INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2007; **55**:127–142
Published online 31 January 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1445

Semi-Lagrangian advection on a spherical geodesic grid

Maria Francesca Carfora*, †

Istituto per le Applicazioni del Calcolo 'Mauro Picone', CNR, via P. Castellino 111, 80131 Napoli, Italy

SUMMARY

A simple and efficient numerical method for solving the advection equation on the spherical surface is presented. To overcome the well-known 'pole problem' related to the polar singularity of spherical coordinates, the space discretization is performed on a geodesic grid derived by a uniform triangulation of the sphere; the time discretization uses a semi-Lagrangian approach. These two choices, efficiently combined in a substepping procedure, allow us to easily determine the departure points of the characteristic lines, avoiding any computationally expensive tree-search. Moreover, suitable interpolation procedures on such geodesic grid are presented and compared. The performance of the method in terms of accuracy and efficiency is assessed on two standard test cases: solid-body rotation and a deformation flow. Copyright © 2007 John Wiley & Sons, Ltd.

Received 14 June 2006; Revised 6 December 2006; Accepted 7 December 2006

KEY WORDS: rotating sphere; geodesic grid; semi-Lagrangian; advection

INTRODUCTION

Spherical geodesic grids have been used in some early meteorological works [1–3]. However, these methods have recently been rediscovered ([4–6] among the others) as more and more researchers have begun to move away from spectral methods towards finite volumes, finite elements and finite difference methods for spherical domains. The global numerical weather prediction model (GME) developed by the German Weather Service [7] uses an icosahedral–hexagonal grid which allows a nearly uniform resolution with less than 20% variation of mesh size over the globe.

These grids are based on the refinements of uniform triangulations of the sphere, in an attempt of better representing the Earth atmosphere. Indeed, conventional models with latitude–longitude

*Correspondence to: Maria Francesca Carfora, Istituto per le Applicazioni del Calcolo 'Mauro Picone', CNR, via P. Castellino 111, 80131 Napoli, Italy.
†E-mail: f.carfora@iac.cnr.it

grids give horizontal resolution that varies systematically and anisotropically with latitude; on the contrary, geodesic grids can offer quasi-uniform resolution, overcoming the 'pole problem'; however, their implementation often suffers from high computational effort and expensive integration schemes which have to be generated automatically as in [7].

The treatment of advection represents a fundamental part in every numerical method for atmospheric circulation: since stability, accuracy and efficiency of the overall models depend on each of the approximations we made, the introduction of more accurate numerical schemes for the advective terms can improve the performance of the model. Moreover, the complexity of the full circulation models makes that all numerical approximations made there are first fully analysed on simpler test equations both from the theoretical and computational point of view. In the last years, semi-Lagrangian techniques became very popular (see [8] for a review), since they guarantee unconditional stability, and then permit the use of large time steps, not constrained by the Courant–Friedrichs–Lewy (CFL) condition. However, in the general literature, semi-Lagrangian techniques approximate the trajectories with straight lines in the longitude–latitude plane, which would be a poor approximation, especially near the poles. Amato and Carfora [9] have already developed a semi-Lagrangian method for the numerical solution of the advection equation on a uniform longitude–latitude grid, with a substepping procedure which permits to accurately reproduce the characteristic lines. The idea of the method is based on a work by Casulli [10] and has been modified to adapt to spherical geometry. In this procedure, the characteristic system is solved with a first-order numerical scheme (Euler method) or also with a Runge–Kutta method. In both cases, with a single-step procedure or with the proposed substepping procedure, the value of the advected field at the departure point of the characteristic lines is then interpolated.

As an alternative to the standard iterative interpolating procedure, which dates back to Ritchie [11], McGregor [12] introduced a departure point calculation procedure that does not involve interpolation. His scheme is based on a truncated Taylor series expansion for the generic departure point about the corresponding arrival point along the characteristic line and requires the evaluation of the partial derivatives of the advected field up to the chosen order of approximation.

The present paper also addresses to the simplified problem of pure advection. It represents a departure from previously published works on solving advection and shallow water equations on the sphere [9, 13, 14] since it proposes the use of an icosahedral grid. Even if these grids have gained a considerable interest in the meteorological community, the related interpolation procedures are still little studied. For this reason, the author developed '*ad hoc*' interpolation procedures and compared them with several published interpolation schemes belonging to three groups: linear schemes, such as the one considered in [7], distance-based schemes and cubic interpolators adapted to spherical triangles. Implementation and cost-efficiency details of these schemes for a simple test case have been reported in details in a recently performed study [15], whose conclusions lead us to prefer radial basis functions (RBFs) and cubic interpolators of Lagrange type, that yield very accurate results at a reasonable cost.

These schemes are there applied to a standard test problem [16] for advection on the sphere; for such a problem, we also consider as a comparison the implementation of McGregor's approach on icosahedral grids as proposed by Giraldo [17].

Moreover, a second and more challenging test problem has been considered: an idealized vortex problem in spherical geometry with an exact solution [18].

DOI: 10.1002/fld

THE NUMERICAL METHOD

Grid generation

The icosahedral–hexagonal grid, already introduced in meteorological modelling in the late 1960s [1, 2], has been gaining increasing interest in recent years. The approach described here was developed in a seminal paper by Baumgardner and Frederickson [19].

To generate the grid, a regular icosahedron is embedded into a sphere such that two of its 12 corners coincide with the North and South poles. Five of the other 10 corners are spaced at equal longitudinal intervals of 72° (= 360°/5) along a latitude circle at 26.565°N, the other five along a latitude circle at 26.565°S.

This spherical icosahedron serves as a macrotriangulation of the sphere which allows the application of common refinement techniques: at every step each midpoint of a triangle edge is connected by geodesic arcs with the midpoints of the two neighbouring sides of the triangles on both sides of it.

This construction procedure yields a grid consisting of $10\,n_i^2 + 2$ grid points (nodes) and $20\,n_i^2$ elementary spherical triangles, where $n_i = 2^i$ is the number of equal intervals into which each side of the original icosahedral triangles is divided. Each of these $10\,n_i^2 + 2$ grid points has six nearest neighbours with the exception of the corners of the original icosahedron, which have only five neighbours. We therefore refer to these 12 special points as *pentagonal* points. Figure 1 shows the grid for the refinement level 4.

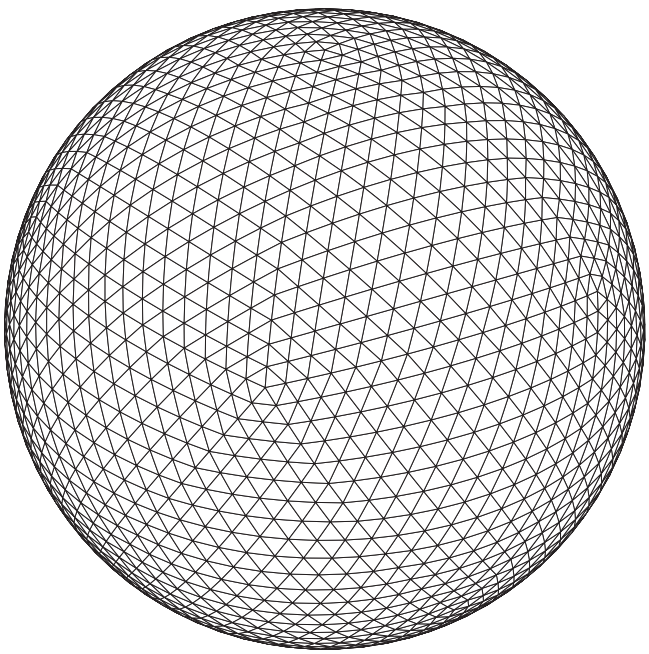

Figure 1. Numerical grid corresponding to refinement level 4.

DOI: 10.1002/fld

The number n_i is a natural parameter for specifying the resolution of the grid. The (minimum) spacing between grid points is then the length of a side of the original icosahedral triangles (about 7054 km for the Earth) divided by n_i. For example, at the refinement level $i = 7$, where $n_i = 128$, we obtain a spacing between grid points of about 55 km.

The icosahedral–hexagonal grid provides a nearly uniform coverage of the sphere even though the hexagonal cells vary somewhat in their exact shape and size, especially those close to the pentagonal points. The pentagons, however, are perfectly regular.

By combining the areas of pairs of the original adjacent icosahedral triangles, as shown in Figure 2, the global grid can logically also be viewed as comprising 10 diamonds, each of which has $n_i \times n_i$ unique grid points. The indexing on a diamond is based on the convention that those $n_i \times n_i$ grid points that are unique to each diamond are numbered from 1 to n_i in the rows and columns of the data arrays. The grid points on the diamonds edges, $(0, 1)$ to $(0, n_i + 1)$ and $(0, n_i + 1)$ to $(n_i, n_i + 1)$, are shared between adjacent diamonds and their data values must be exchanged at each time step. The polar points $(0, 1)$ are each shared by five diamonds: diamonds 1–5 share the North pole and diamonds 6–10 share the South pole.

From the computational point of view the icosahedral–hexagonal grid offers the major advantage that no indirect addressing is required. The data structure is regular and has the dimensions $(0: n_i, n_i + 1, 10)$, that is, consists of 10 logical square arrays of points.

The indices of the neighbouring points are given by fixed offsets from the index of the home point. These operations can be coded to obtain high efficiencies on both vector and scalar computer architectures. Furthermore, the square arrays of points are readily partitioned in a domain decomposition strategy for distributed memory parallel architectures.

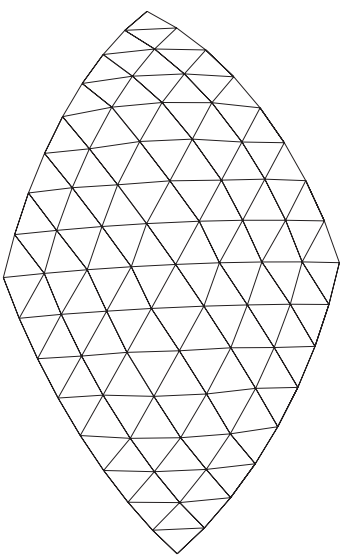

Figure 2. The arranging of two adjacent icosahedral triangles in a diamond for the refinement level 3.

DOI: 10.1002/fld

Time discretization

The two-dimensional advection equation can be written in polar coordinates on a spherical surface of radius R as

$$\frac{\mathrm{d}}{\mathrm{d}t}F(\lambda,\theta)=0 \tag{1}$$

with λ, θ being longitude and latitude, respectively, and

$$\frac{\mathrm{d}}{\mathrm{d}t}()=\frac{\partial}{\partial t}()+\frac{u}{R\cos\theta}\frac{\partial}{\partial\lambda}()+\frac{v}{R}\frac{\partial}{\partial\theta}()=\frac{\partial}{\partial t}()+u\frac{\partial}{\partial x}()+v\frac{\partial}{\partial y}()$$

with x, y being curvilinear coordinates, defined as $\mathrm{d}x=R\cos\theta\,\mathrm{d}\lambda$ and $\mathrm{d}y=R\,\mathrm{d}\theta$, and (u, v) being wind components towards east and north, respectively, $u=\mathrm{d}x/\mathrm{d}t=R\cos\theta\,\mathrm{d}\lambda/\mathrm{d}t$ and $v=\mathrm{d}y/\mathrm{d}t=R\,\mathrm{d}\theta/\mathrm{d}t$.

Its semi-Lagrangian approximation is based on the numerical approximation of the Lagrangian particle paths: we consider a fluid particle arriving at time t^{n+1} in a gridpoint P and we have to determine its departure point $P_\star$ at time t^n. In order to do this, we integrate, backward in time from t^{n+1} to t^n, the following system of ordinary differential equations for any vertex of the grid:

$$\begin{aligned}\frac{\mathrm{d}\lambda}{\mathrm{d}t}&=\frac{u}{R\cos\theta}\\ \frac{\mathrm{d}\theta}{\mathrm{d}t}&=\frac{v}{R}\end{aligned} \tag{2}$$

For each gridpoint P_i, with i ranging from 1 to N_P, the new value of F is then defined as

$$F_i^{n+1}=F_{i_\star}^n \tag{3}$$

Since, in general, the points $P_{i_\star}$ do not belong to the numerical grid, a suitable interpolation method has to be used.

In many models that use this semi-Lagrangian approach, the characteristic lines are approximated with straight lines in the (λ, θ) plane, whose direction is determined iteratively.

In our model, in order to reconstruct more accurately the characteristic lines, we introduced a substepping procedure [9, 10] and solve system (2) evaluating λ, θ in N_τ intermediate time steps, where N_τ is to be chosen. As shown in the cited works, system (2) can be efficiently integrated, backward in time, by a simple explicit Euler method (first-order accurate)

$$\begin{aligned}\lambda^{(k-1)}&=\lambda^{(k)}-\tau\frac{u^{(k)}}{R\cos\theta^{(k)}}\\ \theta^{(k-1)}&=\theta^{(k)}-\tau\frac{v^{(k)}}{R}\end{aligned} \tag{4}$$

with $\tau=\Delta t/N_\tau$ and k ranging from 1 to N_τ; $(\lambda^{(N_\tau)}, \theta^{(N_\tau)})$ refer to the generic grid point P_i and $(\lambda^{(0)}, \theta^{(0)})$ to the corresponding departure point $P_{i_\star}$.

As a comparison, a second-order Runge–Kutta method has been considered and its accuracy improvements evaluated with respect to the higher computational cost. In both cases, to accurately

DOI: 10.1002/fld

integrate system (2) also along near-polar trajectories, we adopt a rotated coordinate system, as commonly suggested in the literature [20], only for situations involving strong cross-polar flows.

The proposed substepping procedure has other advantages: first of all, when the proposed numerical scheme for semi-Lagrangian advection is used in an atmospheric circulation model (even the simple shallow water equations, as shown in [13, 14]), very large time steps can be used without a significant loss of accuracy. Then, the computational cost for the numerical integration of the model can be significantly reduced.

A second advantage, that we will discuss in the next section, concerns the merging of this substepping procedure with the described icosahedral grid: with this choice, the expensive tree-search of the departure point of a characteristic line is no more required, which has represented, up to now, a huge drawback for semi-Lagrangian procedures on irregular grids. Indeed, if the number N_τ of substeps is chosen under a local CFL condition, the departure point is tracked back crossing at most one cell for each substep.

Interpolation

To solve Equation (3) and also to integrate system (2) we need to evaluate variables u, v, F in locations that are, generally, internal to the triangular cells. So we need, first, to identify the cell which every departure point belongs to and, second, to evaluate the advected quantity at this location by an appropriate interpolation procedure.

The more complicated step in such a semi-Lagrangian procedure on unstructured grids seems to be the identification, for any grid point, of the triangle from which the trajectory started at the previous time step. Indeed, many of the published works on this subject accurately describe this search procedure, see e.g. Giraldo [6], where a tree-search algorithm is proposed.

However, this search is not a problem for the presented method. Here, the substepping procedure does exactly the job, since at every substep (backward in time) any Lagrangian point can only move to a neighbouring cell, due to the CFL restriction; then the algorithm easily checks in which cell the point has moved. Finally, when the procedure completed, for any grid point (arrival point) the cell containing the departure point of the related characteristic is already identified and all we need is an interpolation procedure.

While in previous models, which adopted orthogonal longitude–latitude grids, cubic interpolation was a consolidated strategy [8], in the more recent models based on icosahedral grids different interpolation choices have been performed. Then, in a short study recently accepted for publication [15] we compared the extension of several different interpolation procedures (linear, distance based and cubic schemes) to the spherical surface. For completeness, we recall a short definition of the interpolation schemes we consider here. The common starting point is the use of barycentric coordinates relative to spherical triangles, as discussed in [21], where the associated Bernstein–Bézier polynomials on the sphere are also defined

(a) We adopt, as linear interpolator for values of a function on a point P belonging to a spherical triangle T, the linear combination of the function values on the triangle vertices weighted by the normalized barycentric coordinates of point P.
(b) The RBF approximant $s(P)$ at a point P with respect to the given points $P_1, \ldots, P_h$ in R^k, is sought as [22]

$$s(P) = \sum_{i=1}^{h} w_i \Phi(\| P - P_i \|)$$

DOI: 10.1002/fld

where Φ is the RBF to be specified, $\|\cdot\|$ is the standard Euclidean norm and the weights w_i have to be computed so that the estimated function agrees with the observations at points P_i. We consider the performance of the generalization to the sphere of the Hardy's multiquadrics, for $\Phi(r)=(r^2+d^2)^{1/2}$, with r being geodesic distance and d being a tension parameter to be chosen: since we use as interpolation points the seven (six in the case of pentagonal points) grid points closest to P, we fix d of the order of the typical neighbours distance. Determination of the weights w_i then requires the solution of a linear system of size $h=7$.

(c) Theoretical considerations and numerical experiments lead us to suggest the use of Lagrange-type cubic patches for interpolation. Indeed, while the Hermite-type interpolation methods adopted in early papers [23, 24] require gradient estimates for the triangle vertices (and so any inaccuracy in these estimates strongly affects the interpolation results), Lagrange-type interpolation only requires function values. Once identified in the triangulation 10 neighbours of point P which together form a triangle T, in our proposed scheme the approximation at P is written in Bernstein–Bézier form [21] as

$$p(P)=\sum_{i+j+k=3} c_{ijk}B^3_{ijk}(P)$$

where the $B^3_{ijk}(P)$ are the homogeneous Bernstein basis polynomials of degree 3 on T

$$B^3_{ijk}(P)=\frac{3!}{i!j!k!}b_1^i b_2^j b_3^k$$

and (b_1, b_2, b_3)'s are the barycentric coordinates of P in T. The coefficients c_{ijk} have to be computed imposing interpolation at the 10 fixed neighbours, which requires the solution of a linear system. As a remark, we note that the entries of the matrix of this system only depend on the barycentric coordinates of the 10 neighbours of point P in T, that are fixed nodes of the grid. Then, the corresponding matrix can be inverted just once and the resulting coefficients stored.

Computational issues

We report in this section some considerations on the computational cost of the proposed scheme in comparison to the two main trajectory calculation methods in the literature: Ritchie's method and McGregor's method.

The method proposed by Ritchie [11] is iterative and interpolating. Giraldo [17] proved its simplification to the midpoint integration rule on the surface of a sphere:

$$\mathbf{x}_M=\mathbf{x}_A-\frac{\Delta t}{2}\mathbf{u}\left(\mathbf{x}_M, t+\frac{\Delta t}{2}\right)$$

where $\mathbf{x}_M$ and $\mathbf{x}_A$ are the midpoint and the arrival point of the trajectory, respectively. Usually, between three and five iterations are required for this recursive scheme to converge; then, the departure point $\mathbf{x}_\mathrm{D}$ of the trajectory is calculated by

$$\mathbf{x}_\mathrm{D}=\mathbf{x}_A-\Delta t\mathbf{u}\left(\mathbf{x}_M, t+\frac{\Delta t}{2}\right)$$

DOI: 10.1002/fld

Apart from the cost of the interpolation procedure, the scheme then requires three to five evaluation of the velocity for every grid point. The cost of the substepping method we propose here is comparable to the one of Ritchie's method since it requires, apart from the interpolation, one evaluation of the velocity for every substep; the number of substeps is fixed to the smallest integer greater than (or equal to) the local Courant number.

McGregor's non-interpolating method [12] for trajectory calculation in semi-Lagrangian methods was based on a Taylor series expansion of the advected field. McGregor's method has been extended to triangular grids by Giraldo [17], which overcomes the main difficulty in applying this method on unstructured grids, namely the evaluation of derivatives at the grid points, introducing an element by element construction procedure for forming these derivatives. This procedure requires, for the evaluation of any of the first partial derivatives at a grid point, the solution of a linear system which involves one elemental matrix of size three for any of the six neighbours of this point; for the second derivatives 18 neighbours are involved. For higher-order derivatives, the differencing stencil grows even further, involving 36 neighbours for the third derivatives, 60 neighbours for the fourth derivatives, and so on.

It is evident from these considerations that the cost of the whole substepping procedure, involving the determination of the nearest neighbours and the solution of a linear system of size 10 (for cubic interpolation) is definitely minor than the cost of McGregor's method already for $N = 2$ (second derivatives).

NUMERICAL EXPERIMENTS

Here, we consider two numerical experiments to test our advection scheme. These experiments are solid-body rotation of a passive scalar on the surface of the sphere and a deformation flow test for a time-dependent vortex simulated near the poles. The scalar F used in the following advection tests can be viewed as the surface pressure divided by gravity in a shallow water model, or as the pressure difference divided by gravity between the top and the bottom of a model layer in a GCM, or also as the mass per unit area of any single atmospheric component in a GCM, for example water vapour or a chemical constituent.

Solid-body rotation

The problem of solid-body rotation in (λ, θ) coordinates, as in [16] Test N.1, will be first considered. This is quite a standard test for advection schemes over the sphere. We have an initial height profile (a cosine bell) which rotates, with constant angular velocity Ω, around the Earth axis (through the poles). The initial field is given by

$$F(\lambda, \theta) = \begin{cases} 1/2[1 + \cos(\pi r/r_0)] & \text{if } r < r_0 \\ 0 & \text{if } r \geqslant r_0 \end{cases} \tag{5}$$

where r is the geodesic distance between (λ, θ) and the bell centre, initially fixed at $(3\pi/2, 0)$ and the bell radius r_0 is $R/3$; R is the Earth's radius.

We consider this rotation in a spherical coordinate system (λ, θ) having its North pole at the point P (not coinciding with the physical North pole, NP, in general).

DOI: 10.1002/fld

Then, if we indicate by α_0 the angle between the physical and the numerical North pole, so that $(0, \pi/2 - \alpha_0)$ are NP coordinates in the new system, the velocity components of the advecting wind field are given by

$$\begin{aligned} u &= u_0[\cos\theta\cos\alpha_0 + \cos\lambda\sin\theta\sin\alpha_0] \\ v &= -u_0\sin\lambda\sin\alpha_0 \end{aligned} \tag{6}$$

with $u_0 = 2\pi R/(12 \text{ days})$.

The flow field is such that when $\alpha_0 = 0$, the axis of rotation is the polar axis, and when $\alpha_0 = \pi/2$, it is in the equatorial plane. In regular longitude–latitude grids, different choices of α_0 lead to different accuracy in the results, due to the varying spatial resolution of the grid cells. On the contrary, in the geodesic grid the resolution is almost uniform, and the choice of the parameter α_0 is expected not to affect the accuracy of the approximation. However, we tested our model for $\alpha_0 = \pi/2 - 0.05$ radiants, so that the maximum of the advected field is very close to the numerical poles.

We show results of the implementation of the proposed semi-Lagrangian advection scheme endowed with the Euler scheme (4) for time integration: in such a test case, when the analytical velocity field is known and independent of time, the more complicated second-order Runge–Kutta scheme does not introduce any improvement.

Comparison of the interpolation schemes. As far as the interpolation scheme is concerned, the proposed linear interpolator, RBF scheme and cubic Lagrange interpolator has been compared on the considered test problem. In previous comparisons on a smoother test problem [15] RBFs and cubic interpolation both gave good approximation results, and also the simpler linear scheme performed fairly good; on the contrary, when applied to the present test problem, the linear scheme fails, giving excessive smoothing of the solution and also stretching in the direction of flow. Also the accumulation of the error in the RBF scheme is sensibly greater, while the cubic scheme confirms to be accurate and robust: the global relative errors on the advected field in L_1, L_2 and L_∞ norms (indexes l_1, l_2, l_∞ in [16]) have been calculated for the three interpolation schemes after 12 days of simulation, corresponding to a complete rotation of the cosine bell pattern, on a grid of 10 242 points (refinement level 5) with a time step of 120 min, corresponding to a maximum Courant number $C = 1.26$ and reported in Table I.

Comparison of the trajectory calculation methods. As a second test, we compare the performance of the proposed substepping method with respect to the two cited trajectories calculation methods:

Table I. Error indicators for the different interpolation schemes after a full solid-body rotation on a level 5 grid (10 242 points) with $\Delta t = 120'$.

	l_1	l_2	l_∞
Linear	0.5358	0.3458	0.3209
RBF	0.1157	0.1062	0.1093
Cubic	0.0292	0.0207	0.0199

DOI: 10.1002/fld

Table II. Accuracy for the midpoint rule, McGregor's scheme for $N = 1, \ldots, 4$ and the proposed scheme on Test n.1 with levels 3 (Courant = 1.13) and 4 (Courant = 2.27), after one revolution for $\alpha = 0$.

Method	l_2	M_1	l_2	M_1
Exact	0.0917	1.0071	0.0195	0.9988
Midpoint	0.1132	1.0069	0.0386	0.9988
$N = 1$	0.4743	1.0127	0.4684	1.0050
$N = 2$	0.1724	1.0064	0.1506	0.9980
$N = 3$	0.0918	1.0069	0.0210	0.9987
$N = 4$	0.0917	1.0069	0.0206	0.9987
sub	0.1058	1.0053	0.0334	1.0017

the interpolating method of Ritchie, that is equivalent, as proved in [17], to the midpoint rule on the spherical surface, and the non-interpolating method of McGregor as adapted for spherical geodesic grids by Giraldo [17]. We do exactly the same accuracy test of that paper: we fix the refinement levels 3 and 4, corresponding to 642 and 2562 grid points, respectively, and consider a complete revolution of the cosine bell pattern (12 days) with a time step of $432'$. We note that, for such grid resolutions and time step, the maximum Courant numbers are $C = 1.13$ and 2.27, respectively, corresponding to 2 and 3 substeps for our method. Then we report in Table II the results obtained with exact trajectories, with the midpoint rule, with McGregor's scheme for $N = 1, \ldots, 4$ (i.e. using partial derivatives of the advected field up to the order N) and the proposed substepping scheme. Two error indicators have been considered: the l_2 normalized error on the retrieved solution, as defined in [16], and the first moment of the conservation variable

$$M_1 = \frac{\sum F_j}{\sum F_j^{\text{exact}}}$$

where the sums extend to all the $10\, n_i^2+2$ grid points.

As Table II shows, our method performs better than McGregor's method for $N = 1$ and 2 and just slightly better than the midpoint rule. However, McGregor's method with higher-order derivatives ($N = 3$ and 4) yields the best results, but at a much higher computational cost, as discussed in the previous section.

As far as the mass conservation is concerned, even if our method is not inherently conservative, the reported values for the indicator M_1 show that all the considered methods give comparable results.

When resolution increases, the advantages of the substepping procedure over the midpoint rule are expected to be more evident, even if extensive comparisons have not been done.

Detailed solid-body results. In this section, the performance of the proposed method is examined in more details. Executions were made for the four grid resolutions corresponding to refinement levels ranging from 4 (2562 grid points) to 7 (163 842 grid points) and to spatial resolutions ranging from about 440 to 55 km, respectively.

As reported by several authors, and extensively motivated in the error analysis study by Falcone and Ferretti for SL schemes [25], the error is dominated by spatial terms: at realistic grid resolutions

DOI: 10.1002/fld

Table III. Accuracy of the proposed scheme with respect to increasing spatial resolution and fixed time step $\Delta t = 120'$ on Test n.1 after one revolution.

Level	Nodes	Courant	l_1	l_2	l_∞	M_1	CPU time ($''$)
4	2562	0.63	0.63532	0.43963	0.45911	1.054623	10.89
5	10 242	1.26	0.02922	0.02065	0.01993	0.999799	27.62
6	40 962	2.52	0.00711	0.00560	0.00599	1.000003	104.02
7	163 842	5.04	0.00150	0.00141	0.00170	1.000002	624.45

(levels 5–7) the results obtained with time steps of 60 and 120 min are almost undistinguishable. Then we fix the time step, for all resolutions, as $\Delta t = 120'$, so that the related Courant number increases from 0.63 to 5.04 when the refinement level raises.

Table III shows results corresponding to this implementation. The usual error indicators l_1, l_2, l_∞, M_1 have been reported. Refinement levels and related maximum Courant numbers are indicated, and also CPU execution times for the Fortran 90 code on a Pentium IV 3 GHz personal computer, as a partial indication of the computational cost of the whole scheme.

Figure 3 illustrates the time evolution of the l_1, l_2 and l_∞ errors during the simulation for the considered refinement levels.

Figure 4 shows on the left the cosine bell passing over the polar region after 72 h of simulation, for $\alpha = \pi/2 - 0.05$, and on the right the final solution, after 288 h of simulation. Contours of the retrieved solution (solid line) and of the true solution (dashed line) are plotted with uniform intervals of 100 m. We outline that data were interpolated from the geodesic grid to a 64×128 latitude–longitude grid for the visualization, so introducing some artefact in the highest contour. However, the true and the retrieved solution are visually undistinguishable.

First, we observe that the considered scheme, as expected, is numerically very stable also for values of the Courant number higher than one. Mass is not exactly conserved, but the relative mass variation remains very small. It must be stressed that the highest L_∞ errors are often located close to the pentagonal points, i.e. the corners of the 10 diamonds in which the grid is computationally divided. This suggests the need for a more accurate treatment of these points.

Deformation flow

As a second test problem, we consider a deformation flow in spherical geometry. Details of this vortex problem can be found in [18]: the flow field, although smooth and positive definite, is deformational and more challenging than the solid-body rotation.

Let (λ', θ') be the rotated coordinate system with the North pole at (λ_0, θ_0) with respect to the regular spherical coordinate system (λ, θ). This coordinate system can be obtained by the trigonometric relations

$$\begin{aligned} \cos\lambda' \cos\theta' &= \sin\theta_0 \cos\theta \cos(\lambda - \lambda_0) - \cos\theta_0 \sin\theta \\ \sin\lambda' \cos\theta' &= \cos\theta \sin(\lambda - \lambda_0) \\ \sin\theta' &= \cos\theta_0 \cos\theta \cos(\lambda - \lambda_0) + \sin\theta_0 \sin\theta \end{aligned} \tag{7}$$

DOI: 10.1002/fld

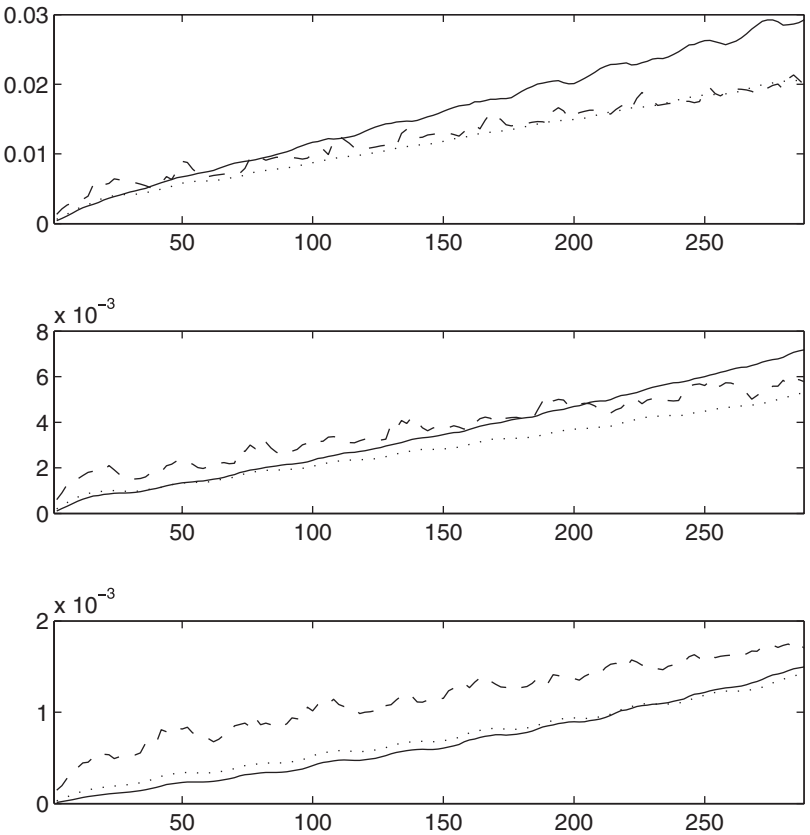

Figure 3. Time evolution of the normalized l_1 (solid line), l_2 (dotted line) and l_∞ (dashed line) errors for the solid-body advection test for refinement levels 5 (top), 6 (middle) and 7 (bottom).

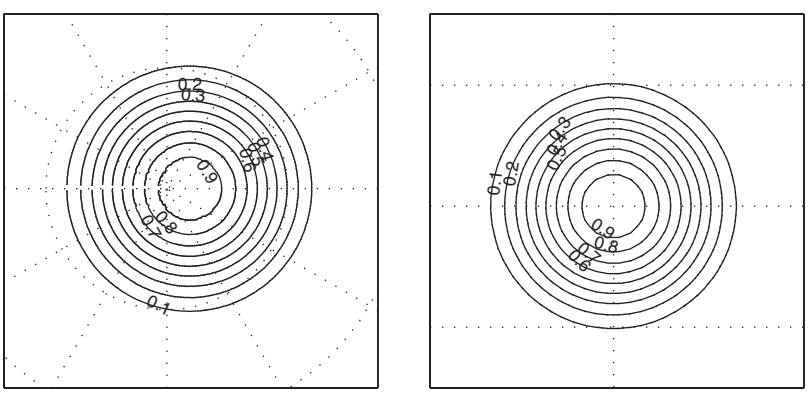

Figure 4. Orthographic projection for solid-body rotation of a cosine-bell passing over the north polar region of the sphere (left) and after a full solid-body rotation (right). The exact solution is shown as dashed contours.

DOI: 10.1002/fld

Consider the rotation of the (λ', θ') system with angular velocity ω' such that

$$\begin{aligned} \frac{\mathrm{d}\lambda'}{\mathrm{d}t} &= \omega' \\ \frac{\mathrm{d}\theta'}{\mathrm{d}t} &= 0 \end{aligned} \tag{8}$$

In this test, the angular velocity ω' varies with respect to the latitude θ'

$$\omega'(\theta') = \begin{cases} 0 & \text{if } \rho' = 0 \\ \dfrac{V_t}{\rho'} & \text{if } \rho' \neq 0 \end{cases}$$

where $\rho' = r_0 \cos(\theta')$ is the radius of the vortex, with r_0 constant, and V_t is the normalized tangential velocity of the vortex

$$V_t = \frac{3\sqrt{3}}{2} \operatorname{sech}^2(\rho') \tanh(\rho')$$

The analytical solution at time t is

$$F(\lambda', \theta', t) = 1 - \tanh\left(\frac{\rho'}{d} \sin(\lambda' - \omega' t)\right)$$

where d is the smoothness parameter for the flow field. The initial condition for the advected scalar is given by $F(\lambda', \theta', 0)$.

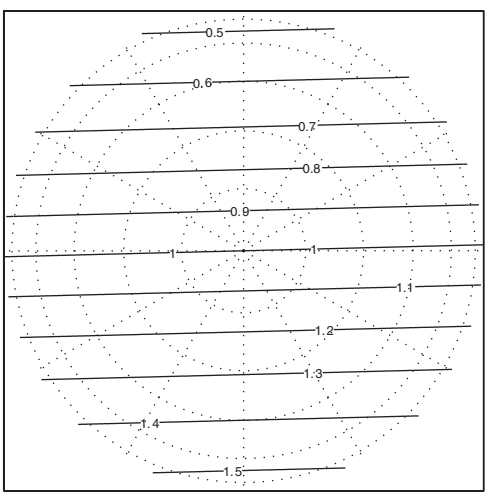

Figure 5. Orthographic projection for the contour lines of the initial field of the deformational problem.

DOI: 10.1002/fld

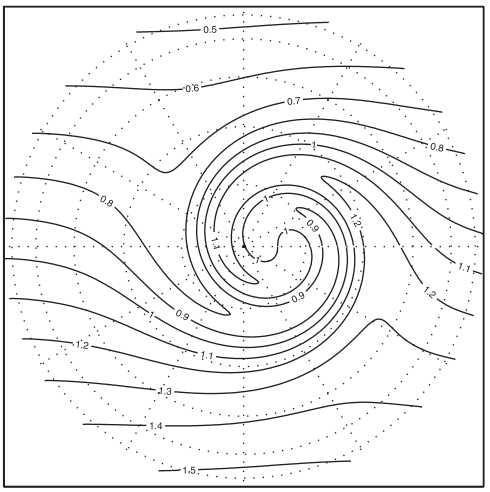

Figure 6. Orthographic projection for the contour lines of the analytical solution of the deformational problem after three time steps.

Table IV. Error indicators for the semi-Lagrangian scheme with cubic interpolation on test problem n.2 for $T = 3$ h. Time step $\Delta t = 5'$.

Level	Courant	l_1	l_2	l_∞	$\max \lvert F - F_{\text{exact}} \rvert$	$1 - M_1$	CPU time ($''$)
4	0.40	3.32e−3	1.09e−2	7.29e−2	0.1120	1.06e−8	3.34
5	0.80	1.33e−3	5.60e−3	4.03e−2	0.0619	3.70e−10	12.51
6	1.60	6.97e−4	3.06e−3	2.66e−2	0.0409	1.14e−11	56.98
7	3.21	3.59e−4	1.60e−3	1.65e−2	0.0259	7.70e−12	284.63

Differentiating Equations (7), with some algebra one can obtain the expression for the velocity components in (λ, θ) coordinates as

$$u = R\omega'[\cos\theta \sin\theta_0 - \sin\theta \cos\theta_0 \cos(\lambda - \lambda_0)]$$

$$v = R\omega' \cos\theta_0 \sin(\lambda - \lambda_0)$$

For the present study, we have set the parameters $(\lambda_0, \theta_0) = (\pi + 0.025, \pi/2.2)$, $r_0 = 3$ and $d = 5$. With these conditions two symmetric vortices are created, one near the North pole and the other near the South pole. This set-up keeps the vortex centres away from the poles to avoid symmetry.

Figures 5 and 6 show a contour plot of the initial field in stereographic polar projection and the corresponding analytical solution after three time units.

Numerical integration of the proposed semi-Lagrangian advection scheme with cubic Lagrange interpolation has been performed with a time step of about 5'(corresponding to 32 time steps, exactly as in [18]). The related Courant numbers and error indicators are reported in Table IV, where the more significant indicator $1 - M_1$ has been adopted instead of M_1; moreover, the absolute error indicator $\max \lvert F - F_{\text{exact}} \rvert$ has been introduced. As can also be seen in Figure 7, the essential

DOI: 10.1002/fld

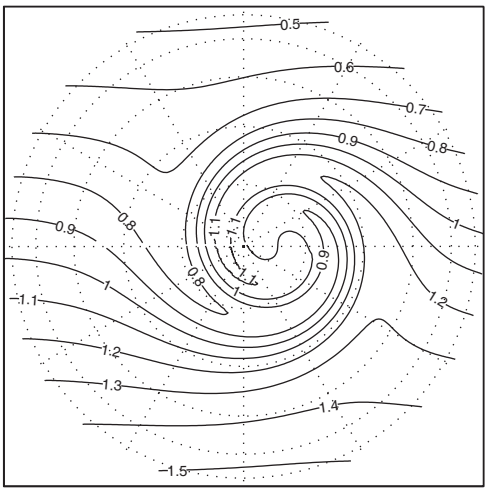

Figure 7. Orthographic projection for the contour lines of the numerical solution of the deformational problem after three time steps (refinement level 7).

details of the exact solution are represented quite well even if, as could be expected, the rate of convergence of the numerical solution is poorer than the one attained in the first test case, where the flow field was smoother.

We tried to do comparisons with the other interpolation schemes, but both linear interpolation and radial basis functions revealed to be unfit to face this more challenging test problem: at the highest resolutions (levels 6 and 7) these schemes broke down, showing very high L_∞ errors, obviously localized close to the vortex centre. Executions done with smaller time steps show little improvement.

FINAL CONSIDERATIONS

A semi-Lagrangian scheme has been presented and tested for 2D transport on the sphere. This scheme is designed on a geodesic grid which results in a quasi-uniform triangulation of the spherical surface. This choice prevents any special treatment for the polar regions; moreover, it perfectly fits the substepping procedure used for the reconstruction of the characteristics. The scheme outperforms schemes based on the traditional regular longitude–latitude grid [9] and was found to be competitive in terms of efficiency for the same resolution with other published trajectory calculation schemes on geodesic grids.

A perspective for the future is an extension of the method to the shallow water equations; between further developments, mainly concerning exact mass conservation, we intend to implement a multiresolution procedure as in [26] to optimize the computational cost of the method.

REFERENCES

1. Sadourny R, Arakawa A, Mintz Y. Integration of the nondivergent barotropic vorticity equation with an icosahedral–hexagonal grid for the sphere. *Monthly Weather Review* 1968; **96**:351–356.

DOI: 10.1002/fld

2. Williamson DL. Integration of the barotropic vorticity equation on a spherical geodesic grid. *Tellus* 1968; **20**:642–653.
3. Cullen MJP. Integration of the primitive equations on a sphere using the finite-element method. *Quarterly Journal of the Royal Meteorological Society* 1974; **100**:555–562.
4. Heikes R, Randall DA. Numerical integration of the shallow-water equations on a twisted icosahedral grid. Part I: Basic design and result of tests. *Monthly Weather Review* 1995; **123**:1862–1880.
5. Stuhne GR, Peltier WR. Vortex erosion and amalgamation in a new model of large scale flow on the sphere. *Journal of Computational Physics* 1996; **128**:58–81.
6. Giraldo FX. Lagrange–Galerkin methods on spherical Geodesic grids. *Journal of Computational Physics* 1997; **136**:197–213.
7. Majewski D, Liermann D, Prohl P, Ritter B, Buchhold M, Hanish T, Paul G, Wergen W. The operational global icosahedral–hexagonal gridpoint model GME: description and high-resolution tests. *Monthly Weather Review* 2002; **130**:319–338.
8. Staniforth A, Côté J. Semi-Lagrangian integration schemes for atmospheric models—a review. *Monthly Weather Review* 1991; **119**:2206–2223.
9. Amato U, Carfora MF. Semi-Lagrangian treatment of advection on the sphere with accurate spatial and temporal approximations. *Mathematical and Computer Modelling* 2000; **32**:981–995.
10. Casulli V. Semi-implicit finite difference methods for the two-dimensional shallow water equations. *Journal of Computational Physics* 1990; **86**:56–74.
11. Ritchie H. Semi-Lagrangian advection on a Gaussian grid. *Monthly Weather Review* 1987; **115**:608–619.
12. McGregor JL. Economical determination of the departure points for the semi-Lagrangian models. *Monthly Weather Review* 1993; **121**:221–230.
13. Carfora MF. An unconditionally stable semi-Lagrangian method for the atmospherical shallow water equations. *International Journal for Numerical Methods in Fluids* 2000; **34**(6):527–558.
14. Carfora MF. Effectiveness of the operator splitting for solving the atmospherical shallow water equations. *International Journal of Numerical Methods for Heat and Fluid Flow* 2001; **11**(3):213–226.
15. Carfora MF. Interpolation on spherical geodesic grids: a comparative study. *Journal of Computational and Applied Mathematics* 2007, in press. DOI: 10.1016/j.cam.2006.10.068.
16. Williamson DL, Drake JB, Hack JJ, Jakob R, Swarztrauber PN. A standard test set for numerical approximations to the shallow water equations in spherical geometry. *Journal of Computational Physics* 1992; **102**:211–224.
17. Giraldo FX. Trajectory calculations of spherical geodesic grids in Cartesian space. *Monthly Weather Review* 1999; **127**:1651–1662.
18. Nair R, Machenhauer B. The mass conservative cell-integrated semi-Lagrangian (CISL) advection scheme on the sphere. *Monthly Weather Review* 2002; **130**:649–667.
19. Baumgardner JR, Frederickson PO. Icosahedral discretization of the two-sphere. *SIAM Journal on Numerical Analysis* 1985; **22**:1107–1115.
20. McDonald A, Bates JR. Semi-Lagrangian integration of a gridpoint shallow water model on the sphere. *Monthly Weather Review* 1989; **117**:130–137.
21. Alfeld P, Neamtu M, Schumaker L. Bernstein–Bézier polynomials on spheres and sphere-like surfaces. *Computer Aided Geometric Design* 1996; **13**:333–349.
22. Buhmann M. Radial basis functions. *Acta Numerica* 2000; 1–38.
23. Lawson CL. C^1 surface interpolation for scattered data on a sphere. *Rocky Mountain Journal of Mathematics* 1984; **14**:177–202.
24. Renka RJ. Interpolation of data on the surface of a sphere. *ACM Transactions on Mathematical Software* 1984; **10**:417–436.
25. Falcone M, Ferretti R. Convergence analysis for a class of high-order semi-Lagrangian advection schemes. *SIAM Journal on Numerical Analysis* 1998; **35**:909–940.
26. Cohen A, Dyn N, Kaber SM, Postel M. Multiresolution schemes on triangles for scalar conservation laws. *Journal of Computational Physics* 2000; **161**:264–286.

DOI: 10.1002/fld