Voronoi, Delaunay and Block Structured Mesh Refinement for Solution of the Shallow Water Equations on the Sphere

HILARY WELLER *

NCAS Climate, Department of Meteorology, University of Reading, UK

HENRY G. WELLER

OpenCFD Limited, UK

AIMÉ FOURNIER

NCAR, Boulder, Colorado, US

Corresponding author address: Dr Hilary Weller, Department of Meteorology, University of Reading, PO Box 243, UK.
E-mail: h.weller@rdg.ac.uk

ABSTRACT

Alternative meshes of the sphere and adaptive mesh refinement could be immensely beneficial for weather and climate forecasts but it is not clear how mesh refinement should be achieved. A finite-volume model which solves the shallow-water equations on any mesh of the surface of the sphere is presented. The accuracy and cost effectiveness of four quasi-uniform meshes of the sphere are compared: a cubed-sphere, reduced latitude-longitude, hexagonal-icosahedral and triangular-icosahedral. On some standard shallow-water tests, the hexagonal-icosahedral mesh performs best and the reduced latitude-longitude mesh performs well only when the flow is aligned with the mesh.

The inclusion of a refined mesh over a disc shaped region is achieved using either gradual Delaunay, gradual Voronoi or abrubt 2:1 block structured refinement. These refined regions can actually degrade global accuracy, presumably due to changes in wave dispersion where the mesh is highly non-uniform. However using gradual refinement to resolve a mountain in an otherwise coarse mesh can improve accuracy for the same cost.

The model prognostic variables are height and momentum colocated at cell centres and (to remove grid-scale oscillations of the A-grid) the mass flux between cells is advanced from the old momentum using the momentum equation. Quadratic and upwind biased cubic differencing are used as explicit corrections to a fast implicit solution that uses linear differencing.

Contents

1. Introduction

Adaptive and variable-resolution modelling of the atmosphere is of increasing interest due to the potential benefits to, for example, regional weather and climate prediction, local resolution of convection and orography and cyclone tracking. These techniques have been under investigation for atmospheric modelling for some time (eg. Staniforth and Mitchell 1978; Berger and Oliger 1984; Skamarock et al. 1989; Dietachmayer and Droegemeier 1992; Skamarock and Klemp 1993; Fiedler and Trapp 1993) but are still mostly research oriented rather than operational (eg. Jablonowski et al. 2006; Läuter et al. 2007; St-Cyr et al. 2008), a noteworthy exception being Bacon et al. (2000). The problems with variable-resolution modelling are less well publicised than the successes. A brash and sweeping generalisation would be that high-order methods for unstructured meshes (such as discontinuous-Galerkin or Spectral-Element) are expensive (eg Läuter et al. 2008) whereas low-order methods such as lower-order finite-volumes and elements are insufficiently accurate (eg Läuter et al. 2005), especially where the mesh changes resolution (eg Lanser et al. 2000).

Spatial changes in resolution can degrade the accuracy globally, particularly of a low-order model, if high resolution is placed arbitrarily to test the numerics rather than to refine a feature of interest (St-Cyr et al. 2008). Alternatively, high local resolution can increase accuracy, particularly of a higher-order model if, for example, either external forcing or orographic forcing are significantly spatially localised (Fournier et al. 2004). Local resolution is particularly beneficial for resolving features which would otherwise be smaller than the stencil or element used for high-order differencing. It is thought that changes in resolution can degrade global accuracy due to spurious wave reflection and scattering (Vichnevetsky 1987; Vichnevtsky and Turner 1991), a problem which is partly rectified by using gradual rather than abrupt mesh refinement.

Low-order models have difficulties in maintaining balances between tightly curved fields with many non-zero derivatives, and the resulting spurious imbalance leads to the break down of balanced but physically unstable jets (eg Jablonowski and Williamson 2006; Läuter et al. 2008). These problems do not occur when the flow is aligned with, for example, a latitude-longitude mesh. Some models on hexagonal or triangular meshes are only second-order accurate where the mesh is uniform and revert to first-order accuracy where the mesh is non-uniform. The resulting errors have motivated the development of algorithms to adjust icosahedral meshes to be as uniform as possible, for example using spring dynamics (Tomita et al. 2002) or deformations to reduce mesh skewness (Heikes and Randall 1995).

High-order models have overcome some of the problems described above but some problems can worsen. Läuter et al. (2008) described the problems with representing a barotropically unstable jet with both a low-order ($k = 2$) and a high-order ($k = 6$) DG model. The low-order model does not respect the tight variations in the balanced flow whereas, once realistic barotropic instabilities develop, the high-order model creates spurious oscillations. Further increases in the order of accuracy do not produce more accurate results for the same number of degrees of freedom because then the span of the points required to fit a higher order polynomial exceeds the width of the jet. Oscillations are also produced when using the spectral-transform method to simulate this test case without diffusion (Galewsky et al. 2004) and using the spectral-element method (St-Cyr et al. 2008).

St-Cyr et al. (2008) and Jablonowski and Williamson (2006) compare models with different mesh types: latitude-longitude, the cubed-sphere and icosahedral, on various two- and three-

dimensional test cases. Some useful conclusions are drawn concerning the different numerical methods and the different meshes. For example, the cubed-sphere and icosahedral meshes can trigger spurious wave-number four and five patterns respectively. However the numerical schemes on the different mesh structures were very different. In contrast, here we will compare these different meshes and different refinement patterns all with the same numerical model.

To our knowledge there has been no systematic study of the advantages and disadvantages of different refinement techniques in the context of spherical shallow-water simulation. Block-structured 2:1 refinement has been popular because structured-mesh models can be nested explicitly (eg. Skamarock and Klemp 1993) so that finer meshes can use shorter time-steps. This type of nesting is used with explicit time-stepping which has severe time-step restrictions, based on the speed of the fastest waves. Implicit two-way nesting with longer variable time-steps has been used by Almgren et al. (1998) for incompressible flows. Block-structured refinement has also been popular as multigrid algorithms for space and time can in principle be used (eg Ruge et al. 1995). Unstructured meshes are particularly popular for ocean modelling since coastlines and bottom topography can be represented accurately (eg Pain et al. 2005) but they have also been used for atmospheric modelling in order to refine gradually (eg Bacon et al. 2000; Läuter et al. 2007). The potential for unstructured meshes to refine anisotropically in an entirely flexible manner and align with the flow has not yet been applied to the global atmosphere.

Here we plan to compare the accuracy of various block-structured and unstructured meshes and refinement patterns. The model is described in section 2 and the meshes are introduced in section 3, with a more detailed description in appendix A. Shallow-water test cases of global steady-state geostrophic flow, flow over an isolated mountain and a barotropically unstable jet are presented in section 4 and conclusions drawn on the suitability of different meshes and the current modelling framework in section 5.

2. Discretisation of the Shallow-Water Equations on Polygons

The shallow-water equations on a rotating sphere have been implemented in the open-source C++ library OpenFOAM (Open Field Operation And Manipulation `www.opencfd.co.uk`), to create AtmosFOAM (which has in part been described by Weller and Weller (2008)).

a. The Shallow-Water Equations

The shallow-water equations (SWEs) are written in 3D vector flux form on a sphere:

$$\frac{\partial h\mathbf{U}}{\partial t} + \nabla \bullet (h\mathbf{U} \otimes \mathbf{U}) = -2\boldsymbol{\Omega} \times h\mathbf{U} - gh\nabla(h + h_0) + \mu\hat{\mathbf{r}} \quad (1)$$

$$\frac{\partial h}{\partial t} + \nabla \bullet (h\mathbf{U}) = 0, \quad (2)$$

where $\mathbf{U}$ is the 3D velocity vector, ∇ is the 3D gradient, $\nabla\bullet$ is the 3D divergence operator, h is the height of the fluid surface above the solid surface, h_0 is the height of the solid surface above a spherical reference height, $\boldsymbol{\Omega}$ is the angular-velocity vector of the sphere, g is the acceleration magnitude due to gravity, $\mathbf{r}$ is the position vector on the surface of the sphere, $\hat{\mathbf{r}} = \mathbf{r}/|\mathbf{r}|$ and μ is the Lagrange multiplier which constrains the motion to follow the sphere (Côté 1988). ($h\mathbf{U} \otimes \mathbf{U} = h\mathbf{U}\mathbf{U}^T$ assuming $\mathbf{U}$ is a column vector.)

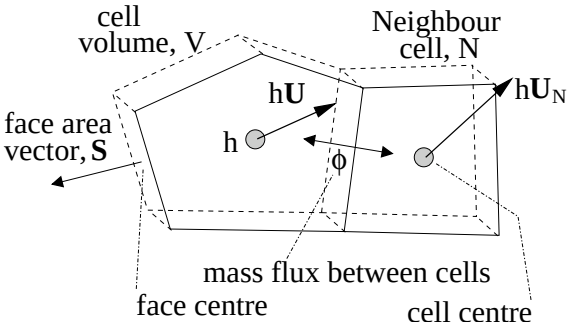

FIG. 1. Cells in an OpenFOAM mesh

b. The Finite-Volume Formulation on Polygons

OpenFOAM uses the finite-volume technique on 3D polyhedral meshes in Cartesian coordinates. The shallow-water model AtmosFOAM uses a spherical shell of polygons in Cartesian coordinates. Rather than using the Lagrange multiplier to constrain the velocity to the surface of the sphere, we remove the radial component of the momentum during each iteration ($h\mathbf{U} \rightarrow h\mathbf{U} - (h\mathbf{U} \bullet \hat{\mathbf{r}})\hat{\mathbf{r}}$), similar to Heinze and Hense (2002).

The prognostic variables are the three components of the cell-average momentum, $h\mathbf{U}$, and the cell average height, h (figure 1). The mass flux between cells, ϕ, is advanced from the previous time-step by solving the momentum equation as described by Rhie and Chow (1983) and so is considered a quasi-prognostic variable. This removes grid-scale oscillations of the A-grid (Arakawa and Lamb 1977). Using local coordinate mappings for every cell individually in order to solve just two components of momentum per cell as in Läuter et al. (2008) would be more efficient but this has not yet been implemented. We will first give an overview of the solution algorithm and then more details about the precise discretisation:

1) OVERVIEW OF SOLUTION ALGORITHM

i. The cell-centred momentum equation is linearised about the solution from the previous iteration and each component solved implicitly.

ii. The momentum equation is also discretised on the faces and is combined with the continuity equation to form a modified Helmholtz equation to predict the height, h, and the face fluxes, ϕ. This allows time-steps much longer than the Courant-Friedrichs-Lewy (CFL) restriction based on the gravity wave speed, c: $\left(\frac{\delta tc}{\delta x}\right)$.

iii. Two iterations are performed each time-step with, higher-order discretisation, non-linear and Coriolis terms updated for the second iteration to improve accuracy and convergence.

The advantage of solving each component of the momentum equation implicitly is the resulting stability of centred, non-diffusive differencing without the need for explicit diffusion or limiters. The algorithm is not monotonic so limiters would be needed for tracers or on a case where the height approaches zero.

2) DISCRETISATION OF THE MOMENTUM EQUATION

We integrate the momentum equation between time-steps 0 (previous) and 1 (current) using Gauss' theorem applied to the divergence term and the height gradient:

$$\tfrac{V}{\delta t}\big((h\mathbf{U})^1-(h\mathbf{U})^0\big)+\sum_f(h\mathbf{U}\otimes\mathbf{U})_f^\alpha\bullet\mathbf{S}=-V\big(2\boldsymbol{\Omega}\times h\mathbf{U}\big)^\alpha-gh\sum_f(h+h_0)_f^\alpha\mathbf{S}, \qquad (3)$$

where V is the cell volume, δt is the time-step, $\sum_f$ means summation over all the faces of a cell, subscript f indicates face average quantities, $\mathbf{S}=\mathbf{S}_f$ is the outward-pointing vector normal to a face with the magnitude of the area of that face (figure 1) and α is the blending factor between the old and new time-steps, *i.e.* $h^\alpha=\alpha h^1+(1-\alpha)h^0$. $\alpha=\frac{1}{2}$ gives second-order, Crank-Nicholson time discretisation. We have used $\alpha=0.55$ to improve stability, but this also reduces the temporal order of accuracy.

Each of the 3 components of the momentum equation is solved separately with other components lagged (*i.e.* most up to date value used from before the start of the implicit solution). A lagged term at time δt is denoted using superscript $\ell(1)$ and a lagged term at time $\alpha\delta t$ by superscript $\ell(\alpha)$ where $h^{\ell(\alpha)}=\alpha h^{\ell(1)}+(1-\alpha)h^0$. The non-lagged terms with superscripts 1 and α at times δt and $\alpha\delta t$ are the terms which are solved implicitly. The Coriolis and height-gradient terms on the right-hand side of equation 3 are all lagged. The non-linear advection term, $\sum_f(h\mathbf{U}\mathbf{U})_f^\alpha\bullet\mathbf{S}$ is linearised by replacing $(h\mathbf{U})_f^\alpha\bullet\mathbf{S}$ with $\phi^{\ell(\alpha)}$. The remaining $\mathbf{U}_f^\alpha$ factor in this term is handled implicitly by interpolating from cell-average momentum and height to face-average velocity:

$$\mathbf{U}_f^\alpha=\lambda(h\mathbf{U})^\alpha/h^{\ell(\alpha)}+(1-\lambda)(h\mathbf{U})_N^\alpha/h_N^{\ell(\alpha)}+\mathbf{C}_{\text{hot}}^{\ell(\alpha)} \qquad (4)$$

where $\lambda=\lambda_f$ is the linear interpolation factor to interpolate between the cell centre and the neighbour cell centre, N, on the other side of face f, and $\mathbf{C}_{\text{hot}}^{\ell(\alpha)}$ are the higher order terms to correct the linear interpolation. The method for finding the stencil and calculating the higher-order terms is given in section c. Equation 4 is substituted into the non-linear advection term and non-lagged, superscripted α terms are replaced with the appropriate combination of old and new time-steps to give the discretised non-linear advection term:

$$\sum_f(h\mathbf{U}\mathbf{U})_f^\alpha\bullet\mathbf{S}=\alpha A^{\ell(1)}(h\mathbf{U})^1+(1-\alpha)A^0(h\mathbf{U})^0+\alpha\mathbf{B}^1+(1-\alpha)\mathbf{B}^0+\mathbf{C}_{\text{hot}}^{\ell(\alpha)} \qquad (5)$$

$$\text{where } A^k=\sum_f\big(\lambda\,\phi^k\big)/h^k,\ \mathbf{B}^k=\sum_f(1-\lambda)\,\phi^{\ell(k)}(h\mathbf{U})_N^k/h_N^{\ell(k)},\ {}_{k=0,1.}$$

$\mathbf{B}^1$ contains all the inter-cell dependencies that will be solved simultaneously, *i.e.* all the off-diagonal elements of a matrix that operates on a column of $h\mathbf{U}^1$ components over all cells. We now substitute the non-linear advection, equation 5, into the discretised momentum equation (3) and rearrange to give all terms involving $(h\mathbf{U})$ at the new time-step on the left hand side:

$$\begin{aligned}(h\mathbf{U})^1\big(\tfrac{V}{\delta t}+\alpha A^{\ell(1)}\big)+\alpha\mathbf{B}^1=(h\mathbf{U})^0\big(\tfrac{V}{\delta t}-(1-\alpha)A^0\big)-(1-\alpha)\mathbf{B}^0-\mathbf{C}_{\text{hot}}^{\ell(\alpha)}\\ -2V\boldsymbol{\Omega}\times(h\mathbf{U})^{l(\alpha)}-gh^{l(\alpha)}\sum_f(h+h_0)_f^{l(\alpha)}\mathbf{S}\end{aligned} \qquad (6)$$

Equation 6 is written as three, sparse matrix equations, one matrix for each component of $(h\mathbf{U})^1$. These matrices have dimensions $N_{cells} \times N_{cells}$ for a mesh with N_{cells} cells and N_N non-zero off-diagonal elements per row for cells with N_N face neighbours (sides). For a CFL limit of about one, the matrix is diagonally dominant and is solved with a bi-conjugate gradient iterative solver with incomplete LU preconditioning.

3) SIMULTANEOUS SOLUTION FOR THE FLUXES AND HEIGHTS

The continuity equation (2) is discretised using Gauss' divergence theorem:

$$\tfrac{V}{\delta t}\big(h^1 - h^0\big) + \alpha \sum_f \phi^1 + (1-\alpha)\sum_f \phi^0 = 0. \tag{7}$$

The equation for the flux, ϕ, is derived by interpolating the discretised momentum equation (6) onto the cell faces, taking the dot product with the face-area vector, $\mathbf{S}$, and rearranging:

$$\phi^1 = D_f \phi^0 - \mathbf{E}_f \bullet \mathbf{S} - \delta t F_f |\mathbf{S}| \nabla_f (h^\alpha + h_0) \tag{8}$$

where $D = \frac{V-(1-\alpha)\,\delta t\, A^0}{V+\alpha\,\delta t\, A^{\ell(1)}}$, $\quad \mathbf{E} = \delta t \frac{\left(\mathbf{B}+\mathbf{C}_{\text{hot}}+2V\boldsymbol{\Omega}\times(h\mathbf{U})\right)^{\ell(\alpha)}}{V+\alpha\,\delta t\, A^{\ell(1)}}$, $\quad F = \frac{Vgh^{\ell(\alpha)}}{V+\alpha\,\delta t\, A^{\ell(1)}}$.

We have not used the lagged cell-centred discretisation of the height gradient from equation 6 to derive equation 8. Instead the height gradient has been re-discretised at the face, making the algorithm staggered and the computational molecule compact:

$$\nabla_f h^\alpha = (h_N^\alpha - h^\alpha)/\delta x + G_{\text{hot}}^{\ell(\alpha)} \tag{9}$$

where δx is the distance between the cell centre and the cell centre of the neighbour, N, on the other side of face f and $G_{\text{hot}}^{\ell(1)}$ contains corrections for higher-order terms which are lagged (see section c).

In order to remove the possible inconsistency which arises because we have separate diagnostic equations for the cell-centred momentum and the fluxes, the old time level flux is interpolated from the old time level momentum following Rhie and Chow (1983):

$$\phi^0 = (h\mathbf{U})_f^0 \bullet \mathbf{S} \tag{10}$$

Because this interpolation is done at the old time-step rather than the current time-step, most of the spurious computational mode of the Arakawa A-grid is removed. Remaining grid-scale oscillations are avoided by the use of upwind biased cubic differencing for $C_{\text{hot}}^{\ell(\alpha)}$ as will be described in section c.

ϕ and h are solved simultaneously by substituting equations (8) and (9) into the continuity equation (7) to create the "pressure equation", a modified Helmholz equation. This is rearranged so that non-lagged terms involving h^1 are on the left hand side.

$$\tfrac{V}{\delta t^2}(h^1 - h^0) - \alpha \sum_f F_f \nabla_f h^\alpha |\mathbf{S}| = \text{explicit terms} \tag{11}$$

Since the higher-order corrections for ∇_f are lagged, as are the non-linear part of the advection

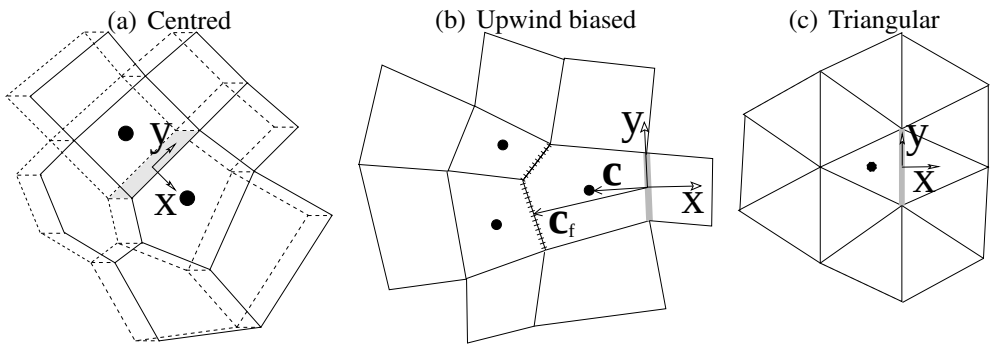

FIG. 2. The 2D stencils of cells for interpolating onto the grey face. The stencils are expanded out from the cells marked by circles.

and the Coriolis force, the matrix equation for h^1 has only N_N non-zero off-diagonal terms per row for cells with N_N face neighbours (sides). The matrix is solved using an incomplete Cholesky pre-conditioned conjugate-gradient iterative solver.

c. Quadratic and Cubic Differencing

We now define how the corrections for the higher-order terms ($\mathbf{C}_{\text{hot}}$ and G_{hot} in sections 2b2 and 2b3) are calculated. First a stencil of cells surrounding each face is found. To make discretisation on polyhedral meshes simple, cell-volume average quantities are approximated by the cell-centre value and face-area averages are approximated by face-centre values. These approximations are second-order accurate.

First we define a local coordinate system centred around the face centre with x in the direction of the face normal (figure 2). Interpolations and face-normal gradients are calculated using a 2D quadratic:

$$\psi = a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 xy + a_5 y^2, \tag{12}$$

where ψ is the field to be operated on. For the interpolation of velocity for the divergence in the momentum equation (which uses an upwind biased stencil) we have found improved accuracy by additionally including some of the terms from the 2D cubic:

$$\psi = a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 xy + a_5 y^2 + a_6 x^3 + a_7 xy^2. \tag{13}$$

These are the terms that can be fitted accurately using stencils shaped as in figure 2. Stevens and Bretherton (1996) used more symmetric stencils and so included different additional terms from a 3D cubic. The interpolant of field Ψ at the face, Ψ_f, and the face-centre gradient normal to the face, ∇_f, are then given by:

$$\Psi_f = a_0, \qquad \nabla_f \Psi = a_1 \tag{14}$$

These polynomials are fit over three different types of stencil:

(a) On non-triangular meshes, we fit the quadratic over a stencil centred on the face we interpolate onto (figure 2(a)). This stencil consists of the cells on either side of the face and all face neighbours of these cells.

(b) For the divergence in the momentum equation (for which we use some additional cubic

terms) we use an upwind biased stencil (figure 2(b)). We first find the one or two faces of the upwind cell that are opposite the face we are interpolating onto. The upwind face is defined to be the face with the maximum $\mathbf{c} \cdot \mathbf{c}_f$ where $\mathbf{c}$ and $\mathbf{c}_f$ are the vectors defining the cell-centre and face-centre positions with respect to the local coordinate system. If the face with the second largest $\mathbf{c} \cdot \mathbf{c}_f$ is within 90% of the maximum then this face is defined as an upwind face also. (Upwind faces marked as railway tracks on figure 2(b).) The stencil then consists of the cells either side of the upwind faces and all face neighbours of these cells.

(c) On meshes of triangles, the stencil of a face consists of the cell upwind of the face and all point neighbours of this cell (fig. 2(c)). The stencil needs to be this big so as to include a well placed upwind-upwind cell.

Depending on the mesh topology, these stencils will have more cells than unknowns in the quadratic or cubic so the system is over-specified. Following Lashley (2002) we find a least-squares fit using singular value decomposition with the immediate upwind and downwind cells of the face weighted by a factor of 1000 relative to the other cells of the stencil so that the fit is most accurate near the face. (We have not found the results to be sensitive to this weighting factor with values ranging from 10 to 10^9.) The singular value decomposition needs to be done only once per cell for a fixed mesh at the beginning of the simulation, leaving just one multiply per cell of each stencil per time-step to calculate an interpolation.

3. Meshes with and without local refinement

We create four different mesh structures at various resolutions both with and without local mesh refinement; hexagonal icosahedral, triangular icosahedral, cubed-sphere and reduced latitude-longutude. Some of the coarsest of each of these that include a disc of refined mesh are shown in figure 3. The refined disc has twice the resolution of its surroundings, has a radius of 25° and is centred at 30° North, 90° East.

We create quasi-uniform hexagonal icosahedral meshes using John Thuburn's mesh generator (Thuburn 1997). Triangular icosahedral meshes are simply the dual of the hexagonal meshes with hexagon and pentagon centres becoming triangle vertices.

The non-uniform Delaunay meshes of triangles (figure 3(d)) are generated by the algorithm described in appendix A based on a "desired resolution" field defined over the globe. The non-uniform Voronoi meshes of polygons (figure 3(a)) are the Voronoi duals of the meshes of triangles. That is, the centre of the circle enclosing each triangle becomes a vertex in the mesh of polygons. These meshes consist entirely of pentagons, hexagons and heptagons.

All the edges of the cubed-sphere mesh without refinement sweep out an equal angle as described by Fournier et al. (2004). The 2:1 refinement as shown in figure 3(b) is achieved by treating the large squares that are adjacent to two small squares as if they had five edges, two of which are in line.

The reduced latitude-longitude (lat-lon) meshes reduce the number of longitudes by a factor of 2 when the mesh spacing reaches half of that at the equator. There is also a polygonal cell at each pole in order to improve the accuracy of the cross-polar flow (figure 3(c)). The coarsenings towards the poles and the refined disc are achieved in the same way as for the cubed-sphere.

Some statistics of all the meshes that we use are given in table 1 — the total number of cells, total number of faces between cells, the distance between adjacent cell centres at the equator

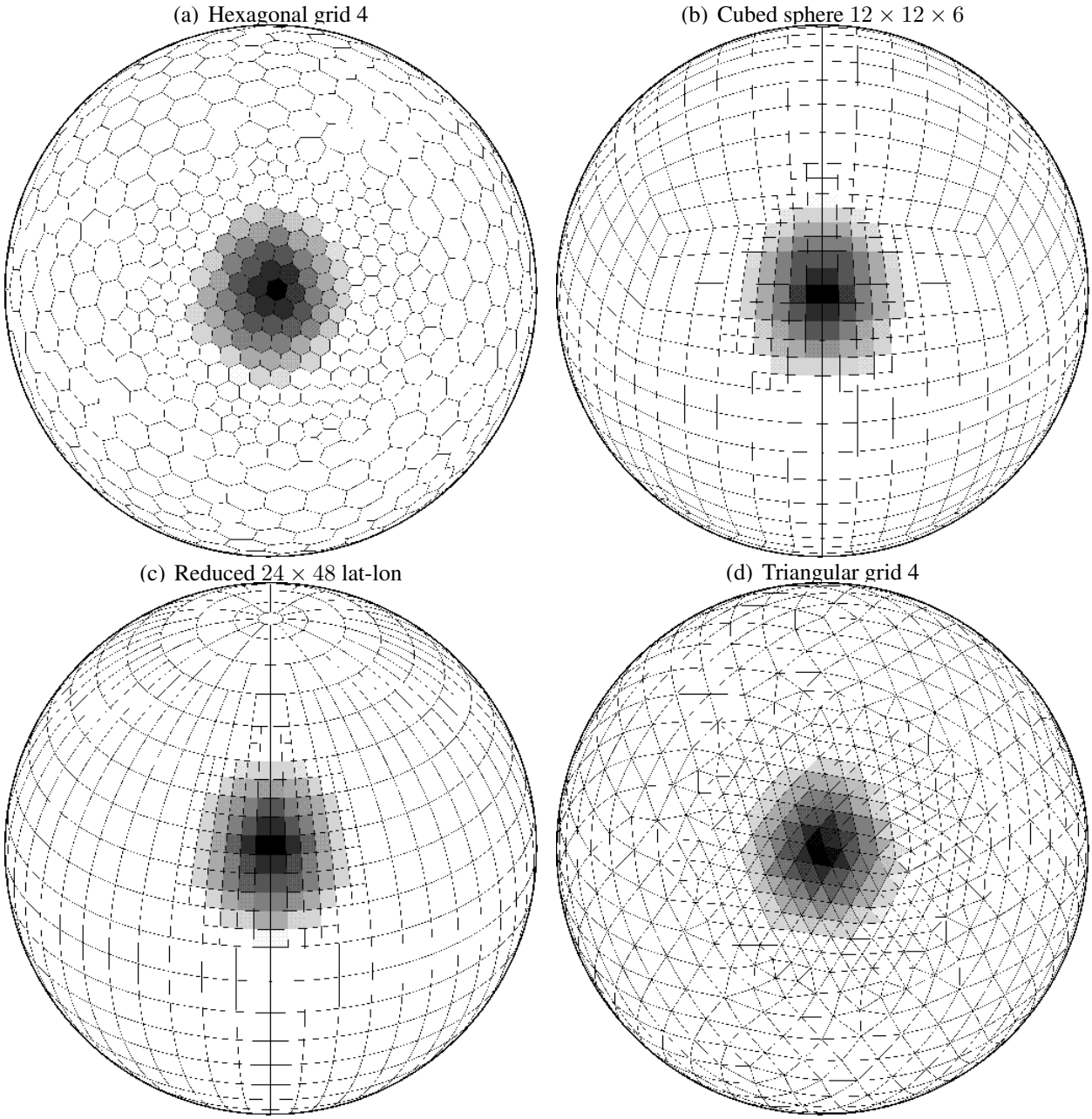

FIG. 3. The four mesh structures with double resolution in a disc at 30° N with a radius of 25°. Viewed from above the refined region. Shading shows the height of a conical mountain of radius 20° at 30°N

	Nominal resolution	Number of cells (without)	Number of faces (without)	Number of cells (with refinement)	Number of faces (with refinement)	δx_{eq} (km)	Reductions at (° lat)
Hexagonal	hex 3	162	480	233	693	1,914	
	hex 4	642	1,920	800	2,394	961	
	hex 5	2,562	7,680	2,950	8,844	480	
	hex 6	10,242	30,720	11,437	34,305	240	
	hex 7	40,962	122,880	44,872	134,610	120	
Cubed Sphere	6×12×12	864	1,728	960	1,932	834	
	6×24×24	3,456	6,912	3,834	7,695	417	
	6×48×48	13,824	27,648	15,330	30,711	208	
	6×72×72	31,104	62,208	34,776	69,631	139	
Reduced lat-lon	24×48	866	1,776	956	1,968	834	74, 82
	36×72	2,090	4,248	2,312	4,711	556	74, 85
	48×96	3,650	7,392	4,058	8,235	417	70, 82, 86
	96×192	14,786	29,760	16,412	33,064	208	67,80,86,88
	144×288	33,842	67,968	37,562	75,487	139	67,81,86,87,89
	576×1152	554,690	1,110,528	-	-	35	67,81,87,88,89,89.7
Triangular	tri 4	1,280	1,920	1,596	2,394	480	
	tri 5	5,120	7,680	5,896	8,844	240	
	tri 6	20,480	30,720	22,870	34,305	120	

TABLE 1. Some details of the different meshes. The meshes with local refinement have twice the resolution in a 25° radius disc centred at 30° North.

(δx_{eq}) and, for the reduced latitude-longitude meshes, the latitudes where the number of longitudes reduces by a factor of 2. The distance between cell centres is not necessarily a good indicator of accuracy or computational cost but is given for ease of comparison with other models. Cost depends on the number of cells, the number of faces and the number of cells in each stencil.

We have not computed results on full latitude-longitude meshes for two reasons; firstly because we have not implemented the necessary polar filter and secondly because the numerical schemes as implemented behave badly if a very large number of cells meet at one point or meet at one polygonal cell at each pole.

4. Shallow-water test cases

a. Williamson et al. (1992) Case 2: Global steady-state geostrophic flow

We first look at the accuracy and efficiency of representing steady-state geostrophically balanced flow (test case two of Williamson et al. 1992) on the different mesh structures defined in section 3. We show results both with and without the arbitrary circular refinement pattern of radius 25° at 30° North. All simulations use a 15-minute time-step for consistency with case 5 in section b. This time-step gives very low flow Courant-Friedrichs-Lewy (CFL) numbers, $\frac{\delta t|\phi|}{h|\mathbf{S}|\delta x}$, no greater than 0.2 on the finest meshes. We are therefore comparing only spatial truncation errors between different meshes, not temporal truncation errors. Figure 4 shows the height errors after five days

of steady-state flow rotated 45° from the geometric North pole for resolutions of all of the mesh structures which imply similar computational costs, both with and without a refined disc. The maximum errors of around 10m should be compared with the total equator-to-pole height difference of 1900m on these moderate resolution meshes. However, for all of the structures, mesh refinement degrades the solution (and at extra cost). In contrast, degree-8 spectral-element discretisation with local refinement increases the global accuracy even when the refinement is placed arbitrarily rather than based on local errors (St-Cyr et al. 2008).

The rotated hexagonal and triangular meshes appear not to project strongly onto the error patterns (figs 4(a) and 4(g)) unlike the rotated cubed-sphere and reduced lat-lon meshes (figs 4(c) and 4(e)). When refined discs are introduced, wave trains of errors form downstream, increasing the global errors despite the improved resolution of part of the domain. This is in line with Vichnevetsky (1987) and Vichnevtsky and Turner (1991) who explain the modifications to wave dispersion when resolution changes.

Using the cubed-sphere mesh, errors are generated as the flow alternates between being aligned with and at 45° to the mesh (figure 4(c)). Otherwise the cubed-sphere mesh gives good accuracy in comparison to the reduced latitude-longitude meshes. Cube edges and cube corners can also slightly increase errors even for much higher-order methods (eg order-15 computation of scalar advection in figure 7 of Taylor et al. 1997).

The mesh coarsening patterns towards the poles of the reduced latitude-longitude mesh lead to large errors because so many coarsening boundaries occur within a few cells of each other. For this mesh, the arbitrary circular refinement pattern makes little difference to the errors. It should be noted that a Fourier filter can be used with a reduced latitude-longitude mesh which may improve accuracy. This has not been done here.

The mesh of triangles is more prone to grid-scale noise, especially at the centres of the flow ($i.e.$ inside the lowest height contours in figs 4(g) and 4(h)) where the wind is very light and the upwinding introduces little numerical diffusion. The staggered component of the algorithm presented in section 2b3 is not beneficial for representing geostrophically balanced flow on triangles; a solution for discritised geostrophic balance does not exist on the triangular C-grid because there are not enough velocity degrees of freedom (Thuburn, pers. comm. 2009, `www.met.rdg.ac.uk/~sws02hs/Newton/Thuburn_waves.pdf`).

The different mesh structures have different computational costs for the same number of cells or for the same nominal resolution because of the differing numbers of faces per cell and the differing stencil sizes for the higher-order differencing (see section 2c). The important metric is the accuracy for the computational cost. We plot the normalised root mean square or ℓ_2 error norm (as defined in Williamson et al. 1992) as a function of computational cost (CPU time) and as a function of number of cells for each mesh structure and for various resolutions in figures 5(a) and 5(b) for this case both unrotated with respect to the geometric North pole, and rotated 45°. Lines showing first- and second-order convergence with resolution are also shown on these graphs, assuming that both CPU time and total number of cells are proportional to $1/\delta x^2$ for a constant time-step. Results on all mesh structures converge with second-order accuracy. These graphs show the superior accuracy for cost of using hexagons and the degradation in solution when an arbitrary refinement pattern is added. The meshes of triangles appear to give poor results when comparing accuracy for number of cells but this is merely because it takes many more triangles to fill space than other cells but results are cheap per triangle so when comparing accuracy for cost triangles give similar results to the meshes of quadrilaterals.

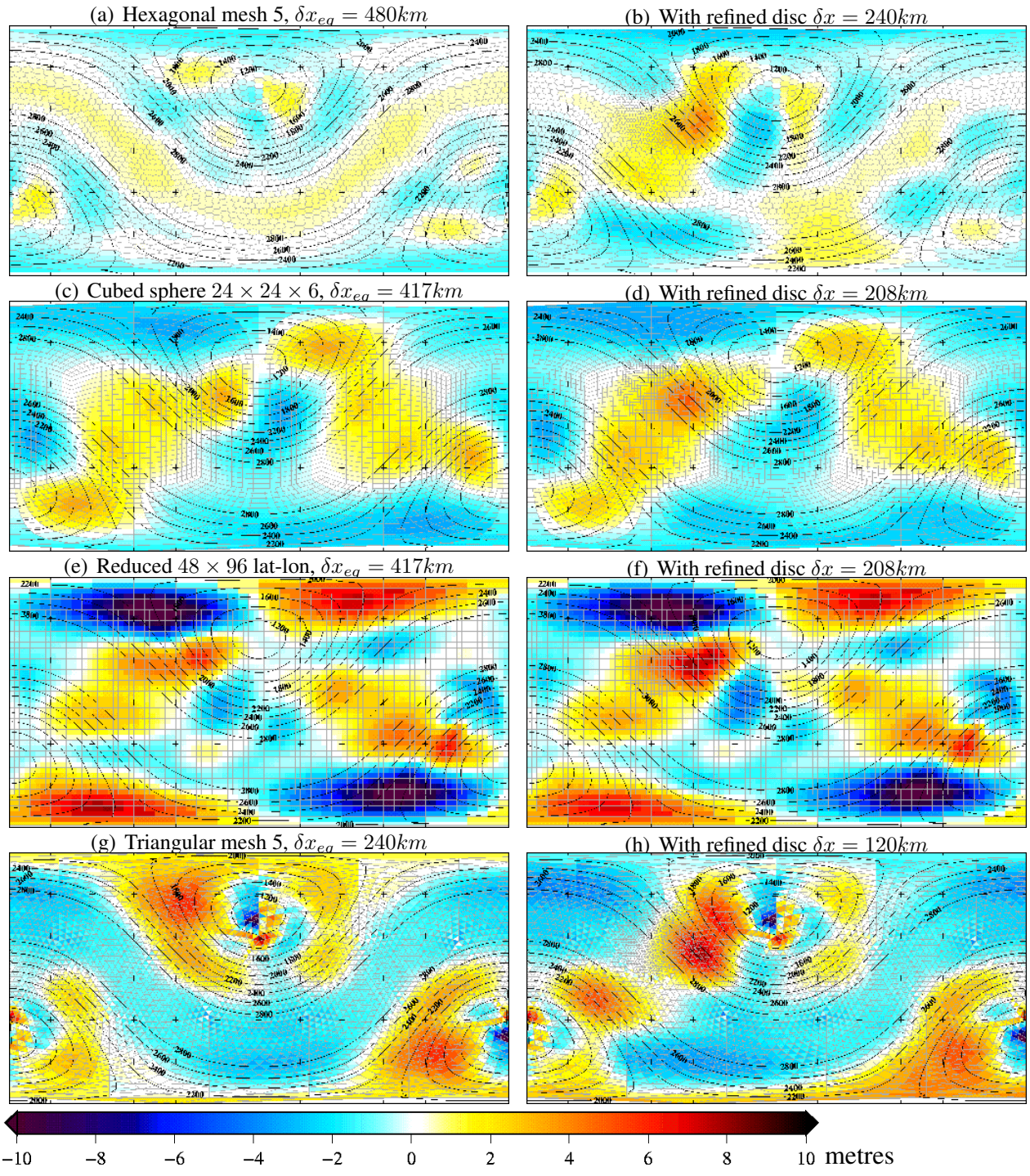

FIG. 4. Total height contours and shaded height errors after 5 days for global steady-state geostrophic flow rotated 45° from North (Williamson et al. 1992). Time step of 15 minutes for all meshes. + every 30° latitude and longitude.

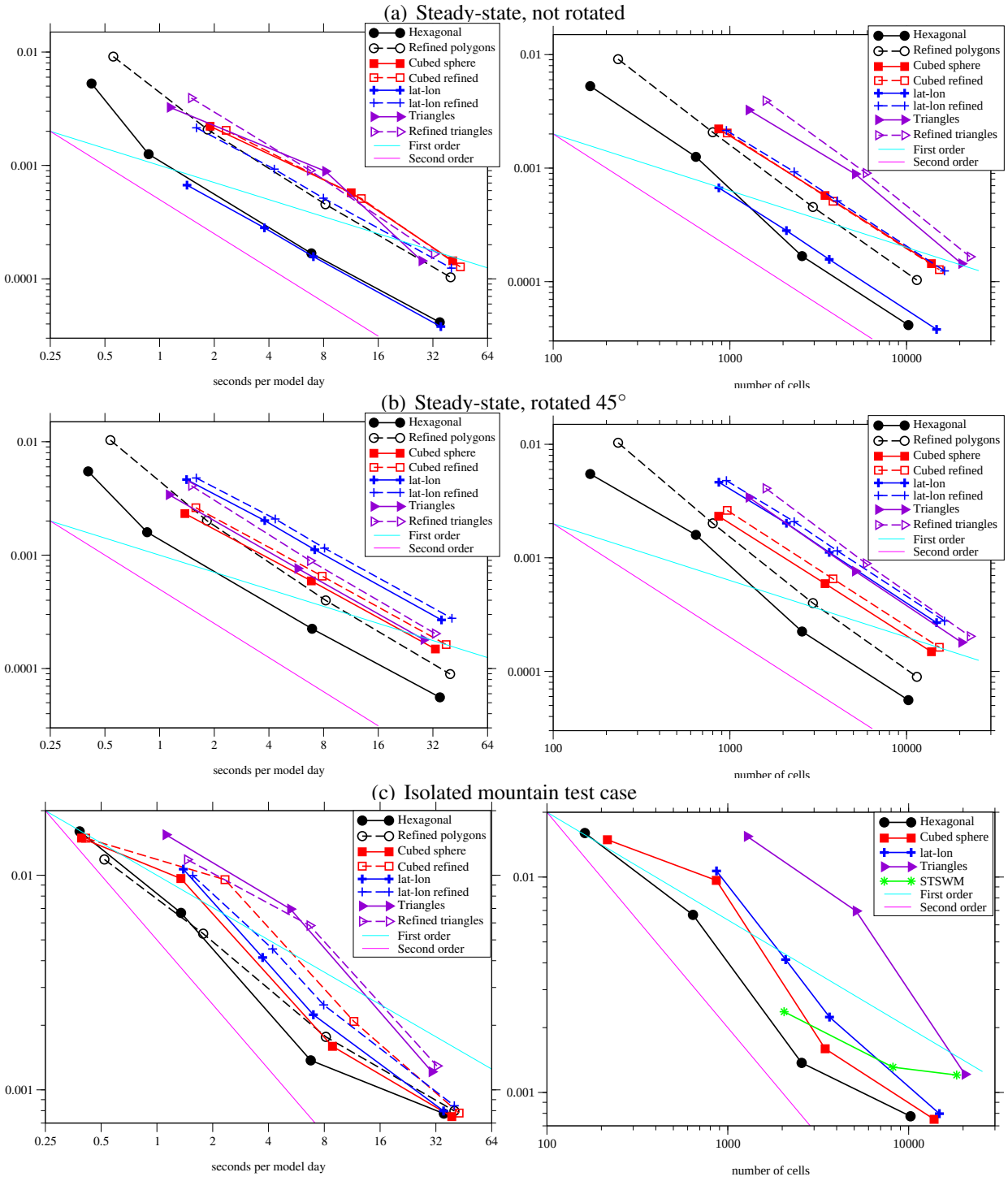

FIG. 5. Normalised root mean square or ℓ_2 error norms for the global steady-state geostrophic flow after 5 days and the flow over a mountain after 15 days (test cases 2 and 5 of Williamson et al. (1992)) as a function of CPU time and number of cells on various mesh structures.

When the flow is not rotated with respect to the geometric North pole, the reduced latitude-longitude mesh without a refinement pattern performs as well as the hexagonal mesh because additional errors are not introduced as the flow does not pass over the coarsening patterns, it travels parallel to them. However when rotated, reduced lat-lon meshes give the worst accuracy for cost. This demonstrates how results can be misleadingly good on test cases with flow aligned with the mesh.

The cost on each mesh structure using the quadratic and cubic fit schemes described in section 2c can be understood by considering the size of stencil necessary for the schemes. For upwinded differencing from cell centres onto face centres, 8 cells are included in the stencil on regular square parts of the meshes, 10 on regular hexagonal regions but 13 triangles are used so as to include a well placed upwind-upwind cell. This can make solution on triangles more expensive than solution on hexagons for meshes with the same number of faces (as opposed to the same number of cells). However meshes of hexagons lead to 7 non-zero elements per row of the matrix to be solved whereas triangles lead to 4 so the matrices resulting from the meshes of triangles will be cheaper to solve.

The stencil sizes can explain why meshes of hexagons are more cost effective than triangles. But based purely on the stencil size and number of non-zero elements per matrix row, meshes of squares should be more cost effective than hexagons. However, to mesh the surface of a sphere with squares, larger distortions must be introduced than using icosahedra, such as at cube corners or latitude-longitude coarsening boundaries. Alternatively for a full latitude-longitude mesh, very thin cells are used towards the poles, which increases cost. This then explains the superior behaviour of the hexagonal meshes, particularly over the rotated meshes of squares.

b. Williamson et al. (1992) Case 5: Flow over an Isolated Mountain

It may be beneficial to use local mesh refinement over orography. This is tested using the four mesh structures and different refinement patterns given in section 3, on the Williamson et al. (1992) test case 5; westerly flow over an isolated mountain. Much of the benefit of using high resolution for orography is in capturing small-scale diabatic effects such as orographically induced rainfall which are not present in the shallow-water equations.

Simulations on all meshes use a time-step of 15 minutes in order to compare with the spectral-model solution at triangular truncation 63 (T63) Jakob et al. (1993). This time-step gives very low flow C.F.L. numbers; no greater than 0.2 on the finest meshes. T63 corresponds to a grid-point resolution of 1.875° or 210km at the equator.

The reference solution and errors after 15 days are shown in figure 6 for the spectral-model (STSWM, Hack and Jakob 1992) at truncation T63 and for the eight AtmosFOAM meshes which have resolution similar to T63 but with and without double resolution around the mountain. The reference solution uses T426 (0.28° or 31km at the equator) and is calculated using a version of STSWM revised by Pilar Rípodas from Deutscher Wetterdienst
(`http://www.icon.enes.org/swm/stswm`). The errors shown in figure 6 are calculated as differences from the initial conditions in comparison to differences from the initial conditions of the reference solution. This means that errors in the initial conditions represented at lower resolution are not included which means that no spectral-ringing is seen in the spectral-model errors. The spectral-model results are interpolated onto the AtmosFOAM meshes using the bicubic interpolation code available also from Deutscher Wetterdienst.

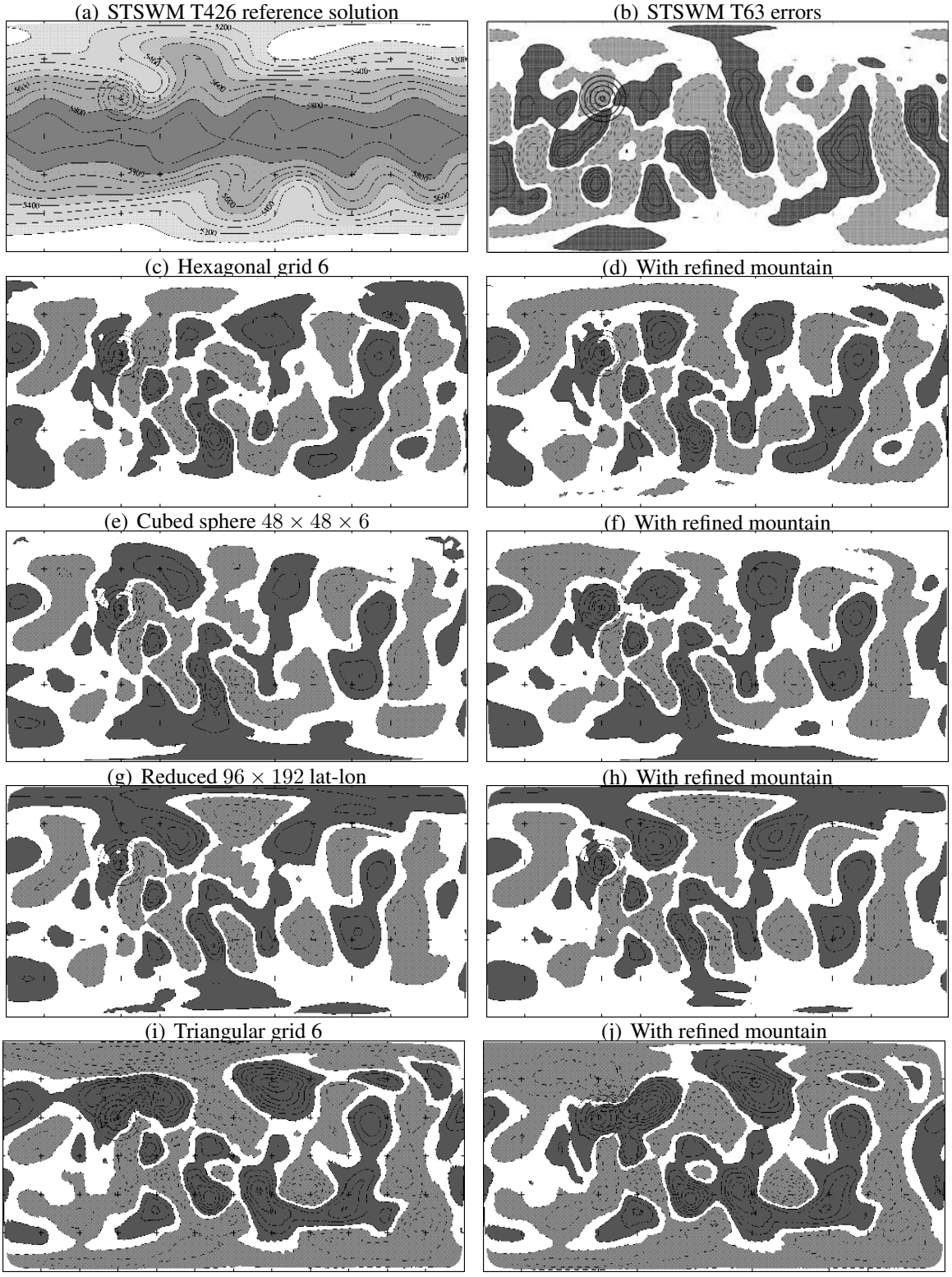

FIG. 6. Height errors after 15 days for the westerly flow over a mid-latitude mountain (Williamson et al. 1992). Time step of 15 minutes for all models. Errors are calculated as differences from initial conditions in comparison to STSWM T426 differences from initial conditions. Dark shaded contours from 2m every 4m. Light shaded dashed contours from -2m every -4m. $+$ every 30° latitude and longitude.

The most accurate AtmosFOAM results in figure 6 come from using uniform hexagonal or uniform cubed-sphere meshes. With these configurations, errors are similar to STSWM at T63. The reduced lat-lon mesh gives larger errors towards the North pole and the triangular mesh gives larger errors globally (despite having similar computational cost). Errors appear to increase slightly for all mesh structures when extra resolution is added around the mountain, and these increased errors come at the additional cost of computing the solution on more cells. Ringler et al. (2008) also found that local resolution does not benefit the global solution when solving this case with low-order finite-volumes.

In order to compare the accuracy for cost and accuracy for number of cells of the different mesh structures more rigorously, we plot the ℓ_2 error norm (Williamson et al. 1992) as a function of the CPU time and number of cells in figure 5(c) for different resolutions of each of the mesh structures. Lines showing first and second-order convergence are also shown. For the graph with respect to number of cells we also include the STSWM rerun at T21, T42 and T63 assuming 32×64, 64×128 and 96×192 grid points respectively. These are the number of grid points used at these truncations but typically spectral-transform models use 50% fewer grid points to achieve the same accuracy than a low-order finite-volume model (Williamson 2008). However the right hand side of figure figure 5(c) shows that AtmosFOAM achieves better accuracy than STSWM at T42 and T63 for the same number of cells for grid points. It is only at T21 that STSWM outperforms most of the AtmosFOAM meshes. AtmosFOAM results generally show second-order convergence with resolution, apart from at high resolution (high computational cost) where both AtmosFOAM and STSWM are converging more slowly to the T426 solution.

At low resolution (short CPU time) local refinement of the mountain gives better price performance only for the hexagonal and triangular meshes which include gradual rather than abrupt refinement. However, once resolution is increased, the mountain is sufficiently well resolved and the additional errors from the refinement pattern lead to inferior price performance.

The hexagonal mesh is most cost-effective, followed by the cubed-sphere, then the reduced latitude-longitude and then the triangular icosahedral mesh. The reduced latitude-longitude mesh does relatively well in this case because the flow is not rotated in comparison to the geometry and so much of the flow is aligned with the mesh. We have not plotted errors of STSWM as a function of CPU time on the same graph as the two codes were not run on the same computer and OpenFOAM is a 3D code running a 2D simulation so there is considerable overhead in taking the 2D projection of the 3D operators.

As would be expected of a spectral-model, the results are particularly accurate at low resolution. However as resolution is increased, the large scale features are adequately represented by the lower-order finite-volume model and the forcing from cone shaped mountain is also better represented.

Conservation of mass, total energy and potential vorticity have been compared with some other models and the results shown in table 2. Where possible the mesh and time-step that we use for AtmosFOAM matches the other published model. The AtmosFOAM conservation is within the range of the other models. Some of the smaller numbers will be influenced by roundoff error and different models were run with different machine precisions.

	Grid and Resolution	Space order	δt (mins)	Normalised Mass	Normalised Energy	Potential Vorticity (s^{-1})
AtmosFOAM	208 km	2	15	4.1×10^{-10}	-5.1×10^{-6}	3.2×10^{-11}
STSWM (Jakob et al. 1993)	lat-lon	T63		6.2×10^{-11}	4.6×10^{-6}	5.0×10^{-17}
AtmosFOAM	84 km	2	15	1.1×10^{-10}	-1.8×10^{-6}	6.8×10^{-21}
			0.75	-1.3×10^{-12}	1.2×10^{-6}	6.2×10^{-21}
Spectral elt. (Taylor et al. 1997)	cubed sphere	8	0.75	$< 10^{-8}$	-4×10^{-9}	$< 10^{-10}$
AtmosFOAM	250 km	2	15	1.6×10^{-11}	1.0×10^{-6}	5.3×10^{-11}
Spectral elt. (Thomas and Loft 2002)	cubed sphere	8	?	$< 10^{-11}$	$\sim 10^{-6}$	$\sim 10^{-17}$
AtmosFOAM	250 km	2	30	8.3×10^{-11}	-1.7×10^{-5}	1.3×10^{-10}
PV (Thuburn 1996)	hexagons	1-3		0	1×10^{-5}	2×10^{-8}
AtmosFOAM	120 km	2	15	-1.6×10^{-11}	-9.4×10^{-5}	9.1×10^{-11}
ICON (Bonaventura and Ringler 2005)	triangles	1-2		10^{-9}	-10^{-4}	5×10^{-7}
AtmosFOAM	156 km	2	15	4.7×10^{-11}	2.8×10^{-7}	4.2×10^{-11}
MMFV (Chen and Xiao 2008)	cubed sphere	4	?	0	-4×10^{-6}	-
D.G. (Läuter et al. 2008)	180 km	$k=6$	δt_{CFL}	0	5.5×10^{-8}	-

TABLE 2. Conservation errors for test case 5 (Williamson et al. 1992): shallow-water westerly flow over a mid-latitude mountain after 15 days. Values from other models are estimated from examination of published graphs and are therefore not precise.

c. Galewsky et al. (2004) Case: Barotropically Unstable Jet

This test case gives interesting results because the initial conditions are are very nearly in balance but physically unstable, and so the solution is sensitive to the initial conditions but also very sensitive to geometrically varying truncation errors. St-Cyr et al. (2008) present results of this test case without explicit diffusion using a spectral-element, sixth-order accurate model on a cubed-sphere mesh and a block-structured finite-volume on a latitude-longitude mesh, both with and without adaptive mesh refinement. Using both models on uniform meshes, good results are obtained once the resolution reaches 140km (1.25°). At coarser resolutions the spectral-element model creates a spurious "wavenumber-4 signal invoked by the computational cubed-sphere mesh" and the finite-volume model tends to smooth out the wave. The finite-volume model has the advantage that the mesh is aligned with the flow and so does not trigger any spurious barotropic instability.

We use this test case to examine the impact of local mesh refinement and different mesh structures on the triggering of barotropic waves. It is possible to adaptively refine meshes so that regions of high vorticity are always well resolved and the jet does not pass over changes in resolution as in St-Cyr et al. (2008) and Weller (2009), but this is not what we are doing. Instead we examine the influence of a statically refined disc which may be motivated by, for example the need for a local weather forecast or resolution of a locally produced tracer.

The vorticity after 6 days on the hexagonal icosahedron, cubed-sphere and reduced latitude-longitude meshes with and without a disc of local refinement of radius 25° centred at 30° North, 90° East are shown in figure 7. The base resolution for each mesh structure is 120-140km ($\approx 1.25°$) with 60-70km ($\approx 0.625°$) in the refined region. We also show results using the reduced latitude-longitude mesh rotated 30°. These are compared with an AtmosFOAM reference solution calculated on a 35km ($\approx 0.3125°$) reduced latitude-longitude mesh which is very similar to the published reference solution (Galewsky et al. 2004).

At 1.25° resolution, all of the meshes apart from the unrotated uniform latitude-longitude mesh produce spurious barotropic instability due to changes with longitude of the mesh structure. These are worst on the cubed-sphere mesh on which the spuriously triggered waves are larger than the wave at 270° longitude which is supposed to have grown the most. The problem with the unrotated cubed-sphere mesh when using this finite-volume model is that the jet tends to follow the mesh rather than lines of constant latitude. More satisfactorily, for both hexagonal meshes, the latitude-longitude mesh with refinement and the rotated latitude-longitude mesh, the spurious barotropic waves have grown less than real barotropic wave. These results highlight the lack of useful information from using this case to demonstrate the merits of a numerical method if results are only shown on a latitude-longitude mesh aligned with the flow. Also of note, only the unrotated reduced latitude-longitude mesh without refinement can maintain the zonal symmetry of balanced initial conditions without the perturbation to trigger the wave (not shown).

If maintaining steady-state but unstable jets at any orientation is important, then these results suggest that this is not possible with a second-order accurate model at moderate resulution. Higher-order accuracy or higher resolution must be used as for the fifth-order accurate spectral-element results presented by St-Cyr et al. (2008). However, Läuter et al. (2008) say of this case: "For the days 4-6 strong gradients develop in the regions of instability and benefit the low-order [schemes]".

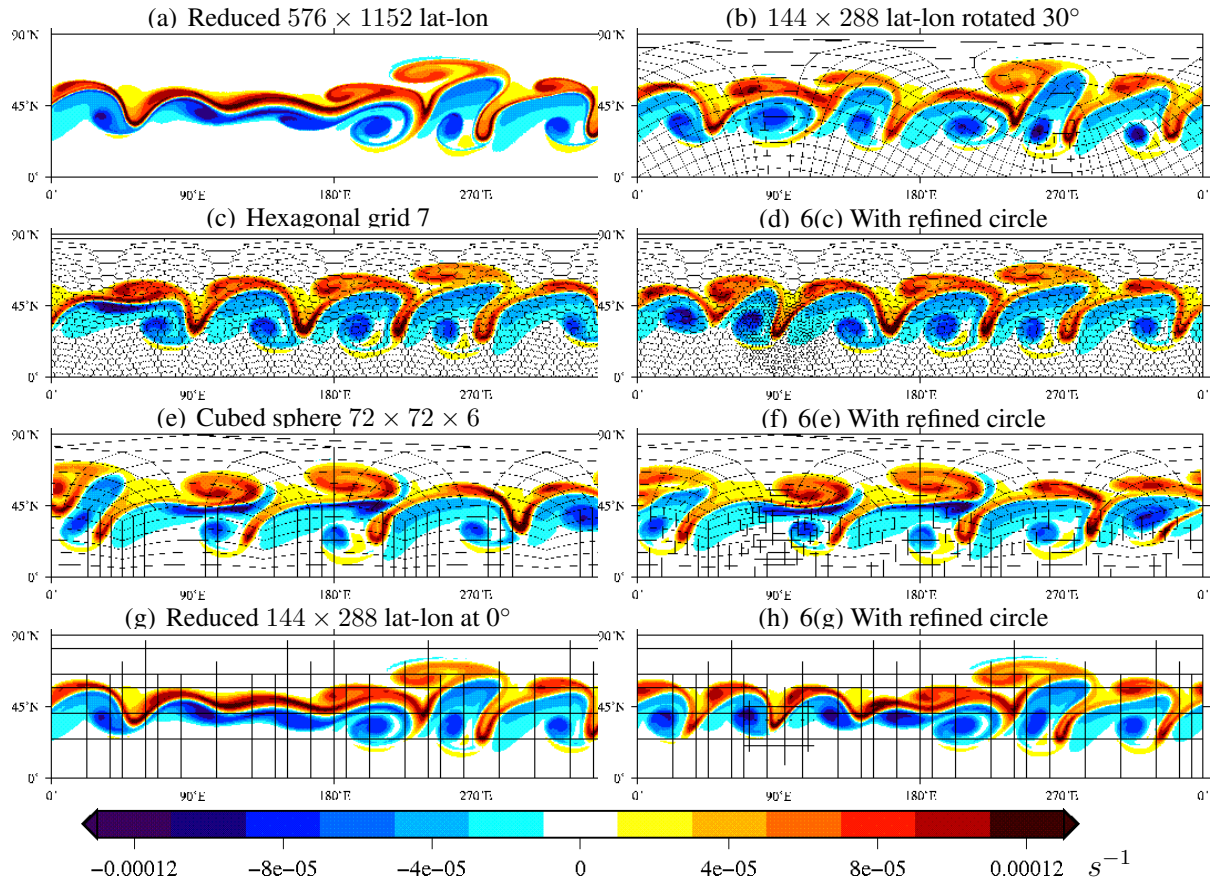

FIG. 7. Relative vorticity after 6 days for the barotropically unstable jet test case of Galewsky et al. (2004). Time step of 10 minutes for (c), (e), (b) and (g), 2.5 minutes for (a) and 5 minutes for the others. The lat-lon and cubed sphere meshes plotted are a factor of 6 coarser and the hexagonal meshes plotted are a factor of 4 coarser than those used to compute the solutions.

5. Summary and Conclusions

We have presented AtmosFOAM, which solves the rotating shallow-water equations on any mesh of the sphere. AtmosFOAM uses a second-order finite-volume formulation with prognostic variables of height and momentum colocated at cell centres. To remove grid-scale oscillations of the A-grid, the mass flux between cells is advanced from the old momentum using the momentum equation. Quadratic and upwind biased cubic differencing are used as explicit corrections to an implicit solution which uses linear differencing. This keeps the matrices involved in the implicit solution sparse and diagonally dominant, and hence cheap to solve while exploiting some accuracy gains of higher-order schemes.

Much of our analysis of these results is quite negative, but the AtmosFOAM results have also been compared with those of a selection of other published results, and AtmosFOAM compares well. For each published model we compare AtmosFOAM on a similar mesh and with the same or a longer time-step.

We present results of some standard test cases on the cubed-sphere and reduced latitude-longitude meshes, both with and without block-structured, static 2:1 refinement, and on hexagonal and triangular icosahedral meshes and Voronoi meshes of polygons and Delaunay meshes of triangles with gradual static local refinement. The latitude-longitude meshes give very good accuracy for cost when the flow is aligned with the mesh. Otherwise the hexagonal icosahedral meshes of polygons are the most cost effective. If a local reduction of truncation errors is required, then this can be achieved with local refinement, but this introduces errors where the mesh changes resolution, and this can actually degrade the global accuracy. This confirms the results of St-Cyr et al. (2008), who found increased global errors due to the inclusion of unnecessary refined mesh in a low-order finite-volume model. However this problem is alleviated when St-Cyr et al. (2008) test a higher-order spectral-element model.

We present solutions of the barotropically unstable jet test case of Galewsky et al. (2004) and find that mesh irregularities can trigger barotropic instability of a similar magnitude to those triggered by the initial perturbation, for the mesh resolution of about 1.25° that we use for this case. Using an unrotated reduced latitude-longitude mesh at 1.25° gives very good results for this case, but rotating the mesh, including a 2:1 refined region or using a hexagonal icosahedral mesh, triggers spurious waves nearly as big as the correctly initialised wave. When the barotropically unstable jet is discretised on a cubed-sphere, the initial jet very nearly follows the mesh lines. Subsequently, the jet tends to follow the cubed-sphere mesh lines which dip towards the equator at the cube corners. This triggers larger barotropic instabilities than the perturbations in the initial conditions.

Gradual Voronoi and Delaunay mesh refinement leads to better accuracy than abrupt block-structured 2:1 refinement. However our mesh-generating algorithms are limited in how gradual they can make the refinement. Factor of 2 increases in resolution tend to occur over two or three cells. We would like to create meshes of polygons with more gradual mesh grading, which might mean using spherical centroidal Voronoi tessellations or relaxing the Delaunay constraint to allow more anisotropic cells and refining based on vector rather than scalar criteria. This may improve the accuracy gains possible from including local mesh refinement.

Acknowledgements

HW acknowledges support from NCAS Climate and is particularly grateful to Julia Slingo for her support. We are grateful to OpenCFD for OpenFOAM and for parallelising some of the additional algorithms described here. Particular thanks to Mattijs Janssens at OpenCFD and Graham Macpherson at Strathclyde University. AF thanks M. Taylor for useful discussions and the University of Reading Department of Meteorology for Visiting Scientist support.

A. Delaunay Mesh Generation

Delaunay meshes in general consist of non-overlapping triangles in which the circle which goes through the three vertices of a triangle does not enclose a vertex of any other triangle. This leads to triangles which are as close as possible to equilateral for a given set of vertices, which may have computational advantages. Delaunay triangulations are computed using the Computational Geometry Algorithms Library (CGAL http://www.cgal.org).

The vertices and mesh are distributed on the sphere using a simple algorithm which appears to share some features of a spherical centroidal Voronoi tessellation (eg Ringler et al. 2008), but a direct comparison is the subject of future work. Our algorithm works as follows:

i. Start from an initial set of points and find the Delaunay triangulation using CGAL. At every step of the algorithm, for every point insertion, removal or movement, CGAL recomputes the Delaunay triangulation based on local changes.

ii. Read in a field describing the desired (isotropic) mesh spacing as a function of position over the globe. This controls the ultimate mesh spacing more directly than the density function as described in Ringler et al. (2008), although our technique may not be optimal.

iii. Loop through all the vertices on the sphere and remove some from regions which are too densely packed. This is decided as follows:

   - We consider vertex $\mathbf{v}$ where the desired resolution is δ_d
   - Vertex $\mathbf{v}$ is connected to n other vertices with edges of length δ_i. If $\frac{1}{n}\sum \delta_i^2 < 0.7\delta_d^2$ then vertex $\mathbf{v}$ is removed. (The factor 0.7 is empirically beneficial.)

iv. Loop through all the triangle edges on the sphere and add a vertex at the midpoint if the edge length, δ_i, is greater than $4/3$ times as long as the desired edge length, δ_d, at the edge midpoint. This criterion means that the edge lengths after adding vertices are always closer to δ_d. This loop is repeated until no more vertices are added.

v. Remove vertices which are only connected to exactly 4 other vertices. The vertices removed are associated with edges shorter than surrounding edges (although not short enough to be removed in step iii). This step removes squares from the Voronoi dual mesh.

vi. For vertices which are directly connected to 8 or more other vertices, add another vertex close to the vertex under consideration. This removes octagons and polygons with more sides from the Voronoi dual.

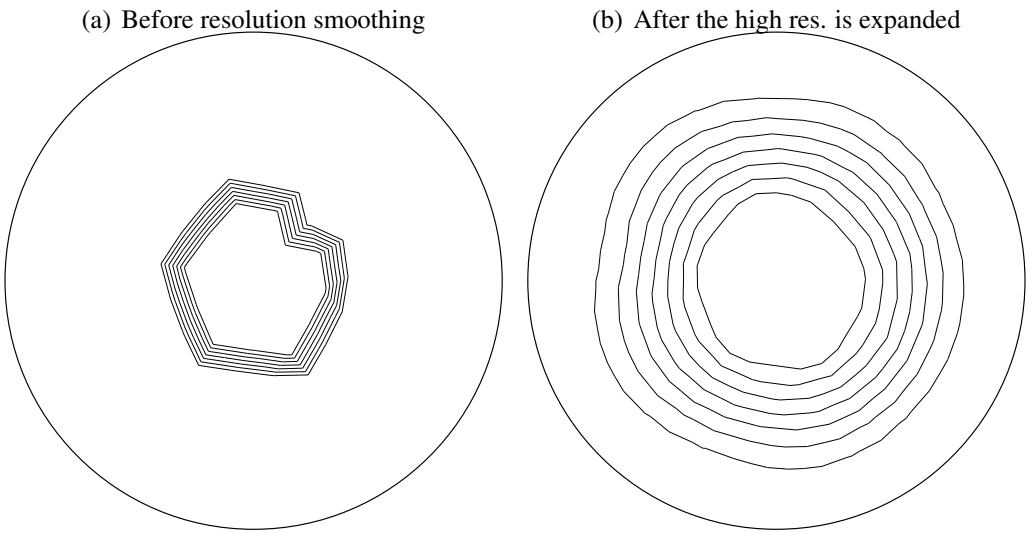

FIG. 8. The desired resolution distributions on hexagonal mesh grid 4 before refinement with contours at 350km to 650km every 100km for resolution centred at 30°North.

Procedures v and vi do not guarantee that there are no squares, octagons or polygons with more sides in the final Voronoi mesh as more can be created as others are removed. However in practice performing steps v and vi once does remove these shapes.

vii. Each triangle edge is assumed to be a critically damped spring with un-stretched length equal to the desired mesh resolution at the location of the centre of the edge. These springs undergo 5 relaxation iterations following Tomita et al. (2002).

For accuracy and stability of the shallow-water equation solver, we want meshes with gradual refinement. The mesh of triangles in figure 3(d) was created using this algorithm and a desired resolution field consisting of $\delta_d = 300km$ inside a disc of radius 20° and $\delta_d = 700km$ outside this radius (figure 8(a)) which is then expanded outwards and smoothed (figure 8(b)). The resulting mesh (figure 3(d)) is not as smooth as the desired resolution because of the tendency of the Delaunay algorithm to make equilateral triangles. Smoother grading might be achieved using a definition of the anisotropic mesh spacing and weighted Delaunay triangulation or by using spherical centroidal Voronoi tessellations.

REFERENCES

Almgren, A., J. Bell, P. Colella, L. Howell, and M. Welcome, 1998: A conservative adaptive projection method for the variable density incompressible Navier-Stokes equations. *J. Comput. Phys.*, **142 (1)**, 1–46.

Arakawa, A. and V. Lamb, 1977: Computational design of the basic dynamical processes of the UCLA general circulation model. *Methods in Computational Physics*, **17**, 173–265.

Bacon, D., et al., 2000: A dynamically adapting weather and dispersion model: The Operational Multiscale Environment Model with Grid Adaptivity (OMEGA). *Mon. Wea. Rev.*, **128 (7)**, 2044–2076.

Berger, M. and J. Oliger, 1984: Adaptive Mesh Refinement for Hyperbolic Partical Differential Equations. *J. Comput. Phys.*, **53**, 484–512.

Bonaventura, L. and T. Ringler, 2005: Analysis of discrete shallow-water models on geodesic Delaunay grids with C-type staggering. *Mon. Wea. Rev.*, **133**, 2351–2373.

CGAL, http://www.cgal.org: Computational Geometry Algorithms Library.

Chen, C. and F. Xiao, 2008: Shallow water model on a cubed-sphere by multi-moment finite volume method. *J. Comput. Phys.*, **227**, 5019–5044.

Côté, J., 1988: A Lagrange multiplier approach for the metric terms of semi-Lagrangian models on the sphere. *Quart. J. Roy. Meteor. Soc.*, **114**, 1347–1352.

Dietachmayer, G. and K. Droegemeier, 1992: Application of continuous dynamic grid adaption techniques to meteorological modeling. Part I: Basic formulation and accuracy. *Mon. Wea. Rev.*, **120 (8)**, 1675–1706.

Fiedler, B. and R. J. Trapp, 1993: A fast dynamic grid adaption scheme for meteorological flows. *Mon. Wea. Rev.*, **121 (10)**, 2879–2888.

Fournier, A., M. Taylor, and J. Tribbia, 2004: The spectral element atmosphere model (SEAM): High-resolution parallel computation and localized resolution of regional dynamics. *Mon. Wea. Rev.*, **132**, 726–748.

Galewsky, J., R. Scott, and L. Polvani, 2004: An initial-value problem for testing numerical models of the global shallow-water equations. *Tellus*, **56A (5)**, 429–440.

Hack, J. and R. Jakob, 1992: Description of a global shallow water model based on the spectral transform method. Tech. Rep. NCAR/TN-343+STR, National Centre for Atmospheric Research, Boulder Colorado.

Heikes, R. and D. Randall, 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. Wea. Rev.*, **123**, 1881–1997.

Heinze, T. and A. Hense, 2002: The shallow water equations on the sphere and their Lagrange-Galerkin-solution. *Meteorology and Atmospheric Physics*, **81 (1-2)**, 129–137.

Jablonowski, C., M. Herzog, J. Penner, R. Oehmke, Q. Stout, B. van Leer, and K. Powell, 2006: Block-structured adaptive grids on the sphere: Advection experiments. *Mon. Wea. Rev.*, **134 (12)**, 3691–3713.

Jablonowski, C. and D. Williamson, 2006: A baroclinic instability test case for atmospheric model dynamical cores. *Quart. J. Roy. Meteor. Soc.*, **132**, 2943–2975.

Jakob, R., J. Hack, and D. Williamson, 1993: Solutions to the shallow water test set using the spectral transform method. Technical note NCAR/TN-388+STR, Climate and Global Dynamics Division, National Center for Atmospheric Research, Boulder, Colorado.

Lanser, D., J. Blom, and J. Verwer, 2000: Spatial discretization of the shallow water equations in spherical geometry using Osher's scheme. *J. Comput. Phys.*, **165 (2)**, Pages 542–565.

Lashley, R., 2002: Automatic generation of accurate advection schemes on unstructured grids and their application to meteorological problems. Ph.D. thesis, Departments of Mathematics and Meteorology, University of Reading.

Läuter, M., F. Giraldo, D. Handorf, and K. Dethloff, 2008: A discontinuous Galerkin method for the shallow water equations in spherical triangular coordinates. *J. Comput. Phys.*, **227**, 10 226–10 242.

Läuter, M., D. Handorf, and D. K., 2005: Unsteady analytic solutions of the spherical shallow water equations. *J. Comput. Phys.*, **210**, 535–553.

Läuter, M., D. Handorf, N. Rakowsky, J. Behrens, S. Frickenhaus, M. Best, D. K., and W. Hiller, 2007: A parallel adaptive barotropic model of the atmosphere. *J. Comput. Phys.*, **223 (2)**, 609–628.

Pain, C., et al., 2005: Three-dimensional unstructured mesh ocean modelling. *Ocean Modelling*, **10 (1-2)**, 5–33.

Rhie, C. and W. Chow, 1983: Numerical study of the turbulent-flow past an airfoil with trailing edge separation. *AIAA Journal*, **21 (11)**, 1525–1532.

Ringler, T., L. Ju, and M. Gunzburger, 2008: A multiresolution method for climate system modeling: applications of spherical centroidal Voronoi tessellations. *Ocean Dynamics*, **Online first**.

Ruge, J., S. McCormick, and S. Yee, 1995: Multilevel adaptive methods for semi-implicit solution of shallow-water equations on a sphere. *Mon. Wea. Rev.*, **123 (7)**, 2197–2205.

Skamarock, W. and J. Klemp, 1993: Adaptive grid refinement for two-dimensional and three-dimensional nonhydrostatic atmospheric flow. *Mon. Wea. Rev.*, **121 (3)**, 788–804.

Skamarock, W., J. Oliger, and R. Street, 1989: Adaptive grid refinement for numerical weather prediction. *J. Comput. Phys.*, **80 (1)**, 27–60.

St-Cyr, A., C. Jablonowski, J. Dennis, H. Tufo, and S. Thomas, 2008: A comparison of two shallow-water models with nonconforming adaptive grids. *Mon. Wea. Rev.*, **136**, 1898–1922.

Staniforth, A. and H. Mitchell, 1978: Variable-resolution finite-element technique for regional forecasting with primitive equations. *Mon. Wea. Rev.*, **106 (4)**, 439–447.

Stevens, D. and S. Bretherton, 1996: A forward-in-time advection scheme and adaptive multilevel flow solver for nearly incompressible atmospheric flow. *J. Comput. Phys.*, **129 (2)**, 284–295.

Taylor, M., J. Tribbia, and M. Iskandarani, 1997: The spectral element method for the shallow water equations on the sphere. *J. Comput. Phys.*, **130 (1)**, 92–108.

Thomas, S. and R. Loft, 2002: Semi-implicit spectral element atmospheric model. *Journal of Scientific Computing*, **17 (1-4)**, 339–350.

Thuburn, J., 1996: A PV-based shallow water model on a hexagonal-icosahedral grid. Tech. Rep. 40, UGAMP.

Thuburn, J., 1997: A PV-based shallow-water model on a hexagonal-icosahedral grid. *Mon. Wea. Rev.*, **125 (9)**, 2328–2347.

Tomita, H., M. Satoh, and K. Goto, 2002: An optimization of the icosahedral grid modified by spring dynamics. *J. Comput. Phys.*, **183 (1)**, 307–331.

Vichnevetsky, R., 1987: Wave propagation and reflection in irregular grids for hyperbolic equations. *Applied Numerical Mathematics*, **3**, 133–166.

Vichnevtsky, R. and L. Turner, 1991: Spurious scattering from discontinously stretching grids in computational fluid dynamics. *Applied Numerical Mathematics*, **8**, 289–299.

Weller, H., 2009: Predicting mesh density for adaptive modelling of the global atmosphere. *Phil. Trans. R. Soc. A*, **invited article to a special issue (to appear)**.

Weller, H. and H. Weller, 2008: A high-order arbitrarily unstructured finite-volume model of the global atmosphere: tests solving the shallow-water equations. *Int. J. Numer. Meth. Fluids*, **56 (8)**, 1589–1596.

Williamson, D., 2008: Equivalent finite volume and Eulerian spectral transform horizontal resolutions established from aqua-planet simulations. *Tellus*, **60A**, 839–847.

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