THE UNIVERSITY OF READING

DEPARTMENTS OF

MATHEMATICS AND METEOROLOGY

Automatic generation of accurate advection schemes on unstructured grids and their application to meteorological problems.

by Raymond K.Lashley

Dissertation submitted for the degree of

Doctor of Philosophy

June 2002

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

Raymond Lashley

June 24, 2002

Abstract

A class of two-dimensional, conservative, shape preserving advection schemes is developed for use in meteorological models on any grid. The advection scheme is formulated in finite volume form to ensure conservation. A local high order polynomial reconstruction of the advected field is integrated over the region swept across each face to generate the fluxes across cell edges. A flux limiter is applied to these fluxes to ensure shape preservation of the advected profile. This procedure is generalised and automated to enable it to be used on any grid on the plane and on the sphere. The method is also included in a shallow water model on the sphere.

The automatic generation of the schemes also enables the order of the polynomial used to be easily changed to vary both the accuracy and the computational cost of the scheme. The effect on the accuracy and computational cost of varying the order of the polynomial and the grid resolution has been investigated on square and triangular grids on the plane, and the icosahedral-hexagonal grid on the sphere. It is found that second to fourth order polynomial are the most efficient when comparing a schemes accuracy with its computational cost.

The icosahedral-hexagonal grid gives a near uniform coverage of the sphere and has no singularities, which makes it particularly suitable for use in meteorological models. The schemes developed in this work have been incorporated into a global shallow water model on an icosahedral-hexagonal grid. A standard set of test cases for shallow water models were used with the model to investigate the effect of using more accurate advection schemes in the model. The use of higher order polynomials in

the advection scheme produces significant improvements in the accuracy of the model without the increase in computational cost associated with increasing the grid resolution.

> To explain all nature is too difficult a task for any one man or even for any one age. 'Tis much better to do a little with certainty, and leave the rest for others that come after you, than to explain all things.
>
> *Sir Isaac Newton*

Acknowledgments

There are so many people that deserve thanks for helping me complete this work that there isn't enough room to mention you all by name. My thanks to you all nonetheless.

Thanks to my whole family; Mum and Dad, without whom I'd never have made it through, or even got to University. Big Bro, for, amongst other things, staying one step ahead in the qualification stakes, until now. Grandma's and Grandad's, a long way away by train but always close to me nevertheless. Aunties, Uncles, Cousins and all the rest.

Huge thanks go to my supervisors, Mike Baines and John Thuburn, their support, suggestions and nagging were invaluable to me. My appreciation also goes to others who have offered advice and guidance on my work; Peter Sweby, Paul Valdes and Nancy Nichols amongst others. To all those who have taught me over the years, thank you for building the foundations. Special thanks to Jeremy Stelling, Paul Evans and Mike Cullingham who went above and beyond the call of duty in their teachings.

To my friends and University colleagues, your help, support, humor and friendship mean more to me than I let on; Stuart, Jo, Paul, Darek, Phil, Dom, ... I think the correct phrase is et. al. My sanity is all your

faults.

To everyone else, Scout leaders, co-workers, RUSU colleagues, fellow X-ers and so many more, thank you. However brief our passings the result was to put me on or help me along this path. Special thanks and apology to John Henderson, who told me the truth I didn't want to hear. It caused us both difficulties but I appreciate it more than anyone will know.

Final thanks to an unknown author of a pre-school maths book, a green one if I remember correctly, for making me want to get a PhD in Mathematics. This was all your fault, but at least I've got it out of my system now.

Contents

Chapter 1

Introduction

The transport of the properties of an air parcel by the wind is one of the most fundamental concepts in meteorology. The transport of the properties of a fluid parcel by a flow is also of great importance in many other physical situations. The equation that governs this transport is the advection equation and can be written as

$$\frac{\partial\left(\rho\phi\right)}{\partial t}+\nabla\cdot\left(\mathbf{v}\rho\phi\right)=0\ , \tag{1.1}$$

for a fluid property with a mass mixing ratio ϕ in a fluid of density ρ with velocity $\mathbf{v}$. The fluid property ϕ could be a chemical tracer, such as ozone, or a physical property of the fluid, such as entropy.

If the velocity $\mathrm{v}(\mathbf{x},t)$ and the initial distribution of $\phi(\mathbf{x},0)$ are known then the distribution of ϕ at any later time can be calculated. If neither the field over the whole domain or how the velocity of the flow varies in time are known, or if suitable boundary conditions are not supplied then equation (1.1) cannot be solved in isolation. However, when the process of advection is combined with the other processes, such as heat effects, in the atmosphere or oceans the system can be closed. Nevertheless, these systems can rarely be solved without making some approximations, either by simplifying the equations used or making some numerical approximations.

These approximations can be made either to the equations themselves or to the way the fields are represented in a model. Approximations to the equations may involve linearising and removing 'small' terms or assuming balance conditions that are only approximate in the real problem. There are also many ways that the problem can be discretised in both space and time. These approximations can be combined in many different ways for many different purposes, including studying individual atmospheric processes, forecast models and climate simulations.

Richardson [36] proposed 'solving' the equations describing the atmosphere by discretising them in space and stepping forward in time by using numerical approximations to the derivatives in the equations. Richardson's first forecast was not a success by todays standards for a number of reasons. Both the model equations and the methods used to solve them were rather basic approximations and the initial data was poorly incorporated, as suggested by Lynch [26]. Despite these initial difficulties, the idea has developed and is now used for forecasting as well as running idealised models of individual atmospheric processes and climate simulations.

Since the advent of computers, numerical methods have been refined as more and more calculations can be performed in shorter and shorter times (see Wiin-Nielsen [57]). These refinements have covered all aspects of the numerical approximations of the governing equations, including advection. The type of schemes that have been used for advection vary from the simple first order Euler scheme to high order linear and nonlinear schemes. Cullen [5] gives an overview of finite difference methods in a meteorological context whilst Rood [38] gives a review of many finite difference schemes for advection. Many of these schemes are one dimensional, linear schemes which have various different orders of accuracy. Higher order one dimensional schemes have since been developed, such as the 'Transient Interpolation Modeling' (TIM) of Leonard [22].

There are several other ways to tackle the problem of numerically solving the advection equation. Overviews of several classes of method are contained in the proceedings of an ECMWF seminar series [6]. Machenhauer [27] explains the development and application of the now widely used spectral method. The semi-Lagrangian method is also widely used at present and is discussed by McDonald [31] and Staniforth and Côté [43]. The finite element method is used in many applications in computational fluid dynamics but not widely in meteorology, although one notable exception to this is the Canadian regional finite element forecast model [45]. Strang and Fix [46] give a description of the finite element method and Temperton [50] puts this in a meteorological context. Finite volume schemes have also been developed such as the Uniformly Third Order Polynomial Interpolation Algorithm (UTOPIA) of Leonard et al. [23] (see also, Rasch [35]) and that used by Thuburn [51].

All these methods are designed to give, amongst other things, a good numerical accuracy as measured by one or more error definitions. As well as this simple numerical accuracy it may also be desirable to impose some of the known features of the advection equation on numerical solutions. Equation (1.1) ensures that the total amount of whatever is being advected must be conserved, given suitable boundary conditions. When the mass of the fluid is conserved, i.e. when

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \ , \tag{1.2}$$

equation (1.1) can also be written in advective form,

$$\frac{\mathrm{D}\phi}{\mathrm{D}t} = 0 \ , \tag{1.3}$$

where $\frac{\mathrm{D}}{\mathrm{D}t}$ is the Lagrangian derivative;

$$\frac{\mathrm{D}}{\mathrm{D}t} \equiv \frac{\partial}{\partial t} + \mathbf{v} \cdot \nabla \ . \tag{1.4}$$

This shows that the mixing ratio of a fluid parcel does not change as it is being advected. There can be no increase in the global maximum mixing ratio, or decrease in the minimum, and there would be no generation of new local maxima or minima.

Either or both of these properties, known as 'conservation' and 'shape preservation' (or monotonicity), can be imposed on numerical solutions by using suitable numerical schemes. Ensuring conservation means that our advection scheme will not create or destroy amounts of whatever is being advected. This is particularly important when the advected fields are interacting with one another and the balance between them is important e.g. the transport of reactive chemicals. Ensuring shape preservation means that oscillations will not be generated near sharp gradients in the field and that no new unphysical maxima or minima will be introduced. This ensures that fields of advected quantities remain positive and do not display unphysical oscillations.

Conservation can be easily ensured by using a finite volume type scheme (e.g. the UTOPIA scheme [23]) or by a more careful formulation of other finite difference or finite element schemes. Enforcing the shape preservation property can lead to more complicated numerical schemes. Godunov's theorem (see LeVeque [25]) showed that any linear, shape preserving advection scheme cannot have an order of accuracy greater than one. This means that any high order, shape preserving scheme must be non-linear and there are a number of methods of generating such schemes. Van Leer incorporated these features into his search for the ultimate conservative difference scheme [17, 18, 19, 20, 21]. The Universal Limiter [22, 51] uses local information about the advected field to set bounds on fluxes that ensure shape preservation. Suresh and Huynh [47] present a fifth order scheme with a limiter that can be applied with Runge-Kutta time stepping in one dimension.

Flux Corrected Transport (FCT) schemes [2, 3, 4, 61] take a similar approach using a first order, shape preserving scheme, and applying carefully constrained 'antidiffusive' fluxes. Total Variation Diminishing (TVD) schemes [48] enforce a property similar to shape preservation which may allow extrema to grow if a nearby one diminishes. Essentially Non-Oscillatory (ENO) schemes [8, 40] do not enforce the shape preservation but do ensure that the magnitude of spurious oscillations decays with increasing resolution.

Making improvements to a numerical solution of the advection equation can be done in two ways. Increasing the computer power devoted to the problem will enable more complex and more accurate schemes or higher resolution to be used. Improving the numerical methods that are used can also improve the numerical solutions obtained. Improving the efficiency of a scheme will allow it to be used on a higher resolution grid. There is also the option to try and improve the accuracy of the schemes themselves, though these improvements will usually increase the computational cost of the schemes. A final possibility is to develop a new scheme that is more accurate than those currently used.

It is important to consider the use of the term 'accuracy' in the context of numerical methods. 'Accuracy' is often used in reference to some measure of the error in a numerical solution, a more accurate scheme is one that gives a better (usually lower) value of this error. Different error measures can behave differently for the same numerical schemes, a simple example of this can be seen in the L-norms of the Laplacian operator in Majewski et al. [28] (Fig. 8). The L_1 and L_2 errors display a second order accuracy as the grid is refined but the L_∞ error is only first order. In this and many other cases, the 'order of accuracy' is used to describe the rate at which the accuracy of the scheme improves as the grid is refined.

It is a well known but often overlooked fact that a 'high order' scheme may have a lower accuracy than a 'low order' scheme at some resolution. Accuracy can also be measured in a qualitative sense whereby a particular property of the solution may be important as well as or instead of an error measure. For example, it may be important that the numerical solution of the advection equation does not contain any spurious oscillations, even if this means that an error measure in increased. In this work there are occasions when we can relate the order of the polynomial used by a scheme to the formal order of accuracy of the scheme and times when this is not possible. For this reason, reference to the 'order' of a scheme refers to the order of the polynomial used, the formal order of accuracy will be referred to explicitly when required.

The use of high order accurate advection schemes can have a significant improvement in meteorological models. An example of this is given by Gregory and West [7] who investigated the use of different schemes to model the 'tape-recorder signal' of water vapour in the stratosphere. This work showed that the signal propagated vertically much too fast when a second order scheme was used. When higher order accurate schemes were used there were strong unphysical oscillations in solutions, limiters were required to remove these oscillations.

An important part of many numerical schemes is the grid on which the problem is discretised. Schemes that have a time step dependent on a Courant-Freidrichs-Levey (CFL) condition will become inefficient if there is a wide range in the distance between grid points or the sizes of cells. The CFL condition constrains the size of the time step, relative to the grid spacing, that can be used to maintain the stability of the scheme. If there is a mix of large and small cells then the small cells will restrict the size of the time step, making it more expensive to use without greatly increasing the accuracy. For a two-dimensional problem the grid will be two dimensional which can cause problems if a one dimensional scheme is used, usually by applying it once in each coordinate direction. This is usually quicker to compute than using a multidimensional scheme but often causes anisotropic distortion of the advected profile when the flow is at an angle to the grid (for example see Thuburn [51]). Many schemes are designed to be used on rectilinear grids and are not so well suited to unstructured grids.

Gridding the sphere has long been a problem with a latitude-longitude approach having singularities at the poles. A number of fixes have been used to overcome this problem within models but there are spherical grids that do not have these singularities. The icosahedral-hexagonal grid has been investigated for use in meteorological problems by Williamson [58], Sadourny et al. [39], Masuda and Ohnishi [30], Heikes and Randall [10, 11] and Thuburn [52, 53] and is now being used by Deutscher Wetterdienst in on operational global forecast model [28, 29]. This grid gives a nearly uniform coverage of the sphere and does not suffer from the ‘pole problem’. The drawback of this grid is that it contains very little regularity so many existing numerical schemes cannot be used. The icosahedral-hexagonal grid is described in detail in chapter 4.

The aim of this work is to develop an accurate, conservative and shape preserving advection scheme that is both fully multidimensional and can be used on unstructured grids. The properties of the scheme are desirable for some of the reasons described above and because we hope to improve the results of the Shallow Water model on the icosahedral-hexagonal grid developed by Thuburn [52, 53]. This model used a multidimensional advection scheme that was conservative and shape preserving but was based on a method for use on a regular hexagonal grid. This scheme caused some problems when used with the icosahedral-hexagonal grid that we might expect a tailor made scheme not to reproduce.

Attempting to develop an advection scheme that produces a numerical solution with all the desired properties (accuracy, conservation, shape preservation) will require some sacrifices to be made. The approach taken here is to try and include these sacrifices within the scheme by developing a method that will generate a family of schemes. Conservation can be assured by discretising the problem into finite volumes and approximating the fluxes between the control volumes (cells). The method is designed to allow greater accuracy to be gained at the expense of computational cost. In common with many advection schemes we shall find a scheme for which the tracer value at the next time step, ϕ^{n+1} can be written in terms of a set of coefficients, c, and the tracer values at the current time step, ϕ^n. This means we can write down an update equation in the form

$$\phi_i^{n+1} = \sum_j c_j \phi_j^n \ , \tag{1.5}$$

where the subscripts on ϕ identify a particular cell of the grid. It is the coefficients, c_j, that we seek to make up the advection scheme. If the method is to be applied to unstructured grids then it will be necessary for the method of finding these c's to be automated. This is because the different sizes and shapes of the cells prevent us from finding a general set of c's for all cases. If the schemes are used to find the fluxes of the advected quantity between the cells then shape preservation can be guaranteed by use of the Universal Limiter as described in Thuburn [51].

The method for generating these schemes is based on a polynomial interpolation of the advected field which is used to approximate the fluxes. This is the same idea as used by Leonard in his ULTIMATE scheme [22] in one dimension and UTOPIA scheme [23] in two dimensions. Chapter 2 begins by showing how automating the basic method used by the ULTIMATE schemes enables the same idea to be applied to non-uniform grids. Extending the ULTIMATE methodology to two dimensions enabled the development of the UTOPIA scheme. Using the idea of automatic generation allows this to be extended to use any order polynomial when generating the advection scheme, not just the second order polynomial[1] used in the UTOPIA scheme. The details of this extension on rectangular grids and the results of idealised test cases are also included in Chapter 2.

The method described in Chapter 2 cannot be applied directly to non-rectangular grids since it relies on properties of the connectivity of the grid. This means that the method must be generalised further to enable its application to other grids. A new way of fitting the polynomial is needed in order to enable the method to be applied to non-rectangular

[1] The UTOPIA scheme was derived using a third order polynomial for interpolation to generate an update equation. It was suggested by Leonard et al. [23] that the scheme could be written in terms of fluxes across cell boundaries that were analogous to the one dimensional quadratic interpolation schemes with extra two dimensional terms. It will be shown later in this work that the UTOPIA scheme is very similar to a scheme generated directly using a two dimensional second order polynomial interpolation

grids. Chapter 3 covers some of the ways this may be done in one dimension and goes on to apply the most suitable way to triangular grids in two dimensions. The results from idealised test cases similar to those in Chapter 2 are also presented.

Chapter 4 applies the method developed in Chapter 3 to a hexagonal-icosahedral grid on the sphere. This does not require any further changes to be made to the method itself but some extra approximations are made because of the spherical geometry. A description of these approximations and their effects in one dimension are given in Chapter 4 along with results of test cases from the application of this method to the spherical icosahedral grid.

In order to test if these schemes have any effect in more complex systems of equations they are used in a shallow water model. The results of the test cases proposed by Williamson et. al. are presented in Chapter 5. Chapter 6 discusses the conclusions that can be drawn from the previous chapters and discusses some of the points raised. This chapter also contains suggestions for further work to be done with this method in order to improve it further.

Chapter 2

Rectangular Grids

2.1 Introduction

In the last chapter several properties that are desirable in a numerical scheme were described, these included, conservation, shape preservation, multidimensionality and good accuracy. Advection schemes with these properties already exist, the ULTIMATE scheme [22] in one dimension and the UTOPIA scheme and Universal Limiter [23], but we will need to extend these methods for our purposes. The ULTIMATE family of schemes uses polynomial interpolation in one dimension, allowing greater accuracy to be achieved by increasing the order of the polynomial at the expense of increasing the computational cost. This ability to vary the accuracy and cost of the scheme is a useful feature that we would like to retain but the current method can only be applied to regular grids. We shall begin in Section 2.2 by automating the process of generating the advection scheme so we can apply the same method of the ULTIMATE schemes to irregular grids in one dimension.

The UTOPIA scheme uses a second order multidimensional polynomial in the same way as the ULTIMATE scheme does in one dimension. We can expand on this idea to use higher order multidimensional polynomials to increase the accuracy and the computational cost of the schemes. The methodology for doing this is covered in Section 2.3.1 and the results of using the schemes are presented in Section 2.4. Once we have achieved an automation of the process of generating advection schemes and extended the schemes to two dimensions we can go on to look at extending the methods to irregular grids.

2.2 One-Dimensional Advection

The advection equation, in one dimension, of a tracer with mixing ratio, ϕ, by a non-uniform velocity $u(t, x)$ can be written,

$$\frac{\partial \phi}{\partial t} = \frac{\partial \left(-u\phi\right)}{\partial x} \ . \tag{2.1}$$

The fluid density, ρ, is taken to be constant in this case so ϕ can be thought of as representing the average tracer amount in the cell. By integrating this equation over a time Δt and a control volume i in space we get the exact conservative form

$$\phi_i^{n+1} = \phi_i^n + \left(\hat{\phi}_{\mathcal{L}} - \hat{\phi}_{\mathcal{R}}\right) , \tag{2.2}$$

where ϕ_i^n is the average mixing ratio in cell i at time $n\Delta t$ and $\hat{\phi}_{\mathcal{L}/\mathcal{R}}$ is the flux across the left/right edge of cell i in time Δt. The flux may be thought of as the average mixing ratio ϕ at the edge over time Δt, multiplied by the Courant number $\frac{u\Delta t}{\Delta x}$, the ratio between the mass of fluid swept across the edge and the total mass in the cell. It can also be thought of as the amount of the tracer swept across the edge during the time step, shown by the bold regions in Figure 2.1. It is these fluxes that we wish to approximate in order to produce an update equation for our numerical method, based on equation (2.2) .

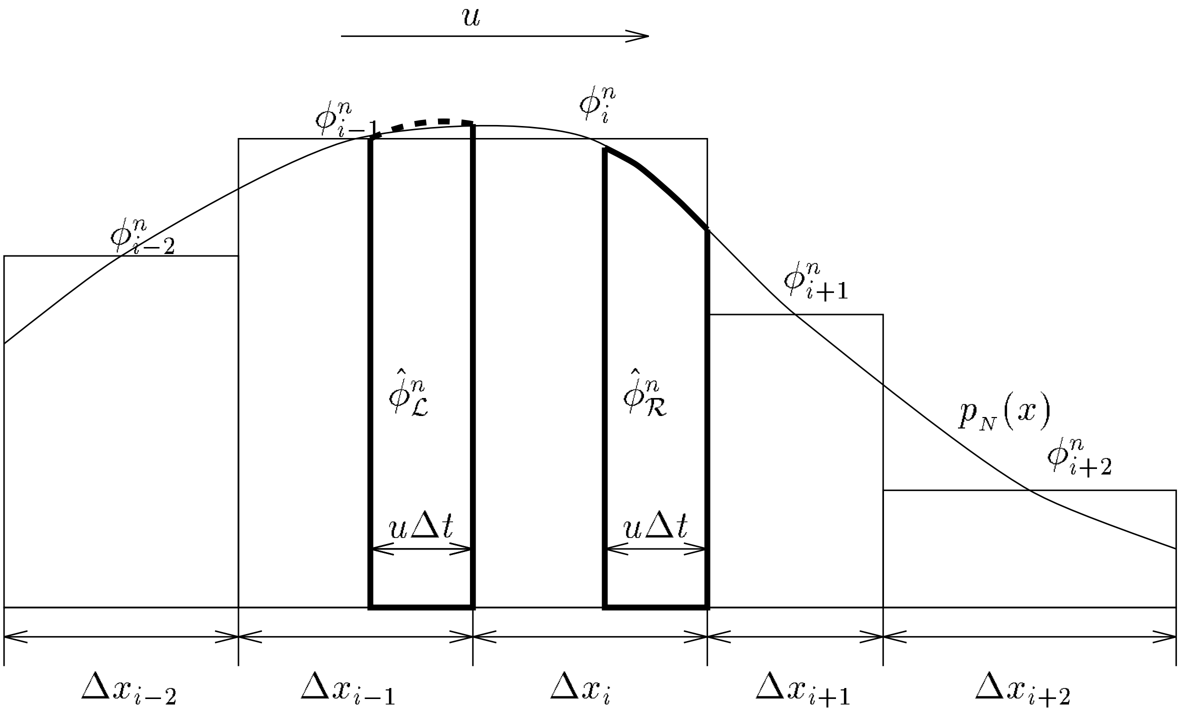


Figure 2.1: Diagram showing the regions used to calculate the fluxes, $\hat{\phi}^n_{\mathcal{L}/\mathcal{R}}$, using the polynomial, p_N. The polynomial P_N is interpolated over $N+1$ cells. The integral of this polynomial over the region swept across an edge in one time step is the approximation to the flux. A different polynomial will be found for each edge.

In his paper [22] Leonard derives update equations for advection of a tracer by a variable advecting velocity using polynomial approximations to the continuous tracer field. An N^{th} order polynomial is interpolated through $N+1$ cell centres spaced either side of the edge to approximate the tracer field. This polynomial can then be used instead of the unknown ϕ-distribution to approximate the fluxes in a similar way to that shown in Figure 2.1. This was done pointwise on a grid with a uniform spacing, Δx, but the method can be extended to a grid with unequal Δx's. For such a grid we would not wish to write down, and then code up, the update equation for each cell in turn. Instead we shall develop the method in such a way that it can easily be coded up to calculate the coefficients in the update equation automatically. This will give us an update equation of the form

$$\phi_i^{n+1} = \phi_i^n + \left(\mathbf{c}_{\mathcal{L}} \cdot \boldsymbol{\phi}_{\mathcal{L}}^n - \mathbf{c}_{\mathcal{R}} \cdot \boldsymbol{\phi}_{\mathcal{R}}^n\right) \ , \tag{2.3}$$

where the $\boldsymbol{\phi}_{\mathcal{L},\ \mathcal{R}}^n$'s are vectors of the mixing ratio of the tracer in the cells surrounding the left/right edges and the $\mathbf{c}$'s are the coefficients that our method will calculate. Equation (2.3) is conservative, given suitable boundary conditions, due to the fact that the left flux of cell i is the right flux of cell $i-1$.

2.2.1 Generating the Advection Scheme

What we seek for our method is an approximation to the amount of tracer swept across each edge at each time-step. In general we do not know the exact region swept across each edge, nor do we have a continuous representation of the tracer field over the domain, both of which would be required to calculate the flux exactly. We therefore seek to construct an approximation to the tracer field and the region swept across each edge in order to approximate the fluxes. In one dimension the region swept across an edge is simply approximated as the region between edge i and $x(i) - u\Delta t$. The reconstruction of the tracer field used here, shown as $p_{_N}$ in Figure 2.1, follows the method of Leonard [22], using a local polynomial reconstruction at each edge. In this case the method is applied to finite volumes.

In order to fit a polynomial over a region we insist that the total amount of the tracer in a cell is the same amount interpolated over the polynomial we seek at time level n. Mathematically,

$$\Delta x_i \phi_i^n = \int_{cell\ i} p_{_N}(x) \mathrm{d}x, \tag{2.4}$$

where N is the order of the polynomial being used. This polynomial has $N+1$ terms and so we insist that the above holds for $N+1$ distinct cells. This generates $N+1$ equations for the $N+1$ unknowns and thus uniquely defines a polynomial. The reason why this will not be a singular problem will be made clear when we investigate the error of the schemes. If $N+1$ is even then the nearest $\frac{N+1}{2}$ cells on each side of the edge are used for a centred scheme, otherwise the nearest $\frac{N}{2}$ cells downwind of the edge and $\frac{N}{2}+1$ cells upwind of the edge are used for an upwinded scheme.

If we define a local coordinate system with the origin at the edge for which we wish to approximate the flux, and write $p_N(x)$ as $\sum_{j=0}^{N} a_j x^j$ then equation (2.4) can be written as

$$\Delta x_i \phi_i^n = \sum_{j=0}^{N} \int_{S_i}^{S_i+\Delta x_i} a_j x^j \mathrm{d}x \ , \tag{2.5}$$

where S_i is the distance of the left edge of cell i from the local origin. Integrating this gives

$$\phi_i^n = \frac{1}{\Delta x_i} \sum_{j=0}^{N} \frac{a_j}{j+1} \left\{ (S_i + \Delta x_i)^{j+1} - S_i^{j+1} \right\} \ . \tag{2.6}$$

Using the binomial expansion we can write this as

$$\phi_i^n = \sum_{j=0}^{N} \frac{a_j}{(j+1)\Delta x_i} \left(S_i^{j+1} + \sum_{k'=1}^{j+1} \left(\frac{(j+1)! \Delta x_i^{k'} S_i^{j+1-k'}}{(j+1-k')! k'!} \right) - S_i^{j+1} \right) \ . \tag{2.7}$$

Substituting $k' = k+1$ and simplifying gives

$$\phi_i^n = \sum_{j=0}^{N} a_j \sum_{k=0}^{j} \frac{j! \Delta x_i^k S_i^{j-k}}{(j-k)!(k+1)} \ . \tag{2.8}$$

By insisting that this holds for each cell in the stencil, we can write down the following $N+1$ equations for the unknown a_j's,

$$\phi_i^n = a_0 + \sum_{j=1}^{N} a_j \sum_{k=0}^{j} \frac{j!}{(j-k)!(k+1)!} \Delta x_i^k S_i^{j-k} \qquad , \ S_i \neq 0, \tag{2.9a}$$

$$\phi_i^n = a_0 + \sum_{j=1}^{N} \frac{a_j \Delta x_i^j}{j+1} \qquad , \ S_i = 0 \ . \tag{2.9b}$$

$S_i = 0$ for the cell to the right of the edge for which we are approximating the flux, i.e. cell i. The solution of these equations for the unknown a_j's (in vector $\mathbf{a}$) is a matrix inversion problem of the form

$$\begin{pmatrix} 1 & P_{1,i-\frac{N+1}{2}} & P_{2,i-\frac{N+1}{2}} & \cdots & \cdots & \cdots & P_{N,i-\frac{N+1}{2}} \\ \vdots & \vdots & \vdots & \ddots & & & \vdots \\ 1 & P_{1,i-1} & P_{2,i-1} & & & & P_{N,i-1} \\ 1 & \frac{\Delta x_i}{2} & \frac{\Delta x_i^2}{3} & & \ddots & & \frac{\Delta x_i^N}{N+1} \\ 1 & P_{1,i+1} & P_{2,i+1} & & & & P_{N,i+1} \\ \vdots & \vdots & \vdots & & & \ddots & \vdots \\ 1 & P_{1,i+\frac{N+1}{2}} & P_{2,i+\frac{N+1}{2}} & \cdots & \cdots & \cdots & P_{N,i+\frac{N+1}{2}} \end{pmatrix} \mathbf{a} = \boldsymbol{\phi} \ . \tag{2.10}$$

The first or last row, depending on the flow direction, will not be included if an even order polynomial is used (i.e. $N+1$ is odd). $P_{n,m}$ is a two dimensional polynomial in Δx_m and S_m consisting of all terms of order n, taken from equation (2.9a);

$$P_{n,m} = \sum_{k=0}^{n} \frac{n!}{(n-k)!(k+1)!} \Delta x_m^k S_m^{n-k} \ . \tag{2.11}$$

This matrix will be ill-conditioned if N is large even if Δx_m is of order one, since S_N will then be large. Assuming that the order of the polynomial being used is much smaller than the number of cells in the domain and that cell sizes vary gradually, S_m will be $O(\Delta x_i)$. In this case if Δx_i is small or large then the matrix will become ill conditioned as the polynomial order increases. This ill-conditioning can be lessened by factoring out a diagonal matrix of powers of a general Δx. Rewriting (2.10) in this way gives

$$\mathsf{B}\ \mathrm{diag}[1, \Delta x, \Delta x^2, \cdots, \Delta x^{N-1}]\mathbf{a} = \boldsymbol{\phi}\ , \tag{2.12}$$

B being what is left of the matrix in equation (2.10) after the factorisation. We can now write down the coefficients, a_j, of our interpolating polynomial,

$$a_j = \frac{1}{\Delta x^j} \sum_{i=0}^{N} \mathsf{B}_{ji}^{-1} \phi_i \quad j = 0, 1, 2, \cdots, N \tag{2.13}$$

where B_{ji}^{-1} is the element in the j^{th} row and $i{+}1^{\text{th}}$ column of B^{-1}. The flux across the boundary, $\hat{\phi}_{\mathcal{L}}$, can now be found by integrating this polynomial over the region swept across the edge in one time step,

$$\hat{\phi}_{\mathcal{L}}^n = \frac{1}{u\Delta t} \int_{-u\Delta t}^{0} \sum_{j=0}^{N} a_j x^j \mathrm{d}x\ . \tag{2.14}$$

Performing this integration and substituting equation (2.13) for the a_j's gives

$$\hat{\phi}_{\mathcal{L}}^n = \sum_{j=0}^{N} \frac{1}{u\Delta t} \frac{1}{\Delta x^j} \sum_{i=0}^{N} \mathsf{B}_{ji}^{-1} \phi_i^n \frac{-(-u\Delta t)^{j+1}}{j+1}\ . \tag{2.15}$$

Simplification yields

$$\hat{\phi}_{\mathcal{L}}^n = \sum_{j=0}^{N} \alpha^j \frac{(-1)^{j+1}}{j+1} \sum_{i=0}^{N} \mathsf{B}_{ji}^{-1} \phi_i^n\ , \tag{2.16}$$

where $\alpha = \frac{u\Delta t}{\Delta x}$ is the Courant number of the flow for the upwind cell at this edge. If the general Δx is chosen to be Δx_i from either the upwind or downwind cell of the edge then α is the Courant number corresponding to that cell. If Δx_i is uniform then α is the Courant number for that edge relative to both cells. We can now write down an equation for the

c's from equation (2.3),

$$c_{i_{\mathcal{L}}} = \begin{pmatrix} \frac{-\mathsf{B}_{i0}^{-1}}{1} \\ \frac{-\mathsf{B}_{i1}^{-1}}{2} \\ \vdots \\ \frac{-\mathsf{B}_{iN}^{-1}}{N+1} \end{pmatrix} \cdot \begin{pmatrix} 1 \\ -\alpha \\ \vdots \\ (-\alpha)^N \end{pmatrix} \qquad , i = 0, 1, \ldots, N \ . \tag{2.17}$$

If the flow is constant in time then $\boldsymbol{c}$ can be calculated once for each edge, otherwise this matrix-vector product will need recomputing each time the flow changes at that edge.

Using this method, we can calculate the flux at each edge as the dot product of a vector of coefficients with a vector of tracer values in cells neighbouring the edge. The vector of coefficients is constant at an edge in equation (2.3) if the flow across that edge is constant, otherwise it can be formed from a matrix-vector product. This vector is dependent on the speed and direction of the flow and the matrix is dependent only on the direction of the flow and the grid. In order to compute this method efficiently, we first form the matrix B for each edge and calculate its inverse. If this is the first time step or if the flow has changed at an edge, then the product of the corresponding B^{-1} and $\boldsymbol{\alpha}$ is formed. The dot product of this vector, or one calculated at a previous time step, with the corresponding $\boldsymbol{\phi}$ gives the flux.

This enables coefficients of α and the ϕ_i's to be calculated for all cells. The coefficients of α depend on the grid size and so need only be calculated once for a regular grid. When this is the case the coefficients of ϕ depend only on α and need only be calculated once for uniform, constant flow. These coefficients are used to calculate the fluxes for the update equation which are in turn used to approximate the tracer distribution at the next time step. This last step is repeated for each time step.

2.2.2 Error Analysis

We define the error in a cell at time t_n to be the difference between the actual amount of the tracer in that cell and the approximated amount, thus

$$e_i^{n+1} = \phi_i^{n+1} - \int_{(i-1)\Delta x}^{i\Delta x} \phi(t_{n+1}, x)\mathrm{d}x \ . \tag{2.18}$$

The right hand side can be written in terms of the values at a previous time and the fluxes across each edge of the cell,

$$e_i^{n+1} = \left(\phi_i^n + \sum_{\text{edges}} \hat{\phi}^n\right) - \left(\int_{(i-1)\Delta x}^{i\Delta x} \phi(t_n, x)\mathrm{d}x \ + \sum_{\text{edges}} \mathrm{FLUX(t_n)}\right) \ , \tag{2.19}$$

or equivalently,

$$e_i^{n+1} = e_i^n + \sum_{\text{edges}} \left(\int_{\mathrm{area_n}} p(x)\mathrm{d}x \ - \int_{\mathrm{area(t_n)}} \phi(t_n, x)\mathrm{d}x\right) \ . \tag{2.20}$$

Adopting the same local origin as before gives the areas as

$$\int_{\mathrm{area(t_n)}} dx \equiv \int_{0-\int_{n\Delta t}^{(n+1)\Delta t} u(t,0)\mathrm{d}t}^{0} dx \ , \tag{2.21a}$$

and

$$\int_{\mathrm{area_n}} dx \equiv \int_{0-\int_{n\Delta t}^{(n+1)\Delta t} u(n\Delta t,0)\mathrm{d}t}^{0} dx \ , \tag{2.21b}$$

where area(t_n) is the true area swept across the edge in time step n, and area$_n$ is the approximation to this area. Then, if $u(t, 0) = u_0$ over this time period,

$$e_i^{n+1} = e_i^n + \sum_{\text{edges}} \int_{-u_0\Delta t}^{0} \{p(x) - \phi(t_n, x)\}\,\mathrm{d}x \ . \tag{2.22}$$

This shows that the error in one cell depends on the accuracy of the polynomial interpolation in the region swept across the edge at that time step. As would be expected, the more accurate the interpolation, the smaller the error will be.

The polynomial is fitted by insisting that

$$\Delta x_i \phi_i^n = \int_{S_i}^{S_{i+1}} p_{_N}(x)\mathrm{d}x \ , \tag{2.23}$$

over $N+1$ cells. The right hand side of this equation can be re-written, using the mean value theorem, as

$$\Delta x_i \phi_i^n = \Delta x_i p_{_N}(x_\zeta) \ , \tag{2.24}$$

where $S_i < x_\zeta < S_{i+1}$. Written in this form we can clearly see that the interpolation of equation (2.10) is $N+1^{\mathrm{th}}$ order accurate in space and thus, from equation (2.22), so is the error in each cell. If $u(t,0) \neq u_0$ then there will also be a contribution to the error from the approximation of the area swept across the edge.

Summing over each cell to find a bound on the global error gives,

$$\left|e^{n+1}\right| \leq |e^n| + 2\sum_i \int_{i\Delta x - u_0\Delta t}^{i\Delta x} \left|p_{i_N}(x) - \phi(t_n, x)\right| \mathrm{d}x \ , \tag{2.25}$$

where i runs over all edges on the grid and $p_{i_N}(x)$ is the N^{th} order polynomial fitted around edge i in the global coordinates. This global error again depends on the accuracy of the polynomial interpolation and so will also be of order $N+1$ if $u = u_0$.

This argument shows that the order of accuracy of these schemes is formally related to the order of the polynomial. Fitting polynomials through data points can generate highly oscillatory polynomials for which the accuracy can vary widely between two nearby points. This means that the argument is only valid for showing how this error changes as the resolution is varied but does not tell us how different order schemes compare for a particular resolution grid.

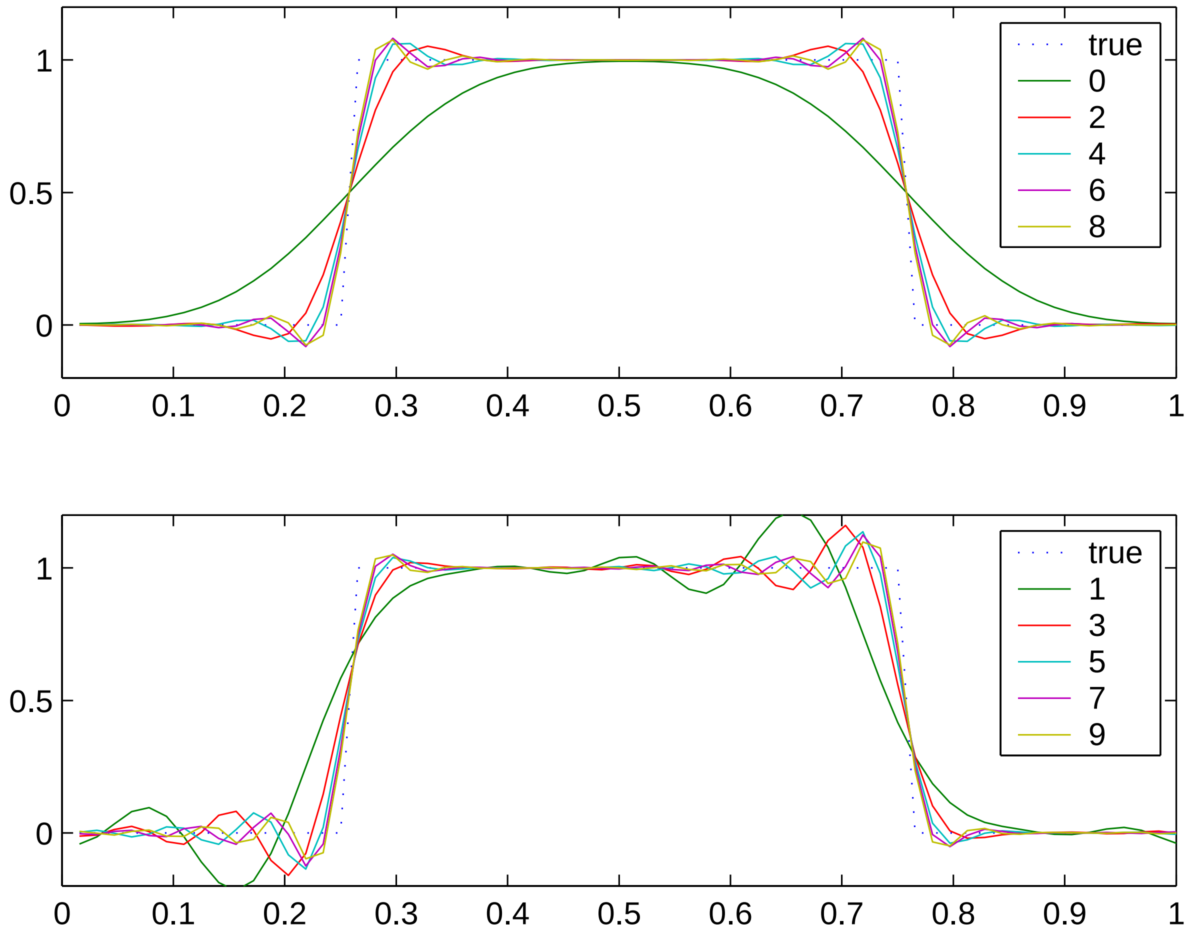


Figure 2.2: Numerical solutions of the advection of a step once around a periodic domain divided into 64 equal cells. The order of the polynomials used in each method is shown in the plots for the even order (top) and odd order (bottom) cases

For some later versions of these schemes we shall not be able to formalise the relationship between polynomial and accuracy orders but we shall continue to refer to schemes by their 'order'. In this work, the term order will be used to refer to the order of the polynomial used for the interpolation and not the order of accuracy of the scheme. In this way the First Order Upwind scheme will be referred to as the zeroth order scheme.

2.2.3 Numerical Results

The results of applying these schemes to a step initial profile on a regular grid are shown in Figure 2.2. A grid of 64 cells was used with a uniform flow of 1 unit/second along a periodic domain of length 1. A time step of 1/128 was used to give a Courant number of $1/2$. The top plot of Figure 2.2 shows the results of the schemes using even order polynomials, these all have the correct phase speed and only produce a few, small oscillations around the steep gradient. By contrast, the odd order schemes advect the profile too slowly and produce more oscillations around the sharp gradient.

Using a higher order polynomial better captures the steep gradient and phase speed but introduces more oscillations into the solution. The oscillations increase in frequency as higher order polynomials are used. For the even order cases the amplitude of these oscillations increases slightly with increasing order but the reverse is true for the odd order cases. Unsurprisingly, these results are very similar to those found by Leonard [22] for the regular grid case. The only difference between the methods on grid points and over finite volumes is the way the polynomials have been fitted.

In the finite volume case, irregularly sized grid cells have been allowed in this computation, due to the process being automated. This means more thought must be given to the choice of a suitable time step. A smaller time step will generally improve the accuracy but for this method will probably have a bigger improvement in the smaller cells than the larger ones. Figure 2.3 shows the results of the same case but on a grid with different sized cells. The 16 cells at either end of the periodic domain have a size $\Delta x = 1/96$, the middle 32 cells have size $\Delta x = 1/48$. A time step of two thirds the previous case was used so that the Courant number in the smaller cells was $1/2$. The results are very similar to those on the regular grid. The smaller grid cells can better maintain a sharp gradient

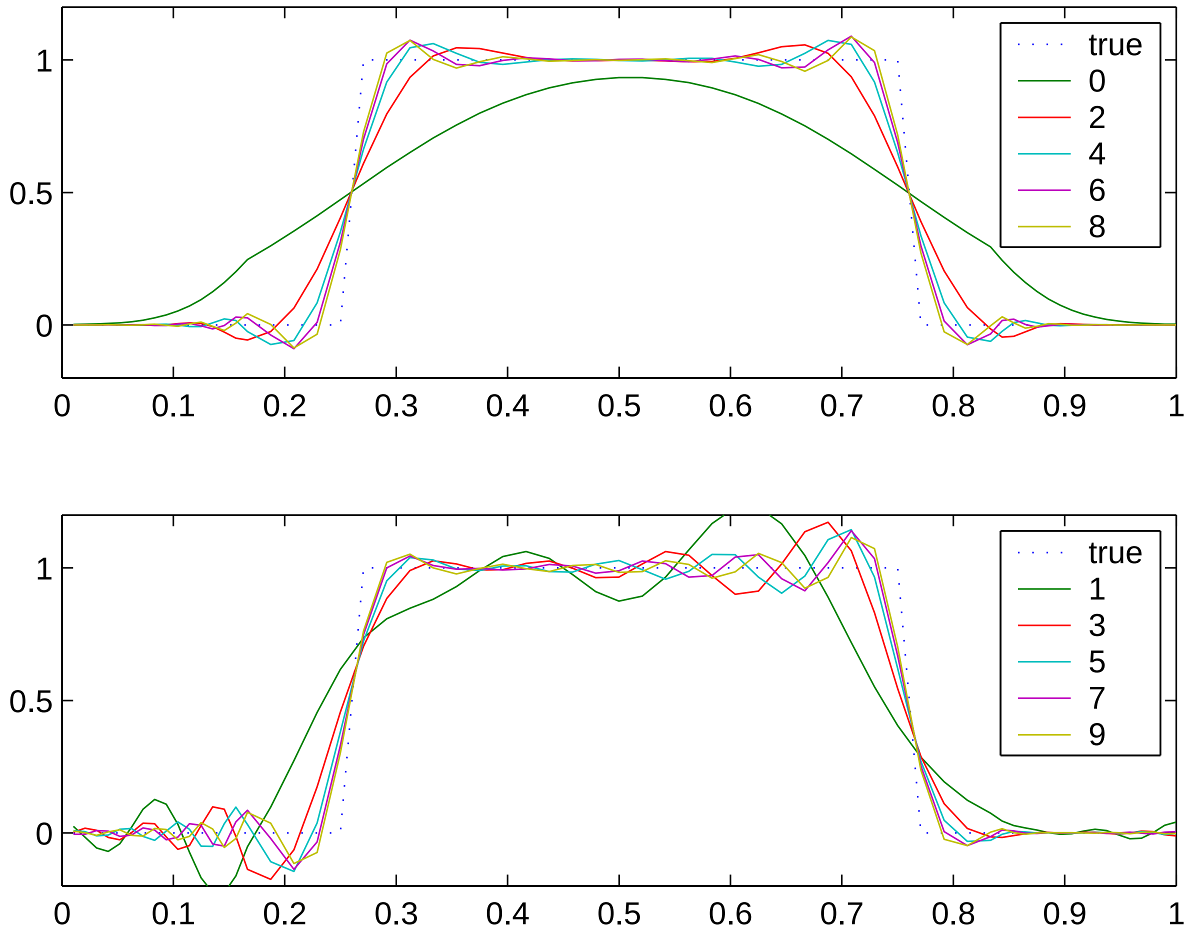


Figure 2.3: Numerical solutions of the advection of a step once around a periodic domain divided into 64 unequal cells. The order of the polynomials used in each method is shown in the plots for the even order (top) and odd order (bottom) cases

but the larger cells cannot so the overall solutions are very similar. These differences in cell sizes do show up in the lower order schemes where the smoothing of the profile or the oscillations cross the boundary between the different sized cells. Where this happens there is a marked change in the gradient of the profile at the boundary.

With these results behaving exactly as we would expect we shall move on to extend the ideas that generated these schemes into two dimensions.

2.3 Two-Dimensional Advection

We can attempt to extend the same method into two dimensions on a rectangular grid simply by using two-dimensional polynomials and integration. Another way to apply the method in two dimensions would be to use dimensional splitting. This would involve applying the one-dimensional method separately in each coordinate direction. This has been shown by Leonard et al. [23] and Thuburn [51] to cause distortion of an advected profile when the the flow is not aligned to the grid.

The UTOPIA scheme [23] uses a similar idea to the one-dimensional schemes but with a two-dimensional polynomial used for the interpolation. This was shown to reduce the anisotropic distortion generated by using the one-dimensional schemes and dimensional splitting. We can extend this idea to use higher order polynomials in a fairly straight forward manner. Leonard et al. [23] suggested extending the UTOPIA scheme by using higher order one dimensional terms ($1,\ x,\ x^2, \dots$) in the coordinate directions but retaining the lower order cross terms ($xy,\ x^2y\ xy^2$).

2.3.1 Generating the Advection Scheme

We again start by insisting that the integral of our polynomial over a cell is equal to the total tracer amount in that cell. Thus

$$\phi^n_{ij}\Delta x\Delta y = \int\int_{cell\ \ ij} \sum_{k=0}^{N}\sum_{l=0}^{k} a_z x^{k-l} y^l \ \mathrm{d}x\mathrm{d}y \ , \tag{2.26}$$

where the index $z = \frac{(k+1)k}{2} + l$ and $i,\ j$ are relative to an, as yet unspecified, origin. We now have a polynomial with $K (= \frac{(N+1)N}{2})$ terms for which we need to find the K coefficients, a_z. If the origin is at a cell vertex then performing this integration gives

$$\phi_{ij}^{n}=\sum_{k=0}^{N}\sum_{l=0}^{k}\frac{a_z}{\Delta x\Delta y}\left(\frac{(i\Delta x)^{k-l+1}}{k-l+1}-\frac{((i-1)\Delta x)^{k-l+1}}{k-l+1}\right)\left(\frac{(j\Delta y)^{l+1}}{l+1}-\frac{((j-1)\Delta y)^{l+1}}{l+1}\right), \tag{2.27}$$

which can be simplified using the binomial expansion to give

$$\phi_{ij}^{n}=\sum_{k=0}^{N}\sum_{l=0}^{k}a_z\Delta x^{k-l}\Delta y^{l}\left(\sum_{m=0}^{k-l}\frac{(-1)^m(k-l)!i^{k-l-m}}{(k-l-m)!m!}\right)\left(\sum_{p=0}^{l}\frac{(-1)^p(l)!j^{l-p}}{(l-p)!p!}\right). \tag{2.28}$$

By insisting that equation (2.28) holds over K cells we can set up a matrix problem of the form, $\mathsf{B}\mathbf{a}=\boldsymbol{\phi}$. By carefully selecting the stencil over which we fit this polynomial, we can ensure that B is non-singular and thus we have a unique N^{th} order polynomial.

If more than $N+1$ cells are used in any row of the grid then, when $l=0$, there are more than $N+1$ distinct values of i with which to find the $N+1$ coefficients of the powers of x. This will force a row degeneracy and the matrix will be singular. The matrix will also be singular if more than $N+1$ cells are used in any column, since when $k=l$ there are more than $N+1$ distinct j values. We therefore begin to build up a stencil by selecting $N+1$ cells in one row $(j=d_2)$ and $N+1$ cells in one column $(i=d_1)$. These cells are centred about the edge for which we wish to approximate the flux or the cell upwind of it (see Figure 2.4 a).

By using these cells to form a matrix in a similar way to the one-dimensional method, we are in practice fitting two one dimensional polynomials over these cells. We can diagonalise the blocks that are dependent only on the powers of x and the powers of y. The remaining terms in the polynomial are multiples of these terms and where i or j is constant this multiple is constant. Because of this constant multiple, the row operations that diagonalise the x and y power terms zero the remaining terms in all except one of these rows. This gives a matrix, B, which has the

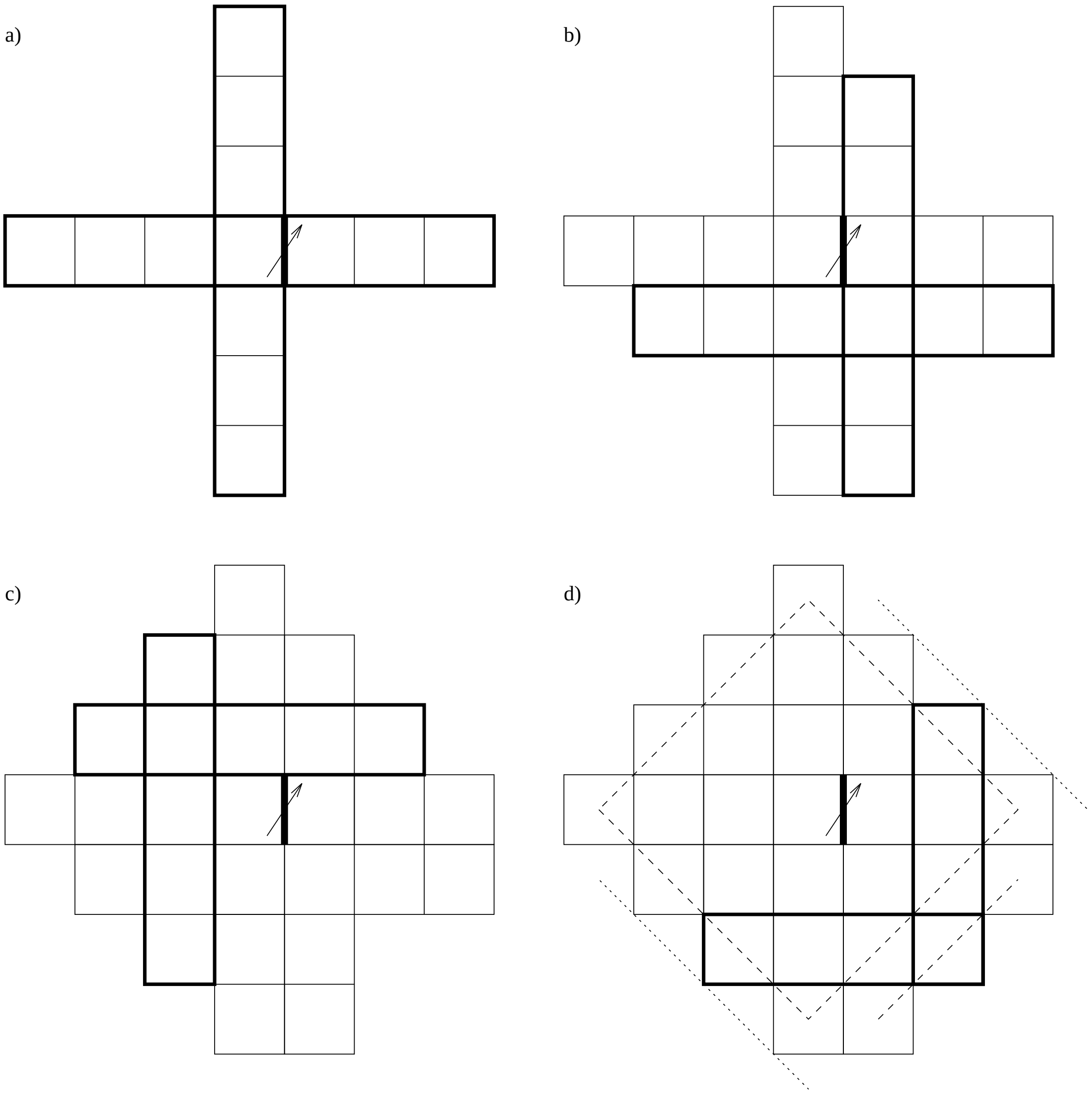


Figure 2.4: Building a stencil for interpolating a sixth order polynomial. a) Seven cells in one row and one column are included for the terms that are powers of x or y. b) Another row and column each of six cells is added for terms xy^n and $x^n y$ where $1 \leq n \leq N - 1$. c) Cells are added to ensure there are five cells in another row and column. d) The final cell is added to make a row and column of four cells. A similar method using a seventh order polynomial would result in the cells along the narrow dotted lines also being included in the stencil.

form,

$$
\left(\begin{array}{c|c|c|c}
1 & (\Delta x)^m & (\Delta y)^n & (\Delta x)^k(\Delta y)^l \\ \cline{2-4}
0 & & & \\
\vdots & N \times N & \emptyset & \\
\vdots & & & \\ \cline{2-3}
\vdots & & & \emptyset \\
\vdots & \emptyset & N \times N & \\
\vdots & & & \\ \cline{2-4}
\vdots & & & \\
\vdots & \multicolumn{2}{c|}{\emptyset} & K-2N-1 \times K-2N-1 \\
\vdots & & & \\
0 & & &
\end{array}\right)
\begin{array}{l}
,\ i = d_1,\ j = d_2 \\
\\
,\ j = d_2 \\
\\
\\
,\ i = d_1 \\
\\
\\
\\
\\
\\
\end{array}
$$

The first row shows the order of the terms on each column where, $n, m = 1, 2, \dots, N$ and $k, l = 1, 2, \dots, N-1$ with $k + l \leq N$. Non-zero blocks have only their sizes shown and the $N \times N$ blocks have been diagonalised. The lower right block contains the information from the remaining cells in the stencil that we have not yet selected. If we now ensure that we have N cells in a different row and a different column (see Figure 2.4 b) we can diagonalise the part of this block dependent on the terms xy^i and x^iy where $i = 1, 2, \dots, N-1$. This row or column is chosen to be down-winded if the previous row/column was up-winded, or up-winded if the last choice was centred or down-winded. We can continue on in this way until the matrix B is fully diagonalised, thus guaranteeing that the matrix is not singular and giving a fast method of calculating it's inverse if required.

Another way to think of this stencil is that it must contain the following; One row of $N+1$ cells for the terms $1,\ x,\ x^2, \dots,\ x^N$, one column of $N+1$ cells for the terms $1,\ y,\ y^2, \dots,\ y^N$ (the row and column must overlap as both fit the constant term) , two rows of at least N cells for the terms $xy,\ x^2y,\dots,\ x^{N-1}y$, two columns of at least N cells for the terms $xy,\ xy^2,\dots,\ xy^{N-1}$, three rows of at least $N-1$ cells for the terms $x^2y^2,\ x^3y^2,\dots,\ x^{N-2}y^2$, three columns of at least $N-1$ cells for the terms $x^2y^2,\ x^2y^3,\dots,\ x^2y^{N-2}$, etc. A third way of building this stencil is to ensure that there is an $(m+1)\times(n+1)$ region for every term x^my^n. These regions are each centred or upwinded about the edge as necessary.

This method of stencil selection will yield a stencil of a particular form, that of a diamond region of cells with cells added along one or three edges of the diamond region. One edge will be used if an even order polynomial is being fitted; which edge is included depends on the direction of the flow. Three edges are included if an odd order polynomial is being used; which edges are used again depends on the flow direction. Figure 2.4 d shows an example of the stencil used for a sixth order polynomial.

Using this method of stencil selection, there are four possible stencils for each edge and the edges have two orientations so there are a total of eight possible stencils. Two of these stencils are identical while of the remaining six stencils, three are transpositions of the other three. If we define the origin to be the top or right node of the edge and refer to the edges by the cardinal directions, then the identical stencils are for flow from the northwest quadrant across the east edge and flow from the opposite direction across the north edge (stencils 3 and b from Figure 2.5). Stencil d is the same as stencil 1, translated one cell northwards, stencil 4 is stencil a translated one cell to the east. Stencil c and stencil 2 are also the same, translated one cell in each coordinate direction. We only consider the stencils for the north and east edges here as the west and south edges are the east and north edges of neighbouring cells.

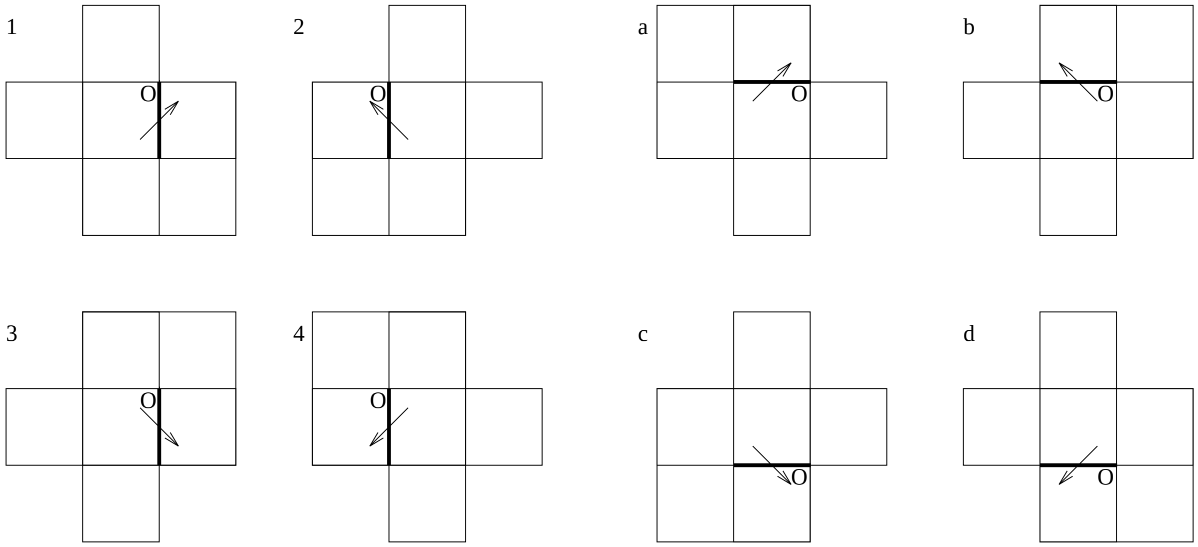


Figure 2.5: Orientation of stencils used for fitting a second order polynomial for all possible flow directions and edge orientations. Stencils 3 and b are identical, the remaining six stencils form pairs that are similar by translation; 1 & d, 2 & c,4 & a.

There are other possible methods for selecting a stencil, but this method does have several advantages. Firstly, and most importantly, it guarantees a non-singular matrix that can be easily inverted using any suitable method (e.g. Gaussian elimination). Secondly, the stencils are all compact and centred or upwinded about the edge across which we are approximating the flux. The stencils also have a simple form that can easily be extended for higher order polynomials and to higher dimensions.

These stencils do not however, guarantee that the matrix B is well conditioned. As before, it becomes more ill-conditioned as the order of the polynomial is increased. Experiments will show (e.g. see Section 2.4.1) that a significant amount of accuracy will be lost when using seventh or higher order polynomials and single precision arithmetic. One way to improve the conditioning of B is to use orthogonal basis functions, such as Legendre polynomials, instead of simple powers of x and y. This will not alter the polynomial we are seeking since it is unique. It should however, improve the conditioning of B and will lead to a different form of the polynomial being found.

Once we have fitted a polynomial over the region around an edge we can then integrate it over the area swept across the edge to approximate the flux. For the purpose of the following derivation we shall continue to use powers of x and y for the interpolated polynomial which can either be calculated from the Legendre polynomials or used directly. The flux in the x-direction is therefore given by

$$FLUX_x = \frac{1}{u\Delta t\Delta y}\int_{-u\Delta t}^{0}\int_{wx-\Delta y}^{wx}\sum_{k=0}^{N}\sum_{l=0}^{k} a_z x^{k-l} y^l \;\mathrm{d}y\;\mathrm{d}x\ , \qquad (2.29_x)$$

where $w = \frac{v}{u}$, the direction of the flow at the edge. The flux in the y-direction can be found from a similar equation,

$$FLUX_y = \frac{1}{v\Delta t\Delta x}\int_{-v\Delta t}^{0}\int_{\frac{y}{w}-\Delta x}^{\frac{y}{w}}\sum_{k=0}^{N}\sum_{l=0}^{k} a_z x^{k-l} y^l \;\mathrm{d}x\;\mathrm{d}y\ , \qquad (2.29_y)$$

where, u, v and hence w may be different from the previous case. Changing the order of the summation and integration and performing the first integration from equation (2.29_x) gives

$$FLUX_x = \sum_{k=0}^{N}\sum_{l=0}^{k}\int_{-u\Delta t}^{0}\frac{a_z x^{k-l}}{u\Delta t\Delta y}\sum_{m=0}^{l}\frac{-l!(wx)^{l-m}\Delta y^{m+1}}{(l-m)!(m+1)!}\;\mathrm{d}x\ . \qquad (2.30)$$

Combining the terms involving powers of x and integrating again gives

$$FLUX_x = \sum_{k=0}^{N}\sum_{l=0}^{k} -1^k l! a_z \sum_{m=0}^{l}\frac{(u\Delta t)^{k-m} w^{l-m}(\Delta y)^m}{(l-m)!(m+1)!(k-m+1)}\ . \qquad (2.31)$$

Substituting for a_z from $\mathbf{a} = \mathsf{B}^{-1}\phi$ and rewriting in terms of Courant numbers gives

$$FLUX_x = \sum_{n=1}^{K}\phi_n\sum_{k=0}^{N}\sum_{l=0}^{k}(-\alpha_x)^k l! \mathsf{B}^{-1}_{zn} w^l \Delta x^k \sum_{m=0}^{l}\frac{\alpha_y^{-m}}{(l-m)!(m+1)!(k-m+1)}\ , \qquad (2.32_x)$$

where B^{-1}_{zn} is the element in the n^{th} column and $z+1^{\mathrm{th}}$ row of B^{-1}, $\alpha_x (= \frac{u\Delta t}{\Delta x})$ is the Courant number of the edge and $\alpha_y = \frac{v\Delta t}{\Delta y}$ which is not strictly a Courant number for this edge. If $\alpha_x = 0$ then this equation will not hold when $k = 0$, but when $\alpha_x = 0$, u must be zero so there will be no flux across the edge. When $\alpha_y = 0$ then v, and therefore w, must be zero, in this case integrating equation (2.29_x) gives

$$FLUX_x = \sum_{n=1}^{K} \phi_n \sum_{k=0}^{N} \sum_{l=0}^{k} \frac{\mathsf{B}_{zn}^{-1}(-u\Delta t)^{k-l}(-\Delta y)^l}{(k-l+1)(l+1)} \ . \qquad (2.33_x)$$

Performing the same procedure for fluxes in the y-direction gives

$$FLUX_y = \sum_{n=1}^{K} \phi_n \sum_{k=0}^{N} \sum_{l=0}^{k} (-\alpha_y)^k (k-l)! \mathsf{B}_{zn}^{-1} \left(\frac{1}{w}\right)^{k-l} \Delta y^k \sum_{m=0}^{l} \frac{\alpha_x^{-m}}{(k-l-m)!(m+1)!(k-m+1)} \ , \qquad (2.32_y)$$

in this case $\alpha_y (= \frac{v\Delta t}{\Delta y})$ is the Courant number for the edge and $\alpha_x = \frac{u\Delta t}{\Delta x}$. If $\alpha_y = 0$ then the flux across the edge is zero, while if $\alpha_x = 0$ then

$$FLUX_y = \sum_{n=1}^{K} \phi_n \sum_{k=0}^{N} \sum_{l=0}^{k} \frac{\mathsf{B}_{zn}^{-1}(-v\Delta t)^{l}(-\Delta x)^{k-l}}{(l+1)(k-l+1)} \ . \qquad (2.33_y)$$

These equations can be written in the form

$$FLUX = (\mathsf{G}\boldsymbol{\alpha}) \cdot \boldsymbol{\phi} \qquad (2.34)$$

where $\boldsymbol{\alpha}$ is a vector of length K containing the terms $\alpha_x^s \alpha_y^t$ and G is a $K \times K$ matrix. s and t are integers between zero and N whose sum is N or less. Except under special conditions, $\boldsymbol{\phi}$ will change each time step, $\boldsymbol{\alpha}$ may also vary with time but G will only change if the stencil changes (i.e. the flow direction changes to a different quadrant). This means that we can precalculate the G's for each edge, then at each time-step the flux can be calculated by forming $\boldsymbol{\alpha}$ and $\boldsymbol{\phi}$ and combining them with the relevant G.

This may seem like an expensive way of computing the fluxes but it is little more so than any other method. Since we need only calculate equation (2.34) at each time step we can compare the cost of this with other methods. Most standard methods can be written in the form of equation (2.34) for example, the UTOPIA scheme of Leonard can be written in this form for stencil 1 in Figure 2.5, with

$$\mathsf{G}_{UTOPIA} = \begin{pmatrix} \frac{5}{6} & \frac{1}{2} & \frac{1}{4} & -\frac{1}{3} & -\frac{1}{4} & -\frac{1}{2} \\ 0 & 0 & -\frac{1}{4} & 0 & 0 & \frac{1}{4} \\ -\frac{1}{6} & 0 & 0 & \frac{1}{6} & 0 & 0 \\ \frac{1}{3} & -\frac{1}{2} & -\frac{1}{4} & \frac{1}{6} & \frac{1}{4} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4} \\ 0 & 0 & \frac{1}{4} & 0 & -\frac{1}{4} & 0 \end{pmatrix}, \ \boldsymbol{\alpha} = \begin{pmatrix} 1 \\ \alpha_x \\ \alpha_y \\ \alpha_x^2 \\ \alpha_x\alpha_y \\ \alpha_y^2 \end{pmatrix} \ \& \ \boldsymbol{\phi} = \begin{pmatrix} \phi^n_{0,0} \\ \phi^n_{0,1} \\ \phi^n_{-1,0} \\ \phi^n_{1,0} \\ \phi^n_{0,-1} \\ \phi^n_{1,-1} \end{pmatrix}.$$

With the UTOPIA scheme we have calculated G on paper where our new method computes it automatically. Since we have used a similar reasoning for developing our method as can be used for deriving the UTOPIA scheme, the matrices G are similar for the UTOPIA scheme and our scheme based on second order polynomial interpolation. For the method described here we get

$$\mathsf{G} = \begin{pmatrix} \frac{5}{6} & \frac{1}{2} & -\frac{1}{4} & -\frac{1}{3} & \frac{1}{3} & -\frac{1}{3} \\ 0 & 0 & \frac{1}{4} & 0 & 0 & \frac{1}{6} \\ -\frac{1}{6} & 0 & 0 & \frac{1}{6} & 0 & 0 \\ \frac{1}{3} & -\frac{1}{2} & \frac{1}{4} & \frac{1}{6} & -\frac{1}{3} & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{3} & \frac{1}{6} \\ 0 & 0 & -\frac{1}{4} & 0 & \frac{1}{3} & 0 \end{pmatrix},$$

which is related to G_{UTOPIA} by

$$\mathsf{G}_{UTOPIA} = \mathsf{G}\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{3}{4} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{6}{4} \end{pmatrix} . \qquad (2.35)$$

The UTOPIA method can be thought of as using a second order polynomial fitted through grid points (Leonard et al. [23]) whilst the new method uses fitting over control volumes. This would cause a slight difference between the coefficients of each scheme and could account for the difference between the two matrices G for each method.

Once the fluxes have been calculated at each edge it is then possible to update the ϕ-value in each cell by simply adding or subtracting the appropriate fluxes,

$$\phi_{ij}^{n+1} = \phi_{ij}^{n} + \sum_{\text{inflow}} FLUX - \sum_{\text{outflow}} FLUX \; . \qquad (2.36)$$

Before undertaking this step we may apply a limiter to these fluxes, in order to ensure shape preservation of the ϕ-field.

This method does not impose a time step restriction on the scheme, any value of Δt can be used and the scheme will work. This does not guarantee stability of the generated scheme but we might reasonably expect the stability conditions of the Lax-Wendroff and UTOPIA schemes to hold for our comparable schemes. For higher order schemes we may be concerned about taking small time steps and using parts of high amplitude, high frequency oscillations to calculate our fluxes. If we apply a suitable flux limiter then this will ensure the stability of the schemes possibly by imposing a stability condition on the Courant number. We shall now take a closer look at such a limiter.

2.3.2 The Flux Limiter

We have a simple form for calculating fluxes that requires a matrix-vector multiplication and a dot product be computed. A flux limiter will require knowledge of the ϕ-field, so if we wish our matrix G to remain constant in time we must use a limiter that post-processes the fluxes and does not alter G. One such flux limiter is the Universal Limiter of Leonard [22] with refinements proposed by Thuburn [51]. The limiter is explained in detail in those papers, here we shall only look at the application of the method itself.

The general principle is to take information about the flow and the distribution of the advected quantity and use them to calculate bounds on the fluxes that will ensure shape preservation. As each edge is both an inflow to one cell and outflow from another, two sets of bounds are generated for each edge. If the flux lies outside these bounds then it is adjusted to be the nearest allowable value. In the worst case the bounds overlap at only one value, the concentration in the upwind cell, and the limiter works as the first order upwind scheme at that edge for that time-step.

This description of the limiter is given in terms of ϕ being a mixing ratio, since we are working with a constant density this will still hold if ϕ is a concentration. The method for generating the initial fluxes will also hold if ϕ is a mixing ratio, which is how it will be treated when the density is not constant.

Using the notation that $\hat{\phi}$ denotes a flux across an edge, the limiter

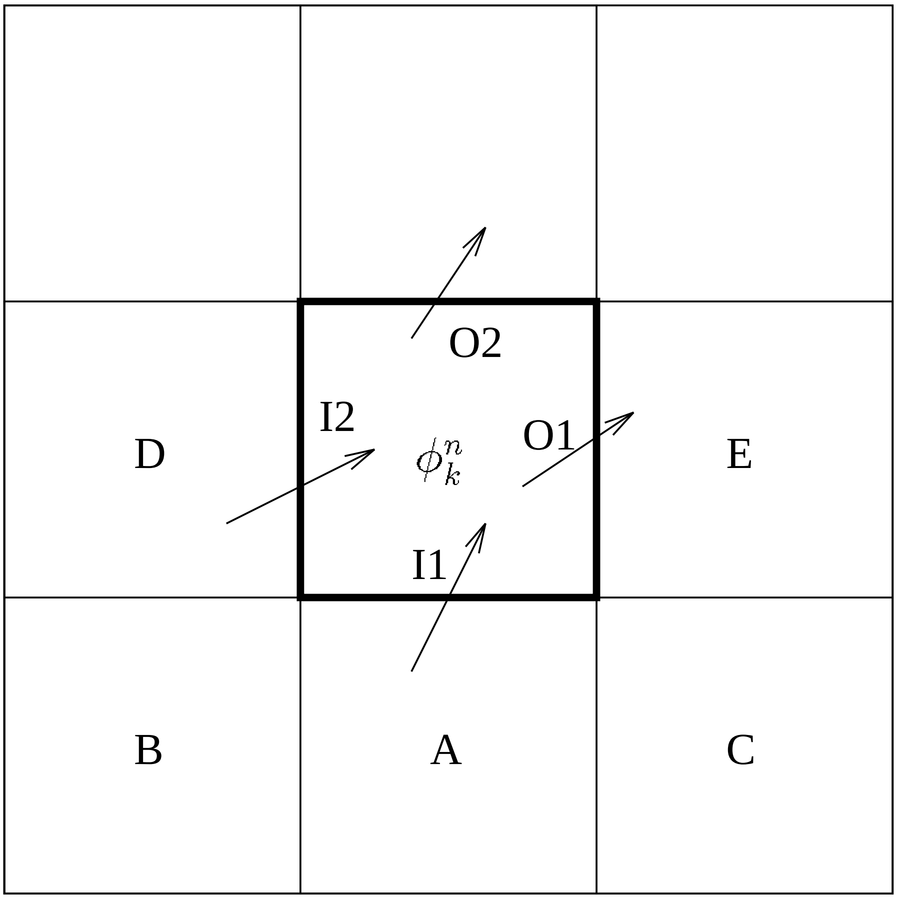


Figure 2.6: Diagram showing the labelling conventions used in describing the flux limiter. The arrows refer to the direction of the flow so I and O refer to inflow and outflow edges of cell k respectively

works as follows;

Step 1: Inflow bounds are defined as (for edge I1 in Figure 2.6),

$$(\hat{\phi}_{\mathrm{I1}}^{(\mathrm{in})})_{\mathrm{min}} = \min\{\phi_{\mathrm{min}}^{\mathrm{up}},\ \phi_k^n\} \ , \qquad (2.37_{\mathrm{min}})$$

$$(\hat{\phi}_{\mathrm{I1}}^{(\mathrm{in})})_{\mathrm{max}} = \max\{\phi_{\mathrm{max}}^{\mathrm{up}},\ \phi_k^n\} \ , \qquad (2.37_{\mathrm{max}})$$

where $\phi_{\mathrm{min}}^{\mathrm{up}}$ is the minimum mixing ratio in an upstream neighbourhood of edge I1. This neighbourhood will include cell A, and also cells B and C if there is flow from these cells into A. If B is included in the upwind region and there is flow from cell D into B, then cell D should be included. Cell E could also be included in this way, even though it is down wind of edge O2.

Step 2: Ensure that the flux lies within these bounds by setting

$$\hat{\phi}'_{\mathrm{I1}} = \min\left\{\hat{\phi}^{(\mathrm{scheme})}_{\mathrm{I1}},\ (\hat{\phi}^{(\mathrm{in})}_{\mathrm{I1}})_{\mathrm{min}}\right\} \ , \tag{2.38}$$

$$\hat{\phi}''_{\mathrm{I1}} = \max\left\{\hat{\phi}'_{\mathrm{I1}},\ (\hat{\phi}^{(\mathrm{in})}_{\mathrm{I1}})_{\mathrm{max}}\right\} \ . \tag{2.39}$$

Step 3: Define upper and lower bounds on ϕ_k^{m+1} by

$$(\phi_k^{n+1})_{\mathrm{min}} = \min_i\left\{\hat{\phi}^{(\mathrm{in})}_i\right\} \ , \tag{2.40_{min}}$$

$$(\phi_k^{n+1})_{\mathrm{max}} = \max_i\left\{\hat{\phi}^{(\mathrm{in})}_i\right\} \ , \tag{2.40_{max}}$$

where i refers to the inflow edges of cell k. If there is no inflow into cell k then $(\phi_k^{n+1})_{\mathrm{min}}$ and $(\phi_k^{n+1})_{\mathrm{max}}$ are set equal to ϕ_k^n

Step 4: Revise the inflow bounds,

$$(\hat{\phi}^{(\mathrm{in})}_{\mathrm{I1}})_{\mathrm{min}} = \min\{\phi^{\mathrm{up}}_{\mathrm{min}},\ \phi''_{\mathrm{I1}}\} \ , \tag{2.41_{min}}$$

$$(\hat{\phi}^{(\mathrm{in})}_{\mathrm{I1}})_{\mathrm{max}} = \max\{\phi^{\mathrm{up}}_{\mathrm{max}},\ \phi''_{\mathrm{I1}}\} \ , \tag{2.41_{max}}$$

Step 5: Calculate the outflow bounds from

$$(\hat{\phi}^{(\mathrm{out})}_k)_{\mathrm{min}} = \frac{\phi_k^n + \sum_{\mathrm{in}}\alpha_i(\hat{\phi}^{(\mathrm{in})}_i)_{\mathrm{max}} - (\phi_k^{m+1})_{\mathrm{max}}\,(1 + \sum_{\mathrm{in}}\alpha_i\sum_{\mathrm{out}}\alpha_j)}{\sum_{\mathrm{out}}\alpha_j} \ , \tag{2.42_{min}}$$

$$(\hat{\phi}^{(\mathrm{out})}_k)_{\mathrm{max}} = \frac{\phi_k^n + \sum_{\mathrm{in}}\alpha_i(\hat{\phi}^{(\mathrm{in})}_i)_{\mathrm{min}} - (\phi_k^{m+1})_{\mathrm{min}}\,(1 + \sum_{\mathrm{in}}\alpha_i\sum_{\mathrm{out}}\alpha_j)}{\sum_{\mathrm{out}}\alpha_j} \ . \tag{2.42_{max}}$$

Step 6: Ensure the flux lies within these bounds,

$$\hat{\phi}'''_{\mathrm{O1}} = \min\left\{\hat{\phi}''_{\mathrm{O1}},\ (\hat{\phi}^{(\mathrm{out})}_k)_{\mathrm{max}}\right\} \ , \tag{2.43}$$

$$\hat{\phi}_{\mathrm{O1}} = \max\left\{\hat{\phi}'''_{\mathrm{O1}},\ (\hat{\phi}^{(\mathrm{out})}_{\mathrm{I1}})_{\mathrm{min}}\right\} \ . \tag{2.44}$$

The α's used here are modified Courant numbers, scaled by the ratio between the density flux across the edge and the density in the cell. In the following work the density remains constant so this ratio is equal to one, where this is not the case then the Courant numbers will need scaling accordingly. For the limiter to ensure shape preservation it also requires that the sum of the Courant numbers over the outflow edges (the denominators in equations (2.42_{min}) and (2.42_{max})) is less than one. This imposes a time step restriction on the scheme which may or may not be stricter than the restriction on the basic scheme itself. The adjusted fluxes can now be used in equation (2.36) to update the ϕ-field.

2.3.3 Error Analysis

We can use a similar argument to that used in the one dimensional case to try and get an idea of how the numerical error of this scheme. We again define the error in a cell at time t_n to be the difference between the actual amount of the tracer in that cell and the approximated amount,

$$e_{ij}^{n+1} = \phi_{ij}^{n+1} - \int_{(j-1)\Delta y}^{j\Delta y} \int_{(i-1)\Delta x}^{i\Delta x} \phi(t_{n+1}, x, y)\ \mathrm{d}x\mathrm{d}y\ . \tag{2.45}$$

If the limiter is not being applied then we can also write the right hand side in terms of the values at a previous time and the fluxes across each edge of the cell, as before,

$$e_{ij}^{n+1} = \left(\phi_{ij}^{n} + \sum_{\text{edges}} \hat{\phi}^{n}\right) - \left(\int_{(j-1)\Delta y}^{j\Delta y} \int_{(i-1)\Delta x}^{i\Delta x} \phi(t_n, x, y)\ \mathrm{d}x\mathrm{d}y\ + \sum_{\text{edges}} \mathrm{FLUX(t_n)}\right)\ . \tag{2.46}$$

Or equivalently,

$$e_{ij}^{n+1} = e_{ij}^{n} + \sum_{\text{edges}} \left(\int\int_{\mathrm{area_n}} P(x,y)\ \mathrm{d}x\mathrm{d}y - \int\int_{\mathrm{area(t_n)}} \phi(t_n, x, y)\ \mathrm{d}x\mathrm{d}y\right)\ . \tag{2.47}$$

If we adopt the same local origin as when we fit the polynomial then the integrals in the exact case are given by

$$\int\!\!\int_{\mathrm{area(t_n)}} \mathrm{d}y\mathrm{d}x \equiv \int_{-\Delta x}^{0} \int_{x-\int_{n\Delta t}^{(n+1)\Delta t} u(t,x,0)\ \mathrm{d}t\ +0-\int_{n\Delta t}^{(n+1)\Delta t} v(t,x,0)\ \mathrm{d}t}^{0} \mathrm{d}y\mathrm{d}x\ , \tag{2.48$_x$}$$

and
$$\int\!\!\int_{\mathrm{area(t_n)}} \mathrm{d}x\mathrm{d}y \equiv \int_{-\Delta y}^{0} \int_{0-\int_{n\Delta t}^{(n+1)\Delta t} u(t,0,y)\ \mathrm{d}t\ +y-\int_{n\Delta t}^{(n+1)\Delta t} v(t,0,y)\ \mathrm{d}t}^{0} \mathrm{d}x\mathrm{d}y\ , \tag{2.48$_y$}$$

and for the approximate case by

$$\int\!\!\int_{\mathrm{area_n}} \mathrm{d}y\mathrm{d}x \equiv \int_{-\Delta x}^{0} \int_{x-\int_{n\Delta t}^{(n+1)\Delta t} u_{ij}^{n}\ \mathrm{d}t\ +0-\int_{n\Delta t}^{(n+1)\Delta t} v_{ij}^{n}\ \mathrm{d}t}^{0} \mathrm{d}y\mathrm{d}x = u_0\Delta t\Delta x\ , \tag{2.49$_x$}$$

and
$$\int\!\!\int_{\mathrm{area_n}} \mathrm{d}x\mathrm{d}y \equiv \int_{-\Delta y}^{0} \int_{0-\int_{n\Delta t}^{(n+1)\Delta t} v_{ij}^{n}\ \mathrm{d}t\ +y-\int_{n\Delta t}^{(n+1)\Delta t} v_{ij}^{n}\ \mathrm{d}t}^{0} \mathrm{d}x\mathrm{d}y = u_0\Delta t\Delta y\ , \tag{2.49$_y$}$$

where $u_{ij}^n = u(n\Delta t, i\Delta x, j\Delta y)$ and $v_{ij}^n = v(n\Delta t, i\Delta x, j\Delta y)$. Then, if the wind is uniform and constant ($u(t,x,y) = u_0$ & $v(t,x,y) = v_0$, over this time period),

$$\begin{aligned} e_{ij}^{n+1} =& e_{ij}^{n} + \sum_{x-\mathrm{edges}} \int_{-\Delta x}^{0} \int_{x-u_0\Delta t-v_0\Delta t}^{0} p(x,y_i) - \phi(t_n,x,y_i)\ \mathrm{d}t\ \mathrm{d}x \\ &+ \sum_{y-\mathrm{edges}} \int_{-\Delta y}^{0} \int_{y-u_0\Delta t-v_0\Delta t}^{0} p(x_i,y) - \phi(t_n,x_i,y)\ \mathrm{d}t\ \mathrm{d}y\ . \end{aligned} \tag{2.50}$$

This shows that the error in one cell again depends on the accuracy of the polynomial interpolations for the regions swept across each edge of the cell at that time step. We can again argue, using the Mean Value Theorem, that fitting polynomials using the integral over an area has the same order of accuracy as fitting a polynomial through a particular point in each cell. The result of this is that a scheme using an N^{th} order interpolating polynomial is $N+1^{\mathrm{th}}$ order accurate in space. If $u(t,0) \neq u_0$

then there will also be a contribution to the error from the approximation of the area swept across the edge. If the limiter is applied to the fluxes then the above argument will not hold when the limiter changes the value of a flux.

Summing over each cell to find a bound on the global error gives,

$$\left|e^{n+1}\right| \leq |e^n| + 2\sum_i \int_{i\Delta x-u_0\Delta t}^{i\Delta x} |p_i(x) - \phi(t_n,x)| \; \mathrm{d}x\mathrm{d}y \ , \qquad (2.51)$$

where i runs over all edges on the grid and $p_i(x)$ is the polynomial fitted around edge i. This error is also $N+1^{\text{th}}$ order accurate when the limiter is not applied to the scheme.

2.4 Test Cases

Three test cases were used to test the schemes produced by this method on a square grid; a uniform flow, a solid body rotational flow and a strong deformational flow. These three cases are frequently used, though not always together, in numerical experiments to show the strengths and weaknesses of numerical advection schemes. Here all three have been used in an attempt to establish whether this method works well for a range of flows or if it has problems with some types of flow. Various error measurements were made along with a measure of the cost of computing the results, the CPU runtime. Generally, higher order schemes are more accurate, as are those run on more refined grids. By comparing these results with the computational cost of generating them using this method should show when a higher order scheme is more cost effective than increasing the resolution.

These test cases give a measure of the success of the scheme which can be compared with other schemes run under the same conditions.

The tests in themselves do not give a complete picture of how well the scheme would perform in the atmospheric problem. The uniform flow, Section 2.4.1, has little resemblance to flows found in the real atmosphere but does provide an indication of how well the scheme handles flow at an angle to the grid. The rotational flow, Section 2.4.2, has more in common with rotating systems in the atmosphere but the flow has a zero strain rate and so does not stretch parcels in the same way. The final test, Section 2.4.3 on deformational flow does include vortices which stretch fluid parcels but these are regular and do not interact with one another as in the real atmosphere. Looking at the results of these tests together should give a good idea of the strengths and weaknesses of the scheme and where it will succeed or have problems in the real atmosphere.

For completeness, several measurements were taken of errors and properties of the advected profile. For the first two cases, the error measures used were the L_1 , L_2 and L_∞ norms and first and second moments of mass, along with a breakdown of the L_2 norm into diffusive and dispersive components. The L_1 norm behaves in the same way as the L_2 norm in these cases and so is not shown, neither is the first moment of mass as this is always unity to machine accuracy because of the conservative nature of this method. The norms are defined as

$$\mathrm{L}_i = \sqrt[i]{\frac{\sum |\phi_T - \phi_N|^i}{\sum \phi_T^i}} \ , \tag{2.52}$$

and the moments of mass as

$$M_i = \frac{\sum \phi_N^i}{\sum \phi_T^i} \ , \tag{2.53}$$

where ϕ_T is the true solution, ϕ_N is the numerical solution and the sums are performed over all cells in the domain.

Other measures taken were of the final maximum and minimum values of the distribution and of how well the scheme maintained the symmetry of solutions (where it should). The maximum and minimum values were used primarily to check that the limiter is working and not allowing the values of the extrema to grow, as was the case in all runs. For symmetric initial conditions in a symmetric flow the solution should also be symmetric, and is to machine accuracy for these test cases. A qualitative account of the results for the third test case is presented here. An analytic solution does exist, but it is complicated to calculate and would add little to the qualitative result.

In all the cases that follow the errors for schemes using 8^{th} and 9^{th} order polynomials on higher resolution grids are worse than those for most lower order schemes. This is because of the ill-conditioning of the matrix B which does not allow accurate numerical inversion and subsequent evaluation of coefficients and fluxes. This may generate schemes that are unstable, but where blow up has not occured the results that have been obtained are shown. There are methods by which the conditioning of these matrices may be further improved, but this is not done here as the results for schemes lower than 8^{th} order provide sufficient information in these cases.

2.4.1 Uniform Flow

For this test case, an initial profile was advected by a constant uniform flow at an angle of 45° to a square grid ($\Delta x = \Delta y$) over a unit square with periodic boundaries. Four grids were used, each double the resolution of the previous one; 20×20 ($\Delta x = 0.05$), 40×40 ($\Delta x = 0.025$), 80×80 ($\Delta x = 0.0125$) and 160×160 ($\Delta x = 0.00625$). The flow speed was one unit per second in each coordinate direction (magnitude $\sqrt{2}$ from southwest to northeast) and a Courant number of one quarter ($\alpha_x = \alpha_y = 0.25$) was used in each direction. This meant that each doubling of the resolution requires twice the number of time steps to be used. The profile is advected once around the periodic domain in one second, requiring 80, 160, 320 or 640 time steps depending on the grid size.

Two initial profiles were used for this test case, a square step and a smooth $\cos^2$ function. The tracer value for the square step initial conditions was one in the region $0.25 < x, y \leq 0.75$ and zero elsewhere. The smooth function was defined as $\cos^2(2\pi R)$ for $R < 0.25$ and zero elsewhere, where R is the distance from the centre of the domain. The step initial conditions test how well the schemes can maintain a sharp gradient in a smooth velocity field and should show how the limiter affects the accuracy in preventing spurious oscillations in the solution. The smooth initial condition tests how well the schemes are able to maintain a localised extreme value.

The L_2 and L_∞ errors are used to show the relative accuracy of the schemes. The former gives a measure of the accuracy over the whole domain whilst the latter gives the largest error in any cell. The L_2 error can also be divided into diffusion and dispersion parts, which give an indication of amount of diffusion and oscillation in the schemes.

Figure 2.7 shows the final time results for a selection of schemes that were generated and applied to the step initial conditions. Using a second order polynomial without the flux limiter, on an 80×80 grid gives the reference solution shown in plot a with which we can compare the other schemes. This behaves as would be expected, with a slight smoothing of the sharp gradient around the edge of the step region and the generation of spurious oscillations. If the same scheme is used but with the flux

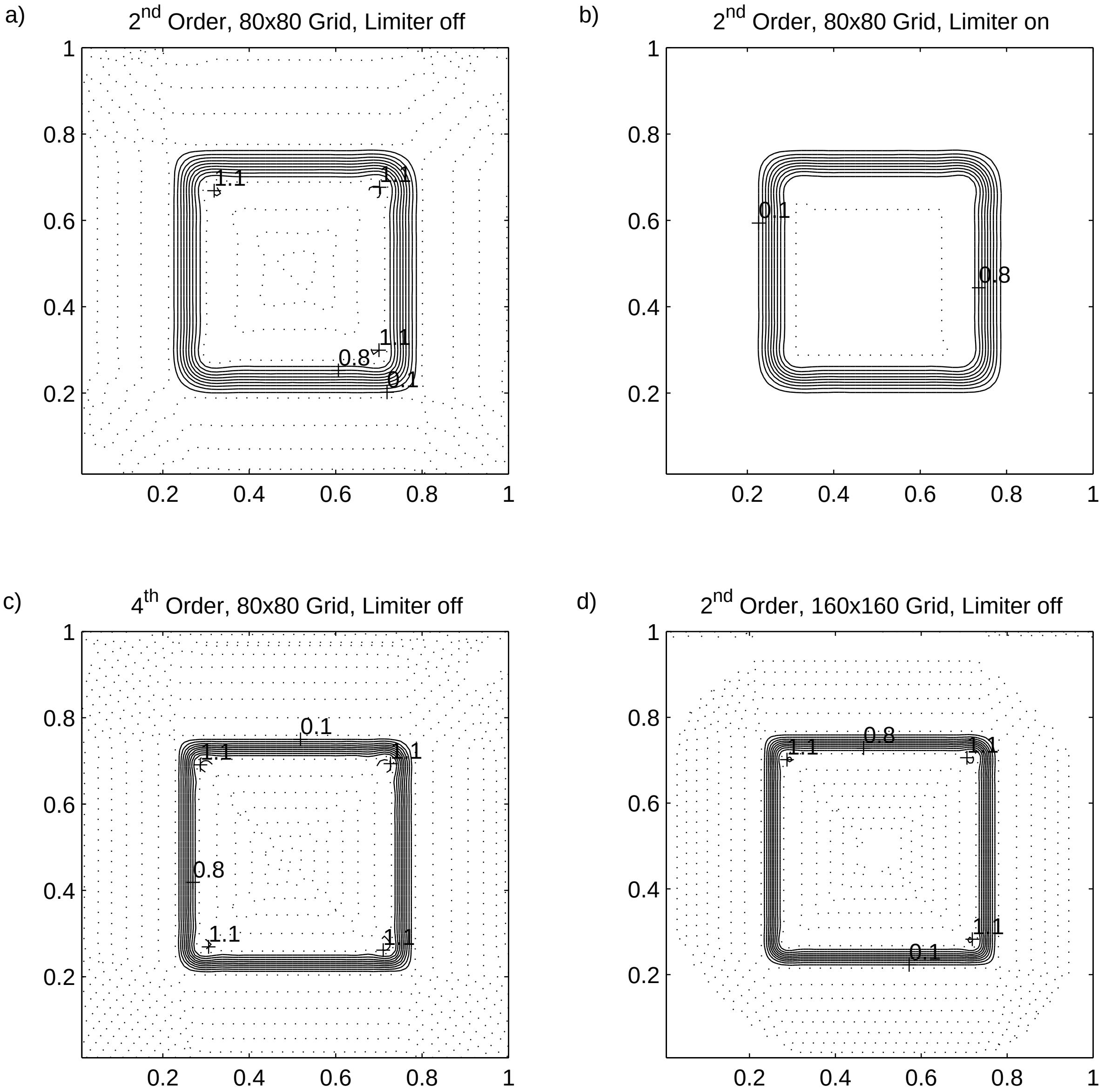


Figure 2.7: Numerical solutions of the advection of a square step once around a periodic domain. Grid sizes, the order of the polynomial used and the state of the limiter are shown above each plot. Contours at 0.0 and 1.0 are dotted lines

limiter applied, then these oscillations are removed, as seen in plot b. The downside of this is that there is a little more smoothing of the sharp edges.

If a fourth order polynomial is used instead of the second order one, then we get the result shown in plot c. This solution contains more oscillations but they are smaller in both magnitude and wavelength, so large gradients are better maintained. If a higher resolution grid is used instead of a higher order polynomial then the results are very similar, as seen by comparing plots c and d. The main differences are that the oscillations are a little smaller on the finer grid and that this result took more than twice as long to compute (as will be seen in plot a of Figure 2.8)

The final solutions of the same schemes applied to the smooth initial condition are not shown but do not exhibit any significant differences in the behaviour of the schemes. Small oscillations are still generated around the non-zero part of the initial profile, though they are smaller in magnitude. The main difference in results between the two profiles is the maintenance of the peak value in the smooth initial profile. This difference is best seen in the later comparison between the L_∞ errors of the different schemes.

Figure 2.8 shows the L_2 and L_∞ norms plotted against the computational cost for both profiles, with and without the use of the flux limiter. This plot and ones like it will be used widely throughout the next few chapters so it is important to take some time to understand what they show. Each symbol on the plot represents one of the error measures for one of the schemes, plotted according to the error value and the computational cost (CPU runtime) taken to calculate the result.

The symbols are joined by lines according to the resolution of the grid

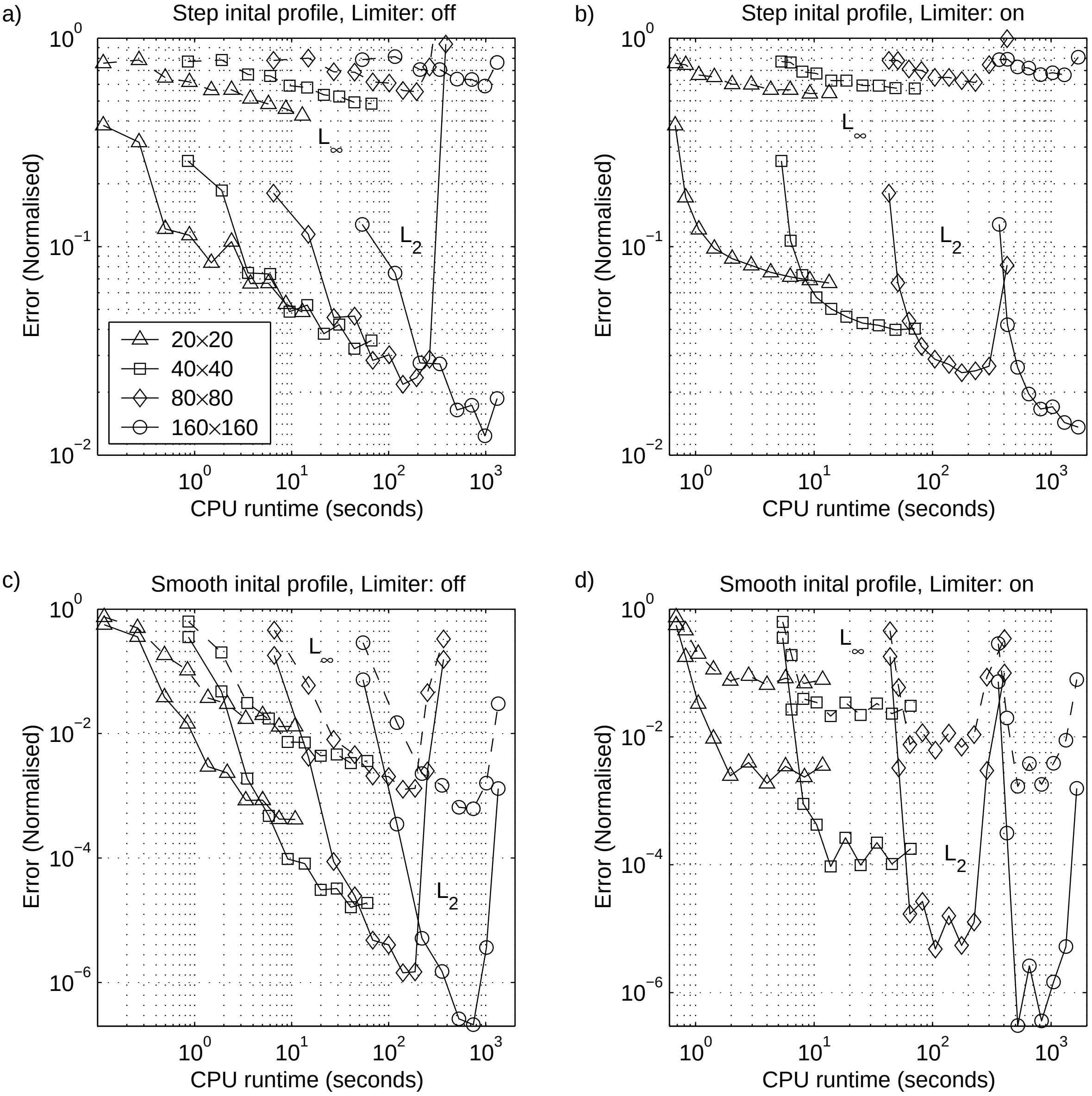


Figure 2.8: L_2 and L_∞ error norms for step profile (top) and smooth profile (bottom) advected by uniform constant flow. Grid sizes are as shown and each order scheme is marked by a symbol.

on which the scheme was used and the type of error being measured. Solid lines represent either the L_2 error or the error due to diffusion (explained later in this section), the dashed lines represent the L_∞ error or the dispersion error. The different symbols that are used represent the different resolutions of the grids, the exact resolutions are shown with each set of plots. On a particular resolution grid the computational cost increases as higher order schemes are used so the first (fastest) symbol on a line represents the zeroth order scheme. The second symbol represents the first order scheme and so on.

The trend in the lines representing each resolution shows how the accuracy (error) varies with the effort (CPU runtime) needed to generate the result as higher order schemes are used. Comparing the lines for the different resolution grids shows how the accuracy varies with the effort required to generate the results when the resolution is also allowed to vary. If all the lines for one error measure overlap then the effort required to generate a result of a given accuracy is the same regardless of whether the scheme order or grid resolution is changed. If the lines do not overlap then either increasing the resolution or the order of the schemes will be more efficient, the graph will tell us which.

We can look at this relationship in two ways, if we wish to achieve a particular accuracy then we can look along the graph from the y-axis to see the most efficient scheme/grid combination that gives the desired accuracy. Alternatively we can consider having a specific amount of computing time in which we wish to find the most accurate solution possible. In this case we can look up along the x-axis for that particular time and find the scheme and grid that will give the most accurate result.

In this particular case we can make several observations about the behaviour of the different order schemes on different resolution grids. Applying the schemes without the flux limiter to the step profile (plot a) shows a general reduction in these error norms as higher order schemes are used. The L_∞ error, in this case, gives an indication of the phase error and how well the sharp gradient is captured. There is an increase in the L_∞ error when increasing the polynomial order from zero to one, caused by the introduction of oscillations into the solution. These oscillations generally decrease in magnitude and increase in frequency as higher order polynomials are used for interpolation. Schemes using even order polynomials generate fewer oscillations than those using odd order polynomials.

When the flux limiter is applied (plot b), these oscillations are removed and the L_∞ error is improved for schemes using low order polynomials. Where higher order polynomials are used the oscillations prevented are small and the lessening of the sharp gradient caused by the limiter dominates this error. The L_∞ error also increases as the resolution is increased, suggesting that the phase error is larger relative to the grid size on the more refined grids. This is most likely caused by the need for more time steps to be taken on these grids, the phase error per time step is smaller on these grids.

The L_2 error gives an indication of the combined error over the whole of the domain. Increasing the order of the polynomial used from even to odd order increases the number and magnitude of oscillations but better captures the steep gradient. These factors counteract one another in terms of the L_2 error to give relatively small improvement in accuracy between even order schemes and those of order one higher. Increasing from an odd to an even order polynomial reduces the magnitude of these oscillations and improves the phase speed, which is shown by the larger improvement in accuracy in plot a. Applying the flux limiter removes these oscillations and improves the accuracy, in the L_2 norm, for low order schemes. For the higher order schemes, these oscillations are smaller and errors due to the poorer representation of the sharp gradient begin to dominate. The effect of this is that the errors appear to be approaching a limit as the order of the polynomial is increased. These properties of the different order schemes appear to be the same here as in the one-dimensional case.

The results for the smooth initial profile, plots c and d, display slightly different behaviour, due to the absence of a sharp gradient in the initial profile. The errors are much smaller than the previous case, which is a result of the smooth profile being more amenable to accurate interpolation by polynomials. The L_∞ error for this case is effectively a measure of the change in the peak value of the solution and this improves with increasing order and resolution. The L_2 error also shows a continued reduction as higher order polynomials and higher resolutions are used. The relationships between even and odd order schemes observed in the previous case are again present in this one.

When the limiter is applied, the effect is similar to the previous case, in that there is a slight improvement in the errors for some of the lower order schemes and a worsening for the higher order schemes. The main difference in this case is that there are oscillations between the errors of high-even and high-odd order schemes. High-even order schemes appear to be approaching a limit form below whilst high-odd order schemes would seem to be approaching the same limit from above. One possible mechanism for this is the difference between the number and size of oscillations in the even and odd order schemes. The limiter will alter the fluxes more where many, large oscillations are being generated and this will have a greater effect on the accuracy.

The x-axis on these plots indicates the computational cost (CPU runtime) for generating and running the schemes. This allows us to make a fairer comparison between the schemes as it shows the computational effort required to reach the accuracy achieved by each scheme. In the un-limited cases, plots a and c show that a given accuracy in the L_2 error can be reached by either increasing the order of the interpolating polynomial or the resolution. In these cases however, the CPU runtime required to gain a given level of accuracy is roughly the same, whichever method of improvement is used. For example, if we want the L_2 error to be less than 0.08 in plot a of Figure 2.8 we can achieve this in around three and a half CPU seconds using either a sixth order scheme on a 20×20 grid or a second order scheme on a 40×40 grid. Alternatively, if we want the most accurate solution possible in forty CPU seconds for the smooth initial profile, an eighth order scheme on a 40×40 grid gives a similar L_2 error to a third order scheme on an 80×80 grid.

When the limiter is used all the schemes become more expensive to run and the errors are generally worse than in the unlimited cases. This is the trade off that is made for ensuring that the numerical solution maintains the shape preservation property of the true solution. The limiter's greater effect on the accuracy for higher order schemes means that it becomes more cost effective to increase the resolution than to increase the polynomial order beyond a certain level. In the smooth case, this level is that of the fourth order polynomial. For the step case this level may be a little higher though it is difficult to give an exact figure without results for resolutions between those shown in plot b.

In order to try and better understand why the errors behave in this way, we can look at the diffusive and dispersive components of the L_2 norm. The square of the L_2 error measure may be written in terms of diffusion and dispersion errors, derived through writing the L_2 error in terms of mean and variance. Following the argument of Takacs [49],

$$L_2^2 = \frac{\sum(\phi_T - \phi_N)^2}{\sum \phi_T^2} = \frac{1}{\sum \phi_T^2}\left(\sigma^2(\phi_T - \phi_N) + (\bar{\phi}_T - \bar{\phi}_N)^2\right) \ , \tag{2.54}$$

where ϕ_T is the true solution, ϕ_N is the numerical solution. By writing the field ϕ in terms of its mean, $\bar{\phi}$, and variance, $\sigma(\phi)$, it can be shown that,

$$L_2^2 = \frac{1}{\sum \phi_T^2}\left((\sigma(\phi_T) - \sigma(\phi_N))^2 + (\bar{\phi}_T - \bar{\phi}_N)^2 + 2(1-\rho)\sigma(\phi_T)\sigma(\phi_N)\right) \ , \tag{2.55}$$

where ρ is a coefficient depending on the correlation between the numerical and exact results. Due to the conservative nature of this method, $\bar{\phi}_T - \bar{\phi}_N = 0$. If there is no diffusion then $\sigma(\phi_T) - \sigma(\phi_N) = 0$, leaving the dispersion error as

$$L_{disp} = \frac{1}{\sum \phi_T} 2(1-\rho)\sigma(\phi_T)\sigma(\phi_N) \ . \tag{2.56}$$

So the diffusion error is

$$L_{diff} = \frac{1}{\sum \phi_T}\left(\sigma(\phi_T) - \sigma(\phi_N)\right)^2 \ . \tag{2.57}$$

Figure 2.9 shows these component parts of the square of the L_2 norm for both profiles. For the step profile, the diffusion error is several orders of magnitude smaller than the dispersion error for both the limited and un-limited schemes. The previously observed behaviour of the schemes being thus shown to come from the error due to dispersion. The differences between using even and odd order polynomials can be seen more

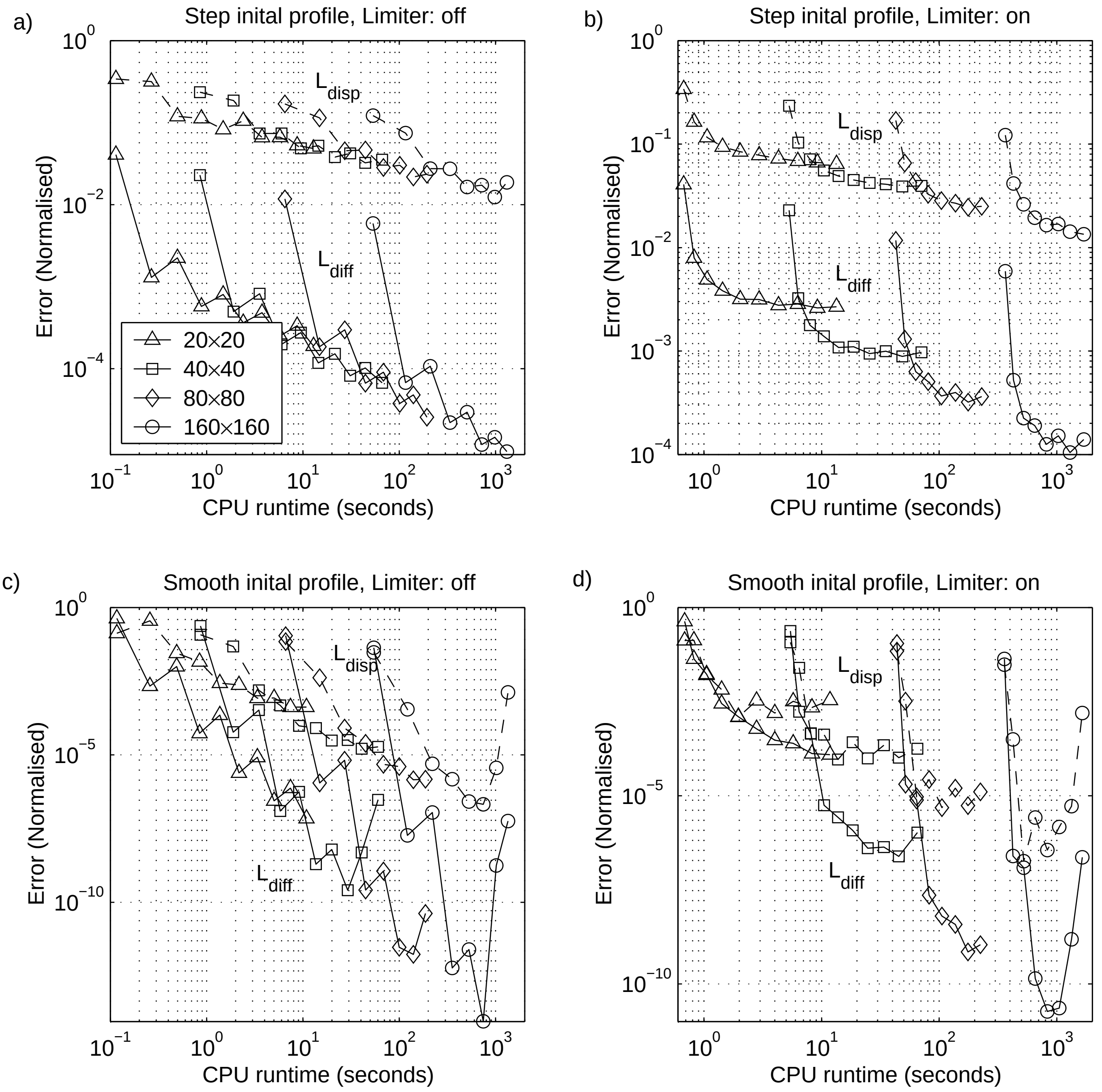


Figure 2.9: L_{diff} and L_{disp} error norms for step profile (top) and smooth profile (bottom) advected by uniform constant flow. Grid sizes are as shown.

clearly from these plots. Increasing from an odd to even order polynomial introduces more diffusion but reduces the dispersion. Increasing from an even to an odd order polynomial has little effect on the dispersion but does reduce the amount of diffusion. In the case of the smooth profile there is a smaller difference between the magnitude of the diffusion and dispersion parts of the error when using low order polynomials. The diffusion error decreases more rapidly than the dispersion part and only becomes relatively small when the polynomial order is increased beyond two (four for the coarsest grid).

These results show that increasing the order of the polynomial will reduce the diffusion by more than it will reduce the dispersion but that the error will continue to improve. This improvement will continue until the matrix inversion part of the algorithm can no longer be accurately computed. The effects of this can be seen for the high order polynomials on the finest grids where the errors begin to increase significantly as the order is increased.

2.4.2 Rotational Flow

For this test case, a counter clockwise solid body rotation was applied to the initial profile of a split cylinder. The rotation speed was 2π times the radius from the centre of a unit domain, in a direction perpendicular to the radius. The boundaries were treated as periodic in the code but due to the method of averaging the flow speed there is no flow across the boundaries. This does not have any effect on the final solution, which is the same as the initial condition, or on the limited schemes, where the tracer does not reach the boundaries. Where the small oscillations in the un-limited schemes reach the boundary the flow is not a true solid body rotation but this has little effect on the quality of the solution.

The initial condition is a cylinder of radius 0.15 and height 1 with a 'slot' of width 0.05 removed leaving a 'bridge' of maximum width 0.05. The same four grids were used as for the previous test case with a maximum Courant number of $\frac{\pi}{8}$ in each direction so one rotation requires $8N$ time-steps, where N is the number of cells in one coordinate direction. This case tests how well the schemes cope with flows at all angles to the grid and with small scale features such as the 'slot' and the 'bridge'.

Before examining the numerical results of these tests it is useful to consider the different approximations this test requires. Firstly, the flow is not uniform across any cell, so an approximation of the flow must be made. To obtain these results, the parallel and perpendicular components of the flow at each edge have been prescribed as their analytic values at the mid-point of that edge. The area swept across each edge is thus approximated as the area swept across the edge by a uniform flow in the region around the edge. This breaches the condition that $u(t, x, y) = u_0$ & $v(t, x, y) = v_0$ in our error analysis, introducing a further error that may either increase or reduce the overall error. It is also important to note that the initial conditions on the discrete grid do not give an exact representation of the stated initial conditions. For the calculation of the error measures the 'true' solution used is that of the discrete representation of the initial condition. This means that they do not take into account any errors from discretising the initial condition.

Figure 2.10 shows some examples of the final numerical solutions for this test case. Plot a shows the result of using a second order polynomial on an 80×80 grid without the flux limiter. This exhibits the same features as the square step in the previous test case, smoothing the sharp gradients

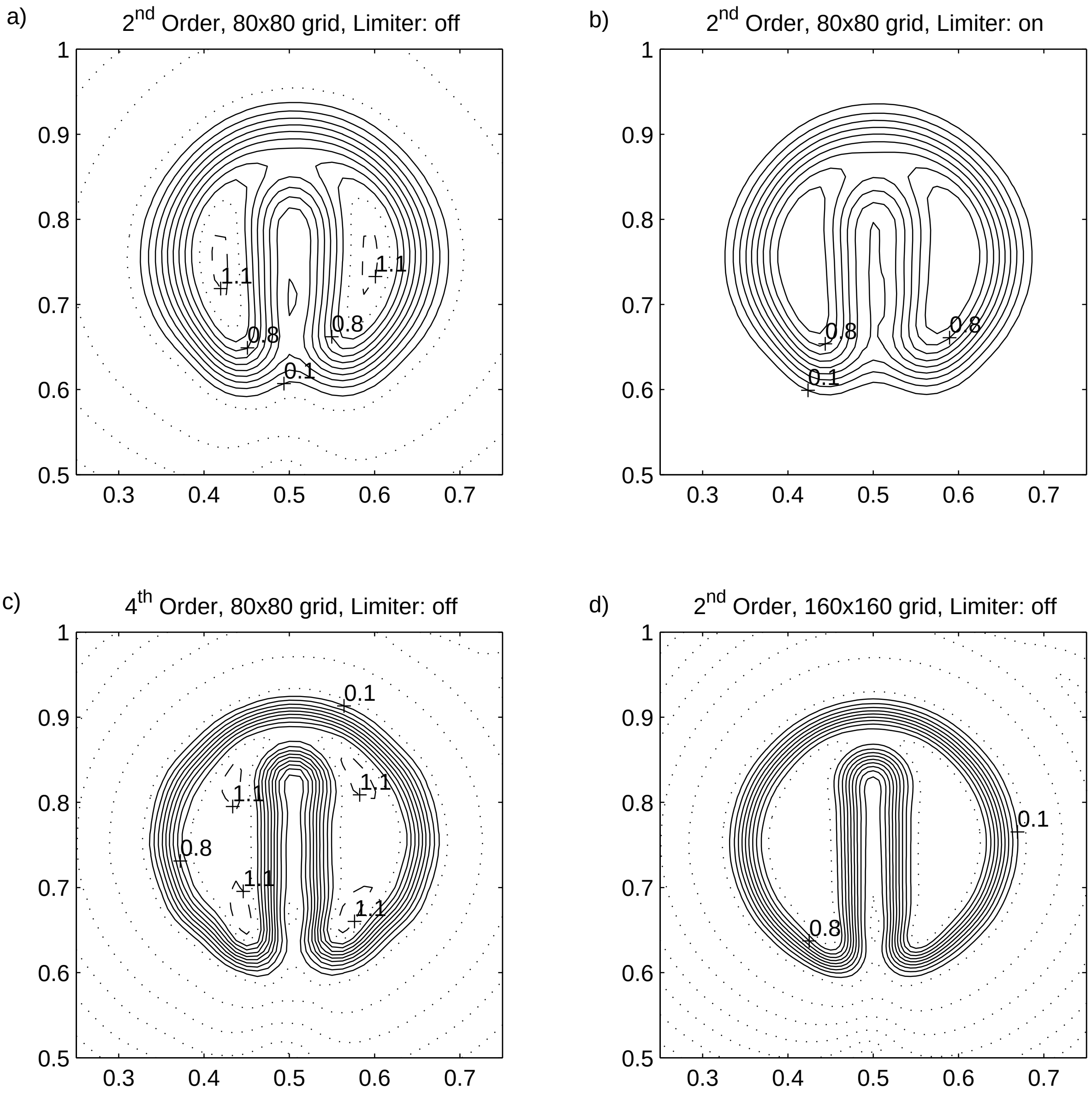


Figure 2.10: Numerical solutions of the advection of a split cylinder by a solid body rotational flow. Grid sizes, the order of the polynomial used and the state of the limiter are shown above each plot. Contours at 0.0 and 1.0 are shown as dotted contours. Only the region around the cylinder is shown.

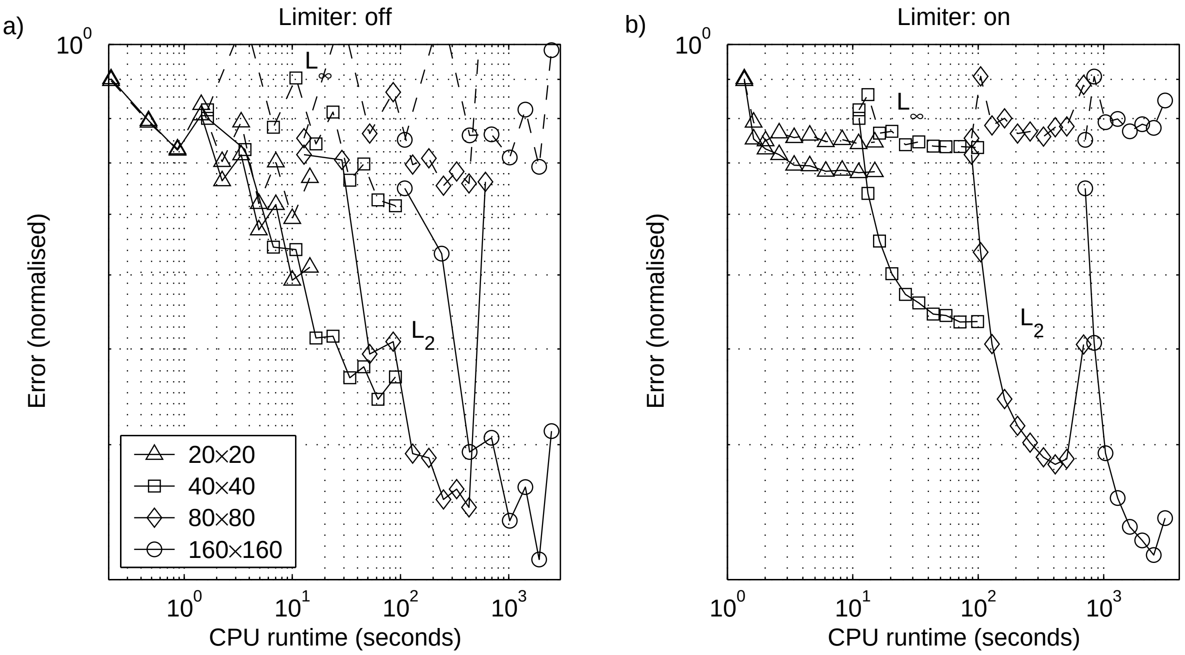


Figure 2.11: L_2 and L_∞ error norms for a split cylinder advected by a solid body rotational flow. Grid sizes are as shown.

and introducing unphysical oscillations in the solution. The result of this is that there is a significant filling of the slot and erosion of the bridge. By limiting the fluxes for this case, as shown in plot b, the oscillations are prevented but the result of lessening the sharp gradients is that there is more filling of the slot.

If we use a fourth order polynomial instead of switching on the limiter, as for plot c, then the oscillations are reduced in magnitude but increase in frequency. The sharp gradients are better maintained so there is little erosion of the bridge or filling of the slot. If the resolution of the grid is doubled instead of increasing the order of the polynomial, then similar results are achieved, though at a higher computational cost. By comparing the smoothness of the contours in plots c and d we can also see that the more refined grid better captures the shape of the initial condition.

Figure 2.11 shows the L_2 and L_∞ error measures for the different grid resolutions and orders. In both the un-limited and limited cases, the er-

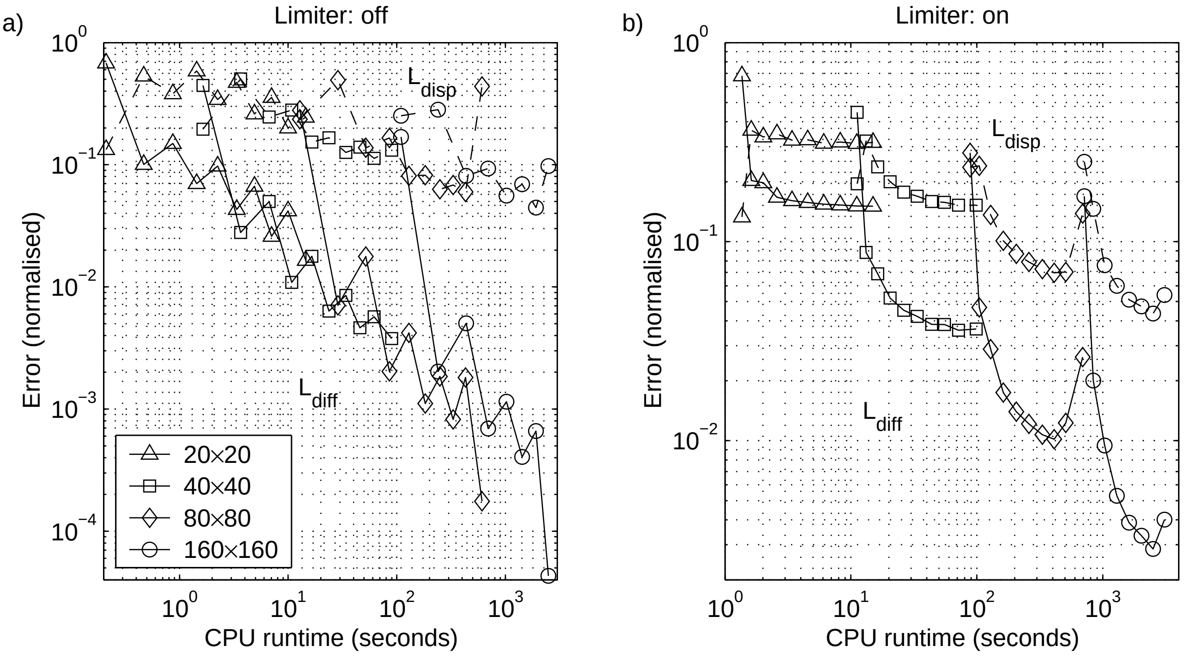


Figure 2.12: $L_{\rm diff}$ and $L_{\rm disp}$ error norms for a split cylinder advected by a solid body rotational flow. Grid sizes are as shown.

rors vary in a similar way as for the previous square step test. There is a general reduction in the error as higher order polynomials are used, though the overall error values are higher than in the previous case. The higher error values are caused by the added approximation of the area being swept across each edge, which is most noticeable in the L_∞ error. For the un-limited case, plot a, the relationship between even and odd order schemes is the same as before but is more marked, with small increases in the error when increasing from even to odd order polynomials. Including the limiter, shown in plot b, removes not only the oscillations in the numerical solutions but also these variations in the L_2 error. The added approximation of the region swept across the edge has little effect on the trends observed in the uniform flow test cases but it does raise the values of the error measures.

The same comparison as before can be made between the diffusive and dispersive parts of the square of the L_2 error and is shown in Figure 2.12.

The diffusion error is much smaller than the dispersion error for all but the first order upwind scheme. The dispersion error is also improved by the limiter removing oscillations, for schemes using odd order polynomials. More diffusion is introduced by the limiter but this is still significantly smaller than the dispersion part of the error. These results are again very similar in form to the previous case, with higher error values due to the added approximation of the area being swept across each edge.

These results combine to show that, as in the previous tests, increasing the order of the polynomial used for interpolation improves the accuracy of the schemes. This improvement in accuracy comes for a smaller computational cost than doubling the resolution for polynomials of up to order eight without the use of the limiter.

2.4.3 Deformational Flow

This case is used by used by Smolarkiewicz to test the multi-dimensional Crowley advection scheme [41] and is discussed further by Staniforth et.al. [42, 44]. The flow field for this case is a set of 16 counter rotating vortices defined by the streamfunction,

$$\psi(x,y) = A\sin(kx)\cos(ky)\,, \tag{2.58}$$

where $A = 8$, $k = 4\pi/L$ and the origin is at the bottom left corner. L is the length of the domain, in this case 100 units. The initial tracer profile is a cone of height one and radius 15 units, centred on the centre of the domain, as shown in Figure 2.13. A maximum Courant number of 0.7 is used and the test is run for the same length of time as in Smolarkiewicz [42], $T = 2637.6$ seconds. On an 80×80 grid, 3021 time steps of approximately 0.87 seconds duration are thus used. Since the flow is

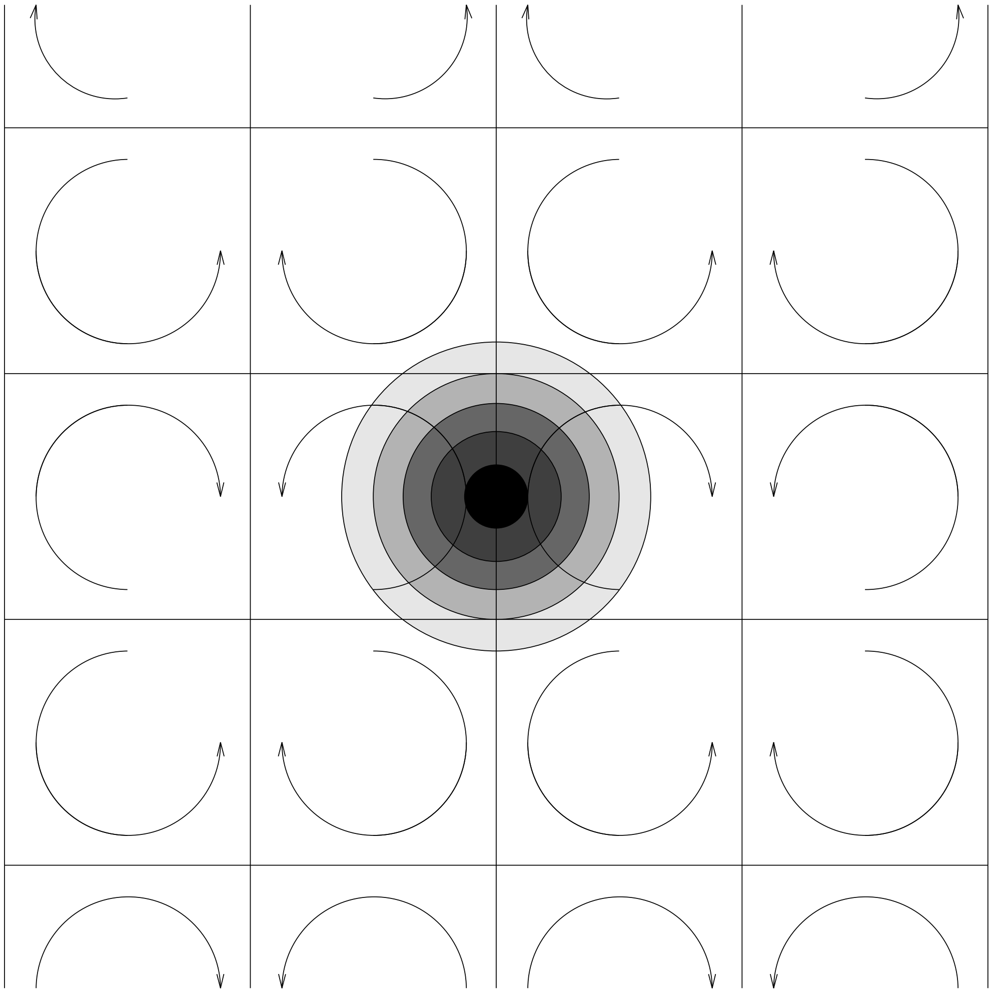


Figure 2.13: Initial conditions (shading) and flow directions (arrows) for deformational flow test case.

constrained to move along streamlines, each vortex is self contained and the solution for all but the six central vortices will be zero for all time. Provided we align our numerical grid with the boundaries of the vortices and apply the flux limiter this property will hold for our numerical solution.

Staniforth et. al. [44] gave an exact solution to this test case and also arguments as to the best way to interpret the results. The vortices act to pull out the flow into thin streamers which are wrapped around the centre of the vortex by the rotation of the flow. We can only hope to approximate the true solution over short time periods since we cannot accurately resolve a pattern of streamers that have widths close to the grid size. Over longer time periods we can only hope to approximate the large scale features of the flow, since a direct comparison between the true and numerical solutions would not be very informative. Results up to time $T/50$ are used here for inter-comparison and a direct comparison with the true solution, figure 3 from [44].

Results for times from $T/50$ up to T are used only to show whether the numerical results represent the large scale features that are present in the true solution. There are three particular features that we would hope to see, the first being that the solution remains zero in all but the six central cells. The next is that the fluid parcels in the four 'outer vortices' remain close to the edge of their vortex and the solution remains zero in the centre of the vortex. The final feature is that the two central vortices each resemble an 'eggcup' shape at time T.

Figure 2.14 shows some results for three limited schemes for a short time after the test is started. Each of these cases exhibits the features we would expect to see. The initial profile is separated into six separate

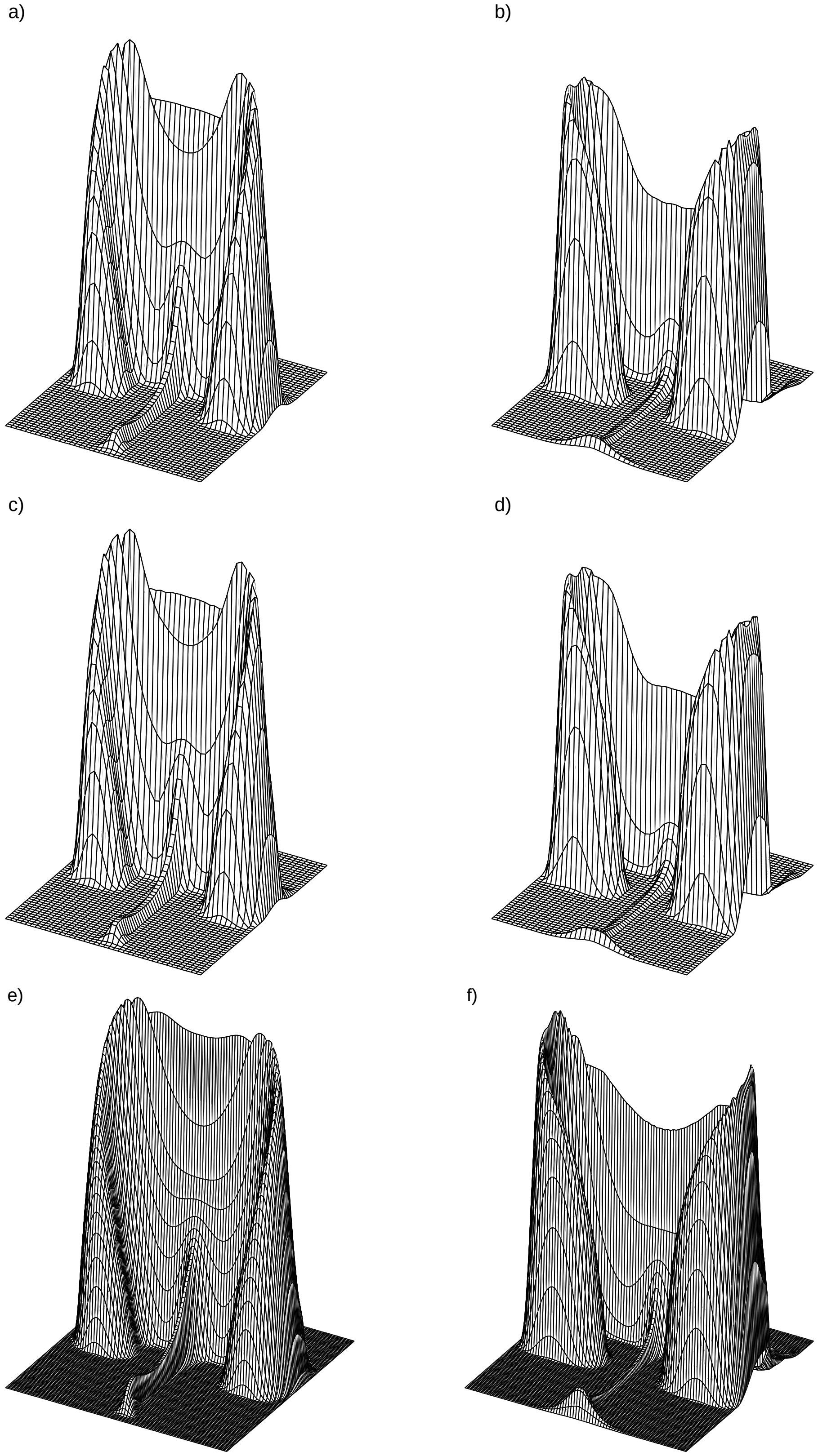


Figure 2.14: Short time results of the deformational flow test case. The plots are arranged as follows: a-b 80×80 grid, second order. c-d 80×80 grid, fourth order. e-f 160×160 grid, second order. a,c,e: $t = \frac{T}{100}$. b,d,f: $t = \frac{3T}{200}$. All schemes have been used with the flux limiter and the x, y and z-axes have the same scale in each plot.

vortices and stretched into filaments which are being wrapped up around the centre of the vortex. The stretching of the filaments at saddle points of the flow (vortex corners) causes new maxima and minima to occur in the solution. The limiter allows these to form in the numerical solution on the boundary of each vortex but also allows smaller oscillations to form along the filaments. These extrema are caused by the same mechanism as those on the boundary but the conditions that create them are due to the discretisation of the wind field. The use of a piecewise constant wind field on a discrete grid case can allow a small amount of unphysical divergence to be generated. The limiter will then allow small oscillations to be generated as they are part of the physical solution of the discrete wind field version of the test case. Despite this problem, the limiter does still ensure that the global maximum and minimum values are not exceeded. The fourth order scheme does a slightly better job of capturing the steep gradients of the filaments than the second order scheme. The higher resolution scheme manages to perform even better, maintaining both steeper gradients and higher peak values, though at a much increased computational cost.

Figure 2.15 shows numerical solutions for the same three schemes as before but after a longer time period. The first column shows the numerical solutions at $t = T/50$, after about one rotation around the vortex. The schemes for the 80×80 grid show little significant difference at this time, only slightly steeper gradients for the fourth order scheme. The scheme on the 160×160 grid shows a small but significant improvement over both lower resolution schemes, again mainly in the capture of the steep gradients.

The middle column of Figure 2.15 shows plots at a time roughly equiv-

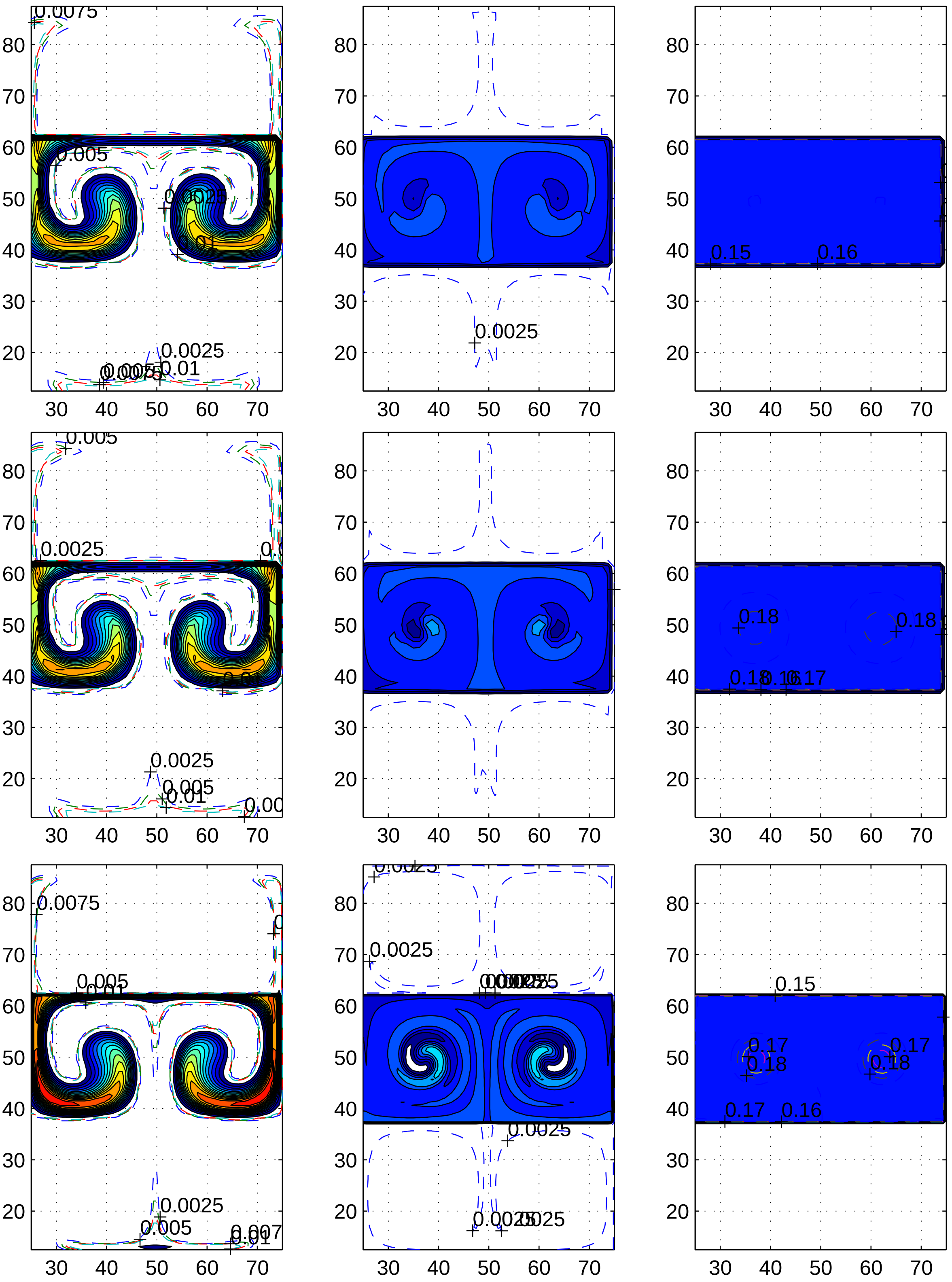


Figure 2.15: Long time results of the deformational flow test case. The plots are arranged as follows: First row 80 × 80 grid, second order. Second row 80 × 80 grid, fourth order. Third row 160 × 160 grid, second order. First column, $t = \frac{T}{50}$. Second column, $t = \frac{T}{10}$. Third column, $t = T$. All schemes have been used with the flux limiter. Filled contours are at intervals of 0.05 and dashed contours as labelled.

alent to five rotations around the vortex. In the four outer vortices, there is little difference between the two schemes on the lower resolution grid. The solution in the central vortices at this time has streamers that are roughly the width of one cell on the 80×80 grid. In the numerical solutions the streamers have merged together and are starting to look like the 'eggcup' suggested by Staniforth et al. [44]. This merging has occured because the numerical schemes have not adequately modelled the narrowness of the streamers, partly due to the accuracy of the schemes but mostly due to aliasing on the grid. There is still a region of high/low contrast rotating around the centre of each vortex, more markedly in the fourth order case than in the second. This contrast is clearer still on the higher resolution grid where these regions also extend further back around the centre of the vortex. The increased resolution also allows the scheme to better capture the smaller features close to the edges of the outer vortices.

The right hand column of Figure 2.15 shows the solutions at the final time, after roughly fifty rotations around the vortex. Each scheme displays the 'eggcup' shape to a greater or lesser extent, at least if the contour interval is carefully chosen as for the second order scheme on the 80×80 grid. The same contour interval shows a clearer 'eggcup' when the fourth order scheme is used and a clearer one still on the higher resolution grid.

These results show that the resolution of the grid can be more important than the order of the numerical scheme used. In this case the test was generating features smaller than the scale of the grid, so a higher resolution grid has a significant advantage in trying to capture these features for longer time runs. The higher order schemes do have a slight advantage over the lower order schemes on a given grid because they are better at maintaining the steep gradients involved for longer and thus resolving the features closer to the grid scale. The one advantage that the lower resolution schemes have is that they are much cheaper to compute. Doubling the resolution means that calculations must be performed for four times as many cells at each time step and that twice the number of time steps need to be taken in order to satisfy any Courant number condition. This means that doubling the resolution causes a roughly eight fold increase in the computational cost of the run. Whether or not this increased computational burden is worth the effort depends on how important it is that the scheme resolves the small scale features.

2.5 Summary

A general method for generating multidimensional advection schemes of improving accuracy on a regular rectangular grid has been developed and implemented in two dimensions. The method works by fitting a polynomial to the discrete tracer field around an edge and then integrating this polynomial over the region swept across that edge to find the flux. The stencil over which the polynomial is fitted is chosen such that there is a unique polynomial fit through the data. A flux limiter can be applied to the fluxes generated in this way if shape preservation is required.

The resulting schemes have been run on a number of test problems with a flux limiter to prevent spurious oscillations and negative values being generated. Numerical results show a large improvement in accuracy for a relatively small cost by increasing the order of the scheme from first to third order and less of an improvement for a greater increase in cost for higher order schemes. For most cases, increasing the order of a scheme beyond eight would be less cost effective (in terms of CPU time) than increasing the resolution of the grid on which the scheme is being run.

Notable exceptions to this rule of thumb are more complicated sets of equations, of which advection makes up only one part. Atmospheric models may not be sensitive to small improvements in the advection scheme, other approximations in the model may introduce more significant errors into the numerical solution. It is also true that increasing the resolution of such models is more computationally expensive than in the cases tested here. The first of these factors suggests that we should increase the resolution rather than the order of the advection scheme, while the second factor suggests the opposite. Whether these factors balance out or one is dominant over the other will be examined when we look at a more complicated shallow water model in Chapter 5.

The results of these schemes compare well, in terms of accuracy, with many commonly used schemes. This is not surprising considering the lower order schemes generated in this way are finite volume versions of known schemes. The scheme based on a zeroth order polynomial is the same as the diffusive first order upwind scheme and the first order scheme turns out to be a two-dimensional generalisation of the Lax-Wendroff scheme. The second order scheme is similar to the UTOPIA scheme of Leonard et al. [23] with higher order schemes being based on the same methodology.

These results show that in this case, and quite possibly others, increasing the order of a scheme gives a larger improvement in the diffusion error than in the dispersive part. This would suggest that at some point simply increasing the order of a scheme will have little discernible effect on the error, which is also shown in these results. Increasing the resolution will cause an improvement in the dispersion error and a lower order scheme may be used for a saving in the cost with little extra error. Finding the exact optimum balance between cost and error for all but the simplest problems would be difficult and would depend on the nature of the flow and our criteria for measuring accuracy. For example, looking at the L_2 error for the uniform test case we could conclude that we could use schemes of up to order eight but for the deformational flow we would want to increase the resolution before using such high order schemes.

The current implementation of this method has not yet been fully optimised for constant flow conditions and is very inefficient for variable flow problems. On a structured grid it is not too difficult to optimise the code for variable flow. Only speed and direction will vary the coefficients needed, so these can be stored when first generated and re-used when flow at another edge or time are near enough identical. On unstructured grids the same idea may be implemented, but with few if any identical edges, storage of all coefficients may become a problem.

The method can be extended to work in three dimensions in a straightforward way. A three dimensional N^{th} order polynomial must be fitted over a stencil which can be selected in the same way as for the two dimensional case. Selecting a row of $N+1$ cells in each coordinate direction followed by different rows of N cells in each direction and so on will generate the three dimensional analogue of the two dimensional stencil. The region swept across the edge can then be found and the integral of the polynomial over the region will give the flux.

The method described in this chapter can be used on rectangular grids but is not applicable to other grids in its current from. If we wish to apply a similar method on a non-rectangular grid then we will need to find a different region over which to fit the interpolating polynomial around an edge. It is this problem that we now consider in the next chapter, along with a change to the way that the polynomial is fitted over the stencil.

Chapter 3

Triangular Grids

3.1 Introduction

In the previous chapter we developed a method for generating finite volume advection schemes on rectangular grids in two dimensions. This method was based on a local polynomial approximation to the advected field, an idea which can be extended for use on other grids. In this chapter we will look at how the method can be applied to both regular and irregular triangular grids in two dimensions. The basic ideas carry over in a straight forward manner, but the application of these ideas introduces some extra problems in this case. These problems are caused by the grid, since we no longer have such a simple method for selecting the stencil over which to fit the polynomial.

The way we selected the stencil on the square grid relied on the regularity of the grid and the alignment of the cells with the coordinate system. Even on a regular triangular grid, the previous method makes little sense as the cell edges are no longer aligned with the coordinate directions. A square has edges in two parallel pairs but a triangle has edges orientated in three directions, which breaks the symmetry our previous idea relied upon. Instead, we must find some other way of selecting a stencil over which to fit the polynomial. There are many ways this could be done but most are ruled out if we do not wish to have to reselect a stencil at each time step. This stipulation rules out any method that uses the wind field as part of the criteria for which cells to use.

If we select compact stencils by adding cells neighbouring those already in the stencil then we may never have a stencil with exactly the same number of cells as terms in our polynomial. If we use the distance of cells from a certain edge or cell then, on a regular grid, we may still end up with two cells that have equal claim to be added to the stencil when we only need to add one more. Since there is no logical way to decide which of these cells to include without resorting to using the wind field, we are left with little choice but to change the way we fit the polynomial over the selected stencil.

Instead of insisting that the integral of the polynomial over a cell is equal to the amount of whatever is being advected in that cell we can insist that the difference between the two, summed over the stencil, minimises a cost function. In order to test whether this is a reasonable approximation to make, we shall return to the one-dimensional problem and examine its effects in that case. We can fit our polynomial over fewer or more cells than there are terms in the polynomial and we can also choose several cost functions. We shall consider these various options after looking at how the method will change.

3.2 One-Dimensional Advection

As in the previous chapter, we begin with the one-dimensional advection equation for a tracer with mixing ratio ϕ by a non-uniform velocity u in a fluid of constant density,

$$\frac{\partial \phi}{\partial t} = -\frac{\partial\,(u\phi)}{\partial x}. \tag{3.1}$$

Integrating this over a time step Δt and a cell in space we get the exact formula for the update equation,

$$\phi_i^{n+1} = \phi_i^n - \left(\hat{\phi}_{\mathcal{R}}^n - \hat{\phi}_{\mathcal{L}}^n\right) \ , \tag{3.2}$$

where ϕ_i^n is again the mixing ratio of tracer ϕ in the i^{th} cell at the n^{th} time step and $\hat{\phi}_{\mathcal{L},\mathcal{R}}^n$ is the flux across the left/right edge of cell i in time Δt. For a fluid of uniform, constant density ϕ can also be thought of as an average concentration. Discretising this equation as before gives

$$\phi_i^{n+1} = \phi_i^n + \left(\mathbf{c}_{\mathcal{L}} \cdot \boldsymbol{\phi}_{\mathcal{L}}^n - \mathbf{c}_{\mathcal{R}} \cdot \boldsymbol{\phi}_{\mathcal{R}}^n\right) \ , \tag{3.3}$$

where the $\boldsymbol{\phi}_{\mathcal{L},\ \mathcal{R}}^n$'s are vectors of the mixing ratio of the tracer in the cells surrounding the left/right edges and the $\mathbf{c}$'s are the coefficients that our method will calculate. In order to calculate the $\mathbf{c}$'s when we do not have the same number of cells in our stencil as there are terms in the polynomial we shall insist that the residual, r, is minimised where

$$r^2 = \sum_{\text{stencil}} \left(\Delta x_i \phi_i^n - \int_{cell\ i} p_N(x)\ \mathrm{d}x\right)^2 \ . \tag{3.4}$$

This minimises, in a least squares sense, the difference between the previously predicted amount of tracer in each cell and the amount approximated by the fitted polynomial.

In the previous chapter we forced $r = 0$ by insisting that the tracer amount approximated by the polynomial was equal to the predicted amount in each cell in the stencil, or

$$\Delta x_i \phi_i^n = \int_{cell\ i} p_N(x)\ \mathrm{d}x. \tag{3.5}$$

We rewrote this as

$$\Delta x_i \phi_i^n = \sum_{j=0}^{N} \int_{S_i}^{S_i+\Delta x_i} a_j x^j \, \mathrm{d}x \ , \tag{3.6}$$

for each cell and ultimately as the matrix equation

$$\mathsf{Ba} = \boldsymbol{\phi} \ . \tag{3.7}$$

If we repeat this process for this case then we again arrive at equation like (3.7) but with the important difference that B is no longer square. If there are fewer cells in our stencil than terms in our polynomial then B will have fewer rows than columns and the system will have an infinite number of solutions, i.e. $r^2 = 0$ for an infinite number of N^{th} order polynomials. If there are more cells in the stencil than terms in the polynomial there will be more rows in B than columns and there will in general be no solutions to equation (3.7), i.e. $r^2 \neq 0$ for any N^{th} order polynomial but will be minimised for one N^{th} order polynomial.

The value of $\mathbf{a}$ that minimise the difference between Ba and $\boldsymbol{\phi}$, in the least squares sense, is found by minimising

$$r = \|\boldsymbol{\phi} - \mathsf{Ba}\|_2 \ . \tag{3.8}$$

This is a discrete form of equation (3.4) for which the values of r are equivalent. The solution to the minimisation problem can be found by pre-multiplying $\boldsymbol{\phi}$ by a pseudo inverse of B, found by taking either a QR factorisation or singular value decomposition (svd) of B. The QR factorisation is cheaper to compute than the svd method but it will fail if there are any degeneracies in the matrix B, i.e. if the rows are not linearly independent. The svd method will always produce the correct result but is more expensive to compute (see chapter 2 of [34] for a full description of these two methods). Since this step is not performed during the time stepping part of the method we can afford to use the svd method without significantly increasing the overall computational cost of the scheme. The svd method has been used here since it cannot be guaranteed that there are no degeneracies in B.

The singular value decomposition decomposes B into three matrices U, Σ and V^T, where U is column-orthogonal, V is row-orthogonal and Σ is a diagonal matrix of the singular values. If B is an $m \times n$ matrix, then U is $m \times n$ and Σ and V are $n \times n$. The matrix B can be written as,

$$\mathsf{B} = \mathsf{U}\Sigma\mathsf{V}^T \ , \tag{3.9}$$

so $\mathbf{a}$ can be found from,

$$\mathbf{a} = \mathsf{V}\Sigma\mathsf{U}^T\boldsymbol{\phi} \ . \tag{3.10}$$

For the cases when B is a square matrix, Σ is the diagonal matrix of the eigenvalues of B and V and U^T are the matrix of eigenvectors and its inverse.

In the over-determined case where B has more rows than columns, this $\mathbf{a}$ minimises the cost function r. In the under-determined case when there are fewer equations than terms in the polynomial the $\mathbf{a}$ that is found gives $r = 0$ and minimises $|\mathbf{a}|$. This second condition is not specified as a condition but is a byproduct of the singular value decomposition. It is not necessarily unwelcome since it will help to reduce any roundoff errors in computations using $\mathbf{a}$. Once suitable coefficients $\mathbf{a}$ of the interpolating polynomial have been found this polynomial can then be integrated across the area swept across the face in one time step giving

$$\hat{\phi} = \frac{1}{u\Delta t}\sum_{j=0}^{N} a_j \int_{-u\Delta t}^{0} x^j \ \mathrm{d}x \ . \tag{3.11}$$

As in the previous chapter, we can perform this integration to give

$$\hat{\phi} = \sum_{j=0}^{N} a_j \left(-u\Delta t\right)^j \ , \tag{3.12}$$

or equivalently

$$\hat{\phi} = \sum_{j=0}^{N} \left(-u\Delta t\right)^j \left[\mathsf{V}\Sigma\mathsf{U}^T\right]_j \cdot \boldsymbol{\phi} \ , \tag{3.13}$$

where $[\ldots]_j$ is the j^{th} row of matrix $[\ldots]$. This will approximate the fluxes but in order to see if these are reasonable approximations we must take a closer look at the schemes generated by this method. There are two cases to be considered. The first is the case of fitting an N^{th} order polynomial over fewer than $N+1$ cells. This is known as the under-determined system because this generates fewer equations than unknowns to be found. The second case involves fitting an N^{th} order polynomial over more than $N+1$ cells. This will give more equations than unknowns and is called the over determined system.

3.2.1 Under-Determined Interpolation

In this case there is generally more than one polynomial that zeroes the cost function r, so one of a number of interpolating polynomials is being used. Each of these polynomials will therefore give the correct solution for a Courant number of one, by removing everything from one cell and placing it in the next. In order to test the method we shall choose a stencil in the same way as in the previous chapter but with one less cell than is required to find a unique N^{th} order polynomial fit.

The coefficients of the polynomial fitted around each edge are calculated in the same way as before. The following argument is being given only to show some of the features of these coefficients. The polynomial fitted by the least squares method zeroes r and is the one which has the smallest value of $|\mathbf{a}|$, i.e.

$$\mathsf{B}\mathbf{a} = \phi \tag{3.14a}$$

$$\text{and} \quad \sum_{i=0}^{N} a_i^2 = a_{\min} \tag{3.14b}$$

where B has the same form as equation (2.10) but one fewer rows. We can write B as $\left[\acute{\mathsf{B}} \mid \mathbf{b}\right] \mathsf{D}_N$ where $\acute{\mathsf{B}}$ is the matrix formed in the fitting of an $N - 1^{th}$ order polynomial over these N cells, $\mathbf{b}$ is the vector from the final column of B and D_N is a diagonal matrix of the powers of Δx from 0 to N. We know from Chapter 2 that $\acute{\mathsf{B}}$ has an inverse so we can pre multiply equation (3.14a) by this and the inverse of D_{N-1} to get

$$\left[I \mid \mathsf{D}_{N-1}^{-1}\acute{\mathsf{B}}^{-1}\mathbf{b}\Delta x^{N}\right] \mathbf{a} = \mathsf{D}_{N-1}^{-1}\acute{\mathsf{B}}^{-1}\phi \ . \tag{3.15}$$

The general j^{th} equation in this set is,

$$a_j + \frac{\Delta x^N}{\Delta x^{j-1}}\left(\acute{\mathsf{B}}_j^{-1} \cdot \mathbf{b}\right) a_N = \frac{1}{\Delta x^{j-1}}\left(\acute{\mathsf{B}}_j^{-1} \cdot \phi\right) \ , \tag{3.16}$$

where the subscript j on a matrix represents the j^{th} row of that matrix. The right hand side of this equation is the j^{th} coefficient, $\acute{a}_j$, of the polynomial found by fitting an $N - 1^{th}$ order polynomial over these cells. We have N equations of this form for the $N+1$ unknowns, the final equation comes from the condition that the sum of the squares of the coefficients,

$$\sum_{j=0}^{N} a_j^2 = a_N^2 + \sum_{j=0}^{N-1}\left(\acute{a}_j - \frac{\Delta x^N}{\Delta x^{j-1}}\left(\acute{\mathsf{B}}_j^{-1} \cdot \mathbf{b}\right) a_N\right)^2 \ , \tag{3.17}$$

is minimised. This is minimised when its derivative with respect to a_N is zero, when

$$2a_N - 2\sum_{j=0}^{N-1}\left(\frac{\Delta x^N}{\Delta x^{j-1}}\left(\acute{\mathsf{B}}_j^{-1} \cdot \mathbf{b}\right)\right)\sum_{j=0}^{N-1}\left(\acute{a}_j - \frac{\Delta x^N}{\Delta x^{j-1}}\left(\acute{\mathsf{B}}_j^{-1} \cdot \mathbf{b}\right) a_N\right) = 0 \ , \tag{3.18}$$

re-arranging gives

$$a_N = \frac{\left(\sum_{j=0}^{N-1} \acute{a}_j\right)\left(\sum_{j=0}^{N-1} \frac{\Delta x^N}{\Delta x^{j-1}}\left(\acute{\mathsf{B}}_j^{-1} \cdot \mathbf{b}\right)\right)}{1 + \left(\sum_{j=0}^{N-1} \frac{\Delta x^N}{\Delta x^{j-1}}\left(\acute{\mathsf{B}}_j^{-1} \cdot \mathbf{b}\right)\right)^2} \,. \tag{3.19}$$

This shows that $a_N \to 0$ as $\Delta x \to 0$ since $\acute{\mathsf{B}}$ and $\mathbf{b}$ do not depend on Δx. The right hand side of equation (3.17) can be rewritten as a quadratic in a_N to also show that $a_N \to \sum_{j=0}^{N-1} \acute{a}_j$ as $\Delta x \to 0$. In effect, as $\Delta x \to 0$, we recover the $N-1^{\text{th}}$ order scheme when we use only N cells in the stencil. This result tells us that the scheme is consistent as $\Delta x \to 0$ through the consistency of the $N-1^{th}$ order scheme. Equation (3.16) suggests a relationship between the polynomial found by fitting a lower order polynomial over the cells and the polynomial we seek. The extra term, $\frac{\Delta x^N}{\Delta x^{j-1}}\left(\sum_{i=0}^{N} \acute{\mathsf{B}}_{ji}^{-1} \mathbf{b}_i\right) a_N x^n$, has no obvious meaning in the context of this method and will have an unknown impact on the schemes being generated.

In practice, applying this method generates unstable schemes with growing oscillations near discontinuities in the gradient of the advected profile. As would be expected from the argument above, the oscillations grow more slowly as the resolution is increased until ill-conditioning of the computation renders any results meaningless. This suggests that this application of the method is unusable without some refinement of the way the polynomial is fitted over the stencils. Even if this were done these results suggest that we would not gain the full accuracy of the N^{th} order polynomial being used.

3.2.2 Over-Determined Interpolation

For this case it is not generally possible to find a polynomial which satisfies $r = 0$, the cost function (3.8). Fitting a polynomial over the cells by a least squares fit in this case does not result in the integral of the polynomial over each cell being equal to the amount of the advected quantity in that cell. Such a scheme would therefore not give the exact solution if a Courant number of one is used, as many one dimensional numerical schemes do.

In order to test this method in one dimension we shall try fitting an N^{th} order polynomial over $N+2$ cells using the method described in Section 3.2. When we do this we find that growing oscillations are generated upwind of discontinuities in the gradient of the advected profile. These oscillations grow much more slowly than those for the under-determined case and are only generated upwind of discontinuities. If these problems are simply caused by no longer insisting on equality between the actual and numerical tracer amounts in each cell, then we may be able to remove these oscillations by insisting this equality holds over the region swept across the edge, i.e. the central cell or cells in the stencil.

A crude but apparently effective way of enforcing this equality in this framework is to apply a weighting to the cost function, i.e.

$$r = \|\mathbf{w} \cdot \boldsymbol{\phi} - \mathbf{w} \cdot (\mathsf{B}\mathbf{a})\|_2 \ , \tag{3.20}$$

where $\mathbf{w}$ is a vector of weights. The weights are unity for all but the central cells and corresponding rows of B in the stencil, for which they are larger. This increases the contribution to the cost function from any inequality over the central cells so a larger weight will reduce the inequality. This weighting is the same as pre-multiplying both sides of $\mathsf{B}\mathbf{a} = \boldsymbol{\phi}$ by a diagonal matrix of the weights. It is simple to implement in the code and carries through both the method and the code without requiring further adaptions. Increasing the weight for one cell forces the difference

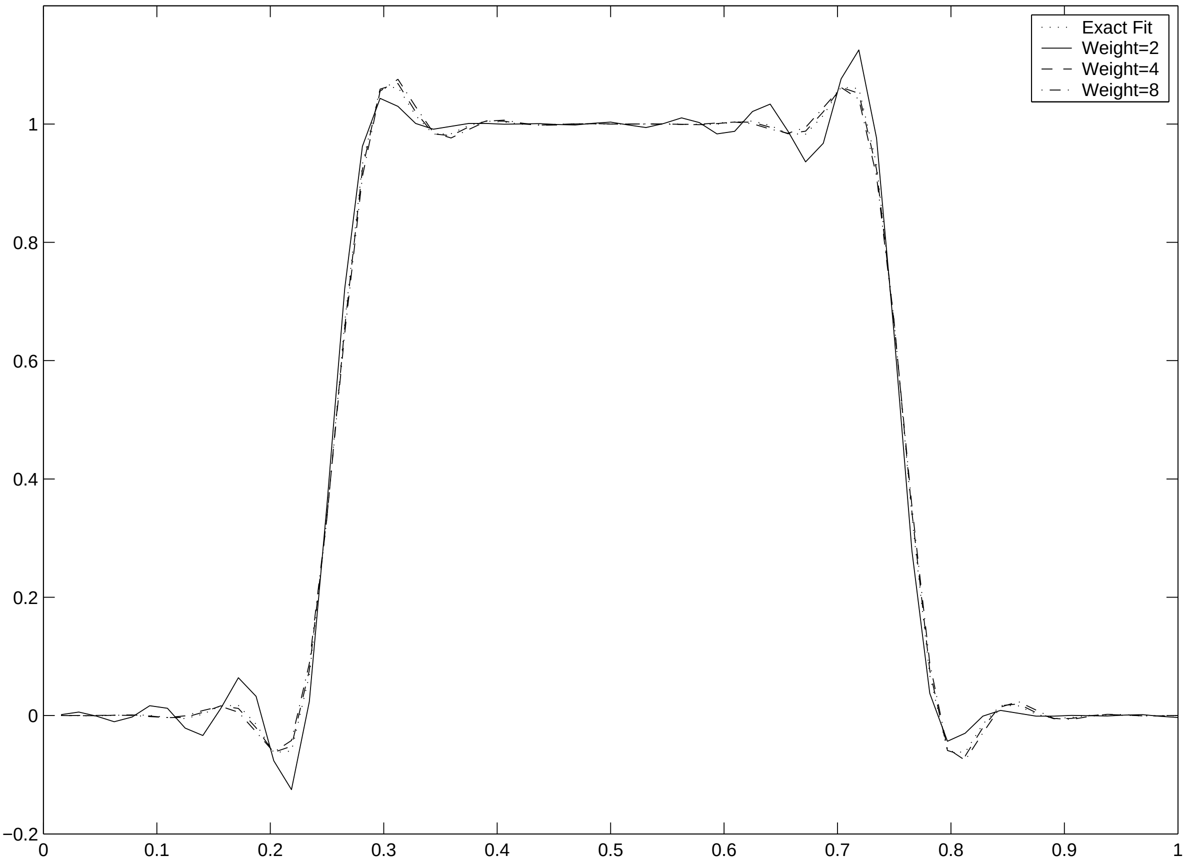


Figure 3.1: Numerical solutions of the advection of a square step once around a periodic domain using fourth order interpolation over six cells (five for the exact fit). 64 grid cells are used. A weighting is applied to the upwind cell during the fitting.

between the actual and numerical amounts of tracer to be smaller in order to minimise the cost function r. All that remains now is to decide on the magnitude of the weights and which cells to apply them to.

For a first attempt at this method we shall apply a weight only to the cell upwind of the edge for which we are approximating the flux. The results of using weights of magnitude 2, 4 and 8 with a fourth order polynomial are shown in Figure 3.1, along with the solution using exact interpolation over five cells. Any of these weights is enough to prevent growing oscillations from occuring, a weight of four or more being enough to generate a solution similar to that when using only five cells. Applying larger weights has very little effect on the schemes until they become large enough to affect the conditioning of B so that the numerics cannot be

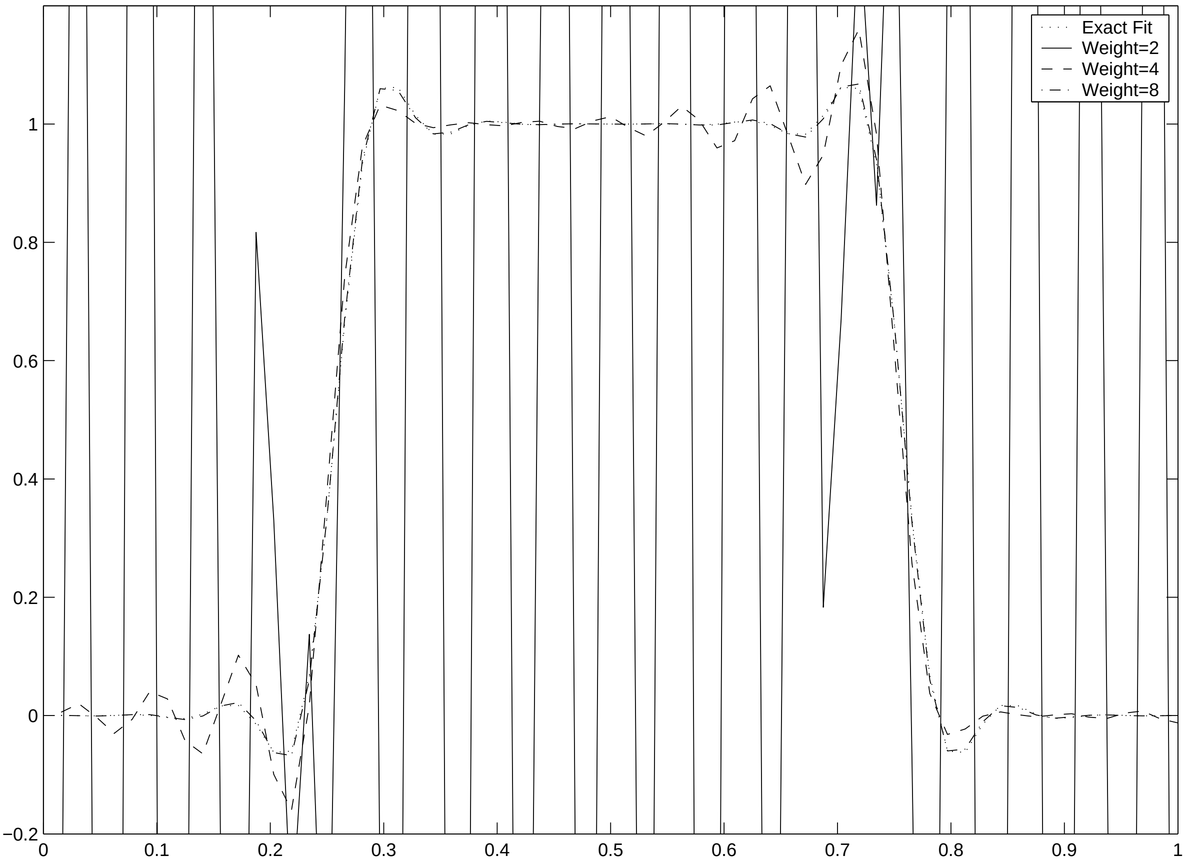


Figure 3.2: Numerical solutions of the advection of a square step once around a periodic domain using fourth order interpolation over six cells (five for the exact fit). 64 grid cells are used. A weighting is applied to three central cells during the fitting.

accurately computed.

If these same weights are used over the upwind cell and one cell to each side for the same test then the numerical results are as shown in Figure 3.2. In this case a larger weight is needed to ensure that growing oscillations are not generated, a weight of about eight or more is enough to recover a solution close to that given by an exactly fitted fifth order polynomial. Another option is to weight the upwind cell by more than its two neighbouring cells, in this case the results are almost identical to those in Figure 3.1.

These results suggest that fitting a polynomial over more than $N + 1$ cells using a weighted least squares fit is a suitable way of generating fluxes for these types of schemes. Experiments will need to be performed on other grids in order to decide on suitable values to use for the weights. Applying these ideas in two dimensions will enable us to tackle problems on two dimensional grids where we need to select stencils that contain more cells than there are terms in the polynomial we are fitting. Before attempting this we shall see what a similar error analysis to that used in the previous chapter can tell us about the accuracy of these schemes.

3.2.3 Error Analysis

Following the same argument that was used in the previous chapter, we define the error in a cell at time t_n to be the difference between the actual amount of the tracer in that cell and the approximated amount

$$e_i^{n+1} = \phi_i^{n+1} - \int_{(i-1)\Delta x}^{i\Delta x} \phi(t_{n+1}, x)\mathrm{d}x \ . \tag{3.21}$$

The right hand side can be written in terms of the values at a previous time and the fluxes across each edge of the cell to give

$$e_i^{n+1} = e_i^n + \sum_{\text{edges}} \left(\int_{\mathrm{area_n}} p(x)\mathrm{d}x - \int_{\mathrm{area(t_n)}} \phi(t_n, x)\mathrm{d}x \right) \ . \tag{3.22}$$

If the continuous and discrete wind fields are the same in this region for this time then

$$e_i^{n+1} = e_i^n + \sum_{\text{edges}} \int_{-u_0\Delta t}^{0} \left\{ p(x) - \phi(t_n, x) \right\} \mathrm{d}x \ . \tag{3.23}$$

Summing over each cell to find a bound on the global error gives,

$$\left| e^{n+1} \right| \leq \left| e^n \right| + 2 \sum_i \int_{i\Delta x - u_0\Delta t}^{i\Delta x} \left| p_{i_N}(x) - \phi(t_n, x) \right| \mathrm{d}x \ , \tag{3.24}$$

where i runs over all edges on the grid and $p_{i_N}(x)$ is the N^{th} order polynomial fitted around edge i in the global coordinates.

This again shows that the accuracy of the scheme depends on the accuracy of the polynomial interpolation over this region. In this case however we do not have as much information about the accuracy of this interpolation as before. We would expect that the accuracy of fitting the polynomial improves as the value of the cost function decreases. This will happen as Δx decreases or the order of the polynomial used increases. We cannot however say anything about an order of accuracy for these schemes. It is for this reason that we refer to the schemes by the order of the polynomial used rather than their formal order of accuracy.

This work in one dimension has shown us that we can fit an N^{th} order polynomial over more than $N+1$ cells and use this in the same method of generating advection schemes as before. This will be useful as we now look at two dimensional grids where we may not be able to find a suitable method of selecting a stencil that gives a uniquely determined system when fitting the polynomial. Finding a method of selecting a stencil is one of the problems that we now need to tackle, along with testing the method to see if the results are as good as on the square grid.

3.3 Two-Dimensional Advection

With the ideas from this one dimensional work we can now attempt to construct a generalisation of the square grid method of the last chapter for triangular grids. The main ideas will be the same as those used in the last chapter but with a new method of selecting a stencil and fitting the polynomial. Following the methodology of the last chapter we can begin to construct the new schemes.

3.3.1 Generating the Advection Scheme

We begin with the two dimensional advection equation in flux form,

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{u}\phi) = 0 \ , \tag{3.25}$$

where ϕ and $\mathbf{u}$ are two dimensional tracer and wind fields respectively. Discretising the time derivative gives

$$\phi_i^{n+1} = \phi_i^n - \sum_{\text{edges}} \hat{\phi} \ , \tag{3.26}$$

with $\hat{\phi}$ representing the flux across an edge of cell i in one time step. We wish to approximate these fluxes by the integral of a polynomial over an approximation of the region swept across the edge at each time step,

$$\hat{\phi} = \frac{\alpha}{\ell \, |\mathbf{n}| \, \Delta t} \int_{|\mathbf{n}|\Delta t} \int_{\ell} p_{_N}(x,y) \mathrm{d}x \mathrm{d}y \ , \tag{3.27}$$

where $\alpha = \frac{\ell|\mathbf{n}|\Delta t}{A}$, ℓ is the length of the edge, A is the area of cell i and $|\mathbf{n}|$ is the magnitude of the flow normal to the edge. The origin of the local $x - y$ coordinate system is defined separately for each cell. The polynomial $p_{_N}$ is found by fitting over a stencil such that

$$\int\int_{\text{cell i}} \sum_{k=0}^{N} \sum_{l=0}^{k} a_z x^{k-l} y^l \mathrm{d}x \mathrm{d}y = A\phi_i^n \ , \tag{3.28}$$

for each cell i in the stencil, where again $z = (k+1)k/2+l$. The difference between this method and that of the square grid now arises since we must find a different way of choosing the stencils over which to fit the polynomial.

On a general triangular grid the cells may not be aligned with one another so we will not be able to find a simple stencil as in the last chapter. The first order upwind scheme will use the cell upwind of an edge so this would be a suitable cell to begin our stencil. If we fit a zeroth order polynomial over this cell we will generate a first order upwind scheme. If we wish to use a first order polynomial then we will need to add two or more cells neighbouring this central cell, since there are three terms in the polynomial. On an irregular grid, distances between the centres of cells or their size could be used to select two cells to add. On a regular grid however, there may be no way to choose between cells other than using the flow direction. Using the flow to choose which cells to add to the stencil would require that a new stencil was chosen whenever the flow changed at that edge. This could be computationally very expensive. Instead we shall include all three neighbouring cells and use a weighted least squares best fit to find our polynomial, as we did in the one-dimensional case.

If we were fitting a second order polynomial then we would need to add more cells neighbouring those already in the stencil. In the case shown in Figure 3.3 this would also give us the 'right' number of cells (10) over which to fit a third order polynomial without needing to perform a least squares fit. On some grids we may need to add more neighbouring cells and find a least squares fit for a third order polynomial. We can continue adding cells in 'halos' of those cells neighbouring the current stencil until we have as many or more cells as there are terms in the polynomial we are seeking. Figure 3.3 shows how the stencil will be built up in this way; first, the darkest upwind cell, 0, is added, then the neighbouring cells labelled 1, then their neighbours, 2, and so on until we have enough cells.

Choosing a stencil in this way does require us to change the stencil if the flow changes so that a different cell becomes the upwind cell of the edge. This does not increase the computational cost significantly since the new upwind cell will already have a stencil as the upwind cell of another edge. In the code, a stencil is generated for each cell and then

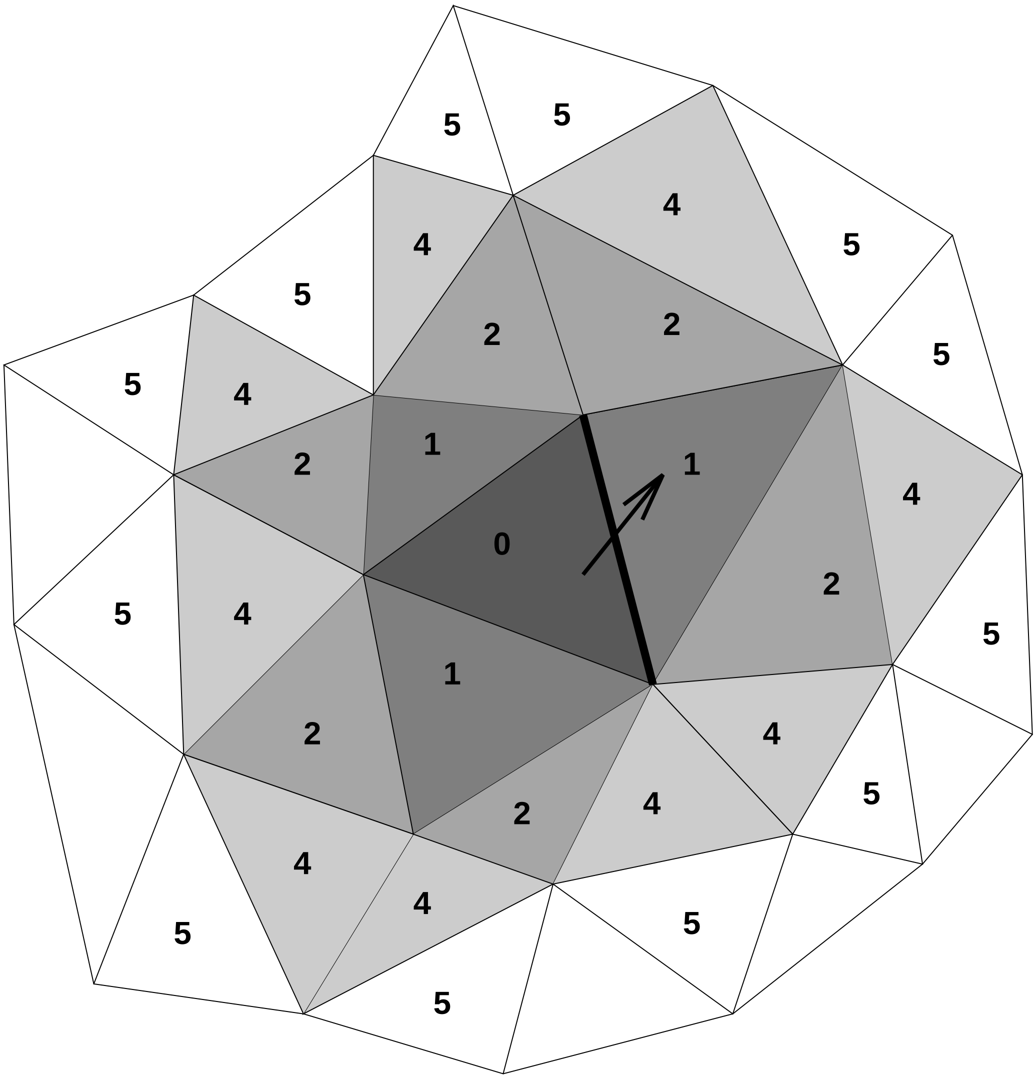


Figure 3.3: Diagram showing the stencils over which polynomials will be fitted for flow across the highlighted edge. The numbers show order of polynomial for which cells will be added to the stencil.

used whenever it is upwind of an edge.

The need for this change could be eliminated altogether by beginning the stencil with the two cells either side of the edge and adding their neighbours. This can cause problems with the zeroth order scheme, which is no longer shape preserving and in some cases is unstable. These problems can also occur when the weighting is applied to these central cells. These cases require special treatment and so this method is not followed here, since we are seeking a general method.

We can now perform the integrations in equation (3.28) to generate the usual matrix equation

$$\mathsf{B}\mathbf{a} = \boldsymbol{\phi} \; . \tag{3.29}$$

For this method of stencil selection B will generally have more rows then columns so we will again need to use a singular value decomposition to find the least squares fit polynomial. In the cases when B is square the singular value decomposition will find the inverse of B where it exists, as it does on every occasion in this work. In order to produce stable schemes we will need to apply weights to both sided of this equation as in the one dimensional case, giving

$$\mathsf{B}_w\mathbf{a} = \boldsymbol{\phi}_w \; , \tag{3.30}$$

where the subscript means that the weighting has been applied to these arrays. We can now find $\mathbf{a}$ from

$$\mathbf{a} = \mathsf{V}_w\Sigma_w^{-1}\mathsf{U}_w^T\boldsymbol{\phi}_w \; . \tag{3.31}$$

The vector $\mathbf{a}$ defines the polynomial $p_{_N}$ in equation (3.27) so we can now calculate the fluxes and update the advected field.

Since the matrix B depends only on the stencil, $\mathsf{V}_w\Sigma_w^{-1}\mathsf{U}_w^T$ can be cal-

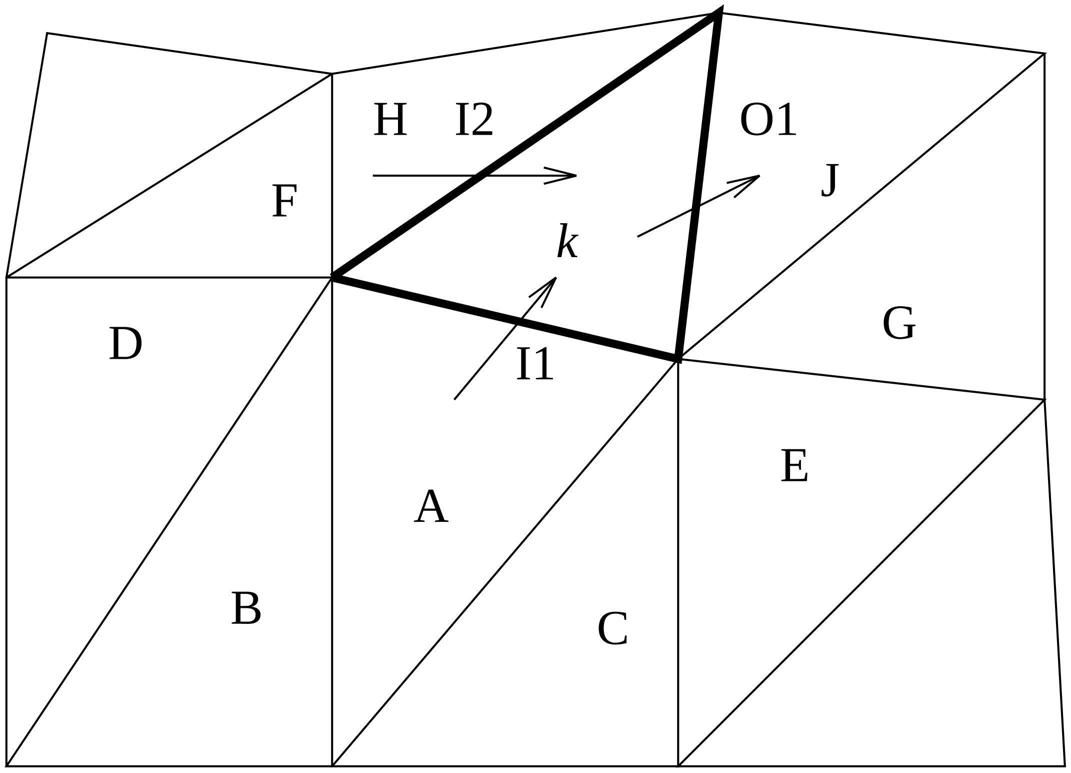


Figure 3.4: Diagram showing the labelling conventions used in describing the flux limiter. The arrows refer to the direction of the flow so I and O refer to inflow and outflow edges of cell k respectively

culated for each cell and the relevant one used for each edge. We then need only compute equations (3.31) and (3.27) and update the tracer field at each time step, limiting the fluxes if required. The irregularity of the grid makes writing down the steps of this process in the same way as in the last chapter a messy task. These calculations are not shown here but result in an equation of the form

$$\hat{\phi} = (\mathsf{G}_w \alpha) \cdot \boldsymbol{\phi}_w \ , \tag{3.32}$$

which is similar to equation (2.34) but includes the weighting and with G_w as a $S \times K$ matrix, S being the number of cells in the stencil and K the number of terms in the polynomial.

The limiter can be applied in the same way as before, the only difference being in the consideration of the region upstream of the edge. In the diagram in Figure 3.4, cell A is upwind of edge I1. Cells B and C are also in the upstream neighbourhood of I1 if there is flow from these cells into A. If B is in the upstream region then we must also include D if there is flow from this cell to B. Cells E, F, G, H and J can also be included in the upstream region in this way.

There are two other minor differences between this case and that of the square grid. The first is that performing exact integration is much less straight-forward on triangles than on rectangles. For this reason Gaussian quadrature was used to approximate integrals. The orders of quadrature used here are such that they integrate polynomials of order up to six exactly. Schemes using higher order polynomials will contain some extra errors but these do not appear to have a large effect on the overall accuracy of the schemes.

The second difference is in the representation of the flow fields. These have been defined in Cartesian co-ordinates and converted before being used by the scheme. The normal and tangential components of the flow field have been calculated at the centre of each edge and used as the values along that edge.

3.4 Test Cases

The same three test cases run on the square grid were used to test the schemes produced by this method on the triangular grid, a uniform flow, a rotational flow and a strong deformational flow. These cases have been chosen for the same reasons as before and so that we can compare results to see what effect the changes in the method have made to the numerical solutions. The same error measures have been used with grids of similar resolution. The grids were set over a unit square domain with equally

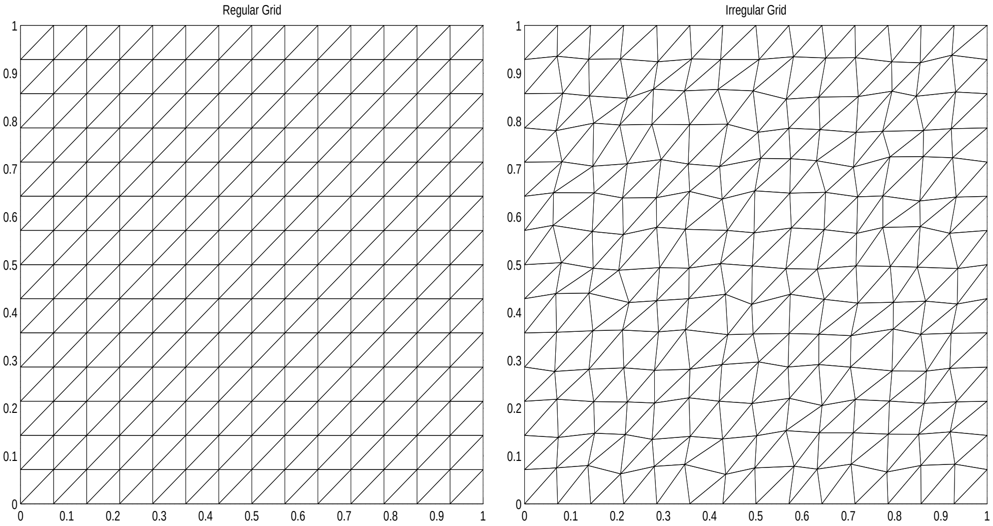


Figure 3.5: The regular and irregular grids used in the test cases on triangular grids.

spaced points along the periodic boundaries, interior points were placed to give both regular and irregular grids as shown in Figure 3.5. The irregular grid is formed by moving the points of the regular grid by a random amount of up to one third of a grid cell in each coordinate direction.

The same measures have been taken in these cases as in the previous chapter for the same reasons as before. Differences between the set up of the tests will be noted in each section along with any changes in the results. As well as comparing the schemes on square and triangular grids, we can also compare the results on regular and irregular triangular grids. For a more detailed discussion of the error measures used see Section 2.4

Initial experiments were run to determine a suitable value of the weight to be applied to the central cell of the stencil. In the same way as in one dimension, there was a wide range of values for the weights for which the final solutions were very similar. In this case however, a weighting of a magnitude of a few hundred was needed to ensure that the generated schemes were stable. The value of the weight used in all the remaining work was taken to be 1000. There were no cases where this value appeared to be too small to ensure stability or large enough to cause ill-conditioning of the numerics.

3.4.1 Uniform Flow

For this test case an initial profile was advected by a constant uniform flow at an angle to the cell edges on the regular grid. Three grid resolutions were used on a unit square domain, each double the resolution of the previous one and of comparable resolution to one of the square grids used previously. The regular grids are 14×14, 28×28 and 56×56 square grids with each square divided into two triangles, giving grids of 392, 1568 and 6272 cells respectively. These also form the basis of the irregular grids. Due to the increase in the memory required to store the stencils and their associated arrays these tests have not been run at a resolution equivalent to the highest resolution square grid. This problem was caused by inefficient coding and has been rectified for later implementations of the method.

The flow speed was 2 units per second in the x-direction 1 in the y-direction, a maximum Courant number of 0.5 was used in either direction. In order to maintain this Courant number, each doubling of the resolution requires twice the number of time steps to be used. The profile is advected twice around the periodic unit domain in the x-direction and once around in the y-direction in one second. This requires 112, 224 or 448 time steps on the regular grid depending on the grid size. More steps are required on the irregular grid since the smaller cells require a smaller time step to maintain the Courant number.

The initial profile used for this test case was a square step function defined as unity over $0.25 < x, y < 0.75$ and zero elsewhere. This test gives an idea of how the schemes cope with sharp gradients in the advected profile. This profile lines up with the regular grid but on the irregular grid we must integrate over each cell to find the initial tracer value in that cell. This was done using the same quadrature routine used within the method. The integration is therefore not exact and does not give the same initial conditions on each grid. For the purposes of measuring the errors, the 'true' final solution is taken to be the same as this initial condition on that grid. This means that the errors measured are due to the advection scheme, not the discretisation on each grid.

Figure 3.6 shows the final numerical solutions for four of the schemes on the 56×56 grid. Plot a shows the result of using a second order polynomial for the interpolation on a regular grid. This is comparable with the results of a similar scheme run on an 80×80 grid shown in Figure 2.7. The sharp gradients are not so well maintained in this case but this is mainly because the grid is better aligned with the profile in the square grid case. Increasing the order of the polynomial to four gives the result shown in plot c. As in the square grid case the sharp gradients are better maintained and there are more, smaller oscillations in the solution. Plot d shows the same order scheme run on an irregular grid. This shows little difference from the regular grid result except where the grid does not align with initial condition, seen as unevenness along the steep gradient. When the limiter is applied to the second order scheme there is a significant difference from previous results.

A more extreme example of this problem is shown in Figure 3.7. When the flow is aligned with one of the edges of the triangles there is very little distortion of the profile. When the flow is at a right angle to that edge there is significant distortion perpendicular to the flow. These results

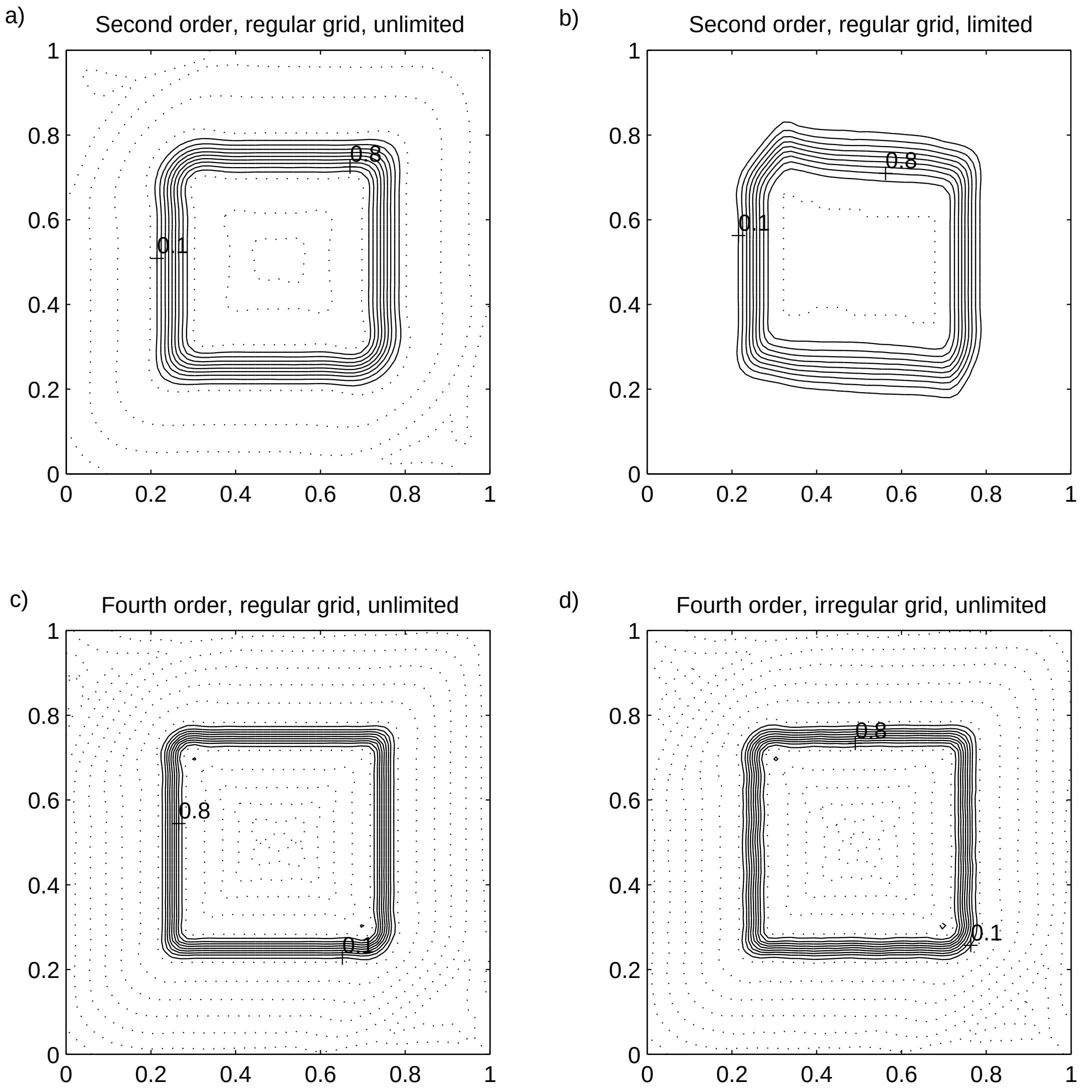


Figure 3.6: Numerical solutions of the advection of a square step around a periodic domain on a grid of 6272 triangles. The order of the polynomial used, the grid type and the state of the limiter are shown above each plot. Contours at 0.0 and 1.0 are shown as dotted lines

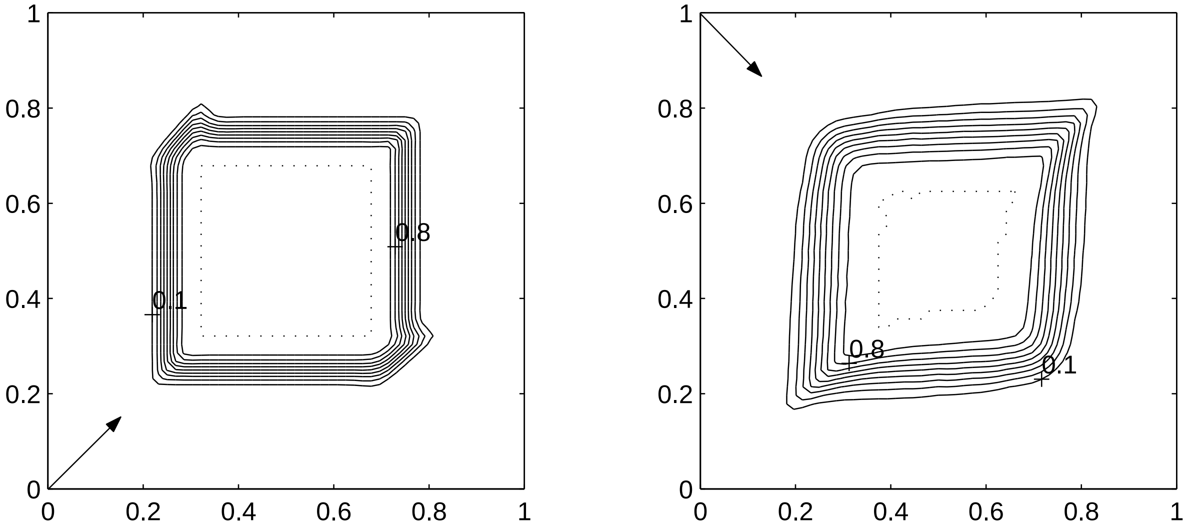


Figure 3.7: Numerical solutions of the advection of a square step around a periodic domain on a grid of 6272 regular triangles by a second order scheme with the limiter applied. The direction of the wind is shown by the arrows on the y-axes. Contours at 0.0 and 1.0 are shown as dotted lines

display the kind of anisotropic distortion that the refinements proposed in Thuburn [51] were designed to reduce. These refinements acted to relax the bounds on the inflow and outflow for each cell. On the triangular grid there will be either one inflow and two outflow edges or vica-versa so the bounds will be restricted by this 'lack of options' as to where the fluxes are distributed. This problem is seen, to a greater or lesser extent, in the majority of tests involving the limiter on the triangular grid. Whilst the basic scheme would appear to be working satisfactorily on this grid, the limiter has this serious drawback.

The L_2 and L_∞ errors for these schemes are shown in Figure 3.8 in the same way as the previous chapter. Plot a shows the results of using these schemes on a regular grid without applying the limiter. As before there is an improvement in the L_2 error as either or both of the order of the scheme and grid resolution is increased. The smaller errors for the cases on the coarsest grid are due to the method of measuring the error not taking into account the fact that the grid cannot resolve the initial

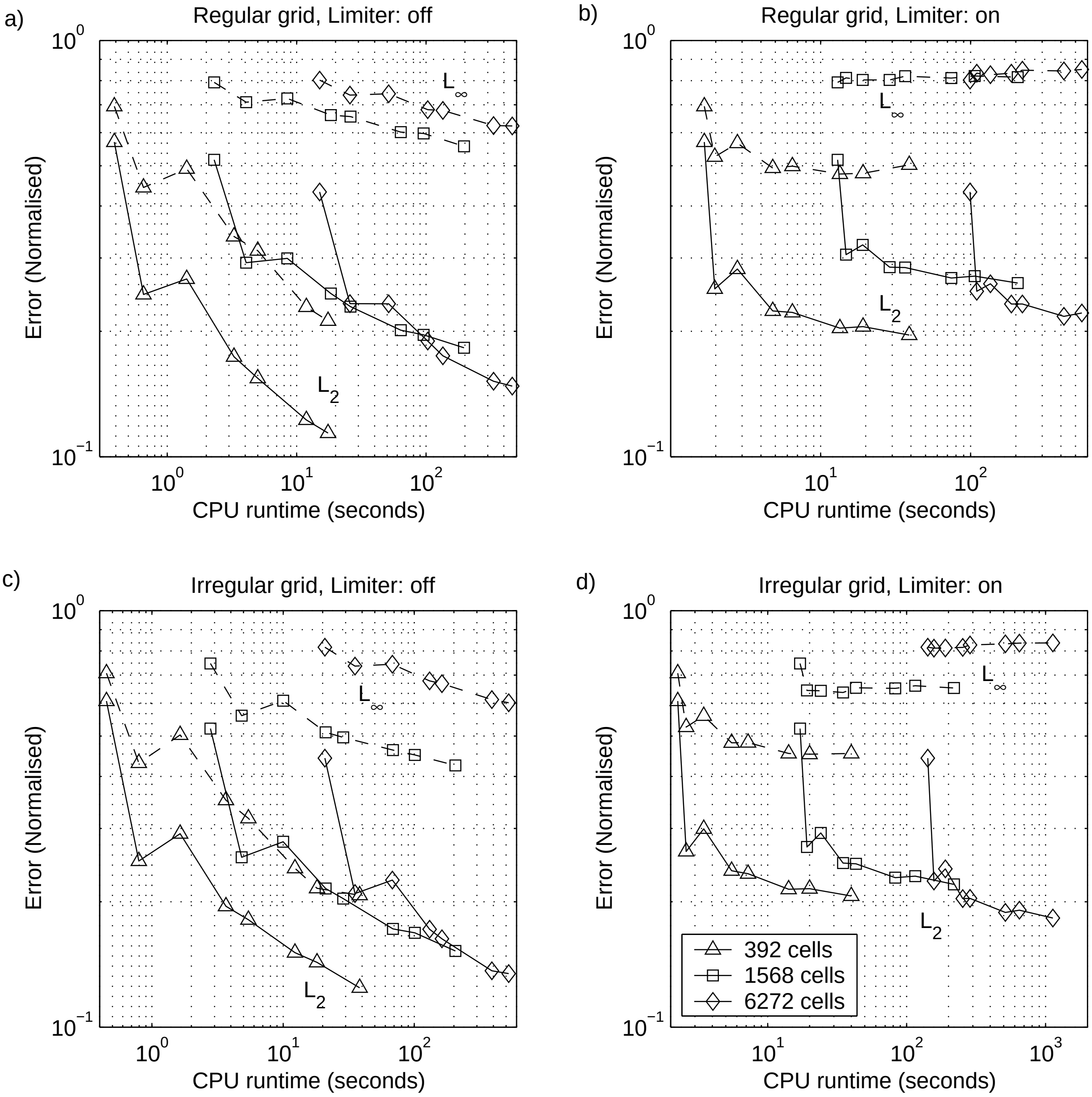


Figure 3.8: L_2 and L_∞ error norms for step profile advected by uniform constant flow on regular (top) and irregular (bottom) triangular grids. Grid sizes are as shown and each order scheme is marked.

condition as well as other cases.

In all other tests on this and the square grid, the initial step region was aligned with the grid cells and was thus more accurately resolved. For this resolution grid the gradient falls in the middle of cells and so is effectively smoothed in the initial condition. It is this initial condition that forms the final solution so the schemes are not trying to maintain as steep a gradient as in other cases. The L_∞ error improves as the order is increased but increases with the resolution. This is again because of the changes in the phase error relative to the resolution of the grid.

We have already seen an example of the distortions caused by the flux limiter, plot b shows the effect this has on the errors. The L_∞ error is increased by the application of the limiter and increases with the order of the polynomial, since the limiter is used more for these schemes. The L_2 error is also increased significantly over the unlimited schemes for the higher order schemes though there is sill a decrease as the scheme order is increased.

The results on the irregular grid, plots c and d, are very similar to those on the regular grid. The main difference is that both the L_2 and L_∞ errors are a little smaller on the irregular grid. This is again because the grid is not aligned with the initial condition and so the initial condition does not contain as sharp a gradient as on the regular grid. This is particularly noticeable in the L_∞ error where there is now a significant increase in the error as the resolution is increased. This suggests that this error comes from the region around the steep gradients and is caused by the phase error.

We can also decompose the L_2 error into diffusive and dispersive parts in the same way as on the square grid. Figure 3.9 shows this decompo-

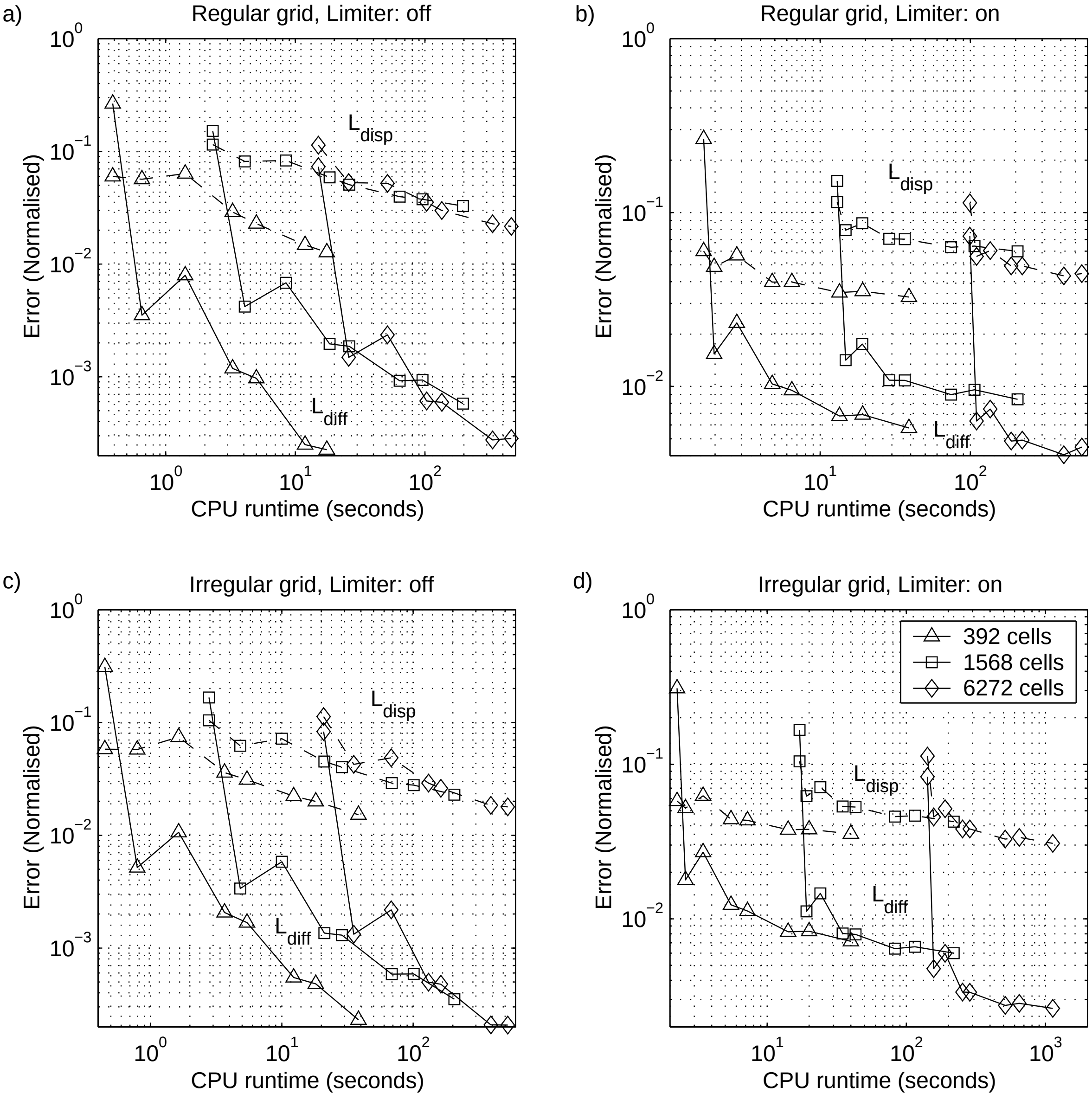


Figure 3.9: L_{diff} and L_{disp} error norms for step profile advected by uniform constant flow on regular (top) and irregular (bottom) triangular grids. Grid sizes are as shown.

sition applied to the same cases as above. In all cases the zeroth order scheme is shown to be highly diffusive but the higher order schemes have a dispersion error significantly larger than the diffusion. The dispersion error decreases with increasing order but more slowly than the diffusion error. This suggests that the higher order schemes maintain the steep gradients more effectively but make little improvement in the phase error.

Although these results on the triangular grids are not as good in quantitative terms as those on the square grid, they are qualitatively very similar. The behaviour of the schemes as the order is increased and the greater effect of the limiter on higher order schemes are particularly significant features present in both cases. The results for the advection of a smooth profile by the same flow as the square step behave in a similar way to the results above and on the square grid. They are not shown here but they are also quantitatively worse than the square grid cases whilst being qualitatively very similar.

The computational cost of generating and running these schemes on the triangular grids is two or three times larger than that of the square grid tests. This is mainly due to the number of stencils that are being used in each case. On the square grid there were only seven different stencils over which a polynomial needed fitting, on the triangular grid there was one stencil for each grid cell. The computational cost of running the schemes, ignoring the cost of their generation, depends only on the number of edges and the number of cells in each stencil. There are roughly $\frac{3}{4}$ the number of edges on the triangular grid compared to a similar resolution square grid but there will be a few more cells in the stencil of a method of comparable order.

3.4.2 Rotational Flow

For this test case an anti-clockwise solid body rotation was applied to an initial profile of a split cylinder in the same way as in Section 2.4.2. The rotation speed was 2π times the distance from the centre of the rotation so the solution after one second was the initial distribution. The same grids as used in the previous section were again used for this test. The time step was chosen to give a maximum Courant number of 0.5 across any edge of the grid. The final numerical solutions for the region around the cylinder from four of the schemes are shown in Figure 3.10

The first of these plots shows the result of using a second order scheme on a regular grid without the limiter. The slot shows a significant amount of filling and there is some erosion of the bridge. This scheme also displays the usual spurious oscillations and negative values that we have seen in other cases. Plot b shows the resulting solution when the flux limiter is applied to the fluxes generated by this scheme. This has removed the spurious oscillations and negative values but has caused more filling of the slot and erosion of the bridge. The solution does not show any of the anisotropic distortion that was seen in uniform flow case, most likely because the flow is rotational and there is no single direction in which the profile is stretched. Increasing the order of the scheme to four gives the result shown in plot c. As before, the steep gradients are better maintained and there are more, smaller oscillations. This leads to less filling of the slot and better maintenance of the bridge. Using the fourth order scheme on the irregular grid gives a solution very similar to the the one on the regular grid.

Figure 3.11 shows the L_2 and L_∞ norms for all cases on the regular

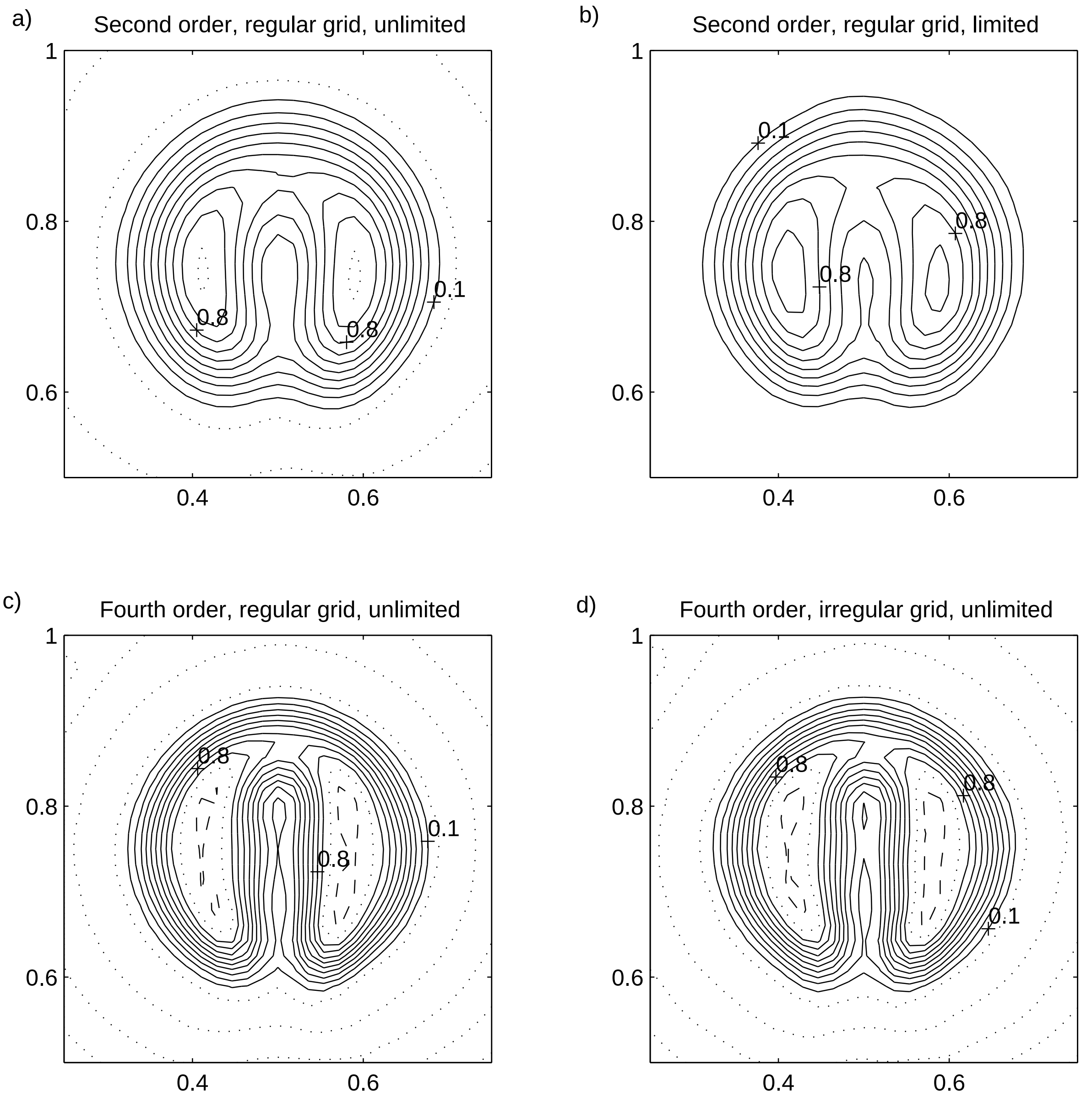


Figure 3.10: Numerical solutions of the advection of a split cylinder by a solid body rotational flow. The order of the polynomial used, the grid type and the state of the limiter are shown above each plot. Contours at 0.0 and 1.0 are shown as dotted lines. Only the region around the cylinder is shown.

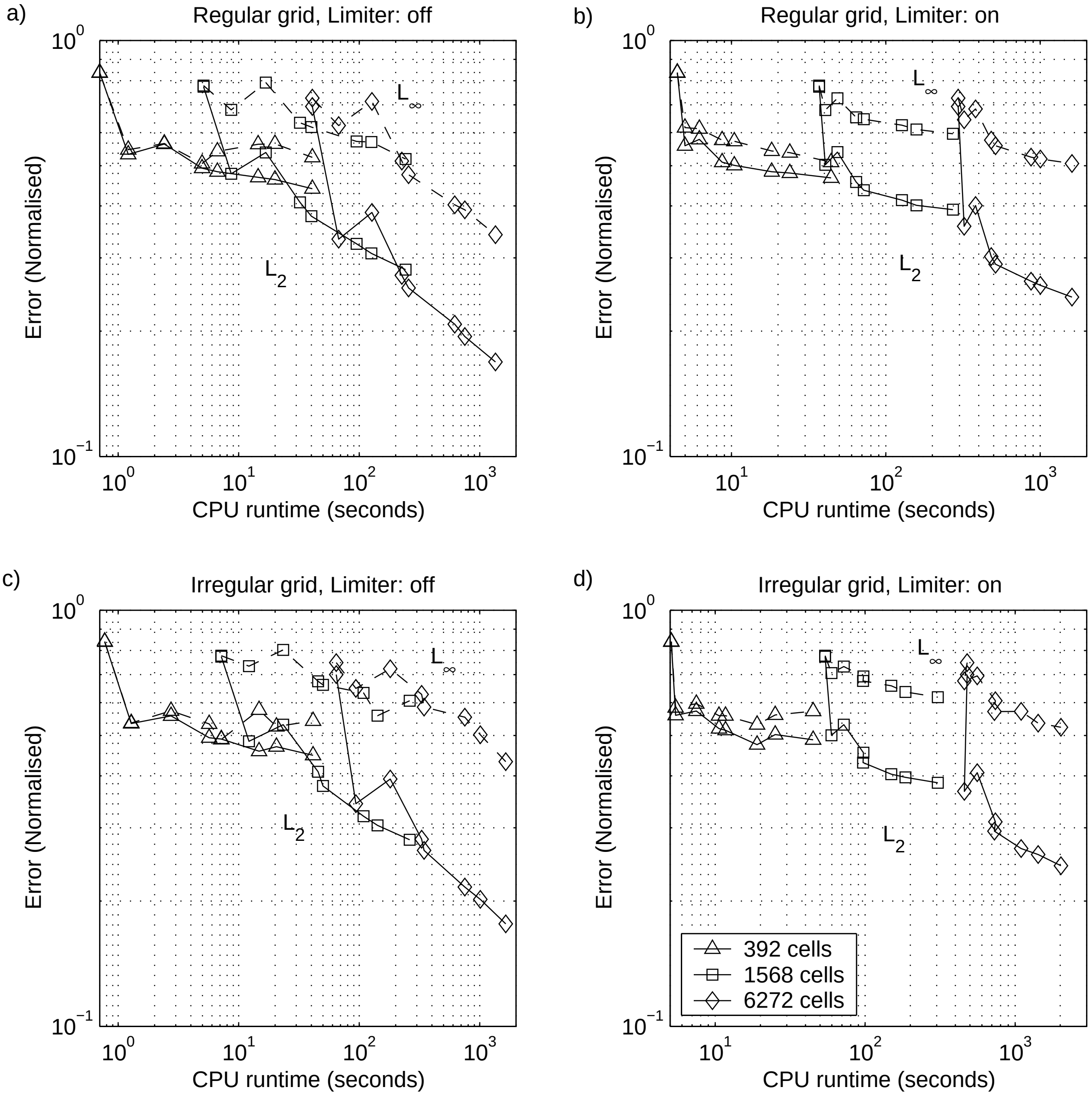


Figure 3.11: L_2 and L_∞ error norms for advection of a split cylinder by a rotational flow on regular (top) and irregular (bottom) triangular grids. Grid sizes are as shown and each order scheme is marked.

and irregular grids run with and without the limiter. On the regular grid without using the limiter the L_2 error behaves as in previous cases, decreasing with increases in the scheme order and grid resolution. On the coarsest grid the L_∞ error and L_2 error are very similar, a result of severe diffusion of the advected profile which loses almost all of the split cylinder shape. This is the case for zeroth order schemes on all grids but at higher resolutions the L_∞ errors is significantly larger than the L_2 error. The two error measures do behave in a similar fashion as the order of the schemes and resolutions of the grid are increased.

The results for schemes run without the limiter on the irregular grid (plot c) are very similar to those on the regular grid. The main difference is that the L_∞ errors are larger than on the regular grid, most likely because of the variation in cell sizes. When the limiter is applied to these schemes (plots b and d) the observed trends are barely altered. There is a general increase in the values of these error measures over the un-limited versions but the trend is still for reduced errors with increased resolution and order.

The diffusion and dispersion errors also behave in a similar fashion to the uniform flow test case. These errors are shown in Figure 3.12 for the regular and irregular grids, with and without the limiter. For the zeroth order cases, the diffusion error is comparable in size to the dispersion error. As the order of the polynomial is increased the diffusion error decreases more rapidly than the dispersion error. The diffusion error is insignificant compared to the dispersion error for all but the lowest order schemes and those on the coarsest grid. When the limiter is applied to these schemes, the order of the scheme must be above two before this distinction can be made.

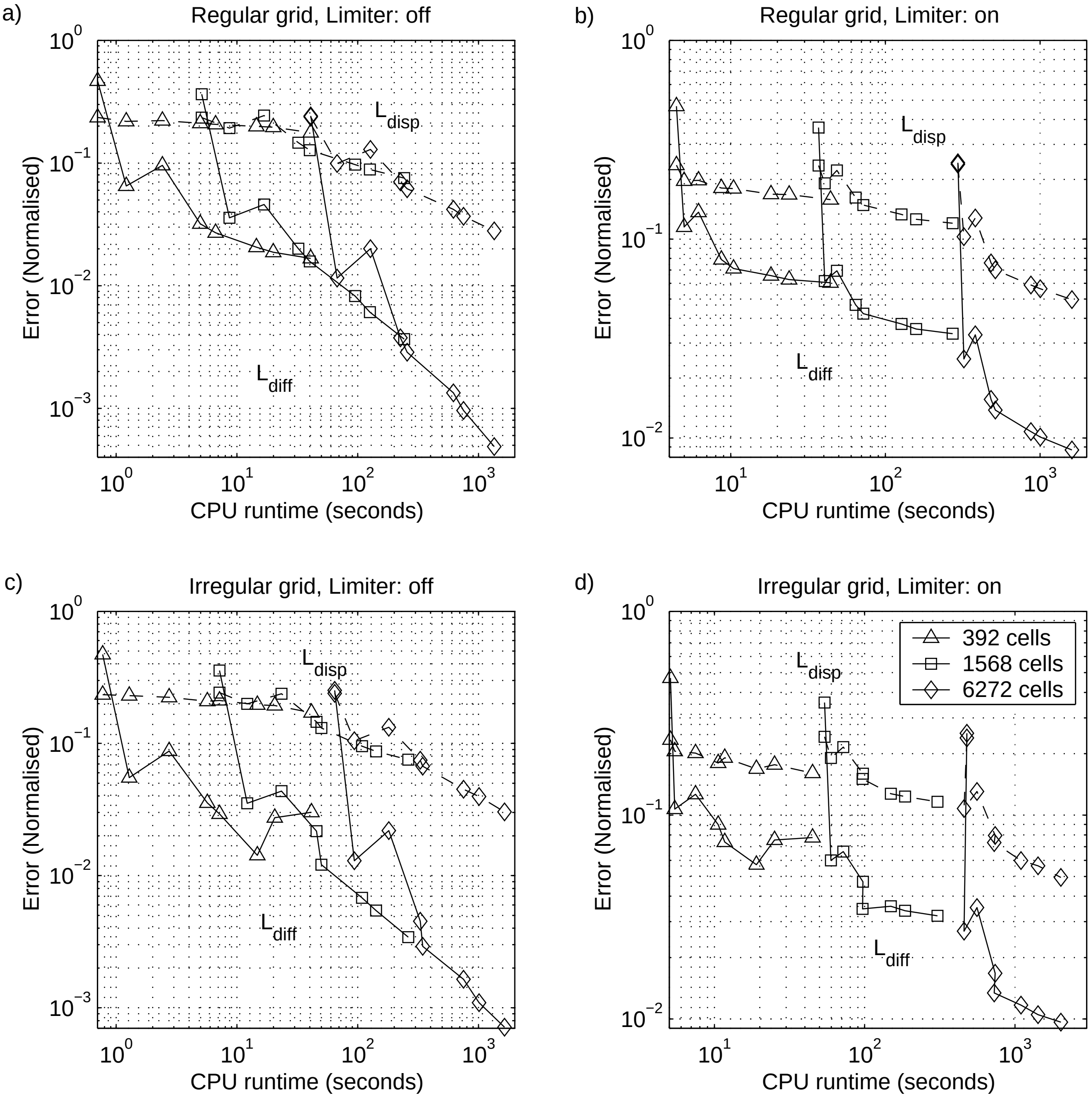


Figure 3.12: L_{diff} and L_{disp} error norms for a split cylinder advected by a solid body rotational flow on regular (top) and irregular (bottom) triangular grids. Grid sizes are as shown.

Once again these results have a very similar form to those of the square grid and to those of the uniform flow test case. In this case the values of the error measures are a little better than on the square grid. This is again because of the way in which the initial conditions were set up in this case. Integrating the specified initial condition over each cell led to some smoothing of the sharp gradient in the initial condition and thus the 'true' solution.

3.4.3 Deformational Flow

This case is set up in exactly the same way as for the square grid test. A flow field of 16 self contained, counter rotating vortices is used to advect the initial profile of a cone of height one unit over a domain 100 units square. The initial condition is only non zero over the six central vortices so the solution should be zero over the remaining vortices for all time. The initial profile is stretched out by the vortices and is wrapped around the centre of each vortex 50 times in time $T = 2637.6$ seconds. We would only expect to accurately model the the flow over a short time period of one or two rotations. After this time we would only hope to model the general features of the solution.

Figure 3.13 shows the short time behaviour of three schemes on regular triangular grids. The three schemes are a second and fourth order scheme on the 6272 cell grid and a second order scheme on a 25088 cell grid, all use the limiter. These results improve slightly for the fourth order scheme when compared to the second order scheme on the coarser grid, the results on the refined grid are much better than both. This was also the case on the square grid but there is an important difference in the results in this case.

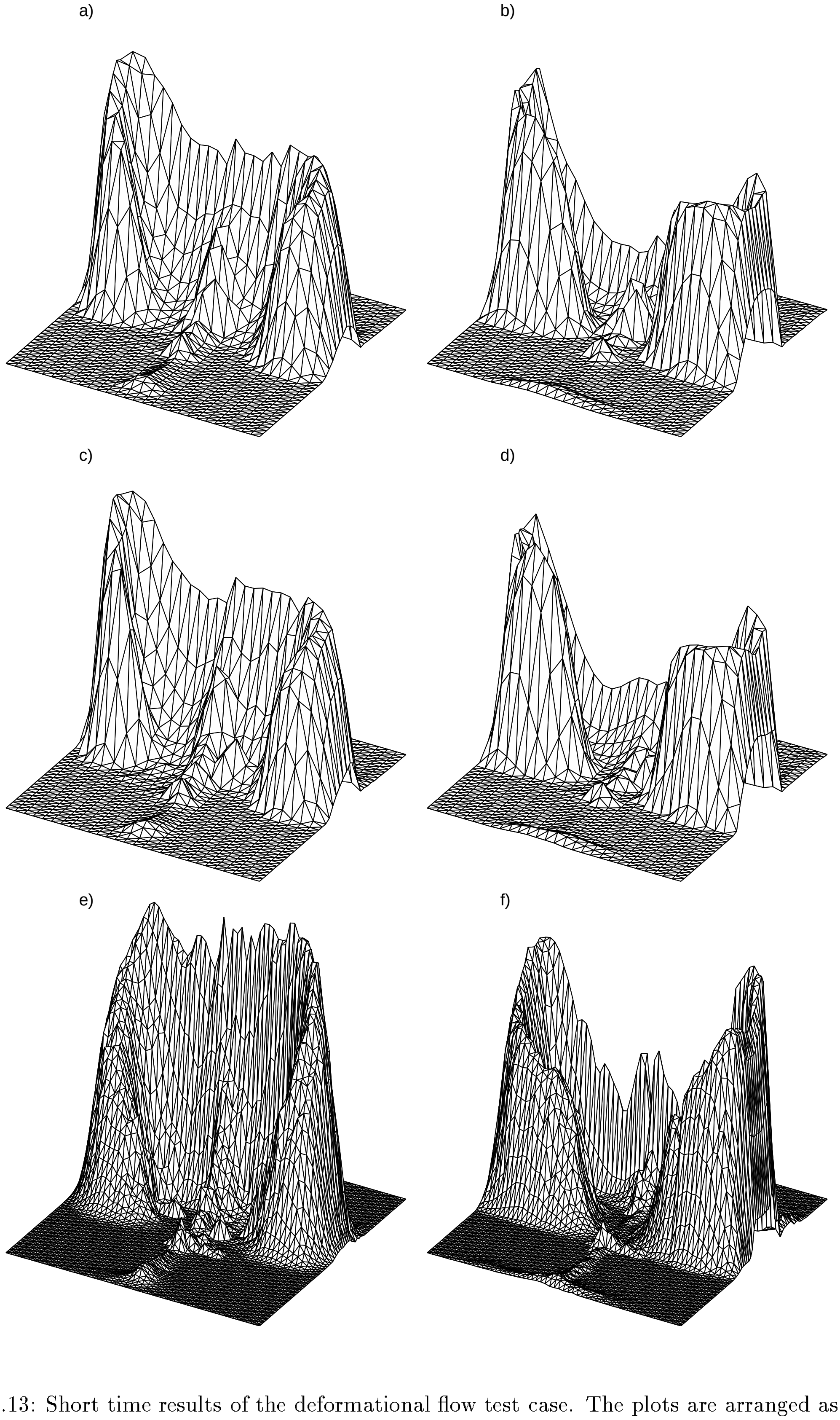


Figure 3.13: Short time results of the deformational flow test case. The plots are arranged as follows: a-b 6272 cell grid, second order. c-d 6272 cell grid, fourth order. e-f 25088 cell grid, second order. a,c,e: $t = \frac{T}{100}$. b,d,f: $t = \frac{3T}{200}$. All schemes have been used with the flux limiter and the x, y and z-axes have the same scale in each plot.

The results on the triangular grid are not symmetric in this case. The flow field and the initial conditions are symmetric and so should be the solution. The triangular grid does not share these symmetries and so neither do the solutions. Even with this difference we would expect the results to be more symmetric than they are. The reason for the asymmetry in these solutions is the anisotropic distortion caused by the flux limiter. This is only manifest when the flow is at an angle to the grid and so there have been different amounts of distortion over each vortex.

The effect of this distortion over the longer time period can be seen in Figure 3.14. After one rotation around the vortex we have still managed to capture the features of the flow with a streamer of tracer being wrapped around the vortex. This streamer is broader than it should be however and this can be seen in the profiles at later times. After five rotations around the vortex at $T/10$ we have lost all features of the tracer field. The only feature of the model that is 'correct' is that none of the tracer has crossed between vortices. The 'eggcup' profile is not visible in any of the profiles at time $\frac{T}{10}$ or later.

The results for this test on the irregular grid are very similar to those of the regular grid and are not shown here. On the irregular grid previously described the grid does not align exactly with the boundaries of the vortices. This will allow tracer to 'leak' from the central vortices into neighbouring ones in which there should be no tracer. In order to prevent this, an irregular grid was generated from the regular one by perturbing only the nodes that were not on the boundaries between the vortices.

The axes and contours are the same in figures 3.13 and 3.14 as in figures 2.14 and 2.15 for the square grid case so a direct comparison can be made between the results. These results do not compare well with

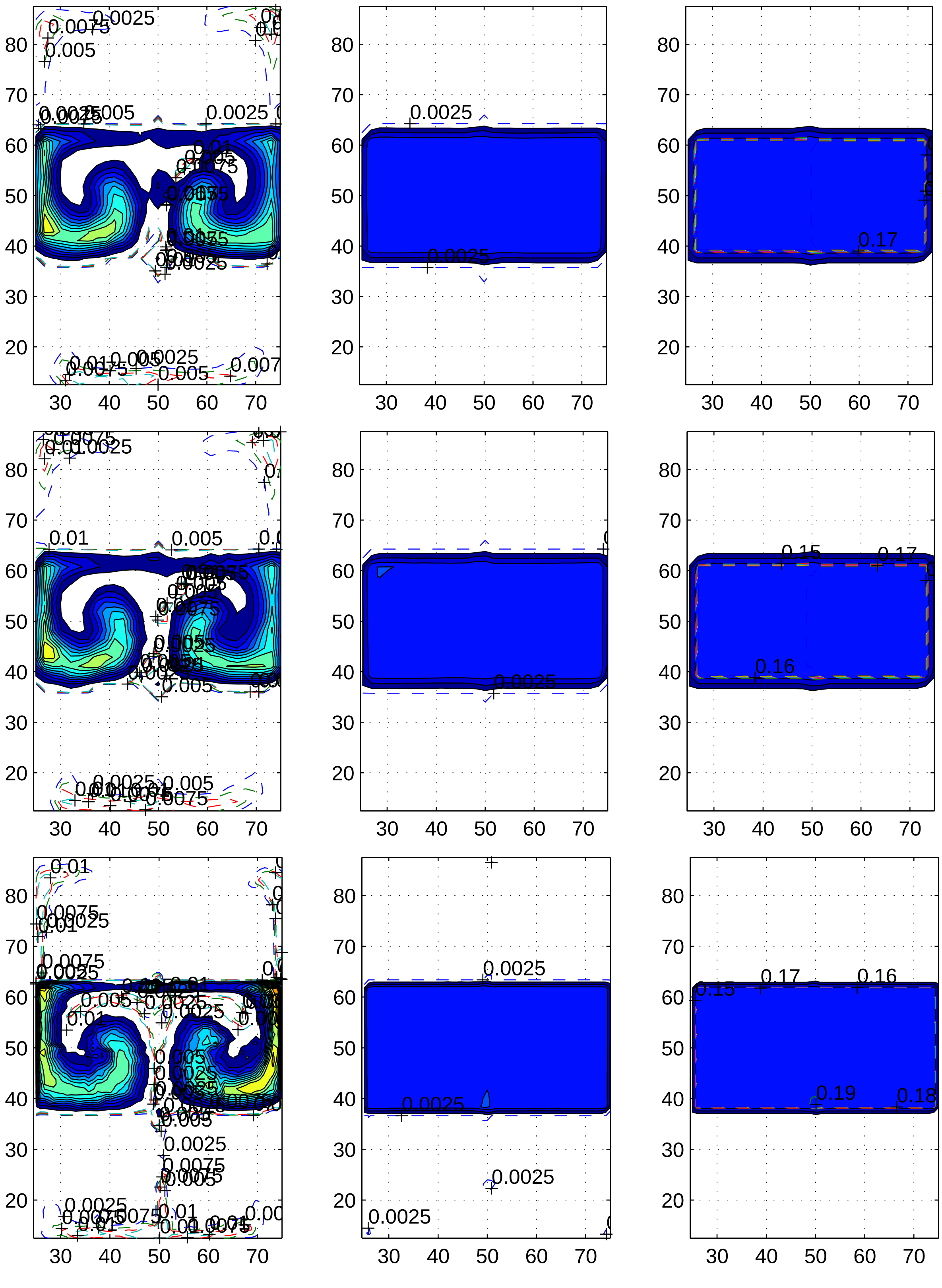


Figure 3.14: Long time results of the deformational flow test case. The plots are arranged as follows: First row 6272 cell grid, second order. Second row 6272 cell grid, fourth order. Third row 25088 cell grid, second order. First column, $t = \frac{T}{50}$. Second column, $t = \frac{T}{10}$. Third column, $t = T$. All schemes have been used with the flux limiter. Filled contours are at intervals of 0.05 and dashed contours as labelled.

those of the square grid, there is much more smoothing of the profile in all cases on the triangular grid. This smoothing is mainly caused by the flux limiter on the triangular grid but there may also be a contribution from the size and shape of the grid cells.

3.5 Summary

The method developed in the previous chapter has been adapted for use on triangular grids. This adaption was made by changing the way the polynomial is fitted over the region around each face. This was necessitated by the method of selecting stencils so as not to require the stencils to change whenever the flow changed. This in turn was desirable in order to keep down the computational cost of the scheme.

This method can now be applied to any grid on the plane. It may be possible to find different ways of selecting stencils on other grids as we did on the square grid. This may be of benefit to the schemes but neither this change nor any others should be required for the method to work. The accuracy of the schemes on other grids cannot be assured but the tests run so far show that the behaviour of the schemes varies little on different grids.

The test cases run using this new method of fitting the polynomial showed little difference in the behaviour of the schemes from the previous method as the grid resolution and scheme order was changed. This suggests that the problem is in some sense well posed since fitting polynomials in two different ways, and finding two different polynomials, has little effect on the overall behaviour of the solution.

The loss of accuracy in the test cases run with this method may be due to a number of factors. The two most likely reasons are the two things to have changed significantly from the last case, the method and the grid. The stencil is broadly similar to that used in the previous case but the method of fitting the polynomial is different. Results in the one-dimensional case showed that there was very little difference between exactly fitted polynomial and one of the same order fitted over one extra cell. This suggests that the reason for the loss in accuracy was due to the different grid used in this case.

The flux limiter displayed significant difficulties in coping with flows at an angle to the triangular grids. This caused a significant loss of accuracy in many cases where the limiter was used. These problems were not present on the square grid and they have not been observed on other grids (e.g. the icosahedral-hexagonal grid of the next chapter). This suggests that there is a specific problem for the limiter with triangular grids. This should not concern us unduly since we are only using this case as a stepping stone to look for ideas for extending the method to unstructured spherical grids. There are many more successful shape preserving schemes that can be applied to triangular grids.

More evidence to implicate the grid as the cause of the loss in errors might be found by returning to the square grid and selecting stencils and fitting the polynomial by the methods of this chapter. Another way is to use this method on other grids to see whether the results are more like those of the triangular or square grid cases. It is this second method that we shall follow in the next chapter by applying the method to an icosahedral-hexagonal grid on the sphere. The reason for making this next step is that the meteorological problems we wish to tackle are defined in a spherical geometry.

Chapter 4

Icosahedral-Hexagonal Grid on the Sphere

4.1 Introduction

The method of generating advection schemes that was applied to triangular grids in Chapter 3 can be applied to any grid on a plane without further adaptions. We wish to use these schemes in meteorological applications so we must also be able to apply the method to grids on the sphere. The basic method can be applied on the sphere but it will require the polynomials and integrals to be found in spherical coordinates. This does not require any adaptions to be made to the method itself but will require changes to the application of the method. Since the polynomials will have a slightly different form on the sphere, so will the matrices based on their integrals over the cells. This in turn will lead to later parts of the method requiring different forms also.

As this method is based on a local approximation to the tracer field over a stencil around each cell it is straightforward to simply approximate this region as a plane. Provided the region is small, i.e. it contains only a small proportion of the total number of cells, we might expect this approximation to create only a small additional error. Before trying this on the sphere we will begin by looking at what effect this has in one dimension by projecting the circle onto a line in Section 4.2. In Section 4.3 we will take a closer look at the icosahedral-hexagonal grid before applying the method to this grid in Section 4.4. The results of some test cases are presented in Section 4.5. The suitability of the method for inclusion in a more complex model is discussed in the final section, Section 4.6, before the method is applied to a shallow water model in the next chapter.

4.2 One Dimensional Advection on a Circle

We have already looked at how to apply this type of method to irregularly spaced grids in one dimension in chapters 2 and 3. In this case the grid will be defined on a circle by a regular angle, $\Delta\theta$, giving cells of uniform arc length, $r\Delta\theta$ where r is the radius of the circle, though this need not be the case. We can select a stencil over which to fit a polynomial in the same way as before. We choose $N+1$ cells centred either on the edge of interest or the cell upwind of it which are projected onto a line before we fit an N^{th} order polynomial over them.

Figure 4.1 shows two ways in which the circle can be projected onto the line. The line has been defined as the line through the two points that form the edges of cell 0. Cell 0 is the central or upwind cell of the stencil so we shall have a different line for each cell. The width of this cell, Δx_0, is found using the cosine rule

$$\Delta x_0 = \sqrt[+]{2r^2(1-\cos\Delta\theta)}\ . \tag{4.1}$$

Δx_1 will have a length that depends on the way in which we project the circle onto the line. If we project along the normal to the circle at the

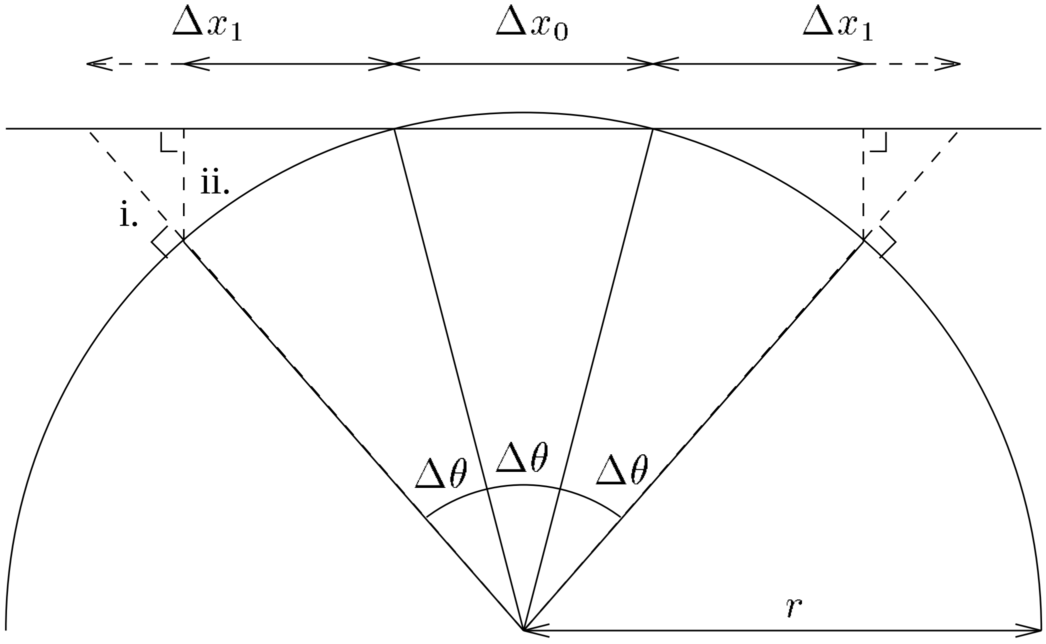


Figure 4.1: Projecting the circle onto the sphere. The distance Δx_1 depends on the method of projection (i. or ii.) and the angle $\Delta\theta$.

edge (method i.) then Δx_1 is given by

$$\Delta x_1 = \Delta x_0 \frac{\cos\left(\frac{\Delta\theta}{2}\right)}{\cos\left(\frac{3\Delta\theta}{2}\right)} \ . \tag{4.2 i}$$

If we project the edge in the direction normal to the line (method ii.) then

$$\Delta x_1 = \Delta x_0 \cos\left(\Delta\theta\right) \ . \tag{4.2 ii}$$

In general, if method i. is used to project each edge

$$\Delta x_n = \Delta x_0 \frac{\cos\left(\frac{(2n-1)\Delta\theta}{2}\right)}{\cos\left(\frac{(2n+1)\Delta\theta}{2}\right)} \ , \tag{4.3 i}$$

and if method ii. is used

$$\Delta x_n = \Delta x_0 \cos\left(n\Delta\theta\right) \ . \tag{4.3 ii}$$

We can see from these equations that $\Delta x_n \to \Delta x_0$ as $\Delta\theta \to 0$, so we would expect the error introduced by this approximation to decrease as the resolution is increased.

If we follow the same method as in Section 2.2, using a second order polynomial and a Courant number of one half, then the equation for the flux (2.16) has the form

$$\hat{\phi}^n = \frac{-\left(1-2f(\Delta\theta)\right)\phi_{-1} + \left(4+12f(\Delta\theta)+8f(\Delta\theta)^2\right)\phi_0 + \left(1-2f(\Delta\theta)\right)\phi_1}{4f(\Delta\theta)\left(1+f(\Delta\theta)\right)\left(1+2f(\Delta\theta)\right)}\ , \qquad (4.4)$$

where the function $f(\Delta\theta)$ depends on the method of projection. The function f takes the following forms,

$$f(\Delta\theta) = 1 \qquad \text{where } \Delta x_n = r\Delta\theta \quad \text{(circular case)} \qquad (4.5)$$

$$f(\Delta\theta) = \cos\left(\frac{\Delta\theta}{2}\right)\sec\left(\frac{3\Delta\theta}{2}\right) \qquad \text{where } \Delta x_n = \Delta x_0 \frac{\cos\left(\frac{(2n-1)\Delta\theta}{2}\right)}{\cos\left(\frac{(2n+1)\Delta\theta}{2}\right)} \qquad (4.5\ \text{i})$$

$$f(\Delta\theta) = \cos(\Delta\theta) \qquad \text{where } \Delta x_n = \Delta x_0\cos(n\Delta\theta)\ . \qquad (4.5\ \text{ii})$$

Some percentage errors in the coefficient of the known ϕ-values are given for both methods of projection in tables 4.1 and 4.2. The correct value for each coefficient is taken as the coefficient from the uniform grid case.

These tables show that both methods of projecting the circle on to the line are second order accurate as the resolution is increased. Method i. is about twice as accurate as method ii. when using a second order polynomial but the errors are similar when using a fourth order polynomial. In both cases the accuracy is lower when using the higher order polynomial. This is partly because the stencil is wider for the higher order case so a greater approximation has been made. Overall the errors are small, less than one percent for grids with a $\Delta\theta$ of 2°. This suggests that either method of projection is suitable for the orders of polynomials and grid resolutions that we will be using in two dimensions. Before applying this

$\Delta\theta$	Percentage error in coefficients					
	Method i.			Method ii.		
	ϕ_{-1}	ϕ_0	ϕ_1	ϕ_{-1}	ϕ_0	ϕ_1
10°	1.5427	0.0000	1.5427	0.7654	0.0000	0.7654
5°	0.3820	0.0000	0.3820	0.1906	0.0000	0.1906
2°	0.0610	0.0000	0.0610	0.0305	0.0000	0.0305
1°	0.0152	0.0000	0.0152	0.0076	0.0000	0.0076

Table 4.1: Percentage errors for the coefficients of a second order scheme found by projecting the circle onto the line. The errors in the coefficients of ϕ_0 are zero to machine accuracy

$\Delta\theta$	Percentage error in coefficients									
	Method i.					Method ii.				
	ϕ_{-2}	ϕ_{-1}	ϕ_0	ϕ_1	ϕ_2	ϕ_{-2}	ϕ_{-1}	ϕ_0	ϕ_1	ϕ_2
10°	6.7171	2.5420	0.0000	2.5420	6.7171	6.8808	1.9904	0.0000	1.9904	6.8808
5°	1.6570	0.6261	0.0000	0.6261	1.6570	1.6671	0.4878	0.0000	0.4878	1.6671
2°	0.2642	0.0998	0.0000	0.0998	0.2642	0.2644	0.0776	0.0000	0.0776	0.2644
1°	0.0660	0.0249	0.0000	0.0249	0.0660	0.0660	0.0194	0.0000	0.0194	0.0660

Table 4.2: Percentage errors for the coefficients of a fourth order scheme found by projecting the circle onto the line. The errors in the coefficients of ϕ_0 are zero to machine accuracy

approximation and our method to the sphere we shall take a closer look at the grid that we will be using.

4.3 Icosahedral-Hexagonal Grid

The icosahedral-hexagonal grid was first introduced as a near-uniform, spherical grid by Sadourny et al. [39] and Williamson [58]. These authors produced models of the barotropic vorticity equation on this grid in 1968. Until recently, little use has been made of the grid in meteorology when compared to the widely used latitude-longitude grid. The grid has been used for integrations of the shallow water equations by several authors, Masuda and Ohnishi [30], Heikes and Randall [10, 11] and Thuburn [53]. More recently an operational numerical weather prediction model has been developed at Deutscher Wetterdienst [29] using an icosahedral-hexagonal grid. A climate model using a modified icosahedral-hexagonal grid is currently under development by Tomita et al. [55].

The grids used by these models have all been called icosahedral-hexagonal grids though they do contain some subtle differences. The method of grid generation described here and used in tests of the advection schemes is also slightly different from these methods. These differences and the suggested reasons for them will be explained along with the method.

All methods begin with a regular icosahedron constructed with its twelve vertices on the surface of a sphere, shown in Figure 4.2 a. For simplicity we shall place two vertices at the poles so that half the remaining vertices lie spaced at 72° intervals on latitude circle 26.565°N and half similarly in the southern hemisphere. Each vertex is surrounded by five

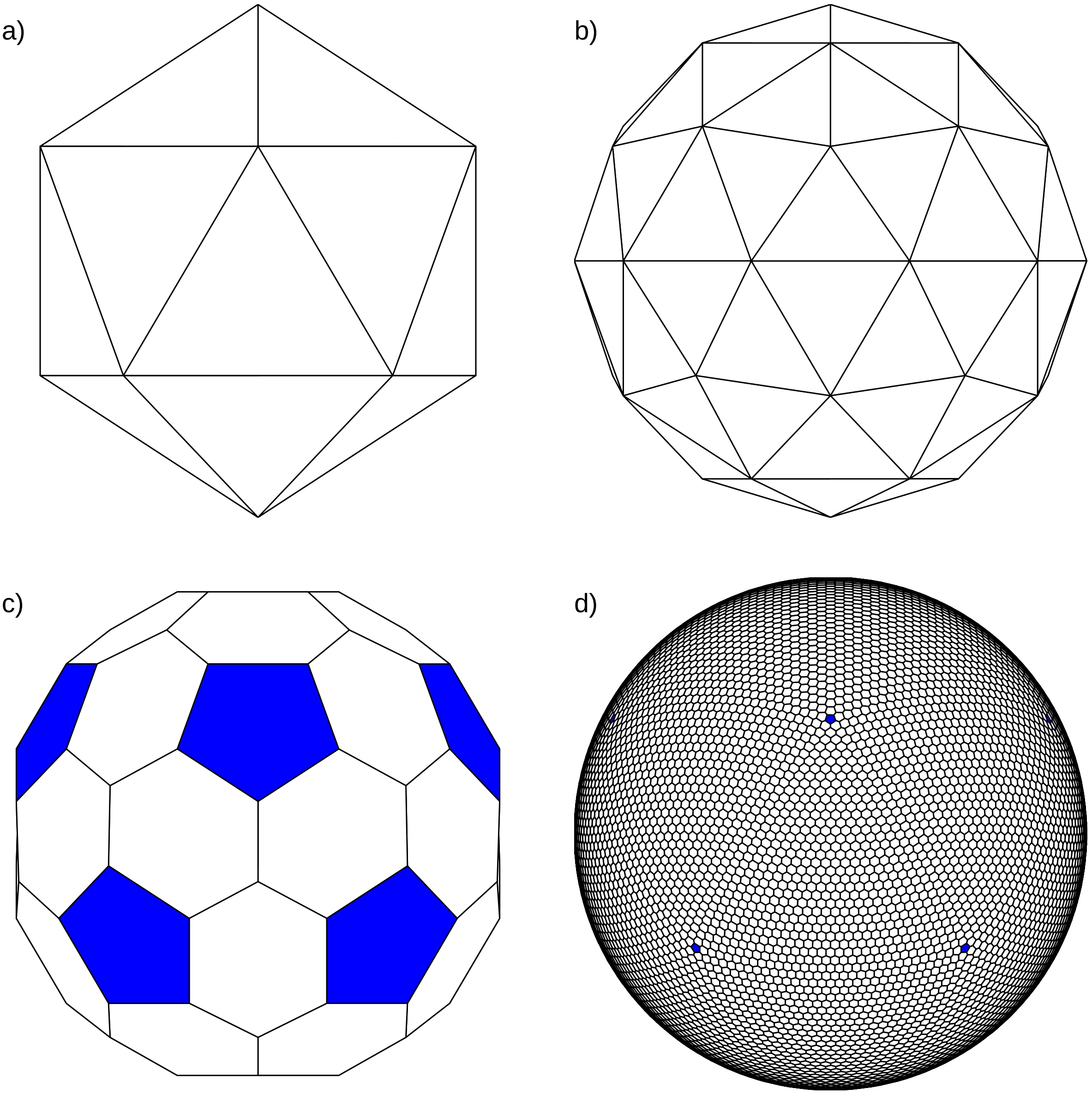


Figure 4.2: Generating the icosahedral-hexagonal grid. a) The icosahedron in a unit sphere. b) Each edge of the icosahedron is bisected and the new nodes projected onto the sphere. c) Grid 2, the dual of grid b, aka. the Bucky ball. c) Grid 6, the dual grid after five successive bisection-projections of the icosahedron. Pentagonal cells are shaded, all other cells are hexagonal

equilateral triangles so the dual of this grid is made up of twelve regular pentagons, a regular dodecahedron.

The dual grid is found by using the vertices on the sphere as vertices to form a Voronoi grid. Each Voronoi cell is defined as the collection of points p, on the sphere S, that are associated with vertex V_k by

$$\{|p - V_k| < |p - V_l| \ \forall \ p \epsilon S \ \& \ l \epsilon \{1, 2, \cdots, N\} \backslash \{k\}\} \ , \tag{4.6}$$

where $|\cdot - \cdot|$ is the great circle distance between two points on the sphere. This means that each cell is associated with a grid point and is made up of the region that is nearer to that grid point than any other. The edges of each cell are made up of points equidistant from two grid points and form the perpendicular bisector of the great circle arc between those points. The vertices of the dual grid are equidistant from three grid points and fall at the centres of the triangles on the original grid.

The twenty equilateral triangles that make up the icosahedron can be divided up in a number of ways to form similar spherical grids. Williamson [58] and Sadourny et al. [39] divided these triangles into as many smaller triangles as required on each plane face and then radially projected all the nodes onto the sphere, Figure 4.2 b. The nodes can then be joined by arcs of great circles to form triangular cells if required. The dual of this grid is shown in Figure 4.2 c.

Heikes and Randall [10, 11] suggest that a more even spacing of grid points is achieved by projecting points onto the sphere between each subdivision of the edges. Each of the twenty edges of the icosahedron is bisected and the new points are projected radially onto the sphere. This forms a grid of 80 triangles and 120 edges, each of which is again bisected and the new points projected onto the sphere. This procedure of bisection and projection can be continued to form a hierarchy of triangular grids on the sphere, the dual grid of each of these triangular grids forms another in the hierarchy of icosahedral-hexagonal grids used in this work. The new vertices added to the grid are each surrounded by six triangles and so are associated with hexagons on the dual grid. There are still twelve pentagons at the vertices from the original icosahedron. The dual grid formed after five successive bisections and projections is shown in Figure 4.2 d, five of the twelve pentagons can be seen still showing the underlying icosahedron.

This grid is not symmetric about the equator and because of this numerical results will not be symmetric when symmetric conditions are used. Heikes and Randall [10, 11] suggest rotating the southern hemisphere through 36° after the first subdivision of the triangles to create a symmetric grid. This 'twisted icosahedral grid' is not used here since the earth and its atmosphere are not symmetric, though some of the later test cases are. The grids described so far each have double the resolution of the previous grid in the hierarchy. Another possible adaption of this method of grid generation was suggested by Majewski et al. [28]. An initial trisection of the edges of the icosahedron, followed by projection and bisections, generates grids with resolutions between those in the original hierarchy.

The spatial operators that Heikes and Randall use on their grid in [10] are not consistent as the cell size goes to zero. This is an artifact of the mid points of the edges and the arcs between cell centres not coinciding. In order to remove this inconsistency the grid was 'tweaked' by minimising

$$R = \sum_{\text{cells}} \sum_{\text{edges}} {r_i}^4 \ , \tag{4.7}$$

where r_i is the distance between the midpoint of an edge and the midpoint of the arc joining the centres of the two cells. Heikes and Randall [11] give a more detailed explanation of the problem and its solution. This adaption is included in the grid used in this work with a slight modification. Instead of solving the global minimisation problem, each new grid point is 'tweaked' in turn and the process is repeated over the globe for twenty iterations at each grid refinement. This grid gives the same results for the test cases used by Heikes and Randall using their numerical schemes.

In this way we have a hierarchy of grids based on the number of subdivisions performed. Grid 1 is defined as a dodecahedron, grid 2 is the grid formed by one subdivision and so on. The number of cells N_{BOX}, edges N_{EDGE} and vertices N_{VERT} on each grid are related to the grid number N_{GRID} by

$$N_{BOX} = 5 \times M + 2 \tag{4.8}$$

$$N_{EDGE} = 15 \times M \tag{4.9}$$

$$N_{VERT} = 10 \times M \tag{4.10}$$

where $M = 2^{(2N_{GRID}-1)}$. Other properties of the grid are shown in table 4.3. These are the same properties shown in Table 1 of each of Heikes and Randall [10, 11] and Thuburn [53]. Comparing these tables shows that tweaking the grid equalises the areas of the grid cells but increases the variation in the distance between cell centres. This is likely to be beneficial in a cell based finite volume scheme but detrimental to a grid point scheme.

Williamson [58] suggested that it was beneficial for the distances between grid points to vary smoothly. Majewski et al. [28] also blame 'local grid nonuniformity' for reducing the slope of the infinity norm for their

N_{GRID}	N_{BOX}	N_{EQ}	A_{AVE} (km^2)	$\frac{A_{MIN}}{A_{MAX}}$	d_{AVE}	$\frac{d_{MIN}}{d_{MAX}}$	T_{EQUIV}	$\circ_{EQUIV}$
1	12	-	4.251×10^7	1.000	7053.9	1.000	T2-T3	60-90
2	42	10	1.215×10^7	0.884	3765.0	0.881	T6	40
3	162	20	3.149×10^6	0.914	1916.2	0.813	T12	20
4	642	40	7.945×10^5	0.937	962.4	0.791	T24	10
5	2562	80	1.991×10^5	0.935	481.7	0.787	T50	5
6	10242	160	4.980×10^4	0.929	240.9	0.785	T100	2.5
7	40962	320	1.245×10^4	0.926	120.5	0.784	T201	1.25

Table 4.3: Properties of the tweaked icosahedral-hexagonal grids. N_{GRID} is the grid number within the hierarchy of grids, N_{BOX} is the number of cells on the grid and N_{EQ} the number round the equator. A and d represent the area of cells and the distance between the centres of neighbouring cells respectively. The suffix $_{AVE}$ refers to an average value over the whole grid whilst $_{MAX}$ and $_{MIN}$ refer to maximum and minimum values. T_{EQUIV} is an approximate equivalent spectral truncation based on the number of degrees of freedom. $\circ_{EQUIV}$ is an approximate equivalent resolution on an $n^\circ \times n^\circ$ grid based on the number of grid cells

Laplace operator. A measure of the uniformity of the cell sizes is shown in table 4.4 which shows the average and extreme ratios of the sizes of neighbouring grid cells. This table shows that the sizes of neighbouring cells varies more smoothly as the grid resolution is increased on both the tweaked and un-tweaked grids. The largest difference in the size of neighbouring cells over the whole grid does not show an improvement as the resolution of the un-tweaked grid is increased. In contrast, this measure of the largest variation in cell sizes shows an improvement as the resolution is increased on the tweaked grid. This shows us that tweaking the grid acts to smooth the variations in cell sizes but does not tell us how this will effect our numerical results. To test this we must adapt our method and apply it to these grids.

N_{GRID}	Untweaked grid		Tweaked grid	
	$\frac{A_I}{A_J}_{AVE}$	$\frac{A_I}{A_J}_{MIN}$	$\frac{A_I}{A_J}_{AVE}$	$\frac{A_I}{A_J}_{MIN}$
1	1.000	1.000	1.000	1.000
2	0.941	0.884	0.942	0.884
3	0.949	0.902	0.978	0.931
4	0.960	0.895	0.990	0.944
5	0.973	0.895	0.995	0.944
6	0.983	0.895	0.997	0.945
7	0.991	0.895	0.998	0.949

Table 4.4: Relations between neighbouring grid boxes on icosahedral-hexagonal grids. N_{GRID} is the grid number within the hierarchy of grids. A represents the area of cells. $_{AVE}$ refers to an average value over the whole grid and $_{MIN}$ refers to the minimum value. $_I$ and $_J$ refer to the largest and smallest of a pair of neighbouring cells respectively.

4.4 Generating Advection Schemes on the Sphere

We can easily apply the method described in the last chapter to these icosahedral-hexagonal grids if we can overcome the problem of the spherical geometry. The results of projecting the circle onto a line in one dimension in Section 4.2 suggest that we can do this by projecting the cells in the stencils onto a plane. The only other part of the method that we need to consider again is the selection of a stencil. This could be done in the same way as before though this can lead to large numbers of cells being added at the same time. Instead a slightly adapted method is used on this grid.

The method begins as before by selecting a stencil for each edge of the grid. We again select a stencil for each cell and use the stencil for the cell upwind of each edge. Each stencil begins with the cell that we are finding a stencil for and (for all but the zeroth order scheme) the five or six surrounding cells. On the triangular grid we continued adding cells that shared one or more edges with cells already in the stencil. In this case we are going to restrict the number of cells added at one time by only adding cells that share two or more edges with those in the stencil. This reduces the number of cells that are added at one time and makes stencils more compact, reducing the error when the stencil is projected onto the plane. An example of this method of selecting the stencil on a hexagonal part of the grid is shown in Figure 4.3

Next we form the matrix B in the same way as before by integrating a general polynomial over each cell in the stencil but first we must project the stencil onto a plane. The plane we use is any one tangent to the sphere at the centre of the central cell in the stencil. This is simple to do by rotating the sphere in three dimensions so that the radius through the centre of the cell lies along the z-axis. Projecting the nearby points onto the plane by method ii. is then done by using only the x and y coordinates, ignoring the z coordinates altogether. If method i. is to be used for projection then a specific z-coordinate of the plane would need to be chosen. We shall be projecting using method ii. since there appears to be little difference between the projections and it is the simpler one. In mathematical terms the local coordinates are found by,

$$\begin{pmatrix} \cos\lambda\cos\varphi & \cos\lambda\sin\varphi & -\sin\lambda \\ -\sin\varphi & \cos\varphi & 0 \\ \sin\lambda\cos\varphi & \sin\lambda\sin\varphi & \cos\lambda \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \approx \begin{pmatrix} x' \\ y' \\ 0 \end{pmatrix}, \qquad (4.11)$$

where λ and φ are the latitude and longitude of the centre of the central cell, x, y and z are the coordinates of a point on the sphere in three dimensions and x' and y' are the associated coordinates on the plane.

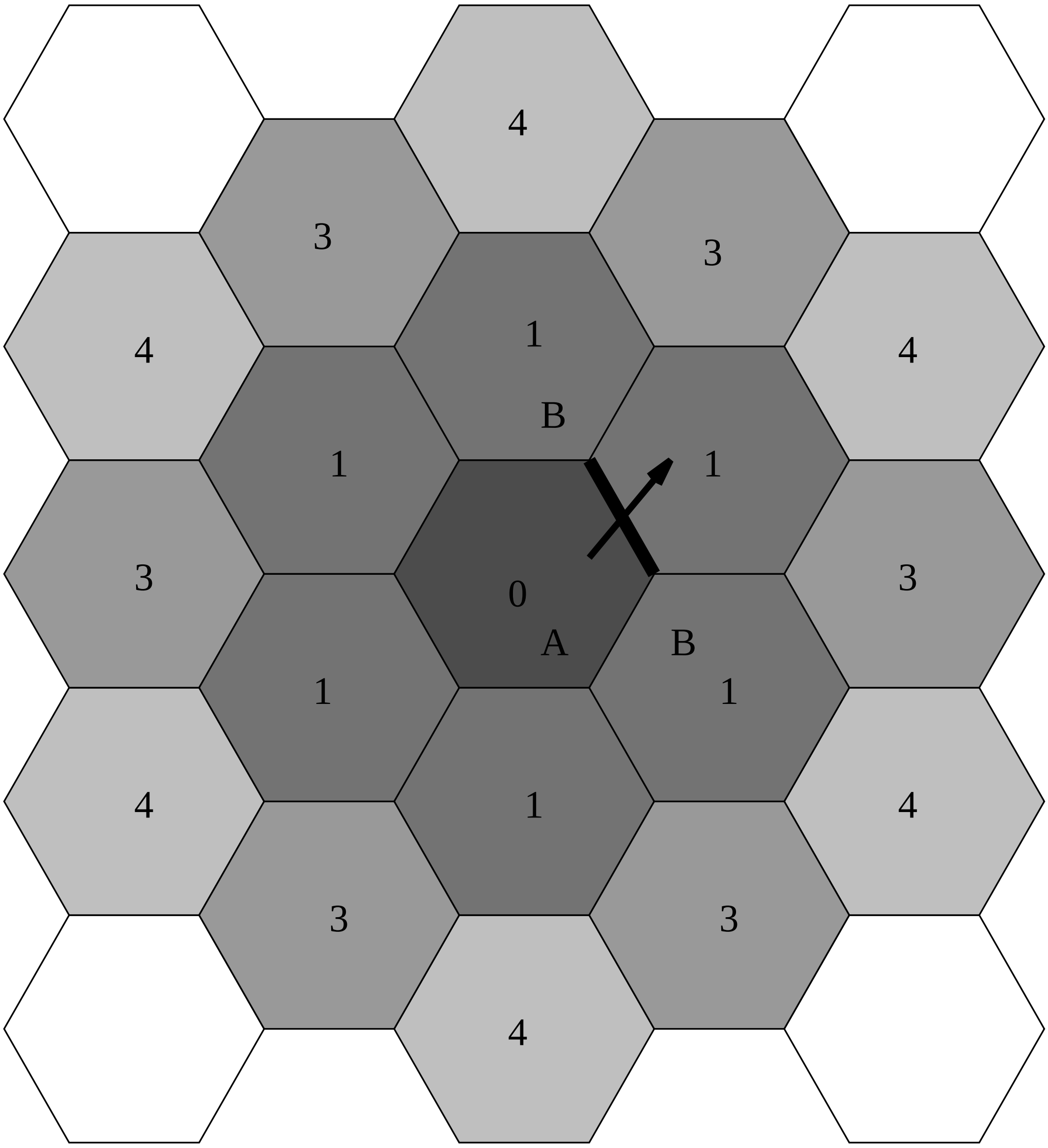


Figure 4.3: Examples of the stencils chosen over which to fit the desired polynomial. The numbers shown are the order of the polynomial for which the cells will need to be added to the stencil. These stencils would be used for the bold edge since the central cell is upwind of this edge.

We can now integrate a general polynomial over the cells projected onto the plane and apply a weighting to form B. As before the weighting can be over just the central cell or this cell and neighbouring ones. Just as on the triangular grid, whenever the magnitude of the weighting of the central cell is above a few hundred there is little difference between the results for either case. A weighting of only the central cell by a factor of 1000 is used in these cases and there was no need to alter this for any of the tests that follow.

Next we find the singular value decomposition of the B's and combine these with the integral of a general polynomial over the regions swept across the corresponding edges, as before. This integral is found on the same projection as the corresponding matrix B was formed. This generates the matrix G_w, the same as in equation (3.32), which is used along with the Courant numbers ($\boldsymbol{\alpha}$), to calculate the flux of the face and the weighted cell average concentrations of the advected quantity ($\boldsymbol{\phi}_w$).

Performing exact integrations on this grid is no easier than it was on the triangular grid. The easiest way to perform integrations over the pentagons and hexagons of the grid is to divide them into three or four triangles and use the same quadrature rules as before. This is slightly more computationally expensive but much simpler than finding quadrature rules for pentagons and hexagons.

The flux limiter works in the same way as on previous grids. The upwind region in this case is given by the cell upwind of the edge, cell A in Figure 4.3. The two cells that share an edge with this cell and a vertex of the edge (labelled B) will also be in the upwind region if there is flow from them across the shared edge. With these changes to the method in place we can now apply the generated schemes to some test cases to check that we have not altered their behaviour.

4.5 Test Cases

The flow used for advection test cases on this grid is that of constant solid body rotation. This is similar to previous tests on other grids and to the first test suggested by Williamson et al. [59] for testing numerical methods for the shallow water equations. These test cases will be referred to as the Williamson test cases and indexed by their numbers in [59]. Two initial conditions have been used, a square step and a cosine bell. The tests using a square step initial condition have been run and measured in the same way as in previous chapters. The cosine bell condition is the first of the seven Williamson test cases and the errors are measured as suggested by the authors. The first test we shall run with the step profile is to use one scheme and to alter the angle of the flow relative to the equator. The second test we shall use involves altering the order of the polynomials and grid resolution used to find the relationship between cost and accuracy as with the previous grids. The f inal test we shall use here is Williamson test case 1, advection of a cosine bell over the pole.

Before beginning these tests we need to define the wind fields on the icosahedral-hexagonal grid. This grid does not have any natural coordinate directions so we have several choices as to how to specify a wind field. We could simply store the wind field at the edges, cell centres or vertices using any coordinate system of our choice. In order to maintain independence from a coordinate system we can store the tangential and normal components of the wind at each edge. These components can be found easily from a streamfunction, ψ, defined at the cell centres. The difference between cell centres is used to find the tangential component, the values at the vertices are found by interpolation and then differenced to find the normal component of the wind. For some test cases there will also be a divergent component of the wind which can be found from a velocity potential, χ, in a similar way.

4.5.1 Flow angle dependence

Many numerical schemes that use dimensional splitting for two dimensional flows also have a dependence on the angle of the flow compared to the grid. With an evenly spaced grid and a two-dimensional advection scheme there should be little dependence of the numerical solution on the angle of flow. On a regular latitude-longitude grid advection around latitude circles is performed at a constant resolution around each latitude. Advection over the poles is performed over a grid of variable resolution around each circle of constant wind speed. These differences will lead to different numerical results, their significance depending on the numerical method being used. On the icosahedral-hexagonal grid the grid spacings are uniform in all directions so we would not expect this difference to occur.

In order to test if there is any flow angle dependence in the numerical solutions of this scheme we shall run tests with the same initial conditions and a constant rotational flow at different angles to the grid. The initial condition is defined as being one in the region $-15° < \lambda,\ \varphi < 15°$ and zero elsewhere. The winds are given as

$$u = u_0 \left(\cos\varphi\cos\alpha + \sin\varphi\cos\lambda\sin\alpha\right) \ , \qquad (4.11_u)$$

$$v = -\ u_0 \sin\lambda\sin\alpha \ , \qquad (4.11_v)$$

where α is the angle of the flow relative to the equator. These are defined in the model in terms of streamfunction and velocity potential,

$$\psi = - au_0(\sin\varphi\cos\alpha - \cos\lambda\cos\varphi\sin\alpha) \ , \qquad (4.12_\psi)$$

$$\chi = 0 \ . \qquad (4.12_\chi)$$

The model is run with realistic Earth values for parameters, including the radius of the sphere, gravitational acceleration and rotation rate of the sphere, though these have no effect in this case. u_0 is chosen so as to advect the profile once around the globe in twelve days and has a value of $u_0 = 2\pi a/(12 \text{ days})$, around 40 ms^{-1}. Flows used are around the equator (α =0°), across the poles (α =90°), at a large angle to a latitude-longitude grid (α =45°) and a small angle to a lat-long grid (α =0.05°).

Figure 4.4 shows the results of applying a second order scheme and the flux limiter to these conditions on grid 6. The four plots show that there is no significant difference between the numerical solutions in each case. This shows that the errors generated are independent of the direction of the flow relative to the grid orientation. These results also show that we do not have the same problems the limiter caused on the triangular grid. There is no noticeable anisotropic distortion caused by the limiter or the grid. Similar results were found using other schemes generated by this method.

4.5.2 Order, accuracy and computational cost

In order to investigate the relationship between the order of the schemes and the accuracy and computational cost, schemes of increasing order have been applied to one case. The same step initial condition as in the previous test was used along with the rotational flow that takes the

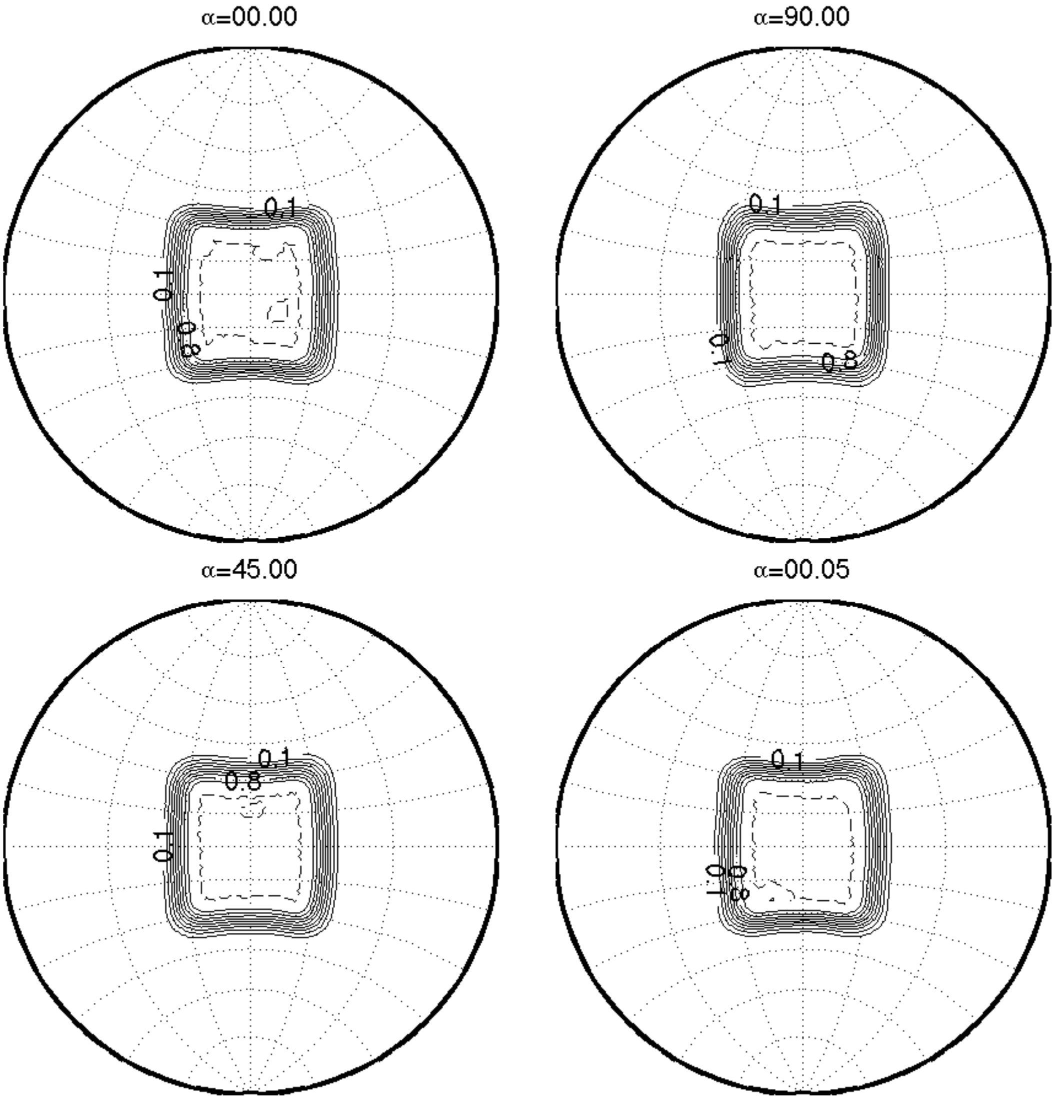
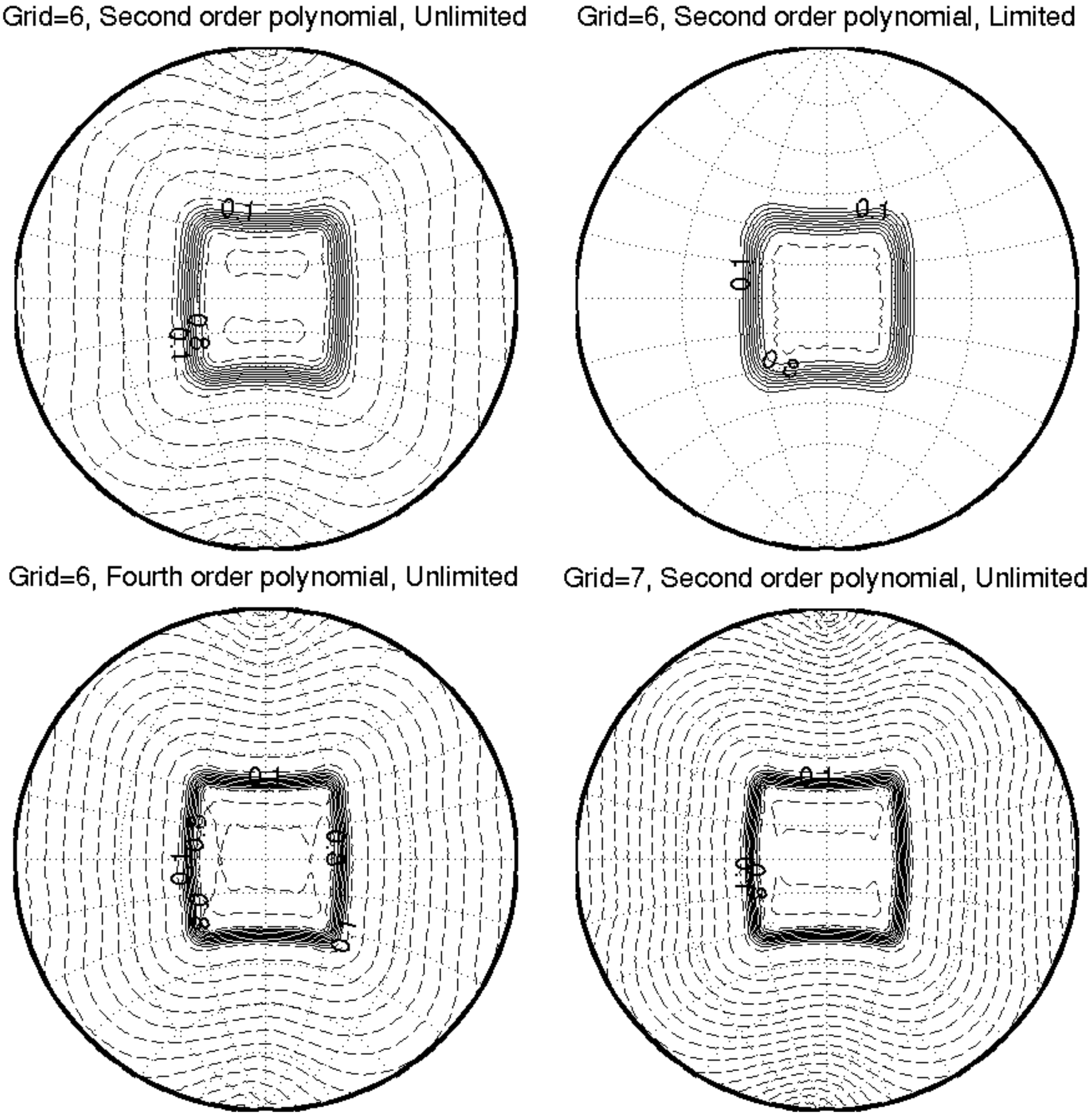


Figure 4.4: Numerical solutions of the advection of a step region around a sphere at an angle α to the equator. The numerical scheme uses a second order polynomial and the flux limiter on grid number 6. The contour interval is 0.1 and contours at 1 are dashed.

profile across the poles (α =90°). This test was run on grids 5, 6 and 7 using schemes based on zeroth to sixth order polynomials both with and without the flux limiter. The same error measures as in previous chapters have been used to measure the error with the CPU runtime is taken as the measure of the computational cost.

Figure 4.5 shows a selection of the solutions generated by these schemes. Using a second order scheme without the limiter on grid 6 gives a solution



Grid=6, Second order polynomial, Unlimited

Grid=6, Second order polynomial, Limited

Grid=6, Fourth order polynomial, Unlimited

Grid=7, Second order polynomial, Unlimited

Figure 4.5: Numerical solutions of the advection of a step region around a sphere across the poles. The grid, order and limiter used are given above each plot. The contour interval is 0.1 with contours at 0 and 1 as dashed lines.

which shows a smoothing of the sharp gradients and unphysical oscillations. When the limiter is used in conjunction with this scheme the oscillations are removed and there is little further smoothing of the sharp gradients. If the limiter is not applied but a fourth order polynomial is used then the steep gradients are better maintained. The higher order polynomial does however introduce more oscillations into the solution. If the resolution of the grid is increased instead of the polynomial order the improvements in the solution are similar. The sharp gradients are better maintained and more, smaller oscillations are introduced into the numerical solution. These relationships between the orders and the features of the numerical solutions are the same as in the tests run on other grids in previous chapters.

Figure 4.6 shows the same error norms as for previous cases for all orders of polynomials on grids 5, 6 and 7. The schemes again behave in a very similar way to previous cases using other grids. Without the limiter the L_2 and L_∞ errors decrease as higher order polynomials are used and the grid resolution is increased. The changes in the L_∞ error at different resolutions are mainly due to the slightly different representations of the initial conditions on the different grids. When the limiter is applied the oscillations generated by these schemes are removed at the expense of lessening the sharp gradients. The effect of removing the oscillations outweighs the lessening of the gradients, in terms of the L_2 and L_∞ errors for low order schemes. For schemes of order four and higher the lessening of the gradients is greater than the accuracy gained from the removal of the oscillations.

Decomposing the square of the L_2 error into diffusive and dispersive components also shows that the results are similar to tests on other grids.

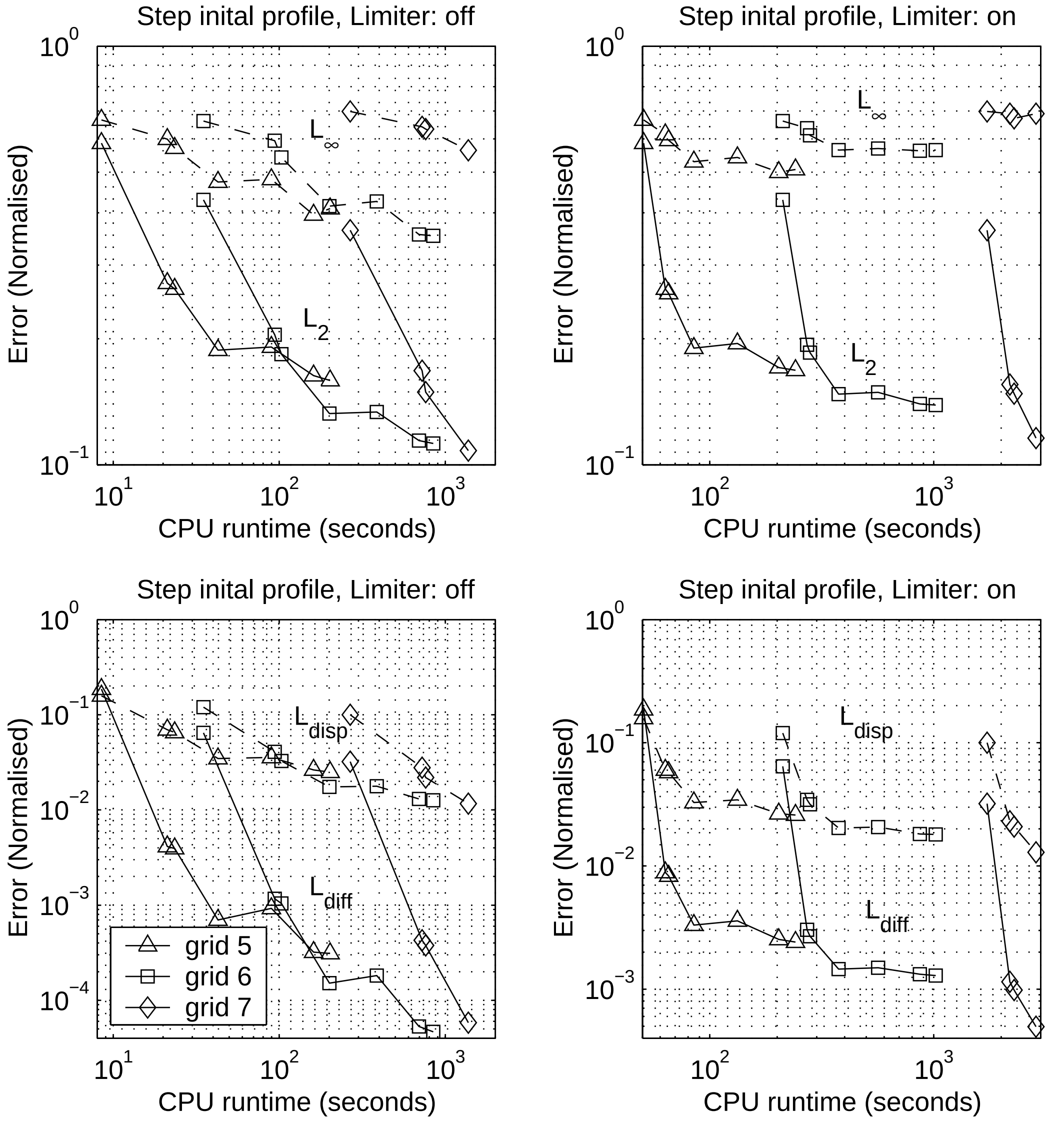


Figure 4.6: Error norms for the advection of a step profile around a sphere across the poles.

For all but the zeroth order scheme the diffusion part is an order of magnitude or more smaller than the dispersion error. Both parts of the error decrease as the order is increased when the limiter is not used. When the limiter is applied these errors do not reduce significantly when schemes above third order are used.

When the CPU runtime is compared to these measures of accuracy we can see how the two are related. As in previous cases increasing the resolution or the order of the scheme increases the computational cost and generally improves the accuracy of the scheme. The improvement in the error is large compared to the increased computational cost for schemes up to and including third order. Increasing the order of the scheme beyond this does improve the accuracy but also increases the cost disproportionately. These higher order schemes may still be more cost effective than increasing the resolution of the grid since this can only be done in large increments.

4.5.3 Test Case 1: Advection of a cosine bell over the pole

This is the first of the seven Williamson test cases for numerical approximations to the shallow water equations in spherical geometry. It is included here since it is a test that involves only pure advection. The shallow water model that is used is described in the next chapter. The model is run in this case with the same wind field prescribed at each time step so that only the advection part of the model is used. The method for generating advection schemes that has been developed in this work has been incorporated into this model.

The test is the advection of a smooth cosine bell of radius $R = a/3$ once around the sphere at various orientations of the advecting winds. The cosine bell is defined as

$$h(\lambda, \varphi) = \begin{cases} \frac{h_0}{2}\left(1 + \cos\left(\frac{\pi r}{R}\right)\right) & r < R \\ 0 & r \geq R \end{cases} \tag{4.13}$$

where $h_0 = 1000\text{m}$ and r is the great circle distance of (λ, φ) from $(\lambda_c, \varphi_c) = (3\pi/2, 0)$:

$$r = a\cos^{-1}\left(\sin\varphi_c \sin\lambda + \cos\varphi_c \cos\varphi \cos(\lambda - \lambda_c)\right) \ . \tag{4.14}$$

The wind field is the same as that used for the two previous test, defined by equations (4.12).

Like all the later tests with the shallow water model, four different advection schemes are used. The original scheme of Thuburn [52] is used on the tweaked grid to check that the new grid does not significantly change the solution. A second order scheme is used on grid 6 for a direct comparison with the original scheme. A fourth order scheme is also used on grid 6 to investigate the effect of improving the advection scheme. A second order scheme is also used on grid 7 to compare the accuracy and computational cost with the fourth order scheme in the lower resolution grid. The flux limiter is used in all cases.

Figure 4.7 shows the numerical solution for each scheme (solid contours) along with the true solution (dashed contours). For the second order scheme on grid 6 there is a significant amount of spreading of the cosine bell and the phase speed is slightly slower than that of the true solution. When the order of the polynomial is increased to four there is a lot less spreading of the profile but the phase speed is still a little slow. The peak value of the cosine shape is also better maintained in this case. The same observations can also be made of the second order scheme run on grid 7. It should also be noted that this solution was computationally more expensive to obtain than using the higher order scheme on the lower resolution grid. These features can also be seen in the difference between the true and numerical solutions shown in Figure 4.8.

Figure 4.9 gives the L_1 , L_2 and L_∞ errors for each scheme. The errors for the second order scheme on grid 6 are very similar to those of the original advection scheme used in this model. This scheme was based on a different method of interpolating a second order polynomial over six cells, so the similarity is reassuring in that both methods make reasonable

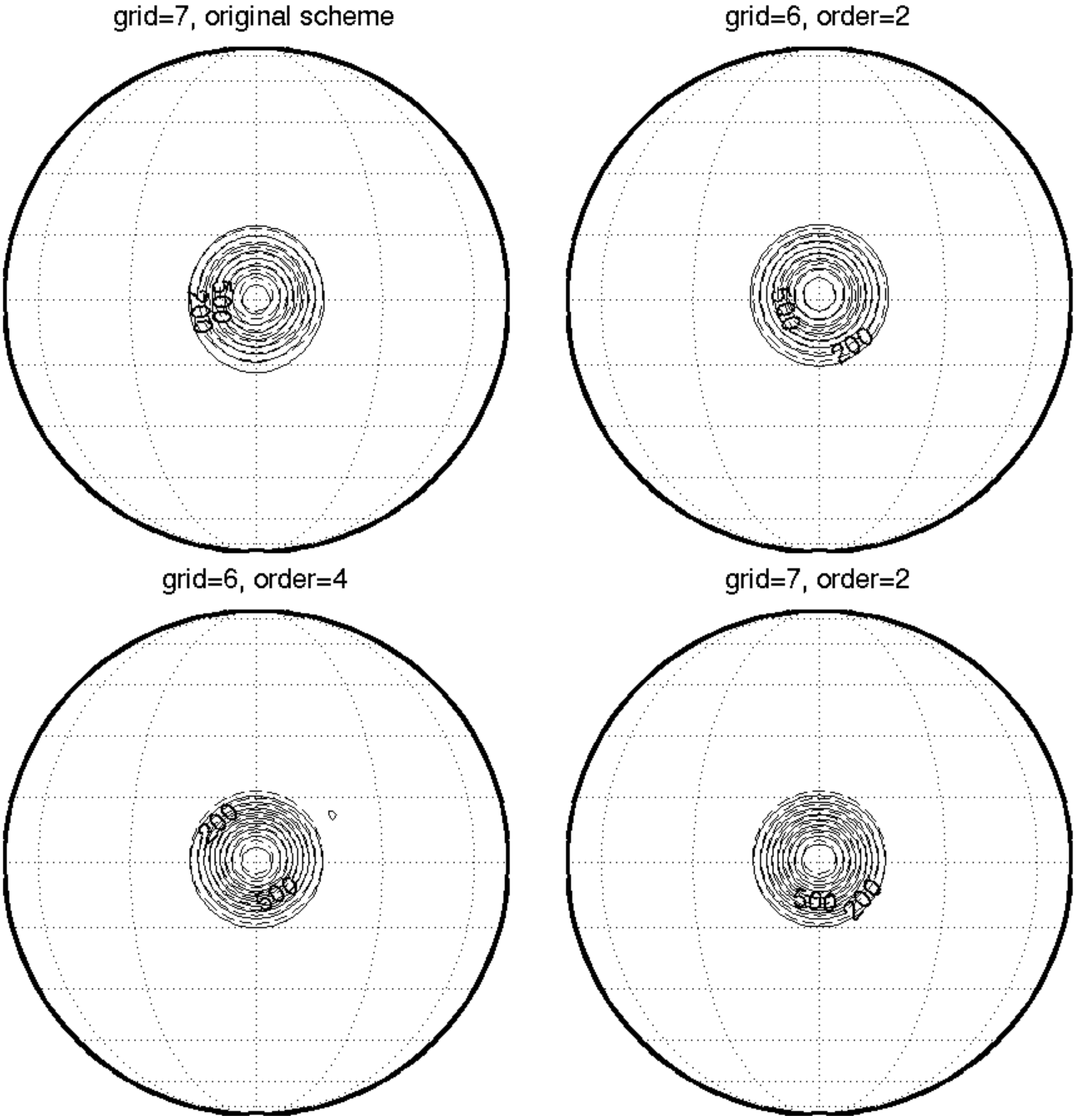


Figure 4.7: Numerical (solid) and exact (dashed) solutions for test case 1 with α =90°, advection across the poles.

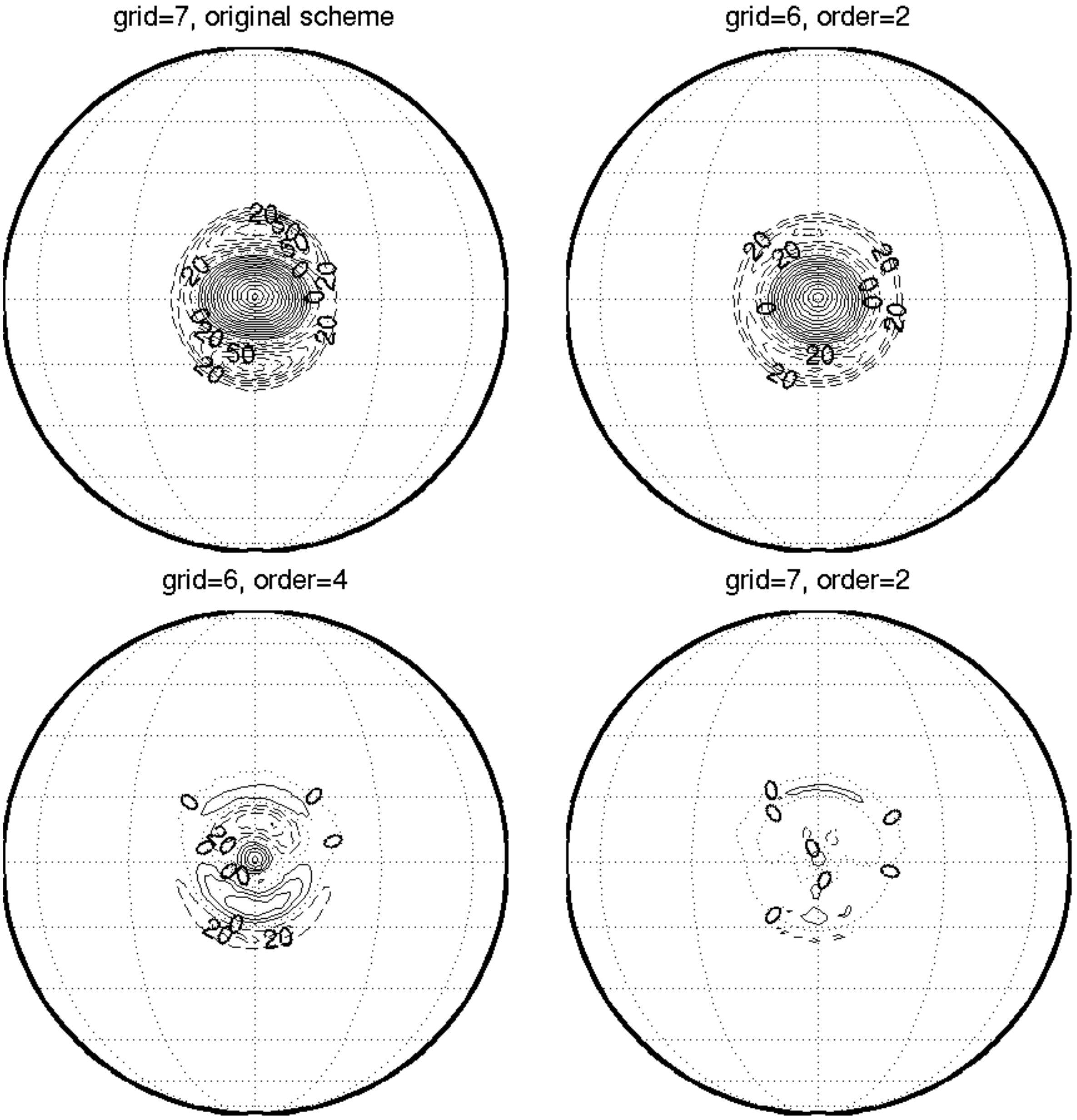


Figure 4.8: Difference between numerical and exact solution for test case 1 with α =90°, advection across the poles.

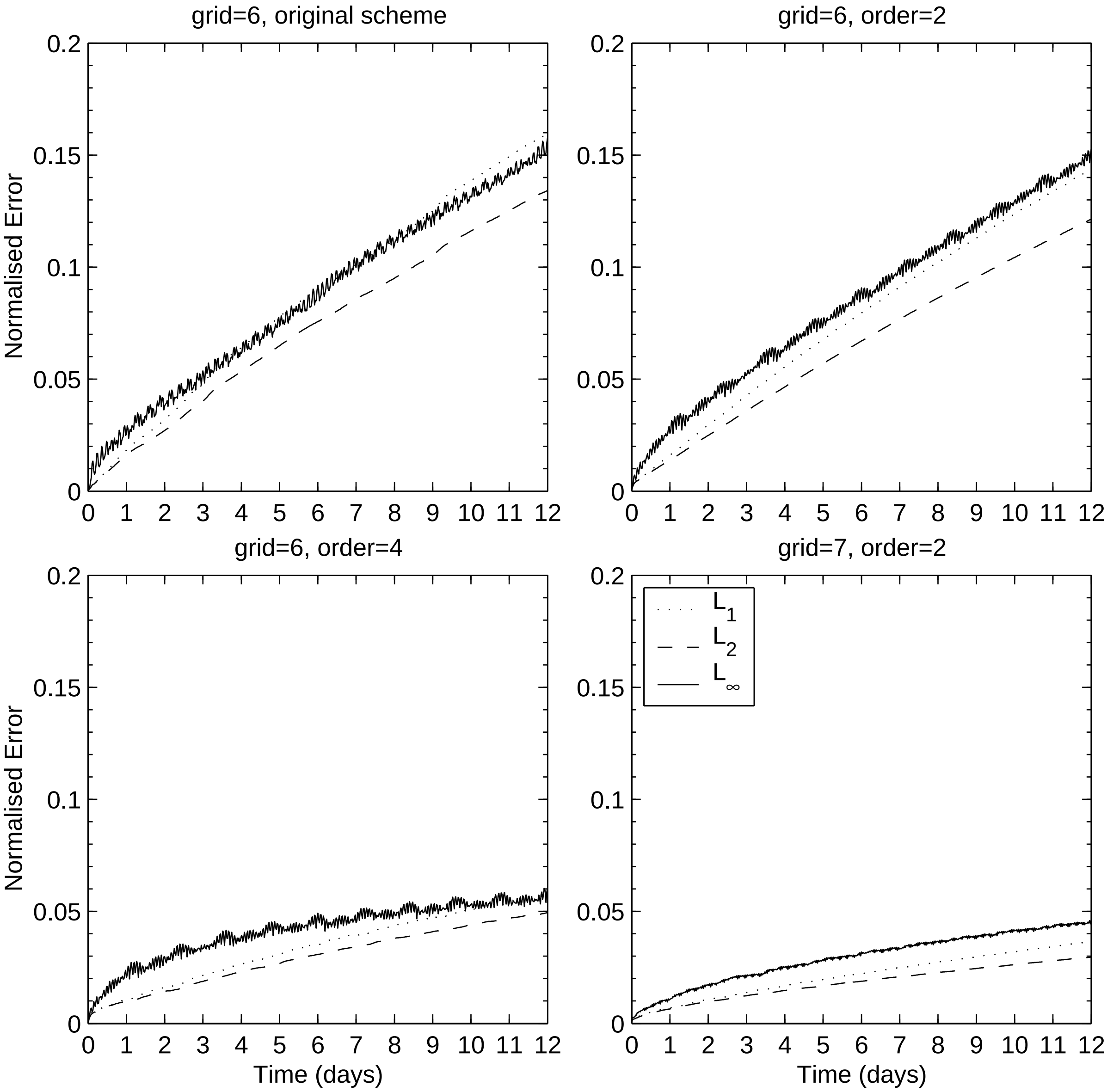


Figure 4.9: Error norms for the same four schemes using test case 1 with α =90°, advection across the poles.

approximations. Increasing the order of the polynomial to four reduces the rate at which the error grows by over one half in this case. These numerical errors show that the second order scheme on grid 7 does give slightly more accurate results than the fourth order scheme on grid 6 though at a much increased computational cost.

All these errors grow smoothly with time showing that the scheme does not encounter problems with any particular part of the grid, i.e. across the poles. The slight oscillations in the L_∞ error occur as the peak of the cosine bell crosses cell boundaries and cell centres. Although the results presented here only show one orientation of the flow, the results are the same for all the other orientations tested, including those of the Williamson test cases.

4.6 Summary

The method developed over the previous chapters has been successfully applied to an icosahedral-hexagonal grid on the sphere. The method itself did not require any adjustments but we did make a local plane approximation to the spherical grid for this to be the case. This idea was investigated in one dimension and shown to make only a small difference to the schemes generated when grid cells were only a few degrees of arc.

The results gained from the application of the method to the icosahedral-hexagonal grid were similar to those of previous tests on square and triangular grids. The changes in accuracy of the results as the order of the scheme is altered behaved in the same way as before. The accuracy improved rapidly as the polynomial order was increased up to three, after which the improvements in accuracy become less as the order is increased further. Coupling the method with the icosahedral grid gives results that are independent of the orientation of the flow relative to the grid.

Using the icosahedral-hexagonal grid has the significant advantage that the results are virtually the same regardless of the orientation of the flow. This is important because it means that significantly larger errors are not being introduced in particular regions of the grid. We have also not had to treat any region of the grid differently from any other as we would have needed to do at the poles of a latitude-longitude grid.

The results presented in this chapter show that the method can be used in a spherical geometry. This means that we now have the flexibility to alter the accuracy and computational cost of advection schemes in the same as we can on a regular grid on the sphere. The next thing to investigate is whether this will have a significant effect in a more complicated model of the atmosphere. Will increasing the accuracy of the advection scheme improve the overall results of the model or will the extra computational effort be wasted? This is investigated in the next chapter where the advection schemes are incorporated into the full shallow water model.

Chapter 5

Shallow Water Model

5.1 Introduction

In the last chapter we applied the method of generating advection schemes developed in earlier chapters to an icosahedral-hexagonal grid on the sphere. The advection schemes generated by the method were shown to achieve reasonable results in pure advection test cases. In this chapter we shall investigate the effects the application of the scheme has on a more complex model, that of the shallow water equations on the sphere. Advection makes up only a part of this model so in order to make a fair test we shall adapt an existing shallow water model to run with the new advection schemes.

The new advection scheme is incorporated into the shallow water model developed by Thuburn [52] and the remaining six Williamson test cases are run (Williamson et al. [59]). The aim of running these tests is to discover if the improved accuracy in the advection routines is carried through to an overall improvement in the model accuracy. If this is the case than it should support the use of PV as a prognostic variable since the advection of PV is important in this case. Investigating the use of PV as a prognostic variable was one of the motivations for the development of the original model.

The original shallow water model will be described in Section 5.2 along with the changes that have been made to incorporate the new advection scheme. Sections 5.3-5.8 will detail the results of the remaining six Williamson test cases. Section 5.9 will discuss these results and what they tell us about the suitability of this scheme for use in meteorological applications.

5.2 The Shallow Water Model

The shallow water equations describe the motion of a fluid of depth h^* over a surface of height h_0. The height of the free surface h, is therefore given by $h = h^* + h_0$, see Figure 5.1. The shallowness comes from assuming that waves on the free surface have a long wavelength L, compared to the average depth of the fluid D. This ensures that the vertical pressure gradient and gravitational forces are close to equilibrium, meaning that the hydrostatic approximation is valid.

The hydrostatic approximation assumes that the gravitational force acting on a fluid parcel is balanced by the force on the parcel due to vertical pressure gradient. If it is also assumed that the fluid is incompressible and viscosity is ignored then the shallow water equations can be derived from the equations of motion for a fluid in a rotating system. The shallow water equations can be written in several different forms, many of which are summarised in Williamson et al. [59]. The form used here has the three equations (in two dimensions) in terms of the fluid depth, the potential vorticity, Q, and the divergence, δ,

$$h_t^* + \nabla \cdot (\mathbf{v}h^*) = 0 \ , \tag{5.1}$$

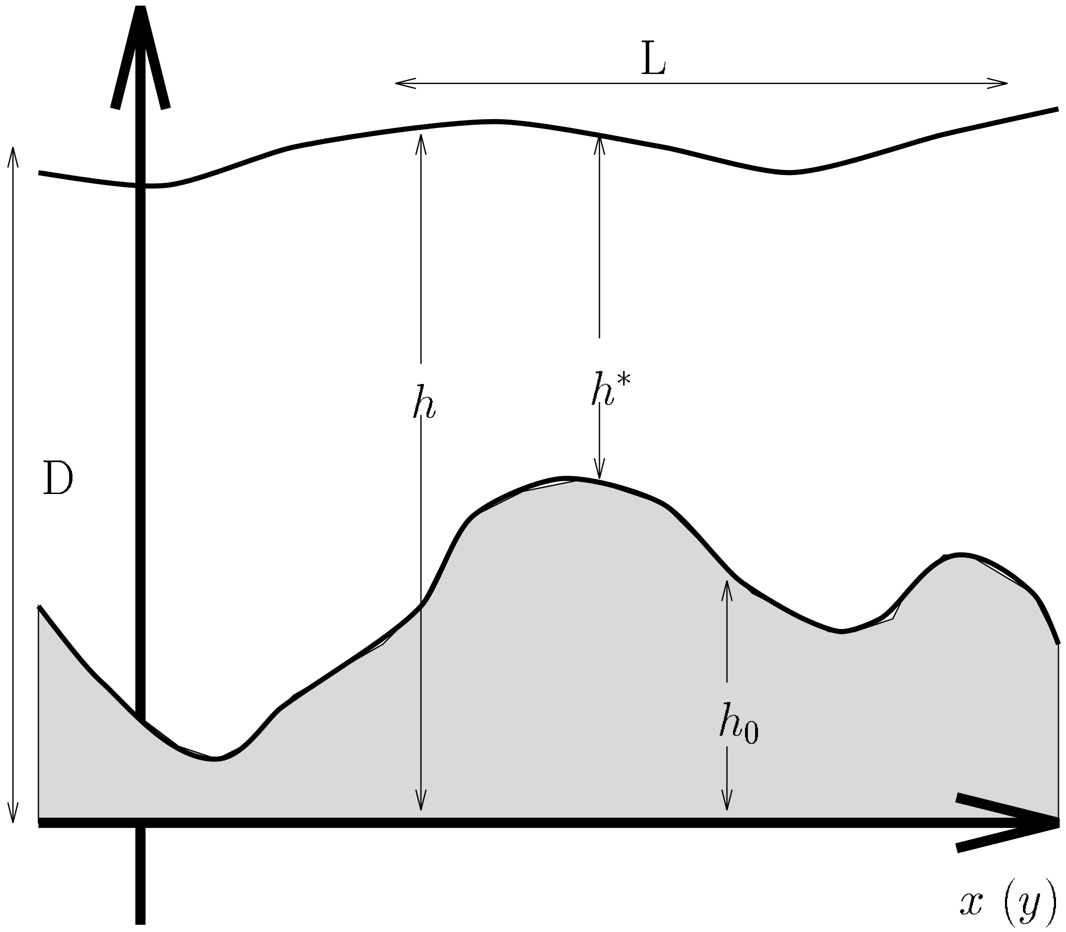


Figure 5.1: Heights and depths in the shallow water system

$$(h^*Q)_t + \nabla \cdot (\mathbf{v}h^*Q) = 0 \ , \tag{5.2}$$

$$\delta_t = -\nabla \cdot \left\{ h^*Q\mathbf{k} \times \mathbf{v} + \nabla \left(gh + \frac{\mathbf{v}^2}{2} \right) \right\} \ , \tag{5.3}$$

where $\mathbf{v}$ is the velocity, g is the acceleration due to gravity and $\mathbf{k}$ is the unit vertical vector.

The basic model used in this work uses potential vorticity, also known as PV, as a prognostic variable. PV is defined in this system as

$$Q = \frac{f + \zeta}{h^*} \ , \tag{5.4}$$

where $\zeta = \mathbf{k}.\nabla \times \mathbf{v}$ is the relative vorticity. ζ is a measure of the rotation in the flow relative to the rotating co-ordinate system. If the system is rotating at a rate Ω then f, the Coriolis parameter at latitude φ, is defined as $f = 2\Omega \sin \varphi$. PV has a number of useful properties in the study of atmospheric dynamics;

i. It is conserved following fluid parcels as far as friction and adiabatic effects are negligible, in that sense it can be thought of as labelling air parcels;

ii. It can be 'inverted' to give wind and mass fields under suitable balance assumptions, i.e. information about balanced parts of the flow is contained in the PV;

iii. Even when adiabatic effects and friction are not negligible, it satisfies a conservation law of the form

$$(h^* Q)_t + \nabla . \mathbf{F} = 0 \ , \tag{5.5}$$

for some flux $\mathbf{F}$ in the shallow water system (a similar conservation law holds for three dimensional flows governed by the primitive equations),

iv. Its behaviour can be thought of as if some substance of mixing ratio Q were confined to move along isentropes[1], see Hoskins et al. [14], Haynes and McIntyre [9], McIntyre [32].

Using PV as a prognostic variable to separate the balanced and unbalanced parts of a flow should enable the use of the most effective numerical method for each part. The use of PV as a prognostic variable in this way has been limited but has been investigated by Bates et al. [1], Ringler et al. [37] and was one of the reason for the development of the original model by Thuburn [53]. Thuburn [53] suggests that;

[1] surfaces of equal potential temperature

> "One possible reason for this may be the lack of adequate advection schemes."

PV has also recently been used by Wlasak [60] as a diagnostic tool to describe balance in data assimilation.

The model used in this work is the same as that used by Thuburn [52, 53]. The original advection scheme used in that model was based on a quadratic interpolation over regular hexagons that is third order accurate on the plane. This method is, like the new schemes we shall use, only formally first order accurate on the grid used. Tests have been run using advection schemes based on second and fourth order polynomial interpolation using the new method. As before, the new schemes are referred to according to the order of the polynomial used for interpolation, not the formal order of accuracy.

The remainder of the model is identical to that of Thuburn described in detail in [52], only a brief description of the rest of the model is given here. The Laplacian of a field, ϕ, is given by,

$$\frac{1}{A_k}\sum_i \frac{e_i}{d_i}(\phi_i - \phi_k) \tag{5.6}$$

where i runs over the edges and neighbouring cells of cell k, e_i and d_i are the length of edge i and distance between the centers of cells k and$i$. This is a second order accurate approximation on regular hexagons but is less accurate on the distorted hexagons of this grid. Poisson's equation is solved using a multigrid elliptic solver that uses the hierachy of grids described in the last chapter. A semi-implicit time step is used which introduces a number of subtleties that are dealt with in [52].

The other difference from the results reported in Thuburn [52, 53] is that the grid was 'tweaked' (but not 'twisted') as described in the last chapter and by Heikes and Randall [11] for these tests. The results of using the original method on the new grid are shown here to distinguish between the changes due to the advection scheme and those caused by the grid. Overall, the change to the grid made little difference to the overall results, what differences there were will be discussed with the relevant test cases.

The same code as used by Thuburn in [52] was used here with a number of adaptions. The main changes were the addition of a subroutine outside the time stepping loop to set up the matrix G and the replacement of the subroutine that generated the fluxes with one using the new method. The flux limiter of the original code was the same as that described in previous chapters and was retained. The only other adaptions to the code were minor changes made for a different formatting of the output and for compatibility with Fortran 90.

All the tests have been run on grids 5, 6 and 7 using second and fourth order advection schemes as well as the original scheme. The results for each of the schemes on grid 6 and the second order scheme on grid 7 are presented here for each test case. The remaining tests are used for convergence tests and to compare timings. These schemes have been chosen because they were amongst the most efficient in earlier tests when comparing accuracy and computational cost. A time step of 30 minutes was chosen for all tests wherever possible. For the fast flows in test case 6 and some of the grid 7 tests a shorter time step was required by the flux limiter, these will be detailed with the individual cases. These occasions will be highlighted when applicable.

The computational cost of using the schemes varies with the number of cells and edges on the grid and with the order of the polynomial used.

Increasing the resolution by one grid level roughly quadruples the number of edges and cells causing an similar increase in the computational cost of the entire model. This increase in cost will be larger still if a smaller time step is required on the finer grid. Increasing the order of the polynomial used in the advection scheme increases the number of cells in the stencil which is the main factor in the cost of the scheme. Increasing from second to fourth order in this case roughly doubles the computational cost of just the advection scheme, a much smaller cost increase than increasing the resolution. This should be remembered when comparing the results of the following test cases.

5.3 Test Case 2:

5.3.1 Global steady state nonlinear zonal geostrophic flow

This case consists of a solid body rotation with a corresponding geostrophic height field, a steady state of the non-linear shallow water equations. The test is run with the poles of the solid body rotation at four different angles to the poles of the grid, $\alpha = 0.0,\ 0.05,\ \pi/2 - 0.05,\ \pi/2$. The streamfunction and velocity potential are then given by

$$\psi = -\, a u_0 \left(\sin\varphi\cos\alpha - \cos\lambda\cos\varphi\sin\alpha\right) \ , \qquad (5.7_\psi)$$

$$\chi = 0 \ . \qquad (5.7_\chi)$$

The corresponding analytic height field is given by

$$gh = gh_0 - \left(a\Omega u_0 + \frac{u_0^2}{2}\right)\left(-\cos\lambda\cos\varphi\sin\alpha + \sin\varphi\cos\alpha\right)^2 \ . \qquad (5.8)$$

The Coriolis parameter is a function of latitude and longitude to enable the axis of rotation to be changed;

$$f = 2\Omega\left(-\cos\lambda\cos\varphi\sin\alpha + \sin\varphi\cos\alpha\right) \ . \qquad (5.9)$$

The other parameter values used are $u_0 = 2\pi a/12$ and $gh_0 = 2.94 \times 10^4$ $\mathrm{m}^2\mathrm{s}^{-2}$.

Since this is a steady state of the shallow water equations we would want our scheme to maintain a solution close to the initial conditions. By discretising the equations onto the icosahedral-hexagonal grid we may no longer have an exact steady state. It would be hoped that there is a steady state of the discrete equations close to that of the continuous case and that the model does not diverge significantly from this state.

5.3.2 Results

The original icosahedral model displayed the five fold symmetry in the grid in the error field when the axis of the solid body rotation was close to that of the grid. The global error measures of the height and wind fields were similar for all the angles of flow relative to the grid. The errors in the spectral model of Jakob et al. [15] are very small but show a large increase in the errors close to the poles of the grid. The error also fluctuates in space between being positive and negative and is also much larger when the rotation and grid poles are aligned than when they are at right angles. The results for the original scheme in this case vary slightly from those presented by Thuburn in [52] and [53] because the tweaked grid is used in these tests, the un-tweaked grid was used previously.

Figure 5.2 shows the difference between the true and numerical solutions for the four schemes mentioned above. There is little obvious difference between the result for the original scheme on the tweaked grid shown here and the un-tweaked grid shown in figure 17 of [52]. Using the new second order scheme on grid 6 produces a different pattern of the error but one which still displays the five-fold symmetry of the grid. The

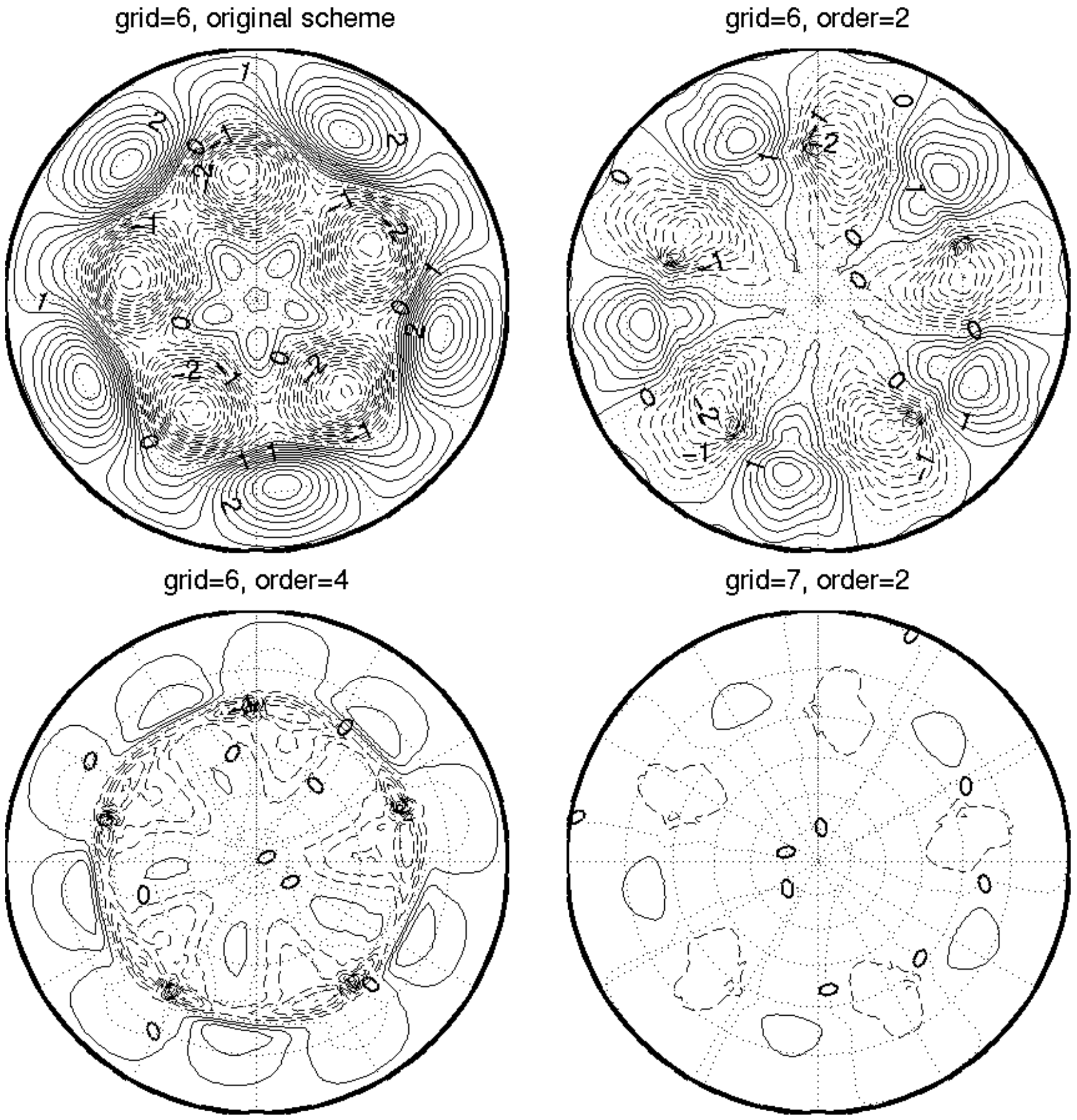


Figure 5.2: Difference between true and numerical height fields for the northern hemisphere after a five day run of test case 2 ($\alpha = 0$). Contour interval is 0.25 m.

maximum values of the error still correspond with the pentagonal cells in this case but have a slightly smaller magnitude. The same comments can be made when comparing the errors of the second and fourth order schemes. The pattern of error is different again but the five fold symmetry is still present and the peak errors occur over the pentagons. The second order scheme run on grid 7 shows a similar pattern of errors to that on grid 6 but greatly reduced in magnitude.

The five-fold symmetry of the errors is not unexpected because of the five-fold symmetry of the grid. The greatest irregularity in the size of the grid cells occurs at the pentagons so we would expect both the greatest difference between the continuous and discrete steady states and the poorest approximation to

$$\frac{\partial()}{\partial t} = 0 \ , \tag{5.10}$$

to occur here. This would lead to the larger errors near the pentagonal cells, particularly in a flow where each lies in a region upwind of another. The advection scheme being used will affect the numerical steady state solution. The difference between the three different cases on grid 6 show that the interactions between the grid and the advection scheme can cause significant differences in the results.

The time evolution of the normalised global error norms for the height field is shown in Figure 5.3. The errors from the original scheme are slightly larger than those from the un-tweaked grid shown in Thuburn [52], figure 18 . There are fewer oscillations in the error as the model settles from the continuous balanced state to the model one on the tweaked grid. These oscillations are more in evidence when the new second order scheme is used but overall the error grows more slowly using the new

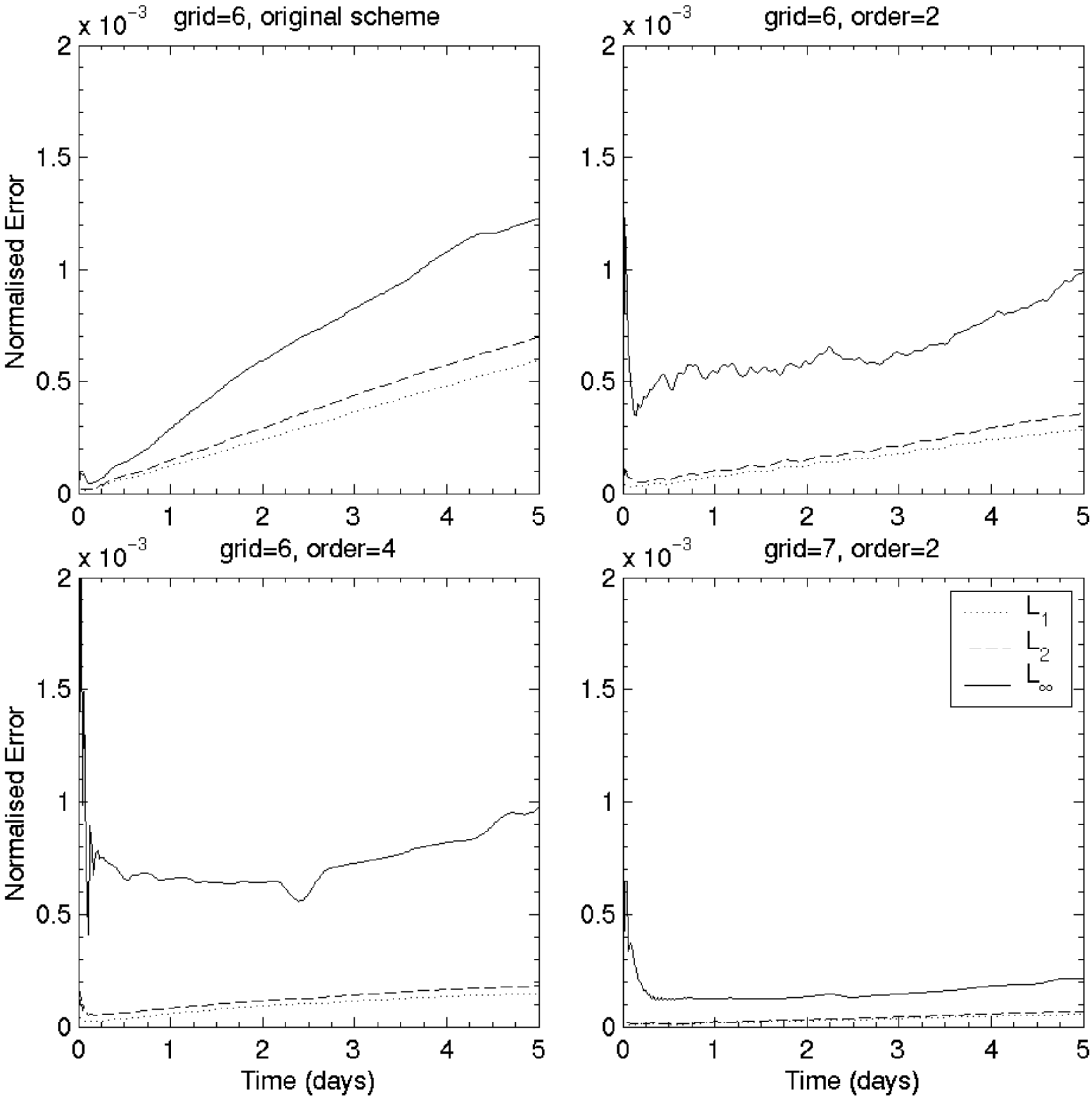


Figure 5.3: Normalised global errors in the height field for a five day run of test case 2 ($\alpha = 0$). Errors sampled at each time step (30 mins.).

scheme. These L_1 and L_2 errors are roughly halved when the order of the scheme is increased to four, and quartered if the grid resolution is increased to grid 7 instead. The L_∞ error shows a similar decrease in value when the resolution is increased but is similar in both the second and fourth order cases on grid 6.

This is more evidence for the peak errors being an artifact of the grid used and not the advection scheme. The oscillations in the errors shortly after the start of the run are caused by fast moving gravity waves, which act to restore geostrophic balance. The discrete balanced state on the more refined grid is closer to the continuous balanced state so the effect is smaller in this case. The higher order, less diffusive advection scheme has a smaller effect on the the gravity waves in the advection step than the lower order scheme. This accounts for the higher errors initially and the faster settling of the model into its balanced state. The smaller diffusive effect of the higher order scheme causes the errors to grow more slowly, as does the even smaller diffusion in the second order scheme on grid 7.

The errors in the diagnosed wind field are shown in Figure 5.4 in the same form as the height errors were displayed. The errors in this case behave in a similar fashion to those of the height error. They are slightly larger for the original scheme on the tweaked grid than the un-tweaked grid. There is a steady increase in the L_1 and L_2 norms with time but this is slowed by using the higher order scheme or a higher resolution grid. The L_∞ error initially displays some of the short time behaviour of the height field though it is not so pronounced. The larger L_∞ error in the fourth order case may be due to this scheme better approximating the discrete steady state where it is significantly different from the true solution. This is not a failing of the advection scheme itself.

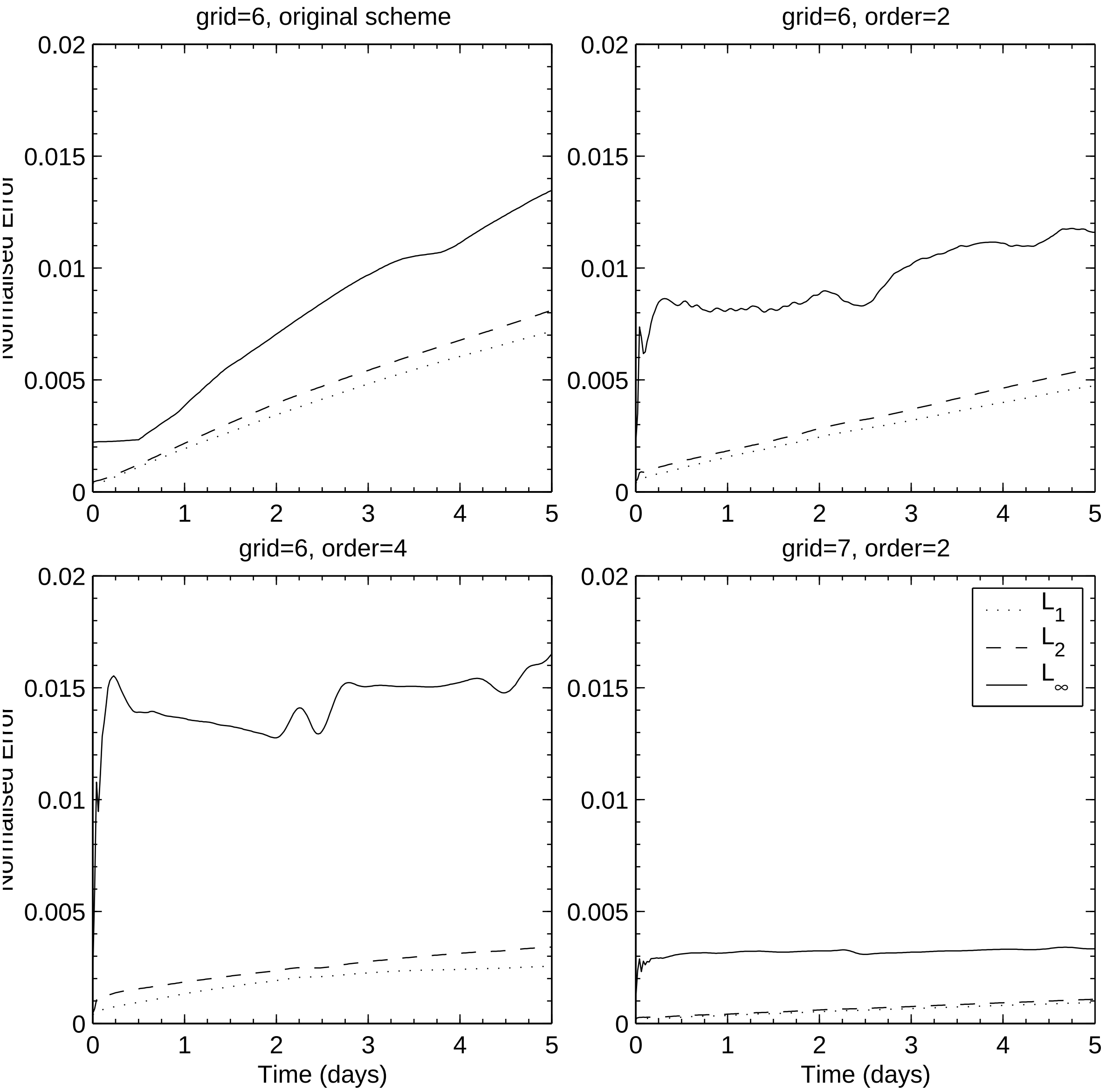


Figure 5.4: Normalised global errors in the wind field for a five day run of test case 2 ($\alpha = 0$). Errors sampled at each time step (30 mins.).

The global height and wind errors are very similar to these results for different angles between the rotational and grid poles. The five-fold symmetry is distorted when $\alpha = 0.05$ and destroyed when α is close to $\pi/2$, as in the original model. The magnitude of the errors when using the fourth order scheme is comparable with the model of Heikes and Randall [12] at this resolution. Where the errors in this case were growing, those of Heikes and Randall oscillate in time about a uniform value. The spectral model of Jakob et al. [15] performs exceptionally well for this test in terms of the global error measures since the initial conditions have an exact spectral representation. Despite this there are still unwanted features in the error fields. The error increases significantly near the poles and the error field is highly oscillatory. These are not significant problems in this case but they do highlight some shortcomings of the spectral model.

5.4 Test Case 3:

5.4.1 Steady state non-linear zonal geostrophic flow with compact support

This case is similar to the previous one but with the wind field only non-zero north of 30°S. The streamfunction and velocity potential are defined as

$$\psi'\left(x'\right) = -\, a\frac{\varphi_a - \varphi_b}{x_e} u_0 \mathrm{e}^{\frac{4}{x_e}} \int_{x_e}^{x'} \mathrm{e}^{-\frac{x_e}{x(x_e - x)}}\, \mathrm{d}x \ , \qquad (5.11_\psi)$$

$$\chi' = 0 \ , \qquad (5.11_\chi)$$

where a primed variable is one in a coordinate system for which the axis of rotation of the flow and the earth are coincident. The parameters used are $u_0 = 2\pi a/(12 \text{ days})$, $\varphi_b = -\pi/6$, $\varphi_e = \pi/2$ and $x_e = 0.3$. For a flow rotating at an angle α to the earths rotation the coordinate systems are related by

$$\sin\varphi' = \sin\varphi\cos\alpha - \cos\varphi\cos\lambda\sin\alpha \ , \tag{5.12}$$

$$\sin\lambda'\cos\varphi' = \sin\lambda\cos\varphi \ . \tag{5.13}$$

The correct quadrant for λ' can be found by ensuring

$$\sin\varphi = \sin\varphi'\cos\alpha + \cos\varphi'\sin\alpha\cos\lambda' \ . \tag{5.14}$$

The Coriolis parameter for the two systems is

$$f = 2\Omega\left(-\cos\lambda\cos\varphi\sin\alpha + \sin\varphi\cos\alpha\right) \ , \tag{5.15}$$

$$f' = 2\Omega\sin\varphi' \ . \tag{5.15$'$}$$

The analytic height field in the prime coordinate that gives a steady state solution must satisfy

$$\frac{(u')^2\tan\varphi'}{a} + \frac{g}{a}\frac{\partial h'}{\partial\varphi'} + fu' = 0 \ . \tag{5.16}$$

The analytic height field is difficult to find exactly from this equation, instead numerical integration is used to find an accurate approximation to h,

$$h = h_0 - \frac{a}{g}\int_{-\frac{\pi}{2}}^{\varphi'}\left(2\Omega\sin\tau + \frac{u'(\tau)\tan\tau}{a}\right)u'(\tau)\mathrm{d}\tau \ . \tag{5.17}$$

The background height is again given by $gh_0 = 2.94 \times 10^4$ m^{2}s^{-2}.

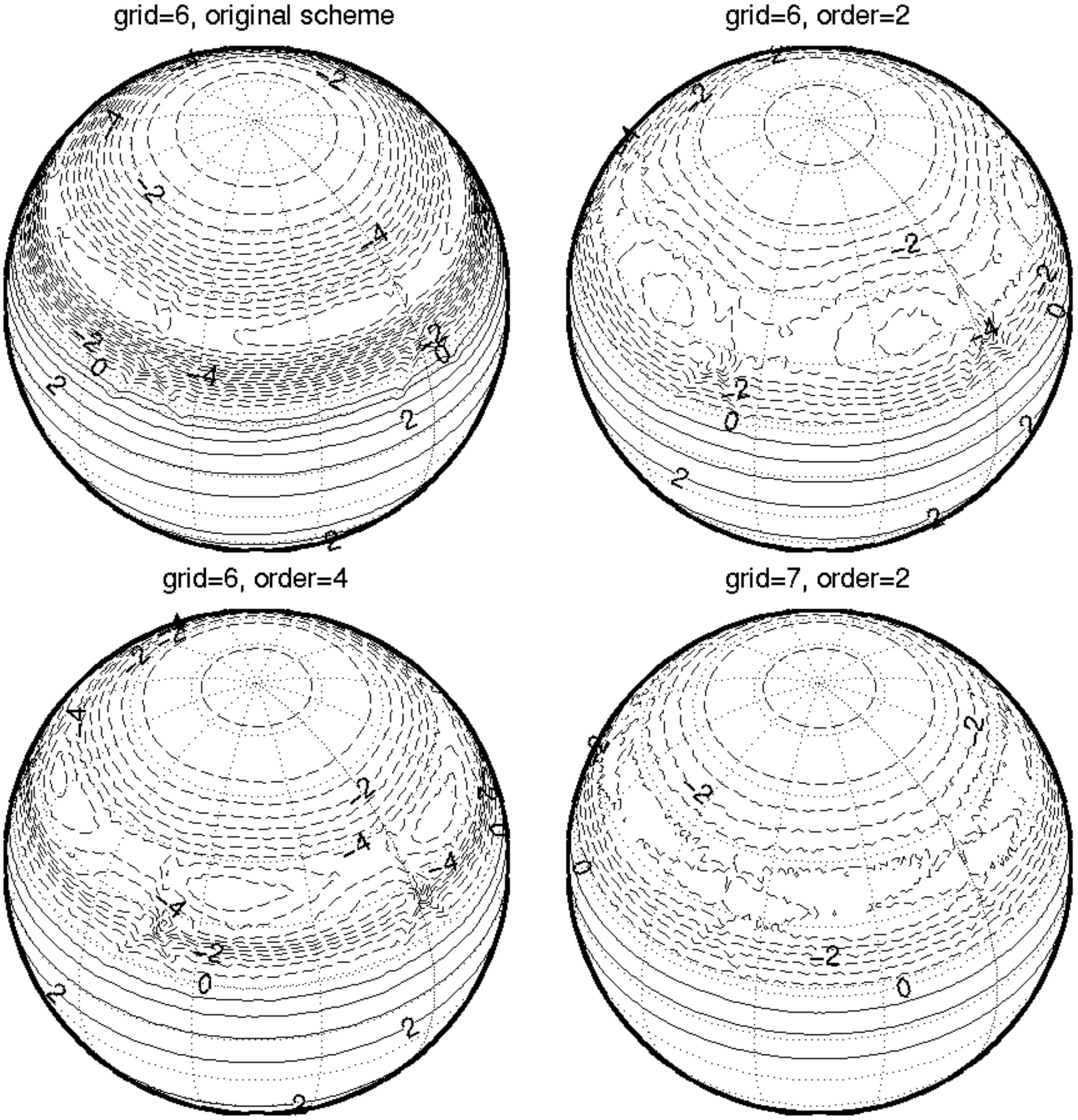
5.4.2 Results

The results for the original model again showed the five fold symmetry of the grid when the axis of the the flow coincided with that of the grid. This was also present but much less evident on grid 6. The h and u errors oscillated in time

> "... presumably about a balanced state of the discrete equations which is not exactly the same as the initial state." (Thuburn [52])

Figure 5.5 shows the difference between the true and model solutions for the same four advection schemes as test case 2. The original scheme used with the tweaked grid shows no obvious difference from using the scheme on the un-tweaked grid. When the new second order scheme is used the difference between the true and numerical solutions is smaller but the five fold symmetry is much clearer to see in a broader region in the error field. If the fourth order scheme is used the height differences at day five are larger than those for the second order scheme. A possible reason for this can be seen in the time evolution of the global error measures which oscillate in time with slightly different phases. When the second order scheme is used on grid 7 the symmetry of the grid is less visible and the overall error is smaller

Figure 5.6 shows the global L_1 , L_2 and L_∞ errors in the height field for these four runs. All cases exhibit the oscillations in the errors seen in the original model. The change in the grid has made little change to the height errors when the original scheme is used. The oscillations in the error are more irregular on the tweaked grid but have a similar magnitude and period. The oscillations in the global error are very similar in form when the new second order scheme is used but the error is smaller in magnitude. Using the fourth order scheme reduces the amplitude of the oscillations slightly but does not improve the magnitude of the errors. Increasing the resolution of the grid does reduce the magnitude of the error but the oscillatory nature of the errors and even the pattern of the oscillations are very similar.



grid=6, original scheme

grid=6, order=2

grid=6, order=4

grid=7, order=2

Figure 5.5: Difference between true and numerical height fields after a five day run of test case 3 ($\alpha = 0$). Plots are centred on $(\frac{3\pi}{2}, \frac{\pi}{2})$ Contour interval is 0.5 m.

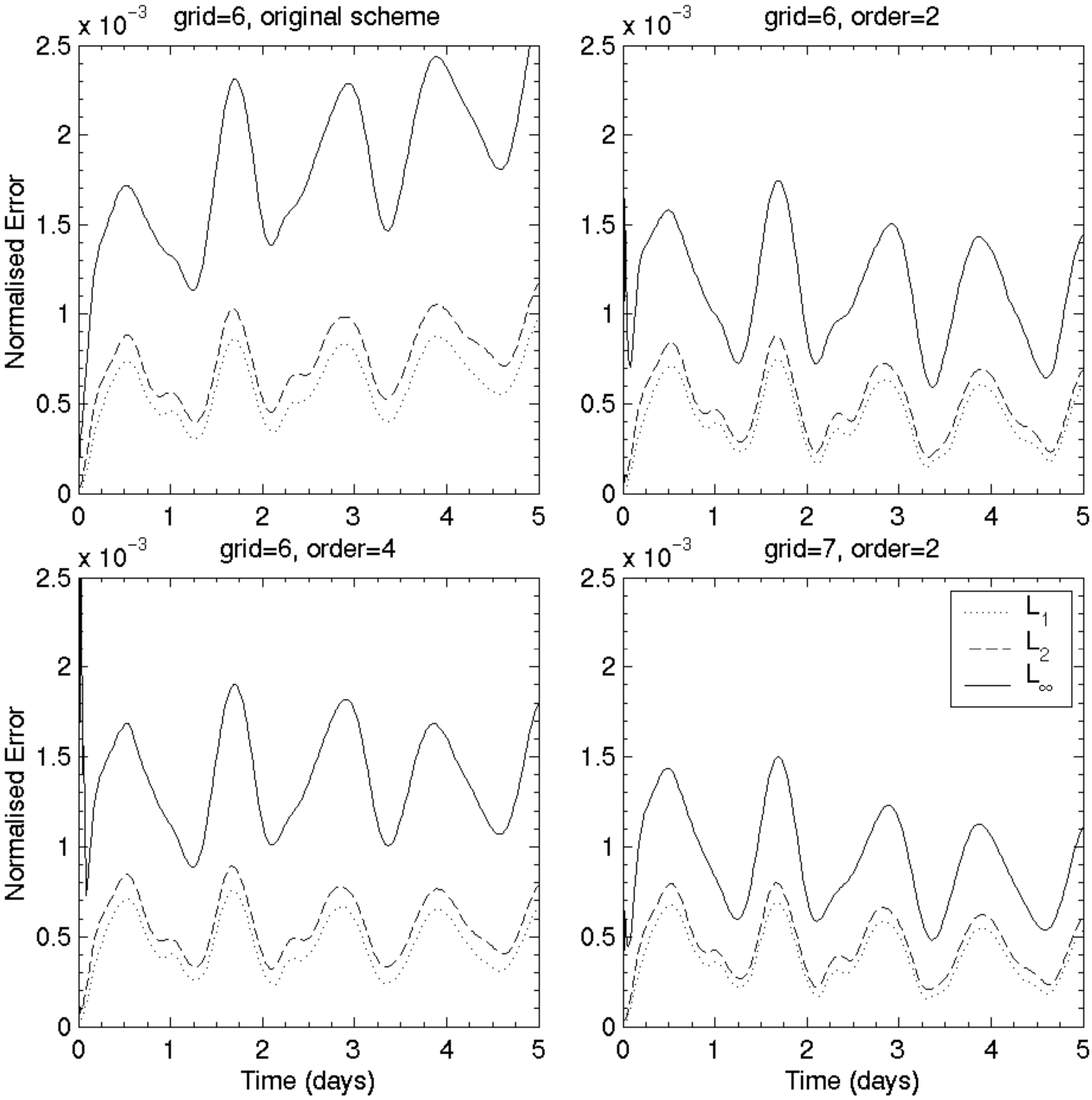


Figure 5.6: Normalised global errors in the height field for a five day run of test case 3 ($\alpha = 0$). Errors sampled at each time step (30 mins.).

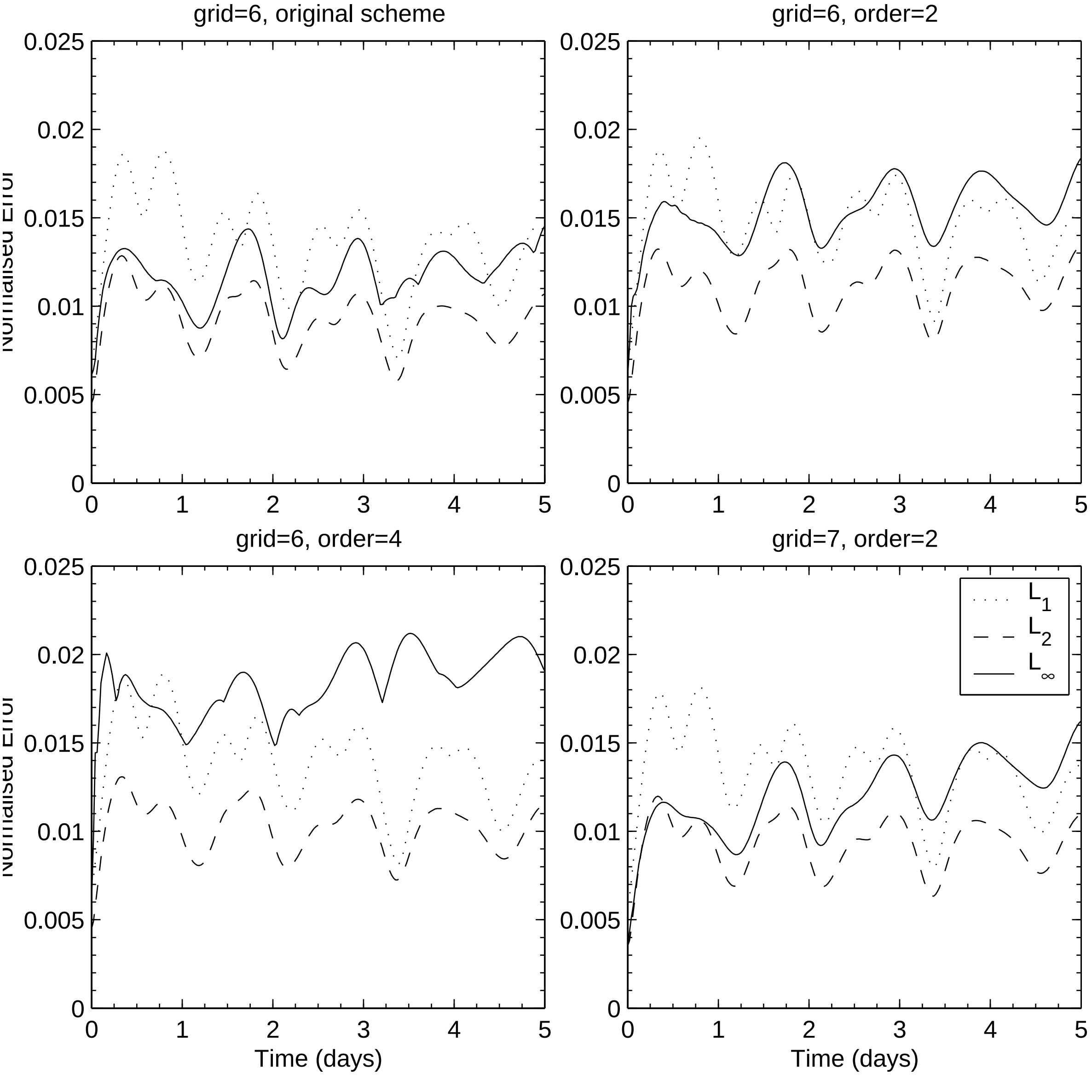


Figure 5.7: Normalised global errors in the wind field for a five day run of test case 3 ($\alpha = 0$). Errors sampled at each time step (30 mins.).

Figure 5.7 shows these same error measures for the wind field. These are also oscillatory in nature and again show the same patterns and magnitudes in the L_1 and L_2 errors for each different advection scheme. This strongly suggests that these errors are not due to errors caused by the advection scheme. The similarity of the results on grid 7 further suggests that these errors are not related directly to the grid resolution. Two possibilities for the cause of this error are the time truncation errors in the divergence equation and initial errors in the discrete height field. The initial discrete height field is unlikely to give a steady state solution, the model state may therefore be oscillating around some discrete steady state. The errors in modelling the small amount of divergent flow caused by the discrete height field may also be causing the errors. Comparing the advection schemes in this case using figures 5.6 and 5.7 cannot therefore go much beyond stating that they all do a good enough job not to seriously affect the results of this test case.

The results for these schemes are very similar when the angle α is changed to $\pi/3$. The main difference is the disappearance of the five fold symmetry in the solutions. When the angle α is taken to be 30° there is a weak four fold symmetry in the pattern of the errors. This is caused by the alignment of the flow with four of the pentagonal cells in this orientation.

Comparing this model with that of Heikes and Randall [12] suggests that there is a significant difference between the two models. Whilst the L_∞ error is slightly larger and more oscillatory in the Heikes and Randall model, the L_1 and L_2 errors are much smaller. The peak errors occured where there was the greatest change in cell sizes in the Heikes and Randall model, where it was suggested that changing the grid so that the cell areas vary more smoothly should reduce the error. The tweaked grid has the effect of smoothing the variations in cell sizes and the peak values are smaller in this case. This support for the suggestion of Heikes and Randall is diminished somewhat by the other, larger changes in the errors between the two models. These could be caused by the different numerical schemes being used or by the different formulations of the shallow water equations. It is most likely a combination of these two factors. Comparison with the Spectral Model is again difficult due to the highly accurate representation of the initial conditions which is only limited by round-off error.

5.5 Test Case 4:

5.5.1 Forced non-linear system with a translating low

Having tested the models on non-linear steady states of the shallow water equations we now move on to tests of unsteady states of the equations. Analytic solutions of unsteady problems are difficult to come by but we are able to find an analytic solution to the forced shallow water system when forcing terms of a certain form are used. The forcings required for describing a flow of a translating low in a mid-latitude westerly jet are given in advective form in Williamson et al. [59]

Converting the forcing of the wind in Cartesian form, $\boldsymbol{F}_v$, to normal and tangential components at box edges, the forcings of the PV, F_Q, and divergence, F_δ, are given by

$$F_Q = \frac{\nabla \times \mathbf{F}_v - QF_h}{h^*} \,, \tag{5.18}$$

$$F_\delta = \nabla \cdot \mathbf{F}_v \tag{5.19}$$

where F_h is the height forcing given along with F_u and F_v in Williamson et al. [59]. The forcings of height, PV and divergence are applied to the fields at each time step. Numerical approximations to the curl and divergence operators are used as would occur in a real application of the model.

The time varying flow is given by

$$\begin{aligned}\tilde{u} =& \bar{u} - \frac{\bar{\psi}_\varphi}{a} \ , && (5.20)\\ \tilde{v} =& \frac{\bar{\psi}_\lambda}{a\cos\varphi} \ , && (5.21)\\ g\tilde{h} =& g\bar{h} + f\bar{\psi} \ , && (5.22)\end{aligned}$$

where

$$\begin{aligned}\bar{u} =& u_0 \sin^{14}(2\varphi) \ , && (5.23)\\ g\bar{h} =& gh_0 - \int_{\frac{\pi}{2}}^{\varphi} \left(af(\tau) + \bar{u}(\tau)\tan\tau\right)\bar{u}(\tau)\mathrm{d}\tau \ , && (5.24)\\ \bar{\psi}(\lambda,\varphi,t) =& \psi_0 \mathrm{e}^{-\sigma\left(\frac{1-C}{1+C}\right)} \ , && (5.25)\end{aligned}$$

with parameters $\psi_0 = -0.03(gh_0/f_0)$, $\sigma = (12.74244)^2$, $gh_0 = 10^5$, $f_0 = 2\Omega\sin(\pi/4)$, and

$$C = \sin\varphi_0 \sin\varphi + \cos\varphi_0 \cos\varphi \cos\left(\lambda - \frac{u_0}{a}t - \lambda_0\right) \ . \qquad (5.26)$$

The streamfunction and velocity potential that this gives are

$$\begin{aligned}\psi =& -\int_{-\frac{\pi}{2}}^{\varphi} a\bar{u}(\tau)\,\mathrm{d}\tau + \bar{\psi}(\lambda,\varphi,t) \ , && (5.27_\psi)\\ \chi =& 0 \ . && (5.27_\chi)\end{aligned}$$

The low centre is initially located at $(\lambda_0, \varphi_0) = (0, \pi/4)$, in the jet prescribed by equation (5.23). Tests are run for values of u_0 of 20 and 40 ms^{-1}, corresponding to a westward translation of the low of about 15.5° or 31° per day.

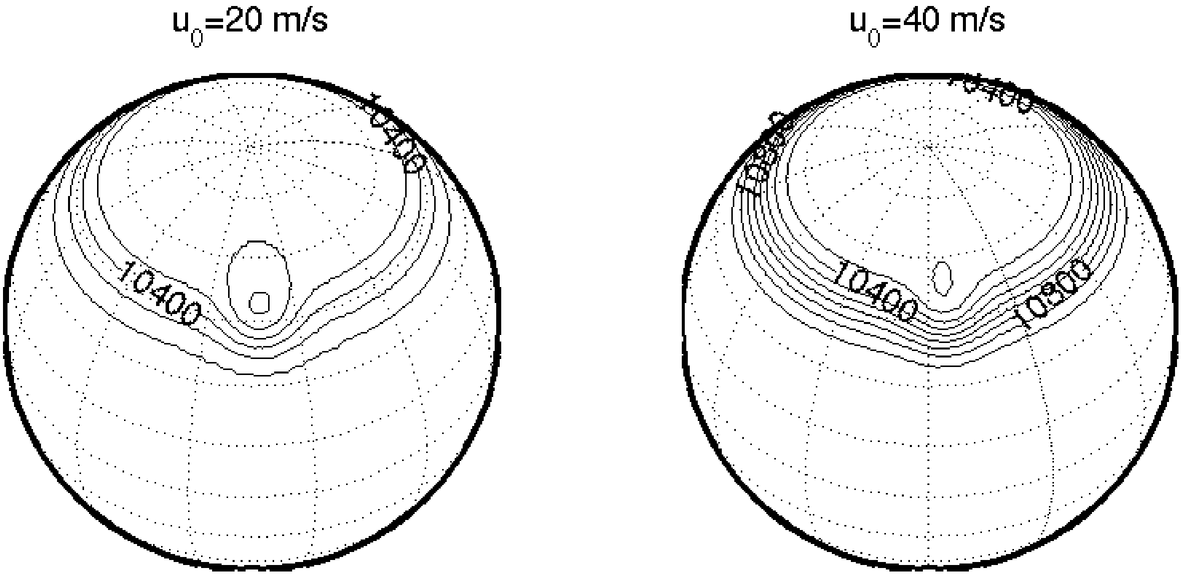


Figure 5.8: Exact solutions for test case of a forced non-linear system of a translating low after 5 days. Plots are centred on longitude of low centre. Contour interval is 100m

5.5.2 Results

Thuburn [52] suggested that the main error in the original model was the filling of the low centre

> "... consistent with a smoothing of the PV maximum (and to some extent the depth minimum) by the advection scheme."

An improved advection scheme should therefore cause less smoothing of the PV field and less filling of the low centre.

The exact solutions for both cases are shown in Figure 5.8 after 5 days. This plot and all others in this section have been rotated to the longitude of the low centre. Looking at the numerical solutions of the height fields shows little difference between the different schemes so instead we shall look at the differences between the true and numerical solutions shown in figures 5.9 and 5.10.

Figure 5.9 shows this difference for the case when $u_0 = 20$ ms^{-1} using the same four schemes used in previous tests. The results for the original

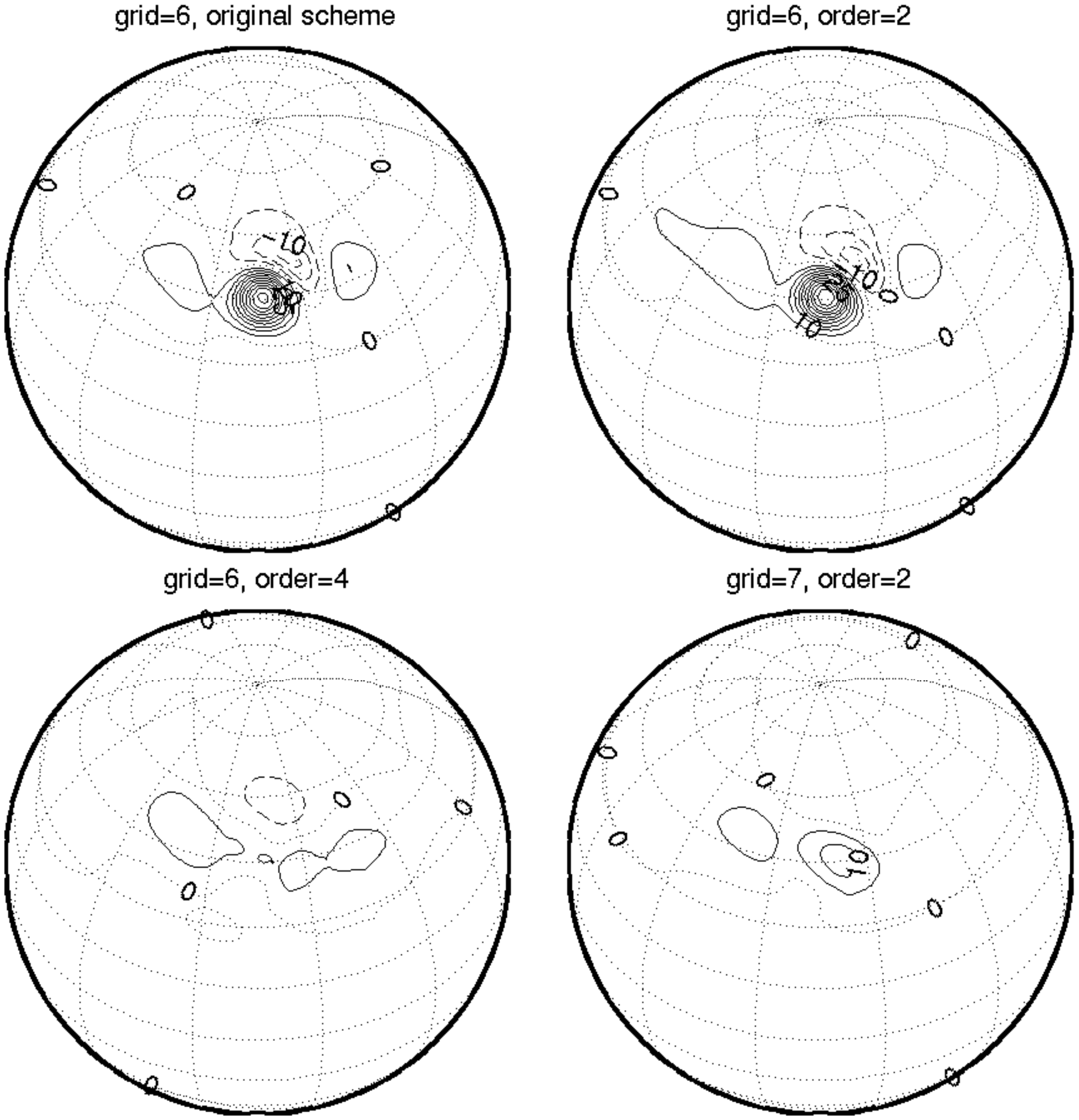


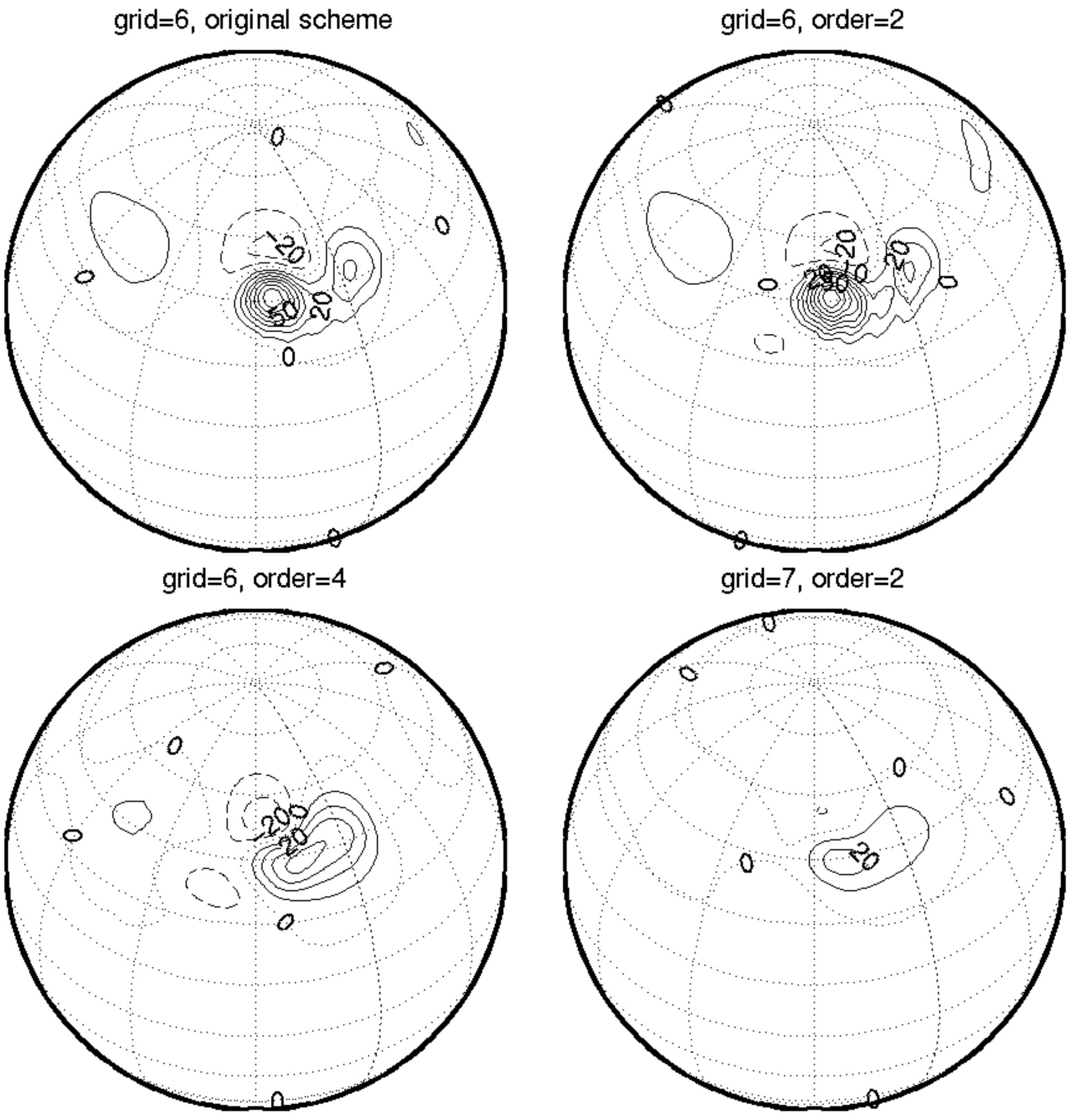
Figure 5.9: Difference between true and numerical height fields after a five day run of test case 4 ($u_0 = 20$ ms^{-1}). Contour interval is 5 m.

scheme on the tweaked grid shows a slight reduction in the filling of the low centre. This may be due to the grid allowing a better approximation of the forcing terms in this case. Using the new second order scheme on the same grid makes very little difference to these results. There is a very small increase in the filling of the low but little else that can be seen in these plots.

Increasing the order of the scheme to four makes a large improvement in the final solution with almost no filling of the low centre. The pattern of errors around the low centre is very similar but is also much reduced from the other cases. This improvement in the fourth order scheme is consistent with a reduced smoothing of the PV maximum (and depth minimum) by the more accurate advection scheme. Using the second order scheme on grid 7 gives a slight filling of the low but little other error.

The results for the case with $u_0 = 40$ ms^{-1} are shown in Figure 5.10. The patterns of errors and the differences between the various cases on grid 6 are largely similar. The main point of note is that the errors are significantly larger in this case. The results for the grid 7 case are better than any of those on grid 6. This may be a result of the increased resolution or because of the smaller times step (15 minutes) required by the flux limiter.

Figure 5.11 shows the global L_1 , L_2 and L_∞ errors in the height field for each of the four runs when $u_0 = 40$ ms^{-1}. The global height errors for the original advection scheme are very similar to those of the same scheme on the un-tweaked grid and to those of new second order scheme. The results for the fourth order scheme are similar to the previous cases for about the first day but do not show the same slow growth at later times.



grid=6, original scheme

grid=6, order=2

grid=6, order=4

grid=7, order=2

Figure 5.10: Difference between true and numerical height fields after a five day run of test case 4 ($u_0 = 40$ ms^{-1}). Contour interval is 10 m.

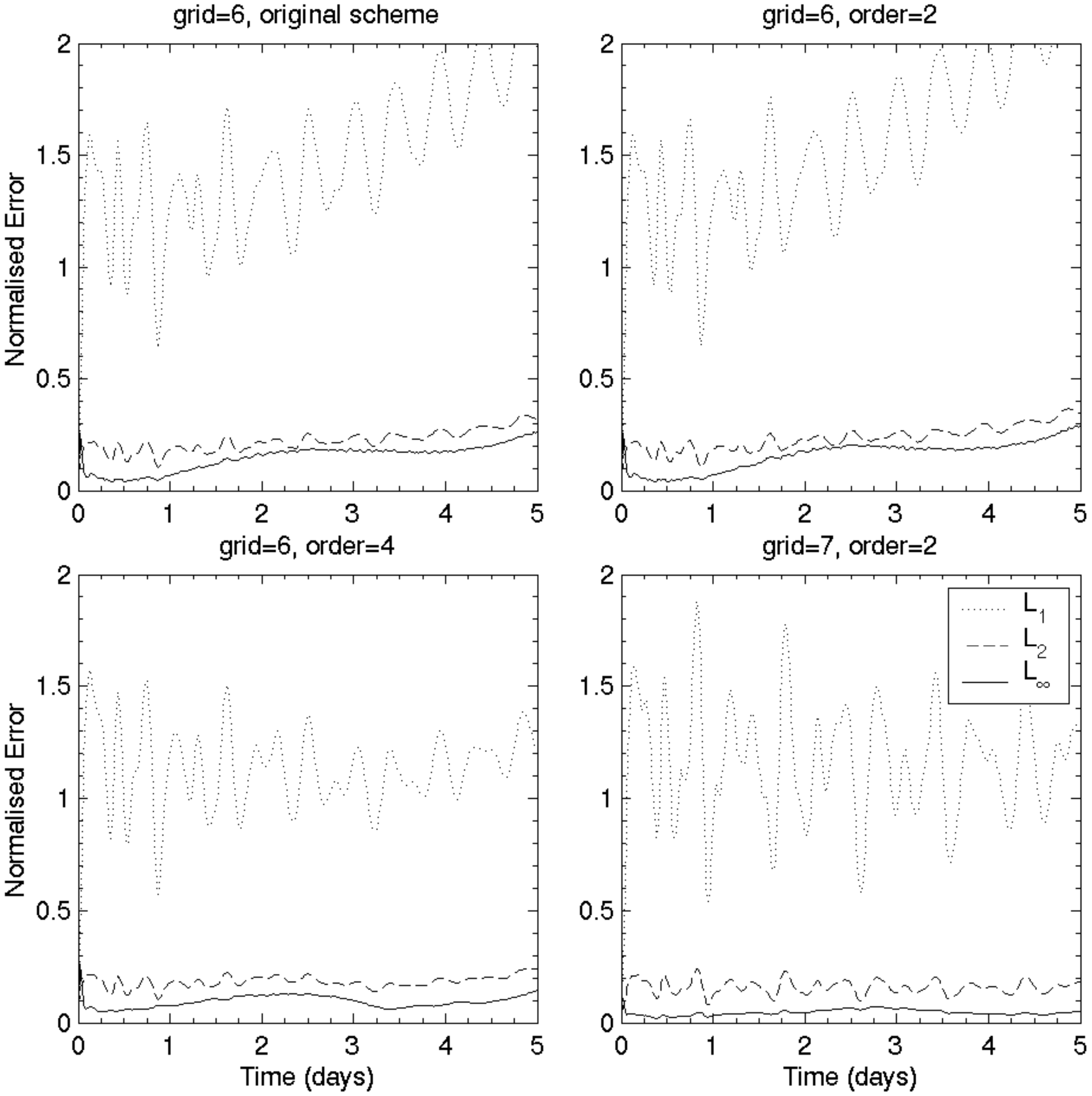


Figure 5.11: Normalised global errors in the height field for a five day run of test case 4 ($u_0 = 40$ m/s). Errors sampled at each time step (30 mins grid 6, 15 mins grid 7).

This suggests that using a fourth order polynomial for interpolation in the advection scheme captures all the features in the advected fields for this case. A second order polynomial captures many of these features but not all, as shown by the slow growth in errors for these cases. On the higher resolution grid using a second order polynomial the height errors do not grow significantly, suggesting that this combination of grid and scheme captures all the features of the advected profile. The results for the slower flow case are broadly the same, the only exception being a very slow growth of the error for the grid 7 case.

Figure 5.12 shows the global L_1 , L_2 and L_∞ wind errors for the same runs as before. The L_2 and L_∞ errors for the original scheme are very similar to those on the un-tweaked grid but the L_1 error is greatly reduced. This may again be because of a better representation of the source terms on the tweaked grid. Using the new second order scheme makes little difference to the results but the fourth order scheme does reduce the rate at which the errors grow. A greater reduction in this rate can be gained by increasing the resolution instead of the order of the scheme.

The results of the model of Heikes and Randall were poor and were not presented in full in [12]. From the comparisons that can be made, this model significantly outperformed that of Heikes and Randall. It correctly located, and produced less filling of the low centre. Comparison with the T42 spectral model show that the icosahedral model has some way to go before it can compete with the spectral model for this case. All the error measures of the spectral model are around a factor of ten smaller for this test. The spatial distribution of the errors for the icosahedral model does not oscillate in the same way as those for the spectral model but this is the only, small, advantage it has.

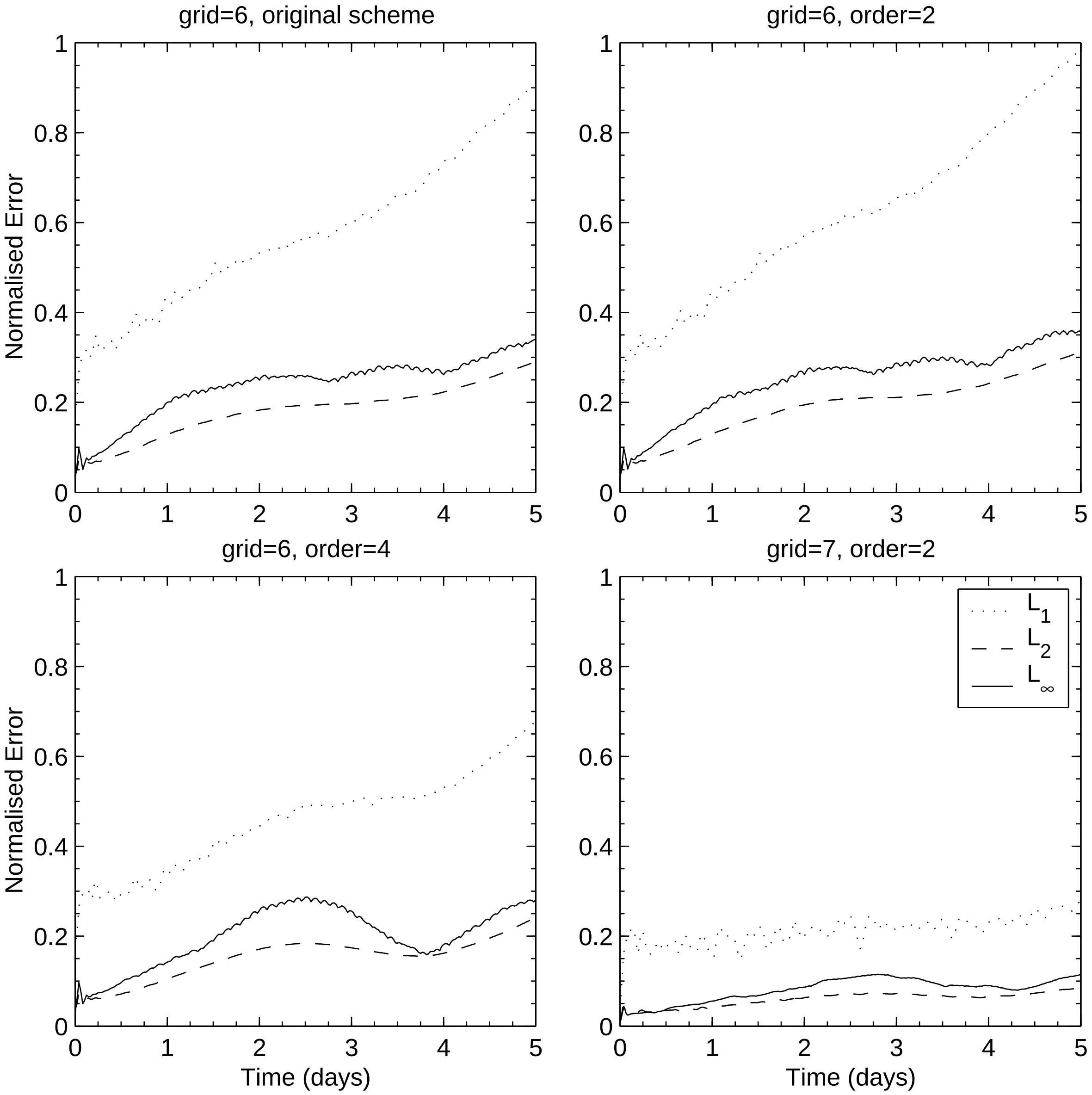


Figure 5.12: Normalised global errors in the wind field for a five day run of test case 4 ($u_0 = 40$ m/s). Errors sampled at each time step (30 mins grid 6, 15 mins grid 7).

5.6 Test Case 5:

5.6.1 Zonal flow over an isolated mountain

This test case is the only one in this set that involves a non-zero orography. An isolated mountain is placed in the flow used in case 2 to test the conservation of integral invariants of the shallow water system along with the usual measures of errors. The normalised integrals of the mass (h^*), total energy $(\frac{1}{2}h^*\mathbf{v}\cdot\mathbf{v} + \frac{1}{2}g(h^2 - h_0^2)\ )$ and potential enstrophy $(\frac{1}{2}h^*Q^2)$ are calculated along with the un-normalised integrals of vorticity (h^*Q) and divergence (δ). These are all conserved quantities for the shallow water equations and so their integrals over the domain should not alter in time. The global integrals of vorticity and divergence are initially zero in this case and should therefore remain so.

The wind field is described by the stream function and velocity potential of equations (5.7) with the parameters; $\alpha = 0$, $h_0 = 5960$ and $u_0 = 20$ m/s. The height of the surface orography, h_0 is given by

$$h_0 = h_{s0}\left(1 - \frac{r}{R}\right) \tag{5.28}$$

where $h_{s0} = 2000$ m, $R = \pi/9$, and $r^2 = \min[R^2,\ (\lambda - \lambda_c)^2 + (\varphi - \varphi_c)^2]$. The centre of the mountain is taken as $(\lambda_c, \varphi_c) = (3\pi/2, \pi/6)$.

There is no known analytic solution for this case so the results are compared with those of a high resolution spectral model. The high resolution model is run with T213 truncation with the archived results stored at T106 truncation. These are used with routines supplied with the data to calculate point values. The archived data and interpolation routines are currently maintained by John Truesdale at NCAR [56]. This reference solution exhibits spectral ringing around the mountain which is clearly visible in the plots of the difference between this 'true' solution and our numerical one (see figures 5.14 and 5.15). This is an artifact of the spectral methods used in the spectral model which is not present in the model we are using and it will contaminate the measures of the error to an unknown extent.

5.6.2 Results

The original model performed well in this test, resolving the features of the large scale flow forced by the mountain at a lower resolution than the one used here. One problem that was noted was the significant drift away from zero of the global integral of vorticity. The suggested reason for this was the inconsistency between the updated divergence, δ^{m+1} and the divergent part of the wind $\mathbf{v}_{div}^{m+1}$ caused by the approximate solution of the elliptic equation

$$\nabla \cdot \mathbf{v}_{div}^{m+1} = \delta^{m+1} \ . \tag{5.29}$$

Changing the advection scheme is unlikely to have a large effect on this problem but the tweaking of the grid should improve the approximation.

The reference solution for this case at days 5, 10 and 15, as converted to the icosahedral grid, is shown in Figure 5.13. The numerical results for this case are again all very similar to these solutions when compared by eye. For this reason we shall again look at plots of the differences between this solution and the numerical ones. These are shown in figures 5.14, for day 5 and 5.15, for day 15.

The first thing that is clear when looking at the difference between the high resolution spectral solution and those on the icosahedral grid is the location of the mountain. This is caused by the spectral ringing around the mountain in the 'true' solution, but the icosahedral models have no

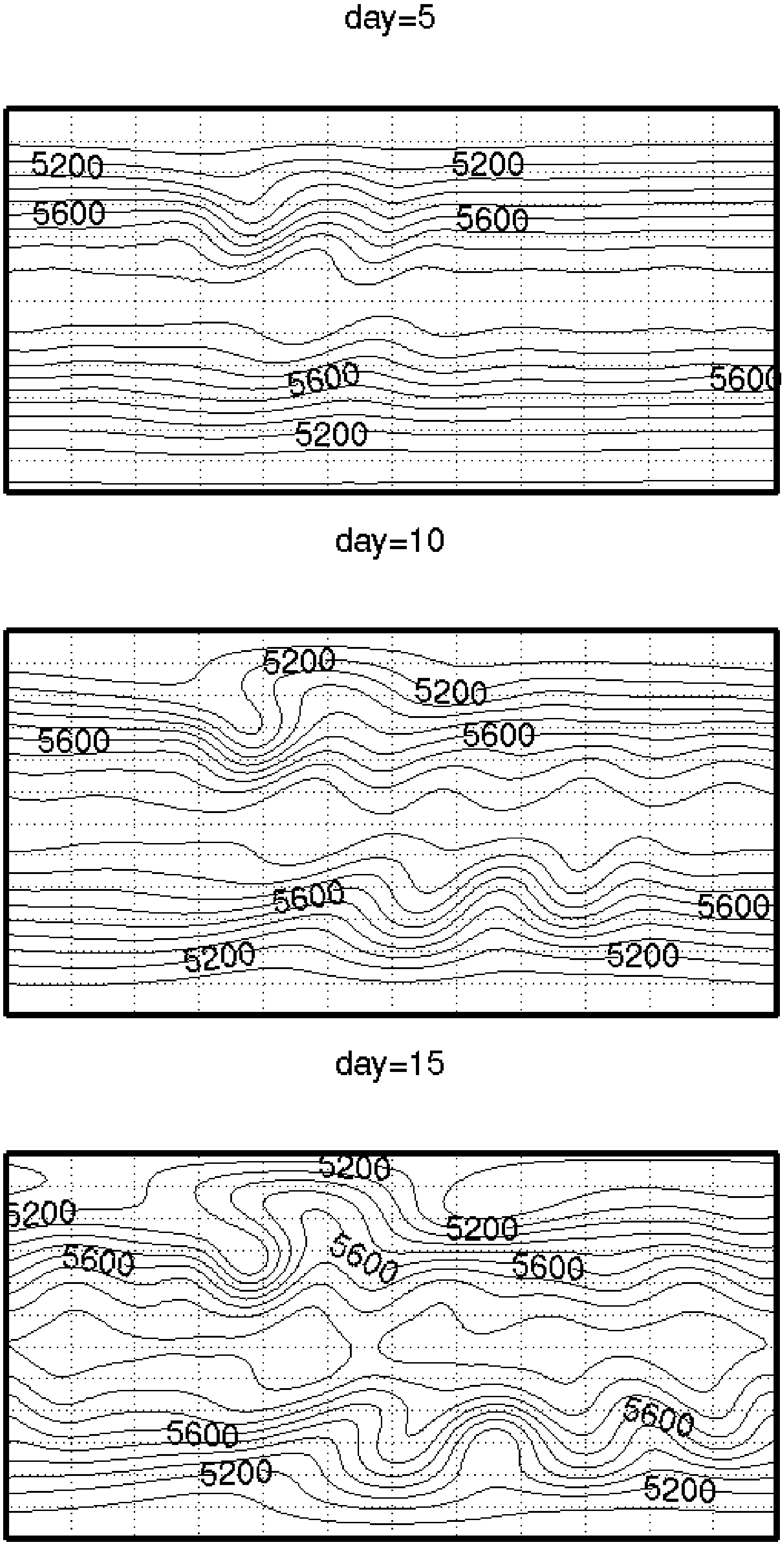


Figure 5.13: High resolution spectral solution of the height field for test case of flow over a mountain after 5, 10 and 15 days

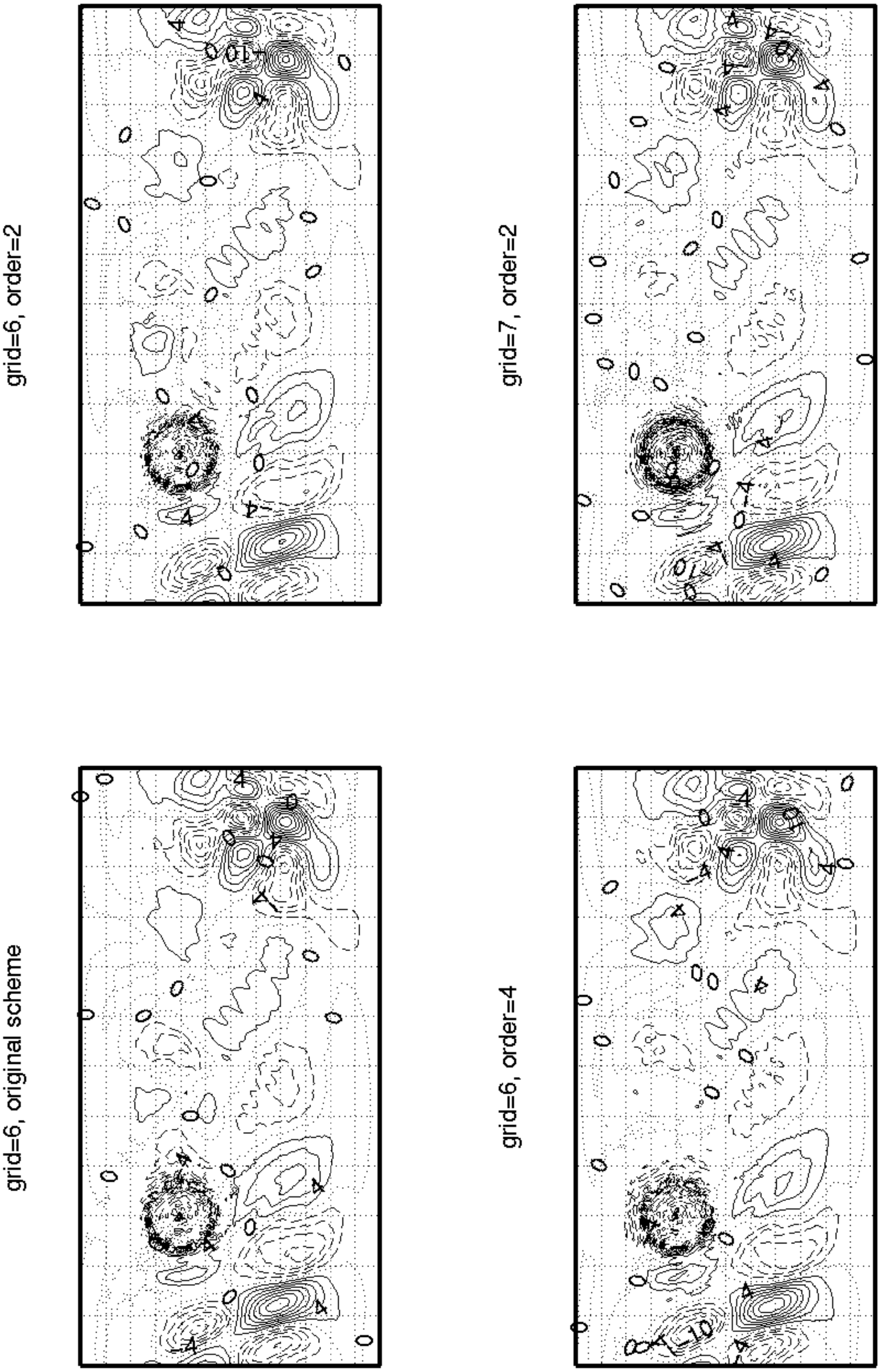


Figure 5.14: Difference between spectral and icosahedral height fields after a five days of a run of test case 5. Contour interval is 2 m.

such problems representing the mountain. After five days, Figure 5.14, the errors downwind of the mountain are smaller than those in other regions of the flow. This suggests that the advection in the flows generated by the mountain is well modelled but that other dynamic processes are not so well captured. In both regions, the differences between the different schemes is remarkably small. The main difference is the slightly higher errors downwind of the mountain when using the new second order scheme over the original one. Using the new fourth order scheme improves slightly over both these schemes and gives similar results to those on the higher resolution grid.

After fifteen days, Figure 5.15, the differences between the schemes are clearer in the region downwind of the mountain. The solution in the southern hemisphere is again similar for all the schemes on grid 6 but appears significantly better on grid 7. This again suggests that the main cause of these errors are not inaccuracies in the advection scheme but some other approximation in the model. Altering the time step for these tests would show if the inaccuracy was caused by time truncation error in any part of the model. The problem could also be caused by the different representations of the reference solution or the mountain on the icosahedral grid. Comparing these results with a high resolution test on the icosahedral grid would help to isolate these potential problems.

Comparing these plots with the spectral model results, Figure 5.13, shows that the pattern of ridges and troughs are weaker in the icosahedral model. The original scheme tends to have a weaker ridge than the reference solution just downwind of the mountain, yet the new second and fourth order schemes produce an even stronger one. These errors are reduced as the order of the scheme and resolution are increased. Aside

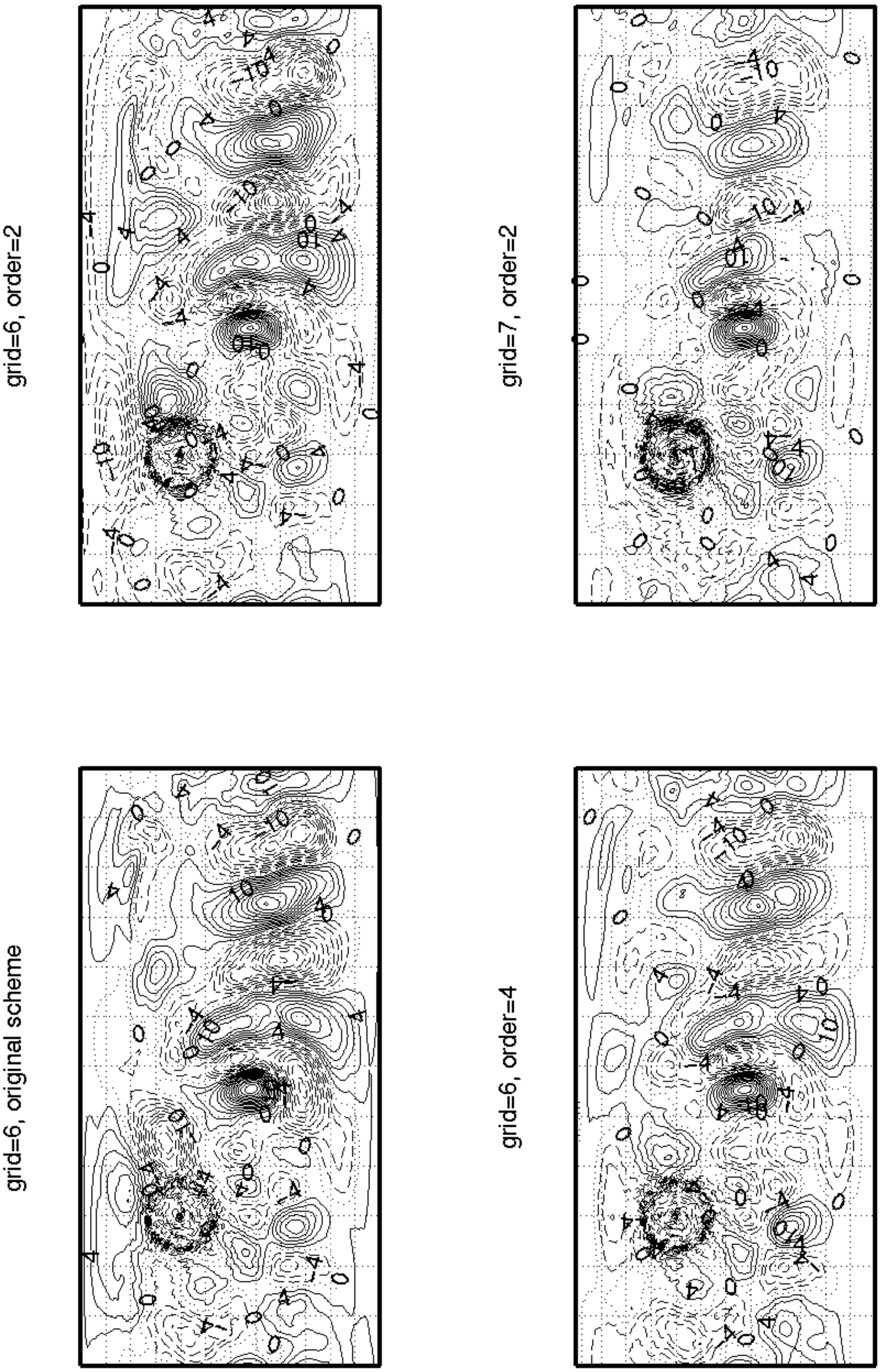


Figure 5.15: Difference between spectral and icosahedral height fields after a fifteen day run of test case 5. Contour interval is 2 m.

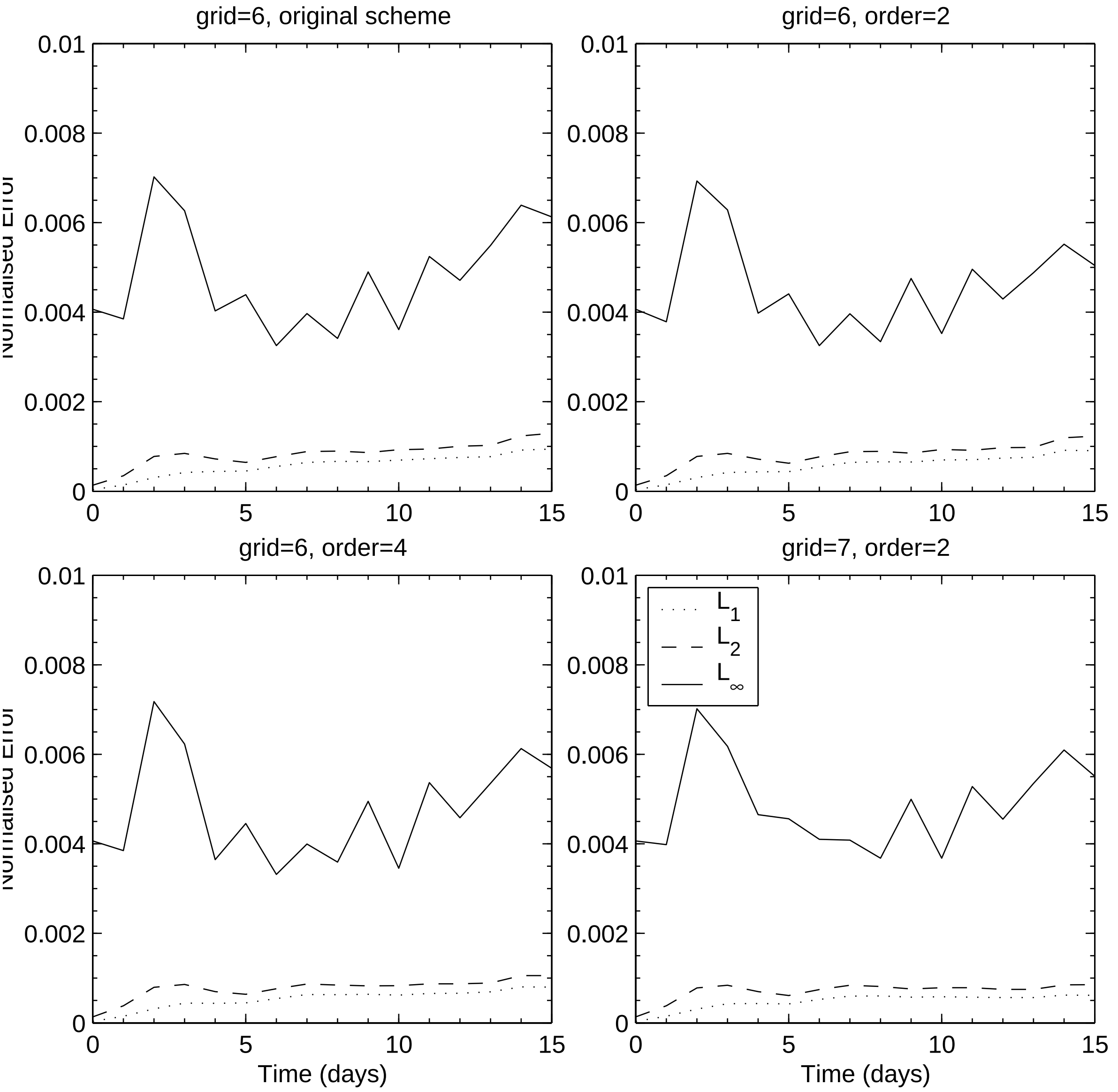


Figure 5.16: Normalised global errors in the height field for a fifteen day run of test case 5. Errors are a comparison between numerical results and a high resolution spectral model, sampled every day.

from this region the main differences are in the peak values of the features in the error fields, which are again reduced by increasing the order of the advection scheme and the resolution of the grid.

The evolution of the global height error (compared to the reference solution) is shown in Figure 5.16. This reinforces the similarities between the results, irrespective of the advection scheme used. There is a slight improvement in these results as the scheme order and grid resolution are increased. This is made more difficult to observe by the relatively small

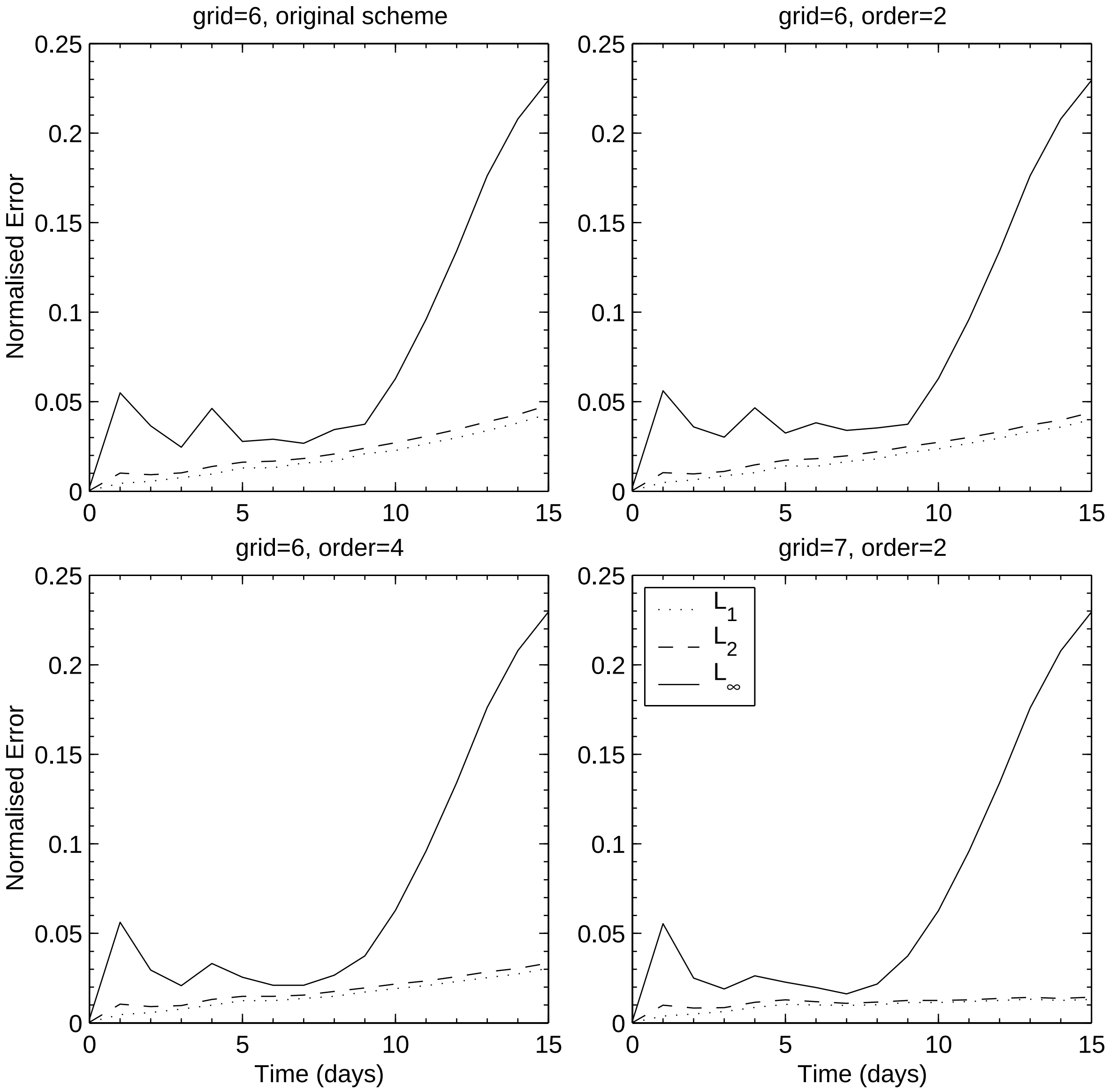


Figure 5.17: Normalised global errors in the wind field for a fifteen day run of test case 5. Errors are a comparison between numerical results and a high resolution spectral model, sampled every day.

errors in this case. The improvements in the global wind field errors as the schemes are changed, Figure 5.17, are more pronounced than for the height field. The rapid increase in the L_∞ error after day nine is the same in all cases but cannot be seen in the L_1 and L_2 errors. This suggests that the cause of this error is localised.

Figure 5.18 shows the evolution with time of the global invariants of the shallow water model. The mass is conserved to machine precision regardless of the advection scheme used and is not shown here. For this

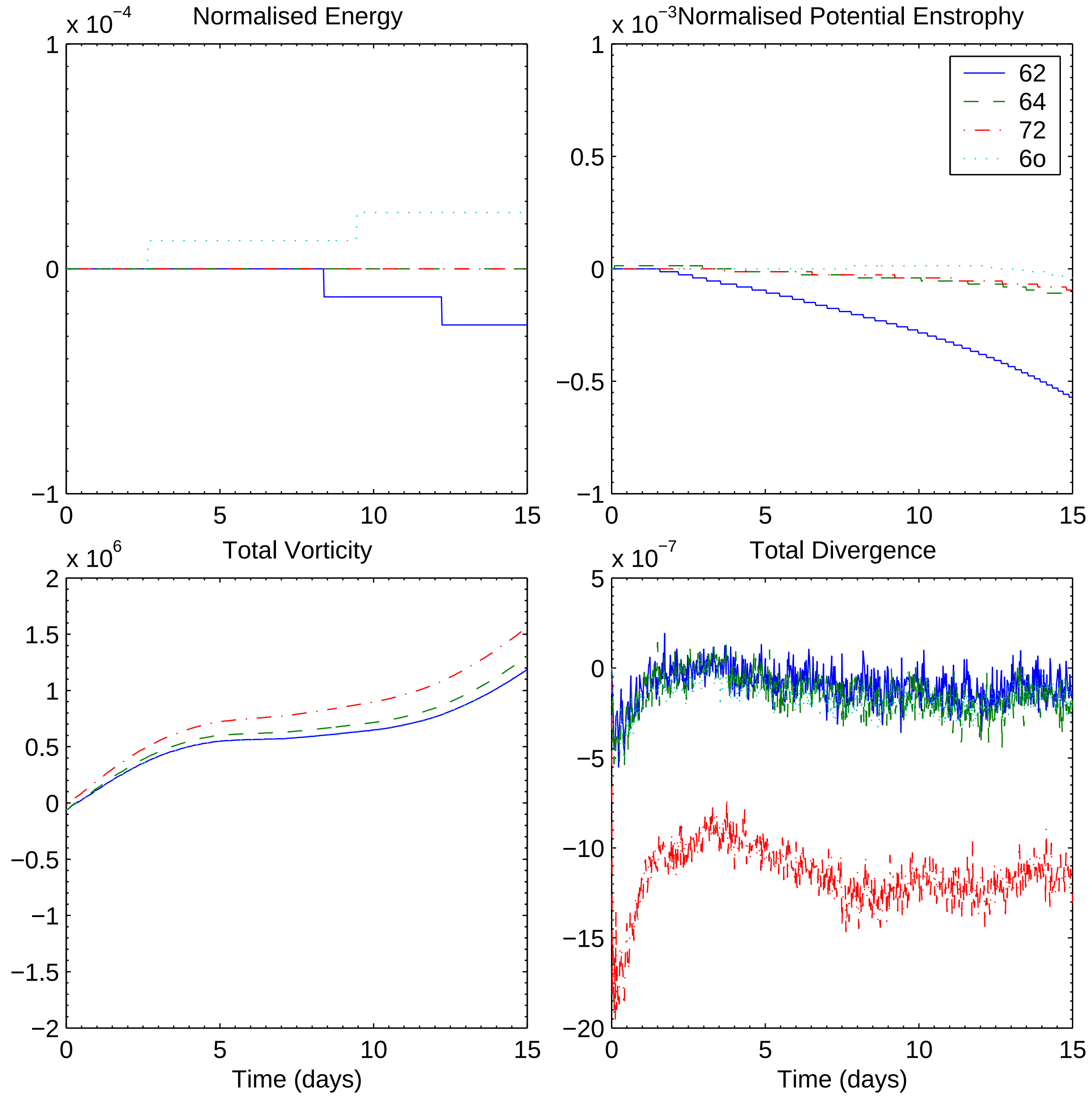


Figure 5.18: Time evolution of the global integral diagnostics sampled at each time step. The schemes are classified first by the grid resolution (6 or 7) and then by the advection scheme (2, 4, or o-the original scheme).

case the energy is also conserved to the accuracy at which the model output is written for all cases. There is a loss of potential enstrophy in the model due to its cascade to unresolved scales. This process can be sped up by inaccuracies in the numerical methods used. This is shown to be much more of a problem for the second order scheme on grid six. The test using the original scheme includes the generation of a significant amount of potential enstrophy. The similarity between these results for the fourth order scheme on grid 6 and the second order scheme on grid 7 suggest that this is more likely to be the cascade to unresolved scales that we would expect.

The total vorticity grows for the same reasons as suggested by Thuburn [52], mentioned above. The convergence of the elliptic solver for equation (5.29) will depend on the smoothness of the divergence field. This in turn is partly dependent on the advection scheme used, the original and second order schemes on grid 6 give the smoothest fields and cause a slower growth in the total vorticity. In all grid 6 cases there is an initial imbalance in the divergence field which reduces back to balance after around two days. This initial imbalance is much larger on grid 7 and the balance is not restored.

Comparing the error fields with those from other models shows that all the models used here perform better than the comparable resolution models of Heikes and Randall. Comparison between the results using grid 6 and the T63 spectral model show results that are broadly similar. Comparing the global diagnostics shows a better conservation of energy and potential enstrophy by these models over those of Heikes and Randall. The total vorticity and divergence are similar for these cases. The spectral model does not conserve mass or energy as the model used here does but does improve on these results for the other integral diagnostics.

5.7 Test Case 6:

5.7.1 Rossby-Haurwitz wave

A Rossby-Haurwitz (R-H) wave is an analytic solution of the non-divergent shallow water equations consisting of steadily propagating profile. This has become a *de facto* standard test case due to its use by many authors, despite the use of different parameters and the fact that R-H waves are not analytic solutions of the full shallow water equations.

The initial, non-divergent, velocity field is specified by the stream function

$$\psi = -a^2\omega \sin\varphi + a^2 K \cos^R \varphi \sin\varphi \cos R\lambda \ , \tag{5.30}$$

where ω, $K = 7.848 \times 10^{-6}$ s^{-1} are constants and $R = 4$ is the zonal wavenumber. The initial height is obtained from the stream function by solving a Charney balance equation so that the initial divergence tendency is zero, giving

$$gh = gh_0 + a^2 A(\varphi) + a^2 B(\varphi) \cos R\lambda + a^2 C(\varphi) \cos 2R\lambda \ , \tag{5.32}$$

where functions A, B and C are given by

$$\begin{aligned} A(\varphi) =& \frac{\omega}{2}(2\Omega{+}\omega)\cos^2\varphi \\ &+\frac{1}{4}K^2\cos^{2R}\varphi\left[(R{+}1)\cos^2\varphi{+}(2R^2{-}R{-}2){-}2R^2\cos^{-2}\varphi\right] \ , \end{aligned} \tag{5.33$_A$}$$

$$B(\varphi) = \frac{2(\Omega+\omega)K}{(R+1)(R+2)}\cos^R\varphi\left[(R^2+2R+2)-(R+1)^2\cos^2\varphi\right] \ , \tag{5.33$_B$}$$

$$C(\varphi) = \frac{1}{4}K^2\cos^{2R}\varphi\left[(R+1)\cos^2\varphi-(R+2)\right] \ . \tag{5.33$_C$}$$

The background height, h_0, is taken to be 8×10^3 m. This test has significantly stronger winds than those used in other cases, so shorter time steps of 15 minutes on grid 6 and 10 minutes on grid 7 have been used.

The 'true' solution is given as the results from a high resolution spectral method as in the previous test case. In this case there are none of the problems with this solution that were seen in the case of flow over a mountain. The parameter $R = 4$ was chosen by Williamson et al. [59] who argued;

> Unstable waves [13] are not chosen, since slightly different perturbations may lead to growth of different unstable modes as might be indicated in Kreiss and Oliger [16].

Thuburn and Li [54] showed, using the icosahedral model amongst others, that perturbations to this wave can project onto unstable modes in this case. This was shown to be a particular problem for the icosahedral model which projected significant errors onto the most unstable mode because of the wavenumber 5 pattern of the grid. Thuburn and Li [54] concluded that;

> The breakdown of the wavenumber 4 flow pattern must be interpreted as a real dynamical instability ... not as a catastrophic failure of the numerical methods.

This behaviour is reduced at higher resolutions and for smaller time steps and should not cause catastrophic problems in these model runs.

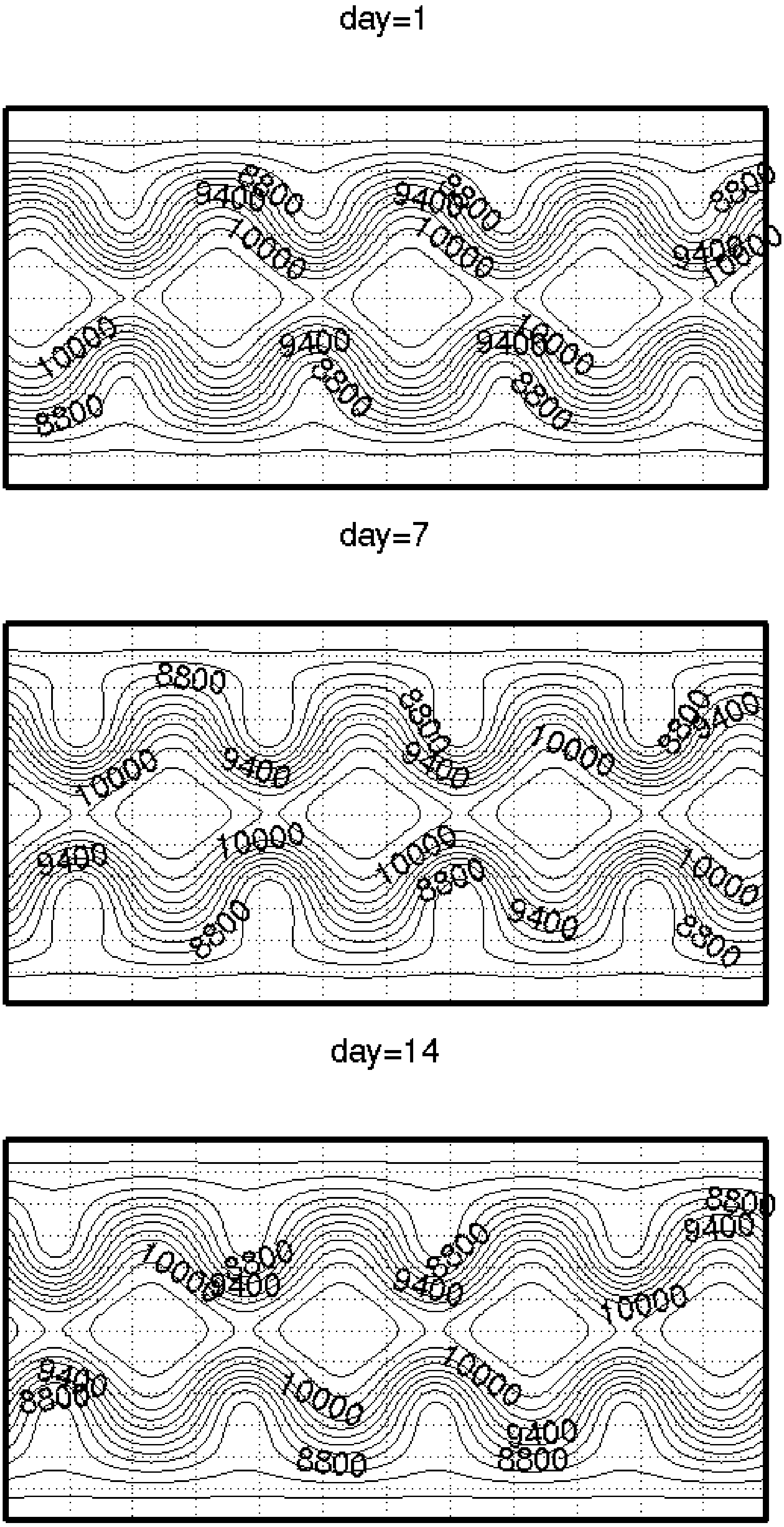


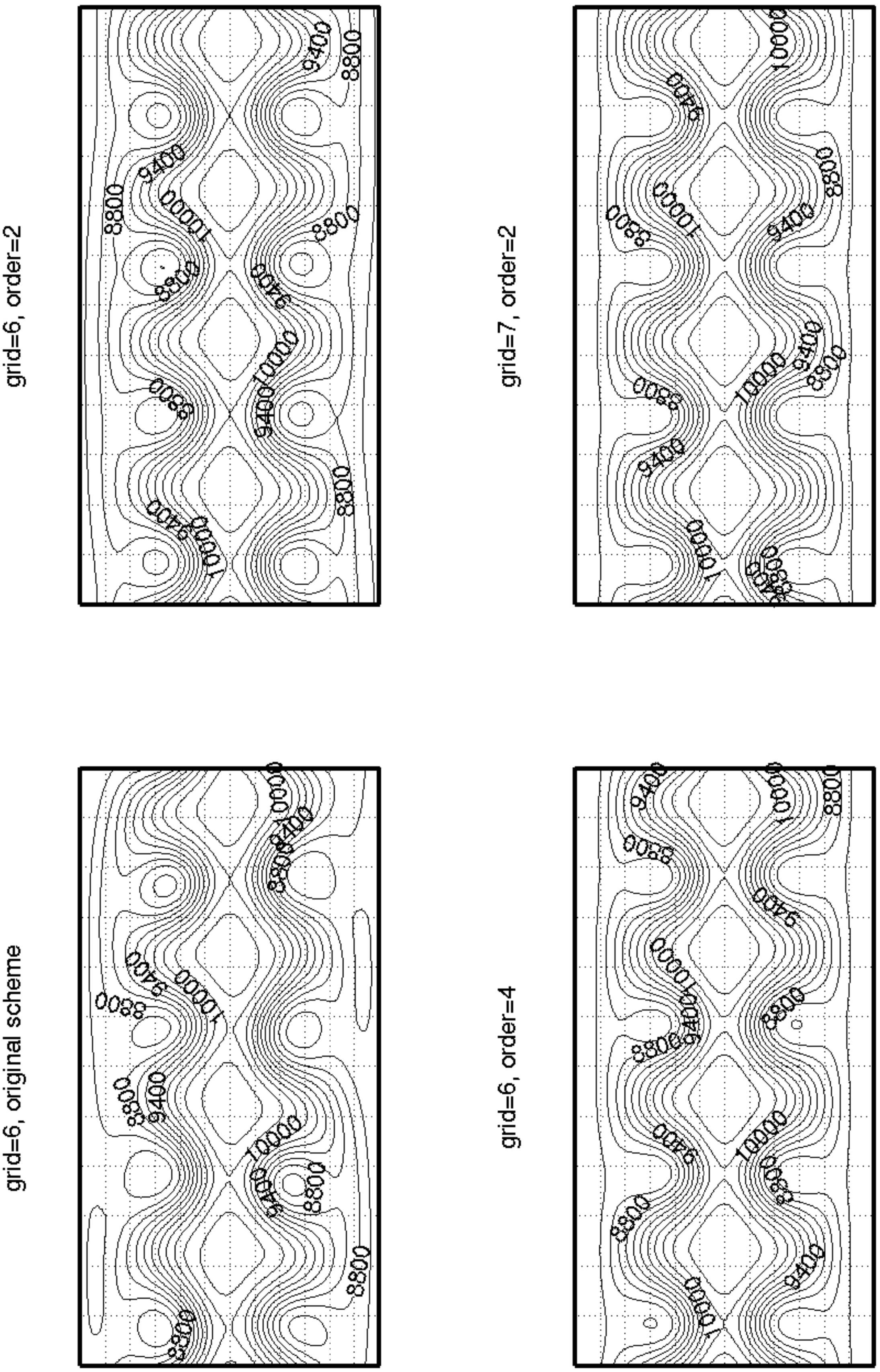
Figure 5.19: Spectral solution for test case of a Rossby-Haurwitz wave after 1, 7 and 14 days. Contour interval is 200 m.

5.7.2 Results

The results of the spectral model at days 1, 7 and 14 are shown in Figure 5.19. Computing this solution has not excited any of the unstable modes in the solution. When this is compared with the results after 14 days, Figure 5.20, for the icosahedral models the loss of symmetry between the waves can clearly be seen. This is more pronounced for the original scheme than for the new schemes suggesting that the original scheme introduces a larger error onto the unstable modes.

Figure 5.21 shows the difference between the icosahedral and reference solution after one day of the run. The largest errors correspond with the gradients of the wave pattern seen in the reference solution and have a wavenumber four structure. The pattern of errors is consistent across all these tests, only the magnitude varies. A closer inspection of these error fields shows some smaller differences between the errors within each wave. These differences are largest in the original scheme and are reduced as higher order versions of the new schemes are used. Once these small errors are introduced they will grow through dynamic instability and produce the asymmetry between the waves that is seen in Figure 5.20. This shows the smaller, more regular error at the final time when the higher order new schemes are used.

The new second order advection scheme has up to now been comparable with the original scheme. In this case however the results of the new scheme do not display such a strong asymmetry, this may be because of the more general treatment of the grid. The original method is based on a scheme used on regular hexagons so any deviation from this pattern will create an error. In this case the variations from regular hexagons



grid=6, original scheme

grid=6, order=2

grid=6, order=4

grid=7, order=2

Figure 5.20: Numerical solutions for test case of a Rossby-Haurwitz wave after 14 days on icosahedral grid. Contour interval is 200 m.

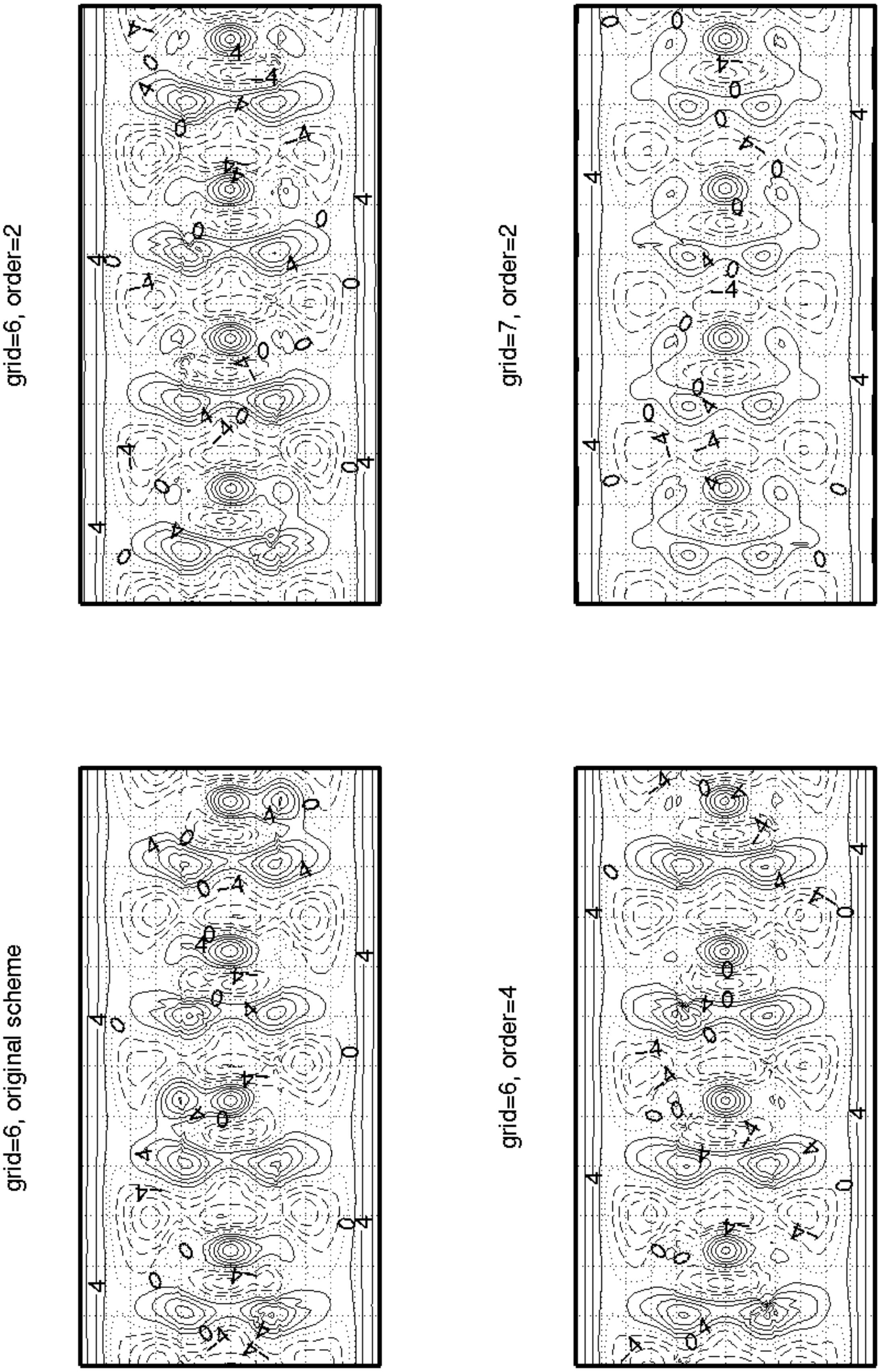


Figure 5.21: Difference between spectral and icosahedral height fields after one day of a run of test case 6. Contour interval is 2 m.

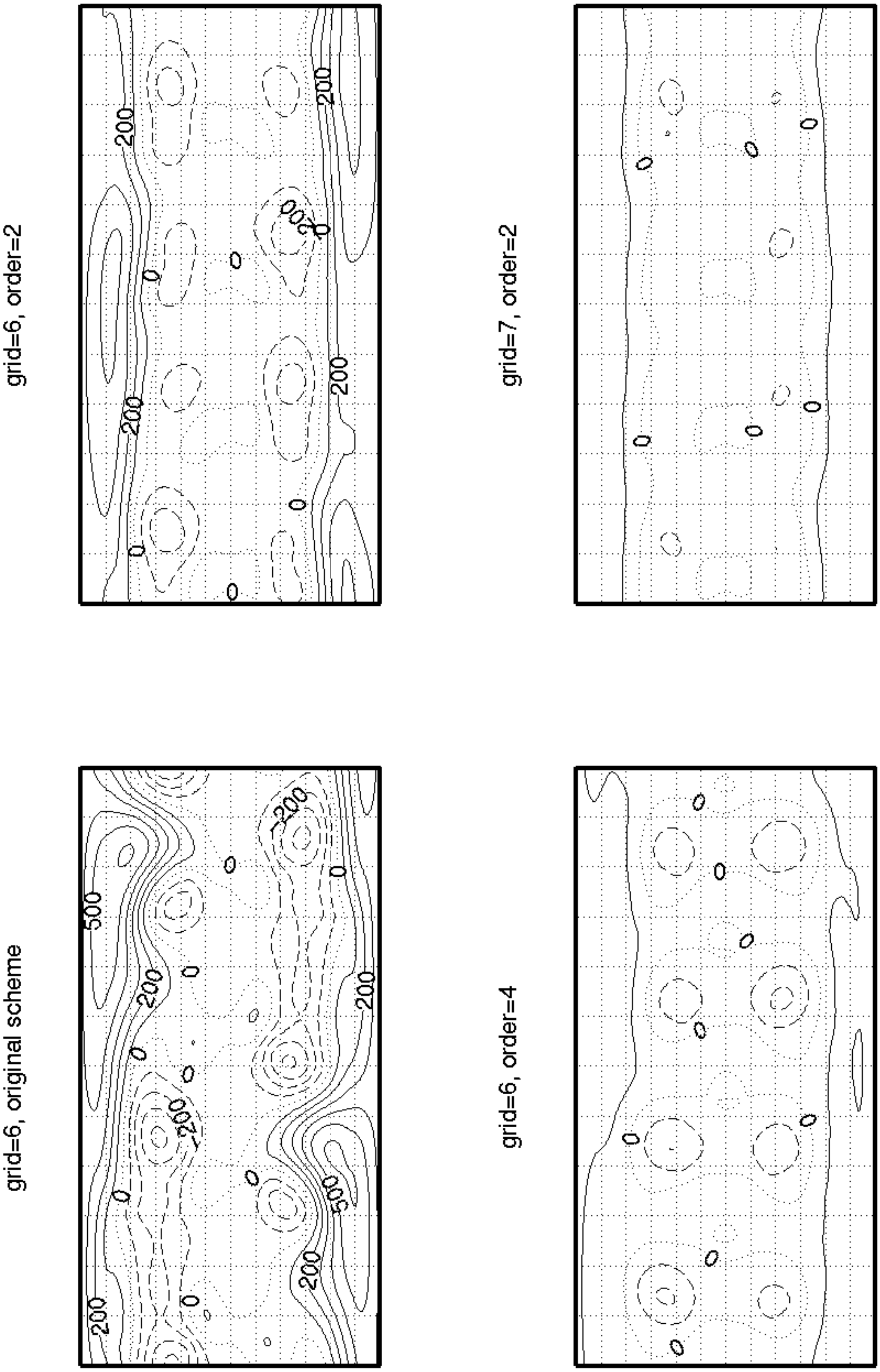


Figure 5.22: Difference between spectral and icosahedral height fields after a fourteen day run of test case 6. Contour interval is 100 m.

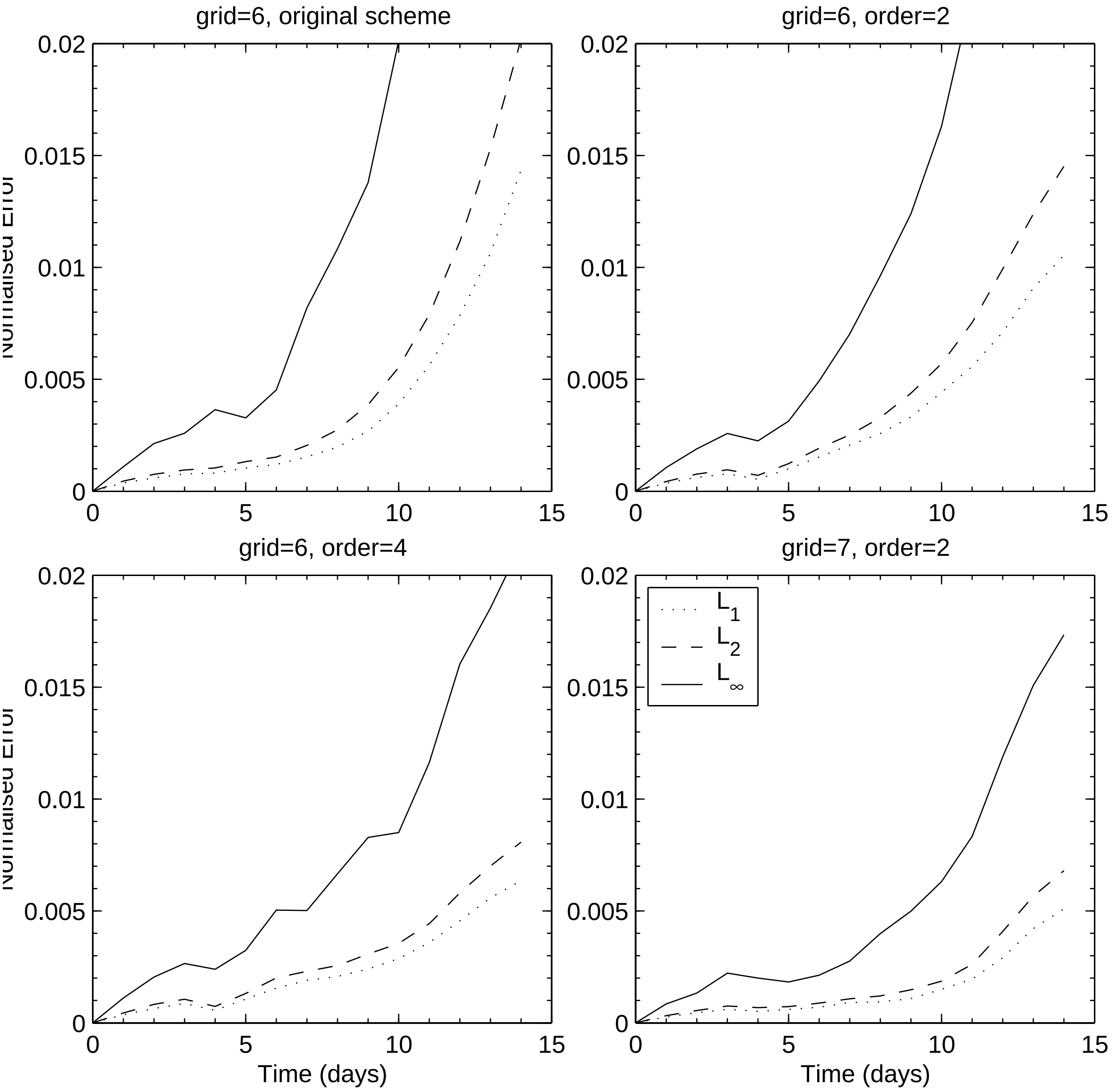


Figure 5.23: Normalised global errors in the height field for a fourteen day run of test case 6. Errors are a comparison between numerical results and a high resolution spectral model, sampled every day.

had a wavenumber five pattern which was duplicated in the error in this case. The new advection scheme generates an advection scheme for each cell and so the error is not so strongly linked to variations in the grid. This resulted in a weaker wavenumber five component of the error being generated and resulted in less growth of the unstable modes at any given time.

The global errors shown in Figure 5.23 illustrate the effect of these growing modes on the global error. For the first few days the errors behave as for previous cases. The global height error increases slightly for the new second order scheme over the original scheme but the new fourth order scheme is an improvement over both. As the unstable modes grow, so the error begins to grow rapidly, this happens sooner and faster for the original and second order scheme. The fourth order scheme projects a smaller error onto the unstable mode so it takes longer for this to grow and contaminate the solution, this also means that the growth is slower. Using the higher resolution grid introduces much smaller errors onto the unstable modes with the result that the global error measures show rapid growth much later in the run. This is partly due to the increased resolution and partly because of the shorter time step used in this case. The same observations can be made for the global wind errors (not shown) for this case.

The normalised global diagnostics, Figure 5.24 again show the improved conservation properties of the new schemes. The mass is again conserved by the model for all cases and is not shown. There is a greater loss of energy using the original scheme on the new grid than on the old. Increasing the resolution appears to be the best way of improving conservation of the energy. The same can be said for the potential enstrophy. There is a significant cascade to small scales for the original and second order schemes on grid 6 and the fourth order scheme appears to generate some potential enstrophy.

The total divergence fluctuates about a value that is slightly larger than zero. These fluctuations are larger when the new advection schemes are used which has an effect on the total vorticity. Larger fluctuations in the divergence field suggest that it is less smooth which will cause the kind of growth in vorticity seen in the last test and described in Thuburn [52].

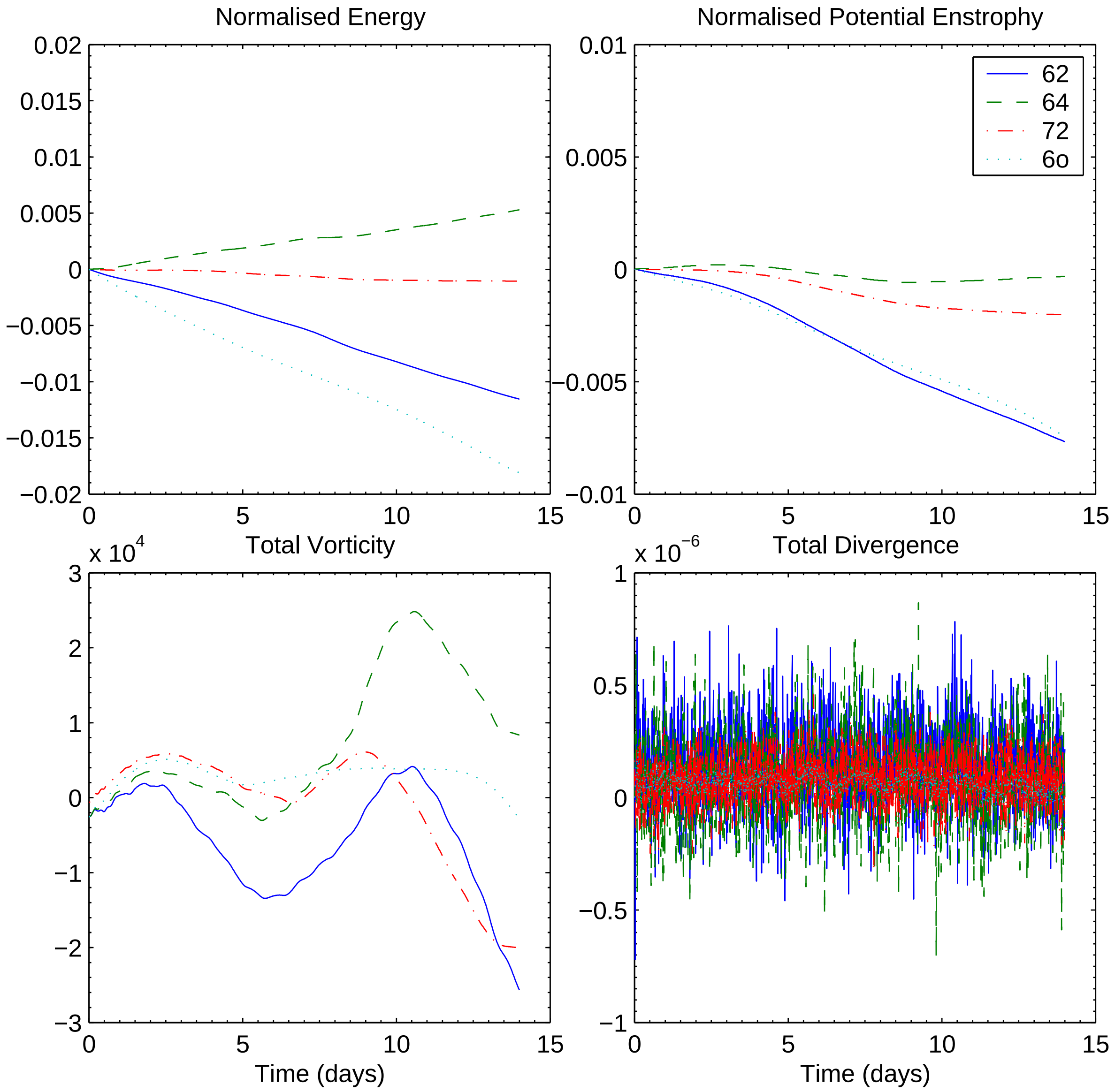


Figure 5.24: Time evolution of the global integral diagnostics sampled at each time step. The schemes are classified first by the grid resolution (6 or 7) and then by the advection scheme (2, 4, or o-the original scheme).

The effect of this can be seen in the time evolution of the total vorticity in the models. This shows less fluctuation in time when the original scheme is used resulting in a better balance at later times.

These results are comparable in magnitude to those of the T63 spectral model tested by Jakob et al. [15]. The results presented here show a significant improvement over those of Heikes and Randall. They display the same change in the structure of the waves as that seen on the twisted icosahedral used by Heikes and Randall [12]. One property of the solutions that was noted in this case was the symmetric nature of the solutions about the equator. In this work the grid is not twisted, the grids for the two hemispheres can be though of as being 180° out of phase, as is the solution in the two hemispheres.

We noted in test case three that a wavenumber four pattern was seen in the error field when the grid was rotated by an angle of $\pi/3$. One way of further improving these results may be to rotate the grid in this way. This would mean that the errors generated have a large wave number four component and only a small fraction of the error is projected onto the dynamically unstable modes.

5.8 Test Case 7:

5.8.1 Analysed 500mb height and wind field initial conditions

Three sets of real atmospheric 500mb height and wind fields from different atmospheric situations are used as initial conditions. The first case (7a) is of 0000GMT on December 21, 1978, which has a strong flow over the north pole. The second case (7b) is of 0000GMT on January 16, 1979, and contains two cutoff lows in a pattern that develops into a blocking situation. The third case (7c) is of 0000GMT on January 9, 1979, with an initially strong zonal flow.

The previous test cases have all been idealised situations that have tested various features of the shallow water model. These test cases are more realistic in that they test how the features of the model combine with one another in producing the final numerical 'forecast'. Included within these tests are small scale features in the PV field and a cascade to smaller scales. From previous results we would expect higher order schemes to better model both the small scale features and the cascade to small scales. We should now be able to see what overall effect these improvements will have on the results when interacting with other causes of inaccuracy in the model.

Since the shallow water equations should not be expected to predict the atmosphere well in these cases, the test is for the model to deal with a variety of atmospheric states. The results of the model are again compared with a reference solution of a high resolution spectral model. What is perhaps more important from our point of view is whether the new advection schemes enable us to better resolve the features of the flow. The schemes may be able to improve the numerical errors but we would also like to see a qualitative improvement in our 'forecasts'.

The wind speeds in these cases all require that a smaller time step is used on grid 7 to satisfy the limiter condition on the Courant number. A 20 minute time step is used for all the following grid 7 results.

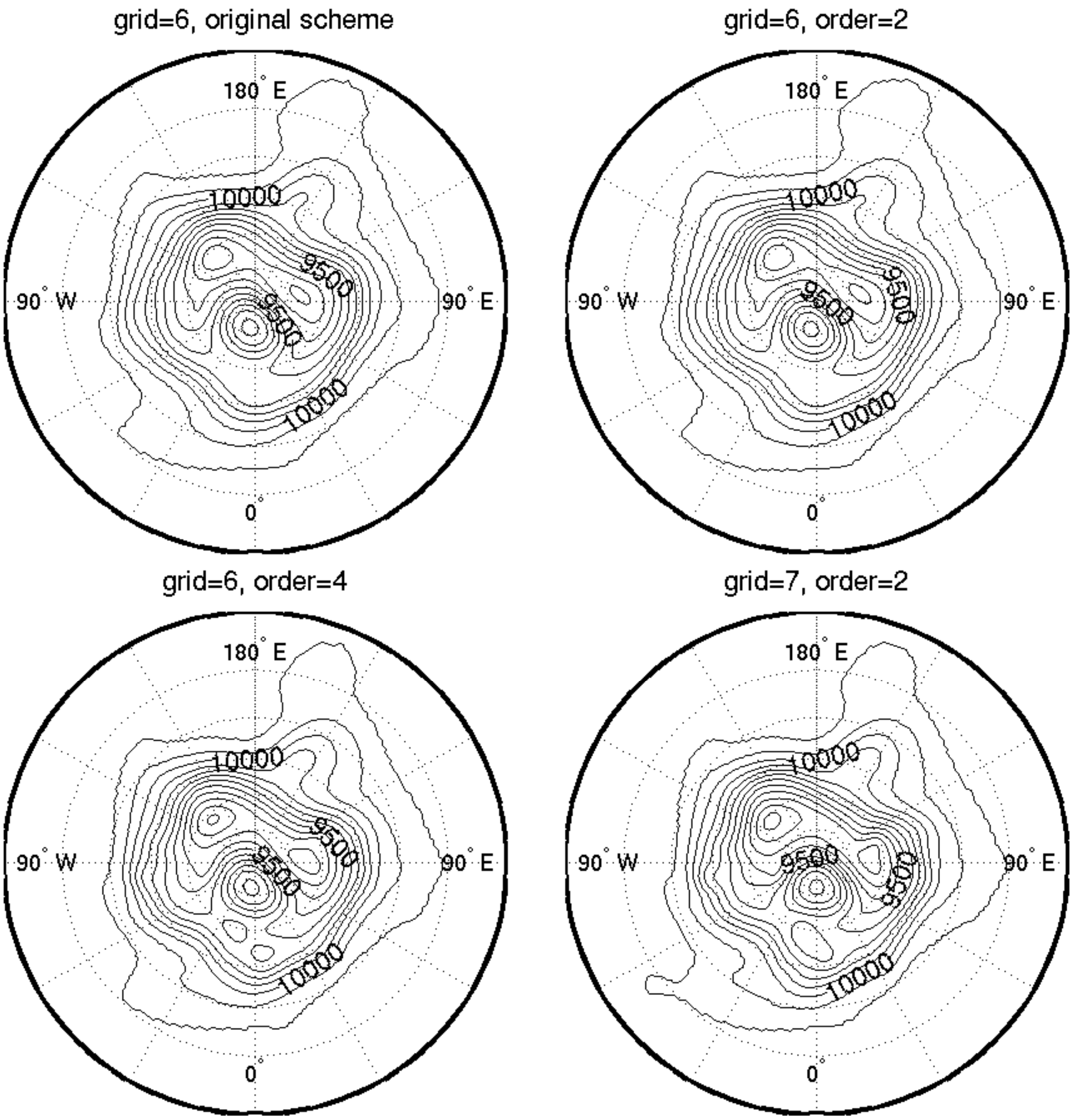


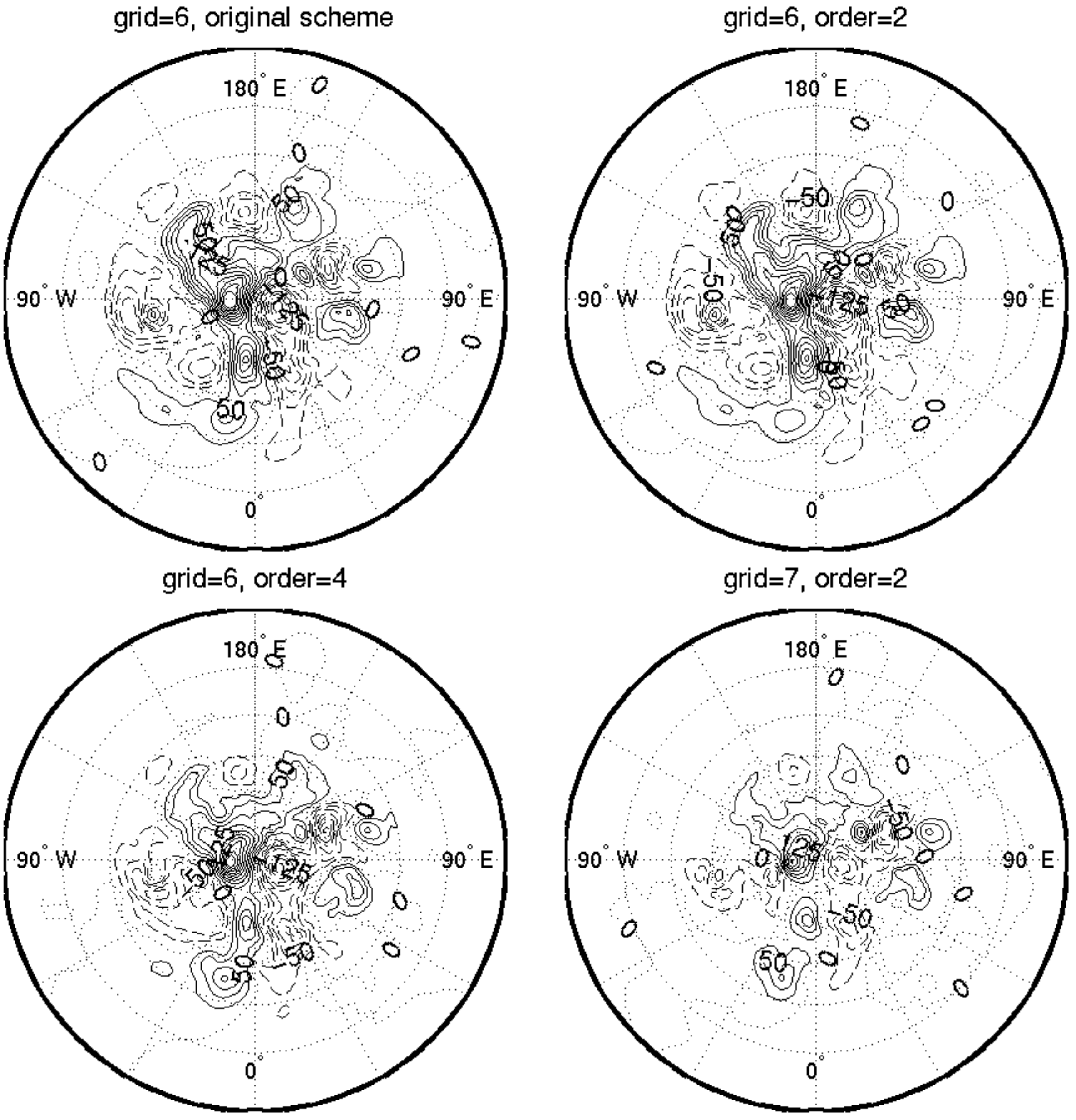
Figure 5.25: Height field in the northern hemisphere after a five day run of test case 7a. Contour interval is 100 m

5.8.2 Results

7a: 21 December 1978

Figure 5.25 shows the final height field after a five day run of test case 7a. The strong flow over the pole in this case should not pose a problem on the icosahedral grid because of its uniform nature. A close inspection of these plots reveals that there are important differences between the results for the different schemes. Looking at the high centre close to the pole using the original and new second order schemes shows virtually no difference between the two schemes. The high centre is slightly further to the east at this time when using the fourth order scheme and is even further east when the higher resolution grid is used. Another feature is the small trough at around 85° west. This feature is somewhat smoothed by the original and second order schemes but is sharper when the fourth order scheme or higher resolution is used.

Both these features can also be seen in the plots of the difference between the icosahedral models and the high resolution spectral model, Figure 5.26. Both features are characterised by positive-negative dipoles in the error field which are reduced when the higher order and higher resolution schemes are used. A number of other improvements can be seen as the higher order scheme or higher resolution grid are used. These improvements carry through to the global error measures of the height field, shown in Figure 5.27. There is again little difference between the original and new second order schemes and a small improvement when the fourth order scheme is used. There is a significant improvement in the global errors when the higher resolution grid is used, all the global error measures being reduced by around a half. The same is true of the


grid=6, original scheme

grid=6, order=2

grid=6, order=4

grid=7, order=2

Figure 5.26: Difference between Icosahedral and spectral height fields in the northern hemisphere after a five day run of test case 7a. Contour interval is 25 m

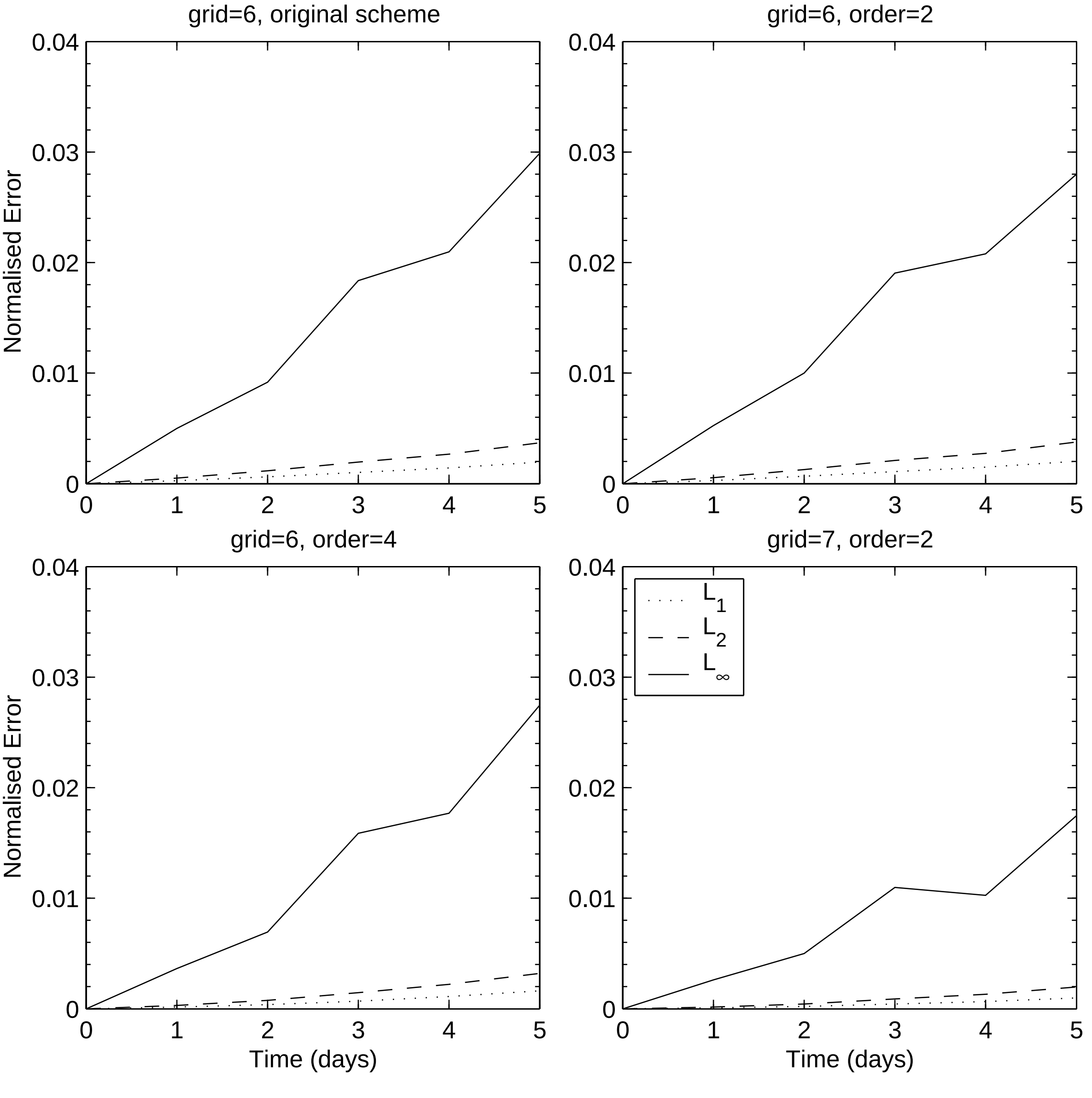


Figure 5.27: Normalised global errors in the height field for a five day run of test case 7a. Errors are a comparison between numerical results and a high resolution spectral model, sampled every day.

errors in the wind field (not shown).

Figure 5.28 shows the time evolution of the global diagnostics for this case, which are very similar to those for the other two initial conditions. As ever, the model conserves mass exactly to round off error. In these tests there is a slight loss of energy in all the models. The loss is greatest for the original and second order schemes on grid 6. Using a fourth order polynomial on this grid significantly improves the conservation of energy, using the higher resolution model provides a further slight improvement. The same relationship can be seen in the conservation of potential enstrophy. In this case however, it is clear that the cascade to unresolved scales is faster at the start of the run than at later times. This suggests there are a lot of small scale features in the initial conditions which are quickly lost through inaccuracies in the schemes. This process is faster for the original and second order schemes initially but the cascade to unresolved scales appears more even across all schemes later in the run.

The fields for these test cases contain an initial imbalance in the total vorticity which is maintained by the schemes used in all cases. There are some differences between the vorticity for the grid 6 schemes however, they are small compared to the magnitude of the field. The fluctuations in the total divergence is also similar for the three grid 6 models but slightly different for the grid 7 model.

The results using the fourth order advection scheme on grid 6 compare favourably with those of the T42 spectral model tested by Jakob et al. [15]. Increasing the resolution to grid 7 and using a second order scheme makes the results comparable with those of the T63 spectral model, in terms of the global error measures.

The reasons for some of the features of the results can be seen in the

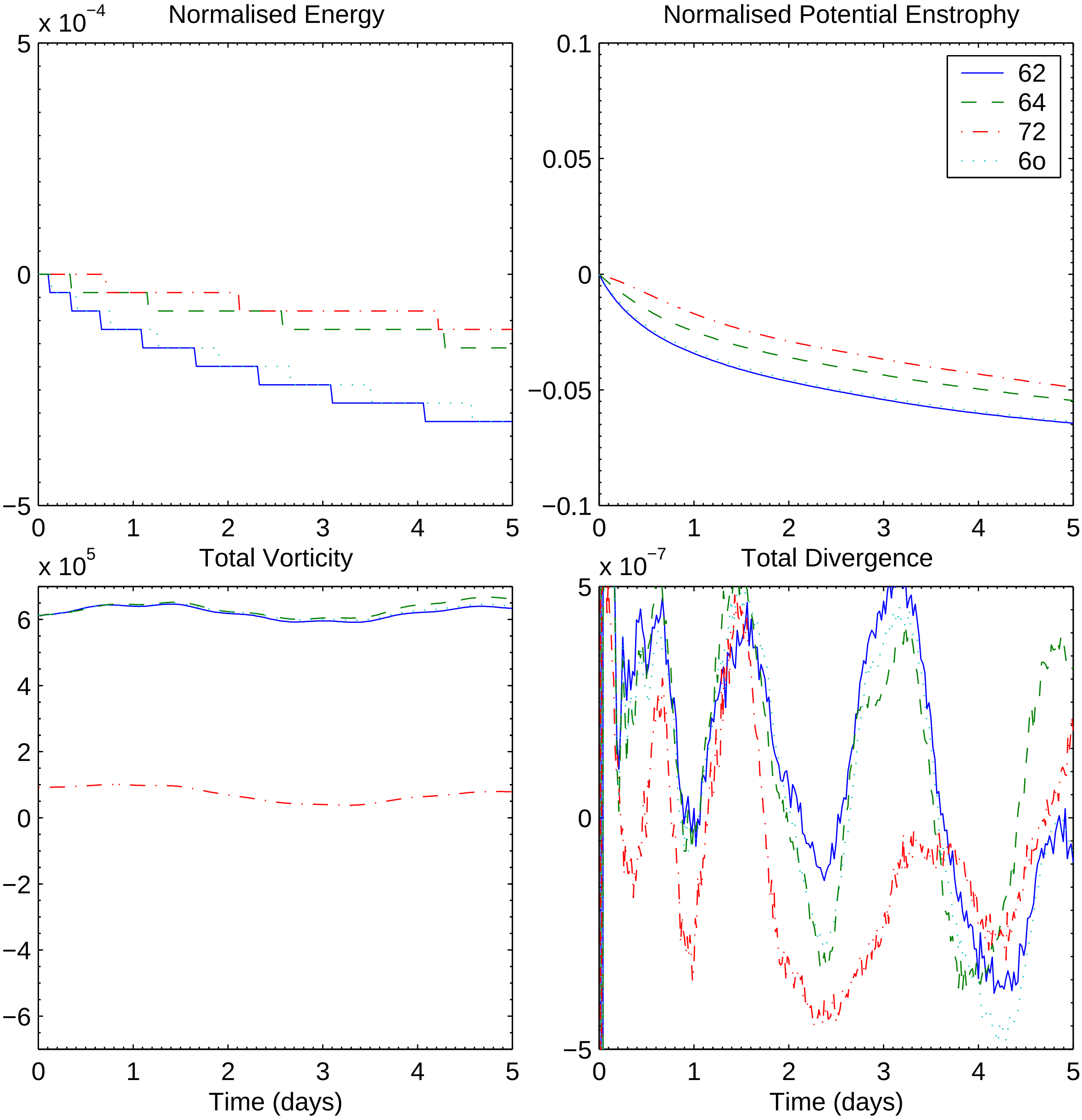


Figure 5.28: Time evolution of the global integral diagnostics sampled at each time step. The schemes are classified first by the grid resolution (6 or 7) and then by the advection scheme (2, 4, or o-the original scheme).

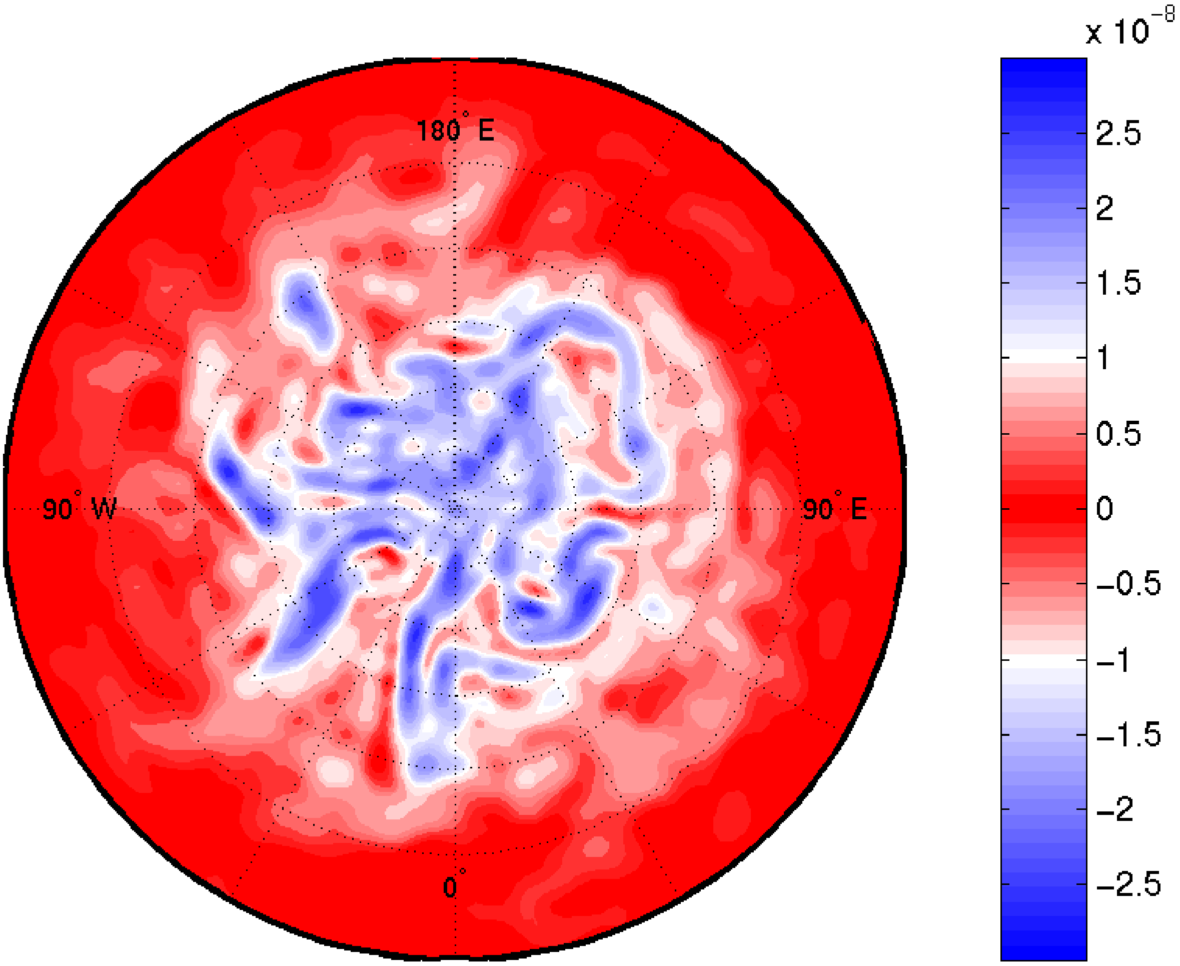


Figure 5.29: Initial PV field in the northern hemisphere for test case 7b. Contour interval is 0.25×10^{-8}, and labels are also $\times 10^{-8}$

PV fields of the models. The initial PV field is shown in Figure 5.29. This contains a lot of small scale features and "steep" gradients in the field. These features will be dissipated faster by the original and second order schemes on grid 6 which will cause the faster cascade of potential enstrophy to unresolved scales. This was seen in the global diagnostics and can also be seen in the PV fields after the first day of the run, Figure 5.30.

Comparing the PV fields of the models after five days, shown in Figure 5.31, shows the greater improvements in the higher order and higher resolution tests. Much of the detail that was still present after one day

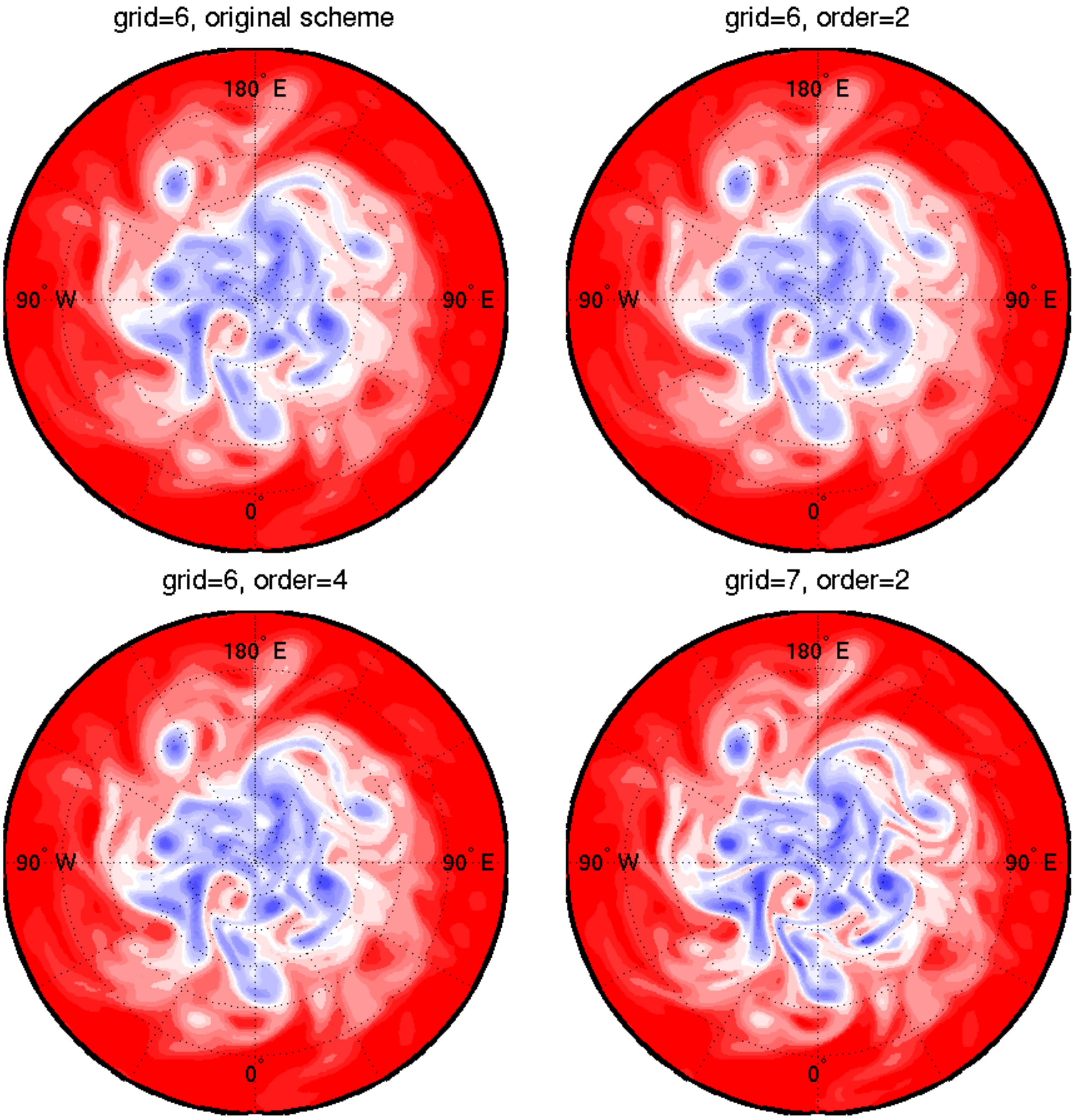


Figure 5.30: PV field in the northern hemisphere after one day of test case 7b. Contour interval is 0.25×10^{-8}, and labels are also $\times 10^{-8}$

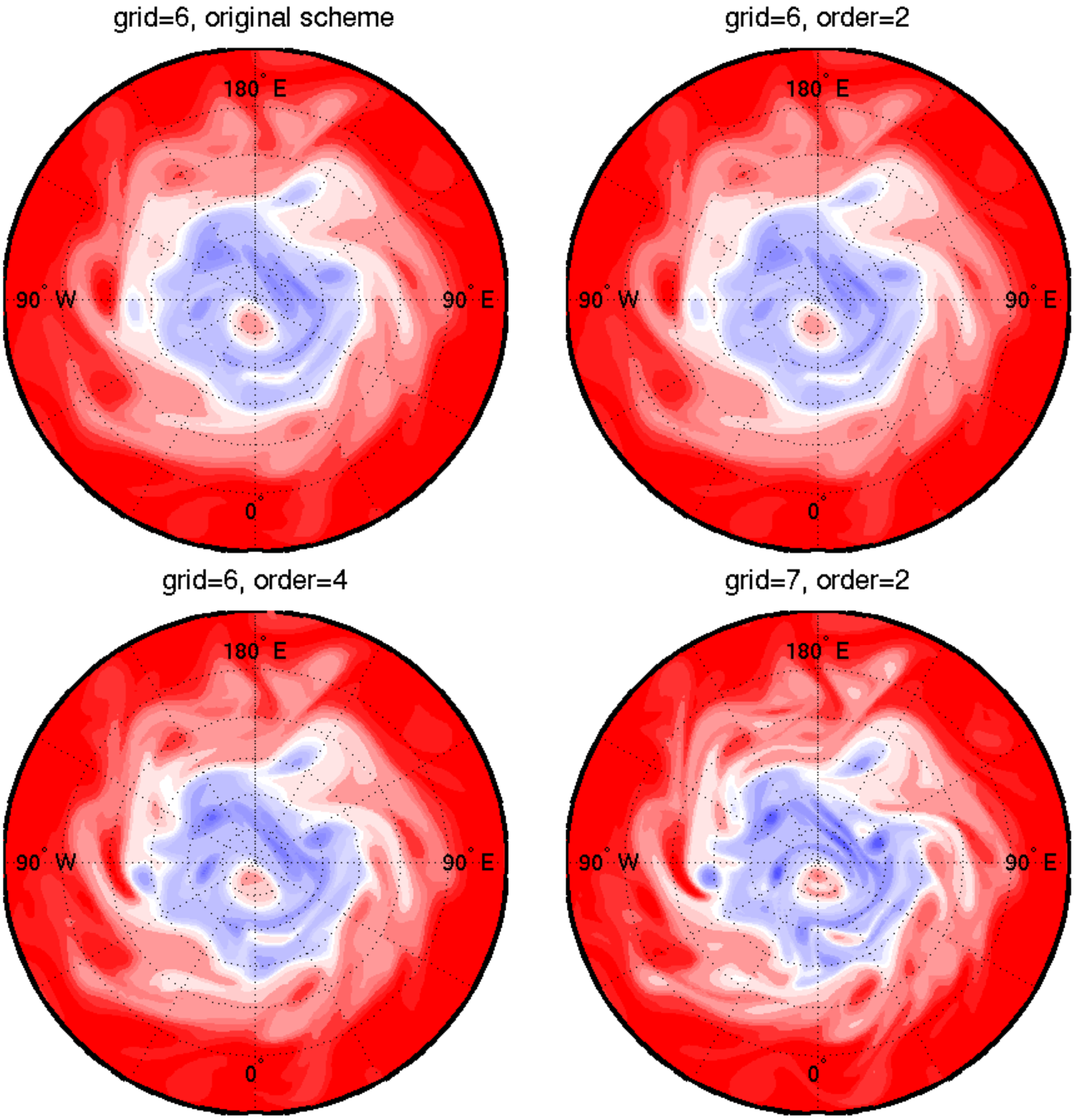


Figure 5.31: PV field in the northern hemisphere after five days of test case 7b. Contour interval is 0.25×10^{-8}, and labels are also $\times 10^{-8}$

has been lost but this has happened to a greater extent to the original and second order schemes. The higher order and higher resolution cases have better maintained the extrema in the PV field and have retained more smaller scale features. One of these can be seen at 85° west where we noted a significant improvement in the error field. This can now be seen as an improvement in the position and strength of a dipole feature in the PV field when the fourth order scheme is used.

7b: 16 January 1979

Looking at the final height fields after five days of case 7b, Figure 5.32 shows few differences between the different models. The main differences in this case are in the values of the local maxima and minima and the steepness of gradients. The only feature that is significantly different is the small trough that appears at around 80° east on grid 7. This region is marked by a strong positive-negative dipole in the error fields for each model, shown in Figure 5.33. This error is reduced as the higher order scheme and higher resolution grid are used but it is still a significant difference from the high resolution spectral result. Other regions of error corresponding to local maxima and minima are also improved by the higher order scheme and higher resolution grid.

The global errors display the same improvements, as in case 7a when the higher order advection scheme and higher resolution grid are used. This is shown here by the global wind errors in Figure 5.34 and is also the true for the height errors and global diagnostics (not shown). These results compare even more favourably with those of the spectral model. The global errors on grid 6 are similar to those of the T63 truncation model of Jakob et al. [15] and those of grid 7 compare well with those of

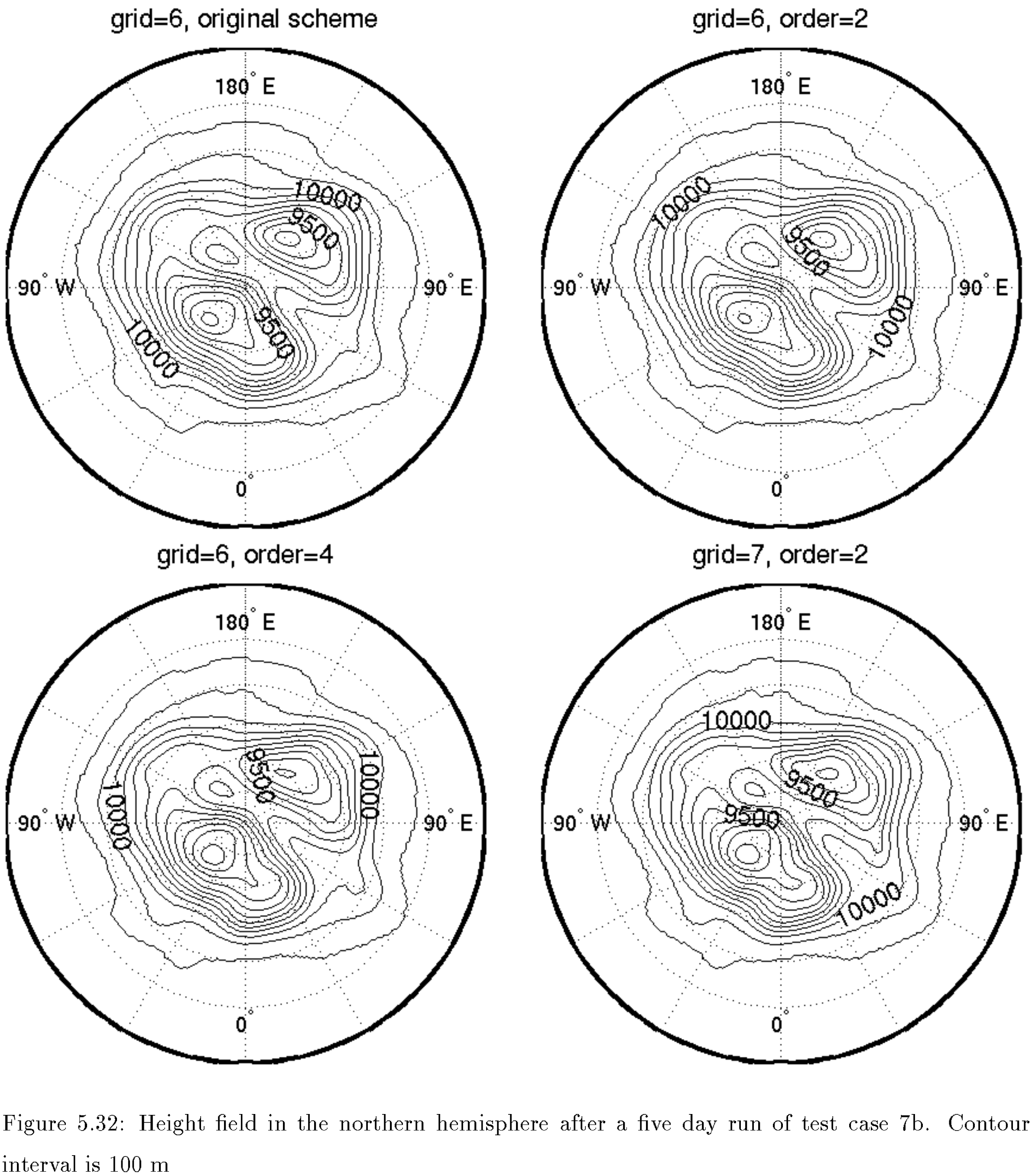
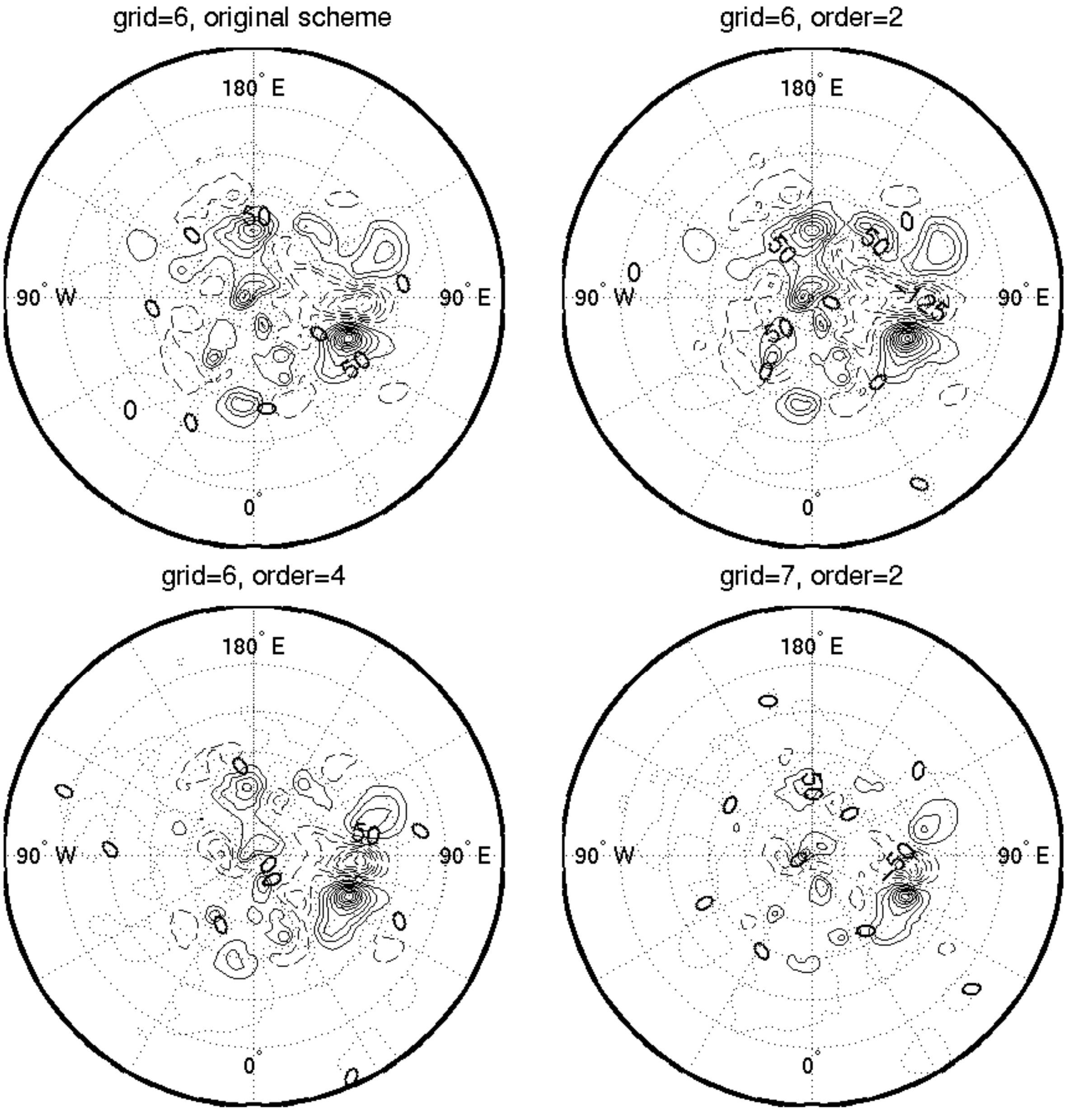


Figure 5.32: Height field in the northern hemisphere after a five day run of test case 7b. Contour interval is 100 m



grid=6, original scheme

grid=6, order=2

grid=6, order=4

grid=7, order=2

Figure 5.33: Difference between Icosahedral and spectral height fields in the northern hemisphere after a five day run of test case 7b. Contour interval is 25 m

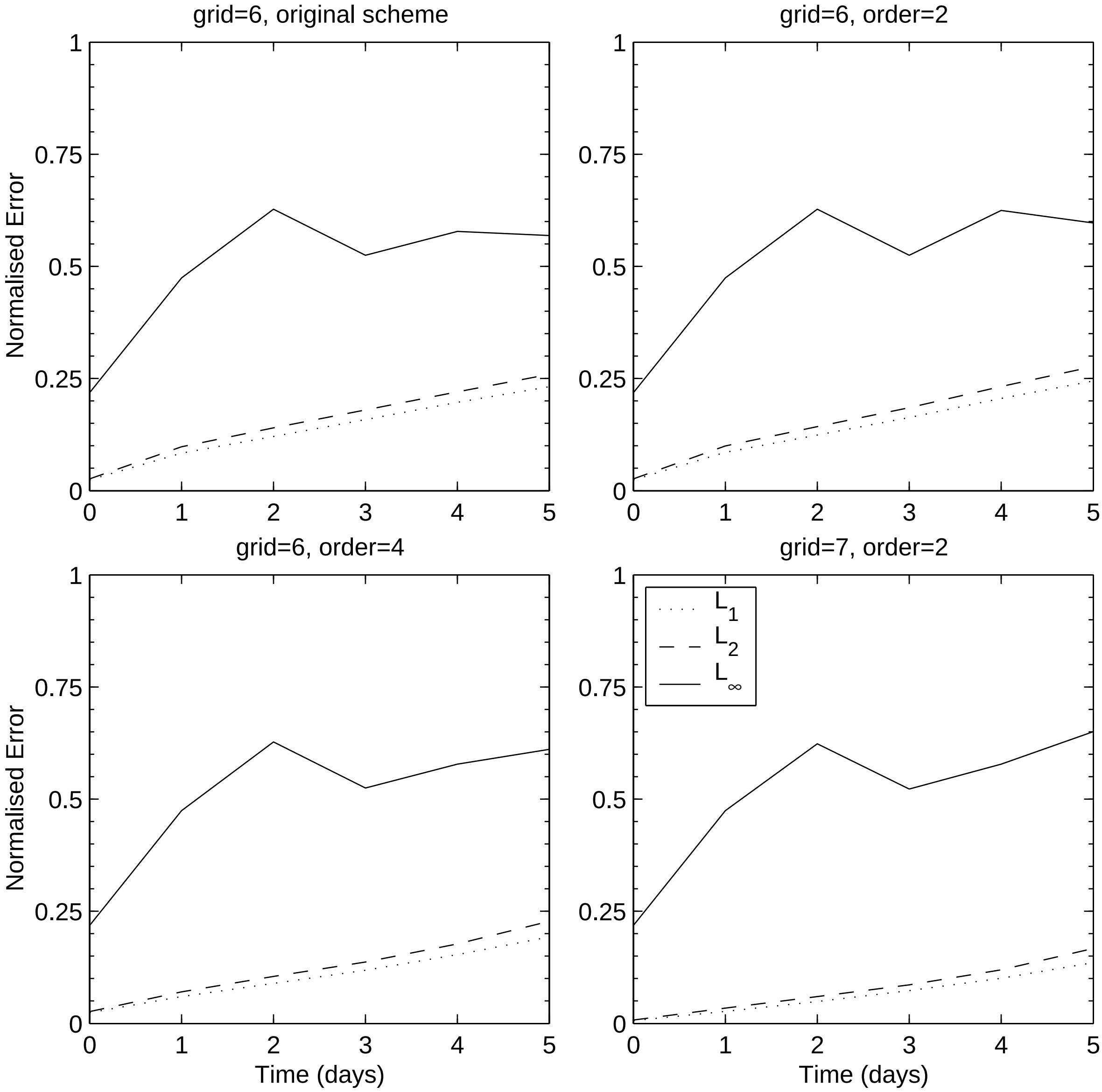


Figure 5.34: Normalised global errors in the wind field for a five day run of test case 7b. Errors are a comparison between numerical results and a high resolution spectral model, sampled every day.

the T106 version.

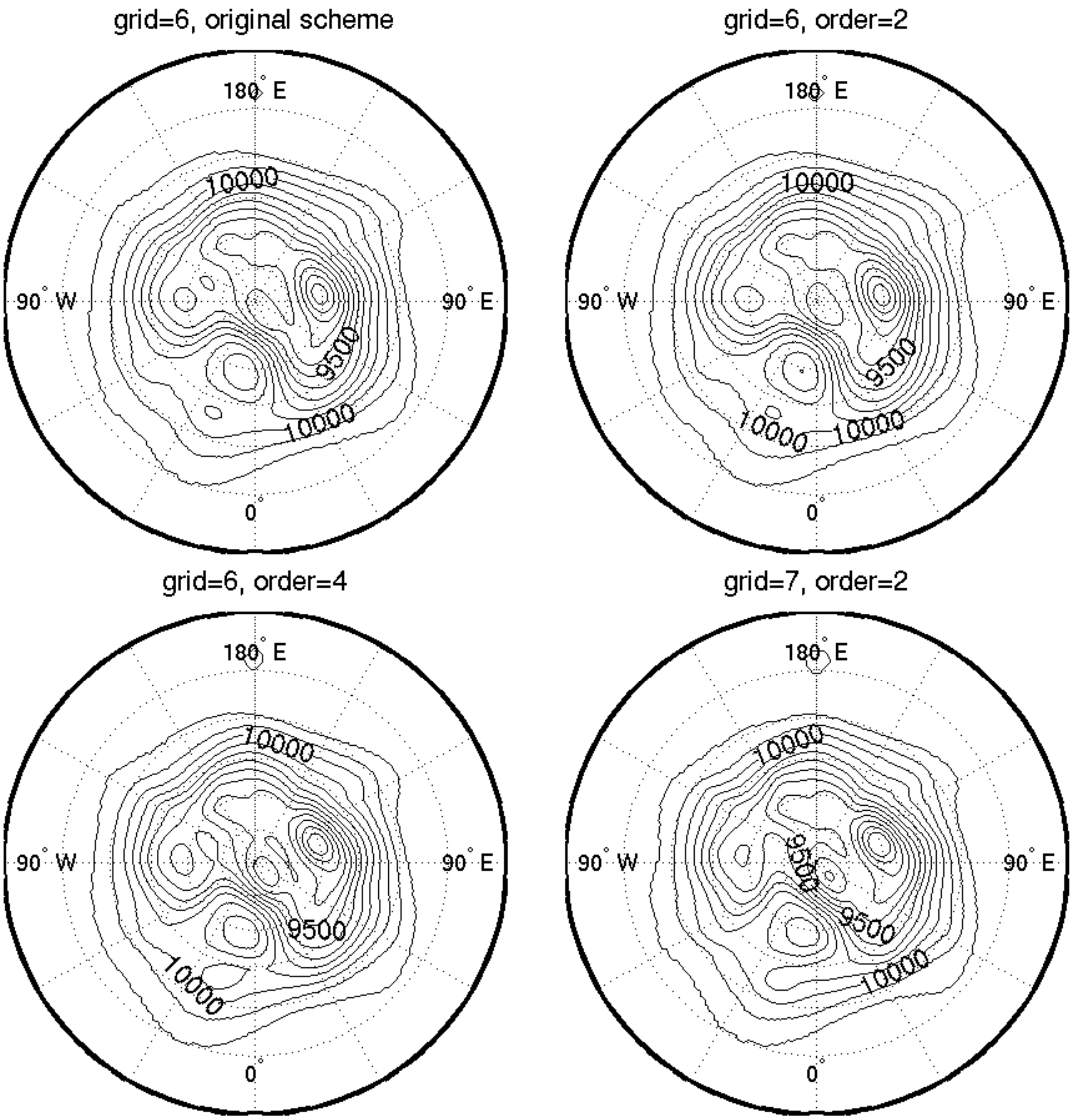
7c: 9 January 1979

The northern hemisphere height field after five days is shown in Figure 5.35. These results display some variation in the strength and location of the weak high over the pole and the low centre to its south at 100° east. The low centre corresponds with a positive-negative dipole in the error field shown in Figure 5.36 for both the original and new second order schemes. This suggests that the correct location for this low centre is further to the east. A weaker dipole suggests that the weak high should be a little to the south along the 45° latitude. Looking at the height field when the fourth order advection scheme is used shows the low centre further to the east. The low centre is in the same location on grid 7 and shows slightly less filling. The weak high is also shifted a little off the pole in this case. These improvements in the error, along with others can be seen in the error field for the fourth order and grid 7 runs.

The global height and wind errors for this case are similar in form and magnitude to the previous two cases and are not shown. The same is also true of the global diagnostics. The grid 6 results for this case are comparable with those of the T43 spectral model and grid 7 results compare well with those of the T63 truncation version.

5.9 Summary

The method of generating advection schemes developed over the previous chapters has been incorporated into a shallow water model. The original model had been tested using the test set of Williamson et al. [59] by Thuburn [52]. These tests were used again here and the results were



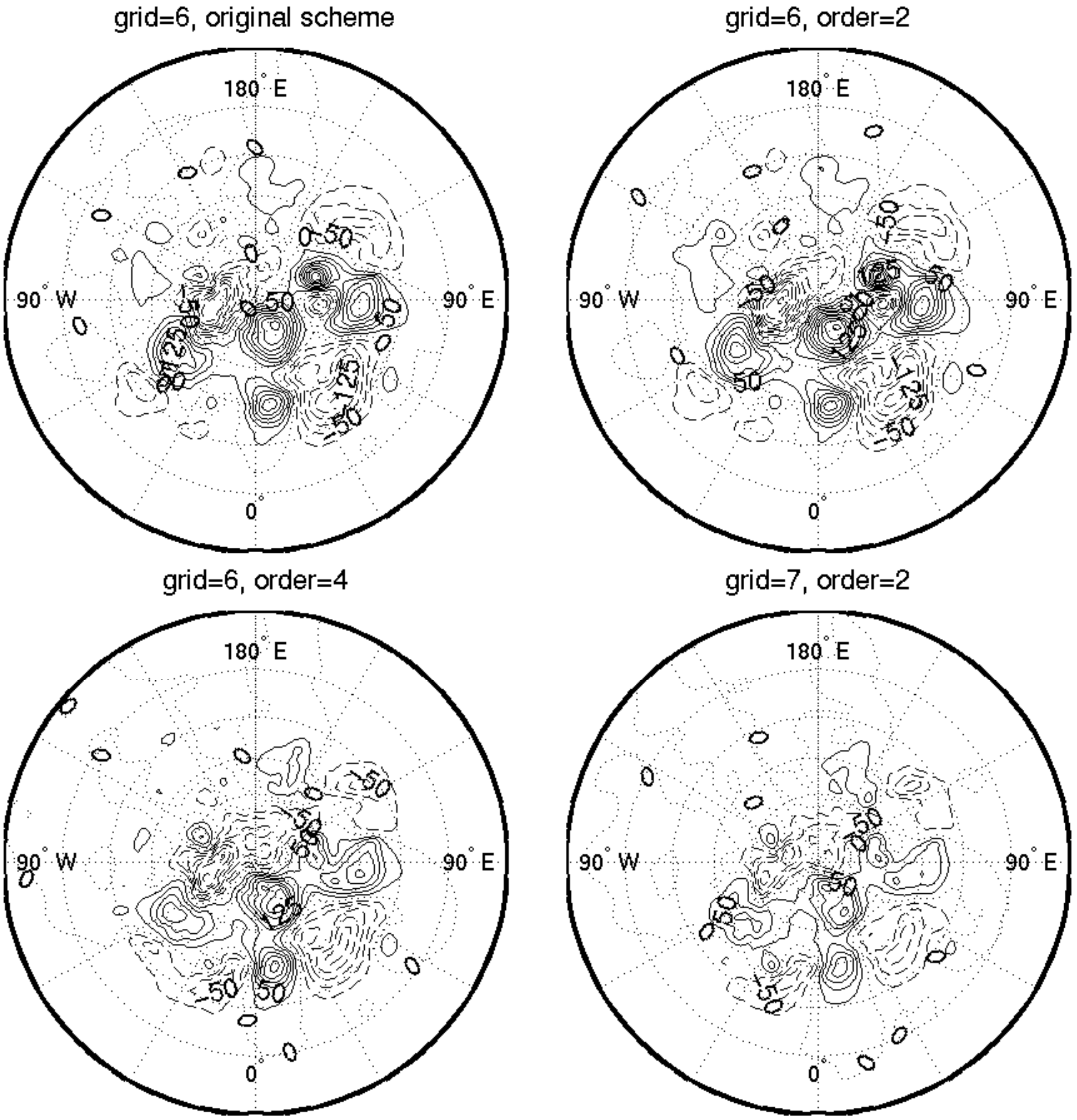
grid=6, original scheme

grid=6, order=2

grid=6, order=4

grid=7, order=2

Figure 5.35: Height field in the northern hemisphere after a five day run of test case 7c. Contour interval is 100 m


grid=6, original scheme

grid=6, order=2

grid=6, order=4

grid=7, order=2

Figure 5.36: Difference between Icosahedral and spectral height fields in the northern hemisphere after a five day run of test case 7c. Contour interval is 25 m

compared with both the original model and other published results (the icosahedral model of Heikes and Randall [12] and the spectral model of Jakob et al. [15]). The model used here used the tweaked icosahedral grid whereas the un-tweaked grid was used by Thuburn in [52]. Tests using the original advection scheme with the tweaked grid were used to distinguish between changes due to the grid and those due to the advection scheme.

The only significant differences between the results when using the original scheme on the two grids occured in tests two and three. These test cases both involved steady states of the equations, so changes to the grid changed the discrete steady state and the accuracy of the approximation to

$$\frac{\partial()}{\partial t} = 0 \; , \tag{5.34}$$

Changing the advection schemes used within the model made little difference to the overall errors but did change their spatial distribution. This suggests that the advection scheme used is not the main cause of the errors in these cases but does have some effect on the results.

In general, the original scheme and the new second order scheme performed in a similar fashion for all the tests. Both these schemes are based on the fitting of a second order polynomial over a similar region upwind of each edge. The similarity of the results suggests that the method of the fitting has little effect on the overall results of the method.

Using the fourth order scheme instead of the second order one made a larger improvement in the results of some tests than in others. In test four for example, the fourth order scheme prevented any significant filling of the low centre. In test case five however, the only significant improvement was in the region downwind of the mountain, a region dominated by advection. The improvements in test case seven appear to be small when looking at the global errors yet there are significant improvements to the 'forecast' when looking at specific regions of the model.

Increasing the resolution of the model had a greater improvement on the results than increasing the order of the advection scheme. This came at an increased computational cost as roughly four times as many cells and edges were used on grid 7 than on grid 6. In test cases two and three there was a large improvement in the results for this increase in computational cost but this was not always the case. Whilst the global errors for test case 7 were improved over the fourth order scheme on the coarser grid, the improvements were small. This suggests that the most cost effective improvement to the second order scheme on any grid is to increase the order of the scheme used. If better accuracy is all that is required then the grid should be refined and the computational price paid.

The results from the test seven cases have shown that the higher order advection schemes generated by the new method have some advantages over the lower order schemes. This is despite the global errors showing only a small improvement in many of the earlier tests. The ability of the higher order schemes to capture smaller scale features of the flow will have little effect on the results in some regions and more in others.

In the real atmosphere there are some regions that are more sensitive to perturbations in a field than others. These perturbations may be caused by any of the physical and numerical processes that are used in forecasting or, in a numerical model, by the numerical methods used. Improving the numerical methods used in these regions can make a big difference to a particular forecast for a particular region. Though we do not generally know where these regions are, improving the global advection scheme should improve the forecast in these regions. This can be done for a much smaller computational cost than increasing the resolution of the model. There are still some extensions that can be made to the advection scheme that have the potential to make it more efficient. These will be looked at briefly in the next chapter, after a review of the main results of this work.

Chapter 6

Summary and Further Work

6.1 Summary

The aim of this work was to develop accurate numerical approximations to the advection equation for use on arbitrary grids. The prime reason for wanting to do this was to improve the accuracy of geophysical models using the icosahedral-hexagonal grid on the sphere. We therefore wanted to ensure that the scheme was conservative and shape preserving as well as accurate and computationally efficient. The nature of the grids necessitated that the method was multidimensional and could be generated automatically for each grid. Using a finite volume scheme and approximating the fluxes across cell boundaries guaranteed conservation. Applying a flux limiter to the fluxes ensured shape preservation but had an adverse effect on other measures of the accuracy of some schemes.

We began by taking the method used to derive the one dimensional QUICKEST and ULTIMATE schemes of Leonard [22] and used them to generate advection schemes on irregularly spaced grids. The method involved fitting an N^{th} order polynomial over $N+1$ cells. On a regular grid (constant Δx) it is easy to write down the fluxes in terms of coefficients and the values of the advected quantity in the cells of the stencil.

On an irregular grid this calculation is much more complicated on paper but can be easily handled by computer code. The numerical results produced using the automatically generated schemes behaved similarly to the QUICKEST scheme and its higher order versions, even on irregular grids.

Leonard [23] extended his ideas to the two-dimensional UTOPIA scheme on a square grid and we did the same. This required being able to fit an N^{th} order polynomial over $\frac{1}{2}(N+1)(N+2)$ cells. A method for selecting a stencil was found that guaranteed a unique polynomial could be found on a rectangular grid. The Universal Limiter of Leonard [24] with the refinements of Thuburn [51] can be applied to the approximate fluxes generated in this way. Both the limited and un-limited schemes were tested using different orders of polynomials and different resolution grids for a range of tests.

The accuracy of the schemes improved as the order of the interpolating polynomial was increased and as the resolution increased. The computational cost of the schemes also increased along with the polynomial order and the grid resolution. In non-deformational flows, the relationship between the accuracy and the computational cost was broadly the same regardless of whether the polynomial order or grid resolution was changed for the un-limited schemes. This means that a given accuracy can be achieved by either increasing the order of the advection scheme or the resolution of the grid. Whichever method is used, the computational cost will be the same. When the limiter was used it had a greater detrimental effect on the accuracy of the schemes using high order polynomials. This led us to the conclusion that the most efficient schemes of this type use between second and fourth order polynomials. It is these schemes that best balance the accurate approximation of the advected field with the need to use the limiter to ensure shape preservation.

Trying to apply the same method of generating advection schemes to triangular grids provided one significant new problem, selecting a stencil over which to fit a polynomial. The problem is that there are some grids for which there is no obvious way to choose between two cells for inclusion in the stencil without considering the flow field. Using the flow field in this way is not a satisfactory solution to the problem since it requires that a computationally expensive part of the algorithm be recomputed whenever the flow field changes. Instead we select a stencil with more than the required number of cells and find a 'best fit' polynomial over this stencil. The best fit is found using a weighted least squares cost function where the weights are chosen to favour the fit over the central cell in the stencil.

The advection schemes generated by this method show a similar relationship between computational cost and accuracy as was seen on the rectangular grids. The trend is for an improvement in accuracy as the order of the polynomial or the resolution is increased. The results again suggest that a polynomial between second and fourth order should be used for maximum efficiency when the limiter is used. The limiter also causes some anisotropic distortion, dependent on the flow angle relative to the grid in this case. This was the only grid for which this was observed.

The same method was then applied to the icosahedral-hexagonal grid on the sphere. The only difficulty in this case was that the grid was spherical whilst the polynomials and integrations had previously been formed on two dimensional planes. This problem was overcome by projecting the region around each cell onto a plane for the polynomial interpolation and approximation of the fluxes. The results of test cases using schemes generated by this method behaved in the same way as on the rectangular and triangular grids. The general trend was for accuracy to improve as the polynomial order or resolution were increased. For best efficiency, comparing the computational cost and accuracy, schemes of between second and fourth order should be used.

The final method of generating advection schemes follows the following pattern;

(i). **Select a stencil:** Begin with one cell and add all cells neighbouring those in the stencil until there are more cells than there are terms in polynomial that is to be 'fitted'.

(ii). **Set up polynomial matrix:** Integrate a general polynomial over each cell in the stencil to get $\mathsf{B}\mathbf{a}$ where $\mathbf{a}$ is the vector of the coefficients of the polynomial. Apply a weighting to B by multiplying by a diagonal matrix of the weights.

(iii). **Singular value decomposition:** Decompose the weighted matrix B_w so that the coefficients, $\mathbf{a}$, of the best fit polynomial can be found from $\mathsf{V}_w\Sigma_w\mathsf{U}_w^T\boldsymbol{\phi}_w$, where $\boldsymbol{\phi}_w$ is the (weighted) amount of the advected quantity in each cell in the stencil.

(iv). **Calculate $\boldsymbol{\phi}$ coefficients:** Integrate the polynomial defined by the relevant $\mathbf{a}$ over the approximate region swept across the edge (in terms of general normal and tangential wind components) to generate the matrix G_w. The relevant $\mathbf{a}$ is the one for the cell upwind of the edge for which the flux is being approximated, this is done treating $\boldsymbol{\phi}$ as an unknown vector.

(v). **Calculate approximate flux:** Substitute in the appropriate values of ϕ and the wind to calculate the flux across the edge.

(vi). **Apply flux limiter:** If required, the flux limiter described in Section 2.3.2 can be applied to adjust the fluxes.

(vii). **Update:** The flux across each edge can now be removed from each upwind cell and placed in the downwind cell.

(viii). **Repeat:** Steps v-vii should be repeated for each time step if the flow continues in the same direction across the edge. Step iv also needs repeating if a different cell becomes the upwind cell.

The schemes generated by this method were then used in a shallow water model in an attempt to discover if the improved treatment of advection improved the overall results. The model that was used had already been developed and tested by Thuburn [52] on an icosahedral-hexagonal grid. Two changes were made from the model reported in [52] and [53], one being the new advection scheme and the other the use of a tweaked icosahedral grid similar to that suggested by Heikes and Randall [11]. The results did show several differences between the new fourth and second order schemes as well as differences from the original scheme that had been used in the model.

The differences in the results caused by the use of the tweaked grid were small and confined mainly to the steady state tests. The more realistic tests showed a small but significant improvement in the results when a fourth order scheme was used over a second order scheme. A much larger improvement in accuracy could be gained by increasing the resolution of the grid. Not only is this computationally expensive but it may not be entirely necessary. The results of using real data in the shallow water model showed that many of the local improvements in the higher resolution model could be gained by using a higher order advection scheme on a coarser grid.

The greatest strength of this method is probably it's flexibility. It is not restricted to any particular grid or to one particular 'order'. This flexibility has however, required that a number of approximations be made and that a significant amount of computation is required for an evolving flow field. It is this need to recalculate coefficients that means the method is not yet suitable for use in high resolution models with varying flow fields, such as a global forecast model. The flexibility of the scheme does enable a number of potential improvements to the way the scheme is implemented. These improvements can increase the efficiency of the scheme but may have some effect on the accuracy.

6.2 Further Work

This thesis has concentrated on the development of a numerical advection scheme that can be used in global models. The ultimate goal of such a scheme is its use in a global numerical weather prediction or climate model. The main problem with the current scheme is computational efficiency. The requirement that the coefficients of the advection scheme need recalculating whenever the flow speed or direction changes is a significant restriction to achieving computational efficiency.

One way around this problem would be to divide the wind field into 'bins' based on the magnitude of the normal and tangential components of the flow at each edge. Step v of the scheme would then only be recalculated when the wind regime moved to a different bin. This would speed up the overall method but would probably also have a detrimental effect on the accuracy. Fewer, bigger bins would mean a greater speed-up but less accuracy. If storage were not a problem then maximum speedup could be gained by storing the coefficients for each bin rather than recalculating them when needed. The relationship between the size of the bins and the accuracy would need to be investigated by running similar tests to those in previous chapters.

Another way of improving the efficiency of the method would be to only use the higher order schemes in regions where this will make a significant improvement in the results. These regions are likely to be where the field is less smooth and a more accurate interpolation of the field will significantly reduce diffusion. By including a simple switch into the model so that higher order schemes are used in regions with steep gradients, much of the accuracy of the higher order scheme can be gained without all the added computational cost. This technique is known as p-type refinement.

A similar idea could be used when several different fields need to be advected. Relatively smooth fields (e.g.height) could use a lower order advection scheme than fields that contain more sharp gradients (e.g.PV). In terms of the shallow water model used in this work we could use a second order scheme for the smooth height field and a fourth order scheme for the PV field. This would require more initial computation and storage but would speed up the time stepping over the fourth order scheme. The loss of accuracy caused would need investigating, again by using similar tests to before.

The accuracy of the schemes depends not only on producing an accurate reconstruction of the advected field but also on the approximation

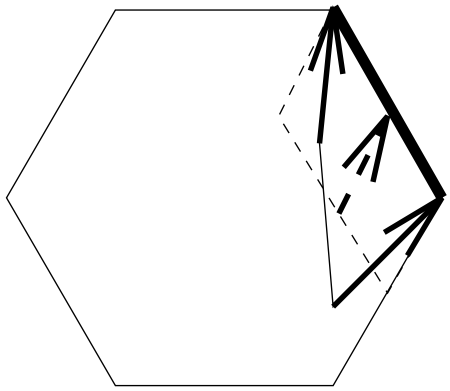


Figure 6.1: The region swept across each edge. The dotted region shows the current approximation to this region. The solid region gives a more accurate representation of this region.

of the region swept across each edge. The spatial interpolations of the advected field use high order polynomials to increase accuracy. The region swept across the edge is approximated simply as a projection of the vertices of the edge in an upwind direction. The direction and magnitude of the wind are found at the centre of each edge by a second order approximation to the derivative of the streamfunction, shown as the doted region in Figure 6.1.

A more accurate approximation of the region swept across the edge could be found by approximating the flow at the vertices and using this to find the required region, the solid region in Figure 6.1. On the icosahedral grid this could be done either by simply interpolating the winds from the edges or by finding an approximate reconstruction of the streamfunction and calculating its gradient at the vertices. Linear interpolation can be used from the three edges or cells associated with each vertex. Quadratic interpolation of the streamfunction could be used for greater accuracy by including the three cells that share two edges with the three cells surrounding the vertex.

Meteorological models are also often used in the studies of particular phenomenon or investigations of new theories and ideas, e.g. a cloud model or the reformulation of equations into new variables. The advantage of this scheme in this case is that the accuracy can be easily improved by using a higher order polynomial. A particular accuracy may be required to capture a certain feature but this may not be known in advance. This method of generating advection schemes allows greater accuracy to be achieved without the need to increase the grid resolution. The overall improvement in accuracy will depend on the relative importance of advective processes in the model. Each model will behave differently as the advection scheme is changed or the resolution is increased, which models will benefit from using this method is difficult to tell in advance. Performing tests comparing accuracy and computational cost on such a model may help to explain this behaviour.

This method of generating advection schemes may also be of particular use in oceanography. Oceans pose two particular problems that are not present in the atmosphere. The first of these is the domain over which the modelling is performed, which not only contains horizontal boundaries but ones that are highly irregular. Using an irregular grid over the domain is a technique that is not widely used but one which this method could easily cope with. The other problem is that the dynamics of the oceans generate strong currents along the western boundaries of ocean basins. These are very important features but pose difficulties for modelling because of their dimension and the steep gradients they cause in some model variables. This method has two benefits in this case, the first is its ability to capture sharper gradients using higher order polynomials. The second benefit is again its ability to handle irregular grids which may be useful for providing increased resolution.

Some of the ideas used with this method could be used to develop a conservative semi-Lagrangian scheme. Conservative semi-Lagrangian schemes have proved difficult to develop though there is growing success in this area, e.g. Nair and Machenauser [33]. Using an exactly fitted local polynomial resconstruction, the total tracer in each cell can be calculated from the region that each cell occupied at the previous time step. Calculating this region and integrating over several local polynomials will cause some problems for this method that must be overcome. A globally fitted function may help with the second of these problems but finding a suitably global function may be difficult or computationally expensive.

These adaptions to the present method are largely designed to make the schemes more efficient. We have already shown that the improvements in the results gained by using higher order polynomials are significant. At present these are somewhat expensive to compute but there are several ways which the efficiency of the scheme may be improved. The automatic generation of the advection schemes allows great flexibility in the way the scheme is applied both in terms of the grids used and the accuracy of the schemes. It is the relation between the accuracy of the schemes and the computational cost of using them that must be exploited to make the method more efficient. If this can be done then this method can be used to advantage in many applications within meteorology and in other fields.

Bibliography

[1] J. R. Bates, Y. Li, A. Brandt, S. F. McCormick, and J. Ruge. A global shallow-water numerical model based on the semi-Lagrangian advection of potential vorticity. *Quaterly Journal of the Royal Meteorological Society*, 121:181–2005, 1995.

[2] J. P. Boris and D. L. Book. Flux corrected transport. I: SHASTA, a fluid transport algorithm that works. *Journal of Computational Physics*, 11:38–69, 1973.

[3] J. P. Boris and D. L. Book. Flux corrected transport. III: Minimal error fct algorithms. *Journal of Computational Physics*, 20:397–431, 1976.

[4] J. P. Boris, D. L. Book, and K. Hain. Flux corrected transport. II: Generalizations of the method. *Journal of Computational Physics*, 11:248–283, 1973.

[5] M. J. P. Cullen. Finite difference methods. In *Numerical methods in atmospheric models*, pages 87–102. ECMWF, Sept 1991.

[6] ECMWF. *Numerical methods in atmospheric models*, Sept 1991.

[7] A. R. Gregory and V. West. The sensitivity of a model's tape-recorder to the choice of advection scheme. *Quaterly Journal of the Royal Meteorological Society*, 2002. preprint.

[8] A. Harten, B. Engquist, S. Osher, and S. R. Chakravarthy. Uniformly high order essentially non-oscillatory schemes, III. *Journal of Computational Physics*, 71:231–303, 1987.

[9] P. H. Haynes and M. E. McItyre. On the conservation and impermeability theorems for potential vorticity. *Journal of Atmospheric Science*, 47:2021–2031, 1990.

[10] R. Heikes and A Randall. Numerical integration of the shallow-water equations on a twisted icosahedral grid. part i: Basic design and results of tests. *Monthly Weather Review*, 123(ii):1862–1880, 1995.

[11] R. Heikes and A Randall. Numerical integration of the shallow-water equations on a twisted icosahedral grid. part II: a detailed description of the grid and an analysis of numerical accuracy. *Monthly Weather Review*, 123(ii):1881–1887, 1995.

[12] R. Heikes and D. A. Randall. The shallow water equations on a spherical geodesic grid. Technical Report 524, Colorado State University, Department of Atmospheric Science, April 1993.

[13] B. J. Hoskins. Stability of the Rossby-Haurwitz wave. *Quaterly Journal of the Royal Meteorological Society*, 99:723–745, 1973.

[14] B. J. Hoskins, M. E. McIntyre, and A. W. Robertson. On the use and significance of isentropic potential vorticity maps. *Quaterly Journal of the Royal Meteorological Society*, 111:877–946, 1985.

[15] R. Jakob, J. J. Hack, and D. L. Williamson. Solutions to the shallow water test set using the spectral transform method. Technical Report TN-388+STR, NCAR, May 1993.

[16] H. O. Kreiss and J. Olger. Comparison of accurate methods for the integration of hyperbolic equations. *Tellus*, 24:199–215, 1972.

[17] B. Van Leer. Towards the ultimate conservative difference scheme I: The quest for monotonicity. In *Lecture Notes in Physics*, volume 18, page 165. 1973.

[18] B. Van Leer. Towards the ultimate conservative difference scheme II: Monotonicity and conservation combined in a second order scheme. *Journal of Computational Physics*, 14:361–370, 1974.

[19] B. Van Leer. Towards the ultimate conservative difference scheme III: Upstream centred finite-difference schemes for ideal compressible flow. *Journal of Computational Physics*, 23:263–275, 1977.

[20] B. Van Leer. Towards the ultimate conservative difference scheme IV: A new approach to numerical convection. *Journal of Computational Physics*, 23:276–299, 1977.

[21] B. Van Leer. Towards the ultimate conservative difference scheme V: A second order sequel to Godunov's method. *Journal of Computational Physics*, 32:101–136, 1979.

[22] R. P. Leonard. *Universal Limiter for Transient Interpolation Modelling of the Advective Transport Equations: The ULTIMATE Conservative Difference Scheme.* NASA technical memorandum: 100916. National Aeronautics and Space Administration, 1988.

[23] R. P. Leonard, M. K. MacVean, and A. P. Lock. Positivity preserving numerical schemes for multidimensional advection. Technical Report 106055 ICOMP-93-05, NASA, 1993.

[24] R. P. Leonard, M. K. MacVean, and A. P. Lock. The flux-integral method for multidimensional convection and diffusion. *Applied mathematical modelling*, 19(6):71–100, 1995.

[25] R. J. LeVeque. *Numerical Methods for Conservation Laws*, chapter 16, pages 173–192. Lectures in Mathematics - ETH Z urich. Birkh auser Verlag, 1992.

[26] P. Lynch. Richardson's barotropic forecast: A reappraisal. *Bulletin of the American Meteorological Society*, 73:35–47, 1992.

[27] B. Machenhauer. Spectral methods. In *Numerical methods in atmospheric models*, pages 3–85. ECMWF, Sept 1991.

[28] D. Majewski, D. Liermann, P. Prohl, B. Ritter, M. Buchhold, T. Hanisch, G. Paul, W. Wergen, and J. Baumgardner. The operational global icosahedral-hexagonal grid point model GME: Description and high resolution tests. *Monthly Weather Review*, 130:319–338, 2002.

[29] D. Majewski, D. Liermann, P. Prohl, B. Ritter, M. Buchhold, T. Hanisch, G. Paul, W. Wergen, and J. Baumgardner. The global icosahedral-hexagonal grid point model GME - operational version and high resolution tests. In *Developments in numerical methods for very high resolution global models*, pages 47–91. ECMWF, 2000.

[30] Y. Masuda and H. Ohnishi. An integration scheme of the primitive equations model with an icosahedral-hexagonal grid system and its application to the shallow water equations. In *Short- and Medium-Range Numerical Weather Prediction*, pages 317–323. Japan Meteorological Society, 1986.

[31] A. McDonald. Semi-lagrangian methods. In *Numerical methods in atmospheric models*, pages 257–271. ECMWF, Sept 1991.

[32] M. E. McIntyre. Atmospheric dynamics: Some fundamentals with observational implications. In J. C. Gille and G. Visconti, editors, *International School Physics "Enrico Fermi", CXV course.*

[33] R.D. Nair and B. Machenhauer. The mass-conservative cell-integrated semi-Lagrangian advection scheme on the sphere. *Monthly Weather Review*, 130:649–667, 2002.

[34] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in Fortran 77: The Art of Scientific Computing*, chapter 2, pages 22–98. Press Syndicate of the University of Cambridge, second edition, 1996.

[35] P. J. Rasch. Conservative shape preserving two-dimensional transport on a spherical reduced grid. *Monthly Weather Review*, 122:1337–1350, 1994.

[36] L. F. Richardson. *Weather prediction by numerical processes.* Cambridge University Press, 1922.

[37] T. D. Ringler, R. P. Heikes, and D. A. Randall. Modelling the atmospheric general circulation using a spherical geodesic grid: A new class of dynamical cores. *Monthly Weather Review*, 128:2471–2490, 2000.

[38] R. B. Rood. Numerical advection algorithms. *Review of Geophysics*, 25:71–100, 1987.

[39] R. Sadourny, A. Arakawa, and Y. Mintz. Integration of the non-divergent barotropic vorticity equation with an icosahedral-

hexagonal grid for the sphere. *Monthly Weather Review*, 96:351–356, 1968.

[40] Chi-Wang Shu. Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. Technical Report ICASE 97-65, NASA, November 1997.

[41] P. K. Smolarkiewicz. The multi-dimensional Crowley advection scheme. *Monthly Weather Review*, 113:1968–1983, 1982.

[42] P. K. Smolarkiewicz. Reply to comments on "Smolarkiewicz's deformational flow". Monthly Weather Review, 1987. pages 901-902.

[43] A. Staniforth and J. Côté. Semi-lagrangian integration schemes for atmospheric models: A review. *Monthly Weather Review*, 1991.

[44] A. Staniforth, J. Côté, and J. Pudykiewicz. Comments on "Smolarkiewicz's deformational flow". Monthly Weather Review, 1987. pages 894-900.

[45] A. N. Staniforth and J. Mailhot. An operational model for regional weather forecasting. *Computational Mathmatics Applications*, 16:1–22, 1988.

[46] G. Strang and G. Fix. *An analysis of the finite element method.* Prentice-Hall, Englewood Cliffs, N.J., 1973.

[47] A. Suresh and H. T. Huynh. Accurate monotonicity-preserving schemes with Runge-Kutta time stepping. *Journal of Computational Physics*, 136:83–99, 1997.

[48] P. K. Sweby. High resolution TVD schemes using flux limiters. *Lectures in Applied Mathematics*, 22:289–309, 1985.

[49] L. L. Takacs. A two-step scheme for the advection equation with minimized dissipation and dispersion errors. *Monthly Weather Review*, 113:1050–1065, 1985.

[50] C. Temperton. Finite element methods. In *Numerical methods in atmospheric models*, pages 103–117. ECMWF, Sept 1991.

[51] J. Thuburn. Multidimensional flux-limited advection schemes. *Journal of Computational Physics*, 123:74–83, 1996.

[52] J. Thuburn. A PV-based shallow water model on a hexagonal-icosahedral grid. Technical Report 40, CGAM, 1996.

[53] J. Thuburn. A PV-based shallow water model on a hexagonal-icosahedral grid. *Monthly Weather Review*, 125:2328–2347, 1997.

[54] J. Thuburn and Y. Li. Numerical simulations of Rossby-Haurwitz waves. *Tellus*, 52A:181–189, 2000.

[55] H. Tomita, M. Tsugawa, M Satoh, and K. Goto. Shallow water model on a modified icosahedral geodesic grid by using spring dynamics. *Journal of Computational Physics*, 174:579–613, 2001.

[56] J. Truesdale. ftp://ftp.cgd.ucar.edu/pub/jet/shallow/.

[57] A. Wiin-Nielsen. The birth of numerical weather prediction. *Tellus*, 43AB:36–52, 1991.

[58] D. L. Williamson. Integration of the barotropic vorticity equation on a spherical geodesic grid. *Tellus*, 20:642–653, 1968.

[59] D. L. Williamson, J. B. Drake, J. J. Hack, R. Jakob, and P. N. Swarztrauber. A standard test set for numerical approximations to

the shallow water equations in spherical geometry. *Journal of Computational Physics*, 102:211–224, 1992.

[60] M. A. Wlasak. *The examination of balanced and unbalanced flow using potential vorticity in atmospheric modelling.* PhD thesis, University of Reading, 2002.

[61] S. T. Zalesak. Fully multidimensional flux corrected transport algorithms for fluids. *Journal of Computational Physics*, 31:335–362, 1979.