

A SOLUTION TO THE POLE PROBLEM FOR THE SHALLOW WATER EQUATIONS ON A SPHERE

G. STARIUS¹

ABSTRACT. We consider a reduced gridding technique for the shallow water equations on a sphere, based on spherical coordinates. In a small vicinity of the poles, a longitudinal derivative is discretized at a grid-point on a parallel, by using points on the great circle through the grid-point and tangent to the parallel. Centered one-dimensional interpolation formulas are used in this process and also in connecting adjacent segments in the reduced grid. The remaining spatial discretization is obtained by simply replacing derivatives by centered equidistant finite difference approximations.

Numerical experiments for scalar advection equations and for the well-known Rossby-Haurwitz test example indicate that the methods developed work surprisingly well. Some advantages are that (i) a fairly uniform grid, with many reductions or segments, can be used, (ii) order of approximation $2p$ in the spatial discretizations requires only $4p + 1$ points and (iii) the local and simple structure of the schemes will make efficient implementation on massively parallel computer systems possible.

The paper is an attempt towards global numerical weather prediction models, by first analyzing the pole problem for reduced latitude-longitude grids.

Keywords: shallow water equations, sphere, reduced grid, pole problem, segment, numerical weather prediction.

AMS Subject Classification: 35L45, 65M06.

1. INTRODUCTION

In this paper we present a class of methods for the numerical solution of the shallow water equations on a sphere, based on reduced latitude-longitude grids. Zonal derivatives close to the poles are discretized at each grid-point on a parallel, by using points on the great circle passing through the grid-point and being tangent to the parallel. A somewhat simplified overview of this technique is illustrated in Figures 1 and 2 below. We shall call it the *Pole Tangent Derivative* method, or just the PTD method. When zonal derivatives are approximated, using only points on the relevant parallel, we refer to the procedure as the *Pole Parallel Derivative* method, or just the PPD method. This has been the standard approximation for difference methods using latitude-longitude grids. We recall that a parallel is a circle with constant latitude.

Consider the use of formulas for the PTD and the PPD methods, with $2p + 1$ points on the relevant great circle and on the relevant parallel, respectively. *The theoretical justification for replacing the ordinary PPD approximation by PTD, in a vicinity of the poles, is that for PTD the discretization error is $O(\Delta\theta^{2p})$, and for PPD only $O(\Delta\theta)$, where $\Delta\theta$ is the latitudinal step.*

Suppose we have $n = 2^\ell$ grid-points near the equator, then $\ell - 3$ reductions, by the factor 2, are needed to obtain 2^3 grid-points on the parallel closest to a pole. This results in $\ell - 2$ segments on each hemisphere. As reported in Section 4, we have successfully used 8 segments,

¹Department of Mathematics, Chalmers University of Technology and the University of Göteborg, Sweden, e-mail: goran@chalmers.se

Manuscript received February 2014.

corresponding to $n = 2^{10}$, in many numerical experiments. This is in sharp contrast to, cf. [2,8], where it is stated that only very few reductions can be made without significant error growth near the poles. We may conclude that the PTD method has made it possible to use many segments. To emphasize the use of many segments, we shall call the methods considered in this paper *segmented methods*.

Our schemes should be seen as a first step towards a global numerical weather prediction (NWP) model. Static, and perhaps also dynamic, adaptivity should be implemented. This would require mesh refinements and eventually also the use of discretizations more suitable than centered approximations, for sharp weather fronts, cf. [8,9]. The question is not how much adaptivity we need, but rather how much we can afford. In this paper we restrict ourselves to the basic methods.

We will only consider Eulerian formulations here, but it would be of interest to use PTD also in connection with semi-Lagrangian schemes. We point out that our methods are not conservative in a strict sense, but in our example, at the end of Section 4, Table 10, the deviations are of the order 0.01% after 10 days, which must be less than needed for weather prediction. See [6], page 2, item 1 with title: Mass conservation.

The paper is organized as follows. In Section 2 we write down the shallow water equations in spherical coordinates and discuss the grid systems employed. We introduce the so called distortion factor, which is used for the segmentation of the sphere. We also present formulas for interpolation and for approximation of derivatives, typical for our schemes. Section 3, which is the central part of this paper, discusses various aspects of the pole problems and their appropriate treatments. In Section 4 several numerical experiments for advective equations and for the Rossby-Haurwitz problem are described. For the latter we also consider conservation properties for mass and energy. Finally, in our concluding Section 5, we consider possible further investigations.

We will now briefly comment on some other methods considered in the literature, and their relation to the methods of this paper. Our main sources are the two review papers, Staniforth and Thuburn [6] and Williamson [11]. Both contain a great number of references, complementing the ones given here.

Most operational global weather forecast models are based on full latitude-longitude grids [6] and semi-Lagrangian schemes [11]. By instead using reduced grids, like in this paper, the number of grid-points can be decreased to about two thirds, which is a considerable saving. Furthermore, the scalability problems at the poles for full latitude-longitude grids [6] are completely removed.

Obviously there are other quasi-uniform grids than ours, for example the geodesic grids, among which the cubed sphere and the icosahedral grids seem to be the most popular. Cubed sphere grids consist of quadrilaterals, and icosahedral grids are generally triangular. Both avoid clustering and scalability problems at the poles. The cubed sphere grids, with angles up to 120 degrees, are briefly commented on in Section 5. The triangular icosahedral grids have some drawbacks, according to [6,11]. Reasonably simple schemes tend to be of lower order, such as second order, *if* applied on a plane uniform grid. In particular higher order schemes support computational modes and noise.

It is becoming more and more important to design NWP methods that can use up to hundreds of thousands of processors in an efficient way. This evolution may lead to the return of explicit time integration, cf. [9]. This in turn will probably lead to the use of Eulerian formulations instead of semi-Lagrangian ones, because it seems to be very hard to construct reasonable semi-Lagrangian schemes of higher order, and the possibility to take longer time steps than for Eulerian schemes has disappeared.

Considering the above and the experiments in Section 4, methods based on grid systems used in this paper, together with PTD, seem to compare favorably with other alternatives. In particular, we refer to the following properties associated with our methods: (i) *Simplicity of grid structure*, (ii) *good scalability*, (iii) *quasi-uniformity*, (iv) *orthogonality*, (v) *quite sufficient conservation of mass for NWP*, and (vi) *simple efficient high order discretizations*. Most test runs in Section 4 were made for spatial discretizations of orders 8 or 10.

2. GOVERNING EQUATIONS, GRID SYSTEMS AND SOME BASIC DISCRETE FORMULAS

Let us first recall the transformation between Cartesian and spherical coordinates

$$\begin{cases} x = a \cos \theta \cos \lambda, \\ y = a \cos \theta \sin \lambda, \\ z = a \sin \theta, \end{cases} \quad 0 \leq \lambda < 2\pi, \quad -\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}, \quad (1)$$

where a is the radius of the earth, λ the longitude and θ the latitude. The advective form of the shallow water equations in spherical coordinates is given by

$$\begin{aligned} u_t &= -(a \cos \theta)^{-1}(uu_\lambda + \Psi_\lambda) - a^{-1}vu_\theta + (f + a^{-1}u \tan \theta)v, \\ v_t &= -(a \cos \theta)^{-1}uv_\lambda - a^{-1}(vv_\theta + \Psi_\theta) - (f + a^{-1}u \tan \theta)u, \\ \Psi_t &= -(a \cos \theta)^{-1}(u_\lambda \Psi + u\Psi_\lambda) - a^{-1}(v\Psi)_\theta + a^{-1}v\Psi \tan \theta, \end{aligned} \quad (2)$$

where Ψ is the geopotential, $f = 2\Omega \sin \theta$ the Coriolis parameter, Ω the rotation rate of the earth and, finally, u and v are the eastward and the northward velocity components, respectively. We notice, in particular, that $\tan \theta$ and $(\cos \theta)^{-1}$ are unbounded near the poles, which is a problem that will be attended to in Sections 3.1 and 3.2.3, respectively. The factor $(\cos \theta)^{-1}$ will be canceled using the PTD method. In Section 3.1 we will represent the velocity vector by its orthogonal projection onto the equatorial plane, where it is expressed in Cartesian coordinates, U and V , say. In a proximity of the poles the system (2) will be expressed in terms of U and V , instead of u and v , which yields a system with bounded coefficients.

Throughout this paper we assume that the solutions are smooth on the whole sphere, in local Cartesian coordinate systems. Further we assume that the same physical mesh-length is suitable everywhere. This means that it is desirable to keep the variation in the physical resolution as small as possible.

Let us now look at the reduced grid systems of interest in this paper. The latitudinal step $\Delta\theta$ is kept constant on the whole sphere, in contrast to the longitudinal step $\Delta\lambda$, which will be increasing and piecewise constant from the equator to the poles. The distances from a grid-point to its nearest neighbors in the λ and θ directions should be as equal as possible. For this reason we define the *distortion factor* as

$$d = \max(\Delta s/\Delta\theta, \Delta\theta/\Delta s) \leq d_{max}, \quad \Delta s = \Delta\lambda \cos \theta. \quad (3)$$

This will be used for partitioning the sphere into segments, within which $\Delta\lambda$ is constant. As we move from the equator to a pole we will sometimes increase $\Delta\lambda$ to $q\Delta\lambda$, where $q > 1$ is predetermined. Suppose $\Delta\theta/\Delta s$ has reached the value d_{max} , then we ought to have the equality

$$q\Delta s/\Delta\theta = \Delta\theta/\Delta s = d_{max} \quad \text{and hence} \quad d_{max} = \sqrt{q},$$

in order to minimize d_{max} for given q .

We now consider the simplest case, namely $q = 2$, which is used in our numerical experiments in Section 4. Let $n = 2^\ell$ be the number of grid-points on a parallel at the equator, set $m = \frac{1}{4}n$

and define

$$\theta_i = (i - \frac{1}{2})\Delta\theta, \quad i = 1, \dots, m, \quad \text{where } \Delta\theta = \frac{1}{2m-1}\pi,$$

which implies $\theta_m = \frac{1}{2}\pi$. Define the segments S_j by the sets

$$\left\{ i \mid 1 \leq i < m, \frac{1}{\sqrt{2}} < \frac{\Delta\theta}{\cos(\theta_i)\Delta\lambda} \leq \sqrt{2}, \Delta\lambda = \frac{2\pi}{2^{\ell+1-j}} \right\}, \quad j = 1, \dots, \ell - 2.$$

The structures of the grid systems for $\ell = 8, 9, 10, 11$ are given in Table 1. Each row in the subtables correspond to a segment S_j . The third and the fourth columns contain the number of parallels corresponding to the segment and the number of grid-points on each parallel, respectively.

TABLE 1. Structures of grid systems with $q = 2$

$\ell = 8 \quad n = 256$				$\ell = 9 \quad n = 512$			
segment	interval	width	length	segment	interval	width	length
1	1-31	31	256	1	1-63	63	512
2	32-49	18	128	2	64-98	35	256
3	50-56	7	64	3	99-113	15	128
4	57-60	4	32	4	114-120	7	64
5	61-62	2	16	5	121-124	4	32
6	63	1	8	6	125-126	2	16
				7	127	1	8
$\ell = 10 \quad n = 1024$				$\ell = 11 \quad n = 2048$			
segment	interval	width	length	segment	interval	width	length
1	1-127	127	1024	1	1-255	255	2048
2	128-197	70	512	2	256-394	139	1024
3	198-227	30	256	3	395-454	60	512
4	228-241	14	128	4	455-483	29	256
5	242-248	7	64	5	484-497	14	128
6	249-252	4	32	6	498-504	7	64
7	253-254	2	16	7	505-508	4	32
8	255	1	8	8	509-510	2	16
				9	511	1	8

From the last four rows in the subtables we see that the pattern for the four segments closest to a pole is independent of the resolution, for $\ell \geq 8$. When n is doubled these four segments are simply squeezed towards the pole. Further notice that the equator is between two adjacent parallels of grid-points, which makes it possible for a program, for a segmented method, to be reflection invariant with respect to the equator. This is not essential, but we think it is a pleasant property. We conclude this section by some discrete formulas typical for our methods. Since such formulas can easily be derived, e.g., by computer algebra, the list below is restricted to only one order of approximation for each formula. For all segments, except the two closest to a pole, we replace all first order derivatives by centered equidistant difference approximations. Let $x_j = jh$, for some mesh-length h , and set $f_j = f(x_j)$, where $f(x)$ is a smooth function. Then we have

$$\begin{aligned} f'(x_j) &= D_{x,8}f(x_j) + \frac{1}{630}h^8 f^{(9)}(\xi_1), \quad \text{where } hD_{x,8}f(x_j) = \\ &= \frac{4}{5}(f_{j+1} - f_{j-1}) - \frac{1}{5}(f_{j+2} - f_{j-2}) + \frac{4}{105}(f_{j+3} - f_{j-3}) - \frac{1}{280}(f_{j+4} - f_{j-4}). \end{aligned} \tag{4}$$

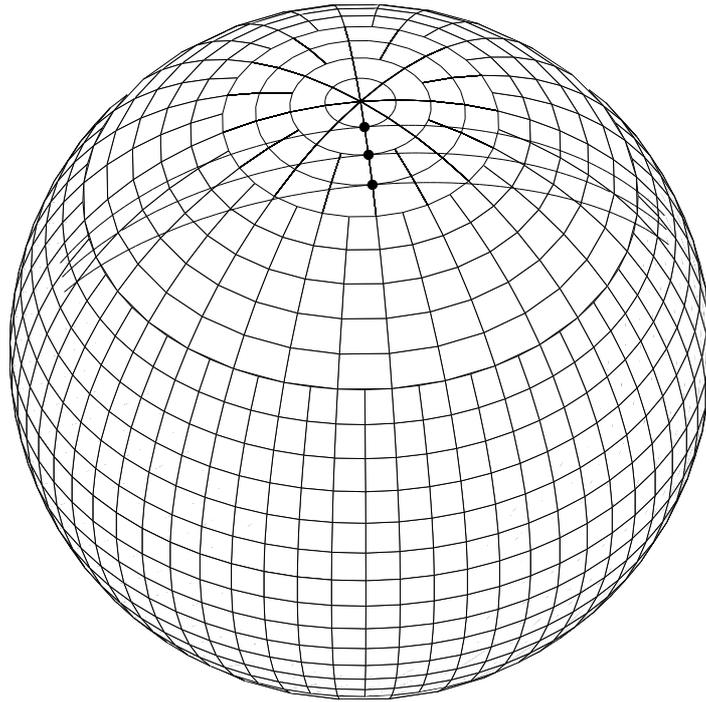


Figure 1. The grid system for $n = 2^6$ points around the equator and with 4 segments. A zonal derivative in a grid-point marked by \bullet will be approximated by using points on the great circle through the grid-point, which is illustrated in Figure 2. Details are given in Section 3.2.3.

For later use we define $D_{x,2p}$, in analogy with the above, to be a centered equidistant difference approximation of order $2p$. When discretizing in a segment close to its boundary towards a pole, see Figure 1, interpolation formulas are needed for the mid-points in the zonal direction

$$f(x_{j+\frac{1}{2}}) = \frac{1225}{2048}(f_j + f_{j+1}) - \frac{245}{2048}(f_{j-1} + f_{j+2}) + \frac{49}{2048}(f_{j-2} + f_{j+3}) - \frac{5}{2048}(f_{j-3} + f_{j+4}) + \frac{1}{936.2\dots}h^8 f^{(8)}(\xi_2). \quad (5)$$

For the two segments closest to a pole we need special formulas. In Section 3.2.3 the zonal derivatives will be replaced by centered non-equidistant difference approximations of the form

$$f'(x_j) = \frac{1}{h} \sum_{k=1}^4 d_k (f(x_j + p_k h) - f(x_j - p_k h)) + O(h^8), \quad (6)$$

where the $p_k \sim k$, $k = 1, \dots, 4$, are given. The coefficients d_k are computed in a preprocessor. For the values at the poles, we use interpolation along meridians of the form

$$f(x_j) = \frac{5}{6}(f_{j+1} + f_{j-1}) - \frac{10}{21}(f_{j+2} + f_{j-2}) + \frac{5}{28}(f_{j+3} + f_{j-3}) - \frac{5}{126}(f_{j+4} + f_{j-4}) + \frac{1}{252}(f_{j+5} + f_{j-5}) - \frac{1}{252}h^{10} f^{(10)}(\xi_3). \quad (7)$$

In Section 4.1. the above choice is shown to compare favorably with using the differential equations, in Cartesian coordinates, at the poles.

3. THE POLE PROBLEM

Below we consider three different problems connected to the poles. Together they are known as the pole problem. The first two are very easy to understand and overcome. The third is more difficult, both to detect and to solve.

The first of them is simply that if we have the same number of grid-points on each parallel, often called a full latitude-longitude grid, the grid-points will cluster near the poles and cause stability problems in the time integration. A reduced grid solves this problem, but unfortunately gives rise to the third problem below.

The second problem has to do with the velocity vector \mathbf{v} . If $\mathbf{v} = (u, v)$, where u and v are the eastward and northward components of the velocity, then at the poles u and v are undefined. We therefore need another representation of the velocity vector in a vicinity of the poles. Such a representation is obtained in Section 3.1 by projecting \mathbf{v} onto the equatorial plane.

The third problem is related to the approximation of zonal derivatives close to a pole, where the number of grid-points per parallel is fairly small. The colatitude for the k :th parallel, counted from a pole, is $k\Delta\theta$. For $k = 1$ we have 8 grid-points, for $k = 2, 3$ there are 16 and so on. For polynomial based difference approximations to zonal derivatives, using grid-points only on the relevant parallel, the error is shown in Section 3.2.1 to be of order only $O(\Delta\theta)$, for any fixed k . However, by using $2p + 1$ points on a suitable great circle through the grid-point the error is $O(\Delta\theta^{2p})$, as already mentioned in the introduction. We recall that the two methods above are referred to as the PPD and the PTD method, respectively.

3.1. A representation of the velocity vector suitable in a vicinity of the poles. For each point (λ, θ) on the sphere, except the poles, the velocity vector \mathbf{v} can be expressed in an orthonormal basis $\mathbf{e}_\lambda, \mathbf{e}_\theta$, consisting of the unit tangents to the parallel and the meridian through the point, respectively, and we have $\mathbf{v} = u\mathbf{e}_\lambda + v\mathbf{e}_\theta$. The orthogonal projection of \mathbf{v} onto the xy -plane, corresponding to (1), is denoted by $P\mathbf{v} = -U\mathbf{e}_x + V\mathbf{e}_y$, where the minus sign for U is chosen in order to get nicer formulas later on. Elementary geometric considerations give

$$P\mathbf{e}_\lambda = -\sin\lambda\mathbf{e}_x + \cos\lambda\mathbf{e}_y \quad \text{and} \quad P\mathbf{e}_\theta = -(\cos\lambda\mathbf{e}_x + \sin\lambda\mathbf{e}_y)\sin\theta$$

and since P is a linear operator we find

$$\begin{pmatrix} U \\ V \end{pmatrix} = A \begin{pmatrix} u \\ \sin\theta v \end{pmatrix}, \quad A = \begin{pmatrix} \sin\lambda & \cos\lambda \\ \cos\lambda & -\sin\lambda \end{pmatrix}, \quad (8)$$

where $A^T = A^{-1} = A$.

Now we shall transform (2) by using (8) to express derivatives of u and v in terms of the new variables U and V . To achieve this we multiply the second equation in (2) by $\sin\theta$ and then write the first two equations in vector form as

$$\begin{pmatrix} u \\ \sin\theta v \end{pmatrix}_t = -\frac{1}{a\cos\theta}u \begin{pmatrix} u \\ \sin\theta v \end{pmatrix}_\lambda - \frac{1}{a}v \begin{pmatrix} u \\ \sin\theta v \end{pmatrix}_\theta - \frac{1}{a} \begin{pmatrix} \Psi_\lambda/\cos\theta \\ \sin\theta\Psi_\theta - \cos\theta v^2 \end{pmatrix} + \left(\frac{\tan\theta}{a}u + f\right) \begin{pmatrix} v \\ -\sin\theta u \end{pmatrix}.$$

Now by using

$$\begin{pmatrix} u \\ \sin\theta v \end{pmatrix} = A \begin{pmatrix} U \\ V \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} \sin\lambda\cos\lambda & \cos^2\lambda \\ -\sin^2\lambda & -\sin\lambda\cos\lambda \end{pmatrix},$$

a somewhat tedious computation yields the transformed system

$$\begin{aligned} \begin{pmatrix} U \\ V \end{pmatrix}_t &= -\frac{1}{a \cos \theta} u \begin{pmatrix} U \\ V \end{pmatrix}_\lambda - \frac{1}{a} v \begin{pmatrix} U \\ V \end{pmatrix}_\theta + 2\Omega \begin{pmatrix} -V \\ U \end{pmatrix} - \\ &\quad - \frac{1}{a} A \begin{pmatrix} \Psi_\lambda / \cos \theta \\ \sin \theta \Psi_\theta - \cos \theta v^2 \end{pmatrix} + \cos \theta \left(\frac{u}{a} + 2\Omega \cos \theta \right) B \begin{pmatrix} U \\ V \end{pmatrix}, \\ \Psi_t &= -\frac{1}{a \cos \theta} (u \Psi_\lambda + (\sin \lambda U_\lambda + \cos \lambda V_\lambda) \Psi) - \\ &\quad - \frac{1}{a} \cos \lambda (U \Psi / \sin \theta)_\theta + \frac{1}{a} \sin \lambda (V \Psi / \sin \theta)_\theta. \end{aligned} \quad (9)$$

In this way we have gotten rid of the unbounded function $\tan \theta$, which is present in (2), and in Section 3.2.3, $(\cos \theta)^{-1}$ will also be canceled.

Mainly for reasons of implementation it would be advantageous to have a simpler system than (9). For high resolution the factor $\sin \theta$ is close to 1 in a vicinity of the northern pole and to -1 near the southern. We therefore replace (8), notice *not* approximate, by the following mappings

$$\begin{pmatrix} u \\ v \end{pmatrix} = A \begin{pmatrix} U \\ V \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} u \\ -v \end{pmatrix} = A \begin{pmatrix} U \\ V \end{pmatrix}, \quad (10)$$

for the northern and the southern poles, respectively. In a manner similar to before we get the transformed systems for the northern and the southern pole as follows.

The North Pole:

$$\begin{aligned} \begin{pmatrix} U \\ V \end{pmatrix}_t &= -\frac{1}{a \cos \theta} u \begin{pmatrix} U \\ V \end{pmatrix}_\lambda - \frac{1}{a} v \begin{pmatrix} U \\ V \end{pmatrix}_\theta - \frac{1}{a} A \begin{pmatrix} \Psi_\lambda / \cos \theta \\ \Psi_\theta \end{pmatrix} + \\ &\quad + (f - \frac{u}{a} \tan(\frac{1}{4}\pi - \frac{1}{2}\theta)) \begin{pmatrix} -V \\ U \end{pmatrix}, \\ \Psi_t &= -\frac{1}{a \cos \theta} (u \Psi_\lambda + (\sin \lambda U_\lambda + \cos \lambda V_\lambda) \Psi) - \\ &\quad - \frac{1}{a} (\cos \lambda (U \Psi)_\theta - \sin \lambda (V \Psi)_\theta) - \frac{1}{a} \tan(\frac{1}{4}\pi - \frac{1}{2}\theta) v \Psi. \end{aligned} \quad (11)$$

The South Pole:

$$\begin{aligned} \begin{pmatrix} U \\ V \end{pmatrix}_t &= -\frac{1}{a \cos \theta} u \begin{pmatrix} U \\ V \end{pmatrix}_\lambda - \frac{1}{a} v \begin{pmatrix} U \\ V \end{pmatrix}_\theta - \frac{1}{a} A \begin{pmatrix} \Psi_\lambda / \cos \theta \\ -\Psi_\theta \end{pmatrix} - \\ &\quad - (f + \frac{u}{a} \tan(\frac{1}{4}\pi + \frac{1}{2}\theta)) \begin{pmatrix} -V \\ U \end{pmatrix} \\ \Psi_t &= -\frac{1}{a \cos \theta} (u \Psi_\lambda + (\sin \lambda U_\lambda + \cos \lambda V_\lambda) \Psi) + \\ &\quad + \frac{1}{a} (\cos \lambda (U \Psi)_\theta - \sin \lambda (V \Psi)_\theta) + \frac{1}{a} \tan(\frac{1}{4}\pi + \frac{1}{2}\theta) v \Psi. \end{aligned} \quad (12)$$

We note that terms of type $\tan(\cdot)$ have now reappeared, but in a bounded form and, in fact, vanish at the poles. From now on we assume that the velocity components U and V are smooth functions at the poles.

3.2. The pole problem for scalar valued functions. We will now study different types of approximations for zonal derivatives in a vicinity of a pole. Firstly, polynomial based formulas, using only points on a parallel, are considered in Section 3.2.1, called the *Pole Parallel Derivative* method, PPD. Then, in Section 3.2.2, we will briefly consider interpolation on parallels by using trigonometric polynomials. Finally, in Section 3.2.3, we focus on the *Pole Tangent Derivative* method, PTD, which is our preference. See Figure 2, which illustrates the situation on the positive x -axis, $\lambda = 0$, for the first three parallels closest to a pole.

3.2.1. The Pole Parallel Derivative method, PPD. Let F be a smooth function in a vicinity of a pole. Since the coordinate z is determined by x and y we can write $F = F(x, y)$. For simplicity, we normalize the scale so that $a = 1$. Whenever the symbol a appears again in the text, it is the radius expressed in meters. We will now look at discretization errors for centered difference approximations of the same kind as (4). On the k th parallel we have

$$D_{\lambda,2p}F_k(\lambda) = D_\lambda F_k(\lambda) + (-1)^{p+1} c_p \Delta \lambda^{2p} D_\lambda^{2p+1} F_k(\lambda) + O(\Delta \lambda^{2p+2}),$$

where $F_k(\lambda) = F(\sin(k\Delta\theta) \cos \lambda, \sin(k\Delta\theta) \sin \lambda)$ and, e.g., $c_3 = 1/140$, $c_4 = 1/630$ and $c_5 = 1/2772$. We recall that $D_\lambda = d/d\lambda$. By repeated differentiation we get

$$D_\lambda^{2p+1} F_k(\lambda)|_{\lambda=0} = (-1)^p \sin(k\Delta\theta) F_y|_{\lambda=0} + O(\sin^2(k\Delta\theta)),$$

and our wanted formula, with only the dominant error term written out, will be

$$D_{\lambda,2p}F_k(\lambda)|_{\lambda=0} = D_\lambda F_k(\lambda)|_{\lambda=0} - c_p \Delta \lambda^{2p} \sin(k\Delta\theta) F_y|_{\lambda=0} + \dots \quad (13)$$

We note that $\lambda = 0$ corresponds to a grid-point on a parallel. The error terms for interpolation are quite similar. The pole problem, of our third kind, can be characterized by stating that we only get a first order approximation for the zonal derivative in $\Delta\theta$, the basic discretization parameter, by using a polynomial based formula containing grid-points on the parallel. Let us recall that for $k = 1$ we have $\Delta\lambda = \frac{1}{4}\pi$, for $k = 2, 3$, $\Delta\lambda = \frac{1}{8}\pi$ and for $k = 4, 5, 6, 7$ $\Delta\lambda = \frac{1}{16}\pi$. The accuracy can still be good, particularly away from the pole, for example $c_4(\frac{1}{16}\pi)^8 \sim 3.510^{-9}$. Actually for simple advection problems, the PPD method seems to work fairly well, which can be seen for example in Section 4.1. When the velocity vector is present the situation is different, as it is, e.g., in the Rossby-Haurwitz problem. We shall return to this in Section 4.2.

3.2.2. Interpolation with trigonometric polynomials on parallels. Interpolation with trigonometric polynomials will be used on the first three parallels, and is an important part of the PTD method. Let us first introduce the trigonometric polynomial

$$S_N(\lambda) = \frac{1}{2} a_0 + \sum_{\nu=1}^N (a_\nu \cos(\nu\lambda) + b_\nu \sin(\nu\lambda)).$$

Three different interpolation methods will be considered below, but only formulated for a parallel closest to a pole. In the simplest case only function values are used at the grid-points, in the second case first order derivatives are used as well, and finally also second order derivatives are added. The derivatives needed will be approximated by PTD, more of this later. For $k = 1$ we have $\Delta\lambda = \frac{1}{4}\pi$, $\lambda_j = j\Delta\lambda$, with $j = 0, \dots, 7$. The following are our three interpolation methods:

- (i) Given the data $F_1(\lambda_j)$, $j = 0, \dots, 7$, find the trigonometric polynomial $S_4(\lambda)$, with $b_4 = 0$, such that $S_4(\lambda_j) = F_1(\lambda_j)$, $j = 0, \dots, 7$.
- (ii) Given the data $F_1(\lambda_j)$, $F'_1(\lambda_j)$, $j = 0, \dots, 7$, find the trigonometric polynomial $S_8(\lambda)$, with $a_8 = 0$, such that $S_8(\lambda_j) = F_1(\lambda_j)$, $S'_8(\lambda_j) = F'_1(\lambda_j)$, $j = 0, \dots, 7$.

- (iii) Given the data $F_1(\lambda_j), F'_1(\lambda_j), F''_1(\lambda_j), j = 0, \dots, 7$, find the trigonometric polynomial $S_{12}(\lambda)$, with $b_{12} = 0$, such that $S_{12}(\lambda) = F_1(\lambda_j), S'_{12}(\lambda_j) = F'_1(\lambda_j), S''_{12}(\lambda_j) = F''_1(\lambda_j), j = 0, \dots, 7$.

The condition numbers for the three systems of linear equations, for the coefficients in the trigonometric polynomials, in the 2-norm, are $\sqrt{2}$, about 10 and about 200, respectively. For $k = 2, 3$ we have $\Delta\lambda = \frac{1}{8}\pi, \lambda_j = j\Delta\lambda$, where $j = 0, \dots, 15$. The condition numbers for (i) and (ii) are now $\sqrt{2}$ and about 20, respectively.

In [4] it is proved that the interpolation error, corresponding to method (i) and 8 points, is $O(\Delta\theta^4)$. The proof can easily be generalized to any even number of points for the case (i), and, with some effort, for the cases (ii) and (iii). The order of approximation is half the number of data values. Thus the interpolation errors, using 8 points, for (ii) and (iii) are $O(\Delta\theta^8)$ and $O(\Delta\theta^{12})$, respectively.

3.2.3. The Pole Tangent Derivative method, PTD. Many variants of this method or procedure are possible. Their common feature is represented in Figures 1 and 2. The order of approximation determines the number of points that will be used on the tilted great circles. To be able to use PTD for $k = 1, 2, 3$ in a convenient way we will work with 32 points per parallel, as used in the third segment. The active grid-points are only those given in Table 1. The interpolation required will be performed using trigonometric polynomials as in (i), (ii) or (iii) above. Since we interpolate and differentiate numerically on meridians in a vicinity of the poles by centered formulas, it is advantageous to make extensions beyond the poles for a number of parallels using 32 points. This will be done in such a way that a column in our array corresponds to a meridian. Below we give the main features of the algorithm for PTD on three parallels, with more details thereafter.

- (1) At the poles we determine a value of a dependent variable by using a formula such as (7) in the four different directions $\lambda = 0, \frac{1}{4}\pi, \frac{1}{2}\pi, \frac{3}{4}\pi$, and then taking the mean value.
- (2) For $k = 2, 3$ we use trigonometric interpolation using 16 function values to determine the 32 values needed. Extensions are then made over the poles for the two parallels.
- (3) For $k = 1$ we approximate F_λ and sometimes even $F_{\lambda\lambda}$ by PTD, i.e., formulas such as (6), and then subtabulation by methods (ii) or (iii) to 32 points. Extensions are then made over the poles for the parallel.
- (4) For $k = 2, 3$ we approximate F_λ by PTD.
- (5) Either we stop here, or else we go back to item 2, where this time also F_λ is used in the interpolation. Continued iteration is also possible, but we have not seen any advantage of this, in our examples.

For each parallel on which we use PTD, data have to be supplied, for the determination of the points in (6), marked by * in Figure 2. This figure corresponds to the following data for the first three parallels:

$k = 1$						$k = 2$						$k = 3$					
m	p	p	p	p	p	m	m	m	p	p	p	m	m	m	m	p	p
4	2	3	4	5	6	2	4	5	4	5	6	1	2	3	4	5	6

Here the letter m stands for a meridian and the letter p for a parallel. An integer under a letter m is the number of the meridian counted from, but not including, the meridian $\lambda = 0$. An integer under a letter p denotes the number of the parallel counted from the pole. For example,

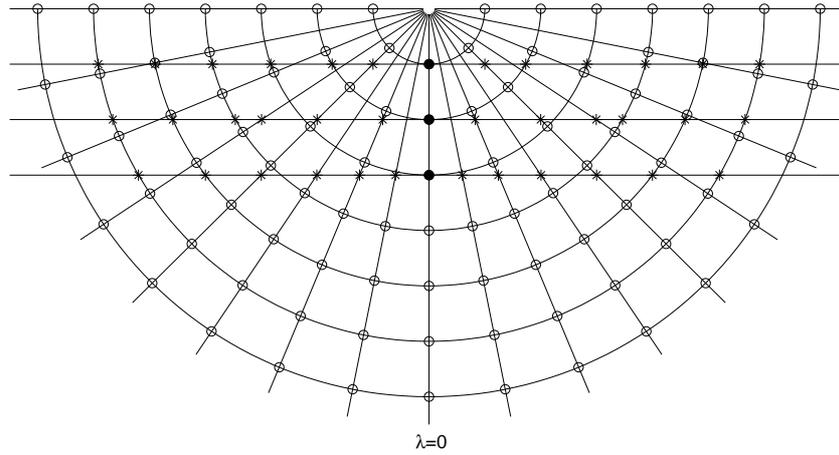


Figure 2. An illustration of the pole tangent derivative method, PTD, used to approximate zonal derivatives on the three parallels closest to a pole. Only grid-points on one meridian are marked by • and great circles, being tangents to parallels through these points are drawn. For each point • we have marked the points on the corresponding great circle by *, that are used in the difference approximation to the zonal derivative. The vertical meridian in the figure corresponds to $\lambda = 0$, which will be referred to in the text.

for the third parallel and the corresponding tangent, containing points marked by * in Figure 2, the first four stars lie on meridians with numbers 1, 2, 3 and 4, and the last two stars lie on parallels with numbers 5 and 6. We decide, e.g., by using a suitable figure, which points * to choose and then determine the corresponding input data by the rules above.

It can easily be checked that the great circle through the point $(\lambda, \theta) = (0, \frac{1}{2}\pi - k\Delta\theta)$ that is also tangent to the parallel at this point is,

$$\begin{cases} x = a \sin(k\Delta\theta) \cos \tau, \\ y = a \sin \tau, \\ z = a \cos(k\Delta\theta) \cos \tau, \end{cases} \quad 0 \leq \tau < 2\pi. \tag{14}$$

Note that $y = 0$ for $\tau = 0$. We shall now express $\partial_\lambda = \partial/\partial\lambda$ in term of a derivative with respect to τ , and ∂_λ^2 in derivatives with respect to τ and θ . For this reason we want to write the angular spherical coordinates (λ, θ) as functions of τ on the tilted great circle. On the northern hemisphere we have

$$(\lambda, \theta) = (\lambda(\tau), \theta(\tau)), \quad \text{where } \theta(0) = \frac{1}{2}\pi - k\Delta\theta \quad \text{and} \quad \lambda(0) = 0.$$

Using (1) and (14) we get

$$\tan \lambda = \tan \tau / \sin(k\Delta\theta) \quad \text{and} \quad \sin \theta = \cos(\tau) \cos(k\Delta\theta)$$

and by differentiation it follows that

$$\lambda'(0) = 1/\sin(k\Delta\theta), \quad \lambda''(0) = 0 \quad \text{and} \quad \theta'(0) = 0, \quad \theta''(0) = -1/\tan(k\Delta\theta).$$

On the southern hemisphere $\tau = 0$ is replaced by $\tau = \pi$ and the last derivative changes to $\theta''(\pi) = 1/\tan(k\Delta\theta)$. Now a smooth function $F(\lambda, \theta)$ on a tilted great circle is reduced to $F(\lambda(\tau), \theta(\tau))$ and $D_\tau F = F_\lambda \lambda' + F_\theta \theta'$. Setting $\tau = 0, \pi$ we arrive at the simple formula

$$F_\lambda = \sin(k\Delta\theta) D_\tau F. \tag{15}$$

In a similar manner, differentiating twice we obtain

$$F_{\lambda\lambda} = \sin^2(k\Delta\theta) D_\tau^2 F \pm \sin(k\Delta\theta) \cos(k\Delta\theta) F_\theta, \tag{16}$$

where the $+$ and $-$ signs are chosen near the northern and the southern poles, respectively. The formula (16) will only be used for $k = 1$.

We now consider the determination of the points $*$ in Figure 2. Therefore we write down the equations for meridians and parallels, in parameter form, denoting the parameters in both cases by σ . The meridian, with constant latitude λ , is given by

$$\begin{cases} x = a \sin \sigma \cos \lambda, \\ y = a \sin \sigma \sin \lambda, \\ z = a \cos \sigma, \end{cases} \quad 0 \leq \sigma < 2\pi, \quad (17)$$

where $\sigma = 0$ corresponds to the northern pole, and the parallel with colatitude $j\Delta\theta$ by

$$\begin{cases} x = a \sin(j\Delta\theta) \cos \sigma, \\ y = a \sin(j\Delta\theta) \sin \sigma, \\ z = a \cos(j\Delta\theta), \end{cases} \quad 0 \leq \sigma < 2\pi. \quad (18)$$

Fortunately, the parameters σ and τ for the points of intersection can easily be determined. In fact, for the intersection between the tilted great circle (14) through $(\lambda, \theta) = (0, \frac{1}{2}\pi - k\Delta\theta)$ and the meridian (17), we have

$$\begin{cases} \tan \tau = \sin(k\Delta\theta) \tan \lambda, \\ \tan \sigma = \tan(k\Delta\theta) / \cos \lambda, \end{cases} \quad (19)$$

and for the intersection between the tilted great circle (14) through $(\lambda, \theta) = (0, \frac{1}{2}\pi - k\Delta\theta)$ and the parallel (18), the two parameters can be determined by

$$\begin{cases} \cos \tau = \cos(j\Delta\theta) / \cos(k\Delta\theta), \\ \cos \sigma = \tan(k\Delta\theta) / \tan(j\Delta\theta), \end{cases} \quad j > k. \quad (20)$$

When we have evaluated the τ -values, τ_1, τ_2, \dots say, on a tilted great circle, then we get $p_1 = \tau_1/\Delta\theta$, $p_2 = \tau_2/\Delta\theta, \dots$, which appear in (6), and the coefficients in the difference approximation can be computed. Before this approximation can be used for a dependent variable, the latter has to be interpolated at the points in the formula, which is done either on a meridian, with step $\Delta\theta$, or on a parallel with 32 points, i.e., with step $\Delta\lambda = \frac{1}{16}\pi$. In both cases the interpolation is centered. The PTD method is invariant under rotation of an angle, $\Delta\lambda = \frac{1}{4}\pi, \frac{1}{8}\pi$, and $\frac{1}{8}\pi$ on the first three parallels, respectively, which considerably simplifies the implementation.

We conclude this section by a simple numerical example, in which we compare different alternatives to approximate zonal derivatives near a pole. Our three alternatives are PPD, PtrigD - which means symbolic differentiation of an interpolating trigonometric polynomial - and PTD. Since high resolution seems reasonable in this context, we let $\ell = 10$, corresponding to 1024 grid-points at the equator. We choose a smooth function with fairly high variation in a vicinity of the northern pole,

$$F(x, y, z) = \sin(20(x + y + z - 1) + 200(x^2 - y^2 + (z - 1)^2)),$$

which can be expressed in terms of λ and θ by using (1). The derivative F_λ will be approximated. For PTD the order of approximation is 8 and for PPD the formula (4) is used. A relative error measure, cf. [10], for the approximative derivatives on a parallel, will be used in Tables 2 and 3. Let $\tilde{\mathbf{a}}$ be an approximation to \mathbf{a} , both in R^d , and define a relative error as

$$\max_j |\tilde{a}_j - a_j| / \max_j |a_j|. \quad (21)$$

Table 2 shows that PDT is superior for $k = 1$ and for $k = 2, 3$ also considerably better than PPD. On the second and third parallels there are 16 points and the PtrigD method is quite accurate.

TABLE 2. Relative errors (21) for approximations to F_λ

parallel k	PPD	PtrigD	PTD	PPD/PTD
1	$4.3 \cdot 10^{-3}$	$6.6 \cdot 10^{-4}$	$2.3 \cdot 10^{-6}$	1870
2	$2.9 \cdot 10^{-4}$	$4.0 \cdot 10^{-7}$	$6.1 \cdot 10^{-6}$	48
3	$9.4 \cdot 10^{-4}$	$6.7 \cdot 10^{-6}$	$4.0 \cdot 10^{-5}$	24

In many papers the pole problem, when using PPD, has been associated with high curvature of the parallel. For this reason we look at the errors in segment 3, in Table 3. The errors are

TABLE 3. Relative errors (21) as function of 'curvature'

Segment 3	PPD			
k	4	5	6	7
relative error	$2.1 \cdot 10^{-5}$	$5.3 \cdot 10^{-5}$	$1.2 \cdot 10^{-4}$	$2.4 \cdot 10^{-4}$

clearly increasing in k , i.e., when the curvature is decreasing. This can easily be understood by studying the physical steps $\cos \theta \Delta \lambda$. Curvature plays no direct part in the pole problem. The culprit is the small number of grid-points used per parallel within a segment, close to a pole, for a reduced grid. However, the problem can be overcome by using the PTD method. This method is fairly easy to implement and its influence on the total computing time negligible, in particular for high resolution.

4. NUMERICAL EXPERIMENTS

In this section we shall use our methods numerically for two test cases proposed by Williamson et al. [10], namely the cosine bell advected around the sphere and the Rossby-Haurwitz problem. All computations were carried out in double precision according to the IEEE standard. Since the cosine bell has discontinuous second order derivatives we shall also use a smooth bell, based on the function

$$sb(x) = e^{-\frac{1}{x(1-x)}}, \text{ for } 0 < x < 1, \text{ and } 0 \text{ elsewhere,}$$

which is infinitely differentiable. We will compare the different methods corresponding to PPD, PTD and PtrigD. For the scalar advection equations the difference between the methods is fairly insignificant, since PPD works quite well for these simple problems. This has been observed earlier, e.g., in [2]. However, the situation is different when the velocity vector is time dependent, i.e., when the equations of motion are involved.

Smoothing is essential for our methods to work properly and will be achieved by adding a so called 'hyperdiffusion' term to each equation. Let L be the difference operator corresponding to the five-point formula for the Cartesian form of $-\nabla^2$ and add to the equations the terms $-C(\theta)L^2u$, $-C(\theta)L^2v$ and $-C(\theta)L^2\Psi$, respectively. The smoothing function is defined as

$$C(\theta) = C_{sm} \cdot R(\theta), \quad R(\theta) = \frac{1 + b\theta}{1 - c\theta}, \quad \text{where } C_{sm} \text{ is given.} \quad (22)$$

By requiring that $\theta = \frac{\pi}{2}(1 + \epsilon)$ is an asymptote to the graph of $R(\theta)$, we obtain $c = \frac{2}{\pi}(1 + \epsilon)^{-1}$ and b can then be determined by a given value of $R(\frac{1}{2}\pi) > 1$. In the experiments below we have used $\epsilon = 0.1$ and whenever $R(\frac{1}{2}\pi)$ is *not* given, the smoothing function $C(\theta)$ equals the constant C_{sm} .

The time integration will be performed by the classical explicit, fourth order Runge-Kutta method, which is convenient to implement. The present paper deals exclusively with spatial

discretizations, and errors given here are intended to be independent of the time integration. For this reason time steps given in tables in Section 4.2 are generally much smaller than needed.

4.1. Advection of solid bodies. In [10] the advective wind is given by

$$u = u_0(\cos \theta \cos \alpha + \sin \theta \cos \lambda \sin \alpha), \quad v = -u_0 \sin \lambda \sin \alpha,$$

where the parameter α is the angle between the axis of the solid body rotation and the polar axis and $u_0 = 2\pi a/12$ m/day, which is about 40 m/s. The advective equation is simply $d\Psi/dt = 0$ or

$$\Psi_t = -\frac{u}{a \cos \theta} \Psi_\lambda - \frac{v}{a} \Psi_\theta.$$

For $\alpha = \frac{1}{2}\pi$, i.e., rotation around the poles, $u = u_0 \sin \theta \cos \lambda$, $v = -u_0 \sin \lambda$ and

$$\Psi_t = -\frac{u_0}{a} \tan \theta \cos \lambda \Psi_\lambda + \frac{u_0}{a} \sin \lambda \Psi_\theta.$$

For $\alpha = 0$, i.e., rotation around the equator, $u = u_0 \cos \theta$, $v = 0$ and

$$\Psi_t = -\frac{u_0}{a} \Psi_\lambda.$$

The two physical rotations are obviously quite similar, which cannot be said about the two initial value problems formulated above. A comparison between the two cases, $\alpha = \frac{1}{2}\pi$ and $\alpha = 0$, might still be possible and of interest. The latter because rotation around the equator with reasonably small support for the bell does neither involve the poles nor any pole problem. Accuracy of the same order, for the two cases, would be an excellent result.

In item 1 of the algorithm in Section 3.2.3 we described how interpolation can be used to obtain the pole values. However, it is also possible to use the differential equation at the poles in Cartesian coordinates (x, y) and then discretize in the most obvious way. This together with PTD is accounted for in Tables 4 and 5 under the name Ppde. The title 'All' means that all methods give about the same result. Angles are given in degrees. The value β_0 for the opening

TABLE 4. The Cosine Bell, $\ell = 8$, $\beta_0 = 120/\pi \cdot 360/256$

Method	α	β	order	$10^8 \cdot C_{sm}$			max absolute error		
PTD	90	β_0	4	18			8.62		
PPD	90	β_0	4	37	18		8.66	8.71	
Ppde	90	β_0	4	26			8.64		
All	0	β_0	4	3	18	37	8.80	8.83	8.91
PTD	90	120	8	11			0.411		
PPD	90	120	8	6			0.420		

angle β of the bell is explained in the concluding Section 5. We see at once that PTD is only slightly better than PPD for the cosine bell, which we believe to be a consequence of the fact that the latter works quite well for scalar advection problems. It can also be seen that the accuracy is actually better for rotation around the poles than around the equator. Cancellation has probably occurred to some extent, but the result is still satisfactory. For the smooth bell and high resolution, $\beta = 120$, the PTD method is significantly more accurate than PPD. But this high accuracy is probably rarely needed. Fortunately, just interpolation for the pole values turned out to be better than the more complicated Ppde method, for the two bell problems. The errors in the last columns of the tables have been minimized in C_{sm} , over suitable subsets of the real numbers, in order to make fair comparisons possible.

TABLE 5. The Smooth Bell, $\ell = 8$, $order = 8$

<i>Method</i>	α	β	$10^8 \cdot C_{sm}$	$\max absolute\ error $
PTD	90	60	3 0.3	0.596 0.676
PPD	90	60	4 1.9	0.686 0.816
Ppde	90	60	0.6	0.613
All	0	60	0.3	0.601
PTD	90	120	0.3	0.0153
PPD	90	120	1.9	0.0755
All	0	120	0.5	0.0139

4.2. **The Rossby-Haurwitz test example.** We will now turn to the Rossby-Haurwitz problem, which has been used frequently for meteorological tests since Phillips [3]. Initial condition for u , v and $\Psi = gH$ are given in [10] and not here. We point out that u , v and H only vary moderately in a proximity of the poles. Despite this we get a sharp difference between PTD and PPD and a noticeable one between PTD and P

4.2.1. *Maximal and minimal values of the height after 10 and 14 days.* In the tables below (9) has been used with only one exception, occurring in Table 7. The simpler variant (11),(12) is actually our method of choice, at least for high resolution. We have used the same smoothing function, corresponding to $C_{sm} = 12 \cdot 10^{-6}$ and $R(\frac{1}{2}\pi) = 6$, except when it was necessary for stability reasons to increase the value of $R(\frac{1}{2}\pi)$. Optimization for each case would be too costly and not very realistic, since it requires good approximations of the solutions to be known. In

TABLE 6. Rossby-Haurwitz, PTD, $n=1024$, Time=10 days, $dt=50$ s

ℓ	order	orderi	$10^6 \cdot C_{sm}$	$R(\pi/2)$	min H	max H
10	10	10	12	6	8208.447	10536.433
	8	8		6	8208.433	10536.420
	6	6		6	8208.394	10536.404
	4	4		8	8208.193	10536.350

Table 6 orderi means order of approximation for the interpolation between adjacent segments. Note that both min H and max H are increasing functions in the order of the method, which is a regular and robust behavior. The following approximations will be used as answers whenever needed

$$\min H \approx 8208.5, \quad \max H \approx 10536.5 \tag{23}$$

The first four lines in Table 7 correspond to Table 6, but with $\ell = 9$ instead of $\ell = 10$. We see

TABLE 7. Rossby-Haurwitz, PTD, $n=512$, Time=10 days, $dt=100$ s

No.	ℓ	order	orderi	$10^6 \cdot C_{sm}$	$R(\pi/2)$	min H	max H
1	9	10	10	12	6	8208.629	10537.290
2		8	8		6	8208.956	10537.329
3		6	6		6	8209.388	10537.390
4		4	4		8	8212.634	10537.750
5		8	10		6	8208.835	10537.312
6(7)		8	8		6	8208.952	10537.331
7		8	8		6	8208.919	10537.319

the same regular pattern but now $\min H$ and $\max H$ are decreasing with the order instead. In the last three lines we have tested some variants of line 2. In line 5 the order of approximation for the interpolation is 10 instead of 8. This has increased the accuracy but only slightly. In line 6(7) we have used PTD on 7 parallels by using 64 points instead of the earlier number, namely 32. Table 7 indicates that PTD on 3 parallels seems to be sufficient. We cannot give any guarantee for this for other problems, more experiments are needed. Finally, in line 7 the simplified variant (11),(12) has been used instead of (9), and by comparing with line 2 we see that line 7 is rather better than worse for the present problem, which makes our choice easy. Table 8 contains our most important experimental comparisons between three different methods

TABLE 8. Comparison: PTD, PPD and PtrigD, Time=10 days

Method	ℓ	dt	order	$10^6 \cdot C_{sm}$	$R(\pi/2)$	$\min H$	$\max H$
PTD	8	200 s	8	12	6	8202.381	10508.508
	9	100 s				8208.956	10537.329
	10	50 s				8208.433	10536.420
PPD	8	200 s	8	12	6	8303.046	10525.319
	9	100 s				8223.945	10533.934
	10	50 s				8211.481	10534.641
PtrigD	8	200 s	8	12	6	8202.421	10508.487
	9	100 s			8	8210.812	10537.516
	10	50 s			14	8208.206	10536.361

at the poles and three different resolutions, $\ell = 8, 9$ and 10 , for the Rossby-Haurwitz problem. PtrigD stands for symbolic differentiation of interpolating trigonometric polynomials. Note that the PTD and PtrigD methods are used only for the three parallels closest to a pole. Since $\min H$ is attained at the poles, we focus on the approximations to this value given in the table. Note also that PtrigD required much more smoothing than the other two, which must be considered a drawback. We see a significant difference between the PTD and the PPD methods at the poles. For the lowest resolution $\ell = 8$ the difference is quite big. The differences are decreasing with increasing resolution, which can be seen as a consequence of the fact that even PPD methods, not surprisingly, give better accuracy for higher resolutions. It is advantageous that PTD is better than or equal to PtrigD for all resolutions, since, in general, trigonometric differentiation requires shorter time steps than ordinary difference approximations.

TABLE 9. Rossby-Haurwitz, PTD, Time=14 days

ℓ	dt	order	$10^6 \cdot C_{sm}$	$R(\pi/2)$	$\min H$	$\max H$
10	50	10	12	6	8111.104	10539.532
		8			8111.223	10539.523
9	100	10	12	6	8146.497	10526.041
		8			8146.019	10525.785

We now turn to Table 9. It is reported in [2] that two reductions per hemisphere can lead to serious growth of the errors, starting after about 10 days. For this reason we have integrated for 14 days, with two different resolutions, $\ell = 10$ and $\ell = 9$, and two orders of approximation, namely 10 and 8. The highest resolution is the most relevant for weather forecasts. The table and Figure 3 indicate that we do not have the same troubles with our methods as those reported in [2]. Let us assume that a substantial amount of accuracy has been lost when switching from

$\ell = 10$ to $\ell = 9$. This implies, for the case $\ell = 9$, that the relative errors for $\min H$ and $\max H$ are about 0.4% and 0.1%, respectively, which seem to be quite modest.

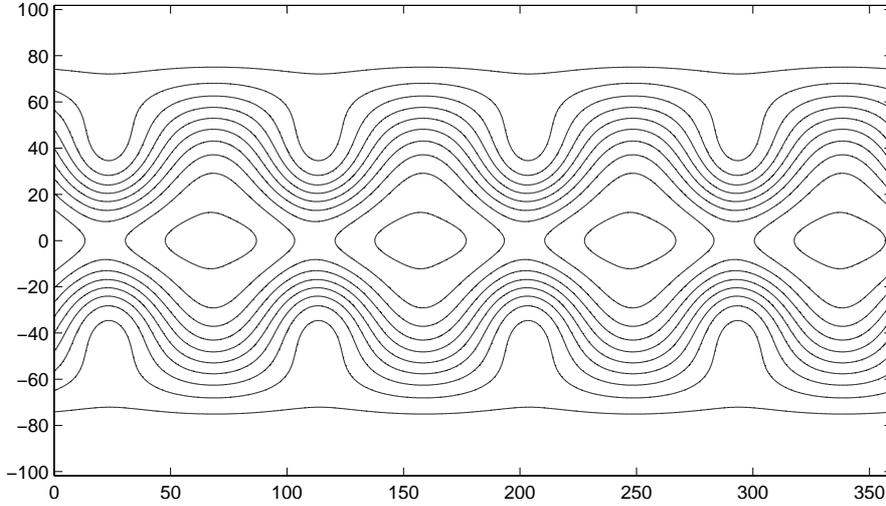


Figure 3. Contour curves for the geopotential for the Rossby-Haurwitz test example after 14 days, from 8100 to 10500 by 240. More details in the first row of Table 9.

4.2.2. *Conservation properties for mass and energy.* Let S be the segment on the sphere defined by $\theta_1 \leq \theta \leq \theta_2$, where, e.g., $\theta_1 = 0$ and $\theta_2 = \frac{1}{2}\pi$ correspond to the northern hemisphere. Our measures for mass and energy on S are

$$\iint_S H dA = a^2 \int_{\theta_1}^{\theta_2} \cos \theta \left(\int_0^{2\pi} H(\lambda, \theta) d\lambda \right) d\theta$$

and

$$\iint_S H(\mathbf{v} \cdot \mathbf{v} + H) dA = a^2 \int_{\theta_1}^{\theta_2} \cos \theta \left(\int_0^{2\pi} H(\mathbf{v} \cdot \mathbf{v} + H) d\lambda \right) d\theta,$$

respectively. The integrals are computed numerically, applying the trapezoidal rule in the λ direction and by using interpolating cubic splines in the θ direction. The answers are obtained by using the values for u , v and H corresponding to $\ell = 10$, and approximations by the lower resolution $\ell = 9$. In Table 10 the latitudes are expressed in degrees and rel_{mass} denotes the

TABLE 10. Rossby-Haurwitz, PTD, Time=10 days, dt=100 s

ℓ	$\theta_1 - \theta_2$	order	$10^6 \cdot C_{sm}$	$R(\pi/2)$	rel_{mass}	rel_{energy}
9	0 - 40	8	12	6	-0.0003	-0.0010
	40 - 60				0.0001	-0.0020
	60 - 80				0.0010	0.0010
	80 - 90				-0.0003	-0.0005
9	0 - 90	8	12	6	-0.0001	-0.0010

relative error for the mass, and analogously for the energy. If, for example, an approximation of the energy is too small, then $\text{rel}_{energy} < 0$. The largest value of rel_{mass} deviates somewhat from the rest of the values in the same column. This might simply depend on the variation of H in λ and θ . Anyhow, the relative errors in Table 10 are fairly small.

5. CONCLUDING REMARKS

Below we propose some investigations, which would constitute a reasonable continuation of this paper.

It would be desirable to investigate *additional test examples*. We have earlier expressed concern about the fact that the solutions to the Rossby-Haurwitz test example vary fairly little in a proximity of a pole. In line 2 in Table 7 we have used PTD on 3 parallels and in line 6, on 7 parallels. A comparison between the pole values for the height shows a difference first in the 7th decimal. However, for other problems with more rapid variation at the poles, we cannot exclude that the situation is different. The use of PTD on 3 or 7 parallels makes little relative change in the overall computing time, for high resolution.

A challenging problem for the segmented methods would be to study *finer segmentations*. A more uniform grid in the physical mesh-lengths leads to a smaller distortion factor (3). This makes it possible to increase the time step without losing accuracy. We recall that a fairly uniform grid is generally considered to be favorable for wave propagation problems. We think it is possible to reduce the computing time by a couple of percent by using a segmentation according to Table 11, which would be a satisfactory side effect.

We now briefly consider the construction of more uniform grids. We believe it is important that the interpolation between adjacent segments is simple and repetitive. Suppose we want to increase the zonal mesh-length by a factor $q = \frac{3}{2}$ or $q = \frac{4}{3}$, instead of by $q = 2$ as earlier. This means that the number of grid-points on a parallel, n_p say, reduces to $\frac{2}{3}n_p$ or $\frac{3}{4}n_p$, respectively, but these numbers have to be integers. In the first case, three mesh-rectangles would be replaced by two and for the second four by three. When changing mesh-lengths, interpolation must now be done both towards the pole and towards the equator. For the special case $n = 2^\ell$ we first use the factor $\frac{4}{3}$ and then $\frac{3}{2}$, which is then repeated. The quantity d_{max} appearing in Section 2 and used for the determination of the segments is now $\sqrt{1.5}$, instead of $\sqrt{2}$. If n contains factors 3 as well it seems quite easy to handle the situation. Many other values of n are possible without introducing a general coupling between two segments.

Table 11 describes two segmentations that differ only at the poles. The one to the left requires nothing new at the poles, compared to Section 3, whereas the one to the right needs further study to be useful. We also note that the distortion factors for the grid systems are the two numbers 1.27 and 1.22, respectively. The number of interpolations is about 4 times greater than for the case $q = 2$. For $\ell = 10$ and $q = 2$, in one of our experiments, the relative computing time for the interpolation between adjacent segments was about 0.66%. We recall that only one-dimensional interpolation is used, cf. the end of Section 2.

We now briefly consider *overlapping grid techniques* [1,7] as a possible alternative to the PTD method. Suppose a segmented grid is used except at small vicinities of the poles, where we use overlapping orthogonal, almost uniform grids. If this is done in such a way that the polar grids only cover a small fixed number of parallels close to the poles, then both scalability problems and extensive interpolation can be avoided. A combination of overlapping and patching will probably be more efficient. We do not expect insufficient conservation of mass for NWP, cf. [7]. Experiments are needed for the ultimate design.

In Ronchi et al. [5], an interesting gridding technique is introduced by combining the cubed sphere decomposition with overlapping grids. They used grids for each subregion such that only one-dimensional interpolation between adjacent curved rectangles was needed, which is a very nice property in this context. The methods do not involve the pole problem in any way. An experiment for the cosine bell with an opening angle $\beta = 120/\pi$, using a fourth order method

with 360 points on the equator, gave the maximal error 10.17. In Table 4 we used 256 points and with opening angle $\beta_0 = (360/256) \cdot 120/\pi$, and found the maximal error 8.62. Presently, we are restricted to $n = 2^\ell$. We will make no further comparison but want to add a quotation from Ronchi et al. [5]: 'Although arguably the most natural coordinates for representing phenomena on a sphere, spherical polar coordinates present several disadvantages (collectively known as the pole problem) when used for numerical computations over the entire spherical surface.' However, if the pole problem were solved, which already may be the case, then grids based on spherical coordinates seem to be the most natural choice.

TABLE 11. Structures of grid systems with $q=4/3$, $q=3/2$ and $n=1024$.

$\ell = 10$	1.27			$\ell = 10$	1.22		
segment	interval	width	length	segment	interval	width	length
1	1-100	100	1024	1	1-100	100	1024
2	101-148	48	768	2	101-148	48	768
3	149-187	39	512	3	149-187	39	512
4	188-205	18	384	4	188-205	18	384
5	206-222	17	256	5	206-222	17	256
6	223-230	8	192	6	223-230	8	192
7	231-239	9	128	7	231-239	9	128
8	240-243	4	96	8	240-243	4	96
9	244-247	4	64	9	244-247	4	64
10	248-249	2	48	10	248-249	2	48
11	250-252	3	32	11	250-251	2	32
12	253-254	2	16	12	252-252	1	24
13	255-255	1	8	13	253-253	1	18
				14	254-254	1	12
				15	255-255	1	6

REFERENCES

- [1] Browning, B.L., Hack, J.J., Swarztrauber, P.N., (1989), A comparison of three numerical methods for solving differential equations on the sphere, *Mon. Wea. Rev.*, 117(5), pp.1058-1075.
- [2] Jablonowski, C., Oehmke, R., Stout, Q.F., (2009), Block-structured adaptive meshes and reduced grids for atmospheric general circulation models, *Philos. Trans. R. Soc., A* 367 1907, pp.4497-4522.
- [3] Phillips, N.A., (1959), Numerical integration of the primitive equations on the hemisphere, *Mon. Wea. Rev.*, 87(9), pp.333-345.
- [4] Purser, R.J., (1988), Accurate numerical differencing near a polar singularity of a skipped grid, *Mon. Wea. Rev.*, 116(5), pp.1067-1076.
- [5] Ronchi, C., Iacono, R., Paulucci, P.S., (1996), The 'cubed sphere': A new method for the solution of partial differential equations in spherical geometry, *J. Comput. Phys.*, 124(1), pp.93-114.
- [6] Staniforth, A., Thuburn, J., (2012), Horizontal grids for global weather and climate prediction models: a review, *Q.J. Meteorol. Soc.*, 138, Issue 662, pp.1-26.
- [7] Starius, G., (1980), On composite mesh difference methods for hyperbolic differential equations, *Numer. Math.*, 35(3), pp.241-255.
- [8] Tolstykh, M.A., Shashkin, V.V., (2012), Vorticity-divergence mass-conserving semi-Lagrangian shallow-water model using the reduced grid on the sphere, *J. Comput. Phys.*, 231(11), pp.4205-4233.
- [9] Ullrich, P.A., Lauritzen, P.H., Jablonowski, C., (2010), A high-order fully explicit 'incremental-remap' semi-Lagrangian shallow-water model, *Int. J. Numer. Meth. Fluids*, 2010. Published online in Wiley Interscience (www.interscience.com). DOI:10.1002/fld.

- [10] Williamson, D.L., Drake, J.B., Hack, J.J., Jacob, R., Swarztrauber, P.N., (1992), A standard test set for numerical approximation to the shallow water equations in spherical geometry, *J. Comput. Phys.*, 102(1), pp.211-224.
- [11] Williamson, D.L., (2007), The Evolution of Dynamical Cores for Global Atmospheric Model, *J. Meteorol. Soc. Jpn.*, Ser.II 85B, special issue, pp.241-269.
-
-



Göran C. Starius - recieved his Ph.D. in Computational Mathematics at Uppsala University, Sweden, in 1976. Became associate professor at Chalmers University of Technology, in 1986. Main scientific interest is numerical solution of partial differential equations. In particular: Overlapping grid methods, Path following and bifurcation, and presently Pde on a sphere.