Journal of Computational Physics 344 (2017) 86–107

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Multidimensional method-of-lines transport for atmospheric flows over steep terrain using arbitrary meshes

James Shaw [a,*], Hilary Weller [a], John Methven [a], Terry Davies [b]

[a] Department of Meteorology, University of Reading, Reading, United Kingdom
[b] Met Office, Exeter, United Kingdom

ARTICLE INFO

Article history:
Received 2 February 2017
Received in revised form 21 April 2017
Accepted 23 April 2017
Available online 2 May 2017

Keywords:
Finite volume
Unstructured mesh
Atmospheric modelling
Least-squares approximation
von Neumann stability analysis

ABSTRACT

Including terrain in atmospheric models gives rise to mesh distortions near the lower boundary that can degrade accuracy and challenge the stability of transport schemes. Multidimensional transport schemes avoid splitting errors on distorted, arbitrary meshes, and method-of-lines schemes have a low computational cost because they perform reconstructions at fixed points.

This paper presents a multidimensional method-of-lines finite volume transport scheme, "cubicFit", which is designed to be numerically stable on arbitrary meshes. Stability conditions derived from a von Neumann stability analysis are imposed during model initialisation to obtain stability and improve accuracy in distorted regions of the mesh, and near steeply-sloping lower boundaries. Reconstruction calculations depend upon the mesh only, needing just one vector multiply per face per time-stage irrespective of the velocity field.

The cubicFit scheme is evaluated using three, idealised numerical tests. The first is a variant of a standard horizontal transport test on severely distorted terrain-following meshes. The second is a new test case that assesses accuracy near the ground by transporting a tracer at the lower boundary over steep terrain on terrain-following meshes, cut-cell meshes, and new, slanted-cell meshes that do not suffer from severe time-step constraints associated with cut cells. The third, standard test deforms a tracer in a vortical flow on hexagonal–icosahedral meshes and cubed-sphere meshes. In all tests, cubicFit is stable and largely insensitive to mesh distortions, and cubicFit results are more accurate than those obtained using a multidimensional linear upwind transport scheme. The cubicFit scheme is second-order convergent regardless of mesh distortions.

1. Introduction

Numerical simulations of atmospheric flows solve equations of motion that result in the transport of momentum, temperature, water species and trace gases. The numerical representation of Earth's terrain complicates the transport problem because the mesh is necessarily distorted next to the lower boundary. As new atmospheric models use increasingly fine mesh spacing, meshes are able to resolve steep, small-scale slopes. Numerical schemes in operational weather forecast models can perform poorly over large mountain ranges, exhibiting small-scale numerical noise in momentum [1], temperature,

* Corresponding author.
E-mail address: js102@zepler.net (J. Shaw).

http://dx.doi.org/10.1016/j.jcp.2017.04.061

humidity [2] and potential vorticity fields [3], or even violating the Courant–Friedrich–Lewy stability constraint resulting in so-called 'grid-point storms' [4]. A transport scheme is desired that yields stable and accurate solutions, particularly near the surface where the weather affects us directly. We present a new transport scheme which is numerically stable on arbitrary meshes and which is generally insensitive to mesh distortions created by steep slopes. It has a low computational cost since most calculations are not repeated every time-step because they depend upon the mesh geometry only.

There are two main methods for representing terrain in atmospheric models: terrain-following layers and cut cells. Both methods modify regular meshes to produce distorted meshes with cells that are aligned in columns. Most operational models use terrain-following layers in which horizontal mesh surfaces are moved upwards to accommodate the terrain. A vertical decay function is chosen so that mesh surfaces slope less steeply with increasing height. The most straightforward is the linear decay function used by the basic terrain-following transform [5] (also called the sigma coordinate), but many atmospheric models suffer from large numerical errors on such meshes [2,6,7]. To reduce such errors, more complex decay functions have been developed so that mesh surfaces are smoother [8,2,9,6].

An alternative to terrain-following layers is the cut cell method. Cut cell meshes are constructed by 'cutting' a regular mesh with a piecewise-linear representation of the terrain. New vertices are created where the terrain intersects mesh edges, and cell volumes that lie beneath the ground are removed. Cut cell meshes can have arbitrarily small cells that impose severe time-step constraints on explicit transport schemes. Several techniques have been developed to alleviate this problem, known as the 'small-cell problem': small cells can be merged with adjacent cells [10], cell volumes can be artificially increased [11], or an implicit scheme can be used near the ground with an explicit scheme used aloft [12].

Another method for avoiding the small-cell problem was proposed by Shaw and Weller [13] in which cell vertices are moved vertically so that they are positioned at the terrain surface. We refer to this alternative method as the slanted cell method in order to distinguish it from the traditional cut cell method. Slanted cell meshes do not suffer from arbitrarily small cells because the horizontal cell dimensions are not modified by the presence of terrain.

Smoothed terrain-following layers, cut cells and slanted cell methods all reduce the amount of mesh distortion but any mesh that represents sloping terrain must necessarily be distorted, at least near the ground. Even when distortions are minimal, transport across mesh surfaces tends to be more common near steep slopes, and this misalignment between the flow and mesh surfaces increases numerical errors [14,2,13]. A huge variety of transport schemes have been developed for atmospheric models, but few are able to account for distortions associated with steep terrain because they treat horizontal and vertical transport separately [15], resulting in numerical errors called 'splitting errors'. Such errors can be reduced by explicitly accounting for transverse fluxes when combining fluxes [16], but splitting errors are still apparent in flows over steep terrain where meshes are highly distorted and metric terms in a terrain-following coordinate transform are large [17].

Transport schemes are often classified as dimensionally-split or multidimensional. Dimensionally-split schemes such as [18,19] calculate transport in each dimension separately before the flux contributions are combined. Such schemes are computationally efficient and allow existing one-dimensional high-order methods to be used. When dimensionally-split schemes are used for horizontal transport, quadrilateral meshes are needed because the mesh dimensions are inherently separable. Special treatment is required at the corners of cubed-sphere panels where local coordinates differ [20,19]. Dimensional splitting is often used for vertical transport and, for similar reasons, dimensionally-split schemes have only been used with terrain-following coordinate transforms and not cut cells. Perhaps confusingly, dimensionally-split schemes are sometimes called multidimensional, too, because they use one-dimensional techniques for multidimensional transport.

Unlike dimensionally-split schemes, multidimensional schemes consider transport in two or three dimensions together. There are several subclasses of multidimensional schemes that include semi-Lagrangian finite volume schemes (also called conservative mesh remapping), swept-area schemes (also called flux-form semi-Lagrangian, incremental remapping, or forward-in-time), and method-of-lines schemes (also called Eulerian schemes). Two-dimensional semi-Lagrangian finite volume schemes such as [21,22] integrate over departure cells that are found by tracing backward the trajectories of cell vertices. These schemes are conservative because departure cells are constructed so that there are no overlaps or gaps, which requires that cell areas are simply-connected domains [23]. SLICE-3D is a three-dimensional semi-Lagrangian finite volume scheme for latitude–longitude meshes that applies separate conservative remappings in each dimension [24]. Swept area schemes such as [25–28] calculate the flux through a cell face by integrating over the upstream area that is swept out over one time-step. Such schemes differ in their choice of area approximation, sub-grid reconstruction, and spatial integration method. Because swept area schemes integrate over the reconstructed field, they typically require a matrix–vector multiply per face per time-stage [28,26]. Method-of-lines schemes such as [29,30] use a spatial discretisation to reduce the transport PDE to an ODE that is typically solved using a multi-stage time-stepping method. A method-of-lines scheme using a spectral element reconstruction was recently developed to achieve accurate solutions near the surface of cut cell meshes [31]. Unlike semi-Lagrangian finite volume schemes, swept-area and method-of-lines schemes achieve conservation for small-scale rotational flows. Such flows can twist the departure domain to such an extent that the domain intersects itself [27]. In two dimensions, a self-intersecting departure domain has a bowtie or hourglass shape. There are many more types of atmospheric transport schemes, but all can be classified according to their treatment of the three spatial dimensions. A more comprehensive overview is presented by Lauritzen et al. [32].

For transport schemes that are ordinarily classified as 'multidimensional', a further distinction ought to made between horizontally-multidimensional and three-dimensional schemes. Most multidimensional schemes are only horizontally-multidimensional because, while the two horizontal dimensions are considered together, horizontal and vertical transport are still treated separately. This separate treatment becomes less justifiable as atmospheric models are using increasingly fine

horizontal mesh spacings that resolve small-scale steep slopes, resulting in greater mesh distortion and possible splitting errors [15]. Three-dimensional schemes avoid any splitting errors over steep slopes, but only a few conservative three-dimensional schemes have been used in atmospheric models. The multi-moment constrained finite volume scheme [33] is a three-dimensional scheme that has been used to simulate nonhydrostatic flows over orography with terrain-following coordinates on a x–z plane [34]. Simulations of subcritical flow around a cylinder have also been performed on a three-dimensional hexahedral–prismatic hybrid mesh [35]. The Multidimensional Positive Definite Advection Transport Algorithm (MPDATA) is another three-dimensional scheme that is suitable for arbitrary meshes. It has been used on triangular unstructured meshes to simulate two-dimensional nonhydrostatic flows over orography [36], and in three-dimensional transport tests [37]. Most recently, MPDATA has been extended to enable semi-implicit integrations of the compressible Euler equations on arbitrary meshes [38]. The three-dimensional method-of-lines scheme developed by Weller and Shahrokhi [39] has been used in two-dimensional flows over orography on Cartesian x–z planes with distorted meshes [13,17]. This finite volume scheme uses a moving least-squares reconstruction that makes it suitable for arbitrary meshes. This least-squares approach has been applied previously to shallow water flows [40], aeronautic [41] and porous media [42] simulations.

In this paper, we present a new multidimensional method-of-lines scheme, "cubicFit", that improves the stability of the Weller and Shahrokhi scheme [39] and avoids all splitting errors. To reconstruct values at cell faces, the scheme fits a multidimensional cubic polynomial over an upwind-biased stencil using a least-squares approach. The implementation uses stability conditions derived from a von Neumann stability analysis to select appropriate polynomial fits for stencils in highly-distorted mesh regions. Almost all of the least-squares procedure depends upon the mesh geometry only and reconstruction weights can be precomputed without knowledge of the velocity field or tracer field. Hence, the computational cost of the cubicFit scheme is lower than most swept-area schemes that require a matrix–vector multiply per face per time-stage. Instead, the computational cost is more comparable to dimensionally-split schemes, with the cubicFit scheme requiring only n multiplies per face per time-stage where n is the size of the stencil. Based on numerical experiments, the scheme is found to be conditionally stable up to maximum Courant numbers of about 1.3 to 3.3.

The remainder of this paper is organised as follows. Section 2 starts by discretising the transport equation using a method-of-lines approach before describing the cubicFit transport scheme and a multidimensional linear upwind transport scheme. Section 3 evaluates the cubicFit scheme using three idealised numerical tests. The first test follows Schär et al. [2], transporting a tracer horizontally above steep mountains on two-dimensional, highly-distorted terrain-following meshes. The second is a new test case designed to assess numerical accuracy next to a mountainous lower boundary. In this test, a tracer placed at the ground is transported over steep slopes by a terrain-following velocity field on terrain-following, cut cell and slanted cell meshes. The third is a standard test of deformational flow on a single-layer spherical Earth, specified by Lauritzen et al. [43], which we use to assess the cubicFit transport scheme on hexagonal–icosahedral meshes and cubed-sphere meshes. Concluding remarks are made in section 4.

2. Transport schemes for arbitrary meshes

The transport of a dependent variable ϕ in a prescribed, non-divergent velocity field $\mathbf{u}$ is given by the equation

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{u}\phi) = 0\,. \tag{1}$$

The time derivative is discretised using a two-stage, second-order Heun method,

$$\phi^{\star} = \phi^{(n)} + \Delta t\, g(\phi^{(n)}) \tag{2a}$$

$$\phi^{(n+1)} = \phi^{(n)} + \frac{\Delta t}{2}\left[g(\phi^{(n)}) + g(\phi^{\star})\right] \tag{2b}$$

where $g(\phi^{(n)}) = -\nabla \cdot (\mathbf{u}\phi^{(n)})$ at time level n. The same time-stepping method is used for both the cubicFit scheme and the multidimensional linear upwind scheme. Although the Heun method is unstable for a linear oscillator [44] and for solving the transport equation using centred, linear differencing, it is stable when it is used for transport schemes with sufficient upwinding.

Using the finite volume method, the velocity field is prescribed at face centroids and the dependent variable is stored at cell centroids. The divergence term in equation (1) is discretised using Gauss's theorem,

$$\nabla \cdot (\mathbf{u}\phi) \approx \frac{1}{\mathcal{V}_c} \sum_{f \in c} \mathbf{u}_f \cdot \mathbf{S}_f \phi_F \tag{3}$$

where subscript f denotes a value stored at a face and subscript F denotes a value approximated at a face using surrounding values. $\mathcal{V}_c$ is the cell volume, $\mathbf{u}_f$ is a velocity vector prescribed at a face, $\mathbf{S}_f$ is the surface area vector with a direction outward normal to the face and a magnitude equal to the face area, ϕ_F is an approximation of the dependent variable at the face, and $\sum_{f \in c}$ denotes a summation over all faces f bordering cell c. Note that equation (3) is a second-order approximation of the divergence term which limits the cubicFit transport scheme to second-order numerical convergence.

This discretisation is applicable to arbitrary meshes. A necessary condition for stability is given by the multidimensional Courant number,

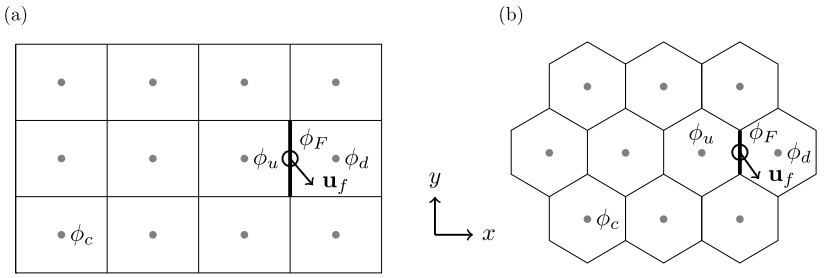

Fig. 1. Upwind-biased stencils for faces far away from the boundaries of two-dimensional (a) rectangular and (b) hexagonal meshes. The stencil is used to fit a multidimensional polynomial to cell centre values, ϕ_c, marked by grey circles, in order to approximate the value ϕ_F at the face centroid marked by an open circle. ϕ_u and ϕ_d are the values at the centroids of the upwind and downwind cells neighbouring the target face, drawn with a heavy line. The velocity vector $\mathbf{u}_f$ is prescribed at face f and determines the choice of stencil at each time-step.

$$\mathrm{Co}_c = \frac{\Delta t}{2\mathcal{V}_c} \sum_{f \in c} |\mathbf{u} \cdot \mathbf{S}_f| \tag{4}$$

such that, for all cells c in the domain, Co_c is less than or equal to some constant that depends upon the spatial and temporal discretisation. Hence, stability is constrained by the maximum Courant number of any cell in the domain.

The accurate approximation of the dependent variable at the face, ϕ_F, is key to the overall accuracy of the transport scheme. The cubicFit scheme and multidimensional linear upwind scheme differ in their approximations, and these approximation methods are described next.

2.1. Cubic fit transport scheme

The cubicFit scheme approximates the value of the dependent variable at the face, ϕ_F, using a least-squares fit over a stencil of surrounding known values. To introduce the approximation method, we will consider how an approximate value is calculated for a face that is far away from the boundaries of a two-dimensional uniform rectangular mesh. For any mesh, every interior face connects two adjacent cells. The velocity direction at the face determines which of the two adjacent cells is the upwind cell. Since the stencil is upwind-biased and asymmetric, two stencils must be constructed for every interior face, and the appropriate stencil is chosen depending on the velocity direction at each face for every time-step.

The upwind-biased stencil for a face f is shown in Fig. 1(a). The wind at the face, $\mathbf{u}_f$, is blowing from the upwind cell c_u to the downwind cell c_d. To obtain an approximate value at f, a polynomial least-squares fit is calculated using the stencil values. The stencil has 4 points in x and 3 points in y, leading to a natural choice of polynomial that is cubic in x and quadratic in y,

$$\phi = a_1 + a_2 x + a_3 y + a_4 x^2 + a_5 xy + a_6 y^2 + a_7 x^3 + a_8 x^2 y + a_9 x y^2 \,. \tag{5}$$

A least-squares approach is needed because the system of equations is overconstrained, with 12 stencil values but only 9 polynomial terms. The stencil geometry is expressed in a local coordinate system with the face centroid as the origin so that the approximated value ϕ_F is equal to the constant coefficient a_1. The stencil is upwind-biased to improve numerical stability, and the multidimensional cubic polynomial is chosen to improve accuracy in the direction of flow [14].

The remainder of this subsection generalises the approximation technique for arbitrary meshes and describes the methods for constructing stencils, performing a least-squares fit with a suitable polynomial, and ensuring numerical stability of the transport scheme.

2.1.1. Stencil construction

For every interior face, two stencils are constructed, one for each of the possible upwind cells. Stencils are not constructed for boundary faces because values of ϕ at boundaries are calculated from prescribed boundary conditions. For a given interior face f and upwind cell c_u, we find those faces that are connected to c_u and 'oppose' face f. These are called the *opposing faces*. The opposing faces for face f and upwind cell c_u are determined as follows. Defining G to be the set of faces other than f that border cell c_u, we calculate the 'opposedness', Opp, between faces f and $g \in G$, defined as

$$\mathrm{Opp}(f, g) \equiv -\frac{\mathbf{S}_f \cdot \mathbf{S}_g}{|\mathbf{S}_f|^2} \tag{6}$$

where $\mathbf{S}_f$ and $\mathbf{S}_g$ are the surface area vectors pointing outward from cell c_u for faces f and g respectively. Using the fact that $\mathbf{a} \cdot \mathbf{b} = |\mathbf{a}|\,|\mathbf{b}| \cos(\theta)$ we can rewrite equation (6) as

$$\mathrm{Opp}(f, g) = -\frac{|\mathbf{S}_g|}{|\mathbf{S}_f|} \cos(\theta) \tag{7}$$

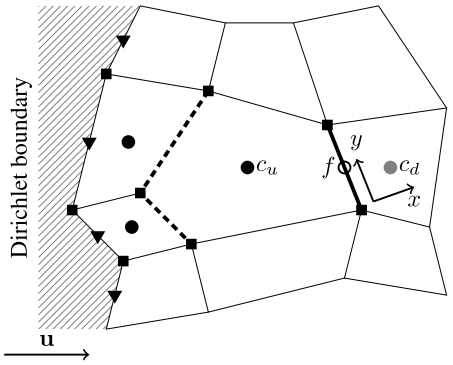

Fig. 2. A fourteen-point, upwind-biased stencil for face f connecting the pentagonal upwind cell, c_u, and the downwind cell c_d. The dashed lines denote the two faces of cell c_u that oppose f, and black circles mark the centroids of the internal cells that are connected to these two opposing faces. The stencil is extended outwards by including cells that share vertices with the three internal cells, where black squares mark these vertices. Four stencil boundary faces, marked by black triangles, are also included. The local coordinate system (x, y) has its origin at the centroid of face f, marked by an open circle, with x normal to f and y perpendicular to x.

where θ is the angle between faces f and g. In this form, it can be seen that Opp is a measure of the relative area of g and how closely it parallels face f.

The set of opposing faces, OF, is a subset of G, comprising those faces with $\text{Opp} \geq 0.5$, and the face with the maximum opposedness. Expressed in set notation, this is

$$\text{OF}(f, c_u) \equiv \{g : \text{Opp}(f, g) \geq 0.5\} \cup \{g : \max_{g \in G}(\text{Opp}(f, g))\} . \tag{8}$$

On a rectangular mesh, there is always one opposing face g, and it is exactly parallel to the face f such that $\text{Opp}(f, g) = 1$.

Once the opposing faces have been determined, the set of internal and external cells must be found. The *internal cells* are those cells that are connected to the opposing faces. Note that c_u is always an internal cell. The *external cells* are those cells that share vertices with the internal cells. Note that c_d is always an external cell. Finally, the *stencil boundary faces* are boundary faces having Dirichlet boundary conditions[1] that share a vertex with the internal cells. Having found these three sets, the stencil is constructed to comprise all internal cells, external cells and stencil boundary faces.

Fig. 2 illustrates a stencil construction for face f connecting upwind cell c_u and downwind cell c_d. The two opposing faces are denoted by thick dashed lines and the centres of the three adjoining internal cells are marked by black circles. The stencil is extended outwards by including the external cells that share vertices with the internal cells, where the vertices are marked by black squares. A boundary at the far left has Dirichlet boundary conditions, and so the four stencil boundary faces are also included in the stencil, where the boundary face centres are marked by black triangles. The resultant stencil contains fourteen points.

2.1.2. Least-squares fit

To approximate the value of ϕ at a face f, a least-squares fit is calculated from a stencil of surrounding known values. First, we will show how a polynomial least-squares fit is calculated for a face on a rectangular mesh. Second, we will make modifications to the least-squares fit that are necessary for numerical stability.

For faces that are far away from the boundaries of a rectangular mesh, we fit the multidimensional polynomial given by equation (5) that has nine unknown coefficients, $\mathbf{a} = a_1 \ldots a_9$, using the twelve cell centre values from the upwind-biased stencil, $\boldsymbol{\phi} = \phi_1 \ldots \phi_{12}$. This yields a matrix equation

$$\begin{bmatrix} 1 & x_1 & y_1 & x_1^2 & x_1 y_1 & y_1^2 & x_1^3 & x_1^2 y_1 & x_1 y_1^2 \\ 1 & x_2 & y_2 & x_2^2 & x_2 y_2 & y_2^2 & x_2^3 & x_2^2 y_2 & x_2 y_2^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{12} & y_{12} & x_{12}^2 & x_{12} y_{12} & y_{12}^2 & x_{12}^3 & x_{12}^2 y_{12} & x_{12} y_{12}^2 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_9 \end{bmatrix} = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{12} \end{bmatrix} \tag{9}$$

which can be written as

$$\mathbf{B}\mathbf{a} = \boldsymbol{\phi} . \tag{10}$$

[1] Boundary faces with Neumann boundary conditions would require extrapolated boundary values to be calculated. This would create a feedback loop in which boundary values are extrapolated from interior values, then interior values are transported using stencils that include boundary values. We have not considered how such an extrapolation could be made consistent with the multidimensional polynomial reconstruction. Hence, boundary faces with Neumann boundary conditions are excluded from the set of stencil boundary faces.

The rectangular matrix $\mathbf{B}$ has one row for each cell in the stencil and one column for each term in the polynomial. $\mathbf{B}$ is called the *stencil matrix*, and it is constructed using only the mesh geometry. A local coordinate system is established in which x is normal to the face f and y is perpendicular to x. The coordinates (x_i, y_i) give the position of the centroid of the ith cell in the stencil. A two-dimensional stencil is also used for the tests on spherical meshes in section 3.3. In these tests, cell centres are projected perpendicular to a tangent plane at the face centre. Previous studies found that results were largely insensitive to the projection method [30,25].

The unknown coefficients $\mathbf{a}$ are calculated using the pseudo-inverse, $\mathbf{B}^+$, found by singular value decomposition,

$$\mathbf{a} = \mathbf{B}^+ \boldsymbol{\phi} \,. \tag{11}$$

Recall that the approximate value ϕ_F is equal to the constant coefficient a_1, which is a weighted mean of $\boldsymbol{\phi}$,

$$a_1 = \begin{bmatrix} b^+_{1,1} \\ b^+_{1,2} \\ \vdots \\ b^+_{1,12} \end{bmatrix} \cdot \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{12} \end{bmatrix} \tag{12}$$

where the weights $b^+_{1,1} \dots b^+_{1,12}$ are the elements of the first row of $\mathbf{B}^+$. Note that the majority of the least-squares fit procedure depends on the mesh geometry only. An implementation may precompute the pseudo-inverse for each stencil during model initialisation, and only the first row needs to be stored. Since each face has two possible stencils depending on the orientation of the velocity relative to the face, the implementation stores two sets of weights for each face. Knowledge of the values of $\boldsymbol{\phi}$ is only required to calculate the weighted mean given by equation (12), which is evaluated once per face per time-stage.

In the least-squares fit presented above, all stencil values contributed equally to the polynomial fit. It is necessary for numerical stability that the polynomial fits the cells connected to face f more closely than other cells in the stencil, as shown by [25,26]. To achieve this, we allow each cell to make an unequal contribution to the least-squares fit. We assign an integer *multiplier* to each cell in the stencil, $\mathbf{m} = m_1 \dots m_{12}$, and multiply equation (10) to obtain

$$\tilde{\mathbf{B}}\mathbf{a} = \mathbf{m} \cdot \boldsymbol{\phi} \tag{13}$$

where $\tilde{\mathbf{B}} = \mathbf{MB}$ and $\mathbf{M} = \mathrm{diag}(\mathbf{m})$. The constant coefficient a_1 is calculated from the pseudo-inverse, $\tilde{\mathbf{B}}^+$,

$$a_1 = \tilde{\mathbf{b}}^+_{\mathbf{1}} \cdot \mathbf{m} \cdot \boldsymbol{\phi} \tag{14}$$

where $\tilde{\mathbf{b}}^+_{\mathbf{1}} = \tilde{b}^+_{1,1} \dots \tilde{b}^+_{1,12}$ are the elements of the first row of $\tilde{\mathbf{B}}^+$. Again, a_1 is a weighted mean of $\boldsymbol{\phi}$, where the weights are now $\tilde{\mathbf{b}}^+_{\mathbf{1}} \cdot \mathbf{m}$. Values for $\mathbf{m}$ are chosen so that the cells connected to face f make a greater contribution to the least-squares fit, as discussed later in section 2.1.4.

For faces of a non-rectangular mesh, or faces that are near a boundary, the number of stencil points and number of polynomial terms may differ: a stencil will have one or more points and, for two-dimensional meshes, its polynomial will have between one and nine terms. Additionally, the polynomial cannot have more terms than its stencil has points because this would lead to an underconstrained system of equations. The procedure for choosing suitable polynomials is discussed next.

2.1.3. Polynomial generation

The majority of faces on a uniform two-dimensional mesh have stencils with more than nine cells. For example, a rectangular mesh has 12 points (Fig. 1(a)), and a hexagonal mesh has 10 points (Fig. 1(b)). In both cases, constructing a system of equations using the nine-term polynomial in equation (5) leads to an overconstrained problem that can be solved using least-squares. However, this is not true for faces near boundaries: stencils that have fewer than nine cells (Fig. 3(a)) would result in an underconstrained problem, and stencils that have exactly nine cells may lack sufficient information to constrain high-order terms. For example, the stencil in Fig. 3(b) lacks sufficient information to fit the x^3 term. In such cases, it becomes necessary to perform a least-squares fit using a polynomial with fewer terms.

For every stencil, we find a set of *candidate polynomials* that do not result in an underconstrained problem. In two dimensions, a candidate polynomial has some combination of between one and nine terms from equation (5). There are two additional constraints that a candidate polynomial must satisfy.

First, high-order terms may be included in a candidate polynomial only if the lower-order terms are also included. More precisely, let

$$M(x, y) = x^i y^j : i, j \geq 0 \text{ and } i \leq 3 \text{ and } j \leq 2 \text{ and } i + j \leq 3 \tag{15}$$

be the set of all monomials of degree at most 3 in x, y. A subset S of $M(x, y)$ is "dense" if, whenever $x^a y^b$ is in S, then $x^i y^j$ is also in S for all $0 \leq i \leq a$, $0 \leq j \leq b$. For example, the polynomial $\phi = a_1 + a_2 x + a_3 y + a_4 xy + a_5 x^2 + a_6 x^2 y$ is a dense

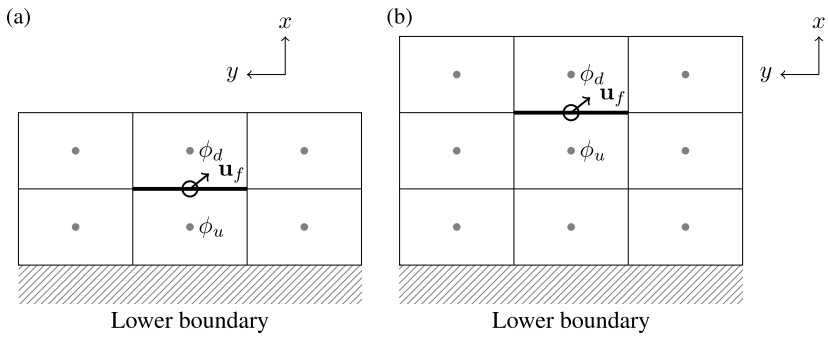

Fig. 3. Upwind-biased stencils for faces near the lower boundary of a rectangular x–z mesh, with (a) a 3×2 stencil for the face immediately adjacent to the lower boundary, and (b) a 3×3 stencil for the face immediately adjacent to the face in (a). Each stencil belongs to the face marked by a thick line. The local coordinate system is shown, having an x direction normal to the face a y direction tangent to the face. For both stencils, attempting a least-squares fit using the nine-term polynomial in equation (5) would result in an underconstrained problem. There is no normal flow at the lower boundary.

subset of $M(x, y)$, but $\phi = a_1 + a_2 x + a_3 y + a_4 x^2 y$ is not because $x^2 y$ can be included only if xy and x^2 are also included. In total there are 26 dense subsets of the two-dimensional polynomial in equation (5).

Second, a candidate polynomial must have a stencil matrix $\mathbf{B}$ that is full rank. The matrix is considered full rank if its smallest singular value is greater than 1×10^{-9}. Using a polynomial with all nine terms and the stencil in Fig. 3(b) results in a rank-deficient matrix and so the nine-term polynomial is not a candidate polynomial.

The candidate polynomials are all the dense subsets of $M(x, y)$ that have a cardinality greater than one with a stencil matrix that is full rank. The final stage of the cubicFit transport scheme selects a candidate polynomial and ensures that the least-squares fit is numerically stable.

2.1.4. Stabilisation procedure

So far, we have constructed a stencil and found a set of candidate polynomials. Applying a least-squares fit to any of these candidate polynomials avoids creating an underconstrained problem. The final stage of the transport scheme chooses a suitable candidate polynomial and appropriate multipliers $\mathbf{m}$ so that the fit is numerically stable.

The approximated value ϕ_F is equal to a_1 which is calculated from equation (14). The value of a_1 is a weighted mean of $\boldsymbol{\phi}$ where $\mathbf{w} = \tilde{\mathbf{b}}_1^+ \cdot \mathbf{m}$ are the weights. If the cell centre values $\boldsymbol{\phi}$ are assumed to approximate a smooth field then we expect ϕ_F to be close to the values of ϕ_u and ϕ_d, and expect ϕ_F to be insensitive to small changes in $\boldsymbol{\phi}$. When the weights $\mathbf{w}$ have large magnitude then this is no longer true: ϕ_F becomes sensitive to small changes in $\boldsymbol{\phi}$ which can result in large, numerically unstable departures from the smooth field $\boldsymbol{\phi}$.

To avoid numerical instabilities, a simplified, one-dimensional von Neumann analysis was performed, presented in Appendix A. The analysis is used to impose three stability conditions on the weights $\mathbf{w}$,

$$0.5 \leq w_u \leq 1 \tag{16a}$$

$$0 \leq w_d \leq 0.5 \tag{16b}$$

$$w_u - w_d \geq \max_{p \in P}(|w_p|) \tag{16c}$$

where w_u and w_d are the weights for the upwind and downwind cells respectively. The *peripheral points* P are the cells in the stencil that are not the upwind or downwind cells, and w_p is the weight for a given peripheral point p. The upwind, downwind and peripheral weights sum to one such that $w_u + w_d + \sum_{p \in P} w_p = 1$.

The stabilisation procedure comprises three steps. In the first step, the set of candidate polynomials is sorted in preference order so that candidates with more terms are preferred over those with fewer terms. If there are multiple candidates with the same number of terms, the minimum singular value of $\mathbf{B}$ is calculated for each candidate, and an ordering is imposed such that the candidate with the larger minimum singular value is preferred. This ordering ensures that the preferred candidate is the highest-order polynomial with the most information content.[2]

In the second step, the most-preferred polynomial is taken from the list of candidates and the multipliers are assigned so that the upwind cell and downwind cell have multipliers $m_u = 2^{10}$ and $m_d = 2^{10}$ respectively, and all peripheral points have multipliers $m_p = 1$. These multipliers are very similar to those used by [25], leading to a well-conditioned matrix $\tilde{\mathbf{B}}$ and a least-squares fit in which the polynomial passes almost exactly through the upwind and downwind cell centre values.

[2] Note that singular values are used for two purposes: first, to test if the matrix $\mathbf{B}$ is full-rank and, second, to impose an ordering on candidates. We have used the minimum singular value, $\sigma_{\min}(\mathbf{B})$, for both purposes. Alternatively, we could use the condition number, $\mathrm{cond}(\mathbf{B})$, which is the ratio of smallest to largest singular value. Experiments revealed that only the candidate ordering was sensitive to the choice of $\sigma_{\min}$ or cond. The most suitable choices of singular value calculations could be explored in future.

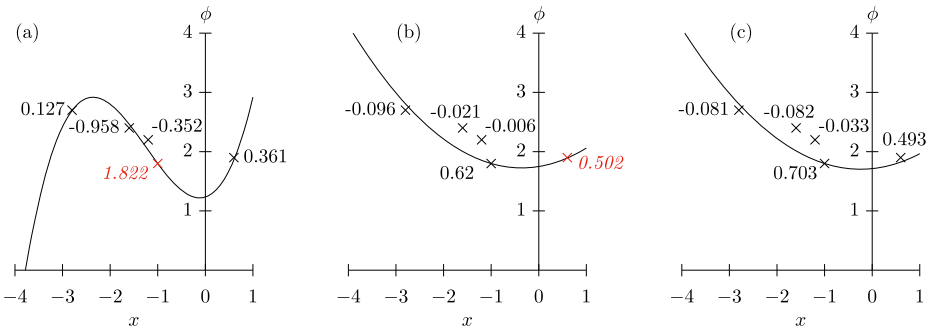

Fig. 4. One-dimensional least-squares fits with a stencil of five points using (a) a cubic polynomial with multipliers $m_u = 1024$, $m_d = 1024$ and $m_p = 1$, (b) a quadratic polynomial with the same multipliers, and (c) a quadratic polynomial with multipliers $m_u = 1024$, $m_d = 1$ and $m_p = 1$. Notice that the curves in (a) and (b) fit almost exactly through the upwind and downwind points immediately adjacent to the y-axis, but in (c) the curve fits almost exactly only through the upwind point immediately to the left of the y-axis. The point data are labelled with their respective weights. Points that have failed one of the stability conditions in equation (16) are marked with italicised labels and are coloured red in the web version of this article. The upwind point is located at $(-1, 1.8)$ and the downwind point at $(0.62, 1.9)$, and the peripheral points are at $(-2.8, 2.4)$, $(-1.6, 2.7)$ and $(-1.2, 2.2)$. The stabilisation procedure (section 2.1.4) calculates weights using only x positions, and values of ϕ are included here for illustration only.

In the third step, we calculate the weights $\mathbf{w}$ and evaluate them against the stability conditions given in equation (16). If any condition is violated, the value of m_d is halved and the conditions are evaluated with the new weights. This step is repeated until the weights satisfy the stability conditions, or m_d becomes smaller than one. In practice, the conditions are satisfied when m_d is either small (between 1 and 4) or equal to 2^{10}. The upwind multiplier m_u is fixed at 2^{10} and the peripheral multipliers m_p are fixed at 1. If the conditions are still not satisfied, then we start again from the second step with the next polynomial in the candidate list.

Finally, if no stable weights are found for any candidate polynomial, we revert to an upwind scheme such that $w_u = 1$ and all other weights are zero. In our experiments we have not encountered any stencil for which this last resort is required. Furthermore, our experiments show that the stabilisation procedure only modifies the least squares fit for stencils near boundaries and for stencils in distorted mesh regions. For stencils in the interior of a uniform rectangular mesh, the least squares fit includes all terms in equation (5) with $m_u = m_d = 2^{10}$.

To illustrate the stabilisation procedure, Fig. 4(a) presents a one-dimensional example of a cubic polynomial fitted through five points, with the weight at each point printed beside it. The stabilisation procedure only uses the x positions of these points and does not use the values of ϕ themselves. The ϕ values are included here for illustration only. Hence, for a given set of x positions, the same set of weights are chosen irrespective of the ϕ values.

For a one-dimensional cubic polynomial fit, the list of candidate polynomials in preference order is

$$\phi = a_1 + a_2 x + a_3 x^2 + a_4 x^3 \,, \tag{17}$$

$$\phi = a_1 + a_2 x + a_3 x^2 \,, \tag{18}$$

$$\phi = a_1 + a_2 x \,, \tag{19}$$

$$\phi = a_1 \,. \tag{20}$$

We begin with the cubic equation (17). The multipliers are chosen so that the polynomial passes almost exactly through the upwind and downwind points that are immediately to the left and right of the y-axis respectively. The stability condition on the upwind point is violated because $w_u = 1.822 > 1$ (equation (16a)). Reducing the downwind multiplier does not help to satisfy the stability condition, so we start again with the quadratic equation (18), and the new fit is presented in Fig. 4(b). Again, the multipliers are chosen to force the polynomial through the upwind and downwind points, but this violates the stability condition on the downwind point because $w_d = 0.502 > 0.5$ (equation (16b)). This time, however, stable weights are found by reducing m_d to one (Fig. 4(c)) and these are the weights that will be used to approximate ϕ_F, where the polynomial intercepts the y-axis.

2.1.5. Future extension to three dimensions

All the procedures used in the cubicFit scheme generalise to three dimensions. The stencil construction procedure described in section 2.1.1 creates a stencil with 12 cells for a face in the interior of a two-dimensional rectangular mesh. In three dimensions, the same procedure creates a stencil with $3 \times 12 = 36$ cells. A three-dimensional stencil has three times as many cells as its two-dimensional counterpart if the mesh has prismatic cells arranged in columns. Hence, the computational cost during integration increases three-fold when moving from two dimensions to three dimensions.

To extend the least squares fit to three dimensions, the two-dimensional polynomial in equation (5) is replaced with its three-dimensional counterpart,

$$\phi = a_1 + a_2 x + a_3 y + a_4 z + a_5 x^2 + a_6 xy + a_7 y^2 + a_8 xz + a_9 yz + a_{10} z^2 + \\ + a_{11} x^3 + a_{12} x^2 y + a_{13} xy^2 + a_{14} x^2 z + a_{15} xz^2 + a_{16} yz^2 + a_{17} y^2 z + a_{18} xyz \,. \tag{21}$$

The procedure for generating candidate polynomials described in section 2.1.3 results in 26 dense subsets in two dimensions and 842 dense subsets in three dimensions. Note that the combinatorial explosion of dense subsets in three dimensions does not increase the computational cost during integration.

The stabilisation procedure described in section 2.1.4 requires further numerical experiments to verify that it is sufficient for three-dimensional flows and arbitrary polyhedral meshes. An initial three-dimensional test with uniform flow and a uniform Cartesian mesh obtained a numerically stable result. For stencils in the interior of the domain, the least squares fit includes all polynomial terms in equation (21) with $m_u = m_d = 2^{10}$. The stabilisation procedure does not modify the least squares fit for these stencils, but we have not explored the three-dimensional extension of cubicFit any further.

2.2. Multidimensional linear upwind transport scheme

The multidimensional linear upwind scheme, called "linearUpwind" hereafter, is documented here since it provides a baseline accuracy for the experiments in section 3. The approximation of ϕ_F is calculated using a gradient reconstruction,

$$\phi_F = \phi_u + \nabla_c \phi \cdot \left(\mathbf{x}_f - \mathbf{x}_c \right) \tag{22}$$

where ϕ_u is the upwind value of ϕ, and $\mathbf{x}_f$ and $\mathbf{x}_c$ are the position vectors of the face centroid and cell centroid respectively. The gradient $\nabla_c \phi$ is calculated using Gauss' theorem:

$$\nabla_c \phi = \frac{1}{\mathcal{V}_c} \sum_{f \in c} \widetilde{\phi}_F \mathbf{S}_f \tag{23}$$

where $\widetilde{\phi}_F$ is linearly interpolated from the two neighbouring cells of face f. The resulting stencil comprises all cells sharing a face with the upwind cell, including the upwind cell itself. For a face in the interior of a two-dimensional rectangular mesh, the stencil for the linearUpwind scheme is a '+' shape with 5 cells. On the same mesh, the stencil for the cubicFit scheme is more than twice the size with 12 cells. For cells adjacent to boundaries having zero gradient boundary conditions, the boundary value is set to be equal to the cell centre value before equation (23) is evaluated. This implementation of the multidimensional linear upwind scheme is included in the OpenFOAM software distribution [45].

3. Results

Three idealised numerical tests are performed to compare the accuracy of the cubicFit transport scheme with the multidimensional linear upwind scheme and with other transport schemes in the literature. The first test transports a tracer horizontally on two-dimensional, highly-distorted terrain-following meshes, following Schär et al. [2]. The second is a new test case that modifies the velocity field and tracer position from the first test in order to challenge the stability and accuracy of the transport schemes near mountainous lower boundaries. The third test evaluates the cubicFit scheme on hexagonal–icosahedral meshes and cubed-sphere meshes with deformational flow on a spherical Earth, as specified by Lauritzen et al. [43].

We have implemented the cubicFit transport scheme and the numerical test cases using the OpenFOAM CFD library because it enables a like-for-like comparison between mesh types and transport schemes. We provide source code archives for the OpenFOAM implementation of the cubicFit scheme [46], the ASAM cut cell mesh generator [47] and associated OpenFOAM converter [48], and the hexagonal–icosahedral mesh generator [49]. For the numerical test cases presented here we also supply the source code [50] and result data [51].

3.1. Horizontal transport over mountains

A two-dimensional transport test was developed by Schär et al. [2] to study the effect of terrain-following coordinate transformations on numerical accuracy. In this standard test, a tracer is positioned aloft and transported horizontally over wave-shaped mountains. The test challenges transport schemes because the tracer must cross mesh layers, which acts to reduce numerical accuracy [2,13]. Here we use a more challenging variant of this test that has steeper mountains and highly-distorted terrain-following meshes. Convergence results are compared using the linearUpwind and cubicFit transport schemes.

The domain is defined on a rectangular x–z plane that is 301 km wide and 25 km high as measured between parallel boundary edges. Boundary conditions are imposed on the tracer density ϕ such that $\phi = 0$ kg m^{-3} at the inlet boundary, and a zero normal gradient $\partial\phi/\partial n = 0$ kg m^{-4} is imposed at the outlet boundary. There is no normal flow at the lower and upper boundaries. A range of mesh spacings are chosen so that $\Delta x : \Delta z = 2 : 1$ to match the original test specification from Schär et al. [2].

The terrain is wave-shaped, specified by the surface elevation h such that

$$h(x) = h^\star \cos^2(\alpha x) \tag{24a}$$

where

$$h^\star(x) = \begin{cases} h_0 \cos^2(\beta x) & \text{if } |x| < a \\ 0 & \text{otherwise} \end{cases} \tag{24b}$$

where $a = 25$ km is the mountain envelope half-width, $h_0 = 6$ km is the maximum mountain height, $\lambda = 8$ km is the wavelength, $\alpha = \pi/\lambda$ and $\beta = \pi/(2a)$. Note that, in order to make this test more challenging, the mountain height h_0 is double the mountain height used by [2].

A basic terrain-following (BTF) mesh is constructed by using the terrain profile to modify the uniform mesh. The BTF method uses a linear decay function so that mesh surfaces become horizontal at the top of the model domain [5],

$$z(x) = (H - h(x))\left(z^\star/H\right) + h(x) \tag{25}$$

where z is the geometric height, H is the height of the domain, $h(x)$ is the surface elevation and $z^\star$ is the computational height of a mesh surface. If there were no terrain then $h = 0$ and $z = z^\star$.

A velocity field is prescribed with uniform horizontal flow aloft and zero flow near the ground,

$$u(z) = u_0 \begin{cases} 1 & \text{if } z \geq z_2 \\ \sin^2\left(\frac{\pi}{2}\frac{z-z_1}{z_2-z_1}\right) & \text{if } z_1 < z < z_2 \\ 0 & \text{otherwise} \end{cases} \tag{26}$$

where $u_0 = 10\ \mathrm{m\,s^{-1}}$, $z_1 = 7$ km and $z_2 = 8$ km.

A tracer with density ϕ has the shape

$$\phi(x,z) = \phi_0 \begin{cases} \cos^2\left(\frac{\pi r}{2}\right) & \text{if } r \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{27a}$$

with radius r given by

$$r = \sqrt{\left(\frac{x - x_0}{A_x}\right)^2 + \left(\frac{z - z_0}{A_z}\right)^2} \tag{27b}$$

where $A_x = 25$ km, $A_z = 3$ km are the horizontal and vertical half-widths respectively, and $\phi_0 = 1\ \mathrm{kg\,m^{-3}}$ is the maximum density of the tracer. At $t = 0$ s, the tracer is centred at $(x_0, z_0) = (-50\text{ km}, 12\text{ km})$ so that the tracer is upwind of the mountain, in the region of uniform flow above z_2.

Tests are integrated for 10 000 s using time-steps chosen for each mesh so that the maximum Courant number is about 0.4. This choice yields a time-step that is well below any stability limit, as recommended by Lauritzen et al. [43]. By the end of integration the tracer is positioned downwind of the mountain. The analytic solution at $t = 10000$ s is centred at $(x_0, z_0) = (50\text{ km}, 12\text{ km})$. Error norms are calculated by subtracting the analytic solution from the numerical solution,

$$\ell_2 = \sqrt{\frac{\sum_c (\phi - \phi_T)^2 \mathcal{V}_c}{\sum_c (\phi_T^2 \mathcal{V}_c)}} \tag{28}$$

$$\ell_\infty = \frac{\max_c |\phi - \phi_T|}{\max_c |\phi_T|} \tag{29}$$

where ϕ is the numerical value, ϕ_T is the analytic value, $\sum_c$ denotes a summation over all cells c in the domain, and $\max_c$ denotes a maximum value of any cell.

Tests were performed using the linearUpwind and cubicFit schemes at mesh spacings between $\Delta x = 250$ m and $\Delta x = 5000$ m. Numerical convergence in the ℓ_2 and ℓ_∞ norms is plotted in Fig. 5. The linearUpwind and cubicFit schemes are second-order convergent at all but the coarsest mesh spacings where errors are saturated for both schemes.

The cubicFit scheme achieves a given ℓ_2 error using a mesh spacing that is almost twice as coarse as that needed by the linearUpwind scheme. Doubling the mesh spacing results in a coarser mesh with four times fewer cells because the $\Delta x : \Delta z$ aspect ratio is fixed. Recall that the stencil for the cubicFit scheme has about twice as many cells as the stencil for the linearUpwind scheme. Hence, for a given ℓ_2 error, the computational cost during integration of the cubicFit scheme is about half the computational cost of the linearUpwind scheme.

This test demonstrates that the cubicFit scheme is second-order convergent in the domain interior irrespective of mesh distortions. In the next test, we assess the numerical accuracy of the transport schemes near a distorted, mountainous lower boundary.

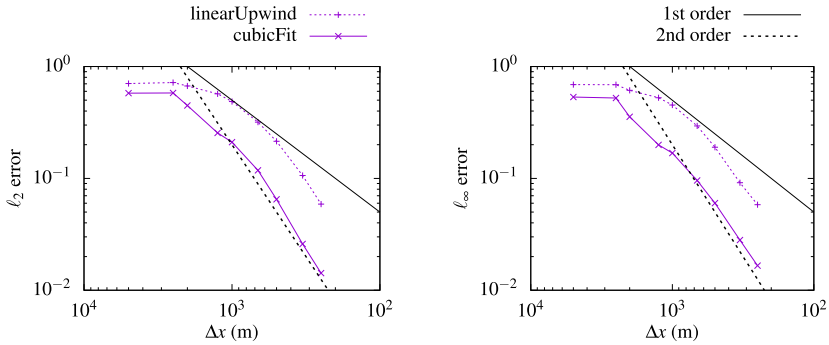

Fig. 5. Numerical convergence of the two-dimensional horizontal transport test over mountains. ℓ_2 errors (equation (28)) and ℓ_∞ errors (equation (29)) are marked at mesh spacings between 5000 m and 250 m using linearUpwind and cubicFit transport schemes on basic terrain-following meshes.

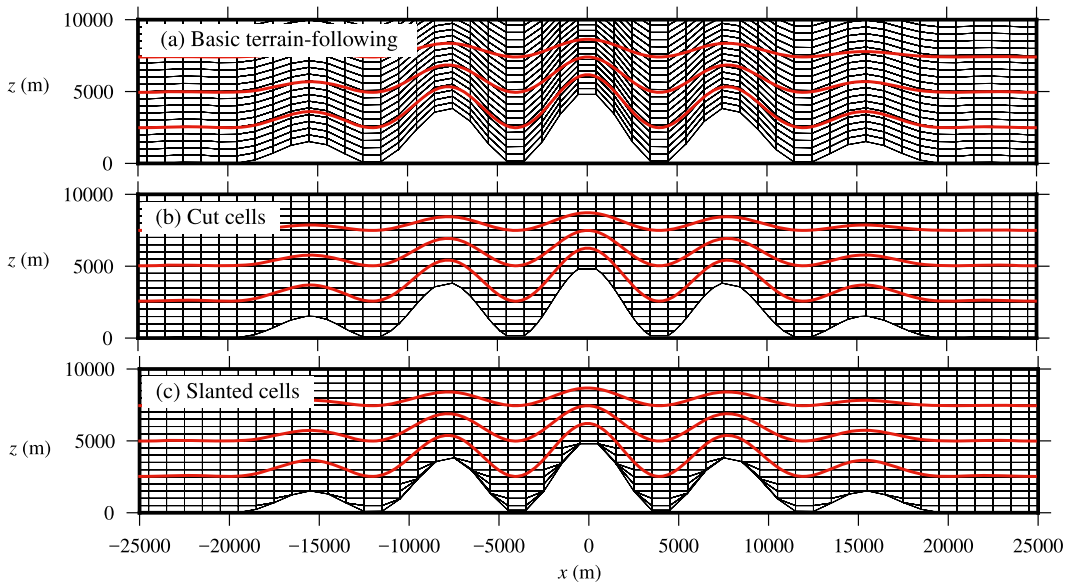

Fig. 6. Two dimensional x–z meshes created with the (a) basic terrain-following, (b) cut cell, and (c) slanted cell methods, and used for the tracer transport tests in section 3.2. Cell edges are marked by thin black lines. The peak mountain height $h_0 = 5$ km. The velocity field is the same for all mesh types with streamlines marked on each panel by thick lines that are coloured red in the web version of the article. The velocity field (equation (30)) follows the lower boundary and becomes entirely horizontal above $H_1 = 10$ km. Only the lowest 10 km for the central region of the domain is shown. The entire domain is 301 km wide and 25 km high.

3.2. Transport over a mountainous lower boundary

The horizontal transport test in the previous section is useful for assessing numerical accuracy on terrain-following meshes, but it presents no particular challenge on cut cell meshes because there is no flow through the distorted cut cells near the ground [52]. Here we present another variant of the standard horizontal transport test that challenges transport schemes on all mesh types. By positioning the tracer next to the ground and modifying the velocity field, we can assess the accuracy of the cubicFit scheme near the lower boundary. Results using the cubicFit scheme are compared with the linearUpwind scheme on basic terrain-following, cut cell and slanted cell meshes.

Cut cell meshes are constructed using the ASAM grid generator [53,54]. Slanted cell meshes are constructed following the approach by Shaw and Weller [13]: vertices that are underground are moved up to the surface and zero-area faces and zero-volume cells are removed. Unlike [13], vertices are never moved downwards.

Following Schär et al. [2], the domain is 301 km wide and 25 km high as measured between parallel boundary edges, with a mesh spacing of $\Delta x = 1000$ m and $\Delta z = 500$ m. The boundary conditions are the same as those used in section 3.1. Cell edges in the central region of the domain are shown in Fig. 6 for each of the three mesh types. Cells in the BTF mesh are highly distorted over steep slopes (Fig. 6(a)) while the cut cell mesh (Fig. 6(b)) and slanted cell mesh (Fig. 6(c)) are orthogonal everywhere except for cells nearest the ground.

Similar to the approach by [13], a velocity field is chosen that follows the terrain at the surface and becomes entirely horizontal aloft. A streamfunction Ψ is used so that the discrete velocity field is non-divergent, such that

Table 1
Time-steps (s) for the two-dimensional transport test over a mountainous lower boundary. The time-steps were chosen so that the maximum Courant number was between 0.36 and 0.46.

Mesh type	Peak mountain height h_0 (km)				
	0	3	4	5	6
BTF	40	16	10	8	5
Cut cell	40	1.6	1.6	0.5	1.5
Slanted cell	40	8	6.25	5	4

$$\Psi(x,z) = -u_0 H_1 \frac{z-h}{H_1-h} \tag{30}$$

where $u_0 = 10\ \mathrm{m\,s^{-1}}$, which is the horizontal velocity where $h(x) = 0$. There is no normal flow at the lower and upper boundaries. The velocity field becomes purely horizontal above $H_1 = 10$ km and, elsewhere, the flow is predominantly horizontal, with non-zero vertical velocities only above sloping terrain. The horizontal and vertical components of velocity, u and w, are given by

$$u = -\frac{\partial \Psi}{\partial z} = u_0 \frac{H_1}{H_1-h}, \quad w = \frac{\partial \Psi}{\partial x} = u_0 H_1 \frac{\mathrm{d}h}{\mathrm{d}x} \frac{H_1-z}{(H_1-h)^2}, \tag{31}$$

$$\frac{\mathrm{d}h}{\mathrm{d}x} = -h_0 \left[\beta \cos^2(\alpha x) \sin(2\beta x) + \alpha \cos^2(\beta x) \sin(2\alpha x) \right]. \tag{32}$$

Unlike the horizontal transport test in [2], the velocity field presented here extends from the top of the domain all the way to the ground.

The flow is deliberately misaligned with the BTF, cut cell and slanted cell meshes away from the ground (Fig. 6) to ensure that flow always crosses mesh surfaces in order to challenge the transport scheme. The value of H_1 is chosen to be much smaller than the domain height H in equation (25) so that flow crosses the surfaces of the BTF mesh. This is evident in Fig. 6(a) where the velocity streamlines are tangential to the mesh only at the ground.

The tracer is again defined by equation (27b) but is now positioned at the ground with $(x_0, z_0) = (-50\ \mathrm{km}, 0\ \mathrm{km})$ with half-widths $A_x = 25$ km and $A_z = 10$ km. Tests are integrated forward for 10 000 s. The time-step was chosen for each mesh so that the maximum Courant number was about 0.4 (Table 1). An analytic solution at 10 000 s is obtained by calculating the new horizontal position of the tracer. Integrating along the trajectory yields t, the time taken to move from the left side of the mountain to the right [13],

$$\mathrm{d}t = \mathrm{d}x / u(x) \tag{33}$$

$$t = \int_0^x \frac{H-h(x)}{u_0 H}\,\mathrm{d}x \tag{34}$$

$$t = \frac{x}{u_0} - \frac{h_0}{16 u_0 H}\left[4x + \frac{\sin 2(\alpha+\beta)x}{\alpha+\beta} \frac{\sin 2(\alpha-\beta)x}{\alpha-\beta} + 2\left(\frac{\sin 2\alpha x}{\alpha} + \frac{\sin 2\beta x}{\beta} \right) \right] \tag{35}$$

By solving this equation we find that $x(t = 10\,000\ \mathrm{s}) = 54342.8$ m when $h_0 = 5$ km.

The tracer density boundary conditions are the same as those in the previous test. Since the cubicFit transport scheme uses values at boundaries with Dirichlet boundary conditions, the cubicFit scheme uses only inlet boundary values in this test case.

Three series of tests were performed using similar configurations. The first series uses a peak mountain height of $h_0 = 5$ km to examine errors on different mesh types using the two transport schemes. The second series varies the peak mountain height to examine the sensitivity of the transport schemes to mesh distortions. The third series verifies accuracy at Courant numbers close to the limit of stability, and examines the longest stable time-step for different mesh types.

For the first series of tests with $h_0 = 5$ km, tracer contours at the initial time $t = 0$ s, half-way time $t = 5000$ s, and end time $t = 10\,000$ s are shown in Fig. 7 using the cubicFit scheme on the BTF mesh. As apparent at $t = 5000$ s, the tracer is distorted by the terrain-following velocity field as it passes over the mountain, but its original shape is restored once it has cleared the mountain by $t = 10\,000$ s. A small phase lag is apparent when the numerical solution marked with solid contour lines is compared with the analytic solution marked with dotted contour lines.

Numerical errors are more clearly revealed by subtracting the analytic solution from the numerical solution. Error fields are compared between BTF, cut cell and slanted cell meshes using the linearUpwind scheme (Figs. 8(a), 8(b) and 8(c) respectively) and the cubicFit scheme (Figs. 8(d), 8(e) and 8(f) respectively). Results are least accurate using the linearUpwind scheme on the slanted cell mesh (Fig. 8(c)). The final tracer is slightly distorted and does not extend far enough towards the ground. The ℓ_∞ error magnitude is reduced by using the linearUpwind scheme on the cut cell mesh (Fig. 8(b)), but the shape of the error remains the same. The cubicFit scheme is less sensitive to the choice of mesh with similar error

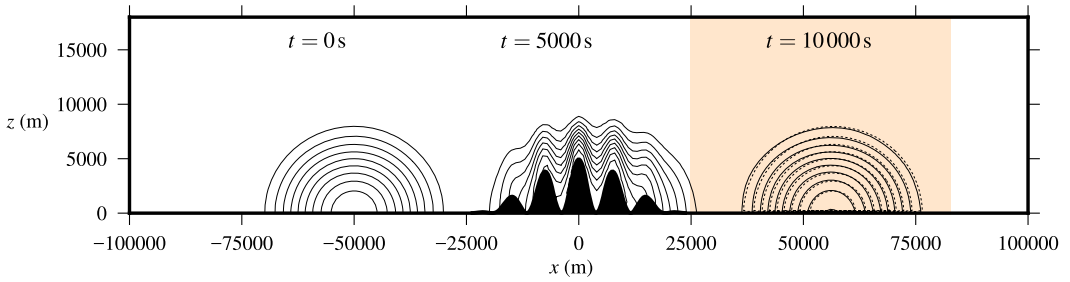

Fig. 7. Evolution of the tracer in the two-dimensional transport test over a mountainous lower boundary. The tracer is transported to the right over the wave-shaped terrain. Tracer contours are every 0.1 kg m^{-3}. The result obtained using the cubicFit scheme on the basic terrain-following mesh is shown at $t = 0$ s, $t = 5000$ s and $t = 10\,000$ s with solid black contours. The analytic solution at $t = 10\,000$ s is shown with dotted contours. The shaded box indicates the region that is plotted in Fig. 8.

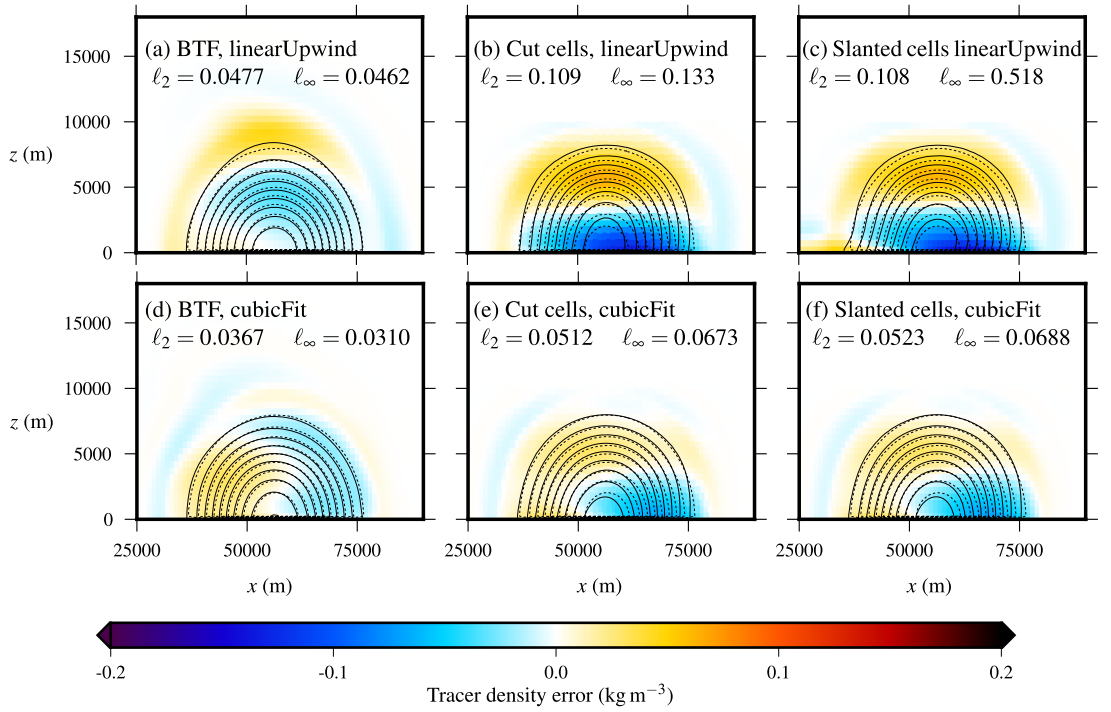

Fig. 8. Tracer contours at $t = 10\,000$ s for the two-dimensional tracer transport tests over a mountainous lower boundary. A region in the lee of the mountain is plotted corresponding to the shaded area in Fig. 7. Results are presented on BTF, cut cell and slanted cell meshes (shown in Fig. 6) using the linearUpwind and cubicFit transport schemes. The numerical solutions are marked by solid black lines. The analytic solution is marked by dotted lines. Contours are every 0.1 kg m^{-3}.

magnitudes on the BTF mesh (Fig. 8(d)), cut cell mesh (Fig. 8(e)) and slanted cell mesh (Fig. 8(f)). Errors using the cubicFit scheme on cut cell and slanted cell meshes are much smaller than the errors using the linearUpwind scheme on the same meshes.

To further examine the performance of the cubicFit scheme in the presence of steep terrain, a second series of tests were performed in which the peak mountain height was varied from 0 km to 6 km keeping all other parameters constant. Results were obtained on BTF, cut cell and slanted cell meshes using the linearUpwind scheme and cubicFit scheme. Again, the time-step was chosen for each test so that the maximum Courant number was about 0.4 (Table 1). The ℓ_2 error was calculated by subtracting the analytic solution from the numerical solution (Fig. 9). Note that the analytic solution is a function of mountain height, with the tracer travelling farther over higher mountains due to non-divergent flow through a narrower channel. In all cases, error increases with increasing mountain height because steeper slopes lead to greater mesh distortions. Errors are identical for a given transport scheme when $h_0 = 0$ km and the ground is entirely flat because the BTF, cut cell and slanted cell meshes are identical. Compared with the cubicFit scheme, the linearUpwind scheme is more sensitive to the mesh type and mountain height. The linearUpwind scheme is unstable on the slanted cell mesh with a peak mountain height $h_0 = 6$ km despite using a Courant number of 0.428. In contrast, the cubicFit scheme is less sensitive to

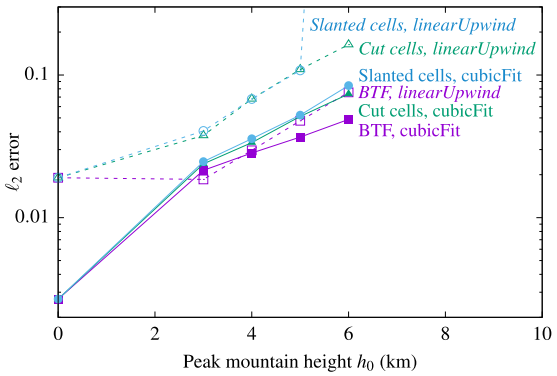

Fig. 9. Error measures for the two-dimensional tracer transport tests over a mountainous lower boundary. Peak mountain heights h_0 are from 0 km to 6 km. Results are compared on BTF, cut cell and slanted cell meshes using the linearUpwind and the cubicFit schemes. At $h_0 = 0$ km the terrain is entirely flat and the BTF, cut cell and slanted cell meshes are identical. At $h_0 = 6$ km the linearUpwind scheme is unstable on the slanted cell mesh.

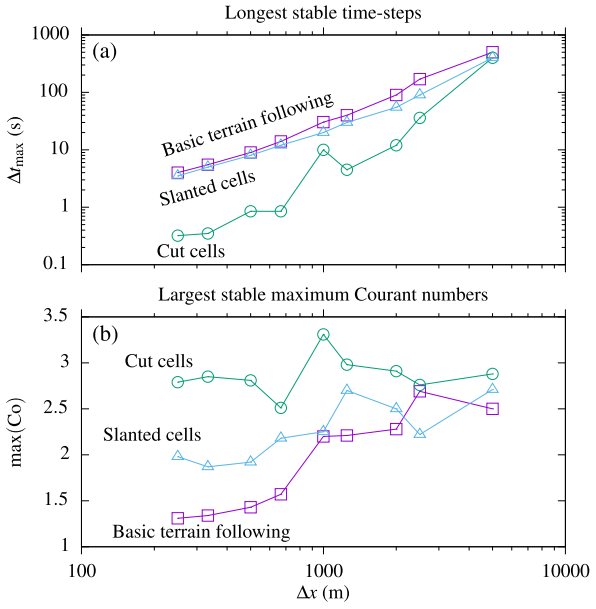

Fig. 10. (a) Longest stable time-steps, Δt_{max}, and (b) largest stable maximum Courant numbers, max(Co), for the two-dimensional tracer transport test over a mountainous lower boundary. Results were obtained on basic terrain-following, cut cell and slanted cell meshes at mesh spacings between $\Delta x = 5000$ m and $\Delta x = 250$ m. The largest stable maximum Courant numbers were calculated from the corresponding longest stable time-steps using equation (4).

the mesh type and errors grow more slowly with increasing mountain height. The cubicFit scheme yields stable results in all tests.

A final series of tests were performed to determine the stability limit of the cubicFit scheme with the two-stage Heun time-stepping scheme (equation (2b)). The tracer was transported on BTF, slanted cell and cut cell meshes with a variety of mesh spacings between $\Delta x = 5000$ m and $\Delta x = 250$ m. Δz was chosen so that a constant aspect ratio is preserved such that $\Delta x/\Delta z = 2$. For each test, the time-step was increased until the result became unstable. The largest stable time-steps, Δt_{max}, are presented in Fig. 10(a). BTF meshes permit the longest time-steps of all three mesh types since cells are almost uniform in volume. As expected, the longest stable time-step scales linearly with BTF mesh spacing. There is no such linear scaling on cut cell meshes because these meshes can have arbitrarily small cells. The time-step constraints on cut cell meshes are the most severe of the three mesh types. Slanted cell meshes have a slightly stronger time-step constraint than BTF meshes but still exhibit similar linear scaling with mesh spacing. Furthermore, a dynamical model that uses slanted cell meshes instead of BTF meshes is expected to calculate pressure gradients more accurately [13].

Fig. 10(b) presents the largest stable maximum Courant numbers, max(Co), which were calculated by substituting $\Delta t = \Delta t_{max}$ into equation (4). On basic terrain following meshes, the maximum Courant number tends towards about 1.3 with finer mesh spacings. No such trend is found on cut cell or slanted cell meshes. Cut cell meshes permit the largest maximum Courant numbers of around 3, but the largest stable time-steps on cut cell meshes are still smaller than corresponding time-steps on basic terrain following and slanted cell meshes.

This paper focuses on the spatial discretisation of the cubicFit scheme, but the stability limit depends also upon the choice of time-stepping. As such, we have not calculated a theoretical Courant number limit, although such an analysis should be possible using the techniques in [55].

The transport tests presented in this section demonstrate that the cubicFit scheme is suitable for flows over very steep terrain on two-dimensional terrain-following, cut cell and slanted cell meshes. The cubicFit scheme is less sensitive to the mesh type and mountain steepness compared to the linearUpwind scheme. The linearUpwind scheme becomes unstable over very steep slopes but the cubicFit scheme is stable for all tests. The accuracy of the cubicFit scheme was largely insensitive to the choice of time-step. In the next section, we evaluate the cubicFit scheme using more complex, deformational flows on icosahedral meshes and cubed-sphere meshes.

3.3. Deformational flow on a sphere

The tests so far have used flows that are mostly uniform on meshes that are based on rectangular cells. To ensure that the cubicFit transport scheme is suitable for complex flows on a variety of meshes, we use a standard test of deformational flow on a spherical Earth, as specified by Lauritzen et al. [43]. Results are compared between linearUpwind and cubicFit schemes using hexagonal–icosahedral meshes and cubed-sphere meshes. Hexagonal–icosahedral meshes are constructed by successive refinement of a regular icosahedron following the approach by [28,56,57] without any mesh twisting. Cubed-sphere meshes are constructed using an equi-distant gnomic projection of a cube having a uniform Cartesian mesh on each panel [58].

Following appendix A9 in [32], the average equatorial spacing $\Delta\lambda$ is used as a measure of mesh spacing. It is defined as

$$\Delta\lambda = 360^\circ \frac{\overline{\Delta x}}{2\pi R_e} \tag{36}$$

where $\overline{\Delta x}$ is the mean distance between cell centres and $R_e = 6.3712 \times 10^6$ m is the radius of the Earth.

The deformational flow test specified by Lauritzen et al. [43] comprised six elements:

1. a convergence test using a Gaussian-shaped tracer
2. a “minimal” resolution test using a cosine bell-shaped tracer
3. a test of filament preservation
4. a test using a “rough” slotted cylinder tracer
5. a test of correlation preservation between two tracers
6. a test using a divergent velocity field

We assess the cubicFit scheme using the first two tests only. We do not consider filament preservation, correlation preservation, or the transport of a “rough” slotted cylinder because no shape-preserving filter has yet been developed for the cubicFit scheme. Stable results were obtained when testing the cubicFit scheme using a divergent velocity field, but no further analysis is made here.

The first deformational flow test uses an infinitely continuous initial tracer that is transported in a non-divergent, time-varying, rotational velocity field. The velocity field deforms two Gaussian ‘hills’ of tracer into thin vortical filaments. Half-way through the integration the rotation reverses so that the filaments become circular hills once again. The analytic solution at the end of integration is identical to the initial condition. A rotational flow is superimposed on a time-invariant background flow in order to avoid error cancellation. The non-divergent velocity field is defined by the streamfunction Ψ,

$$\Psi(\lambda, \theta, t) = \frac{10 R_e}{T} \sin^2(\lambda') \cos^2(\theta) \cos\left(\frac{\pi t}{T}\right) - \frac{2\pi R_e}{T} \sin(\theta) \tag{37}$$

where λ is a longitude, θ is a latitude, $\lambda' = \lambda - 2\pi t/T$, and $T = 12$ days is the duration of integration. The time-step is chosen such that the maximum Courant number is about 0.4.

The initial tracer density ϕ is defined as the sum of two Gaussian hills,

$$\phi = \phi_1(\lambda, \theta) + \phi_2(\lambda, \theta)\,. \tag{38}$$

An individual hill ϕ_i is given by

$$\phi_i(\lambda, \theta) = \phi_0 \exp\left(-b\left(\frac{|\mathbf{x} - \mathbf{x}_i|}{R_e}\right)^2\right) \tag{39}$$

where $\phi_0 = 0.95$ kg m^{-3} and $b = 5$. The Cartesian position vector $\mathbf{x} = (x, y, z)$ is related to the spherical coordinates (λ, θ) by

$$(x, y, z) = (R_e \cos\theta \cos\lambda, R_e \cos\theta \sin\lambda, R_e \sin\theta)\,. \tag{40}$$

The centre of hill i is positioned at $\mathbf{x}_i$. In spherical coordinates, two hills are centred at

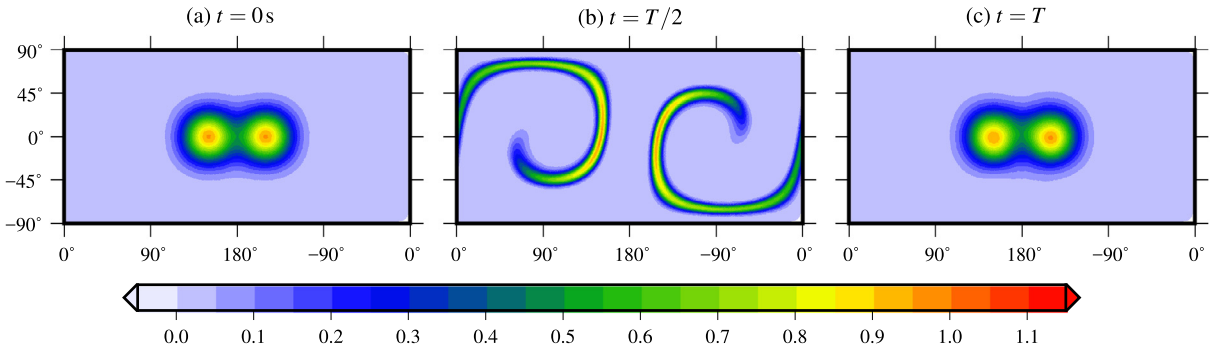

Fig. 11. Tracer fields for the deformational flow test using initial Gaussian hills. The tracer is deformed by the velocity field before the rotation reverses to return the tracer to its original distribution: (a) the initial tracer distribution at $t = 0$ s; (b) by $t = T/2$ the Gaussian hills are stretched into a thin S-shaped filament; (c) at $t = T$ the tracer resembles the initial Gaussian hills except for some distortion and diffusion due to numerical errors. Results were obtained with the cubicFit scheme on a hexagonal–icosahedral mesh with an average equatorial mesh spacing of $\Delta\lambda = 0.542°$.

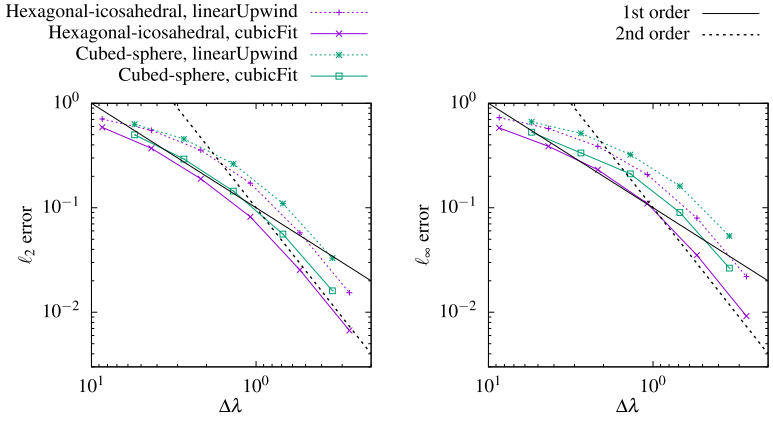

Fig. 12. Numerical convergence of the deformational flow test on the sphere using initial Gaussian hills. ℓ_2 errors (equation (28)) and ℓ_∞ errors (equation (29)) are marked at mesh spacings between 8.61° and 0.271° using the linearUpwind scheme (dotted lines) and the cubicFit scheme (solid lines) on hexagonal–icosahedral meshes and cubed-sphere meshes.

$$(\lambda_1, \theta_1) = (5\pi/6, 0) \tag{41}$$

$$(\lambda_2, \theta_2) = (7\pi/6, 0) \tag{42}$$

The results in Fig. 11 are obtained using the cubicFit scheme on a hexagonal–icosahedral mesh with $\Delta\lambda = 0.542°$. The initial Gaussian hills are shown in Fig. 11(a). At $t = T/2$ the tracer has been deformed into an S-shaped filament (Fig. 11(b)). By $t = T$ the tracer has almost returned to its original distribution except for some slight distortion and diffusion that are the result of numerical errors (Fig. 11(c)).

To determine the order of convergence and relative accuracy of the linearUpwind and cubicFit schemes, the same test was performed at a variety of mesh spacings between $\Delta\lambda = 8.61°$ and $\Delta\lambda = 0.271°$ on hexagonal–icosahedral meshes and cubed-sphere meshes. The results are shown in Fig. 12. The solution is slow to converge at coarse resolutions, and this behaviour agrees with the results from Lauritzen et al. [43]. Both linearUpwind and cubicFit schemes achieve second-order accuracy at finer mesh spacings. For any given mesh type and mesh spacing, the cubicFit scheme is more accurate than the linearUpwind scheme. Results are more accurate using hexagonal–icosahedral meshes compared to cubed-sphere meshes. It is not known whether the larger errors on cubed-sphere meshes are due to mesh non-uniformities at panel corners but there is no evidence of grid imprinting in the error fields (not shown).

A slightly more challenging variant of the same test is performed using a quasi-smooth tracer field defined as the sum of two cosine bells,

$$\phi = \begin{cases} b + c\phi_1(\lambda, \theta) & \text{if } r_1 < r, \\ b + c\phi_2(\lambda, \theta) & \text{if } r_2 < r, \\ b & \text{otherwise.} \end{cases} \tag{43}$$

The velocity field is the same as before. This test is used to determine the "minimal" resolution, $\Delta\lambda_m$, which is specified by Lauritzen et al. [43] as the coarsest mesh spacing for which $\ell_2 \approx 0.033$.

Table 2
Minimal resolutions for the cubicFit and linearUpwind schemes in the test of deformational flow using cosine bells. Italicised values have been extrapolated using the second-order convergence obtained at coarser mesh spacings. For comparison with existing models, some results are also included for unlimited versions of the transport schemes from the intercomparison by Lauritzen et al. [32].

Transport scheme	Mesh type	Minimal resolution (°)
linearUpwind	Cubed-sphere	*0.15*
FARSIGHT, grid-point semi-Lagrangian [59]	Cubed-sphere	0.1875
linearUpwind	Hexagonal–icosahedral	*0.2*
SLFV-SL, swept-area scheme [60]	Hexagonal–icosahedral	0.25
cubicFit	Cubed-sphere	*0.25*
cubicFit	Hexagonal–icosahedral	0.3
ICON-FFSL, swept-area scheme [60]	Triangular–icosahedral	0.42

The minimal resolution for the cubicFit scheme on a hexagonal–icosahedral mesh is about $\Delta\lambda_m = 0.3°$. Tests were not performed at mesh spacings finer than $\Delta\lambda = 0.271°$ but approximate minimal resolutions have been extrapolated from the second-order convergence that is found at fine mesh spacings. These minimal resolutions are presented in Table 2 along with a selection of transport schemes having similar minimal resolutions from the model intercomparison by Lauritzen et al. [32].

The series of deformational flow tests presented here demonstrate that the cubicFit scheme is suitable for transport on spherical meshes based on quadrilaterals and hexagons. The cubicFit scheme is largely insensitive to the mesh type, and results are more accurate compared to the linearUpwind scheme for a given mesh type and mesh spacing. Neither scheme requires special treatment at the corners of cubed-sphere panels.

4. Conclusion

Atmospheric models are using increasingly fine horizontal mesh spacings that resolve steep slopes in terrain resulting in highly-distorted meshes, increased numerical errors and numerical instabilities. We have presented a new multidimensional method-of-lines transport scheme, cubicFit, that enforces stability conditions derived from a von Neumann stability analysis to make the scheme stable over steep terrain on highly-distorted, arbitrary meshes. The scheme has a low computational cost at runtime, requiring only n multiplies per face per time-stage using a stencil with n cells. Stability condition calculations are pre-computed during model initialisation since they depend upon the mesh geometry only.

The cubicFit scheme was compared to a multidimensional linear upwind scheme using three idealised numerical tests. The first test transported a tracer horizontally above steep slopes on highly-distorted, two-dimensional terrain-following meshes. The cubicFit scheme was second-order convergent regardless of mesh distortions. The second test transported a tracer over a mountainous lower boundary using terrain-following, cut cell and slanted cell meshes. The cubicFit scheme was generally insensitive to the type of mesh and less sensitive to terrain steepness compared to the multidimensional linear upwind scheme, and the scheme maintained accuracy up to its stability limit. The third test evaluated the transport schemes in a standard deformational flow field on hexagonal–icosahedral meshes and cubed-sphere meshes. In all tests, compared to the multidimensional linear upwind scheme, the cubicFit transport scheme was more stable and more accurate.

Acknowledgements

James Shaw acknowledges support from a PhD studentship funded jointly by NERC grant NE/K500860/1 and the University of Reading with CASE support from the Met Office. We are grateful to the Leibniz Institute for Tropospheric Research for providing their cut cell mesh generator, to the three anonymous reviewers for their helpful questions, and to Dr Shing Hing Man for his assistance with candidate polynomial generation. We also thank Dr Tristan Pryer at the University of Reading for useful discussions about the cubicFit transport scheme.

Appendix A. One-dimensional von Neumann stability analysis

Two analyses are performed in order to find stability conditions that are used to constrain the weights $\mathbf{w} = \tilde{\mathbf{b}}_{\mathbf{1}}^{+} \cdot \mathbf{m}$ as appear in equation (14). The first analysis uses a two-cell approximation to derive separate stability conditions involving the upwind weight w_u and downwind weight w_d. The second analysis uses three cells to derive a stability condition that involves all weights in a stencil.

Two-cell analysis

We start with the conservation equation for a dependent variable ϕ that is discrete-in-space and continuous-in-time

$$\frac{\partial \phi_j}{\partial t} = -v \frac{\phi_R - \phi_L}{\Delta x} \tag{44}$$

where v is the velocity, and the left and right fluxes, ϕ_L and ϕ_R, are weighted averages of the neighbouring cell centres. Assuming that v is positive

$$\phi_L = \alpha_u \phi_{j-1} + \alpha_d \phi_j \tag{45}$$

$$\phi_R = \beta_u \phi_j + \beta_d \phi_{j+1} \tag{46}$$

where $\phi_{j-1}, \phi_j, \phi_{j+1}$ are cell centre values, and j denotes a cell centre position $x = j\Delta x$ where Δx is a uniform mesh spacing. α_u and β_u are the upwind weights and α_d and β_d are the downwind weights for the left and right fluxes respectively, and $\alpha_u + \alpha_d = 1$ and $\beta_u + \beta_d = 1$.

At a given time $t = n\Delta t$ at time-level n and with a time-step Δt, we assume a wave-like solution with an amplification factor A, such that

$$\phi_j^{(n)} = A^n e^{i\,jk\Delta x} \tag{47}$$

where $\phi_j^{(n)}$ denotes a value of ϕ at position j and time-level n. Using this to rewrite the left-hand side of equation (44)

$$\frac{\partial \phi_j}{\partial t} = \frac{\partial}{\partial t}\left(A^{t/\Delta t}\right) e^{ijk\Delta x} = \frac{\ln A}{\Delta t} A^n e^{ikj\Delta x} \tag{48}$$

hence equation (44) becomes

$$\frac{\ln A}{\Delta t} = -\frac{v}{\Delta x}\left(\beta_u + \beta_d e^{ik\Delta x} - \alpha_u e^{-ik\Delta x} - \alpha_d\right) \tag{49}$$

$$\ln A = -c\left(\beta_u - \alpha_d + \beta_d \cos k\Delta x + i\beta_d \sin k\Delta x - \alpha_u \cos k\Delta x + i\alpha_u \sin k\Delta x\right) \tag{50}$$

where the Courant number $c = v\Delta t/\Delta x$. Let $\Re = \beta_u - \alpha_d + \beta_d \cos k\Delta x - \alpha_u \cos k\Delta x$ and $\Im = \beta_d \sin k\Delta x + \alpha_u \sin k\Delta x$, then

$$\ln A = -c\left(\Re + i\Im\right) \tag{51}$$

$$A = e^{-c\Re} e^{-ic\Im} \tag{52}$$

and the complex modulus of A is

$$|A| = e^{-c\Re} = \exp\left(-c\left(\beta_u - \alpha_d + (\beta_d - \alpha_u)\cos k\Delta x\right)\right)\,. \tag{53}$$

For stability we need $|A| \le 1$ and, imposing the additional constraints that $\alpha_u = \beta_u$ and $\alpha_d = \beta_d$, then

$$(\alpha_u - \alpha_d)(1 - \cos k\Delta x) \ge 0 \quad \forall k\Delta x \tag{54}$$

and, given $0 \le 1 - \cos k\Delta x \le 2$, then

$$\alpha_u - \alpha_d \ge 0\,. \tag{55}$$

Additionally, we do not want more damping than a first-order upwind scheme (where $\alpha_u = \beta_u = 1$, $\alpha_d = \beta_d = 0$), having an amplification factor, A_{up}, so we need $|A| \ge |A_{\text{up}}|$, hence

$$\exp\left(-c\left(\alpha_u - \alpha_d\right)(1 - \cos k\Delta x)\right) \ge \exp\left(-c\left(1 - \cos k\Delta x\right)\right) \quad \forall k\Delta x \tag{56}$$

therefore

$$\alpha_u - \alpha_d \le 1\,. \tag{57}$$

Now, knowing that $\alpha_u + \alpha_d = 1$ (or $\alpha_d = 1 - \alpha_u$) then, using equations (55) and (57), we obtain the first two stability conditions,

$$0.5 \le \alpha_u \le 1 \text{ and} \tag{58}$$

$$0 \le \alpha_d \le 0.5\,. \tag{59}$$

Three-cell analysis

We start again from equation (44) but this time approximate ϕ_L and ϕ_R using three cell centre values,

$$\phi_L = \alpha_{uu}\phi_{j-2} + \alpha_u \phi_{j-1} + \alpha_d \phi_j \tag{60}$$

$$\phi_R = \alpha_{uu}\phi_{j-1} + \alpha_u \phi_j + \alpha_d \phi_{j+1} \tag{61}$$

having used the same weights α_{uu}, α_u and α_d for both left and right fluxes. Substituting equation (47) into equation (44) we find

$$A = \exp\left(-c\left[\alpha_{uu}\left(e^{-ik\Delta x} - e^{-2ik\Delta x}\right) + \alpha_u\left(1 - e^{-ik\Delta x}\right) + \alpha_d\left(e^{ik\Delta x} - 1\right)\right]\right) \tag{62}$$

so that, if the complex modulus $|A| \leq 1$ then

$$\alpha_u - \alpha_d + (\alpha_{uu} - \alpha_u + \alpha_d)\cos k\Delta x - \alpha_{uu}\cos 2k\Delta x \geq 0\,. \tag{63}$$

If $k\Delta x = \pi$ then $\cos k\Delta x = -1$ and $\cos 2k\Delta x = 1$ and $\alpha_u - \alpha_d \geq \alpha_{uu}$. If $k\Delta x = \pi/2$ then $\cos k\Delta x = 0$ and $\cos 2k\Delta x = -1$ and $\alpha_u - \alpha_d \geq -\alpha_{uu}$. Hence we find that

$$\alpha_u - \alpha_d \geq |\alpha_{uu}|\,. \tag{64}$$

When the same analysis is performed with four cells, α_{uuu}, α_{uu}, α_u and α_d, by varying $k\Delta x$ we find that equation (64) holds replacing $|\alpha_{uu}|$ with $\max(|\alpha_{uu}|, |\alpha_{uuu}|)$. Hence, we generalise equation (64) to obtain the final stability condition

$$\alpha_u - \alpha_d \geq \max_{p \in P} |\alpha_p| \tag{65}$$

where the peripheral cells P is the set of all stencil cells except for the upwind cell and downwind cell, and α_p is the weight for a given peripheral cell p. We hypothesise that the three stability conditions (equations (58), (59) and (64)) are necessary but not sufficient for a transport scheme on arbitrary meshes.

The stability of the one-dimensional transport equation discretised in space and time could be analysed using existing techniques [55], but we have only analysed the spatial stability of the cubicFit scheme. Numerical experiments presented in section 3.2 demonstrate that the cubicFit scheme is generally insensitive to the time-step, provided that it is below a stability limit.

Appendix B. Mesh geometry on a spherical Earth

The cubicFit transport scheme is implemented using the OpenFOAM CFD library. Unlike many atmospheric models that use spherical coordinates, OpenFOAM uses global, three-dimensional Cartesian coordinates with the z-axis pointing up through the North pole. In order to perform the experiments on a spherical Earth presented in section 3.3, it is necessary for velocity fields and mesh geometries to be expressed in these global Cartesian coordinates.

Velocity field specification

The non-divergent velocity field in section 3.3 is specified as a streamfunction $\Psi(\lambda, \theta)$. Instead of calculating velocity vectors, the flux $\mathbf{u}_f \cdot \mathbf{S}_f$ through a face f is calculated directly from the streamfunction,

$$\mathbf{u}_f \cdot \mathbf{S}_f = \sum_{e \in f} \mathbf{e} \cdot \mathbf{x}_e \Psi(e) \tag{66}$$

where $e \in f$ denotes the edges e of face f, $\mathbf{e}$ is the edge vector joining the two vertices of the edge, $\mathbf{x}_e$ is the position vector of the edge midpoint, and $\Psi(e)$ is the streamfunction evaluated at the same position. Edge vectors are directed in a counter-clockwise orientation.

Spherical mesh construction

Since OpenFOAM does not support two-dimensional spherical meshes, instead, we construct meshes that have a single layer of cells that are 2000 m deep, having an inner radius $r_1 = R_e - 1000$ m and an outer radius $r_2 = R_e + 1000$ m. By default, OpenFOAM meshes comprise polyhedral cells with straight edges and flat faces. This is problematic for spherical meshes because face areas and cell volumes are too small. For tests on a spherical Earth, we override the default configuration and calculate our own face areas, cell volumes, face centres and cell centres that account for the mesh curvature. Note that the new centres are no longer centroids, but they are consistent with the horizontal transport tests on a sphere presented in section 3.3.

A face is classified as either a surface face or radial face. A surface face has any number of vertices, all of equal radius. A radial face has four vertices with two different radii, r_1 and r_2, and two different horizontal coordinates, (λ_1, θ_1) and (λ_2, θ_2). A radial face centre is modified so that it has a radius R_e. The latitudinal and longitudinal components of a radial face centre need no modification. The face area A_f for a radial face f is the area of the annular sector,

$$A_f = \frac{d}{2}\left|r_2^2 - r_1^2\right| \tag{67}$$

where d is the great-circle distance between (λ_1, θ_1) and (λ_2, θ_2).

To calculate the centre of a surface face f, a new vertex is created that is positioned at the mean of the face vertices. Note that this centre position, $\tilde{\mathbf{c}}_f$, is used in intermediate calculations and it is not the face centre position. Next, the surface face is subdivided into spherical triangles that share this new vertex [61]. The face centre direction and radius are calculated separately. The face centre direction $\hat{\mathbf{r}}$ is the mean of the spherical triangle centres weighted by their solid angle,

$$\hat{\mathbf{r}} = \frac{\sum_{t \in f} \Omega_t \left(\mathbf{x}_{t,1} + \mathbf{x}_{t,2} + \tilde{\mathbf{c}}_f \right)}{\left| \sum_{t \in f} \Omega_t \left(\mathbf{x}_{t,1} + \mathbf{x}_{t,2} + \tilde{\mathbf{c}}_f \right) \right|} \tag{68}$$

where $t \in f$ denotes the spherical triangles t of face f, Ω_t is spherical triangle's solid angle which is calculated using l'Huilier's theorem, $\mathbf{x}_{t,1}$ and $\mathbf{x}_{t,2}$ are the positions of the vertices shared by the face f and spherical triangle t, and $\tilde{\mathbf{c}}_f$ is the position of the centre vertex shared by all spherical triangles of face f. The face centre radius r is the mean radius of the face vertices, again weighted by the solid angle of each spherical triangle,

$$r = \frac{\sum_{t \in f} \Omega_t \left(\left| \mathbf{x}_{t,1} \right| + \left| \mathbf{x}_{t,2} \right| \right) / 2}{\Omega_f} \tag{69}$$

where the solid angle Ω_f of face f is the sum of the solid angles of the constituent spherical triangles,

$$\Omega_f = \sum_{t \in f} \Omega_t \,. \tag{70}$$

We use equations (68) and (69) to calculate the centre $\mathbf{c}_f$ of the face f,

$$\mathbf{c}_f = r\,\hat{\mathbf{r}} \tag{71}$$

The area vector $\mathbf{S}_f$ of the surface face f is the sum of the spherical triangle areas [61],

$$\mathbf{S}_f = r^2 \Omega_f \hat{\mathbf{r}} \,. \tag{72}$$

Cell centres and cell volumes are corrected by considering faces that are not normal to the sphere such that

$$\frac{\left(\mathbf{S}_f \cdot \mathbf{c}_f \right)^2}{\left| \mathbf{S}_f \right|^2 \left| \mathbf{c}_f \right|^2} > 0 \,. \tag{73}$$

Let $\mathcal{F}$ be the set of faces satisfying equation (73). Then, the cell volume $\mathcal{V}_c$ is

$$\mathcal{V}_c = \frac{1}{3} \sum_{f \in \mathcal{F}} \mathbf{S}_f \cdot \mathbf{c}_f \tag{74}$$

which can be thought of as the area A integrated between r_1 and r_2 such that $\int_0^R A(r)\,\mathrm{d}r = \int_{r_1}^{r_2} r^2 \Omega \,\mathrm{d}r = \frac{1}{3}\Omega \left(r_2^3 - r_1^3 \right)$. The cell centre is modified so that it has a radius R_e, which is consistent with radial faces.

Edges can be classified in a similar manner to faces where surface edges are tangent to the sphere and radial faces are normal to the sphere. The edge midpoints $\mathbf{x}_e$ are used to calculate the face flux for non-divergent velocity fields (equation (66)). For transport tests, corrections to edge midpoints are unnecessary. Due to the choice of r_1 and r_2 during mesh construction, the midpoint of a radial edge is at a radial distance of R_e which is necessary for the correct calculation of non-divergent velocity fields. The position of surface edge midpoints is unimportant because these edges do not contribute to the face flux since $\mathbf{e} \cdot \mathbf{x}_e = 0$. Edge lengths are the straight-line distance between the two vertices and not the great-circle distance. Again, the edge lengths are not corrected because it makes no difference to the face flux calculation.

References

[1] R.L. Walko, R. Avissar, The Ocean-Land-Atmosphere Model (OLAM). Part II: formulation and tests of the nonhydrostatic dynamic core, Mon. Weather Rev. 136 (11) (2008) 4045–4062, http://dx.doi.org/10.1175/2008MWR2523.1.

[2] C. Schär, D. Leuenberger, O. Fuhrer, D. Lüthi, C. Girard, A new terrain-following vertical coordinate formulation for atmospheric prediction models, Mon. Weather Rev. 130 (10) (2002) 2459–2480, http://dx.doi.org/10.1175/1520-0493(2002)130<2459:ANTFVC>2.0.CO;2.

[3] K.P. Hoinka, G. Zängl, The influence of the vertical coordinate on simulations of a PV streamer crossing the alps, Mon. Weather Rev. 132 (7) (2004) 1860–1867, http://dx.doi.org/10.1175/1520-0493(2004)132<1860:TIOTVC>2.0.CO;2.

[4] S. Webster, A. Brown, D. Cameron, C. Jones, Improvements to the representation of orography in the Met Office Unified Model, Q. J. R. Meteorol. Soc. 129 (591) (2003) 1989–2010, http://dx.doi.org/10.1256/qj.02.133.

[5] T. Gal-Chen, R.C. Somerville, On the use of a coordinate transformation for the solution of the Navier–Stokes equations, J. Comput. Phys. 17 (2) (1975) 209–228, http://dx.doi.org/10.1016/0021-9991(75)90037-6.

[6] J.B. Klemp, A terrain-following coordinate with smoothed coordinate surfaces, Mon. Weather Rev. 139 (7) (2011) 2163–2169, http://dx.doi.org/10.1175/MWR-D-10-05046.1.

[7] S.D. Eckermann, J.P. McCormack, J. Ma, T.F. Hogan, K.A. Zawdie, Stratospheric analysis and forecast errors using hybrid and sigma coordinates, Mon. Weather Rev. 142 (1) (2014) 476–485, http://dx.doi.org/10.1175/MWR-D-13-00203.1.

[8] A.J. Simmons, D.M. Burridge, An energy and angular-momentum conserving vertical finite-difference scheme and hybrid vertical coordinates, Mon. Weather Rev. 109 (4) (1981) 758–766, http://dx.doi.org/10.1175/1520-0493(1981)109<0758:AEAAMC>2.0.CO;2.

[9] D. Leuenberger, M. Koller, O. Fuhrer, C. Schär, A generalization of the SLEVE vertical coordinate, Mon. Weather Rev. 138 (9) (2010) 3683–3689, http://dx.doi.org/10.1175/2010MWR3307.1.

[10] H. Yamazaki, T. Satomura, N. Nikiforakis, Three-dimensional cut-cell modelling for high-resolution atmospheric simulations, Q. J. R. Meteorol. Soc. 142 (696) (2016) 1335–1350, http://dx.doi.org/10.1002/qj.2736.

[11] J. Steppeler, H.-W. Bitzer, M. Minotte, L. Bonaventura, Nonhydrostatic atmospheric modeling using a z-coordinate representation, Mon. Weather Rev. 130 (8) (2002) 2143–2149, http://dx.doi.org/10.1175/1520-0493(2002)130<2143:NAMUAZ>2.0.CO;2.
[12] S. Jebens, O. Knoth, R. Weiner, Partially implicit peer methods for the compressible Euler equations, J. Comput. Phys. 230 (12) (2011) 4955–4974, http://dx.doi.org/10.1016/j.jcp.2011.03.015.
[13] J. Shaw, H. Weller, Comparison of terrain following and cut cell grids using a non-hydrostatic model, Mon. Weather Rev. 144 (6) (2016) 2085–2099, http://dx.doi.org/10.1175/MWR-D-15-0226.1.
[14] B. Leonard, M. MacVean, A. Lock, Positivity-Preserving Numerical Schemes for Multidimensional Advection, Tech. Rep. 106055, NASA, 1993.
[15] J. Kent, P.A. Ullrich, C. Jablonowski, Dynamical core model intercomparison project: tracer transport test cases, Q. J. R. Meteorol. Soc. 140 (681) (2014) 1279–1293, http://dx.doi.org/10.1002/qj.2208.
[16] B. Leonard, A. Lock, M. MacVean, Conservative explicit unrestricted-time-step multidimensional constancy-preserving advection schemes, Mon. Weather Rev. 124 (11) (1996) 2588–2606, http://dx.doi.org/10.1175/1520-0493(1996)124<2588:CEUTSM>2.0.CO;2.
[17] Y. Chen, H. Weller, S. Pring, J. Shaw, Dimension splitting errors and a long time-step multi-dimensional scheme for atmospheric transport, Q. J. R. Meteorol. Soc. (2017), in press, arXiv:1701.06907.
[18] S.-J. Lin, R.B. Rood, Multidimensional flux-form semi-Lagrangian transport schemes, Mon. Weather Rev. 124 (9) (1996) 2046–2070, http://dx.doi.org/10.1175/1520-0493(1996)124<2046:MFFSLT>2.0.CO;2.
[19] W. Guo, R.D. Nair, J.-M. Qiu, A conservative semi-Lagrangian discontinuous Galerkin scheme on the cubed sphere, Mon. Weather Rev. 142 (1) (2014) 457–475, http://dx.doi.org/10.1175/MWR-D-13-00048.1.
[20] W.M. Putman, S.-J. Lin, Finite-volume transport on various cubed-sphere grids, J. Comput. Phys. 227 (1) (2007) 55–78, http://dx.doi.org/10.1016/j.jcp.2007.07.022.
[21] A. Iske, M. Käser, Conservative semi-Lagrangian advection on adaptive unstructured meshes, Numer. Methods Partial Differ. Equ. 20 (3) (2004) 388–411, http://dx.doi.org/10.1002/num.10100.
[22] P.H. Lauritzen, R.D. Nair, P.A. Ullrich, A conservative semi-Lagrangian multi-tracer transport scheme (CSLAM) on the cubed-sphere grid, J. Comput. Phys. 229 (5) (2010) 1401–1424, http://dx.doi.org/10.1016/j.jcp.2009.10.036.
[23] P.H. Lauritzen, C. Jablonowski, M.A. Taylor, R.D. Nair, Numerical Techniques for Global Atmospheric Models, vol. 80, Springer Science & Business Media, 2011.
[24] M. Zerroukat, T. Allen, A three-dimensional monotone and conservative semi-Lagrangian scheme (SLICE-3D) for transport problems, Q. J. R. Meteorol. Soc. 138 (667) (2012) 1640–1651, http://dx.doi.org/10.1002/qj.1902.
[25] R.K. Lashley, Automatic Generation of Accurate Advection Schemes on Unstructured Grids and Their Application to Meteorological Problems, Ph.D. thesis, University of Reading, 2002.
[26] W.C. Skamarock, M. Menchaca, Conservative transport schemes for spherical geodesic grids: high-order reconstructions for forward-in-time schemes, Mon. Weather Rev. 138 (12) (2010) 4497–4508, http://dx.doi.org/10.1175/2010MWR3390.1.
[27] P.H. Lauritzen, C. Erath, R. Mittal, On simplifying 'incremental remap'-based transport schemes, J. Comput. Phys. 230 (22) (2011) 7957–7963, http://dx.doi.org/10.1016/j.jcp.2011.06.030.
[28] J. Thuburn, C. Cotter, T. Dubos, A mimetic, semi-implicit, forward-in-time, finite volume shallow water model: comparison of hexagonal–icosahedral and cubed-sphere grids, Geosci. Model Dev. 7 (3) (2014) 909–929, http://dx.doi.org/10.5194/gmd-7-909-2014.
[29] H. Weller, H.G. Weller, A. Fournier, Voronoi, Delaunay, and block-structured mesh refinement for solution of the shallow-water equations on the sphere, Mon. Weather Rev. 137 (12) (2009) 4208–4224, http://dx.doi.org/10.1175/2009MWR2917.1.
[30] W.C. Skamarock, A. Gassmann, Conservative transport schemes for spherical geodesic grids: high-order flux operators for ODE-based time integration, Mon. Weather Rev. 139 (9) (2011) 2962–2975, http://dx.doi.org/10.1175/MWR-D-10-05056.1.
[31] J. Steppeler, J. Klemp, Advection on cut-cell grids for an idealized mountain of constant slope, Mon. Weather Rev. 145 (5) (2017) 1765–1777, http://dx.doi.org/10.1175/MWR-D-16-0308.1.
[32] P. Lauritzen, P. Ullrich, C. Jablonowski, P. Bosler, D. Calhoun, A. Conley, T. Enomoto, L. Dong, S. Dubey, O. Guba, et al., A standard test case suite for two-dimensional linear transport on the sphere: results from a collection of state-of-the-art schemes, Geosci. Model Dev. 7 (1) (2014) 105–145, http://dx.doi.org/10.5194/gmd-7-105-2014.
[33] S. Ii, F. Xiao, High order multi-moment constrained finite volume method. Part I: Basic formulation, J. Comput. Phys. 228 (10) (2009) 3669–3707, http://dx.doi.org/10.1016/j.jcp.2009.02.009.
[34] X. Li, C. Chen, X. Shen, F. Xiao, A multimoment constrained finite-volume model for nonhydrostatic atmospheric dynamics, Mon. Weather Rev. 141 (4) (2013) 1216–1240, http://dx.doi.org/10.1175/MWR-D-12-00144.1.
[35] B. Xie, F. Xiao, A multi-moment constrained finite volume method on arbitrary unstructured grids for incompressible flows, J. Comput. Phys. 327 (2016) 747–778.
[36] P.K. Smolarkiewicz, J. Szmelter, A nonhydrostatic unstructured-mesh soundproof model for simulation of internal gravity waves, Acta Geophys. 59 (6) (2011) 1109–1134, http://dx.doi.org/10.2478/s11600-011-0043-z.
[37] P.K. Smolarkiewicz, J. Szmelter, MPDATA: an edge-based unstructured-grid formulation, J. Comput. Phys. 206 (2) (2005) 624–649, http://dx.doi.org/10.1016/j.jcp.2004.12.021.
[38] C. Kühnlein, P.K. Smolarkiewicz, An unstructured-mesh finite-volume MPDATA for compressible atmospheric dynamics, J. Comput. Phys. 334 (2017) 16–30, http://dx.doi.org/10.1016/j.jcp.2016.12.054.
[39] H. Weller, A. Shahrokhi, Curl free pressure gradients over orography in a solution of the fully compressible Euler equations with implicit treatment of acoustic and gravity waves, Mon. Weather Rev. 142 (12) (2014) 4439–4457, http://dx.doi.org/10.1175/MWR-D-14-00054.1.
[40] L. Cueto-Felgueroso, I. Colominas, J. Fe, F. Navarrina, M. Casteleiro, High-order finite volume schemes on unstructured grids using moving least-squares reconstruction. Application to shallow water dynamics, Int. J. Numer. Methods Eng. 65 (3) (2006) 295–331, http://dx.doi.org/10.1002/nme.1442.
[41] L. Cueto-Felgueroso, I. Colominas, X. Nogueira, F. Navarrina, M. Casteleiro, Finite volume solvers and moving least-squares approximations for the compressible Navier–Stokes equations on unstructured grids, Comput. Methods Appl. Mech. Eng. 196 (45) (2007) 4712–4736, http://dx.doi.org/10.1016/j.cma.2007.06.003.
[42] R.P. Laurent White, D. Trenev, Flow simulation in heterogeneous porous media with the moving least-squares method, SIAM J. Sci. Comput. 39 (2) (2017) B323–B351, http://dx.doi.org/10.1137/16M1070840.
[43] P.H. Lauritzen, W.C. Skamarock, M. Prather, M. Taylor, A standard test case suite for two-dimensional linear transport on the sphere, Geosci. Model Dev. 5 (3) (2012) 887–901, http://dx.doi.org/10.5194/gmd-5-887-2012.
[44] D.R. Durran, Numerical Methods for Wave Equations in Geophysical Fluid Dynamics, vol. 32, Springer Science & Business Media, 2013.
[45] CFD Direct, OpenFOAM user guide: numerical schemes, http://cfd.direct/openfoam/user-guide/fvschemes/, 2016.
[46] H. Weller, J. Shaw, AtmosFOAM: OpenFOAM applications and libraries for atmospheric modelling, http://dx.doi.org/10.5281/zenodo.247327, 2017.
[47] Leibniz Institute for Tropospheric Research, J. Shaw, ASAM cut cell grid generator, http://dx.doi.org/10.5281/zenodo.242374, 2017.
[48] J. Shaw, GMV-to-OpenFOAM file format converter, http://dx.doi.org/10.5281/zenodo.242387, 2017.
[49] J. Thuburn, C.J. Cotter, T. Dubos, H. Weller, J. Shaw, Hexagonal/triangular geodesic mesh generator, http://dx.doi.org/10.5281/zenodo.245327, 2017.
[50] J. Shaw, AtmosTests: idealised atmospheric test suite, http://dx.doi.org/10.5281/zenodo.247480, 2017.

[51] J. Shaw, Test result data for idealised numerical atmospheric transport, http://dx.doi.org/10.5281/zenodo.259328, 2017.
[52] B. Good, A. Gadian, S.-J. Lock, A. Ross, Performance of the cut-cell method of representing orography in idealized simulations, Atmos. Sci. Lett. 15 (1) (2014) 44–49, http://dx.doi.org/10.1002/asl2.465.
[53] M. Jähn, O. Knoth, M. König, U. Vogelsberg, ASAM v2.7: a compressible atmospheric model with a Cartesian cut cell approach, Geosci. Model Dev. 8 (2) (2015) 317–340, http://dx.doi.org/10.5194/gmd-8-317-2015.
[54] U. Vogelsberg, ASAM Wiki: grid generator, http://asamwiki.tropos.de/index.php?title=Grid_Generator, 2010.
[55] M. Baldauf, Stability analysis for linear discretisations of the advection equation with Runge–Kutta time integration, J. Comput. Phys. 227 (13) (2008) 6638–6659, http://dx.doi.org/10.1016/j.jcp.2008.03.025.
[56] R. Heikes, D.A. Randall, Numerical integration of the shallow-water equations on a twisted icosahedral grid. Part I: basic design and results of tests, Mon. Weather Rev. 123 (6) (1995) 1862–1880, http://dx.doi.org/10.1175/1520-0493(1995)123<1862:NIOTSW>2.0.CO;2.
[57] R. Heikes, D.A. Randall, 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 (6) (1995) 1881–1887, http://dx.doi.org/10.1175/1520-0493(1995)123<1881:NIOTSW>2.0.CO;2.
[58] A. Staniforth, J. Thuburn, Horizontal grids for global weather and climate prediction models: a review, Q. J. R. Meteorol. Soc. 138 (662) (2012) 1–26, http://dx.doi.org/10.1002/qj.958.
[59] J. White, J.J. Dongarra, High-performance high-resolution semi-Lagrangian tracer transport on a sphere, J. Comput. Phys. 230 (17) (2011) 6778–6799, http://dx.doi.org/10.1016/j.jcp.2011.05.008.
[60] H. Miura, An upwind-biased conservative advection scheme for spherical hexagonal–pentagonal grids, Mon. Weather Rev. 135 (12) (2007) 4038–4044, http://dx.doi.org/10.1175/2007MWR2101.1.
[61] G. Van Brummelen, Heavenly Mathematics: The Forgotten Art of Spherical Trigonometry, Princeton University Press, 2013.