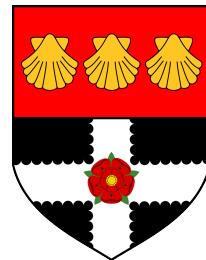


**UNIVERSITY OF READING**

Department of Meteorology



# **Vertical Coordinates Over Orography**

by

Matthew Jones

Supervisor

Dr Hilary Weller

A dissertation submitted in partial fulfilment of the requirement  
for the degree of Master of Science in  
Atmosphere, Oceans and Climate

August 2013



# *Abstract*

## **Vertical Coordinates Over Orography**

by Matthew Jones

As the resolution of numerical models increases, orography is resolved in more detail with steeper slopes. These steeper slopes can lead to large errors when using terrain-following coordinates, so a better way of representing orography in a numerical model is needed. One way to do this is by using more orthogonal grids. Throughout this project grids are used rather than coordinates because they are easier to implement and grids can be created to represent terrain-following coordinates. This dissertation reviews different terrain-following coordinates and different ways of creating more orthogonal grids. A new technique for making terrain-following coordinates more orthogonal is also described.

Various grids were created and compared by using each to solve the linear advection equation using OpenFOAM with and without orography. For the tests with orography, the grids were constructed using terrain-following coordinates from the literature. A more orthogonal grid was then created using the smooth level vertical (SLEVE) coordinates and velocity potential contours.

For the tests without orography, the contributions from dispersion and diffusion errors were diagnosed; these were used to inform the analysis of the tests which contained orography. SLEVE produced the smallest errors of the terrain-following coordinate based grids - similar shaped errors were seen for all the terrain-following coordinate based grids but with much smaller magnitude. The error was mainly due to the grid not being aligned with the flow. The more orthogonal grid was shown to perform slightly better than the SLEVE grid.

A limiting factor in this project was the test case and the equations solved, so further testing is needed for this new grid, particularly a test which solves the momentum equation for the velocity. However, no detrimental effect was found for the test case used caused by making a terrain-following coordinate based grid more orthogonal.

## *Acknowledgements*

First and foremost I would like to thank my supervisor, Dr Hilary Weller, for all her help and advice throughout this project. I would also like to thank the Computational Atmospheric Dynamics Group for their help with my presentation.

Finally, I would like to thank my family and friends for their help and support throughout this year.

# Contents

<b>Abstract</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>iv</b>
<b>Abbreviations</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation and ways of representing terrain in a numerical model . . . . .	1
1.2 Objectives . . . . .	4
1.3 Dissertation Outline . . . . .	4
1.4 Terrain-following coordinates . . . . .	5
1.4.1 Information on the Jacobian . . . . .	5
1.4.2 Basic terrain following (BTF) coordinates . . . . .	6
1.4.3 Hybrid terrain following (HTF) coordinates . . . . .	7
1.4.4 Smooth level vertical (SLEV) coordinates . . . . .	8
1.4.5 Smoothed terrain-following (STF) coordinates . . . . .	9
1.5 Orthogonal grids . . . . .	10
1.5.1 What makes orthogonal coordinates better? . . . . .	10
1.5.2 Methods to create orthogonal and near-orthogonal grids . . . . .	11
1.6 Pressure gradient errors . . . . .	15
1.7 Numerical truncation error . . . . .	16
1.8 Approach taken in this project . . . . .	17
<b>2 Methodology</b>	<b>19</b>
2.1 Description of the numerical model used . . . . .	19
2.1.1 OpenFOAM . . . . .	20
2.1.2 Description of the discretisations used . . . . .	20
2.1.2.1 Spatial discretisations . . . . .	20
2.1.2.2 Temporal discretisations . . . . .	22
2.1.3 Courant number on an unstructured grid . . . . .	23
2.2 Test case used in this project . . . . .	24
2.2.1 Initial conditions . . . . .	25
2.2.2 Domain . . . . .	25
2.2.3 Terrain . . . . .	26
2.3 Creating the terrain-following coordinate based grids . . . . .	26
2.4 Near-orthogonal grid generation . . . . .	28
2.5 Errors with scalar transport due to horizontal differencing . . . . .	31
2.6 Description of error diagnostics . . . . .	31

<b>3 Results</b>	<b>35</b>
3.1 Tracer advection over flat ground . . . . .	35
3.1.1 Tracer advection using centred, linear, advection . . . . .	38
3.1.2 Tracer advection using first order upwind advection . . . . .	42
3.1.3 Tracer advection using a van Leer limiter . . . . .	43
3.2 Properties of the grids used for tracer advection over orography . . . . .	44
3.3 Tracer advection over orography . . . . .	46
3.3.1 Tracer advection using a grid based on BTF coordinates . . . . .	46
3.3.2 Tracer advection using a grid based on HTF coordinates . . . . .	49
3.3.3 Tracer advection using a grid based on SLEVE coordinates . . . . .	50
3.3.4 Tracer advection using the new near-orthogonal grid . . . . .	50
3.3.5 Summary of the tracer advection over orography . . . . .	51
3.4 Analysis of the grids for SLEVE and the near-orthogonal grid . . . . .	52
3.5 Error diagnostics for all the grids . . . . .	53
<b>4 Summary and Conclusions</b>	<b>59</b>
4.1 Conclusions on the objectives . . . . .	59
<b>5 Further Work</b>	<b>63</b>
<b>Bibliography</b>	<b>67</b>

# Abbreviations

<b>BTBS</b>	Backwards in Time, Backwards in Space
<b>BTCS</b>	Backwards in Time, Centred in Space
<b>BTF</b>	Basic Terrain Following
<b>HTF</b>	Hybrid Terrain Following
<b>PDE</b>	Partial Differential Equation
<b>SLEVE</b>	Smooth LEvel VErtical
<b>STF</b>	Smoothed Terrain Following
<b>PMA</b>	Parabolic Monge-Ampère
<b>TV</b>	Total Variation
<b>TVD</b>	Total Variation Diminishing



# Chapter 1

## Introduction

This project investigates how the use of different grids used in a numerical model affects the error produced by orography.

### 1.1 Motivation and ways of representing terrain in a numerical model

As the horizontal resolution in numerical models increases, steeper slopes are simulated for terrain. Terrain is usually represented using terrain-following coordinates which suffer from reduced accuracy in areas of steep terrain (Klemp, 2011). This implies that as the resolution of numerical models increases, a different way of representing terrain is needed.

The representation of terrain by terrain-following coordinates allows a structured grid to be used where the vertical coordinate, let it be  $Z$ , replaces the original vertical coordinate. So, for example, the Cartesian coordinates  $(x, y, z)$  would become  $(x, y, Z)$ . The terrain following coordinate,  $Z$ , can be formulated so that the grid near the ground follows the undulations of the terrain and the layers aloft can be more horizontal.

Schär et al. (2002) describes some advantages and disadvantages of terrain-following coordinates. Some advantages over other coordinates include the fact that they map the terrain onto a rectangular style grid; this suits the way that domains are discretised on computers. Also, the lower boundary condition can be simplified because the transformed vertical wind disappears at the surface. Finally, terrain-following coordinates can easily allow unequally

spaced vertical grid levels, being denser near the surface and sparser aloft. However, terrain-following coordinates do have some disadvantages such as: the difficulty in formulating the horizontal pressure gradient because of the sloping computational surfaces. The horizontal and vertical terms in the advection equation need to be treated consistently to ensure that the correct numerical cancellations take place when approximating spatial discretisations on sloping coordinate surfaces. If these cancellations do not occur, nonhomogenous terms can arise in the linear wave equation; this can produce distortions in wave solutions (Klemp et al., 2003), increasing the error.

In terrain-following coordinates, a high degree of non-orthogonality can occur where there are steep slopes (Steppeler et al., 2003); this also causes a high aspect ratio due to deformation of the coordinates in the presence of steep terrain (Lundquist et al., 2010). These two factors, along with other sources of error (discussed later), contribute to reduce the accuracy of the numerical model in the presence of steep terrain (Lundquist et al., 2010); these are reasons to try and improve on terrain following coordinates.

Sigma coordinates are a form of terrain-following coordinates introduced by Gal-Chen and Somerville (1975) are based on pressure. They are coordinates in which the vertical coordinate is proportional to the pressure at the surface, so that the vertical coordinate is defined as

$$Z = p/p_s \quad (1.1)$$

where  $p$  is the pressure and  $p_s = p_s(x, y, t)$  is the pressure at the surface (Holton, 2004). With this definition, the lowest coordinate follows the ground and the velocity is always zero at the ground, thus simplifying the lower boundary condition (Holton, 2004). Sigma coordinates produce a similar grid near the terrain to terrain-following coordinates and so would suffer from the same problems due to non-orthogonality. An example of sigma coordinates is shown in Figure 1.1(b).

Vertical coordinates based on temperature, isentropic coordinates, define the vertical coordinates based on levels of constant potential temperature,  $\theta$ . These are advantageous because there is reduced vertical transport through the vertical levels and lateral mixing is along isentropic surfaces, giving no unwanted cross isentropic mixing (Benjamin et al., 2004). Thus spurious entropy growth when using coordinates not based on  $\theta$  is avoided (Benjamin et al., 2004). These can be blended with terrain-following surfaces near the ground so that the layers do not intersect with the ground (Konor and Arakawa, 1997).

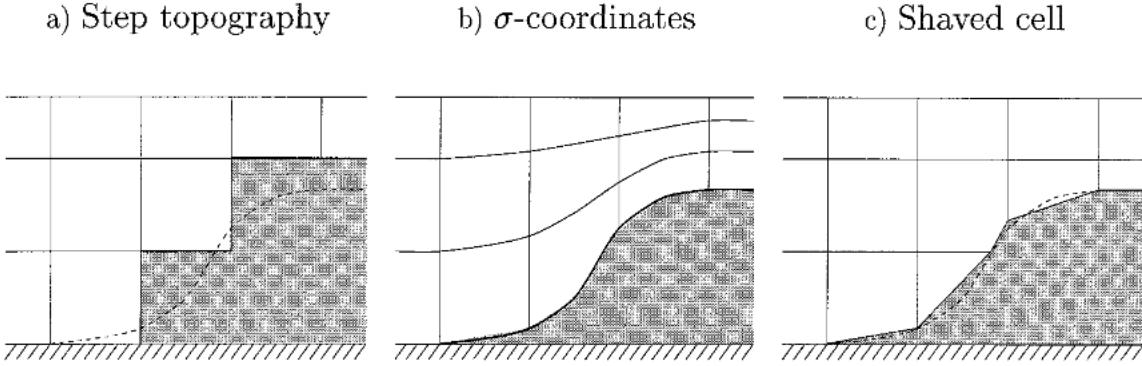


FIGURE 1.1: Figure from Adcroft et al. (1997) showing examples of step topography, sigma coordinates (as an example of terrain-following coordinates) and shaved cells. The dashed line shows the shape of the topography.

An alternative to terrain-following coordinates is to use height as a vertical coordinate; this can overcome errors due to the deformation of the coordinates. Some coordinates based on height build the shape of the orography using a step function (Steppeler et al., 2002) (an example is shown in Figure 1.1(a)). Because of the discontinuity of the orographic function, these coordinates do not have sufficient convergence properties (Steppeler et al., 2003). An improvement on these coordinates would be to use shaved cells that have parallel horizontal grid lines where the cells near the terrain are ‘shaved’ to represent the shape of the terrain as shown in Figure 1.1(c). It can be seen where the cells have been cut where they are intersected by the terrain profile approximating the shape of the terrain. These have produced good results for adiabatic mesoscale flows but the problem with this technique is that they would require large modifications to current numerical models and parametrisation that use terrain-following methods (Schär et al., 2002). These two methods are orthogonal and preserve the aspect ratio so do not suffer the same truncation errors that affect terrain-following coordinates. However, when unequally spaced vertical levels are used, high terrain can suffer from low resolution in the boundary layer (Steppeler et al., 2003).

There are obvious advantages to terrain-following coordinates over the other coordinates such as following terrain and so simplifying boundary conditions. The downsides of terrain following coordinates could possibly be overcome by making them more orthogonal and so taking advantage of the benefits of orthogonal coordinates, such as shaved cells. The aim of this project was to review some of the existing terrain following coordinates using a simple test case to compare the errors produced, and explore the use of more orthogonal coordinates, then compare the created grid with the terrain-following coordinates. The theory is that the

increased orthogonality will increase the accuracy of the numerical model.

## 1.2 Objectives

The objectives of this project are as follows:

- Review different terrain following coordinates from the literature.
- Review ways of creating orthogonal coordinates from the literature.
- Run a scalar transport over flat ground to analyse where errors are coming from for different numerical schemes.
- Run a scalar transport to test different grids based on terrain-following coordinates.
- Create a more orthogonal grid, compare it with the terrain-following coordinates and assess its suitability for further testing.

The tracer transport tests conducted in this project are only the first step in testing the hypothesis that more orthogonal grids will improve the representation of orography. The biggest improvements when using more orthogonal grids would be expected when simulating equations with pressure gradients and orography, however tracer transport tests are an important first step.

## 1.3 Dissertation Outline

The remainder of this chapter discusses terrain-following coordinates and ways of creating orthogonal coordinates from the literature. Chapter 2 gives information on the methodology used to create the grids used in this project and details ways of quantifying the errors produced during scalar transport. Chapter 3 discusses the results obtained from the different grids and some properties of the grids along with comparisons of different spatial discretisation schemes for a test with no orography and then gives some more detailed analysis of the grids. Chapter 4 concludes the results and discussion and Chapter 5 gives suggestions of work to further the work in this project.

## 1.4 Terrain-following coordinates

### 1.4.1 Information on the Jacobian

Before moving on to terrain-following coordinates, it is useful to have some understanding of the Jacobian. When solving equations on a grid which is not aligned along the Cartesian coordinates ( $x, y, z$ ) there are two approaches. Firstly, in a structured grid approach, new coordinates can be defined in which the grid is aligned with the coordinates (like with the terrain following coordinates). Or, secondly, in an unstructured grid (geometrically unstructured but logically structured) approach, the Cartesian coordinates are maintained and the grid is not aligned with the coordinates. For the former approach it is necessary to calculate the Jacobian for transforming derivatives between the old and new set of coordinates. For example when transforming between the old Cartesian coordinates ( $x, y, z$ ) and a new set of coordinates ( $X, Y, Z$ ) the Jacobian would be defined as,

$$J = \begin{vmatrix} \frac{\partial x}{\partial X} & \frac{\partial x}{\partial Y} & \frac{\partial x}{\partial Z} \\ \frac{\partial y}{\partial X} & \frac{\partial y}{\partial Y} & \frac{\partial y}{\partial Z} \\ \frac{\partial z}{\partial X} & \frac{\partial z}{\partial Y} & \frac{\partial z}{\partial Z} \end{vmatrix} \quad (1.2)$$

where  $\partial/\partial X$  and  $\partial/\partial Y$  denote the derivatives on the  $Z$  surfaces (Weisstein, 2013a) then, for example the equation,

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{u}\phi) = 0 \quad (1.3)$$

where  $\mathbf{u}$  is the velocity and  $\phi$  is the tracer being advected, becomes, in two dimensions,

$$\frac{\partial}{\partial t}(J^{-1}\phi) + \frac{\partial}{\partial X}(J^{-1}u\phi) + \frac{\partial}{\partial Z}(J^{-1}W\phi) = 0 \quad (1.4)$$

where  $W = DZ/Dt$  is the vertical velocity in the new coordinates (Schär et al., 2002).

When using the unstructured grid approach derivatives can be calculated using Gauss' divergence theorem,

$$\int_V \nabla \cdot \mathbf{u} dV = \int_S \mathbf{u} \cdot d\mathbf{S} \quad (1.5)$$

where  $V$  is the volume of the cell which has surfaces  $S$  with the outward pointing normal  $d\mathbf{S}$  (OpenFOAM Programmer's Guide).

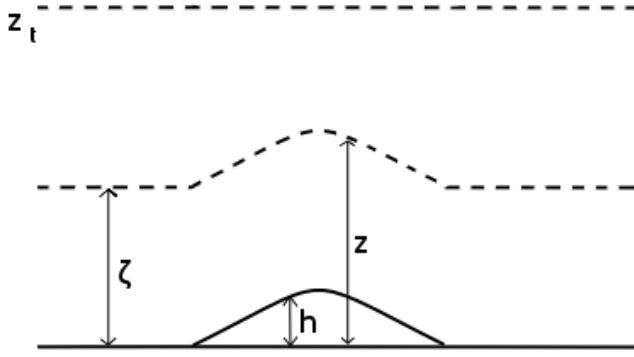


FIGURE 1.2: Visual representation of the variables in equation (1.6) for an example orography profile.

An assumption for this project is that on distorted grids the error from both approaches can be similar. So it is worth reporting on the literature regarding the truncation errors of the Jacobian approach when attempting to understand the truncation errors that come from the approach of using an unstructured grid. This is done throughout the discussion on terrain-following coordinates and in Section 1.7 on truncation error.

#### 1.4.2 Basic terrain following (BTF) coordinates

The BTF coordinates described in Schär et al. (2002) and Klemp (2011) keep the most information about the shape of the orography aloft because the orographic influence decreases linearly with height following the equation,

$$z = (z_t - h) \frac{\zeta}{z_t} + h \quad (1.6)$$

where  $z$  is the height of the computational surfaces,  $z_t$  is the height of the upper boundary of the domain,  $\zeta$  is height that the surfaces would have if there was no orography present and  $h$  is the profile of the terrain (Klemp, 2011) as shown in Figure 1.2. BTF coordinates give a similar coordinate profile to the often used sigma coordinates (Klemp, 2011) and an example is shown in Figure 1.3 from Klemp (2011).

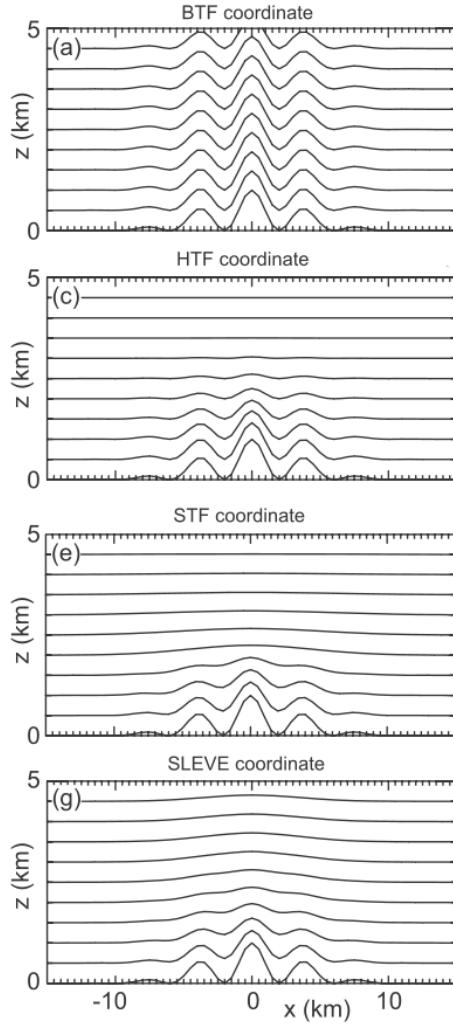


FIGURE 1.3: Figure adapted from Klemp (2011) showing the different terrain following coordinates discussed in Section 1.4.

### 1.4.3 Hybrid terrain following (HTF) coordinates

The HTF coordinates are similar to BTF coordinate except instead of a linear decay of the effect of the shape of the terrain there is an exponential decay. The form of the HTF coordinates used here is from Schär et al. (2002) and is hybrid-like (terrain-following at the surface moving to horizontal aloft) when a scale height shorter than the top of the domain is used; this is the height in which the terrain shape decays by approximately  $1/e$ . The equation describing HTF is,

$$z = \zeta + h \frac{\sinh[(z_t - \zeta)/s]}{\sinh[z_t/s]} \quad (1.7)$$

where the symbols have the same definition as with BTF and  $s$  is the scale height (Schär et al., 2002). An example of HTF from Klemp (2011) is shown in Figure 1.3.

Having an exponentially decaying function has advantages. Aloft, the computational levels are more horizontal and less distorted which reduces the truncation error - errors due to approximating the pressure gradient term and the horizontal differencing. Having exponential decay as opposed to linear decay means that the Jacobian is affected (Schär et al., 2002), reducing the horizontal gradient of its inverse and so reducing the magnitude of the truncation error due to the grid transformation. This implies that there should be obvious improvement for HTF over BTF shown in Section 3.3.

#### 1.4.4 Smooth level vertical (SLEVE) coordinates

The SLEVE coordinates, introduced by Schär et al. (2002), splits the terrain into a large scale and a small scale contribution and then has the small scale contribution decay faster than the larger scale one. The decay is achieved in a similar way to HTF, in which there is a scale height for an exponential decay, but for SLEVE there is a separate scale height for each contribution of the terrain. This is described by the equation,

$$z = \zeta + h_1(x, y)b_1(\zeta) + h_2(x, y)b_2(\zeta) \quad (1.8)$$

where the vertical decay functions are of the form,

$$b_i(\zeta) = \frac{\sinh[(z_t - \zeta)/s_i]}{\sinh[z_t/s_i]} \quad (1.9)$$

where  $i = 1, 2$ , and  $h_1$  and  $h_2$  are the large and small scale effects from the terrain respectively (Schär et al., 2002). An example of SLEVE is shown in Figure 1.3 from Klemp (2011).

Having two separate decay functions for the small and large scale contributions of the terrain allows the small scale to be decayed much faster than the large one, enabling the coordinates to smooth faster, giving a reduction in the truncation errors, even compared to HTF.

There is, however, a problem with SLEVE. At levels close to the ground there exists large compression (Leuenberger et al., 2010), giving levels that are closer together than is desirable. Therefore, the maximum time step may be limited by the thickness of the layer. Therefore, to keep the Courant number small, the timestep would need to decrease; this is particularly important for convection resolving models because of the vertical velocities involved. This is because large fluxes into very thin layers can cause instabilities in a numerical model. This can be a problem when high turbulent fluxes occur such as at the peak of mountains (Leuenberger

et al., 2010). Lastly, if the first layer is too close to the terrain, violations of boundary layer assumptions can occur, especially turbulent fluxes of heat and momentum - related to Obunkhov scaling which states that the depth of the lowest model level must be much larger than the roughness scale for all grid points (Leuenberger et al., 2010).

The solution which Leuenberger et al. (2010) arrived at was to generalise the decay function so that

$$b_i(\zeta) = \frac{\sinh[(z_t/s_i)^n - (\zeta/s_i)^n]}{\sinh[(z_t/s_i)^n]} \quad (1.10)$$

where  $n \geq 1$  and is real. This is equivalent to the original formulation of the SLEVE coordinates when  $n = 1$ . Leuenberger et al. (2010) found that the optimal value for  $n$  is 1.35 which gives the least compression of lower levels while still decaying the terrain profile sufficiently.

#### 1.4.5 Smoothed terrain-following (STF) coordinates

STF coordinates (shown in Figure 1.3), introduced by Klemp (2011), take a slightly different approach to smoothing the coordinates with increasing height. Klemp (2011) generalised BTF to,

$$z = (z_t - A'h_s) \frac{\zeta}{z_t} + A'h_s \quad (1.11)$$

where  $A'$  controls the rate that the coordinates change from terrain following at the ground to horizontal aloft and  $h_s$  is a smoothed form of  $h$  that can be formulated to change with height. The fact that the terrain profile can be smoothed more with increasing height gives STF more flexibility than SLEVE (Klemp, 2011). Multiple techniques could be considered to smooth  $h_s$  with height such as Fourier filtering which could remove higher wave numbers from the profile or a smoother could be implemented on the terrain profile at each level, smoothing each level more with height (Klemp, 2011).

HTF, SLEVE and STF all share the problem that they have unknown parameters which need to be defined. When using a global model these parameters would need to be set to values that would be useful over the entire domain, which could be problematic.

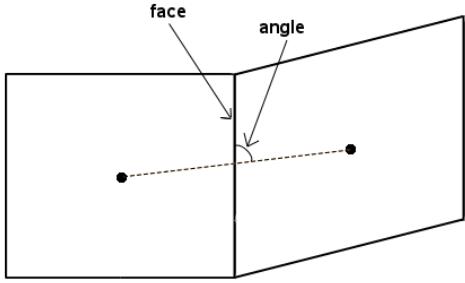


FIGURE 1.4: Definition of the angle between the line joining the centre of two neighbouring cells and the face between them.

## 1.5 Orthogonal grids

Most of the ways of generating orthogonal grids look at the production of a grid rather than a set of coordinates. Orthogonal coordinates are defined when the angle between the line between two cell centres and the face or edge between the cells is  $90^\circ$  - the definition of the face and the angle is shown in Figure 1.4.

### 1.5.1 What makes orthogonal coordinates better?

It is well known that the accuracy of a numerical model is higher on orthogonal grids than non-orthogonal grids (Eça, 1998), and there are a number of reasons for this. On an orthogonal grid the grid lines are orthogonal with the boundary, so boundary conditions are generally easier to implement (Eça, 1998). Also mentioned in Eça (1998) is that the implementation of turbulence models is often simpler on an orthogonal grid because, normally, information along perpendicular directions is required. Putman and Lin (2007) mention that many model formulations require an orthogonal grid so using a non-orthogonal grid can affect the accuracy of the model. Truncation errors in the finite difference expressions used are reduced on orthogonal grids (Sankaranarayanan and Spaulding, 2003). Sankaranarayanan and Spaulding (2003) show that the truncation errors are a function of the grid angle and the aspect ratio of the grid, leading to the improved accuracy on a more orthogonal grid. These examples of improved accuracy, however, are for discretisation of grids in the horizontal plane rather than the vertical. Local orthogonality leads to less additional terms in numerical equations Ringler (2011) which is likely to improve the accuracy of a model. By creating a more orthogonal grid, it could be possible to take advantage of some of these advantages.

In three dimensions with complex geometry, a perfectly orthogonal grid may not be possible to create (Eça, 1998) (on a logically structured grid), but when the coordinates are near-orthogonal the magnitude of the extra terms introduced to the equations of motion are much smaller in comparison to when the non-orthogonal coordinates are being used (Thompson et al., 1997), implying errors are likely to be smaller. Thompson et al. (1997) notes that when there is a severe departure from the grid being orthogonal, more truncation error is introduced and so it is worth pursuing near-orthogonal grids because of the reduction in this truncation error the closer that you get to orthogonality.

### 1.5.2 Methods to create orthogonal and near-orthogonal grids

There are various different methods to create orthogonal and near-orthogonal grids for an arbitrary domain in two dimensions. The main categories that the investigated methods fall into are: field methods, trajectory methods and conformal mapping. Of these three only trajectory methods would be usable in three-dimensions because conformal mappings do not exist for most three-dimensional cases and most of the equations that can be solved for the field method only have solutions in two-dimensions. A couple of other techniques, that do not fit into these categories, are also touched upon: rotating basis vectors, Voronoi grids and parabolic Monge-Ampère, all of these are described below.

The first set of methods, known as field methods, create grids based on the solutions to partial differential equations (PDEs) (Eça, 1998). The PDEs can be elliptic, parabolic or hyperbolic (Eça, 1998). Adamson (1984) describes a method of producing a boundary-fitted orthogonal curvilinear grid by solving elliptic equations. The method described involves a function which can be defined to control how packed the cells get to try and give more control over the spacing of grid points. Another approach is to consider Laplace's equation, as described in Eça (1998), using the solutions of the stream function ( $\nabla^2\psi = 0$ ) and the velocity potential ( $\nabla^2\chi = 0$ ) to define the horizontal and vertical grid lines respectively. Eça (1998) also describes how hyperbolic equations can be solved to grow a grid out from a boundary and how the orthogonality of these methods helps to prevent skewness in the grid.

The second set of methods are known as trajectory methods (Eça, 1998) - McNally (1972) also describes a similar method for an arbitrary two-dimensional domain. The grids are started by creating a grid using a well-known non-orthogonal method, for example a terrain-following

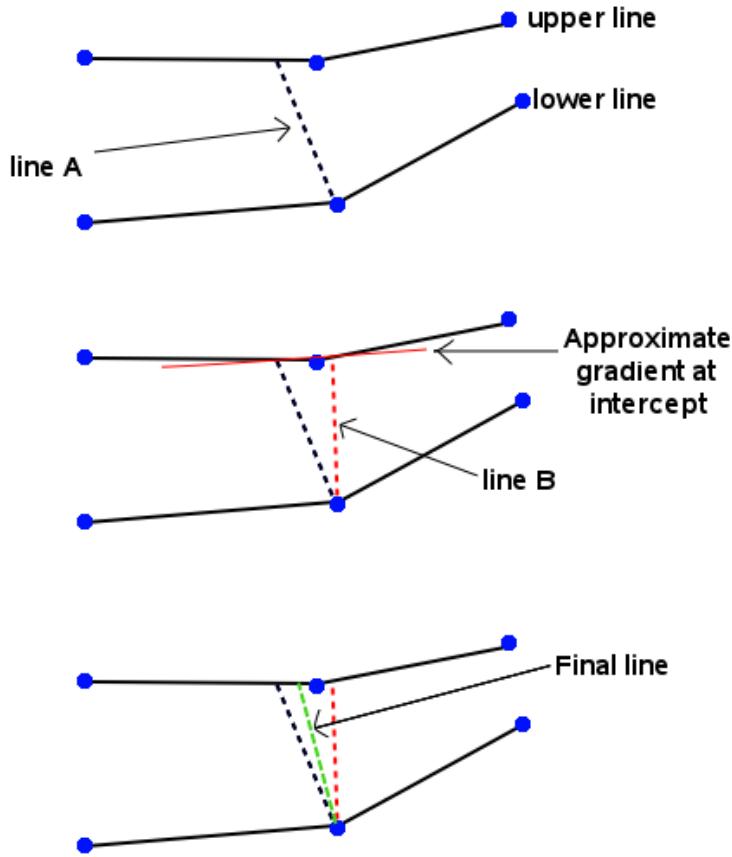


FIGURE 1.5: Visual description of the trajectory technique described in Section 1.5.2. The blue dots show the positions of the original vertical lines from the non-orthogonal grid.

coordinate based grid, with one set of grid lines are removed to generate a new set as follows (this method is shown in Figure 1.5):

- Firstly, on one of the boundaries the grid points for the new lines are defined so that the spacing of the grid lines is defined on this boundary. Then there is a marching process from that boundary to produce the new lines.
- On the current line retained from the non-orthogonal grid (let this line be called the lower line) a new line is made by creating the normal of this line which intercepts the next line up (let this line above be called the upper line and the line created be called line A).

- Using the gradient of the upper line, a new line is created back through the original point which is perpendicular to the tangent of the upper line at the intercept between line A and the upper line (let this line be called line B).
- The final line between upper and lower lines from the non-orthogonal grid is between the point started from on the lower line and halfway between the intercepts of line A and B with the upper line (Thompson et al., 1997).

This process is then repeated on the upper line until the top of the domain is reached. Then this marching process is repeated at the next boundary point along the domain. Thompson et al. (1997) notes that the line created will not be orthogonal with either the line started from or the line intersected. This method could cause issues with tangling (where the grid lines end up crossing) so a method of restricting the spacing between the grid points would need to be implemented.

Conformal mapping is a technique which preserves the angles at the intercepts between curves (Putman and Lin, 2007), so could be used to create an orthogonal grid by transforming a regular grid with no orography to include the effect of orography. Conformal mappings produce few additional terms when approximating PDEs (Thompson et al., 1997), however, there is very little control over the positioning of grid points and conformal mappings do not generally exist in three dimensions (Thompson et al., 1997). Putman and Lin (2007) showed, for the geometry tested, that despite conformal mapping producing more orthogonal grids and preserving the aspect ratio very well, the size of the cells become too small at high resolutions therefore making it an inefficient technique to use.

Li et al. (2012) discuss a way of rotating the basis vectors of a grid to make the grid more orthogonal in the vertical for an atmospheric model. Rotating the basis vectors has the effect of smoothing coordinates with height, which can produce similar horizontal levels as terrain-following coordinates, but this method keeps the coordinates more orthogonal. They show this method produces good results for different rotation parameters - these increase the amount of rotation of the basis vectors with height. The best results shown were for the smoothing of the vertical levels shown by the green line in Figure 1.6. This gave a very dramatic smoothing of the coordinate levels with height which could cause problems because of low level compression causing errors.

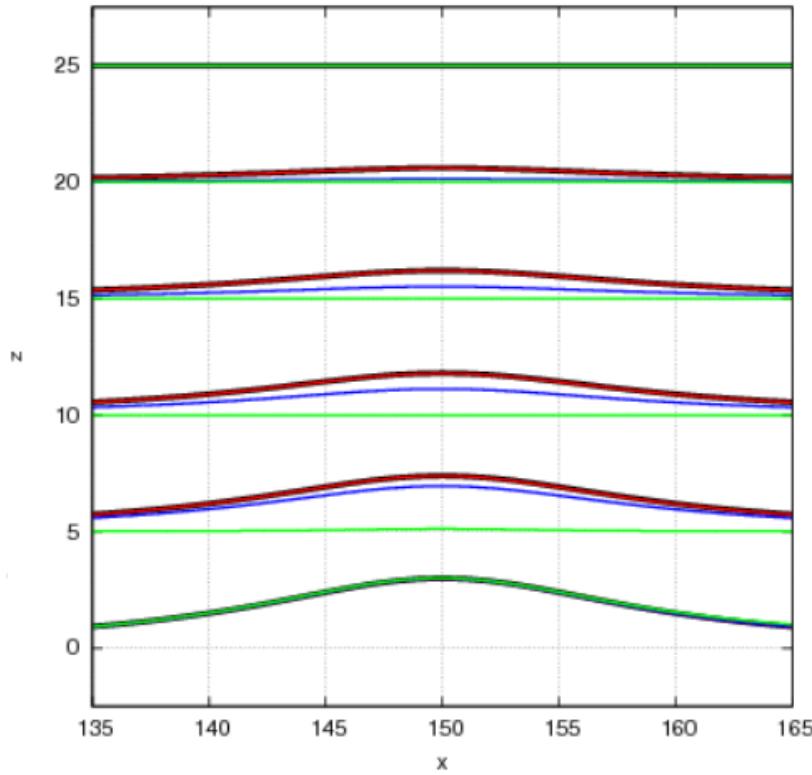


FIGURE 1.6: Figure from Li et al. (2012) showing the smoothing of the vertical coordinate levels for different rotation parameters each shown by a different colour.

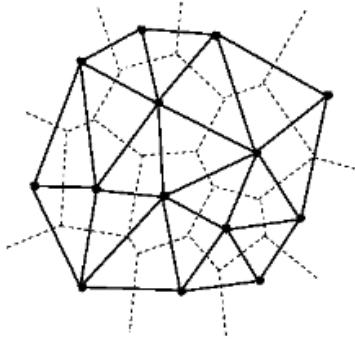


FIGURE 1.7: Figure from Van Laerhoven (2013) showing Delaunay triangles (solid line) and a Voronoi grid (dotted lines).

Voronoi grids can be used in the horizontal to generate arbitrarily structured grids which are orthogonal. Voronoi grids can be uniform or non-uniform and can be generated to give high quality orthogonal grids created from Voronoi tessellations, which can be either triangles (known as Delaunay triangles) or hexagons (Ju et al., 2011) shown in Figure 1.7. Ju et al. (2011) describes methods to create these grids and also notes that they are well suited to create variable resolution grids.

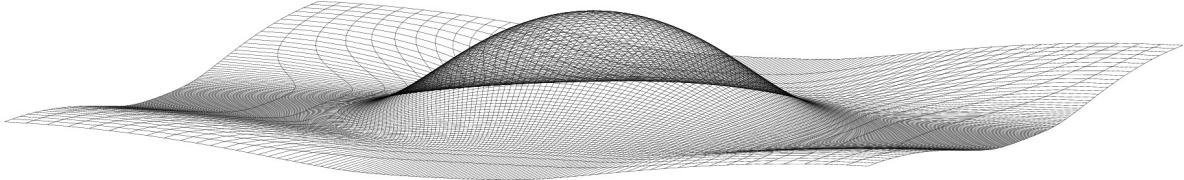


FIGURE 1.8: Example of the use of PMA where the grid points have been moved towards the peak of the terrain. (P.A. Browne, personal communication, July 24, 2013)

A method of moving grid points in three-dimensions, known as parabolic Monge-Ampère (PMA) (Browne et al., in preparation), has potential to be used to create more orthogonal grids. In PMA the grid points are moved according to a monitor function to bunch grid points in specific areas. An example is shown in Figure 1.8 where the grid points have been moved towards the peak of the hill shown. This could be used to create more orthogonal grids by moving all the points in the grid horizontally towards the peaks of the terrain along surfaces defined by terrain following coordinates then implementing Laplacian smoothing (all the points are averaged with their neighbours over multiple iterations, smoothing the grid and making it more orthogonal). Another way to create a more orthogonal grid using PMA would be to move the points towards the peaks depending on the terrain effect at each level rather than the terrain itself. Both of these methods would curve the vertical grid surfaces towards the peaks in the terrain making them more orthogonal.

## 1.6 Pressure gradient errors

The smoothing of terrain aloft does not help near the ground where the coordinates still have to follow the orographic profile. Here, the most critical discretisation error is due to the horizontal pressure gradient (because of sloping coordinate surfaces); this can induce spurious circulations and possibly cause numerical instability if the terrain is of sufficient incline (Zängl, 2012). This error occurs because the horizontal pressure gradient cannot be taken along the same coordinate surface (shown in Figure 1.9 at  $p_{j+1,k}$ ) because of the sloping and instead, information from other layers is needed which is not necessarily in the right place (for example  $p_{j+1,k+2}$  on Figure 1.9), introducing error. Zängl (2012) develops a way of calculating a truly horizontal pressure gradient to reduce this error. This is done by using the two most horizontal cells next to the cell where the pressure gradient is being calculated (in the case shown in Figure 1.9 this would be  $p_{j+1,k+1}$  and  $p_{j+1,k+2}$ ) then interpolating between them to get a better representation of the pressure horizontally. This work is not needed in this project

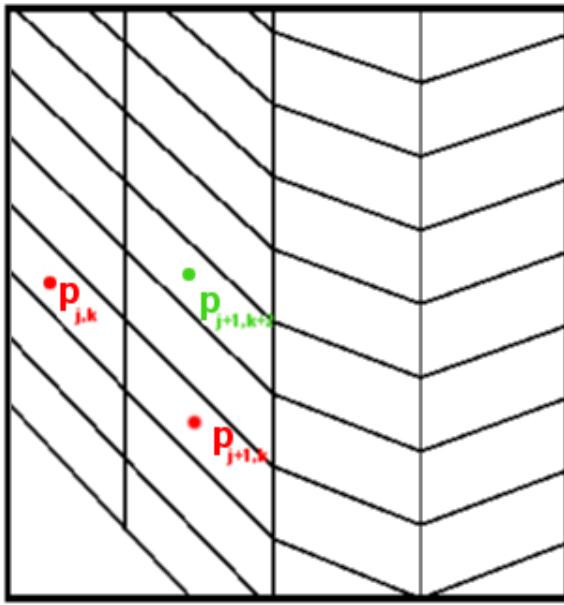


FIGURE 1.9: A section from the grid for BTF showing an example of where the pressure gradient could produce errors.

however, because horizontal pressure gradients are not being calculated due to not solving for the wind.

## 1.7 Numerical truncation error

Schär et al. (2002) conducted a theoretical analysis of the truncation error, caused by taking a finite number of elements from a Taylor series expansion, for arbitrary grid transformations. They found the truncation error can be separated into the truncation error appearing with a uniform grid and those due to the grid transformation - the truncation error due to the transformation of the grid has the same leading order as the truncation error that exists when there is a uniform grid. The relative importance of each of the truncation errors due to a uniform grid and the grid transformation depends on the scale of the anomalies in the flow. The transformation errors will dominate for larger scale anomalies and regular numerical errors will be dominant for smaller scale anomalies. This implies that the transport of large scale anomalies will be influenced by small scale transformation errors.

The Jacobian gives some sort of idea about how distorted the coordinates are, which can then be used to give an indication of the relative sizes of the two parts of the truncation error. Because the truncation error due to the grid deformation is related to the Jacobian,

and increases with the Jacobian (Schär et al., 2002), this implies that the truncation error due to the grid distortion is greater with more skewed coordinates, which would be as expected. When the coordinates tend to the original uniform coordinates, the Jacobian tends to unity and the truncation error becomes the truncation error present on a uniform grid. This would imply that the smoother coordinates have a smaller error which would be closer to the error due to the numerical model in absence of terrain, which is what was found as discussed later. This is also why most of the literature discusses ways of smoothing the coordinates aloft.

## 1.8 Approach taken in this project

Instead of formulating the equations in the different sets of coordinates, as is normally done, equations for an unstructured grid were used and the points in the grid (the vertices of the cells) were moved to represent the different coordinate systems and the orography; this allowed OpenFOAM to be used, using an unstructured grid approach. To test the different vertical coordinates, a simulation of a two dimensional scalar advection was run on the test case described in Schär et al. (2002) then the results were compared to the analytical solution produced using the same grid which gave some idea of the errors that were occurring. This was then used to compare different vertical coordinates and more orthogonal grids. First, non-orthogonal sets of coordinates were tested and the most suitable of these used to create a more orthogonal grid. Throughout all the tests, the velocities were not solved for but instead the wind profiles were exactly defined. This means that a large part of the advantage of an orthogonal grid is lost because pressure gradients are only needed in the prediction of velocity. However, this is an important first test of orthogonal grids.



# Chapter 2

## Methodology

In this section, the methodology used to allow the comparison between grids is described along with a description of the approximations being made. Also ways of quantifying errors are defined.

### 2.1 Description of the numerical model used

The numerical model being used uses a finite volume approximation. A finite volume method predicts the values of variables on small volumes by calculating the quantities entering and leaving the cells using the flux on the faces (Weller, 2012).

With flow that is incompressible and non-divergent, the advection equation can be described in flux (conservative) form,

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

where  $\mathbf{u}$  is the velocity and  $\phi$  is the scalar being advected. The second term in this equation can be discretised using Gauss' divergence theorem to give the approximation,

$$V\nabla \cdot \mathbf{u}\phi \approx \int_V \nabla \cdot \mathbf{u}\phi dV = \int_S \phi \mathbf{u} \cdot d\mathbf{S} \approx \sum_f \phi \mathbf{u} \cdot d\mathbf{S} \quad (2.2)$$

where the flux,  $\phi \mathbf{u} \cdot d\mathbf{S}$ , is summed over all faces,  $f$ , of the cell and  $V$  is the volume of the cell (Weller, 2012). The numerical schemes described in this section give approximations to this equation. These are then used in OpenFOAM when running the solver scalarTransportFoam,

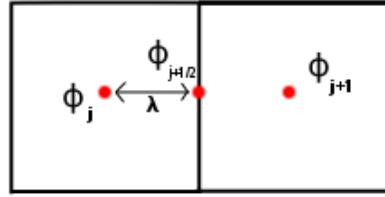


FIGURE 2.1: Example of where the values from the cell centres at  $\phi_j$  and  $\phi_{j+1}$  are interpolated onto the face between the two cells at  $\phi_{j+1/2}$ .  $\lambda$  is shown as the distance between the centre of the cell  $j$  and the face.

which solves for the transport of a passive tracer (OpenFOAM User’s Guide). The tracer concentration in cell  $(j, k)$  at time  $n$  is defined as  $\phi_{j,k}^{n+1}$

### 2.1.1 OpenFOAM

OpenFOAM is a C++ library that allows users to write code to solve partial differential equations on any structured or unstructured grid. OpenFOAM was used so that the same fluid flow solver was used for each of the different grids, representing different terrain-following coordinates, so that a good comparison can be made between each grid.

### 2.1.2 Description of the discretisations used

Numerical approximations are needed to find  $\phi$  on the faces for equation (2.2) from the values of  $\phi$  at the cell centres. Various numerical discretisations were used and tested to compare the terrain-following coordinates. Van Leer was the main spatial discretisation scheme used but linear and upwind were also tested to try to get a sense of what errors were occurring because of the numerical scheme. For the temporal discretisation, the main scheme was a second order implicit backwards scheme but the Crank-Nicolson scheme was also tested. These schemes are described below.

#### 2.1.2.1 Spatial discretisations

All of the spatial differencing schemes interpolate the value of  $\phi$  at the cell centres onto the cell face (as shown in Figure 2.1). Linear is a centred differencing scheme which is second order accurate on a uniform grid, but unbounded. Unbounded means that if the solution starts

between two values, it will not necessarily stay between those values - commonly because of spurious oscillations in the solution. A centred differencing scheme uses the values from the cells either side of the face to calculate fluxes. The approximation of the flux for equation (2.2) is given by (for one dimension) (Weller, 2012)

$$\phi_{j+1/2} = \lambda\phi_j + (1 - \lambda)\phi_{j+1} \quad (2.3)$$

where  $\lambda$  is the distance from the cell centre at  $\phi_j$  and the cell face between  $\phi_j$  and  $\phi_{j+1}$ , as shown in Figure 2.1.

Upwind is a first order bounded scheme which calculates the fluxes using information on the upwind direction (OpenFOAM Programmer's Guide). This is useful to combine with linear because of its bounded nature. The equation for the approximation of flux for upwind in one dimension is,

$$f_{j+1/2} = u \begin{cases} \phi_j & \text{for } u > 0 \\ \phi_{j+1} & \text{for } u < 0 \end{cases} \quad (2.4)$$

Van Leer is a limiter for a total variation diminishing (TVD) scheme. In a second order advection scheme, at each time step, there are unphysical, spurious oscillations between each grid point which can be measured. The measurement of the change in these oscillations is called the the total variation (TV). A TVD scheme is one in which the change in total variation between timesteps does not increase (Weller, 2012). The total variation can be measured using, in one dimension,

$$TV = \sum_{j=0}^{N-1} |\phi_{j+1} - \phi_j| \quad (2.5)$$

where N is the total number of points, so when the total variation does not increase with time (stays the same or decreases) the scheme is TVD (Weller, 2012). Upwind is the only linear TVD scheme, the others are all non-linear and described by the equation,

$$f_{j+1/2} = \Psi_{j+1/2} f_H + (1 - \Psi_{j+1/2}) f_L \quad (2.6)$$

where the flux,  $f_{j+1/2}$  is a weighted average of a  $f_H$ , a high order flux and  $f_L$ , a low order flux (Weller, 2012). In the case of van Leer used in OpenFOAM the high order flux is the linear differencing scheme and the low order flux is the upwind scheme where the orders are second

and first respectively (OpenFOAM Programmer’s Guide). The limiter function,  $\Psi$ , for van Leer is (Weller, 2012)

$$\Psi(r) = \frac{r + |r|}{1 + r} \quad (2.7)$$

where  $r$  is a function of the upwind local gradient,

$$r_{j+1/2} = \frac{\phi_j - \phi_{j-1}}{\phi_{j+1} - \phi_j} \quad (2.8)$$

which is for, as in the test case for this project, positive wind speeds (Weller, 2012).

In OpenFOAM, van Leer is a blend between centred linear and upwind (OpenFOAM Programmer’s Guide) so to understand the errors of van Leer, the errors of linear and upwind will also be studied.

When van Leer was initially used in this project, there were small negative values being produced of order around 0.5% of the initial tracer anomaly; this is small but because van Leer is TVD this should not be happening. This could have been because the van Leer limiter was not working correctly on the unstructured grids. To solve this, a strictly limited version of van Leer was used with an additional limiter to keep the tracer values between 0 and 1 (OpenFOAM User’s Guide). This meant the errors reduced closer to the values that would be expected because of round-off errors rather than ones due to the numerical scheme.

### 2.1.2.2 Temporal discretisations

Temporal discretisations are needed to advance the model solution forward in time. The main temporal discretisation scheme used was backwards, which is second order accurate and is implicit (OpenFOAM Programmer’s Guide). Implicitness means the scheme is stable for any timestep but there is the added complication and computational cost of having to invert a matrix at every timestep (Press et al., 1992). When solving the ordinary differential equation (ODE),

$$\frac{d\phi}{dt} = f(\phi) \quad (2.9)$$

which has the initial conditions,

$$\phi(t = 0) = \phi_0 \quad (2.10)$$

the backwards differencing scheme used can be described by the equation,

$$\phi^{n+1} = \frac{4}{3}\phi^n - \frac{1}{3}\phi^{n-1} + \frac{2}{3}\Delta t f(\phi^{n+1}) \quad (2.11)$$

where n represents the time level n at time  $n\Delta t$  (OpenFOAM Programmer's Guide). Second order backwards is also useful because it damps high frequency oscillations in the solution.

The Crank-Nicolson (CN) time stepping scheme is also implicit and second order accurate (OpenFOAM Programmer's Guide) but does not damp high frequency oscillations. CN uses a trapezoidal method to discretise equations (Weller, 2012). The equation for CN is given by,

$$\phi^{n+1} = \phi^n + \frac{1}{2}\Delta t[f(\phi^n, t^n) + f(\phi^{n+1}, t^{n+1})] \quad (2.12)$$

where  $f(\phi, t) = \frac{d\phi}{dt}$  and  $\phi^n$  is the value of the tracer at time level  $t^n$  (Langer et al., 2013).

CN can be off-centred, which blends the CN scheme with implicit Euler (OpenFOAM Programmer's Guide). This damps the high frequency oscillations, but reduces the order of accuracy to one. No off-centring was used for these simulations to maintain second order accuracy in the time discretisation.

### 2.1.3 Courant number on an unstructured grid

The Courant number gives an idea for what spatial and temporal resolutions a model is stable for. In one-dimension the courant number is defined as (Weller, 2012),

$$\alpha = \frac{u\Delta t}{\Delta x} \quad (2.13)$$

In more than one dimension, OpenFOAM uses a generalisation of the Courant number for arbitrary dimensions on unstructured grids,

$$\alpha = \frac{\sum |\mathbf{u} \cdot d\mathbf{S}|}{V} \Delta t \quad (2.14)$$

where the sum is over all the dot products of the velocity and surface vector on all the faces of the grid box, which has volume  $V$  (OpenFOAM Software, (OpenFOAM, 2011)).

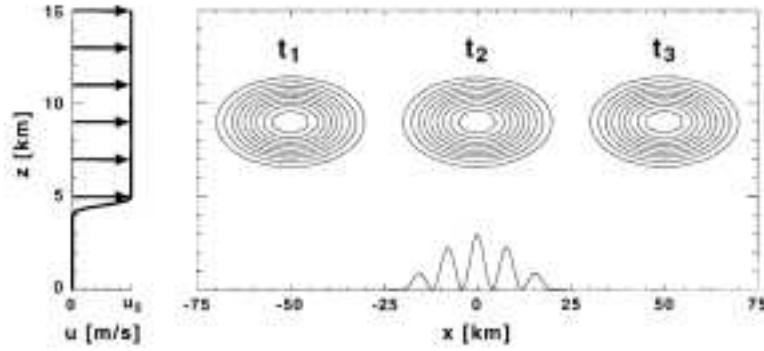


FIGURE 2.2: Initial conditions as described in Schär et al. (2002) also showing the shape of the terrain used later when terrain is introduced (Figure from Schär et al. (2002)).

## 2.2 Test case used in this project

To be able to compare the different grids, a consistent test is needed. When comparing the test case run with the different grids, the error is expected to be highest for BTF, then HTF, then SLEVE, then the created near-orthogonal coordinates. The resulting comparison between the errors produced for each grid and over flat ground is presented in Chapter 3.

The test case that was used for comparing the different coordinate systems is described in Schär et al. (2002) and is shown in Figure 2.2. This is a two-dimensional advection of a passive tracer, described by equation (2.16), through a grid which is distorted due to the presence of orography at the surface. The wind profile is described by equation (2.15). The analytical solution to this problem is that the tracer keeps its shape perfectly and moves at the same speed as the wind. The difficulty with this problem arises because distortions in the grid causes the flow to not be aligned with the grid. Also, additional errors can be introduced due to effects like non-orthogonality, skewness and non-uniformity. This simple two-dimensional advection test case is used because it is useful to quantify the errors occurring because of transformations in the grid (Schär et al., 2002). The winds are not being solved for because this is just meant to be a simple first step for a comparison between grids.

### 2.2.1 Initial conditions

The wind profile is specified as

$$u(z) = u_0 \begin{cases} 1 & \text{for } z_2 \leq z \\ \sin^2\left(\frac{\pi}{2} \frac{z-z_1}{z_2-z_1}\right) & \text{for } z_1 \leq z \leq z_2 \\ 0 & \text{for } z \leq z_1 \end{cases} \quad (2.15)$$

where  $u_0 = 10\text{ms}^{-1}$ ,  $z_1 = 4\text{km}$  and  $z_2 = 5\text{km}$ . The wind profile has no wind below 4km which then smoothly increases up to a height of 5km where it reaches the maximum value of  $u_0$  and then stays constant above this level. The wind is orientated horizontally only. The stagnant air below 4km was at this height so that when orography was introduced there would be no complications with the terrain (which had a maximum height of 3km) such as defining non-divergent winds along the surface.

The initial tracer is described by

$$\phi(x, z) = \phi_0 \begin{cases} \cos^2\left(\frac{\pi r}{2}\right) & \text{for } r \leq 1 \\ 0 & \text{otherwise} \end{cases} \quad (2.16)$$

with

$$r = \left( \left( \frac{x - x_0}{A_x} \right)^2 + \left( \frac{z - z_0}{A_z} \right)^2 \right)^{1/2} \quad (2.17)$$

with  $\phi_0 = 1$  is the maximum tracer value and half widths  $A_x = 25\text{km}$  and  $A_z = 3\text{km}$ . At  $t = 0$  the tracer centre is set to the position  $x_0 = -50\text{km}$  and  $z_0 = 9\text{km}$ .

### 2.2.2 Domain

The domain for the test case has a length of 300km and a depth of 25km which is split into a 300x50 grid meaning  $\Delta x = 1\text{km}$  and  $\Delta z = 0.5\text{km}$ . With this test case, at  $t = 0\text{s}$  the tracer centre would be at -50km, at  $t = 5000\text{s}$  the tracer centre would be at 0km and at  $t = 10000\text{s}$  the tracer would be centred at 50km as shown in Figure 2.2.

The domain was produced according to the size and spatial resolution described above. However, OpenFOAM generates grids that are three-dimensional so when generating the grid for

a two-dimensional problem, a depth of one is used (y-direction) and the faces on either side of the domain (faces orthogonal to the y-axis) are empty.

### 2.2.3 Terrain

The terrain profile used was from Schär et al. (2002) which is described by,

$$h(x) = \cos^2\left(\frac{\pi x}{\lambda}\right) h^*(x) \quad (2.18)$$

where

$$h^*(x) = \begin{cases} h_0 \cos^2\left(\frac{\pi x}{2a}\right) & \text{for } |x| \leq a \\ 0 & \text{for } |x| > a \end{cases} \quad (2.19)$$

where  $\lambda = 8\text{km}$  and is the wavelength of the small scale component of the terrain;  $h_0 = 3\text{km}$  and is the maximum height of the terrain; and  $a = 25\text{km}$  and is the half width of the component of the terrain  $h^*$ . The initial conditions with the terrain profile are shown in Figure 2.2.

This terrain profile simulates the effect of complex topography by having a fine wave-like features of a similar scale to the terrain used in high resolution, regional numerical weather prediction (Schär et al., 2002).

## 2.3 Creating the terrain-following coordinate based grids

The grids for all the terrain-following coordinates were needed to set up the model, so are included here, however the production of these grids also forms part of the results: the grids are discussed in more detail in Sections 3.3 and 3.4 of the results.

The non-orthogonal grids (BTF, HTF and SLEVE) were implemented using the equations in the previous chapter (equations (1.6), (1.7) and (1.8) for BTF, HTF and SLEVE respectively). For HTF, the scale factor used was the same used in Schär et al. (2002) of  $s = 8\text{km}$ . For SLEVE the two scale factors were again used as in Schär et al. (2002) of  $s_1 = 15\text{km}$  for the large scale and  $s_2 = 2.5\text{km}$  for the small scale. Also in SLEVE the terrain effects were split into large scale terrain contribution,  $h_1$ , and small scale contribution,  $h_2$ ,

$$h_1(x) = \frac{1}{2}h^*(x) \quad (2.20)$$

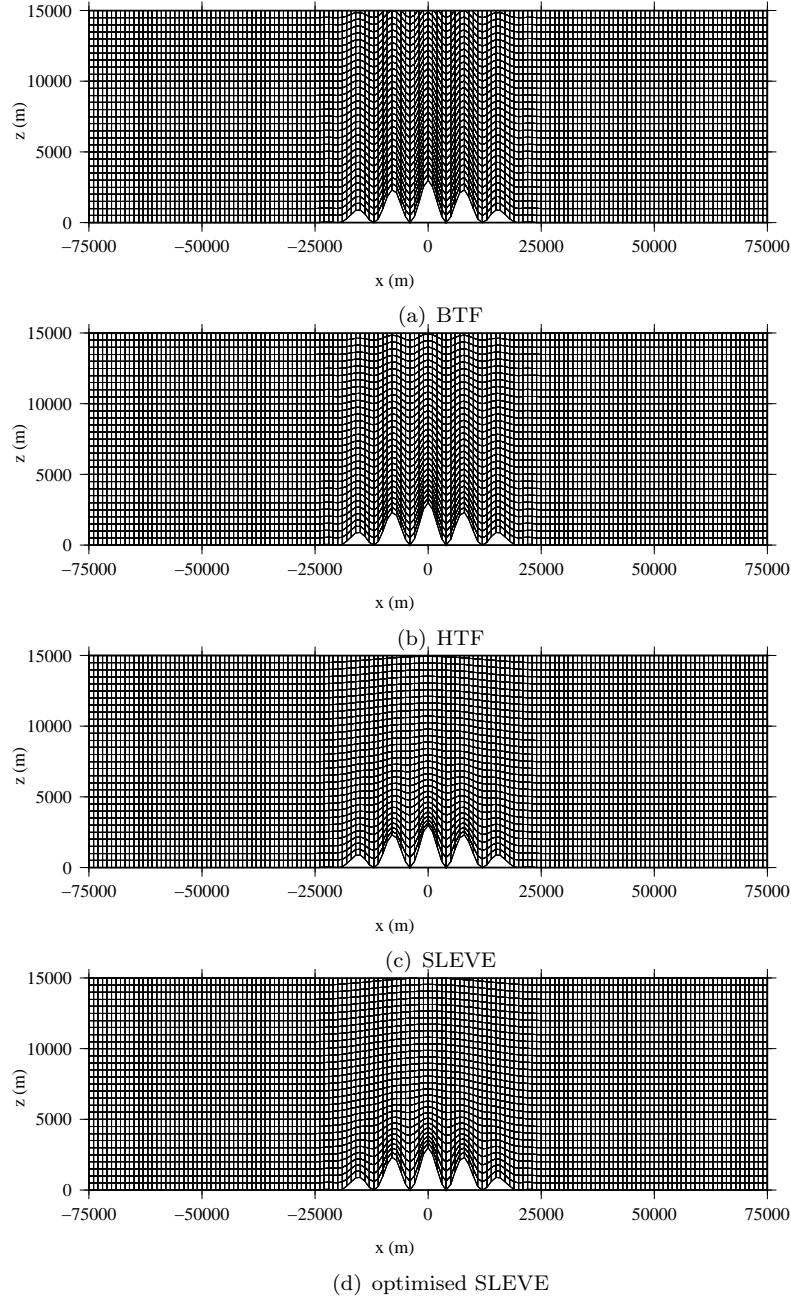


FIGURE 2.3: Produced non-orthogonal grids showing (a) BTF, (b) HTF, (c) SLEVE and (d) optimised SLEVE for the terrain profile described in Section 2.2.3.

$$h_2(x) = h(x) - h_1(x) \quad (2.21)$$

as described in Schär et al. (2002). The grids created are shown in Figure 2.3.

The results for each of these non-orthogonal grids are discussed in Section 3.3.

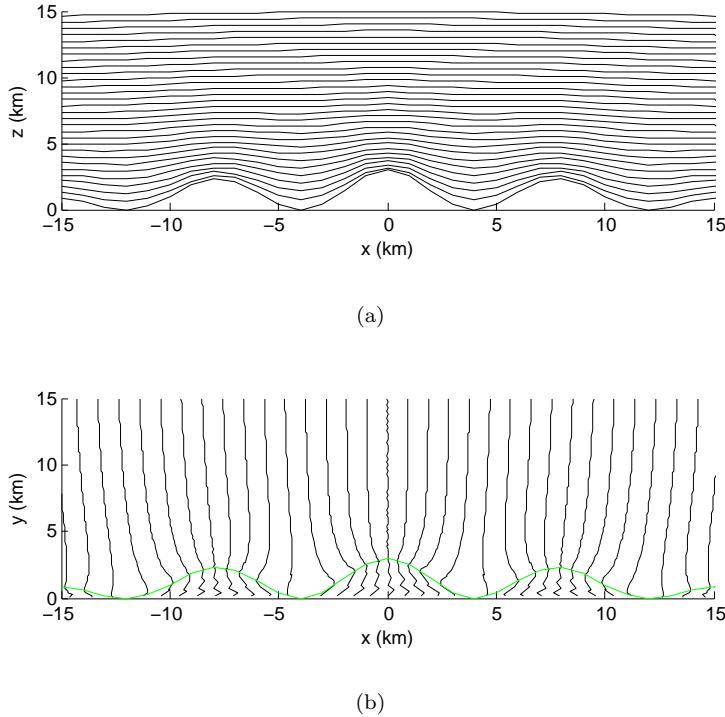


FIGURE 2.4: (a) contours from SLEVE (b) the velocity potential contours used to create the new near-orthogonal grid in Figure 2.6. The green line in (b) shows the terrain.

## 2.4 Near-orthogonal grid generation

To generate a more orthogonal grid than the grids based on terrain-following coordinates, a technique similar to the field technique of solving elliptical equations was implemented. Normally an elliptic equation would be solved for both the horizontal and vertical grid lines but for this grid it was decided to use the horizontal grid lines from the SLEVE coordinates for simplicity and it appears to have not been done before. The vertical grid lines were calculated by solving for the velocity potential using the equation,

$$\nabla^2 \chi = 0 \quad (2.22)$$

where  $\chi$  is the velocity potential. The contours were created using OpenFOAM with the boundary conditions for the velocity potential set at  $\chi = 150m^2s^{-1}$  at the inlet of the domain and  $\chi = -150m^2s^{-1}$  at the outlet of the domain. At the top and bottom of the domain the gradient of the velocity potential was set to be zero. The contour intervals were set to give the approximately same spacing as for the terrain-following coordinates to keep the horizontal resolution around 1km.

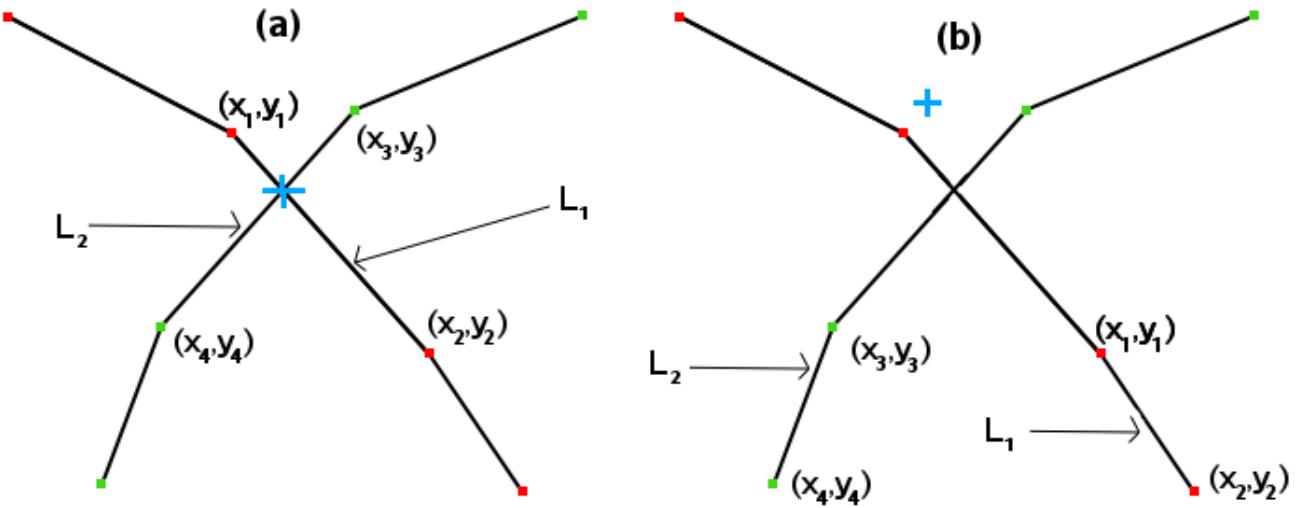


FIGURE 2.5: Line segments  $L_1$  and  $L_2$  with the intercept between them shown by the blue +. Two cases are shown (a) where the intercept has been calculated correctly and (b) where the initial guess of the line segments is wrong so the intercept was not correctly found.

Using the contours of  $\chi$  (shown in Figure 2.4) the intercept was found with the SLEVE layers by running code to find the points on the contour nearest to where the intercept was thought to be. Let the line segment between the points on the velocity potential contour ( $(x_1, y_1)$  and  $(x_2, y_2)$ ) be  $L_1$  and the line segment between the points on the contour from SLEVE ( $(x_3, y_3)$  and  $(x_4, y_4)$ ) be  $L_2$  (shown in Figure 2.5) then the intercept can be calculated using the method described by Weisstein (2013b). In this method the intercept between the two lines,  $L_1$  and  $L_2$ , is given by the determinants,

$$P_x = \frac{\begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{vmatrix} \begin{vmatrix} x_1 & 1 \\ x_2 & 1 \\ x_3 & 1 \\ x_4 & 1 \end{vmatrix}}{\begin{vmatrix} x_1 & 1 \\ x_2 & 1 \\ x_3 & 1 \\ x_4 & 1 \end{vmatrix} \begin{vmatrix} y_1 & 1 \\ y_2 & 1 \\ y_3 & 1 \\ y_4 & 1 \end{vmatrix}}, \quad P_y = \frac{\begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{vmatrix} \begin{vmatrix} y_1 & 1 \\ y_2 & 1 \\ y_3 & 1 \\ y_4 & 1 \end{vmatrix}}{\begin{vmatrix} x_1 & 1 \\ x_2 & 1 \\ x_3 & 1 \\ x_4 & 1 \end{vmatrix} \begin{vmatrix} y_1 & 1 \\ y_2 & 1 \\ y_3 & 1 \\ y_4 & 1 \end{vmatrix}} \quad (2.23)$$

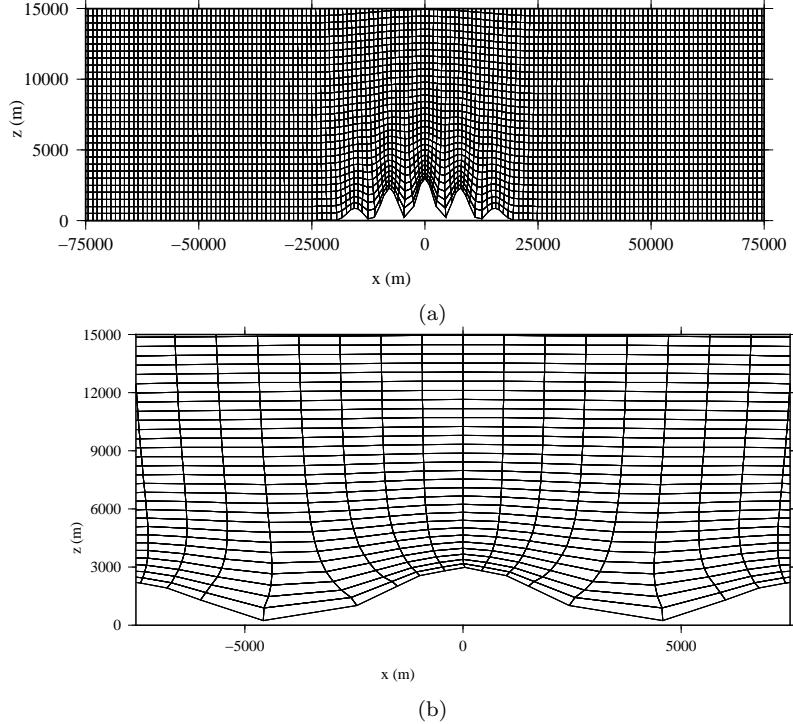


FIGURE 2.6: The more orthogonal grid produced (a) and a zoomed in portion of the grid (b) to show the curvature of the vertical lines more clearly.

where  $P_x$  and  $P_y$  are the points of the intercept. This simplifies to

$$(P_x, P_y) = \left( \frac{(x_1y_2 - y_1x_2)(x_3 - x_4) - (x_1 - x_2)(x_3y_4 - y_3x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)}, \frac{(x_1y_2 - y_1x_2)(y_3 - y_4) - (y_1 - y_2)(x_3y_4 - y_3x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)} \right) \quad (2.24)$$

which can then be implemented to find the intercept between the lines  $L_1$  and  $L_2$ . This method assumes that the lines are of infinite length so it is possible that if the initial guess of where the points on the contours are is wrong then the intercept will not be between the points on  $L_1$  and  $L_2$  as shown in Figure 2.5(b) (the intercept found needs to be between the points  $(x_1, y_1)$  and  $(x_2, y_2)$  on  $L_1$  and  $(x_3, y_3)$  and  $(x_4, y_4)$  on  $L_2$  to ensure that the intercept lies on both line segments, where the intercept is pictured in Figure 2.5(a), rather than not on these segments because the initial guess was wrong as shown in Figure 2.5(b)). This meant that the code then checked that the intercept was between the points on each line and if it was not then the initial guess of the points was moved along the line in the right direction, then the process was repeated. When all the intercepts had been found a grid was generated from them which is shown in Figure 2.6.

The results comparing the error for this grid is in Section 3.3.4 of the results.

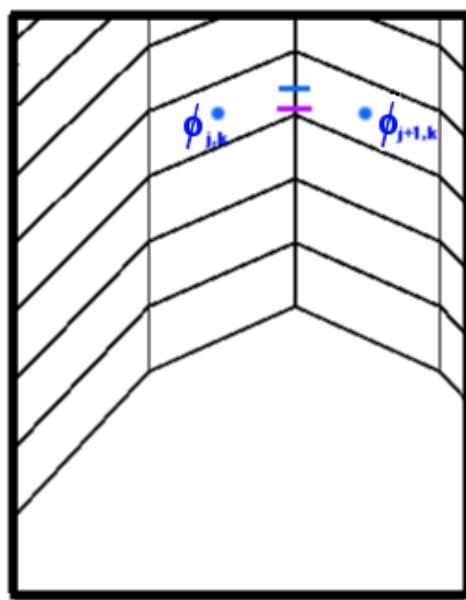


FIGURE 2.7: A section from the grid for BTF showing errors because of horizontal differencing due to skewness in the grid.

## 2.5 Errors with scalar transport due to horizontal differencing

A problem arises from the slant in the cells when running scalar transport (shown in Figure 2.7) occurring above the peaks in the terrain, or where there is high curvature in the terrain. This is because the flux, calculated from the centre of the cells, is assumed to be at the centre of the face between the two cells shown by the blue line, whereas in fact the flux calculated is a better representation of the flux at the position of the purple line - from now this effect will be known as skewness. Skewness can be reduced by reducing the aspect ratio ( $\Delta x / \Delta z$ ) by decreasing the horizontal grid spacing (Lundquist et al., 2010) which Schär et al. (2002) showed would increase the accuracy of the solution. This was, however, not implemented in this project because of the emphasis on comparing the coordinate systems rather than increasing the resolution.

## 2.6 Description of error diagnostics

Williamson et al. (1992) defined a series of tests for numerical approximations for shallow water equations in spherical geometry. Some of the errors defined by Williamson et al. (1992) for the advection of a cosine bell over a pole, were useful to quantify the error produced from

the test case in this project despite not being in spherical geometry or using the shallow water equations. The reason that these are still useful is that the error quantities are defined using either integrals over the entire domain, mean values, or maxima and minima from the tracer field, which are all still relevant quantities.

Normalised global errors can be defined as

$$l_1 = I[|\phi(x, z) - \phi_T(x, z)|]/I[|\phi_T(x, z)|] \quad (2.25)$$

$$l_2 = \{I[(\phi(x, z) - \phi_T(x, z))^2]\}^{1/2}/\{I[\phi_T(x, z)^2]\}^{1/2} \quad (2.26)$$

$$l_\infty = \max_{\text{all } x, z} |\phi(x, z) - \phi_T(x, z)| / \max_{\text{all } x, z} |\phi_T(x, z)| \quad (2.27)$$

where  $I$  defines a discrete approximation to the global integral

$$I(\phi) = \int \int \phi(x, z) dx dz \quad (2.28)$$

where  $\phi(x, z)$  are the values of the tracer field from the model, and  $\phi_T(x, z)$  are the values of the analytical solution. The error defined by  $l_1$  contains the most information from the smaller errors in the solution, whereas  $l_2$  puts more weight on the larger errors and  $l_\infty$  only includes information on the maximum values. So on a scale of the information included from the smallest errors are taken most into account, at one end would be  $l_1$  and the other end would be  $l_\infty$  where the latter includes no information from the small errors.

Williamson et al. (1992) also gives equations for calculating the normalised mean,

$$M = (\bar{\phi} - \bar{\phi}_T)/\bar{\phi}_0 \quad (2.29)$$

where  $\phi_0(x, z)$  is the initial tracer conditions and  $\bar{\phi}$  denotes the mean; normalised variance,

$$V = \{I[(\phi - \bar{\phi})^2] - I[(\phi_T - \bar{\phi}_T)^2]\}/I[(\phi_0 - \bar{\phi}_0)^2] \quad (2.30)$$

and normalised maximum and minimum in the error,

$$\phi_{max} = (\max_{\text{all } x, z} \phi(x, z) - \max_{\text{all } x, z} \phi_T(x, z))/\Delta\phi \quad (2.31)$$

$$\phi_{min} = (\min_{\text{all } x, z} \phi(x, z) - \min_{\text{all } x, z} \phi_T(x, z))/\Delta\phi \quad (2.32)$$

where  $\Delta\phi$  is the difference between the maximum and minimum values of the tracer in the initial conditions. These diagnostics were used on the test cases in Section 3.5 of the results.



# Chapter 3

## Results

In this chapter, the results of advecting a tracer over orography using the grids based on the terrain following coordinates in the previous chapter and the near-orthogonal grid will be shown. These will be compared with results without orography using various discretisations. Also covered will be some more detailed analysis of the grids and the error.

### 3.1 Tracer advection over flat ground

Figure 3.1 shows the results from the model run with no orography using the van Leer spatial differencing scheme. As expected from the literature (Schär et al., 2002) the tracer was centred at -50km, 0km and 50km for times 0, 5000s and 10000s respectively. The tracer also keeps its shape well without any obvious dispersion with only a small amount of diffusion shown by the small decay in the maximum of the tracer. Figure 3.2 shows the error fields for the different differencing schemes used. These were used to show where the errors were coming from with van Leer by looking at the errors produced using upwind and linear, because, as previously discussed, van Leer is a combination of upwind and linear.

Before moving on to the discussion of the results for advection over flat ground with linear, upwind and van Leer, it is useful to look at the Fourier transforms of the tracer field at the beginning and the end of each model run for each of the spatial differencing schemes. This allows a comparison of the power in different wavelengths between the initial conditions and the tracer after advection, which can be used to see whether dispersion or diffusion occurs. To analyse the phase speed errors and diffusion, a Fourier transform was taken of a section

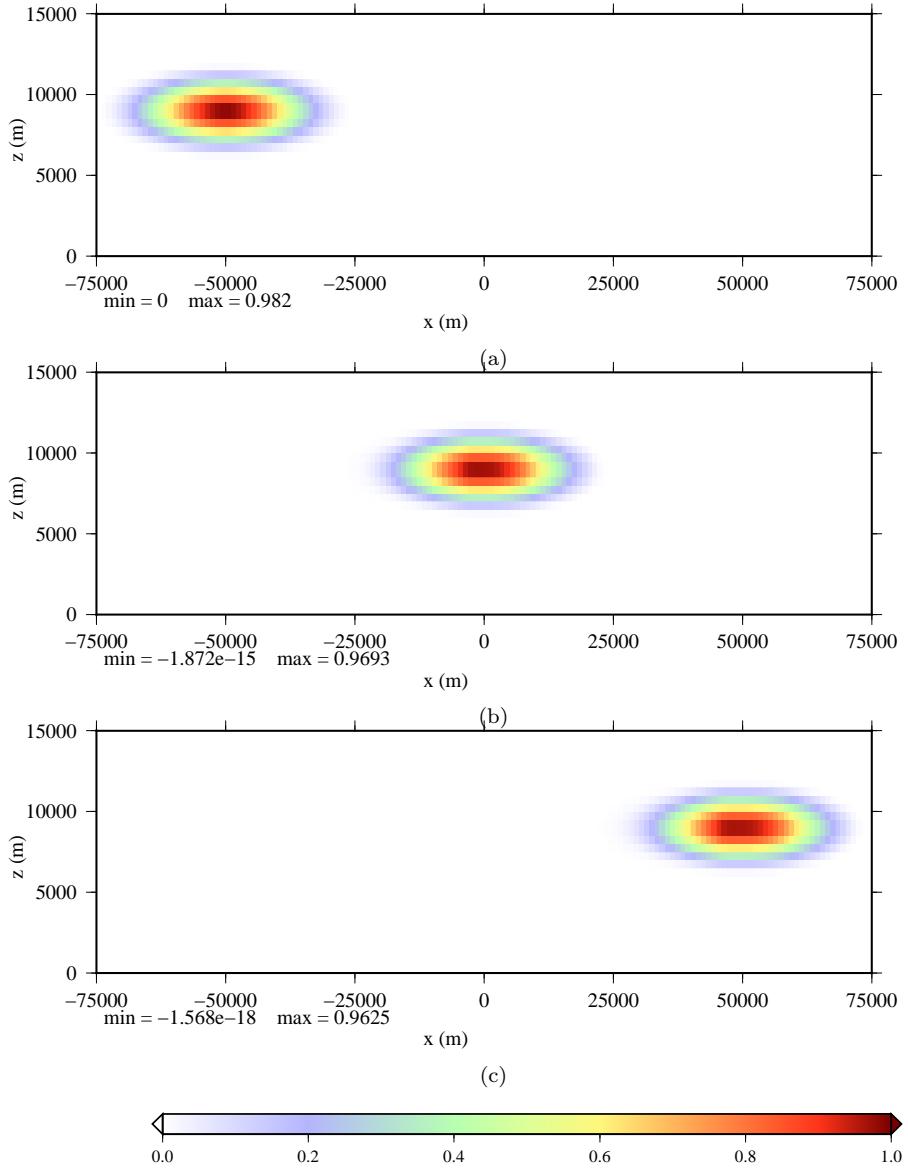


FIGURE 3.1: Results from the tracer advection run without orography. The y-axis shows height above the surface in metres and the x-axis shows the distance along the ground in metres. (a) shows the initial conditions of the tracer at  $t=0$ s, (b) shows the solution at  $t=5000$ s and (c) shows the solution at  $t=10000$ s. The minimum and maximum values from the tracer field are shown at the bottom left of each graph and the colours show the value of the tracer.

through the centre of the initial tracer field. To determine whether dispersion was present, this power spectrum was used in conjunction with the phase ratios for upwind and linear in Figure 3.3. Von Neumann stability analysis was carried out to calculate the phase speed of the numerical method in comparison to the exact phase speed and amplification factors were also calculated (shown in Figure 3.4). The upwind scheme can be described by the backwards in time backwards in space (BTBS) finite differencing formula,

$$3\phi_j^{n+1} - 4\phi_j^n + \phi_j^{n-1} = -2\alpha(\phi_j^{n+1} - \phi_{j-1}^{n+1}) \quad (3.1)$$

and the linear scheme can be described by the backwards in time centred in space (BTCS) finite differencing formula,

$$3\phi_j^{n+1} - 4\phi_j^n + \phi_j^{n-1} = -\alpha(\phi_{j+1}^{n+1} - \phi_{j-1}^{n+1}) \quad (3.2)$$

The numerical phase speed is compared to the analytical phase speed using a Courant number of 0.1. To determine whether artificial diffusion has occurred, the power spectrums of the initial and final tracer field were compared to see if there were any changes. This is because diffusion will remove power at shorter wavelengths causing the tracer to spread out. This can also be used with a comparison of the maximum tracer values and the amplification factor to see whether diffusion has occurred. If the amplification factor is equal to one the scheme is stable and non-diffusive and if it is less than one the scheme is stable but diffusive (Weller, 2012). The phase ratios in Figure 3.3 show the physical modes of BTBS and BTCS moving too slowly for shorter wavelengths then tending towards one (showing them moving at the correct speed) for longer wavelengths and the computational modes for each are moving backwards slower than the true speed. For each mode the waves are stationary for a wavelengths of twice the grid spacing. The amplification factors in Figure 3.4 show that the physical mode of BTCS is stable and non-diffusive and the computational mode is stable and very diffusive. For BTBS, the physical mode is stable but diffusive, more so for shorter wavelengths and the computational mode is stable and very diffusive.

The power spectrum of the slice through the initial conditions is shown in Figure 3.5, along with the power spectrum from the solution after the scalar transport for each of the tests with no orography (linear, upwind and van Leer). This was produced in Matlab using the inbuilt FFT (fast Fourier transform) function. A power spectrum shows the power in each wavelength of a curve. To get the power spectrum, a sample frequency of  $1/\Delta x$  was used

with a window length along the whole slice. The power is then calculated by multiplying the Fourier coefficients with their complex conjugate, then one side of the power spectrum is plotted. Only one side was plotted because the Fourier transform produces a symmetric power spectrum about the Nyquist frequency and so only half needs to be plotted to show the component frequencies. Figure 3.5 shows that the initial conditions are composed mainly of longer wavelength waves, shown by the high power at lower frequencies, which make up the peak in the tracer. Also, there are many smaller contributions from shorter wavelengths, shown by lower power on higher frequencies, which cause destructive interference away from the peak, totally cancelling each other out causing the flattened area of the tracer profile. These power spectrums will be discussed in the sections below.

### 3.1.1 Tracer advection using centred, linear, advection

The error for the linear scheme (shown in Figure 3.2(a)) is small compared to the initial amplitude of the tracer - around 2.5% of the initial maximum in the tracer. There are obvious dispersion errors shown by the alternating regions of positive and negative error. This pattern continues, decaying in amplitude, all the way back to the initial position of the tracer.

Looking at the power spectrum for linear in Figure 3.5, it is very similar to that of the initial conditions. This suggests that the error for linear is mainly due to dispersion errors rather than diffusion errors because the power spectrum shows, generally, the same frequencies at almost the same power as the initial conditions, showing that it is composed of almost the same waves. These waves have just advanced at different rates, producing the error pattern seen due to phase errors. Also, the amplification factor for BTCS (in Figure 3.4) being equal to one for the physical mode reinforces the fact that there is little diffusion occurring.

The pattern seen in the error field (Figure 3.2(a)) for the linear scheme can be explained further with the help of the phase speed ratios shown in Figure 3.3. There is a clear wave-like structure in the error in Figure 3.2(a) with maximum amplitude near the centre of the tracer which decays in amplitude as you move further away. This can be explained by looking at the phase speed ratios for the physical mode in Figure 3.3. The shorter wave lengths move slower than the analytical solution, and as the wavelengths get longer they tend more towards the correct speed; this explains the wave-like pattern in the solution in Figure 3.2(a) because the shorter wavelengths are being retarded and so can be seen behind the main tracer position. The main error seen in Figure 3.2(a) suggests that the tracer is moving too slowly, shown by

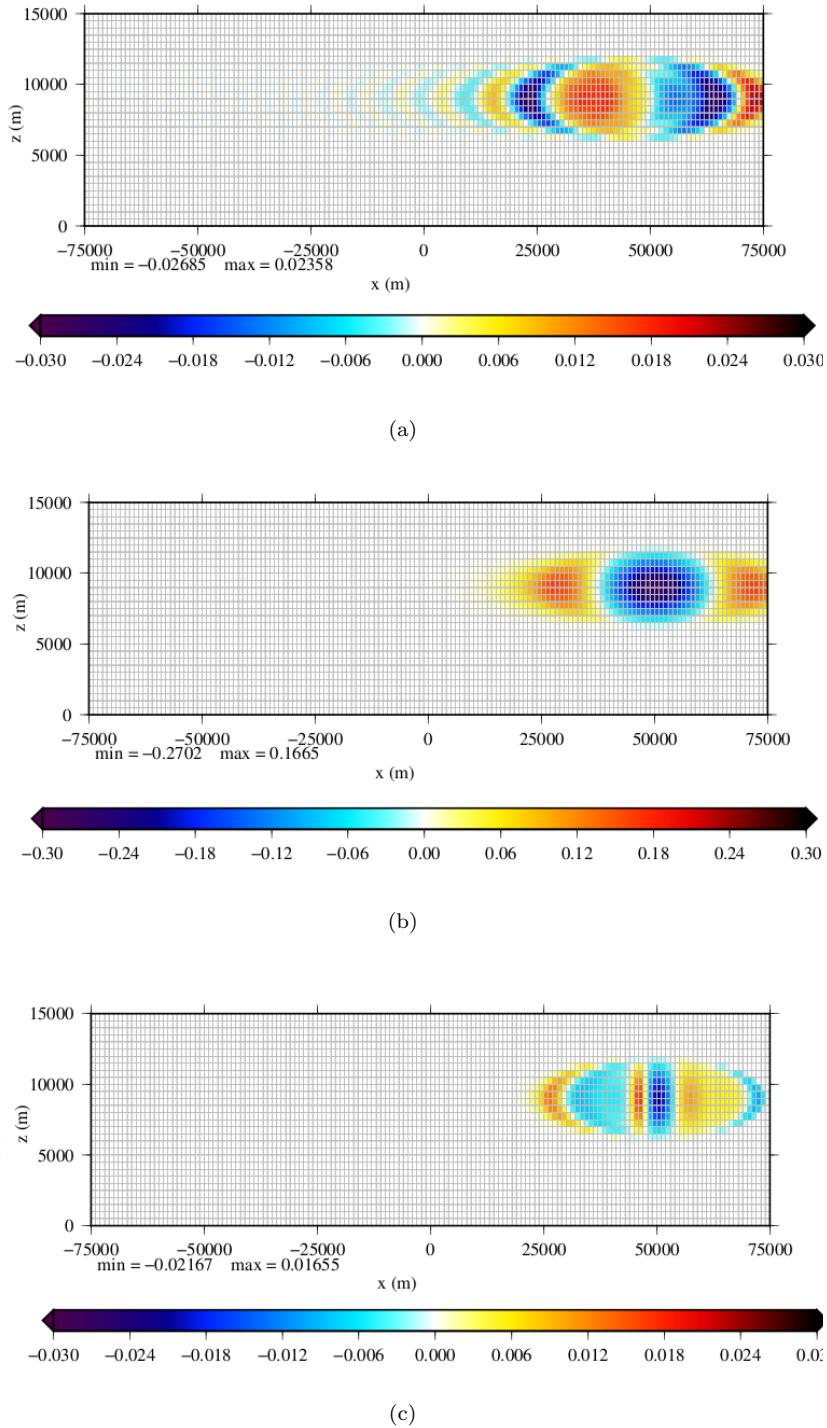


FIGURE 3.2: Error at  $t=10000$ s (difference between the numerical solution and the analytical solution) run with the linear spatial differencing scheme (a), the upwind differencing scheme (b) and the van Leer differencing scheme (c) for a section of the domain. The colours show the error.

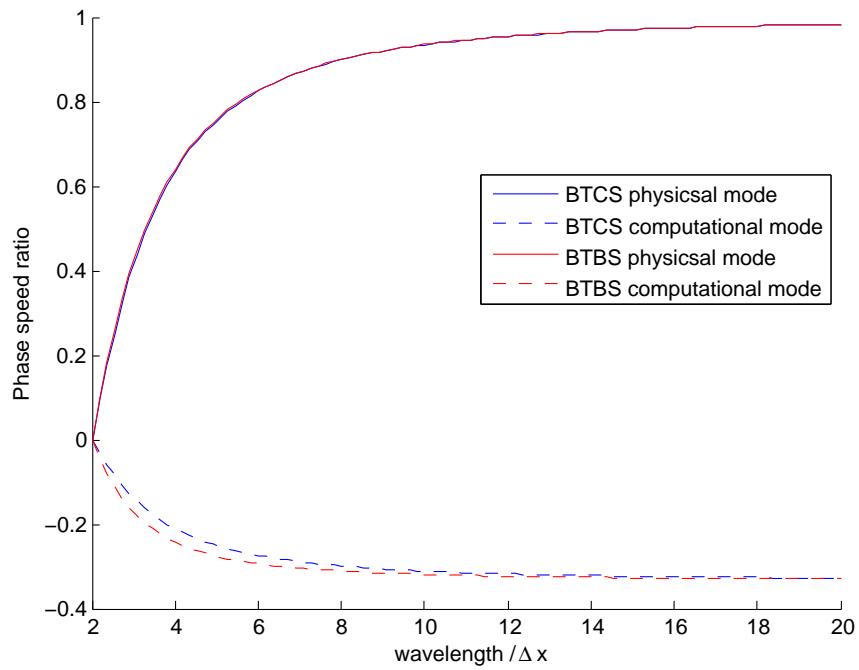


FIGURE 3.3: Phase speed ratio for BTBS (red) and BTCS (blue).

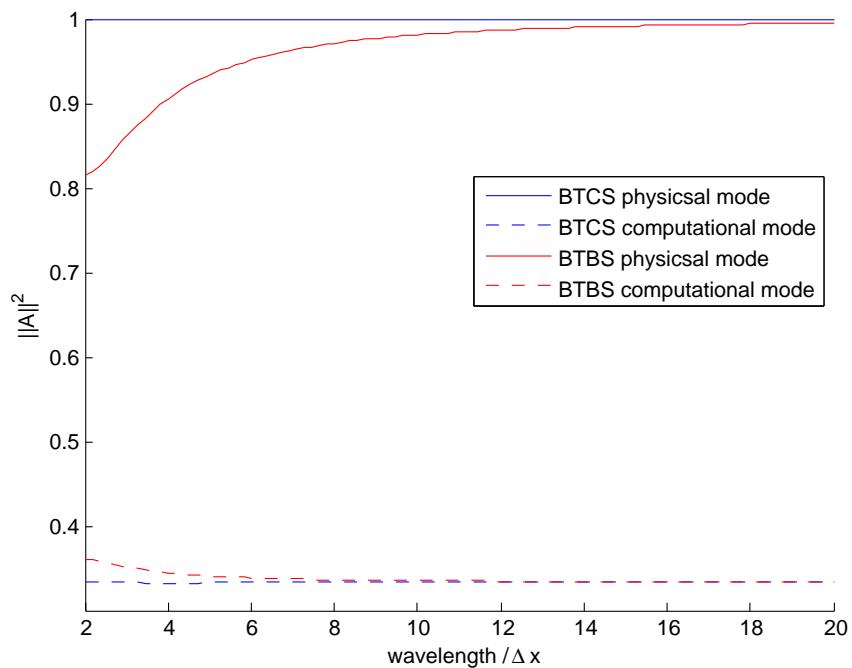


FIGURE 3.4: Amplification factor for BTBS (red) and BTCS (blue).

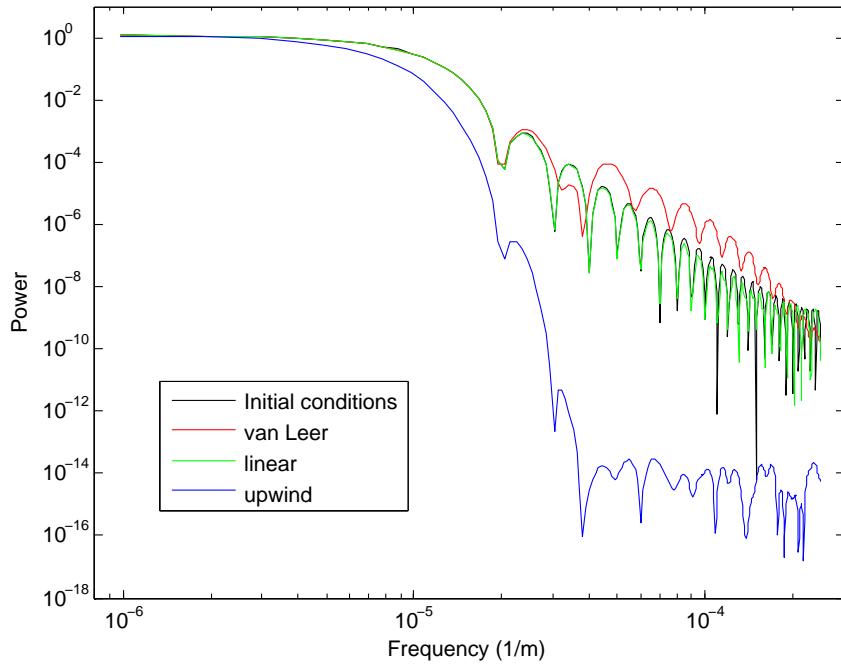


FIGURE 3.5: Power spectrum of the tracer initial conditions (shown in black) with the power spectrums for the tracer after the advection also shown: van Leer in red, linear in green and upwind in blue.

the large area of negative error from around 50km to 70km. The positive error in front of this suggests that a small amount of diffusion is occurring, which can be seen by looking closely at the difference in the power spectrums for the initial conditions and linear in Figure 3.5, and noting the slight difference which could be caused by a small amount of diffusion. However, the amplification factor for BTCS, in Figure 3.4, shows that the physical mode for BTCS is not diffusive so this may not be because of diffusion. This is also not due to dispersion because the phase speed ratio in Figure 3.3 never increases above one, so no waves move faster than the true speed of the tracer. These two reasons mean that it is not clear what has caused the positive error in the front of the tracer error.

The effect of the computational mode for BTCS in Figure 3.3 can been seen by looking at the whole width of the domain for the tracer advection run with linear shown in Figure 3.6. In this figure, there is an area of error centred at around -150km; this is there because of the computational mode advecting some of the tracer backwards. From the phase speed ratios in Figure 3.3 the computational mode should be advecting backwards at around a third of the speed of the wind, however, it moves backwards at around the same speed, having moved the same distance backwards as the rest of the tracer has moved forwards. It is unknown what

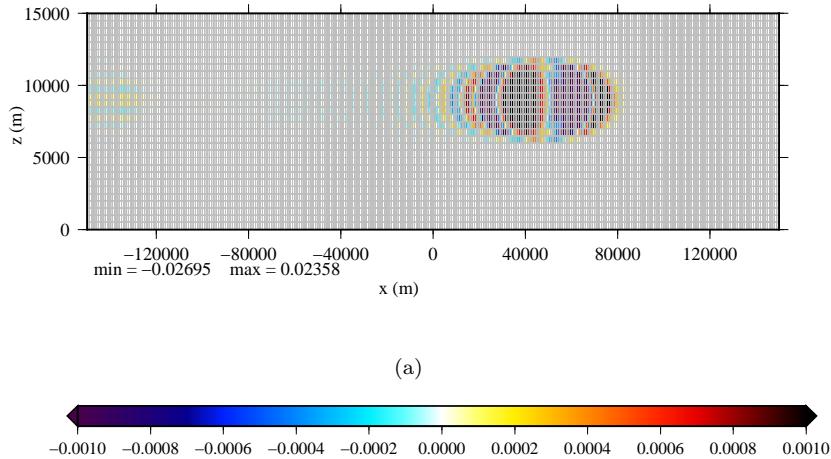


FIGURE 3.6: More detailed look at the error for linear over the width of the whole domain, note the different scale for the error compared to Figure 3.2(a).

has caused this increased phase speed in the computational mode.

### 3.1.2 Tracer advection using first order upwind advection

The errors for upwind, shown in Figure 3.2(b), are much larger than for linear; this was expected since upwind is first order and linear is second order. The errors reach up to about 27% of the maximum in the initial tracer and the negative pattern in the middle of the tracer and positive on either side indicates that there has been significant diffusion occurring, possibly hiding any phase errors. The power spectrum for upwind in Figure 3.5 shows that the composition of the tracer after the model was run is very different to that of the initial conditions because the power at short wavelengths has been destroyed. This shows lots of diffusion has occurred as was thought from the error in Figure 3.2(a).

There could, of course, have been phase errors but the amplitude would have been too small to see over the diffusion errors - the errors for upwind are an order of magnitude greater than those for linear. Figure 3.3 indicates that there should be phase errors at shorter wavelengths (by looking at the phase speed ratios for BTBS) but by looking at Figure 3.5 the shorter wavelengths have a much lower power and so the high frequency waves have been damped.

Figure 3.3 also shows a computational mode for BTBS, which was not seen when the error for upwind was looked at in more detail (error over the whole domain for upwind is not shown). The lack of an observed computational mode for upwind is either because upwind is so diffusive so the computational mode is diffused out (this is backed up by how diffusive

the physical and computational modes are for BTBS in Figure 3.4), or could be because the computational mode is manifesting from the spatial discretisation rather than the temporal one (linear produces a computational mode because it is second order in space as well as one because of the time discretisation whereas upwind produces one only because of the time discretisation).

### 3.1.3 Tracer advection using a van Leer limiter

The error for van Leer (Figure 3.2(c)) looks significantly different to the other two schemes, and shows phase errors of a different shape to those seen in the linear scheme.

With van Leer being a blended differencing scheme between upwind and linear (OpenFOAM Programmer's Guide), it would make sense that the errors shown in Figure 3.2(c) could be explained by considering the errors in upwind and linear. The error for van Leer shows that there are phase errors involved because of the wave-like structure in the tracer error. The phase speed ratios are very similar for upwind and linear which suggests that the phase speed ratio is similar for van Leer. This would give a similar pattern in the error, but since van Leer is TVD, higher frequency waves are filtered out. This can be seen by the lack of the tail in the error seen in Figure 3.2(a). This is due to the van Leer limiter suppressing grid scale oscillations, but this does not stop the retardation of some of the wavelengths, causing the error seen.

It can be seen from Figure 3.5 that the composing waves of the results from the model running with van Leer are quite different to the initial conditions and to linear - showing higher power for some wavelengths and for high frequencies the power drops below the power for the initial conditions, suggesting that some diffusion has occurred. However, the error shown in Figure 3.2(c) suggests that the error is more down to dispersion than diffusion, because of the power spectrum for van Leer more closely resembling linear than upwind, and the removal of power at high frequencies is scale selective, thus the diffusion errors are smaller than for upwind.

Because, in the maths, the computational mode for linear and upwind was caused by the time discretisation it could be expected that there would be a computational mode for van Leer (for all of the tests without orography the backwards time discretisation was used). However, when the error for van Leer was looked at in more detail (not shown) there was no evidence of

this. This is probably because van Leer filters out grid scale oscillations so the computational mode could not be seen.

Due to the superior behaviour of van Leer over linear and upwind, van Leer will be used for the advection over orography. The errors that are produced from van Leer have also been explained with the help of upwind and linear.

### 3.2 Properties of the grids used for tracer advection over orography

Attributes of each grid are shown in Table 3.1.  $\Delta z_{min}$  shows the amount of compression that occurred due to the effect of the terrain decreasing faster with height for some coordinates - especially SLEVE.  $z_{min}$  for the optimised version of SLEVE shows much less compression at the peaks than SLEVE, as it was designed to (Schär et al., 2002).  $\Delta z_{max}$  shows that for BTF and HTF the height between levels only decreases, whereas for SLEVE and the coordinates derived from it (optimised SLEVE and the near-orthogonal grid)  $\Delta z_{max}$  increases due to the small scale and large scale portions of the terrain decaying at different rates.

$\Delta x_{min}$  and  $\Delta x_{max}$  for all the non-orthogonal coordinates are constant because the vertical coordinate lines are parallel and uniform, whereas for the near-orthogonal grid  $\Delta x_{min}$  decreases because the vertical grid lines curve towards the peaks of the terrain. This makes the cells thinner at the peaks and wider at the troughs, causing  $\Delta x_{max}$  to increase.

The near-orthogonal grid showed some slight improvement over the SLEVE coordinate that it is based on for  $\Delta z_{max}$  and  $\Delta z_{min}$ . The range for  $\Delta z$  is smaller than for SLEVE, showing that the change of vertical lines has helped the cells to become less distorted, which could be shown by looking at the uniformity and skewness of the grid.  $\Delta x$ , however, shows that the near-orthogonal grid has significantly more range in the x-direction than SLEVE due to the vertical grid lines curving away from the troughs and towards the peaks. This effect can also be seen in the slightly larger  $\alpha_{max}$  (in Table 3.2) that the near-orthogonal has compared to SLEVE of 0.16027 compared to 0.13855.

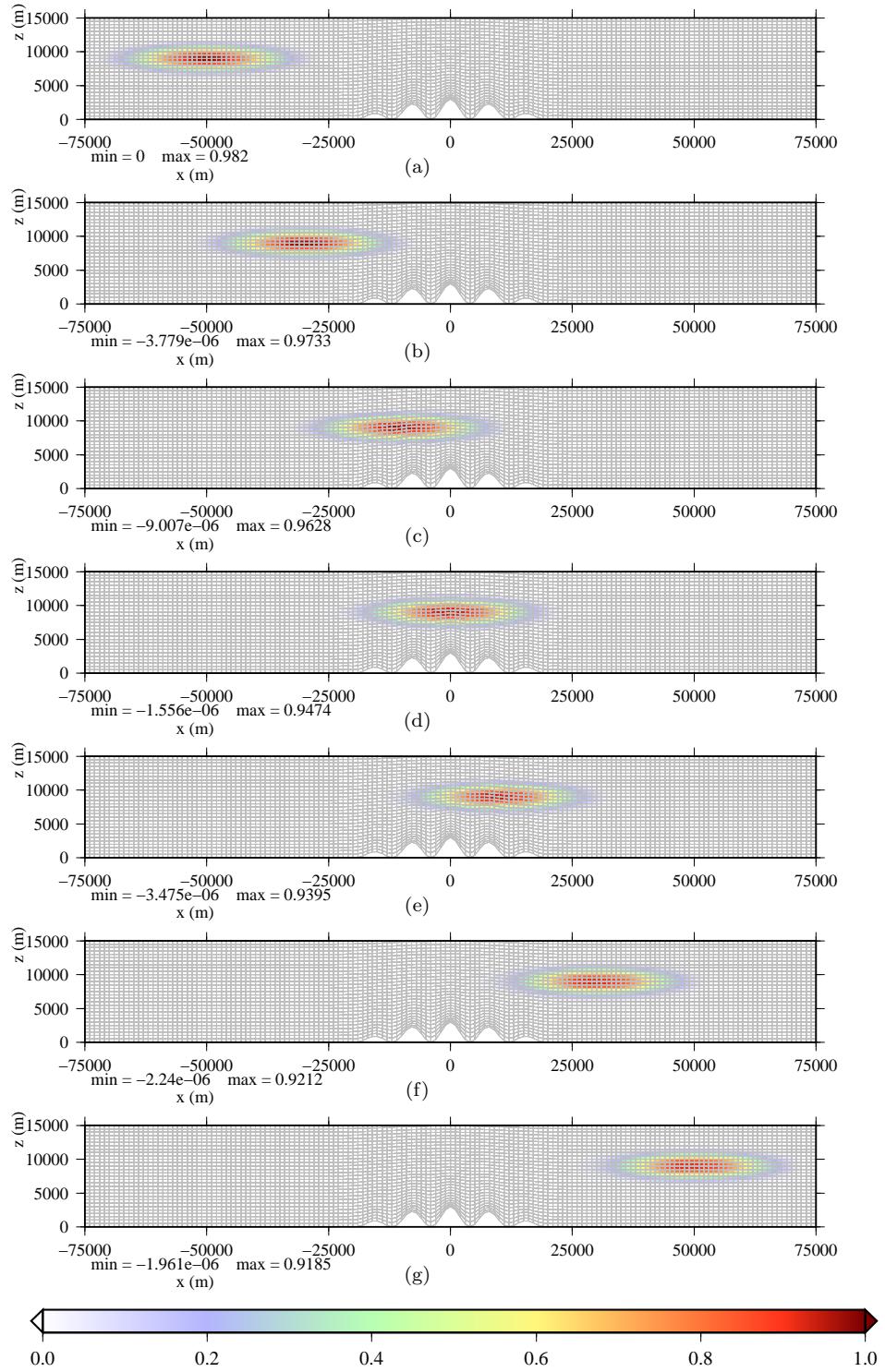


FIGURE 3.7: Example solutions from SLEVE at (a) 0s, (b) 2000s, (c) 4000s, (d) 5000s, (e) 6000s, (f) 8000s and (g) 10000s. The colours show the value of the tracer.

Coordinates	$\Delta z_{min}$ (m)	$\Delta z_{max}$ (m)	$\Delta x_{min}$ (m)	$\Delta x_{max}$ (m)
BTF	440	500	1000	1467.6
HTF	317.5	500	1000	1467.6
SLEVE	175.2	705.5	1000	1467.6
optimised SLEVE	257.7	680.5	1000	1467.6
near-orthogonal	191.3	653.9	422.4	2286.3

TABLE 3.1: Attributes of each grid created.

Coordinates	$\alpha_{mean}$	$\alpha_{max}$	$\Delta t$ (s)
BTF	0.04252	0.14696	5
HTF	0.08267	0.23744	10
SLEVE	0.08122	0.13855	10
optimised SLEVE	0.08123	0.13465	10
near-orthogonal	0.08126	0.16027	10

TABLE 3.2: Courant numbers and timestep for the model run with each grid.

### 3.3 Tracer advection over orography

The results from the simulation are shown in Figure 3.7 (using SLEVE as an example) at various times throughout the model run showing that the tracer moves about the right speed and keeps its shape well. Comparisons of the different errors with respect to the analytical solution are shown for all the grids over orography for  $t=5000$ s and  $t=10000$ s in Figures 3.8 and 3.9 respectively. The Courant numbers for the model runs, along with the timesteps used, are shown in Table 3.2. For all the model runs over orography with the different grids the Courant numbers ( $\alpha_{mean}$  and  $\alpha_{max}$ ) were kept below 0.2 apart from for HTF with  $\alpha_{max} = 0.23744$ , but this did not make any discernible difference to the solution (not shown).

#### 3.3.1 Tracer advection using a grid based on BTF coordinates

The error for BTF, shown in Figures 3.8(a) and 3.9(a) produced the largest error compared to the other non-orthogonal grids, as expected from the literature (Schär et al., 2002; Klemp, 2011), because it is the most distorted aloft.

The error at  $t=5000$ s (Figure 3.8(a)) shows a wave-like pattern caused by distortions in the model solution as it passes through the distorted grid. Artificial vertical diffusion can also be seen by the spreading out of the model solution vertically, shown by the negative error in the centre and the positive error above and below, caused by the non-alignment between the flow and the grid. The model solution has spread out quite a bit more compared to the other,

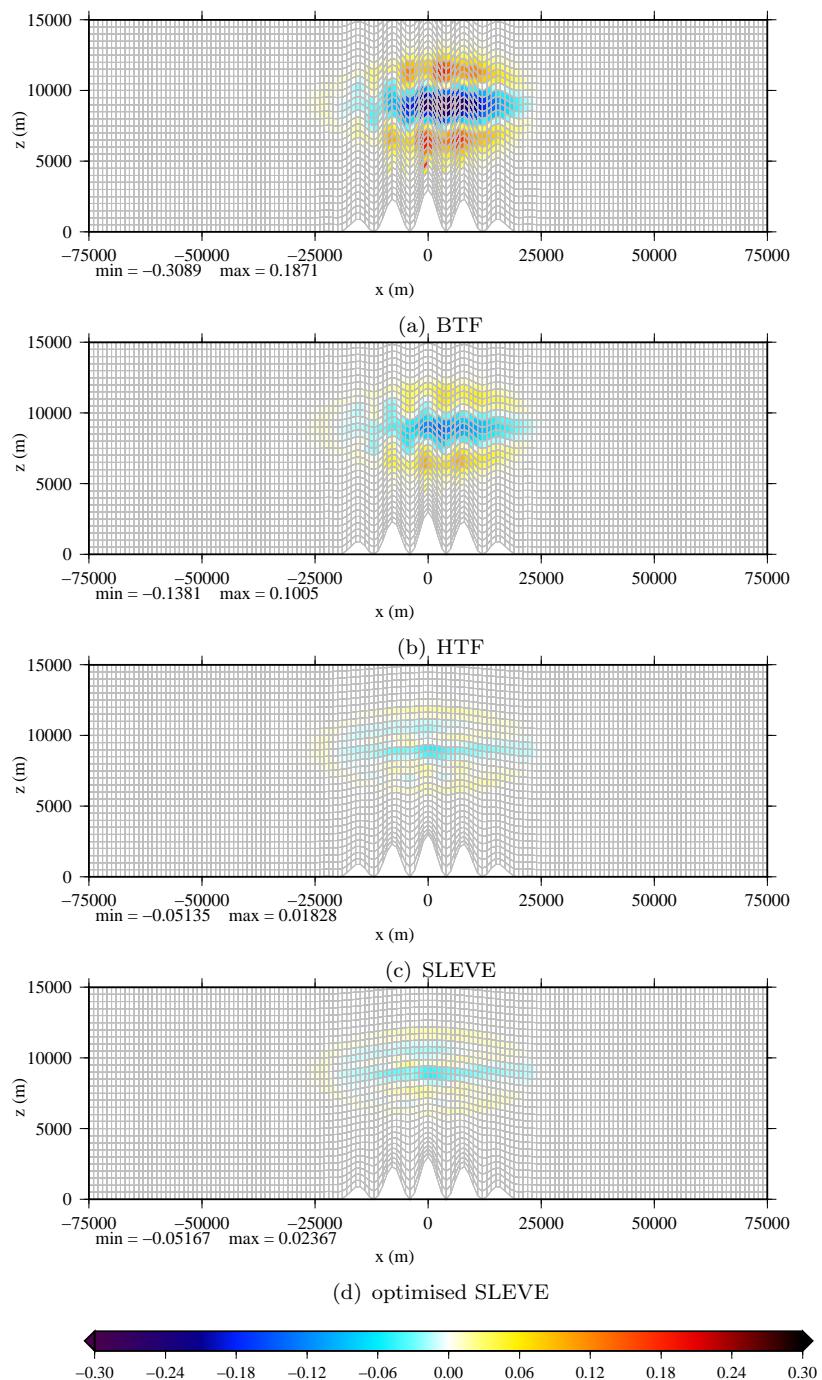


FIGURE 3.8: Error at  $t=5000$ s (numerical solution minus analytical solution) for BTF (a), HTF (b), SLEVE (c) and the optimised version of SLEVE (d). The colours show the error.

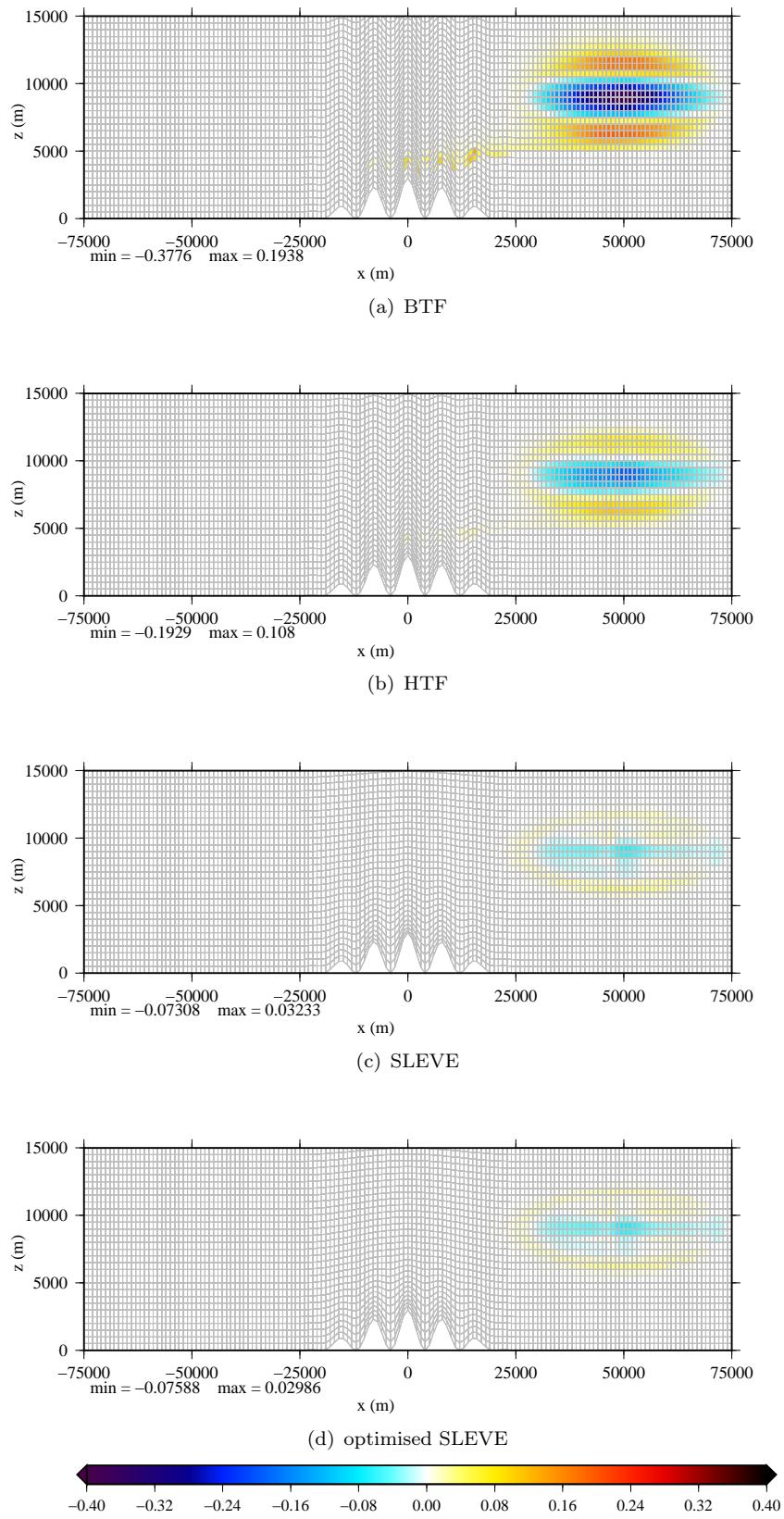


FIGURE 3.9: Error at  $t=10000$ s (numerical solution minus analytical solution) for BTF (a), HTF (b), SLEVE (c) and the optimised version of SLEVE (d). The colours show the error.

terrain-following coordinate based grids. At  $t=10000\text{s}$  (Figure 3.9(a)) the model solution is still spread out vertically compared to the analytical solution. Also note the positive error behind the anomaly at both times which suggests the tracer is advecting too slowly, which is seen in all of the coordinates and looks about the same magnitude. This is due to the numerical schemes rather than the grid because it was also seen in the cases without orography.

The large error for BTF compared to the other terrain-following coordinate based grids is caused by the grid being more distorted because the computational surfaces aloft contain more of the effect of the terrain, showing the slower linear decay of the terrain effect. The comparison between the truncation error due to orography and without orography in Schär et al. (2002) shows that the errors with the transformed coordinates are much larger than the errors when on a uniform grid. This suggests that the large distortions in the BTF grid cause a larger truncation error because the horizontal gradient of the inverse of the Jacobian is large. The size of these errors overpowers the errors due to the numerical scheme.

The general shape of the error field, at  $t=10000\text{s}$ , with a negative error in the centre of the analytical tracer position and positive error above and below, is similar to the errors seen in Lundquist et al. (2010). The shape in the error suggests that there is artificial diffusion occurring, causing a vertical spread in the anomaly. This diffusion can cause the anomaly to be more susceptible to coordinate transformations (Schär et al., 2002); this is a possible reason why the errors are quite large when compared to the initial maximum. This also explains the ‘tail-like’ error seen behind the anomaly probably caused by the spreading out of the anomaly into an area of slower wind, and so being left behind, and then the strong truncation errors that close to the peaks of the terrain could have increased the size of the error on the tail.

### 3.3.2 Tracer advection using a grid based on HTF coordinates

The results from HTF (Figures 3.8(b) and 3.9(b)) show a significant improvement over BTF with the maximum and minimum errors being approximately half that of BTF for  $t=10000\text{s}$  and still much smaller generally for  $t=5000\text{s}$ . A similar shape in the error at  $t=10000\text{s}$  can be seen compared to BTF and the error at  $t=5000\text{s}$  looks similar although does not have the same vertical extent because the grid is not as distorted. The error for HTF looks similar but smaller than BTF because the schemes being used are the same (so having the same reasons for the errors) but being lower in magnitude because the grid is less distorted aloft.

### 3.3.3 Tracer advection using a grid based on SLEVE coordinates

The error field for SLEVE for  $t=10000$ s (Figure 3.9(c)) again looks similar, showing the same general shape as BTF and HTF but with much smaller magnitudes. The error at  $t=5000$ s (Figure 3.8(c)) however looks significantly different, showing a wave-like pattern vertically with alternating regions of positive and negative error. This vertical pattern was probably cased by diffusion between the layers. The extra smoothing provided by decaying the small scale contributions faster than the large ones in the SLEVE coordinates reduces the truncation error. Again, the error field looks similar because the same schemes are being used.

The optimisation of SLEVE (Figures 3.9(d) and 3.8(d)) did not make that much difference for the test case being conducted because the tracer does not advect near the surface. This lack of difference is shown by the similarity between the solutions for SLEVE and optimised SLEVE in Figures 3.8 and 3.9. However, optimised SLEVE did show slightly larger errors at  $t=5000$ s probably due to the grid being smoothed more slowly.

When SLEVE was tested, the backwards time stepping scheme was changed to the Crank Nicolson scheme to test if this made any difference. When it was used there was no discernible difference between two schemes (not shown); this was thought to be because the Courant number was kept small so the difference between the two schemes was negligible.

SLEVE gave the best results from the implemented coordinate systems because it produced the smoothest coordinates aloft, however, STF has been shown to produce better results than SLEVE (Leuenberger et al., 2010). STF was not implemented because it was decided that SLEVE was sufficient to base a more orthogonal grid on.

### 3.3.4 Tracer advection using the new near-orthogonal grid

The error compared to the analytical solution for the near-orthogonal grid is shown in Figure 3.10. At  $t=5000$ s and  $t=10000$ s, the error is similar to the error from SLEVE due to the grids being similar aloft. The same positive error behind the tracer was seen for this grid, as it was for the other grids, because of the numerical schemes. The error is slightly smaller, however, showing that either the increased orthogonality or other attributes of the grid have reduced the error; this is probably not because of the increased orthogonality because the velocities are not being solved for.

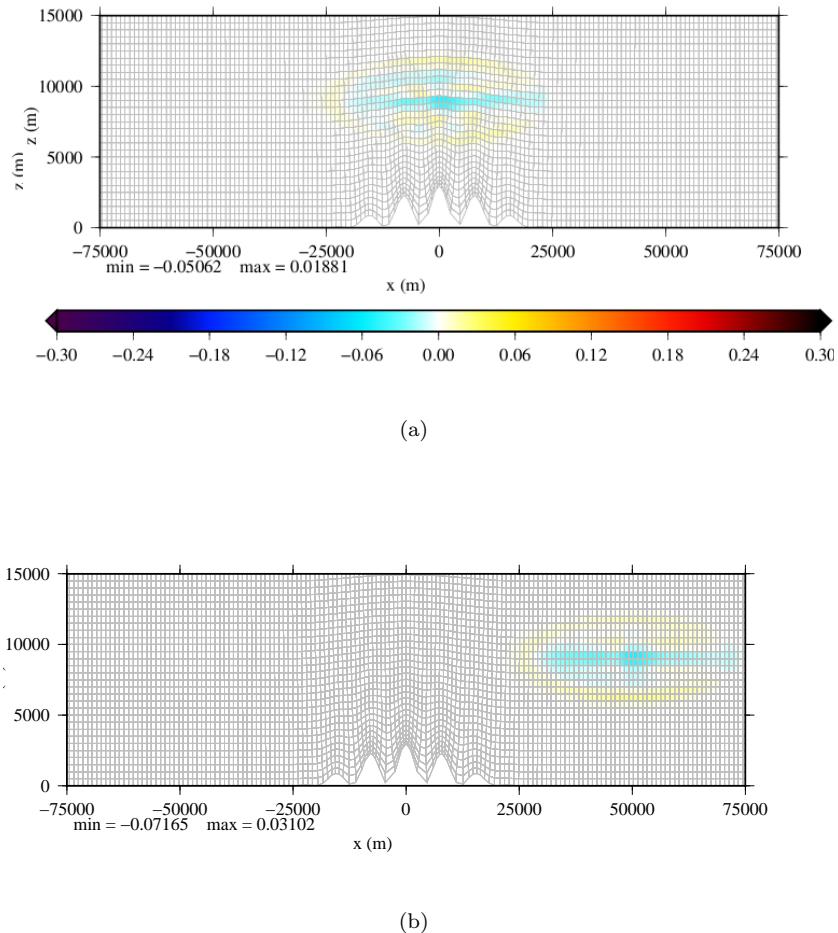


FIGURE 3.10: Error for the near orthogonal grid created using the velocity potential contours compared to the analytical solution at 5000s (a) and 10000s (b). The colours show the error.

### 3.3.5 Summary of the tracer advection over orography

In this section, the near-orthogonal grid gave the smallest error, closely followed by SLEVE. The similarity in the error between these two is thought to be because of similarity of the grids aloft. The increased orthogonality is not thought to be the reason for the reduced error. The next section will investigate other properties of the two grids to attempt to find the reason for the reduced error in the near-orthogonal grid.

### 3.4 Analysis of the grids for SLEVE and the near-orthogonal grid

A measure of the orthogonality of SLEVE and the near-orthogonal grid is shown in Figures 3.11(a) and 3.11(b). This measure of orthogonality is defined as the difference between the angle defined in Figure 1.4 and  $90^\circ$ . Figures 3.11(a) and 3.11(b) show that the near-orthogonal grid is closer to being orthogonal than the SLEVE coordinates, with the angle being closer to  $90^\circ$  throughout the grid. The near orthogonal grid is not completely orthogonal because the contours are not defined correctly near the surface, also the velocity potential contours are not defined to be orthogonal to the SLEVE coordinates (they are defined to be orthogonal with the streamlines from solving  $\nabla^2\psi = 0$ ). If the velocities were being solved for, this could have decreased the error seen for the near-orthogonal grid compared to SLEVE. Because of this it is unlikely that the reduction in error is because of the orthogonality, it is more likely to be because of the skewness and uniformity of the grid. The skewness of the grid and the cell length compared to the length of the unaltered grid (1km)(the grid with no orography) are also shown in Figure 3.11 at various levels chosen to try and represent the whole grid. The skewness is defined as the distance between the centre of the face between two cells and the intercept between a line connecting the centre of the cells and the face between the two cells.

The cell length shown in Figures 3.11(c) and 3.11(d) can give a sense of the non-uniformity in the grid by looking at the variability of the cell length along each of the levels shown. For SLEVE the cell lengths are unchanged from the original cell lengths at the edges of the domain and higher in the domain (curve flattens around the 12km level). The cells increase in length where the curvature of the grid increases, mainly low in the domain, near the surface around the slopes in the terrain. This is due to the cells stretching because the cell width in the x-direction is the same, whereas the height of the cell in the z-direction is greatly increased. For the near-orthogonal grid, the cell length is even more variable because of the curvature of the vertical coordinate lines towards the peaks in the terrain causing the cells to reduce in length at the peaks and increase in length in the troughs. Figure 3.11(d) shows the non-uniformity near the surface (2km) for the near-orthogonal grid is much greater than for SLEVE varying between 70% and 120% as opposed to 100% to around 110%. At the level of the tracer (around 7km to 12km) the variability is much less than the variability near the surface but the near-orthogonal grid is still much more variable than SLEVE. This increased non-uniformity is likely to increase the error. Figure 3.11(d) also shows that towards the edges

of the domain the cells are longer than the original length of the cells for the near-orthogonal grid; this appeared to have little effect on the error.

The skewness of the grid (Figures 3.11(e) and 3.11(f)) improved for the near orthogonal grid, especially at low levels near the peaks, but stayed similar in the troughs. The improvement at the peaks is likely to be due to cells becoming smaller rather than the grid becoming more orthogonal. This is because the centre of the cells are closer to the face between them, meaning that the intersection between the line connecting the centres of the cells and the cell face is more likely to be closer to the centre of the cell face. This slight decrease in skewness is likely to improve the results slightly, so can explain the reduced error seen for the near-orthogonal grid.

### 3.5 Error diagnostics for all the grids

The diagnostics from Williamson et al. (1992) (equations (2.25)-(2.32)) can be used to give an idea of what errors were occurring and when for the different coordinates.  $l_2$  is not shown because it gives similar information to  $l_1$ , and  $l_\infty$  is not shown because it gives similar information to the maximum error. To be able to get some insight as to what is causing these errors it is useful to see where the tracer is sat relative to the orography at some of the times not shown on the error graphs, but are shown in Figure 3.7 using SLEVE as an example.  $l_1$  (from equation (2.25)) is shown in Figure 3.12, the normalised minimum and maximum in the error (equations (2.32) and (2.31)) are also shown in Figure 3.12. The normalised variance (equation (2.30)) is shown in Figure 3.13 along with the normalised mean (equation (2.29)).

The normalised maximum gives an indication of how the peak in the tracer has changed by comparing it to the analytical solution. For all the grids, the maximum decreases over time steadily for the first around 3000s, then decreases faster when the peak in the tracer enters the area where the grid is more distorted (shown by the tracer moving into the distorted region between Figures 3.7(b) and 3.7(c) at times 2000s and 4000s respectively), showing that there is more diffusion happening during this period. Then when the peak moves out of the distorted region (at around 7000s shown on Figures 3.7(e) and 3.7(f)) the maximum decreases less over time. The differences between the different grids occur when the peak in the tracer moves through the distorted grid, with the diffusion being greater on the more distorted grids: BTF showing the most diffusion; then HTF; then SLEVE, optimised SLEVE

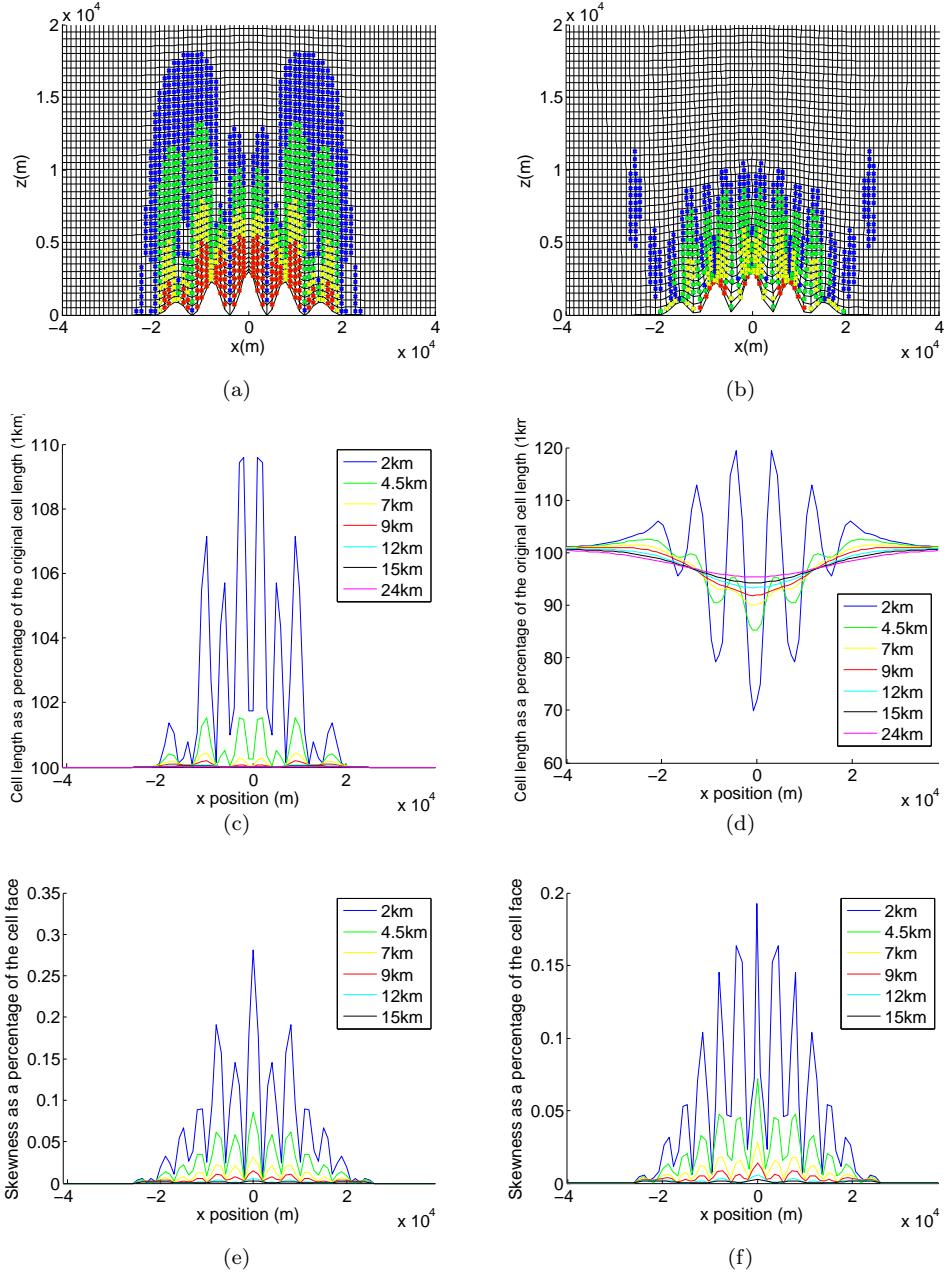


FIGURE 3.11: Analysis of the grid for SLEVE ((a), (c) and (e)) and the near orthogonal grid created using the velocity potential contours ((b), (d) and (f)). (a) and (b) shows the difference between the angle between a line connecting each cell centre and the face between them (shown in Figure 1.4), and 90°: red being an angle greater than 10°; yellow greater than 5°; green greater than 2°; and blue greater than 1°. (c) and (d) show the non-uniformity of the cells along the computational level described by the height of the level at the edge of the domain. (e) and (f) show the skewness at the level described by the height of the computational level at the edge of the domain. The non-uniformity and skewness at the surface are not shown because they were much higher in magnitude than at any other level and any structure in the other levels could not be seen.

and the near-orthogonal grid all showing about the same amount of reduction in the maximum. The diffusion is caused by the flux on the upper and lower, slanted, faces of the cell being over or under estimated because the wind is not aligned with these faces; this is worse for the more distorted grids. There is a more obvious structure seen in the grids other than BTF because of the maximum in the normalised maximum being lower; this structure shows the maximum drop more suddenly than smooth off. This is possibly due to the half widths of the finer scale structure of the orography being 3km which would coincide with the distance that the tracer has travelled in 300s (the width of the drop in Figure 3.12(d)). The steepness of the slopes could be causing larger error where they are steepest, but then the error reducing at the peaks and troughs where the coordinates are, relatively, flatter. This could indicate that the largest errors that cause diffusion are around the areas of slopes rather than errors at the peaks and troughs and so due to the non-alignment of the flow and the grid (because of the slanting cells) rather than the skewness (skewness higher at the peaks due to the opposite slanting of the cells). This is a reason why the diffusion is higher on the less smoothed grids. This effect is most prevalent in HTF but is also seen in SLEVE, optimised SLEVE and the near-orthogonal grid - the difference in shape could be to do with how much more the grid is smoothed compared to HTF, suggesting that the areas of sloping surfaces are larger and less frequent than HTF, seen by the reduced number of sharp drops in the maximum for example for SLEVE in Figure 3.12(f).

All the minimums in Figure 3.12 shows that the minimum in the error stays at approximately zero. This is expected because of the bound nature of the TVD advection scheme keeping the range of the values between 0 and 1 and so, away from the tracer; the tracer value is zero for both the model solutions and the analytical solutions, so the error in the minimum is also zero.

The normalised variance shown in Figure 3.13 and described by equation (2.30) indicates how the difference in the variance between the model and analytical solutions vary with time. Variance is a measure of the spread of the values in the tracer field so the normalised variance in the Figure 3.13 gives a measure of how spread out the values in the model solution are compared to the analytical solution and so gives another measure of diffusion. For all of the coordinates the shape of the normalised variance is very similar, only varying much in magnitude. This general shape shows the variance decreasing much less at the beginning and the end than in the centre of the curve (2000s-8000s) which is when the tracer is in the area where the coordinates are more distorted. As the tracer moves through the distorted area of

the grid the variance drops much faster, most for BTF (Figure 3.13(b)) and least for the near-orthogonal grid (Figure 3.13(j)). This drop in the variance shows that the spread of values in the model solution is getting smaller compared to the analytical solution, showing that more diffusion is occurring. This agrees with the analysis using the maximum values of the error in the tracer, but shows there is a slight difference between SLEVE and the near-orthogonal grid with the variance in the latter decreasing less. This indicates that the near-orthogonal grid performs slightly better with regards to variance in the solution agreeing with the reduced error for the near orthogonal grid between Figures 3.9(c) and 3.10(b) for SLEVE and the near-orthogonal grid respectively.

The global normalised error ( $l_1$ ) in Figure 3.12 shows that in the areas where the terrain has no effect the error increases slowly then when the tracer enters the distorted grid (around 2000s) the error increases more rapidly. The more distorted the grid, the faster the error increases - BTF is the worst and the near orthogonal coordinates are the best. Note the different scales on the y-axis making the error appear to be increasing faster initially for SLEVE and the near-orthogonal grid, whereas this only appears so because the error increases less overall because there is less distortion in the grid than for BTF and HTF.

The normalised mean in Figure 3.13 shows that the mass is being conserved only to about  $10^{-4}$  which seems large for an energy conserving model. The boundaries of the domain are set to have a zero gradient, which means that tracer can move out of the domain. When the boundaries were checked, there were small values of tracer at the boundaries implying that some of the tracer had moved out. This means that the mass is not conserved because tracer had moved out of the domain.

Possible reasons, other than increased orthogonality, as to why the near orthogonal grid was giving better results than SLEVE were given in the last two sections. The main reason thought was because of the reduced skewness in the grid. The error diagnostics support the reduced error from Section 3.3 showing a reduction in error for the near-orthogonal grid compared to the SLEVE coordinate based grid.

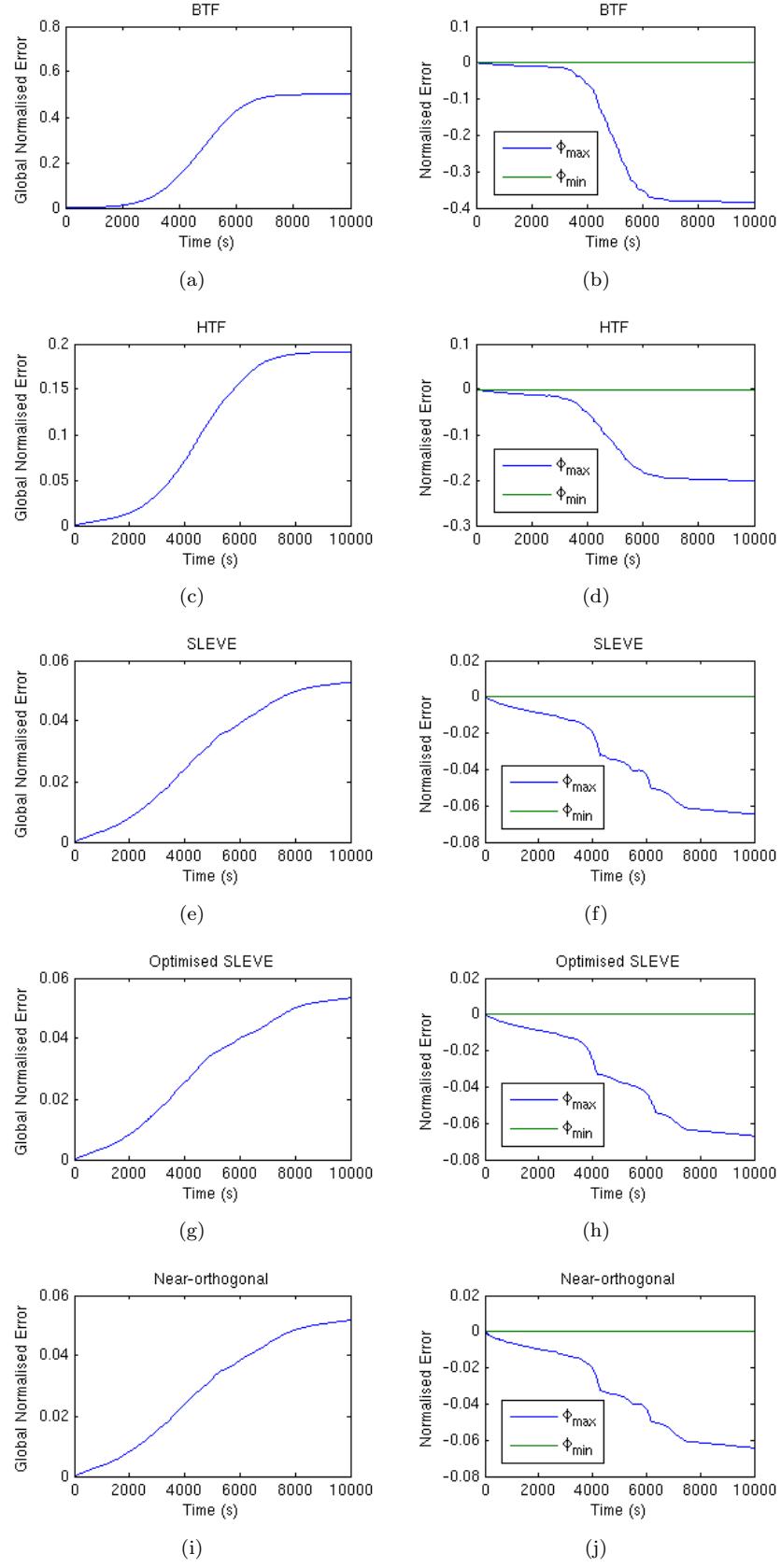


FIGURE 3.12: Graphs showing global normalised error  $l_1$  (Figures (a), (c), (e), (g) and (i)) and normalised maximum and minimum in the error with respect to the analytical solution (Figures (b), (d), (f), (h) and (j)) from Williamson et al. (1992) for BTF, (a) and (b); HTF, (c) and (d); SLEVE, (e) and (f); optimised SLEVE, (g) and (h); and the near-orthogonal grid, (i) and (j).

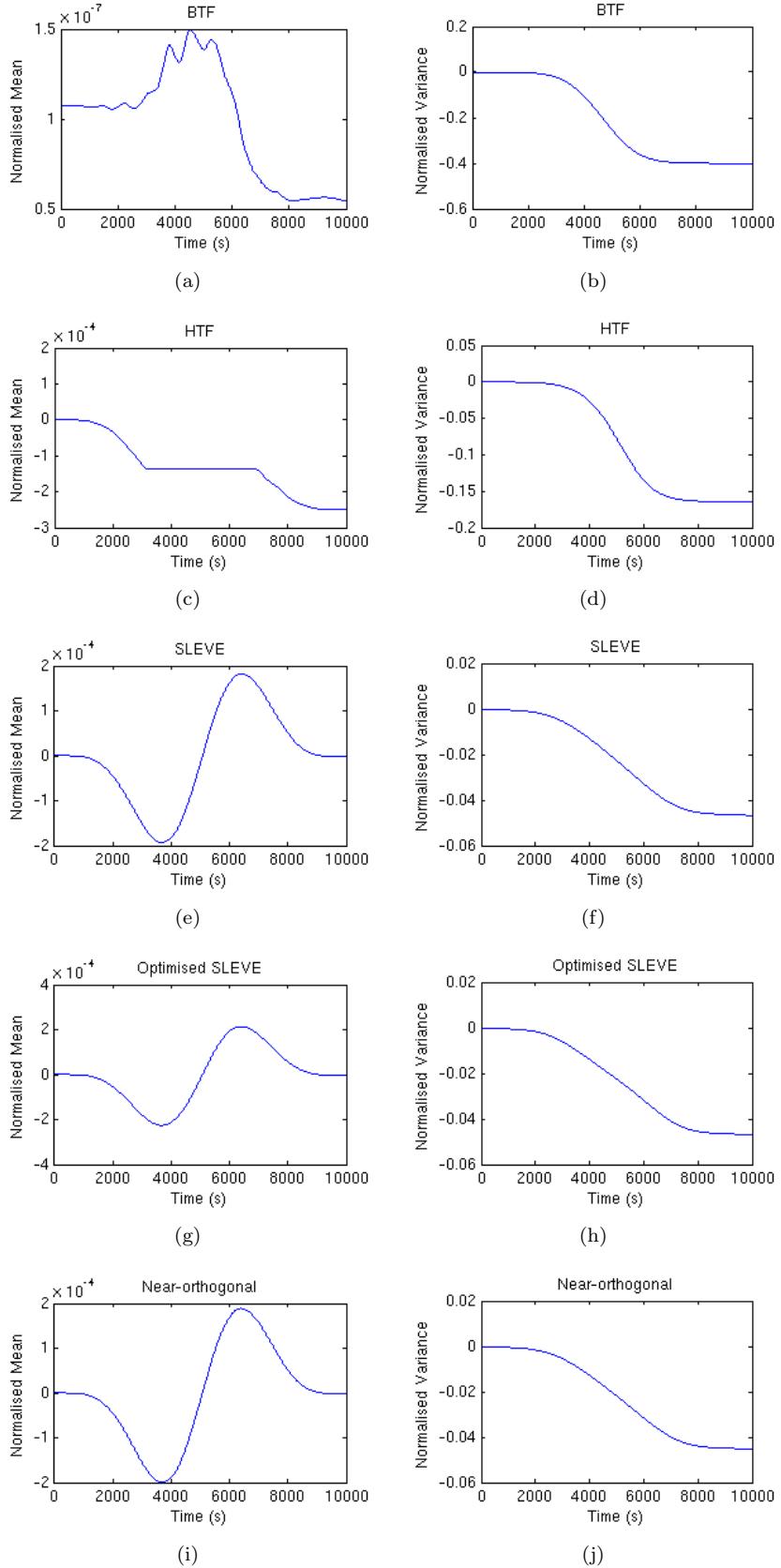


FIGURE 3.13: Graphs showing normalised mean (Figures (a), (c), (e), (g) and (i)) and normalised variance (Figures (b), (d), (f), (h) and (j)) from Williamson et al. (1992) for BTF, (a) and (b); HTF, (c) and (d); SLEVE, (e) and (f); optimised SLEVE, (g) and (h); and the near-orthogonal grid, (i) and (j).

# Chapter 4

## Summary and Conclusions

In this project a simulation was carried out of a passive tracer that was advected over orography designed to simulate complex terrain, using a test case from Schär et al. (2002) for a test case with no orography, and one with orography for grids based on terrain-following coordinates and a created, more orthogonal, grid.

### 4.1 Conclusions on the objectives

- Review different terrain following coordinates from the literature and
- Review ways of creating orthogonal coordinates from the literature.

Various different methods of including the effect of orography in a numerical model were looked at, the main method being terrain-following coordinates. Of these, STF was considered to be the best from the literature (but was not implemented), and SLEVE was the best tested because it produced the smallest error. The main problem with terrain following coordinates is their lack of orthogonality and the need with HTF, SLEVE and STF to define parameters, for example scale heights, to be useful over the whole domain. This led to looking at various ways of producing more orthogonal coordinates, which were also reviewed to see which would be suitable to make terrain-following coordinates more orthogonal - solving for the velocity potential in combination with the SLEVE coordinates was chosen because of the ease of implementation and because this combination has not been done before.

- **Run a scalar transport over flat ground to analyse where errors are coming from for different numerical schemes.**

This gave some insight as to what errors were produced because of the numerical scheme and what errors would be due to the inclusion of the terrain. This was useful when analysing the results from the rest of the project and was used as a control simulation to see how the errors would be on a regular, aligned grid. The test carried out was a comparison between different numerical schemes in a domain without orography which was done by looking at the error fields for each scheme after advection in conjunction with power spectrums from Fourier transforms and numerical phase speed in comparison to the exact phase speed. The errors using the linear scheme were found to be mainly due to dispersion errors, but some diffusion was also seen along with a computational mode. For upwind, the error was found to be due mainly to diffusion with any dispersion errors masked by how large the diffusion errors were. Using the van Leer limiter the errors were also mainly due to dispersion with a small amount of diffusion but the tail in the error seen using linear was filtered out. Van Leer was considered the best scheme, so was used when orography was included.

- **Run a scalar transport to test different grids based on terrain-following co-ordinates.**

This next stage was to introduce orography into the domain and compare grids based on terrain-following coordinates. Of these grids SLEVE performed the best but the patterns in all the grids were similar, showing that the reasons for the errors were also similar. The main error found in all these grids was vertical diffusion to do with the grids lines not being aligned with the direction of the flow. As expected, SLEVE was the best grid tested and could be used as a base to create a more orthogonal grid.

- **Create a more orthogonal grid, compare it with the terrain-following co-ordinates and assess its suitability for further testing.**

When the created near-orthogonal grid was implemented the errors produced were slightly smaller. This was possibly to do with the reduced skewness in the grid. To further investigate the errors on the different grids, the non-uniformity, orthogonality and skewness were calculated, along with time series of error diagnostics.

The near-orthogonal grid performed slightly better than the SLEVE based grid for the globalised normal error, the variance, the normalised maximum and when compared to

the analytical solution, for the test case used. These results could change if a different test was done which was closer to the ground: the near-orthogonal grid would probably perform worse because of the increased non-uniformity compared to SLEVE closer to the surface. However, being more orthogonal, the near-orthogonal coordinates would be expected to perform much better if the velocities were being solved for because of the increased orthogonality shown.

For the test case used in this project it has been shown that making coordinates more orthogonal slightly improves the error over the grids based on terrain-following coordinates and has no serious detrimental effect. However, having a test which would solve for velocities would show if the increased orthogonality is beneficial.

It would also be useful to be able to test these coordinates in three-dimensions which could be possible because velocity potential surfaces can be defined in that way. In a logically structured rectangular, three-dimensional domain a similar method could be used to find the intercepts between surfaces. This however could not be done for a spherical domain because the velocity potential cannot be defined on a sphere.



# Chapter 5

## Further Work

There are a number of areas which could be investigated to further the work done in this project:

- **Solving for velocities (solving the Euler equations).**

The first thing which would be useful to look at is a test which solves for the velocities. This would be the next step to see if making coordinates more orthogonal is beneficial to a grid. A test which could be implemented is a resting atmosphere simulation like the one described by Klemp (2011) or Lundquist et al. (2010). In this simulation the model is initiated with a horizontally homogeneous thermodynamic sounding, and when initially at rest the atmosphere should remain at rest. This type of test challenges the accuracy of the calculations of the horizontal pressure gradient (Klemp, 2011) and the results from more orthogonal grids created could be compared with the results from Klemp (2011).

Another test which could be done would be the dry flow past idealised topography from Schär et al. (2002) and for different slope inclinations, as in Lundquist et al. (2010). This also solves for velocities so would be good to test the more orthogonal grids. Any of the tests so far mentioned could also be carried out for a real terrain sample such as the one in Schär et al. (2002) to see how the grids compare over real terrain.

- **Try more methods of generating orthogonal grids.**

The next thing which could be done would be to test more methods of generating orthogonal grids to see how these compare with each other and how the attributes of the grids

differ from each other. The current method to create the near-orthogonal coordinates could be changed to calculate the horizontal coordinate lines using stream lines, by solving  $\nabla^2\psi = 0$ , as is done usually with the velocity potential in a field method. Another method which would be good to compare would be the trajectory method described in Section 1.5.2 along with a way to create the grid using conformal mapping. The method of rotating the unit vectors described by Li et al. (2012) would be good to look at to see how this works, but at the time of writing this dissertation a paper on this was not released.

Using PMA as described in Browne et al. (in preparation) would be interesting method to produce more orthogonal grids. PMA can be formulated in two-dimensions (Browne et al., in preparation) and so a two dimensional test case would be a good place to start this work. By either moving the points along horizontal levels defined by terrain-following coordinates relative to the height of the terrain on all levels then then implementing Laplacian smoothing (where all the points are averaged with their neighbours over multiple iterations, smoothing the grid and making it more orthogonal), or by moving the points relative to the terrain effect on each level. This would produce a more orthogonal grid. The method could then be moved into three-dimensions to produce three-dimensional grids to test.

- **Moving into three-dimensions.**

An extension of this work into three-dimensions would also be an interesting route to further this work. As mentioned in the conclusion, this could be possible for simple, rectangular domains. The method mentioned by Li et al. (2012), possibly the trajectory method and using PMA could all be used in three-dimensions. This extension into three-dimensions would give a more of an idea of how more orthogonal coordinates would perform in a real, three-dimensional, atmospheric model. This would allow three-dimensional tests to be performed similar to the ones in Kent et al. (2013) (these are on a sphere but could be altered for the use on a rectangular domain) to assess ability of transport schemes on the new coordinates for their suitability in dynamical cores of atmospheric numerical models.

- **Comparing the unstructured grid technique with transforming coordinates.**

Another thing which would be interesting to look at would be how the unstructured grid method used in this dissertation compares with the usual method used in atmospheric

models, of transforming the coordinates and then forming the equations of motion in the new coordinates. This would be useful to see if the way of testing grids used in this dissertation and the results seen would be comparable to results seen when transforming coordinates. There is some evidence of this being the case from results in Schär et al. (2002) and Lundquist et al. (2010) with the results for the tracer advection being similar to the results in this project, but further investigation would be useful.



# Bibliography

- Adamson, J., 1984: General method for producing a boundary-fitted orthogonal curvilinear mesh. *Appl. Math. Model.*, **Vol. 8**, 231–237.
- Adcroft, A., C. Hill, and J. Marshall, 1997: Representation of topography by shaved cells in a height coordinate ocean model. *Mon. Wea. Rev.*, 2293–2315.
- Benjamin, S. G., G. A. Grell, J. M. Brown, and T. G. Smirnova, 2004: Mesoscale weather prediction with the RUC hybrid isentropic-terrain-following coordinate model. *Mon. Wea. Rev.*, **132**, 473–494.
- Browne, P. A., C. J. Budd, C. Piccolo, and M. Cullen, in preparation: Fast three dimensional r-adaptive mesh redistribution.
- Erçal, L., 1998: Orthogonal generation systems. *Handbook of Grid Generation*, J. F. Thompson, B. K. Soni, and N. P. Weatherill, Eds., CRC Press, Chapter 7.
- Gal-Chen, T. and R. C. J. Somerville, 1975: Numerical solution of the Navier-Stokes equations with topography. *J. Comp. Phys.*, **17.3**, 276–310.
- Holton, J. R., 2004: *An Introduction to Dynamic Meteorology*. 4th ed., Elsevier Academic Press.
- Ju, L., T. Ringler, and M. Gunzburger, 2011: Voronoi tessellations and their application to climate and global modelling. *Numerical Techniques for Global Atmospheric Models*, P. H. Lauritzen, C. Jablonowski, M. A. Taylor, and R. D. Nair, Eds., Springer, 313–342.
- Kent, J., P. A. Ullrich, and C. Jablonowski, 2013: Dynamical core model intercomparison project: tracer transport test cases. *Q. J. R. Meteor. Soc.*, doi: [10.1002/qj.2208](https://doi.org/10.1002/qj.2208).
- Klemp, J. B., 2011: A terrain-following coordinate with smoothed coordinate surfaces. *Mon. Wea. Rev.*, **July 2011**, 2163–2169.

- Klemp, J. B., W. C. Skamarock, and O. Fuhrer, 2003: Numerical consistency of metric terms in terrain-following coordinates. *Mon. Wea. Rev.*, **July 2003**, 1229–1239.
- Konor, C. S. and A. Arakawa, 1997: Design of an atmospheric model based on a generalized vertical coordinate. *Mon. Wea. Rev.*, **125**, 1649–1673.
- Langer, S. A., A. C. E. Reid, V. R. Coffman, G. Doğan, S. I. Haan, R. E. Garcia, and R. C. Lua, 2013: OOF2 the manual: Crank-nicolson. URL <http://www.ctcms.nist.gov/~langer/oof2man/RegisteredClass-CrankNicolson.html>, [Online; accessed 22-July-2013].
- Leuenberger, D., M. Koller, O. Fuhrer, and C. Schär, 2010: A generalisation of the SLEVE vertical coordinate. *Mon. Wea. Rev.*, **September 2010**, 3683–3689.
- Li, Y., B. Wang, and D. Wang, 2012: An orthogonal curvilinear terrain-following coordinate for atmospheric models. *Solution to partial differential equations on the sphere.*, Isaac Newton Institute Seminar Series., 2012.09.26, URL <http://www.newton.ac.uk/programmes/AMM/seminars/2012092609001.pdf>, [Online; accessed 10-July-2013].
- Lundquist, K. A., F. K. Chow, and J. K. Lundquist, 2010: Numerical errors in the presence of steep topography: analysis and alternatives. *AMS 14th Mountain Meteorological Conference*, Lake Tahoe, CA, United States, paper 10.1.
- McNally, W. D., 1972: Fortran program for generating a two-dimensional orthogonal mesh between two arbitrary boundaries. Tech. rep., National Aeronautics and Space Administration. URL [www.pdas.com/refs/tnd6766.pdf](http://www.pdas.com/refs/tnd6766.pdf), [Online; accessed 20-July-2013].
- OpenFOAM, 2011: CourantNo.H. OpenFOAM Software.
- OpenFOAM Programmer's Guide, 2013: *OpenFOAM, the Open Source CFD Toolbox, Programmer's Guide, version 2.2.0*. OpenFOAM.
- OpenFOAM User's Guide, 2013: *OpenFOAM, the Open Source CFD Toolbox, User's Guide, version 2.2.0*. OpenFOAM.
- Press, W. H., B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, 1992: *Numerical Recipes in FORTRAN 77: Volume 1, Volume 1 of Fortran Numerical Recipes: The Art of Scientific Computing (Vol. 1)*. Cambridge University Press.
- Putman, W. M. and S. J. Lin, 2007: Finite-volume transport on various cubed-sphere grids. *J. Comp. Phys.*, **227**, 55–78.

- Ringler, T. D., 2011: Momentum, vorticity and transport: considerations of the design of a finite-volume dynamical core. *Numerical Techniques of Global Atmospheric Models*, P. H. Lauritzen, C. Jabolowski, M. A. Taylor, and R. D. Nair, Eds., Springer, 143–184.
- Sankaranarayanan, S. and M. L. Spaulding, 2003: A study of the effects of grid non-orthogonality on the solution of shallow water equations in boundary-fitted coordinate systems. *J. Comp. Phys.*, **184**, 299–320.
- 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.*, **October 2002**, 2459–2480.
- Steppeler, J., H. W. Bitzer, M. Minotte, and L. Bonaventura, 2002: Nonhydrostatic atmospheric modelling using a z-coordinate representation. *Mon. Wea. Rev.*, **130**, 2143–2149.
- Steppeler, J., R. Hess, U. Schättler, and L. Bonaventura, 2003: Review of numerical methods for nonhydrostatic weather prediction models. *Meteor. Atmos. Phys.*, **88**, 287–301.
- Thompson, J. F., Z. U. A. Warsi, and C. Wayne Mastin, 1997: *Numerical Grid Generation: Foundations and Applications*. URL <http://www.erc.msstate.edu/publications/gridbook/>, [Online; accessed 1-July-2013].
- Van Laerhoven, K., 2013: Voronoi diagrams & delaunay triangulation. URL <http://mathworld.wolfram.com/Jacobian.html>, [Online; accessed 06-August-2013].
- Weisstein, E. W., 2013a: “*Jacobian*” From MathWorld—A Wolfram Web Resource. URL <http://mathworld.wolfram.com/Jacobian.html>, [Online; accessed 06-August-2013].
- Weisstein, E. W., 2013b: “*Line-Line Intersection*.” From MathWorld—A Wolfram Web Resource. URL <http://mathworld.wolfram.com/Line-LineIntersection.html>, [Online; accessed 11-June-2013].
- Weller, H., 2012: MTMW12: Introduction to numerical methods. University of Reading, University Lecture Course, University of Reading.
- Williamson, D. L., J. B. Drake, J. J. Hack, R. Jakob, and P. N. Swarztrauber, 1992: A standard test set for numerical approximations to the shallow-water equations in spherical geometry. *J. Comp. Phys.*, **102.1**, 211–224.
- Zängl, G., 2012: Extending the numerical stability limit of terrain-following coordinate models over steep slopes. *Mon. Wea. Rev.*, **140**, 3722–3733.