LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-JRNL-463840

High order finite difference and finite volume methods for convection on the sphere

B. Sjogreen

December 9, 2010

Journal of Scientific Computing

Disclaimer

High order finite difference and finite volume methods for convection on the sphere

Björn Sjögreen

December 8, 2010

Abstract

Numerical schemes used for computational climate modeling and weather prediction are often of second order accuracy. It is well-known that methods of formal order higher than two offer a significant potential gain in computational efficiency. We here present two classes of high order methods for discretization on the surface of a sphere, first finite difference schemes satisfying the summation-by-parts property on the cube sphere grid, secondly finite volume discretizations on unstructured grids with polygonial cells. Furthermore, we also implement the seventh order accurate weighted essentially non-oscillatory (WENO7) scheme for the cube sphere grid. For the finite difference approximation, we prove a stability estimate, derived from projection boundary conditions. We the finite volume method, we develop the implementational details by working in a local coordinate system. We apply the schemes to compute convection on a sphere, which is a well established test problem. We compare the performance of the methods with respect to accuracy, computational efficiency, and ability to capture discontinuities.

1 Introduction

Many mathematical models used for weather prediction or climate modeling are described by systems of partial differential equations (PDE) defined on a sphere. This paper will develop and analyze high order accurate discretizations of finite difference and finite volume type for PDEs on the sphere. To

avoid the practical complications of realistic climate models, which can be very complex, we will develop and compare the discretizations of high order for the simple case of passive advection of a scalar quantity, $\phi(t, \mathbf{x})$,

$$\phi_t + div(\phi \mathbf{u}) = 0, \quad \mathbf{x} \in S, \;\; t > 0. \tag{1}$$

The velocity field $\mathbf{u} = \mathbf{u}(t, \mathbf{x})$ is a given vector valued function, tangential to the sphere. The domain of definition is the surface of a sphere with radius r, $S = \{\mathbf{x} = (x, y, z) \,|\, x^2 + y^2 + z^2 = r^2\}$. In climate modeling computations, the radius is usually $6.37122 \times 10^6\, m$. Passive advection is a model problem for the more complicated partial differential equations used in climate prediction, and it is of interest in its own right for describing the transport of material, e.g., pollutants, in the atmosphere.

This report addresses both finite difference methods and finite volume methods. Finite difference methods are easy to define to high order of accuracy for any type of partial differential equation, they are easy to implement, and are computationally efficient. The drawback with finite difference methods is that they require a logically rectangular grid, which can be difficult to generate for complicated geometries. Finite volume methods are more flexible with respect to geometry, and can be defined on both structured and unstructured grids, making discretizations with grid refinement relatively straightforward. However, finite volume methods are difficult to implement to high order of accuracy, and are awkward to define for partial differential equations that are not naturally on conservation form.

Another popular method used for climate modeling is the spectral element method (SEM), see, e.g. [9]. SEM have many similarities with the finite difference discretization by summation-by-parts operators described below. In the SEM, the weak form of the PDE is discretized directly. The discrete weak form therefore satisfies a summation-by-parts identity, obtained as the discretized integration-by-parts identity used to derive the weak form.

Element based methods, such as the finite element method (FEM) or discontinuous Galerkin methods (DG) are well suited for unstructured grids. FEM and DG methods have excellent geometrical flexibility, but are less computationally efficient than finite difference methods or the spectral element method.

In the next section, we will develop high order finite difference discretizations for the sphere, S, using the cubed-sphere approach, see, e.g., [7]. We will use both purely centered schemes, and the weighted essentially non-oscillatory scheme (WENO) for capturing of steep gradients. In section 4,

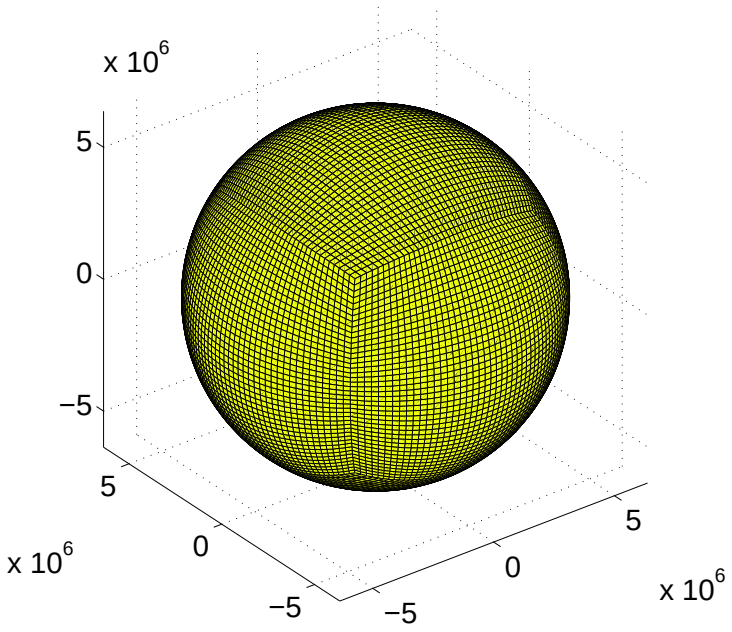

Figure 1: Cubed-sphere grid.

we develop a third and fourth order accurate finite volume discretizations for the same domain, S, using an unstructured grid of hexagons. A additional complication compared with the finite volume method in the plane is that, for high order methods on the sphere, it is necessary to take into account the curvature of the cells. Finally, Section 5 will give some numerical comparisons of the methods when applied to a simple vortex convection problem.

2 The Equation on the cube-sphere grid

We will first consider S discretized by six adjacent curvilinear grid patches, a so called cube sphere grid. The patches have the same number of grid points, and the grid points coincides on the patch boundaries, thus there are no hanging nodes. Figure 1 shows an example grid.

Each grid patch is a mapping from a square to a patch of the sphere. In

general we consider mappings from $(x, y, z) \in R^3$ to $(r, \xi, \eta) \in R^3$ given by

$$x = r\, a(\xi, \eta) \tag{2}$$
$$y = r\, b(\xi, \eta) \tag{3}$$
$$z = r\, c(\xi, \eta) \tag{4}$$

where a, b, c is a mapping from the square $0 \le \xi \le 1$, $0 \le \eta \le 1$ to a domain on the surface of the unit sphere, $a^2 + b^2 + c^2 = 1$. The domain on the surface of a sphere with radius r_0 is the level surface $r = r_0$. The mapping transforms the Cartesian representation of a vector, $\mathbf{f}$, $(f^{(x)},\ f^{(y)},\ f^{(z)})$ into the curvilinear components $(f^{(r)},\ f^{(\xi)},\ f^{(\eta)})$ according to

$$\begin{aligned}
f^{(x)} &= \frac{1}{r} a f^{(r)} + \frac{a_\xi}{\sqrt{a_\xi^2 + b_\xi^2 + c_\xi^2}} f^{(\xi)} + \frac{a_\eta}{\sqrt{a_\eta^2 + b_\eta^2 + c_\eta^2}} f^{(\eta)} \\
f^{(y)} &= \frac{1}{r} b f^{(r)} + \frac{b_\xi}{\sqrt{a_\xi^2 + b_\xi^2 + c_\xi^2}} f^{(\xi)} + \frac{b_\eta}{\sqrt{a_\eta^2 + b_\eta^2 + c_\eta^2}} f^{(\eta)} \\
f^{(z)} &= \frac{1}{r} c f^{(r)} + \frac{c_\xi}{\sqrt{a_\xi^2 + b_\xi^2 + c_\xi^2}} f^{(\xi)} + \frac{c_\eta}{\sqrt{a_\eta^2 + b_\eta^2 + c_\eta^2}} f^{(\eta)}
\end{aligned} \tag{5}$$

The divergence in the mapped coordinate is

$$div(\mathbf{f}) = \frac{1}{r^2}(r^2 f^{(r)})_r + \frac{1}{J}\left(\frac{J}{r}\frac{1}{\sqrt{a_\xi^2 + b_\xi^2 + c_\xi^2}} f^{(\xi)}\right)_\xi + \frac{1}{J}\left(\frac{J}{r}\frac{1}{\sqrt{a_\eta^2 + b_\eta^2 + c_\eta^2}} f^{(\eta)}\right)_\eta$$

where J is the determinant of the Jacobian of the mapping,

$$\begin{pmatrix} x_r & y_r & z_r \\ x_\xi & y_\xi & z_\xi \\ x_\eta & y_\eta & z_\eta \end{pmatrix} = \begin{pmatrix} a & b & c \\ r a_\xi & r b_\xi & r c_\xi \\ r a_\eta & r b_\eta & r c_\eta \end{pmatrix}. \tag{6}$$

Obviously, $J = r^2 d(\xi, \eta)$, where $d(\xi, \eta)$ is the determinant of the Jacobian with $r = 1$.

The PDE to solve on each grid patch of the sphere is hence,

$$(J\phi)_t + \left(\frac{J}{r}\frac{1}{\sqrt{a_\xi^2 + b_\xi^2 + c_\xi^2}}\phi u^{(\xi)}\right)_\xi + \left(\frac{J}{r}\frac{1}{\sqrt{a_\eta^2 + b_\eta^2 + c_\eta^2}}\phi u^{(\eta)}\right)_\eta = 0, \tag{7}$$

because the given velocity field will always have $u^{(r)} = 0$.

It is advantageous to rewrite (1) on skew-symmetric form,

$$\phi_t + \frac{1}{2} div(\phi \mathbf{u}) + \frac{1}{2}(\nabla \phi)^T \mathbf{u} + \frac{1}{2}\phi div \mathbf{u} = 0, \tag{8}$$

before discretization. Multiplication of (8) by ϕ and integration over a domain $\Omega \in R^3$, lead to the estimate

$$\begin{aligned} \frac{d}{dt}||\phi||^2 = 2(\phi, \phi_t) = -(\phi, div(\phi \mathbf{u})) - (\phi, (\nabla \phi)^T \mathbf{u}) - (\phi, \phi div \mathbf{u}) \\ = B.T - (\phi^2, div \mathbf{u}) \end{aligned} \tag{9}$$

by use of the divergence theorem. "B.T" denotes boundary terms and the scalar product over the domain is defined by

$$(u, v) = \int_\Omega uv \, dx \, dy \, dz$$

with corresponding norm $||u||^2 = (u, u)$. If the given velocity field is divergence free, and the boundary conditions are such that the boundary terms are zero, then the norm estimate

$$||\phi(t)|| = ||\phi(0)|| \tag{10}$$

follows.

3 Discretization by centered finite differences

The grid on a patch is obtained by discretizing, $\xi_i = (i-1)h$, $i = 1, 2, \ldots, N$ and $\eta_j = (j-1)h$, $j = 1, 2, \ldots, N$ where the number of points N and the step size h are related by $(N-1)h = 1$. All patches will have the same h and N. An analytic mapping, see [7], generates the grid on patch p,

$$(x_{p,i,j}, y_{p,i,j}, z_{p,i,j}), \quad p = 1, \ldots, 6, \quad i = 1, \ldots, N \;\; j = 1, \ldots, N.$$

We did not attempt to smooth, or otherwise improve the grid. However it is important for the discretization technique that we will use, that all six grid patches have right handed coordinate systems. Let $\phi_{p,i,j}(t)$ denote the solution at point (i, j) at time t on grid patch p, $p = 1, \ldots, 6$. When time

is discrete we let $\phi^n_{p,i,j}$ denote the solution at (i,j) at time $t_n = n\Delta t$. The patch index, p, will be left out when it is clear from the context.

The estimate (10) can be done for a finite difference discretization if the discretization is done by operators that satisfy the summation-by-parts (SBP) identity. In one space dimension, for grid functions u_j and v_j defined on $j = 1, 2, \ldots, N$, the summation-by-parts identity is

$$(u, Dv)_{1,h} = -(Du, v)_{1,h} - u_1 v_1 + u_N v_N \tag{11}$$

where the difference operator, D, is an even order centered operator away from the boundaries, and modified near boundaries to satisfy (11). The discrete scalar product is defined as

$$(u, v)_{1,h} = h \sum_{j=1}^{N} \sigma_j u_j v_j$$

where the positive weights σ_j are different from one near the boundaries. The boundary modification leads to some reduction in the formal order of accuracy at the boundaries. Derivation of SBP operators and scalar products with boundary orders/interior orders 1/2, 2/4, 3/6, and 4/8 can be found in [8]. The simplest example is the first order boundary/second order interior operator, defined as

$$Du_j = \begin{cases} (u_{j+1} - u_j)/h & j = 1 \\ (u_{j+1} - u_{j-1})/2h & 2 \le j \le N-1 \\ (u_j - u_{j-1})/h & j = N \end{cases}, \tag{12}$$

with $\sigma_1 = \sigma_N = 1/2$ and $\sigma_j = 1$ for $2 \le j \le N-1$.

The semi-discrete approximation of the skew symmetric form of the equations in transformed coordinates is

$$\begin{aligned} J_{i,j} \frac{d}{dt} \phi_{i,j}(t) = -\frac{1}{2}(D_I(p_{i,j}\phi_{i,j}u^{(\xi)}_{i,j}) + D_J(q_{i,j}\phi_{i,j}u^{(\eta)}_{i,j})) - \\ \frac{1}{2}(D_I(p_{i,j}u^{(\xi)}_{i,j}) + D_J(q_{i,j}u^{(\eta)}_{i,j}))\phi_{i,j} - \\ \frac{1}{2}(p_{i,j}u^{(\xi)}_{i,j} D_I\phi_{i,j} + q_{i,j}u^{(\eta)}_{i,j} D_J\phi_{i,j}), \end{aligned} \tag{13}$$

where the metric coefficients are evaluated from the given grid as

$$p_{i,j} = J_{i,j} / \sqrt{(D_I x_{i,j})^2 + (D_I y_{i,j})^2 + (D_I z_{i,j})^2} \tag{14}$$

$$q_{i,j} = J_{i,j} / \sqrt{(D_J x_{i,j})^2 + (D_J y_{i,j})^2 + (D_J z_{i,j})^2} \tag{15}$$

The difference operators D_I and D_J are SBP operators that act in the i- or j-indices respectively. $J_{i,j}$ is computed as the determinant of the matrix (6) with derivatives evaluated numerically by D_I and D_J. Similarly (5), with metric derivatives approximated by D_I and D_J, gives the transformed velocity components, $u^{(\xi)}$ and $u^{(\eta)}$, from given Cartesian velocity components.

The skew-splitting can be done in different ways, depending on how the three factors in the conservative derivatives of (13) are split. The approximation (13) is made such that it is possible to carry over the estimate (10) to the numerical approximation.

In order to do this, we define the two discrete scalar products over grid patch p, $p = 1, \ldots, 6$,

$$(u, v)_{hJ,p} = \sum_{j=1}^{N} \sum_{i=1}^{N} \sigma_i \sigma_j u_{p,i,j} v_{p,i,j} J_{p,i,j} h^2$$

and

$$(u, v)_{h,p} = \sum_{j=1}^{N} \sum_{i=1}^{N} \sigma_i \sigma_j u_{p,i,j} v_{p,i,j} h^2.$$

The scalar product over the entire sphere is the sum over the six grid patches,

$$(u, v)_{hJ} = \sum_{p=1}^{6} (u, v)_{hJ,p} \qquad (u, v)_h = \sum_{p=1}^{6} (u, v)_{h,p}$$

In the SBP framework, boundary conditions consists of first applying the one sided operators at all points, and then project the solution onto the space where boundary conditions are satisfied, see [6]. For example, assume that the boundary point (i, j) in one patch coincides with the the boundary point (k, l) in another patch. Let ϕ' be the solution obtained after taking one time step (or Runge-Kutta stage), before boundary conditions are imposed. The projection boundary conditions consist in updating

$$\phi_{i,j} = (\phi'_{i,j} + \phi'_{k,l})/2 \tag{16}$$

$$\phi_{k,l} = (\phi'_{i,j} + \phi'_{k,l})/2 \tag{17}$$

At the corner points where three patches meet, the same condition but as an average over the three values is imposed.

Theorem 1 *Let the interface conditions be given by (16) and (17) (and similarly at corners where three patches meet). Then the estimate*

$$\frac{d}{dt}||\phi(t)||^2_{Jh} = -(\phi(t)^2, D_I(pu^{(\xi)}) + D_J(qu^{(\eta)}))_h \tag{18}$$

holds.

Proof: On a single patch, take the scalar product of (13) by $\phi_{i,j}(t)$ and use the summation-by-parts identity to obtain

$$\begin{aligned}
&\frac{d}{dt}||\phi(t)||^2_{Jh,r} = \\
&\quad -(\phi(t), D_I(p\phi u^{(\xi)}))_{h,r} - (\phi(t), D_I(pu^{(\xi)})\phi(t))_{h,r} - (\phi(t), pu^{(\xi)} D_I\phi(t))_{h,r} \\
&-(\phi(t), D_J(q\phi u^{(\eta)}))_{h,r} - (\phi(t), D_J(qu^{(\eta)})\phi(t))_{h,r} - (\phi(t), qu^{(\eta)} D_J\phi(t))_{h,r} = \\
&-(\phi(t)^2, D_I(pu^{(\xi)}) + D_J(qu^{(\eta)}))_{h,r} - \sum_{j=1}^{N} \sigma_j(\phi(t)^2_{N,j} p_{N,j} u^{(\xi)}_{N,j} - \phi(t)^2_{1,j} p_{1,j} u^{(\xi)}_{1,j})h \\
&\qquad - \sum_{i=1}^{N} \sigma_i(\phi(t)^2_{i,N} q_{i,N} u^{(\eta)}_{i,N} - \phi(t)^2_{i,1} q_{i,1} u^{(\eta)}_{i,1})h.
\end{aligned} \tag{19}$$

Next, taking the sum over the six patches leads to

$$\frac{d}{dt}||\phi(t)||^2_{Jh} = -(\phi(t)^2, D_I(pu^{(\xi)}) + D_J(qu^{(\eta)}))_h + B.$$

Where the boundary terms B consists of sums over the edges, i.e., the last two sums of (19) from the six different patches. The terms in B can be ordered in pairs corresponding to the same boundary location. For example, if (N, j) and $(1, k)$ are identical locations on the sphere but on different patches, we study the pair

$$\phi(t)^2_{N,j} q_{N,j} u^{(\xi)}_{N,j} - \phi(t)^2_{1,k} q_{1,k} u^{(\xi)}_{1,k}.$$

It follows from the general theory in [6], that ϕ here satisfies the interface conditions, so that $\phi_{N,j} = \phi_{1,k}$. Therefore, $B = 0$ if

$$q_{N,j} u^{(\xi)}_{N,j} = q_{1,k} u^{(\xi)}_{1,k}. \tag{20}$$

To see why this is true, note that the Cartesian coordinates x, y, z are continuous across the patch interface, as is its derivatives along the interface, x_η, y_η, z_η. Take the scalar product between the cross product vector

$\mathbf{v} = (x\ y\ z) \times (x_\eta\ y_\eta\ z_\eta)$ and the equations (5) to obtain

$$pu^{(\xi)} = \mathbf{v} \cdot (f^{(x)}\ f^{(y)}\ f^{(z)}) - \mathbf{v} \cdot (a_\eta\ b_\eta\ c_\eta) \frac{u^{(\eta)}}{\sqrt{a_\eta^2 + b_\eta^2 + c_\eta^2}}$$

Note that $J = -(x_\xi\, y_\xi\, z_\xi) \cdot \mathbf{v}$. It is obvious that the right hand side only depend on the Cartesian coordinates and its derivatives with respect to η. Therefore also the left hand side is continuous across the patch interface and (20) follows. The description here is valid also for the discrete problem, since is unchanged if the derivatives with respect to ξ and η are replace by the finite difference operators D_I and D_J respectively. Finally, (20) applied at all interface points implies that $B = 0$, and the estimate (18) follows. □

Note that if the discretization of the velocity field is perfectly divergence free, the norm over the entire sphere surface of the solution is conserved in time.

3.1 WENO finite difference schemes

Equation (7) is a conservation law in two space dimensions with ξ-direction flux

$$f^{(\xi)} = \frac{J}{r} \frac{1}{\sqrt{a_\xi^2 + b_\xi^2 + c_\xi^2}} \phi u^{(\xi)} = p u^{(\xi)} \phi.$$

and a similar expression for the η direction flux. The WENO discretization of (7) is of flux difference form where the fluxes are based on the splitting $f_i^{(\xi)} = f_i^{(\xi)+} + f_i^{(\xi)-}$ with

$$f_i^{(\xi)+} = p_i u_i^{(\xi)+} \phi_i \quad f_i^{(\xi)-} = p_i u_i^{(\xi)-} \phi_i$$

where $u^{(\xi)-} = \min(0, u^{(\xi)})$ and $u^{(\xi)+} = \max(0, u^{(\xi)})$, and p_i is the metric coefficient from (14). The η-direction flux is similar. The flux difference $h_{i+1/2} - h_{i-1/2}$ approximates the flux derivative. The numerical fluxes are of the split form

$$h_{i+1/2} = h_{i+1/2}^+ + h_{i+1/2}^-$$

where $h_{i+1/2}^+$ is a weighted sum of flux stencils of $f_i^{(\xi)+}$ and $h_{i+1/2}^-$ is a weighted sum of stencils of $f_{i+1}^{(\xi)-}$. The weighting is solution adaptive and is designed to give more weight to stencils that cover smooth parts of the solution and less

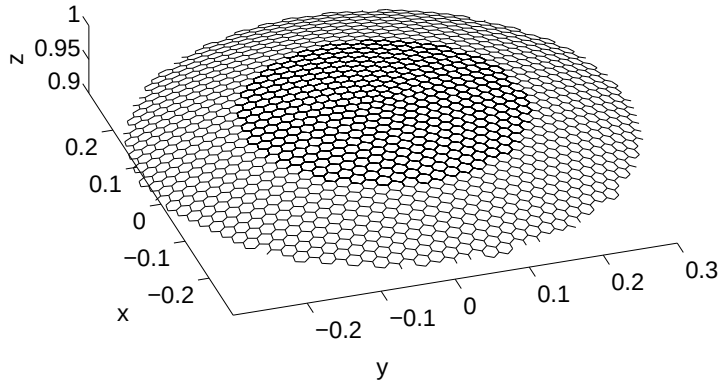

Figure 2: Unstructured grid on the sphere, only the top part of the grid is displayed.

weight to stencils that intersects discontinuities. When all stencils equally smooth the weights adapt to give optimal order of accuracy. The $2r-1$th order WENO scheme uses weighted sums of r rth order accurate flux stencils. For a complete description, see [5]. We have here implemented the case $r = 4$, thus making the scheme a sum of 4th order stencils, with an seventh order accuracy when the solution is smooth.

On the cube sphere grid, the WENO scheme is biased away from the boundary, by forcing the weights of stencils that would require points outside the domain to zero. This leads to local third order accuracy on the patch boundaries.

4 Finite volume discretization on an unstructured grid

We are given a decomposition of the surface of the sphere, S, into cells, A_i,

$$S = \cup_{i=1}^{N} A_i,$$

where the cells have disjoint interiors and their boundaries, ∂A_i, are piecewise circular arcs. The number of sides can be different for different cells. Figure 2 shows the upper part of an example grid.

Finite volume schemes for advection on the sphere are obtained by discretizing the integral form of (1),

$$\frac{d}{dt}\frac{1}{A}\int_A \phi\, dS + \frac{1}{A}\int_{\partial A} \phi \mathbf{u}\cdot\mathbf{n}\, ds = 0. \tag{21}$$

where A is a subset of S and $\mathbf{n}$ is normal to the boundary of A and tangential to S. s is a curve parameter along the cell boundary. (21) can be derived by integration of (1) in three-dimensional space over a cell between two spherical shells, use of Gauss' theorem, and letting the distance between the spherical shells go to zero.

The cell average of a function ϕ over cell A_i, $\overline{\phi_i}$, and the area of the cell, $|A_i|$, are defined by

$$\overline{\phi_i}(t) = \frac{1}{|A_i|}\int_{A_i} \phi\, dS \quad \text{and} \quad |A_i| = \int_{A_i} dS$$

respectively.

The finite volume scheme stores the cell averages as unknown variables, the semi-discrete approximation being

$$\frac{d}{dt}\overline{\phi_i}(t) = -\frac{1}{|A_i|}\sum_{j\in N(i)} g_{i,j}$$

where $g_{i,j}$ is numerical flux between cells A_i and A_j. $N(i)$ is the set of neighbors of A_i.

The finite volume approximation will reconstruct the given cell averages to a piecewise polynomial representation $R(\mathbf{x}, \overline{\phi})$, such that R is a polynomial for $\mathbf{x} \in A_i$, with possible discontinuities over ∂A_i. R is constructed to satisfy the conservation property

$$\overline{\phi_j} = \frac{1}{|A_j|}\int_{A_j} R(\mathbf{x}, \overline{\phi})\, dS \quad j \in T(i)$$

and the rth order accuracy condition

$$R(\mathbf{x}, \overline{\phi}) = \phi(\mathbf{x}) + \mathcal{O}(\Delta^r),$$

where $T(i)$ is the set of cells forming the stencil at i and Δ is a typical cell linear size. On an edge directed from $\mathbf{x}_1$ to $\mathbf{x}_2$ will denote the limit values of

the reconstructed polynomials from the left and the right sides, by ϕ_L and ϕ_R respectively.

The side between the cells is the circular arc between the two side end points on the sphere, say $\mathbf{x}_1$ and $\mathbf{x}_2$. It follows from (21) that $g_{i,j}$ is an approximation of

$$\int_{\mathbf{x}_1}^{\mathbf{x}_2} \phi(\mathbf{u}\cdot\mathbf{n})\, ds. \tag{22}$$

The parametrization of the edge,

$$\mathbf{x}(\beta) = (\sin(\alpha_{12}-\beta)\mathbf{x}_1 + \sin\beta\mathbf{x}_2)/\sin\alpha_{12} \quad 0\le\beta\le\alpha_{12} \tag{23}$$

where $\cos\alpha_{12} = (\mathbf{x}_1\cdot\mathbf{x}_2)/r^2$, gives

$$\int_{\mathbf{x}_1}^{\mathbf{x}_2} \phi\mathbf{u}\cdot\mathbf{n}\, ds = \int_0^{\alpha_{12}} \phi(t,\mathbf{x}(\beta))(\mathbf{u}(t,\mathbf{x}(\beta))\cdot\mathbf{n})|\mathbf{x}'(\beta)|\, d\beta.$$

It is straightforward to verify that $|\mathbf{x}(\beta)| = r$ for any β and that the unit normal vector is $\mathbf{n} = (\mathbf{x}_1\times\mathbf{x}_2)/r^2$. Furthermore, it follows from (23) that $|\mathbf{x}'(\beta)| = r$. We define the numerical flux by approximating the parametrized integral by the two point Gaussian numerical quadrature formula, and obtain

$$g_{i,j} = \frac{\alpha_{12}}{2}\left(g(\phi_L(t,\mathbf{x}(\beta^-)),\phi_R(t,\mathbf{x}(\beta^-))) + g(\phi_L(t,\mathbf{x}(\beta^+)),\phi_R(t,\mathbf{x}(\beta^+)))\right). \tag{24}$$

The first order accurate numerical flux function, $g(\phi_L,\phi_R)$, satisfies the consistency condition

$$g(\phi,\phi) = r\phi(\mathbf{u}\cdot\mathbf{n}).$$

The Gaussian points are

$$\beta^- = \frac{\alpha_{12}}{2}(1-\frac{1}{\sqrt{3}}) \quad \beta^+ = \frac{\alpha_{12}}{2}(1+\frac{1}{\sqrt{3}}).$$

The numerical experiments in Sec. 5 will use the upwind flux

$$g(\phi_L,\phi_R) = \begin{cases} r\phi_L(\mathbf{u}\cdot\mathbf{n}) & (\mathbf{u}\cdot\mathbf{n}) > 0 \\ r\phi_R(\mathbf{u}\cdot\mathbf{n}) & (\mathbf{u}\cdot\mathbf{n}) < 0 \end{cases}.$$

The velocities $\mathbf{u}$ are here assumed to be given functions that can be evaluated anywhere. In a realistic climate simulation, these are computed by auxiliary

differential equations and it might then be necessary to interpolate or to reconstruct the velocity to the Gauss points as well.

The reconstruction algorithm is based on the following local parametrization of S at the cell i,

$$\mathbf{x}(u,v) = \frac{r}{|\hat{\mathbf{x}}(u,v)|}\hat{\mathbf{x}}(u,v)$$

where $\hat{\mathbf{x}}(u,v)$ is the tangent plane at the cell center, $\mathbf{x}_i$,

$$\hat{\mathbf{x}}(u,v) = \mathbf{x}_i + u\mathbf{e}_u + v\mathbf{e}_v.$$

Thus $|\mathbf{x}_i| = r$, and $\hat{\mathbf{x}}(0,0) = \mathbf{x}_i$. $\mathbf{e}_u$ is chosen arbitrarily as a unit vector in the tangent plane at $\mathbf{x}_i$, and $\mathbf{e}_v = \mathbf{x}_i \times \mathbf{e}_u / r$. The inverse mapping is given by

$$u = (\mathbf{e}_u \cdot \mathbf{x})\frac{r^2}{\mathbf{x}\cdot\mathbf{x}_i} \quad v = (\mathbf{e}_v \cdot \mathbf{x})\frac{r^2}{\mathbf{x}\cdot\mathbf{x}_i}$$

The integral over a spherical polygon, A, is transformed to the integral over a polygon, B, in the (u,v) coordinates by

$$\int_A f(\mathbf{x})\,dA = \int_B f(\mathbf{x}(u,v))\left|\frac{\partial \mathbf{x}}{\partial u} \times \frac{\partial \mathbf{x}}{\partial v}\right| du\,dv.$$

where is straightforward to verify that

$$\left|\frac{\partial \mathbf{x}}{\partial u} \times \frac{\partial \mathbf{x}}{\partial v}\right| = \frac{r^3}{|\hat{\mathbf{x}}(u,v)|^3},$$

because the basis vectors $\mathbf{x}_i$, $\mathbf{e}_u$, and $\mathbf{e}_v$ are orthogonal.

We determine the reconstruction polynomial for the cell i as a function of u and v. For third order of accuracy, we use the a quadratic polynomial,

$$R_i(u,v) = c_{00} + c_{10}u + c_{01}v + c_{20}u^2 + c_{11}uv + c_{02}v^2.$$

and a similar cubic polynomial for fourth order. The polynomial is required to satisfy

$$\bar{\phi}_j = \frac{1}{A_j}\int_j R_i\,dA = \frac{1}{A_j}\int_{B_j} R_i(u,v)\frac{r^3}{|\hat{\mathbf{x}}(u,v)|^3}\,du\,dv$$

for $j \in T(i)$. This leads to the linear system of equations

$$\sum_{k,l} a_{j,kl}c_{kl} = A_j\bar{\phi}_j, \;\; j \in T(i) \tag{25}$$

for c_{kl}, with coefficients

$$a_{j,kl} = \int_{B_j} u^k v^l \frac{r^3}{|\hat{\mathbf{x}}(u,v)|^3}\, du\, dv. \tag{26}$$

The same reasoning as given in [3] shows that the point values are reconstructed to rth order accuracy by a polynomial of degree $r-1$.

Denote $\mathbf{v} = (u\ v)$ and apply the numerical quadrature formula for a triangle, T,

$$\int_T f(\mathbf{v})\, du\, dv = \frac{|T|}{60}(3\sum_{i=3}^{3} f(\mathbf{v}_i) + 8 \sum_{1\le i<j\le 3} f(\mathbf{v}_{ij}) + 27 f(\mathbf{v}_{123})), \tag{27}$$

in the u-v plane to evaluate the coefficients (26). Here $\mathbf{v}_i$ are the corners of the triangle, $\mathbf{v}_{ij}$ is the midpoint on the side from $\mathbf{v}_i$ to $\mathbf{v}_j$, and $\mathbf{v}_{123}$ is the barycenter. (27) is exact for polynomials of degree three and lower, see [1]. The polygonal cells are decomposed into sub triangles by joining the vertices and the barycenter, and (27) is applied to each sub triangle.

To summarize, the finite volume approximation consists by the following steps

1. Determine the stencil $T(i)$
2. Compute $a_{j,kl}$ in (25) by use of (27)
3. Solve (25) for c_{kl}, which determines the reconstruction polynomial.
4. Compute the numerical flux from (24) by evaluating the reconstruction polynomial at the Gaussian points.

For the third order method, the stencil $T(i)$ at point i contains the cell i itself and five other cells to determine the six coefficients in the third order reconstruction polynomial. For fourth order of accuracy, the stencil must have at least ten cells. Because the grid used consists of spherical hexagons and pentagons, the third order polynomial is conveniently determined by using all direct neighbors of pentagons, and all direct neighbors except one of hexagons. In the numerical experiments below, the most downwind cell is excluded in the case of hexagons, where the downwindness of cell j from the center cell i is determined by the size of $\mathbf{u}\cdot\mathbf{d}/|\mathbf{u}|$, where $\mathbf{u}$ is the velocity

vector at the cell center of i and $\mathbf{d}$ the vector from the cell center of j to the cell center of i. This computation is done in the projected $u-v$ coordinates.

Other possibilities for stencil selection are a) to use the stencil that gives the reconstruction polynomial with smallest variation in the cell, or b) to use all possible neighbors and second neighbors and solve (25) as an overdetermined system by the least squares method, or c) to use the WENO idea by building the stencil as a weighted average lower order stencils, as described in [4, 11]. All these methods can easily be included in the algorithm above by modifying the determination of $T(i)$ in step 1.

5 Numerical experiments

5.1 Finite difference schemes

We solve the standard test example of a vortex convected on the surface of a sphere, described in, e.g., [2]. The velocity field is divergence free, and corresponds to a wind rotating around an axis with a constant angle α between the polar axis and the axis of rotation. The stream function of the velocity is given by

$$\psi = -u_0 z\cos\alpha + u_0 x\sin\alpha = u_0 r(c\cos\alpha - a\sin\alpha)$$

from which the velocity components on the curvilinear grid are obtained as

$$u^{(\xi)} = -\frac{r\sqrt{a_\xi^2+b_\xi^2+c_\xi^2}}{J}\frac{\partial\psi}{\partial\eta} \tag{28}$$

$$u^{(\eta)} = \frac{r\sqrt{a_\eta^2+b_\eta^2+c_\eta^2}}{J}\frac{\partial\psi}{\partial\xi} \tag{29}$$

We define the discretized velocities as

$$u_{i,j}^{(\xi)} = -\frac{1}{p_{i,j}}D_J\psi_{i,j} \tag{30}$$

$$u_{i,j}^{(\eta)} = \frac{1}{q_{i,j}}D_I\psi_{i,j} \tag{31}$$

$$\tag{32}$$

to obtain zero discrete divergence,

$$D_I(p_{i,j}u_{i,j}^{(\xi)}) + D_J(q_{i,j}u_{i,j}^{(\eta)}) = -D_I(D_J\psi_{i,j}) + D_J(D_I\psi_{i,j}) = 0.$$

accuracy	L^∞-error with $N = 45$	L^∞-error with $N = 89$	order
1/2	0.41	0.12	1.78
2/4	0.0302	0.0071	2.09
3/6	0.0125	0.0028	2.16
4/8	0.0198	0.0034	2.54

Table 1: Errors in the numerical solution after one period for methods of different formal accuracy

We solve the problem on a sphere with $r = 6.37122 \times 10^6\, m$ with diagonal wind, $\alpha = \pi/4$, and with $u_0 = 2\pi r/(12 \cdot 24 \cdot 3600)\, ms^{-1}$, which means that the period of the flow is 12 days. The standard test example has initial data

$$\phi(\theta, \lambda) = \begin{cases} \frac{h_0}{2}(1 + \cos \pi s) & s < 1 \\ 0 & s \geq 1 \end{cases} \tag{33}$$

where $s = 3 \arccos(\sin\theta_c \sin\theta + \cos\theta_c \cos\theta \sin(\lambda - \lambda_c))$, and $h_0 = 1000\, m$. θ_c, λ_c is a fixed starting position. The data is given in spherical coordinates θ, λ. At a given grid point, $(x_{i,j}, y_{i,j}, z_{i,j})$, we compute θ, λ by inverting the spherical coordinate transformation

$$\begin{cases} x = r \cos\lambda \cos\theta \\ y = r \sin\lambda \cos\theta \\ z = r \sin\theta \end{cases} .$$

Figure 3 shows the initial data.

To assess the error in the computation, Fig. 4 shows the norm of the error as function of time for the four different methods with boundary accuracy/interior accuracy 1/2, 2/4, 3/6, and 4/8 on a grid with $N = 45$ (corresponding to the case c44 in [7]). All computations used the standard fourth order Runge-Kutta scheme with CFL number 0.8. The norms are normalized as suggested in [2]. Similarly, Fig. 5 shows the same quantities as Fig. 4, but with $N = 89$.

Table 1 displays the errors after one period with the different accuracy schemes. The observed convergence order between the coarse and fine grids, shown in the last column of Table 1 indicates all methods converges approximately with second order of accuracy. The error is, however, smaller for the

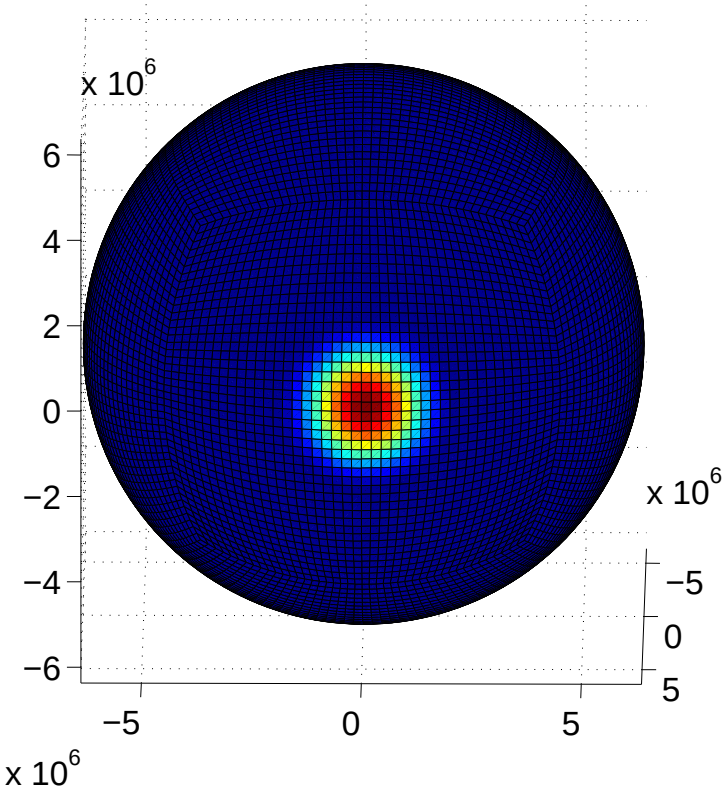

Figure 3: Cosine bell initial data.

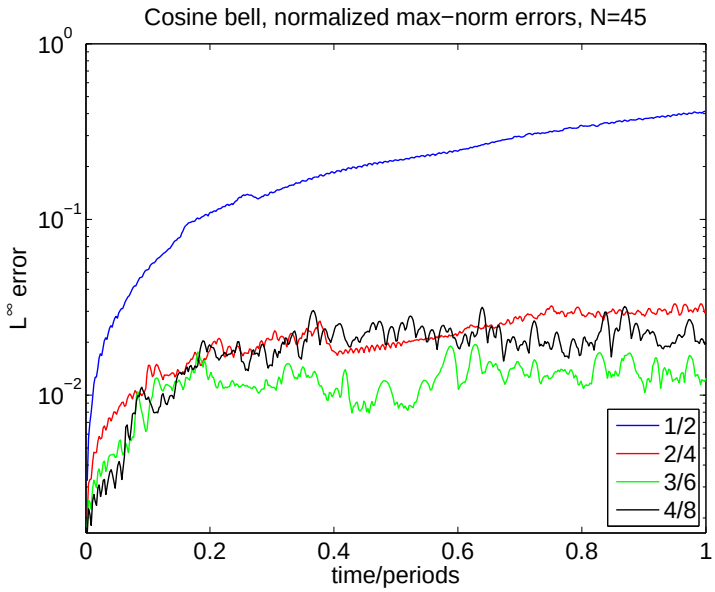

Figure 4: Error vs. time with cosine bell initial data, $N = 45$, 1 period.

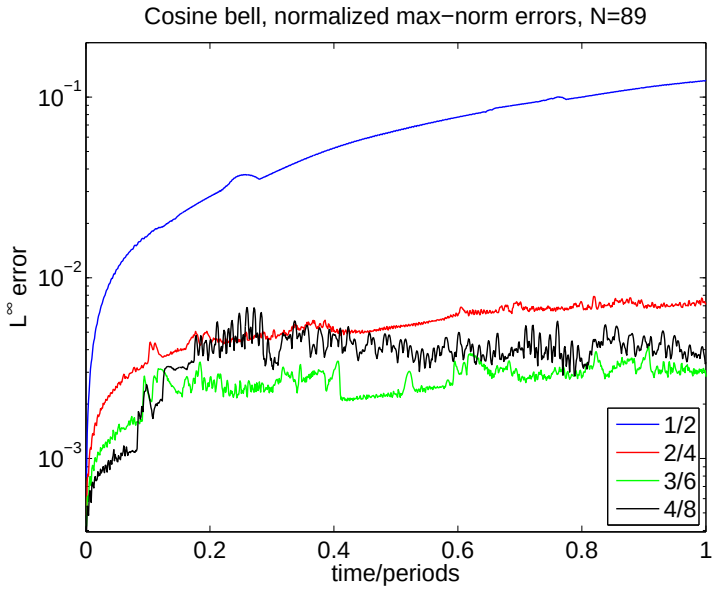

Figure 5: Error vs. time with cosine bell initial data, $N = 89$, 1 period.

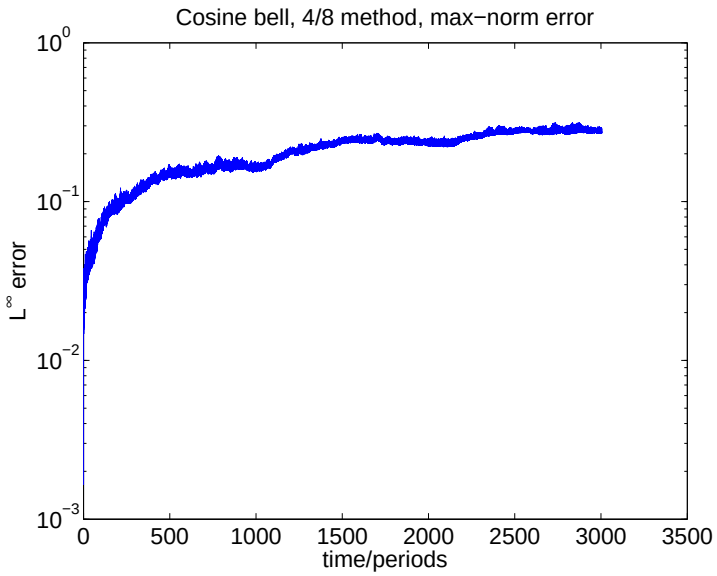

Figure 6: Error vs. time with cosine bell initial data, $N = 45$, 3000 periods.

higher order schemes. Here the sixth order scheme has the smallest absolute error, and the eight order scheme shows the best convergence rate. The reason that the schemes do not converge with the formal order of accuracy is, of course, that the initial data is non-smooth, there is a discontinuity in the derivative.

For long term climate effects, it is necessary to integrate the equation for many years. The period of 12 days, means that there are 3000 periods in 100 years, which could be a reasonable integration time. We show in Fig. 6 the error in time when the initial data (33) is integrated for 3004.3 periods (corresponding to 1.4 million time steps) with the 4/8 method on the grid with $N = 45$. At the final time, the norm error is close to 30 %. Nevertheless, the vortex is still visible in solution shown in Fig. 7a. The exact solution, shown in Fig. 7b, we conclude that the dispersive errors are small.

A second test, we replace the initial data by the smooth

$$\phi(\theta, \lambda) = \frac{h_0}{2} exp(-Ks^2) \tag{34}$$

where $K = 0.451$. h_0 and s are the same as in (33). Figure 8 shows the initial data (34). Figures 9 and 10 shows the evolution of the maximum norm error

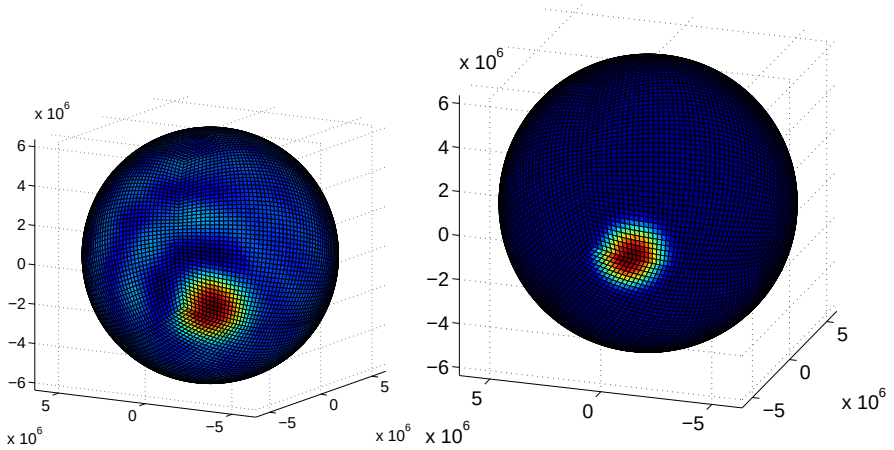

Figure 7: Solution after 3000.4 periods with cosine bell initial data, $N = 45$.

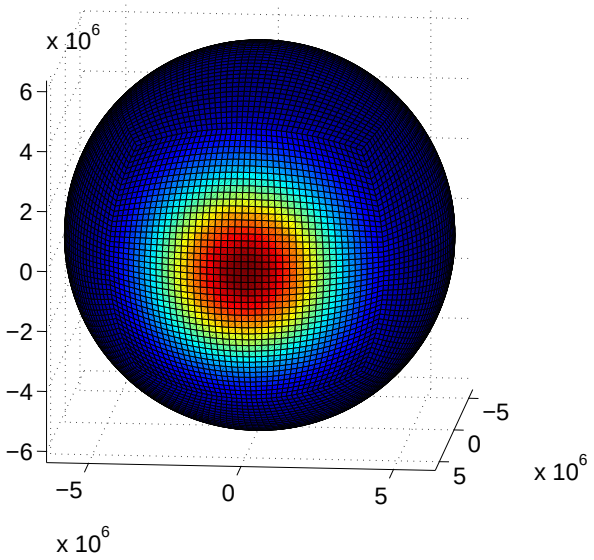

Figure 8: Smooth initial data.

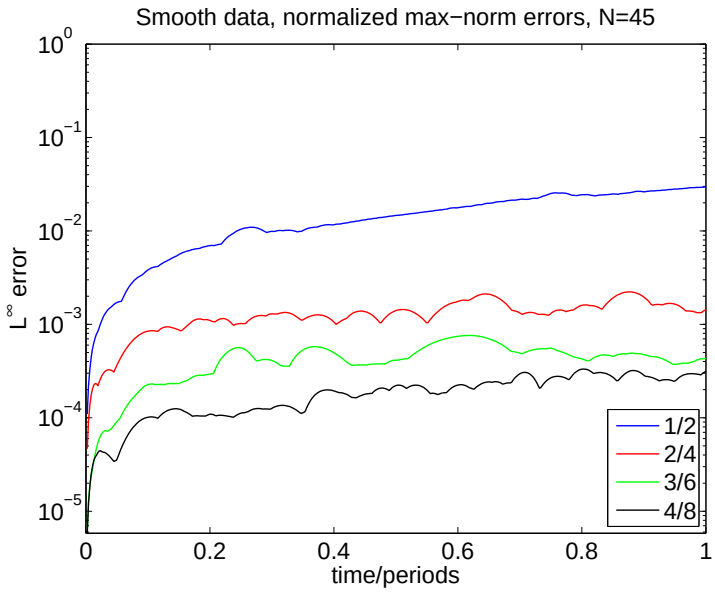

Figure 9: Error vs. time with smooth initial data, $N = 45$, 1 period.

in time on grids with $N = 45$ and $N = 89$ respectively.

Table 2 shows the error after one period. The observed convergence rate in the last column of Table 2 convergence rate is now close to the expected rate, which should be at least equal to the formal order of the boundary operators.

5.2 Finite volume schemes

We use a grid with 40962 cells, having the edge lengths varying between $0.013r$ and $0.0071r$. The corresponding resolution for the cube sphere case is $N = 83$, because $6 \times 83^2 \approx 41000$.

The initial data $\phi(t, \mathbf{0})$ is first transformed to cell averages, by the parametrization and numerical quadrature procedure described in Sec. 4. The error is evaluated by comparing the computed cell averages with the cell averages of the exact solution, also computed by the numerical quadrature procedure.

Fig. 11 shows the error in maximum norm when convecting the cosine bell function (33) on the unstructured grid on period in time. The stencil is third order accurate, and biased in the upwind direction. Fig. 12 displays the

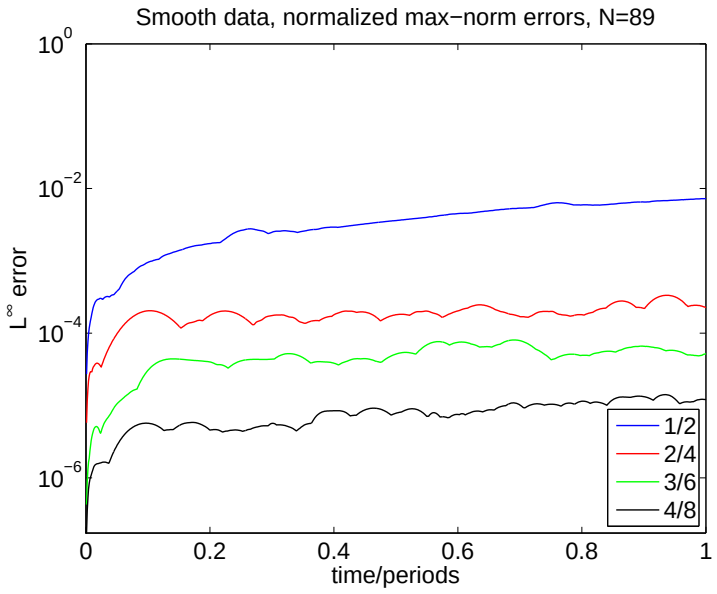

Figure 10: Error vs. time with smooth initial data, $N = 89$, 1 period.

accuracy	L^∞-error with $N = 45$	L^∞-error with $N = 89$	order
1/2	0.0296	0.00725	2.03
2/4	0.00144	2.259×10^{-4}	2.68
3/6	0.000290	2.853×10^{-5}	3.35
4/8	0.000317	1.205×10^{-5}	4.72

Table 2: Errors in the numerical solution after one period for methods of different formal accuracy

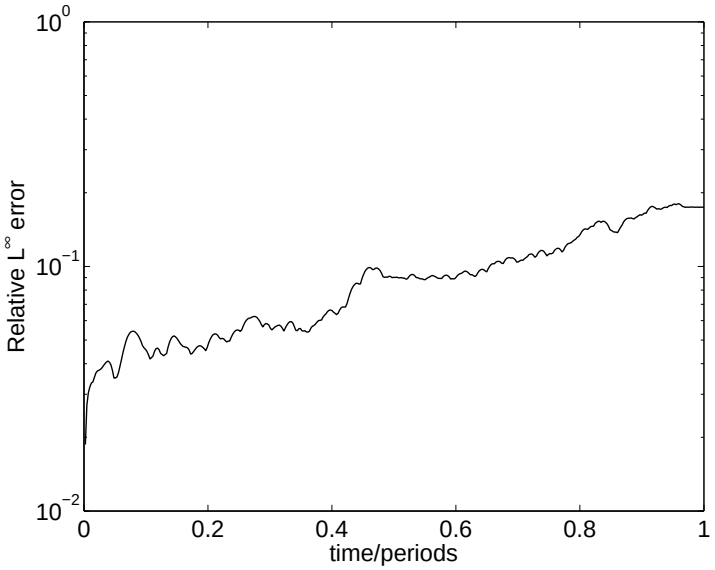

Figure 11: Error vs. time for the third order finite volume scheme when convecting the cosine bell function.

time evolution of the error norm, when convecting the smooth function (34), with the same method as used for the cosine bell function. As expected, the smoother function has a smaller error. Table 3 presents a comparison of the finite difference and finite volume methods. To have a fair comparison, the finite difference scheme is run with $N = 83$, using the method of order 6/3. This gives at least the same formal order of accuracy as the third order finite volume scheme, and a comparable number of grid points. Table 3 shows that the error when convecting the smooth initial data (34) is considerably smaller than with the other two methods. Furthermore, the CPU time of the finite difference scheme is around 16 times smaller than for the finite volume method. The last column of Table 3 shows the minimum and maximum of the numerical solution when convecting a step function ('top hat') that jumps between 0 and 500. The upwind character of the finite volume approximation makes it better at keeping the minimum and maximum values in range. The same is true for the WENO7 scheme, which is designed for shock capturing computations. The property that the solution do not undershoot or overshoot is important in many climate modeling applications. This property can be

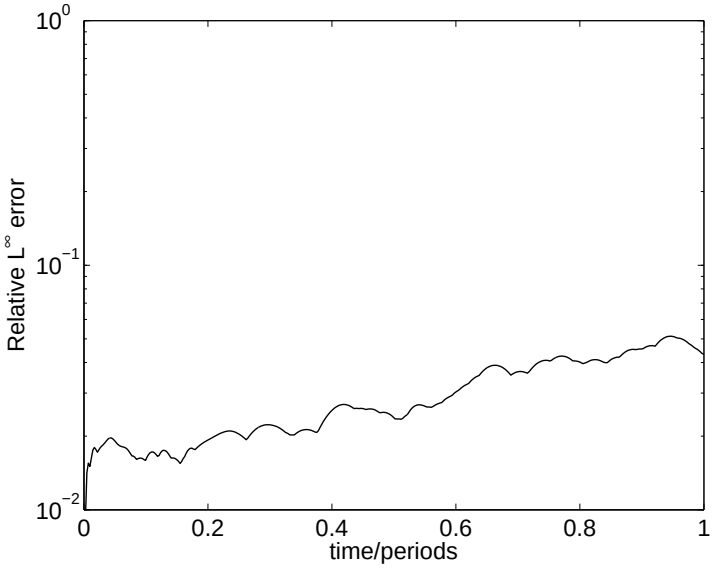

Figure 12: Error vs. time for the third order finite volume scheme when convecting a smooth function.

	CPU time	Error smooth data	Top hat range
FV	422 s	4.00×10^{-2}	$[-20, 514]$
FD	26.8 s	3.42×10^{-5}	$[-122, 605]$
WENO7	42.7 s	7.36×10^{-3}	$[-3, 512]$

Table 3: Comparison of the finite volume (FV), finite difference (FD) and finite difference WENO7 schemes. Cpu time and error after one period of convecting smooth initial data. Top hat range shows range of the numerical solution after one period of convection of a top hat function with jump from 0 to 500.

guranteed for the finite volume scheme by introducing the positivity limiter presented in [10].

The the formal accuracy drops to third order at the boundary for the WENO7 scheme, with the present implementation of boundary operators. The accuracy can, however, be improved by better choices of boundary operators and is the subject of ongoing research.

6 Conclusions

We have presented a high order finite difference scheme for convection on the sphere, using the cube sphere discretization, which is proved to be stable in L^2 norm. Furthermore, we have also presented an upwind high order finite volume discretization for unstructured hexagonal meshes on the sphere. Numerical comparisons have shown the relative strengths and weaknesses of the two methods.

For the cube sphere discretization, it is possible to hybridize the current methods to obtain a scheme that combines the accuracy of the centered finite difference scheme with the shock capturing capabilities of the WENO7 scheme. Further development is needed to implement the schemes for more application relevant differential equations, e.g., the shallow water equations. Further testing of the unstructured algorithm would be interesting, since the various possible choices of stencils have not been fully explored, and is outside the scope of the current work. Some of the implementation details could be applicable to other finite volume like approximations, e.g., the discontinuous Galerkin method.

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

References

[1] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems*. North-Holland, 1978.

[2] D.L.Williamson, J.B.Drake, J.J.Hack, R.Jakob, and P.N.Swarztrauber. A standard test set for numerical approximations to the shallow water equations in spherical geometry. *J. Comput. Phys.*, 102:211–224, 1992.

[3] A. Harten and S. R. Chakravarthy. Multi-dimensional ENO schemes for general geometries. Report 91-76, ICASE, 1991.

[4] C. Hu and C.-W. Shu. Weighted essentially non-oscillatory schemes on triangular meshes. *J. Comput. Phys.*, 150:97–127, 1999.

[5] G.-S. Jiang and C.-W. Shu. Efficient implementation of weighted ENO schemes. *J. Comput. Phys.*, 126:202–228, 1996.

[6] P. Olsson. Summation by parts, projections, and stability. I. *Math. Comp.*, 64:1035–1065, 1995.

[7] William M. Putman and Shian-Jiann Lin. Finite-volume transport on various cube-sphere grids. *J. Comput. Phys.*, 227:55–78, 2007.

[8] B. Strand. Summation by parts for finite difference approximations for d/dx. *J. Comput. Phys.*, 110:47–67, 1994.

[9] M. Taylor, J. Tribbia, and M. Iskandarani. The spectral element method for the shallow water equations on the sphere. *J. Comput. Phys.*, 130:92–108, 1997.

[10] X. Zhang and C.-W. Shu. On maximum-principle-satisfying high order schemes for scalar conservation laws. *J. Comput. Phys.*, 229:3091–3120, 2010.

[11] Yong-Tao Zhang and Chi-Wang Shu. Third order WENO schemes on three dimensional tetrahedral meshes. *Commun. Comput. Phys.*, 5:836–848, 2009.