_{\bar{h}} \tag{2.5}
\]

where \( S_Q \) denotes the source term for \( Q \). For accuracy in representing balance and imbalance, following Smith and Dritschel (2006) and Mohebalhojeh and Dritschel (2007) we use the prognostic variables \( Q \), the velocity divergence \( \delta \), and the acceleration \( (D\mathbf{v}/Dt) \) divergence \( \gamma \) to evolve the SW equations on the sphere. Using the standard notations \( u \) for the velocity component in the longitudinal (\( \lambda \)) direction and \( \beta \) for the northward gradient of \( f \), it is straightforward to show that the acceleration divergence is equal to \( f\zeta - \beta u - c^2 \nabla^2 \bar{h} \). Finally, the prognostic equations for \( \delta \) and \( \gamma \) read

\[
\frac{\partial \delta}{\partial t} = \gamma - 2 \left[ \frac{\partial u}{\partial \phi} \left( \frac{\partial u}{\partial \phi} + \zeta \right) + \frac{\partial v}{\partial \phi} \left( \frac{\partial v}{\partial \phi} - \delta \right) \right] - \nabla \cdot (\delta \mathbf{v}) - \frac{\mathbf{v}^2}{a^2} \tag{2.6}
\]

\[
\frac{\partial \gamma}{\partial t} = c^2 \nabla^2 \{ \nabla \cdot [(1 + \bar{h}) \mathbf{v}] - S_{\bar{h}} \} + \frac{2\Omega_E}{a^2} \frac{\partial B}{\partial \lambda} - \nabla \cdot (Z\mathbf{v}) \tag{2.7}
\]
in which $v$ is the velocity component in the latitudinal ($\phi$) direction, $a$ is the Earth’s radius, $B \equiv c^2 \tilde{h} + \frac{1}{2} |v|^2$ is the Bernoulli pressure, and $Z = f(f + \zeta)$. Compared with the unforced equations, the $\delta$ equation remains intact and the $\gamma$ equation changes only due to the $-c^2 \nabla^2 S_h$ term on the right-hand side of (2.7).

In the numerical experiments reported in this paper, we nondimensionalize horizontal lengths by the Earth’s radius $a = 6.37122 \times 10^6$ m and time by one day $T_{\text{day}} = 2\pi/\Omega_E$ with $\Omega_E = 7.292 \times 10^{-5}$ s$^{-1}$. The nondimensional gravity wave speed $c$ is equal to $\sqrt{gH_0(T_{\text{day}}/a)} \approx 4.25$, where $g$ and $H_0$ are set to 9.80616 ms$^{-2}$ and $10^4$ m, respectively. Note that by definition $\tilde{h}$ and $\tilde{h}_e$ are nondimensional and that by the nondimensionalization used we can set $a = 1$, $\Omega = 2\pi$, $f = 4\pi \sin \phi$, and $\beta = 4\pi \cos \phi$ in (2.6) and (2.7) as well as in the definition of the prognostic variables $(Q, \delta, \gamma)$.

For ease of comparison, it is useful here to write the corresponding equations for the single-layer quasigeostrophic model on the $f$ plane used in Dritschel and Ambaum (2006). The Rossby–Ertel PV becomes quasigeostrophic PV, denoted by $q$, defined by $q = \nabla^2 \psi - \frac{\psi}{L_R^2}$ in which $\psi$ is the streamfunction and $L_R = c/f$ is the Rossby radius with $f$ being a constant on the $f$ plane. The thermal forcing becomes

$$S_q = \frac{\psi_e - \psi}{\tau L_R^2}$$

(2.8)

where $S_q$ denotes the source term for $q$, and $\psi_e$ is the equilibrium streamfunction field. As far as the DCASL algorithms are concerned, one should note the significant differences that arise between the thermally-forced quasigeostrophic and shallow water equations. In order of significance, the differences are in (a) the absence of PV in $S_q$ and the presence of PV in $S_Q$, (b) the linearity of $S_q$ and the nonlinearity of $S_Q$, (c) the inversion relations for $\psi$
and \( \tilde{h} \), and (d) the filtering and admissibility of inertia–gravity waves (imbalance). These differences may have implications for the construction of DCASL algorithms, which will be discussed in detail.

3. Numerical algorithms

The DCASL algorithm of Dritschel and Ambaum (2006) is extended to the forced shallow water equations on the sphere. For the reasons described shortly, the algorithm thus obtained will be called type I DCASL. The type I DCASL algorithm is constructed by a combination of the contour representation and contour advection used in the adiabatic CASL algorithm as described in Mohebalhojeh and Dritschel (2007) and a grid-based algorithm. The combination is made possible by a partitioning of the PV field into an adiabatic part \( Q_a \) and a diabatic part \( Q_d \), that is \( Q = Q_a + Q_d \). The time evolution of \( Q_a \) and \( Q_d \) are governed by

\[
\frac{DQ_a}{Dt} = 0 \quad \text{and} \quad \frac{DQ_d}{Dt} = S_Q.
\]

(3.1)

To deal with non-smooth forcing, we seek to minimize the norm of the grid-based part of the solution for PV. To this end, in addition to using multiple PV sets, a new type of DCASL called type II DCASL is introduced that makes simultaneous use of both contour and grid representations for \( Q_d \). The introduction of type II DCASL paves the way for a fully-Lagrangian DCASL. In the rest of this section, we describe these four algorithms: (a) a standard semi-Lagrangian, (b) the type I DCASL, (c) the type II DCASL, and (d) the fully-Lagrangian DCASL.
a. Semi-Lagrangian algorithm

The forced advection equation (2.5) is discretized according to

\[
Q(\mathbf{x}, t_{n+1}) = Q(\mathbf{x}_o, t_n) + \frac{\Delta t}{4} \left[ S_Q(\mathbf{x}, t_n) + 2S_Q(\mathbf{x}_m, t_{n+1/2}) + S_Q(\mathbf{x}_o, t_n) \right]
\] (3.2)

where \(\Delta t\) is time step, \(\mathbf{x}\) is the position of a fluid element (a grid-point in the semi-Lagrangian implementation), while \(\mathbf{x}_o\) and \(\mathbf{x}_m\) are the departure points for that fluid element at, respectively, \(t_n = n\Delta t\) and \(t_{n+1/2} = (n + 1/2)\Delta t\). The computation of the back trajectory and thus the departure points is carried out using the second-order implicit midpoint method (Smolarkiewicz and Pudykiewicz 1992; Durran 1999)

\[
\mathbf{x}_o = \mathbf{x} - \Delta t \mathbf{v}(\mathbf{x}_m, t_{n+1/2})
\] (3.3)

where \(\mathbf{x}_m = (\mathbf{x} + \mathbf{x}_o)/2\). The implicit equation (3.3) is solved by two iterations, sufficing for second-order accuracy. To circumvent pole problems, the trajectory computations are carried out in a Cartesian coordinate system attached to the center of the sphere. Since the grid points are shifted half a grid length away from the pole, no problematic trajectory computation at the poles is required. The velocity field at the midpoint is computed by time extrapolation,

\[
\mathbf{v}(\mathbf{x}, t_{n+1/2}) = \frac{3}{2}\mathbf{v}(\mathbf{x}, t_n) - \frac{1}{2}\mathbf{v}(\mathbf{x}, t_{n-1}).
\] (3.4)

The velocity fields at \((\mathbf{x}_m, t_{n+1/2})\) and \((\mathbf{x}_o, t_n)\) are evaluated from neighboring points by a piecewise bilinear interpolation. Similar to the computation of \(\mathbf{v}(\mathbf{x}, t_{n+1/2})\) in (3.4), the source term \(S_Q(\mathbf{x}, t_{n+1/2})\) is computed by time extrapolation from \(S_Q(\mathbf{x}, t_n)\) and \(S_Q(\mathbf{x}, t_{n-1})\). A piecewise bicubic Lagrange interpolation is used to compute the source term \(S_Q\) at the
midpoint \( \mathbf{x}_m \) and at the departure point \( \mathbf{x}_o \) as well as to compute \( Q \) at the departure point. For reference, hereafter we call the resulting algorithm SL.

\[ \]

\[ b. \text{Type I DCASL} \]

Apart from the use of multiple PV sets and technicalities of the spherical problem, this is essentially the algorithm of Dritschel and Ambaum (2006). The adiabatic PV is treated exactly as \( Q \) in adiabatic CASL, that is, represented as contours and converted to the inversion grid using the contour-to-grid transform. No source/sink comes into the advection of \( Q_a \), except a regularization procedure called “contour surgery” carried out periodically at time intervals \( T_s \) approximately equal to \( 1/10 \) to \( 1/20 \) of the advective time scale (see Dritschel and Ambaum 1997). Having an Eulerian representation only, the diabatic PV is solved using the semi-Lagrangian method described above. At certain regular time intervals \( T_m \) chosen to be a multiple of \( T_s \) (Dritschel and Ambaum 2006), the \( Q_a \) and \( Q_d \) fields are merged on an Eulerian ultra-fine grid (typically 8 times finer in each direction) and recontoured to reconstruct a new Lagrangian representation, i.e. a new \( Q_a \) with PV levels at \((\pm 1/2, \pm 3/2, \pm 5/2, \ldots)\Delta_0 Q\), where \( \Delta_0 Q \) is the contour interval. The recontouring is nothing but the realization of the grid-to-contour transform. Since the contour-to-grid transform is not the inverse of the grid-to-contour transform, the merging process of \( Q_a \) and \( Q_d \) will be generally incomplete and thus will generate a residual. The process can be repeated by recontouring the residual successively using PV levels at \((\pm 1/2, \pm 3/2, \pm 5/2, \ldots)\Delta_n Q\), where \( \Delta_n Q \) is the contour interval at iteration \( n \). The smaller residual left from the last iteration using the finest contour interval is set to \( Q_d \). The result will be the use of multiple PV sets.
for $Q_a$. A schematic illustration is shown in Fig. 1a.

To be clear, it is useful to restate here that the type I DCASL involves three spatial grids and two time intervals. The spatial grids are the main grid used for PV inversion and the computations required for the inertia–gravity wave part of the solution, the fine grid used for contour-to-grid conversion, and the ultra-fine grid used for the merging and recontouring. The two time intervals are used for periodic operation of surgery and the merging of $Q_a$ and $Q_d$, respectively.

c. Type II DCASL

The type II DCASL differs from the type I DCASL described above only in the treatment of $Q_d$. Unlike type I, in type II we use a contour representation for both $Q_a$ and $Q_d$. The procedure is as follows. At the end of the merging process, $Q_d$ is set to the residual left from the merging of $Q_a$ and $Q_d$ (see Fig. 1a). Let us denote the resulting residual PV field by $Q_d(x, T_m)$, where $T_m$ is the time interval for merging of $Q_a$ and $Q_d$. The treatment of $Q_d$ can be described algorithmically in the following loop carried out at time intervals of $J\Delta t$, $J$ being an integer, in which a new set or sets of contours are generated for $Q_d$. Partitioning $Q_d$ into a contour part $Q_{d,c}$ and a grid part $Q_{d,g}$ at the end of the each merging time interval, we set $Q_{d,g}(x, T_m) = Q_d(x, T_m)$ and $Q_{d,c}(x, T_m) = 0$. The operations carried out in this loop are then:

- Step 1: Over $J\Delta t$ time intervals, integrate $Q_{d,c}$ by contour advection and $Q_{d,g}$ by a semi-Lagrangian advection, e.g. that described in (3.1) where $Q$ is replaced by $Q_{d,g}$.
- Step 2: Generate a set (or sets) of contours for this updated $Q_{d,g}$ field. Reset $Q_{d,g}$ to the residual left from the contouring process. Include this new set of contours in the contour representation $Q_{d,c}$.

In this way, at $J\Delta t$ time intervals a new set (or sets) of contours is generated (see Fig. 1b). This loop can be repeated, in principle, up to the next merging of $Q_a$ and $Q_d$. The main constraining factor here is the memory cost of retaining large sets of contours. To reduce the number of contours used in representing $Q_{d,c}$, a periodic merging and recontouring of $Q_{d,c}$ and $Q_{d,g}$ in the same way as that described for $Q_a$ and $Q_d$ can be invoked (Fig. 1c). The time interval for merging of $Q_{d,c}$ and $Q_{d,g}$ can also be chosen to be a multiple of the surgery time. For $J$, the ideal choice is 1 such that at each time step a new set (or sets) of contours is generated for $Q_{d,g}$. The actual choices for the latter merging time interval and $J$ can be determined by computational cost and memory considerations.

Compared with the type I DCASL, it is useful to note that the type II involves two additional time intervals, one for the periodic contouring of the diabatic PV and one for the periodic merging and recontouring the grid and contour parts of the diabatic PV.

\textit{d. Fully-Lagrangian DCASL}

This algorithm makes no use of semi-Lagrangian integration. It can be readily obtained from the type II DCASL by setting (i) $J = 1$ and (ii) $Q_{d,g} = 0$. The latter is possible when the residuals of each contouring and recontouring in the type II DCASL are sufficiently small using multiple PV sets. In fact, setting $Q_{d,g} = 0$ means there is no need for a semi-Lagrangian integration. As a result, the algorithm can be simplified as follows:
• Step 1: At each time step, generate a set (or sets) of contours for \( S_{\Delta t} \). Include the generated set or sets in the contour representation for \( Q_d \).

• Step 2: Periodically recontour \( Q_d \) on the ultra-fine grid. Here we must perform a contour-to-grid transform on all the PV sets used for \( Q_d \), combine the resulting gridded fields, contour by a desired number of contour sets, and finally reset the residual to zero.

In essence, this algorithm contours the source term for PV with sufficient resolution to make it possible to truncate the grid part of the diabatic PV.

4. Diabatic extension of GSP04 test case

The GSP04 test case is modified in such a way to make it a suitable, challenging test case for the thermally-forced spherical shallow water equations. The test case starts with a prescribed localized zonal jet in the same form as in GSP04, that is

\[
          u(\phi) = \frac{u_{\text{max}}}{e_n} \exp \left[ \frac{1}{(\phi - \phi_0)(\phi - \phi_1)} \right],
\]

for \( \phi_0 < \phi < \phi_1 \) and zero otherwise, and finds the associated, balanced depth field \( \tilde{h}_z \).

In (4.1), \( u_{\text{max}} \) is the maximum speed, \( \phi_1 \) is the latitude of the northern boundary of the jet, \( \phi_0 \) is the latitude of the southern boundary of the jet, and \( e_n \) is a normalizing factor making \( u(\phi,0) \) equal to \( u_{\text{max}} \) at the center of the jet. The constants used are \( \phi_0 = \pi/7 \), \( \phi_1 = \pi/2 - \phi_0 \), \( e_n = \exp[-4/(\phi_1 - \phi_0)^2] \), and \( u_{\text{max}} = 80 \text{ ms}^{-1} \) in dimensional units. The nondimensional maximum jet speed is equal to \( 80(T_{\text{day}}/a) \sim 1.08 \). The zonal jet’s mid-point is located at \( \phi = \pi/4 \). A perturbation \( \tilde{h}' \) with nonzero global mean is then added to \( \tilde{h}_z \) in
order to trigger instability and make the flow unsteady. The presence of thermal forcing provides an alternative, convenient way of triggering instability. The modification is thus carried out by removing the perturbation \( \tilde{h}' \) and instead introducing an equilibrium depth field that brings about a significant disequilibrium to the initial state. The removal of \( \tilde{h}' \) is particularly desirable for the PV-based algorithms with partitioning of mass (layer depth) to a global mean and a perturbation. Unlike in GSP04, no re-adjustment of global mean mass is then required. The equilibrium perturbation depth field \( \tilde{h}_e \) is generated by a simple 30 degrees clockwise rotation of the initial depth field \( \tilde{h}_z \) about the \( y \)-axis of the Cartesian coordinate system \((x, y, z)\) attached to the center of the sphere. The contour maps of \( \tilde{h}_z, \tilde{h}_e \) and the initial disequilibrium depth field \( \tilde{h}_z - \tilde{h}_e \) are shown in Fig. 2. The simple rotation generates localized as well as broad-scale regions of disequilibrium in the depth field. The two dipolar structures centered around the jet maximum provide clear examples of local thermal disequilibrium in the depth and thus in the PV field. The inclusion of local sources/sinks in the mass and PV fields adds some realistic features to this test case, making it even more challenging for numerical algorithms.

Details of the setup of the GSP04 test case for PV-based algorithms can be found in Mohebalhojeh and Dritschel (2007). The setup of parameters specific to the diabatic extension is given here. We have used \( \tau = 5 \) days and \( H_e = H_0 \). For the DCASL algorithms, contour surgery is performed at 0.1 time intervals, except each 0.8 time interval where surgery is replaced by a recontouring of \( Q_a \) and \( Q_d \). That is, the time interval for merging the adiabatic and diabatic parts of PV is equal to 0.8. In the type II DCASL algorithm, the time interval for merging the contour and grid parts of the diabatic PV is set to 0.2. Time steps of \((5.0, 2.5, 1.25, 0.625) \times 10^{-3}\) are used at \(128 \times 128, 256 \times 256, 512 \times 512\) and \(1024 \times 1024\).
resolutions, respectively (see Table 1 of Mohebalhojeh and Dritschel 2007 for additional information).

With the nondimensionalization mentioned at the end of Section 2, for an order one Rossby number, low Froude number flow the nondimensional PV is expected to take values in the range $[-8\pi, 8\pi]$. For one PV set with $\Delta_0Q = (\pi/2, \pi/4, \pi/8, \pi/16)$, the errors in the contour representation of the initial grid PV computed as described in Mohebalhojeh and Dritschel (2007) are $(4.2, 1.6, 0.9, 0.7) \times 10^{-2}$, respectively. The error is measured by

$$L_2(Q) = \frac{\langle |Q(t, \lambda, \phi) - Q_{\text{ref}}(t, \lambda, \phi)|^2 \rangle^{1/2}}{\langle |Q_{\text{ref}}(t, \lambda, \phi)|^2 \rangle^{1/2}}$$

(4.2)

where $Q_{\text{ref}}$ is the reference solution and $\langle \rangle$ denotes the domain area average. The grid resolution is $128 \times 128$. The error stops converging for PV jumps smaller than $\pi/16$ when the same grid resolution is used. An explanation is needed here. The main factor responsible for the nonconvergence behavior has been found to be the subgrid scale averaging that is used when the contour representation is transformed to the inversion grid. In brief, the averaging includes an interpolation from the fine grid to the inversion grid and a distance-weighted distribution of the PV jump of each contour crossing to its two neighboring grid points in the latitudinal direction. As a sub-grid-scale averaging, it brings a benign smearing of the piecewise uniform, discontinuous, contour representation with some positive impact on the representation of imbalance. Although it is possible to achieve convergence with one PV set by removing the averaging altogether, the use of multiple PV sets instead offers a more economical way forward while keeping the averaging. For $(1, 2, 3, 4)$ PV sets with $\Delta_0Q = \pi/8$, $\varepsilon = 10^{-1}$, and $128 \times 128$ resolution, the errors are, respectively, $(0.864, 0.159, 0.089, 0.057) \times 10^{-2}$. Guided by these observations and numerical experimentation, to provide adequate
.resolution of \( Q \) in a contour representation, we have used \( \Delta_0 Q = \pi/8 \) for \( Q_a \), \( \Delta_0 Q = \pi/16 \) for \( Q_d \) in type II DCASL, \( \Delta_0 Q = \pi/256 \) for contours of \( S_Q \Delta t \) in the fully-Lagrangian DCASL, and with \( \varepsilon = 10^{-1} \) where needed. For the fully-Lagrangian DCASL only, \( \Delta_0 Q \) is halved with each doubling of the spatial resolution (and each halving of the time step to keep the Courant number fixed).

To start with, a qualitative description of the flow evolution is given. The thermal disequilibrium rapidly induces a small but finite meridional velocity field that slowly perturbs the PV. Comparing the actual velocity field and that obtained by PV inversion by means of Bolin–Charney balance (see Section 4b and the Appendix) tells us that the induced meridional velocity is largely balanced. The small imbalance at \( t = 0 \) due to the presence of thermal forcing has comparatively little impact on the subsequent evolution of the PV field. The perturbation to PV induced in this way leads to the onset of instability around time \( t = 4 \) and the subsequent roll up of the jet into complex vortical structures with increasingly sharp gradients of PV. A similar evolution is seen in the contour maps of the vorticity field shown in Fig. 3 for the type I DCASL algorithm with one PV set at \( t = (0, 2, 4, 6, 8, 10) \) days for \( 256 \times 256 \) resolution. The domain-maximum value of the local Rossby number \( R_{\text{max}} \), where \( R_{\text{O}} = |\zeta|/f_c \), rises from \( R_{\text{max}} = 1.09 \) at \( t = 0 \) to \( R_{\text{max}} = \sqrt{2} \) around \( t = 5.0 \), then fluctuates around \( \sqrt{2} \) until about \( t = 20.0 \), then decays to 0.98 by the end of simulation at \( t = 25 \). Here \( f_c \) denotes the Coriolis parameter \( f \) at the center of the jet. The domain-maximum value of local Froude number \( F_{\text{max}} \), where \( F_{\text{r}} = |v|/c \), stays close to 0.26 from \( t = 0.0 \) to around \( t = 7.0 \), then fluctuates and decays to 0.13 by the end of the simulation. The diabatic extension to GSP04 generates an order one Rossby number, low Froude number regime of flow.
The rest of this section is focused on providing evidence on two aspects of the DCASL algorithms. The first is on their effectively higher resolution for the vortical part of the flow resulting from the use of a Lagrangian representation for PV. This higher resolution is achieved with almost no adverse effect on the inertia–gravity part of the solution. The second is related to the way diabatic PV is handled in type I DCASL.

a. Representation of vortical flow

We first qualitatively describe how the vortical flow is represented in the SL and DCASL algorithms. To this end, we suffice to compare the vorticity field at $t = 10$, a time when instability is fully developed, thermal forcing has had a full effect, and the flow is still predictable in an initial-value-problem sense. Theoretically, it is the PV that best represents vortical flow. It is important, however, to see the desirable properties in the vorticity field with its more direct link to the velocity field and thus advection. In this regard, one should note that in the PV-based numerical algorithms for shallow water, $\zeta$ is a diagnostic variable whose computation involves both $Q$ and $\tilde{h}$. Shown in Fig. 4 is $\zeta$ for the SL algorithm at $128 \times 128$, $256 \times 256$, $512 \times 512$, and $1024 \times 1024$ resolutions. The corresponding results for the type I DCASL algorithm with one PV set are presented in Fig. 5 at $128 \times 128$, $256 \times 256$, and $512 \times 512$ resolutions. There are two main points emerging by comparing the two figures. The first point is evident. Much sharper gradients are captured by the DCASL algorithm at each resolution. The smearing of gradients by the SL algorithm is particularly serious at $128 \times 128$ resolution, leading to a significant loss of accuracy as judged by the highest resolution SL results. The second, less clear point is the presence of features in the
DCASL solutions that are captured by SL only at much higher resolutions. An example of such a feature is the small vortex seen near the left edge of the DCASL images around $-175^\circ$ longitude, $55^\circ$ latitude. This vortex with a longitudinal width of near to $5^\circ$ is totally missed by SL even at $512 \times 512$ resolution. In the longitudinal direction, across the vortex there are nearly 2, 4, 7, 14 grid points for $128 \times 128$, $256 \times 256$, $512 \times 512$, $1024 \times 1024$ resolutions, respectively. A second example is the small vortex near $60^\circ$ longitude, $30^\circ$ latitude seen in all the DCASL solutions with varying strengths. This feature is absent in the SL solutions at $128 \times 128$ and $256 \times 256$ resolutions. Snapshots of PV and vorticity at later times in this case as well as in the extensive long-term simulations of the polar stratospheric vortex (Polvani et al. 1995; Rong and Waugh 2004) indicate that the emergence of much finer vortical structures in the DCASL simulations is indeed a generic feature of these nonlinear flows.

Let us now quantify the relative strength of the vortical flow in the DCASL and SL integrations. The mass-weighted moments of potential vorticity $C_n$ and potential palinstrophy $P$ defined per unit area as

\begin{align*}
C_n &= \frac{1}{2} \langle (1 + \tilde{h}) Q^n \rangle, \quad (4.3a) \\
\tilde{C}'_0 &= \frac{1}{2} \langle (1 + \tilde{h}) Q^n \rangle - \frac{1}{2} \langle f^n \rangle, \quad (4.4) \\
P &= \frac{1}{2} \langle (1 + \tilde{h}) |\nabla Q|^2 \rangle \quad (4.3b)
\end{align*}

are used to provide global information on the distribution of PV. Among the infinity of such moments, Fig. 6 presents the time variation of $C'_2$ and $C'_4$ for the SL algorithm and the type I DCASL algorithm with three PV sets at various resolutions. For a better comparison, the flow-independent part of the moments has been subtracted to define $C'_n$,
The global quantities $C_2'$ and $C_4'$ are computed on an ultra-fine grid 8 times finer than the actual or inversion grid in each resolution. A piecewise bicubic Lagrange interpolation is carried out to find the ultra-fine fields of $Q$ and $\tilde{h}$ for both the SL and DCASL results. This way of computing the moments makes no direct use of the Lagrangian information available in the contour representation for $Q$ in DCASL. For the type I DCASL with three PV sets, it is also possible to convert the adiabatic PV ($Q_a$) from a contour representation directly to the ultra-fine grid and merge it with the interpolated diabatic PV ($Q_d$). The results plotted by $\circ$, $+$, and $\times$, respectively at $128 \times 128$, $256 \times 256$, and $512 \times 512$ resolutions, are for the inclusion of Lagrangian information in the computation of the moments. For brevity and explicitness, in referring to the latter results we simply use the term “with Lagrangian information”.

For the thermally-forced shallow water equations, $C_n'$ is not conserved in general. The time evolution equation for $C_n'$ can be obtained by a straightforward manipulation of the diabatic PV and mass equations, leading to

$$\frac{dC_n'}{dt} = \frac{(1 - n)}{2} \langle Q^n S_{\tilde{h}} \rangle. \tag{4.5}$$

That is, the inner product of the source term for mass $S_{\tilde{h}}$ and $Q^n$ determines the time evolution of $C_n'$. Although no analytical solution to (4.5) is available to lead us to the exact behavior of the moments, a meaningful comparison across the algorithms can be made by carefully monitoring the variation of $C_n'$ against resolution. In the present case, Fig. 6 points to three stages in the evolution of $C_2'$ and $C_4'$. The stages are (i) a gentle reduction in the first 5 days where the flow is dominantly zonal; (ii) a steeper reduction for $t \in [5, 15]$ where the flow is strongly non-zonal; and (iii) a gentle reduction for $t > 15$ where the flow becomes
largely zonal once more by the action of thermal forcing. From the start of stage (ii) onwards, each increase in resolution leads to a clear shift to larger values for both $C'_2$ and $C'_4$. The main differences between the results for the SL algorithm and the type I DCASL algorithm with three PV sets appear in stage (ii) and persist to stage (iii). The fact that the flow is predictable in an initial-value-problem sense for a significant part of stage (ii) makes the comparison even more important. The DCASL results at a given resolution are comparable to the SL results at two-to-four times higher resolution. With Lagrangian information, the differences become even more dramatic. For example, the $256 \times 256$ DCASL results ($\circ$) stand well above the $1024 \times 1024$ SL (thin solid).

The differences can be made quantitative by computing the relative differences between the SL and the type I DCASL with three PV sets, namely $C'_{rel,n} = \{C'_n(SL) - C'_n(DCASL)\}/C'_n(DCASL)$. The results are shown in Fig. 7 for $C'_2$ and $C'_4$ together where relative differences are also plotted for potential palinstrophy $P$. For both $C'_2$ and $C'_4$, the relative differences are negative in all three stages mentioned above. That is, at each resolution persistently lower values of $C'_2$ and $C'_4$ are obtained by the SL algorithm. The relative differences are most negative at stage (ii), when the flow is maximally nonzonal. In stage (ii), in the fully-developed state of instability, the SL algorithm underestimates $C'_2$ by between 20% to 10% across resolutions from $128 \times 128$ to $512 \times 512$. The corresponding underestimate for $C'_4$ is between 30% to 15%. For potential palinstrophy $P$, the onset of instability results in a substantial drop in $P$ for SL compared with the DCASL algorithm. The computation of $\nabla Q$ involved in estimating $P$ is carried out in the ultra-fine grid. The derivatives in estimating $P$ are computed using a spectral transform in longitude and compact fourth-order differencing in latitude, which is subject to errors we expect for grid-based computations. Nonetheless,
these estimates provide us with a valid comparison between the SL and DCASL algorithms as regards the magnitude of gradients maintained globally. Compared with the DCASL solutions, once instability develops, from 60% to 80% weaker gradients are maintained in the SL solutions. Interestingly, there is a remarkable agreement across resolutions. An important consequence of this loss of gradients is a very slow convergence rate of SL.

Further analysis shows that the above comparative results for the behavior of the vortical flow are insensitive to the particular DCASL algorithm chosen. To illustrate this robustness, in Fig. 8 we present the relative differences in $C'_2$, $C'_4$, and $P$ between the solutions obtained by the type I DCASL with one and three PV sets. In $C'_2$ and $C'_4$, the relative differences between the latter two DCASL solutions are nearly one tenth of those between the SL and the type I DCASL with three PV sets. While relative differences between the solutions of the SL and the type I DCASL with three PV sets show greater values for $C'_4$, the relative differences between the two type I DCASL solutions show greater values for $C'_2$. Since fine-scale details make a greater relative contribution to $C'_4$ than to $C'_2$, we can conclude that they are represented equally well in the two DCASL algorithms. For $P$ in stages (i) and (ii) described above, the relative differences between the two type I DCASL solutions stay less than 10%, substantially smaller than those between the solutions of SL and the type I DCASL with three PV sets. The growth observed in stage (iii) at 256 resolution has to be treated with caution. First, stage (iii) is beyond predictability in an initial-value problem sense. Second, the growth is not repeated at the higher 512 resolution. Third, the numerical errors in the grid-based computation of $P$ may affect the results.

It is important here to point out that the computational price paid by the DCASL algorithms to attain higher (Lagrangian) resolutions is much lower than that of the SL
algorithm using higher (Eulerian) resolutions. To illustrate, Table 1 compares the CPU times of the algorithms and resolutions tested relative to that of the lowest resolution SL for the 25-day experiments. Comparing e.g. the results for the $256 \times 256$ type I DCASL solution and the $1024 \times 1024$ SL solution, $(31, 32, 20)$-fold reductions in computational cost are obtained for the type I DCASL solutions obtained with one, two, and three PV sets, respectively. Even the most expensive DCASL, the type II with three PV sets for both adiabatic and diabatic PV, offers a six fold reduction. In the codes used to generate the results in this paper, all contour operations are carried out in Cartesian coordinates in a system attached to the center of the sphere. The computational cost of DCASL can be further reduced by rewriting the contour-to-grid conversion, node redistribution (see Dritschel and Ambaum 1997), and surgery routines in a latitude-longitude grid.

**b. Imbalance**

To properly assess accuracy, it is imperative to investigate the response of the unbalanced part of the flow to the higher resolution of the vortical flow available to the DCASL algorithms. To this end, we use the “diabatic Bolin–Charney” balance and PV inversion as described in the Appendix to decompose each instantaneous state of the flow to a balanced part representing vortical flow and an unbalanced part representing inertia–gravity waves. In the Bolin–Charney balance relations for the thermally-forced shallow water equations, the mass source term $S_h$ comes into the balanced divergence equation (A2). Taking into account this diabatic effect on balanced divergence has a noticeable positive impact on the quantitative results reported shortly. It is thus justifiable to use the term diabatic Bolin–Charney
balance to make a clear distinction from the unforced balance relations.

In Fig. 9, the squared $L_2$ norm of imbalance at $256 \times 256$ resolution is presented for the solutions of the type I DCASL with one PV set, the fully-Lagrangian DCASL with three PV sets for both $Q_a$ and $Q_d$, and SL. The $L_2$ norm is that defined in (A4). From the fully-Lagrangian to pure semi-Lagrangian solution, one can see that the representation of imbalance is fairly equal on this measure. In this case, the algorithm dependency of imbalance is clearly less than the sensitivity to initial conditions, as shown by the solid line in Fig. 9 showing the imbalance obtained for the solution of the type I DCASL with one PV set initialized by diabatic Bolin–Charney PV inversion. In any case, the use of balanced initial conditions will lead to an even better agreement among the algorithms.

The agreement on imbalance suggested by Fig. 9 can be quantified by comparing the time-averaged imbalance for each solution with that for a reference solution. More precisely, at each resolution we take the reference solution to be that given by the type I DCASL with three PV sets. For the time-averaged squared $L_2$ norm of imbalance $\|X_{imb}\|^2$, the relative differences $\left(\|X_{imb}\|^2 - \|X_{imb,ref}\|^2\right)/\|X_{imb,ref}\|^2$ calculated for each algorithm are summarized in Table 2. For SL, time-averaged imbalance is always less than that for the reference DCASL solution. This is consistent with our previous findings (Mohebalhojeh and Dritschel 2000, 2004) that smoother PV gradients lead to generally smaller values for imbalance. The relative differences are small, no greater than 6% and about 10 times smaller than those for potential palinstrophy $P$ in Fig. 7. More than anything, it confirms the successful handling of substantially sharper gradients of PV in the DCASL algorithms brought about by the use of $\delta$ and $\gamma$ as prognostic variables alongside PV. Unlike the SL, the DCASL algorithms show no sign-definite behavior. That is, relative differences of both positive and negative signs
are observed. Among the DCASL algorithms, the relative differences are below 3%, about half that for the SL algorithm, indicating less sensitivity in the representation of imbalance. This is a direct consequence of their close agreement in the representation of the dominant balanced, vortical flow (Fig. 8).

For the type I DCASL algorithms, Table 2 also gives separately results for the integrations when, at each merging of the adiabatic ($Q_a$) and diabatic ($Q_d$) PV, the latter is set to zero after recontouring. The results for these integrations are presented in the three bottom rows of Table 2. Setting $Q_d$ to zero after recontouring leads to non-conservation of grid PV at times of recontouring. The study of non-conservative — dissipative, anti-dissipative, or mixed — properties of recontouring needs a separate place. It is worth mentioning, however, that successive application of recontouring using finer PV jumps can turn it into a suitable regularization method to handle Lagrangian complexity. The aim here is simply to examine the likely effect of non-conservation on the representation of balance and imbalance. The effects are expected to depend on the magnitude and structure of the recontouring residuals that are set to zero. For integrations with $Q_d$ set to zero after recontouring, for the type I DCASL the relative differences take larger values at $128 \times 128$ and $512 \times 512$ solutions with one set of PV contours and at $256 \times 256$ solution with two sets of PV contours. With three sets of contours, the relative differences are less than 1%, comparable to those for the type I DCASL with one and two PV sets that keep the residuals. Confirming our expectation, concomitant with diminishing residuals with each extra set of PV contours, we can see a clear quantitative convergence of imbalance to the corresponding values obtained for the algorithms that conserve grid PV during the merging process. When recontouring behaves dissipatively, as is the case here with the three sets of PV contours, one can safely set $Q_d$ to
zero with no discernible, adverse effect on imbalance. When recontouring is not dissipative, maintaining PV conservation in the merging process becomes imperative.

c. Diabatic PV in type I DCASL

In the thermally forced shallow water equations, the source term for PV involves the PV itself. Interacting with nonlinear advection, the consequent forcing of diabatic PV by total PV may lead to the development of fine-scale structures in diabatic PV, with a rate much faster than that in the quasi-geostrophic case considered by Dritschel and Ambaum (2006). The determining factor to investigate is the growth of diabatic PV by the combined effects of advection/source/sink terms. To this end, we monitor the time evolution of $Q_d$ by computing $||Q_d||, ||v \cdot \nabla Q||, ||S_h||, ||S_Q||, ||v \cdot \nabla Q_d||/||S_Q||$, and $||v \cdot \nabla Q_d||/||v \cdot \nabla Q||$ at each time step. Here $||\ldots||$ denotes the usual $L_2$ norm. The main features of the time evolution of $Q_d$ are revealed by the selected results presented in Fig. 10 for the type I DCASL with one, two, and three PV sets at $128 \times 128$ resolution. Cycles of growth followed by sudden drop are observed in $||Q_d||, ||v \cdot \nabla Q_d||/||S_Q||$, and $||v \cdot \nabla Q_d||/||v \cdot \nabla Q||$. Being a grid-based computation, the magnitude of advection is subject to considerable numerical error. Nevertheless, the main points emerging from Fig. 10 appear robust. First, with two and three PV sets, the evolution of $Q_d$ is never dominantly advective. Second, from the onset of instability and beyond, the advection of $Q_d$ barely exceeds levels two orders of magnitude smaller than that of $Q$ in the type I DCASL with two and three PV sets. Third, despite the emergence of fine-scale structures arising from the source term, the smallness of the $Q_d$ advection means that any loss of numerical accuracy caused by the use of a grid-based algorithm for $Q_d$ will have an
insignificant effect on the representation of PV.

5. Concluding remarks

Making use of both contour and grid representations to evolve potential vorticity in PV-based models of the shallow water equations, the grid-to-contour and contour-to-grid transforms are exploited to construct diabatic CASL (DCASL) algorithms. The contour and grid representations for PV are merged periodically to reconstruct a contour (Lagrangian) representation in a process called recontouring. The fact that the two transforms are not inverses is addressed by introducing a sequence of piecewise uniform, discontinuous functions with successively finer PV jumps to represent PV. This technique is to ensure that the residual left from recontouring contains minimal fine-scale features. Further, provision is made to periodically contour the diabatic PV in order to accurately represent fine-scale features in the source term for diabatic PV, leaving only the broad-scale features to the grid-based algorithm to solve. The ultimate goal in introducing successively finer PV jumps and using a contour representation for diabatic PV is to minimize the contribution of the grid representation for diabatic PV. The two techniques together lead us to a pure contour-advective (fully-Lagrangian) algorithm. A spectrum of algorithms from the conventional, pure semi-Lagrangian to pure contour-advective is then at our disposal to investigate.

Unlike harmonic basis functions used in spectral methods, piecewise uniform, discontinuous functions work best in representing sharp gradients and fine-scale structures. Once combined with a conventional algorithm like a standard semi-Lagrangian one that is capable of representing the smooth part of the solution, the DCASL algorithms extend the range of
scales resolvable beyond that accessible to the conventional algorithm. The result is reflected in the higher accuracy of DCASL algorithms. The higher resolving power and numerical accuracy of DCASL algorithms are carefully examined in the diabatic extension of the GSP04 test case with a significant localized contribution to thermal forcing. The wide spectrum of thermal forcing as well as the presence of self-interaction in the evolution equation for PV (2.5) make the task of the diabatic PV-based algorithms significantly more challenging than in the quasi-geostrophic setting studied in Dritschel and Ambaum (2006).

In terms of the representation of the vortical flow, all the DCASL algorithms tested lead to solutions significantly distinct from that given by the pure SL algorithm. This is manifested clearly in the time evolution of potential enstrophy and higher moments, as well as of potential palinstrophy. On these global measures of vortical activity, the DCASL results differ very little during the time that the flow is predictable in an initial-value sense. The significantly stronger vortical flow at each resolution is achieved with little, if any, adverse effect on the unbalanced part of the flow. As measured by the diabatic Bolin–Charney PV inversion, imbalance is about 3% to 6% higher for a type I DCASL algorithm with three PV sets compared with that for the pure SL algorithm. Such small differences, likely to arise dynamically from the presence of sharp PV gradients, is a measure of success for (a) the prognostic variables $\delta$ and $\gamma$ used alongside $Q$ and (b) the merging followed by recontouring process carried out periodically in the DCASL algorithms. By maintaining a small diabatic solution, the numerical errors in the SL solution are kept sufficiently small.
Acknowledgments.

A.R.M. thanks the UK Natural Environment Research Council for a Research Fellowship, and the Universities of St Andrews and Tehran for providing support during this research. We would also like to thank Mohammad Mirzaei for help in preparing the figures.
Bolin–Charney balance & PV inversion

The Bolin–Charney balance relations are obtained by setting to zero the variable Ξ and its first time derivative, where Ξ is defined by

$$\Xi = (f - c^2 \nabla^2 \bar{h} - \beta u_\psi) - 2 \left[ \frac{\partial u_\psi}{\partial \phi} \left( \frac{\partial u_\psi}{\partial \phi} + \zeta \right) + \left( \frac{\partial v_\psi}{\partial \phi} \right)^2 \right] - \frac{|\psi|^2}{a^2}. \quad (A1)$$

In (A1), $\psi = \hat{z} \times \nabla \psi$ is the rotational velocity, $\psi$ denotes the streamfunction, $u_\psi$ and $v_\psi$ are, respectively, the longitudinal and latitudinal components of $\psi$, and $\hat{z}$ is the unit vector in the local vertical direction. The balance relation $\Xi = 0$ together with the definition of PV, the Poisson equation $\nabla^2 \psi = \zeta$, constitute a closed system of equations to solve for $\bar{h}$, $\zeta$, $\psi$, and $\psi$ from the known instantaneous distribution of PV. The balanced divergence $\delta$ is obtained from $\partial \Xi / \partial t = 0$, which gives

$$(c^2 \nabla^2 - f^2) \delta = \nabla \cdot (f \nabla \psi) - c^2 \nabla^2 \left[ \nabla \cdot (\bar{h} \nabla) - S_{\bar{h}} \right] +$$

$$2 \frac{\partial}{\partial t} \left[ \frac{\partial u_\psi}{\partial \phi} \left( \frac{\partial u_\psi}{\partial \phi} + \zeta \right) + \left( \frac{\partial v_\psi}{\partial \phi} \right)^2 \right] + \frac{2}{a^2} (u_\psi \frac{\partial u_\psi}{\partial t} + v_\psi \frac{\partial v_\psi}{\partial t}), \quad (A2)$$

where

$$\frac{\partial \psi}{\partial t} = \hat{z} \times \nabla \nabla^{-2} \left\{ -\nabla \cdot [(f + \zeta) \nabla] \right\}. \quad (A3)$$

With (A3), the Helmholtz decomposition $\psi = \hat{z} \times \nabla \psi + \nabla \chi$, and the Poisson equation $\nabla^2 \chi = \delta$, Eq. (A2) can be solved to determine $\delta$ and $\psi$. The balanced fields can be subtracted from the actual shallow water fields to define the unbalanced fields. For the state vector $(u, v, \bar{h})$
of the system, we define the $L_2$ norm (Mohebalhojeh and Dritschel 2001; Mohebalhojeh and Dritschel 2004)

$$
||X|| = \left( \frac{1}{2} H \int_{\pi/2}^{\pi/2} \int_0^{2\pi} \left[ (u^2 + v^2) + c^2 \tilde{h}^2 \right] \cos \phi \, d\phi \, d\lambda \right)^{1/2}.
$$

(A4)

The factor $H/2$ is simply used to make $||X||^2$ a linearized available energy with a physical meaning as well. We measure imbalance by $||X_{imb}|| = ||X - X_b||$, where $X$, $X_b$, and $X_{imb}$ stand for the actual, balanced, and unbalanced state vectors.
REFERENCES


List of Figures

1. A schematic illustration of (a) merging and recontouring $Q_a$ and $Q_d$, carried out in type I and type II DCASL, (b) the loop carried out in type II where at $J\Delta t$ time intervals the diabatic PV is contoured, and (c) the repeat of the latter loop for the desired number of times, 4 here, followed by merging and recontouring of the grid and contour parts of the diabatic PV. Note that panel (a) shows an example of twice recontouring leading to generation of two sets of contours for adiabatic PV. 

2. The initial depth field $\tilde{h}_z$ (left panel), the equilibrium depth field $\tilde{h}_e$ (middle panel), and $\tilde{h}_z - \tilde{h}_e$ (right panel). The contour interval is 0.01 and the zero contour has not been plotted.

3. The vorticity field at $t = (0, 2, 4, 6, 8, 10)$ days obtained by the type I DCASL algorithm with one PV set. The contour interval is 2 (days)$^{-1}$, starting at $\pm 1$ (days)$^{-1}$ (levels shown are $\pm 1, \pm 3, \pm 5, \ldots$ (days)$^{-1}$). The same contouring scheme is used in subsequent figures. The solid and dotted lines represent positive and negative contours. The resolution is $256 \times 256$. Only the northern hemisphere is shown in this and subsequent figures.

4. The vorticity field at $t = 10$ computed by the SL algorithm at $128 \times 128, 256 \times 256, 512 \times 512$ and $1024 \times 1024$ resolutions from top to bottom, respectively.

5. The vorticity field at $t = 10$ computed by the type I DCASL algorithm with one PV set at $128 \times 128, 256 \times 256, 512 \times 512$ resolutions from top to bottom, respectively.
Time variation of $C_2'$ (left panel) and $C_4'$ (right panel) for the SL algorithm at $128 \times 128$ (thin dotted), $256 \times 256$ (thin dash-dotted), $512 \times 512$ (thin dashed), $1024 \times 1024$ (thin solid) resolutions and for the type I DCASL algorithms with three PV sets at $128 \times 128$ (thick dotted), $256 \times 256$ (thick dash-dotted) and $512 \times 512$ (thick dashed) resolutions. Also shown are the results for the same DCASL algorithms when the contribution of $Q_a$ is computed directly using contour-to-grid transform carried out on the ultra-fine grid. The latter results are shown at $128 \times 128$ ($\circ$), $256 \times 256$ ($+$), and $512 \times 512$ ($\times$) resolutions.

The relative differences in (a) potential enstrophy $C_2''$, (b) potential enstrophy squared $C_4''$ and (c) potential palinstrophy $P$ between the solutions of the SL and the type I DCASL with three PV sets at $128 \times 128$ (dotted), $256 \times 256$ (dashed) and $512 \times 512$ (solid) resolutions. At each resolution, the DCASL results are taken as the reference solution and the relative differences are computed against them.

The relative differences in (a) potential enstrophy $C_2''$, (b) potential enstrophy squared $C_4''$ and (c) potential palinstrophy $P$ between the solutions obtained using one PV set and three PV sets at $128 \times 128$ (dotted), $256 \times 256$ (dashed) and $512 \times 512$ (solid) resolutions. As in the previous figure, the results for the DCASL with three PV sets are taken as the reference solution.

The squared $L_2$ norm of imbalance at $256 \times 256$ resolution for the solutions of the SL (dotted line), the type I DCASL with one PV set (dashed line), and the fully-Lagrangian DCASL with three PV sets for $Q_a$ and $Q_d$ (dashed-dotted line). The solid line is for the solution of the type I DCASL with one PV set when initialized by Bolin–Charney PV inversion.
\[ \|Q_d\| \text{ (left panel)}, \|v \cdot \nabla Q_d\|/\|S_Q\| \text{ (middle panel), and } \|v \cdot \nabla Q_d\|/\|v \cdot \nabla Q\| \text{ (right panel)} \]

for the solutions of the type I DCASL with one PV set (dotted), two PV sets (dashed), and three PV sets (solid) at 128 \times 128 resolution.
FIG. 1. A schematic illustration of (a) merging and recontouring $Q_a$ and $Q_d$, carried out in type I and type II DCASL, (b) the loop carried out in type II where at $J\Delta t$ time intervals the diabatic PV is contoured, and (c) the repeat of the latter loop for the desired number of times, 4 here, followed by merging and recontouring of the grid and contour parts of the diabatic PV. Note that panel (a) shows an example of twice recontouring leading to generation of two sets of contours for adiabatic PV.
Fig. 2. The initial depth field $\tilde{h}_z$ (left panel), the equilibrium depth field $\tilde{h}_e$ (middle panel), and $\tilde{h}_z - \tilde{h}_e$ (right panel). The contour interval is 0.01 and the zero contour has not been plotted.
Fig. 3. The vorticity field at $t = (0, 2, 4, 6, 8, 10)$ days obtained by the type I DCASL algorithm with one PV set. The contour interval is 2 (days)$^{-1}$, starting at $\pm 1$ (days)$^{-1}$ (levels shown are $\pm 1$, $\pm 3$, $\pm 5$, \ldots (days)$^{-1}$). The same contouring scheme is used in subsequent figures. The solid and dotted lines represent positive and negative contours. The resolution is $256 \times 256$. Only the northern hemisphere is shown in this and subsequent figures.
Fig. 4. The vorticity field at $t = 10$ computed by the SL algorithm at $128 \times 128$, $256 \times 256$, $512 \times 512$ and $1024 \times 1024$ resolutions from top to bottom, respectively.
Fig. 5. The vorticity field at $t = 10$ computed by the type I DCASL algorithm with one PV set at $128 \times 128$, $256 \times 256$, and $512 \times 512$ resolutions from top to bottom, respectively.
Fig. 6. Time variation of $C_2^2$ (left panel) and $C_4^4$ (right panel) for the SL algorithm at $128 \times 128$ (thin dotted), $256 \times 256$ (thin dash-dotted), $512 \times 512$ (thin dashed), $1024 \times 1024$ (thin solid) resolutions and for the type I DCASL algorithms with three PV sets at $128 \times 128$ (thick dotted), $256 \times 256$ (thick dash-dotted) and $512 \times 512$ (thick dashed) resolutions. Also shown are the results for the same DCASL algorithms when the contribution of $Q_a$ is computed directly using contour-to-grid transform carried out on the ultra-fine grid. The latter results are shown at $128 \times 128$ (○), $256 \times 256$ (+), and $512 \times 512$ (×) resolutions.
FIG. 7. The relative differences in (a) potential enstrophy $C'_2$, (b) potential enstrophy squared $C'_4$, and (c) potential palinstrophy $P$ between the solutions of the SL and the type I DCASL with three PV sets at $128 \times 128$ (dotted), $256 \times 256$ (dashed) and $512 \times 512$ (solid) resolutions. At each resolution, the DCASL results are taken as the reference solution and the relative differences are computed against them.
**Fig. 8.** The relative differences in (a) potential enstrophy $C_2'$, (b) potential enstrophy squared $C_4'$ and (c) potential palinstrophy $P$ between the type I DCASL solutions obtained using one PV set and three PV sets at $128 \times 128$ (dotted), $256 \times 256$ (dashed) and $512 \times 512$ (solid) resolutions. As in the previous figure, the results for the DCASL with three PV sets are taken as the reference solution.
Fig. 9. The squared $L_2$ norm of imbalance at $256 \times 256$ resolution for the solutions of the SL (dotted line), the type I DCASL with one PV set (dashed line), and the fully-Lagrangian DCASL with three PV sets for $Q_a$ and $Q_d$ (dashed-dotted line). The solid line is for the solution of the type I DCASL with one PV set when initialized by Bolin–Charney PV inversion.
FIG. 10. $\|Q_d\|$ (left panel), $\|v.\nabla Q_d\|/\|S_Q\|$ (middle panel), and $\|v.\nabla Q_d\|/\|v.\nabla Q\|$ (right panel) for the solutions of the type I DCASL with one PV set (dotted), two PV sets (dashed), and three PV sets (solid) at $128 \times 128$ resolution.
List of Tables

1. The CPU time over the 25-day diabatic GSP04 experiment relative to that of SL run at 128 × 128 resolution. .................................................. 47

2. The relative difference of the time-averaged squared $L_2$ norm of imbalance for various algorithms and resolutions. At each resolution, the reference solution is taken to be that given by the type I DCASL with three PV sets. ........................................ 48
Table 1. The CPU time over the 25-day diabatic GSP04 experiment relative to that of SL run at 128 \times 128 resolution.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>128 \times 128</th>
<th>256 \times 256</th>
<th>512 \times 512</th>
<th>1024 \times 1024</th>
</tr>
</thead>
<tbody>
<tr>
<td>SL</td>
<td>1.00</td>
<td>10.88</td>
<td>109.88</td>
<td>1117.75</td>
</tr>
<tr>
<td>Type I DCASL with one PV set</td>
<td>3.26</td>
<td>36.33</td>
<td>505.12</td>
<td>—</td>
</tr>
<tr>
<td>Type I DCASL with two PV sets</td>
<td>3.51</td>
<td>34.82</td>
<td>248.33</td>
<td>—</td>
</tr>
<tr>
<td>Type I DCASL with three PV sets</td>
<td>5.23</td>
<td>55.36</td>
<td>397.66</td>
<td>—</td>
</tr>
<tr>
<td>Type II DCASL with three PV sets</td>
<td>16.30</td>
<td>192.18</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Fully-Lagrangian</td>
<td>14.58</td>
<td>174.83</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>
Table 2. The relative difference of the time-averaged squared $L_2$ norm of imbalance for various algorithms and resolutions. At each resolution, the reference solution is taken to be that given by the type I DCASL with three PV sets.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>128 × 128</th>
<th>256 × 256</th>
<th>512 × 512</th>
</tr>
</thead>
<tbody>
<tr>
<td>SL</td>
<td>−0.05879</td>
<td>−0.02950</td>
<td>−0.05702</td>
</tr>
<tr>
<td>Type I DCASL with one PV set</td>
<td>+0.00355</td>
<td>+0.00429</td>
<td>−0.01536</td>
</tr>
<tr>
<td>Type I DCASL with two PV sets</td>
<td>−0.01088</td>
<td>+0.02012</td>
<td>−0.00325</td>
</tr>
<tr>
<td>Type II DCASL with three PV sets</td>
<td>+0.01130</td>
<td>+0.02469</td>
<td></td>
</tr>
<tr>
<td>Fully-Lagrangian DCASL</td>
<td>+0.02536</td>
<td>+0.01609</td>
<td></td>
</tr>
<tr>
<td>Type I DCASL with one PV set (zero residual)</td>
<td>−0.09234</td>
<td>−0.00015</td>
<td>+0.14287</td>
</tr>
<tr>
<td>Type I DCASL with two PV sets (zero residual)</td>
<td>+0.028844</td>
<td>+0.06106</td>
<td>−0.00520</td>
</tr>
<tr>
<td>Type I DCASL with three PV sets (zero residual)</td>
<td>−0.00246</td>
<td>−0.00048</td>
<td>−0.00984</td>
</tr>
</tbody>
</table>