RMetS
Royal Meteorological Society

A conservative adaptive wavelet method for the shallow-water equations on staggered grids

T. Dubos[a]* and N. K-R. Kevlahan[b]

[a]*Laboratoire de Météorologie Dynamique/IPSL, École Polytechnique, Palaiseau, France*
[b]*Department of Mathematics and Statistics, McMaster University, Hamilton, Canada*

*Correspondence to: T. Dubos, LMD/IPSL, École Polytechnique, Palaiseau 91128, France.
E-mail: dubos@lmd.polytechnique.fr

This article presents the first dynamically adaptive wavelet method for the shallow-water equations (SWEs) on a staggered hexagonal C-grid. Pressure is located at the centres of the primal grid (hexagons) and velocity is located at the edges of the dual grid (triangles). Distinct bi-orthogonal second-generation wavelet transforms are developed for the pressure and the velocity. These wavelet transforms are based on second-order accurate interpolation and restriction operators. Together with compatible restriction operators for the mass flux and Bernoulli function, they ensure that mass is conserved and that there is no numerical generation of vorticity when solving the SWEs. Grid refinement relies on appropriate thresholding of the wavelet coefficients, allowing error control in both the quasi-geostrophic and inertia–gravity wave regimes. The SWEs are discretized on the dynamically adapted multiscale grid using a mass and potential-enstrophy-conserving finite-difference scheme. The conservation and error control properties of the method are verified by applying it to a propagating inertia–gravity wave packet and to rotating shallow-water turbulence. Significant savings in the number of degrees of freedom are achieved even in the case of rotating shallow-water turbulence. The numerical dissipation introduced by the grid adaptation is quantified. The method has been designed so it can be extended easily to the icosahedral subdivision of the sphere. This work provides important building blocks for the development of fully adaptive general circulation models. Copyright © 2013 Royal Meteorological Society

Key Words: climate modelling; numerical methods; shallow-water equations; wavelets

Received 1 July 2012; Revised 20 November 2012; Accepted 5 December 2012; Published online in Wiley Online Library

Citation: T. Dubos, NK-R. Kevlahan. 2013. A conservative adaptive wavelet method for the shallow-water equations on staggered grids. *Q. J. R. Meteorol. Soc.* DOI:10.1002/qj.2097

1. Introduction

1.1. *Adaptive methods for climate and weather models*

Atmospheric and oceanic flows span a wide range of scales and at the same time are organized into wave packets, eddies, jets and currents. Therefore it seems likely that numerical modelling of the atmosphere and oceans would be more efficient and accurate if the model resolution was not uniform but refined locally where small-scale features need to be resolved. To some extent, this idea has gained in popularity in recent years with the increasing use of stretched grids (Krinner *et al.*, 1997) and nested models to achieve higher resolution locally for regional numerical weather forecasting or regional climate modelling. However the grid refinement remains static (i.e. the grid does not evolve in time) in operational practice. While this can be justified when there is *a priori* knowledge of the location of the small-scale features, such as ocean boundary currents, this is a strong limitation since many small-scale features (e.g. fronts, hurricanes or oceanic eddies) occur at unpredictable locations. To efficiently resolve such phenomena, dynamic grid adaptivity is necessary, and indeed the idea of dynamic grid adaptivity for numerical modelling of the atmosphere and ocean was explored as early as the 1980s (Skamarock *et al.*, 1989). Despite continuous progress on dynamically adaptive methods for numerical modelling of the atmosphere and ocean (Ruge *et al.*, 1995; Bacon *et al.*, 2000; Läuter *et al.*, 2007; St-Cyr *et al.*, 2008; Jablonowski *et al.*, 2009; Weller *et al.*, 2009; Nikiforakis, 2009; Chen *et al.*, 2010), it is fair to say that dynamical adaptivity has not yet made its way into operational models, with the notable exception of the Operational Multiscale Environment model with Grid Adaptivity (OMEGA; Bacon *et al.*, 2000). This state of affairs is caused by a number of issues specific to the modelling of the atmosphere and ocean. Prominent amongst these are the numerical properties of adaptive methods, the grid refinement criteria, the complexity versus efficiency trade-off, and the interplay with subgrid-scale physics. We address these issues at least partially in the present work, with the exception of subgrid-scale physics.

As discussed in more detail in the next subsection, numerical models of the atmosphere and ocean prefer numerical methods with exact discrete conservation properties since they improve the stability and fidelity of these under-resolved models. Devising non-adaptive numerical methods with such properties is not straightforward, and any adaptive method with inferior conservation properties would be of questionable interest. How to exacly conserve total mass is well understood when grid refinement is obtained by recursive subdivison of control volumes (e.g. Berger and Oliger, 1984; St-Cyr *et al.*, 2008) or by remapping (e.g. Farrell *et al.*, 2009), but we are not aware of dynamically adaptive methods with a consistent treatment of vorticity. Furthermore, no existing approach seems applicable to the case of a hierarchy of non-nested control volumes, like spherical–hexagonal grids. In this case, a fine-resolution hexagon overlaps with several coarse-resolution hexagons, and a simple flux-counting approach fails. More generally, the need for dynamical adaptivity should not restrict the set of available numerical schemes. Instead, an approach is needed that can render dynamically adaptive any appropriate numerical method. The wavelet-based approach we develop here can be tailored to a wide range of pre-existing numerical methods, including non-Cartesian staggered grids, with the restriction that discrete differential operators should possess a compact stencil.

Dynamical adaptivity needs refinement criteria to refine and coarsen the computational mesh at each time step. Coarsening is important since the failure to coarsen regions where it would have been acceptable increases the computational cost without increasing the fidelity of the simulation. Refinement ensures uniform accuracy, or more importantly allows resolution of phenomena that would otherwise remain unresolved. Often heuristic, gradient-based or vorticity-based refinement criteria are used (St-Cyr *et al.*, 2008). For numerical weather forecasting (NWP), it is unacceptable to miss small-scale high-impact phenomena due to inappropriate refinement criteria, but the additional cost of the refinement should not prevent the forecast from being calculated in a reasonable time. For climate modelling any systematic bias introduced by the refinement criteria would be problematic. Thus, the development of robust, objective, and sharp refinement criteria remains an open problem (e.g. Berger and Colella, 1989; Nikiforakis, 2009; Weller, 2009) and their suitability to the various forms of atmosphere and ocean modelling needs to be demonstrated. The refinement strategy we propose uses in a novel way the error-control capabilities inherent to wavelet decompositions. A refinement strategy based on error control is not suitable in all situations, especially when subgrid-scale physics come into play, and our strategy is not universally applicable, especially when complex physics and dynamics are coupled. Nevertheless our results suggest that our approach is effective and robust in idealized situations, where *a priori* knowledge of the dominant balances is available.

Dynamically adaptive methods can appear complicated. This complexity seems incompatible with the need of community-based research codes to evolve continuously and to be easily modified to explore new ideas. This sets the bar high for the efficiency gains that must be demonstrated in order for adaptive methods to gain acceptance and means that efficient implementations on present and future massively parallel architectures are needed. We do not address here the computational challenge, which is not specific to applications to the atmosphere and oceans. However, experience in other fields, such as the FLASH (Fryxell *et al.*, 2000) code used by the astrophysics community, suggests that it is possible to build efficient and easily modifiable dynamically adaptive codes (Popinet, 2003). However, we do provide partial answers to the question of whether a substantial gain in computational efficiency can be achieved in practice. This is clearly the case for the favourable situation of a small coherent structure isolated in a laminar flow, but we show it is also true for the more realistic situation where the computational domain is densely populated by a large number of interacting coherent structures which move and evolve unpredictably. Indeed, a common source of scepticism towards dynamical adaptivity is that it should not, for example, track only a few oceanic mesoscale eddies, but a whole field of eddies, while still delivering significant efficiency gains.

Finally, a formidable issue is the interaction between a dynamically adaptive method and subgrid-scale physics. Even with dynamically adaptive grids, numerical simulations of the atmosphere and the ocean will remain spatially unresolved for decades to come. Subgrid-scale physics will therefore remain an essential part of the models, adaptive or not. How the parametrization of subgrid-scale phenomena varies with the local resolution, how it interacts with the numerics, and how the refinement strategy should incorporate subgrid-scale phenomena, are open problems. This work does not address this issue. We note, however, that the advent of variable-resolution static grids has stimulated research on scale-aware parametrizations (Chen *et al.*, 2011).

1.2. *Adaptive numerical methods for geophysical flows*

There are four main approaches to increase accuracy and efficiency in numerical methods for solving partial differential equations (PDEs): *h*-refinement, *p*-refinement, *r*-refinement and mimetic methods. In *h*-refinement, increased accuracy is achieved by fixing the order of the method (e.g. a second-order finite-volume discretization) and refining or coarsening the grid locally to achieve a constant error tolerance. This approach is best for problems with non-smooth solutions, or when high-order discretizations are not available. In p-refinement the grid resolution is kept fixed and the order of the method is modified locally (e.g. spectral element methods). This approach is optimal for problems with smooth solutions where higher-order discretizations are available. It is possible to combine these approaches in hp-refinement, such as hp-FEM (finite-element methods), where one seeks a compromise between a high-order method on a coarse grid where the solution is smooth and a low-order method on a fine grid where the solution is not smooth. Finally, r-refinement conserves the number of grid points (or computational elements), but redistributes them to minimize the error. r-refinement is commonly used in finite-element methods, and vortex methods (Cottet and Poncet, 2002) may be considered a type of r-refinement method. Note that, for problems with a fixed distribution of active scales, e.g. linear advection, h- and r-refinement are essentially equivalent.

Mimetic approaches, in contrast, do not attempt to decrease the error by changing the distribution of grid points or the order of the discretization, rather they ensure that the discrete version of the PDE retains important symmetry and conservation properties of the continuous equations. For example, if the continuous differential operator is skew-symmetric, the discretization of this operator would also be skew-symmetric. Similarly, the discrete equations could ensure mass conservation, or have a discrete maximum principle equivalent to that of the continuous equations. Mimetic discretizations are especially useful when h- or p-refinement is not sufficient to resolve all the active scales of motion, i.e. the simulation must remain under-resolved because of limited computational resources. Climate and weather models are important examples of such under-resolved simulations, since the active scales of motion in the atmosphere and oceans range from $O(10^4)$ km to fractions of a millimetre, making fully-resolved simulations impossible, even with optimal adaptive techniques. It is often claimed that mimetic methods produce qualitatively more accurate results than non-mimetic methods, especially for under-resolved problems (Verstappen and Veldman, 1997a,b). In other words, mimetic methods produce consistent approximations, even when the convergence error is large.

The so-called TRiSK scheme introduced recently by Ringler *et al.* (2010) is a mimetic finite-volume/finite-difference method that discretizes the rotating shallow-water equations (SWEs) on arbitrarily structured C-grids on the sphere and ensures discrete conservation of mass and either total energy or potential enstrophy (to within time integration error). This discretization also ensures that the potential enstrophy is *compatible* and *consistent*. Compatibility means that there is no spurious numerical generation of potential vorticity and consistency means that a constant potential vorticity field q remains constant for all time: $\mathrm{D}_t q = 0$. The TRiSK scheme is a generalization of the mimetic C-grid scheme first investigated by Sadourny (1975). The mimetic properties of the TRiSK scheme ensure that it performs well for under-resolved problems like climate modelling and weather prediction. However, convergence tests show that it is a low-order method, with convergence rates between first- and second-order accuracy. Thus, a dynamically adaptive h-refinement grid structure is needed to fully exploit its mimetic properties while ensuring sufficiently accurate results on inhomogeneous and non-stationary problems, such as atmosphere and ocean dynamics. (p-refinement is not possible since the scheme has fixed order.) The goal of the present work is to develop a dynamically adaptive multiscale wavelet h-refinement grid structure for the TRiSK discretization that retains its important mimetic properties.

Dynamically adaptive wavelet Galerkin and finite-difference/finite-volume methods for PDEs have been developed over the past 15 years (Dumont and Lebon, 1996; Vasilyev and Paolucci, 1996; Fröhlich and Schneider, 1996, 1997; Schneider *et al.*, 1997; Vasilyev and Bowman, 2000; Roussel *et al.*, 2003; Kevlahan and Vasilyev, 2005; Roussel and Schneider, 2010) for a variety of nonlinear PDEs, especially in fluid dynamics and combustion. In particular, Cohen *et al.* (2003) developed an adaptive wavelet multiresolution method for conservation laws on a triangular grid that shares some features of the method we develop here. In fact, we previously developed an adaptive wavelet collocation method (AWCM) for solving PDEs on the sphere using second-generation bi-orthogonal wavelets (Mehra and Kevlahan, 2008). These adaptive wavelet methods dynamically refine the computational grid to achieve the desired error tolerance at each time step. In particular, it can be shown that nonlinear wavelet filtering (the basis of wavelet adaptivity) gives an optimal N-term approximation to a sufficiently smooth function (Cohen *et al.*, 2002). However, to the best of our knowledge, all these methods use collocated grids and tensor-products of one-dimensional (i.e. separable) wavelets.

Another way of achieving adaptivity is to use a so-called adaptive mesh refinement (AMR) method such as GeoClaw (George and LeVeque, 2006). As in the TRiSK method, GeoClaw also uses a conservative finite-volume approach, but on a logically Cartesian grid mapped to the sphere. The similarities and difference between the AMR and wavelet methods for providing dynamically adaptive grids are discussed in more detail in section 4.

Although these approaches work well for fully-resolved simulations, or when efficient subgrid-scale models are available (Goldstein *et al.*, 2005), they cannot be used for the numerical models of geophysical fluid dynamics which use staggered grids for pressure and velocity and mimetic discretizations. In particular, in the C-grid used in the TRiSK model, pressure is discretized at the centres of a primal hexagonal grid, while vorticity is discretized at the centres of the dual triangular grid (velocity components are located at the mid-point of the triangle edges). Developing an adaptive wavelet method for such a C-grid discretization presents new fundamental challenges compared with all existing approaches on collocated Cartesian grids:

1. Separate wavelet transforms (and associated wavelets) are required for the pressure and velocity variables.
2. The grids for pressure (hexagons) and velocity (triangles) are non-Cartesian and the hexagons are not nested when the grid is refined.
3. A non-separable vector-valued transform is required for the velocity. Such vector-valued wavelet transforms do not currently exist.
4. The dynamical refinement of the pressure and velocity grids must produce consistent, controllable, errors for the tendencies $\partial_t p$ and $\partial_t \mathbf{u}$. This is non-trivial since each tendency involves both independent variables.

5. The adaptive scheme must retain the important mimetic properties of the original TRiSK scheme: conserve mass and use a compatible and consistent discretization of the potential vorticity.

Although our ultimate goal is to develop an adaptive wavelet method for the TRiSK scheme on the sphere, in order to simplify the presentation the current article derives and verifies the method for the planar C-grid. The hexagonal–triangular planar C-grid is topologically similar to the subdivision of the icosahedral grid on the sphere considered by Ringler *et al.* (2010), apart from the 12 pentagonal points. The main differences are technical and due to the non-uniformity of the dyadically subdivided grids on the sphere, and the method presented here is currently being extended to the sphere.

The dynamically adaptive wavelet method derived here represents a new class of adaptive wavelet methods for PDEs on non-Cartesian staggered grids. It also forms part of a broader effort to investigate the potential of adaptive numerical methods for climate and weather models to increase accuracy (by ensuring spatially and temporally *uniform* error control) and to make optimal use of the available computational resources. In the present method, the local resolution of the pressure and velocity grids is controlled by a single tolerance parameter which coarsens or refines the local grid by locally removing or adding scales to the dyadic multiscale grid to maintain the desired L^∞ error in the pressure and velocity tendencies at each time step. Because this is a multiscale method using a hierarchy of grids, the adapted grid is always structured and there are no spurious errors associated with jumps in grid spacing. In addition to providing dynamically adaptive grid refinement, this approach also allows for static grid refinement and is natural for multiscale modelling of physical processes.

In the following section we derive the relevant wavelet transforms and introduce the adaptive wavelet method for the TRiSK scheme. In section 3, the error control estimates are verified and the method is used to solve the problem of a propagating wave packet in the inertia–gravity wave regime and rotating shallow-water turbulence in the quasi-geostrophic regime.

2. Method

2.1. TRiSK discretization of the rotating shallow-water equations

In preparation for deriving the dynamically adaptive wavelet method, we briefly review the relevant properties of the TRiSK scheme. For complete details, the reader is referred to the original papers (Thuburn *et al.*, 2009; Ringler *et al.*, 2010). The discrete equations are derived from the vector-invariant form of the momentum equations:

$$\frac{\partial p}{\partial t} = -\mathrm{div}(p\mathbf{u}), \qquad (1)$$

$$\frac{\partial \mathbf{u}}{\partial t} = -q\,(p\mathbf{u}^{\perp}) - \mathrm{grad}B, \qquad (2)$$

where p is the fluid pressure (proportional to fluid thickness), $\mathbf{u}$ is the fluid velocity, $\mathbf{u}^{\perp} = \mathbf{k}\times\mathbf{u}$, $q = (\mathbf{k}\cdot\mathrm{curl}\,\mathbf{u} + f)/p$ is the potential vorticity, $B = p + gb + K$ is the Bernoulli function and $K = |\mathbf{u}|^2/2$ is the kinetic energy. The three parameters in the system are gravity, g, the Coriolis parameter, $f = 2\Omega\sin\phi$ (where Ω is the rotation of the Earth and ϕ is the latitude), and bottom topography, b. For simplicity we assume $b = 0$ and $g = 1$ in the remainder of the article. Velocity divergence and vorticity (which together form a complete description of the flow field) are derived from the velocity equation (2).

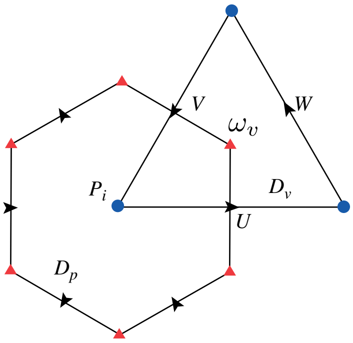


Figure 1. Discrete system for the rotating shallow-water equations on the C-grid with hexagonal primal mesh cells P_i and triangular dual mesh cells D_v. Pressure p_i is located at the hexagon centres x_i (indicated by the blue circles and the subscript i), vorticity ω_v is located at triangle centres x_v (indicated by the red triangles and subscript v). Velocity components u_e are located at the intersection of the triangle and hexagon edges x_e (indicated by the black arrows and the subscript e) and are normal to the hexagon edges and tangential to the triangle edges. U, V, and W are the projections of the velocity vector in the three directions tangential to the triangle edges. Mass is integrated over the hexagons and circulation is integrated over the triangles. This figure is available in colour online at wileyonlinelibrary.com/journal/qj

The vector-invariant rotating SWEs (1), (2) are discretized on the staggered C-grid. On this grid pressure and divergence are located at the centres of the primal grid (the hexagon centres, or triangle vertices) and vorticity (or circulation) is located at the centres of the dual grid (the triangle centres, or hexagon vertices). Gradients, fluxes and velocities are located at the coincident mid-points of the triangle and hexagon edges. The C-grid configuration is shown in Figure 1. The resulting discretized system is

$$\frac{\partial p_i}{\partial t} = -[\mathrm{div}\,F_e]_i, \qquad (3)$$

$$\frac{\partial \mathbf{u}_e}{\partial t} = F_e^{\perp}\widehat{q}_e - [\mathrm{grad}\,B_i]_e, \qquad (4)$$

where $F_e = \widehat{p}_e u_e$ is the thickness flux (i.e. the flux of pressure normal to a hexagon edge) and $F_e^{\perp}$ is the thickness flux in the direction perpendicular to F_e (i.e. the flux of pressure normal to a triangle edge). Ringler *et al.* (2010) show that with appropriate definitions for the four discrete scalars $(\widehat{p}_e, \widehat{q}_e, K_i, F_e^{\perp})$ and the three discrete differential operators $([\mathbf{k}\cdot\mathrm{curl}(\,)]_v, [\mathrm{grad}(\,)]_e, [\mathrm{div}(\,)]_i)$ the associated potential vorticity equation is consistent with an underlying thickness evolution equation and is compatible with the discrete momentum equation. As mentioned earlier, the TRiSK scheme can conserve *either* the total energy or potential enstrophy; we choose to implement the latter version in the adaptive wavelet method.

The TRiSK scheme for the rotating SWEs thus consists of prognostic finite-volume equations for pressure (1) and finite-difference equations for velocity (2) which specify the derived scalars $\widehat{p}_e$, $\widehat{q}_e$, K_i, $F_e^{\perp}$. Although the discrete equations do not include either physical or numerical dissipation of total energy, the adaptive wavelet method introduces dissipation due to the coarsening associated with neglecting weak gradients during grid adaptation. This numerical dissipation is quantified for rotating shallow-water turbulence simulations in section 3.

In the remainder of this section, we derive the dynamically adaptive wavelet method for the TRiSK discretization of the rotating SWEs (1) and (2). This method is based on a bi-orthogonal second-generation wavelet multiresolution analysis, and retains the conservation, consistency and compatibility properties of the TRiSK discretization, while providing dynamical error control by automatically coarsening or refining the computational grid to maintain a desired error tolerance ε in the tendencies $\partial_t p$ and $\partial_t \mathbf{u}$ at each time step. The next section reviews briefly the main features of a second-generation bi-orthogonal wavelet transform.

2.2. *Bi-orthogonal second-generation wavelet transform*

A wavelet transform takes a signal on a fine grid and decomposes it into a sequence of increasingly smooth approximations (spanned by the *scaling functions*) and the details which are lost between successive smooth approximations (spanned by the *wavelets*). Bi-orthogonal second-generation wavelets (Sweldens, 1996) are particularly well-suited to numerical approximation of PDEs since in this case the scaling functions and wavelets correspond directly to unique grid points. Such wavelets can be constructed in physical space using a simple procedure called *lifting*. In this section we give the basic idea behind the bi-orthogonal wavelet transforms used in this article. The interested reader is referred to Mallat (1998) for full details about wavelet multiresolution analysis and wavelet filtering.

The wavelet transform can be understood by considering a single two-scale transform from a fine scale $j+1$ to a coarse scale j. Following Daubechies and Sweldens (1998), let us consider a signal $\mathbf{x} = \{x_k\}_{k\in Z}$, $x_k \in \mathbb{R}$. The first step in the wavelet transform is the *restriction* (or coarse-graining) of the signal, which in the simplest case splits the signal into even components $\mathbf{x}_e = \{x_{2k}\}$ and odd components $\mathbf{x}_o = \{x_{2k+1}\}$ and then deletes the odd components (i.e. sub-samples). The deleted points are then *predicted* (or prolonged) using the coarse values x_{2k}, for example using polynomial interpolation P. The difference between the predicted and the actual deleted points are the wavelet coefficients $\widetilde{\mathbf{x}} = \{\widetilde{x}_m\}$, i.e. $\widetilde{\mathbf{x}} = \mathbf{x}_o - P(\mathbf{x}_e)$, where the wavelet locations m are the locations of the odd values x_{2k+1}. Computing a prediction and recording the detail is the first lifting step. At this stage we have a transform $(\mathbf{x}_e, \mathbf{x}_o) \to (\mathbf{x}_e, \widetilde{\mathbf{x}})$. The prediction step using linear interpolation is described graphically in Figure 2. If the wavelet coefficient is small, then the associated point at the finer scale can be interpolated with high accuracy from the coarse points. In other words, the wavelet coefficient measures the variation of the signal (or, equivalently, the local interpolation error) at a particular location and scale.

However, after this basic restriction and prolongation lifting step, the frequency localization is poor due to aliasing and the average is not conserved, i.e. $\overline{\mathbf{x}}_e \neq \overline{\mathbf{x}}$. To correct these (or other) deficiencies, additional lifting steps can be performed. These additional lifting (or *update*) steps modify the values at the coarse grid points by adding linear combinations of the wavelet coefficients, i.e. $\mathbf{y} = \mathbf{x}_e + U(\widetilde{\mathbf{x}})$. For example, to preserve the running average the missing grey area in Figure 2 must be restored. This can be achieved for the linear prolongation operator by adding $\widetilde{x}_m/4$ to the neighbouring values on the coarse grid. Note that the complete transform from values on even and odd points to smoothed values on a coarse grid and wavelet coefficients on the neglected grid, $(\mathbf{x}_e, \mathbf{x}_o) \to (\mathbf{y}, \widetilde{\mathbf{x}})$, is fully invertible. The general two-scale lifted wavelet transform is given in Algorithm 1.

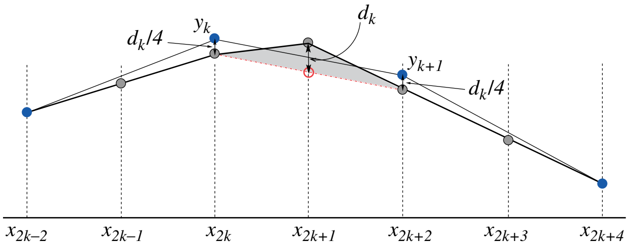


Figure 2. Lifted wavelet transform using linear interpolation. The thick line is the original signal on the fine grid and the thin line is the smooth signal on the coarse grid obtained by linear interpolation and lifting. The first stage in the smoothing (or restriction) is simply to neglect the odd points x_{2k+1}. The wavelet coefficient $\widetilde{x}_m$ is then the difference between the actual signal value x_{2k+1} and the value predicted by linear interpolation (indicated by the open red circle). In order to retain the moving average of the signal, the coarse grid values x_{2k} and x_{2k+2} are lifted by adding $\widetilde{x}_m/4$ to obtain the coarse values y_k and y_{k+1}. (Figure adapted from Daubechies and Sweldens, 1998). This figure is available in colour online at wileyonlinelibrary.com/journal/qj

Algorithm 1. General two-scale lifted wavelet transform from complete set of fine-scale values $x_k^{j+1} \cup x_m^{j+1}$ to coarse-scale values x_k^j and wavelet coefficients $\widetilde{x}_m^j$. $\mathcal{M}^j$ is the set of wavelet points at scale j (i.e. those fine points that are predicted from the coarse points), $\mathcal{K}_m$ is the stencil used for the prediction at point $m \in \mathcal{M}^j$, and $\widetilde{s}^j_{k,m}$ and $s^j_{k,m}$ are respectively the filter coefficients for the prediction and update steps.

Analysis:

$$\begin{aligned} \widetilde{x}_m^j &= x_m^{j+1} - \sum_{k\in\mathcal{K}_m^j} \widetilde{s}^j_{k,m} x_k^{j+1}, \\ x_k^j &= x_k^{j+1} + \sum_{m\in\mathcal{M}_k^j} s^j_{k,m} \widetilde{x}_m^j. \end{aligned}$$

Synthesis:

$$\begin{aligned} x_k^{j+1} &= x_k^j - \sum_{m\in\mathcal{M}_k^j} s^j_{k,m} \widetilde{x}_m^j, \\ x_m^{j+1} &= \widetilde{x}_m^j + \sum_{k\in\mathcal{K}_m^j} \widetilde{s}^j_{k,m} x_k^j. \end{aligned}$$

For a one-dimensional signal of length 2^J, the two-scale transform described above is repeated 2^{J-1} times to generate the complete wavelet transform,

$$\mathbf{x} \to \left\{x_k^0, \widetilde{x}_m^j\right\}, \quad k \in \mathcal{K}_0, \quad j = 0, \ldots, J-1, \quad m \in \mathcal{M}^j. \quad (5)$$

The full wavelet transform (5) has linear computational complexity, $O(N)$, for a signal of length N and is invertible.

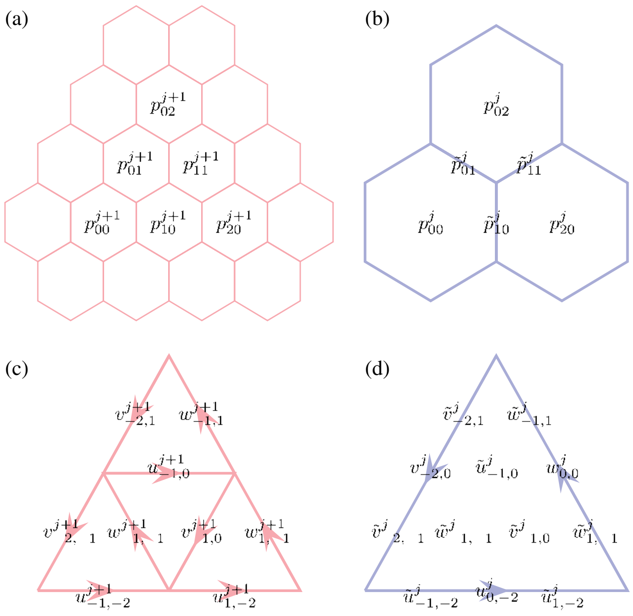


Figure 3. Arrangement of scaling function and wavelet coefficients on the multiscale triangular–hexagonal C-grid. (a, b): pressure on the hexagonal grid. (c, d): velocity on the triangular grid. (a, c): scaling function coefficients at the fine scale $j+1$. (b, d): scaling function and wavelet coefficients at the coarse scale j. The coarse scaling function coefficients p^j and u^j, v^j, w^j are smoothed versions of the pressure and velocity respectively. The wavelet coefficients are the difference between the exact value on the fine grid and the value interpolated from the coarse grid using the coarse scaling function coefficients. On each coarse triangular edge, the two velocity wavelet coefficients are equal and opposite, e.g. $\widetilde{u}^j_{-1,-2} + \widetilde{u}^j_{1,-2} = 0$. This figure is available in colour online at wileyonlinelibrary.com/journal/qj

The two-scale wavelet transform given by Algorithm 1 is very general: for such a transform to exist in any number of dimensions, one simply needs a multiscale sequence of grids, a restriction operator from grid $j+1$ to grid j and a prolongation operator from grid j to grid $j+1$ (the *prediction*). One can then define suitable lifting (or *update*) steps to improve the properties of the wavelet transform (e.g. increase the number of vanishing moments or conserve the mean).

Figure 3 shows the positions of the scaling functions and wavelets for pressure and velocity for one coarsening step of the C-grid. We use this multiscale grid structure to construct the adaptive wavelet scheme for the TRiSK equations. Note that, although the grid points for the pressure are nested (i.e. $x^j_i \subset x^{j+1}_i$), the associated hexagonal finite volumes containing the mass are not. Conversely, the grid points for velocity are not nested, but the triangular finite volumes containing the circulation are. The fact that the grid points and finite volumes are not nested makes developing the wavelet transforms for pressure and velocity more challenging.

Finally note that, if a wavelet coefficient is smaller than the desired tolerance ε, then the associated grid point may be neglected since it can be interpolated from values at the coarse grid points with error $O(\varepsilon)$. Removing wavelets with small magnitude is called nonlinear wavelet filtering. Nonlinear wavelet filtering generates a multiscale hierarchy of adapted grids and ensures that the pressure and velocity can be constructed to the desired tolerance ε on the adapted grid. This error control is at the heart of the method that we develop in the following subsections.

$$\begin{tikzcd}
 & p^0_i, u^0_e \arrow[r, red, "T"] \arrow[d, blue, "I"] & B^0_i, F^0_e, q^0_e F^{\perp 0}_e \arrow[r, red, dashed, "T"] & \frac{\partial}{\partial t}(p^0_i, u^0_e) & \\
\tilde{p}^0_m, \tilde{u}^0_n \arrow[r, blue] & p^1_i, u^1_e \arrow[r, red, "T"] \arrow[d, blue, "I"] & B^1_i, F^1_e, q^1_e F^{\perp 1}_e \arrow[r, red, dashed, "T"] & \frac{\partial}{\partial t}(p^1_i, u^1_e) \arrow[u, green, dashed, "R"] \arrow[r, green] & \frac{\partial}{\partial t}(\tilde{p}^0_m, \tilde{u}^0_n) \\
\tilde{p}^1_m, \tilde{u}^1_n \arrow[r, blue] & p^2_i, u^2_e \arrow[r, red, "T"] & B^2_i, F^2_e, q^2_e F^{\perp 2}_e \arrow[r, red, "T"] & \frac{\partial}{\partial t}(p^2_i, u^2_e) \arrow[u, green, dashed, "R"] \arrow[r, green] & \frac{\partial}{\partial t}(\tilde{p}^1_m, \tilde{u}^1_n)
\end{tikzcd}$$

Figure 4. Non-conservative computation of the tendencies $\partial_t \widetilde{p}^j_m$, $\partial_t \widetilde{u}^j_n$, $\partial_t p^0_i$, $\partial_t u^0_e$ from $\widetilde{p}^j_m$,$\widetilde{u}^j_n$, p^0_i and u^0_e. In the above method conservation of total mass is *not* guaranteed. Blue arrows correspond to step 1, red arrows to step 2 and green arrows to step 3. The interpolation, restriction and TRiSK operators are labeled as I, R and T respectively. Dashed arrows indicate that either restriction or TRiSK operators are used. In this minimal example the finest level is at $j = 2$. Therefore the lowermost red T-arrow is solid as no data is available from level $j+1$ to apply the restriction operators. Additional levels can be added by repeating the intermediate line $j = 1$. This figure is available in colour online at wileyonlinelibrary.com/journal/qj

The next section describes the algorithm for calculating the tendencies $\partial_t p$ and $\partial_t \mathbf{u}$ on the adapted grids, and the subsequent two sections derive the wavelet transforms for the pressure and velocity satisfying the particular requirements of the tendency algorithm. Finally, we analyze the conditions for the stability of the grid adaptation algorithm and determine the appropriate scalings for the pressure and velocity filter thresholds to control the tendency error.

2.3. *Algorithm for calculating the tendency*

In the adaptive scheme, adaptivity is achieved by neglecting those wavelet coefficents $\widetilde{p}^j_m$ and $\widetilde{u}^j_n$ that are below a predefined threshold. Therefore, the quantities evolved in time are the (active) wavelet coefficients $\widetilde{p}^j_m$ and $\widetilde{u}^j_n$, together with the scaling coefficients p^0_i and u^0_e at the coarsest scale. A standard time-stepping scheme is used. The main task is then to compute the tendencies $\partial_t \widetilde{p}^j_m$, $\partial_t \widetilde{u}^j_n$, $\partial_t p^0_i$, $\partial_t u^0_e$ from $\widetilde{p}^j_m$,$\widetilde{u}^j_n$, p^0_i and u^0_e. This procedure involves the calculation of various intermediate quantities such as the scaling coefficients p^j_i and u^j_e and their tendencies $\partial_t p^j_i$ and $\partial_t u^j_e$, as well as the mass flux F^j_e and the Bernoulli function B_i as depicted in Figures 4 and 6. As discussed below, the adaptive computation of tendencies depicted in Figure 4 does not guarantee that total mass $\sum_i p^0_i$ is conserved. Therefore we introduce and implement a slightly different method depicted in Figure 6, which does guarantee conservation of total mass and a consistent vorticity budget across scales.

Both adaptive calculations involve two categories of discrete operators:

- *One-scale operators* involve a single resolution level j. These operators compute the mass flux F^j_e, kinetic energy K^j_i, Bernoulli function B^j_i, potential vorticity q_e, rotated mass flux $F^{\perp}_e$ from which the tendencies $\partial_t p^j_i$ and $\partial_t u^j_e$ are computed. One-scale operators are given by the underlying numerical method, in this case the TRiSK scheme.
- *Two-scale operators* involve two resolution levels j and $j+1$. These operators perform the interpolation and restriction steps of the wavelet transforms for pressure and velocity. They are essentially independant from the equations being solved. In the conservative

$$\begin{tikzcd}
B_i^0,\, F_e^0,\, u_e^0 \arrow[r, "\mathrm{grad}^0,\, \mathrm{div}^0, \mathrm{curl}^0"] & \mathrm{grad}\ B_i^0,\ \mathrm{div}\ F_e^0,\ \mathrm{curl}\ u_e^0 \\
B_i^1,\, F_e^1,\, u_e^1 \arrow[u, "R"] \arrow[r, "\mathrm{grad}^1,\, \mathrm{div}^1,\, \mathrm{curl}^1"] & \mathrm{grad}\ B_i^1,\ \mathrm{div}\ F_e^1,\ \mathrm{curl}\ u_e^1 \arrow[u, "R"']
\end{tikzcd}$$

Figure 5. Commutation diagram of gradient, divergence and curl with the appropriate restriction operators.

$$\begin{tikzcd}
 & p_i^0, u_e^0 \arrow[r, dashed, "T"] \arrow[d, "I"] & B_i^0, F_e^0, q_e^0 F_e^{\perp 0} \arrow[r, "T"] & \frac{\partial}{\partial t}(p_i^0, u_e^0) & \\
\tilde{p}_i^0, \tilde{u}_e^0 \arrow[r] & p_i^1, u_e^1 \arrow[r, dashed, "T"] \arrow[d, "I"] & B_i^1, F_e^1, q_e^1 F_e^{\perp 1} \arrow[u, dashed, "R"] \arrow[r, "T"] & \frac{\partial}{\partial t}(p_i^1, u_e^1) \arrow[r] & \frac{\partial}{\partial t}(\tilde{p}_i^0, \tilde{u}_e^0) \\
\tilde{p}_i^1, \tilde{u}_e^1 \arrow[r] & p_i^2, u_e^2 \arrow[r, "T"] & B_i^2, F_e^2, q_e^2 F_e^{\perp 2} \arrow[u, dashed, "R"] \arrow[r, "T"] & \frac{\partial}{\partial t}(p_i^2, u_e^2) \arrow[r] & \frac{\partial}{\partial t}(\tilde{p}_i^1, \tilde{u}_e^1)
\end{tikzcd}$$

Figure 6. Conservative adaptive computation of the tendencies $\partial_t\widetilde{p}^j_m$, $\partial_t\widetilde{u}^j_n$, $\partial_t p_i^0$, $\partial_t u_e^0$ from $\widetilde{p}^j_m$,$\widetilde{u}^j_n$, p_i^0 and u_e^0. In the above method conservation of total mass *is* guaranteed. Blue arrows correspond to step 1 (synthesis, algorithm 2.2), red arrows to step 2 and green arrows to step 3. The interpolation, restriction and TRiSK operators are labeled as I, R and T respectively. Dashed arrows indicate that either restriction or TRiSK operators are used. In this minimal example the finest level is at $j = 2$. Therefore the lowermost red T-arrow is solid as no data is available from level $j + 1$ to apply the restriction operators. Additional levels can be added by repeating the intermediate line $j = 1$. This figure is available in colour online at wileyonlinelibrary.com/journal/qj

method, distinct restriction operators for the mass flux F_e^j and Bernoulli function B_i^j are introduced.

2.3.1. *Non-conservative adaptive algorithm*

Let us first describe and discuss the non-conservative computation of tendencies depicted in Figure 4. The computation involves three steps:

1. An inverse wavelet transform is executed to compute the scaling coefficients p_i^j and u_e^j from $\widetilde{p}^j_m$, $\widetilde{u}^j_n$, p_i^0 and u_e^0. The data flow is from the coarsest level to the finest level at which non-zero wavelet coefficients exist.
2. The TRiSK operators are applied separately at each grid level to compute the tendencies $\partial_t p_i^j$ and $\partial_t u_e^j$.
3. A forward wavelet transform is executed to compute the tendencies $\partial_t\widetilde{p}^j_m$, $\partial_t\widetilde{u}^j_n$, $\partial_t p_i^0$, $\partial_t u_e^0$ from the tendencies $\partial_t p_i^j$ and $\partial_t u_e^j$. The data flow is from the finest level to the coarsest level.

The total operation count is proportional to the number of active wavelet coefficients provided all the above steps are actually performed on a subset of the indices e and i only. This is possible because the task is only to compute $\partial_t\widetilde{p}^j_m$, $\partial_t\widetilde{u}^j_n$ wherever the coefficients $\widetilde{p}^j_m$, $\widetilde{u}^j_n$ are active.

Notice that steps 2 and 3 can provide two conflicting values for $\partial_t p_i^j$ and $\partial_t u_e^j$. Indeed, $\partial_t p_i^j$ and $\partial_t u_e^j$ can be obtained either from p_i^j and u_e^j by applying the one-scale operators (step 2), or from $\partial_t p_i^{j+1}$ and $\partial_t u_e^{j+1}$ by applying the restriction operators (step 3). The latter must be preferred when possible because fine-grid computations are more accurate. The appropriate subsets would therefore be determined as follows, following the data flow of the calculation backwards:

1. Given the set of active wavelet coefficients, determine separately at each level j those indices i and e that are required to compute $\partial_t\widetilde{p}^j_m$, $\partial_t\widetilde{u}^j_n$ from the tendencies $\partial_t p_i^{j+1}$ and $\partial_t u_e^{j+1}$. This depends on the stencil of two-scale operators.
2. Starting from the finest level, find which $\partial_t p_i^j$ and $\partial_t u_e^j$ can be obtained by applying the restriction operators to $\partial_t p_i^{j+1}$ and $\partial_t u_e^{j+1}$. For those which cannot (because not all the necessary $\partial_t p_i^{j+1}$ and $\partial_t u_e^{j+1}$ are available), determine those indices i and e for which we need p_i^j and u_e^j as an input to the one-scale (TRiSK) operators. This depends on the stencil of the one-scale operators.
3. Given those indices i and e for which we need p_i^j and u_e^j as an input to the one-scale (TRiSK) operators, find the minimal set of indices i and e which allow the inverse wavelet transform to be performed. Indeed, we must guarantee that all indices needed at level j to interpolate to level $j + 1$ where required are present. This is done starting from the finest level.

In this process the adapted grids emerge as sets of indices i and e and are a by-product of the set of active wavelet coefficients and the stencils of the one-scale and two-scale operators.

An important role of the two-scale operators is to blend together the computations performed at various resolution levels. This blending occurs when one chooses between applying the finite-difference operators on the current resolution level and applying a restriction operator to tendencies already computed at the immediately finer level. However this blending also breaks the exact discrete conservation of total mass. In fact, the property $\sum_i \partial_t p_i^0 = 0$ relies on $\partial_t p_i^0$ being computed as the divergence of the mass flux, and on the overall cancellation of the mass fluxes as each flux is counted once positively and once negatively. This cancellation does not occur in the presence of a mix of $\partial_t p_i^0$ computed as a flux divergence and $\partial_t p_i^0$ computed as the restriction from $\partial_t p_i^1$. To recover the exact discrete conservation of total mass, we introduce a new, conservative adaptive algorithm.

2.3.2. *Conservative adaptive algorithm*

The most obvious way to recover the property $\sum_i \partial_t p_i^0 = 0$ is to insist that *all* $\partial_t p_i^0$ be computed as the divergence of a mass flux. This is what we do in the algorithm depicted in Figure 6. In this modified algorithm, the restriction operator is not applied to the pressure tendency $\partial_t p_i^{j+1}$ but to the mass flux F_e^{j+1} instead. This modification is admissible if the flux restriction operator, the discrete divergence operator, and the pressure restriction operator are compatible in the following sense: restricting the mass flux from level $j + 1$ to level j then computing the discrete divergence on level j must produce the same result as computing the mass flux divergence at level $j + 1$ then restricting this pressure tendency to level j, i.e. the diagram presented in Figure 5 commutes. This property means that the mass budget can be expressed in a consistent way at all resolution levels. We address the problem of constructing the restriction operator for the mass flux together with the description of the wavelet transform for pressure.

Algorithm 2. Conservative adaptive computation of tendencies.

1. Execute an inverse wavelet transform to compute the scaling coefficients p_i^j and u_e^j from $\widetilde{p}_m^j$, $\widetilde{u}_n^j$, p_i^0 and u_e^0. The data flow is from the coarsest level to the finest level at which non-zero wavelet coefficients exist.
2. Compute the mass flux F_e, Bernoulli function B_i and and potential vorticity flux $F_e^\perp \widehat{q}_e$ in a loop starting at the finest level:
   - (a) Compute B_e^j, F_e^j by restriction from level $(j+1)$ where possible.
   - (b) Where restriction is not possible, compute F_e^j and B_e^j using the TRiSK operators applied to p_i^j and u_e^j.
   - (c) Compute$F_e^{j\perp}\widehat{q}_e^j$ by restriction from level $(j+1)$ where possible.
   - (d) Where restriction is not possible, compute $F_e^{j\perp}$ from F_e^j and $\widehat{q}_e^j$ from u_e^j and p_i^j using the TRiSK operators.
3. At each level separately, apply the TRiSK operators to B_e^j, F_e^j and $F_e^{j\perp}\widehat{q}_e^j$ to compute $\partial_t p_i^j$ and $\partial_t u_e^j$ then obtain the tendencies $\partial_t \widetilde{p}_m^j$, $\partial_t \widetilde{u}_n^j$.

A similar idea is applied to the gradient of the Bernoulli function: instead of restricting $[\text{grad}\, B_i]_e$, we restrict B_i and compute its gradient at each scale. This guarantees the absence of spurious generation of vorticity by the adaptive method. As explained in subsection 2.5, the compatible restriction operator for the Bernoulli function is simply the subsampling operator $B_i^{j+1} \mapsto B_i^j = B_i^{j+1}$. With these modifications, the computation involves four steps (Algorithm 2).

For Algorithm 2 to work, one must take care that each operator finds the input data it needs to compute the output data needed by the next operator. Therefore, before the computation is performed, the subsets of indices on which each operator is applied is determined following the data flow backwards (Algorithm 3). Starting from the set of active wavelet coefficients, other subsets are computed in sequence by taking into account the stencil of each operator encountered during the algorithm. An important resulting subset is the subset of scaling coefficients needed as input by the TRiSK operators. Notice that, because the TRiSK operators also use values from first or second neighbours as input, this 'input' subset is wider than the 'output' subset of scaling coefficients, tendencies of which are computed. Nevertheless all the necessary 'input' scaling coefficients can always be computed by the inverse wavelet transform, even if this computation involves inactive wavelet coefficients. Indeed, in that case the missing wavelet coefficients are simply assumed to be zero, and the inverse wavelet transform effectively performs an interpolation. This way the multiscale nature of the grid is hidden from the TRiSK operators, which continue to operate on their full, unmodified stencil.

Algorithm 3. Subsets involved in the adaptive computation of tendencies.

1. Given the set of active wavelet coefficients, determine separately at each level j those indices i and e that are required to compute $\partial_t \widetilde{p}_m^j$, $\partial_t \widetilde{u}_n^j$ from the tendencies $\partial_t p_i^j$ and $\partial_t u_e^j$. Then determine which F_e^j and B_i^j need to be computed.
2. Starting from the finest level,
   - (a) Find which $F_e^{j\perp}\widehat{q}_e^j$ can be obtained by applying the velocity restriction operator. For those which cannot, determine which p_i^j, u_e^j and F_e^j are needed as an input to the TRiSK operators.
   - (b) Find which F_e^j and B_i^j can be obtained by restriction from level $(j+1)$. For those which cannot, determine which p_i^j and u_e^j are needed as an input to the TRiSK operators.
3. Given those indices i and e for which we need p_i^j and u_e^j as an input to the TRiSK operators, find the minimal set of indices i and e which allow the inverse wavelet transform to be performed.

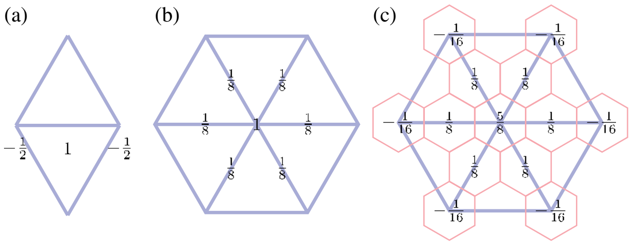


Figure 7. Stencils of the (a) interpolation, (b) lifting and (c) restriction operators for pressure. This figure is available in colour online at wileyonlinelibrary.com/journal/qj

2.4. *Wavelet transform of pressure*

To define the wavelet transform of pressure, one needs to define an interpolation and a lifting. Since pressure sits at hexagon centres, at scale $j+1$ one can distinguish between hexagons whose centres coincide with the centre of a larger hexagon from level j, and the remainder which we call 'new' hexagons. The interpolation formula interpolates from the centres of larger hexagons to the centres of 'new' hexagons. We choose a second-order centred linear interpolation formula. The resulting stencil for the computation of the wavelet coefficients is presented in Figure 7(a).

The lifting step serves to define the restriction operator. The restricted pressure is defined on the coarser grid level j as a linear combination of the pressure at the finer level $j+1$ at the same point and the nearby wavelet coefficients. We choose to use only wavelet coefficients from nearest neighbours. Elementary linear algebra shows that with a weight of 1/8, the restricted pressure field has the same average as the fine pressure field. The corresponding stencil is presented in Figure 7(b). Combining this stencil with a stencil for the wavelet coefficient one obtains the stencil of

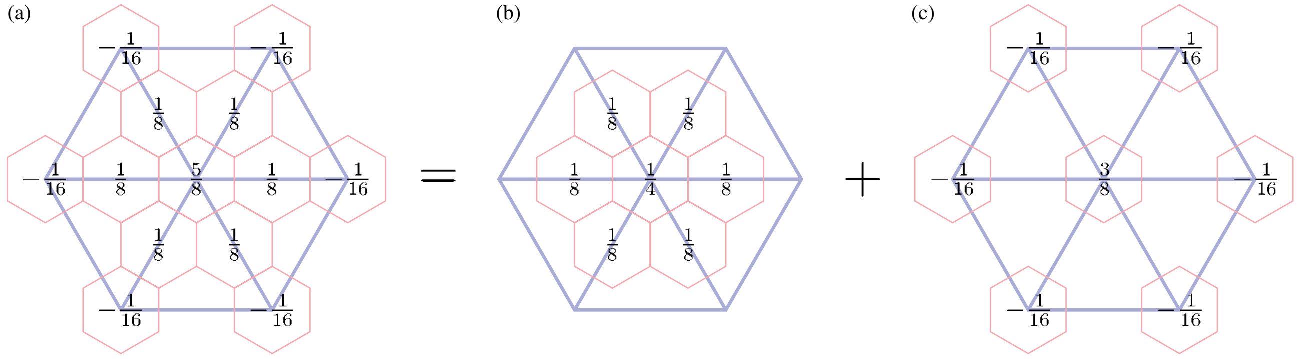


Figure 8. Decomposition of (a) the pressure restriction operator R_p^j into a sum of (b) an area-weighted restriction operator R_0^j and (c) a remainder δR_p^j. The remainder is proportional to the seven-point Laplacian on the coarse grid. This figure is available in colour online at wileyonlinelibrary.com/journal/qj

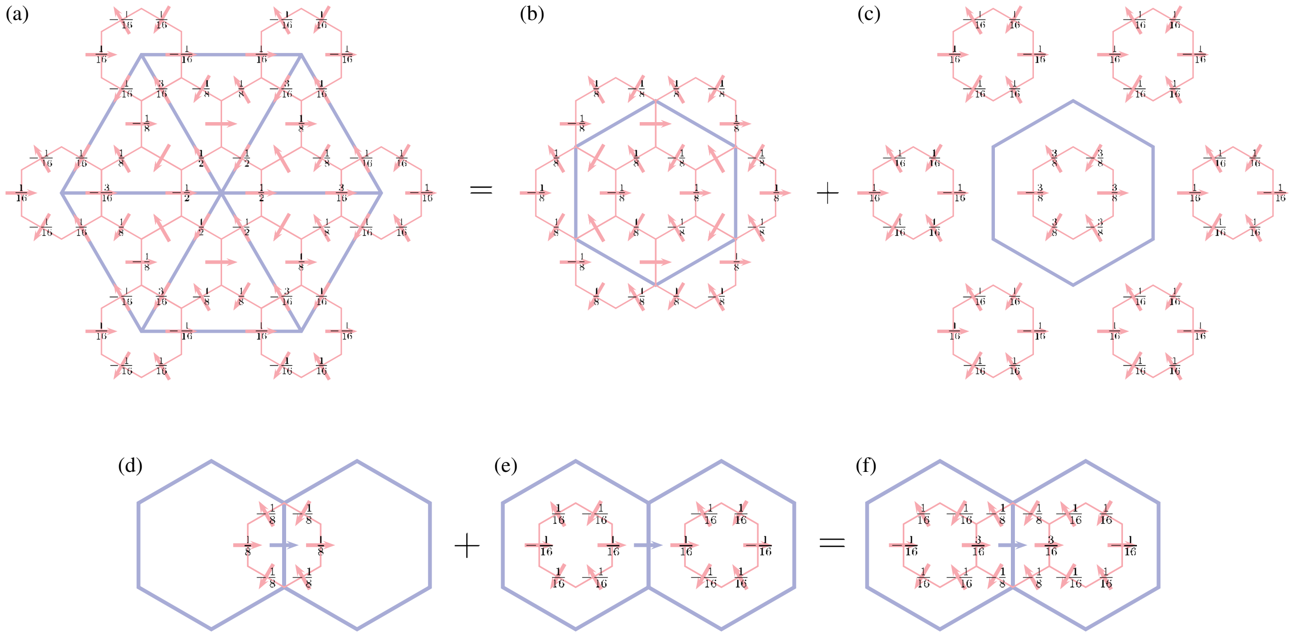


Figure 9. (a) Stencil of $R_p^j \circ \mathrm{div}^{j+1}$ (div operator followed by the pressure restriction operator) and its decomposition into (b, c) two simpler operators for which a compatible flux restriction operator can be found readily. (d) basic and (e) corrective flux restriction operators R_{0f}^j and $D^j \circ \mathrm{div}^{j+1}$ combined to give (f) the final mass flux restriction operator R_f^j, compatible with the mass restriction operator R_p^j and the div operators div^j, div^{j+1} in the sense of Figure 5. This figure is available in colour online at wileyonlinelibrary.com/journal/qj

the restriction operator $R_p^j : p_i^{j+1} \mapsto p_i^j$ (Figure 7(c)). We decompose the pressure restriction operator into two parts,

$$R_p^j = R_0^j + \delta R_p^j ,$$

where R_0^j is an area-weighted restriction operator (with weights 1/4 and 1/8) and δR_p^j is a remainder (Figure 8).

The construction of the pressure wavelet transform is then a straightforward application of the lifting scheme. We now describe a novel aspect, which is the definition of a mass flux restriction operator compatible with the pressure restriction (Figure 5). The problem is to find an operator R_f^j such that

$$\mathrm{div}^j\{R_f^j\,(F_e^{j+1})\} = R_p^j\,\{\mathrm{div}^{j+1}(F_e^{j+1})\}.$$

Figure 9(a, b, c) shows the decomposition of $R_p^j \circ \mathrm{div}^{j+1}$ into two operators: $R_0^j \circ \mathrm{div}^{j+1}$ and $\delta R_p^j \circ \mathrm{div}^{j+1}$. There is a natural flux restriction R_{0f}^j compatible with R_0^j: assuming that the flux divergence is constant within each fine hexagon, one can compute a flux through the edges of coarse hexagons. The stencil of this basic flux restriction operator R_{0f} is presented in Figure 9(d, e, f). Concerning the remainder $\delta R_p^j \circ \mathrm{div}^{j+1}$, it turns out that δR_p^j has non-zero weights only at coarse points, and that those weights are proportional to the seven-point finite-difference Laplacian (Figure 8). It is therefore already in divergence form $\delta R_p^j = \mathrm{div}^j \circ D^j$, where D^j is proportional to a finite-difference gradient. Then

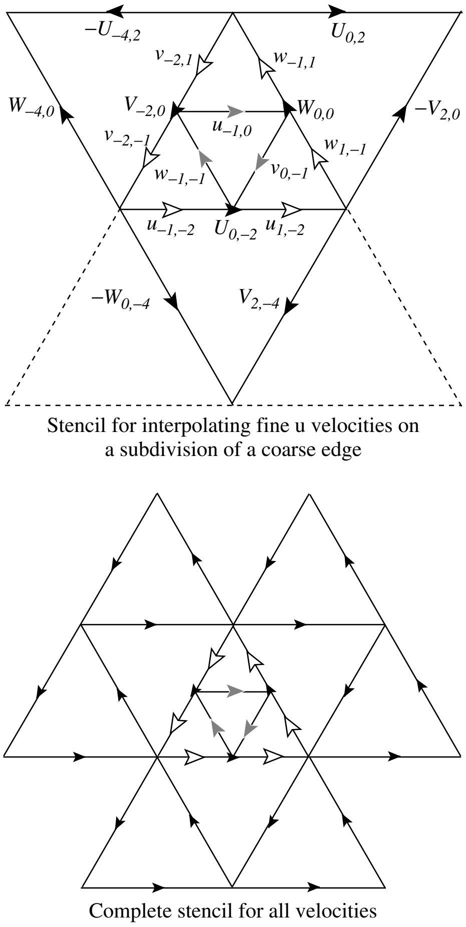
$$R_p^j \circ \mathrm{div}^{j+1} \quad = \quad \mathrm{div}^j \circ R_{0f}^j + \mathrm{div}^j \circ D^j \circ \mathrm{div}^{j+1}.$$

This provides a solution for R_f^j

$$R_f^j = R_0^j + D^j \circ \mathrm{div}^{j+1}.$$

The stencils of the corrective flux restriction $D^j \circ \mathrm{div}^{j+1}$ and the final flux restriction R_f^j are presented in Figure 9(d, e, f).

Notice that the weights presented in Figure 9 omit the metric factors (multiplication/division by lengths and areas).



Stencil for interpolating fine u velocities on a subdivision of a coarse edge

Complete stencil for all velocities

Figure 10. Stencil for circulation-conserving second-order interpolation of velocity (filled black arrows). Velocities on the bisection of coarse edges (unfilled black arrows) are obtained by linear interpolation. The velocities on inner fine triangle (grey arrows) are then obtained by assuming constant circulation over the coarse triangle. The dotted lines indicate the mirror-symmetric stencil. This stencil works for both valence 6 points (hexagon points) and and valence 5 points (pentagon points) on subdivisions of the icosahedral sphere.

As a result, the weights of the flux restriction operators presented in Figure 9(d, e, f) must be multiplied by 2. One can then check that R^j_{0f} is exact for all affine vector fields. Furthermore, the corrective flux restriction $D^j \circ \mathrm{div}^{j+1}$ vanishes on affine vector fields since such fields have a constant divergence and D^j vanishes on constant fields. R^j_f is exact on affine vector fields and is therefore second-order accurate.

2.5. *Wavelet transform of velocity*

The wavelet transform of velocity on the adapted grid must conserve circulation and, because the velocity is defined in terms of its tangential components (U, V, W) on the edges of the triangular grid, it is a vector-valued (non-separable) transform. Further, we show in section 2.6 that the interpolation operator (i.e. the prolongation operator) used in the wavelet transform must be at most second-order accurate to ensure stability of the dynamically adapted grid for the second-order finite-volume operators used in the TRiSK scheme. Finally, the interpolation stencil must be consistent with both the valence 5 (pentagonal) and valence 6 (hexagonal) points on subdivisions of the icosahedral sphere.

In the following derivation, we consider two scales: a coarse scale j and a fine scale $j+1$, where the fine grid is obtained as the bisection of the coarse triangle edges. As explained above, the wavelet coefficients are the differences between the actual velocity values on the fine grid and the values predicted by interpolating from neighbouring velocities on the coarse grid. The circulation-conserving restriction is simply the average of the two neighbouring fine velocities on a coarse edge (the interior fine velocities are not involved in the restriction). Because the discrete gradient is simply a finite difference between two neighbouring vertices, the restriction operator for the Bernoulli function compatible with the velocity restriction is simply the subsampling operator $B^{j+1}_i \mapsto B^j_i = B^{j+1}_i$. Indeed $B^j_l - B^j_n = (B^j_l - B^j_m) + (B^j_m - B^j_n)$, where l and n are coarse-grid points and m is their midpoint. Figure 10 shows the geometry of the velocity interpolation and restriction. The wavelet transform is implemented hierarchically, starting from the finest scale J, in the usual way and no additional lifting steps are required.

Let us consider interpolating the two fine velocity values $u_{-1,-2}$ and $u_{1,-2}$ shown in Figure 10. (By symmetry, the fine v and w velocities on the other coarse edges are interpolated in a similar way.) The two-dimensional linear interpolating function for the velocity $\mathbf{IU}(x,y) = \{IU_x(x,y), IU_y(x,y)\}$ has the form

$$\begin{aligned} IU_x(x,y) &= a_1 + a_2\,x + a_3\,y\,, \\ IU_y(x,y) &= b_1 + b_2\,x + b_3\,y\,, \end{aligned}$$

where the interpolated velocities on triangle edges IU, IV, IW are defined in terms of IU_x and IU_y by suitable projections. Thus, six equations for the six unknown coefficients $(a_1, a_2, a_3, b_1, b_2, b_3)$ are needed to determine the velocity interpolation operator.

To approximate the rotational part of the velocity field, and to conserve circulation around the coarse triangle, the first three equations ensure that the line integral of the interpolated velocity along each of the coarse triangle edges equals the line integral of the actual velocity:

$$\begin{aligned} U_{0,-2} &= IU(0,-2)\,, \\ V_{-2,0} &= IV(-2,0)\,, \\ W_{0,0} &= IW(0,0)\,. \end{aligned}$$

(Note that taking point values of the interpolated velocities is exact for linear functions.) The final three equations approximate the irrotational part of the velocity and involve line integrals along the outer edges of the three neighbouring coarse triangles,

$$\begin{aligned} U_{0,-2} - V_{2,0} &= IU(0,2) - IV(2,0), \\ W_{-4,0} - U_{-4,2} &= IW(-4,0) - IU(-4,2), \\ V_{2,-4} - W_{0,-4} &= IV(2,-4) - IW(0,-4). \end{aligned}$$

The alternating signs of the velocity on each pair of exterior edges (shown by the arrows in Figure 10) capture the irrotational component of the velocity field. These rotational and irrotational conditions on the line integrals form a linear system with a unique solution for the unknown interpolation coefficients. To reduce the error of this second-order approximation, the interpolation is made symmetrical by averaging the values for $u_{-1,-2}$ and $u_{1,-2}$ with those obtained using the mirror-image stencil indicated by the dotted lines in Figure 10. Interpolation formulae for the remaining coarse edge velocities $v_{-2,-1}$, $v_{-2,1}$, $w_{1,-1}$, $w_{-1,1}$ are obtained by rotation of the stencil for the u coarse edge velocities.

After all fine velocities on bisections of coarse edges (unfilled black arrows on Figure 10) have been obtained using the interpolation described above, the velocities on the edges of the fine triangle in the interior of the coarse triangle (grey arrows on Figure 10) are found by assuming that vorticity is constant over the coarse triangle, for example:

$$u_{-1,0} = \frac{1}{2}\left(U_{0,-2} + V_{-2,0} + W_{0,0}\right) - v_{-2,1} - w_{-1,1}.$$

Since the restriction operator is the average of neighbouring fine velocities and the restriction of a prolongation must be the identity (i.e. $R \circ I = \mathrm{Id}$), the wavelet coefficients on coarse edges are redundant,

$$\begin{aligned} \widetilde{u}_{-1,-2} &= -\widetilde{u}_{1,-2}, \\ \widetilde{v}_{-2,-1} &= -\widetilde{v}_{-2,1}, \\ \widetilde{w}_{1,-1} &= \widetilde{w}_{-1,1}. \end{aligned}$$

Thus, there are nine linearly independent wavelet coefficients associated with the coarse velocities $U_{0,-2}$, $V_{-2,0}$, $W_{0,0}$: $\widetilde{u}_{-1,-2}$, $\widetilde{v}_{-2,-1}$, $\widetilde{w}_{1,-1}$, $\widetilde{u}_{-1,0}$, $\widetilde{u}_{1,0}$, $\widetilde{v}_{0,-1}$, $\widetilde{v}_{0,1}$, $\widetilde{w}_{-1,-1}$, $\widetilde{w}_{1,1}$.

2.6. *Error control and adaptivity*

2.6.1. *Adaptivity*

In the preceding sections we derived a scalar wavelet transform for the pressure p_i and a vector wavelet transform for the velocity components u_e, v_e, w_e. These wavelet transforms are both based on linear polynomial interpolation (and hence are second-order accurate), but they use different stencils and implicitly define different wavelets and scaling functions. A wavelet transform of a function f is

$$\mathbf{f} = \mathbf{f}_k^0 \phi_k^0 + \sum_{j=0}^{J-1} \sum_{m \in \mathcal{M}^j} \widetilde{f}_m^j \psi_m^j, \tag{6}$$

where ϕ_k^0 are the scaling functions spanning the coarsest scale $j = 0$ and ψ_m^j are the wavelet functions spanning the difference between two successive scales $(j+1)$ and j. As described in section 2.2, the wavelet coefficient $\widetilde{f}_m^j$ measures the interpolation error at a particular location m and scale j. Adaptive bi-orthogonal wavelet methods are based on the fact that removing wavelet coefficients with a magnitude lower than a particular threshold ε_f (and the associated grid points) generates an adaptive grid, and the error of a function $f_>$ reconstructed on the adaptive grid is controlled by ε_f (Vasilyev and Bowman, 2000),

$$\begin{aligned} ||f_> - f||_\infty &\le C_1\, \varepsilon_f, & (7) \\ ||f_> - f||_\infty &\le C_2\, N^{-m/D}, & (8) \end{aligned}$$

where N is the number of grid points in the adapted grid, m is the order of interpolation and D is the dimension. Here $m = 2$ and $D = 2$ and so

$$||f_> - f||_\infty \le C_2\, N^{-1}. \tag{9}$$

Combining inequalities (7) and (9) shows that the number of active grid points N scales like $1/\varepsilon_f$. Since one order of accuracy is lost with each derivative, if the wavelet coefficients for f are filtered with a threshold ε, the error control relation for the kth derivative $D_k f$ is given by

$$||D_k f_> - D_k f||_\infty \le c_k \varepsilon_f^{1-k/m}. \tag{10}$$

An adaptive wavelet collocation method for a time-dependent PDE filters the solutions at each time step, and then adds an adjacent zone of neighbouring points in position and scale to allow for the change in the solution over one time step (Vasilyev and Bowman, 2000; Kevlahan and Vasilyev, 2005). One nearest neighbour in position is sufficient for a CFL value of one, and one neighbour in scale is sufficient for quadratic nonlinearities (which can at most reduce the scale by a factor of two in one time step). An important feature of the method is that its computational complexity is linear in the number of active grid points N.

In the following two sections we show that, since the TRiSK discretization is less than second-order accurate, the wavelet transforms for pressure and velocity should be at most second-order accurate, and we determine how to scale the separate thresholds for pressure, ε_p, and velocity, ε_u, to ensure a uniform relative error ε for both tendencies $\partial_t p$ and $\partial_t \mathbf{u}$.

2.6.2. *Stability of the adaptive method*

It is reasonable to assume that the grid adaptation strategy for the numerical solution of a time-dependent PDE depends on the order of accuracy of the discretization. In AWCMs on collocated grids, this is not an issue since differential operators are approximated using the same polynomial interpolation as used in the wavelet transform. However, in general different interpolations are used for the discretization of the dependent and independent variables. In this section, we analyze how the order of accuracy of the discretization of a PDE constrains the order of the wavelet transform used to adapt the grid.

Because this is essentially a problem of advective stability, let us consider the linear advection equation

$$\frac{\partial p}{\partial t} + \mathbf{u} \cdot \nabla p = 0. \tag{11}$$

We assume that the velocity- and length-scales are respectively U and L, the fluctuations of p are of order δp and the local grid size is h. If the polynomial interpolation used in the prolongation (i.e. prediction) operators of the wavelet transform are order m, then (7) shows that the error in p due to wavelet filtering with threshold $\varepsilon_p = \varepsilon\, \delta p$ is

$$\mathrm{Err}(p) \sim \varepsilon\, \delta p \sim h^m\, p^{(m)} \sim \delta p \left(\frac{h}{L}\right)^m. \tag{12}$$

Now, if the pressure gradient is discretized using an nth-order accurate method, the discretization error of ∇p can be estimated as

$$\mathrm{Err}(\nabla p) \sim h^n\, p^{(n+1)} \sim \frac{\delta p}{h} \left(\frac{h}{L}\right)^{n+1}. \tag{13}$$

Combining (12) and (13) relates the error in the pressure gradient due to wavelet filtering to that due to the discrete approximation of the gradient operator,

$$\varepsilon_{\nabla p} \sim \frac{\delta p}{h} \varepsilon^{(n+1)/m}.$$

Now, if we assume explicit time-stepping and demand that each time step introduces an error smaller than ε_p (so the time integration error does not cause runaway grid refinement), we find that

$$U\,\Delta t\,\frac{\delta p}{h}\varepsilon^{(n+1)/m} < \varepsilon\,\delta p,$$

or

$$\frac{U\,\Delta t}{h} < \varepsilon^{(m-n-1)/m}. \qquad (14)$$

The left-hand side of (14) is the CFL stability criterion, and thus in order for the limit on the time step imposed by the grid adaptation to be no stricter than that imposed by the time integration scheme, we require that $m \le n+1$. On the plane the TRiSK scheme is second-order accurate, however on the sphere it is only between first- and second-order accurate due to grid deformation (Ringler *et al.*, 2010). Thus we require that $m \le 2$. Choosing second-order interpolation for prediction step in the wavelet transform therefore ensures that grid adaptation is stable when combined with a time integration scheme stable for a CFL criterion less than one.

2.6.3. *Controlling the error in the tendencies $\partial_t p$ and $\partial_t \mathbf{u}$*

Here we identify suitable thresholds for the pressure and velocity coefficients ensuring a prescribed relative error in the tendencies $\partial_t p$ and $\partial_t \mathbf{u}$. Because our method uses different wavelet transforms for pressure and velocity (and these quantities exist on dual grids), and because the approximation of differential operators are not the same as the interpolation operators used in the wavelet transforms, error control of the tendency is non-trivial.

In the inviscid SWEs (1),(2), we assume that pressure fluctuations are weak, $p = c^2 + \delta p$, $\delta p \ll c^2$ (where c is the wave speed), and that the flow has characteristic velocity- and length-scales U and L. We further assume that the Burgers number $Bu = (c/fL)^2 = O(1)$. To simplify the analysis, we consider both the *inertia–gravity wave* and *geostrophic* regimes. Each regime leads to different relative scalings of the pressure and velocity thresholds to provide a uniform error bound on the errors in the tendencies $\partial_t p$ and $\partial_t \mathbf{u}$.

In the *inertia–gravity wave* regime, the relevant (fast) time-scale is $T \sim L/c$, and energy is equally divided between kinetic and potential (e.g. gravity waves),

$$\delta p \sim cU, \qquad \partial_t p \sim c^2\frac{U}{L}, \qquad \partial_t u \sim U\frac{c}{L}.$$

The equations may then be linearized to give the linear inertia–gravity wave equations

$$\begin{aligned} \frac{\partial p}{\partial t} + c^2\,\nabla\cdot u &= 0, \\ \frac{\partial \mathbf{u}}{\partial t} + f\mathbf{u}^\perp + \nabla p &= 0. \end{aligned}$$

Our goal is to ensure that filtering the pressure and velocity with thresholds ε_p and ε_u respectively produces the same relative error ε in the tendencies $\partial_t p$ and $\partial_t \mathbf{u}$, i.e. we require that

$$\begin{aligned} \varepsilon\frac{c^2 U}{L} \sim \mathrm{Err}(\partial_t p) &\sim c^2\,\mathrm{Err}(\nabla\cdot\mathbf{u}) \sim \frac{c^2}{h}\varepsilon_u, && (15) \\ \varepsilon\,\frac{cU}{L} \sim \mathrm{Err}(\partial_t \mathbf{u}) &\sim f\,\mathrm{Err}(\mathbf{u}^\perp) + \mathrm{Err}(\nabla p) \\ &\sim f\varepsilon_u + \frac{1}{h}\varepsilon_p. && (16) \end{aligned}$$

Equation (16) shows immediately that

$$\varepsilon_p \sim \varepsilon\,\frac{h}{L}\,Uc.$$

Taking the strictest of the conditions imposed by (15) and (16) shows that

$$\varepsilon_u \sim \varepsilon\,\frac{h}{L}\,U\,\mathrm{Min}\left(1, \frac{R_\mathrm{d}}{h}\right) \sim \varepsilon\,\frac{h}{L}\,U,$$

where we used the fact that $R_\mathrm{d}/h > 1$ since the Rossby radius $R_\mathrm{d} = c/f$ must be resolved. Thus, in the inertia–gravity wave regime, the relative scalings of the pressure and velocity thresholds ε_p and ε_u to ensure a relative error ε in the tendency are

$$\frac{\varepsilon_p}{cU} \sim \frac{\varepsilon_u}{U} \sim \varepsilon\frac{h}{L}. \qquad (17)$$

If $m = 2$, then (12) shows that $h/L \sim \varepsilon^{1/2}$, and finally

$$\begin{aligned} \varepsilon_p &\sim cU\,\varepsilon^{3/2}, && (18) \\ \varepsilon_u &\sim U\,\varepsilon^{3/2}. && (19) \end{aligned}$$

Filtering the pressure and velocity (and hence adapting the grid) using the above wavelet thresholds ensures that the tendencies in pressure and velocity have a relative error no larger than ε at each time step.

In the *quasi-geostrophic regime*, the Rossby number is small, $Ro = U/fL \ll 1$, the flow is close to geostrophic balance, $f\,\mathbf{u}^\perp \simeq \nabla p$, and the pressure fluctuations scale like $\delta p \sim Lf\,U$. The flow then changes on the slow time-scale $T = L/U \gg L/c$, and the magnitudes of the tendencies are estimated as $\partial_t p \sim U^2 f$, $\partial_t \mathbf{u} \sim U^2/L$. As for the inertia–gravity wave case, but using the slow time-scale and without linearizing the pressure tendency, the errors in the trends are

$$\begin{aligned} \varepsilon\,fU^2 \sim \mathrm{Err}(\partial_t p) &\sim c^2\,\mathrm{Err}(\nabla\cdot\mathbf{u}) + U\,\mathrm{Err}(\nabla p) \\ &\sim \frac{c^2}{h}\varepsilon_u + \frac{U}{h}\varepsilon_p, && (20) \\ \varepsilon\,\frac{U^2}{L} \sim \mathrm{Err}(\partial_t \mathbf{u}) &\sim f\,\mathrm{Err}(\mathbf{u}^\perp) + \mathrm{Err}(\nabla B) \\ &\sim f\varepsilon_u + \frac{1}{h}\varepsilon_p. && (21) \end{aligned}$$

The scaling of ε_p and ε_u are determined by the strictest of conditions (20) and (21). The pressure threshold is

$$\varepsilon_p \sim \varepsilon h f U\,\mathrm{Min}\,(1, Ro) \sim \varepsilon\,\frac{h}{L}\,U,$$

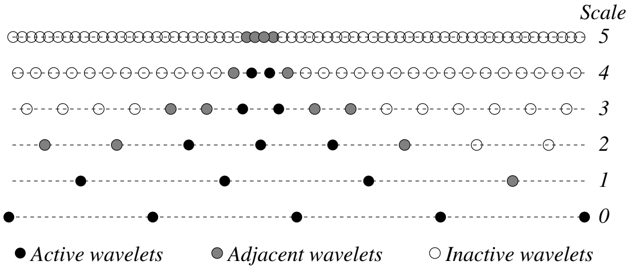


Figure 11. One-dimensional example showing adjacent wavelets in position and scale that are added to the grid of active wavelets to account for the change in the solution over one time step.

and the velocity threshold is

$$\begin{aligned}\varepsilon_u &\sim \varepsilon U \,\mathrm{Min}\left(\frac{hfU}{c^2},\, Ro\right),\\ &\sim \varepsilon Ro\, U \,\mathrm{Min}\left(Bu^{-1}\frac{h}{L},\, 1\right),\\ &\sim \varepsilon\,\frac{Ro}{Bu}\,\frac{h}{L}\,U.\end{aligned}$$

Thus, in the quasi-geostrophic regime the relative scalings of the pressure and velocity thresholds ε_p and ε_u to ensure a relative error ε in the tendency are

$$\frac{\varepsilon_p}{c^2} \sim \frac{\varepsilon_u}{fL} \sim \varepsilon\,\frac{h}{L}\,\frac{U^2}{c^2}.$$

Since $Bu = O(1)$ and $L \sim c/f$ for geostrophic flows, the final scalings for the pressure and velocity thresholds become

$$\frac{\varepsilon_p}{fLU} \sim \frac{\varepsilon_u}{U} \sim \varepsilon\frac{h}{L}Ro. \tag{22}$$

As in the inertia–gravity regime $h/L \sim \varepsilon^{1/2}$, and so the resolution-independent scalings in the quasi-geostrophic regime are

$$\begin{aligned}\varepsilon_p &\sim f\,U\,L\,Ro\,\varepsilon^{3/2}, & (23)\\ \varepsilon_u &\sim U\,Ro\,\varepsilon^{3/2}. & (24)\end{aligned}$$

If we compare the scaling (18), (19) in the inertia–gravity wave regime to the scaling (23), (24) in the quasi-geostrophic regime, we remark that the ratio $\varepsilon_p/\varepsilon_u$ is the same, and that the quasi-geostrophic scaling of $\varepsilon_p, \varepsilon_u$ is more stringent by a factor $Ro \ll 1$. Therefore when both the fast and slow modes are important, the quasi-geostrophic scaling will provide adequate error control.

We have therefore shown that wavelet filtering with the appropriately scaled thresholds ε_p and ε_u provides uniform control of the tendencies $\partial_t p$ and $\partial_t \mathbf{u}$ at each time step. As mentioned earlier, once the pressure and velocity fields have been filtered, adjacent points must be added in position and scale to allow for the translation or steepening of the solution after one time step. The active and adjacent zone wavelets are shown for a one-dimensional example in Figure 11. One further modification to the adapted grid determined must also be made. In some cases the pressure may be much smoother than the velocity (or vice versa). This could lead to a grid that is inconsistent with the finite-volume discretization of the SWEs. In other words, there may not be enough pressure (or velocity) points present locally to properly approximate the fluxes and source terms in the SWEs at the appropriate points. For example, we require accurate pressure gradient estimates at velocity points to evaluate the velocity tendency. To correct this possible inconsistency of the pressure and velocity grids, a final step is added to the grid adaptation strategy to ensure that velocity points have a sufficient number of pressure point neighbours (and vice versa).

In the following section we verify the error control estimates for wavelet filtering of p, $\mathbf{u}$, $\partial_t p$, and $\partial_t \mathbf{u}$, derived in this section and apply the complete adaptive wavelet algorithm to solve the rotating SWEs for a wave packet (in the inertia–gravity wave regime) and shallow-water turbulence (in the geostrophic regime). The TRiSK equations on the wavelet-adapted grid are integrated in time using the explicit strongly stability-preserving five-stage fourth-order Runge–Kutta method of Spiteri and Ruuth (2002)which is stable for a CFL limit ≤ 1.5.

3. Verification of error control and examples

3.1. *Inertia–gravity wave and quasi-geostrophic test cases*

As mentioned in the previous section, the rotating SWEs have two principal regimes: gravity waves (where acceleration and pressure gradient are approximately balanced) and quasi-geostrophic (where the Coriolis force is balanced by the pressure gradient). We consider test cases that focus on each of these regimes. The parameters for the gravity wave and quasi-geostrophic cases are shown in Table 1.

The initial condition for the inertia–gravity wave case is a wave packet with wavenumber k and frequency ω. The packet has a Gaussian envelope and contains approximately four wavelengths. The wave packet initial condition is

$$\begin{aligned}u(x,y) &= U\cos(kx-\omega t)\,\mathrm{e}^{-r^2/L^2},\\ v(x,y) &= \frac{Uf}{\omega}\sin(kx-\omega t)\,\mathrm{e}^{-r^2/L^2},\\ P(x,y) &= c^2\left(1+\frac{Uk}{\omega}\cos(kx-\omega t)\right)\mathrm{e}^{-r^2/L^2},\end{aligned}$$

where $f=1$, $c=1$, $k=4\pi$, $\omega = k \simeq \left(f^2+k^2c^2\right)^{1/2}$, and $U=0.1$. These choices satisfy the requirement that $kU/\omega = 0.1 \ll 1$ and ensure that perturbations from rest $P=c^2$, $\mathbf{u}=0$ are small. We consider both the linear and nonlinear inertia–gravity wave cases, in order to separate the effect of advection from the effect of generation of small scales, which is only present in the nonlinear case. The physical domain is a lozenge $16L$ on each side and the coarsest scale has 32^2 grid points. In the linear case the number of levels of resolution is controlled only by the tolerance ε, while in the nonlinear case the maximum number of levels is set to six, corresponding to a maximum resolution 2048^2, because the nonlinear case quickly develops a shock in pressure which would require an infinite number of levels to resolve. Figure 12 shows the gravity wave initial condition and the associated adapted grid determined by a relative tendency tolerance $\varepsilon \simeq 0.05$ (corresponding to pressure and velocity tolerances $\varepsilon_P = \varepsilon_u = 1.1\times10^{-3}$). The gravity wave simulations include a Laplacian viscous term with a small kinematic viscosity $\nu = 10^{-4}$ in the velocity equations.

Table 1. Parameters for gravity wave and quasi-geostrophic decaying turbulence test cases.

Case	c	f	U	L	T	Ro	Bu	Rd	Fr
Gravity wave	1	1	0.1	1	10	0.1	1	1	0.1
Quasi-geostrophic	10	10	1	1	1	0.1	1	1	0.1

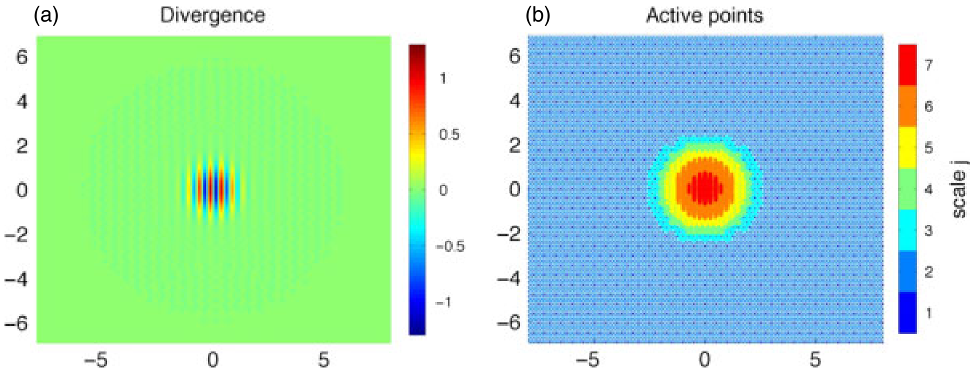


Figure 12. Initial conditions for the inertia−gravity wave case. The maximum allowable resolution is 2048^2 and the period cell has sides of length 16. (a) shows the velocity divergence and (b) the initial adapted grid (pressure and velocity wavelet points) used for the simulations. The tendency threshold is set to $\varepsilon = 0.05$, corresponding to pressure and velocity thresholds $\varepsilon_P = \varepsilon_u = 1.118\times10^{-3}$. This figure is available in colour online at wileyonlinelibrary.com/journal/qj

The computed solutions should remain well-localized for all times and are a test of how well the adaptive wavelet approach can track the advection of an isolated solution with small-scale internal structure in the gravity wave regime.

The second test case is more challenging: decaying rotating shallow-water turbulence in the quasi-geostrophic regime. This case is challenging because the flow is statistically homogeneous and isotropic at intermediate times (when it is strongly turbulent) and the grid adaptation must track the development of small scales via the enstrophy cascade, and the subsequent coarsening as the nonlinearity is depleted at long times. It is not obvious *a priori* that an adaptive method is advantageous in this case. The initial condition is an array of nine vortex pairs, each of which is generated by the perturbed pressure field

$$P(x, y) = c^2 \left(1 + \frac{2l}{L}\,\mathrm{e}^{-r^2/L^2}\right)$$

with $c = 10$. The velocity is then found from the pressure by assuming geostrophic balance of the Coriolis force and pressure gradient. Here $L = 1$ and $l = 0.15$ to ensure that the rms velocity $u = 1$ and the initial eddy turnover time $L/u = 1$. The corresponding Taylor (or friction) scale $\lambda = \sqrt{E_K/\Omega} = 0.4$, where $E_K = 0.5 \int |\mathbf{u}|^2\, \mathrm{d}A$ is the kinetic energy and $\Omega(t) = 0.5 \int \omega^2\, \mathrm{d}A$ is the enstrophy. It is important to note that essentially all energy and enstrophy dissipation is due to the wavelet filtering ($\nu = 0$), which smooths sufficiently weak velocity and pressure gradients at all scales (i.e. gradients are retained at the smallest scales if they are sufficiently intense). As we will see below, this flow develops intense turbulence with an effective Taylor-scale Reynolds number $Re_\lambda = u\lambda/\nu_{\mathrm{eff}}(\varepsilon) \approx 300$. Figure 13 shows the potential vorticity and active points for the turbulence initial condition. The initial grid is not strongly compressed because the initial condition is relatively homogeneous. The physical domain is a lozenge $9L$ on each side and thus contains 22.5 Taylor scales. The coarsest scale has 2×2 grid points and up to ten levels of refinement are permitted for a maximum of eleven dyadic scales, or a maximum resolution of 2048^2. Note that only eight out of a maximum of eleven levels are used to resolve the initial condition, although all eleven are needed at later times.

3.2. Error control of pressure, velocity and tendency

Before considering the adaptive simulation of the rotating SWEs, we first verify that our method does indeed adaptively control the errors in the pressure and velocity for the turbulence initial condition. For this we evaluate the initial fields on a fine grid of resolution 8192^2, compute the wavelet coefficients, discard those less than a threshold ε, and compute the filtered values (scaling coefficients) on the fine grid by inverse discrete wavelet transform. The difference with the unfiltered initial condition, measured in various norms, is plotted as a function of ε in Figure 14(a, d). The error scales linearly with ε, confirming that separate nonlinear wavelet filtering of the pressure and velocity wavelet coefficients controls the error. Furthermore the number $N(\varepsilon)$ of active wavelet coefficients scales as expected for second-order interpolation, $N(\varepsilon) \propto \varepsilon^{-1}$ (Figure 14(b, e)). Note that control of pressure and velocity errors does not depend on the whether the test field is in the gravity wave or geostrophic regime.

Having demonstrated the error control properties of the scalar- and vector-valued wavelet transforms, we still need to verify that we can control the errors in the tendencies $\partial_t p$ and $\partial_t \mathbf{u}$ by filtering the pressure and velocity using the scalings derived in section 2.6. The inertia−gravity wave and geostrophic regimes have different scalings and so we consider both the gravity wave and turbulence initial conditions as test cases. Now a reference value of the tendencies is computed on the fine grid using the TRiSK operators and the fine-grid initial condition. The tendencies of the active wavelet coefficients are computed using the adaptive Algorithm 2, then inverse-wavelet transformed to yield tendencies of the pressure and velocity on the fine grid. Differences with the reference tendencies measured in

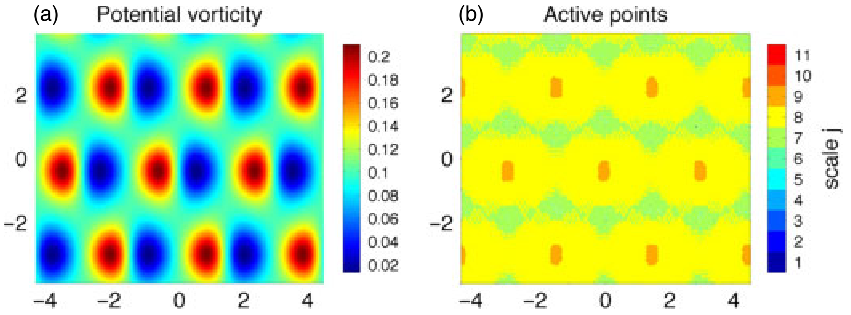


Figure 13. Initial conditions for turbulence case. The maximum resolution is 2048^2 and the periodic cell has sides of length 9 (22.5 Taylor scales λ). (a) shows the potential vorticity and (b) the initial adapted grid (pressure and velocity points) used for the turbulence simulation. The tendency threshold is set to $\varepsilon = 0.15$, which corresponds to pressure and velocity thresholds $\varepsilon_p = \varepsilon_u = 5.8095 \times 10^{-2}$. This figure is available in colour online at wileyonlinelibrary.com/journal/qj

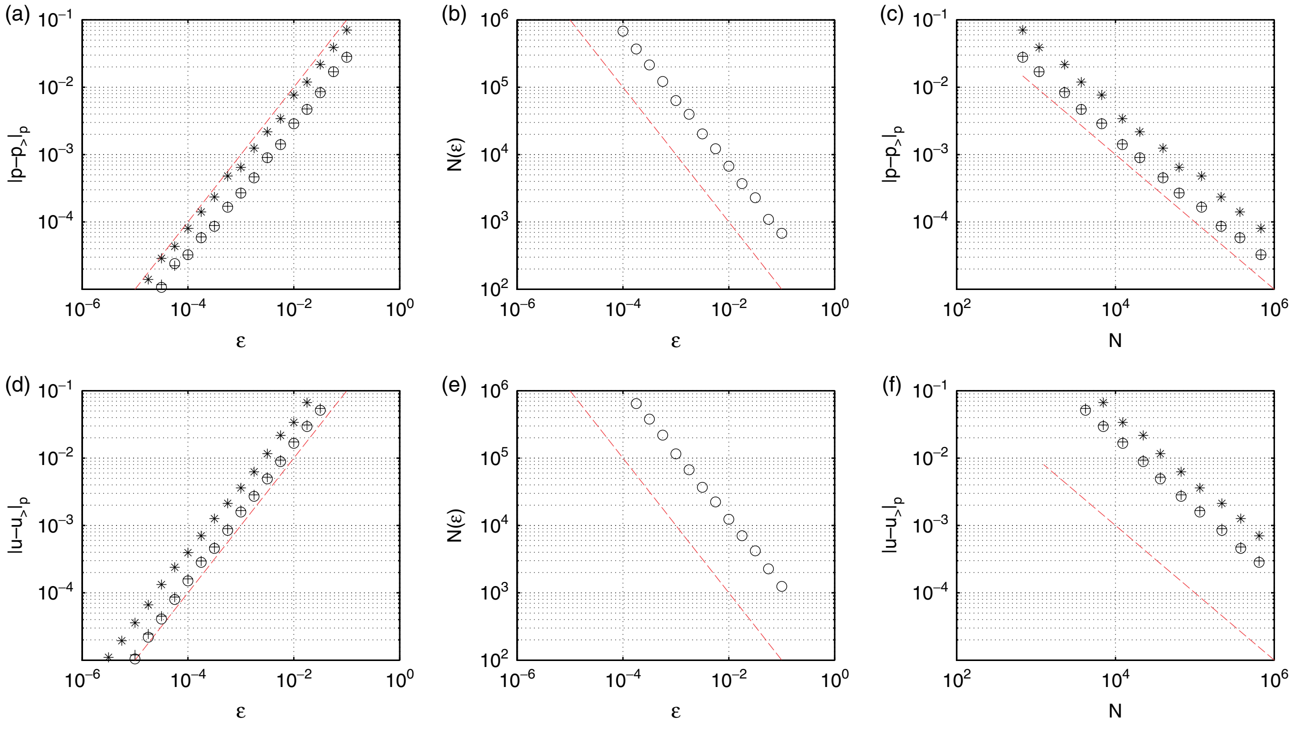


Figure 14. Control of error in (a, b, c) pressure and (d, e, f) velocity. ε is the threshold, and $N(\varepsilon)$ is the number of active pressure or velocity wavelet coefficients. ○ denotes the L^1 error norm, + the L^2 error norm and ∗ the L^∞ error norm. The straight lines give the theoretical scalings for second-order accurate interpolation in two dimensions (i.e. error ∝ N^{-1}, and $N \propto \varepsilon^{-1}$). Note that the most important property is that the error is proportional to (and hence controllable by) the threshold ε. This figure is available in colour online at wileyonlinelibrary.com/journal/qj

various norms are plotted in Figure 15 as a function of the threshold ε determining the thresholds ε_p and ε_u. Figure 15 confirms that, if the pressure and velocity thresholds are scaled as in (17), then the relative errors in both the pressure and velocity tendencies are of the same order and are both controlled by the single tendency threshold ε. (This test uses a homogeneous version of the gravity wave initial condition, rather than the wave packet version, to avoid the weak potential enstrophy and non-periodic boundary effects associated with the wave packet envelope.) Similarly, in the quasi-geostrophic case Figure 16 confirms that, if the pressure and velocity thresholds are scaled as in (22), then the errors in both the pressure and velocity tendencies are of the same order and both are controlled by the single tendency threshold ε. Both Figures 15 and 16 show that the error drops abruptly at a critical small value of ε. This value corresponds to the point at which the maximum allowed number of levels has been reached and thus ε no longer determines the maximum scale (which should be larger). Since the error is compared with the tendency on the non-adapted (full) grid, further reduction in ε simply fills out the grid and leads to an artificially low error. (A maximum level is required since these convergence studies require calculating the tendency on the equivalent non-adapted grid, and available computational resources limit this grid to 8192^2.)

Finally, we perform a convergence study of the linear gravity wave case to show that the tendency tolerance ε does indeed control the accumulated time integration error for p and $\mathbf{u}$. The simulation parameters are as in Table 1, the computational domain is $8L$ on each side, the kinematic viscosity is set to zero (so that any dissipation is due entirely to the adaptivity) and the equations are integrated until time $T = 8$ (i.e. one orbit of the periodic domain). The errors of the adaptive simulations are calculated with respect to an equivalent single-scale non-adaptive inviscid TRiSK

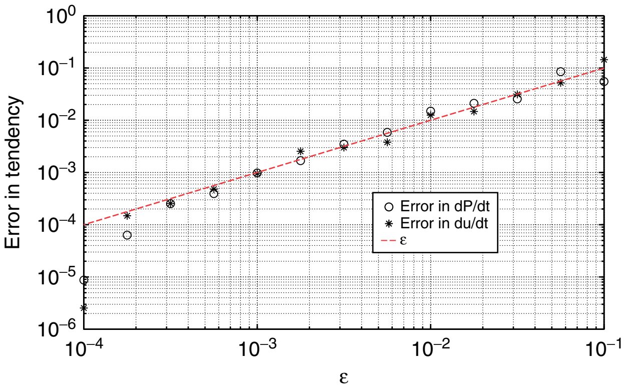


Figure 15. Control of tendency error in the inertia–gravity wave regime. The thresholds for pressure and velocity are respectively $\varepsilon_P = c\,U\,\varepsilon^{3/2}$ and $\varepsilon_u = U\,\varepsilon^{3/2}$, to ensure that the error in the tendencies $\partial_t p$ and $\partial_t \mathbf{u}$ are $O(\varepsilon)$. This figure is available in colour online at wileyonlinelibrary.com/journal/qj

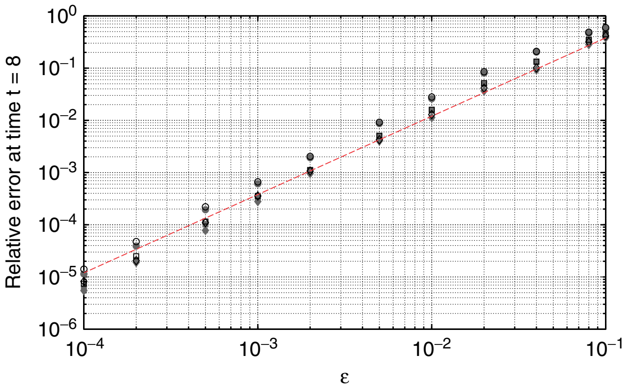


Figure 17. Accumulated time integration errors at $t = 8$ for the linear gravity wave case as a function of tolerance ε: L^1 error (○), L^2 error (□), L^∞ error (◇). Errors in p are indicated by open symbols and errors in $\mathbf{u}$ by filled symbols. The straight line shows a $\varepsilon^{3/2}$ scaling. This figure is available in colour online at wileyonlinelibrary.com/journal/qj

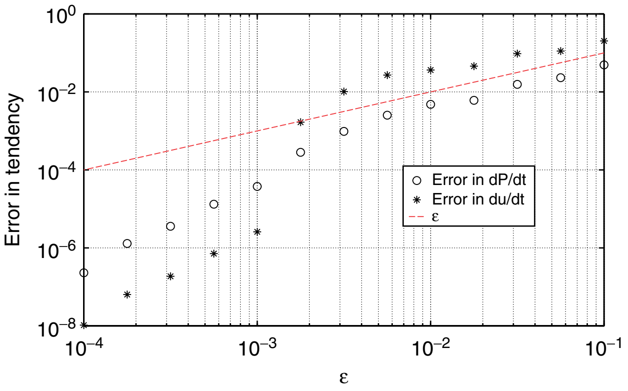


Figure 16. Control of the tendency error in quasi-geostrophic regime. The thresholds for pressure and velocity are respectively $\varepsilon_P = Ro\,U\,f\,L\,\varepsilon^{3/2}$ and $\varepsilon_u = Ro\,U\,L\,\varepsilon^{3/2}$, to ensure that the relative error in the tendencies $\partial_t p$ and $\partial_t \mathbf{u}$ is $O(\varepsilon)$. The error decreases more rapidly for $\varepsilon \leq 10^{-3}$ because the maximum allowable refinement level $j = 12$ has been reached. This figure is available in colour online at wileyonlinelibrary.com/journal/qj

simulation at the maximum allowed resolution of 1024^2. (We use a smaller resolution and domain size due to the computational expense of the non-adaptive simulation.) Figure 17 shows that the accumulated relative error scales as $\varepsilon^{3/2}$ in the L_1, L_2, and L_∞ error norms, which means that the tendency error is proportional to the pressure and velocity thresholds ε_p and ε_u in this case. Most importantly, the scalings for tolerances ε_p and ε_u derived in section 2.6.3 have ensured the same relative errors for pressure and velocity. Note that the error scaling for the highly localized wavepacket case considered here is better than the estimated upper-bound scaling of ε found for the homogeneous gravity wave case (Figure 15). As for all time-marching numerical schemes, the accumulated error grows linearly in time.

3.3. *Simulations of inertia–gravity waves*

We now perform numerical simulations using the adaptive evaluation of tendencies. The velocity divergence field and the grid of active wavelets for the linear and nonlinear inertia–gravity wave cases after one orbit of the periodic domain are shown in Figures 18 and 19 respectively. In both cases the adaptive grid has tracked the advection of the wave packet, retaining a significant compression ratio. In the linear case the wave packet does not steepen, although it spreads in the y direction. Both the spreading and the lack of generation of small scales are reflected in the active grid. The isolated grid points in the wake of the wavepacket in Figure 18 are the remains of the very weak wave train left behind the wavepacket (which has been largely dissipated). In contrast, as shown in Figure 19, nonlinearity steepens the waves making up the wavepacket, which generates a sequence of very strong gradients. The active grid has successfully refined to track the local steepening and spreading of the wavepacket.

Figure 20 displays the evolution with time of several measures of the efficiency of the adaptive method: number $N(t)$ of active points, cpu cost per active point, compression ratio and cpu cost per time step. The steepening of the nonlinear inertia–gravity wave is reflected in the development of the grid. The compression ratio decreases as the wavefront broadens and steepens, however the computational cost per grid point is roughly constant. This shows that the computational cost is approximately proportional to the number of active points (i.e. linear computational complexity), as needed for an effective dynamically adaptive method. In contrast, the number of active grid points for linear inertia–gravity wave is approximately constant (results not shown). This is expected since the linear wave packet is simply advected and spread out, which should conserve the total number of active points.

3.4. *Rotating shallow-water turbulence*

This case tests the ability of the adaptive wavelet method to provide high compression and accurate results in the quasi-geostrophic regime. We are particularly concerned with the effect of the choice of tendency tolerance ε on the turbulence simulations. Although we consider the case of freely decaying turbulence to avoid issues with the precise nature of the forcing, the initial condition is sufficiently energetic that the flow develops a strongly turbulent quasi-stationary regime at $t \approx 25$ that lasts for about 20 eddy turnover times. Turbulence intensity later decreases due to merging of like-sign vortices until at about $t = 100$ only one positive and one negative vortex remain. This scenario is qualitatively the same as the dynamics of two-dimensional incompressible Navier–Stokes turbulence

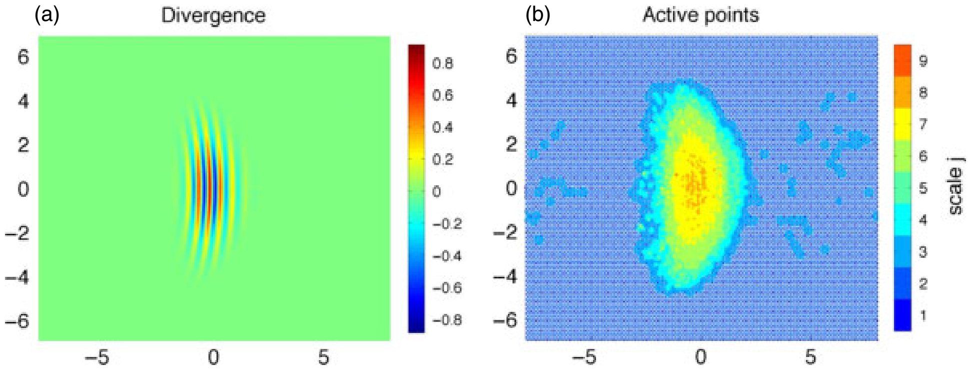


Figure 18. (a) Velocity divergence and (b) active wavelets for the linear inertia–gravity wave at $t = 16$ (one orbit of periodic domain). Note that the grid has tracked the advection and spreading of the wavefront. This figure is available in colour online at wileyonlinelibrary.com/journal/qj

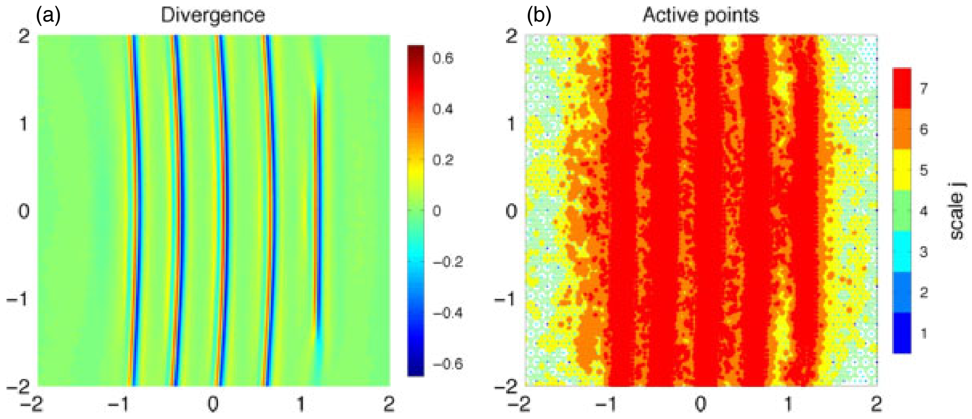


Figure 19. As Figure 18, but for the nonlinear wave and zoomed to show the small-scale structure. The grid has tracked the advection and steepening of the nonlinear wave. This figure is available in colour online at wileyonlinelibrary.com/journal/qj

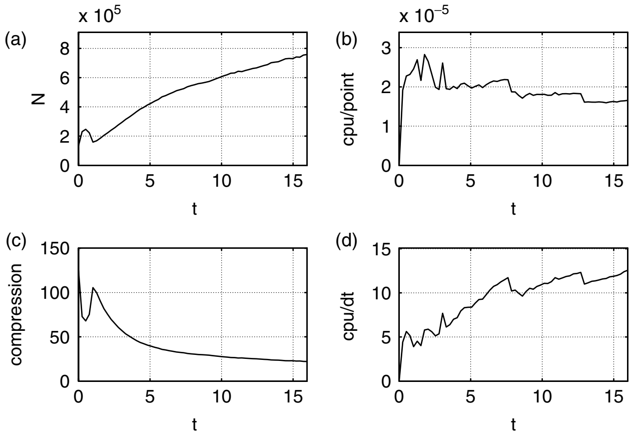


Figure 20. Compression properties of nonlinear inertia–gravity wave computation. The compression decreases as the wave packet steepens and broadens, but the cpu time per active point is approximately constant.

(Boffetta and Ecke, 2012). Because the energy dissipation rate determines the effective Reynolds number, in Figure 21 we compare two simulations with tolerances $\varepsilon = 0.15$ and $\varepsilon = 0.30$ to investigate how changing the tolerance affects the turbulence. Figure 22 shows how the grid progressively coarsens as the potential vorticity decays.

Figure 23 compares the potential vorticity and active wavelets points for the high and low tolerance simulations just after the onset of turbulence. (Note that the turbulence takes 20–25 eddy turnover times to develop because the vortex array initial condition is geostrophic, and thus meta-stable. This instability takes slightly longer to develop in the $\varepsilon = 0.15$ case.) Both simulations are qualitatively similar: as in two-dimensional incompressible Navier–Stokes turbulence, the flow is dominated by large-scale coherent vortices and small-scale vorticity filaments. However, in the $\varepsilon = 0.30$ case the filaments are slightly less intense since this flow is a bit more dissipative. The active wavelet points shown in Figure 21 indicate that the filaments in the $\varepsilon = 0.15$ are resolved to scales about twice as small as in the $\varepsilon = 0.30$ case ($j = 11$ compared to $j = 10$). However, it is important to recall that, unlike a large-eddy simulation (LES), the adaptive wavelet simulation does resolve the same proportion of total energy for all scales, i.e. all scales are available if they are sufficiently energetic.

A major question of this section is whether the adaptive wavelet method can still achieve significant compression ratios for statistically homogeneous and isotropic flow with a dense distribution of coherent structures. Figure 21 has already revealed that even in the strongly turbulent regime there are significant sub-regions of the flow that need be resolved only coarsely. In fact, the effective resolution varies from 2048^2 ($j = 11$) in the most intense vorticity filaments to an effective resolution of only 32^2 ($j = 5$) or 64^2 ($j = 6$) in the regions outside the coherent vortices and filaments. The coherent vortices are resolved at effective resolutions of 256^2 ($j = 8$) or 512^2 ($j = 9$). The properties of the grid compression are summarized in Figure 23, which shows that the compression ratio is about 50 times when the turbulence is most active (between $t = 25$ and $t = 40$) and then rises monotonically to about 280 times during the final period of

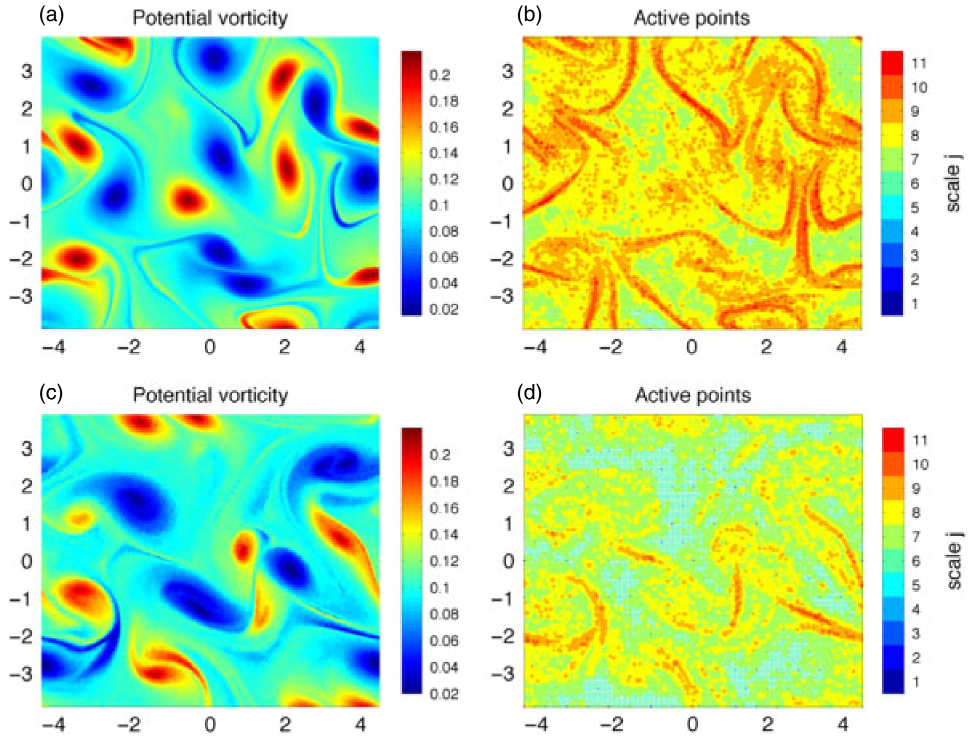


Figure 21. (a, c) Potential vorticity and (b, d) adapted grids for decaying quasi-geostrophic turbulence just after the turbulence is fully developed: (a, b) at $t = 25.03$ for tolerance $\varepsilon = 0.15$, and (c, d) at $t = 21.80$ for tolerance $\varepsilon = 0.30$. Two and three levels of refinement have been added, respectively, compared to the initial condition. There are significant coarse regions between the coherent structures and filaments. This figure is available in colour online at wileyonlinelibrary.com/journal/qj

decay. As in the inertia–gravity wave case, the computational cost per active point is approximately constant (i.e. linear computational complexity on the adapted grid).

An important characteristic of turbulence is that its energy spectrum should follow a power law, typically about $E(k) \propto k^{-3.3}$ for decaying two-dimensional incompressible Navier–Stokes turbulence (Boffetta and Ecke, 2012). There are no theoretical predictions for rotating shallow-water turbulence, but one expects power-law scaling because of the similar structure of the equations. In order to calculate the Fourier energy spectra, the velocity is first interpolated onto the full $j = 11$ (2048^2) grid before calculating the vorticity and velocity divergence and transforming to Fourier space using the appropriate lozenge-shaped periodic unit cell. Note that, because the periodic unit cell is a lozenge, the maximum available isotropic wavenumber is smaller than for the usual square periodic unit cell.

Figure 24 compares the isotropic energy spectra for the rotational and divergent parts of the flows at same times as the fields shown in Figure 23. One of the principal advantages of wavelet-based adaptive methods compared to LES is that wavelet methods capture the full range of active length-scales, even at high tolerances ε. This is clear from Figure 23, where the vortical energy spectra of both simulations are very similar over the full range of length-scales. Unsurprisingly, the vortical energy spectra scale like k^{-4} over most length-scales due to the predominance of vorticity filaments. The main difference between the two simulations is in the divergent energy spectrum: the $\varepsilon = 0.30$ simulation actually has significantly more rotational energy, except at small scales. This may be because the velocity divergence in the higher-tolerance simulation is noisier at intermediate and small scales. Both simulations show an equipartition of energy between vortical and divergent modes at small scales, possibly with a short range of k^{-3} scaling for both modes. In brief, although the higher tolerance simulation is more dissipative, it still captures the correct scaling of the energy spectra over the full range of scales.

As mentioned above, energy and potential enstrophy dissipation are due to the wavelet filtering process. Figure 25 shows the decay of (available) energy $E(t)$ and (available) potential enstrophy $Z(t)$:

$$\begin{aligned} E(t) &= \frac{1}{2}\int p(p + |\mathbf{u}|^2)\,\mathrm{d}A - \frac{c^4}{2}A, \\ Z(t) &= \frac{1}{2}\int \frac{(f+\omega)^2}{p}\mathrm{d}A - \frac{f^2}{2c^2}A, \end{aligned}$$

where $\omega = \operatorname{curl}\mathbf{u}$, and $A = 81 \times \sqrt{3}/2$ is the area of periodic domain. Notice that we subtract the values corresponding to the state of rest to emphasize the non-trivial variations. There is an initial rapid decrease in energy for $0 \le t \le 20$ as the instability develops, followed by a period of slower decay associated with the decaying two-dimensional turbulence. In contrast, the potential enstrophy decreases most rapidly just after the turbulence forms due to the formation and dissipation of intense small-scale vorticity filaments. The higher tolerance $\varepsilon = 0.30$ simulation destabilizes to turbulence earlier ($t = 18$ compared to $t = 25$) and, as expected, is more dissipative once the turbulence has developed. As explained below, the decay rate of energy and enstrophy is relatively insensitive to the tolerance level ε, except during the most turbulent period.

Although it is clear that the $\varepsilon = 0.15$ simulation is less dissipative and more turbulent than the $\varepsilon = 0.30$ simulation, it is useful to quantify this difference in terms of an effective Reynolds number. To define an effective Reynolds number,

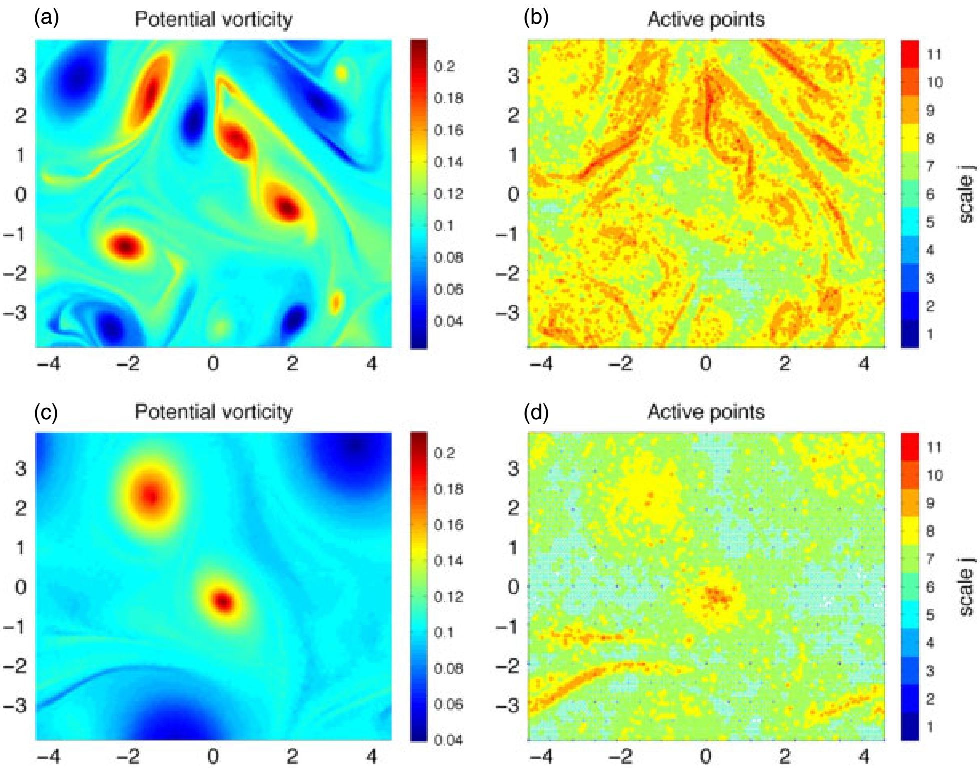


Figure 22. (a, c) Potential vorticity and (b, d) adapted grids for decaying quasi-geostrophic turbulence computed with tolerance $\varepsilon = 0.15$ at (a, b) an intermediate time (t =39.8) and (c, d) a late (t =74.8) time. This figure is available in colour online at wileyonlinelibrary.com/journal/qj

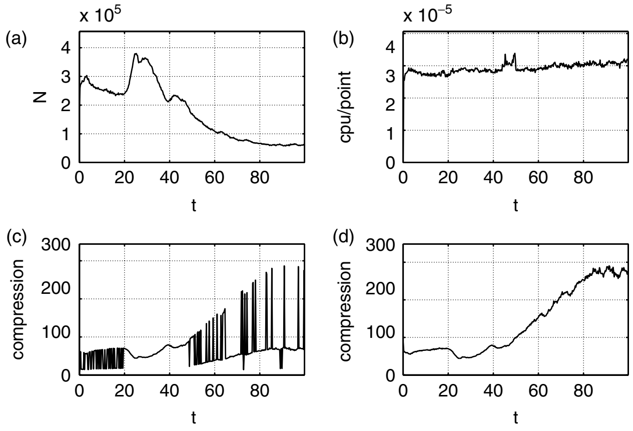


Figure 23. Adaptive properties of the turbulence simulation with tolerance $\varepsilon = 0.15$. (a) shows the total number of active points which is greatest when the turbulence is most intense (around $t = 25$). (b) shows that the computational cost per grid point is roughly constant, independent of the number of active points and the compression. (c) shows the compression based on the maximum active scale at any given time (and hence jumps when the maximum active scale changes), and (d) shows the compression rate measured based on the maximum allowable resolution (i.e. 2048^2).

we need to estimate an effective viscosity ν_{eff} due to the wavelet filtering. For quasi-geostrophic flows the equation for the total kinetic energy is approximately

$$\frac{\mathrm{d}E(t)}{\mathrm{d}t} = -2\nu\, c^2\, \Omega(t).$$

Since we know $E(t)$ and $\Omega(t)$, we can calculate an effective viscosity due to the wavelet filtering,

$$\nu_{\text{eff}}(t) = \frac{-1}{2\,c^2\,\Omega(t)}\,\frac{\mathrm{d}E(t)}{\mathrm{d}t}, \tag{25}$$

and hence an effective time-dependent Reynolds number

$$Re_\lambda(t) = \frac{u(t)\,\lambda(t)}{\nu_{\text{eff}}(t)}. \tag{26}$$

This dynamic Reynolds number measures the turbulence intensity at any time and would be roughly constant for a statistically stationary flow. Tests of (25) on non-adaptive simulations suggest that the estimate of the effective viscosity is accurate to within less than one percent for the resolutions considered here.

Figure 26 shows the energy decay rate, Taylor scale, effective viscosity and effective Reynolds numbers as a function of time for the two turbulence simulations. Because we use a bi-orthogonal basis and the exact (i.e. non-adaptive) dissipation is zero, the effective viscosity should scale with the tolerance ε roughly like $\nu_{\text{eff}} \propto \mathrm{d}E/\mathrm{d}t = 0 + O(\varepsilon)$, and $Re_\lambda = O(\varepsilon^{-1})$. The results confirm that the Reynolds number is roughly proportional to $1/\varepsilon$ once the turbulence has developed, $25 \le t \le 100$, with $\overline{Re}_\lambda(\varepsilon = 0.15) \approx 387 \pm 62$ and $\overline{Re}_\lambda(\varepsilon = 0.30) = 199 \pm 58$. For comparison, the two-dimensional soap film turbulence experiments of Kellay and Gordburg (2002) had a Taylor Reynolds number of about 100.

The relation between energy decay rate and effective viscosity and Reynolds number is not straightforward. During the initial laminar period, before the instability has developed, the effective viscosity and Reynolds numbers of the two simulations are similar. However, once the instability has developed, the expected scaling with ε is maintained throughout both the turbulent, $25 \le t \le 50$, and final decay, $t > 50$, periods. Interestingly, Figure 26 shows that the decay rate of total energy, $\mathrm{d}E/\mathrm{d}t$, is approximately proportional to ε only during the most turbulent phase. During the initial laminar and final decay periods, the energy decay

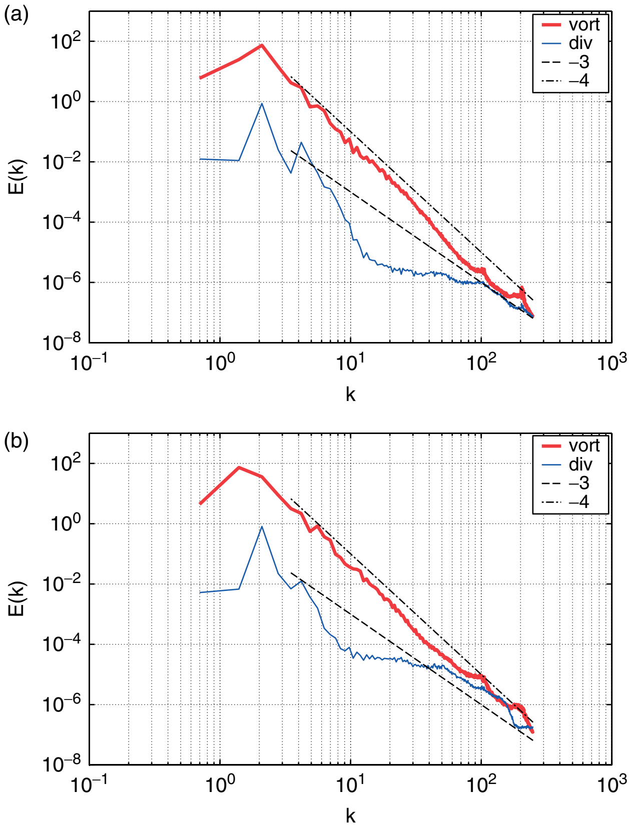


Figure 24. Vertical and divergent parts of the energy spectrum for the decaying shallow-water turbulence simulations with tolerances (a) $\varepsilon = 0.15$ at $t = 25.03$, and (b) $\varepsilon = 0.30$ at $t = 21.80$. Note that the vortical part of the two spectra are very similar, while the divergent part of the $\varepsilon = 0.15$ spectrum has less energy at intermediate and small scales. This figure is available in colour online at wileyonlinelibrary.com/journal/qj

rate is almost equal for both the $\varepsilon = 0.15$ and $\varepsilon = 0.30$ simulations, presumably because the flow is much smoother outside the turbulent regime and thus less dissipative. The energy decay rate is actually relatively insensitive to ε, probably because the wavelet coefficients in the adjacent zone are not reset to 0 at each time step. Nevertheless, because the equations for the effective viscosity (25) and Reynolds number (26) involve other time-dependent quantities Ω, u, and λ, the effective viscosity and Reynolds numbers do scale approximately as expected with ε once the unstable dynamics have developed.

The effective Reynolds number defined here allows one to accurately estimate the time-dependent Reynolds number of any turbulence simulation involving non-physical dissipation, such as AMR, hyper-dissipation and numerically dissipative schemes (e.g. upwind schemes). It could also be useful even when a standard Laplacian dissipation term is used since it measures the actual effective Reynolds number depending on the dynamics of the flow at a particular time. As mentioned earlier, one of the main advantages of the wavelet approach is that, because it filters weak gradients and not small scales, it models a higher Reynolds number for a given number of grid points and captures the full range of active length-scales. For fixed initial conditions, we found that the Taylor Reynolds number scales roughly like $1/\varepsilon$, but this scaling might change for higher Reynolds numbers as the active small scales become increasingly concentrated.

3.5. *Computational cost compared with a non-adaptive method*

In the previous section we verified that the computational cost of the adaptive wavelet solver for the two-dimensional rotating SWEs is proportional to the number of points in the adaptive grid, i.e. the cpu time per active grid point is approximately independent of the number of grid points and number of refinement levels. Combining the error control results from section 2.6 with the Reynolds number estimates from section 3.4 suggests that the number of active grid points (and hence computational cost) should scale like $N \propto Re_\lambda^{3/2}$, consistent with the estimate from the adaptive wavelet simulations of Alam *et al.* (2007). Boffetta and Musacchio (2010)'s high-resolution pseudo-spectral direct numerical simulations of forced two-dimensional Navier–Stokes turbulence suggest that the ratio of the Taylor scale to the viscous scales increases with Reynolds number like $\lambda/l_\nu \propto Re_\lambda^2$, and thus the number of grid points required for a non-adaptive simulation should increase like $N \propto Re_\lambda^4$. Although no similar results exist for rotating shallow-water turbulence, the structure of the two flows are very similar and thus it is reasonable to expect that adaptivity will drastically reduce the computational cost in this case as well. As found by Alam *et al.* (2007), the compression ratio should increase significantly with increasing Reynolds number as the flow becomes more intermittent.

In order to estimate the overhead of the adaptive wavelet method, we have compared the actual computational times of adaptive and non-adaptive non-multiscale (i.e. single-scale, uniform-grid) simulations of the same problem (vortex merging) using the TRiSK discretization of the SWEs. Both methods were implemented in `matlab` and run serially on the same machine. The actual difference in cpu time depends on the number of refinement levels and maximum resolution, but for a maximum resolution of 1024^2, the adaptive wavelet code is about 23 times slower *per active grid point* for eight levels of refinement and nine times slower for three levels of refinement. It turns out that most of the increased overhead is due to the extensive use of sparse matrices in the adaptive code to ensure that memory use scales with the number of active grid points. Apparently the overhead due to indirect addressing in three-dimensional models using an unstructured horizontal grid experience can be made small by an adequate implementation (MacDonald *et al.*, 2011; Skamarock *et al.*, 2012). If indirect addressing is indeed cheap in a realistic setting, the only remaining overhead lies in the additional interpolation/restriction operations required by the adaptive calculation. In order to measure this overhead, we force the non-adaptive code to use sparse matrices as well and compare it to the adaptive code. In that case the adaptive code is only nine and three times slower respectively per active grid point. Although the cost per active grid point is higher, the high compression ratios means that the adaptive code is nevertheless seven times faster with eight levels of refinement and 18 times faster with three levels of refinement than the non-sparse non-adaptive code. When compared to the sparse non-adaptive code, the adaptive code is 18 and 50 times faster for eight and three levels of refinement respectively. Because the sparse arithmetic and numerical calculations are slow in `matlab`, we expect that the actual overhead in a Fortran 95 implementation would be about 2–3 times.

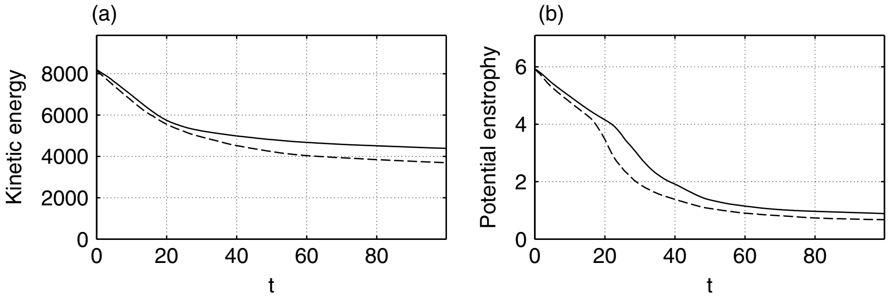


Figure 25. (a) Total energy and (b) potential enstrophy for the turbulence simulation with tolerances $\varepsilon = 0.15$ (solid) and $\varepsilon = 0.30$ (dashed).

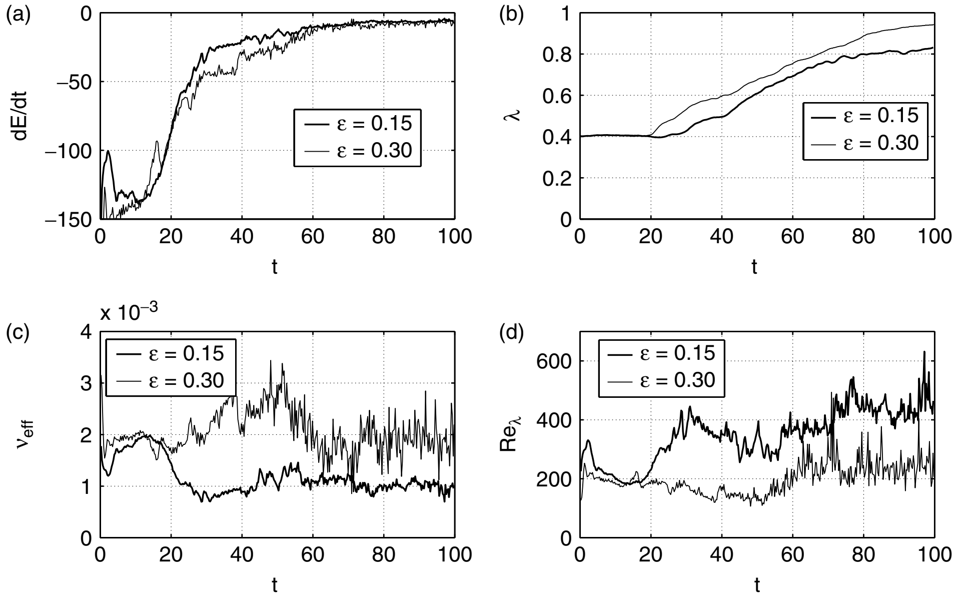


Figure 26. Dynamical quantities compared for turbulence simulations with tolerances $\varepsilon = 0.3$ and $\varepsilon = 0.15$: (a) energy decay dE/dt, (b) Taylor scale λ, (c) the effective viscosity $\nu_{\rm eff}$, and (d) the effective Taylor Reynolds number Re_λ. The effective Reynolds number appears to be roughly inversely proportional to the tolerance after the initial laminar period.

Besides reducing the cpu time for intermittent and inhomogeneous flows, the adaptive wavelet method also decreases memory use. Without such an adaptive method, memory limitations are often the main constraint in limiting the size of feasible problems. As stated in the introduction, the adaptive wavelet method is best seen as a ‘capability’ method: it allows the computation of problems that would not be feasible using non-adaptive method because of both cpu and memory constraints. If the problem is small (or the solution is homogeneous), it is better to use traditional non-adaptive methods.

4. Conclusions and perspectives

This article has presented a novel adaptive wavelet implementation of the TRiSK method (Ringler *et al.*, 2010) for the rotating SWEs on a multiscale structured staggered C-grid. The TRiSK scheme was designed from the beginning to accommodate unstructured, variable-resolution static grids (Ringler *et al.*, 2011). The approach followed here to make the TRiSK scheme dynamically adaptive was not to deform the grid to provide the high resolution where desired, but to work with a fixed hierarchy of structured grids and neglect degrees of freedom where and when they were deemed superfluous. It would be interesting in the future to compare a static variant of our dynamically adaptive method to the TRiSK scheme on a variable-resolution grid, especially with respect to well-known issues of statically refined grids like the spurious reflection/refraction of marginally resolved waves (Long and Thuburn, 2011).

The approach proposed here has two main purposes. First, it serves as the first step towards developing a fully adaptive discretely conservative dynamical core for weather and climate models on the sphere. Secondly, it initiates a new class of adaptive wavelet solvers for conservative PDEs on non-Cartesian staggered grids and introduces a non-separable vector-valued wavelet transform. We have verified that the algorithm provides effective error control for numerical simulations of the rotating SWEs on a flat two-dimensional periodic domain in both the inertia–gravity wave and quasi-geostrophic regimes. In addition, we have shown that the method has linear computational complexity in the number of active grid points, gives high grid compression ratios and is accurate even for relatively large tolerances. We made a special effort to understand the effect of nonlinear wavelet filtering on energy dissipation and on the turbulence dynamics, and proposed an effective Taylor Reynolds number as a way of characterizing the turbulence.

As mentioned in the introduction, the wavelet method developed here shares some features of the AMR schemes popular in engineering and astrophysics. The main difference is that in the AMR approach the numerical discretization of the PDE is modified locally to be consistent with the nested grid structure providing the adaptivity (Berger and Leveque, 1998). In contrast, the wavelet approach isolates the multiscale grid structure and dynamical adaptivity from the numerical discretization of the PDE. The wavelet interpolation and restriction operators provide the necessary input values to the underlying numerical scheme, so that it does not 'know' that it operates on an adapted grid. Conversely the AMR algorithm does not use interpolation and restriction operators (with appropriate commutation properties) to calculate the tendencies and other discretized quantities. Because of this, AMR methods must treat the boundaries between refined patches explicitly in order to avoid spurious errors. There are further differences: the AMR approach typically uses relatively coarse granularity where each refined patch contains thousands of cells, whereas the wavelet method refines individual grid points to control the error locally (although granularity can be decreased using the wavelet blocks approach described below). The two methods also use different error control strategies. AMR uses a variety of methods to control grid refinement, including Richardson extrapolation with a grid twice as fine in each direction as the existing grid to flag those cells requiring refinement (Berger and Colella, 1989). The coarse (i.e. actual) solution is first projected onto the finer grid, and the fine-grid solution is advanced in time the equivalent of one coarse time step. The difference between coarse and fine solutions gives a direct estimate of the local truncation error in time. In contrast, in the present wavelet approach the local interpolation error is used directly to determine whether the grid is sufficient to resolve the local structure that has developed over one time step, or whether it needs to be refined (or can be coarsened). Since an adjacent grid in scale (i.e. at half the local scale) is added to the grid determined by the local interpolation error, this also gives an estimate of the local time integration error. Unlike wavelet methods, AMR approaches do not guarantee control of the L^∞ error. Despite their differences, it is unclear at present which method is more efficient and accurate in practice.

Atmospheric and oceanic flows are strongly turbulent, with eddies developing virtually everywhere and on a wide range of scales. This fact is a legitimate source of scepticism towards the effectiveness of an adaptive strategy, since it suggests that the grid would need to be refined everywhere. Therefore we included a test case featuring a statistically homogeneous and isotropic turbulent flow with a dense distribution of coherent structures. Our results show that high compression and physical fidelity can be achieved even in this seemingly unfavourable situation.

The current adaptive wavelet approach has been developed specifically for the TRiSK discretization, but it would be relatively straightforward to modify it for other discretizations of the SWEs on staggered grids. The multiscale structure and conservative restriction and prolongation operators developed here should also be useful for static grid refinement in meteorology (e.g. embedding a regional model in a global model) and for analyzing observational and model-generated data. Although the basic features of the method are well-established and quite general, the particular implementation presented here should be seen as a proof-of-concept, rather than an operational code. In order to extend what has been developed here to build a fully adaptive climate or weather model on the sphere, much computational and analytical work remains.

First, the method needs to be extended to the sphere, parallelized, and made computationally efficient for large problems. Since we have been careful to develop the method so that it is consistent with the multiscale spherical C-grid (i.e. dyadic subdivisions of the icosahedron projected onto the sphere), this should be relatively straightforward. The main challenges are the non-uniformity of the grid on the sphere (especially near the twelve valence 5 points) and the need to encode the local discrete spherical geometry in the calculation of fluxes and source terms. In addition, the convergence of TRiSK or similar operators on arbitrary refinements of the icosahedral grid is degraded, or lost, even in the non-adaptive case, unless some global grid optimization is performed (Heikes and Randall, 1995; Tomita, 2002; Xu, 2006).

The principal computational challenge is efficient parallelization (including dynamic load balancing), although we can take advantage of existing approaches used in other adaptive methods, such as AMR. The extension of the adaptive wavelet to spherical C-grids is under way. For problems with relatively few points at the smallest scales, computational efficiency could be improved by using multiscale time-stepping where coefficients at each scale j are advanced using the time step $\Delta t_j = 2^{-j}\Delta t_0$ appropriate to that scale (Domingues *et al.*, 2008; Hejazialhosseini *et al.*, 2010). Another technique for improving the computational efficiency of an adaptive wavelet method is to use wavelet blocks (Hejazialhosseini *et al.*, 2010). The idea behind wavelet blocks is to reduce the number of sequential operations required to access a particular wavelet or scaling coefficient by using a coarser data structure. Instead of using a quad-tree data structure (as here) where each coefficient is a leaf, granularity is increased by using leaves that are large blocks of coefficients. This make the method less adaptive, but reduces the cost of tree operations by an order of magnitude or more. Varying the size of the blocks allows the user to tune the method for the particular application and, since the adapted grid is usually block-like, the impact on adaptivity should be small.

Secondly, the current two-dimensional method must be extended to three dimensions. For geophysical flows on the sphere, the simplest strategy would be to use a constant number of model levels, either terrain-following (for the atmosphere) or terrain-intersecting (for the ocean), and a horizontal adapted grid common to all levels. This would facilitate whole-column calculations like convective adjustment and radiative transfer, but would also mean that there is no grid adaptivity in the vertical dimension. In order to model ocean circulation, solid boundaries would need to be introduced. Note that adaptive wavelet methods are particularly well-adapted to vortical boundary-layer flows (Kevlahan and Vasilyev, 2005) and so should be advantageous for resolving features such as western boundary currents.

Finally, although the adaptive wavelet approach allows for uniform error control and should better resolve the small-scale active features of the flow, weather and climate models will necessarily remain strongly under-resolved, at least for some phenomena such as cloud formation. Thus, a major research effort is required to analyze and understand the interplay between a dynamically adapting grid and subgrid parametrizations. The lack of such understanding is one of the principal reasons holding back the adoption of dynamically adaptive weather and climate models. The relatively simple case of subgrid-scale modelling for incompressible Navier–Stokes turbulence in LES suggests that existing subgrid parametrizations might perform well in an adaptive wavelet method (Vasilyev *et al.*, 2008; de Stefano and Vasilyev, 2010). In fact, Nikiforakis (2009) proposed that adaptive subgrid-scale parametrizations for climate and weather modelling could be based on the approaches used for LES. We hope that the results presented here are a significant step forward in the effort to develop the next generation of dynamically adaptive weather and climate models.

Acknowledgements

NKRK was supported by an NSERC Discovery Grant and by an invited Professorship at École Polytechnique. NKRK and TD benefitted from the Mobility Program in Science and Technology run by the French Embassy in Canada. This work was supported by IPSL, project DYNAMICO. We would like to thank Todd Ringler and an anonymous reviewer for their comments and suggestions which helped improve the manuscript.

References

Alam J, Kevlahan NK-R, Vasilyev OV. 2007. Scaling of space–time modes with Reynolds number in two-dimensional turbulence. *J. Fluid Mech.* **570**: 217–226.

Bacon DP, Ahmad NN, Boybeyi Z, Dunn TJ, Hall MS, Lee PCS, Sarma RA, Turner MD, Waight KT, Young SH, Zack JW. 2000. A dynamically adapting weather and dispersion model: the Operational Multiscale Environment model with Grid Adaptivity (OMEGA). *Mon. Weather Rev.* **128**: 2044–2076.

Berger MJ, Colella P. 1989. Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.* **82**: 64–84. DOI: 10.1016/0021-9991(89)90035-1

Berger MJ, Leveque RJ. 1998. Adaptive mesh refinement using wave-propagation algorithms for hyperbolic systems. *SIAM J. Numer. Anal.* **35**: 2298–2316.

Berger MJ, Oliger J. 1984. Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.* **53**: 484–512.

Boffetta G, Ecke RE. 2012. Two-dimensional turbulence. *Ann. Rev. Fluid Mech.* **44**: 427–451. DOI: 10.1146/annurev-fluid-120710-101240

Boffetta G, Musacchio S. 2010. Evidence for the double cascade scenario in two-dimensional turbulence. *Phys. Rev. E* **82**: 016307, DOI: 10.1103/PhysRevE.82.016307

Chen C, Xiao F, Li X. 2010. An adaptive multimoment global model on a cubed sphere. *Mon. Weather Rev.* **139**: 523–548.

Chen Q, Gunzburger M, Ringler TD. 2011. A scale-invariant formulation of the anticipated potential vorticity method. *Mon. Weather Rev.* **139**: 2614–2629.

Cohen A, Dahmen W, DeVore RA. 2002. Sparse evaluation of compositions of functions using multiscale expansions. Tech. report 222. IGPM, RWTH: Aachen

Cohen A, Kaber SM, Müller S, Postel M. 2003. Fully adaptive multiresolution schemes for conservation laws. *Math. Comput.* **72**: 183–225.

Cottet G-H, Poncet P. 2002. Particle methods for direct numerical simulations of three-dimensional wake flows. *J. Turbulence* **3**: 1–9.

Daubechies I, Sweldens W. 1998. Factoring wavelet transforms into lifting steps. *J. Fourier Anal. Appl.* **4**: 247–270.

de Stefano G, Vasilyev O. 2010. Stochastic coherent adaptive large-eddy simulation of forced isotropic turbulence. *J. Fluid Mech.* **646**: 453–470.

Domingues MO, Gomes SM, Roussel O, Schneider K. 2008. An adaptive multiresolution scheme with local time stepping for evolutionary PDEs. *J. Comput. Phys.* **227**: 3758–3780. DOI: 10.1016/j.jcp.2007.11.046

Dumont S, Lebon F. 1996. Wavelet-Galerkin method for periodic heterogeneous media. *Comput. Struct.* **61**: 55–65.

Farrell PE, Piggott MD, Pain CC, Gorman GJ, Wilson CR. 2009. Conservative interpolation between unstructured meshes via supermesh construction. *Comput. Methods Appl. Math.* **198**: 2632–2642.

Fröhlich J, Schneider K. 1996. Numerical simulation of decaying turbulence in an adaptive wavelet basis. *Appl. Comput. Harmon. Anal.* **3**: 393–397.

Fröhlich J, Schneider K. 1997. An adaptive wavelet-vaguelette algorithm for the solution of PDEs. *J. Comput. Phys.* **130**: 174–190.

Fryxell B, Olson K, Ricker P, Timmes FX, Zingale M, Lamb DQ, MacNeice P, Rosner R, Truran JW, Tufo H. 2000. FLASH: An adaptive mesh hydrodynamics code for modeling astrophysical thermonuclear flashes. *Astrophys. J. Suppl.* **131**: 273–334. DOI: 10.1086/317361

George DL, LeVeque RJ. 2006. Finite-volume methods and adaptive refinement for tsunami propagation and inundation. *Sci. Tsunami Hazards* **24**: 319–328.

Goldstein DE, Vasilyev OV, Kevlahan NK-R. 2005. CVS and SCALES simulation of 3D isotropic turbulence. *J. Turbulence* **6**: 37, DOI: 10.1080/14685240500460790

Heikes R, Randall DA. 1995. Numerical integration of the shallow-water equations on a twisted icosahedral grid. Part II. A detailed description of the grid and an analysis of numerical accuracy. *Mon. Weather Rev.* **123**: 1881–1887.

Hejazialhosseini B, Rossinelli D, Bergdorf M, Koumoutsakos P. 2010. High-order finite-volume methods on wavelet-adapted grids with local time-stepping on multicore architectures for the simulation of shock-bubble interactions. *J. Comput. Phys.* **229**: 8364–8383. DOI: 10.1016/j.jcp.2010.07.021

Jablonowski C, Oehmke RC, Stout QF. 2009. Block-structured adaptive meshes and reduced grids for atmospheric general circulation models. *Phil. Trans. R. Soc. London A* **367**: 4497–4522.

Kellay H, Goldburg WI. 2002. Two-dimensional turbulence: a review of some recent experiments. *Reports Prog. Phys.* **65**: 845–894. DOI: 10.1088/0034-4885/65/5/204

Kevlahan NK-R, Vasilyev OV. 2005. An adaptive wavelet collocation method for fluid–structure interaction at high Reynolds numbers. *SIAM J. Sci. Comput.* **26**: 1894–1915.

Krinner G, Genthon C, Li Z-X, Le Van P. 1997. Studies of the Antarctic climate with a stretched-grid general circulation model. *J. Geophys. Res.–Atmos.* **102**: D12, 13731–13745. DOI: 10.1029/96JD03356

Läuter M, Handorf D, Rakowsky N, Behrens J, Frickenhaus S, Best M, Dethloff K, Hiller W. 2007. A parallel adaptive barotropic model of the atmosphere. *J. Comput. Phys.* **223**: 609–628.

Long D, Thuburn J. 2011. Numerical wave propagation on non-uniform one-dimensional staggered grids. *J. Comput. Phys.* **230**: 2643–2659.

MacDonald AE, Middlecoff J, Henderson T, Lee JL. 2011. A general method for modeling on irregulargrids. *Int. J. High Perform. Comput.* **25**: 392–403.

Mallat S. 1998. *A wavelet tour of signal processing*. Academic Press.

Mehra M, Kevlahan NK-R. 2008. An adaptive wavelet collocation method for the solution of partial differential equations on the sphere. *J. Comput. Phys.* **227**: 5610–5632. DOI: 10.1016/j.jcp.2008.02.004

Nikiforakis N. 2009. Mesh generation and mesh adaptation for large-scale earth-system modelling. *Phil. Trans. R. Soc. London A* **367**: 4473–4481.

Popinet S. 2003. Gerris: a tree-based adaptive solver for the incompressible euler equations in complex geometries. *J. Comput. Phys.* **190**: 572–600.

Ringler TD, Thuburn J, Klemp JB, Skamarock WC. 2010. A unified approach to energy conservation and potential vorticity dynamics for arbitrarily structured C-grids. *J. Comput. Phys.* **229**: 3065–3090. DOI: 10.1016/j.jcp.2009.12.007

Ringler TD, Jacobsen D, Gunzburger M, Ju L, Duda M, Skamarock W. 2011. Exploring a multiresolution modeling approach within the shallow-water equations. *Mon. Weather Rev.* **139**: 3348–3368.

Roussel O, Schneider K. 2010. Coherent vortex simulation of weakly compressible turbulent mixing layers using adaptive multiresolution methods. *J. Comput. Phys.* **229**: 2267–2286. DOI: 10.1016/j.jcp.2009.11.034

Roussel O, Schneider K, Tsigulin A, Bockhorn H. 2003. A conservative fully adaptive multiresolution algorithm for parabolic PDEs. *J. Comput. Phys.* **188**: 493–523. DOI: 10.1016/S0021-9991(03)00189-X

Ruge JW, McCormick SF, Yee SYK. 1995. Multilevel adaptive methods for semi-implicit solution of shallow-water equations on a sphere. *Mon. Weather Rev.* **123**: 2197–2205.

Sadourny R. 1975. The dynamics of finite-difference models of the shallow-water equations. *J. Atmos. Sci.* **32**: 680–689.

Schneider K, Kevlahan NK-R, Farge M. 1997. Comparison of an adaptive wavelet method and nonlinearly filtered pseudo-spectral methods for two-dimensional turbulence. *Theor. Comput. Fluid Dyn.* **9**: 191–206.

Skamarock WC, Oliger J, Street RL. 1989. Adaptive grid refinement for numerical weather prediction. *J. Comput. Phys.* **80**: 27–60.

Skamarock WC, Klemp JB, Duda MG, Fowler LD, Park S-H, Ringler TD. 2012. A multiscale non-hydrostatic atmospheric model using centroidal Voronoi tesselations and C-grid staggering. *Mon. Weather Rev.* **140**: 3090–3105.

Spiteri RJ, Ruuth SJ. 2002. A new class of optimal high-order strong-stability-preserving time discretization methods. *SIAM J. Numer. Anal.* **40**: 469–491.

St-Cyr A, Jablonowski C, Dennis JM, Tufo HM, Thomas SJ. 2008. A comparison of two shallow-water models with nonconforming adaptive grids. *Mon. Weather Rev.* **136**: 1898–1922.

Sweldens W. 1996. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Appl. Comput. Harmon. Anal.* **3**: 186–200.

Thuburn J, Ringler TD, Skamarock WC, Klemp JB. 2009. Numerical representation of geostrophic modes on arbitrarily structured C-grids. *J. Comput. Phys.* **228**: 8321–8335. DOI: 10.1016/j.jcp.2009.08.006

Tomita H. 2002. An optimization of the icosahedral grid modified by spring dynamics. *J. Comput. Phys.* **183**: 307–331.

Vasilyev OV, Bowman C. 2000. Second-generation wavelet collocation method for the solution of partial differential equations. *J. Comput. Phys.* **165**: 660–693.

Vasilyev OV, Paolucci S. 1996. A dynamically adaptive multilevel wavelet collocation method for solving partial differential equations in a finite domain. *J. Comput. Phys.* **125**: 498–512.

Vasilyev OV, de Stefano G, Goldstein DE. 2008. Lagrangian dynamic SGS model for stochastic coherent adaptive large-eddy simulation. *J. Turbulence* **9**: 1–14.

Verstappen RWCP, Veldman AEP. 1997a. Direct numerical simulation of turbulence at lower costs. *J. Eng. Math.* **32**: 143–159.

Verstappen RWCP, Veldman AEP. 1997b. Spectro-consistent discretization of Navier–Stokes: a challenge to RANS and LES. *J. Eng. Math.* **34**: 163–179.

Weller H. 2009. Predicting mesh density for adaptive modelling of the global atmosphere. *Phil. Trans. R. Soc. London A* **367**: 4523–4542.

Weller H, Weller HG, Fournier A. 2009. Voronoi, Delaunay and block-structured mesh refinement for solution of the shallow-water equations on the sphere. *Mon. Weather Rev.* **137**: 208–422.

Xu G. 2006. Discrete Laplace–Beltrami operator on sphere and optimal spherical triangulations. *Internat. J. Comput. Geom. Appl.* **16**: 75–93.