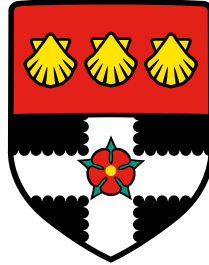UNIVERSITY OF READING

DEPARTMENT OF METEOROLOGY

Numerical representation of mountains in atmospheric models

James Shaw

PhD Atmosphere Ocean and Climate

March 2018

Declaration: I confirm that this is my own work and the use of all material from other sources has been properly and fully acknowledged. — James Shaw

Abstract

Numerical weather and climate models are using increasingly fine meshes that resolve small-scale, steeply-sloping terrain. Terrain-following meshes become highly distorted above such steep slopes, degrading the numerical balance between the pressure gradient and gravity. Furthermore, existing models often prefer dimensionally-split transport schemes for their computational efficiency, but such schemes can suffer from splitting errors above steep slopes. The cut cell method offers an alternative that avoids most mesh distortions, but arbitrarily small cut cells can impose severe time-step constraints on explicit transport schemes. This thesis makes three contributions to improve atmospheric simulations, particularly in the vicinity of steeply-sloping terrain.

First, a multidimensional finite volume transport scheme is formulated to obtain accurate solutions on arbitrary, highly-distorted meshes. Stability conditions derived from a von Neumann stability analysis are imposed during model initialisation to obtain stability and improve accuracy near steeply-sloping lower boundaries. Reconstruction calculations depend upon the mesh only, needing just one vector multiply per face per time-stage. The scheme achieves second-order convergence across a series of tests using highly-distorted terrain-following meshes and cut cell meshes. The scheme is extended using the k-exact method to achieve third-order convergence on distorted meshes without increasing the computational cost during integration.

Second, a new type of mesh is designed to avoid severe mesh distortions associated with terrain-following meshes and avoids severe time-step constraints associated with cut cells. Numerical experiments compare the new mesh with terrain-following and cut cell meshes, revealing that the new mesh simultaneously achieves an accurate balance between the pressure gradient and gravity, and avoids severe time-step constraints.

Third, a new two-dimensional test case is proposed that excites the Lorenz computational mode. The new test is used to compare results from a nonhydrostatic model with Lorenz staggering with those from a model variant with a newly-developed generalised Charney–Phillips staggering for arbitrary meshes.

Contents

1 Introduction

Atmospheric models are using increasingly fine mesh spacing to resolve small-scale processes and improve weather and climate forecasts (Wedi, 2014). These finer meshes resolve small-scale, steeply sloping terrain that is poorly represented by traditional terrain-following meshes and traditional numerical methods (Schär et al., 2002), motivating research into alternative vertical meshes including improved terrain-following meshes (Schär et al., 2002; Klemp, 2011) and cut-cell meshes (Jähn et al., 2015; Yamazaki et al., 2016), and improved numerical methods (Zängl, 2012; Steppeler and Klemp, 2017).

Traditional terrain-following meshes become highly distorted over steeply sloping terrain, resulting in increased numerical errors: transport across distorted terrain-following mesh layers can produce spurious numerical diffusion and grid-scale oscillations (Hoinka and Zängl, 2004; Schär et al., 2002), and spurious flows result from errors in the calculation of the pressure gradient where terrain-following mesh layers are not perpendicular to gravity (Klemp, 2011; Zängl, 2012). Cut cell meshes can reduce numerical errors associated with transport and pressure gradient calculations (Good et al., 2014), but the cut cell method creates meshes with arbitrarily small cut cells that impose severe time-step constraints on explicit transport schemes (Klein et al., 2009).

1.1 Vertical meshes to represent the atmosphere above terrain

Terrain-following meshes have been in widespread operational use since atmospheric models first included a numerical representation of terrain (Steppeler et al., 2003), with Phillips (1957) having formulated the σ coordinate, also known as the basic terrain-following coordinate or basic terrain-following mesh (Gal-Chen and Somerville, 1975). Above sloping terrain, basic terrain-following meshes distort every model layer , with only the upper boundary being entirely horizontal. Terrain-following mesh distortions become more severe with increasingly steep

slopes, reducing the numerical accuracy of transport schemes and pressure gradient calculations in particular.

In a mesoscale model forecast over the Alps, transport across terrain-following mesh layers produced spurious numerical diffusion in water vapour and relative vorticity fields near the tropopause where vertical gradients are strong (Hoinka and Zängl, 2004). Schär et al. (2002) found that lower-order transport schemes are inaccurate in the presence of basic terrain-following mesh distortions, with the transported tracer exhibiting numerical diffusion and grid-scale oscillations. Furthermore, such errors are not confined to atmospheric models: in coupled ocean/sea-ice model experiments performed by Naughten et al. (2017), an inaccurate transport scheme produced numerical oscillations, leading to supercooling and spurious sea-ice production.

Pressure gradient errors near steep slopes result in spurious circulations that can degrade simulated slope flows, along-valley flows, orographically-induced precipitation and cold air pools (Zängl et al., 2004). Comparing model simulations with field campaign observations in the Salt Lake valley, Fast (2003) found that simulated winds were too strong at night, when observed winds were weak and cold air pools formed. Zängl et al. (2004) performed a model intercomparison using an idealised test with a stratified atmosphere initially at rest above an isolated mountain with steep slopes. After one simulated day, pressure gradient errors produced maximum spurious vertical velocities between $0.4\,\mathrm{m\,s^{-1}}$ and $3\,\mathrm{m\,s^{-1}}$ across different models. Pressure gradient errors are also problematic using terrain-following meshes to represent steep ocean bathymetry: Luo et al. (2002) simulated an ocean initially at rest above an isolated seamount, and found that spurious vertical velocities increased with steeper seamount slopes.

To improve the accuracy of transport schemes and pressure gradient calculations, terrain-following mesh layers can be smoothed so that mesh distortions further above the lower boundary are reduced. While the layers of a basic terrain-following mesh are distorted throughout the domain, the layers of a hybrid terrain-following mesh are gradually flattened and become purely horizontal at a specified height below the domain top (Simmons and Burridge, 1981). Compared to the basic terrain-following mesh, the hybrid terrain-following mesh has been found to improve forecasts, particularly in the stratosphere where hybrid terrain-following mesh layers are horizontal (Eckermann et al., 2014). Variants of the hybrid terrain-following mesh have become widely adopted in atmospheric models (Davies et al., 2005; Donner et al., 2011) as well as some ocean models (Burchard and Petersen, 1997; Halliwell, 2004). More sophisticated methods have been developed that produce even smoother terrain-following meshes, including the smooth level vertical (SLEVE) mesh (Schär et al., 2002; Leuenberger et al., 2010) used in the icosahedral nonhydrostatic ICON model (Zängl et al., 2015).

Despite their associated numerical errors, terrain-following meshes are attractive because

their rectangular structure is simple to process by computer, they can be straightforwardly linked with parameterization schemes, and the boundary layer resolution can be improved simply by using variable spacing of vertical layers (Schär et al., 2002). Nevertheless, terrain-following meshes cannot avoid distortions near the surface, and terrain-following cell volumes approach zero as sloping terrain approaches a 90° cliff. The cut cell mesh is an alternative in which the mesh does not follow the terrain but, instead, cells that lie entirely below the terrain are removed, and those that intersect the surface are modified in shape so that they more closely fit the terrain. The resulting mesh is entirely undistorted except for cells that have been cut. Cut cells are also known as shaved cells (Adcroft et al., 1997; Adcroft, 2013) or embedded boundaries (Schwartz et al., 2015; Devendran et al., 2017).

The cut cell method can create arbitrarily small cells that severely constrain the maximum time-step for explicit methods (Klein et al., 2009), and several approaches have been tried in order to alleviate the problem. Yamazaki and Satomura (2010) combine small cells with horizontally or vertically adjacent cells. Steppeler et al. (2002) employ a thin-wall approximation to increase the computational volume of small cells without altering the terrain. Jebens et al. (2011) avoid the time-step restriction associated with explicit schemes by using an implicit method for cut cells and a semi-explicit method elsewhere.

In an idealised test with a stratified atmosphere initially at rest above a mountain, Good et al. (2014) found that spurious circulations became increasingly severe with increasingly steep slopes represented by terrain-following meshes, but such errors were eliminated by using cut cell meshes. In a comparison of terrain-following and cut cell meshes using real initial data, Steppeler et al. (2013) found that 5-day forecasts of precipitation and wind over Asia were more accurate in the cut cell model, although this result depended upon an old version of a model being used.

Another alternative method for representing terrain is the Eta coordinate (Mesinger et al., 1988), which creates terrain profiles having a staircase pattern. Mesinger et al. (1988) found that the Eta coordinate improves the accuracy of pressure gradient calculations compared to basic terrain-following meshes, and Mesinger et al. (2012) later refined the formulation to allow diagonal transport of momentum and temperature immediately above sloping terrain, making the Eta coordinate similar to the cut cell method.

1.2 Horizontal meshes to represent a spherical Earth

We have seen that increasingly fine mesh spacing poses problems for traditional terrain-following meshes and traditional numerical methods, but further numerical and computational issues also arise with finer meshes. Traditionally, global atmospheric models have used uniform latitude-

longitude meshes to represent a spherical Earth but, with increasingly fine horizontal mesh spacing, the cells of latitude-longitude meshes become very small near the Earth's poles, causing a bottleneck in parallel computation (Staniforth and Thuburn, 2012) and placing severe time-step constraints on explicit methods. In addition to the small cell problem near the poles, computer storage and computation time increase dramatically when horizontal mesh spacing is reduced uniformly over a latitude-longitude mesh: halving the horizontal mesh spacing results in four times as many cells and simulations require a smaller time-step.

In response to these problems, a variety of alternative horizontal representations have been proposed. Alternative, quasi-uniform meshes avoid small cells near the poles of latitude-longitude meshes (Staniforth and Thuburn, 2012), and some models are already using quasi-uniform meshes: the ICON model uses an icosahedral mesh (Zängl et al., 2015), the Global Environmental Multiscale model uses a Yin-Yang mesh comprising two overlapping sections arranged like a tennis ball (Qaddouri and Lee, 2011), and the Met Office are preferring a cubed-sphere mesh for their next-generation GungHo model (N. Wood 2017, personal communication). To improve the scalability of computational resources with finer mesh spacing, static mesh refinement and dynamic adaptive mesh techniques create meshes with fewer cells while retaining the numerical accuracy achieved with a uniformly fine mesh (Jablonowski et al., 2009).

These alternative meshes alleviate many of the computational and numerical problems that arise due to finer horizontal mesh spacing, but they introduce problems of their own. Unlike latitude-longitude meshes, quasi-uniform meshes have non-zero skewness or non-orthogonality that produces grid imprinting errors and excites computational modes (Weller et al., 2012). Mesh refinement and adaptive mesh techniques also create mesh geometries with non-orthogonalities or hanging nodes (Marras et al., 2016).

Some recent studies have applied mesh refinement and adaptive mesh techniques to vertical meshes to better resolve cloud processes (Müller et al., 2013) and flows over mountains (Yamazaki and Satomura, 2012). The vertical discretisation used by Yamazaki and Satomura (2012) supports a computational mode (Thuburn and Woollings, 2005) that can be avoided by using an alternative staggering of variables. The Charney–Phillips staggering is free from computational modes, but the Charney–Phillips staggering has yet to be generalised for arbitrary vertical meshes.

1.3   Aims and research outline

With such a wide choice of horizontal and vertical meshes and numerical schemes, it is important that next-generation atmospheric models are designed so that the choice of mesh and choices of numerical schemes can be deferred until later in the development process, or changed during

operation as new techniques emerge (Ford et al., 2013; Theurich et al., 2016). This thesis makes four contributions to improve numerical accuracy for flows over steep slopes:

1. a new approach for stabilising finite volume transport schemes on arbitrary meshes,

2. a high-order finite volume formulation for transport on arbitrary meshes,

3. a new vertical mesh for representing terrain,

4. a new test case to excite the Lorenz computational mode on a two-dimensional x–z Cartesian plane.

All numerical tests are performed using two-dimensional flows on Cartesian planes and spherical shells, but there is nothing precluding the extension of all four formulations to three-dimensional geometries.

Throughout the thesis, we use the OpenFOAM software package (The OpenFOAM Foundation) to implement numerical schemes and numerical experiments, enabling like-for-like comparisons between different model variants and different types of mesh. We provide source code archives for the OpenFOAM implementation of the new transport schemes (Weller et al., 2018; Shaw, 2018d) and supporting tools (Shaw et al., 2018; Shaw, 2018e), the ASAM cut cell mesh generator (Leibniz Institute for Tropospheric Research and Shaw, 2018) and associated OpenFOAM converter (Shaw, 2018c), and the hexagonal–icosahedral mesh generator (Thuburn et al., 2018). For the numerical test cases presented, we also supply the source code (Shaw, 2018b) and resulting data (Shaw, 2018f). For convenience, we have pre-installed all necessary software in an executable Singularity container (Shaw, 2018a). The source code for the thesis itself is also available (Shaw, 2018g).

Chapter 2 formulates a new finite volume transport scheme to achieve numerical stability over steep terrain represented by highly-distorted, arbitrary meshes. It is second-order convergent on quasi-uniform spherical meshes, terrain-following and cut cell meshes. Chapter 3 proposes a modification to the formulation, using the k-exact method (Barth, 1995) to achieve higher than second-order convergence in the interior of distorted meshes without increasing the computational cost during integration. Chapter 4 introduces a new type of vertical mesh that simulateously avoids severe mesh distortions associated with traditional terrain-following meshes, while avoiding arbitrarily small cells associated with cut cell meshes. Numerical experiments verify that a more accurate balance is achieved between the pressure gradient and gravity, and we find that the new mesh permits maximum time-steps comparable to those permitted by terrain-following meshes. Chapter 5 extends the work of Arakawa and Konor (1996) to create a new two-dimensional test case that excites the Lorenz computational mode, enabling a more straightforward assessment of models using Lorenz or Charney–Phillips staggerings. Motivated

by the emerging need for vertical mesh refinement and mesh adaptivity, the Charney–Phillips staggering is generalised for arbitrary meshes, and the formulation is assessed using the new two-dimensional test case. Closing remarks are made in chapter 6.

2 Numerically stable transport over steep slopes

Highlights

- The new cubicFit transport scheme is second-order convergent regardless of mesh distortions or the choice of velocity field

- Sub-grid reconstructions are mostly precomputed depending on the mesh geometry alone

- Misalignment of the velocity field with mesh layers is the primary source of numerical error and not simply mesh distortions

---

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 (Kent et al., 2014), resulting in numerical errors called 'splitting errors'. Such errors can be reduced by explicitly accounting for transverse fluxes when combining fluxes (Leonard et al., 1996), 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 (Chen et al., 2017).

Transport schemes are often classified as dimensionally-split or multidimensional. Dimensionally-split schemes such as Lin and Rood (1996); Guo et al. (2014) 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 (Putman and Lin, 2007; Guo

et al., 2014). 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 multi-dimensional, 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 Iske and Käser (2004); Lauritzen et al. (2010) 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 (Lauritzen et al., 2011b). SLICE-3D is a three-dimensional semi-Lagrangian finite volume scheme for latitude-longitude meshes that applies separate conservative remappings in each dimension (Zerroukat and Allen, 2012). Swept-area schemes such as Lashley (2002); Skamarock and Menchaca (2010); Lauritzen et al. (2011a); Thuburn et al. (2014) 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 (Thuburn et al., 2014; Skamarock and Menchaca, 2010). Method-of-lines schemes such as Weller et al. (2009); Skamarock and Gassmann (2011) 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 (Steppeler and Klemp, 2017). 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 (Lauritzen et al., 2011a). 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. (2014).

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 (Kent et al., 2014). 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 (Ii and Xiao, 2009) is a three-dimensional scheme that has been used to simulate non-hydrostatic flows over orography with terrain-following coordinates on a x–z plane (Li et al., 2013). Simulations of subcritical flow around a cylinder have also been performed on a three-dimensional hexahedral-prismatic hybrid mesh (Xie and Xiao, 2016). 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 (Smolarkiewicz and Szmelter, 2011), and in three-dimensional transport tests (Smolarkiewicz and Szmelter, 2005). Most recently, Kühnlein and Smolarkiewicz (2017) extended MPDATA to enable semi-implicit integrations of the compressible Euler equations on arbitrary meshes, and MPDATA has also been extended to achieve third-order convergence (Waruszewski et al., 2018). The three-dimensional method-of-lines scheme developed by Weller and Shahrokhi (2014) has been used in two-dimensional flows over orography on Cartesian x–z planes with distorted meshes (Chen et al., 2017). This finite volume scheme uses a moving weighted least-squares reconstruction (Lashley, 2002; Thuburn et al., 2014) that makes it suitable for arbitrary meshes. Similar least-squares approaches have been applied previously to shallow water flows (Cueto-Felgueroso et al., 2006), aeronautic (Cueto-Felgueroso et al., 2007) and porous media (White et al., 2017) simulations.

This chapter presents a new multidimensional method-of-lines scheme, 'cubicFit', that improves the stability of the scheme by Weller and Shahrokhi (2014) 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. This stabilisation procedure has similarities to the Multidimensional Optimal Order Detection (MOOD) method (Clain et al., 2011; Diot et al., 2013). However, MOOD is an *a posteriori* method that detects discontinuities in the solution. In contrast, the cubicFit stabilisation procedure is an *a priori* method that depends upon the mesh geometry only, and reconstruction weights can be precomputed without knowledge of the velocity field or tracer field. Wang et al. (2018) propose a different stabilisation approach that regularises ill-conditioned matrices that are associated with least-squares fits over highly-distorted mesh regions, though the technique has so far only been tried with element-free Galerkin methods.

For the cubicFit transport scheme, 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 with an $m \times n$ matrix where m is the size of the stencil and n is associated with the order of accuracy. Instead, the computational cost of the cubicFit scheme is more comparable to dimensionally-split schemes, requiring only m multiplies per face per time-stage.

The remainder of this chapter is organised as follows. Section 2.1 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 that is included in the Open-FOAM software distribution (CFD Direct, 2018b). Subsequent sections evaluate the cubicFit scheme in a series of three idealised numerical tests. The test in section 2.2 follows Schär et al. (2002), transporting a tracer horizontally above steep mountains on two-dimensional, highly-distorted terrain-following meshes. Section 2.3 formulates a new tracer transport test that uses a terrain-following velocity field to challenge transport schemes on cut cell meshes. Finally, in section 2.4, we assess the cubicFit transport scheme on hexagonal-icosahedral meshes and cubed-sphere meshes using a standard test of deformational flow on a single-layer spherical Earth, as specified by Lauritzen et al. (2012).

2.1   Transport schemes for arbitrary meshes

The transport of a tracer density ϕ in a prescribed, non-divergent velocity field $\mathbf{u}$ is given by the flux-form equation (Nair and Lauritzen, 2010)

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{u}\phi) = 0 \; . \tag{2.1}$$

The time derivative is discretised using an explicit, two-stage, second-order Heun scheme,

$$\phi^\star = \phi^{(n)} + \Delta t \, g(\phi^{(n)}) \tag{2.2a}$$

$$\phi^{(n+1)} = \phi^{(n)} + \frac{\Delta t}{2} \left[g(\phi^{(n)}) + g(\phi^\star) \right] \tag{2.2b}$$

where $g(\phi^{(n)}) = -\nabla \cdot (\mathbf{u}\phi^{(n)})$ at time level n. This two-stage second-order time-stepping scheme is similar to the three-stage second-order time-stepping scheme used later in a model of the fully compressible Euler equations (section 4.3), which needs an additional time-stage to converge upon the semi-implicit solution. The two-stage second-order time-stepping scheme is used for both the cubicFit scheme and the multidimensional linear upwind scheme. Although the Heun scheme is unstable for a linear oscillator (Durran, 2013) and for solving the transport equation using centred, linear differencing, it is stable when it is used for transport schemes with sufficient upwinding (Hundsdorfer and Verwer, 2013, p. 149).

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 (2.1) is discretised using Gauss's theorem,

$$\nabla \cdot (\mathbf{u}\phi) \approx \frac{1}{\mathcal{V}_c} \sum_{f \in c} \mathbf{u}_f \cdot \mathbf{S}_f \phi_F \tag{2.3}$$

where subscript f denotes a value stored at a face and subscript F denotes a value approximated at a face from 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.

This discretisation is applicable to arbitrary meshes. A necessary condition for stability is given by the multidimensional Courant number (Weller and Shahrokhi, 2014),

$$\text{Co}_c = \frac{\Delta t}{2\mathcal{V}_c} \sum_{f \in c} |\mathbf{u} \cdot \mathbf{S}_f| \tag{2.4}$$

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 the multidimensional linear upwind scheme differ in their approximations, and these approximation methods are described next.

Cubic fit transport scheme

The cubicFit scheme is based on the method-of-lines scheme by Weller and Shahrokhi (2014). The value of the dependent variable at the face, ϕ_F, is approximated 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 figure 2.1a. 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

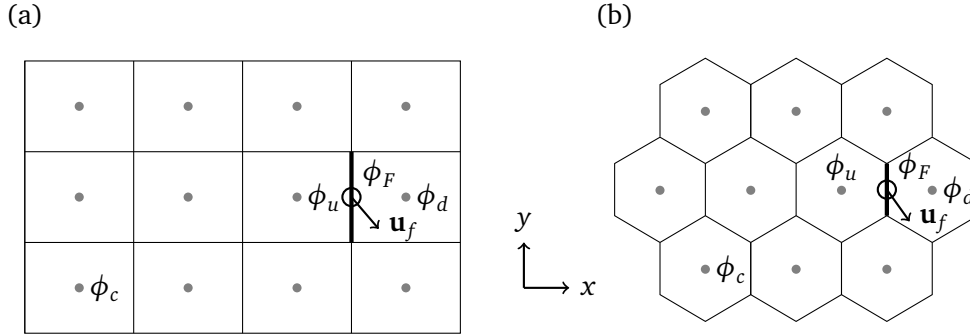(a)   (b)

Figure 2.1: Upwind-biased stencils for faces far away from the boundaries of two-dimensional (a) rectangular and (b) hexagon 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.

and 3 points in y, leading to a natural choice of polynomial that is cubic in x and quadratic in y,

$$\phi = a_1 + a_2 x + a_3 y + a_4 x^2 + a_5 xy + a_6 y^2 + a_7 x^3 + a_8 x^2 y + a_9 xy^2 \,. \qquad (2.5)$$

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 (Leonard et al., 1993).

   The remainder of this section 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.

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

$$\text{Opp}(f,g) \equiv -\frac{\mathbf{S}_f \cdot \mathbf{S}_g}{|\mathbf{S}_f|^2} \tag{2.6}$$

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 (2.6) as

$$\text{Opp}(f,g) = -\frac{|\mathbf{S}_g|}{|\mathbf{S}_f|} \cos(\theta) \tag{2.7}$$

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

$$\text{OF}(f,c_u) \equiv \{g : \text{Opp}(f,g) \geq 0.5\} \cup \{g : \max_{g \in G}(\text{Opp}(f,g))\} \, . \tag{2.8}$$

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.

Figure 2.2 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.

---

[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.
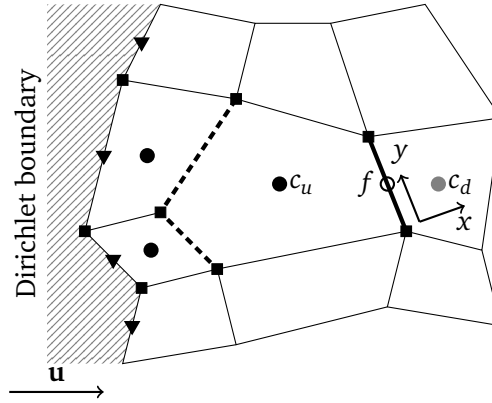
Figure 2.2: A fourteen-point, upwind-biased stencil for face f connecting the pentagonal upwind cell, c_u, and the downwind cell c_d. The dashed lines denote the two faces of cell c_u that oppose f, and black circles mark the centroids of the internal cells that are connected to these two opposing faces. 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.

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 (2.5) 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

$$
\begin{bmatrix}
1 & x_1 & y_1 & x_1^2 & x_1 y_1 & y_1^2 & x_1^3 & x_1^2 y_1 & x_1 y_1^2 \\
1 & x_2 & y_2 & x_2^2 & x_2 y_2 & y_2^2 & x_2^3 & x_2^2 y_2 & x_2 y_2^2 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
1 & x_{12} & y_{12} & x_{12}^2 & x_{12} y_{12} & y_{12}^2 & x_{12}^3 & x_{12}^2 y_{12} & x_{12} y_{12}^2
\end{bmatrix}
\begin{bmatrix}
a_1 \\
a_2 \\
\vdots \\
a_9
\end{bmatrix}
=
\begin{bmatrix}
\phi_1 \\
\phi_2 \\
\vdots \\
\phi_{12}
\end{bmatrix}
\tag{2.9}
$$

which can be written as

$$
\mathbf{Ba} = \boldsymbol{\phi} \; . \tag{2.10}
$$

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 that is a translation, rotation and rescaling of the physical, Cartesian coordinate system: the origin is positioned at the centroid of face f, with x normal to the face and y perpendicular to x, and the distance between the origin and the upwind cell centroid is one. The local 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 2.4. 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 (Skamarock and Gassmann, 2011; Lashley, 2002).

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

$$\mathbf{a} = \mathbf{B}^+ \boldsymbol{\phi} \ . \tag{2.11}$$

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

$$a_1 = \begin{bmatrix} b_{1,1}^+ & b_{1,2}^+ & \cdots & b_{1,12}^+ \end{bmatrix} \begin{bmatrix} \phi_1, \phi_2, \cdots, \phi_{12} \end{bmatrix}^{\mathsf{T}} \tag{2.12}$$

where the weights $b_{1,1}^+ \ldots b_{1,12}^+$ 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 (2.12), 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 Lashley (2002); Skamarock and Menchaca (2010). 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, $\mathbf{m} = m_1 \ldots m_{12}$, and multiply equation (2.10) by $\mathbf{M} = \mathrm{diag}(\mathbf{m})$ to obtain

$$\tilde{\mathbf{B}}\mathbf{a} = \mathbf{M}\boldsymbol{\phi} \tag{2.13}$$

where $\tilde{\mathbf{B}} = \mathbf{M}\mathbf{B}$. The constant coefficient a_1 is then calculated from the pseudo-inverse $\tilde{\mathbf{B}}^+$,

$$a_1 = \tilde{\mathbf{b}}_1^+ \cdot \mathbf{m} \cdot \boldsymbol{\phi} \tag{2.14}$$

where $\tilde{\mathbf{b}}_1^+ = \tilde{b}_{1,1}^+ \ldots \tilde{b}_{1,12}^+$ 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 $\tilde{\mathbf{b}}_1^+ \cdot \mathbf{m}$. 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.
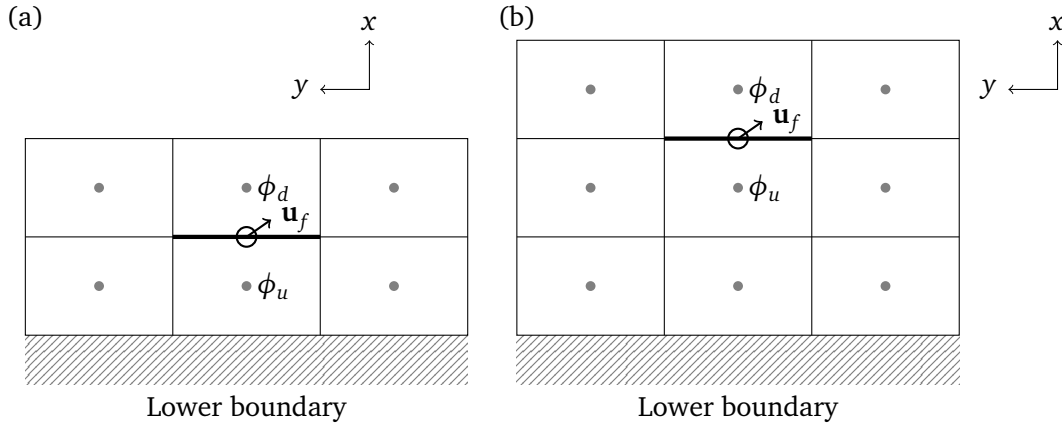
Figure 2.3: Upwind-biased stencils for faces near the lower boundary of a rectangular x–z mesh, with (a) a 3 × 2 stencil for the face immediately adjacent to the lower boundary, and (b) a 3 × 3 stencil for the face immediately adjacent to the face in (a). Each stencil belongs to the face marked by a 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 (2.5) would result in an underconstrained problem. There is no normal flow at the lower boundary.

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 cells 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 cells because this would lead to an underconstrained system of equations. The procedure for choosing suitable polynomials is discussed next.

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 (figure 2.1a), and a hexagonal mesh has 10 points (figure 2.1b). In both cases, constructing a system of equations using the nine-term polynomial in equation (2.5) 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 (figure 2.3a) 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 figure 2.3b 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 undercon-

strained problem. In two dimensions, a candidate polynomial has some combination of between one and nine terms from equation (2.5). 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

$$M(x, y) = x^i y^j : i, j \geq 0 \text{ and } i \leq 3 \text{ and } j \leq 2 \text{ and } i + j \leq 3 \tag{2.15}$$

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 $\phi = a_1 + a_2 x + a_3 y + a_4 x y + a_5 x^2 + a_6 x^2 y$ is a dense 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 $x y$ and x^2 are also included. In total there are 26 dense subsets of the two-dimensional polynomial in equation (2.5).

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 figure 2.3b results in a rank-deficient matrix and so the nine-term polynomial is not a candidate polynomial for that stencil.

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.

Achieving numerical stability

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 (2.14). 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, simplified, one-dimensional von Neumann analyses were performed in order to impose stability conditions on the weights $\mathbf{w}$. 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.

The two-cell analysis starts with the conservation equation for a dependent variable ϕ that is discrete-in-space and continuous-in-time

$$\frac{\partial \phi_j}{\partial t} = -v\frac{\phi_R - \phi_L}{\Delta x} \tag{2.16}$$

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

$$\phi_L = \alpha_u \phi_{j-1} + \alpha_d \phi_j \tag{2.17}$$

$$\phi_R = \beta_u \phi_j + \beta_d \phi_{j+1} \tag{2.18}$$

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

$$\phi_j^{(n)} = A^n e^{ijk\Delta x} \tag{2.19}$$

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 (2.16)

$$\frac{\partial \phi_j}{\partial t} = \frac{\partial}{\partial t}\left(A^{t/\Delta t}\right)e^{ijk\Delta x} = \frac{\ln A}{\Delta t}A^n e^{ikj\Delta x} \tag{2.20}$$

hence equation (2.16) becomes

$$\frac{\ln A}{\Delta t} = -\frac{v}{\Delta x}\left(\beta_u + \beta_d e^{ik\Delta x} - \alpha_u e^{-ik\Delta x} - \alpha_d\right) \tag{2.21}$$

$$\ln A = -c\left(\beta_u - \alpha_d + \beta_d \cos k\Delta x + i\beta_d \sin k\Delta x - \alpha_u \cos k\Delta x + i\alpha_u \sin k\Delta x\right) \tag{2.22}$$

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

$$\ln A = -c\left(\Re + i\Im\right) \tag{2.23}$$

$$A = e^{-c\Re}e^{-ic\Im} \tag{2.24}$$

and the complex modulus of A is

$$|A| = e^{-c\Re} = \exp\left(-c\left(\beta_u - \alpha_d + (\beta_d - \alpha_u)\cos k\Delta x\right)\right) . \tag{2.25}$$

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

$$(\alpha_u - \alpha_d)(1 - \cos k\Delta x) \geq 0 \quad \forall k\Delta x \tag{2.26}$$

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

$$\alpha_u - \alpha_d \geq 0 \ . \tag{2.27}$$

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 $|A| \geq |A_{\text{up}}|$, hence

$$\exp\left(-c\left(\alpha_u - \alpha_d\right)\left(1 - \cos k\Delta x\right)\right) \geq \exp\left(-c\left(1 - \cos k\Delta x\right)\right) \quad \forall k\Delta x \tag{2.28}$$

therefore

$$\alpha_u - \alpha_d \leq 1 \ . \tag{2.29}$$

Now, knowing that $\alpha_u + \alpha_d = 1$ (or $\alpha_d = 1 - \alpha_u$) then, using equations (2.27) and (2.29), we obtain the first two stability conditions,

$$0.5 \leq \alpha_u \leq 1 \text{ and} \tag{2.30}$$
$$0 \leq \alpha_d \leq 0.5 \ . \tag{2.31}$$

The three-cell analysis starts again from equation (2.16) but this time approximate ϕ_L and ϕ_R using three cell centre values,

$$\phi_L = \alpha_{uu}\phi_{j-2} + \alpha_u\phi_{j-1} + \alpha_d\phi_j \tag{2.32}$$
$$\phi_R = \alpha_{uu}\phi_{j-1} + \alpha_u\phi_j + \alpha_d\phi_{j+1} \tag{2.33}$$

having used the same weights α_{uu}, α_u and α_d for both left and right fluxes. Substituting equation (2.19) into equation (2.16) we find

$$A = \exp\left(-c\left[\alpha_{uu}\left(e^{-ik\Delta x} - e^{-2ik\Delta x}\right) + \alpha_u\left(1 - e^{-ik\Delta x}\right) + \alpha_d\left(e^{ik\Delta x} - 1\right)\right]\right) \tag{2.34}$$

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

$$\alpha_u - \alpha_d + \left(\alpha_{uu} - \alpha_u + \alpha_d\right)\cos k\Delta x - \alpha_{uu}\cos 2k\Delta x \geq 0 \ . \tag{2.35}$$

Let $y = \cos k\Delta x$ then

$$\alpha_u - \alpha_d + \left(\alpha_{uu} - \alpha_u + \alpha_d\right)y - \alpha_{uu}(2y^2 - 1) \geq 0. \tag{2.36}$$

For stability, equation (2.36) should hold for $y \in [-1, 1]$. The minumum is either in the interval $-1 < y < 1$, or it is at $y = -1$ or $y = 1$. When $y = 1$ then, trivially, $0 \geq 0$. When $y = -1$ then

$$\alpha_u - \alpha_d \geq \alpha_{uu} \tag{2.37}$$

If the minimum is in the interval $-1 < y < 1$ then the turning points of equation (2.36) can be found by differentiation such that

$$\alpha_{uu}(1 - 4y) - \alpha_u + \alpha_d = 0 \tag{2.38}$$

hence

$$y = \frac{\alpha_{uu} - \alpha_u + \alpha_d}{4\alpha_{uu}} \ . \tag{2.39}$$

Since $-1 < y < 1$ and the quadratic equation (2.36) must have $\alpha_{uu} < 0$ for the turning point to be a minimum, then

$$5\alpha_{uu} < \alpha_u - \alpha_d < -3\alpha_{uu} \ . \tag{2.40}$$

The left-hand side of equation (2.36) is zero when $y = 1$, and so any minimum in the interval $-1 < y < 1$ must be less than zero, making the scheme unstable. Hence, the minimum of equation (2.36) must occur for $y \leq 1$ or $y \geq 1$, so

$$\alpha_u - \alpha_d \geq -3\alpha_{uu}. \tag{2.41}$$

which is more permissive than the alternative, $\alpha_u - \alpha_d \leq 5\alpha_{uu}$. Guided by equations (2.27), (2.37) and (2.41), we choose the final stability condition

$$\alpha_u - \alpha_d \geq \max_{p \in P} |\alpha_p| \tag{2.42}$$

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.

The three stability conditions (equations 2.30, 2.31 and 2.42) are used to impose three stability conditions on the weights **w**,

$$0.5 \leq w_u \leq 1 \tag{2.43a}$$

$$0 \leq w_d \leq 0.5 \tag{2.43b}$$

$$w_u - w_d \geq \max_{p \in P}(|w_p|) \tag{2.43c}$$

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 $w_u + w_d + \sum_{p \in P} w_p = 1$. Inspection of stencil weights for stable and unstable numerical experiments confirms that equation (2.43) provides suitable stability conditions to achieve a stable transport scheme on arbitrary meshes.

The stability of the one-dimensional transport equation discretised in space and time could be analysed using existing techniques (Baldauf, 2008), but we have only analysed the spatial stability of the cubicFit scheme. Numerical experiments presented in section 4.2 demonstrate that the cubicFit scheme is generally insensitive to the time-step, provided that it is below a stability limit.

Stabilisation procedure

Equipped with three stability conditions in equation 2.43, we develop a stabilisation procedure that achieves numerical stability on arbitrary meshes. 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 **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 Lashley (2002), 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.

In the third step, we calculate the weights **w** and evaluate them against the stability conditions given in equation (2.43). 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 satisified 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

---

[2]Note that singular values are used for two purposes: first, to test if the matrix **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, cond(**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.
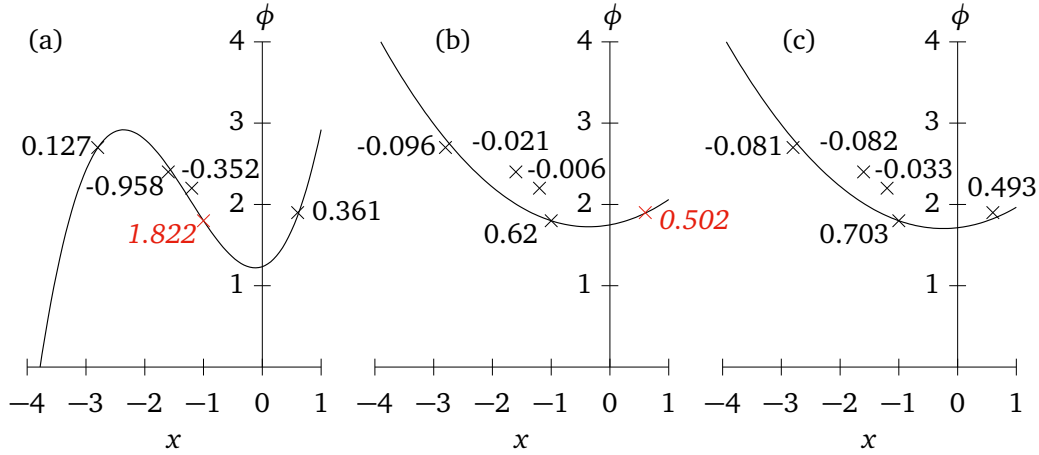
Figure 2.4: One-dimensional least-squares fits with a stencil of five points using (a) a cubic polynomial with multipliers $m_u = 1024$, $m_d = 1024$ and $m_p = 1$, (b) a quadratic polynomial with the same multipliers, and (c) a quadratic polynomial with multipliers $m_u = 1024$, $m_d = 1$ and $m_p = 1$. 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 (2.43) are marked in red with italicised labels. 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) calculates weights using only x positions, and values of ϕ are included here for illustration only.

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 (2.5) with $m_u = m_d = 2^{10}$.

To illustrate the stabilisation procedure, figure 2.4a 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 uses only 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

$$\phi = a_1 + a_2 x + a_3 x^2 + a_4 x^3 \;, \tag{2.44}$$

$$\phi = a_1 + a_2 x + a_3 x^2 \;, \tag{2.45}$$

$$\phi = a_1 + a_2 x \;, \tag{2.46}$$

$$\phi = a_1 \;. \tag{2.47}$$

We begin with the cubic equation (2.44). 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 2.43a). Reducing the downwind multiplier does not help to satisfy the stability condition, so we start again with the quadratic equation (2.45), and the new fit is presented in figure 2.4b. 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 2.43b). This time, however, stable weights are found by reducing m_d to one (figure 2.4c) and these are the weights that will be used to approximate ϕ_F, where the polynomial intercepts the y-axis.

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 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 (2.5) is replaced with its three-dimensional counterpart,

$$
\begin{aligned}
\phi = a_1 + a_2 x + a_3 y + a_4 z + a_5 x^2 + a_6 xy + a_7 y^2 + a_8 xz + a_9 yz + a_{10}z^2 + \\
a_{11}x^3 + a_{12}x^2 y + a_{13}xy^2 + a_{14}x^2 z + a_{15}xz^2 + a_{16}yz^2 + a_{17}y^2 z + a_{18}xyz \,. \quad (2.48)
\end{aligned}
$$

The procedure for generating candidate polynomials described in section 2.1 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 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 (2.48) 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 the cubicFit scheme any further.

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 that follow. The approximation of ϕ_F is calculated using a gradient reconstruction,

$$\phi_F = \phi_u + \nabla_c \, \phi \cdot \left(\mathbf{x}_f - \mathbf{x}_c \right) \tag{2.49}$$

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's theorem:

$$\nabla_c \, \phi = \frac{1}{\mathcal{V}_c} \sum_{f \in c} \widetilde{\phi}_F \mathbf{S}_f \tag{2.50}$$

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 (2.50) is evaluated. This implementation of the multidimensional linear upwind scheme is included with OpenFOAM (CFD Direct, 2018b).

2.2   Horizontal transport over mountains

A two-dimensional transport test was developed by Schär et al. (2002) 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. When terrain-following meshes are used, this test challenges transport schemes because the tracer must cross mesh layers, which acts to reduce numerical accuracy (Schär et al., 2002). Here we use a more challenging variant of the test that has steeper mountains and highly-distorted terrain-following meshes. Numerical convergence and numerical error structures are compared using the linearUpwind and cubicFit transport schemes on terrain-following meshes and cut cell meshes.

The domain is defined on a rectangular x–z plane that is 300 km wide as measured between the outermost cell centres, and 25 km high as measured between upper and lower boundary edges. Boundary conditions are imposed on the tracer density ϕ such that $\phi = 0 \, \text{kg m}^{-3}$ at the inlet boundary, and a zero normal gradient $\partial \phi / \partial n = 0 \, \text{kg m}^{-4}$ is imposed at the outlet boundary. There is no normal flow at the lower and upper boundaries.

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

$$h(x) = h^\star \cos^2(\alpha x) \tag{2.51a}$$

where

$$h^\star(x) = \begin{cases} h_0 \cos^2(\beta x) & \text{if } |x| < a, \\ 0 & \text{otherwise,} \end{cases} \tag{2.51b}$$

where $a = 25\,\text{km}$ is the mountain envelope half-width, $h_0 = 6\,\text{km}$ is the maximum mountain height, $\lambda = 8\,\text{km}$ 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 Schär et al. (2002).

A basic terrain-following (BTF) mesh is constructed by using the terrain profile to modify the uniform rectangular mesh. The BTF method uses a linear decay function so that mesh layers become horizontal at the top of the model domain (Gal-Chen and Somerville, 1975),

$$z(x) = (H - h(x))(z^\star/H) + h(x) \tag{2.52}$$

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 layer. 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,

$$u(z) = u_0 \begin{cases} 1 & \text{if } z \geq z_2, \\ \sin^2\left(\frac{\pi}{2} \frac{z - z_1}{z_2 - z_1}\right) & \text{if } z_1 < z < z_2, \\ 0 & \text{otherwise,} \end{cases} \tag{2.53}$$

where $u_0 = 10\,\text{m}\,\text{s}^{-1}$, $z_1 = 7\,\text{km}$ and $z_2 = 8\,\text{km}$. This results in uniform horizontal flow above $8\,\text{km}$ and zero flow at $7\,\text{km}$ and below.

The discrete velocity field is defined using a streamfunction, Ψ. Given that $u = -\partial\Psi/\partial z$, the streamfunction is found by vertical integration of the velocity profile,

$$\Psi(z) = -\frac{u_0}{2} \begin{cases} (2z - z_1 - z_2) & \text{if } z > z_2, \\ z - z_1 - \frac{z_2 - z_1}{\pi} \sin\left(\pi \frac{z - z_1}{z_2 - z_1}\right) & \text{if } z_1 < z \leq z_2, \\ 0 & \text{otherwise.} \end{cases} \tag{2.54}$$

A tracer with density ϕ is positioned upstream above the height of the terrain. It has the shape

$$\phi(x, z) = \phi_0 \begin{cases} \cos^n\left(\frac{\pi r}{2}\right) & \text{if } r \leq 1, \\ 0 & \text{otherwise,} \end{cases} \tag{2.55a}$$

with radius r given by

$$r = \sqrt{\left(\frac{x - x_0}{A_x}\right)^2 + \left(\frac{z - z_0}{A_z}\right)^2} \tag{2.55b}$$

where $A_x = 25\,\text{km}$, $A_z = 3\,\text{km}$ are the horizontal and vertical half-widths respectively, $\phi_0 = 1\,\text{kg m}^{-3}$ is the maximum density of the tracer, and the exponent $\mathfrak{n} = 2$. At $t = 0\,\text{s}$, 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 $10\,000\,\text{s}$ using a time-step 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. (2012). By the end of integration the tracer is positioned downwind of the mountain. The analytic solution at $t = 10\,000\,\text{s}$ is centred at $(x_0, z_0) = (50\,\text{km}, 12\,\text{km})$ with its shape unchanged from the initial condition.

To measure numerical convergence, a range of mesh spacings are chosen so that $\Delta x : \Delta z = 2{:}1$ to match the original test specification by Schär et al. (2002). Tests were performed using the linearUpwind and cubicFit schemes on BTF meshes and cut cell meshes with mesh spacings between $\Delta x = 250\,\text{m}$ and $\Delta x = 5000\,\text{m}$. Error norms are calculated by subtracting the analytic solution from the numerical solution,

$$\ell_2 = \sqrt{\frac{\sum_c (\phi - \phi_T)^2 \, \mathscr{V}_c}{\sum_c \left(\phi_T^2 \mathscr{V}_c\right)}} \tag{2.56}$$

$$\ell_\infty = \frac{\max_c |\phi - \phi_T|}{\max_c |\phi_T|} \tag{2.57}$$

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. The linearUpwind and cubicFit schemes are second-order convergent in the ℓ_2 norm (figure 2.5a) and ℓ_∞ norm (figure 2.5b) at all but the coarsest mesh spacings where errors are saturated for both schemes.

We can estimate the computational cost of the cubicFit scheme relative to the linearUpwind scheme by considering the mesh spacing necessary for a given ℓ_2 error, and the size of the stencil used by each scheme. 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 of the cubicFit scheme per time-stage is about half the computational cost of the linearUpwind scheme.

Next, we examine the structure of numerical errors using the linearUpwind and cubicFit transport schemes on BTF and cut cell meshes with $\Delta x = 1000\,\text{m}$ and $\Delta z = 500\,\text{m}$. To obtain a
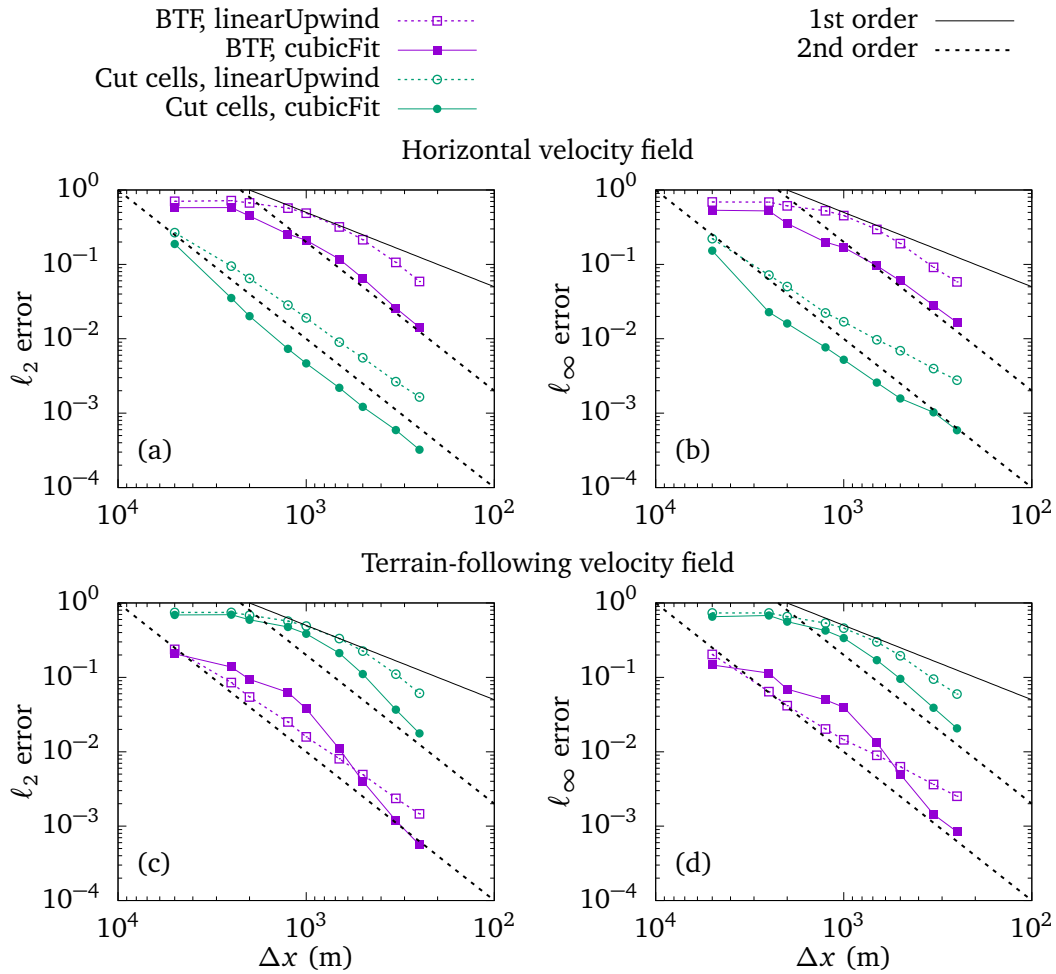
Figure 2.5: Numerical convergence of the two-dimensional tracer transport tests over mountains using (a, b) horizontal and (c, d) terrain-following velocity fields. ℓ_2 errors (equation 2.56) and ℓ_∞ errors (equation 2.57) are marked at mesh spacings between $\Delta x = 5000\,$m and $\Delta x = 250\,$m using linearUpwind and cubicFit transport schemes on basic terrain-following and cut cell meshes.
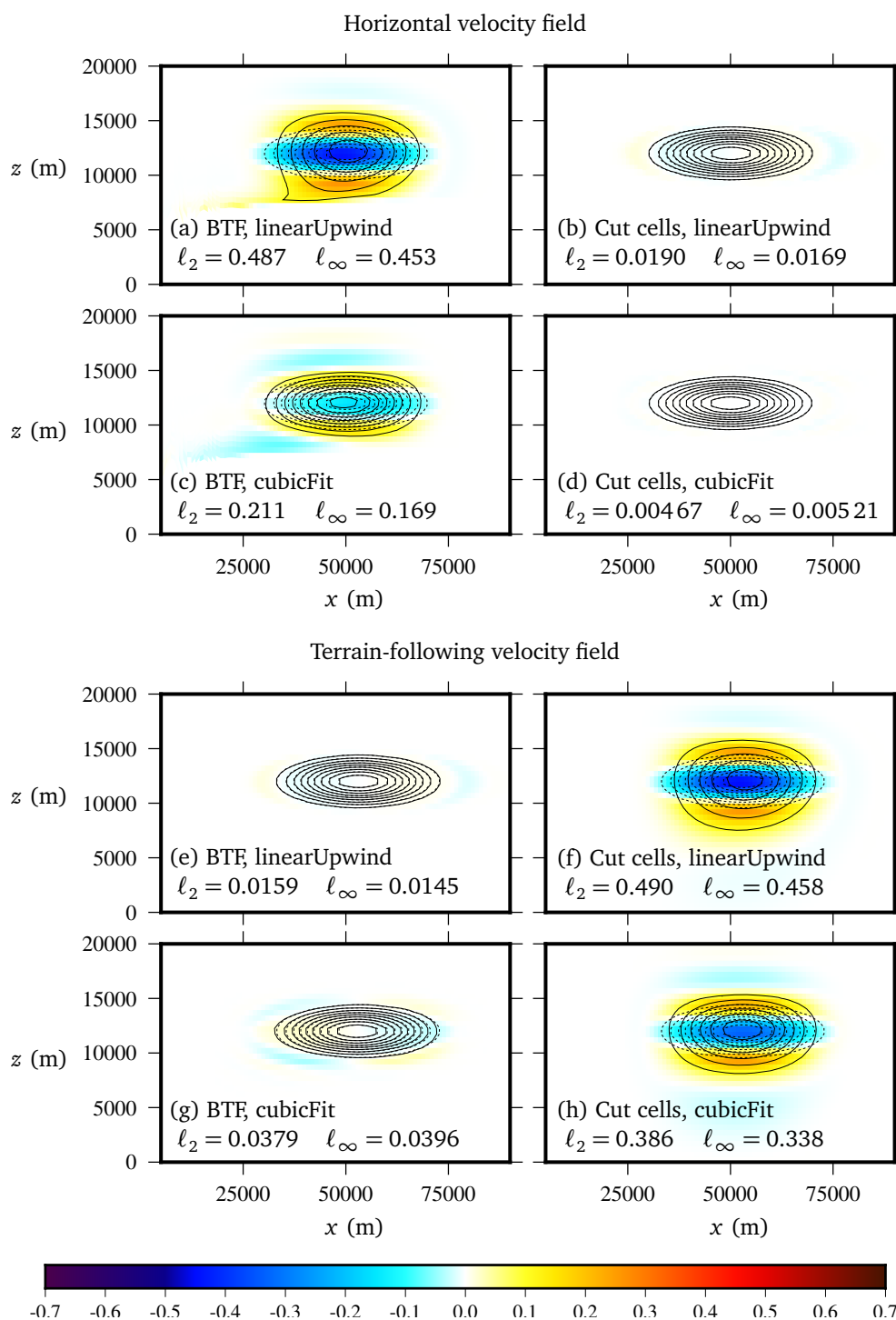
Figure 2.6: Tracer contours at the end of integration for the two-dimensional tracer transport tests over mountains using (a, b, c, d) horizontal and (e, f, g, h) terrain-following velocity fields. The numerical solution, marked with solid lines, and the analytic solution, marked with dashed lines, are plotted every 0.1. Tracer contours overlay a colour error field, calculated by subtracting the analytic solution from the numerical solution. Only the lowest 20 km in the lee of the mountain is plotted. The entire domain is 300 km wide and 25 km high.

maximum Courant number of about 0.4, we choose $\Delta t = 40$ s on the cut cell mesh where the flow is aligned with mesh layers and there are no fluxes through upper and lower cell faces. Since there is no flow below $z = 7$ km, the time-step is not constrained by small, cut cells next to the lower boundary. On the BTF mesh, Δt is only 8 s because the flow is misaligned with mesh layers, with fluxes through all four faces of cells above sloping terrain.

The highly-distorted BTF mesh presents a particular challenge to the linearUpwind scheme with the final numerical solution, marked by solid lines, losing its correct shape and maximum intensity compared to the analytic solution marked by dashed lines (figure 2.6a). The linearUpwind scheme produces a much more accurate solution on the cut cell mesh, with only small phase errors apparent in figure 2.6b. Accuracy is much improved using the cubicFit scheme: on the BTF mesh, shape and maximum intensity are similar to the analytic solution (figure 2.6c) and, on the cut cell mesh, numerical errors are so small they are not visible (figure 2.6d). The numerical and analytic contours overlay a colour error field that reveals horizontal streaks of error on the BTF mesh (figure 2.6a, 2.6c) that were generated above the steepest mountain peaks before becoming trapped in the region of zero flow below $z = 7$ km.

The horizontal transport test demonstrates that the cubicFit scheme is second-order convergent in the domain interior irrespective of mesh distortions. Numerical errors are largest on terrain-following meshes, due either to misalignment of the flow with mesh layers, or to mesh distortions. In the next section, we propose a new test in order to identify the primary cause of these numerical errors.

2.3 Transport in a terrain-following velocity field

In the horizontal transport test, results were least accurate on the BTF mesh where the mesh was most distorted and flow was misaligned with mesh layers. Here, we formulate a new tracer transport test in which the velocity field is everywhere tangential to the basic terrain-following mesh layers. The flow is then aligned with the BTF mesh layers, but the cells in the linearUpwind and cubicFit stencils are not uniformly distributed because the BTF mesh is distorted. Conversely, the flow is misaligned with the cut cell mesh layers but, except for cut cells next to the ground, the cut cell mesh is undistorted. This test determines whether the primary source of numerical error is due to mesh distortions or misalignment of the flow with mesh layers.

The domain size, mountain profile, initial tracer profile and boundary conditions are the same as those in the horizontal transport test in section 2.2. The discrete velocity field is calculated using a streamfunction Ψ in the same way as the horizontal transport test. Here, we define a different streamfunction that yields a velocity field that follows the BTF mesh layers given by

equation (2.52) such that

$$\Psi(x,z) = -u_0 H_1 \frac{z-h}{H_1-h} \tag{2.58}$$

where $u_0 = 10\,\mathrm{m\,s^{-1}}$, which is the horizontal velocity where $h(x) = 0$. The velocity field follows the lower boundary and becomes entirely horizontal at $H_1 = H = 25\,\mathrm{km}$, hence, there is no normal flow at the lower and upper boundaries. In the domain interior, 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

$$u = -\frac{\partial \Psi}{\partial z} = u_0 \frac{H_1}{H_1-h}, \quad w = \frac{\partial \Psi}{\partial x} = u_0 H_1 \frac{\mathrm{d}h}{\mathrm{d}x} \frac{H_1-z}{(H_1-h)^2} , \tag{2.59}$$

$$\frac{\mathrm{d}h}{\mathrm{d}x} = -h_0 \left[\beta \cos^2(\alpha x) \sin(2\beta x) + \alpha \cos^2(\beta x) \sin(2\alpha x) \right] . \tag{2.60}$$

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

An analytic solution at $10\,000\,\mathrm{s}$ 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 at $-a$, to the right side of the mountain at a,

$$\mathrm{d}t = \mathrm{d}x/u(x) \tag{2.61}$$

$$t = \int_{-a}^{a} \frac{H_1 - h(x)}{u_0 H_1} \,\mathrm{d}x \tag{2.62}$$

$$t = \frac{2a}{u_0} - \frac{h_0}{16u_0 H_1} \left[4x + \frac{\sin 2(\alpha+\beta)x}{\alpha+\beta} + \frac{\sin 2(\alpha-\beta)x}{\alpha-\beta} + 2\left(\frac{\sin 2\alpha x}{\alpha} + \frac{\sin 2\beta x}{\beta} \right) \right]_{-a}^{a} \tag{2.63}$$

Because the velocity field is non-divergent, the flow accelerates over mountain ridges and the tracer travels 2997.162 m further compared to the tracer in a purely horizontal velocity field. The vertical tracer position is unchanged downwind of the mountains because flow is parallel to the mesh layers.

To enable comparisons with the horizontal transport test, results are obtained using the linearUpwind and cubicFit transport schemes on BTF and cut cell meshes with $\Delta x = 1000\,\mathrm{m}$ and $\Delta z = 500\,\mathrm{m}$. To obtain a maximum Courant number of about 0.4, we choose $\Delta t = 25\,\mathrm{s}$ on the BTF mesh where flow is aligned with mesh layers. The cut cell mesh suffers from the small cell problem, having a more stringent time-step constraint of $\Delta t = 8\,\mathrm{s}$. Recall that, in this test, there is flow everywhere in the domain, and it is flow through arbitrarily small cut cells that imposes the more stringent time-step constraint.

Figure 2.7 shows results using the cubicFit scheme on the BTF mesh, illustrating the evolution of the tracer with snapshots plotted every 5000 s. At $t = 5000\,\mathrm{s}$, the tracer is distorted by
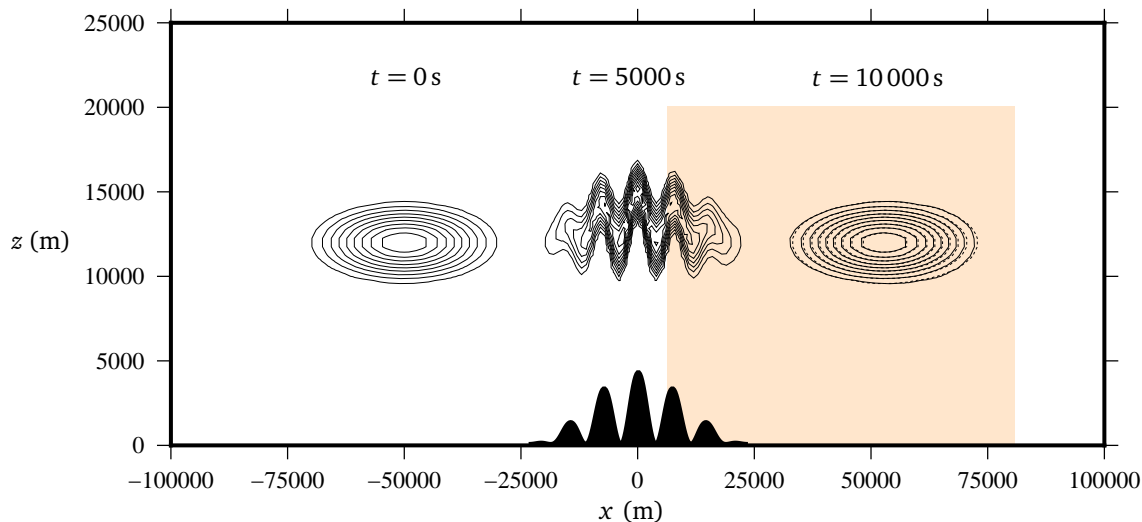
Figure 2.7: Tracer contours transported above mountains in a terrain-following velocity field at $t = 0$ s, 5000 s, and 10 000 s using the cubicFit transport scheme on a BTF mesh. The analytic solution at $t = 10\,000$ s is plotted with dashed contours. All contour intervals are 0.1. The terrain profile is shown immediately above the x axis. The region highlighted in orange marks the region plotted in the panels of figure 2.6. Only the central 200 km of the domain is shown. The entire domain is 300 km wide and 25 km high.

the terrain-following velocity field but, by $t = 10\,000$ s, the tracer has correctly returned to its original shape, with some phase errors apparent when comparing the numerical solution (solid contours) with the analytic solution (dashed contours). The region highlighted in orange corresponds to the region plotted in figure 2.6, where tracer contours and numerical errors are plotted at $t = 10\,000$ s.

Unlike the horizontal transport test, results are most accurate on the BTF mesh (linearUpwind, figure 2.6e; cubicFit, figure 2.6g) and least accurate on the cut cell mesh (linearUpwind, figure 2.6f; cubicFit, figure 2.6h). Hence, we conclude that the accuracy of the transport schemes depends upon alignment of the flow with mesh layers, and accuracy is mostly unaffected by mesh distortions. The error structures on the cut cell mesh in this test (2.6f, 2.6h) are similar to the error structures on the BTF mesh in the horizontal transport test (2.6a, 2.6c), and the phase error using the linearUpwind scheme on the BTF mesh (2.6e) closely resembles the error on the cut cell mesh in the horizontal transport test (2.6b).

Perhaps surprisingly, errors are slightly larger using the cubicFit scheme on the BTF mesh (2.6g) compared to those obtained using the linearUpwind scheme (2.6e). At finer mesh spacings, however, cubicFit is more accurate on BTF and cut cell meshes in both the ℓ_2 norm (figure 2.5c) and ℓ_∞ norm (figure 2.5d). Once again, both transport schemes are second-order convergent

irrespective of mesh distortions or misalignment of the flow with mesh layers.

In both horizontal and terrain-following transport tests, which are both variations on the standard test case by Schär et al. (2002), the linearUpwind and cubicFit transport schemes are second-order convergent irrespective of mesh distortions or misalignment of the flow with mesh layers. Together, the horizontal and terrain-following transport tests demonstrate that numerical accuracy depends primarily on the alignment of the flow with mesh layers.

2.4 Deformational flow on a sphere

The tests presented 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. (2012). Results are compared between the 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 Thuburn et al. (2014); Heikes and Randall (1995a,b) 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 (Staniforth and Thuburn, 2012).

Following appendix A9 in Lauritzen et al. (2014), the average equatorial spacing $\Delta\lambda$ is used as a measure of mesh spacing. It is defined as

$$\Delta\lambda = 360° \frac{\overline{\Delta x}}{2\pi R_e} \tag{2.64}$$

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

The deformational flow test specified by Lauritzen et al. (2012) 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 smooth 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,

$$\Psi(\lambda, \theta, t) = \frac{10R_e}{T} \sin^2\left(\lambda'\right) \cos^2(\theta) \cos\left(\frac{\pi t}{T}\right) - \frac{2\pi R_e}{T} \sin(\theta) \tag{2.65}$$

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,

$$\phi = \phi_1(\lambda, \theta) + \phi_2(\lambda, \theta) . \tag{2.66}$$

An individual hill ϕ_i is given by

$$\phi_i(\lambda, \theta) = \phi_{\max} \exp\left(-b\left(\frac{|\mathbf{x} - \mathbf{x}_i|}{R_e}\right)^2\right) \tag{2.67}$$

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

$$(x, y, z) = (R_e \cos\theta \cos\lambda, R_e \cos\theta \sin\lambda, R_e \sin\theta) . \tag{2.68}$$

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

$$(\lambda_1, \theta_1) = (5\pi/6, 0) , \tag{2.69}$$

$$(\lambda_2, \theta_2) = (7\pi/6, 0) . \tag{2.70}$$

The results in figure 2.8 are obtained using the cubicFit scheme on a hexagonal-icosahedral mesh with $\Delta\lambda = 0.542°$. The initial Gaussian hills are shown in figure 2.8a. At $t = T/2$ the tracer has been deformed into an S-shaped filament (figure 2.8b). 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 (figure 2.8c).
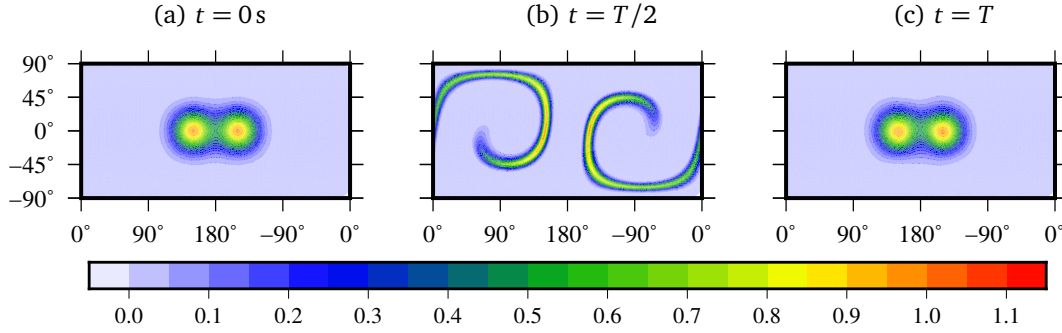
Figure 2.8: Tracer fields for the deformational flow test using initial Gaussian hills. The tracer is deformed by the velocity field before the rotation reverses to return the tracer to its original distribution: (a) the initial tracer distribution at $t = 0\,\text{s}$; (b) by $t = T/2$ the Gaussian 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 $\Delta\lambda = 0.542°$.
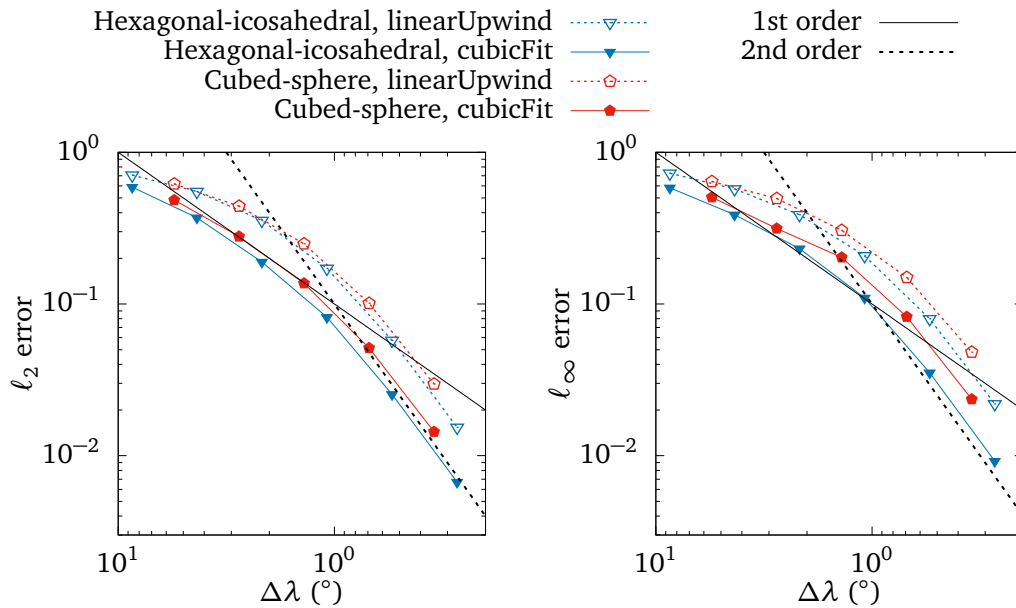

Figure 2.9: Numerical convergence of the deformational flow test on the sphere using initial Gaussian hills. ℓ_2 errors (equation 2.56) and ℓ_∞ errors (equation 2.57) are marked at mesh spacings between 8.61° and 0.271° using the linearUpwind scheme (dotted lines) and the cubicFit scheme (solid lines) on hexagonal-icosahedral meshes and cubed-sphere meshes.

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 betweeen $\Delta\lambda = 8.61°$ and $\Delta\lambda = 0.271°$ on hexagonal-icosahedral meshes and cubed-sphere meshes. The results are shown in figure 2.9. The solution is slow to converge on coarser meshes, and this behaviour agrees with the results from Lauritzen et al. (2012). Both linearUpwind and cubicFit schemes achieve second-order convergence 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,

$$
\phi = \begin{cases} b + c\phi_1(\lambda,\theta) & \text{if } r_1 < r, \\ b + c\phi_2(\lambda,\theta) & \text{if } r_2 < r, \\ b & \text{otherwise,} \end{cases}
\tag{2.71}
$$

where $b = 0.1$, $c = 0.9$, $r = R_e/2$ and an individual hill ϕ_i is given by

$$
\phi_i = \begin{cases} \frac{\phi_{\max}}{2}\left(1 + \cos\left(\frac{\pi r_i}{r}\right)\right) & \text{if } r_i < r, \\ 0 & \text{otherwise,} \end{cases}
\tag{2.72}
$$

where $\phi_{\max} = 1$ and r_i is the great circle distance between the centre point (λ_i, θ_i) and point (λ, θ) such that

$$
r_i(\lambda, \theta) = R_e \arccos\left(\sin\theta_i \sin\theta + \cos\theta_i \cos\theta \cos(\lambda - \lambda_i)\right) .
\tag{2.73}
$$

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. (2012) as the coarsest mesh spacing for which $\ell_2 \approx 0.033$. Hence, a transport scheme that achieves a larger minimal resolution is more accurate than one that achieves a smaller minimal resolution.

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 2.1 along with a selection of transport schemes having similar minimal resolutions from the model intercomparison by Lauritzen et al. (2014).

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

Transport scheme	Mesh type	Minimal resolution (°)
linearUpwind	Cubed-sphere	*0.15*
FARSIGHT, grid-point semi-Lagrangian (White and Dongarra, 2011)	Cubed-sphere	0.1875
linearUpwind	Hexagonal-icosahedral	*0.2*
SLFV-SL, swept-area scheme (Miura, 2007)	Hexagonal-icosahedral	0.25
cubicFit	Cubed-sphere	*0.25*
cubicFit	Hexagonal-icosahedral	0.3
ICON-FFSL, swept-area scheme (Miura, 2007)	Triangular-icosahedral	0.42

Table 2.1: 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. (2014).

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.

3 High-order transport for arbitrary meshes

Highlights

- The new highOrderFit transport scheme is third-order convergent or higher on distorted and undistorted meshes

- During integration, the highOrderFit scheme has the same computational cost as the cubicFit scheme, only requiring m multiplies per face per time-stage using a stencil of m cells

- High-order 'k-exact' polynomial reconstructions are obtained by calculating high-order volume and surface moments exactly

Atmospheric models are using increasingly fine meshes to make more accurate forecasts, but high-order numerical schemes offer another possible route to improving accuracy. Choosing a higher-order scheme can be more computationally efficient than choosing a finer mesh (Waruszewski et al., 2018), and numerical experiments performed by Ullrich (2014) to compare the effective resolution of transport schemes identify third- or fourth-order schemes as the 'sweet spot' where computational efficiency is maximised.

A high-order transport scheme is ordinarily defined as one with a formal accuracy greater than second-order. The *order of convergence* observed in numerical experiments may be less than the formal *order of accuracy* if the transported field is insufficiently smooth, and strong gradients in the form of weather fronts and temperature inversions mean that atmospheric fields are generally not smooth enough to obtain high-order convergence (Holdaway et al., 2008). Even if high-order convergence is unattainable, high-order schemes offer other advantages over second-order schemes: high-order schemes can reduce dispersion and diffusion errors (Ullrich

37

and Jablonowski, 2012; Waruszewski et al., 2018), reduce grid imprinting (McCorquodale et al., 2015), and increase the effective resolution of the scheme (Ullrich, 2014).

High-order schemes are often formulated by introducing additional degrees of freedom within each cell. Such schemes are called 'compact schemes' because sub-grid reconstructions are performed within each cell, only requiring data exchange with immediately adjacent cells. Hence, compact schemes have near-optimal parallel scalability, making them attractive for massively parallel atmospheric simulations (Ullrich, 2014). Discontinuous Galerkin (DG) schemes belong to the class of compact schemes, and DG schemes have been tried in some atmospheric research models (Nair et al., 2005; Giraldo and Restelli, 2008). High-order DG schemes prognose values at Gauss points within each cell in order to approximate integral values using Gaussian quadrature. The position of Gauss points can be straightforwardly calculated for tetrahedral and hexahedral reference cells, but no straightforward method is available for arbitrary polyhedra (Costa et al., 2017). Furthermore, numerical quadrature calculations in DG schemes can be expensive (Dumbser et al., 2007), motivating alternative, quadrature-free DG schemes (Atkins and Shu, 1998; Nair, 2015). Another quadrature-free compact scheme is the multi-moment constrained finite volume formulation which achieves high-order accuracy by storing several prognostic moments collocated at cell centres (Ii and Xiao, 2009). Transporting a tracer using a compact scheme usually requires the storage of multiple values per cell, and these storage requirements increase with the order of accuracy, with a fourth-order accurate DG scheme requiring the storage of up to 10 values per cell (Ullrich et al., 2010). The transport scheme by Skamarock and Gassmann (2011) is a compact scheme for hexagonal meshes that, unusually, only requires the cell average values of immediately adjacent cells, using them to calculate a second-order derivative that cancels low-order errors in the Taylor series expansion. The resulting scheme is high-order accurate on uniform hexagonal meshes, but it is formally only first-order accurate on non-uniform meshes.

Non-compact schemes store only cell average values, and high-order reconstructions are obtained on uniform or non-uniform meshes by using a larger stencil of cells. High-order polynomial reconstructions over non-compact stencils have been used in fully compressible finite volume models that employ Godunov-type schemes (Ullrich and Jablonowski, 2012) using cubed-sphere meshes, or Weighted Essentially Non-Oscillatory (WENO) schemes (Tsoutsanis and Drikakis, 2016) using arbitrary polyhedral meshes. Godunov-type schemes and WENO schemes are well-suited for representing nonlinear dynamics with discontinuous solutions (LeVeque, 2002), but they are often computationally expensive. Computationally cheaper are high-order swept area schemes, which also reconstruct high-order polynomials over non-compact stencils. A swept area scheme calculates a face flux by Gaussian integration of the polynomial over the upstream swept area, which typically requires a matrix-vector multiply per face per time-stage

(Thuburn et al., 2014). Computationally cheaper than the swept area approach is the k-exact method (Barth, 1995), which requires only one dot product of two vectors per face per time-stage. The k-exact method is so-called because it exactly reconstructs a polynomial of degree k or less, represented by a non-compact stencil of cell average values.

For the numerical experiments presented in chapter 2, the cubicFit transport scheme achieves only second-order convergence even though it includes high-order polynomial terms. The cubicFit scheme uses a sub-grid reconstruction that fits a polynomial over known values stored at cell centre points, and it is this point-wise approach that limits the scheme to second-order convergence. In this chapter, we apply the k-exact method, constraining the polynomial fit so that the average of the polynomial integrated over a cell volume equals the cell average value. The computationally expensive spatial integration calculations rely on the mesh geometry alone, with just m multiplies per face per time-stage using a stencil of m cells. In this way, we obtain a high-order transport scheme which retains the low computational cost of the cubicFit transport scheme. Since it has much in common with the cubicFit scheme, we name this high-order transport scheme 'highOrderFit'.

In section 3.1, we formulate the highOrderFit transport scheme using the k-exact method. We go on to perform numerical experiments to compare the order of convergence of the highOrderFit scheme and the cubicFit scheme: section 3.2 performs the standard test of horizontal flow over mountains using terrain-following and cut cell meshes and, following Chen et al. (2017), section 3.3 performs a test of deformational flow on a two-dimensional Cartesian plane represented by uniform meshes and meshes with distortions similar to those of a cubed-sphere.

3.1 High-order finite volume formulation

Averaging the flux-form transport equation (2.1) over a volume $\mathcal{V}$ and using Gauss's divergence theorem,

$$\frac{1}{\mathcal{V}} \int_{\mathcal{V}} \frac{\partial \phi}{\partial t} d\mathcal{V} = -\frac{1}{\mathcal{V}} \int_{\partial \mathcal{V}} \phi \mathbf{u} \cdot \hat{\mathbf{n}} dA \tag{3.1}$$

where $\hat{\mathbf{n}}$ is the outward unit normal vector. Using the method of lines, the time derivative is discretised using the classical fourth-order Runge–Kutta time-stepping scheme (Durran, 2013, p. 53), and the spatial discretisation is described next. For a polygonal cell with faces f equation (3.1) becomes

$$\frac{1}{\mathcal{V}} \int_{\mathcal{V}} \frac{\partial \phi}{\partial t} d\mathcal{V} = -\frac{1}{\mathcal{V}} \sum_{f} \int_{\mathcal{A}_f} \phi \mathbf{u} \cdot \hat{\mathbf{n}} d\mathcal{A}_f \tag{3.2}$$

where $\mathscr{A}_f$ is the area of face f. If ϕ is a sufficiently smooth field then it can be approximated to P-order accuracy by replacing ϕ with a polynomial interpolant ψ,

$$\psi = \sum_{|\mathbf{p}| \leq P} a_{\mathbf{p}} (\mathbf{x} - \mathbf{x}_0)^{\mathbf{p}} \tag{3.3}$$

where $a_{\mathbf{p}}$ are unknown polynomial coefficients, $\mathbf{x}_0$ is a fixed position, and P is the total polynomial order. Note that we use multi-index notation such that $|\mathbf{p}| = p_1 + \ldots + p_n$ and

$$a_{\mathbf{p}} (\mathbf{x} - \mathbf{x}_0)^{\mathbf{p}} = a_{\mathbf{p}} \prod_{d=1}^{D} \left(x_d - x_{0_d} \right)^{p_d} \tag{3.4}$$

where D is the number of physical dimensions. As an example, the exponents $\mathbf{p}$ in two dimensions (x, y) with $|\mathbf{p}| \leq 1$ are $(0, 0)$, $(1, 0)$ and $(0, 1)$, hence the two-dimensional polynomial interpolant for a total polynomial order $P = 1$ is

$$\psi = a_{0,0} + a_{1,0} (x - x_0) + a_{0,1} (y - y_0) . \tag{3.5}$$

Replacing ϕ in (3.2) with ψ in (3.3) we obtain an expression for the face flux[1],

$$\int_{\mathscr{A}} \phi \mathbf{u} \cdot \hat{\mathbf{n}} d\mathscr{A} = \mathbf{u}_f \cdot \hat{\mathbf{n}} \sum_{|\mathbf{p}| \leq P} a_{\mathbf{p}} \mathfrak{m}_{\mathscr{A}}^{\mathbf{p}} \tag{3.6}$$

where $\mathfrak{m}_{\mathscr{A}}^{\mathbf{p}} = \int_{\mathscr{A}} (\mathbf{x} - \mathbf{x}_0)^{\mathbf{p}} d\mathscr{A}$ is the $\mathbf{p}$-th moment of area $\mathscr{A}$. Therefore, the face flux can be calculated by finding the the polynomial coefficients $a_{\mathbf{p}}$.

Following the same approach as the cubicFit transport scheme, taking a total polynomial order $P = 3$ gives 9 polynomial terms with polynomial coefficients calculated using the same upwind-biased stencil. For every cell in the stencil we require that the average of the polynomial integrated over a cell volume equals the cell average value,

$$\langle \psi \rangle_{\mathscr{V}} = \langle \phi \rangle_{\mathscr{V}} \tag{3.7}$$

where the average over volume $\mathscr{V}$ is

$$\langle \psi \rangle_{\mathscr{V}} = \frac{1}{\mathscr{V}} \int_{\mathscr{V}} \psi d\mathscr{V} . \tag{3.8}$$

Using equations (3.3) and (3.8) we can rewrite equation (3.7) as

$$\frac{1}{\mathfrak{m}_{\mathscr{V}}^{0}} \sum_{|\mathbf{p}| \leq P} a_{\mathbf{p}} \mathfrak{m}_{\mathscr{V}}^{\mathbf{p}} = \langle \phi \rangle_{\mathscr{V}} \tag{3.9}$$

---

[1] Equation (3.6) assumes that there is little variation in the velocity field $\mathbf{u}$ at the grid scale, such that $\mathbf{u}$ can be approximated by a velocity $\mathbf{u}_f$ that is constant over the face f. Such an assumption is justifiable for atmospheric flows that are predominantly large-scale (Methven and Hoskins, 1999).

where $\mathfrak{m}_{\mathcal{V}}^{\mathbf{p}} = \int_{\mathcal{V}} (\mathbf{x} - \mathbf{x}_0)^{\mathbf{p}} \, d\mathcal{V}$ is the $\mathbf{p}$-th moment of volume $\mathcal{V}$, and the zeroth moment $\mathfrak{m}_{\mathcal{V}}^{\mathbf{0}}$ is the volume. For m polynomial terms and a stencil with n cells, we calculate a face flux by choosing $\mathbf{x}_0$ to be the position of the face centre, then we write the linear system

$$\begin{bmatrix} \mathfrak{m}_{\mathcal{V}_1}^{\mathbf{p}_1}/\mathfrak{m}_{\mathcal{V}_1}^{\mathbf{0}} & \cdots & \mathfrak{m}_{\mathcal{V}_1}^{\mathbf{p}_m}/\mathfrak{m}_{\mathcal{V}_1}^{\mathbf{0}} \\ \vdots & & \vdots \\ \mathfrak{m}_{\mathcal{V}_n}^{\mathbf{p}_1}/\mathfrak{m}_{\mathcal{V}_n}^{\mathbf{0}} & \cdots & \mathfrak{m}_{\mathcal{V}_n}^{\mathbf{p}_m}/\mathfrak{m}_{\mathcal{V}_n}^{\mathbf{0}} \end{bmatrix} \begin{bmatrix} a_{\mathbf{p}_1} \\ \vdots \\ a_{\mathbf{p}_m} \end{bmatrix} = \begin{bmatrix} \langle \phi \rangle_{\mathcal{V}_1} \\ \vdots \\ \langle \phi \rangle_{\mathcal{V}_n} \end{bmatrix} \tag{3.10}$$

which can be written as

$$\mathbf{B}\mathbf{a} = \boldsymbol{\phi} \tag{3.11}$$

where $\mathbf{B}$ is the stencil matrix, which is constructed using only the mesh geometry. The high-OrderFit scheme generates stencils using the same procedure as the cubicFit scheme. Assuming a stencil comprises at least as many cells as there are polynomial coefficients then $n \geq m$ and the matrix equation can be solved using a least-squares approach to find the unknown coefficients $\mathbf{a}$.

To obtain a stable transport scheme, we follow the approach of the cubicFit scheme by introducing multipliers $\mathbf{m}$ to obtain

$$\tilde{\mathbf{B}}\mathbf{a} = \mathbf{m} \cdot \boldsymbol{\phi} \tag{3.12}$$

where $\tilde{\mathbf{B}} = \mathbf{M}\mathbf{B}$ and $\mathbf{M} = \mathrm{diag}(\mathbf{m})$. The upwind cell and downwind cell have multipliers $m_u = 2^{10}$ and $m_d = 2^{10}$ respectively, and all peripheral points have multipliers $m_p = 1$.

The calculation of high-order cell volume moments and surface moments are required by equations (3.10) and (3.6) respectively. These volume and surface moments can be calculated exactly using the method of Tuzikov et al. (2003). We follow their method but, in order to avoid any degenerate triangles, we introduce a centre point shared by all triangles instead of triangulating polygons with only existing vertices.

While the highOrderFit transport scheme uses a total polynomial order $P = 3$ for stencils in the domain interior, a total polynomial order $P = 1$ is used for stencils near the boundary having fewer than 12 cells. This reduction in total polynomial order ensures that matrix equations are not underconstrained. This thesis does not assess the accuracy of the highOrderFit scheme near boundaries, and so the more sophisticated boundary treatment implemented in the cubicFit scheme has not been implemented in the highOrderFit scheme.

3.2 High-order transport over mountains

Section 2.2 presented a more challenging variant of the standard two-dimensional transport test formulated by Schär et al. (2002) to assess the numerical accuracy of the cubicFit scheme

transporting a tracer horizontally above very steep slopes represented by highly-distorted terrain-following meshes. In this section, we use the original test as specified by Schär et al. (2002) that has shallower slopes than the test presented in section 2.2. We make only one modification to the original test, choosing a smoother initial tracer to allow high-order convergence to be achieved (Holdaway et al., 2008). This test is used to measure the order of convergence of the cubicFit scheme and the highOrderFit scheme on undistorted cut cell meshes and distorted terrain-following meshes.

The test follows the specification by Schär et al. (2002), with the same domain size and boundary conditions as in section 2.2. The mountain profile is given by equation (2.51) and the prescribed velocity field is given by equation (2.53). As originally specified by Schär et al. (2002), the peak mountain height $h_0 = 3$ km, and the transition from zero flow near the ground to uniform horizontal flow aloft occurs between $z_1 = 4$ km and $z_2 = 5$ km. All other parameters relating to the mountain profile and velocity field are the same as those given in section 2.2.

The tracer density is given by equation (2.55) and is centred at $(x_0, z_0) = (-50$ km, 9 km$)$. In order to allow high-order convergence to be achieved, the exponent $\mathfrak{n} = 4$ such that the $\cos^{\mathfrak{n}}$ hill has $\mathfrak{n} - 1$ continuous derivatives (Holdaway et al., 2008). All other tracer parameters are the same as those given in section 2.2.

Tests are integrated for 10 000 s using the classical fourth-order Runge–Kutta time-stepping scheme with both cubicFit and highOrderFit transport schemes, and a time-step is chosen for each mesh so that the maximum Courant number is about 0.4. The analytic solution at $t = 10\,000$ s is centred at $(x_0, z_0) = (50$ km, 9 km$)$. To measure the order of convergence of the cubicFit scheme and the highOrderFit scheme, tests are performed using mesh spacings between $\Delta x = 5000$ m and $\Delta x = 250$ m. The vertical mesh spacing Δz is chosen so that $\Delta x : \Delta z = 2 : 1$ to match the original test specification by Schär et al. (2002).

The ℓ_2 and ℓ_∞ errors are measured on a series of basic terrain-following meshes and cut cell meshes (figure 3.1). The highOrderFit transport scheme achieves third-order convergence on basic terrain-following meshes and fourth-order convergence on cut cell meshes. While we have not formally analysed the order of accuracy of the highOrderFit scheme, we might expect to obtain fourth-order convergence under these ideal test conditions: first, the highOrderFit scheme uses a cubic reconstruction; second, cut cell meshes are undistorted away from the lower boundary and, third, the horizontal flow is aligned with the cut cell mesh.

The cubicFit transport scheme achieves second-order convergence on cut cell meshes, and the scheme achieves third-order convergence on basic terrain-following meshes, with error measures very similar to those obtained with the highOrderFit scheme. It is perhaps surprising that higher-order convergence is achieved when the mesh is distorted and flow is misaligned with the mesh. Further tests using very fine meshes should confirm whether third-order convergence is
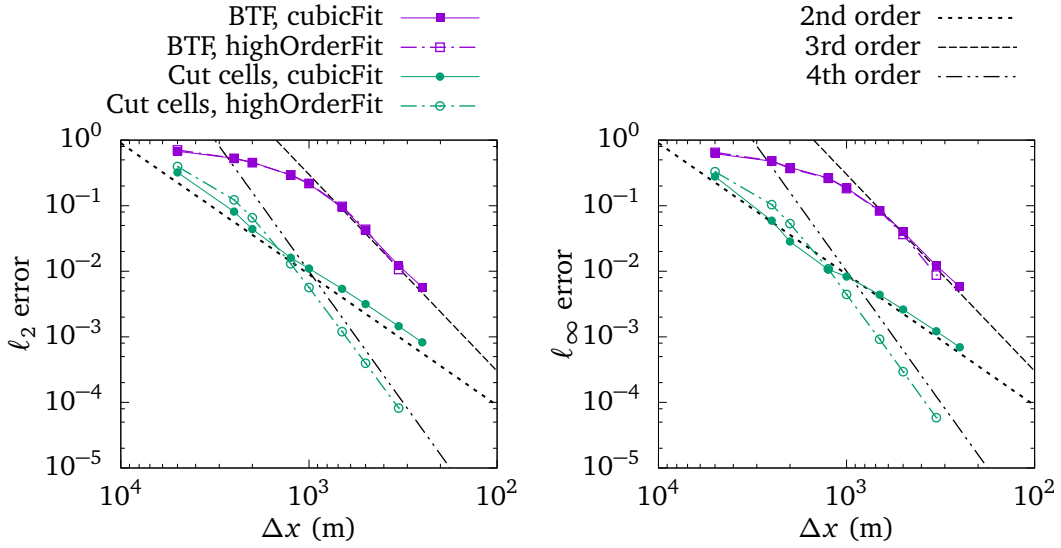
Figure 3.1: Numerical convergence in a test transporting a $\cos^4$ tracer horizontally over mountains. ℓ_2 (equation 2.56) and ℓ_∞ errors (equation 2.57) are marked at mesh spacings between $\Delta x = 5000\,\mathrm{m}$ and $\Delta x = 250\,\mathrm{m}$ using cubicFit and highOrderFit transport schemes on basic terrain-following and cut cell meshes.

maintained. We might expect that the cubicFit scheme tends towards second-order convergence at finer mesh spacings, otherwise the cubicFit scheme would become more accurate on basic terrain-following meshes than cut cell meshes!

At coarser mesh spacings, the cubicFit scheme is more accurate than the highOrderFit scheme. It should be possible to increase the accuracy of the highOrderFit scheme by adjusting the multiplier values used in the least-squares fit, but this has not been explored. At finer mesh spacings, the highOrderFit scheme becomes more accurate than the cubicFit scheme thanks to its higher order of convergence.

Further numerical experiments were performed using steeper slopes represented by basic terrain-following meshes, but the highOrderFit scheme produced instabilities above the steepest slopes. The stabilisation procedure used in the cubicFit scheme has not been implemented in the highOrderFit scheme, but examination of the stencils reveals that, using the highOrderFit scheme, some stencil weights violate the stability conditions given in equation (2.43). It is likely that the selective removal of some high-order terms for particularly distorted stencils could stabilise the highOrderFit scheme on highly-distorted meshes.

This series of horizontal transport tests demonstrates that, under favourable conditions with a sufficiently smooth tracer and uniform flow, the highOrderFit scheme is capable of fourth-order and third-order convergence on undistorted and distored meshes respectively. The tests also demonstrated a surprising result that the cubicFit scheme is seemingly capable of third-order

convergence only on distorted, terrain-following meshes. In the next section, we evaluate the highOrderFit scheme on distorted and undistorted meshes using a more challenging, deformational flow.

3.3   Deformational flow on a plane

The standard test case by Lauritzen et al. (2012) of deformational flow on a two-dimensional spherical shell was adapted by Chen et al. (2017) for use on a two-dimensional Cartesian plane. Since the highOrderFit formulation described in section 3.1 has not been extended to spherical geometry, we use the test case by Chen et al. (2017) to measure the order of convergence of the highOrderFit transport scheme in a time-varying, rotational velocity field. Tests are performed on uniform meshes, and meshes with distortions similar to those found on the cubed-sphere.

Following Chen et al. (2017), the domain is defined on a rectangular x–y plane that is 2π wide and π tall. The domain is periodic in the x direction with no normal flow imposed at the upper and lower boundaries. The discrete velocity field is defined using the streamfunction,

$$\Psi = \frac{\hat{\Psi}}{T}\sin^2\left(2\pi\left(\frac{x}{2\pi}-\frac{t}{T}\right)\right)\cos^2(y)\cos\left(\frac{\pi t}{T}\right)-\frac{2\pi y}{T} \tag{3.13}$$

where $\hat{\Psi}=10$, and $T=5$ is the duration of integration, after which time the analytic solution is equal to the initial condition. The initial tracer density ϕ is defined as the sum of two Gaussian hills,

$$\phi = \phi_1(x,y)+\phi_2(x,y) \tag{3.14}$$

where an individual hill ϕ_i is given by

$$\phi_i(x,y) = \phi_0\exp\left(-b\left(|\mathbf{x}-\mathbf{x}_i|\right)^2\right) \tag{3.15}$$

where $\phi_0 = 0.95\,\mathrm{kg\,m^{-3}}$ and $b=5$. The initial tracer field has two hills centred at

$$(x_1,y_1) = (5\pi/6,0)\,, \tag{3.16}$$

$$(x_2,y_2) = (7\pi/6,0)\,. \tag{3.17}$$

Tests were performed using the cubicFit and highOrderFit schemes using uniform meshes and meshes with distortions similar to a cubed-sphere mesh. Uniform meshes comprise square cells so that $\Delta x : \Delta y = 1:1$. Distorted meshes modify the corresponding uniform mesh using a coordinate transform,

$$x^\star = x, \quad y^\star = \begin{cases} \pi\frac{y-f}{\pi-2f} & \text{if } y \geq f, \\ \pi\frac{y-f}{\pi+2f} & \text{otherwise,} \end{cases} \tag{3.18}$$
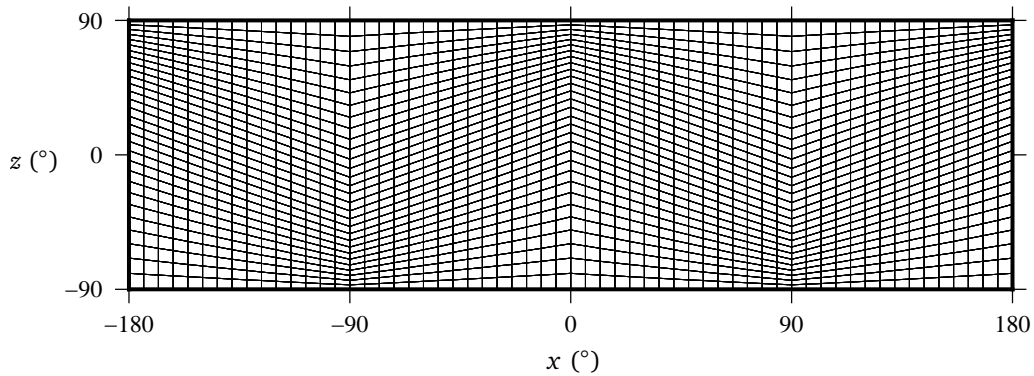
Figure 3.2: A distorted mesh on a Cartesian plane that has distortions similar to a cubed-sphere mesh. This coarse distorted mesh has 60×30 cells such that $\Delta x = 6°$.

where (x, y) are the physical coordinates, $(x^\star, y^\star)$ are the computational coordinates, and f is given by

$$f = \begin{cases} \tan(30°)\left(\frac{\pi}{4} - |x|\right) & \text{if } |x| \leq \frac{\pi}{2}, \\ \tan(30°)\left(|x| - \frac{3\pi}{4}\right) & \text{otherwise.} \end{cases} \tag{3.19}$$

Figure 3.2 illustrates a resulting distorted mesh with 60×30 cells. The classical fourth-order Runge–Kutta time-stepping scheme is used for both cubicFit and highOrderFit transport schemes, and tests are integrated using a time-step chosen for each mesh so that the maximum Courant number is about 0.4.

To measure numerical convergence, a range of mesh spacings are chosen between $\Delta x = 6°$ and $\Delta x = 0.375°$, and ℓ_2 and ℓ_∞ errors are calculated for the cubicFit and highOrderFit schemes on each mesh (figure 3.3). Similar to the results of deformational flow on a sphere in section 2.4, both the cubicFit scheme and the highOrderFit scheme are slow to converge on coarser meshes. At finer mesh spacings, the cubicFit scheme achieves second-order convergence and the highOrderFit scheme achieves third-order convergence. For both schemes, errors are slightly larger switching from a uniform mesh to a distorted mesh, but the order of convergence remains unchanged.

Unlike the results of transporting a smooth tracer over mountains (section 3.2), the highOrderFit scheme did not achieve fourth-order convergence for this deformational flow test case on a uniform mesh. It might be that fourth-order convergence can be obtained only with a uniform flow (J. Kent 2018, personal communication). Alternatively, it might be that the initial Gaussian tracer field, which is infinitely differentiable, becomes insufficiently smooth due to filamentation by the deformational velocity field.
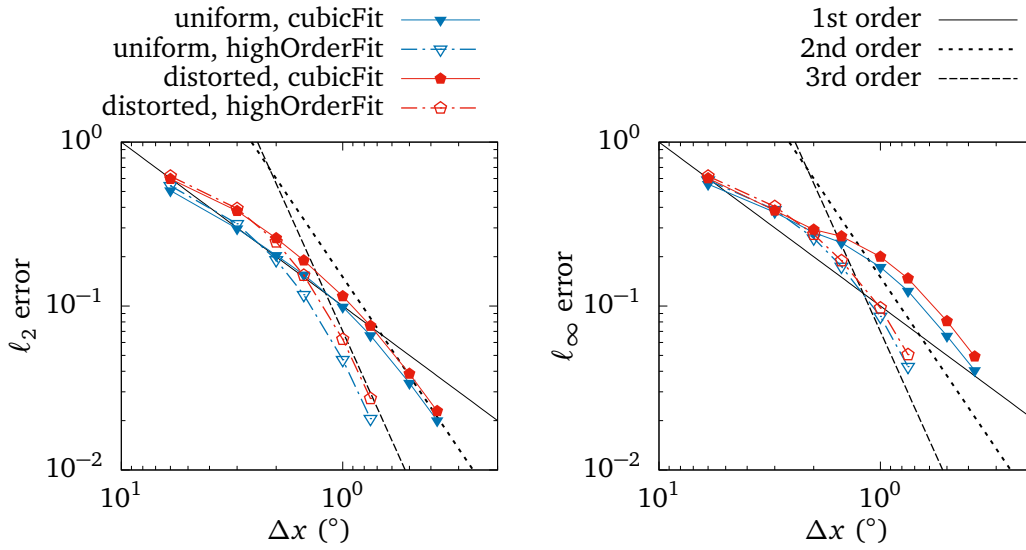
Figure 3.3: Numerical convergence of the deformational flow test on a Cartesian plane. ℓ_2 errors (equation 2.56) and ℓ_∞ errors (equation 2.57) are marked at mesh spacings between 6° and 0.375° using cubicFit and highOrderFit transport schemes on uniform and distorted meshes.

Results presented in this chapter demonstrate that, assuming a sufficiently smooth tracer, the highOrderFit transport scheme achieves third-order convergence or higher, irrespective of mesh distortions or the choice of velocity field. Thanks to its high-order convergence, the highOrderFit scheme is more accurate than the cubicFit scheme on all but the coarsest meshes.

4 A new mesh for representing the atmosphere above terrain

Highlights

- The new slanted cell mesh permits longer time-steps than those permited by the cut cell mesh, with time-steps comparable to terrain-following meshes

- Resting atmosphere simulations are more accurate using the new slanted cell method compared to terrain-following methods

- Unlike the multidimensional linear upwind scheme, the cubicFit scheme is numerically stable over very steep slopes represented by slanted cell meshes

---

Two sources of numerical error receive particular attention in atmospheric models: errors associated with transport terms and errors associated with the pressure gradient term. The previous two chapters developed transport schemes that reduce numerical errors associated with transport over mountains. This chapter seeks to reduce errors associated with the balance between the pressure gradient and gravity by representing the atmosphere above terrain with a new type of mesh, the 'slanted cell' mesh.

Pressure gradient errors result in spurious flows that are especially apparent for diurnal valley flows where synoptic-scale winds are weak (Fast, 2003), and even small velocity errors can produce large errors in derived quantities such as relative vorticity and potential vorticity (Hoinka and Zängl, 2004). These numerical errors are particularly large using terrain-following meshes with steeply sloping terrain (Zängl, 2012). Terrain-following meshes are typically implemented using a coordinate transform that introduces metric terms into the equations of motion. The
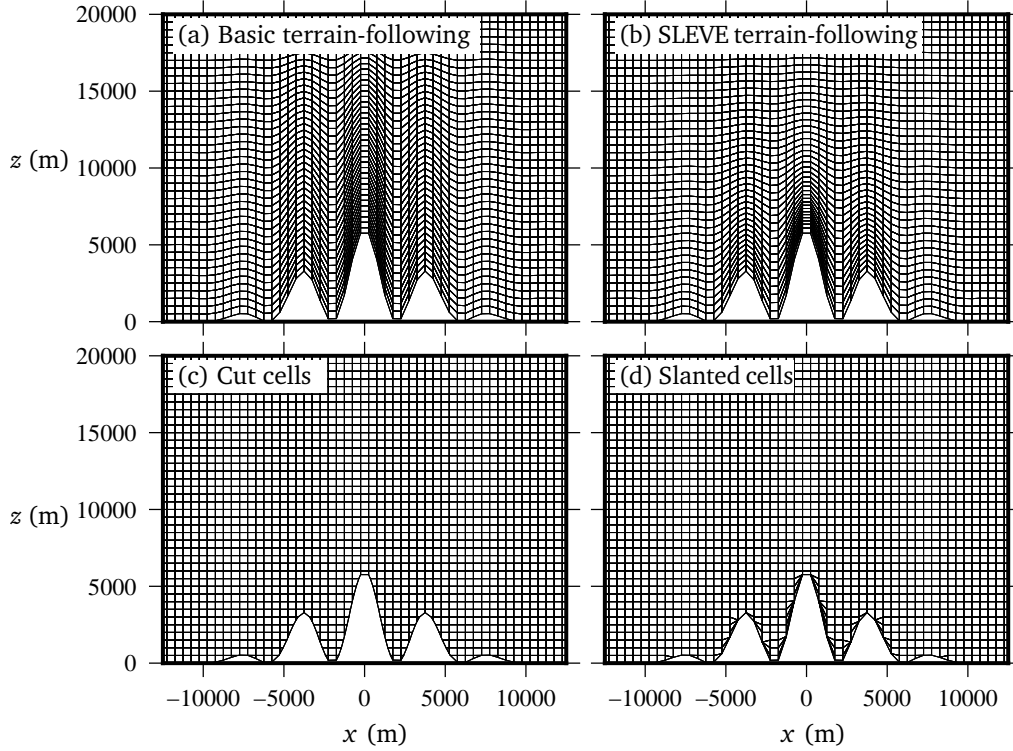
47

Figure 4.1: Two dimensional x-z meshes created with the (a) basic terrain-following, (b) SLEVE, (c) cut cell, and (d) slanted cell methods, used for the resting atmosphere tests in section 4.4. Cell edges are marked by thin black lines. The peak mountain height $h_0 = 6$ km. Only the central region of the domain is shown. The entire domain is 200 km wide and 20 km high.

horizontal pressure gradient $\partial p / \partial x|_z$ can be written as (Mahrer, 1984)

$$\left.\frac{\partial p}{\partial x}\right|_z = \left.\frac{\partial p}{\partial x}\right|_{z^\star} + \left.\frac{\partial z^\star}{\partial x}\right|_z \frac{\partial p}{\partial z^\star} \tag{4.1}$$

where $\partial / \partial x|_z$ denotes a horizontal derivative at a fixed height in the physical domain, and $\partial / \partial x|_{z^\star}$ denotes a horizontal derivative at a fixed model level in the computational domain, with $z^\star$ held constant. The first term on the right hand side of equation (4.1) is the change in pressure along the terrain-following coordinate surface, and the second term corrects for the vertical contribution in the first. These terms tend to be large and of opposite sign over steep terrain, and cancellation errors between the two terms result in pressure gradients errors that drive spurious flows.

There are two common approaches to reducing errors associated with terrain-following meshes. The first approach reduces the influence of the terrain on the mesh by choosing a vertical decay function that smooths mesh layers rapidly with height. The smooth level vertical (SLEVE) decay function formulated by Schär et al. (2002) and later improved by Leuenberger

et al. (2010), is one such approach. The SLEVE method achieves a less distorted terrain-following mesh in the middle and top of the domain than the BTF method. The difference is illustrated in figure 4.1, in which the SLEVE mesh (figure 4.1b) is noticeably smoother than the equivalent BTF mesh (figure 4.1a). The SLEVE method partitions the terrain height h into coarse-scale and fine-scale components, h_1 and h_2, such that $h = h_1 + h_2$, with each component having a different exponential decay. The transformation is defined as (Leuenberger et al., 2010)

$$z = z^{\star} + h_1 b_1 + h_2 b_2 \tag{4.2}$$

where the vertical decay functions are given by

$$b_i = \frac{\sinh\left((H/s_i)^n - (z^{\star}/s_i)^n\right)}{\sinh\left(H/s_i\right)^n} \tag{4.3}$$

and s_1 and s_2 are the scale heights of coarse-scale and fine-scale terrain respectively. The exponent n was introduced by Leuenberger et al. (2010) in order to increase cell thickness in the layers nearest the ground, allowing longer timesteps. In their experiments, Leuenberger et al. (2010) found the exponent to have an optimal value of $n = 1.35$. Choosing $n = 1$ gives the decay functions used by Schär et al. (2002). While SLEVE can produce very smooth meshes, the coordinate transform becomes non-monotonic and the SLEVE mesh becomes tangled if parameter values are chosen poorly (Schär et al., 2002; Leuenberger et al., 2010), making it difficult to produce a very smooth global mesh with real terrain (C. Kühnlein 2015, personal communication). Many operational atmospheric models use some form of coordinate smoothing (Eckermann, 2009), and this approach has been found to reduce pressure gradient errors (Schär et al., 2002; Leuenberger et al., 2010; Klemp, 2011).

The second approach to reducing pressure gradient errors is to improve the accuracy in calculating the horizontal pressure gradient itself. Instead of calculating the horizontal pressure gradient in the computational domain, the techniques proposed by Klemp (2011) and Zängl (2012) both involve interpolation onto z levels in the physical domain. This gave them the flexibility to design more accurate horizontal pressure gradient discretizations using more appropriate stencils.

Another approach is to improve the balance the between pressure gradient and gravity by reducing the spurious solenoidal term using a curl-free gradient formulation (Thuburn and Cotter, 2012), and this formulation was included in the nonhydrostatic model by Weller and Shahrokhi (2014). Since their model calculates gradients in the Exner function of pressure and does not calculate gradients of pressure directly, the formulation cannot be exactly curl-free. Nevertheless, Weller and Shahrokhi (2014) found that switching from a horizontal gradient formulation to the curl-free gradient formulation improved idealised simulations over orography represented by terrain-following meshes.

Cut cell meshes (also known as shaved cell meshes or embedded boundary meshes) are less distorted than any smoothed terrain-following mesh, and some studies have shown examples where cut cells produce more accurate results when compared to terrain-following meshes (Good et al., 2014; Steppeler et al., 2013). Although cut cell meshes are almost entirely undistorted, when explicit methods are used with cut cells, the small cell problem must be overcome in order to avoid severe time-step constraints.

We seek a new type of mesh that improves the balance between the pressure gradient and gravity compared to terrain-following methods, and avoids the severe time-step constraints associated with arbitrarily small cut cells. Section 4.1 describes the slanted cell method which is designed to satisfy these criteria. Section 4.2 presents a new two-dimensional test that challenges transport schemes by transporting a tracer along the ground through slanted cells, and this test is used to measure time-step constraints for terrain-following, cut cell and slanted cell meshes. Section 4.3 outlines the discretisation of the fully compressible model taken from Weller and Shahrokhi (2014) which includes a curl-free pressure gradient formulation. In section 4.4, the fully compressible model is used to simulate a standard resting atmosphere test case (Klemp, 2011), comparing results using terrain-following, cut cell and slanted cell meshes.

4.1   Slanted cell method

The slanted cell method is straightforward, and slanted cell meshes are always free of mesh tangling by construction. Starting from a uniform rectangular mesh, all cell vertices that lie beneath the orography are moved up to the surface. Additionally, to avoid creating very thin cells, all vertices up to $2\Delta z/5$ above the orography can be moved down to the surface. Where all four of a cell's vertices are moved, the cell has zero volume and so it is removed. Where two vertices at the same horizontal location are moved up to the surface they will occupy the same point; this results in a zero-length edge that is removed to create a triangular cell. Figure 4.2 shows how a 2×3-cell, uniform rectangular mesh is transformed into a slanted cell mesh. Cells c_5 and c_6 are removed because they have zero volume, and the zero-length edge at point q is removed to create a triangular cell, c_4. Point p is moved down because it is within $2\Delta z/5$ of the surface, avoiding the creation of a very thin cell. We have not explored the sensitivity of results using values other than $2\Delta z/5$, but we did find that this approach reduces numerical errors on some meshes with very thin slanted cells.

The slanted cell method does generate some small cells but, unlike the cut cell method, the width of slanted cells is never altered. Since a no normal flow condition is imposed at the lower boundary, flow must be parallel to the surface and there is only very weak flow across the long, upper face of slanted cells. Hence, slanted cell meshes should not suffer from severe time-step
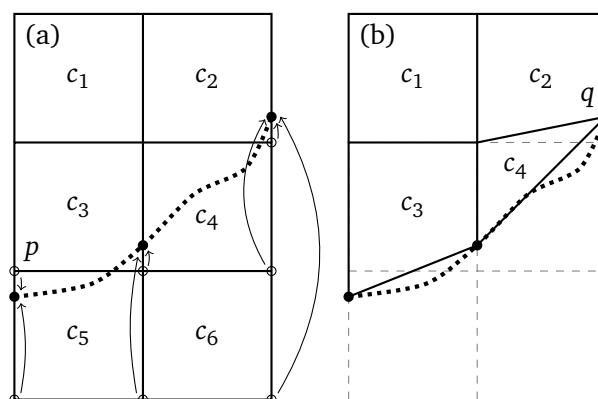
Figure 4.2: Illustration of a slanted cell mesh (a) before, and (b) after construction. The terrain surface, denoted by a heavy dotted line, intersects a uniform rectangular mesh comprising six cells, $c_1, \ldots, c_6$. The cell vertices, marked by open circles, are moved to the points at which the terrain intersects vertical cell edges, marked by filled circles. Cells that have no volume are removed. Where a cell has two vertices occupying the same point, the zero-length edge that joins those vertices is removed. In this illustration, cells c_5 and c_6 are removed because they have no volume, and the zero-length edge at point q is removed to create a triangular cell, c_4. Point p is moved down because it is within $2\Delta z/5$ of the surface, avoiding the creation of a thin cell.

constraints associated with arbitrarily small cut cells because slanted cells are never shortened in the direction of flow. An example of a slanted cell mesh is illustrated in figure 4.1d for comparison with the equivalent BTF (figure 4.1a), SLEVE (figure 4.1b), and cut cell (figure 4.1c) meshes, with the same mesh spacing and mountain profile used for all meshes.

4.2 Transport over a mountainous lower boundary

The two-dimensional tests performed in chapter 2 transported tracers positioned well above the terrain surface. Here we formulate a new test, positioning the tracer at the ground in order to assess the accuracy of transport schemes immediately above a mountainous lower boundary. Results are compared between the cubicFit scheme and the linearUpwind scheme on basic terrain-following, cut cell and slanted cell meshes. The test presents a particular challenge to transport schemes as they must transport the tracer through arbitrarily small cut cells and distorted slanted cells.

The domain size and mountain profile are the same as those in the horizontal tracer transport test in section 2.2, with a mesh spacing of $\Delta x = 1000\,\text{m}$ and $\Delta z = 500\,\text{m}$. In order to present the most challenging test on slanted cell meshes vertices are not moved downwards, and so thin cells remain near mountain peaks. Cell edges in the central region of the domain are shown in
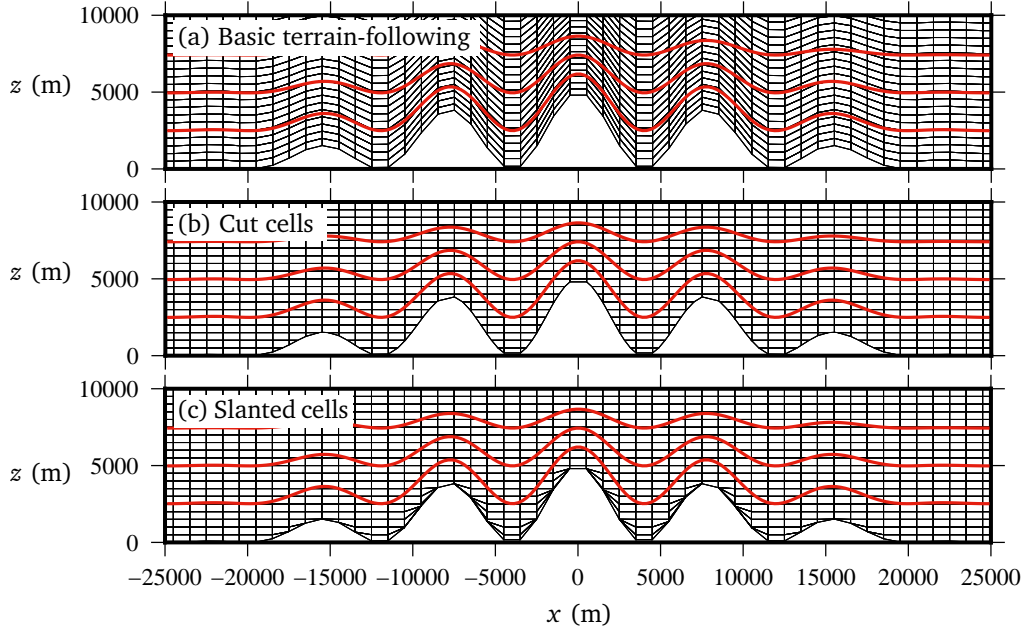
Figure 4.3: Two dimensional x-z meshes created with the (a) basic terrain-following, (b) cut cell, and (c) slanted cell methods, used for the tracer transport tests in section 4.2. Cell edges are marked by thin black lines. The peak mountain height $h_0 = 5$ km. The velocity field is the same for all mesh types with streamlines marked on each panel by thick red lines. The velocity field (equation 2.58) follows the lower boundary and becomes entirely horizontal above $H_1 = 10$ km. The mesh spacing is $\Delta x = 1000$ m and $\Delta z = 500$ m. Only the lowest 10 km for the central region of the domain is shown. The entire domain is 300 km wide and 25 km high.

figure 4.3 for each of the three mesh types. Cells in the BTF mesh are highly distorted over steep slopes (figure 4.3a) while the cut cell mesh (figure 4.3b) and slanted cell mesh (figure 4.3c) are orthogonal everywhere except for cells nearest the ground.

A velocity field is prescribed using equation (2.58) so that the flow follows the terrain at the surface and becomes entirely horizontal above $H_1 = 10$ km. The value of H_1 is chosen to be much smaller than the domain height H in equation (2.52) so that flow crosses the surfaces of the BTF mesh. This is evident in figure 4.3a where the the velocity streamlines are tangential to the mesh only at the ground. The flow is deliberately misaligned with the BTF, cut cell and slanted cell meshes away from the ground (figure 4.3) to ensure that flow always crosses mesh surfaces in order to challenge the transport schemes.

The tracer is defined again by equation (2.55) but is now positioned at the ground with $(x_0, z_0) = (-50 \text{ km}, 0 \text{ km})$ 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 4.1). An analytic solution at 10 000 s is obtained by calculating

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

Table 4.1: 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.
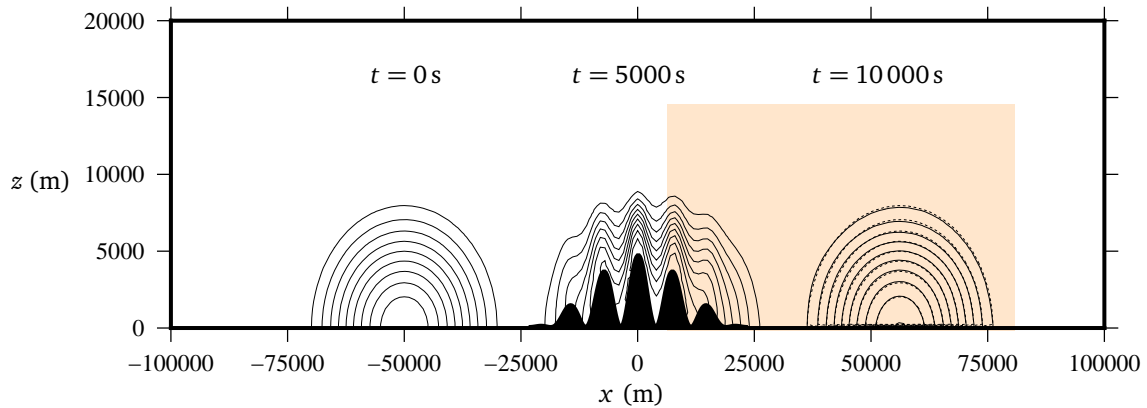

Figure 4.4: Evolution of the tracer in the two-dimensional transport test over a mountainous lower boundary. The tracer is transported to the right over the wave-shaped terrain. Tracer contours are every $0.1 \, \mathrm{kg \, m^{-3}}$. The result obtained using the cubicFit scheme on the basic terrain-following mesh is shown at $t = 0 \, \mathrm{s}$, $t = 5000 \, \mathrm{s}$ and $t = 10\,000 \, \mathrm{s}$ with solid black contours. The analytic solution at $t = 10\,000 \, \mathrm{s}$ is shown with dotted contours. The mesh spacing is $\Delta x = 1000 \, \mathrm{m}$ and $\Delta z = 500 \, \mathrm{m}$. The shaded box indicates the region that is plotted in figure 4.5.

the new horizontal position of the tracer using equation (2.63). By solving this equation we find that $x(t = 10\,000 \, \mathrm{s}) = 6244.087 \, \mathrm{m}$ when $h_0 = 5 \, \mathrm{km}$.

The tracer density boundary conditions are the same as those in section 2.2. 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 $h_0 = 5 \, \mathrm{km}$ to examine errors on different mesh types using the linearUpwind and cubicFit transport schemes. The second series varies the peak mountain height to examine the sensitivity of the two 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.
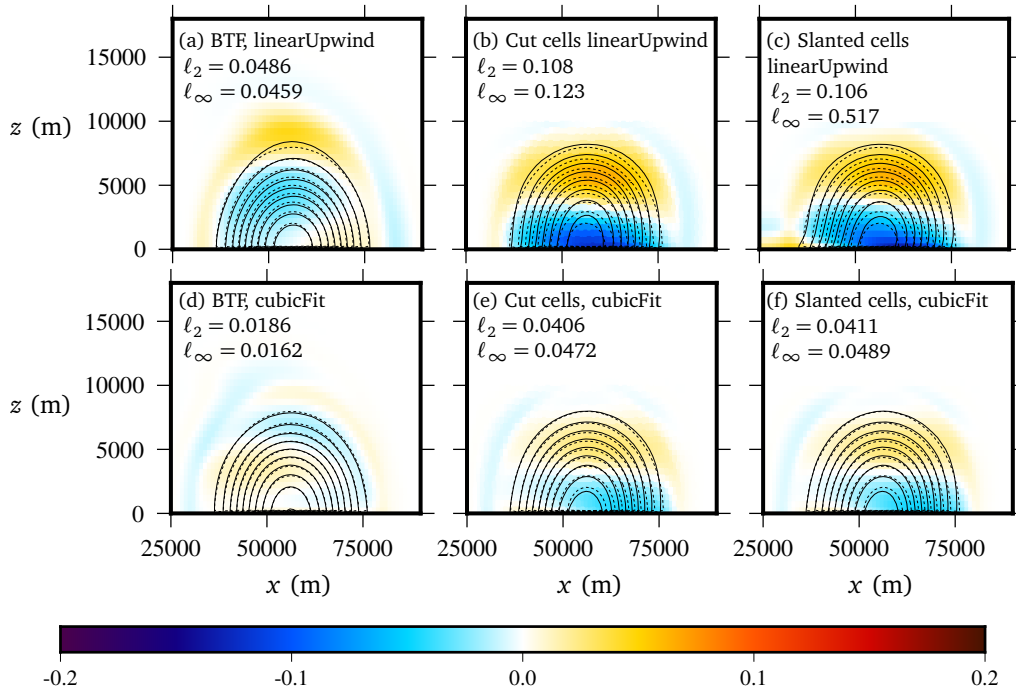
Figure 4.5: Tracer contours at $t = 10\,000\,\text{s}$ for the two-dimensional tracer transport tests over a mountainous lower boundary. A region in the lee of the mountain is plotted corresponding to the shaded area in figure 4.4. Results are presented on BTF, cut cell and slanted cell meshes (shown in figure 4.3) 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 $0.1\,\text{kg}\,\text{m}^{-3}$. The mesh spacing is $\Delta x = 1000\,\text{m}$ and $\Delta z = 500\,\text{m}$.

Comparison of numerical accuracy between mesh types and transport schemes

For the first series of tests with $h_0 = 5\,\text{km}$, tracer contours at the initial time $t = 0\,\text{s}$, half-way time $t = 5000\,\text{s}$, and end time $t = 10\,000\,\text{s}$ are shown in figure 4.4 using the cubicFit scheme on the BTF mesh. As apparent at $t = 5000\,\text{s}$, the tracer is distorted by the terrain-following velocity field as it passes over the mountain as expected, and its original shape is restored once it has cleared the mountain by $t = 10\,000\,\text{s}$. Slight errors are apparent at $t = 10\,000\,\text{s}$ 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 (figures 4.5a, 4.5b and 4.5c respectively) and the cubicFit scheme (figures 4.5d, 4.5e and 4.5f respectively). Results are least accurate using the linearUpwind scheme on the slanted cell mesh (figure 4.5c) with the final tracer being slightly distorted.
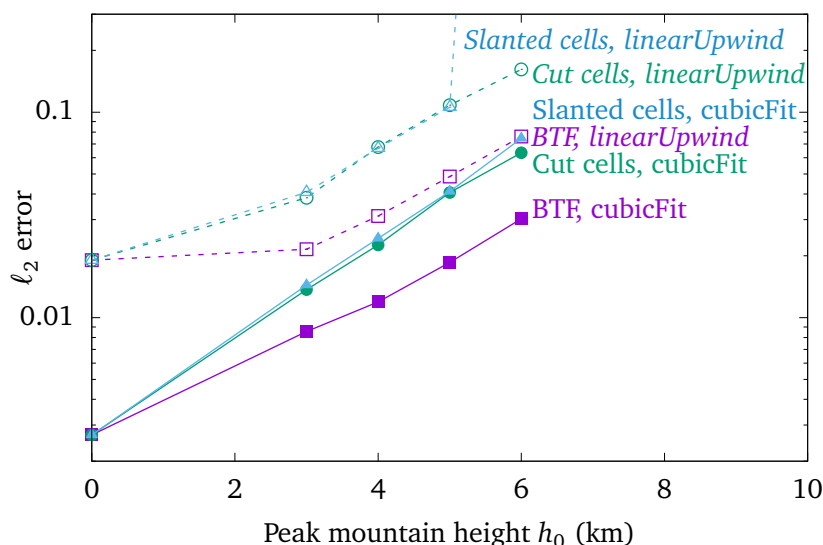
Figure 4.6: Error measures for the two-dimensional tracer transport tests over a mountainous lower boundary. Peak mountain heights h_0 are from 0 km to 6 km. Results are compared on BTF, cut cell and slanted cell meshes using the linearUpwind and the cubicFit schemes. At $h_0 = 0$ km the terrain is entirely 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.

The ℓ_∞ error magnitude is reduced by using the linearUpwind scheme on the cut cell mesh (figure 4.5b), but the shape of the error remains the same. On the BTF mesh (figure 4.5d), cut cell mesh (figure 4.5e) and slanted cell mesh (figure 4.5f), the cubicFit scheme is more accurate than the linearUpwind scheme.

Numerical stability and numerical accuracy with increasingly steep slopes

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 4.1). The ℓ_2 error was calculated by subtracting the analytic solution from the numerical solution (figure 4.6). 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. The linearUpwind
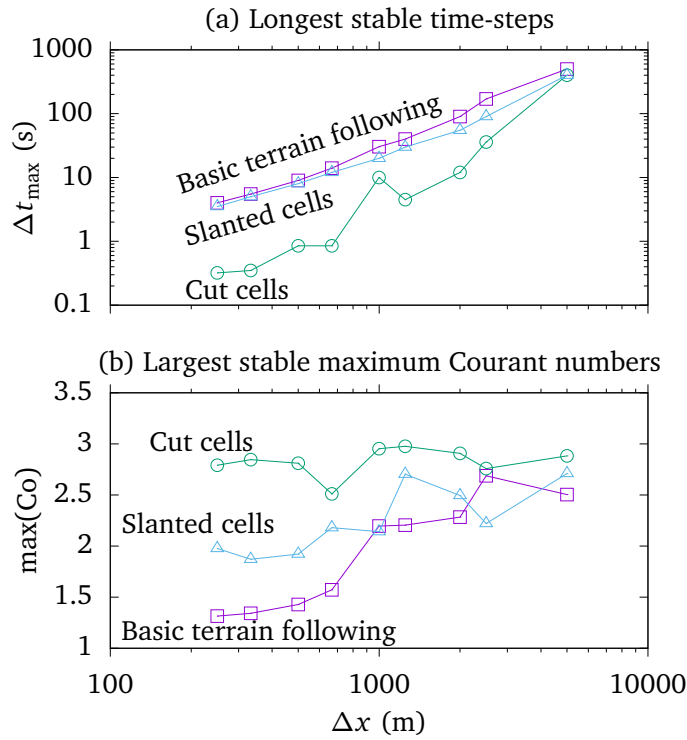
Figure 4.7: (a) Longest stable time-steps, Δt_{max}, and (b) largest stable maximum Courant numbers, max(Co), for the two-dimensional tracer transport test over a mountainous lower boundary. Results were obtained on basic terrain-following, cut cell and slanted cell meshes at mesh spacings between $\Delta x = 5000\,\text{m}$ and $\Delta x = 250\,\text{m}$. The largest stable maximum Courant numbers were calculated from the corresponding longest stable time-steps using equation (2.4). The peak mountain height $h_0 = 6\,\text{km}$.

scheme is unstable on the slanted cell mesh with a peak mountain height $h_0 = 6\,\text{km}$ despite using a Courant number of 0.428. The cubicFit scheme yields stable results in all tests, and cubicFit is more accurate than linearUpwind in all tests.

Numerical stability limits of the cubicFit transport scheme

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 2.2). The tracer was transported on BTF, slanted cell and cut cell meshes with a variety of mesh spacings between $\Delta x = 5000\,\text{m}$ and $\Delta x = 125\,\text{m}$. Δz was chosen so that a constant aspect ratio is preserved such that $\Delta x : \Delta z = 2:1$. The peak mountain height $h_0 = 6\,\text{km}$. For each test, the time-step was adjusted in order to find the largest stable time-step, Δt_{max} (figure 4.7a). 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 more stringent time-step constraint than BTF meshes but still exhibit similar linear scaling with mesh spacing.

Figure 4.7b presents the largest stable maximum Courant numbers, max(Co), which were calculated by substituting $\Delta t = \Delta t_{\max}$ into equation (2.4). 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 thesis focuses on the spatial discretisation of the cubicFit scheme, but the stability limit depends also upon the choice of time-stepping. We have not calculated a theoretical Courant number limit, although such an analysis should be possible using the techniques of Baldauf (2008).

This new test case demonstrates that the cubicFit transport scheme is more accurate than the linearUpwind scheme on all meshes, and only the cubicFit scheme can achieve stable results on slanted cell meshes with very steep slopes. The slanted cell method exhibits a time-step constraint that scales linearly with mesh spacing, and slanted cells avoid severe time-step constraints associated with arbitrarily small cut cells. Next, we incorporate the cubicFit transport scheme into a model of the fully compressible Euler equations.

4.3 Discretisation of the fully compressible Euler equations

The finite volume model of the fully compressible Euler equations is taken from Weller and Shahrokhi (2014), given by

$$\text{Momentum} \qquad \frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = \rho \mathbf{g} - c_p \rho \theta \nabla \Pi - \mu \rho \mathbf{u} \qquad (4.4a)$$

$$\text{Continuity} \qquad \frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{u} = 0 \qquad (4.4b)$$

$$\text{Thermodynamic equation} \qquad \frac{\partial \rho \theta}{\partial t} + \nabla \cdot \rho \mathbf{u} \theta = 0 \qquad (4.4c)$$

$$\text{Ideal gas law} \qquad \Pi^{(1-\kappa)/\kappa} = \frac{R \rho \theta}{p_0} \qquad (4.4d)$$

where ρ is the density, $\mathbf{u}$ is the velocity field, $\mathbf{g}$ is the gravitational acceleration, c_p is the heat capacity at constant pressure, $\theta = T (p_0/p)^\kappa$ is the potential temperature, T is the temperature,

p is the pressure, $p_0 = 1000\,\text{hPa}$ is a reference pressure, $\Pi = (p/p_0)^\kappa$ is the Exner function of pressure, and $\kappa = R/c_p$ is the gas constant to heat capacity ratio. μ is a damping function that can be used to absorb momentum in a sponge layer near the upper boundary.

The model uses the C-grid staggering in the horizontal and the Lorenz staggering in the vertical such that θ, ρ and Π are stored at cell centroids and the covariant component of velocity at cell faces. The model is configured in an inertial frame without Coriolis forces.

Acoustic and gravity waves are treated implicitly and transport terms are treated explicitly. The trapezoidal implicit treatment of fast waves and the Hodge operator suitable for non-orthogonal meshes are described in the appendix to Shaw and Weller (2016). To avoid time-splitting errors between transport and fast waves, transport is time-stepped using a three-stage, second-order Runge-Kutta scheme. The transport terms of the momentum equation (4.4a) and thermodynamic equation (4.4c) are discretised in flux form using either the linearUpwind scheme or the cubicFit scheme as desired.

This model is suitable for arbitrary meshes and includes a curl-free pressure gradient formulation. In the next section, we use this model to compare the accuracy of hydrostatic balance calculations using terrain-following, cut cell and slanted cell meshes.

4.4 Stratified atmosphere initially at rest

Diurnal valley and slope flows are associated with weak synoptic-scale winds, and cold air that sinks along sloping terrain can stagnate for days after becoming trapped in topographic basins (Chow et al., 2013). The test case by Klemp (2011) is an idealised representation of such phenomena, in which a wave-shaped mountain is submerged in a stably stratified atmosphere at rest in hydrostatic balance. The analytic solution is time-invariant, but numerical errors in calculating pressure gradients can give rise to spurious flows which become stronger over steeper terrain (Klemp, 2011). Results are compared using terrain-following, cut cell and slanted cell meshes.

Following Klemp (2011), the domain is 200 km wide and 20 km high, and the mesh spacing is $\Delta x = \Delta z^\star = 500\,\text{m}$. All boundary conditions are no normal flow. The wave-shaped mountain profile has a surface height, h, given by

$$h(x) = h_0 \exp\left(-\left(\frac{x}{a}\right)^2\right)\cos^2(\alpha x) \tag{4.5}$$

where $a = 5\,\text{km}$ is the mountain half-width $\lambda = 4\,\text{km}$ is the wavelength and h_0 is the peak mountain height. For the optimised SLEVE mesh, the coarse-scale component h_1 is specified as

$$h_1(x) = \frac{1}{2}h_0 \exp\left(-\left(\frac{x}{a}\right)^2\right). \tag{4.6}$$
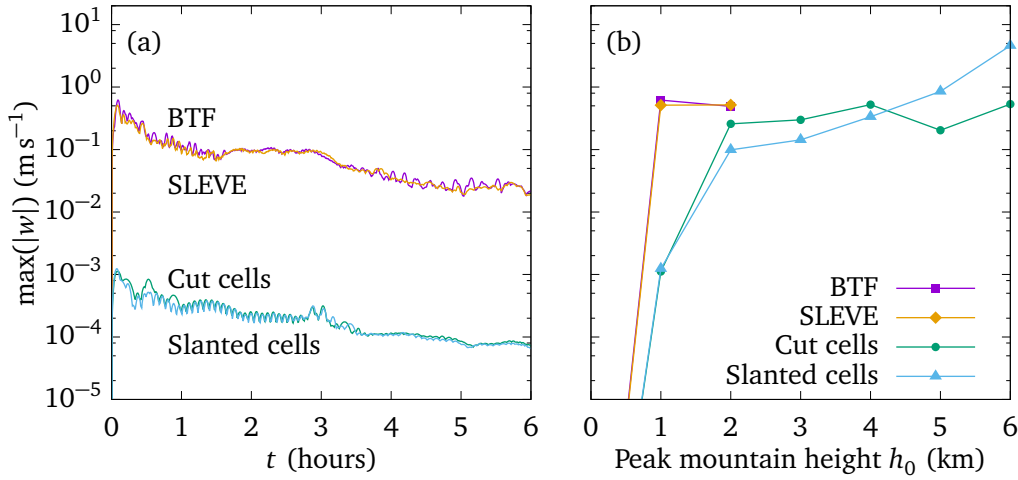
Figure 4.8: Spurious vertical velocities in the resting atmosphere test using BTF, SLEVE, cut cell and slanted cell meshes. (a) Time series of spurious vertical velocities for a peak mountain height $h_0 = 1$ km, with the maximum absolute value calculated at each time-step. (b) Sensitivity to peak mountain height h_0, with the maximum absolute value calculated across all time-steps.

To accommodate a range of mountain heights we choose a coarse scale height $s_1 = 20$ km and a fine scale height $s_2 = 8$ km. Following Leuenberger et al. (2010) the optimal exponent value of $n = 1.35$ is used. These parameter values result in a SLEVE mesh that is more distorted than the SLEVE mesh used by Klemp (2011), but the choice is necessary to avoid mesh tangling with mountains higher than 1 km.

The initial potential temperature field has a nonlinear vertical profile in the lower atmosphere, with $\theta(z = 0) = 288$ K and a constant static stability with Brunt-Väisälä frequency $N = 0.01\,\text{s}^{-1}$ everywhere, except for a more stable layer of $N = 0.02\,\text{s}^{-1}$ between $2\,\text{km} \leq z \leq 3\,\text{km}$. The Exner function of pressure is calculated so that it is in discrete hydrostatic balance in the vertical direction (Weller and Shahrokhi, 2014).

Momentum and potential temperature are transported using the cubicFit transport scheme. The damping function μ is set to $0\,\text{s}^{-1}$. Unlike Klemp (2011), there is no eddy diffusion in the equation set.

The test is integrated forward by 6 hours using a time-step of $\Delta t = 25\,\text{s}$ on the BTF, SLEVE, cut cell and slanted cell meshes with a peak mountain height $h_0 = 1$ km. For each mesh, the maximum absolute vertical velocity is calculated at each time-step as a measure of the spurious flow generated by numerical errors. In agreement with Klemp (2011), magnitudes of vertical velocity peak shortly after integration begins and magnitudes are larger on more distorted meshes (figure 4.8a). However, magnitudes are much smaller comparing results on the terrain-following meshes with those from Klemp (2011): results in figure 4.8a, which use a curl-free pressure

gradient formulation, have maximum absolute vertical velocities of $0.62\,\mathrm{m\,s^{-1}}$, compared with a maximum of $\sim 7\,\mathrm{m\,s^{-1}}$ found by Klemp (2011) using their improved horizontal pressure gradient formulation. The results on terrain-following meshes in figure 4.8a have similar maximum errors as Weller and Shahrokhi (2014) but, due to the more stable split into implicitly and explicitly treated terms (described in the appendix to Shaw and Weller (2016)), the errors decay over time due to the dissipative nature of the transport scheme. Unlike the result from Klemp (2011), spurious flows are similar on both terrain-following meshes even though the SLEVE mesh is less distorted than the BTF mesh.

Compared to results on the terrain-following meshes, spurious flows are two orders of magnitude smaller on the cut cell mesh and the slanted cell mesh with a maximum absolute vertical velocity of $\sim 1 \times 10^{-3}\,\mathrm{m\,s^{-1}}$. Good et al. (2014) found the maximum vertical velocity in their cut cell model was $1 \times 10^{-12}\,\mathrm{m\,s^{-1}}$, which is better than any result obtained here. It is worth noting that our model stores values at the geometric centre of cut cells, whereas the model used by Good et al. (2014) has cell centres at the centre of the uncut cell, resulting in the centre of some cut cells being below the ground (S.-J. Lock 2014, personal communication). This means that the mesh is effectively regular when calculating horizontal and vertical gradients, and this would account for the very small velocities found by Good et al. (2014).

To evaluate the slanted cell method with steeper slopes, we perform a second series of tests with peak mountain heights ranging from $h_0 = 0\,\mathrm{km}$ to $h_0 = 6\,\mathrm{km}$. The BTF, SLEVE, cut cell and slanted cell meshes with the largest peak mountain height of $h_0 = 6\,\mathrm{km}$ are shown in figure 4.1. To obtain a single measure of spurious flow for a given mesh, the maximum absolute vertical velocity is calculated across all time-steps (figure 4.8b). The most accurate results are obtained without mountains where $h_0 = 0\,\mathrm{km}$ when all meshes become identical, with $\max(|w|) \sim 1 \times 10^{-11}\,\mathrm{m\,s^{-1}}$. Using terrain-following meshes, the model becomes unstable beyond $h_0 = 2\,\mathrm{km}$. Using cut cell meshes, maximum vertical velocities are almost constant at $\sim 0.5\,\mathrm{m\,s^{-1}}$ beyond $h_0 = 1\,\mathrm{km}$ because cut cell mesh distortions are largely independent of mountain height. Using slanted cell meshes, maximum vertical velocities are one to two orders of magnitude smaller than those found on terrain-following meshes at a given mountain height. Unlike results on terrain-following meshes, slanted cell meshes yield stable results for all mountain heights, although maximum vertical velocities increase with peak mountain height as slanted cells become increasingly distorted. Up to a peak mountain height of $h_0 = 4\,\mathrm{km}$, slanted cell meshes produce results that are more accurate than those obtained for any other mesh.

In summary, spurious velocities in the resting atmosphere test were similar on both types of terrain-following mesh, with errors being much smaller compared to those from Klemp (2011). The maximum absolute vertical velocity was decreased by one to two orders of magnitude using cut cell and slanted cell meshes so we conclude that, in this test, mesh distortion, or lack of

alignment of the mesh with surfaces of constant gravitational potential, is the primary cause of numerical error. The resting atmosphere test presented a challenge to the pressure gradient formulation but the resultant spurious flows presented no particular challenge to the cubicFit transport scheme. We will turn our attention to transport-dominated flow in the next chapter.

5 A new test case to excite the Lorenz computational mode

Highlights

- A new idealised two-dimensional test case reveals spurious grid-scale waves excited by the Lorenz computational mode

- The Charney–Phillips staggering is generalised for arbitrary meshes, reducing to the classical Charney–Phillips staggering on traditional meshes

- A new fully compressible Euler model that implements the generalised Charney–Phillips formulation is free from spurious grid-scale waves

The Lorenz computational mode arises from having one too many degrees of freedom in the Lorenz staggering of variables, and it is often excited by thermal forcing, producing spurious, vertical, two-grid alternating waves (Schneider, 1987; Arakawa and Konor, 1996). In the Lorenz staggering (Lorenz, 1960), the pressure and vertical velocity variables are staggered, with the thermodynamic variable collocated with the pressure variable (figure 5.1a). Hydrostatic models calculate the hydrostatic balance equation and nonhydrostatic models calculate the vertical momentum equation and, in both cases, vertical averaging of the thermodynamic variable means that these spurious grid-scale waves persist because they become invisible to the model (Arakawa and Konor, 1996). Spurious grid-scale waves have been attributed to the Lorenz computational mode in the Global Environmental Multiscale 3 model (Girard et al., 2014), and the Korea Institute of Atmospheric Prediction Systems Integrated Model (Yi and Park, 2017) amongst others, and these non-physical waves can lead to spurious rainfall in atmospheric

Figure 5.1: (a) Lorenz and (b) Charney–Phillips vertical staggering of variables. Dashed lines mark the vertical position of cell centres and solid lines mark the vertical position of horizontal faces. The Lorenz staggering collocates the thermodynamic variable θ with the pressure variable p and horizontal velocity u. The Charney–Phillips staggering collocates the thermodynamic variable with the vertical velocity w. Adapted from Holdaway et al. (2013).

models (Hollingsworth, 1995), inaccurate simulations of idealised hurricanes (Zhu and Smith, 2003), and spurious instabilities in ocean models (Bell and White, 2017).

The computational mode can be at least partially controlled by using sufficient vertical diffusion (Chang, 1992; Zadra et al., 2004), or by using a higher-order vertical discretisation (Untch and Hortal, 2004; Guerra and Ullrich, 2016; Yi and Park, 2017), but the computational mode can only be properly eliminated by choosing an alternative staggering of variables that removes the extra degree of freedom. One such alternative is the Charney–Phillips staggering (Charney and Phillips, 1953), in which the thermodynamic variable is collocated with vertical velocity (figure 5.1b), avoiding any vertical averaging in calculating the vertical momentum equation. Due to the errors associated with the Lorenz computational mode, the Global Environmental Multiscale model switched from a Lorenz staggering to a Charney–Phillips staggering between model versions 3 and 4 (Girard et al., 2014), and a Charney–Phillips staggering is also used in the Met Office Unified Model (Davies et al., 2005) and the Global/Regional Assimilation and Prediction System (Yang et al., 2007). Thuburn and Woollings (2005) exhaustively tested different combinations of vertical coordinates, prognostic variables and their staggerings, and found that a Charney–Phillips staggering has better dispersion properties than a Lorenz staggering for any given choice of vertical coordinate and prognostic variables. Numerical experiments performed by Cullen et al. (1997) revealed that a model with a Charney–Phillips staggering reduced spurious gravity waves and had better geostrophic adjustment compared to the same model with a Lorenz staggering.

While the Charney–Phillips staggering avoids vertical averaging of the thermodynamic variable in calculating the vertical momentum equation, Davies et al. (2005) notes that horizontal

pressure gradient calculations can involve vertical averaging of the thermodynamic variable which is inaccurate in the lowest layers where there are strong temperature gradients. Holdaway et al. (2013) note that, in calculating the Richardson number for boundary layer schemes, averaging that is necessary with the Charney–Phillips staggering is avoided by using a Lorenz staggering.

Previous studies have used a variety of test cases to compare different model variants using Lorenz and Charney–Phillips staggering. One of the earliest comparisons was made by Arakawa and Moorthi (1988) who found that, without additional diffusion, their numerical solutions of a baroclinic instability test were dominated by short-wave noise. Later, Arakawa and Konor (1996) performed the same test to find that a model using a Charney–Phillips staggering did not suffer from spurious noise, with the Charney–Phillips model needing no additional diffusion. In the same study, Arakawa and Konor (1996) proposed new test cases that use thermal forcing to excite the Lorenz computational mode, and these tests clearly reveal spurious grid-scale waves that grow and persist throughout the simulation. Based on the work of Arakawa and Konor (1996), Untch and Hortal (2004) developed a new test case, using thermal forcing to excite spurious grid-scale waves in a 600-day integration of a global, 3D model.

While these tests have proved useful, they are not ideally suited for dynamical core development: the test case by Untch and Hortal (2004) uses a three-dimensional global domain with a long simulation time, and the test cases by Arakawa and Konor (1996) were developed only for a simplified, vertically discrete model in pressure coordinates. Hence, we propose a new, idealised test case on a two-dimensional x–z Cartesian plane, based on the work of Arakawa and Konor (1996), to compare the accuracy of models using Lorenz or Charney–Phillips staggerings. The new test case intended to aid the development and intercomparison of modern, nonhydrostatic dynamical cores. The test completes quickly, using a coarse two-dimensional mesh specified in Cartesian coordinates. We compare test results between the fully compressible Euler model with a Lorenz staggering (section 4.3), and a variant of this same model that includes a new generalisation of the Charney–Phillips staggering for arbitrary meshes.

After describing the generalised Charney–Phillips formulation in section 5.1, we compare Lorenz and generalised Charney–Phillips model variants in section 5.2 using the standard mountain waves test case specified by Schär et al. (2002). Section 5.3 presents the new test case based on the original test specified by Arakawa and Konor (1996). We verify that the Lorenz computational mode is excited using the fully compressible Euler model with a Lorenz staggering, and we verify that the Charney–Phillips model variant is free of spurious grid-scale waves. Finally, we explore the sensitivity to mesh distortions using the generalised Charney–Phillips model variant.

5.1   Generalising the Charney–Phillips staggering for arbitrary meshes

The generalisation of the Lorenz staggering for arbitrary meshes is straightforward (Weller and Shahrokhi, 2014) but this is not true for the Charney–Phillips staggering, which is only suitable for structured meshes with cells stacked in columns. On a finite volume mesh, variables are ordinarily placed at cell centres or cell faces. In the Charney–Phillips staggering, the thermodynamic variable is placed at only those cell faces that lie on the vertical coordinate surfaces, and vertically-oriented faces have no thermodynamic information. This existing staggering is unsuitable for arbitrary finite volume meshes because faces can have any orientation.

A generalised Charney–Phillips staggering will be particularly relevant to atmospheric models that use vertical mesh refinement techniques. Mesh refinement has received growing attention in atmospheric modelling literature because it could enable atmospheric models to produce more accurate forecasts with less computation (Behrens, 2006; Jablonowski et al., 2009). While much of the literature concentrates on horizontal mesh refinement, some investigations have been made into vertical refinement on two-dimensional x–z Cartesian planes: Müller et al. (2013) have used conforming refinement of triangular meshes for simulating the standard rising bubble and density current test cases, Van Hooft et al. (2018) have used block-structured adaptive mesh refinement for direct numerical simulations of the atmospheric boundary layer, and Yamazaki and Satomura (2012) have used nonconforming block-refinement to better resolve the atmosphere immediately above idealised mountains.

According to Thuburn and Woollings (2005), the vertical discretisation used by Yamazaki and Satomura (2012) supports computational modes and instabilities, although these errors were not excited by the test cases performed by Yamazaki and Satomura (2012). The Charney–Phillips staggering is not susceptible to such errors, but we are not aware of any existing literature that combines mesh refinement with a Charney–Phillips staggering. By allowing for an arbitrary mesh, a generalised Charney–Phillips formulation should be suitable for any type of mesh, including conforming and non-conforming mesh refinement, terrain-following meshes, cut cell meshes and slanted cell meshes.

Generalised Charney–Phillips formulation

The generalised Charney–Phillips model is a new variant of the fully compressible Euler model with a Lorenz staggering, as documented by Weller and Shahrokhi (2014) and summarised in section 4.3. The model variant uses a newly-formulated generalisation of the Charney–Phillips staggering for arbitrary meshes. The primary difference between the Lorenz and Charney–Phillips formulations is their treatment of the prognostic thermodynamic variable: the gener-
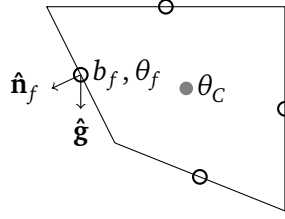
Figure 5.2: A quadrilateral cell with the prognostic thermodynamic variable b_f stored at face centres marked by open circles. b_f is calculated from the potential temperature θ_f such that $b_f = \theta_f \hat{\mathbf{g}} \cdot \hat{\mathbf{n}}_f$ where $\hat{\mathbf{n}}_f$ is the unit vector outward normal to face f, and $\hat{\mathbf{g}}$ is the unit vector of gravitational acceleration. The potential temperature at the cell centre, θ_C, is reconstructed from surrounding values of b_f using equation (5.2).

alised Charney–Phillips formulation stores the prognostic thermodynamic variable b_f at all cell faces such that $b_f = \theta_f \hat{\mathbf{g}} \cdot \hat{\mathbf{n}}_f$ where f is a face, θ_f is the potential temperature at the face, $\hat{\mathbf{g}}$ is the unit vector of gravitational acceleration and $\hat{\mathbf{n}}_f$ is the unit vector that is outward normal to the face. This arrangement is illustrated in figure 5.2. To transport the thermal field, first, potential temperature is transported in advective form using first-order time-stepping,

$$\theta_f^{n+1} = \theta_f^n - \Delta t \mathbf{u}_f \cdot \left(\nabla_c \theta_f^\ell \right)_F \tag{5.1}$$

where θ_f^{n+1} is the value of θ_f at the new time-step, θ_f^ℓ is the lagged value from the previous time-stepping iteration, $\mathbf{u}_f$ is the velocity, $(\cdot)_F$ denotes an interpolation from cell centres to faces, and ∇_c denotes a cell centre gradient (Weller and Shahrokhi, 2014). Next, b_f is calculated such that $b_f = \theta_f \hat{\mathbf{g}} \cdot \hat{\mathbf{n}}_f$. On a Cartesian mesh with no diagonal faces, b_f is zero for entirely vertical faces and $b_f = \theta_f$ for entirely horizontal faces.

Potential temperature at the cell centre is reconstructed from bordering faces,

$$\theta_C = \hat{\mathbf{g}} \cdot \left(\sum_{f \in c} \hat{\mathbf{n}}_f \mathbf{S}_f \right)^{-1} \cdot \sum_{f \in c} \mathbf{S}_f b_f \tag{5.2}$$

where θ_C is the reconstructed potential temperature. On a rectangular Cartesian mesh with no diagonal faces, θ_C is simply a linear interpolation from the face values immediately above and below the cell centre, and the generalised Charney–Phillips formulation reduces to the standard Charney–Phillips formulation.

Finally, θ_f is recalculated from b_f and θ_C,

$$\theta_f := |\hat{\mathbf{g}} \cdot \hat{\mathbf{n}}_f \theta_f| + \left(1 - |\hat{\mathbf{g}} \cdot \hat{\mathbf{n}}_f| \right) (\theta_C)_F . \tag{5.3}$$

This ensures that values of θ_f on vertical faces is calculated from nearby b_f values and is not retained across time-steps.

The generalised Charney–Phillips model variant makes two other modifications to the Lorenz model variant in order to simplify implementation: first, gravity waves are treated explicitly and, second, first-order Euler semi-implicit time-stepping is used with deferred correction of explicit terms (equation 5.1).

5.2   Schär mountain waves test

Chapter 4 assessed a finite volume model of the fully compressible Euler equations with a Lorenz staggering using a test case with a stratified atmosphere initially at rest above an isolated mountain. We now turn our attention to a transport-dominated test case that presents a challenge to the transport schemes within the dynamical model. As specified by Schär et al. (2002), the test prescribes flow over idealised terrain with small-scale and large-scale undulations that induce propagating and evanescent gravity waves. We use the test to compare results from the two finite volume model variants with Lorenz and generalised Charney–Phillips staggerings against the reference solution from Melvin et al. (2010).

Following Melvin et al. (2010), the domain is 300 km wide and 30 km high, and the mesh spacing is $\Delta x = 500$ m and $\Delta z^\star = 300$ m. The mountain profile has the same form as equation (4.5), but the mountain waves test has a lower peak mountain height of $h_0 = 250$ m. As in the resting atmosphere test (section 4.4), $a = 5$ km is the mountain half-width and $\lambda = 4$ km is the wavelength.

A uniform horizontal wind $(u, w) = (10, 0)\,\mathrm{m\,s^{-1}}$ is prescribed in the domain interior and at the inlet boundary. No normal flow is imposed at the top and bottom boundaries and the velocity field has a zero gradient outlet boundary condition.

The initial thermodynamic conditions have constant static stability with $N = 0.01\,\mathrm{s^{-1}}$ everywhere such that

$$\theta(z) = \theta_0 \exp\left(\frac{N^2}{g}z\right) \tag{5.4}$$

where the temperature at $z = 0$ is $\theta_0 = 288$ K. Potential temperature values are prescribed at the inlet and upper boundary using equation (5.4), and a zero gradient boundary condition is applied at the outlet. At the ground, fixed gradients are imposed by calculating the component of $\nabla\theta$ normal to each face using the vertical derivative of equation (5.4). For the Exner function of pressure, hydrostatic balance is prescribed on top and bottom boundaries and the inlet and outlet are zero normal gradient.

Sponge layers are added to the upper 10 km and leftmost 10 km at the inlet boundary to damp the reflection of waves. The damping term μ in the momentum equation (4.4a) is a

function adapted from Melvin et al. (2010) such that

$$\mu(x, z) = \mu_{\text{upper}} + \mu_{\text{inlet}} \tag{5.5a}$$

$$\mu_{\text{upper}}(z) = \begin{cases} \overline{\mu} \sin^2\left(\frac{\pi}{2} \frac{z - z_B}{H - z_B}\right) & \text{if } z \geq z_B, \\ 0 & \text{otherwise,} \end{cases} \tag{5.5b}$$

$$\mu_{\text{inlet}}(x) = \begin{cases} \overline{\mu} \sin^2\left(\frac{\pi}{2} \frac{x_I - x}{x_I - x_0}\right) & \text{if } x < x_I, \\ 0 & \text{otherwise,} \end{cases} \tag{5.5c}$$

where $\overline{\mu} = 1.2\,\text{s}^{-1}$ is the damping coefficient, $z_B = 20\,\text{km}$ is the bottom of the sponge layer, $H = 30\,\text{km}$ is the top of the domain, $x_0 = -150\,\text{km}$ is the leftmost limit of the domain and $x_I = -140\,\text{km}$ is the rightmost extent of the inlet sponge layer. The sponge layer is active only on entirely horizontal faces so that only vertical momentum is damped. Note that, while the domain itself is 30 km in height, for the purposes of generating basic terrain-following meshes, the domain height is set to 20 km because the sponge layer occupies the uppermost 10 km.

The test is integrated forward by five hours using a time-step of 8 s. At the end of the simulation, gravity waves are apparent in the contours of vertical velocity (figure 5.3). Results are presented for the Lorenz model variant, with momentum and potential temperature being transported using the linearUpwind scheme (figure 5.3a) and the cubicFit scheme (figure 5.3b), and for the generalised Charney–Phillips model variant (figure 5.3c), and all are in general agreement with the reference solution from Melvin et al. (2010), reproduced in figure 5.3d. All four results presented in figure 5.3 were obtained using the same basic terrain-following mesh.

Spurious distortions are visible in the vertical velocity contours using the Lorenz model variant and the linearUpwind transport scheme (figure 5.3a), and similar error structures have been found in previous studies that were attributed to numerical errors associated with basic terrain-following mesh distortions (Schär et al., 2002; Klemp et al., 2003). In agreement with these previous findings, we find that spurious gravity wave distortions can be avoided by switching from a basic terrain-following mesh to a slanted cell mesh or cut cell mesh (results not shown). We also find that spurious gravity wave distortions can be avoided by transporting momentum and potential temperature on a basic terrain-following mesh using the cubicFit scheme (figure 5.3b). Avoiding such spurious gravity waves distortions using either approach produces solutions that closely match the reference solution (figure 5.3d). Given these results, we can attribute spurious gravity wave distortions to transport scheme errors associated with flow that is misaligned with mesh layers. Unlike the results obtained by Shaw and Weller (2016) that used an older formulation of the cubicFit scheme, potential temperature errors are negligible for all types of mesh when using the most recent formulation of the cubicFit scheme documented in chapter 2.

(a) Lorenz staggering, linearUpwind transport

(b) Lorenz staggering, cubicFit transport

(c) Generalised Charney–Phillips staggering

(d) Reference solution by Melvin et al. (2010)
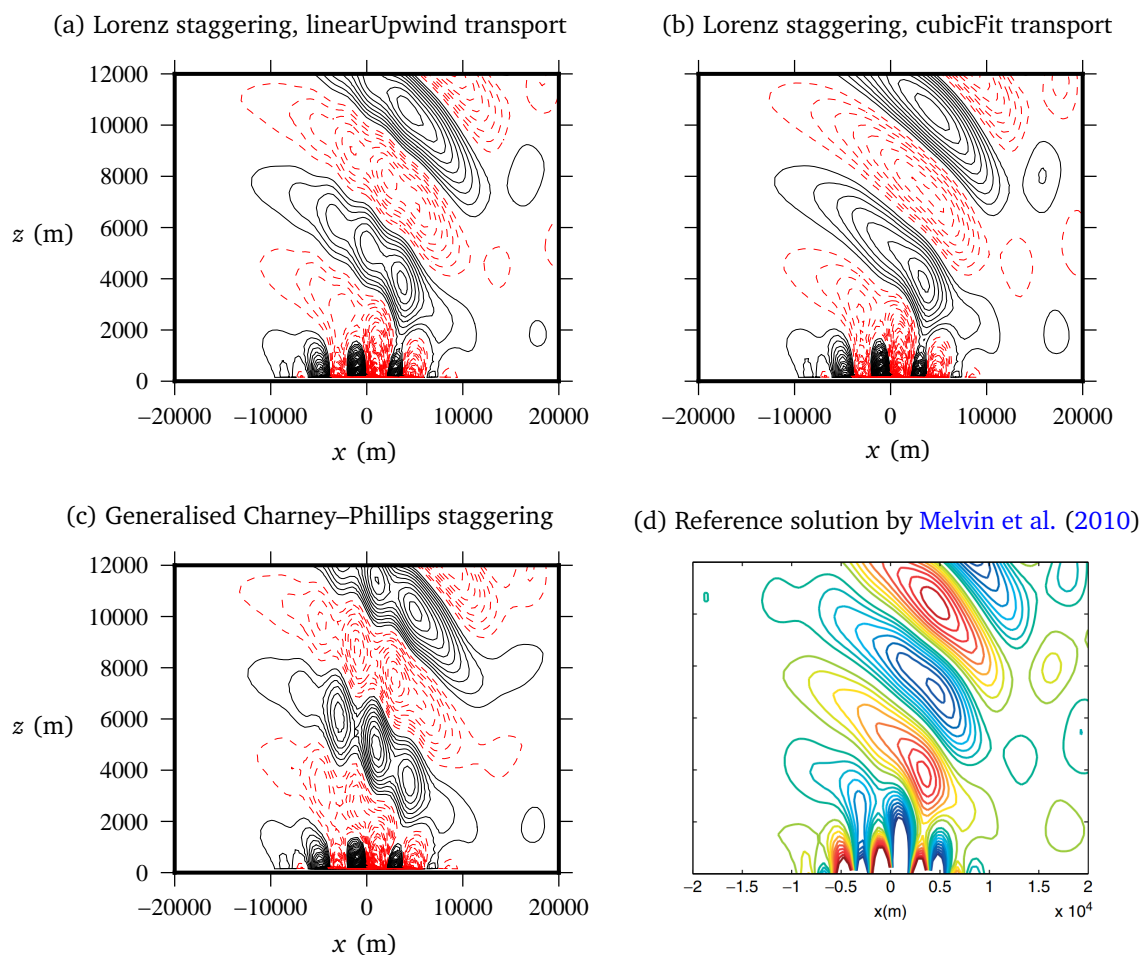
Figure 5.3: Vertical velocities at the end of integration of the Schär mountain waves test case. Results obtained using a basic terrain-following mesh and Lorenz staggering, with potential temperature and momentum transported by (a) the linearUpwind scheme and (b) the cubicFit scheme, and (c) using a basic terrain-following mesh and generalised Charney–Phillips staggering. For comparison, (d) provides a reference solution obtained with a mass-conserving semi-implicit semi-Lagrangian model (Melvin et al., 2010). Contours are plotted every $0.05\,\mathrm{m\,s^{-1}}$. In figures (a), (b) and (c), ascending velocities are marked by solid black lines and descending velocities are marked by dashed red lines. Only the lowest 12 km in the central region of the domain is shown. The entire domain is 300 km wide and 30 km high.

As seen in figure 5.3c, the generalised Charney–Phillips model variant produces gravity waves with spurious distorted structures similar to those obtained using the Lorenz model variant with the linearUpwind scheme (figure 5.3a). In addition, as evidenced by the density of vertical velocity contour lines in figure 5.3c, the generalised Charney–Phillips model variant produces gravity wave amplitudes that are too large compared to the reference solution.

In summary, all solutions obtained here are in general agreement with the reference solution from Melvin et al. (2010). However, on the basic terrain-following mesh, numerical errors lead to spurious gravity wave distortions using the generalised Charney–Phillips model variant and using the Lorenz model variant with momentum and potential temperature transported by the linearUpwind scheme. Using the more accurate cubicFit scheme in the Lorenz model variant produces the correct solution on the basic terrain-following mesh that is free from any spurious gravity wave distortions.

Knowing that an improved transport scheme was responsible for the improved gravity wave solution using the Lorenz model variant, we conjecture that the generalised Charney–Phillips model variant produces less accurate results because the model uses a transport scheme that is insufficiently accurate. In the next section we perform a further comparison between Lorenz and generalised Charney–Phillips model variants using a new test case to excite the Lorenz computational mode.

5.3 A two-dimensional standing waves test case

Having verified that the generalised Charney–Phillips formulation produces a reasonable mountain waves solution, we must also verify that the formulation is free from the Lorenz computational mode. Since existing tests are not well-suited for nonhydrostatic model evaluation, we design a new, two-dimensional standing waves test case, based on the original specification by Arakawa and Konor (1996). Results are compared between Lorenz and generalised Charney–Phillips model variants. To explore the applicability of the generalised Charney–Phillips formulation to arbitrary vertical meshes, we also compare results between distorted and undistorted meshes.

The domain is 30 km high and 600 km wide between the outermost faces, and the mesh spacing is $\Delta x = 10$ km and $\Delta z = 1$ km. The lower boundary is flat with no mountain profile. The upper and lower boundaries are no normal flow, and the domain is horizontally periodic.

The initial potential temperature profile is the sum of a stably-stratified profile and a grid-scale perturbation near the ground. The stably-stratified profile has $\theta(z = 0) = 250$ K and a constant static stability with Brunt-Väisälä frequency $N = 0.02$ s^{-1}. The potential temperature

perturbation θ' is defined as

$$\theta' = \begin{cases} S\theta'_0 \sin(\frac{2\pi x}{\lambda}) & \text{if } |x| \leq \frac{\lambda}{2}, \\ 0 & \text{otherwise,} \end{cases} \tag{5.6a}$$

where S is given by

$$S = \begin{cases} -1 & \text{if } 1\,\text{km} \leq z < 2\,\text{km}, \\ 1 & \text{if } 2\,\text{km} \leq z < 3\,\text{km}, \\ 0 & \text{otherwise,} \end{cases} \tag{5.6b}$$

with the maximum amplitude $\theta'_0 = 0.5\,\text{K}$ and the wavelength $\lambda = 100\,\text{km}$. Using a Lorenz staggering, this arrangement produces grid-scale waves in the central region of the domain in two adjacent layers near the ground (figure 5.4a). Using a generalised Charney–Phillips staggering, the perturbation is non-zero at the lowest two interior mesh layers above the lower boundary (not shown). The definition given by equation (5.6) ensures that the potential temperature perturbation integrated over the domain is zero. Using the Lorenz model variant, potential temperature is transported using the linearUpwind scheme. Using the Charney–Phillips model variant, potential temperature is transported in advective form (equation 5.1).

At the upper and lower boundaries zero gradients are imposed on the potential temperature field for the Lorenz model variant; for the Charney–Phillips model variant, fixed potential temperature values are prescribed using equation 5.4. The Exner function of pressure is calculated so that it is in discrete hydrostatic balance with the perturbed potential temperature field.

A sponge layer is added to the upper 10 km. The damping function is given by

$$\mu(z) = \begin{cases} \overline{\mu} \sin^2\left(\frac{\pi}{2} \frac{z-z_B}{H-z_B}\right) & \text{if } z \geq z_B, \\ 0 & \text{otherwise,} \end{cases} \tag{5.7}$$

where $\overline{\mu} = 1.2\,\text{s}^{-1}$ is the damping coefficient, $z_B = 20\,\text{km}$ is the bottom of the sponge layer and $H = 30\,\text{km}$ is the top of the domain. The sponge layer is active only on entirely horizontal faces so that only vertical momentum is damped.

The test is integrated forward by 48 hours using a time-step of $\Delta t = 25\,\text{s}$. The initial potential temperature perturbation generates gravity waves that spread rapidly through the domain. In addition to these gravity waves, using the Lorenz model variant, a grid-scale standing wave slowly spreads vertically, occupying the entire depth of the atmosphere (figure 5.4b), and the initial perturbation, though weakened in amplitude, persists throughout the duration of the simulation. No standing waves are produced by the generalised Charney–Phillips model variant (figure 5.4c). When the mesh is refined vertically such that $\Delta z = 500\,\text{m}$ then the initial potential
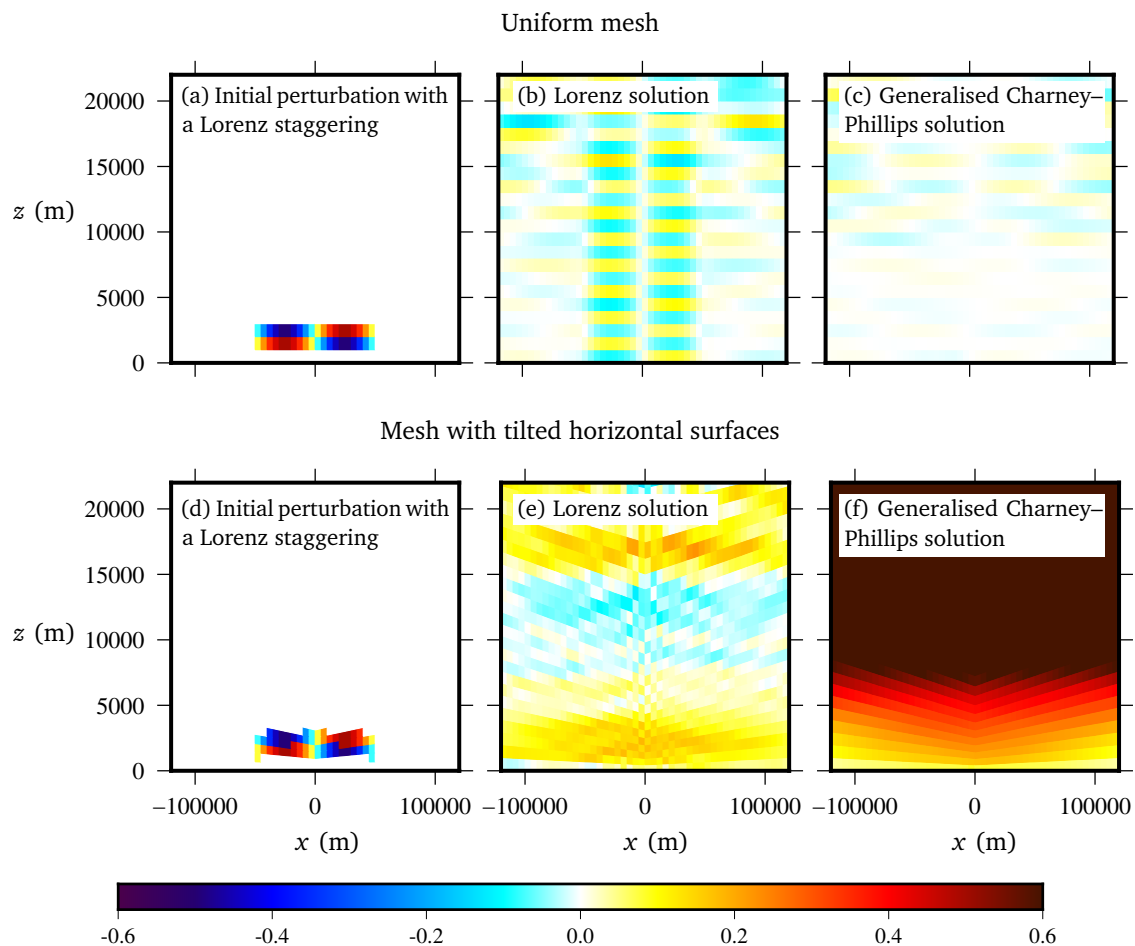
Figure 5.4: Differences in potential temperature for the standing waves test case. On the uniform mesh and the horizontally tilted mesh, (a, d) a grid-scale potential temperature perturbation near the surface is added to an initial, stably-stratified profile; the difference between the initial, unperturbed, stably-stratified potential temperature profile and the final solution are shown using (b, e) the Lorenz model variant, and (c, f) the generalised Charney–Phillips model variant. The largest differences in potential temperature on the horizontally tilted mesh using the generalised Charney–Phillips model variant is about 1.5 K, which is not representable by the colour scale used here. Only the lowest 22 km in the central region of the domain is shown. The entire domain is 600 km wide and 30 km high.
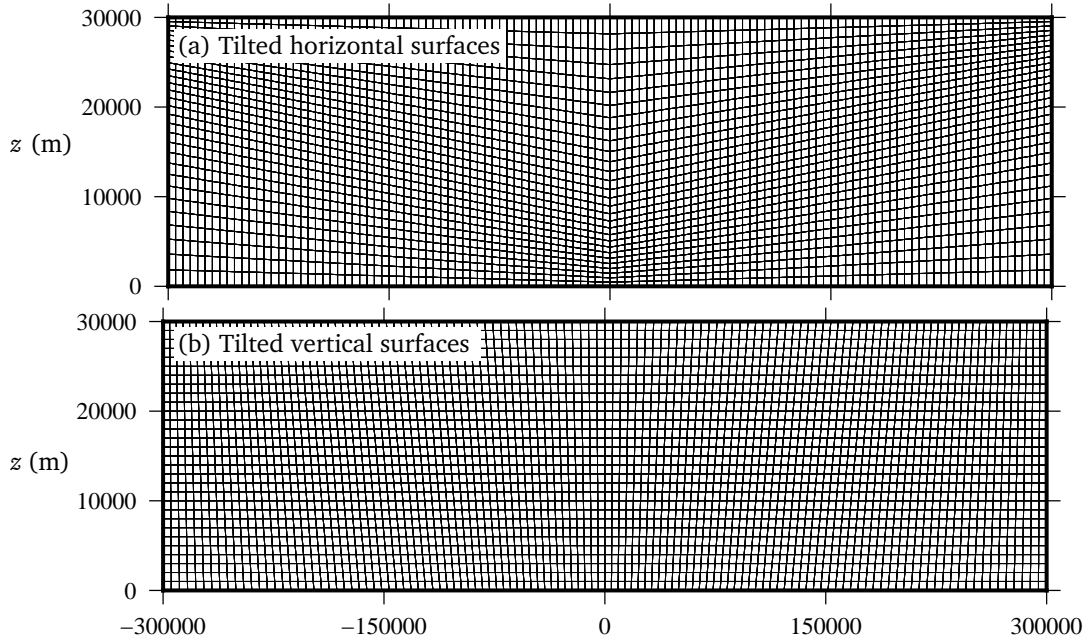
Figure 5.5: Distorted meshes used for the standing waves test with (a) horizontally tilted and (b) vertically tilted surfaces.

temperature perturbation is no longer at the grid scale, occupying four mesh layers rather than two. On this refined mesh, neither the Lorenz model variant nor the generalised Charney–Phillips model variant produce any standing waves (not shown). Since they are produced only by an initial grid-scale potential temperature perturbation using the Lorenz model variant, we conclude that the grid-scale standing waves are a spurious feature excited by the Lorenz computational mode.

To assess the suitability of the generalised Charney–Phillips formulation for arbitrary vertical meshes, we perform the same standing waves test case on rectangular domains with tilted horizontal surfaces (figure 5.5a) and tilted vertical surfaces (figure 5.5b). To allow for periodic lateral boundaries, each mesh is split into left and right blocks of equal size, with the right block mirroring the left. For the left-hand block of the horizontally tilted mesh, horizontal surfaces are distorted such that the ratio of minimum and maximum vertical edge lengths at $x = -300$ km is $1 : 16$, and the ratio at $x = 0$ km is $16 : 1$. The vertically tilted mesh is constructed in a similar manner, with a ratio of minimum and maximum horizontal edge lengths at $z = 0$ km of $81 : 100$, and a ratio at $z = 30$ km of $100 : 81$.

Using the horizontally tilted mesh, large-scale responses are produced by both the Lorenz model variant (figure 5.4e) and generalised Charney–Phillips model variant (figure 5.4f), and these large-scale responses are very different from the small-scale gravity waves produced by
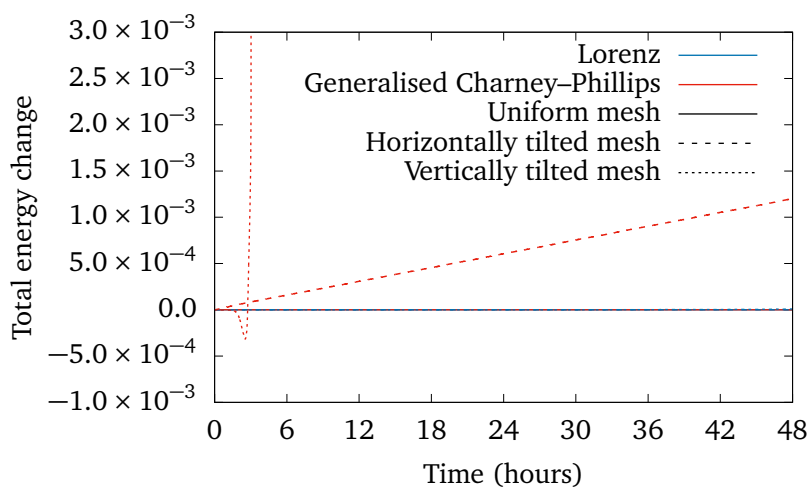
Figure 5.6: Total normalised energy changes for the standing waves test case using Lorenz and generalised Charney–Phillips model variants on a uniform mesh, a horizontally tilted mesh and a vertical tilted mesh. Energy changes are negligible using the Lorenz model variant on all meshes, and using the generalised Charney–Phillips model variant on the uniform mesh.

the Lorenz and generalised Charney–Phillips models using the uniform mesh (figures 5.4b and 5.4c respectively). Using the Lorenz model variant and the horizontally tilted mesh, a vertical tripole structure is seen in the final potential temperature field, with some grid-scale features visible near the centre of the domain throughout the depth of the atmosphere (figure 5.4e). A similar solution is produced by the Lorenz model variant using the vertically tilted mesh (not shown). In contrast to the solutions produced by the Lorenz model variant, using the generalised Charney–Phillips model variant and the horizontally tilted mesh produces a solution that is everywhere too warm (figure 5.4f). Using the generalised Charney–Phillips model variant and the vertically tilted mesh, the atmosphere warms rapidly before the model becomes unstable after about 2.5 hours. Note that, since the initial potential temperature perturbation is defined in Cartesian coorinates and not relative to model layers, the discrete initial potential temperature perturbation differs slightly between the uniform mesh (figure 5.4a), horizontally tilted mesh (figure 5.4d) and vertically tilted mesh (not shown). However, we do not expect this slight initial difference to result in such dramatically different solutions.

To better examine the thermal errors produced by the generalised Charney–Phillips model variant, we calculate total energy change over time, normalised by the initial total energy, where the total energy is the sum of the kinetic energy, potential energy and internal energy (figure 5.6). Energy changes are negligible using the Lorenz model variant on all meshes, and using the generalised Charney–Phillips model variant on the uniform mesh. Using the

generalised Charney–Phillips model and the horizontally tilted mesh, total energy increases linearly with time, which corresponds with the spurious warm atmosphere seen in figure 5.4f. Using the vertically tilted mesh, a rapid increase in total energy is observed at about $t = 2.5$ hours, just before the model becomes unstable. Since the generalised Charney–Phillips model variant transports potential temperature using an advective-form scheme and not a flux-form scheme, it is likely that the observed energy changes and associated potential temperature errors are due to a lack of conservation on distorted meshes.

Here we have presented a new standing waves test case that has been used to clearly excite the Lorenz computational mode, and we have demonstrated that the generalised Charney–Phillips formulation is free from the Lorenz computational mode on a uniform mesh. The generalised Charney–Phillips model variant suffers from inaccurate or unstable solutions on distorted meshes, but we expect that a more accurate transport scheme could avoid such a severe lack of conservation and improve solutions on distorted meshes.

6 Conclusions and future work

Atmospheric models are using increasingly fine horizontal mesh spacings that resolve steep slopes in terrain resulting in highly-distorted meshes and increased numerical errors. This thesis makes four contributions to reduce numerical errors for flows over steep slopes. First, we 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 while maintaining second-order convergence over steep terrain on highly-distorted, arbitrary meshes. The scheme has a low computational cost at runtime, requiring only m multiplies per face per time-stage using a stencil with m cells. Stability condition calculations are pre-computed during model initialisation since they depend upon the mesh geometry only. A new transport test case with a terrain-following velocity field was formulated which reveals that numerical transport errors are primarily due to misalignment of the velocity field with mesh layers and not simply mesh distortions. In all tests, compared to the multidimensional linear upwind scheme, the cubicFit scheme is more stable and more accurate. The cubicFit transport scheme is second-order convergent on two-dimensional meshes over steeply sloping terrain, cubed sphere meshes and hexagonal-icosahedral meshes, irrespective of the velocity field or mesh distortions.

Second, a high-order multidimensional method-of-lines transport scheme, highOrderFit, was developed. The highOrderFit scheme uses k-exact polynomial reconstructions that are obtained by calculating high-order volume and surface moments exactly. All computationally expensive reconstruction calculations depend upon the mesh geometry alone. During integration, the highOrderFit scheme requires only m multiplies per face per time-stage using a stencil of m cells, meaning that, at runtime, the highOrderFit scheme has the same computational cost and storage requirements as the cubicFit scheme. Transport tests demonstrate that the highOrderFit scheme achieves at least third-order convergence in the domain interior irrespective of the velocity field or mesh distortions.

The highOrderFit transport scheme offers a promising route to obtaining high-order conver-

gence with a low computational cost, but we are aware of three improvements that could be investigated:

1. By selectively removing high-order polynomial terms for highly-distorted stencils, the cu-bicFit scheme achieves stable results where, on the same mesh, the highOrderFit scheme is unstable. We expect that a similar stabilisation procedure could be included in the highOrderFit scheme to obtain stable results on more highly-distorted meshes. This sta-bilisation procedure should also permit stable transport over arbitrary lower boundaries and, to further improve accuracy near boundaries, high-order boundary conditions might be derived following existing approaches (Devendran et al., 2017; Schwartz et al., 2015).

2. The highOrderFit scheme achieves high-order convergence that makes it more accurate than the cubicFit scheme at finer mesh spacings but, at coarser mesh spacings, the cubic-Fit scheme is more accurate. For both transport schemes, the multipliers that appear in the weighted least-squares fit are used to ensure that the polynomial fits the upwind and downwind cells almost exactly, and this is necessary to achieve stability on highly-distorted meshes. On less distorted meshes, however, more accurate results can be obtained by mod-ifying the multiplier values so that the polynomial has a less exact fit through the upwind and downwind cells, but has better least-squares fit to the stencil overall. Devendran et al. (2017) note that these multipliers introduce extra degrees of freedom into the system, and a strategy for optimising these values is the subject of future work. We hope that if such a strategy were included in the highOrderFit scheme then more accurate results could be obtained on coarser meshes.

3. The highOrderFit scheme has only been tested on Cartesian planes, and further research is necessary to extend the scheme to spherical geometry. It is not clear that a projection from the sphere onto a local tangent plane, as used by Sjögreen (2012), will achieve high-order convergence. Instead, it is more likely that high-order convergence could be achieved by taking the method by Tuzikov et al. (2003) for calculating exact polyhedral surface and volume moments and extending it to spherical polyhedra.

Third, we proposed a new type of mesh, the slanted cell mesh, for representing the atmo-sphere above steeply sloping terrain. The slanted cell mesh is designed to avoid severe mesh distortions associated with terrain-following meshes, and to avoid severe time-step constraints associated with arbitrarily small cut cells. In a test of a stratified atmosphere at rest, spurious circulations were reduced by switching from the highly-distorted basic terrain-following mesh to the more uniform slanted cell mesh. A new test case was formulated to challenge transport schemes over a steeply sloping lower boundary. Unlike the multidimensional linear upwind

scheme, the cubicFit scheme is numerically stable over these very steep slopes. The test reveals that the slanted cell mesh permits longer time-steps than those permitted by cut cell meshes, since slanted cells are always long in the direction of flow.

In this thesis, all tests that represent terrain with x–z meshes have used only uniform vertical mesh spacing. Operational atmospheric models have non-uniform vertical mesh spacing, using finer mesh spacing near the ground to resolve boundary layer processes. Fine vertical mesh spacing near the ground is also desirable for resolving diurnal flows along mountain slopes, which are typically 1 m to 20 m deep (Chow et al., 2013, p. 39). Since we have established that the numerical accuracy of a transport scheme depends primarily on the alignment of flow with the mesh (section 2.3), we might expect a terrain-following mesh to be better-suited than a cut cell mesh for representing slope flows. The tests presented here demonstrate that transport over mountain slopes is more accurate using terrain-following meshes (section 4.2), but that a more accurate balance between the pressure gradient and gravity is achieved using cut cell meshes or slanted cell meshes (section 4.4). Hence, future work might seek an improved mesh that blends the best features of both terrain-following and slanted cell meshes.

Finally, a two-dimensional test case was developed to excite the Lorenz computational mode. The test is based on the original specification by Arakawa and Konor (1996) that was developed for a simplified, vertically-discrete model in pressure coordinates. Test results were compared using two models of the fully compressible Euler equations: one variant having a Lorenz staggering, and the other model variant using a newly-formulated Charney–Phillips staggering generalised for arbitrary meshes. The test case verifies that the generalised Charney–Phillips model variant is free from the Lorenz computational mode, and we hope that the new test case might aid in the development and intercomparison of future dynamical cores.

The research presented in this thesis creates new opportunities for accurate atmospheric transport with low computational cost and improved numerical balance between the pressure gradient and gravity, and we have provided a series of new, idealised test cases for evaluating dynamical cores. In combination, the slanted cell mesh, the highOrderFit transport scheme and numerical stabilisation techniques from the cubicFit scheme offer a route to more accurate numerical simulations of the atmosphere in the vicinity of steeply sloping terrain.

Appendices

A 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 2.4, 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 2.4 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,

$$\mathbf{u}_f \cdot \mathbf{S}_f = \sum_{e \in f} \mathbf{e} \cdot \mathbf{x}_e \Psi(e) \tag{A.1}$$

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 2.4.

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,

$$A_f = \frac{d}{2}|r_2^2 - r_1^2| \tag{A.2}$$

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 (Van Brummelen, 2013). 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,

$$\hat{\mathbf{r}} = \frac{\sum_{t \in f} \Omega_t \left(\mathbf{x}_{t,1} + \mathbf{x}_{t,2} + \tilde{\mathbf{c}}_f \right)}{|\sum_{t \in f} \Omega_t \left(\mathbf{x}_{t,1} + \mathbf{x}_{t,2} + \tilde{\mathbf{c}}_f \right)|} \tag{A.3}$$

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,

$$r = \frac{\sum_{t \in f} \Omega_t \left(|\mathbf{x}_{t,1}| + |\mathbf{x}_{t,2}| \right)/2}{\Omega_f} \tag{A.4}$$

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

$$\Omega_f = \sum_{t \in f} \Omega_t \ . \tag{A.5}$$

We use equations (A.3) and (A.4) to calculate the centre $\mathbf{c}_f$ of the face f,

$$\mathbf{c}_f = r\,\hat{\mathbf{r}} \tag{A.6}$$

The area vector $\mathbf{S}_f$ of the surface face f is the sum of the spherical triangle areas (Van Brummelen, 2013),

$$\mathbf{S}_f = r^2 \Omega_f \hat{\mathbf{r}} \ . \tag{A.7}$$

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

$$\frac{\left(\mathbf{S}_f \cdot \mathbf{c}_f\right)^2}{|\mathbf{S}_f|^2 |\mathbf{c}_f|^2} > 0 \ . \tag{A.8}$$

Let $\mathscr{F}$ be the set of faces satisfying equation (A.8). Then, the cell volume $\mathscr{V}_c$ is

$$\mathscr{V}_c = \frac{1}{3} \sum_{f \in \mathscr{F}} \mathbf{S}_f \cdot \mathbf{c}_f \tag{A.9}$$

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 A.1). 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.

Bibliography

Adcroft, A., 2013: Representation of topography by porous barriers and objective interpolation of topographic data. *Ocean Model.*, **67**, 13–27, doi:10.1016/j.ocemod.2013.03.002.

Adcroft, A., C. Hill, and J. Marshall, 1997: Representation of topography by shaved cells in a height coordinate ocean model. *Mon. Wea. Rev.*, **125 (9)**, 2293–2315, doi:10.1175/1520-0493(1997)125<2293:ROTBSC>2.0.CO;2.

Arakawa, A., and C. S. Konor, 1996: Vertical differencing of the primitive equations based on the Charney-Phillips grid in hybrid σ–p vertical coordinates. *Mon. Wea. Rev.*, **124 (3)**, 511–528, doi:10.1175/1520-0493(1996)124%3C0511:VDOTPE%3E2.0.CO;2.

Arakawa, A., and S. Moorthi, 1988: Baroclinic instability in vertically discrete systems. *J. Atmos. Sci.*, **45 (11)**, 1688–1708, doi:10.1175/1520-0469(1988)045<1688:BIIVDS>2.0.CO;2.

Atkins, H. L., and C.-W. Shu, 1998: Quadrature-free implementation of discontinuous Galerkin method for hyperbolic equations. *AIAA J.*, **36 (5)**, 775–782, doi:10.2514/2.436.

Baldauf, M., 2008: Stability analysis for linear discretisations of the advection equation with Runge–Kutta time integration. *J. Comp. Phys.*, **227 (13)**, 6638–6659, doi:10.1016/j.jcp.2008.03.025.

Barth, T. J., 1995: Aspects of unstructured grids and finite-volume solvers for the Euler and Navier–Stokes equations. NASA, 1994–05 Lecture Series, von Karman Institute for Fluid Dynamics.

Behrens, J., 2006: *Adaptive Atmospheric Modeling*. Springer, doi:10.1007/3-540-33383-5.

Bell, M. J., and A. A. White, 2017: Analytical approximations to spurious short-wave baroclinic instabilities in ocean models. *Ocean Model.*, **118**, 31–40, doi:10.1016/j.ocemod.2017.08.001.

Burchard, H., and O. Petersen, 1997: Hybridization between σ- and z-co-ordinates for

improving the internal pressure gradient calculation in marine models with steep bottom slopes. *Int. J. Numer. Meth. Fluids*, **25 (9)**, 1003–1023, doi:10.1002/(SICI)1097-0363(19971115)25:9<1003::AID-FLD600>3.0.CO;2-E.

CFD Direct, 2018a: OpenFOAM CFD software. http://openfoam.org/.

CFD Direct, 2018b: OpenFOAM user guide: Numerical schemes. http://cfd.direct/openfoam/user-guide/fvschemes/.

Chang, E. K., 1992: Resonating neutral modes of the Eady model. *J. Atmos. Sci.*, **49 (24)**, 2452–2463, doi:10.1175/1520-0469(1992)049<2452:RNMOTE>2.0.CO;2.

Charney, J. G., and N. A. Phillips, 1953: Numerical integration of the quasi-geostrophic equations for barotropic and simple baroclinic flows. *J. Meteor.*, **10 (2)**, 71–99, doi:10.1175/1520-0469(1953)010%3C0071:NIOTQG%3E2.0.CO;2.

Chen, Y., H. Weller, S. Pring, and J. Shaw, 2017: Comparison of dimensionally-split and multi-dimensional atmospheric transport schemes for long time-steps. *Quart. J. Roy. Meteor. Soc.*, **143 (708)**, 2764–2779, doi:10.1002/qj.3125.

Chow, F. K., S. F. De Wekker, and B. J. Snyder, 2013: *Mountain weather research and forecasting: recent progress and current challenges*. Springer, doi:10.1007/978-94-007-4098-3.

Clain, S., S. Diot, and R. Loubere, 2011: A high-order finite volume method for systems of conservation laws—Multi-dimensional Optimal Order Detection (MOOD). *J. Comp. Phys.*, **230 (10)**, 4028–4050, doi:10.1016/j.jcp.2011.02.026.

Costa, R., S. Clain, and G. J. Machado, 2017: A sixth-order finite volume scheme for the steady-state incompressible Stokes equations on staggered unstructured meshes. *J. Comp. Phys.*, **349**, 501–527, doi:10.1016/j.jcp.2017.07.047.

Cueto-Felgueroso, L., I. Colominas, J. Fe, F. Navarrina, and M. Casteleiro, 2006: High-order finite volume schemes on unstructured grids using moving least-squares reconstruction. Application to shallow water dynamics. *Int. J. Numer. Meth. Engng.*, **65 (3)**, 295–331, doi:10.1002/nme.1442.

Cueto-Felgueroso, L., I. Colominas, X. Nogueira, F. Navarrina, and M. Casteleiro, 2007: Finite volume solvers and moving least-squares approximations for the compressible Navier–Stokes equations on unstructured grids. **196 (45)**, 4712–4736, doi:10.1016/j.cma.2007.06.003.

Cullen, M., T. Davies, M. Mawson, J. James, S. Coulter, and A. Malcolm, 1997: An overview of numerical methods for the next generation UK NWP and climate model. *Atmos.-Ocean*, **35 (Supplement 1)**, 425–444, doi:10.1080/07055900.1997.9687359.

Davies, T., M. Cullen, A. Malcolm, M. Mawson, A. Staniforth, A. White, and N. Wood, 2005:

A new dynamical core for the Met Office's global and regional modelling of the atmosphere. *Quart. J. Roy. Meteor. Soc.*, **131 (608)**, 1759–1782, doi:10.1256/qj.04.101.

Devendran, D., D. Graves, H. Johansen, and T. Ligocki, 2017: A fourth-order Cartesian grid embedded boundary method for Poisson's equation. *Comm. App. Math. Comp. Sci.*, **12 (1)**, 51–79, doi:10.2140/camcos.2017.12.51.

Diot, S., R. Loubere, and S. Clain, 2013: The Multidimensional Optimal Order Detection method in the three-dimensional case: very high-order finite volume method for hyperbolic systems. *Int. J. Numer. Meth. Fluids*, **73 (4)**, 362–392, doi:10.1002/fld.3804.

Donner, L. J., and Coauthors, 2011: The dynamical core, physical parameterizations, and basic simulation characteristics of the atmospheric component AM3 of the GFDL global coupled model CM3. *J. Clim.*, **24 (13)**, 3484–3519, doi:10.1175/2011JCLI3955.1.

Dumbser, M., M. Käser, V. A. Titarev, and E. F. Toro, 2007: Quadrature-free non-oscillatory finite volume schemes on unstructured meshes for nonlinear hyperbolic systems. *J. Comp. Phys.*, **226 (1)**, 204–243, doi:10.1016/j.jcp.2007.04.004.

Durran, D. R., 2013: *Numerical methods for wave equations in geophysical fluid dynamics*, Vol. 32. Springer Science & Business Media, doi:10.1007/978-1-4419-6412-0.

Eckermann, S., 2009: Hybrid $\sigma-p$ coordinate choices for a global model. *Mon. Wea. Rev.*, **137 (1)**, 224–245, doi:10.1175/2008MWR2537.1.

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

Fast, J. D., 2003: Forecasts of valley circulations using the terrain-following and step-mountain vertical coordinates in the Meso-Eta model. *Wea. Forecasting*, **18**, 1192–1206, doi:10.1175/1520-0434(2003)018<1192:FOVCUT>2.0.CO;2.

Ford, R., M. Glover, D. Ham, C. Maynard, S. Pickles, and G. Riley, 2013: GungHo phase 1 computational science recommendations. Tech. Rep. 587, Met Office. Available at http://www.metoffice.gov.uk/media/pdf/8/o/FRTR587Tagged.pdf.

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

Giraldo, F. X., and M. Restelli, 2008: A study of spectral element and discontinuous Galerkin methods for the Navier–Stokes equations in nonhydrostatic mesoscale atmospheric modeling: Equation sets and test cases. *J. Comp. Phys.*, **227 (8)**, 3849–3877,

doi:10.1016/j.jcp.2007.12.009.

Girard, C., and Coauthors, 2014: Staggered vertical discretization of the Canadian Environmental Multiscale (GEM) model using a coordinate of the log-hydrostatic-pressure type. *Mon. Wea. Rev.*, **142 (3)**, 1183–1196, doi:10.1175/MWR-D-13-00255.1.

Good, B., A. Gadian, S.-J. Lock, and A. Ross, 2014: Performance of the cut-cell method of representing orography in idealized simulations. *Atmos. Sci. Lett.*, **15 (1)**, 44–49, doi:10.1002/asl2.465.

Guerra, J. E., and P. A. Ullrich, 2016: A high-order staggered finite-element vertical discretization for non-hydrostatic atmospheric models. *Geosci. Model Dev.*, **9 (5)**, 2007, doi:10.5194/gmd-9-2007-2016.

Guo, W., R. D. Nair, and J.-M. Qiu, 2014: A conservative semi-Lagrangian discontinuous Galerkin scheme on the cubed sphere. *Mon. Wea. Rev.*, **142 (1)**, 457–475, doi:10.1175/MWR-D-13-00048.1.

Halliwell, G. R., 2004: Evaluation of vertical coordinate and vertical mixing algorithms in the HYbrid-Coordinate Ocean Model (HYCOM). *Ocean Model.*, **7 (3)**, 285–322, doi:10.1016/j.ocemod.2003.10.002.

Heikes, R., and D. A. Randall, 1995a: Numerical integration of the shallow-water equations on a twisted icosahedral grid. Part I: Basic design and results of tests. *Mon. Wea. Rev.*, **123 (6)**, 1862–1880, doi:10.1175/1520-0493(1995)123<1862:NIOTSW>2.0.CO;2.

Heikes, R., and D. A. Randall, 1995b: Numerical integration of the shallow-water equations on a twisted icosahedral grid. Part II: A detailed description of the grid and an analysis of numerical accuracy. *Mon. Wea. Rev.*, **123 (6)**, 1881–1887, doi:10.1175/1520-0493(1995)123<1881:NIOTSW>2.0.CO;2.

Hoinka, K. P., and G. Zängl, 2004: The influence of the vertical coordinate on simulations of a PV streamer crossing the Alps. *Mon. Wea. Rev.*, **132 (7)**, 1860–1867, doi:10.1175/1520-0493(2004)132<1860:TIOTVC>2.0.CO;2.

Holdaway, D., J. Thuburn, and N. Wood, 2008: On the relation between order of accuracy, convergence rate and spectral slope for linear numerical methods applied to multiscale problems. *Int. J. Numer. Meth. Fluids*, **56 (8)**, 1297–1303, doi:10.1002/fld.1644.

Holdaway, D., J. Thuburn, and N. Wood, 2013: Comparison of Lorenz and Charney–Phillips vertical discretisations for dynamics–boundary layer coupling. Part I: Steady states. *Quart. J. Roy. Meteor. Soc.*, **139 (673)**, 1073–1086, doi:10.1002/qj.2016.

Hollingsworth, A., 1995: *A Spurious Mode in the 'Lorenz' Arrangement of f and T which Does Not*

Exist in the 'Charney–Phillips' Arrangement. 211, European Centre for Medium-Range Weather Forecasts.

Hundsdorfer, W., and J. G. Verwer, 2013: *Numerical solution of time-dependent advection-diffusion-reaction equations*, Vol. 33. Springer-Verlag Berlin Heidelberg, 472 pp., doi:10.1007/978-3-662-09017-6.

Ii, S., and F. Xiao, 2009: High order multi-moment constrained finite volume method. Part I: Basic formulation. *J. Comp. Phys.*, **228 (10)**, 3669–3707, doi:10.1016/j.jcp.2009.02.009.

Iske, A., and M. Käser, 2004: Conservative semi-Lagrangian advection on adaptive unstructured meshes. *Numer. Methods. Partial Differ. Equ.*, **20 (3)**, 388–411, doi:10.1002/num.10100.

Jablonowski, C., R. C. Oehmke, and Q. F. Stout, 2009: Block-structured adaptive meshes and reduced grids for atmospheric general circulation models. *Philos. Trans. Roy. Soc. A*, **367 (1907)**, 4497–4522, doi:10.1098/rsta.2009.0150.

Jähn, M., O. Knoth, M. König, and U. Vogelsberg, 2015: ASAM v2.7: a compressible atmospheric model with a Cartesian cut cell approach. *Geosci. Model Dev.*, **8 (2)**, 317–340, doi:10.5194/gmd-8-317-2015.

Jebens, S., O. Knoth, and R. Weiner, 2011: Partially implicit peer methods for the compressible Euler equations. *J. Comp. Phys.*, **230 (12)**, 4955–4974, doi:10.1016/j.jcp.2011.03.015.

Kent, J., P. A. Ullrich, and C. Jablonowski, 2014: Dynamical core model intercomparison project: Tracer transport test cases. *Quart. J. Roy. Meteor. Soc.*, **140 (681)**, 1279–1293, doi:10.1002/qj.2208.

Klein, R., K. Bates, and N. Nikiforakis, 2009: Well-balanced compressible cut-cell simulation of atmospheric flow. *Philos. Trans. Roy. Soc. A*, **367 (1907)**, 4559–4575, doi:10.1098/rsta.2009.0174.

Klemp, J. B., 2011: A terrain-following coordinate with smoothed coordinate surfaces. *Mon. Wea. Rev.*, **139 (7)**, 2163–2169, doi:10.1175/MWR-D-10-05046.1.

Klemp, J. B., W. C. Skamarock, and O. Fuhrer, 2003: Numerical consistency of metric terms in terrain-following coordinates. *Mon. Wea. Rev.*, **131 (7)**, 1229–1239, doi:10.1175/1520-0493(2003)131<1229:NCOMTI>2.0.CO;2.

Kühnlein, C., and P. K. Smolarkiewicz, 2017: An unstructured-mesh finite-volume MPDATA for compressible atmospheric dynamics. *J. Comp. Phys.*, **334**, 16–30, doi:10.1016/j.jcp.2016.12.054.

Lashley, R. K., 2002: Automatic generation of accurate advection schemes on unstructured grids and their application to meteorological problems. Ph.D. thesis, University of Reading, 223 pp.

Lauritzen, P., and Coauthors, 2014: 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)**, 105–145, doi:10.5194/gmd-7-105-2014.

Lauritzen, P. H., C. Erath, and R. Mittal, 2011a: On simplifying 'incremental remap'–based transport schemes. *J. Comp. Phys.*, **230 (22)**, 7957–7963, doi:10.1016/j.jcp.2011.06.030.

Lauritzen, P. H., C. Jablonowski, M. A. Taylor, and R. D. Nair, 2011b: *Numerical techniques for global atmospheric models*, Vol. 80. Springer Science & Business Media, doi:10.1007/978-3-642-11640-7.

Lauritzen, P. H., R. D. Nair, and P. A. Ullrich, 2010: A conservative semi-Lagrangian multi-tracer transport scheme (CSLAM) on the cubed-sphere grid. *J. Comp. Phys.*, **229 (5)**, 1401–1424, doi:10.1016/j.jcp.2009.10.036.

Lauritzen, P. H., W. C. Skamarock, M. Prather, and M. Taylor, 2012: A standard test case suite for two-dimensional linear transport on the sphere. *Geosci. Model Dev.*, **5 (3)**, 887–901, doi:10.5194/gmd-5-887-2012.

Leibniz Institute for Tropospheric Research, and J. Shaw, 2018: ASAM cut cell grid generator. Zenodo, doi:10.5281/zenodo.1174692.

Leonard, B., A. Lock, and M. MacVean, 1996: Conservative explicit unrestricted-time-step multi-dimensional constancy-preserving advection schemes. *Mon. Wea. Rev.*, **124 (11)**, 2588–2606, doi:10.1175/1520-0493(1996)124<2588:CEUTSM>2.0.CO;2.

Leonard, B., M. MacVean, and A. Lock, 1993: Positivity-preserving numerical schemes for multi-dimensional advection. Tech. Rep. 106055, NASA.

Leuenberger, D., M. Koller, O. Fuhrer, and C. Schär, 2010: A generalization of the SLEVE vertical coordinate. *Mon. Wea. Rev.*, **138 (9)**, 3683–3689, doi:10.1175/2010MWR3307.1.

LeVeque, R. J., 2002: *Finite volume methods for hyperbolic problems*, Vol. 31. Cambridge university press, doi:10.1017/CBO9780511791253.

Li, X., C. Chen, X. Shen, and F. Xiao, 2013: A multimoment constrained finite-volume model for nonhydrostatic atmospheric dynamics. *Mon. Wea. Rev.*, **141 (4)**, 1216–1240, doi:10.1175/MWR-D-12-00144.1.

Lin, S.-J., and R. B. Rood, 1996: Multidimensional flux-form semi-Lagrangian transport schemes. *Mon. Wea. Rev.*, **124 (9)**, 2046–2070, doi:10.1175/1520-0493(1996)124<2046:MFFSLT>2.0.CO;2.

Lorenz, E. N., 1960: Energy and numerical weather prediction. *Tellus*, **12 (4)**, 364–373, doi:10.1111/j.2153-3490.1960.tb01323.x.

Luo, Y., C. Guan, and D. Wu, 2002: An Eta-coordinate version of the Princeton Ocean Model. *J. Oceanogr.*, **58 (4)**, 589–597, doi:10.1023/A:1021270911741.

Mahrer, Y., 1984: An improved numerical approximation of the horizontal gradients in a terrain-following coordinate system. *Mon. Wea. Rev.*, **112 (5)**, 918–922, doi:10.1175/1520-0493(1984)112<0918:AINAOT>2.0.CO;2.

Marras, S., and Coauthors, 2016: A review of element-based Galerkin methods for numerical weather prediction: Finite elements, spectral elements, and discontinuous Galerkin. *Arch. Comput. Method. E.*, **23**, 673–722, doi:10.1007/s11831-015-9152-1.

McCorquodale, P., P. A. Ullrich, H. Johansen, P. Colella, and Coauthors, 2015: An adaptive multiblock high-order finite-volume method for solving the shallow-water equations on the sphere. *Comm. App. Math. Comp. Sci.*, **10 (2)**, 121–162, doi:10.2140/camcos.2015.10.121.

Melvin, T., M. Dubal, N. Wood, A. Staniforth, and M. Zerroukat, 2010: An inherently mass-conserving iterative semi-implicit semi-Lagrangian discretization of the non-hydrostatic vertical-slice equations. *Quart. J. Roy. Meteor. Soc.*, **136 (648)**, 799–814, doi:10.1002/qj.603.

Mesinger, F., Z. I. Janjić, S. Ničkovic, D. Gavrilov, and D. G. Deaven, 1988: The step-mountain co-ordinate: Model description and performance for cases of Alpine lee cyclogenesis and for a case of an Appalachian redevelopment. *Mon. Wea. Rev.*, **116 (7)**, 1493–1518, doi:10.1175/1520-0493(1988)116<1493:TSMCMD>2.0.CO;2.

Mesinger, F., and Coauthors, 2012: An upgraded version of the Eta model. *Meteor. Atmos. Phys.*, **116 (3-4)**, 63–79, doi:10.1007/s00703-012-0182-z.

Methven, J., and B. Hoskins, 1999: The advection of high-resolution tracers by low-resolution winds. *J. Atmos. Sci.*, **56 (18)**, 3262–3285, doi:10.1175/1520-0469(1999)056<3262:TAOHRT>2.0.CO;2.

Miura, H., 2007: An upwind-biased conservative advection scheme for spherical hexagonal-pentagonal grids. *Mon. Wea. Rev.*, **135 (12)**, 4038–4044, doi:10.1175/2007MWR2101.1.

Müller, A., J. Behrens, F. X. Giraldo, and V. Wirth, 2013: Comparison between adaptive and uniform discontinuous Galerkin simulations in dry 2D bubble experiments. *J. Comp. Phys.*, **235**, 371–393, doi:10.1016/j.jcp.2012.10.038.

Nair, R. D., 2015: Quadrature-free implementation of a discontinuous Galerkin global shallow-water model via flux correction procedure. *Mon. Wea. Rev.*, **143 (4)**, 1335–1346, doi:10.1175/MWR-D-14-00174.1.

Nair, R. D., and P. H. Lauritzen, 2010: A class of deformational flow test cases for linear transport problems on the sphere. *J. Comp. Phys.*, **229 (23)**, 8868–8887,

doi:10.1016/j.jcp.2010.08.014.

Nair, R. D., S. J. Thomas, and R. D. Loft, 2005: A discontinuous Galerkin transport scheme on the cubed sphere. *Mon. Wea. Rev.*, **133 (4)**, 814–828, doi:10.1175/MWR2890.1.

Naughten, K. A., B. K. Galton-Fenzi, K. J. Meissner, M. H. England, G. B. Brassington, F. Colberg, T. Hattermann, and J. B. Debernard, 2017: Spurious sea ice formation caused by oscillatory ocean tracer advection schemes. *Ocean Model.*, **116**, 108–117, doi:10.1016/j.ocemod.2017.06.010.

Phillips, N. A., 1957: A coordinate system having some special advantages for numerical forecasting. *J. Meteor.*, **14 (2)**, 184–185, doi:10.1175/1520-0469(1957)014<0184:ACSHSS>2.0.CO;2.

Putman, W. M., and S.-J. Lin, 2007: Finite-volume transport on various cubed-sphere grids. *J. Comp. Phys.*, **227 (1)**, 55–78, doi:10.1016/j.jcp.2007.07.022.

Qaddouri, A., and V. Lee, 2011: The Canadian Global Environmental Multiscale model on the Yin-Yang grid system. *Quart. J. Roy. Meteor. Soc.*, **137**, 1913–1926, doi:10.1002/qj.873.

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

Schneider, E. K., 1987: An inconsistency in vertical discretization in some atmospheric models. *Mon. Wea. Rev.*, **115 (9)**, 2166–2169, doi:10.1175/1520-0493(1987)115<2166:AIIVDI>2.0.CO;2.

Schwartz, P., and Coauthors, 2015: High-accuracy embedded boundary grid generation using the divergence theorem. *Comm. App. Math. Comp. Sci.*, **10 (1)**, 83–96, doi:10.2140/camcos.2015.10.83.

Shaw, J., 2018a: AtmosTests Singularity container. Zenodo, doi:10.5281/zenodo.1183332.

Shaw, J., 2018b: AtmosTests: suite of idealised atmospheric tests. Zenodo, doi:10.5281/zenodo.1181452.

Shaw, J., 2018c: GMV-to-OpenFOAM file format converter. Zenodo, doi:10.5281/zenodo.1174703.

Shaw, J., 2018d: highOrderFit: a high-order OpenFOAM interpolation scheme for advective fluxes. Zenodo, doi:10.5281/zenodo.1174733.

Shaw, J., 2018e: ninjaopenfoam: OpenFOAM ninja-build autogenerator. Zenodo, doi:10.5281/zenodo.1181427.

Shaw, J., 2018f: OpenFOAM test result data accompanying Numerical representation of mountains in atmospheric models'. Zenodo, doi:10.5281/zenodo.1183278.

Shaw, J., 2018g: Source files for the PhD thesis, 'Numerical representation of mountains in atmospheric models'. Zenodo, doi:10.5281/zenodo.1187031.

Shaw, J., and H. Weller, 2016: Comparison of terrain-following and cut-cell grids using a non-hydrostatic model. *Mon. Wea. Rev.*, **144 (6)**, 2085–2099, doi:10.1175/MWR-D-15-0226.1.

Shaw, J., H. Weller, and W. McIntyre, 2018: AtmosFOAM-tools: Common applications and libraries for AtmosFOAM. Zenodo, doi:10.5281/zenodo.1181443.

Shaw, J., H. Weller, J. Methven, and T. Davies, 2017: Multidimensional method-of-lines transport for atmospheric flows over steep terrain using arbitrary meshes. *J. Comp. Phys.*, **344**, 86–107, doi:10.1016/j.jcp.2017.04.061.

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

Sjögreen, B., 2012: High order finite difference and finite volume methods for advection on the sphere. *J. Sci. Comput.*, **51 (3)**, 703–732, doi:10.1007/s10915-011-9527-x.

Skamarock, W. C., and A. Gassmann, 2011: Conservative transport schemes for spherical geodesic grids: High-order flux operators for ODE-based time integration. *Mon. Wea. Rev.*, **139 (9)**, 2962–2975, doi:10.1175/MWR-D-10-05056.1.

Skamarock, W. C., and M. Menchaca, 2010: Conservative transport schemes for spherical geodesic grids: High-order reconstructions for forward-in-time schemes. *Mon. Wea. Rev.*, **138 (12)**, 4497–4508, doi:10.1175/2010MWR3390.1.

Smolarkiewicz, P. K., and J. Szmelter, 2005: MPDATA: An edge-based unstructured-grid formulation. *J. Comp. Phys.*, **206 (2)**, 624–649, doi:10.1016/j.jcp.2004.12.021.

Smolarkiewicz, P. K., and J. Szmelter, 2011: A nonhydrostatic unstructured-mesh sound-proof model for simulation of internal gravity waves. *Acta Geophys.*, **59 (6)**, 1109–1134, doi:10.2478/s11600-011-0043-z.

Staniforth, A., and J. Thuburn, 2012: Horizontal grids for global weather and climate prediction models: a review. *Quart. J. Roy. Meteor. Soc.*, **138 (662)**, 1–26, doi:10.1002/qj.958.

Steppeler, J., H.-W. Bitzer, M. Minotte, and L. Bonaventura, 2002: Nonhydrostatic atmospheric modeling using a z-coordinate representation. *Mon. Wea. Rev.*, **130**, 2143–2149, doi:10.1175/1520-0493(2002)130<2143:NAMUAZ>2.0.CO;2.

Steppeler, J., R. Hess, U. Schättler, and L. Bonaventura, 2003: Review of numerical meth-

ods for nonhydrostatic weather prediction models. *Meteor. Atmos. Phys.*, **82**, 287–301, doi:10.1007/s00703-001-0593-8.

Steppeler, J., and J. Klemp, 2017: Advection on cut-cell grids for an idealized mountain of constant slope. *Mon. Wea. Rev.*, **145 (5)**, 1765–1777, doi:10.1175/MWR-D-16-0308.1.

Steppeler, J., S.-H. Park, and A. Dobler, 2013: Forecasts covering one month using a cut-cell model. *Geosci. Model Dev.*, **6**, 875–882, doi:10.5194/gmd-6-875-2013.

Theurich, G., and Coauthors, 2016: The Earth System Prediction Suite: Toward a coordinated U.S. modeling capability. *Bull. Amer. Meteor. Soc.*, **97 (7)**, 1229–1247, doi:10.1175/BAMS-D-14-00164.1.

Thuburn, J., C. Cotter, and T. Dubos, 2014: 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)**, 909–929, doi:10.5194/gmd-7-909-2014.

Thuburn, J., and C. J. Cotter, 2012: A framework for mimetic discretization of the rotating shallow-water equations on arbitrary polygonal grids. *SIAM J. Sci. Comput.*, **34 (3)**, B203–B225, doi:10.1137/110850293.

Thuburn, J., C. J. Cotter, T. Dubos, H. Weller, and J. Shaw, 2018: Hexagonal/triangular geodesic mesh generator. doi:10.5281/zenodo.1174742.

Thuburn, J., and T. Woollings, 2005: Vertical discretizations for compressible Euler equation atmospheric models giving optimal representation of normal modes. *J. Comp. Phys.*, **203 (2)**, 386–404, doi:10.1016/j.jcp.2004.08.018.

Tsoutsanis, P., and D. Drikakis, 2016: A high-order finite-volume method for atmospheric flows on unstructured grids. *J. Coupled Syst. Multiscale Dyn.*, **4 (3)**, 170–186, doi:10.1166/jcsmd.2016.1104.

Tuzikov, A., S. Sheynin, and P. V. Vasiliev, 2003: Computation of volume and surface body moments. *Pattern Recognit.*, **36 (11)**, 2521–2529, doi:10.1016/S0031-3203(03)00127-4.

Ullrich, P., 2014: Understanding the treatment of waves in atmospheric models. Part 1: The shortest resolved waves of the 1D linearized shallow-water equations. *Quart. J. Roy. Meteor. Soc.*, **140 (682)**, 1426–1440, doi:10.1002/qj.2226.

Ullrich, P. A., and C. Jablonowski, 2012: MCore: A non-hydrostatic atmospheric dynamical core utilizing high-order finite-volume methods. *J. Comp. Phys.*, **231 (15)**, 5078–5108, doi:10.1016/j.jcp.2012.04.024.

Ullrich, P. A., C. Jablonowski, and B. Van Leer, 2010: High-order finite-volume methods for the shallow-water equations on the sphere. *J. Comp. Phys.*, **229 (17)**, 6104–6134,

doi:10.1016/j.jcp.2010.04.044.

Untch, A., and M. Hortal, 2004: A finite-element scheme for the vertical discretization of the semi-Lagrangian version of the ECMWF forecast model. *Quart. J. Roy. Meteor. Soc.*, **130 (599)**, 1505–1530, doi:10.1256/qj.03.173.

Van Brummelen, G., 2013: *Heavenly mathematics: The forgotten art of spherical trigonometry*. Princeton University Press, 217 pp.

Van Hooft, J., S. Popinet, C. Van Heerwaarden, S. Van Der Linden, S. De Roode, and B. Van de Wiel, 2018: Towards adaptive grids for atmospheric boundary-layer simulations. *Bound.-Layer Meteorol.*, doi:10.1007/s10546-018-0335-9, accepted.

Wang, Q., W. Zhou, Y. Cheng, G. Ma, X. Chang, Y. Miao, and E. Chen, 2018: Regularized moving least-square method and regularized improved interpolating moving least-square method with nonsingular moment matrices. *Appl. Math. Comput.*, **325**, 120–145, doi:10.1016/j.amc.2017.12.017.

Waruszewski, M., C. Kühnlein, H. Pawlowska, and P. K. Smolarkiewicz, 2018: MPDATA: Third-order accuracy for variable flows. *J. Comp. Phys.*, doi:10.1016/j.jcp.2018.01.005, accepted.

Wedi, N. P., 2014: Increasing horizontal resolution in numerical weather prediction and climate simulations: illusion or panacea? *Philos. Trans. Roy. Soc. A*, **372 (2018)**, 20130 289, doi:10.1098/rsta.2013.0289.

Weller, H., and A. Shahrokhi, 2014: Curl free pressure gradients over orography in a solution of the fully compressible Euler equations with implicit treatment of acoustic and gravity waves. *Mon. Wea. Rev.*, **142 (12)**, 4439–4457, doi:10.1175/MWR-D-14-00054.1.

Weller, H., J. Shaw, W. McIntyre, P. Browne, and H. Yamazaki, 2018: Atmos-FOAM: OpenFOAM applications and libraries supporting atmospheric modelling. Zenodo, doi:10.5281/zenodo.1181446.

Weller, H., J. Thuburn, and C. J. Cotter, 2012: Computational modes and grid imprinting on five quasi-uniform spherical C grids. *Mon. Wea. Rev.*, **140**, 2734–2755, doi:10.1175/MWR-D-11-00193.1.

Weller, H., H. G. Weller, and A. Fournier, 2009: Voronoi, Delaunay, and block-structured mesh refinement for solution of the shallow-water equations on the sphere. *Mon. Wea. Rev.*, **137 (12)**, 4208–4224, doi:10.1175/2009MWR2917.1.

White, J., and J. J. Dongarra, 2011: High-performance high-resolution semi-Lagrangian tracer transport on a sphere. *J. Comp. Phys.*, **230 (17)**, 6778–6799, doi:10.1016/j.jcp.2011.05.008.

White, L., R. Panchadhara, and D. Trenev, 2017: Flow simulation in heterogeneous porous

media with the moving least-squares method. *SIAM J. Sci. Comput.*, **39 (2)**, B323–B351, doi:10.1137/16M1070840.

Xie, B., and F. Xiao, 2016: A multi-moment constrained finite volume method on arbitrary unstructured grids for incompressible flows. *J. Comp. Phys.*, **327**, 747–778, doi:10.1016/j.jcp.2016.09.054.

Yamazaki, H., and T. Satomura, 2010: Nonhydrostatic atmospheric modeling using a combined Cartesian grid. *Mon. Wea. Rev.*, **138**, 3932–3945, doi:10.1175/2010MWR3252.1.

Yamazaki, H., and T. Satomura, 2012: Non-hydrostatic atmospheric cut cell model on a block-structured mesh. *Atmos. Sci. Lett.*, **13 (1)**, 29–35, doi:10.1002/asl.358.

Yamazaki, H., T. Satomura, and N. Nikiforakis, 2016: Three-dimensional cut-cell modelling for high-resolution atmospheric simulations. *Quart. J. Roy. Meteor. Soc.*, **142**, 1335–1350, doi:10.1002/qj.2736.

Yang, X., J. Chen, J. Hu, D. Chen, X. Shen, and H. Zhang, 2007: A semi-implicit semi-Lagrangian global nonhydrostatic model and the polar discretization scheme. *Sci. China Ser. D Earth Sci.*, **50 (12)**, 1885–1891, doi:10.1007/s11430-007-0124-7.

Yi, T.-H., and J.-R. Park, 2017: Vertical discretization with finite elements for a global hydrostatic model on the cubed sphere. *J. Comp. Phys.*, **338**, 339–356, doi:10.1016/j.jcp.2017.02.067.

Zadra, A., M. Buehner, S. Laroche, and J.-f. Mahfouf, 2004: Impact of the GEM model simplified physics on extratropical singular vectors. *Quart. J. Roy. Meteor. Soc.*, **130 (602)**, 2541–2569, doi:10.1256/qj.03.208.

Zängl, G., 2012: Extending the numerical stability limit of terrain-following coordinate models over steep slopes. *Mon. Wea. Rev.*, **140**, 3722–3733, doi:10.1175/MWR-D-12-00049.1.

Zängl, G., L. Gantner, G. Hartjenstein, and H. Noppel, 2004: Numerical errors above steep topography: A model intercomparison. *Meteor. Z.*, **13 (2)**, 69–76, doi:10.1127/0941-2948/2004/0013-0069.

Zängl, G., D. Reinert, P. Rípodas, and M. Baldauf, 2015: The ICON (ICOsahedral Non-hydrostatic) modelling framework of DWD and MPI-M: Description of the non-hydrostatic dynamical core. *Quart. J. Roy. Meteor. Soc.*, **141 (687)**, 563–579, doi:10.1002/qj.2378.

Zerroukat, M., and T. Allen, 2012: A three-dimensional monotone and conservative semi-Lagrangian scheme (SLICE-3D) for transport problems. *Quart. J. Roy. Meteor. Soc.*, **138 (667)**, 1640–1651, doi:10.1002/qj.1902.

Zhu, H., and R. K. Smith, 2003: Effects of vertical differencing in a minimal hurricane model. *Quart. J. Roy. Meteor. Soc.*, **129 (589)**, 1051–1069, doi:10.1256/qj.02.78.