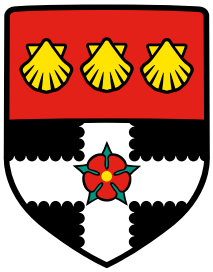
University of Reading

Department of Meteorology



Numerical representation of mountains in atmospheric models

James Shaw

TODO: *date*

Abstract

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Contents

1 Introduction

2 Existing methodologies

TODO:

- *meshes: BTF, SLEVE?, cut cells*
- *exnerFoamH here or in chapter 4?*

$$\text{Momentum} \qquad \frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot \rho \mathbf{u} \otimes \mathbf{u} = \rho \mathbf{g} - c_p \rho \theta \nabla \Pi - \mu \rho \mathbf{u} \tag{2.1a}$$

$$\text{Continuity} \qquad \frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{u} = 0 \tag{2.1b}$$

$$\text{Thermodynamic equation} \qquad \frac{\partial \rho \theta}{\partial t} + \nabla \cdot \rho \mathbf{u} \theta = 0 \tag{2.1c}$$

$$\text{Ideal gas law} \qquad \Pi^{(1-\kappa)/\kappa} = \frac{R \rho \theta}{p_0} \tag{2.1d}$$

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 used TODO: *for the sponge layer in the gravity waves test in section nnn*.

TODO: *The fully-compressible 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 without Coriolis forces.*

Acoustic and TODO: *gravity waves are treated implicitly* and advection is treated explicitly. The trapezoidal implicit treatment of fast waves and the Hodge operator suitable for non-orthogonal grids are described in TODO: *appendix or ref to shaw-weller2016?*. To avoid time-splitting errors between the advection and the fast waves, the advection is time-stepped using a three-stage, second-order Runge-Kutta scheme. The advection terms of the momentum and θ equations, (2.1a) and (2.1c), are discretised in flux form using TODO: *the cubicFit transport scheme ... or linearUpwind*.

3 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, not simply mesh distortions

TODO: *Motivation*

- *Conservation of tracer, therefore Eulerian*
- *Numerically stable and accurate on arbitrary meshes (because no consensus on "best" mesh: hex or cubed-sphere? TF or cut cells?)*
- *Avoid all splitting errors (because they get worse with steeper slopes represented by TF meshes)*
- *Method-of-lines approach permits low computational cost*

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

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{u}\phi) = 0 \,. \tag{3.1}$$

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

$$\phi^{\star} = \phi^{(n)} + \Delta t \; g(\phi^{(n)}) \tag{3.2a}$$

$$\phi^{(n+1)} = \phi^{(n)} + \frac{\Delta t}{2}\left[g(\phi^{(n)}) + g(\phi^{\star})\right] \tag{3.2b}$$

where $g(\phi^{(n)}) = -\nabla \cdot (\mathbf{u}\phi^{(n)})$ at time level n. The same time-stepping method is used for both the cubicFit scheme and the multidimensional linear upwind scheme. Although the Heun method

is unstable for a linear oscillator (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.

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 (3.1) is discretised using Gauss's theorem:

$$\nabla \cdot (\mathbf{u}\phi) \approx \frac{1}{\mathcal{V}_c} \sum_{f \in c} \mathbf{u}_f \cdot \mathbf{S}_f \phi_F \tag{3.3}$$

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. Note that equation (3.3) is a second-order approximation of the divergence term which limits the cubicFit transport scheme to second-order numerical convergence.

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

$$\mathrm{Co}_c = \frac{\Delta t}{2\mathcal{V}_c} \sum_{f \in c} |\mathbf{u} \cdot \mathbf{S}_f| \tag{3.4}$$

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

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

3.1 Cubic fit transport scheme

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

The upwind-biased stencil for a face f is shown in figure 3.1a. 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

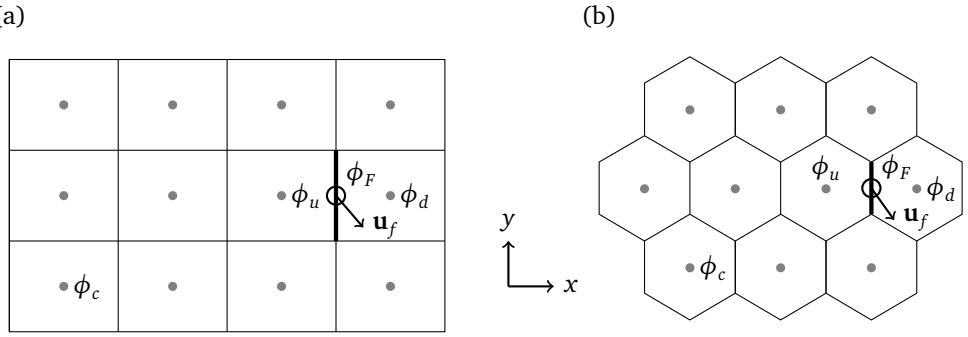


Figure 3.1: Upwind-biased stencils for faces far away from the boundaries of two-dimensional (a) rectangular and (b) hexagon meshes. The stencil is used to fit a multidimensional polynomial to cell centre values, ϕ_c, marked by grey circles, in order to approximate the value ϕ_F at the face centroid 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.

polynomial least-squares fit is calculated using the stencil values. The stencil has 4 points in x and 3 points in y, leading to a natural choice of polynomial that is cubic in x and quadratic in y,

$$\phi = a_1 + a_2 x + a_3 y + a_4 x^2 + a_5 xy + a_6 y^2 + a_7 x^3 + a_8 x^2 y + a_9 x y^2 \,. \tag{3.5}$$

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

$$\mathrm{Opp}(f, g) \equiv -\frac{\mathbf{S}_f \cdot \mathbf{S}_g}{|\mathbf{S}_f|^2} \tag{3.6}$$

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 (3.6) as

$$\mathrm{Opp}(f, g) = -\frac{|\mathbf{S}_g|}{|\mathbf{S}_f|} \cos(\theta) \tag{3.7}$$

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 Opp ≥ 0.5, and the face with the maximum opposedness. Expressed in set notation, this is

$$\mathrm{OF}(f, c_u) \equiv \{g : \mathrm{Opp}(f, g) \geq 0.5\} \cup \{g : \max_{g \in G}(\mathrm{Opp}(f, g))\} \,. \tag{3.8}$$

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

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 3.2 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.

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 (3.5) that has nine unknown coefficients, $\mathbf{a} = a_1 \ldots a_9$,

[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.

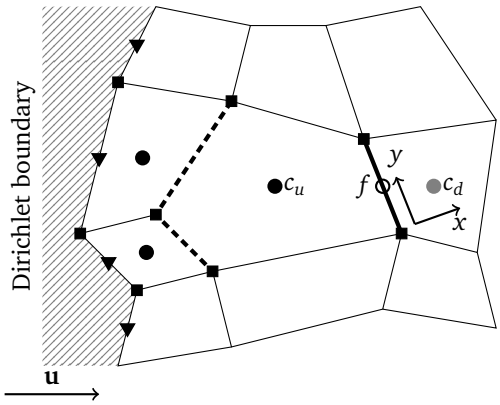


Figure 3.2: A fourteen-point, upwind-biased stencil for face f connecting the pentagonal upwind cell, c_u, and the downwind cell c_d. The dashed lines denote the two faces of cell c_u that oppose f, and black circles mark the centroids of the internal cells that are connected to these two opposing faces. 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.

using the twelve cell centre values from the upwind-biased stencil, $\boldsymbol{\phi} = \phi_1 \dots \phi_{12}$. This yields a matrix equation

$$\begin{bmatrix} 1 & x_1 & y_1 & x_1^2 & x_1y_1 & y_1^2 & x_1^3 & x_1^2y_1 & x_1y_1^2 \\ 1 & x_2 & y_2 & x_2^2 & x_2y_2 & y_2^2 & x_2^3 & x_2^2y_2 & x_2y_2^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{12} & y_{12} & x_{12}^2 & x_{12}y_{12} & y_{12}^2 & x_{12}^3 & x_{12}^2y_{12} & x_{12}y_{12}^2 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_9 \end{bmatrix} = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{12} \end{bmatrix} \tag{3.9}$$

which can be written as

$$\mathbf{Ba} = \boldsymbol{\phi} \ . \tag{3.10}$$

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 in which x is normal to the face f and y is perpendicular to x. The coordinates (x_i, y_i) give the position of the centroid of the ith cell in the stencil. A two-dimensional stencil is also used for the tests on spherical meshes in section 3.5. 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,

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

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

$$a_1 = \begin{bmatrix} b^+_{1,1} \\ b^+_{1,2} \\ \vdots \\ b^+_{1,12} \end{bmatrix} \cdot \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{12} \end{bmatrix} \tag{3.12}$$

where the weights $b^+_{1,1} \dots b^+_{1,12}$ are the elements of the first row of $\mathbf{B}^+$. Note that the majority of the least-squares fit procedure depends on the mesh geometry only. An implementation may precompute the pseudo-inverse for each stencil during model initialisation, and only the first row needs to be stored. Since each face has two possible stencils depending on the orientation of the velocity relative to the face, the implementation stores two sets of weights for each face. Knowledge of the values of $\boldsymbol{\phi}$ is only required to calculate the weighted mean given by equation (3.12), 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, $\mathbf{m} = m_1 \dots m_{12}$, and multiply equation (3.10) to obtain

$$\tilde{\mathbf{B}}\mathbf{a} = \mathbf{m} \cdot \boldsymbol{\phi} \tag{3.13}$$

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

$$a_1 = \tilde{\mathbf{b}}^+_1 \cdot \mathbf{m} \cdot \boldsymbol{\phi} \tag{3.14}$$

where $\tilde{\mathbf{b}}^+_1 = \tilde{b}^+_{1,1} \dots \tilde{b}^+_{1,12}$ are the elements of the first row of $\tilde{\mathbf{B}}^+$. Again, a_1 is a weighted mean of $\boldsymbol{\phi}$, where the weights are now $\tilde{\mathbf{b}}^+_1 \cdot \mathbf{m}$. 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 3.1.

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 3.1a), and a hexagonal mesh has 10 points

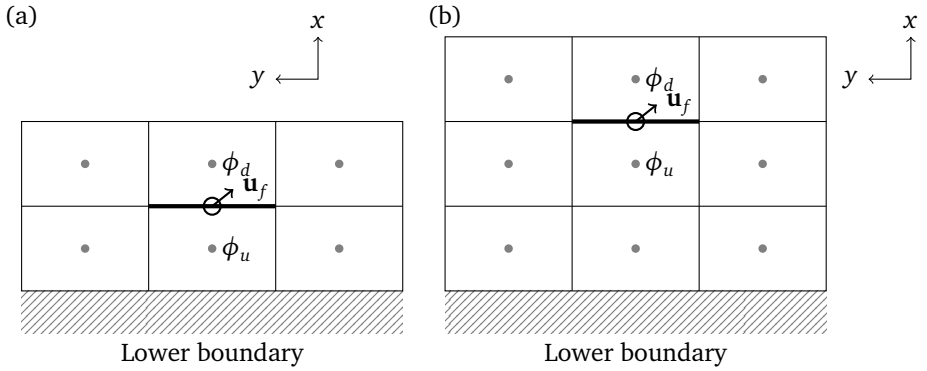


Figure 3.3: Upwind-biased stencils for faces near the lower boundary of a rectangular x–z mesh, with (a) a 3×2 stencil for the face immediately adjacent to the lower boundary, and (b) a 3×3 stencil for the face immediately adjacent to the face in (a). Each stencil belongs to the face marked by a thick 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 (3.5) would result in an underconstrained problem. There is no normal flow at the lower boundary.

(figure 3.1b). In both cases, constructing a system of equations using the nine-term polynomial in equation (3.5) 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 3.3a) 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 3.3b 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 underconstrained problem. In two dimensions, a candidate polynomial has some combination of between one and nine terms from equation (3.5). 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

$$M(x,y) = x^i y^j : i,j \geq 0 \text{ and } i \leq 3 \text{ and } j \leq 2 \text{ and } i+j \leq 3 \tag{3.15}$$

be the set of all monomials of degree at most 3 in x, y. A subset S of $M(x,y)$ is "dense" if, whenever $x^a y^b$ is in S, then $x^i y^j$ is also in S for all $0 \leq i \leq a$, $0 \leq j \leq b$. For example, the polynomial $\phi = a_1 + a_2 x + a_3 y + a_4 xy + a_5 x^2 + a_6 x^2 y$ is a dense subset of $M(x,y)$, but $\phi = a_1 + a_2 x + a_3 y + a_4 x^2 y$ is not because $x^2 y$ can be included only if xy and x^2 are also included. In total there are 26 dense subsets of the two-dimensional polynomial in equation (3.5).

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 3.3b results in a rank-deficient matrix and so the nine-term polynomial is not a candidate polynomial.

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

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 (3.14). 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.

Two-cell analysis

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

$$\frac{\partial \phi_j}{\partial t} = -v \frac{\phi_R - \phi_L}{\Delta x} \tag{3.16}$$

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

$$\phi_L = \alpha_u \phi_{j-1} + \alpha_d \phi_j \tag{3.17}$$

$$\phi_R = \beta_u \phi_j + \beta_d \phi_{j+1} \tag{3.18}$$

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

$$\phi_j^{(n)} = A^n e^{ijk\Delta x} \tag{3.19}$$

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 (3.16)

$$\frac{\partial \phi_j}{\partial t} = \frac{\partial}{\partial t}\left(A^{t/\Delta t}\right) e^{ijk\Delta x} = \frac{\ln A}{\Delta t} A^n e^{ikj\Delta x} \tag{3.20}$$

hence equation (3.16) becomes

$$\frac{\ln A}{\Delta t} = -\frac{\nu}{\Delta x}\left(\beta_u + \beta_d e^{ik\Delta x} - \alpha_u e^{-ik\Delta x} - \alpha_d\right) \tag{3.21}$$

$$\ln A = -c\left(\beta_u - \alpha_d + \beta_d \cos k\Delta x + i\beta_d \sin k\Delta x - \alpha_u \cos k\Delta x + i\alpha_u \sin k\Delta x\right) \tag{3.22}$$

where the Courant number $c = \nu\Delta t/\Delta x$. 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

$$\ln A = -c\left(\Re + i\Im\right) \tag{3.23}$$

$$A = e^{-c\Re} e^{-ic\Im} \tag{3.24}$$

and the complex modulus of A is

$$|A| = e^{-c\Re} = \exp\left(-c\left(\beta_u - \alpha_d + (\beta_d - \alpha_u)\cos k\Delta x\right)\right) . \tag{3.25}$$

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

$$(\alpha_u - \alpha_d)(1 - \cos k\Delta x) \geq 0 \quad \forall k\Delta x \tag{3.26}$$

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

$$\alpha_u - \alpha_d \geq 0 . \tag{3.27}$$

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

$$\exp\left(-c\left(\alpha_u - \alpha_d\right)(1 - \cos k\Delta x)\right) \geq \exp\left(-c\left(1 - \cos k\Delta x\right)\right) \quad \forall k\Delta x \tag{3.28}$$

therefore

$$\alpha_u - \alpha_d \leq 1 . \tag{3.29}$$

Now, knowing that $\alpha_u + \alpha_d = 1$ (or $\alpha_d = 1 - \alpha_u$) then, using equations (3.27) and (3.29), we obtain the first two stability conditions,

$$0.5 \leq \alpha_u \leq 1 \text{ and} \tag{3.30}$$

$$0 \leq \alpha_d \leq 0.5 . \tag{3.31}$$

Three-cell analysis

We start again from equation (3.16) but this time approximate ϕ_L and ϕ_R using three cell centre values,

$$\phi_L = \alpha_{uu}\phi_{j-2} + \alpha_u \phi_{j-1} + \alpha_d \phi_j \tag{3.32}$$

$$\phi_R = \alpha_{uu}\phi_{j-1} + \alpha_u \phi_j + \alpha_d \phi_{j+1} \tag{3.33}$$

having used the same weights α_{uu}, α_u and α_d for both left and right fluxes. Substituting equation (3.19) into equation (3.16) we find

$$A = \exp\left(-c\left[\alpha_{uu}\left(e^{-ik\Delta x} - e^{-2ik\Delta x}\right) + \alpha_u\left(1 - e^{-ik\Delta x}\right) + \alpha_d\left(e^{ik\Delta x} - 1\right)\right]\right) \tag{3.34}$$

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

$$\alpha_u - \alpha_d + (\alpha_{uu} - \alpha_u + \alpha_d)\cos k\Delta x - \alpha_{uu}\cos 2k\Delta x \geq 0\ . \tag{3.35}$$

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

$$\alpha_u - \alpha_d \geq |\alpha_{uu}|\ . \tag{3.36}$$

When the same analysis is performed with four cells, α_{uuu}, α_{uu}, α_u and α_d, by varying $k\Delta x$ we find that equation (3.36) holds replacing $|\alpha_{uu}|$ with $\max(|\alpha_{uu}|, |\alpha_{uuu}|)$. Hence, we generalise equation (3.36) to obtain the final stability condition

$$\alpha_u - \alpha_d \geq \max_{p \in P} |\alpha_p| \tag{3.37}$$

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 3.30, 3.31 and 3.36) are used to impose three stability conditions on the weights $\mathbf{w}$,

$$0.5 \leq w_u \leq 1 \tag{3.38a}$$

$$0 \leq w_d \leq 0.5 \tag{3.38b}$$

$$w_u - w_d \geq \max_{p \in P}(|w_p|) \tag{3.38c}$$

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 $w_u + w_d + \sum_{p \in P} w_p = 1$. We hypothesise that the stability conditions given by equation (3.38) are necessary but not sufficient for a transport scheme on arbitrary meshes.

The stability of the one-dimensional transport equation discretised in space and time could be analysed using existing techniques (Baldauf, 2008), but we have only analysed the spatial stability of the cubicFit scheme. Numerical experiments presented in section 4.1 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 3.38, 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 $\mathbf{B}$ is calculated for each candidate, and an ordering is imposed such that the candidate with the larger minimum singular value is preferred. This ordering ensures that the preferred candidate is the highest-order polynomial with the most information content.[2]

In the second step, the most-preferred polynomial is taken from the list of candidates and the multipliers are assigned so that the upwind cell and downwind cell have multipliers $m_u = 2^{10}$ and $m_d = 2^{10}$ respectively, and all peripheral points have multipliers $m_p = 1$. These multipliers are very similar to those used by 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 $\mathbf{w}$ and evaluate them against the stability conditions given in equation (3.38). 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 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 (3.5) with $m_u = m_d = 2^{10}$.

To illustrate the stabilisation procedure, figure 3.4a 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 only uses the x positions of these points and does not use the values of ϕ themselves. The ϕ values are included here for illustration only. Hence, for a given set of x positions, the same set of weights are chosen irrespective of the ϕ values.

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

$$\phi = a_1 + a_2 x + a_3 x^2 + a_4 x^3 \,, \tag{3.39}$$

$$\phi = a_1 + a_2 x + a_3 x^2 \,, \tag{3.40}$$

$$\phi = a_1 + a_2 x \,, \tag{3.41}$$

$$\phi = a_1 \,. \tag{3.42}$$

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

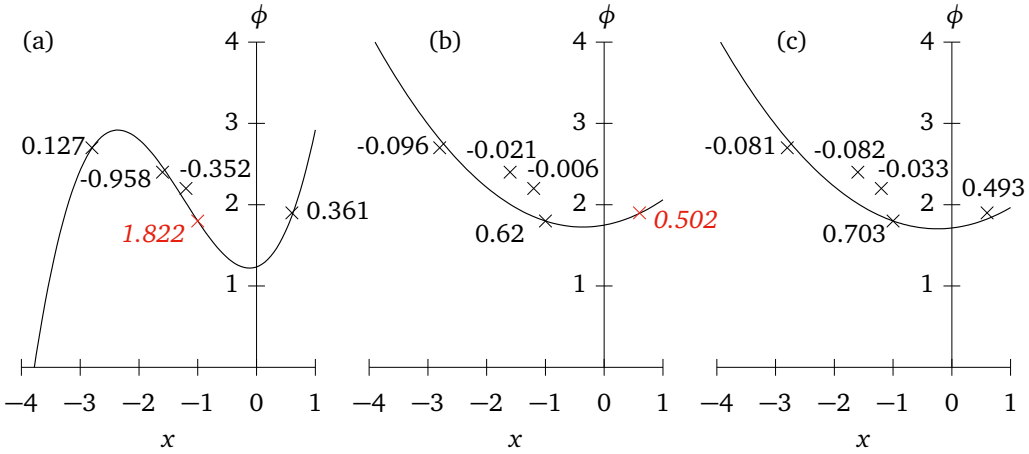


Figure 3.4: One-dimensional least-squares fits with a stencil of five points using (a) a cubic polynomial with multipliers $m_u = 1024$, $m_d = 1024$ and $m_p = 1$, (b) a quadratic polynomial with the same multipliers, and (c) a quadratic polynomial with multipliers $m_u = 1024$, $m_d = 1$ and $m_p = 1$. 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 (3.38) 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 3.1) calculates weights using only x positions, and values of ϕ are included here for illustration only.

We begin with the cubic equation (3.39). 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 3.38a). Reducing the downwind multiplier does not help to satisfy the stability condition, so we start again with the quadratic equation (3.40), and the new fit is presented in figure 3.4b. 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 3.38b). This time, however, stable weights are found by reducing m_d to one (figure 3.4c) 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 3.1 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 (3.5) is replaced with its three-dimensional counterpart,

$$\phi = a_1 + a_2 x + a_3 y + a_4 z + a_5 x^2 + a_6 xy + a_7 y^2 + a_8 xz + a_9 yz + a_{10} z^2 + a_{11} x^3 + a_{12} x^2 y + a_{13} xy^2 + a_{14} x^2 z + a_{15} xz^2 + a_{16} yz^2 + a_{17} y^2 z + a_{18} xyz \,. \quad (3.43)$$

The procedure for generating candidate polynomials described in section 3.1 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 3.1 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 (3.43) with $m_u = m_d = 2^{10}$. The stabilisation procedure does not modify the least squares fit for these stencils, but we have not explored the three-dimensional extension of cubicFit any further.

3.2 Multidimensional linear upwind transport scheme

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

$$\phi_F = \phi_u + \nabla_c \phi \cdot \left(\mathbf{x}_f - \mathbf{x}_c\right) \quad (3.44)$$

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

$$\nabla_c \phi = \frac{1}{\mathcal{V}_c} \sum_{f \in c} \widetilde{\phi}_F \mathbf{S}_f \quad (3.45)$$

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 (3.45) is evaluated. This implementation of the multidimensional linear upwind scheme is included in the OpenFOAM software distribution (CFD Direct, 2016).

3.3 Horizontal transport over mountains

TODO:

- *Meshes: BTF, cut cells*
- *Schemes: cubicFit, linearUpwind, others?*
- *Plot: Tracer and error contours (for schaerAdvect and tfAdvect)*
- *Conclusion: 2nd-order convergence*
- *Conclusion: numerical errors are due either to grid distortions, or misalignment of the flow with the mesh*

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. The 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 this test that has steeper mountains and highly-distorted terrain-following meshes. Convergence results are compared using the linearUpwind and cubicFit transport schemes.

The domain is defined on a rectangular x–z plane that is 301 km wide and 25 km high as measured between parallel boundary edges. Boundary conditions are imposed on the tracer density ϕ such that $\phi = 0\,\mathrm{kg\,m^{-3}}$ at the inlet boundary, and a zero normal gradient $\partial\phi/\partial n = 0\,\mathrm{kg\,m^{-4}}$ is imposed at the outlet boundary. There is no normal flow at the lower and upper boundaries. A range of mesh spacings are chosen so that $\Delta x : \Delta z = 2:1$ to match the original test specification from Schär et al. (2002).

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

$$h(x) = h^\star \cos^2(\alpha x) \tag{3.46a}$$

where

$$h^\star(x) = \begin{cases} h_0 \cos^2(\beta x) & \text{if } |x| < a \\ 0 & \text{otherwise} \end{cases} \tag{3.46b}$$

where $a = 25\,\mathrm{km}$ is the mountain envelope half-width, $h_0 = 6\,\mathrm{km}$ is the maximum mountain height, $\lambda = 8\,\mathrm{km}$ 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).

TODO: *might have already described BTF in a previous chapter* A basic terrain-following (BTF) mesh is constructed by using the terrain profile to modify the uniform mesh. The BTF method uses a linear decay function so that mesh surfaces become horizontal at the top of the model domain (Gal-Chen and Somerville, 1975),

$$z(x) = (H - h(x))(z^\star/H) + h(x) \tag{3.47}$$

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

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

$$u(z) = u_0 \begin{cases} 1 & \text{if } z \geq z_2 \\ \sin^2\left(\frac{\pi}{2}\frac{z-z_1}{z_2-z_1}\right) & \text{if } z_1 < z < z_2 \\ 0 & \text{otherwise} \end{cases} \tag{3.48}$$

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

A tracer with density ϕ has the shape

$$\phi(x,z) = \phi_0 \begin{cases} \cos^2\left(\frac{\pi r}{2}\right) & \text{if } r \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{3.49a}$$

with radius r given by

$$r = \sqrt{\left(\frac{x-x_0}{A_x}\right)^2 + \left(\frac{z-z_0}{A_z}\right)^2} \tag{3.49b}$$

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

Tests are integrated for 10 000 s using time-steps chosen for each mesh so that the maximum Courant number is about 0.4. This choice yields a time-step that is well below any stability limit, as recommended by Lauritzen et al. (Lauritzen et al., 2012). By the end of integration the tracer is positioned downwind of the mountain. The analytic solution at $t = 10\,000\,\mathrm{s}$ is centred at $(x_0, z_0) = (50\,\mathrm{km}, 12\,\mathrm{km})$. Error norms are calculated by subtracting the analytic solution from the numerical solution,

$$\ell_2 = \sqrt{\frac{\sum_c (\phi - \phi_T)^2 \mathcal{V}_c}{\sum_c \left(\phi_T^2 \mathcal{V}_c\right)}} \tag{3.50}$$

$$\ell_\infty = \frac{\max_c |\phi - \phi_T|}{\max_c |\phi_T|} \tag{3.51}$$

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

Tests were performed using the linearUpwind and cubicFit schemes at mesh spacings between $\Delta x = 250\,\mathrm{m}$ and $\Delta x = 5000\,\mathrm{m}$. Numerical convergence in the ℓ_2 and ℓ_∞ norms is plotted in figure 3.6. The linearUpwind and cubicFit schemes are second-order convergent at all but the coarsest mesh spacings where errors are saturated for both schemes.

The cubicFit scheme achieves a given ℓ_2 error using a mesh spacing that is almost twice as coarse as that needed by the linearUpwind scheme. Doubling the mesh spacing results in a coarser mesh with four times fewer cells because the $\Delta x : \Delta z$ aspect ratio is fixed. Recall that the

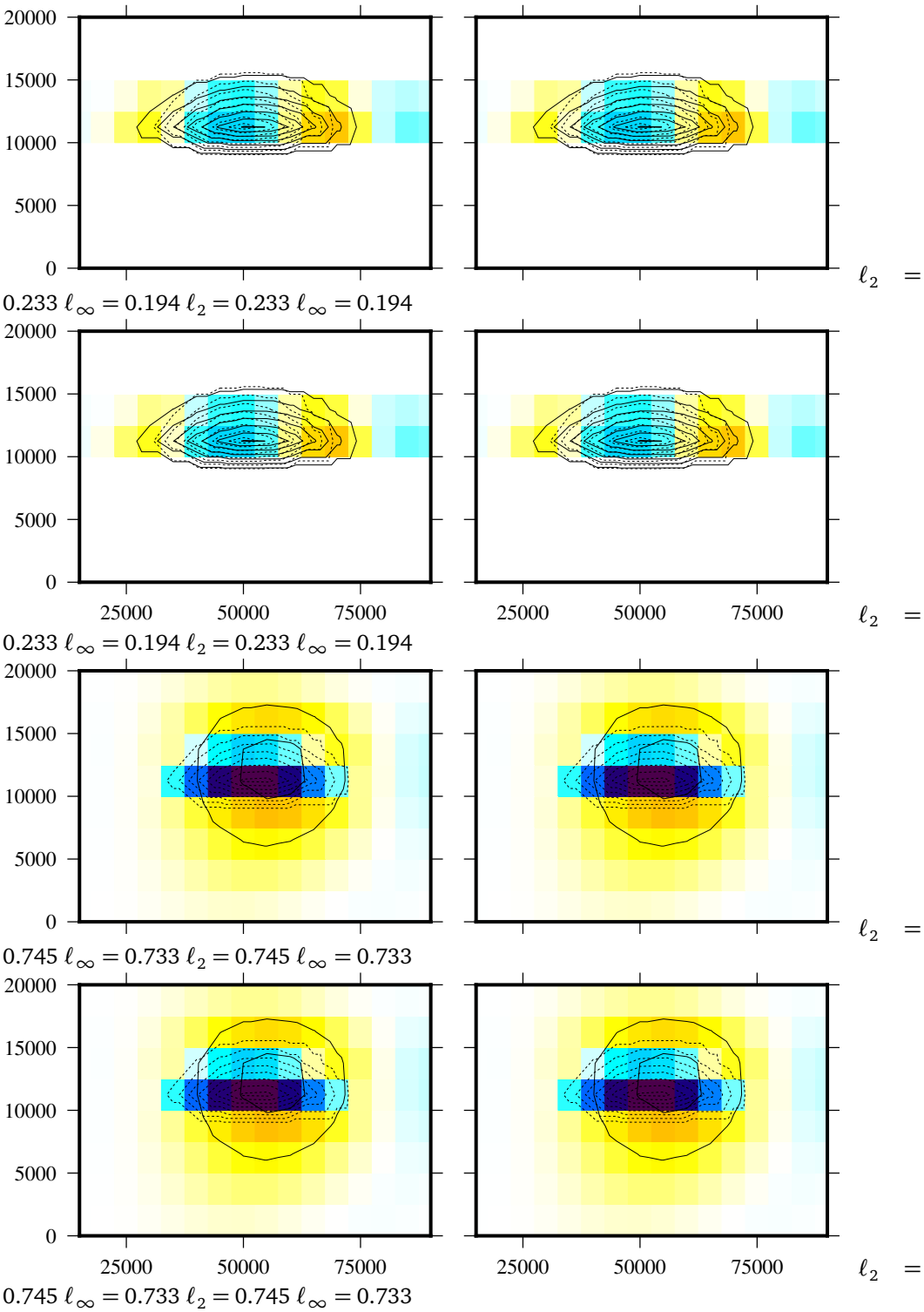


Figure 3.5: TODO: *numerical and analytic tracer contours, error heatmap for schaerAdvect and tfAdvect*

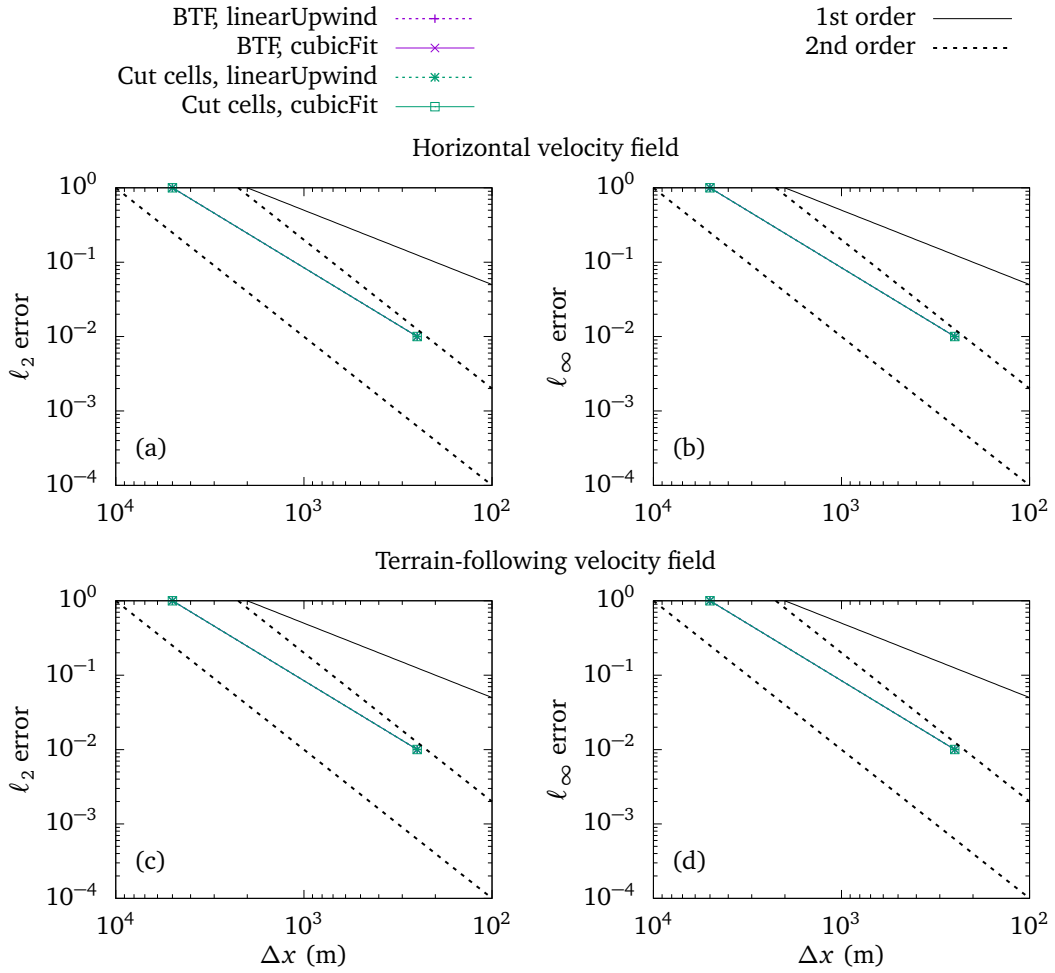


Figure 3.6: Numerical convergence of the two-dimensional tracer transport tests over mountains using (a, b) horizontal and (c, d) terrain-following velocity fields. ℓ_2 errors (equation 3.50) and ℓ_∞ errors (equation 3.51) are marked at mesh spacings between 5000 m and 250 m using linearUpwind and cubicFit transport schemes on basic terrain-following and cut cell meshes.

stencil for the cubicFit scheme has about twice as many cells as the stencil for the linearUpwind scheme. Hence, for a given ℓ_2 error, the computational cost during integration of the cubicFit scheme is about half the computational cost of the linearUpwind scheme.

This test demonstrates that cubicFit is second-order convergent in the domain interior irrespective of mesh distortions. TODO: *In the next test, ...*

3.4 Transport in a terrain-following velocity field

TODO:

- *Meshes: BTF, cut cells*
- *Schemes: cubicFit, linearUpwind, others?*
- *Conclusion: Misalignment of the velocity field with mesh layers is the primary source of numerical error*
- *Conclusion: 2nd-order convergence again*

3.5 Deformational flow on a sphere

TODO: *The tests so far have used flows that are mostly uniform on meshes that are based on rectangular cells.* To ensure that the cubicFit transport scheme is suitable for complex flows on a variety of meshes, we use a standard test of deformational flow on a spherical Earth, as specified by Lauritzen et al. (2012). Results are compared between linearUpwind and cubicFit schemes using hexagonal-icosahedral meshes and cubed-sphere meshes. Hexagonal-icosahedral meshes are constructed by successive refinement of a regular icosahedron following the approach by 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

$$\Delta\lambda = 360^\circ \frac{\overline{\Delta x}}{2\pi R_e} \tag{3.52}$$

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

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

$$\Psi(\lambda,\theta,t) = \frac{10R_e}{T}\sin^2\left(\lambda'\right)\cos^2\left(\theta\right)\cos\left(\frac{\pi t}{T}\right) - \frac{2\pi R_e}{T}\sin\left(\theta\right) \tag{3.53}$$

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,

$$\phi = \phi_1(\lambda,\theta) + \phi_2(\lambda,\theta)\,. \tag{3.54}$$

An individual hill ϕ_i is given by

$$\phi_i(\lambda,\theta) = \phi_0 \exp\left(-b\left(\frac{|\mathbf{x}-\mathbf{x}_i|}{R_e}\right)^2\right) \tag{3.55}$$

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

$$(x,y,z) = (R_e\cos\theta\cos\lambda, R_e\cos\theta\sin\lambda, R_e\sin\theta)\,. \tag{3.56}$$

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

$$(\lambda_1,\theta_1) = (5\pi/6, 0) \tag{3.57}$$

$$(\lambda_2,\theta_2) = (7\pi/6, 0) \tag{3.58}$$

The results in figure 3.7 are obtained using the cubicFit scheme on a hexagonal-icosahedral mesh with $\Delta\lambda = 16.9°$. The initial Gaussian hills are shown in figure 3.7a. At $t = T/2$ the tracer

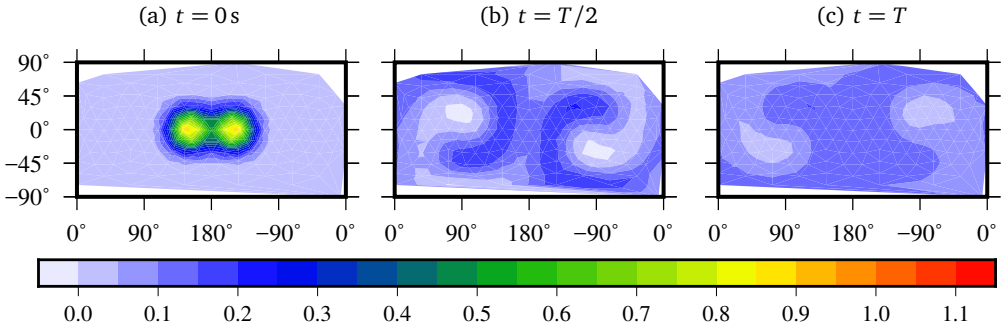


Figure 3.7: Tracer fields for the deformational flow test using initial Gaussian hills. The tracer is deformed by the velocity field before the rotation reverses to return the tracer to its original distribution: (a) the initial tracer distribution at $t = 0\,\mathrm{s}$; (b) by $t = T/2$ the Gaussian 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 $\Delta\lambda = 16.9°$.

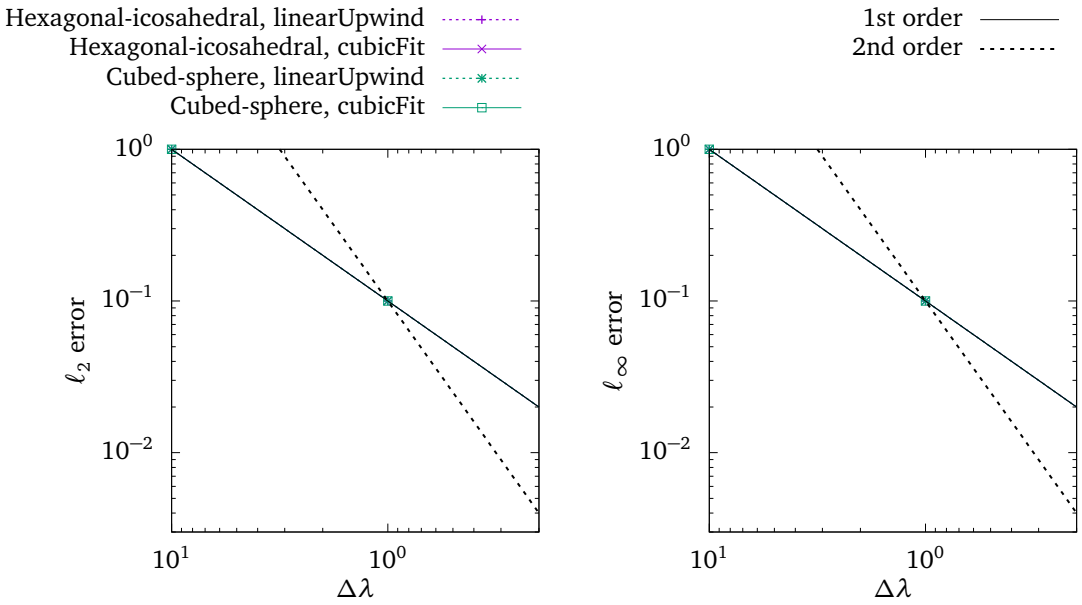


Figure 3.8: Numerical convergence of the deformational flow test on the sphere using initial Gaussian hills. ℓ_2 errors (equation 3.50) and ℓ_∞ errors (equation 3.51) are marked at mesh spacings between 16.9° and 16.9° using the linearUpwind scheme (dotted lines) and the cubicFit scheme (solid lines) on hexagonal-icosahedral meshes and cubed-sphere meshes.

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 3.1: 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).

has been deformed into an S-shaped filament (figure 3.7b). 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 3.7c).

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 $\Delta\lambda = 16.9°$ and $\Delta\lambda = 16.9°$ on hexagonal-icosahedral meshes and cubed-sphere meshes. The results are shown in figure 3.8. The solution is slow to converge at coarse resolutions, and this behaviour agrees with the results from Lauritzen et al. (2012). Both linearUpwind and cubicFit schemes achieve second-order accuracy at smaller 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,

$$\phi = \begin{cases} b + c\phi_1(\lambda, \theta) & \text{if } r_1 < r, \\ b + c\phi_2(\lambda, \theta) & \text{if } r_2 < r, \\ b & \text{otherwise.} \end{cases} \tag{3.59}$$

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 $\ell_2 \approx 0.033$.

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 $\Delta\lambda = 16.9°$ 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 3.1 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 scheme is largely insensitive to the mesh type, and results are more accurate compared to the linearUpwind scheme for a given mesh type and mesh spacing. Neither scheme requires special treatment at the corners of cubed-sphere panels.

4 A new mesh for representing the atmosphere above terrain

Highlights

- The new slanted cell mesh permits longer time-steps than cut cells, with time-steps comparable to terrain-following meshes
- Pressure gradient calculations are more accurate using slanted cells compared to terrain-following meshes
- Unlike the multidimensional linear upwind scheme, the cubicFit scheme is numerically stable over very steep slopes

TODO: *Motivation*

- *cut cells can improve pressure gradient accuracy over TF layers, but they have arbitrarily small cells and they're not simple to construct*
- *we seek a mesh that improves pressure gradient errors, but avoids arbitrarily small cells and is easier to construct than cut cell meshes*
- *to ensure cubicFit is numerically stable and accurate on slanted cell meshes with steep slopes, we transport a tracer along the ground through the slanted cells*

4.1 Transport over a mountainous lower boundary

Following Schär et al. (2002), the domain is 301 km wide and 25 km high as measured between parallel boundary edges, with a mesh spacing of $\Delta x = 1000\,\text{m}$ and $\Delta z = 500\,\text{m}$. The boundary conditions are the same as those used in section 3.3. Cell edges in the central region of the domain are shown in figure 4.1 for each of the three mesh types. Cells in the BTF mesh are highly distorted over steep slopes (figure 4.1a) while the cut cell mesh (figure 4.1b) and slanted cell mesh (figure 4.1c) are orthogonal everywhere except for cells nearest the ground.

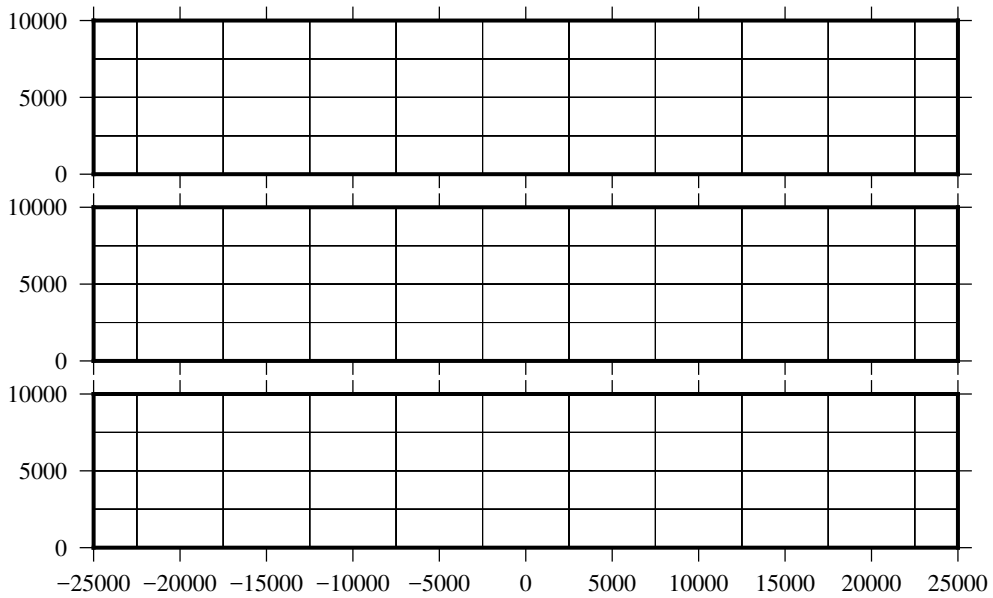


Figure 4.1: Two dimensional x-z meshes created with the (a) basic terrain-following, (b) cut cell, and (c) slanted cell methods, and used for the tracer transport tests in section 4.1. Cell edges are marked by thin black lines. The peak mountain height $h_0 = 5\,\text{km}$. The velocity field is the same for all mesh types with streamlines marked on each panel by thick red lines. The velocity field (equation 4.1) follows the lower boundary and becomes entirely horizontal above $H_1 = 10\,\text{km}$. Only the lowest 10 km for the central region of the domain is shown. The entire domain is 301 km wide and 25 km high.

A velocity field is chosen that follows the terrain at the surface and becomes entirely horizontal aloft. A streamfunction Ψ is used so that the discrete velocity field is non-divergent, such that

$$\Psi(x,z) = -u_0 H_1 \frac{z-h}{H_1-h} \tag{4.1}$$

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

$$u = -\frac{\partial \Psi}{\partial z} = u_0 \frac{H_1}{H_1-h}, \quad w = \frac{\partial \Psi}{\partial x} = u_0 H_1 \frac{\mathrm{d}h}{\mathrm{d}x} \frac{H_1-z}{(H_1-h)^2}\,, \tag{4.2}$$

$$\frac{\mathrm{d}h}{\mathrm{d}x} = -h_0 \left[\beta \cos^2(\alpha x)\sin(2\beta x) + \alpha \cos^2(\beta x)\sin(2\alpha x)\right]\,. \tag{4.3}$$

Unlike the horizontal transport test in Schär et al. (2002), the velocity field presented here extends from the top of the domain all the way to the ground.

The flow is deliberately misaligned with the BTF, cut cell and slanted cell meshes away from the ground (figure 4.1) to ensure that flow always crosses mesh surfaces in order to challenge the transport scheme. The value of H_1 is chosen to be much smaller than the domain height

	Peak mountain height h_0 (km)				
Mesh type	0	3	4	5	6
BTF	40	40	40	40	40
Cut cell	40	40	40	40	40
Slanted cell	40	40	40	40	40

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 4.2: Evolution of the tracer in the two-dimensional transport test over a mountainous lower boundary. The tracer is transported to the right over the wave-shaped terrain. Tracer contours are every 0.1 kg m^{-3}. The result obtained using the cubicFit scheme on the basic terrain-following mesh is shown at $t = 0$ s, $t = 5000$ s and $t = 10\,000$ s with solid black contours. The analytic solution at $t = 10\,000$ s is shown with dotted contours. The shaded box indicates the region that is plotted in figure 4.3.

H in TODO: *equation* (**??**) so that flow crosses the surfaces of the BTF mesh. This is evident in figure 4.1a where the the velocity streamlines are tangential to the mesh only at the ground.

The tracer is defined again by equation (3.49) 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 the new horizontal position of the tracer. Integrating along the trajectory yields t, the time taken to move from the left side of the mountain to the right:

$$\mathrm{d}t = \mathrm{d}x/u(x) \tag{4.4}$$

$$t = \int_0^x \frac{H_1 - h(x)}{u_0 H_1}\,\mathrm{d}x \tag{4.5}$$

$$t = \frac{x}{u_0} - \frac{h_0}{16 u_0 H_1}\left[4x + \frac{\sin 2(\alpha+\beta)x}{\alpha+\beta} + \frac{\sin 2(\alpha-\beta)x}{\alpha-\beta} + 2\left(\frac{\sin 2\alpha x}{\alpha} + \frac{\sin 2\beta x}{\beta}\right)\right] \tag{4.6}$$

By solving this equation we find that $x(t = 10\,000\,\text{s}) = 54\,342.8$ m when $h_0 = 5$ km.

The tracer density boundary conditions are the same as those in section 3.3. Since the cubicFit transport scheme uses values at boundaries with Dirichlet boundary conditions, the cubicFit scheme uses only inlet boundary values in this test case.

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

A 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 4.2 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, but its original shape is restored once it has cleared the mountain by $t = 10\,000\,\text{s}$. A small phase lag is apparent when the numerical solution marked with solid contour lines is compared with the analytic solution marked with dotted contour lines.

Numerical errors are more clearly revealed by subtracting the analytic solution from the numerical solution. Error fields are compared between BTF, cut cell and slanted cell meshes using the linearUpwind scheme (figures 4.3b, 4.3a and 4.3c respectively) and the cubicFit scheme (figures 4.3d, 4.3e and 4.3f respectively). Results are least accurate using the linearUpwind scheme on the slanted cell mesh (figure 4.3c). The final tracer is slightly distorted and does not extend far enough towards the ground. The ℓ_∞ error magnitude is reduced by using the linearUpwind scheme on the cut cell mesh (figure 4.3a), but the shape of the error remains the same. The cubicFit scheme is less sensitive to the choice of mesh with similar error magnitudes on the BTF mesh (figure 4.3d), cut cell mesh (figure 4.3e) and slanted cell mesh (figure 4.3f). Errors using the cubicFit scheme on cut cell and slanted cell meshes are much smaller than the errors using the linearUpwind scheme on the same meshes.

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 4.4). 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 $h_0 = 0\,\text{km}$ and the ground is entirely flat because the BTF, cut cell and slanted cell meshes are identical. Compared with the cubicFit scheme, the linearUpwind scheme is more sensitive to the mesh type and mountain height. The linearUpwind scheme is unstable on the slanted cell mesh with a peak mountain height $h_0 = 6\,\text{km}$ despite using a Courant number of 0.348. In contrast, the cubicFit scheme is less sensitive to the mesh type and errors grow more slowly with increasing mountain height. The cubicFit scheme yields stable results in all tests.

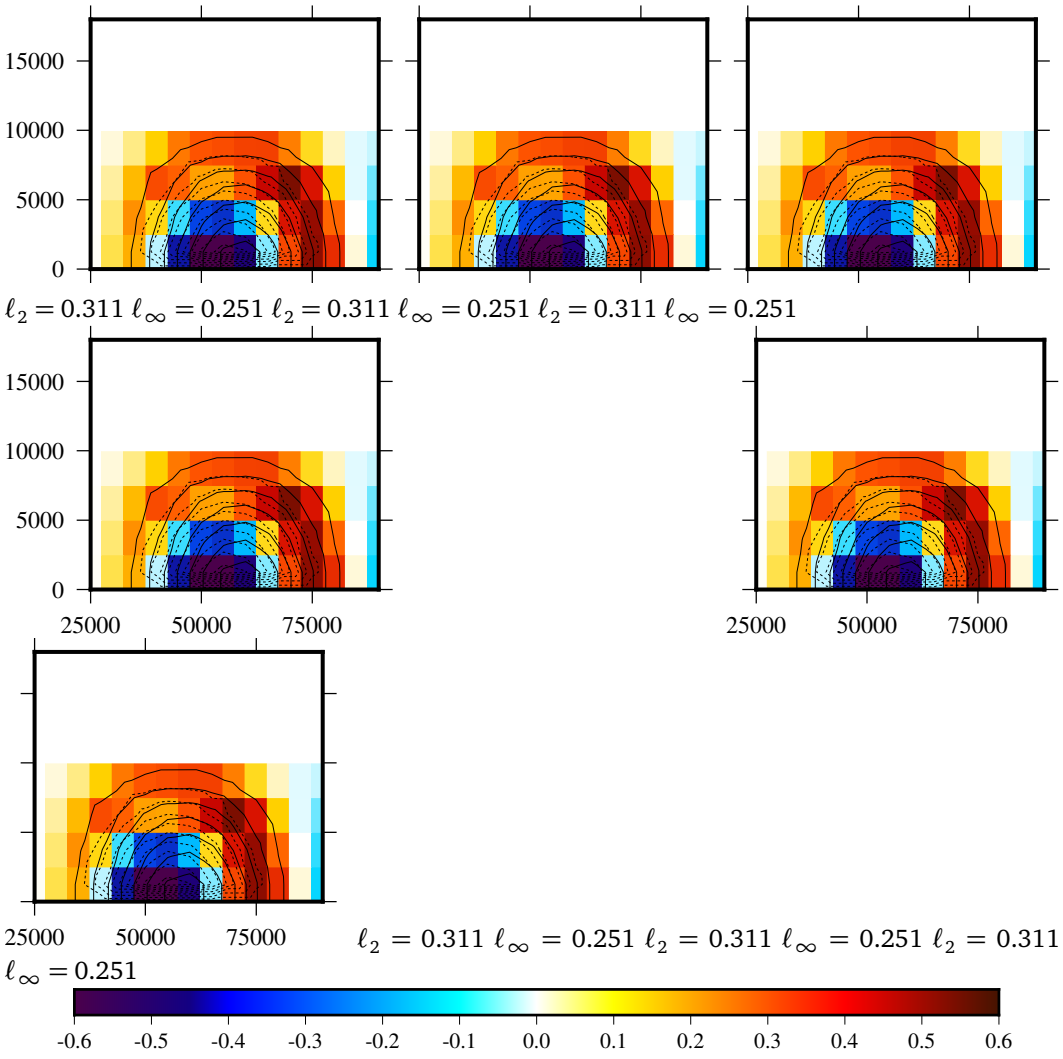


Figure 4.3: Tracer contours at $t = 10\,000$ s for the two-dimensional tracer transport tests over a mountainous lower boundary. A region in the lee of the mountain is plotted corresponding to the shaded area in figure 4.2. Results are presented on BTF, cut cell and slanted cell meshes (shown in figure 4.1) using the linearUpwind and cubicFit transport schemes. The numerical solutions are marked by solid black lines. The analytic solution is marked by dotted lines. Contours are every 0.1 kg m^{-3}.

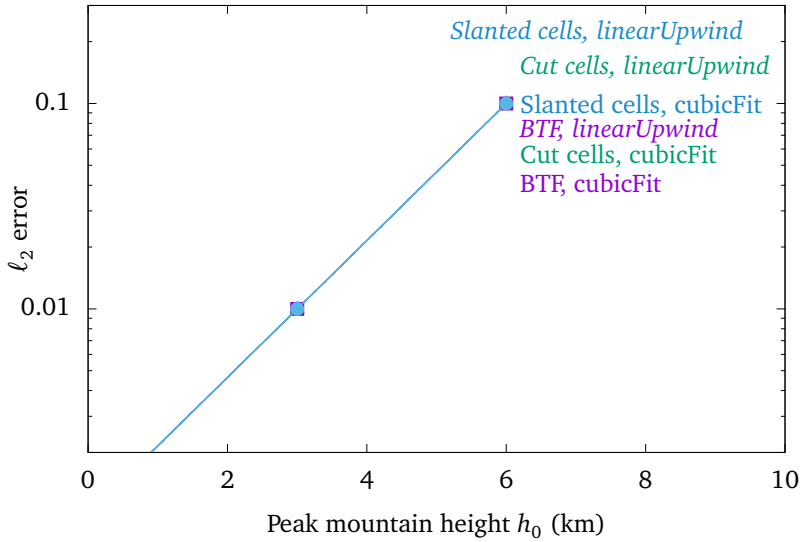


Figure 4.4: Error measures for the two-dimensional tracer transport tests over a mountainous lower boundary. Peak mountain heights h_0 are from 0 km to 6 km. Results are compared on BTF, cut cell and slanted cell meshes using the linearUpwind and the cubicFit schemes. At $h_0 = 0$ km the terrain is entirely 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.

Numerical stability limits of the cubicFit transport scheme on different mesh types

A final series of tests were performed to determine the stability limit of the cubicFit scheme with the two-stage Heun time-stepping scheme (TODO: *equation 3.2*). The tracer was transported on BTF, slanted cell and cut cell meshes with a variety of mesh spacings between $\Delta x = 5000$ m and $\Delta x = 125$ m. Δz was chosen so that a constant aspect ratio is preserved such that $\Delta x / \Delta z = 2$. For each test, the time-step was increased until the result became unstable. The largest stable time-steps, Δt_{max}, are presented in figure 4.5a. BTF meshes permit the longest time-steps of all three mesh types since cells are almost uniform in volume. As expected, the longest stable time-step scales linearly with BTF mesh spacing. There is no such linear scaling on cut cell meshes because these meshes can have arbitrarily small cells. The time-step constraints on cut cell meshes are the most severe of the three mesh types. Slanted cell meshes have a slightly stronger time-step constraint than BTF meshes but still exhibit similar linear scaling with mesh spacing. TODO: *Furthermore, a dynamical model that uses slanted cell meshes instead of BTF meshes is expected to calculate pressure gradients more accurately... which is going to be the next test (resting atmosphere)*

Figure 4.5b presents the largest stable maximum Courant numbers, max(Co), which were calculated by substituting $\Delta t = \Delta t_{\text{max}}$ into TODO: *equation* (3.4). On basic terrain following meshes, the maximum Courant number tends towards about 1.3 with finer mesh spacings. No

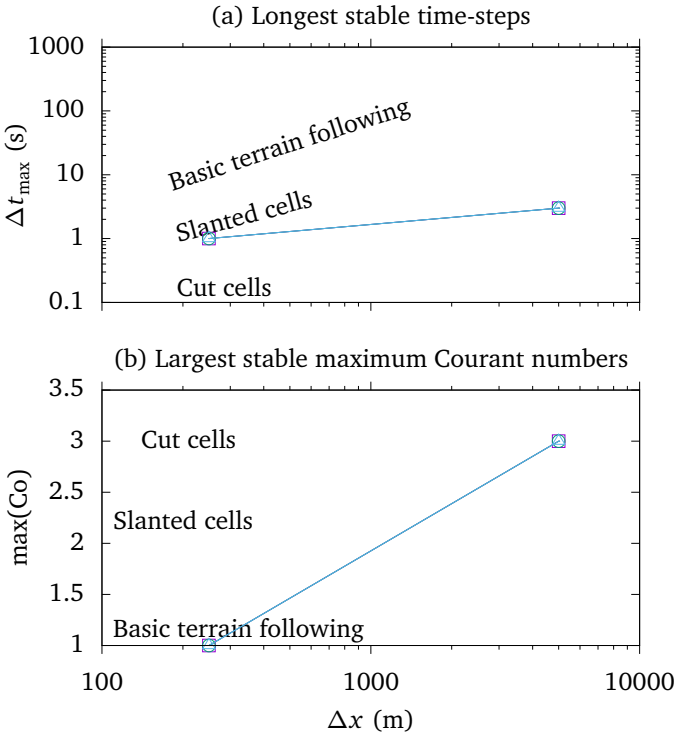


Figure 4.5: (a) Longest stable time-steps, $\Delta t_{\max}$, and (b) largest stable maximum Courant numbers, max(Co), for the two-dimensional tracer transport test over a mountainous lower boundary. Results were obtained on basic terrain-following, cut cell and slanted cell meshes at mesh spacings between $\Delta x = 5000\,\mathrm{m}$ and $\Delta x = 250\,\mathrm{m}$. The largest stable maximum Courant numbers were calculated from the corresponding longest stable time-steps using equation (3.4).

such trend is found on cut cell or slanted cell meshes. Cut cell meshes permit the largest maximum Courant numbers of around 3, but the largest stable time-steps on cut cell meshes are still smaller than corresponding time-steps on basic terrain following and slanted cell meshes.

This paper focuses on the spatial discretisation of the cubicFit scheme, but the stability limit depends also upon the choice of time-stepping. As such, we have not calculated a theoretical Courant number limit, although such an analysis should be possible using the techniques in (Baldauf, 2008).

4.2 Stratified atmosphere initially at rest

TODO:

- *Meshes: BTF, cut cell, slanted cell (with upward vertex shifts), SLEVE?*
- *Schemes: cubicFit, linearUpwind?*
- *Plot: time series of max(abs(w)) for $h_0 = 1\,\mathrm{km}$?*

- *Plot: time-maximum of max(abs(w)) versus h_0*
- *Conclusion: slanted cells have spurious velocities comparable to cut cells, much better than BTF, somewhat better than SLEVE*

An idealised 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 velocities which become more severe over steeper terrain (Klemp, 2011). TODO: *Cut cell meshes are often suggested as a technique for reducing these spurious circulations (?Jebens et al., 2011; Good et al., 2014).*

The test setup follows the specification by Klemp (2011). The domain is 200 km wide and 20 km high, and the mesh spacing is $\Delta x = \Delta z^\star = 500\,\mathrm{m}$. All boundary conditions are no normal flow.

The wave-shaped mountain profile has a surface height, h, given by

$$h(x) = h_0 \exp\left(-\left(\frac{x}{a}\right)^2\right)\cos^2(\alpha x) \tag{4.7}$$

where $a = 5\,\mathrm{km}$ is the mountain half-width, $h_0 = 1\,\mathrm{km}$ is the maximum mountain height and $\lambda = 4\,\mathrm{km}$ is the wavelength. For the optimised SLEVE mesh, the large-scale component h_1 is specified as

$$h_1(x) = \frac{1}{2}h_0 \exp\left(-\left(\frac{x}{a}\right)^2\right) \tag{4.8}$$

and, following Leuenberger et al. (2010), $s_1 = 4\,\mathrm{km}$ is the large scale height, $s_2 = 1\,\mathrm{km}$ is the small scale height, and the optimal exponent value of $n = 1.35$ is used.

The initial potential temperature field has $\theta(z = 0) = 288\,\mathrm{K}$ and a constant static stability with Brunt-Väisälä frequency $N = 0.01\,\mathrm{s}^{-1}$ everywhere, except for a more stable layer of $N = 0.02\,\mathrm{s}^{-1}$ between $2\,\mathrm{km} \le z \le 3\,\mathrm{km}$. The Exner function of pressure is calculated so that it is in discrete hydrostatic balance in the vertical direction (Weller and Shahrokhi, 2014). The damping function, μ, is set to $0\,\mathrm{s}^{-1}$. Unlike Klemp (2011), there is no eddy diffusion in the equation set.

The test was integrated forward by 5 hours using a timestep $\Delta t = 100\,\mathrm{s}$ on the BTF, TODO: *SLEVE*, cut cell and slanted cell meshes. TODO: *Maximum vertical velocities are presented in figure ??. In agreement with Klemp (2011), vertical velocities are larger on more distorted meshes. However, magnitudes are smaller comparing results on the terrain following meshes with those from Klemp (2011). The results presented in figure ??, which use a curl-free pressure gradient, have maximum spurious values of w of* $0.33\,\mathrm{m\,s}^{-1}$ *on the BTF mesh, compared with a maximum of* $\sim 7\,\mathrm{m\,s}^{-1}$ *found by Klemp (2011) using their improved horizontal pressure gradient formulation. The results in figure ?? have the same maximum errors as Weller and Shahrokhi (2014) but, due to the more stable split into implicitly and explicitly treated terms (described in the appendix), the errors decay over time due to the dissipative nature of the advection scheme.*

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

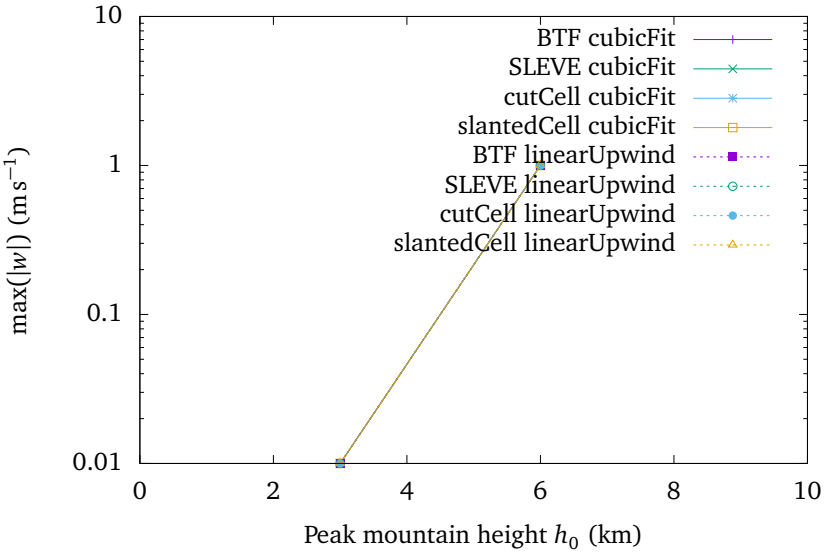


Figure 4.6: TODO:

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. This would account for the very small velocities found by Good et al. (2014).

TODO: *In summary, spurious velocities in the resting atmosphere test were similar on both terrain following meshes, with lower errors compared to those from Klemp (2011). The maximum vertical velocity was significantly decreased on the cut cell mesh, so we conclude that non-orthogonality, or lack of alignment of the mesh with surfaces of constant gravitational potential are a significant cause of numerical error in this test.*

4.3 Schär mountain waves

TODO:

- *Meshes: BTF, cut cell, slanted cell*
- *Schemes: linearUpwind, cubicFit*
- *Plot: incorrect w contours using linearUpwind, compare with Melvin et al. 2010*
- *Plot: theta diff heatmap*
- *Plot: schaerWaves and thermalAdvect vertical cross-sections of theta diff*

- *Conclusion: unlike linearUpwind, cubicFit gives the correct solution (at least on BTF? is linearUpwind better on cut/slanted cells?)*
- *Conclusion: should we test max(dt) again? we've not yet done so for a dynamical test. I'd hope that slanted cells are much better than cut cells.*

4.4 Transporting a thermal profile in a terrain-following velocity field

TODO:

- *Meshes: BTF, cut cell, slanted cell*
- *Schemes: cubicFit*
- *Plot: theta diff heatmap*
- *Conclusion: schaerWaves errors due to theta cubicFit advection errors*

5 Generalising the Charney–Phillips staggering for arbitrary meshes

6 Conclusion

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 3.5, it is necessary for velocity fields and mesh geometries to be expressed in these global Cartesian coordinates.

Velocity field specification

The non-divergent velocity field in section 3.5 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 $r_1 = R_e - 1000\,\text{m}$ and an outer radius $r_2 = R_e + 1000\,\text{m}$. By default, OpenFOAM meshes comprise polyhedral cells with straight edges and flat faces. This is problematic for spherical meshes because face areas and cell volumes are too small. For tests on a spherical Earth, we override the default configuration and calculate our own face areas, cell volumes, face centres and cell centres that account for the mesh curvature. Note that the new centres are no longer centroids, but they are consistent with the horizontal transport tests on a sphere presented in section 3.5.

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

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.

CFD Direct, 2016: OpenFOAM user guide: Numerical schemes. http://cfd.direct/openfoam/user-guide/fvschemes/.

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.

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.

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

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.

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.

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.

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., 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.

Leonard, B., M. MacVean, and A. Lock, 1993: Positivity-preserving numerical schemes for multidimensional 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.

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.

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.

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.

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.

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.

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

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.

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.