A finite volume transport scheme for atmospheric flows over steep terrain
TODO: *(working title)*

James Shaw[a,*], Hilary Weller[a]

[a]*Department of Meteorology, University of Reading, Reading, United Kingdom*

Abstract

TODO: *abstract*

Keywords: TODO: *keywords*

1. Introduction

First, we present a multidimensional advection scheme that is computationally cheap and suitable for complex flows on a variety of meshes. Second, we present a new type of Cartesian mesh, the slanted cell mesh, that avoids the small cell problem associated with cut cell meshes. We apply the advection scheme to tests over steep orography and show that accurate results are obtained on slanted cell meshes. Finally, we challenge the multidimensional advection scheme using a test of deformational flow on a geodesic mesh.

2. Types of meshes

2.1. Meshes for representing terrain

TODO: *describe BTF, cut cell method and slanted cell method. Cite ASAM for the cut cell method. Describe slanted cell method and cite [8].*

$$z = (H - h)\left(z^\star / H\right) + h \tag{1}$$

2.2. Meshing a spherical Earth

TODO: *Describe icosahedral meshes (cite one or more of [9, 2, 3])*
TODO: *Describe cubed-sphere mesh (cite someone?)*

Following appendix A9 in [5], the average equatorial spacing $\Delta\lambda$ is used as a measure of mesh spacing. It is defined as

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

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.

*Corresponding author
Email address: js102@zepler.net (James Shaw)

3. Transport schemes for arbitrary meshes

The cubicFit transport scheme is described here for arbitrary two-dimensional meshes and arbitrary, single-layer spherical meshes. Section 4 compares results using the cubicFit scheme with results using the linearUpwind transport scheme, and so a description of the linearUpwind scheme is also provided here.

TODO: *move the derivation from the advection equation and timestepping details to here since they're common to both cubicFit and linearUpwind*

TODO: *somewhere in here we need to state the multidimensional Courant number and note that the Courant number may vary between cells belonging to the same mesh. We should then say that the stability limit for both schemes is* $\max \mathrm{Co} \leq 1$.

3.1. cubicFit transport scheme

TODO: *I haven't said exactly how we do 2D stencils on the sphere: how local coordinates are established, projection onto a local 2D plane*

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

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{u}\phi) = 0 \tag{3}$$

The time derivative is discretised using a three-stage, second-order Runge-Kutta scheme:

$$\phi^{\star} = \phi^{(n)} + \Delta t f(\phi^{(n)}) \tag{4a}$$

$$\phi^{\star\star} = \phi^{(n)} + \frac{\Delta t}{2}\left(f(\phi^{(n)}) + f(\phi^{\star})\right) \tag{4b}$$

$$\phi^{(n+1)} = \phi^{(n)} + \frac{\Delta t}{2}\left(f(\phi^{(n)}) + f(\phi^{\star\star})\right) \tag{4c}$$

where $f(\phi^{(n)}) = -\nabla \cdot (\mathbf{u}\phi^{(n)})$ at time level n.

Using the finite volume method, the wind field is prescribed at face centroids and the dependent variable is stored at cell centroids. The divergence term in equation (3) is discretised using Gauss's theorem:

$$\nabla \cdot (\mathbf{u}\phi) \approx \frac{1}{\mathcal{V}_c} \sum_{f \in c} \mathbf{u}_f \cdot \mathbf{S}_f \phi_F \tag{5}$$

where $\mathcal{V}_c$ is the cell volume, $\mathbf{u}_f$ is a wind 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, and $\sum_{f \in c}$ denotes a summation over all faces f bordering cell c. The value of the dependent variable at the face, ϕ_F, is approximated by a least squares fit over a stencil of surrounding cell centre values. The approximation method to calculate ϕ_F is described here.

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 wind direction at the face determines which of the two adjacent cells is the upwind cell. Since the stencil is upwind-biased, two stencils must be constructed for every interior face, and the appropriate stencil is chosen for each face depending on the wind direction at that face for every timestep.

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

$$\phi = a_1 + a_2 x + a_3 y + a_4 x^2 + a_5 xy + a_6 y^2 + a_7 x^3 + a_8 x^2 y + a_9 x y^2 \tag{6}$$

A least squares approach is needed because the system of equations is overconstrained, with 12 stencil values but only 9 polynomial terms. If the stencil geometry is expressed in a local coordinate system with the face centroid as the origin, then the approximated value ϕ_F is equal to the constant coefficient a_1.

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.

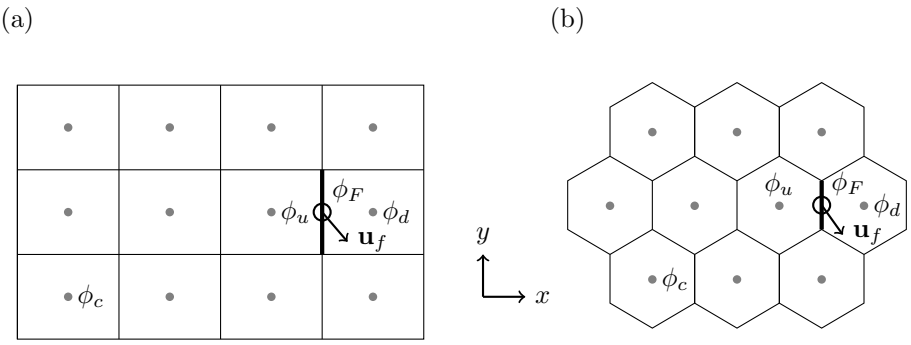

Figure 1: Upwind-biased stencils for faces far away from the boundaries of two-dimensional (a) rectangular and (b) hexagon meshes. The stencil is used to fit a multidimensional polynomial to cell centre values, ϕ_c, marked by grey circles, in order to approximate the value ϕ_F at the face centroid marked 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 wind vector $\mathbf{u}_f$ is prescribed at face f and determines the choice of stencil at each timestep.

3.1.1. Stencil construction

For every interior face, two stencils are constructed, one for each of the possible upwind cells. For a given 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

$$\mathrm{Opp}(f, g) \equiv -\frac{\mathbf{S}_f \cdot \mathbf{S}_g}{|\mathbf{S}_f|^2} \tag{7}$$

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 (7) as

$$\mathrm{Opp}(f, g) = -\frac{|\mathbf{S}_g|}{|\mathbf{S}_f|}\cos(\theta) \tag{8}$$

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

$$\mathrm{OF}(f, c_u) \equiv \{g : \mathrm{Opp}(f, g) \geq 0.5\} \cup \{g : \max_{g \in G}(\mathrm{Opp}(f, g))\} \tag{9}$$

On a rectangular mesh, there is always one opposing face that is exactly parallel to the face f.

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. Having found these two sets of cells, the stencil is constructed to comprise all internal and external cells.

Figure 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. The resultant stencil contains 13 cells.

3.1.2. Least squares fit

To approximate the value at a face f, a least squares fit is calculated from a stencil of surrounding cell centre values. First, we will show how a polynomial least squares fit is calculated for a face on a rectangular mesh. Second, we will

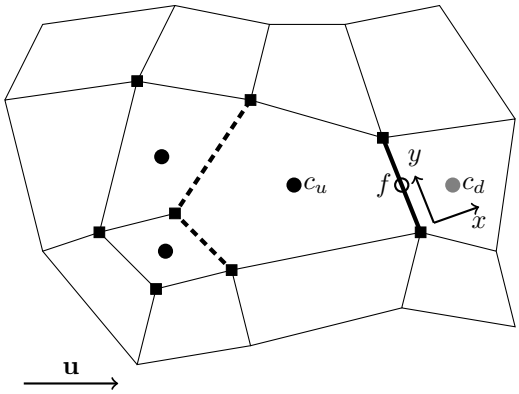

Figure 2: A thirteen-cell, upwind-biased stencil for face f connecting the pentagonal upwind cell, c_u, and the downwind cell c_d. The dashed lines denote the two faces of cell c_u that oppose f, and black circles mark the centroids of the internal cells that are connected to these two opposing faces. The stencil is extended outwards by including cells that share vertices with the three internal cells, where black squares mark these vertices. 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.

make modifications to the least squares fit that are necessary for numerical stability. Finally, we will describe how the approach is applicable to faces of arbitrary meshes.

For faces that are far away from the boundaries of a rectangular mesh, we fit the multidimensional polynomial given by equation (6) that has nine unknown coefficients, $\mathbf{a} = a_1 \dots a_9$, using the twelve cell centre values from the upwind-biased stencil, $\boldsymbol{\phi} = \phi_1 \dots \phi_{12}$. This yields a matrix equation

$$\begin{bmatrix} 1 & x_1 & y_1 & x_1^2 & x_1y_1 & y_1^2 & x_1^3 & x_1^2y_1 & x_1y_1^2 \\ 1 & x_2 & y_2 & x_2^2 & x_2y_2 & y_2^2 & x_2^3 & x_2^2y_2 & x_2y_2^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{12} & y_{12} & x_{12}^2 & x_{12}y_{12} & y_{12}^2 & x_{12}^3 & x_{12}^2y_{12} & x_{12}y_{12}^2 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_9 \end{bmatrix} = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{12} \end{bmatrix} \tag{10}$$

which can be written as

$$\mathbf{Ba} = \boldsymbol{\phi} \tag{11}$$

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

$$\mathbf{a} = \mathbf{B}^+\boldsymbol{\phi} \tag{12}$$

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

$$a_1 = \begin{bmatrix} b_{1,1}^+ \\ b_{1,2}^+ \\ \vdots \\ b_{1,12}^+ \end{bmatrix} \cdot \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{12} \end{bmatrix} \tag{13}$$

where the weights $b_{1,1}^+ \dots b_{1,12}^+$ are the elements of the first row of $\mathbf{B}^+$.

In the least squares fit presented above, all stencil values contributed equally to the polynomial fit. Lashley [4] showed that it is necessary for numerical stability that the polynomial fits the cells connected to face f more closely

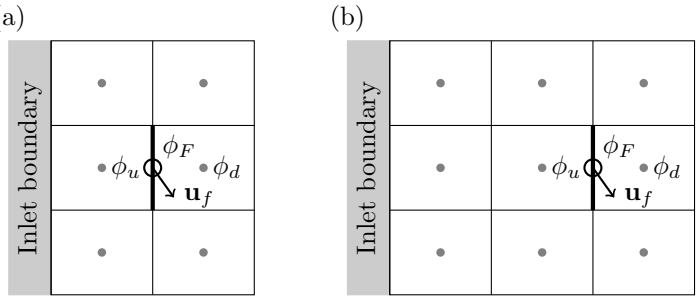

Figure 3: Upwind-biased stencils for faces near the left-hand boundary of a rectangular mesh, with (a) a 2×3 stencil for the face immediately adjacent to the left-hand boundary, and (b) a 3×3 stencil for the face immediately adjacent to the face in (a). For both stencils, attempting a least squares fit using the nine-term polynomial in equation (6) would result in an underconstrained problem.

than other cells in the stencil. 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 by equation (11) to obtain

$$\tilde{\mathbf{B}}\mathbf{a} = \mathbf{m} \cdot \boldsymbol{\phi} \tag{14}$$

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

$$a_1 = \tilde{\mathbf{b}}_\mathbf{1}^+ \cdot \mathbf{m} \cdot \phi \tag{15}$$

where $\tilde{\mathbf{b}}_\mathbf{1}^+ = \tilde{b}_{1,1}^+ \dots \tilde{b}_{1,12}^+$ are the elements of the first row of $\tilde{\mathbf{B}}^+$. Again, a_1 is a weighted mean of $\boldsymbol{\phi}$, where the weights are now $\tilde{\mathbf{b}}_\mathbf{1}^+ \cdot \mathbf{m}$. Note that, once $\tilde{\mathbf{b}}_\mathbf{1}^+$ is calculated for each stencil, only the first row needs to be stored.

For faces of a non-rectangular mesh, or faces that are near a boundary, the number of stencil cells and number of polynomial terms may differ: a stencil will have two 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.

3.1.3. Polynomial generation

The majority of faces on a uniform two-dimensional mesh have stencils with more than nine cells. For example, a rectangular mesh has 12 points (figure 1a), and a hexagonal mesh has 10 points (figure 1b). In all three cases, constructing a system of equations using the nine-term polynomial in equation (6) 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 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 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 between one and nine terms and includes a combination of the terms in equation (6). There are two additional constraints that a candidate polynomial satisfy.

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

$$M(x, y) = x^i y^j : i, j \geq 0 \text{ and } i + j \leq 3 \tag{16}$$

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 $x^a y^b$ and $x^c y^d$ are in S with $a \leq c$ and $b \leq d$, then $x^i y^j$ is also in S for all $a < i < c$, $b < j < d$. 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.

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 3b results in a rank-deficient matrix and so the nine-term polynomial would not be a candidate polynomial.

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

3.1.4. Stabilisation procedure

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

The approximated value ϕ_F is equal to a_1 which is calculated from equation (15). The value of a_1 is a weighted mean of $\boldsymbol{\phi}$ where $\mathbf{w} = \tilde{\mathbf{b}}_{\mathbf{1}}^{+} \cdot \mathbf{m}$ 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 departures from the smooth field $\boldsymbol{\phi}$.

A one-dimensional von Neumann analysis was performed to obtain stability constraints on the weights $\mathbf{w}$. The analysis is presented in the appendix, and it shows that the weights must satisfy three constraints:

$$0.5 \leq w_u \leq 1 \tag{17a}$$

$$0 \leq w_d \leq 0.5 \tag{17b}$$

$$w_u - w_d \geq \max_{p \in P}(|w_p|) \tag{17c}$$

where w_u and w_d are the weights for the upwind and downwind cells respectively. The *peripheral cells* 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 cell p.

The stabilisation procedure comprises three steps. In the first step, the candidate polynomials are sorted in preference order so that candidates with the most terms are preferred over those with fewer terms. If there are multiple candidates with the same number of terms, the candidate with the largest minimum singular value is preferred. This ordering ensures that the preferred candidate is the highest-order polynomial with the most information content.

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 cells have multipliers $m_p = 1$. These multipliers are very similar to those used by Lashley [4], 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 $\mathbf{w}$ from the least squares fit and evaluate them against the stability constraints given in equation (17). If any constraint is violated, the value of m_d is halved and the constraints are evaluated with the new weights. This step is repeated until the weights satisfy the stability constraints, or m_d becomes smaller than one. In practice, the constraints are satisified when m_d is either small (between 1 and 4) or equal to 2^{10}. If the constraints are still not satisfied, then we start again from the second step with the next-preferred 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 fact, we have not encountered any stencil for which this last resort is required.

To illustrate the stabilisation procedure, figure 4a presents a one-dimensional example of a cubic polynomial fitted through five points, with the weight at each point printed above it. In preference order, the candidate polynomials are

$$\phi = a_1 + a_2 x + a_3 x^2 + a_4 x^3 \tag{18}$$

$$\phi = a_1 + a_2 x + a_3 x^2 \tag{19}$$

$$\phi = a_1 + a_2 x \tag{20}$$

$$\phi = a_1 \tag{21}$$

We begin with the cubic equation. 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 constraint on the

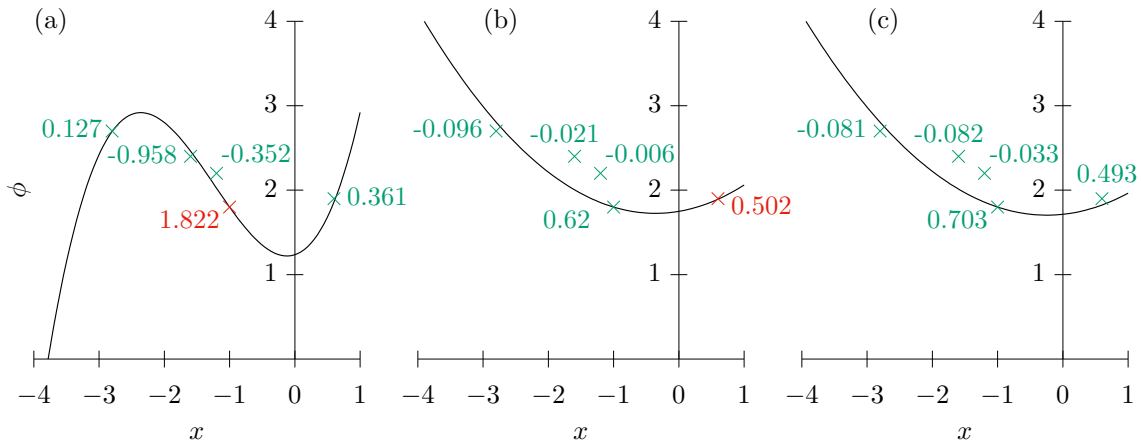

Figure 4: A one-dimensional least squares fits to a stencil of five points using (a) a cubic polynomial with multipliers $m_u = 1024$, $m_d = 1024$ and $m_p = 1$, (b) a quadratic polynomial with the same multipliers, and (c) a quadratical polynomial with multipliers $m_u = 1024$, $m_d = 1$ and $m_p = 1$. 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 constraints in equation (17) are marked in red. 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 $(-1.2, 2.2)$.

Figure 5: TODO: *I'll probably need a figure that shows linearUpwind's stencil and shows how it approximates ϕ_F*

upwind point is violated because $w_u = 1.822 > 1$ (equation 17a). Reducing the downwind multiplier does not help to satisfy the constraint, so we start again with the quadratic equation (figure 4b). Again, the multipliers are chosen to force the polynomial through the upwind and downwind points, but this violates the constraint on the downwind point because $w_d = 0.502 > 0.5$ (equation 17b). This time, however, stable weights are found by reducing w_d to one (figure 4c) and these are the weights that will be used to approximate ϕ_F, where the polynomial intercepts the y-axis.

3.2. linearUpwind transport scheme

TODO: *describe OpenFOAM's linearUpwind scheme and cite something? OpenFOAM docs? OpenFOAM github?*
TODO: *I need to mention that we use a Dirichlet boundary condition of $\phi = 0\,\mathrm{kg\,m^{-3}}$ at the ground. Results are unstable if I use zero gradient boundary condition.*

4. Results

TODO: *the 'slug' advection test case makes slanted cells look bad compared to BTF. should we redress the balance with a different test?* TODO: *should I use RK4 or will RK2 suffice?*
TODO: *somewhere mention that the second-order convergence is a limitation of the divergence discretisation. With more DoF a higher order should be achievable.*
TODO: *should I do eigenmode analysis? this would prove stability for particular meshes for arbitrary wind fields*
TODO: *somewhere define error norms*

4.1. Horizontal transport with mesh refinement

Coping with sudden changes in mesh spacing is necessary for some types of mesh refinement and mesh adaptivity. We should use the horizontal advection test from Schär et al. [7] on a BTF mesh with a 2x/4x/8x refined mesh in part of the mountainous region. The tracer will have to pass into and out of this refined region. This test will also help to familiarise the reader with this standard test. We will modify parts of this test in the following subsection in order to test advection over the lower boundary.

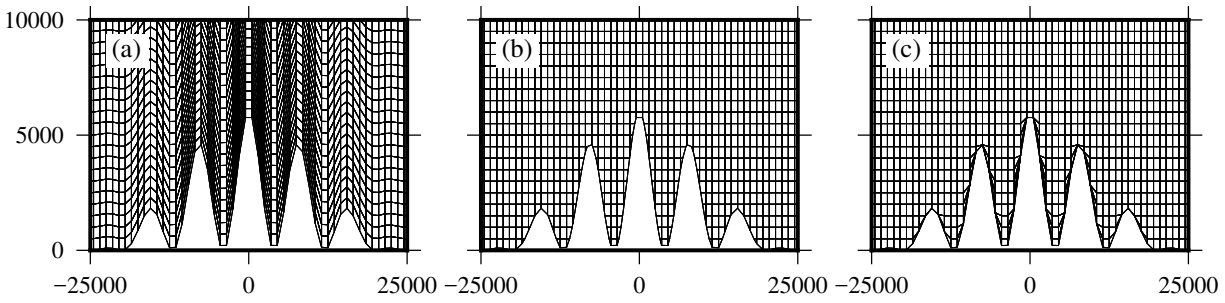

Figure 6: Cell edges of (a) basic terrain-following, (b) cut cell, and (c) slanted cell two-dimensional x–z meshes used for the tracer transport tests in section 4.2. Only the lowest 10 km for the central region of the domain in shown. The entire domain is 301 km wide and 25 km high.

TODO: *I tried this with horizontal advection, no orography, with 4x mesh refinement in a region of the mesh. Results for linearUpwind and cubicFit were very similar to corresponding results without any refinement. So I'm not sure this test will be particularly revealing.*

4.2. Transport over a mountainous lower boundary

A two-dimensional transport test over mountains was developed in [7] 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 terrain. This test presents no particular challenge on cut cell meshes because there is zero velocity and zero tracer density near the ground [1]. Here we present a variation of this standard test that challenges transport schemes on all mesh types. By positioning the tracer next to the ground and modifying the velocity field, we can assess the accuracy of the cubicFit scheme near the lower boundary. Results using the cubicFit scheme are compared with the linearUpwind scheme on basic terrain-following, cut cell and slanted cell meshes.

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. The domain is subdivided into a 301×50 mesh such that $\Delta x = 1\,\text{km}$ and $\Delta z = 500\,\text{m}$. A boundary condition of $\phi = 0\,\text{kg}\,\text{m}^{-3}$ is imposed at the inlet, ground and top boundaries, and the outlet boundary is open with $\partial\phi/\partial x = 0\,\text{kg}\,\text{m}^{-4}$. TODO: *the outlet boundary is currently fixedValue=0! this needs fixing*

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

$$h(x) = h^\star \cos^2(\alpha x) \tag{22a}$$

where

$$h^\star(x) = \begin{cases} h_0 \cos^2(\beta x) & \text{if } |x| < a \\ 0 & \text{otherwise} \end{cases} \tag{22b}$$

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 [7].

Basic terrain-following, cut cell and slanted cell meshes are constructed by modifying the uniform 301×50 mesh using this terrain profile. The details of the various mesh generation methods were given in section 2.1. Cell edges in the central region of the domain are shown in figure 6 for each of the three mesh types. Cells in the BTF mesh are highly distorted over steep slopes (figure 6a) while the cut cell mesh (figure 6b) and slanted cell mesh (figure 6c) are orthogonal everywhere except for cells nearest the ground.

A velocity field is chosen so that velocities are everywhere tangential to the basic terrain-following coordinate surfaces given by equation (1). This velocity field ensures that there is no normal flow at the lower boundary. A streamfunction Ψ is used so that the discrete velocity field is non-divergent, such that

$$\Psi(x, z) = -u_0 H \frac{z - h}{H - h} \tag{23}$$

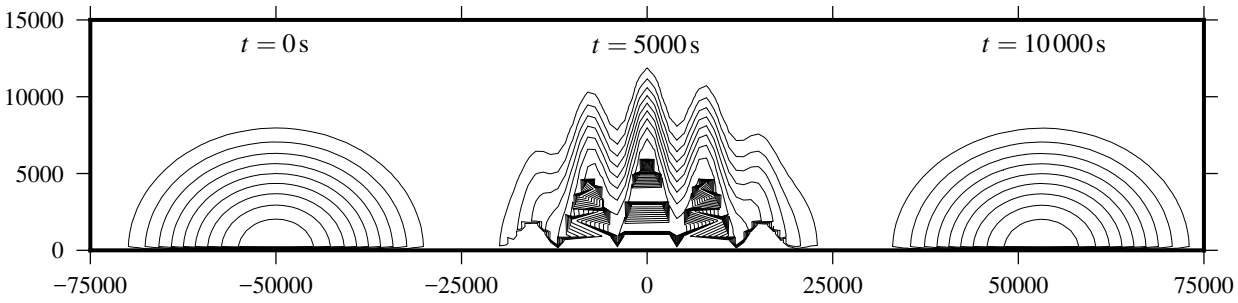

Figure 7: Evolution of the tracer in the two-dimensional transport test over steep terrain. The tracer is transported to the right over the wave-shaped terrain. Tracer contours are every 0.1 kg m^{-3}. TODO: *try to mask off below the mountain to hide the messy contouring*

where $u_0 = 10\,\mathrm{m\,s^{-1}}$, which is the horizontal velocity where $h(x) = 0$. The horizontal and vertical components of velocity, u and w, are then given by

$$u = -\frac{\partial \Psi}{\partial z} = u_0 \frac{H}{H-h}, \quad w = \frac{\partial \Psi}{\partial x} = u_0 H \frac{\mathrm{d}h}{\mathrm{d}x} \frac{H-z}{(H-h)^2} \tag{24}$$

$$\frac{\mathrm{d}h}{\mathrm{d}x} = -h_0 \left[\beta \cos^2(\alpha x) \sin(2\beta x) + \alpha \cos^2(\beta x) \sin(2\alpha x)\right] \tag{25}$$

Unlike the horizontal transport test in [7], the velocity field presented here extends from the top of the domain all the way to the ground.

At $t = 0\,\mathrm{s}$, a tracer with density ϕ is positioned upwind of the mountain at the ground. It has the shape

$$\phi(x,z) = \phi_0 \begin{cases} \cos^2\left(\frac{\pi r}{2}\right) & \text{if } r \le 1 \\ 0 & \text{otherwise} \end{cases} \tag{26}$$

with radius r given by

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

where $A_x = 25\,\mathrm{km}$, $A_z = 10\,\mathrm{km}$ 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 $(x_0, z_0) = (-50\,\mathrm{km}, 0\,\mathrm{km})$ so that the tracer is upwind of the mountain and centred at the ground.

Tests are integrated forward for 10 000 s, by which time the tracer has moved downwind of the mountain. TODO: *state timesteps for BTF/cut cell/slanted cell* 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:

$$\mathrm{d}t = \mathrm{d}x / u(x) \tag{28}$$

$$t = \int_0^x \frac{H - h(x)}{u_0 H}\,\mathrm{d}x \tag{29}$$

$$t = \frac{x}{u_0} - \frac{h_0}{16 u_0 H}\left[4x + \frac{\sin 2(\alpha+\beta)x}{\alpha+\beta} + \frac{\sin 2(\alpha-\beta)x}{\alpha-\beta} + 2\left(\frac{\sin 2\alpha x}{\alpha} + \frac{\sin 2\beta x}{\beta}\right)\right] \tag{30}$$

This equation is solved numerically to find that $x(t = 10\,000\,\mathrm{s}) = 51\,577.4\,\mathrm{m}$.

Tracer contours at the initial time $t = 0\,\mathrm{s}$, half-way time $t = 5000\,\mathrm{s}$, and end time $t = 10\,000\,\mathrm{s}$ are shown in figure 7 using the linearUpwind scheme on the BTF mesh. As apparent at $t = 5000\,\mathrm{s}$, the tracer is distorted by the

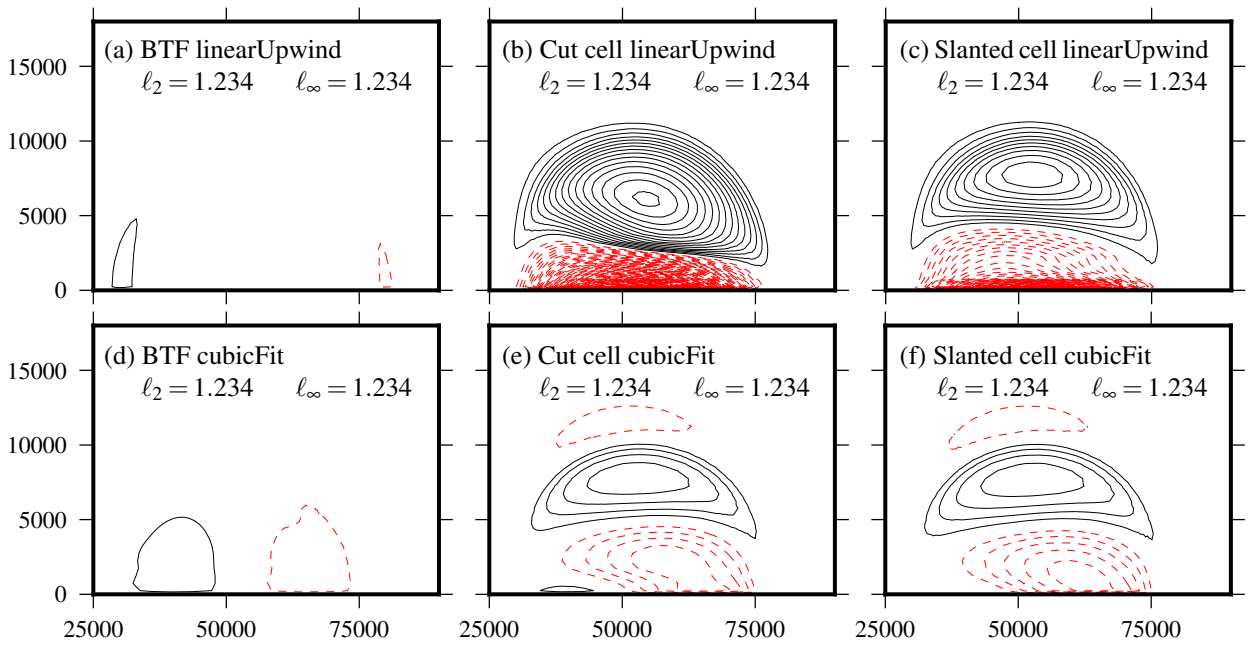

Figure 8: Error contours at $t = 10\,000$ s for the two-dimensional tracer transport tests. Results are presented on BTF, cut cell and slanted cell meshes (shown in figure 6) using the linearUpwind and cubicFit transport schemes. Error contours are every 0.01 kg m^{-3} with solid black lines marking positive errors and dashed red lines marking negative errors.

terrain-following velocity field as it passes over the mountain, but its original shape is restored once it has cleared the mountain by $t = 10\,000$ s.

Numerical errors are calculated by subtracting the analytic solution from the numerical solution. Errors are compared between BTF, cut cell and slanted cell meshes using the linearUpwind scheme (figures 8a, 8b and 8c respectively) and the cubicFit scheme (figures 8d, 8e and 8f respectively). Results are most accurate on the BTF mesh. This is to be expected because the velocity field is exactly aligned with the mesh [8]. Small phase errors are visible using linearUpwind (figure 8a), while slightly larger phase errors are apparent using cubicFit (figure 8d). We surmise that errors are larger using cubicFit because the scheme's larger stencil includes data from cells above and below those cells that lie along the trajectory.

Results are least accurate using the linearUpwind scheme on the cut cell mesh (figure 8b). The final tracer is distorted and does not extend far enough towards the ground. The error magnitude is reduced by using the linearUpwind scheme on the slanted cell mesh (figure 8c), but the error's shape remains the same. The cubicFit scheme is less sensitive to the choice of mesh with similar error magnitudes on the cut cell mesh (figure 8e) and slanted cell mesh (figure 8f). 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 meshe. Nevertheless, errors on the BTF mesh are at least four times smaller than errors on cut cell or slanted cell meshes using the cubicFit scheme.

Another series of tests were performed on BTF, slanted cell and cut cell meshes using 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 $\Delta x/\Delta z = 2$. In order to verify that cubicFit is accurate near the stability limit, timesteps were chosen so that the maximum Courant number was close to one. TODO: *(check this is true:) Results were accurate in all tests and the scheme was largely insensitive to the choice of timestep.*

This series of tests also enables a comparison of longest stable timesteps between mesh types. The longest stable timestep for a maximum Courant number of one can be calculated as $\Delta t_{\max} = \Delta t / \max \mathrm{Co}$ where Δt is the timestep used in a particular test run and max Co is the maximum Courant number for that test run. TODO: *For example, on the BTF mesh with* $\Delta x = 123$ m *and* $\Delta t = 456$ s *the maximum Courant number was* max Co = 789. *The longest stable timestep is then* $\Delta t_{\max} = 345$ s.

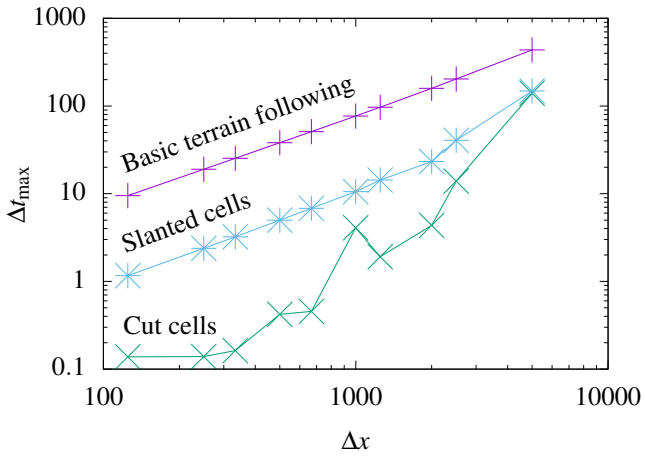

Figure 9: Longest stable timesteps, $\Delta t_{\max}$, for the two-dimensional tracer transport test on basic terrain-following, cut cell and slanted cell meshes at mesh spacings between $\Delta x = 5000$ m and $\Delta x = 125$ m. The tests were integrated with a maximum Courant number close to 1, while $\Delta t_{\max}$ is calculated as the timestep corresponding to a maximum Courant number of exactly 1. TODO: *I have only calculated the numbers for this plot, I haven't yet run all the tests themselves at Co close to one. I have at least run some tests at Co* ≈ 1 *and there were no issues.*

The longest stable timesteps for BTF, cut cell and slanted cell meshes are presented in figure 9. BTF meshes permit the longest timesteps of all three meshes since cells are almost uniform in volume, and the velocity field coincides with mesh layers so that cells are maximally long in the direction of flow. As expected, the longest stable timestep 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 timestep constraints on cut cell meshes are the most severe of the three mesh types. Slanted cell meshes have a timestep constraint that is more severe than BTF meshes, but still exhibit the same, predictable linear scaling with mesh spacing.

The transport tests presented in this section demonstrate that the cubicFit scheme is suitable for flows over steep terrain on two-dimensional terrain-following and cut cell meshes. The cubicFit scheme is less sensitive to the mesh type compared to the linearUpwind scheme. In the next section, we evaluate the cubicFit scheme using more complex, deformational flows on icosahedral meshes and cubed-sphere meshes.

4.3. Deformational flow on a sphere

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 [6]. Results are compared between linearUpwind and cubicFit schemes using icosahedra and cubed-spheres.

The deformational flow test in [6] comprised six elements:

1. a convergence test using a Gaussian-shaped tracer
2. a "minimal" resolution test using a cosine-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 only tests 1, 2 and 6. We do not consider filament preservation or the transport of a "rough" slotted cylinder because no shape-preserving filter has yet been developed for cubicFit. TODO: *why is tracer correlation out of scope for this paper?*

TODO: *how much detail do I need about OpenFOAM's global Cartesian coordinates, lack of 2D meshes and our correction for spherical geometry?*

4.3.1. Numerical order of convergence using Gaussian hills

The first deformational flow test uses a C^∞ 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:

$$\Psi(\lambda, \theta, t) = \frac{10R_e}{T} \sin^2(\lambda') \cos^2(\theta) \cos\left(\frac{\pi t}{T}\right) - \frac{2\pi R_e}{T} \sin(\theta) \tag{31}$$

where λ is a longitude, θ is a latitude, $T = 1.0368 \times 10^6$ s is the duration of integration, and $\lambda' = \lambda - 2\pi t/T$.

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

$$\phi = \phi_1(\lambda, \theta) + \phi_2(\lambda, \theta) \tag{32}$$

An individual hill ϕ_i is given by

$$\phi_i(\lambda, \theta) = \phi_0 \exp\left(-b\left(\frac{|\mathbf{x} - \mathbf{x}_i|}{R_e}\right)^2\right) \tag{33}$$

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

$$(x, y, z) = (R_e \cos\theta \cos\lambda, R_e \cos\theta \sin\lambda, R_e \sin\theta) \tag{34}$$

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

$$(\lambda_1, \theta_1) = (5\pi/6, 0) \tag{35}$$

$$(\lambda_2, \theta_2) = (7\pi/6, 0) \tag{36}$$

The results in figure 10 are obtained using the cubicFit scheme on TODO: *whatever high-res mesh I choose*. The initial Gaussian hills are shown in figure 10a. At $t = T/2$ the tracer has been deformed into an S-shaped filament (figure 10c). By $t = T$ the tracer has almost returned to its original distribution except for some distortion and diffusion that are the result of numerical errors (figure 10d).

4.3.2. "Minimal" resolution using cosine bells

TODO: *what motivates us to perform this test?*

The non-divergent velocity field is the same as convergence test in section 4.3.1. A quasi-smooth tracer field is defined as the sum of two cosine bells:

$$\phi = \begin{cases} b + c\phi_1(\lambda, \theta) & \text{if } r_1 < r \\ b + c\phi_2(\lambda, \theta) & \text{if } r_2 < r \\ b & \text{otherwise} \end{cases} \tag{37}$$

4.3.3. Transport under divergent flow conditions using cosine bells

5. Conclusions

The advection scheme is

- suitable for complex flows on a variety of meshes
- computationally cheap at runtime, with more expensive computations depending only on the mesh geometry
- TODO: *convergence*
- stable for Courant numbers up to 1

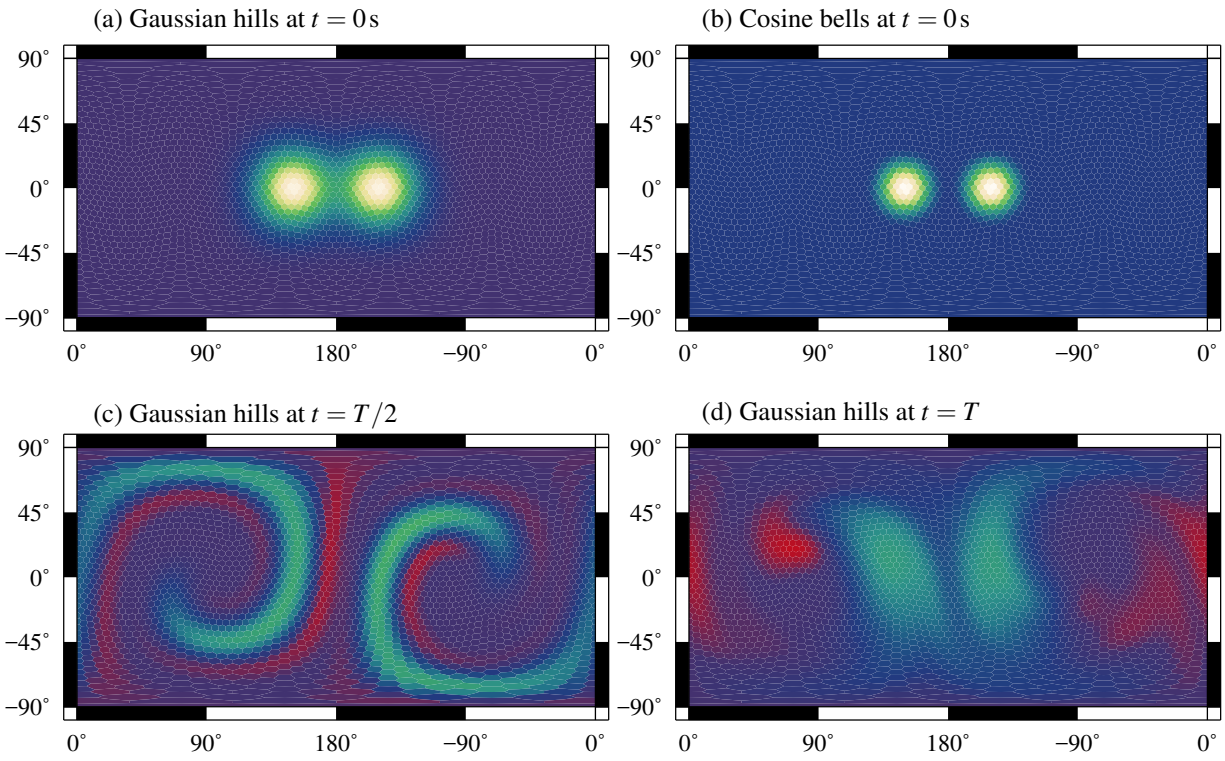

Figure 10: Tracer fields for the deformational flow test using (a) initial Gaussian hills and (b) initial cosine bells. The tracer is deformed by the velocity field before the rotation reverses to return the tracer to its original distribution: (c) by $t = T/2$ the Gaussian hills are stretched into a thin S-shaped filament; (d) at $t = T$ the tracer resembles the initial Gaussian hills except for some distortion and diffusion due to numerical errors. TODO: *plot at a high resolution using whichever mesh gives better results.*
TODO: *would it be clearer to use contours? or do heatmap plots aid intercomparison?*

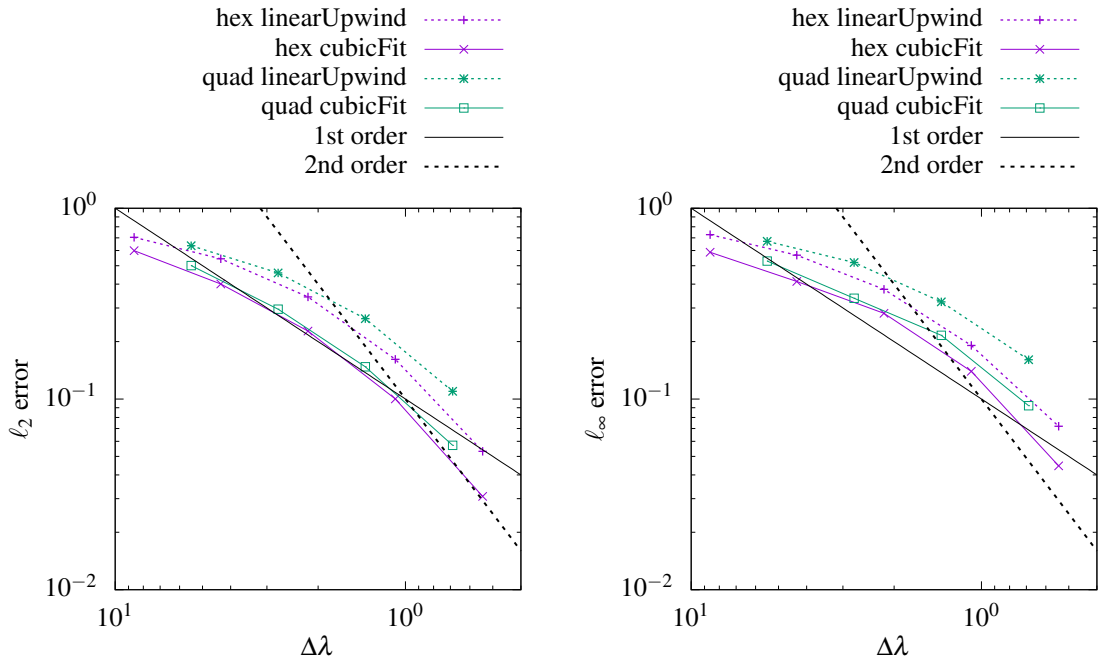

Figure 11: TODO: *deformational flow l2 and linf convergence plots comparing cubed sphere and hexagons, cubicFit and linearUpwind. This figure is comparable to Lauritzen et al. [6] figure 4.*

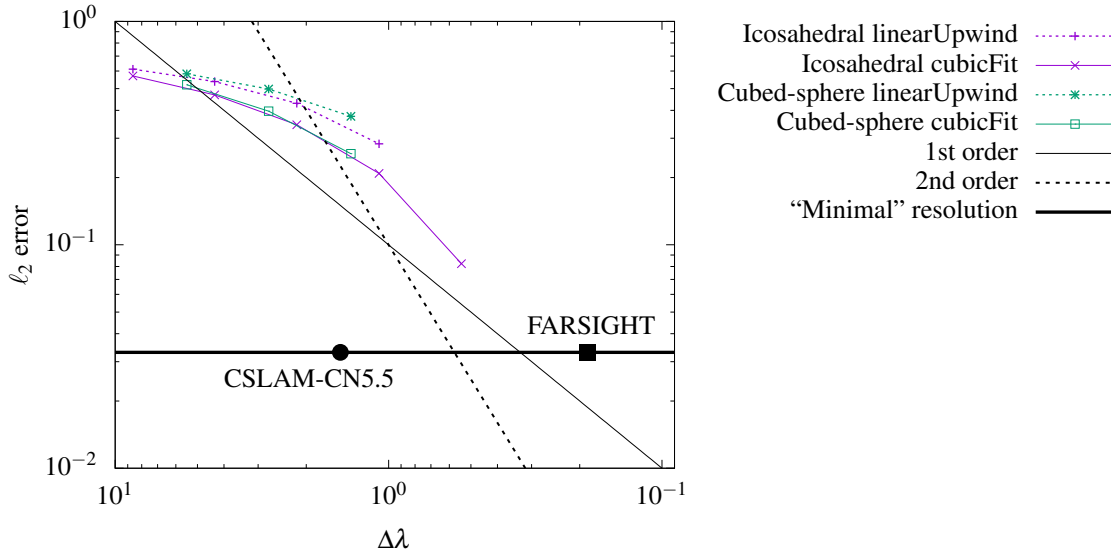

Figure 12: TODO: ℓ_2 *convergence for non-divergent deformational flow using Cosine bells. Used to find "minimal" resolution. Plot for hexagons and cubed sphere, cubicFit and linearUpwind. Plot a heavy line for minimal resolution, as in Lauritzen et al. [6] figure 5.*

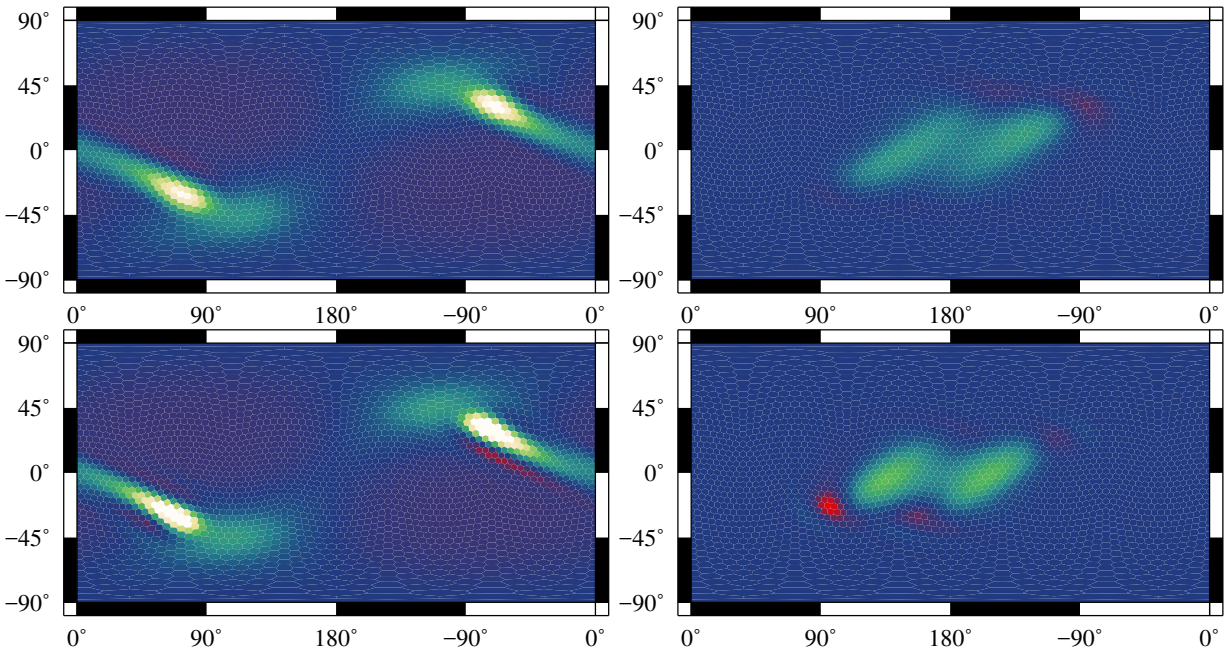

Figure 13: TODO: *divergent flow at $t = T/2$ and $t = T$ comparing cubed sphere and hexagons, cubicFit and linearUpwind. Corresponds to Lauritzen et al. [6] figure 9.*

6. Acknowledgements

TODO: *Supervisors, funding bodies. ASAM group for the mesh generator—I should ask permission to use cut cell meshes in this paper. Dr Tristan Pryer. Dr Shing Hing Man.*

Appendix: One-dimensional von Neumann stability analysis

Two analyses are performed in order to find stability constraints on the weights $\mathbf{w} = \tilde{\mathbf{b}}_1^+ \cdot \mathbf{m}$ as appear in equation (15). The first analysis uses two points to derive separate constraints on the upwind weight w_u and downwind weight w_d. The second analysis uses three points to derive a constraint that considers all weights in a stencil.

Two-point analysis

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

$$\frac{\partial \phi_j}{\partial t} = -u \frac{\phi_R - \phi_L}{\Delta x} \tag{38}$$

where the left and right fluxes, ϕ_L and ϕ_R are weighted averages of the neighbouring points. Assuming that u is positive

$$\phi_L = \alpha_u \phi_{j-1} + \alpha_d \phi_j \tag{39}$$

$$\phi_R = \beta_u \phi_j + \beta_d \phi_{j+1} \tag{40}$$

where 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 $\beta_u + \beta_d = 1$. A subscript j denotes the value at a given point $x = j\Delta x$ where Δx is a uniform mesh spacing.

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

$$\phi_j^{(n)} = A^n e^{ijk\Delta x} \tag{41}$$

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 (38)

$$\frac{\partial \phi_j}{\partial t} = \frac{\partial}{\partial t}\left(A^{t/\Delta t}\right) e^{ijk\Delta x} = \frac{\ln A}{\Delta t} A^n e^{ikj\Delta x} \tag{42}$$

hence equation (38) becomes

$$\frac{\ln A}{\Delta t} = -\frac{u}{\Delta x}\left(\beta_u + \beta_d e^{ik\Delta x} - \alpha_u e^{-ik\Delta x} - \alpha_d\right) \tag{43}$$

$$\ln A = -c\left(\beta_u - \alpha_d + \beta_d \cos k\Delta x + i\beta_d \sin k\Delta x - \alpha_u \cos k\Delta x + i\alpha_u \sin k\Delta x\right) \tag{44}$$

where the Courant number $c = u\Delta t/\Delta x$. Let $\mathfrak{R} = \beta_u - \alpha_d + \beta_d \cos k\Delta x - \alpha_u \cos k\Delta x$ and $\mathfrak{I} = \beta_d \sin k\Delta x + \alpha_u \sin k\Delta x$, then

$$\ln A = -c\left(\mathfrak{R} + i\mathfrak{I}\right) \tag{45}$$

$$A = e^{-c\mathfrak{R}} e^{-ic\mathfrak{I}} \tag{46}$$

and the complex modulus and complex argument of A are found to be

$$|A| = e^{-c\mathfrak{R}} = \exp\left(-c\left(\beta_u - \alpha_d + (\beta_d - \alpha_u)\cos k\Delta x\right)\right) \quad \text{and} \tag{47}$$

$$\arg(A) = -c\mathfrak{I} = -c\left(\beta_d + \alpha_u\right)\sin k\Delta x \tag{48}$$

For stability, we need $|A| \leq 1$ and for advection in the correct direction we need $\arg(A) < 0$ for $c > 0$, so

$$\beta_u - \alpha_d + (\beta_d - \alpha_u)\cos k\Delta x \geq 0 \quad \forall k\Delta x \quad \text{and} \tag{49}$$

$$\beta_d + \alpha_u > 0 \tag{50}$$

Imposing the additional constraints that $\alpha_u = \beta_u$ and $\alpha_d = \beta_d$:

$$|A| = \exp\left(-c\left(\alpha_u - \alpha_d\right)\left(1 - \cos k\Delta x\right)\right) \tag{51}$$

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

$$\alpha_u - \alpha_d \geq 0 \tag{52}$$

which provides a lower bound on α_u:

$$\alpha_u \geq \alpha_d \tag{53}$$

Additionally, we do not want more damping than an upwind scheme (where $\alpha_u = \beta_u = 1$, $\alpha_d = \beta_d = 0$), having an amplification factor, A_{up}:

$$|A_{\text{up}}| = \exp\left(-c\left(1 - \cos k\Delta x\right)\right) \tag{54}$$

So we need $|A| \geq |A_{\text{up}}|$:

$$-c\left(\alpha_u - \alpha_d\right)\left(1 - \cos k\Delta x\right) \geq -c\left(1 - \cos k\Delta x\right) \tag{55}$$

$$\alpha_u - \alpha_d \leq 1 \tag{56}$$

$$\alpha_u \leq 1 + \alpha_d \tag{57}$$

which provides an upper bound on α_u. Combining with eqn (53) we can bound α_u on both sides:

$$\alpha_d \leq \alpha_u \leq 1 + \alpha_d \tag{58}$$

Now, knowing that $\alpha_u + \alpha_d = 1$ (or $\alpha_d = 1 - \alpha_u$), then

$$1 - \alpha_u < \alpha_u \leq 1 + (1 - \alpha_u) \tag{59}$$

$$0.5 \leq \alpha_u \leq 1 \tag{60}$$

and, since $\alpha_u + \alpha_d = 1$, then

$$0 \leq \alpha_d \leq 0.5 \tag{61}$$

Three-point analysis

We start again from equation (38) but this time approximate ϕ_L and ϕ_R using three points:

$$\phi_L = \alpha_{uu}\phi_{j-2} + \alpha_u\phi_{j-1} + \alpha_d\phi_j \tag{62}$$

$$\phi_R = \alpha_{uu}\phi_{j-1} + \alpha_u\phi_j + \alpha_d\phi_{j+1} \tag{63}$$

having used the same weights α_{uu}, α_u and α_d for both left and right fluxes. Substituting equation (41) into equation (38) we find

$$A = \exp\left(-c\left[\alpha_{uu}\left(e^{-ik\Delta x} - e^{-2ik\Delta x}\right) + \alpha_u\left(1 - e^{-ik\Delta x}\right) + \alpha_d\left(e^{ik\Delta x} - 1\right)\right]\right) \tag{64}$$

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

$$\alpha_u - \alpha_d + (\alpha_{uu} - \alpha_u + \alpha_d)\cos k\Delta x - \alpha_{uu}\cos 2k\Delta x \geq 0 \tag{65}$$

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

$$\alpha_u - \alpha_d \geq |\alpha_{uu}| \tag{66}$$

and, when the same analysis is performed with four points, α_{uuu}, α_{uu}, α_u and α_d, we find that the same condition holds replacing α_{uu} with α_{uuu}. Hence, we generalise equation (66) to find the final stability constraint

$$\alpha_u - \alpha_d \geq \max_{p \in P} |\alpha_p| \tag{67}$$

where the peripheral cells P is the set of all stencil cells except for the upwind and downwind cell, and α_p is the weight for a given peripheral cell p.

[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**, 44–49, doi:10.1002/asl2.465.

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

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

[4] 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.

[5] 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**, 105–145, doi:10.5194/gmd-7-909-2014.

[6] 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**, 887–901, doi:10.5194/gmd-5-887-2012.

[7] 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**, 2459–2480, doi:10.1175/1520-0493(2002)130<2459:ANTFVC>2.0.CO;2.

[8] Shaw, J., and H. Weller, 2016: Comparison of terrain following and cut cell grids using a non-hydrostatic model. *Mon. Wea. Rev.*, doi:10.1175/MWR-D-15-0226.1.

[9] 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**, 909–929, doi:10.5194/gmd-7-909-2014.