

An adaptive wavelet collocation method for the solution of partial differential equations on the sphere

Mani Mehra^{*}, Nicholas K.-R. Kevlahan

Department of Mathematics and Statistics, McMaster University, Hamilton, ON, Canada L8S 4K1

Received 6 December 2006; received in revised form 3 February 2008; accepted 5 February 2008

Available online 15 February 2008

Abstract

A dynamic adaptive numerical method for solving partial differential equations on the sphere is developed. The method is based on second generation spherical wavelets on almost uniform nested spherical triangular grids, and is an extension of the adaptive wavelet collocation method to curved manifolds. Wavelet decomposition is used for grid adaption and interpolation. An $O(N)$ hierarchical finite difference scheme based on the wavelet multilevel decomposition is used to approximate Laplace–Beltrami, Jacobian and flux-divergence operators. The accuracy and efficiency of the method is demonstrated using linear and nonlinear examples relevant to geophysical flows. Although the present paper considers only the sphere, the strength of this new method is that it can be extended easily to other curved manifolds by considering appropriate coarse approximations to the desired manifold (here we used the icosahedral approximation to the sphere at the coarsest level).
© 2008 Elsevier Inc. All rights reserved.

Keywords: Wavelets; Lifting scheme; Second generation wavelets; Partial differential equations; Spherical triangulation; Adaptive grid; Numerical method

1. Introduction

Wavelet methods have been used to develop accurate and fast algorithms for integral and differential equations, especially those whose solutions are highly localized in position and scale. The numerical solution of such problems on uniform grids is impractical since high resolution computations are required only in regions where sharp transitions occur. However, until now these adaptive wavelet methods have been limited to flat geometries (i.e. manifolds with zero curvature).

The efficient numerical solution of partial differential equations (PDEs) defined on the sphere is necessary in geodesy, climate modelling, and numerical weather prediction. Current climate and weather models are not adaptive, which means local features (e.g. hurricanes) are not resolved properly and must be parameterized [1]. Other PDEs on the sphere arise from mean curvature flow, surface diffusion flow [2], Willmore flow [3] and the reaction–diffusion equations modeling morphogenesis [4,5]. These PDEs are often solved on spherical

^{*} Corresponding author.

E-mail addresses: mmehra@math.mcmaster.ca (M. Mehra), kevlahan@mcmaster.ca (N.K.-R. Kevlahan).

triangular grids. Spherical triangular grids avoid the so-called pole problem that exists in conventional latitude–longitude grids. The singularities at the poles lead to a variety of numerical difficulties, including a severe limitation on the time step size unless special measures are undertaken (such as local re-scaling of the equations). Because they avoid these problems, quasi-uniform triangulations are gaining popularity in the climate modelling community. These almost uniform spherical triangular grids are the fundamental idea behind second generation spherical wavelets. We limit ourselves here to the construction of bases arising from interpolating subdivision schemes.

In recent years, there has been a growing interest in developing wavelet based adaptive numerical algorithms for solving PDEs [6–10]. In flat geometries the adaptive wavelet collocation method (AWCM) was pioneered by Bertoluzza [11] for time-independent problems, and has since been developed for more general problems using second generation wavelets [12–15]. This approach combines the adaptivity and error control of the adaptive wavelet method with the flexibility of collocation. Wavelets are used to adapt the computational grid (and hence compress the solution), while finite differences are used to approximate derivatives. The accuracy of both the solution and derivatives are controlled by a tolerance parameter ϵ . The objective of the present work is to extend the AWCM to PDEs defined on the sphere, or, more generally, on curved manifolds.

There are two classes of numerical methods for solving PDEs on curved surfaces. In the *explicit* approach considered here the surface is approximated explicitly using Delaunay triangulation and Voronoi diagrams. This approach has been popular in computer graphics [16,17], climate modelling [18] and aeronautical engineering [19]. In the *implicit* approach [20] the surface is first embedded in a higher dimensional computational space (e.g. a 2D surface is embedded in a flat 3D space) using an appropriate embedding function. The location of the surface is defined as a level set of the embedding function. The embedded PDE is then solved on a fixed Cartesian grid in the higher dimensional space using any suitable method (e.g. finite differences or particle methods), but only in a small band near the level set.

Implicit (or level set) methods have two principal advantages compared with explicit methods. First, they inherit the simplicity, efficiency and good convergence properties of numerical methods defined on Cartesian grids. Secondly, they deal easily with surfaces whose shape or topology changes since the shape of the surface is encoded by a (time-dependent) embedding function and thus the computational grid does not need to change. This feature is especially important for biological applications such as cell morphogenesis. The level set method also allows surfaces to be constructed from experimentally derived image data [21]. In an interesting recent paper Bergdorf and Koumoutsakos [22] have combined a wavelet multi resolution analysis with a particle-based implicit method to solve PDEs on one-dimensional curves embedded in a flat two-dimensional space. This work shows that implicit methods can be made adaptive. However, the computational overhead could be significant for adaptive implicit methods on two-dimensional surfaces since the adaptivity must necessarily take place in the three-dimensional embedding space. In addition, as noted by Bergdorf & Koumoutsakos [22], for nonlinear problems areas of high strain could lead to artificially small time steps.

While explicit methods based on triangulated surfaces are simple and fast, they often have a lower order of accuracy and less stability than methods based on Cartesian grids. However, in this paper we show that the convergence error can be controlled using adaptive wavelet refinement of the triangulated grid, as in the case of Cartesian grids. In the case of the sphere, using the optimal spherical geodesic grid [23] ensures convergence in terms of local mean curvature. The optimal spherical grid also has a lower truncation error than conventional spherical geodesic grids. We will show that our adaptive explicit method has no difficulty solving nonlinear problems.

On balance, adaptive explicit methods, such as the one presented in this paper, are usually preferred when high accuracy is required on fixed, relatively simple, surfaces. On the other hand, an implicit method is usually the best choice when the shape or topology of the surface is changing.

A multiresolution analysis (MRA) of the sphere using second generation wavelets was proposed over 12 years ago by Schroder & Sweldens [24]. However, until now no one has attempted to use this MRA as the basis for an adaptive PDE solver. In fact, the MRA on the sphere and other curved manifolds has been used almost exclusively by the computer graphics community for surface mesh refinement (e.g. [16,25]).

Adaptive solution of PDEs on curved manifolds requires a non-trivial extension of adaptive wavelet methods for flat space. The main difficulty is due to the fact that we require simultaneous convergence of the

approximations to both the differential operator and the local geometry of the surface (e.g. local mean curvature) as the mesh is refined. The convergence of differential operators on curved manifolds was not well understood until recently. In fact, some commonly used approximations have been shown to be non-convergent [23]. For this reason, even non-adaptive numerical methods for PDEs on the sphere [5] and on other curved manifolds [20,21,26] remain an active area of research.

The spherical advection equation in flux form is the simplest model of large scale atmospheric dynamics, and is a special case of the shallow water equations. The shallow water equations have been solved previously on non-adaptive spherical grids using finite difference and finite volume methods [18,27,28,5]. However, these methods are nonadaptive. Our adaptive wavelet method provides local grid refinement and error control. This approach should allow the climate modelling community to perform large scale simulations of of atmosphere and ocean dynamics. The present method is particularly well suited to problems which require resolution of localized structures with both very small and very large scales (e.g. long distance propagation of Tsunami waves). Current simulations do not make optimal use of computational resources because adaptive error-controlled methods are not used in operational weather forecasting and climate modelling. We focus here on spatial discretization using adaptive wavelet methods. Improved time integration schemes will be developed in future work. To the best of our knowledge this paper is the first attempt to solve PDEs on the sphere using the AWCM.

Atmospheric turbulence is approximately two-dimensional, and is therefore a good application for the spherical AWCM. Past research in the application of wavelets to two-dimensional turbulence been restricted to flat geometries, which is severely limiting for geophysical applications. In this paper we also measure how well the spherical AWCM compresses two-dimensional turbulence on the sphere. Nonlinear wavelet filtering splits the flow field into coherent and incoherent parts, corresponding to the weak and strong wavelet coefficients, respectively. This filtering is the basis of coherent vortex simulation (CVS), introduced in [29]. CVS separates the Gaussian (noise) and non-Gaussian (coherent) parts of a two-dimensional turbulent flow, and is an efficient method for computing and modeling two-dimensional turbulence [29,30]. In this paper we are taking the first step towards developing CVS for two-dimensional turbulence on the sphere.

The paper is organized as follows. Section 2 gives a brief introduction to second generation wavelets on the sphere. The AWCM for solving spherical PDEs is introduced in Section 3. An application of the method to the compression of turbulence flow data concludes this section. Finally, Section 4 presents numerical results for the solution of a variety of linear and nonlinear PDEs. The main results are summarized and future directions are outlined in Section 5.

2. Wavelets on the sphere

2.1. Subdividing the sphere

The construction of spherical wavelets in [24] relies on a recursive partitioning of the sphere into spherical triangles. This is done starting from a platonic solid whose faces are spherical triangles. Fig. 1 shows this subdivision scheme for several levels, beginning with the icosahedron. Choosing the icosahedron as the starting point, the resulting triangulation has the least imbalance in area between its constituent triangles. Such imbalances, starting from the tetrahedron or octahedron, can lead to visible artifacts. Here we will consider only the icosahedral subdivision for which $\#\mathcal{K}^j = 10 \times 4^j + 2$ at subdivision level j .

Let S be a triangulation of the sphere S and denote the set of all vertices obtained after subdivisions with $\mathcal{S}^j = \{p_k^j \in S | k \in \mathcal{K}^j\}$, where \mathcal{K}^j is an index set. The vertices of the original platonic solid are in \mathcal{S}^0 , \mathcal{S}^1 contains those vertices and all new vertices on the edge midpoints. Since $\mathcal{S}^j \subset \mathcal{S}^{j+1}$ we also let $\mathcal{K}^j \subset \mathcal{K}^{j+1}$. Let $\mathcal{M}^j = \mathcal{K}^{j+1} / \mathcal{K}^j$ be the indices of the vertices added when going from level j to $j + 1$.

2.2. Second generation wavelets

Wavelets are basis functions which represent a given function at multiple levels of detail. Due to their local support in both space and frequency, they provide sparse approximations of functions. Locality in space follows from their compact support, while locality in frequency follows from their smoothness and vanishing

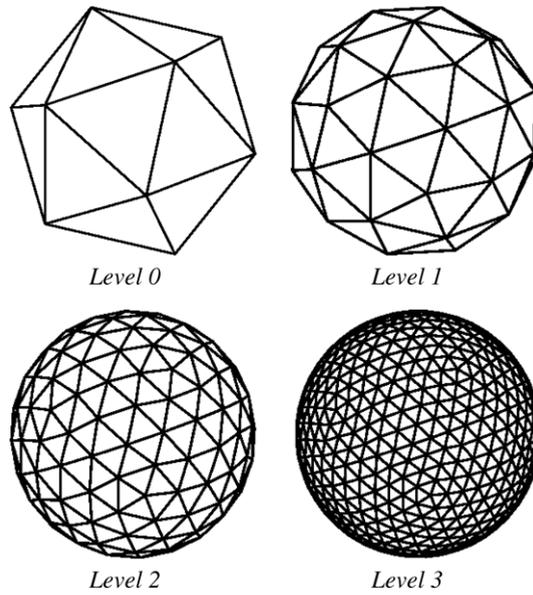


Fig. 1. Dyadic icosahedral triangulation of the sphere.

moment properties. Fast $O(\mathcal{N})$ algorithms (where \mathcal{N} is the number of significant wavelet coefficients) exist to calculate the wavelet coefficients. These properties make wavelets efficient for many computational problems.

In the classical wavelet setting, i.e. on the real line, wavelets are defined from the dyadic translates and dilates of a scaling function. The idea behind second generation wavelets is to build wavelets which retain desirable properties like localization and fast wavelet transform, but adapted to a much more general setting than the real line, e.g. complex geometries or curved manifolds. In order to consider wavelets on a manifold we need to construct wavelets adapted to the desired surface.

Adaptive construction relies on the observation that translation and dilation are not fundamental to obtain wavelets with the desired localization properties. Classical wavelets and their corresponding filters are constructed via the Fourier transform. This is because translation and dilation become algebraic operations after Fourier transform. In the setting of second generation wavelets, translation and dilation invariance are lost, and the Fourier transform method can no longer be used. In order to define second generation wavelets, we first introduce the multiresolution analysis (following [31]).

A second generation multiresolution analysis of the sphere provides a sequence of subspaces $\mathcal{V}^j \subset L_2(S)$, with $j \geq 0$ and sphere $S = \{p = (p_x, p_y, p_z) \in \mathbb{R}^3 : \|p\| = a\}$ such that

- $\mathcal{V}^j \subset \mathcal{V}^{j+1}$ (subspaces are nested),
- $\bigcup_{j \geq 0} \mathcal{V}^j = L_2(S)$,
- Each \mathcal{V}^j has a Riesz basis of scaling functions $\{\phi_k^j | k \in \mathcal{K}^j\}$,

Since $\phi_k^j \in \mathcal{V}^j \subset \mathcal{V}^{j+1}$, for every scaling function ϕ_k^j filter coefficients $h_{k,l}^j$ exists such that

$$\phi_k^j = \sum_{l \in \mathcal{K}^{j+1}} h_{k,l}^j \phi_l^{j+1}. \tag{1}$$

Note that the filter coefficients $h_{k,l}^j$ can be different for every $k \in \mathcal{K}^j$ at a given level $j \geq 0$. Therefore each scaling function satisfies a different refinement relation. Each multiresolution analysis is accompanied by a dual multiresolution analysis consisting of nested spaces $\tilde{\mathcal{V}}^j$ with bases given by the dual scaling functions $\tilde{\phi}_k^j$, which are biorthogonal to the scaling functions:

$$\langle \phi_k^j, \tilde{\phi}_{k'}^j \rangle = \delta_{k,k'} \quad \text{for } k, k' \in \mathcal{K}^j, \tag{2}$$

where $\langle f, g \rangle = \int_S fgdw$ is the inner product on the sphere. The dual scaling functions satisfy refinement relations with coefficients $\tilde{h}_{k,l}^j$.

One of the crucial steps when building a multiresolution analysis is the construction of the wavelets. They encode the difference between two successive levels of representation, i.e. they form a Riesz basis for the space \mathcal{W}^j , which is the complement of \mathcal{V}^j in \mathcal{V}^{j+1} (i.e. $\mathcal{V}^{j+1} = \mathcal{V}^j \oplus \mathcal{W}^j$). By construction the wavelets form a Riesz basis for $L^2(S)$ and allow a function to be represented by its wavelet coefficients. Since $\mathcal{W}^j \subset \mathcal{V}^{j+1}$, we have

$$\psi_k^j = \sum_{l \in \mathcal{K}^{j+1}} g_{k,l}^j \phi_l^{j+1}. \tag{3}$$

The spherical wavelets ψ_m^j have \tilde{d} vanishing moments if \tilde{d} independent polynomials $P_i, 0 \leq i < \tilde{d}$ exist such that

$$\langle \psi_m^j, P_i \rangle = 0 \quad \forall j \geq 0, m \in \mathcal{M}^j, \tag{4}$$

where \mathcal{M}^j is an index set. Here the polynomials P_i are defined as the restriction to the sphere of polynomials on \mathbb{R}^3 .

2.3. Interpolating subdivision schemes and fast wavelet transform

Scaling functions $\{\phi_k^j | j \geq 0, k \in \mathcal{K}^j\}$ are called interpolating if a set of points $\{p_k^j | j \geq 0, k \in \mathcal{K}^j\}$ with $p_k^j = p_{2k}^{j+1}$ exists, so that

$$\forall k, k' \in \mathcal{K}^j : \phi_k^j(p_{k'}) = \delta_{k,k'}. \tag{5}$$

In the case of interpolating scaling functions, we can always take the dual scaling functions to be Dirac distributions, $\tilde{\phi}_k^j(p) = \delta(p - p_k^j)$, which are necessarily biorthogonal. The set of filters resulting from interpolating scaling functions (with Diracs as their formal dual), can be seen as a dual lifting of the Lazy wavelet [31]. So we have an interpolating multiresolution analysis which is dually lifted from the Lazy wavelet. The Lazy wavelet transform [31] is an orthogonal transform that simply subsamples the coefficients. The filters of the Lazy wavelet transform are given as

$$h_{k,l}^j = \tilde{h}_{k,l}^j = \delta_{k,l} \quad \text{and} \quad g_{k,l}^j = \tilde{g}_{k,l}^j = \delta_{m,l}.$$

For a vertex-based scheme we may think of the grid points $m \in \mathcal{M}^j$ as located on the midpoint of some parent edge, while the endpoints of a given edge form the even indices $k \in \mathcal{K}^j$, and their union is $l \in \mathcal{K}^j \cup \mathcal{M}^j = \mathcal{K}^{j+1}$ given the set of all indices. We consider all index sets as defined locally around a point p_m^{j+1} which is locally denoted by $m \in \mathcal{M}^j$ in Fig. 2. For each m the filters range only over some small neighborhood. We will refer to elements in these neighborhoods by a local naming scheme, $k \in \mathcal{K}_m \subset \mathcal{K}^j$. For all vertex bases the unlifted scaling coefficients are simply subsampled during analysis and upsampled during synthesis, while finding the wavelet coefficients involves some additional computation.

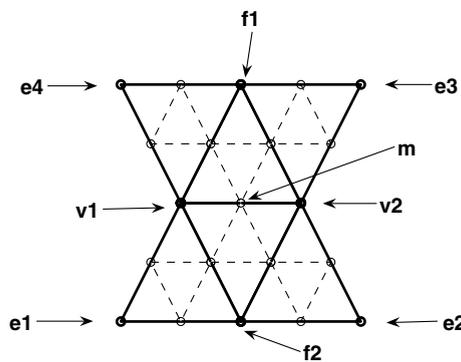


Fig. 2. The members of the neighborhoods used in wavelet bases ($m \in \mathcal{M}^j, \mathcal{K}_m = \{v_1, v_2, f_1, f_2, e_1, e_2, e_3, e_4\}$). The open circles and dashed lines represent the grid at the next finer level of resolution $j + 1$.

Analysis I(j) :

$$\begin{aligned} \forall k \in \mathcal{K}^j : c_k^j &= c_k^{j+1}, \\ \forall m \in \mathcal{M}^j : d_m^j &= c_m^{j+1} - \sum_{k \in \mathcal{K}_m} \tilde{s}_{k,m}^j c_k^j, \end{aligned} \tag{6}$$

Synthesis II(j) :

$$\begin{aligned} \forall k \in \mathcal{K}^j : c_k^{j+1} &= c_k^j, \\ \forall m \in \mathcal{M}^j : c_m^{j+1} &= d_m^j + \sum_{k \in \mathcal{K}_m} \tilde{s}_{k,m}^j c_k^j. \end{aligned} \tag{7}$$

The linear interpolatory basis uses the stencil $k \in \mathcal{K}_m = \{v_1, v_2\}$ (see Fig. 2) for analysis and synthesis

$$\begin{aligned} d_m^j &:= c_m^{j+1} - \frac{1}{2}(c_{v_1}^{j+1} + c_{v_2}^{j+1}), \\ c_m^{j+1} &:= d_m^j + \frac{1}{2}(c_{v_1}^{j+1} + c_{v_2}^{j+1}). \end{aligned} \tag{8}$$

The resulting scaling function is the hat function which is continuous, but not differentiable. The butterfly basis uses a stencil with eight neighboring points (see Fig. 2) and is differentiable if a smooth map exists from the sphere to a regular planar triangulation. The filter coefficients for the butterfly basis are $\tilde{s}_{v_1} = \tilde{s}_{v_2} = 1/2$, $\tilde{s}_{f_1} = \tilde{s}_{f_2} = 1/8$, $\tilde{s}_{e_1} = \tilde{s}_{e_2} = \tilde{s}_{e_3} = -1/16$, as proposed by Dyn et al. [17].

A disadvantage of this multiresolution analysis is that it cannot provide Riesz bases for $L_2(S)$. The dual functions do not even belong to $L_2(S)$. It turns out that the wavelet does not have a vanishing integral since it coincides with a scaling function. One can apply the primal lifting scheme to overcome this drawback by ensuring that the primal wavelet has at least one vanishing moment. This yields

$$\begin{aligned} \tilde{h}_{k,l}^j &= \delta_{k,l} + \sum_m s_{k,m}^j \tilde{g}_{m,l}^j, \\ g_{m,l}^j &= \delta_{m,l} - \sum_m s_{k,m}^j h_{k,l}^j. \end{aligned} \tag{9}$$

The resulting wavelet can be written as

$$\psi_m^j = \phi_m^{j+1} - \sum_{k \in \mathcal{K}_m = \{v_1, v_2\}} s_{k,m}^j \phi_k^j. \tag{10}$$

The weights $s_{k,m}^j$ are chosen so that resulting wavelet has a vanishing integral

$$s_{k,m}^j = I_m^{j+1} / 2I_k^j \quad \text{with} \quad I_k^j = \int \int_S \phi_k^j dw. \tag{11}$$

During analysis lifting is done (at each level j) after the d_m^j computation, while during synthesis it is the first step, followed by the regular synthesis step.

Analysis II(j) :

$$\forall m \in \mathcal{M}^j, \quad \forall k \in \mathcal{K}_m = \{v_1, v_2\} : c_k^j = c_k^j + s_{k,m}^j d_m^j, \tag{12}$$

Synthesis I(j) :

$$\forall m \in \mathcal{M}^j, \quad \forall k \in \mathcal{K}_m = \{v_1, v_2\} : c_k^j = c_k^j - s_{k,m}^j d_m^j.$$

Lifting does not improve compression, but reduces error for the same compression rate.

3. Adaptive wavelet collocation method

3.1. Grid adaptation

The major strength of wavelet decomposition is its ability to compress functions. In comparison with other common bases for function spaces on the sphere, such as spherical harmonic or spherical splines, wavelet

bases allow for simultaneous localization in both space and frequency. For functions containing isolated small scales on a large-scale background (i.e. intermittent functions) most wavelet coefficients are small. These small wavelet coefficients can be neglected, thus compressing the function. By contrast, such functions have slowly decaying spectral coefficients, and therefore truncation of the spherical harmonics leads to spurious oscillations. The spherical spline basis coefficients cannot be truncated efficiently.

Grid adaptation occurs naturally in wavelet methods (e.g. [8,9]). Note that all the adaptive wavelet algorithms cited in the literature can be considered from two quite different points of view. First, they can be considered as providing compression on some fine, but fixed, level: many wavelet coefficients are small and can be replaced by zero. This is the point of view taken by wavelet algorithms for data compression, or for the sparse representation of non-local operators such as integral of pseudo differential operators. Secondly, they can be considered to be closely related to classical h -versions of adaptive finite element methods, i.e. higher accuracy of the approximation is achieved by locally refined grids instead of by increasing the degree of the trial functions (p -refinement). We will take the second point of view in order to exploit the localization properties of the wavelet multiscale basis to develop an adaptive multiscale algorithm for PDEs on the sphere.

Following the construction of second generation spherical wavelets in Section 2.2, a function $u(p) \in L_2(S)$ can be approximated as

$$u(p) = \sum_{k \in \mathcal{K}^0} c_k^0 \phi_k^0(p) + \sum_{j=0}^{\infty} \sum_{m \in \mathcal{M}^j} d_m^j \psi_m^j(p). \tag{13}$$

More precisely, we can rewrite Eq. (13) as the sum of two terms composed of wavelets whose amplitudes are, respectively, above and below some prescribed threshold ϵ ,

$$u(p) = u_{\geq}(p) + u_{<}(p), \tag{14}$$

where

$$u_{\geq}(p) = \sum_{k \in \mathcal{K}^0} c_k^{J_0} \phi_k^{J_0}(p) + \sum_{j=J_0}^{\infty} \sum_{\substack{m \in \mathcal{M}^j \\ |d_m^j| \geq \epsilon}} d_m^j \psi_m^j(p), \tag{15}$$

$$u_{<}(p) = \sum_{j=J_0}^{\infty} \sum_{\substack{m \in \mathcal{M}^j \\ |d_m^j| < \epsilon}} d_m^j \psi_m^j(p), \tag{16}$$

where J_0 is the coarsest level of approximation. Donoho [32] has shown that, for smooth enough u ,

$$\|u(p) - u_{\geq}(p)\|_{\infty} \leq c_1 \epsilon, \tag{17}$$

and the number of significant coefficients $N(\epsilon) = \mathcal{N}$ depends on ϵ ,

$$N(\epsilon) \leq c_2 \epsilon^{-n/d}, \tag{18}$$

where d is the order of interpolation, n is the dimension of the problem and the coefficients c_i depend on $u(p)$. Combining relations (17) and (18) gives the following bound on the error in terms of $N(\epsilon)$

$$\|u(p) - u_{\geq}(p)\|_{\infty} \leq c_3 N(\epsilon)^{-d/n}. \tag{19}$$

Note that d controls the number of zero moments of the interpolating scaling function. Relation (19) shows that the wavelet basis can represent a function with significantly fewer degrees of freedom, while still retaining an accuracy $O(\epsilon)$. However, in order to realize the benefits of wavelet compression, we need to be able to reconstruct $u_{\geq}(p)$ from the subset $N(\epsilon) \subset \mathcal{N}$ of significant grid points. Note that every wavelet $\psi_l^j(p)$ is uniquely associated to a collocation (or grid) point. Consequently, a collocation point should be omitted from the computational grid if the associated wavelet is omitted from the approximation. This procedure results in a set of nested adaptive computational grids $S_{\geq}^j \subset S^j$, such that $S_{\geq}^j \subset S_{\geq}^{j+1}$ for any $j < J - 1$, where J is the finest level of resolution in approximation $u_{\geq}(p)$.

When solving the evolution equations an additional criterion for grid adaption should be added. The computational grid should consist of significant grid points and those grid points that could become significant

during a time step. In other words, at any instant of time the computational grid should consist of the $N(\epsilon)$ significant grid points plus those grid points in an adjacent zone in both position and scale that could become significant in one time step. We say that the wavelet $\psi_l^j(p)$ belongs to the adjacent zone of wavelet $\psi_k^j(p)$ if the following relations are satisfied,

$$|j - j'| \leq L, \quad |2^{j'-j}k - l| \leq M, \tag{20}$$

where L determines the extent to which coarser and finer scales are included in the adjacent zone. This allows for the development of details on finer scales such as shocks or eddies. For quadratic nonlinearities $L = 1$ (since the scale resolution is dyadic) [8]. The parameter M defines the width of adjacent zone at the same level. This allows for the transport of information along characteristics. A CFL criterion of one corresponds to $M = 1$.

The grid adaptation algorithm is described in Algorithm 1. The application of this algorithm is illustrated in the following examples.

Algorithm 1: Grid adaptation

Choose Parameters:

- A coarsest level J_0 .
- A threshold parameter $\epsilon > 0$.
- Positive adjacent zone constants M and L .

Iterative grid adaptation:

$m = 0$

$S_{\geq, m} = S^{J_0}$

do while $m = 0$ or $S_{\geq, m} \neq S_{\geq, m-1}$ or $\|u - u_{\geq, m}\|_{\infty} > \epsilon$

Sample function $u(p)$ on $S_{\geq, m}$ to give $u_{\geq, m}$.

$m = m + 1$

Forward wavelet transform

Compression: retain only significant coefficients $|d_k^j| \geq \epsilon$ to initialize $S_{\geq, m}$.

Reconstruction check: (described in [12])

add grid points needed to calculate significant coefficients.

add all points at coarsest level: $S^{J_0} \subset S_{\geq, m}$.

Adaptation: add grid points associated with adjacent zone.

Inverse wavelet transform: interpolates $u_{\geq, m}$ onto new grid $S_{\geq, m}$

end

Converged results: $S_{\geq} = S_{\geq, m}, u_{\geq} = u_{\geq, m}$.

3.1.1. Example 1: grid adaptation to a localized test function

In this example we choose a localized test function that should be well compressed in a wavelet basis. The function

$$u(\theta, \phi) = 2 \exp \left[-\frac{(\theta - \theta_0)^2 + (\phi - \phi_0)^2}{\nu} \right] - \tanh \left[\frac{\cos(\theta) \sin(\phi) + \sin(\pi/3)}{2\tau} \right] \tag{21}$$

is a superposition of a Gaussian peak and a circular shock, where $-\pi \leq \theta \leq \pi$ and $-\pi/2 \leq \phi \leq \pi/2$ are longitude and latitude, respectively. This function is displayed in Fig. 3.

We have verified relation (18) for linear wavelets (i.e. $d = 2$) in Fig. 4. Relation (19), which was proved in [33], is verified in Fig. 5a. Fig. 5b shows that the error $\|u(p) - u_{\geq}(p)\|_{\infty} < C\epsilon$, and can therefore be controlled by setting the tolerance ϵ to the desired error. Fig. 5 shows that the Butterfly wavelets perform better than linear bases, confirming our assertion that smoother bases give more compression and lifting schemes give smaller errors.

The adaptive computational grid S_{\geq} is presented in Fig. 6 for different J for the test function shown in Fig. 3. The grid is fine only in regions where the function has a strong gradient.

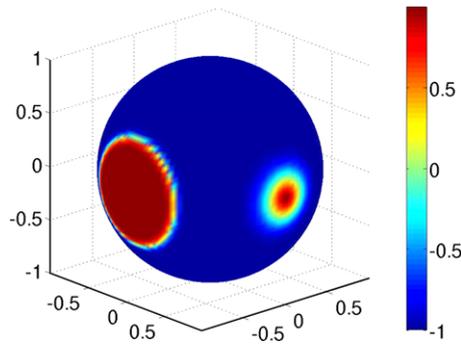


Fig. 3. Test function (21) with $\theta_0 = 0$, $\phi_0 = 0$, $v = 1/(2\pi^2)$, $\tau = 10^{-2}$.

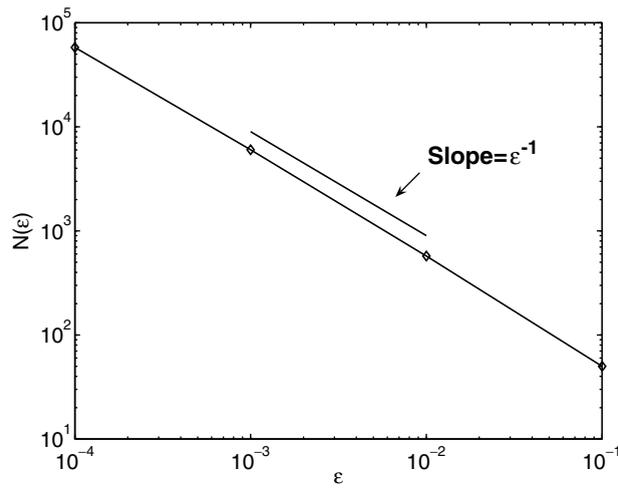


Fig. 4. Relation between $N(\epsilon)$ and ϵ for linear wavelets.

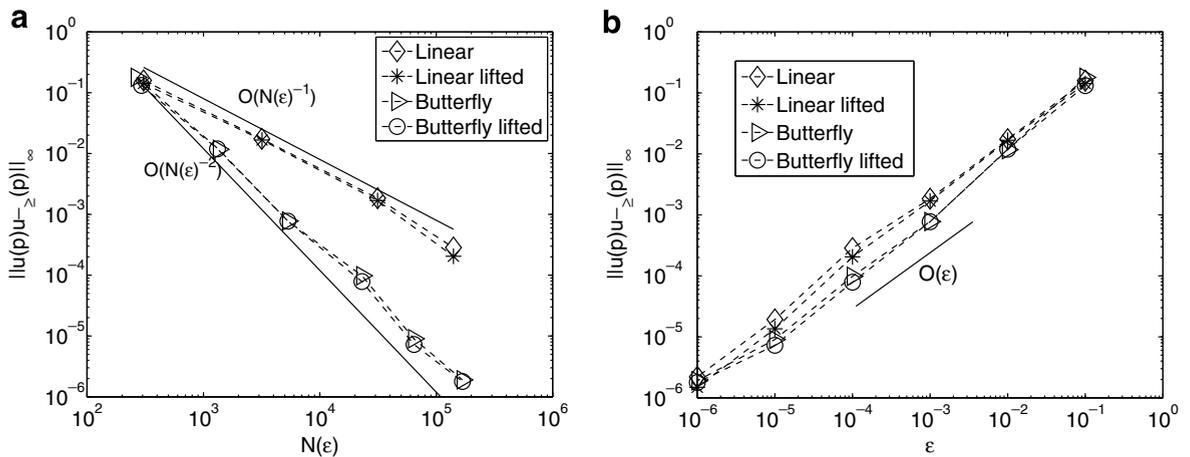


Fig. 5. Convergence result of the $u_{\geq}(p)$ for the test function as shown in Fig. 3 for different choices of bases: (a) error as a function of the number of grid points $N(\epsilon)$; (b) error as a function of the tolerance ϵ .

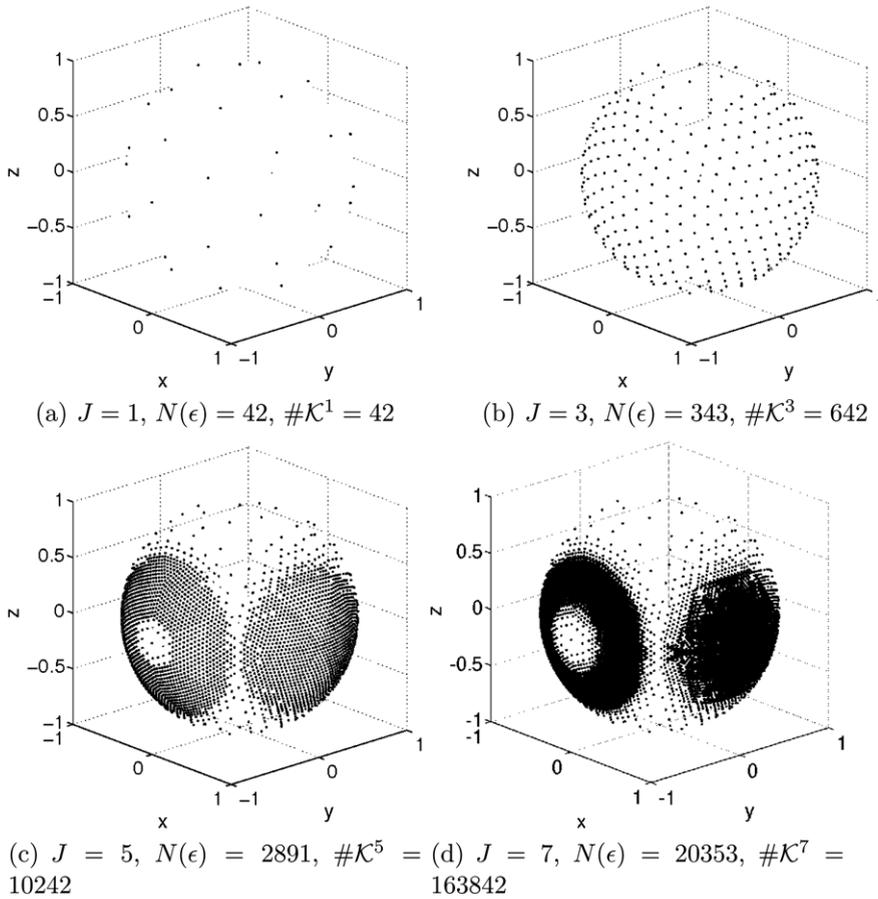


Fig. 6. Adapted grid S_{\geq} for the test function as shown in Fig. 3 at $\epsilon = 10^{-5}$ and $M, L = 1$ for $J = 1, 3, 5, 7$.

3.1.2. Example 2: grid adaptation to 2D turbulence

In this example we study the compression of turbulence data by spherical wavelets. This is necessary for the efficient storage or transmission of the enormous amounts of meteorological and climatological data resulting from numerical simulation or observation. We use a random phase multiscale model for the turbulence data, where the energy spectrum is given in terms of total spherical harmonic wavenumber n by

$$E(n) = \frac{An^{\gamma/2}}{(n + n_0)^{\gamma}}, \tag{22}$$

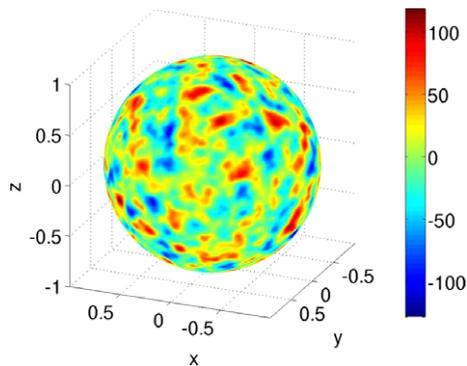


Fig. 7. Turbulence data corresponding to $\gamma = 20$ and $n_0 = 7$.

and the complex phase of each mode is chosen randomly. The value of \mathcal{A} determines the r.m.s. velocity U , while the parameters n_0 and γ , are used to control, respectively, the peak location and the width of the spectrum. The turbulence data corresponding to $\gamma = 20$ and $n_0 = 7$ in (22) is shown in Fig. 7. In Fig. 8 we compare the energy spectra obtained by retaining only the 40% and 1% most significant coefficients. At 40% we retain 95.14% of the total energy, and accurately reproduce the energy spectrum at all scales. Note that even the 1% case (i.e. a compression of 100 times) reproduces qualitatively the energy spectrum at all scales.

The relation between ϵ and $N(\epsilon)$ is shown in Fig. 9. Note that even when the wavelet threshold parameter $\epsilon \rightarrow 0$, the number of discarded incoherent wavelet modes may be still large due to the good compression properties of wavelets for turbulence. In Fig. 9 the total number of modes is $O(10^7)$, but with only minimal thresholding ($\epsilon = 10^{-7}$) we achieve a compression rate of 10 times. This reflects the fact that many wavelet coefficients are essentially zero and can therefore be discarded without losing a significant amount of energy.

Percentage relative error as a function of percentage of coefficients used is shown in Fig. 10. When we increase ϵ , the compression rates get distinctly better, as shown in Fig. 10. For example, only 67% of coefficients are required to achieve an error of order 0.01%.

3.2. Calculation of differential operators on an adaptive grid

When solving PDEs numerically, it is necessary to approximate differential operators of a function from the value of the function at collocation points. A procedure for approximating differential operators, which takes advantage of multiresolution wavelet decomposition, fast wavelet transform, and finite difference differentiation is discussed in detail by Vasilyev and Bowman [12] for the one-dimensional case and by Vasilyev [13] for multiple dimensions, but restricted to rectangular domains and flat geometries.

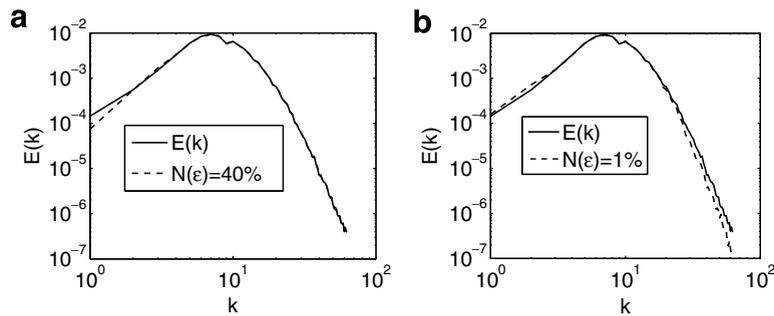


Fig. 8. Energy spectrum of turbulence data in Fig. 7.

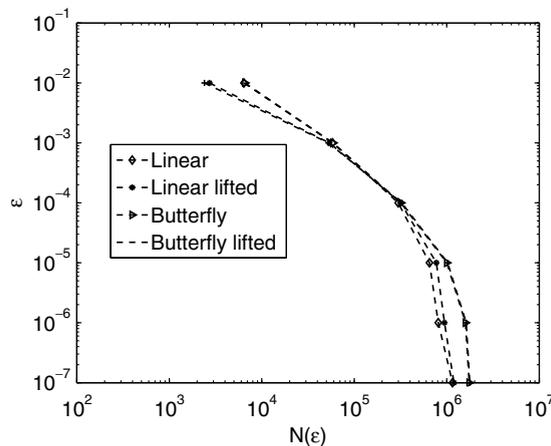


Fig. 9. Relation between ϵ and $N(\epsilon)$ using different bases for the turbulence data shown in Fig. 7.

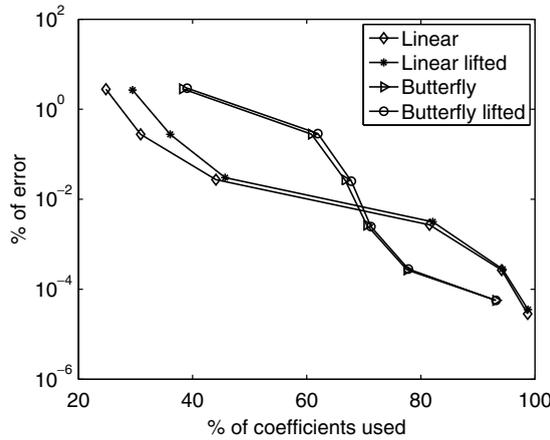


Fig. 10. Relative error as a function of the number of coefficients used for different choices of basis functions for turbulence data.

We will consider the spherical advection and spherical diffusion equations in the following section. Therefore, we outline the procedure for finding the flux term present in spherical advection equation and for approximating the Laplace–Beltrami operator. It is well known that any vector field may be separated into curl-free and divergence-free parts using Helmholtz’s decomposition. On a two-dimensional surface these two components can be expressed in terms of scalar potential fields,

$$\mathbf{V} = \mathbf{k} \times \nabla\psi + \nabla\chi, \tag{23}$$

where \mathbf{k} is the unit vector normal to the surface, ψ is the stream function representing the divergence-free part, and χ is the velocity potential representing the curl-free part. Then the flux term in the spherical advection equation can be expressed in the form of flux-divergence and Jacobian operators,

$$\nabla \cdot (\mathbf{V}u) = \nabla \cdot (u\nabla\chi) - J(u, \psi), \tag{24}$$

where the Jacobian operator is defined by $J(\alpha, \beta) = \mathbf{k} \cdot (\nabla\alpha \times \nabla\beta)$ for arbitrary scalars α and β .

Now we describe the approximation of the Laplace–Beltrami, Jacobian and flux-divergence operators on triangulated spheres, which takes advantage of the multiresolution properties of the wavelet multiresolution analysis. For any point p on the surface of S , it is known that [34]

$$\Delta_S p = 2H(p) \in \mathbb{R}^3, \tag{25}$$

where $H(p)$ is the mean curvature normal at p , i.e. $\|H(p)\|$ is the mean curvature, $H(p)/\|H(p)\|$ is the unit surface normal. Let p_i^j be a vertex of the triangulation at resolution j ; and $p_k^j, k \in N(i)$ be the neighboring vertices around p_i^j . The numerical approximation of the Laplace–Beltrami operator on the sphere Δ_S as proposed in [35] is then

$$\Delta_S u(p_i^j) = \frac{1}{A_S(p_i^j)} \sum_{k \in N(i)} \frac{\cot \alpha_{i,k} + \cot \beta_{i,k}}{2} [u(p_k^j) - u(p_i^j)], \tag{26}$$

where $\alpha_{i,k}$ and $\beta_{i,k}$ are the angles shown in Fig. 11, $N(i)$ is the set of nearest neighbor vertices of the vertex p_i^j . $A_S(p_i^j)$ is the area of the one-ring neighborhood given by

$$A_S(p_i^j) = \frac{1}{8} \sum_{k \in N(i)} (\cot \alpha_{i,k} + \cot \beta_{i,k}) \|p_k^j - p_i^j\|^2. \tag{27}$$

Using Gauss’s theorem we can approximate the Jacobian and the flux-divergence similarly,

$$J_S(u(p_i^j), \psi(p_i^j)) = \frac{1}{6A_S(p_i^j)} \sum_{k \in N(i)} (u(p_i^j) + u(p_k^j)) (\psi(p_{k+1}^j) - \psi(p_{k-1}^j)), \tag{28}$$

$$\nabla_S \cdot (u(p_i^j), \nabla_S \chi(p_i^j)) = \frac{1}{2A_S(p_i^j)} \sum_{k \in N(i)} \frac{\cot \alpha_{i,k} + \cot \beta_{i,k}}{2} (u(p_i^j) + u(p_k^j)) (\chi(p_k^j) - \chi(p_i^j)). \tag{29}$$

The differentiation procedure is based on the interpolating properties of second generation wavelets. We recall that the wavelet coefficients measure the difference between the approximations of a function at successive

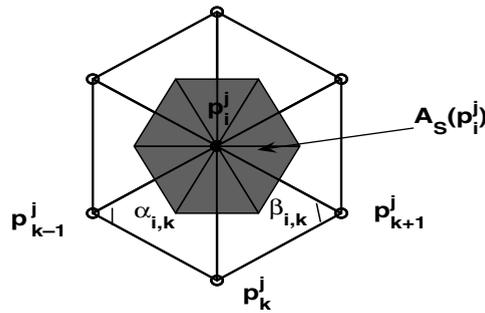


Fig. 11. The definition of the angles $\alpha_{i,j}, \beta_{i,j}$, neighboring vertices and area A_S used in evaluating derivatives on a spherical triangulation of a surface. Note that these parameters encode information about the local curvature of the surface.

levels of resolution j and $j + 1$. Thus, if there are no points in the immediate vicinity of a grid point p_i^j , i.e. $|d_k^j| < \epsilon$ for all $k \in N(i)$ and the points $p_k^{j+1}, k \in N(i)$, are not present in S_{\geq}^{j+1} , then there exists some neighborhood of p_i^j, Ω_i^j , where the function can be interpolated by a wavelet interpolant based on $s_{k,m}^j (k \in \mathcal{K}_m)$,

$$\left| u(p) - \sum_{k \in \mathcal{K}_m} s_{k,m}^j \phi_k^j(p) \right| \leq c_3 \epsilon, \tag{30}$$

where the coefficients $s_{k,m}^j$ can be chosen according to stencil in Fig. 2. The procedure for finding derivatives at all grid points is given in the following algorithm. The wavelets are used to do what they do well: compress and interpolate. Finite differences do the rest: differentiate polynomials. At the end of this procedure we have derivatives at all grid points with a uniform bound on the approximation error. The computational cost of calculating derivatives is same as the cost of forward and inverse transforms.

Next we examine the accuracy of our calculation of the Laplace–Beltrami operator on an adaptive grid. Numerical differentiation will reduce the order of scheme by one. Therefore, in light of relation (19), we have the following error bound on the derivative

Algorithm 2: Approximation of differential operators on an adaptive grid

For a function $u(p)$ defined on S_{\geq} ,

- Forward wavelet transform.
- Recursively reconstruct the function starting from the coarsest level of resolution. On each level j find derivatives of the function at grid points that belong to \mathcal{M}_{\geq}^j using the appropriate weighted average (26,28,29).

$$\|\Delta_S u(p) - \Delta_S u_{\geq}(p)\|_{\infty} \leq c_4 \epsilon \leq c_5 N(\epsilon)^{-(d-2)/n}. \tag{31}$$

The relation (31) is verified numerically for the test function presented in Fig. 3 and convergence results are presented in Fig. 12.

Furthermore, for any arbitrary scalars α and β the analytic operators satisfy

$$\int \int_S \Delta_S \alpha \, dB = 0, \quad \int \int_S J_S(\alpha, \beta) \, dw = 0, \quad \int \int_S \nabla_S(\alpha, \nabla_S \beta) = 0, \tag{32}$$

where the integrations are carried over the entire sphere. Therefore, the numerical approximation of these operators on an adaptive grid should satisfy

$$\sum_{\substack{J_0 \leq j \\ i \in \mathcal{K}^j}} A_S \Delta_S u(p_i^j) = O(\epsilon), \quad \sum_{\substack{J_0 \leq j \\ i \in \mathcal{K}^j}} A_S J_S(u(p_i^j), \psi(p_i^j)) = O(\epsilon), \quad \sum_{i \in \mathcal{K}^j}^{J_0 \leq j} A_S \nabla_S(u(p_i^j), \nabla_S \chi(p_i^j)) = O(\epsilon). \tag{33}$$

Relations (33) have been verified numerically. Eq. (33) also shows that the AWCM conserves kinetic energy and entropy under advection in purely rotational flows.

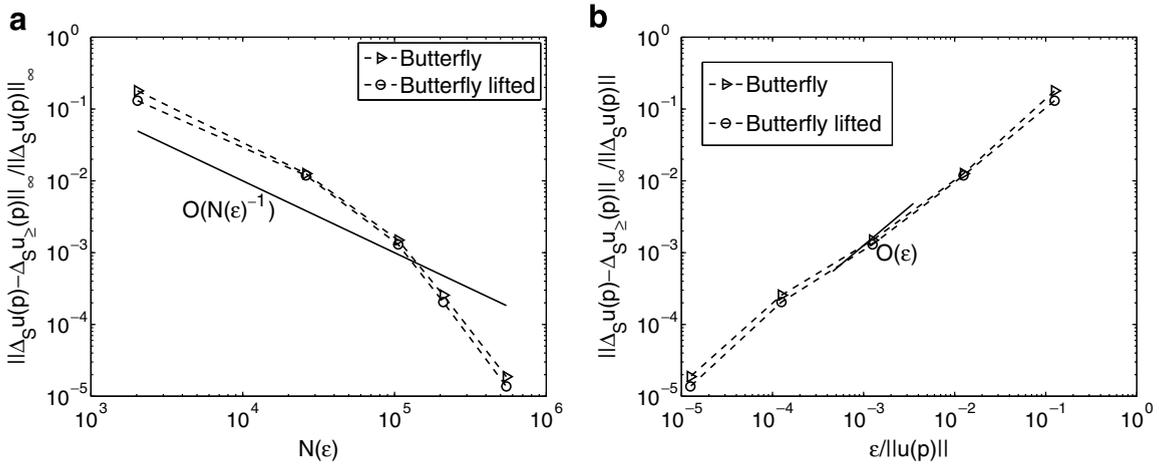


Fig. 12. Convergence result for the approximate Laplace–Beltrami operator on an adaptive grid for the test function in Fig. 3: (a) error as a function of the number of grid points $N(\epsilon)$; (b) error as a function of the tolerance ϵ .

3.3. Numerical algorithm for PDEs

We now have all the ingredients necessary to construct a dynamically adaptive algorithm for the solution of PDEs on the sphere. The three basic steps of this algorithm are as follows:

- (1) Knowing the solution $\mathbf{u}_{\geq}(t)$ on the adaptive grid, we compute the values of wavelet coefficients corresponding to each component of the solution using the fast wavelet transform. For a given threshold ϵ we update $S_{\geq}^{t+\Delta t}$ based on the magnitude of wavelet coefficients. We also add an adjacent zone to the significant coefficients to allow for the change in the solution during one time step, as described in Section 3.1.
- (2) If there is no change between computational grids S_{\geq}^t and $S_{\geq}^{t+\Delta t}$, we go directly to next step. Otherwise we interpolate the values of the solution at the collocation points $S_{\geq}^{t+\Delta t}$, which are not included in S_{\geq}^t .
- (3) We integrate the resulting system of ordinary differential equations in time (e.g. using Runge–Kutta) to obtain new values $\mathbf{u}_{\geq}(t + \Delta t)$ at positions on adaptive grid $S_{\geq}^{t+\Delta t}$, and go back to step 1.

The bold symbols $\mathbf{u} = (u_1, \dots, u_n)$ and $\mathbf{k} = (k_1, \dots, k_n)$ denote n -dimensional vectors. The adjacent zone allows the grid to automatically refine or coarsen as necessary to resolve the solution at each time step. The adaptive wavelet method provides an optimal \mathcal{N} -term approximation to the solution at each time step, given the desired accuracy ϵ .

4. Adaptive solution of PDEs on the sphere: example problems

In order to illustrate the accuracy and efficiency of the proposed numerical method, we apply it to the spherical diffusion equation, the spherical advection equation (which is a standard geophysical test known as the advection of a cosine bell) and a nonlinear reaction–diffusion equation. For simplicity, we consider the radius of sphere to be $a = 1$.

4.1. Problem formulations

4.1.1. Spherical diffusion equation

For the first test problem we consider diffusion equation

$$u_t = \nu \Delta_S u + f, \tag{34}$$

where f is a localized source chosen such a way that the solution of diffusion equation is given by

$$u(\theta, \phi, t) = 2 \exp \left[-\frac{(\theta - \theta_0)^2 + (\phi - \phi_0)^2}{v(t+1)} \right]. \quad (35)$$

The initial condition is obtained from the analytical solution, and the problem is solved for parameters values $v = 1/(4\pi^2)$, $\theta_0 = 0$, $\phi_0 = 0$.

4.1.2. Advection of the cosine bell

As a second test problem we consider a case which is very similar to the solid body rotation test case suggested by Williamson [36]. This test case uses the spherical advection equation,

$$\frac{\partial u}{\partial t} + \mathbf{V} \cdot \nabla_S u = 0. \quad (36)$$

The stationary divergence-free advecting velocity field $\mathbf{V} = (v_1, v_2)$ is given by

$$\begin{aligned} v_1 &= u_0 \left(\cos(\phi) \cos(\alpha) + \sin(\phi) \cos \left(\theta + \frac{3\pi}{2} \right) \sin(\alpha) \right), \\ v_2 &= -u_0 \sin \left(\theta + \frac{3\pi}{2} \right) \sin(\alpha). \end{aligned} \quad (37)$$

where u_0 the advection speed and α is the angle between the axis of solid body rotation and the polar axis of spherical coordinate system. We set $u_0 = 2\pi a$ so that the rotation period is equal to one. The initial condition is

$$u(\theta, \phi) = \begin{cases} \frac{1}{2} (1 + \cos(\frac{\pi r}{R})) & \text{if } r < R, \\ 0 & \text{if } r \geq R, \end{cases} \quad (38)$$

where radius $R = a/3$ and $r = a \arccos[\sin(\phi_c) \sin(\phi) + \cos(\phi_c) \cos(\phi) \cos(\theta - \theta_c)]$, which is the geodesic distance between (ϕ, θ) and the centre $(\phi_c, \theta_c) = (0, 0)$. This pattern is simply advected without changing shape.

4.1.3. Nonlinear reaction–diffusion equation

Finally, we consider a nonlinear PDE. Solving a nonlinear PDE tests the ability of our adaptive method to track the development of localized structure from an unstructured initial condition. We solve a nonlinear reaction–diffusion equation on the sphere [4],

$$\frac{\partial u_1}{\partial t} = \alpha u_1 (1 - r_1 u_1^2) + u_2 (1 - r_2 u_1) + D \delta \Delta_S u_1, \quad (39)$$

$$\frac{\partial u_2}{\partial t} = \beta u_2 \left(1 + \frac{\alpha r_1}{\beta} u_1 u_2 \right) + u_1 (\gamma + r_2 u_2) + \delta \Delta_S u_2. \quad (40)$$

The parameters are chosen as in the first case considered in [4]: $r_1 = 3.5$, $r_2 = 0$, $D = 0.516$, $\delta = 0.0045$, $\alpha = 0.899$, $\beta = -0.91$ and $\gamma = -\alpha$. This choice of parameters eventually produces a system of bands with defects (spots) at both poles. The fact that the solution of this nonlinear equation on the sphere is one of the few known qualitatively makes it a good test case.

The initial conditions are $u_1 = u_2 = 0$, except on a narrow band near the equator, where u_1 and u_2 take random values between 0 and 0.5. Note that the evolution to the steady state is extremely slow: Varea et al. [4] required over 1.5 million time steps to reach the stationary banded solution at $t = 1500$. In order to efficiently calculate such slow dynamics we use the stiffly stable Krylov time integration technique [37] for this calculation. Unlike the Runge–Kutta method used in the other cases, the Krylov method allows for very large time steps during the slow evolution characteristic of this problem. We continue the calculation until the banded structure is well established.

4.2. Numerical results

4.2.1. Spherical diffusion equation

This example tests the AWCM’s ability to accurately and efficiently solve a diffusive equation on an adapted grid. In this case, the refined part of the grid should spread and coarsen as time increases. The initial grid is determined from the initial condition using Algorithm 1. The solution of the diffusion equation and the dynamically adapted computational grids S_{\geq}^t are shown in Fig. 13 for $\epsilon = 10^{-4}$. In order to demonstrate the

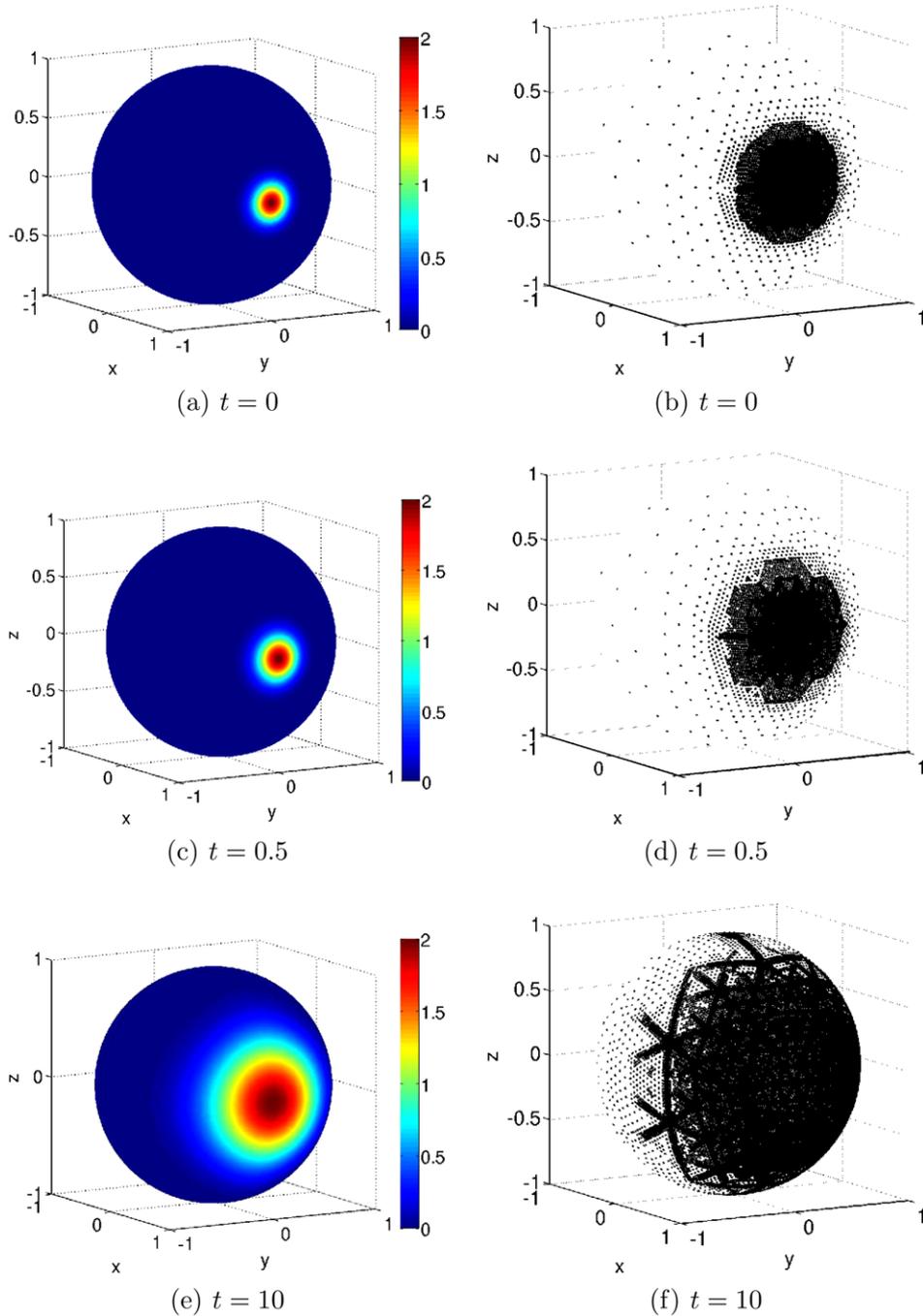


Fig. 13. Evolution of the solution and dynamically adapted grid S_{\geq}^t for the spherical diffusion equation with tolerance $\epsilon = 10^{-4}$.

efficiency of the adaptive algorithm we need to compare the number of grid points used in the adaptive and nonadaptive methods. This can be measured by compression coefficient $C = N(\epsilon = 0)/N(\epsilon)$. The larger the compression rate, the more efficient the adaptive algorithm. A compression coefficient of one indicates that there is no compression (i.e. the grid is nonadaptive).

The time evolution of the compression coefficient is shown in Fig. 14. Initially at $t = 0$ approximately 16 times fewer grid points are required for the computation (compared to a non-adaptive calculation), and the compression coefficient decreases as the initial Gaussian peak diffuses.

Next, we study the convergence of the AWCM. Eq. (18) predicts that decreasing ϵ reduces the error (by increasing the number of grid points $N(\epsilon)$). Hence, the approximation of the solution is controlled by ϵ and d . In Fig. 15, we present the results of a convergence study for the Butterfly wavelet using the diffusion equation. This shows that we can control the error at a particular fixed time (using a given time integration scheme) by adjusting ϵ . These results confirm the analytical error estimate (17).

4.2.2. Advection of cosine bell

This case consists of advecting the cosine bell with a stationary velocity field (37). We run the test for $\alpha = \frac{\pi}{2} - 0.05$ (this is the most unfavorable case for latitude longitude grids). To get a better idea of how the adaptive wavelet algorithm distributes points in position and scale we have plotted the initial grid at each scale in Fig. 16a–e and whole adaptive grid at the finest scale in Fig. 16f. Choosing smaller ϵ leads to denser

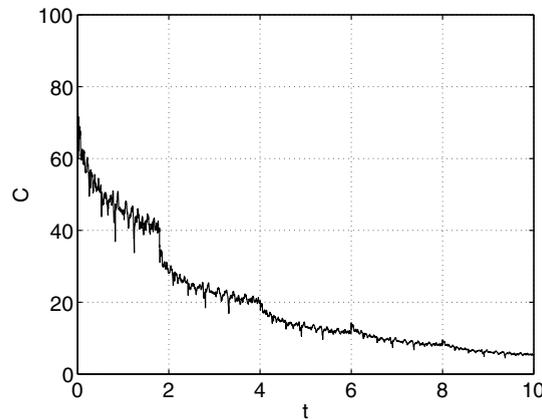


Fig. 14. The compression coefficient C as a function of time for the spherical diffusion equation with tolerance $\epsilon = 10^{-4}$.

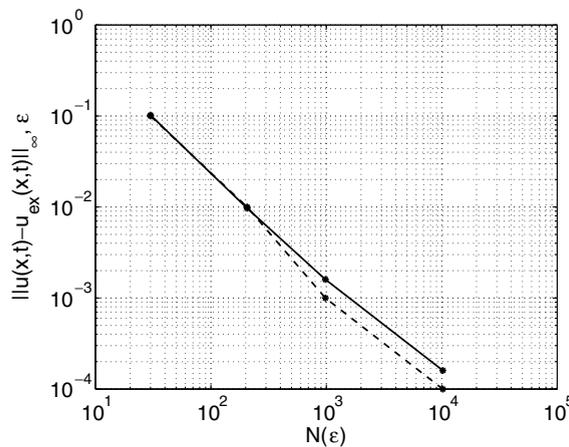


Fig. 15. The L_∞ error of the solution (solid line) at time $t = 0.5$, $\epsilon = 10^{-4}$ for the spherical diffusion equation. The dashed line shows the value of ϵ as a function of $N(\epsilon)$.

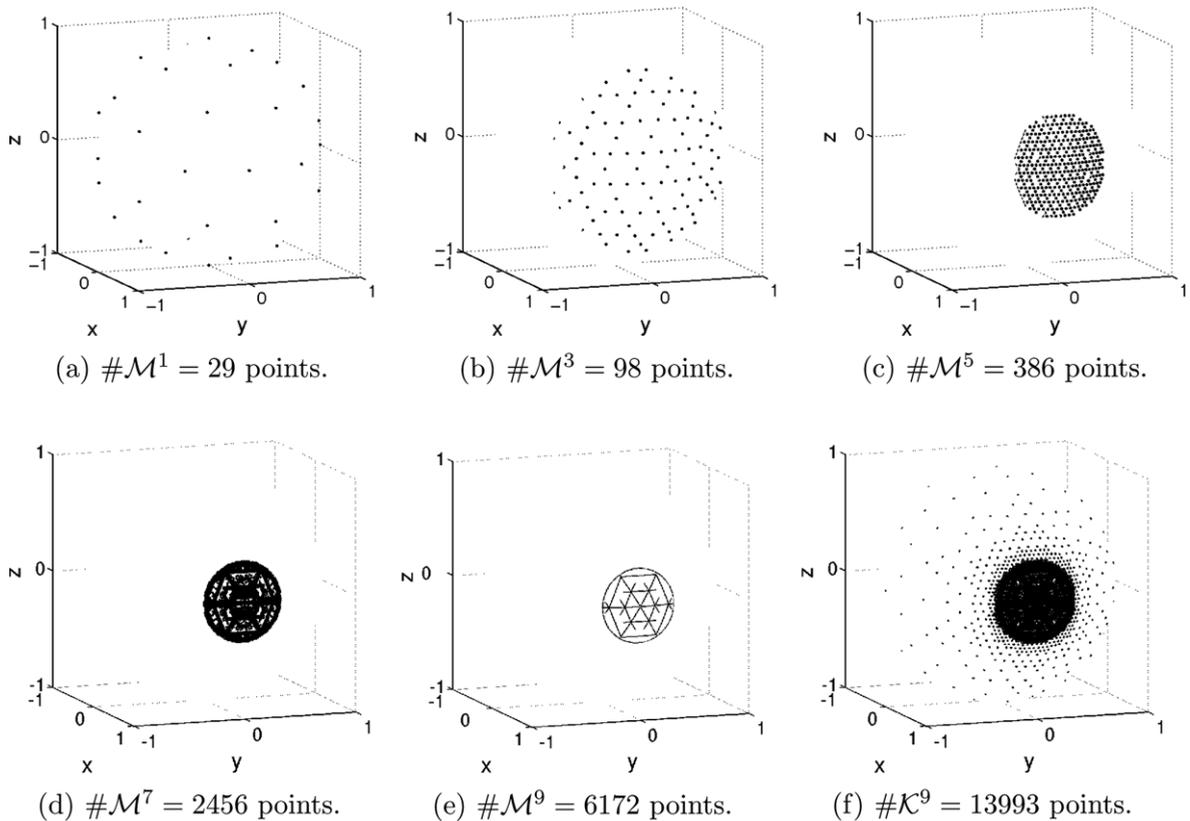


Fig. 16. The initial adaptive grid for the advection of the cosine bell, $S_{\mathbb{V}}^{\sigma=0}$ for tolerance $\epsilon = 10^{-5}$. (a–e) Grids at scales 1, 3, 5, 7, 9, respectively. (f) The complete grid, which is the union of the grids at scales 0–9.

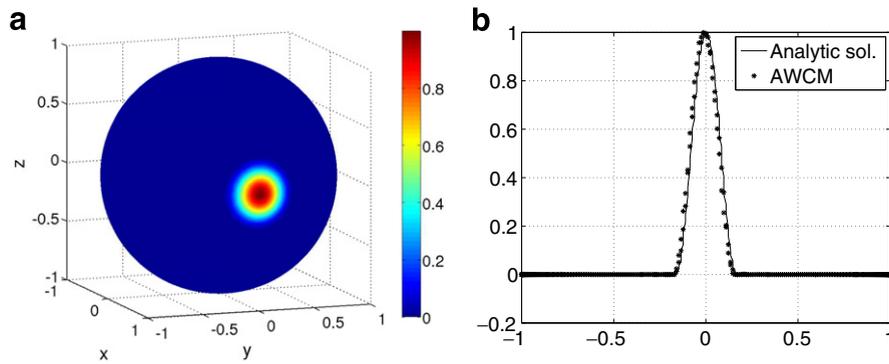


Fig. 17. Solid body rotation of cosine bell using AWCM after one rotation ($t = 1$) with tolerance $\epsilon = 10^{-5}$: (a) solution; (b) cross-section through peak compared to exact solution.

computational grids, and hence require more computation. Typically, $\epsilon = O(10^{-3})$ gives acceptable accuracy for time-dependent PDEs. The numerical solution obtained with the AWCM at $t = 1$ is presented in Fig. 17a and compared with the analytical solution by taking the slice in one direction is presented in Fig. 17b. The dynamically adapted grid for the solution of Fig. 17a is shown in Fig. 18. We see that the grid refinement somewhat trails the rotating cosine. This systematic difference is probably due to the dispersive error of the finite difference method. However, this dispersive error does not grow with time and therefore does not significantly perturb the calculation. The compression coefficient is shown in Fig. 19. It oscillates, but its average

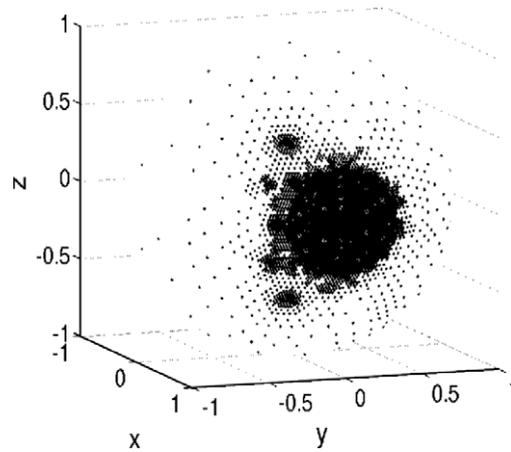


Fig. 18. Adapted grid for the solution presented in Fig. 17a.

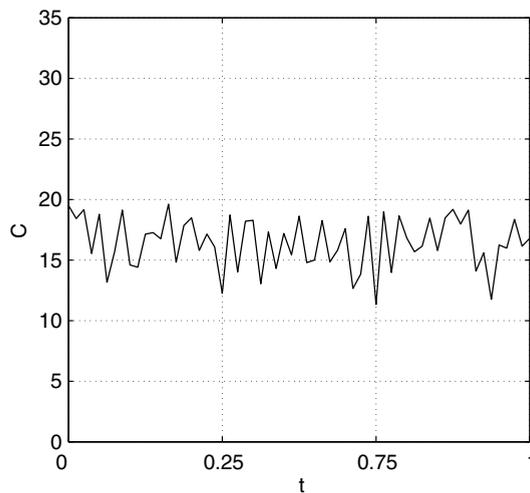


Fig. 19. The compression coefficient C for the advection of the cosine bell as a function of time with tolerance $\epsilon = 10^{-5}$.

is roughly constant (about 17 times compression). The slight decrease in the compression coefficient over time is due to the fact that spurious dispersive oscillations behind the moving bell errors are resolved.

We have also tested the AWCM for $\alpha = 0$ and observed no significant dependence of the grid structure on α . This confirms the quasi-uniformity of the reduced grid and the stability of our scheme. The dynamically adapted grid at $t = 1$ is shown in Fig. 18 and the time evolution of the L_∞ error is shown in Fig. 20. The grid shows some spurious upwind refinement, but is still quite close to the initially adapted grid in Fig. 16f. The error increases monotonically, as expected in any time marching method.

4.2.3. Nonlinear reaction–diffusion equation

We now present the solution of the nonlinear reaction–diffusion equation. The evolution of the solution and corresponding adaptive grid using the AWCM are shown in Figs. 21 and 22, respectively. The compression coefficient of the initial condition is 6.4. It then decreases until $t \approx 34$ before increasing again to reach a roughly constant value of 10 (i.e. 10 times compression). It is interesting to note that although \mathbf{u} is initially confined to a single narrow band, the AWCM is able to track the emergence of banded structures over the entire sphere. This demonstrates that our adaptive method is able to efficiently and accurately capture the emergence of the localized structures which characterize the solution of nonlinear PDEs.

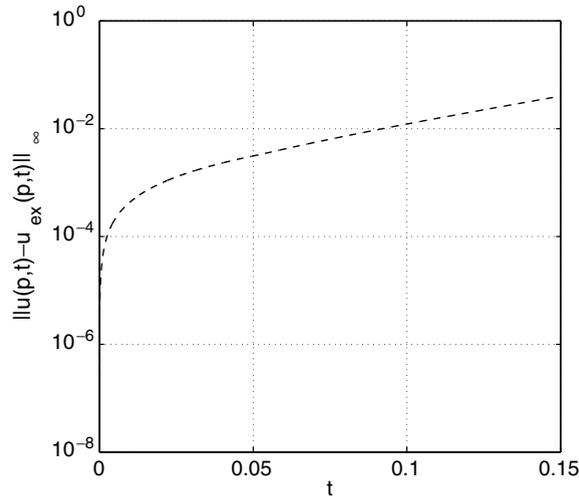


Fig. 20. Time evolution of the pointwise L_∞ error of the advected cosine bell solution.

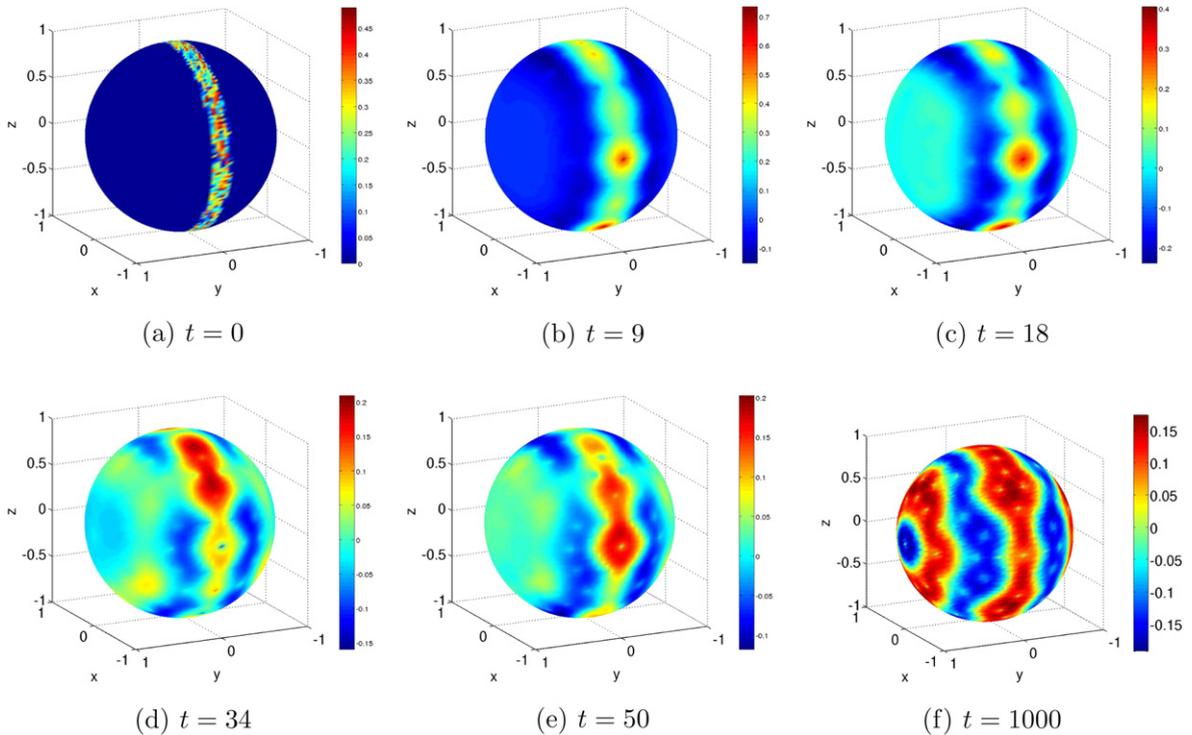


Fig. 21. The solution of the nonlinear reaction–diffusion equation for tolerance $\epsilon = 10^{-2}$. Only the first component u_1 is shown.

5. Summary and future work

In this work a complete adaptive wavelet collocation method (AWCM) for the numerical solution of PDEs on the sphere has been developed. Wavelet decomposition is used for grid adaption and interpolation, while an $O(\mathcal{N})$ hierarchical finite difference scheme over the triangulated surface is used for the Laplace–Beltrami, Jacobian and flux-divergence differential operators. This finite difference scheme takes advantage of wavelet multilevel decomposition and the adaptive grid. The method is verified by solving the spherical diffusion

equation and spherical advection equation (which is a simplified form of the shallow water equation on the sphere). The results show that the computational grid adapts efficiently to the local gradients of the solution, refining or coarsening as necessary to satisfy the specified error tolerance. This error control is done automatically by specifying one parameter at the beginning of the simulation: the wavelet filtering parameter ϵ . Furthermore, the solution is obtained on a near optimal grid: the AWCN uses the minimal number of grid points necessary to obtain the specified error $O(\epsilon)$. The localized advection and diffusion problems considered here arise, for example, in the tracking of storm fronts, hurricanes and breaking waves in the the simulation of global atmospheric dynamics.

We have also solved a nonlinear reaction–diffusion equations, starting from an unstructured initial condition. We show that our adaptive method is able to correctly predict the emergence of banded structures over the whole sphere starting from a random initial condition localized near the equator. The ability of an adaptive method to track emergent structure is essential for solving nonlinear problems.

As a first step towards applying our method to two-dimensional turbulence on the sphere we studied the compression of model atmospheric turbulence data by spherical wavelets. We found that approximately half of the modes are sufficient to achieve 10^{-2} accuracy in the reconstructed velocity field, and that 1% of the modes are sufficient to qualitatively capture the energy spectrum at all scales.

The simple examples considered here suggest that the AWCN may prove useful in the many large-scale geophysical simulations of atmosphere and ocean dynamics. Specific examples include the simulation of transport of pollutants in the Earth’s atmosphere as well as the forecasting of moisture and cloud water fields in numerical weather prediction and climate models.

The multilevel adaptive approach used here is in some respects similar to [22], where a wavelet MRA is used to provide adaptive re-gridding for a Lagrangian particle based level set method. It would be interesting to

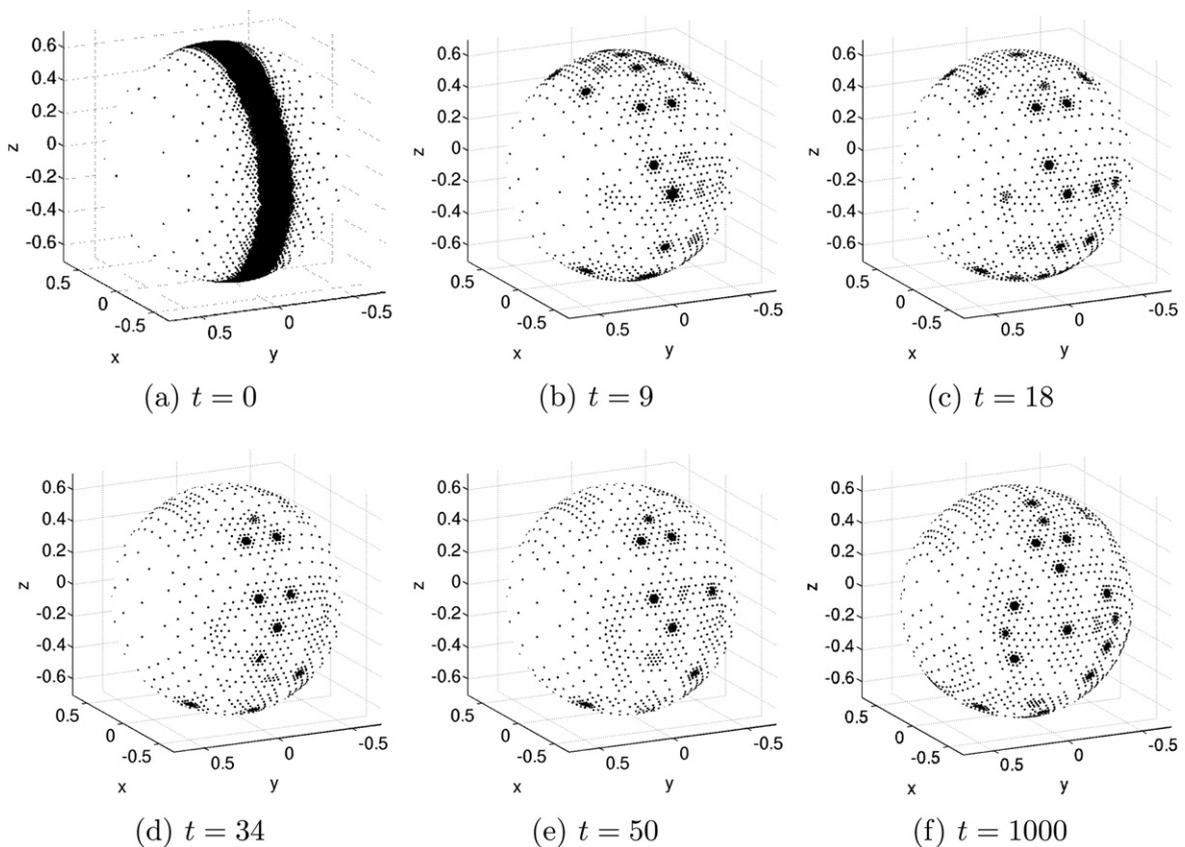


Fig. 22. The dynamically adapted grid for the solution of the nonlinear reaction–diffusion equation. The grids correspond to the solutions shown in Fig. 21.

combine the level set idea with the well-established adaptive wavelet collocation method on a three-dimensional Cartesian grid [14].

It should be noted that the overall accuracy of the algorithm depends on the accuracy of the derivatives. The current approximations are only first- or second-order accurate, but we are currently investigating wavelets based on optimal spherical triangulations [23]. The advantages of such triangulations are that the discrete mean curvature approximation is exact, and truncation error of derivatives is minimized.

Another future area of research is the incorporation of a multilevel elliptic solver to solve PDE constraints for evolution problems, such as the Poisson equation in the pressure correction method for the incompressible Navier–Stokes equations. This would allow us to solve the full incompressible Navier–Stokes equations on the sphere, taking advantage of wavelet multilevel decomposition and compression. In flat geometries this has been done successfully in [14]. This work is currently underway.

In this paper we developed an AWCM method on the sphere, but our approach can be extended to other curved manifolds as follows. For an arbitrary manifold M we need to construct wavelets which are parameterized over a polyhedral base mesh. The initial step of converting an arbitrary surface into a multiresolution form is therefore to determine a base mesh K_0 that is topologically equivalent to the given surface. One then finds appropriate parameterizations of M over K_0 as described in [16]. Once the parameterizations are found for a given polyhedral surface, the base mesh can be sub-divided, as in the case of the sphere, to generate a multi-resolution analysis of M .

Acknowledgments

M.M. gratefully acknowledges the partial support provided by NSERC. We thank Jahrul Alam for helpful discussions. N.K.R.K. gratefully acknowledges support from NSERC. This work was made possible by the facilities of the Shared Hierarchical Academic Research Computing Network (SHARCNET).

References

- [1] D. Stensrud, *Parameterization Schemes: Keys to Understanding Numerical Weather Prediction Models*, Cambridge University Press, 2007.
- [2] U.F. Mayer, Numerical solutions for the surface diffusion flow in three space dimensions, *Comp. Appl. Math.* 20 (3) (2001) 361–379.
- [3] G. Simonett, The Willmore flow for near spheres, *Differ. Integr. Equat.* 14 (8) (2001) 1005–1014.
- [4] C. Varea, J.L. Aragon, R.A. Barrio, Turing pattern on a sphere, *Phys. Rev. E* 60 (4) (1999) 4588–4592.
- [5] J. Pudykiewicz, Numerical solution of the reaction–advection–diffusion equation on the sphere, *J. Comput. Phys.* 213 (2006) 358–390.
- [6] A. Cohen, W. Dahmen, R. DeVore, Adaptive wavelet methods for elliptic operator equations: convergence rates, *Math. Comput.* 70 (2001) 27–75.
- [7] S. Dahlke, W. Dahmen, R. Hochmuth, R. Schneider, Stable multiscale bases and local error estimation for elliptic problems, *Appl. Numer. Math.* 23 (1997) 21–47.
- [8] J. Liandrat, P. Tchamitchian, Resolution of the 1d regularized burgers equation using a spatial wavelet approximation, *Tech. Rep.* 90-83, ICASE, 1990.
- [9] M. Holmstrom, J. Walden, Adaptive wavelet methods for hyperbolic pde's, *J. Comput. Phys.* 13 (1998) 19–49.
- [10] O. Roussel, K. Schneider, A. Tsigulin, H. Bockhorn, A conservative fully adaptive multiresolution algorithm for parabolic pdes, *J. Comput. Phys.* 188 (2003) 493–523.
- [11] S. Bertoluzza, Adaptive wavelet collocation for the solution of steady-state equations, *SPIE Proc.* 2491 (1995) 947–956.
- [12] O.V. Vasilyev, C. Bowman, Second generation wavelet collocation method for the solution of partial differential equations, *J. Commun. Phys.* 165 (2000) 660–693.
- [13] O.V. Vasilyev, Solving multi-dimensional evolution problems with localized structure using second generation wavelets, *Int. J. Comp. Fluid Dyn.* 17 (2) (2003) 151–168.
- [14] O.V. Vasilyev, N.K.-R. Kevlahan, An adaptive multilevel wavelet collocation method for elliptic problems, *J. Comput. Phys.* 206 (2005) 412–431.
- [15] N. Kevlahan, O. Vasilyev, An adaptive wavelet collocation method for fluid–structure interaction at high Reynolds numbers, *SIAM J. Sci. Comput.* 26 (6) (2005) 1894–1915.
- [16] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lonsbery, W. Stuetzle, Multiresolution analysis of arbitrary meshes, in: *Proceedings of SIGGRAPH 95*, ACM, 1995, pp. 173–182.
- [17] N. Dyn, D. Levin, J. Gregory, A butterfly subdivision scheme for surface interpolation with tension control, *Trans. Graph.* 9 (2) (1990) 160–169.
- [18] R. Heikes, D.A. Randall, Numerical integration of the shallow-water equations on a twisted icosahedral grid part: basic design and results of test, *Mon. Weather Rev.* 123 (1995) 1862–1880.

- [19] R. Agarwal, Computational fluids dynamics of whole body aircraft, *Ann. Rev. Fluid Mech.* 31 (1999) 125–169.
- [20] M. Bertalmio, L.T. Cheng, S. Osher, Variational problems and partial differential equations on implicit surfaces, *J. Comput. Phys.* 174 (2001) 759–780.
- [21] I.F. Sbalzarini, A. Hayer, A. Helenius, P. Koumoutsakos, Simulations of anisotropic diffusion on curved biological surfaces, *Biophys. J.* 90 (2006) 878–885.
- [22] M. Bergdorf, P. Koumoutsakos, A lagrangian particle-wavelet method, *Multiscale Model Simul.* 5 (2006) 980–995.
- [23] G. Xu, Discrete Laplace–Beltrami operator on sphere and optimal spherical triangulation, *Int. J. Comp. Geom. Appl.* 16 (2006) 75–93.
- [24] P. Schroder, W. Sweldens, Spherical wavelets: efficiently representing functions on the sphere, in: *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, 1995*, pp. 161–172.
- [25] S. Valette, R. Prost, Wavelet-based progressive compression scheme for triangle meshes: wavemesh, *IEEE Trans. Vis. Comput. Graph.* 10 (2) (2004) 123–129.
- [26] I.G. Novak, F. Gao, Y.S. Choi, D. Resasco, J.C. Schaff, B.M. Slepchenko, Diffusion on a curved surface coupled to diffusion in the volume: application to cell biology, *J. Comput. Phys.* 226 (2007) 1271–1290.
- [27] G.R. Stuhne, W.R. Peltier, New icosahedral grid-point discretization of the shallow water equation on the sphere, *J. Comput. Phys.* 148 (1999) 23–58.
- [28] D. Majewski, D. Liermann, P. Prohl, B. Ritter, M. Buchhold, T. Hanish, G. Paul, W. Wergen, The operational icosahedral–hexagonal gridpoint model gme: description and high-resolution tests, *Mon. Weather Rev.* 130 (2002) 319–338.
- [29] M. Farge, K. Schneider, N.K.-R. Kevlahan, Non-gaussianity and coherent vortex simulation for two-dimensional turbulence using an adaptive orthogonal wavelet basis, *Phys. Fluids* 11 (1999) 2187–2201.
- [30] J. Alam, N.-R. Kevlahan, O. Vasilyev, Scaling of space–time modes with reynolds number in two-dimensional turbulence, *J. Fluid Mech.* 570 (2007) 217–226.
- [31] W. Sweldens, The lifting scheme: a construction of second generation wavelets, *SIAM J. Math. Anal.* 29 (2) (1998) 511–546.
- [32] D.L. Donoho, Interpolating wavelet transforms, *Tech. Rep. 408*, Department of Statistics, Stanford University, 1992.
- [33] R.A. Devore, B. Jawerth, B.J. Lucier, Image compression through wavelet transform coding, *IEEE Trans. Inform. Theory* 38 (2) (1992) 719–746.
- [34] T.J. Willmore, *Riemannian Geometry*, Clarendon Press, 1993.
- [35] M. Meyer, M. Desbrun, P. Schroder, A. Barr, Discrete differential geometry operator for triangulated 2-manifolds, in: *Visualization and Mathematics III, 2002*, pp. 35–57.
- [36] D.L. Williamson, J.B. Drake, J.J. Hack, P.N.S.R. Jacob, A standard test set for numerical approximations to the shallow water equations in spherical geometry, *J. Comput. Phys.* 102 (1992) 211–224.
- [37] W.S. Edwards, L.S. Tuckerman, R.A. Friesner, D.C. Sorensen, Krylov methods for the incompressible Navier–Stokes equations, *J. Comput. Phys.* 110 (1994) 82–102.