University of Reading

School of Mathematics, Meteorology and Physics

Discretization On Non-uniform Meshes: Tests solving shallow-water equations

By

FAWZI B ALBUSAIDI

August 2008

This dissertation is submitted to the Department of Mathematics and Meteorology
in partial fulfilment of the requirements for the degree of Master of Science

Acknowledgments

In the Name of Allah, the Compassionate, the Merciful. I thank Allah for given me knowledge, strength and patience to finish my MSc.

Personally, I thank my beloved wife Shaima Albusaidi for her love, encouragement and patience. Also I would like to thank Mr. Ahmed Alharthy for his encouragement and support even from a distance.

Finally, I thank the most two people who took me all the way to the end of this dissertation Dr.Hilary Weller for the good guidance, explanation and long lasting patience and to Professor.Mike Baines for his supportive, right ideas and guidance.

Declaration

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

Abstract

In this dissertation, we study transition between coarse mesh and fine mesh. We introduce a blend of the linear and the quadratic schemes which gives more accurate results than just using linear scheme on non-uniform hexagonal icosahedral mesh. We have tested the blend scheme using two tests of different complexity of Williamson [1], test case 2, the global steady state geostrophically balanced flow and test case 5, zonal flow over an isolated mountain.

We have run these two tests using a shallow-water equations solver called AtmosFOAM [2, 4] on spherical meshes in cartesian co-ordinates using the finite volume method, with the three schemes, the linear scheme, the quadratic and the blend scheme and we compare the results. The blend scheme uses the linear scheme where the mesh is nearly uniform and uses the quadratic elsewhere in order to improve accuracy over linear for minimal extra cost.

Contents

Acknowledgments **2**

Abstract **3**

1 Introduction **8**

1.1 General Background 9

1.2 Background to the Dissertation 10

2 Discretization on Non-uniform Grids **13**

2.1 Non-uniform Grid 14

2.2 Numerical Method 16

2.2.1 Linear Approximation 16

2.2.2 Quadratic Approximation 16

2.2.3 Results 18

2.3 Least Squares Approach 19

2.4 Error Analysis 21

2.5 Conclusion 22

3 The Blending Scheme 23

3.0.1 Setting the Blend Scheme . . . 23

3.1 Testing The Blend Scheme . . . 25

4 Shallow-Water Equations Test Cases 28

4.1 AtmosFOAM . . . 29

4.2 Main Setup . . . 31

4.3 Williamson *etal* test Case 2 . . . 32

4.3.1 Results . . . 32

4.4 Williamson *etal* test Case 5 . . . 35

4.4.1 Results . . . 36

4.5 Conclusion . . . 39

5 Conclusion 40

5.1 Future Work . . . 41

Bibliography 42

List of Figures

2.1 Showing the test setup domain of the staggered grid with a region of non-uniform grid intervals. 15

2.2 Illustrate how the discretization is done with Arakawa C-grid. 16

2.3 Plots showing the errors of using linear and quadratic approximation from calculating the $\frac{dh}{dx}$ at u points. 18

2.4 Illustration of the grid points intervals which been tested using Least Square method. 19

2.5 Illustration to calculate the accuracy of the non-uniform staggered grids. . . 21

3.1 graph illustrates the way the Blend factor were calculated, where w is the linear weight for each face. 24

3.2 Graphs showing the output of various values of w. 24

3.3 Graphs showing the errors from calculating the gradient of the quadratic equation using the linear and the blend schemes. 26

3.4 Graphs showing the errors from calculating the gradient of the cubic equations using the linear and the blend schemes. 27

4.1 Components of the cells in AtmosFOAM. 29

4.2 Height norm errors $l_2(h)$ against CPU time taken after 5 days after applying different combination of linear and quadratic scheme to the five terms gradient, divergence, Laplacian, Interpolation method and height gradient. Errors calculated as differences with respect to the initial condition. 33

4.3 Errors for the Uniform and Non-uniform hexagonal icosahedral mesh after 5 days. Errors calculated as differences with respect to the initial conditions. The errors rang from -15 - 15 34

4.4 (a) Time steps in seconds against height error norms. (b) Errors for the Non-uniform hexagonal icosahedral mesh after 5 days. Errors calculated as differences with respect to initial conditions. The errors rang from -15-15. . 35

4.5 Height norm errors $l_2(h)$ against CPU time taken after 15 days after applying a combination of linear and quadratic scheme to the five terms gradient, divergence, Laplacian, Interpolation method and height gradient. Errors calculated as differences with respect to the given reference solutions. 36

4.6 Errors for the Uniform and Non-uniform hexagonal icosahedral mesh after 15 days. Errors calculated as differences with respect to the given reference solutions. The errors rang from -160 - 160 38

4.7 (a) Time steps in seconds against height error norms. (b) Errors for the Non-uniform hexagonal icosahedral mesh after 15 days. Errors calculated as differences with respect to the given reference solutions. The errors rang from -160 - 160 39

Chapter 1

Introduction

We study numerical modeling of the atmosphere using mesh adaptation, in particular the transition between coarse and fine meshes. To obtain high resolution over the area of interest operationally we run two atmospheric models, one global low resolution and the other limited high resolution models. However, this is very expansive in regards to CPU time cost. Also as time goes by global models get higher in resolution, which means extra cost of CPU time. The need for adaptive and variable modeling increases to compete with the same accuracy and efficiency as the uniform grids models. However, there are still a number of challenges to be looked at before this technique can compete with the well known global atmospheric models on uniform grids.

One of the challenges that we are going to concentrate on, is well described in the paper [8], the authors show that spurious wave scattering do occurs on a nonuniform grid where the hyperbolic equations are approximated with finite differences, when the grid size function as compared to the position has a discontinuous first derivative, as in equation 1.1 below. Then the authors derived the reflection ratio expression and verified it by experimental data.

$$\frac{d\,(meshsize)}{d\,(position)} \tag{1.1}$$

In chapter 2, will be looking at testing the accuracy of the linear and the quadratic approximations when applied to equally and non equally spaced grid.

In chapter 3, we will set a blend of both the linear and the quadratic schemes, and test the blend scheme by looking at the maximum errors in calculating the gradients for solutions of different order of polynomials, such as quadratic and cubic.

In chapter 4, we will focus on the behavior of uniform and nonuniform hexagonal icosahedral meshes when applying the linear and quadratic schemes to the five terms occurring in Shallow Water Equations (SWEs), on the sphere, as SWEs describe many of the atmospheric phenomena in the horizontal with a special velocity generated by the geostrophic approximation it is a good test tool for mostly any proposed numerical scheme before implementing it on the more complex primitive equations. They will be described in sections 4.1 and 4.2 showing which one has the biggest impact on the global error. We will show how accurate the results are and the CPU time cost. Also we are going to apply the blend of both schemes (which is linear where the mesh is nearly uniform and quadratic elsewhere) in order to improve the accuracy over the linear scheme for minimal extra cost. To do so, we run two test cases of Williamson [1] using AtmosFOAM, which is an open source global shallow water model written using OpenFOAM. We are going to describe AtmosFOAM set up briefly in this chapter, but for more information about it and OpenFOAM please refer to [2, 4].

1.1 General Background

First we make some general comments about models from the literature. In general, the Operational Multiscale Environment Model with Grid Adaptivity (OMEGA) and its Atmospheric Dispersion Model [7], counts as the first operational atmospheric flow system based on unstructured grid techniques for atmospheric simulation and real-time hazard production. The grid resolution of the OMEGA unstructured grid can vary from 100 km to 1 km horizontally and from a few tens of meters in the boundary layer to 1 km in the free atmosphere vertically. The OMEGA unstructured grid permits additional grid elements at any point in space and time, which made it naturally scale spanning. The OMEGA can easily adapt its grid to any stationary or dynamic features in the developing weather pattern. To know more about the OMEGA model and system please refer to [7].

1.2 Background to the Dissertation

In [5], the SWEs are spatially discretized using Osher's scheme, as this scheme of upwind type is better in solving a hyperbolic system of equations. To avoid the pole problem, the Osher's scheme was applied on the latitude-longitude grid on a stereographic grid (as in stereographic coordinates the pole singularity does not exist). The Osher's scheme was applied across the sphere on a joined grid connecting the two grids at high latitude, as the latitude-longitude grid is preferable on lower latitudes, which provides a good spatial discretization for explicit integration methods. Therefore, it reduces the time step limitation incurred by the pole singularity when using only latitude-longitude grid. If the time step limitation is not significant, then the reduced latitude-longitude grid is used provided the grid is kept sufficiently fine in the polar region to resolve flow over the pole.

In the paper, PV-Based Shallow-Water Model on a Hexagonal-Icosahedral Grid [6], the author developed a new global shallow water model. The model uses a hexagonal-icosahedral grid, potential vorticity as a prognostic variable, and a conservative, shape-preserving scheme for the advection of mass, potential vorticity, and tracers. The time scheme used is semi-implicit, where the maximum time step is limited by the advection speed instead of gravity wave phase speed, which gives a stable scheme. The author has used combination of above numerical methods to avoid problems of the traditional numerical methods, such as pole problems, spurious oscillations and negatives in advected quantities.

To illustrate the performance of the model, he presents a set of results from a standard set of test cases. These show that overall accuracy of the model is comparable to the other grid point models except in a pure advection test case where the model's advection scheme shows little diffusivity. Also, in these test cases two sources of errors were noticed, the dissipation existing in the advection scheme and the grid structure led to easily seen symmetry errors, as the hexagonal grid boxes are not perfectly regular in certain regions of the grid. Which in turn, led to larger truncation errors in the edvection scheme in those regions.

In the paper [2], the authors introduce solutions of the SWEs for the case-5 of [1], which is westerly flow over mid-latitude mountain from a finite-volume model written using OpenFOAM, which has an advantage of handling any mesh structure. The authors applied a second-/third-order scheme to three different meshes of the globe, a hexagonal-icosahedral mesh and two of the reduced latitude-longitude meshes. To improve the accuracy, they introduced a scheme to fit a 2D cubic polynomial approximately on the stencil around each cell. The results they got are as accurate as the reference solution calculated using equivalent resolution spectral model.

In the paper [4], the authors present AtmosFOAM, which is shallow water equation solver written using OpenFOAM. OpenFOAM technique based on an implicit finite-volume on three-dimensional polyhedral meshes. The authors describe AtmosFOAM as ”second order on all meshes, free of spurious computational modes and conserves mass and divergence exactly and momentum, energy, potential enstrophy and potential vorticity accurately”. The authors claims that this combination of numerical methods has not been achieved before on unstructured or block-structured meshes. Which was achieved by blending the C-grid and the A-grid. The authors also present results of different test cases on different type of meshes. There results shows that the hexagonal and locally refined cubed sphere meshes give the most accurate results for the computational cost, followed by the cubed sphere. Also they a new test case to excite grid-scale oscillations.

The tests used in this dissertation will be taken from Williamson et al [1]. Here the authors have proposed a seven test cases for numerical methods that solve the shallow water equations in spherical geometry. These tests were made to evaluate and identify the potential trade-offs of the proposed numerical methods before they been applied to a full baroclinic atmospheric model and they were presented in order of complexity. They consist of advection across the poles, steady state geostrophically balanced flow of both global and local scales, forced nonlinear advection of an isolated low, zonal flow impinging on an isolated mountain, Rossby-Haurwitz wave and observed atmospheric states.

Chapter 2

Discretization on Non-uniform Grids

The objective of this chapter to test the accuracy of the linear and the quadratic approximations when applied to equal and non equal spaced grid.

The two-dimensional shallow-water equations on a rotating plane with constant Coriolis and no diffusion consist of the momentums and continuity equations:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} - fv = -g\frac{\partial h}{\partial x} \tag{2.1}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + fu = -g\frac{\partial h}{\partial y} \tag{2.2}$$

$$\frac{\partial h}{\partial t} + h\frac{\partial u}{\partial x} + h\frac{\partial v}{\partial y} = 0 \tag{2.3}$$

where u and v are the velocity in x and y directions respectively, h is the height of the fluid surface above the solid surface, f is the Coriolis force and g is the scalar acceleration due to gravity. These equations may also be written in vector form:

$$\frac{\partial \mathbf{V}}{\partial t} + f\mathbf{k} \times \mathbf{V} + \nabla h = 0 \tag{2.4}$$

where $\mathbf{V}$ is the velocities (u, v) and $\mathbf{k}$ is the unit normal vector, If we do some scale analysis to see which are the most important terms to help us in our test case. The scale values are for $V = 10ms^{-1}$, $L = 10^2m$, $D = 10^2m$, $f \approx 10^{-4}s^{-1}$ and $g = 10ms^{-2}$, where $\mathbf{V}$ the velocity scales, D and L are the characteristic scales of the motion in the vertical (depth) and horizontal (length) respectively. The magnitudes of the three terms are:

$$\frac{\partial \mathbf{V}}{\partial t} \approx \frac{V^2}{L}; \qquad f\mathbf{k} \times \mathbf{V} \approx 2\Omega V; \qquad \nabla h \approx \frac{gD}{L}.$$

$$1 \qquad\qquad 10 \qquad\qquad 10 \quad ms^{-2}$$

where vector Ω describe the rotation of a system. The acceleration is an order of magnitude smaller than the remaining terms. The Coriolis term and the pressure gradient term are of the same order of magnitude, which is called Geostrophic Balance. In this test case, for simplicity, we are going to ignore the smaller term and concentrate on the velocity component in the x-direction:

$$u = -\frac{g}{f}\frac{\partial h}{\partial x}. \tag{2.5}$$

We will therefore, consider various discretization to estimate $\frac{dh}{dx}$ at u points.

2.1 Non-uniform Grid

We consider the spatially staggered grid approach, which stores the numerically approximated values of the velocity component in the x-direction, u and the height h over a computational domain at different grid points in space on the x-axis. The heights (h) are stored at the cell centers and the component of velocities (u) at x-faces of the cells in space on the x-axis as in figure-2.1.

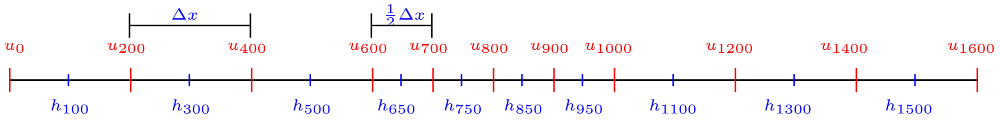


Figure 2.1: Showing the test setup domain of the staggered grid with a region of non-uniform grid intervals.

The grid at figure-2.1 has three regions:

1. Region $x < 600$ with even Δx.
2. Region $600 < x < 1000$ even half spacing $\frac{1}{2}\Delta x$.
3. Region $x > 1000$ with even Δx again.

In order to solve the momentum equations, we need to estimate $\frac{dh}{dx}$ at u points. Then we can compare the exact solution for $\frac{dh}{dx}$ with various discretization, such as linear and quadratic approximations. We start with the simplest quadratic equation of the height (2.6):

$$h = h_0(1 - \frac{x}{3000})^2 \tag{2.6}$$

where $h_0 = 10,000m$, and x is the horizontal distance. The exact solution for $\frac{dh}{dx}$ is:

$$\frac{dh}{dx} = -\frac{h_0}{1500}(1 - \frac{x}{3000}) \tag{2.7}$$

and also we going to test with the slightly more complex cubic equation of the height (2.8):

$$h = h_0(1 - \frac{x}{3000})^3 \tag{2.8}$$

The exact solution for $\frac{dh}{dx}$ is:

$$\frac{dh}{dx} = -\frac{h_0}{1000}(1 - \frac{x}{3000})^2 \tag{2.9}$$

2.2 Numerical Method

2.2.1 Linear Approximation

We are going to use the Arakawa's C-grid scheme to discretized the gradient equation using linear differencing as in figure-2.2 and equation (2.10).

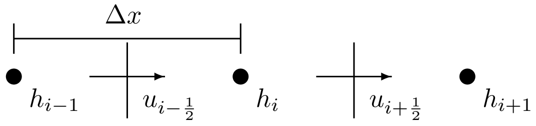


Figure 2.2: Illustrate how the discretization is done with Arakawa C-grid.

$$\left.\frac{\partial h}{\partial x}\right|_{i-\frac{1}{2}} \approx \frac{h_i - h_{i-1}}{\Delta x} \tag{2.10}$$

We will compare the exact gradient calculated from equations (2.7) and (2.9) with the discretized in the equation (2.10) with h values specified by the equations (2.7) and (2.9) and compare the errors.

2.2.2 Quadratic Approximation

We approximate height (h) by using the quadratic equation:

$$h = a + bx + cx^2 \tag{2.11}$$

where a, b and c are unknown and can be found from three values of h at the h points surrounding a u point. From a, b and c we can calculate the gradient at the u point. For example, to estimate the height gradient at $x = 600$, we use the known values of h at

$x = 500$, 650 and 750:

$$h_{500} = a + b500 + c(500)^2 \tag{2.12}$$

$$h_{650} = a + b650 + c(650)^2 \tag{2.13}$$

$$h_{750} = a + b750 + c(750)^2 \tag{2.14}$$

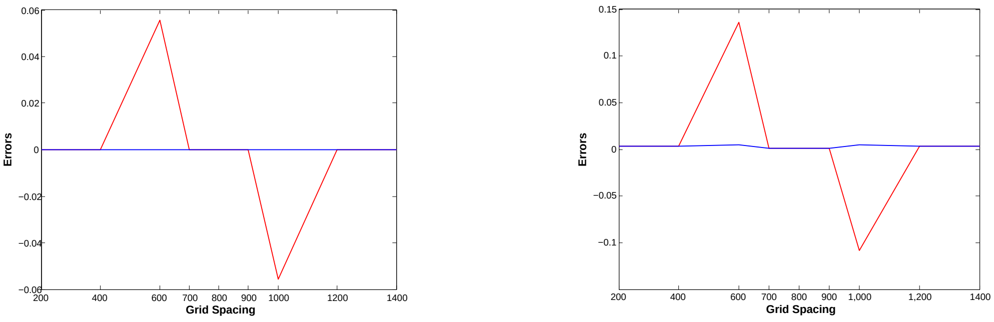
The left hand side (h_{500}, h_{650} and h_{750}) is calculated from equation (2.6) or equation (2.8). We find a, b and c and we find the approximation for the gradient by differentiating equation (2.11) as below:

$$\frac{dh}{dx} = b + 2cx \tag{2.15}$$

Then again we compare the exact gradient calculated from equations (2.7) and (2.9) with the from discretized in the equation (2.15) with h values specified by the equations (2.7) and (2.9) and compare the errors.

2.2.3 Results

In figure-2.3a we see that using the height equation (2.6) the errors using linear approximation on uniform grids are zero, except at the two points where the grid spacing are changing where it gives errors. Using the quadratic approximation gave also zero errors on the uniform grids and almost zero at the points where the grid spacing changes. In figure-2.3b using the height equation (2.8) the errors using linear approximation on uniform grids are almost zero, except at the two points where the grid spacing are changing it gives higher errors. By using the quadratic approximation gave also almost zero errors on the uniform grids and better but not zero at the points where the grid spacing changes.



(a) For the quadratic height equation.

(b) For the cubic height equation.

Figure 2.3: Plots showing the errors of using linear and quadratic approximation from calculating the $\frac{dh}{dx}$ at u points.

2.3 Least Squares Approach

Least squares approximation, which is often used when we have more equations then the unknowns. This approximation approach occurs when a function is given explicitly and we want to find a simpler type of function, such as a polynomial, which can be used to determine the approximation values of the given function. Suppose that we have a function $h \in C[a,b]$ and the polynomial $P_n(x)$ of degree required n that will minimize the error as in equation (2.16).

$$Error^2 = \int_a^b \left[h(x) - (P_n(x))\right]^2 dx \tag{2.16}$$

Now, back to our test case, the least squares approximation for the two height equations (2.6 and 2.8) is of order n=2:

$$Error^2 = \int_a^b \left[(h(x)) - \left(a + bx + cx^2\right)\right]^2 dx \tag{2.17}$$

For example to find $\frac{dh}{dx}$ at $x = 700m$, we integrate the equations (2.17) using the values of h at x = 500, 650, 750 and 850m, we fit a quadratic minimizing the error in a least squares approach as in figure-2.4.

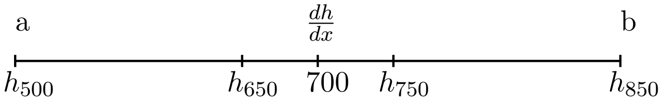


Figure 2.4: Illustration of the grid points intervals which been tested using Least Square method.

Then the minimum error is at the point where the partial derivatives of the error function with respect to the coefficients are all zero. For equation (2.6), the resulting equation from evaluating the partial derivative with respect to a, $\frac{\partial}{\partial a}$ is:

$$-2\left(\int_{500}^{850}\left(1000\left(1-\frac{x}{3000}\right)^2\right)dx-\int_{500}^{850}\left(a+bx+cx^2\right)dx\right) \tag{2.18}$$

The resulting equation from evaluating the partial derivative with respect to b, $\frac{\partial}{\partial b}$ is:

$$-2\left(\int_{500}^{850}x\left(1000\left(1-\frac{x}{3000}\right)^2\right)dx-\int_{500}^{850}\left(x\left(a+bx+cx^2\right)\right)dx\right) \tag{2.19}$$

The resulting equation from evaluating the partial derivative with respect to c, $\frac{\partial}{\partial c}$ is:

$$-2\left(\int_{500}^{850}x^2\left(1000\left(1-\frac{x}{3000}\right)^2\right)dx-\int_{500}^{850}\left(x^2\left(a+bx+cx^2\right)\right)dx\right) \tag{2.20}$$

Solving these three equations (2.18), (2.19) and (2.20) we get the values of $a = 10000$, $b = -\frac{20}{3}$ and $c = \frac{1}{900}$. Therefor, the function h for the height (2.6) is:

$$h(x) = 10000 - \frac{20}{3}x + \frac{1}{900}x^2 \tag{2.21}$$

Differentiating (2.21) we get the approximation for $\frac{dh}{dx}$:

$$\frac{dh}{dx} = -\frac{20}{3} + \frac{2}{900}x \tag{2.22}$$

We do similar steps for the height equation (2.8) and we get the values of $a = \frac{158251}{16}$, $b = -\frac{17101}{1800}$ and $c = \frac{31}{12000}$. Therefor, the the approximation $\frac{dh}{dx}$ for the height (2.8) is:

$$\frac{dh}{dx} = -\frac{17101}{1800}x + c\frac{31}{12000}x^2 \tag{2.23}$$

Solving the equation (2.6) using least squares approximation with same polynomial of degree reduce the errors to zero. While, solving the equation (2.8) using least squares approximation with smaller polynomial of degree increases the errors.

2.4 Error Analysis

Checking for the order of accuracy for the uniform and non-uniform staggered grids, if we have three grid points with different h spacing h_1 and h_2 and a function $f(x)$ is known at points x_0 and x_2 and we want to find $\frac{\partial f}{\partial x}$ at x_1 as in figure-2.5. Then to approximate the derivative of the function $f(x_1)$ we use Taylor expansion over points $f(x_0)$ and $f(x_2)$:

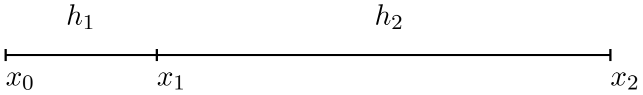


Figure 2.5: Illustration to calculate the accuracy of the non-uniform staggered grids.

$$f(x_0) = f(x_1) - h_1 f'(x_1) + \frac{h_1^2}{2} f''(x_1) + O(h^3) \tag{2.24}$$

$$f(x_2) = f(x_1) + h_2 f'(x_1) + \frac{h_2^2}{2} f''(x_1) + O(h^3) \tag{2.25}$$

If the distance were equal ($h_1 = h_2 = h$) then we get 2^{nd} order accuracy:

$$f(x_2) - f(x_0) = 2hf'(x_1) + O(h^3) \tag{2.26}$$

$$f'(x_1) = \frac{dh}{dx} = \frac{f(x_2) - f(x_1)}{2h} + O(h^2) \tag{2.27}$$

If the distance were not equal ($h_1 \neq h_2$) then we get 1^{st} order accuracy:

$$f(x_2) - f(x_0) = (h_2 + h_1)f'(x_1) + (\frac{h_2^2 - h_1^2}{2})f''(x_1) + O(h^3) \tag{2.28}$$

$$f'(x_1) = \frac{dh}{dx} = \frac{f(x_2) - f(x_0)}{h_2 + h_1} + O(h) \tag{2.29}$$

Using non-uniform grids we lose the accuracy of the approximation at the points of interface at which spacing changes form 2^{nd} to 1^{st} order of accuracy when we are calculating the gradients.

2.5 Conclusion

Linear scheme are better for uniform grids and when it comes to non-uniform the scheme gives errors at the points of interface at which spacing changes for any simple quadratic equation. Using the quadratic scheme resolve this problem but not completely. Linear scheme solving cubic equation gave higher errors at at the points of interface at which spacing changes and with quadratic scheme it reduces the errors but not to that extend. Applying finite difference approximations using non-uniform grids we loss the accuracy of the approximation at the points of interface at which spacing changes form 2^{nd} to 1^{st} order of accuracy.

In general, what ever order polynomial we use to discretized the gradients, on a uniform grid the order of accuracy is one more than the order of the polynomial and on non-uniform grids the order of accuracy is equal to the order of the polynomial.

Chapter 3

The Blending Scheme

As we going to see later, the quadratic scheme is more accurate than the linear scheme but also more costly with regard to CPU time. Getting a scheme that is going to be more accurate and almost as cheap as linear it is worth looking at a blending of both the linear and the quadratic schemes.

3.0.1 Setting the Blend Scheme

The quadratic scheme will be used only where the meshes are changing and the linear scheme elsewhere to reduce the errors that the linear scheme gives and the computational cost of using the quadratic scheme alone. The blending scheme is set by a blending factor between the linear and the quadratic schemes. The blend is linear when the factor is equal to one and quadratic when it is equal to zero and uses a combination of both schemes when it is in between, as in figure 3.1, which were linearized using equation 3.1.

$$linearBlend[i] = 10 * w[i] - 4.4 \tag{3.1}$$

where w is the linear weight for each face, calculated straightforward by a linear differencing equation 3.2.

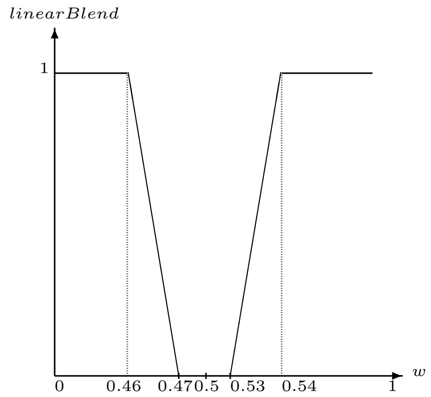
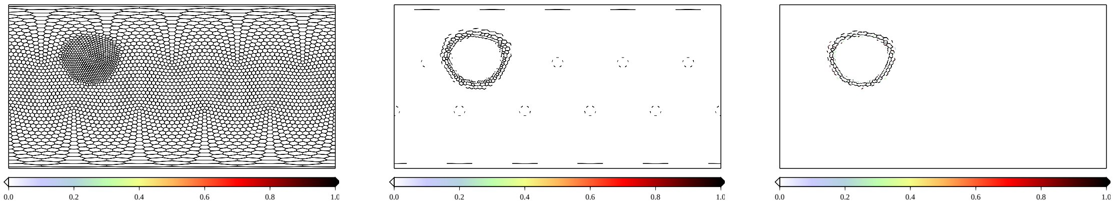


Figure 3.1: graph illustrates the way the Blend factor were calculated, where w is the linear weight for each face.

$$w = \frac{\mathbf{x}_P - \mathbf{x}_f}{|\mathbf{x}_P - \mathbf{x}_N|} \tag{3.2}$$

where x_P, x_N are the centers of two neighbour cells and x_f is the face center separating the two cells.



(a) Using only quadratic scheme.

(b) Using linear and quadratic schemes.

(c) Using quadratic where the meshes are changing and linear elsewhere.

Figure 3.2: Graphs showing the output of various values of w.

figure (3.2), shows using different values of the w, as when the equation (3.1) equal to zero we will be only using the quadratic scheme, figure (3.2a). When using the more of the linear than the quadratic schemes (3.2b) and finally when using the quadratic scheme only where the meshes are changing and the linear scheme elsewhere (3.2c).

The blended scheme is not currently implemented efficiently since both the linear and the quadratic schemes differencing are calculated and then blended. If this blended schemes appears to be beneficial it can be implemented so that it is almost as efficient as linear (summing when the blending coefficient is mostly zero). We will therefore not discuss the CPU time of using the blended scheme, only the accuracy.

3.1 Testing The Blend Scheme

In order to test the blend scheme we look at the maximum errors in calculating the gradients for equations of different order of polynomials, such as quadratics and cubic. For the quadratic equation on the sphere (3.3).

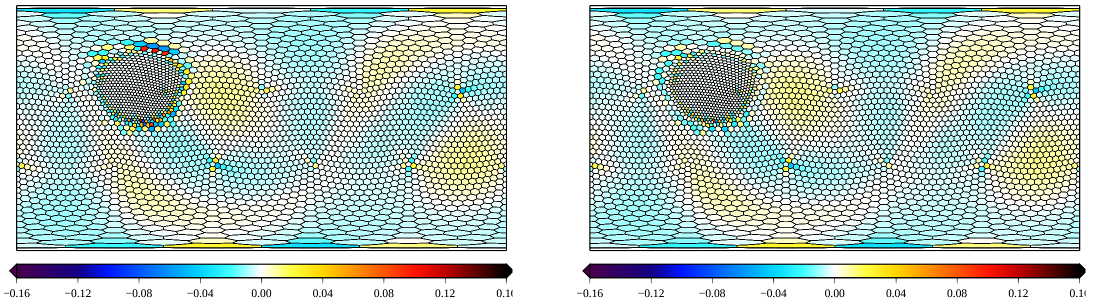
$$\begin{aligned} p &= (x-y)(x-z) \qquad (3.3) \\ &= x^2 - xz - xy + yz \end{aligned}$$

The gradient for p is:

$$\nabla p = \begin{pmatrix} 2x - z - y \\ -x + z \\ -x + y \end{pmatrix}$$

and the magnitude of the gradient:

$$|\nabla p| = \sqrt{(2x-z-y)^2 + (z-x)^2 + (y-x)^2}$$



(a) Linear, ∇ error $1.0 \times 10^{-}1$.

(b) Blend, ∇ error $8.3 \times 10^{-}2$.

Figure 3.3: Graphs showing the errors from calculating the gradient of the quadratic equation using the linear and the blend schemes.

Figure 3.3a shows the errors for calculating the gradient of the quadratic equation using the linear scheme and clearly the errors are maximum where the meshes are changing. When applying the blend scheme figure 3.3b, the errors are reduced where the meshes are changing.

For the cubic equation on the sphere equation (3.4).

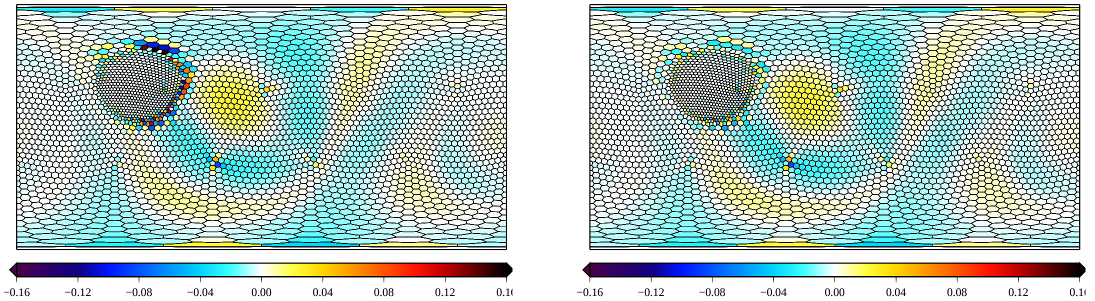
$$\begin{aligned} p &= (x-y)(x-z)(x-1) \qquad (3.4) \\ &= x^3 - x^2 - x^2z + xz - x^2y + xy + xyz - yz \end{aligned}$$

The gradient for p is:

$$\nabla p = \begin{pmatrix} 3x^2 - 2x - 2xz + z - 2xy + y + yz \\ -x^2 + x + xz - z \\ -x^2 + x + xy - y \end{pmatrix}$$

and the magnitude of the gradient:

$$|\nabla p| = \sqrt{(3x^2 - 2x - 2xz + z - 2xy + y + yz)^2 + (-x^2 + x + xz - z)^2 + (-x^2 + x + xy - y)^2}$$



(a) Linear, maximum ∇ error 1.6×10^{-1}.

(b) Blend, maximum ∇ error $9.5\times^{-2}$.

Figure 3.4: Graphs showing the errors from calculating the gradient of the cubic equations using the linear and the blend schemes.

Figure 3.4a shows the errors for calculating the gradient of the cubic equation using the linear scheme and clearly it the errors are maximum where the meshes are changing. When applying the blend scheme figure 3.4b, the errors are reduced where the meshes are changing.

Chapter 4

Shallow-Water Equations Test Cases

In this chapter we focus on two main objectives. Firstly, the behaviour of uniform and nonuniform hexagonal icosahedral mesh when applying the linear and quadratic schemes to SWEs on the sphere and how accurate the results are and the CPU time cost. Secondly, applying a blend of both schemes which is linear where the mesh is nearly uniform and quadratic elsewhere in order to improve accuracy over linear for minimal extra cost.

To do so, we run two test cases of [1] using AtmosFOAM, which is an open source global shallow water model written using OpenFOAM. Both OpenFOAM and AtmosFOAM are described in [2, 4]. As the Shallow Water Equations describe many of the atmosphere phenomena in the horizontal dynamical aspects which makes it a good test tool for mostly any proposed numerical scheme before implementing it on the more complex primitive equations.

4.1 AtmosFOAM

In this section we briefly describe AtmosFOAM which is described more fully in [2, 4]. AtmosFOAM solves the SWE's on spherical meshes in Cartesian co-ordinates using the finite-volume method with linear or quadratic differencing schemes.

The two-dimensional SWE's consist of the momentum and continuity equations:

$$\frac{\partial h\mathbf{U}}{\partial t} + \nabla \bullet (h\mathbf{U}\mathbf{U}) = -2\boldsymbol{\Omega} \times h\mathbf{U} - gh\nabla(h + h_0) \tag{4.1}$$

$$\frac{\partial h}{\partial t} + \nabla \bullet (h\mathbf{U}) = 0 \tag{4.2}$$

where $\mathbf{U}$ is the horizontal vector velocity, the ∇ operator is the horizontal gradient operator, the $\nabla\bullet$ operator is the horizontal divergence operator, h is the fluid depth, h_0 is the height of the solid surface above a reference height, Ω is the angular vector velocity of the earth and g is the scalar gravitational constant.

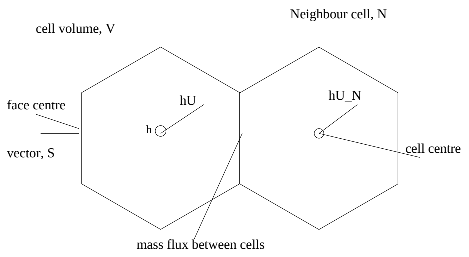


Figure 4.1: Components of the cells in AtmosFOAM.

The prognostic variables are cell-average momentum ($h\mathbf{U}$), height (h) and the mass flux between the cells (ϕ) as in figure 4.1. This system is over specified and the method of removing the possible inconsistency between ($h\mathbf{U}$) and (ϕ) is described in [4].

The finite-volume discretization to calculate divergences, cell gradients and the Laplacian were done using Gauss theorem, as from [4] for the cell centered scalar and vector quantities Ψ and $\boldsymbol{\Psi}$:

$$\nabla \bullet \boldsymbol{\Psi} \approx \frac{1}{V}\int_V \nabla \bullet \boldsymbol{\Psi} dV \approx \frac{1}{V}\sum \boldsymbol{\Psi}_{\mathbf{f}} \bullet S, \tag{4.3}$$

$$\nabla_c \Psi \approx \frac{1}{V}\int_V \nabla \Psi dV \approx \frac{1}{V}\sum \Psi_f S, \tag{4.4}$$

$$\nabla^2 \Psi \approx \frac{1}{V}\int_V \nabla^2 \Psi dV \approx \frac{1}{V}\sum \nabla_f \Psi \left|S\right|, \tag{4.5}$$

where V is the cell volume, $\sum$ the sum over all the faces of a cell, subscript f means interpolation from cell averages to face averages, ∇_c the cell-average gradient and ∇_f is the gradient in direction S, discretized at the face.

Interpolations from cell average quantities to face average quantities (Ψ to Ψ_f) are done by assuming that average quantities are represented by cell centre or face centre quantities and then using linear or bi-quadratic differencing. The simplest numerical solution is given by the linear differencing where we calculate the face average by:

$$\boldsymbol{\Psi}_{\mathbf{f}} = \lambda \boldsymbol{\Psi}_{\mathbf{P}} + (1-\lambda)\boldsymbol{\Psi}_{\mathbf{N}},$$

and the gradient at face average by:

$$\nabla_f \boldsymbol{\Psi} = \frac{\boldsymbol{\Psi}_{\mathbf{N}} - \boldsymbol{\Psi}_{\mathbf{P}}}{|\mathbf{x}_{\mathbf{N}} - \mathbf{x}_{\mathbf{P}}|}$$

where $\boldsymbol{\Psi}$ is the cell average at the face f between the neighboring cells P and N with corresponding cells centres $\mathbf{x}_{\mathbf{P}}$ and $\mathbf{x}_{\mathbf{N}}$ with $\lambda = \frac{|\mathbf{x}_{\mathbf{N}} - \mathbf{x}_{\mathbf{f}}|}{|\mathbf{x}_{\mathbf{N}} - \mathbf{x}_{\mathbf{P}}|}$. which is only second-order accurate on uniform regular mesh. For more accuracy, we use the quadratic differencing that uses the bi-quadratic polynomial:

$$\psi = a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 xy + a_5 y^2. \tag{4.6}$$

where, x and y are local co-ordinates of the system at each face center, in the plane of the two-dimensional geometry with the x direction normal to the face. Then the interpolation at the face will be given by:

$$\mathbf{\Psi_f} = a_0,$$

and the face-center gradient is given by:

$$\nabla_f = a_1.$$

The numerical solution obtained when using a bi-quadratic interpolation is more accurate than the solution from the linear differencing on the uniform regular mesh. Moreover, it is also second order where the mesh is non-uniform.

4.2 Main Setup

We run two test cases of Williamson [1] on two different meshes, one nearly uniform hexagonal icosahedral mesh and one mesh with local refinement using polygon. We apply different schemes to the SWEs:

1. Linear scheme (l).
2. Quadratic scheme (q).
3. A blend of both schemes (b)

for these five numerical approximations terms respectively:
The gradient, the divergence, the Laplacian, the Interpolation method and the height gradient. In order to evaluate those schemes used and to see which of the above terms have the biggest impact on the global errors, we compare the:-

1. The Time taken to run in seconds (CPU time).

2. The Normalized global errors of the height($l(h)$).

The $l(h)$ error norms of the height are defined as:

$$l_1(h) = \frac{\sum_{cells} V(h - h_{true})}{\sum_{cells} V h_{true}} \tag{4.7}$$

$$l_2(h) = \sqrt{\frac{\sum_{cells} V(h - h_{true})^2}{\sum_{cells} V h_{true}^2}} \tag{4.8}$$

$$l_\infty(h) = \frac{max_{allVh_{true}} \sum_{cells} |V(h - h_{true})|}{max_{allVh_{true}} \sum_{cells} |V h_{true}|} \tag{4.9}$$

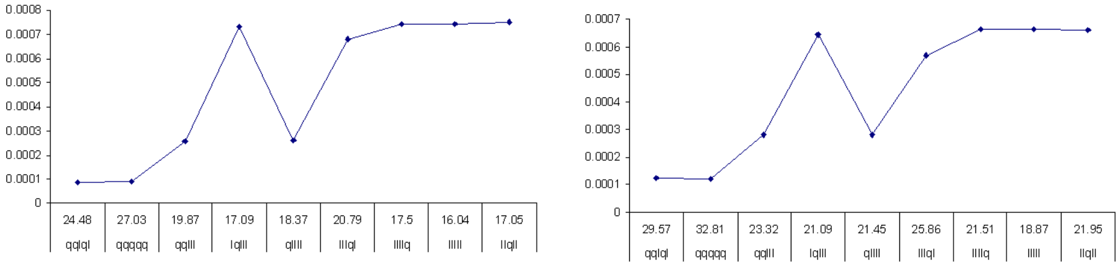
where h_{true} is the exact , reference solution.

4.3 Williamson *etal* test Case 2

Testing case 2 of the Williamson et al [1] global steady state nonlinear zonal geostrophic flow. This case were done to observe the behavior of steady zonal flow with corresponding geostrophic height field on the uniform and non-uniform meshes when applied different scheme to it. The results after 5 days using time steps 20 minutes giving maximum C.F.L number of 0.05 for the uniform and of 0.11 for the nonuniform were compared with the initial conditions.

4.3.1 Results

In order to evaluate which of the five terms numerical approximation that were described in sections 4.1 and 4.2 have the biggest impact on the global error, we plot the l_2 error norm as a function of the scheme used in those five terms respectively for both uniform and nonuniform hexagonal icosahedral mesh in the figure 4.2. This shows that the gradient is the most important term and as equally important as using quadratic on the divergence as



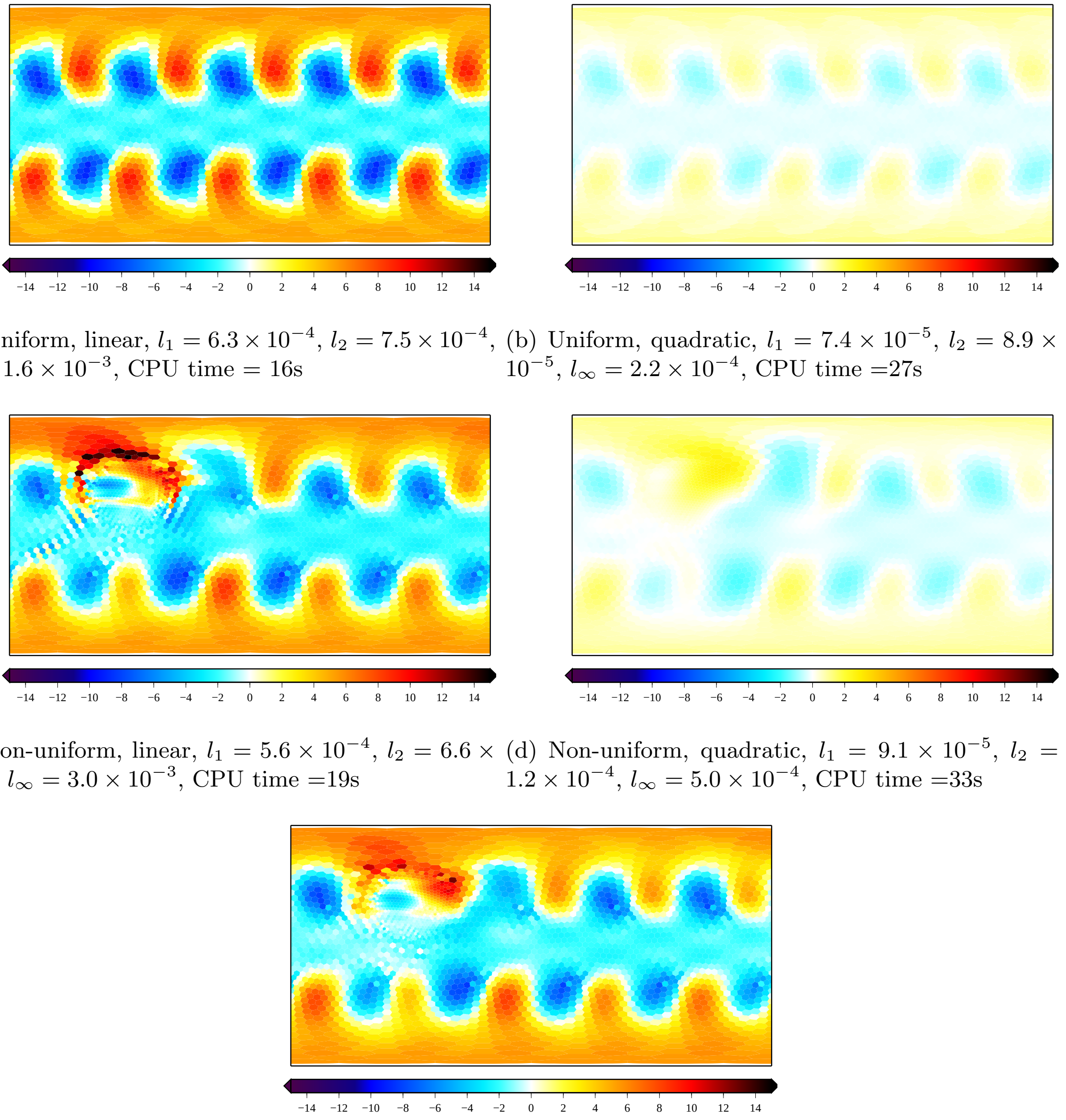
(a) Uniform Meshes

(b) Nonuniform Meshes

Figure 4.2: Height norm errors $l_2(h)$ against CPU time taken after 5 days after applying different combination of linear and quadratic scheme to the five terms gradient, divergence, Laplacian, Interpolation method and height gradient. Errors calculated as differences with respect to the initial condition.

well. The combination of those two and the interpolation method gives the smallest errors and is as good as using quadratic scheme for the all five terms with slightly better CPU time. We going to test the blend scheme on these three terms and for the rest of the terms we will use linear scheme.

The errors in figure 4.3 are calculated after 5 days in comparison to the initial conditions. In general, we get large oscillations (symmetry errors), and that appear due to the hexagonal grid are not completely uniform, they are nearly uniform as described in [6]. The errors using just linear scheme for the five terms are much larger than just using the quadratic scheme for both uniform and non-uniform meshes (figure 4.3 from a-d) as expected.



(a) Uniform, linear, $l_1 = 6.3 \times 10^{-4}$, $l_2 = 7.5 \times 10^{-4}$, $l_\infty = 1.6 \times 10^{-3}$, CPU time = 16s

(b) Uniform, quadratic, $l_1 = 7.4 \times 10^{-5}$, $l_2 = 8.9 \times 10^{-5}$, $l_\infty = 2.2 \times 10^{-4}$, CPU time =27s

(c) Non-uniform, linear, $l_1 = 5.6 \times 10^{-4}$, $l_2 = 6.6 \times 10^{-4}$, $l_\infty = 3.0 \times 10^{-3}$, CPU time =19s

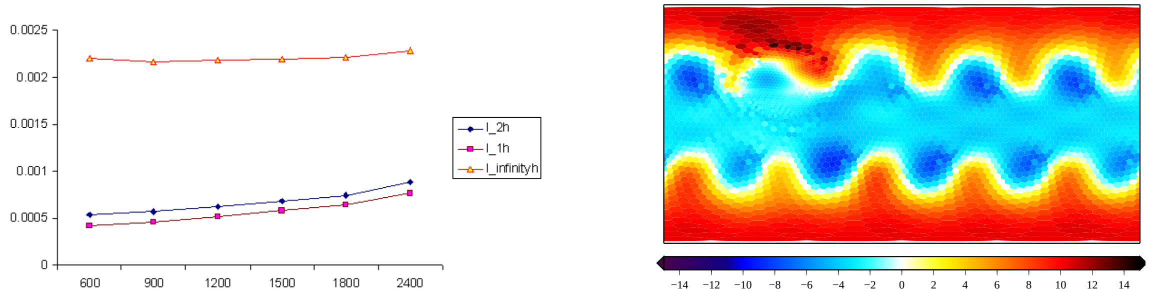
(d) Non-uniform, quadratic, $l_1 = 9.1 \times 10^{-5}$, $l_2 = 1.2 \times 10^{-4}$, $l_\infty = 5.0 \times 10^{-4}$, CPU time =33s

(e) Non-uniform, Blend, $l_1 = 5.2 \times 10^{-4}$, $l_2 = 6.2 \times 10^{-4}$, $l_\infty = 2.2 \times 10^{-3}$, CPU time =30 , using equation 3.1

Figure 4.3: Errors for the Uniform and Non-uniform hexagonal icosahedral mesh after 5 days. Errors calculated as differences with respect to the initial conditions. The errors rang from -15 - 15

Most of the errors around the refine area are at the region where the nearly uniform hexagonal mesh. The errors when using the blend scheme for the non-uniform mesh figure (4.3e) gives slightly smaller errors at the refine mesh than using only the linear scheme figure (4.3c), and the oscillations also small.

When using different time steps we get different errors norms, as figure (4.4a) shows, as we reduce the time steps the errors norms get smaller. But the CPU time and the oscillations get bigger and the opposite is true. More oscillations were expected as we use smaller time steps which resolves the gravity wave much better. But at time 600s the l_∞ got larger due to increase in oscillations. As an example, figure (4.4b) shows the non-uniform mesh using the blend scheme at time step of 40 minutes and the C.F.L number 0.22, where the errors increased and the oscillations decreased compared with figure (4.3e).



(a) Using the blend scheme

(b) Non-uniform, Blend, $l_1 = 7.7 \times 10^{-4}$, $l_2 = 8.9 \times 10^{-4}$, $l_\infty = 2.3 \times 10^{-3}$, CPU time=16s, time step 40 minutes.

Figure 4.4: (a) Time steps in seconds against height error norms. (b) Errors for the Non-uniform hexagonal icosahedral mesh after 5 days. Errors calculated as differences with respect to initial conditions. The errors rang from -15-15.

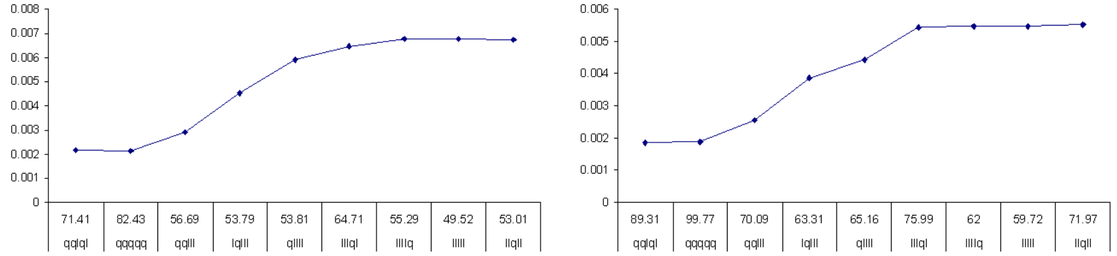
4.4 Williamson *etal* test Case 5

Testing Case 5 of the Williamson et al [1] Zonal flow over an isolated mountain. This case were done to observe the behavior of steady zonal flow impinging on a mountain on the uniform and non-uniform meshes when applied different scheme to it. The results after 15 days using time steps 20 minutes giving maximum C.F.L number of 0.05 for the uniform and of 0.11 for the nonuniform were compared with the given reference solution.

The reference solution was calculated using the Spectral transform (STSWM [9]) using T213 resolution. The version of STSWM has been revised by Pilar Ripodas from Deutscher Wetterdienst [$http://www.icon.enes.org/swm/stswm$]. The STSWM solution is interpolated onto the AtmosFOAM grid using bicubic interpolation, also from [$http://www.icon.enes.org/swm/stswm$].

4.4.1 Results

Again we evaluate which of five terms numerical approximation that were described in sections 4.1 and 4.2 have the biggest impact on the global error by plotting the l_2 error norm as a function of the scheme used in those five terms respectively for both uniform and nonuniform hexagonal icosahedral mesh in the figure 4.5. Looking at this two plots we find that the divergence is the most important, than the gradient. While, in figure 4.3 the gradient were the opposite and that due to the flow as it hits the mountain it create



(a) Uniform Meshes

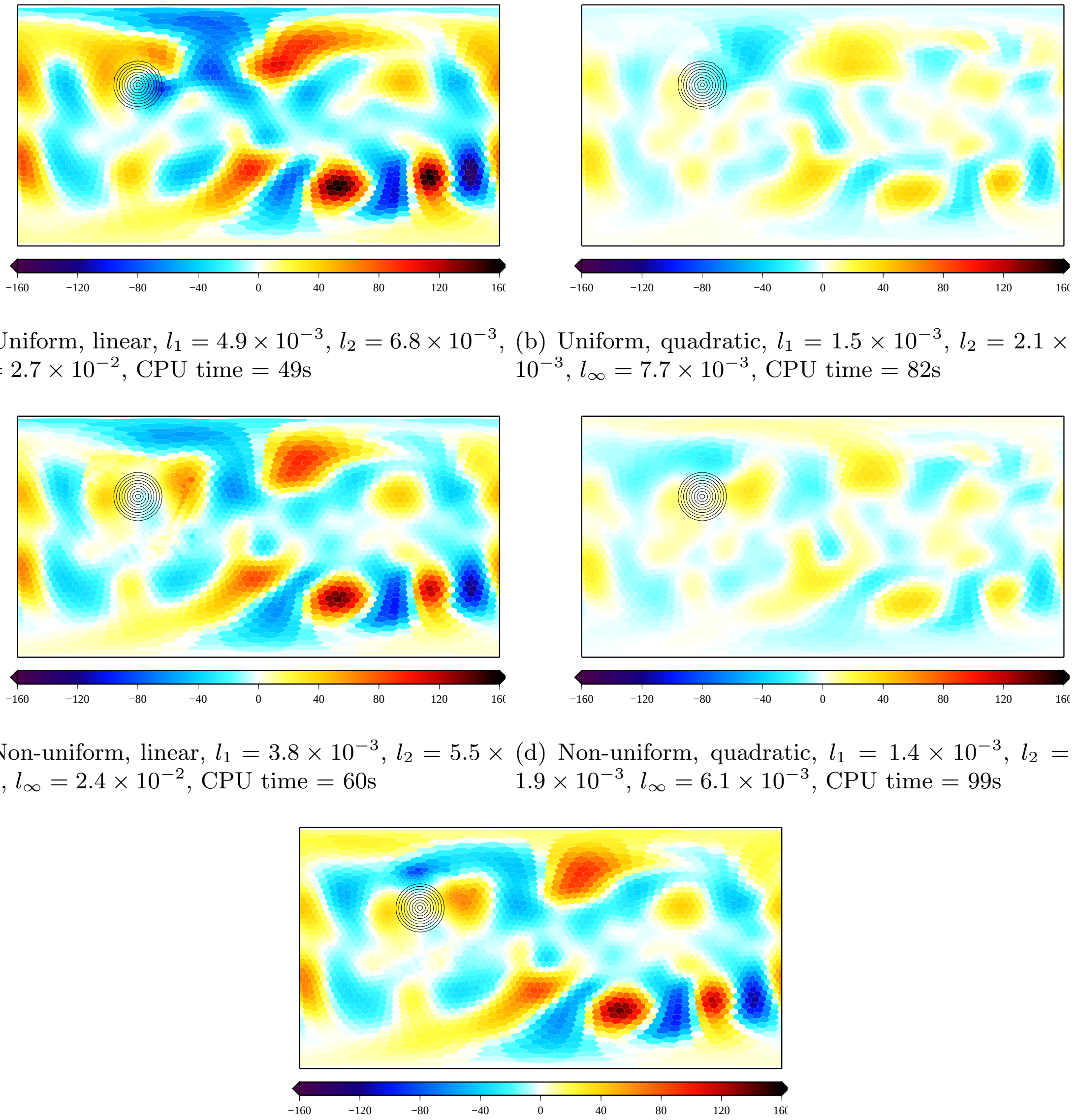
(b) Non-uniform Meshes

Figure 4.5: Height norm errors $l_2(h)$ against CPU time taken after 15 days after applying a combination of linear and quadratic scheme to the five terms gradient, divergence, Laplacian, Interpolation method and height gradient. Errors calculated as differences with respect to the given reference solutions.

divergence. Also as before the combination of the gradient and the divergence with the quadratic scheme gives small errors. Also as before using combination of those two and the interpolation method gave us the smallest errors and is as good as using the quadratic scheme on all the five terms with slightly better CPU time. Again we are going to test the blend scheme on these three terms and for the rest of the terms we will use the linear scheme.

The errors in figure 4.6 are calculated after 15 days in comparison to the reference solutions which were given. Due to large errors in this case the errors in the previous case are there but too small to be visualized. The errors using just the linear scheme for the five terms are much larger than just using the quadratic scheme for both uniform and non-uniform meshes figure 4.6 from a - d.

The errors when using the blend scheme for the three terms on a non-uniform mesh figure 4.6e gives smaller errors at the refine mesh than just using the linear scheme figure 4.6c.



(a) Uniform, linear, $l_1 = 4.9 \times 10^{-3}$, $l_2 = 6.8 \times 10^{-3}$, $l_\infty = 2.7 \times 10^{-2}$, CPU time = 49s

(b) Uniform, quadratic, $l_1 = 1.5 \times 10^{-3}$, $l_2 = 2.1 \times 10^{-3}$, $l_\infty = 7.7 \times 10^{-3}$, CPU time = 82s

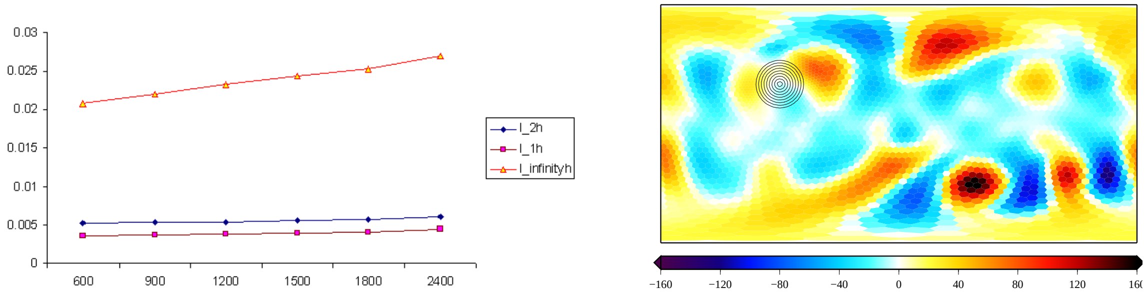
(c) Non-uniform, linear, $l_1 = 3.8 \times 10^{-3}$, $l_2 = 5.5 \times 10^{-3}$, $l_\infty = 2.4 \times 10^{-2}$, CPU time = 60s

(d) Non-uniform, quadratic, $l_1 = 1.4 \times 10^{-3}$, $l_2 = 1.9 \times 10^{-3}$, $l_\infty = 6.1 \times 10^{-3}$, CPU time = 99s

(e) Non-uniform, blend, $l_1 = 3.8 \times 10^{-3}$, $l_2 = 5.3 \times 10^{-3}$, $l_\infty = 2.3 \times 10^{-2}$, CPU time =65s

Figure 4.6: Errors for the Uniform and Non-uniform hexagonal icosahedral mesh after 15 days. Errors calculated as differences with respect to the given reference solutions. The errors rang from -160 - 160

As before with different time steps we get different errors norms. As in figure 4.7a shows as we reduce the time steps the errors norms get smaller. But the CPU time and the oscillations get larger and the opposite is true. More oscillations were expected as we use smaller time steps which resolve the gravity wave better. As an example figure 4.7b shows the non-uniform mesh with the blend scheme at time step of 40 minutes and the C.F.L number 0.22 errors increased and the oscillations decreased compared with figure 4.6e.



(a) Using blend scheme

(b) Non-uniform, Blend, $l_1 = 4.5 \times 10^{-3}$, $l_2 = 6.1 \times 10^{-3}$, $l_\infty = 2.7 \times 10^{-2}$, CPU time = 49s, time step 40 minutes

Figure 4.7: (a) Time steps in seconds against height error norms. (b) Errors for the Non-uniform hexagonal icosahedral mesh after 15 days. Errors calculated as differences with respect to the given reference solutions. The errors rang from -160 - 160

4.5 Conclusion

We have shown that the gradient and the divergence are the most important terms to approximate with. The combination of those two and the interpolation method helps to minimize the errors. We have also shown that we do get errors with the uniform grids and using the quadratic scheme gives much less errors than using the linear scheme but the CPU costs are higher. Finally we shown by applying a blend of the linear and the quadratic schemes it does improve the accuracy from just linear scheme.

Chapter 5

Conclusion

In this dissertation we have introduced a blend scheme, which calculates the gradients using quadratic polynomials where the mesh is non-uniform (the transition between coarse and fine meshes) and linear polynomials where the mesh is linear or nearly linear. We tested the blend scheme by calculating the gradient with polynomials of different order and compared these results with the ones obtained from the linear scheme. In chapter 2, we showed that for any polynomial of order n used to discretize the gradients, the accuracy of the scheme is of order $n + 1$ on a uniform grid and of order n on a non-uniform grid. Thus, the blend scheme scheme is expected to be more accurate than the linear scheme on a non-uniform grid and as accurate as linear on uniform grid. This was affirmed by the numerical results in chapter 3.

Later, we have evaluated the linear and the quadratic schemes on the full shallow water equations solver AtmosFOAM [2, 4] using two different test cases of Williamson et al. [1], test case 2, the global steady state nonlinear zonal geostrophic flow and test case 5, zonal flow over an isolated mountain on uniform and non-uniform hexagonal icosahedral meshes.

The results of these two test cases showed that:

1. The combination of numerical approximation terms of the gradient, the divergence and the interpolation method have the biggest impact on the global error.
2. Using the linear scheme resulted in larger errors at the transition between coarse and fine meshes but was less costly in CPU time.
3. Using the quadratic scheme resulted in smaller errors at the transition between coarse and fine meshes but more costly in CPU time.
4. Using the blend of the both schemes improved the accuracy from the linear scheme at the transition between coarse and fine meshes.

5.1 Future Work

More tests with different complexity need to be done on the blend scheme. As the blended scheme is not currently implemented efficiently since both the linear and the quadratic schemes differencing are calculated and then blended. It worth implementing so that it is almost as efficient as linear (summing when the blending coefficient is mostly zero), to compare the CPU cost with linear and quadratic schemes.

Bibliography

[1] Williamson D. L, Drake J. B, Hack J. J, Jakob R and Swarztrauber P. N. A Standard Test Set for Numerical Approximations to the Shallow Water Equations in Spherical Geometry, *Journal of Computational Physics*, **102**, 1992, 211–224.

[2] Hilary Weller and Henry Weller. A high-order arbitrarily unstructured finite-volume model of the global atmosphere: Tests solving the shallow-water equations, *International Journal for numerical methods in fluids*, **56**, 2008, 1589–1596.

[3] Jorn Behrens. *Adaptive Atmospheric Modeling*, Springer, 2006.

[4] Hilary Weller, Henry Weller and Aime Fournier. Towards a Polyhedral Model of the Global Atmosphere: Shallow-Water Test Cases, *Submitted to Journal of Computational Physics*, 2008.

[5] D.Lanser, J.G.Blom and J.G.Verwer. Spatial Discretization of the Shallow Water Equations in Spherical Geometry Using Osher's Scheme, *Journal of Computational Physics*, **165**, 2000, 542–565.

[6] John Thuburn. A PV-Based Shallow-Water Model on a Hexagonal-Icosahedral Grid, *Monthly Weather Review*, **125**, 1996, 2328–2347.

[7] David P. Bacon, Nash'at N. Ahmed, Zafer Boybeyi, Thomas J. Dunn, Mary S. Hall, Pius C. S. Lee, R. Ananthakrishna Sarma and Mark D. Turner. A Dynamically Adapting Weather and Dispersion Model: The Operational Multiscale

Environment Model with Grid Adaptivity (OMEGA), *Monthly Weather Review*, **128**, 1999, 2044–2076.

[8] R. Vichnevetsky and L. H. Turner. Spurious scattering from discontinuously stretching grids in computational fluid dynamics, *Applied Numerical Mathematics*, **8**, 1991, 280–299.

[9] R. J.J. Hack and R. Jakob. Description of a global shallow water model based on the spectral transform method, *Technical Report NCAR/TN-343+STR*, Center for Atmospheric Research, Boulder Colorado, February 1992.