UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Superior d'Enginyeries Industrial, Aeroespacial i Audiovisual de Terrassa

Implementation of bottom topography for Shallow Water simulations

Document:

Report

Author:

David Ferrer Argullós

Director/Co-director:

Enrique García Melendo / Arnau Miró

Degree:

Bachelor in Aerospace Engineering

Examination session:

Spring, 2023

BACHELOR FINAL THESIS

Contents

List of Figures

List of Tables

List of abbreviations / Glossary

Add a lost of abbreviations or a glossary if it is necessary.

Chapter 1

Introduction

1.1 Object

The object of this bachelor thesis is to elaborate a Matlab code to simulate different atmospheric phenomena on a planetary scale on a rotating planet with the aim of introducing a topography into the model in order to simulate how the presence of this alters the behaviour of the atmosphere on a large scale. The value of this study is that the results obtained can be a great source of information about how the presence of islands or mountainous terrain on geophysical flows changes the behavior of the atmosphere.

1.2 Scope

The project scope is the development of the code and the results of the different code versions and the validation results of the code functions

1.3 Requirements

The principal tool that will be used in the project to develop the code is Matlab. Also it is required a good knowledge in Shallow water models which can be acquired through the documentation

1.4 Rationale

The aim of this project is to study the behaviour of atmospheric phenomena on a planetary scale. Understanding these phenomena can be very useful in predicting and making weather models. The used methodology (Shallow Water model) is an efficient way to simplify the equations governing atmospheric physics, without sacrificing large-scale models with limited computational resources. The results of the methodology are good enough despite the simplification applied to it. The methodology has also been used in other studies. [1] [2]. Because the purpose of the work is that the code can simulate real phenomena, the code will be first used for a simulation with the topography of Amsterdam Island [3].

Chapter 2

Background and fundamentals

2.1 Introduction to planetary atmospheres

2.2 Shallow water model

The Shallow-Water model is essentially a subset of the Navier-Stokes equations with a series of simplifications for our purpose, planetary atmosphere simulations. The Navier–Stokes equations are useful because they describe the physics of many phenomena of scientific and engineering interest. They may be used to model the weather, ocean currents, water flow in a pipe or air flow around a wing.

2.2.1 Hypothesis

In the Shallow water model, we consider a thin layer of fluid, which thickness on the vertical scale, named D is much smaller than the horizontal scale, named L .

But for large-scale motion in the atmosphere or oceans, the movement aspect ratio is usually small indicating that the phenomenon to be studied is produced in a thin sheet of fluid where the trajectories of the fluid particles are predominantly horizontal.

2.2.2 Rotating coordinate frame

If we take in to account the momentum equations, the forces between the relative velocity and the Coriolis acceleration. The relation between this two concepts is the Rossby number

This value indicates the length scale of the flow structures evolving in the domain and time, which is highly related with the phase speed of the present gravity waves ($U = \sqrt{gD}$)

When the time scale of fluid motion exceeds the planet's rotation period, the Coriolis effect beats the relative acceleration obtaining a Rossby number less than unit (Ro $<$ 1).

This is caused by the fluid tendency to reduce its relative acceleration at large-scale while Coriolis effect remains constant.

When the time scale of fluid motion exceeds the planet's rotation period, the Coriolis effect beats relative acceleration obtaining a Rossby number less than unit (Ro < 1). This is caused by the tendency of the fluid to reduce its relative acceleration at large-scale motion, while the Coriolis effect remains constant.

2.2.3 Shallow water equations

In this section we will now do the deduction of the equations.

If we analyse the magnitude of the terms in the mass conservation equation

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \tag{2.1}$$

$$\frac{U}{L} + \frac{U}{L} + \frac{W}{D} = 0 \tag{2.2}$$

Where the W stands for the vertical velocity and $W < \delta$

Now we take a look on the momentum equations:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z} - fv = -\frac{1}{\rho}\frac{\partial p}{\partial x} \tag{2.3}$$

$$\frac{U}{T} \quad \frac{U^2}{L} \quad \frac{U^2}{L} \quad \frac{UW}{D} \quad fU = \frac{P}{\rho L} \tag{2.4}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z} + fu = -\frac{1}{\rho}\frac{\partial p}{\partial y} \tag{2.5}$$

$$\frac{U}{T} \quad \frac{U^2}{L} \quad \frac{U^2}{L} \quad \frac{UW}{D} \quad fU = \frac{P}{\rho L} \tag{2.6}$$

$$\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z} = -\frac{1}{\rho}\frac{\partial p}{\partial z} - g \tag{2.7}$$

$$\frac{W}{T} \quad \frac{UW}{L} \quad \frac{UW}{L} \quad \frac{W^2}{D} \quad = \frac{P}{\rho D} \tag{2.8}$$

It is important to note that the order of magnitude of each term is written bellow in the according characteristic scale.

Also, equations 2.3 and 2.5 state the pressure field P as a function of L,T, U, f and ρ so that the horizontal pressure gradient enters as a forcing term in the horizontal plane, otherwise the flow would not be accelerated. Therefore, from the z component of the momentum equation 2.7, the ratio of total derivative of the velocity to the pressure gradient in vertical direction is given by :

$$O(\rho\frac{dw/dt}{\partial p/\partial z}) = \rho\frac{[W/T, WU/L]_{max}}{P/D} = \delta^2\frac{[1/T, U/L]_{max}}{[1/T, U/L, f]_{max}} \tag{2.9}$$

If we study this correlation, we can conclude that the order of magnitude should be δ^2 ¡¡ 1 given typical values of the Rossby number (3.8). Consequently, we can apply the hidrostatic approximation, which integrated within the boundary conditions gives the following expression:

$$p = \rho g(h + h_b - z) + p_0 \tag{2.10}$$

where p_0 evaluates the constant uniform pressure at the surface of the fluid layer.

If the horizontal pressure gradients are independent from z,

$$\frac{\partial p}{\partial x} = \rho g\frac{\partial(h + h_b)}{\partial x} \tag{2.11}$$

$$\frac{\partial p}{\partial y} = \rho g\frac{\partial(h + h_b)}{\partial y} \tag{2.12}$$

the horizontal accelerations must be also independent. Consequently, the horizontal velocities will remain independent from the z coordinate and this also implies that the vertical velocity can be integrated from the mass conservation equation 2.1.

In order to obtain a lineal funcition in the vertical direction, boundary conditions like no normal flow at the surface $\nabla(h_b - z)\vec{v} = 0$, resulting in:

$$w = -z(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}) + u\frac{\partial h_b}{\partial x} + v\frac{\partial h_b}{\partial y} \tag{2.13}$$

In order to complete the system of equations of the model, we must add the corresponding kinematic condition $w = \frac{d(h+h_b)}{dt}$ on the fluid layer system.

Finally, the system of equations defining the model results in this set of equations:

$$\left.\begin{array}{c} \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} - fv = -g\frac{\partial \eta}{\partial x} \\ \frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + fu = -g\frac{\partial \eta}{\partial y} \\ \frac{\partial \eta}{\partial t} + \frac{\partial hu}{\partial x} + \frac{\partial hv}{\partial y} = 0 \\ h - D - \eta + h_B = 0 \end{array}\right\} \tag{2.14}$$

2.2.4 Geostrophic balance

As it has been mentioned in section 2.2.2, in large-scale motion models as the ones are intended to study, the Coriolis term is the predominant in the momentum equations. If we neglect the terms of the inertial forces from the model equations (Eq.2.14) and the pressure gradients, it is obtained the following relation between the velocity and pressure field:

$$-fv = -\frac{1}{\rho}\frac{\partial p}{\partial x} \tag{2.15}$$

$$+fu = -\frac{1}{\rho}\frac{\partial p}{\partial y} \tag{2.16}$$

where the Coriolis parameter can be written in terms of the latitude (φ) as:

$$f = 2\Omega sin(\varphi) \tag{2.17}$$

Explicar una mica més per a què és important (tema rotació huracans)

2.2.5 Vorticity

The flow has a tendency to rotate around a point, which is described by the relative vorticity. The conservation of the potential vorticity is introduced in the model as a passive tracer reminiscent to a cloud for the visualisation of the results. Its expression can be readjusted to the model variables. In the same way as the SW equations, it is expressed in a Cartesian frame of coordinates.

Since in the SW model density is an irrelevant constant, potential vorticity is written independently of density.

In the model implemented in this project, f = 2Ω but in general if the model is applied on a sphere, f is the Coriolis parameter, which corresponds to the vertical component of the planetary vorticity.

2.2.6 Coordinate system

The derivation of the model and expressions used so far are computed into a Cartesian coordinate system for simplicity. In simplified cases in the context of Shallow-Water domains, these expressions are sufficient. However, for extended simulations of planetary atmospheres or even oceans, the formulation need to be modified in order to capture all the physics.

Cartesian Coordinates

The Cartesian coordinates system is very useful to verify and test simple cases. The goal is to visualize the evolution of the system from an initial perturbation. Different parameters can be analysed like the phase speed on the gravity waves or the positive/negative waves inferences,

For cases studying large atmospheres domains, the model equations are not accurate enough.

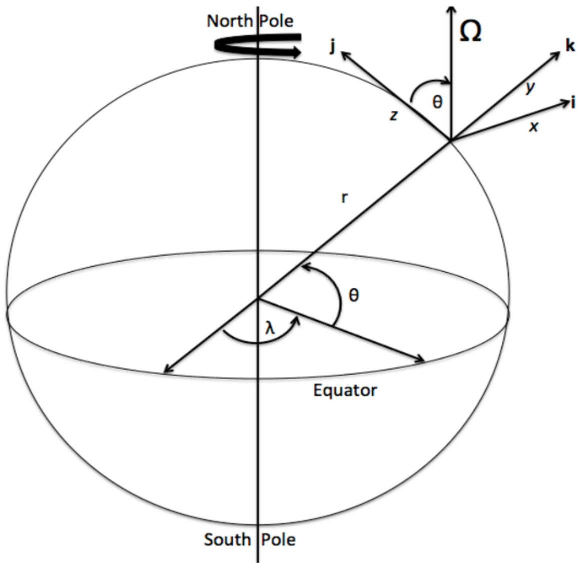

Figure 2.1: Representation of Cartesian coordinate frame

Ellipsoidal coordinates

Most of the planets have a similar shape to an oblate spheroid, this is an ellipsoid of revolution along its rotation axis meaning that the polar radius is smaller than the equatorial radius RP (flattening of poles). The use of ellipsoidal coordinate system is highly recommended, specially for large regions of atmosphere where high resolution and results accuracy is demanded.

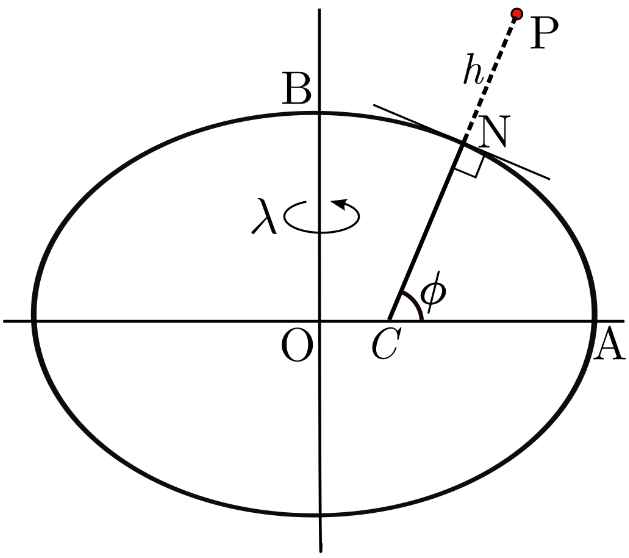

Figure 2.2: Title of the figure (Source: xxxx)

Here is the set of equations of the Shallow water model in an ellipsoidal coordinate system:

Chapter 3

Methodology

3.1 Equation computation

In order to compute the model, the domain has been discretized using the Arakawa-C grid and a finite differences approximation as it is shown in figure 3.1

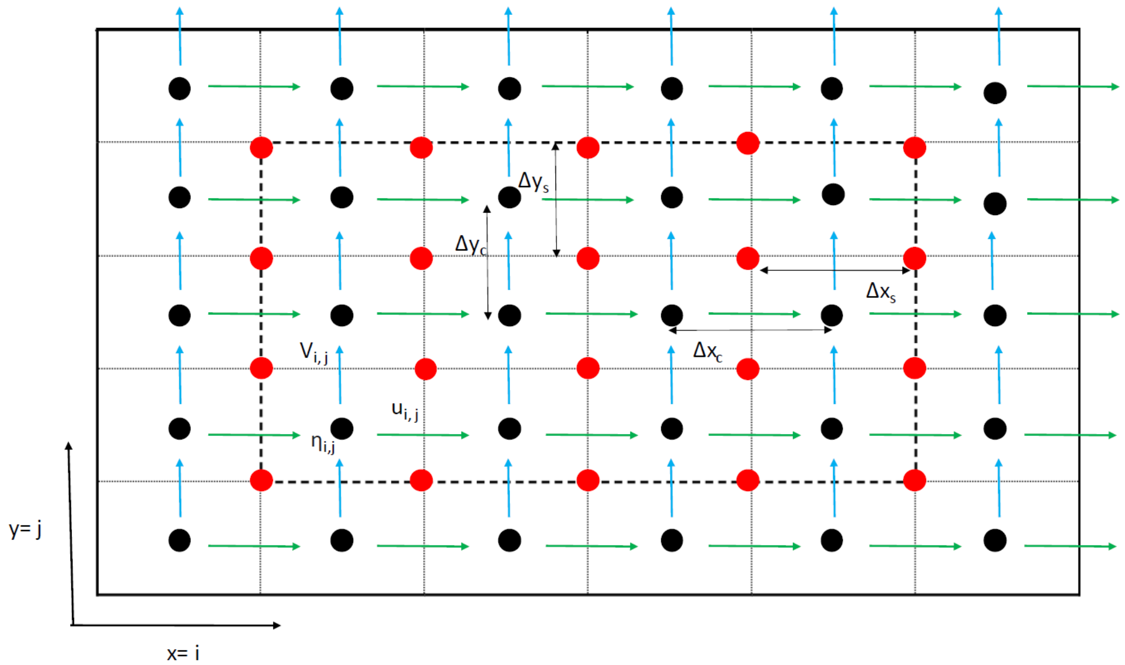

Figure 3.1: Arakawa C-grid (Source: Own)

The full development on the equations can be seen in [1] , [2] and Durran, D.R(2013)

P1u term

Looking at the x-axis component in the momentum equation, the P1u is defined as:

Splitting each velocity component, a similar expression is obtained discretized according to the ux direction.

$$\tilde{P_{1u}} = -\Delta t \nabla \cdot (u\vec{u}) = c_w^+ u_w^+ + c_w^- u_w^- - c_e^+ u_e^+ - c_e^- u_e^- + c_s^+ u_s^+ + c_s^- u_s^- - c_n^+ u_n^+ - c_n^- u_n^- \tag{3.1}$$

$$\left.\begin{aligned}
c_w^+ &= \frac{1}{2}(\frac{1}{2}(u_{i-1,j} + |u_{i-1,j}|) + \frac{1}{2}(u_{i,j} + |u_{i,j}|))\frac{\Delta t}{\Delta x} \\
c_w^- &= \frac{1}{2}(\frac{1}{2}(u_{i-1,j} - |u_{i-1,j}|) + \frac{1}{2}(u_{i,j} - |u_{i,j}|))\frac{\Delta t}{\Delta x} \\
c_e^+ &= \frac{1}{2}(\frac{1}{2}(u_{i,j} + |u_{i,j}|) + \frac{1}{2}(u_{i+1,j} + |u_{i+1,j}|))\frac{\Delta t}{\Delta x} \\
c_e^- &= \frac{1}{2}(\frac{1}{2}(u_{i,j} - |u_{i,j}|) + \frac{1}{2}(u_{i+1,j} - |u_{i+1,j}|))\frac{\Delta t}{\Delta x} \\
c_s^+ &= \frac{1}{2}(\frac{1}{2}(v_{i,j-1} + |v_{i,j-1}|) + \frac{1}{2}(v_{i+1,j-1} + |v_{i+1,j-1}|))\frac{\Delta t}{\Delta y} \\
c_s^- &= \frac{1}{2}(\frac{1}{2}(v_{i,j-1} - |v_{i,j-1}|) + \frac{1}{2}(v_{i+1,j-1} - |v_{i+1,j-1}|))\frac{\Delta t}{\Delta y} \\
c_n^+ &= \frac{1}{2}(\frac{1}{2}(v_{i,j} + |v_{i,j}|) + \frac{1}{2}(v_{i+1,j} + |v_{i+1,j}|))\frac{\Delta t}{\Delta y} \\
c_n^- &= \frac{1}{2}((v_{i,j} - |v_{i,j}|) + \frac{1}{2}(v_{i+1,j} - |v_{i+1,j}|))\frac{\Delta t}{\Delta y}
\end{aligned}\right\} \tag{3.2}$$

$$\left.\begin{aligned}
u_w^+ &= u_{i-1,j} + \frac{1}{2}\Psi(r_w^+)(1 - C_w^+)(u_{i,j} - u_{i-1,j}) \\
u_w^- &= u_{i,j} - \frac{1}{2}\Psi(r_w^-)(1 + C_w^-)(u_{i,j} - u_{i-1,j}) \\
u_e^+ &= u_{i,j} + \frac{1}{2}\Psi(r_e^+)(1 - C_e^+)(u_{i+1,j} - u_{i,j}) \\
u_e^- &= u_{i+1,j} - \frac{1}{2}\Psi(r_e^-)(1 + C_e^-)(u_{i+1,j} - u_{i,j}) \\
u_s^+ &= u_{i,j-1} + \frac{1}{2}\Psi(r_s^+)(1 - C_s^+)(u_{i,j} - u_{i,j-1}) \\
u_s^- &= u_{i,j} - \frac{1}{2}\Psi(r_s^-)(1 + C_s^-)(u_{i,j} - u_{i,j-1}) \\
u_n^+ &= u_{i,j} + \frac{1}{2}\Psi(r_n^+)(1 - C_n^+)(u_{i,j+1} - u_{i,j}) \\
u_n^- &= u_{i,j+1} - \frac{1}{2}\Psi(r_n^-)(1 + C_n^-)(u_{i,j+1} - u_{i,j})
\end{aligned}\right\} \tag{3.3}$$

$$\left.\begin{aligned}
r_w^+ &= \frac{u_{i-1,j} - u_{i-2,j}}{u_{i,j} - u_{i-1,j}} \\
r_w^- &= \frac{u_{i+1,j} - u_{i,j}}{u_{i,j} - u_{i-1,j}} \\
r_e^+ &= \frac{u_{i,j} - u_{i-1,j}}{u_{i+1,j} - u_{i,j}} \\
r_e^- &= \frac{u_{i+2,j} - u_{i+1,j}}{u_{i+1,j} - u_{i,j}} \\
r_s^+ &= \frac{u_{i,j-1} - u_{i,j-2}}{u_{i,j} - u_{i,j-1}} \\
r_s^- &= \frac{u_{i,j+1} - u_{i,j}}{u_{i,j} - u_{i,j-1}} \\
r_n^+ &= \frac{u_{i,j} - u_{i,j-1}}{u_{i,j+1} - u_{i,j}} \\
r_n^- &= \frac{u_{i,j+2} - u_{i,j+1}}{u_{i,j+1} - u_{i,j}}
\end{aligned}\right\} \tag{3.4}$$

P1v term

Looking at the y-axis component in the momentum equation, the P1v is defined as:

Splitting each velocity component, a similar expression is obtained discretized according to the vy direction.

$$\tilde{P_{1v}} = -\Delta t \nabla \cdot (v\vec{u}) = c_w^+ v_w^+ + c_w^- v_w^- - c_e^+ v_e^+ - c_e^- v_e^- + c_s^+ v_s^+ + c_s^- v_s^- - c_n^+ v_n^+ - c_n^- v_n^- \tag{3.5}$$

$$\left.\begin{aligned}
c_w^+ &= \frac{1}{2}(\frac{1}{2}(u_{i-1,j} + |u_{i-1,j}|) + \frac{1}{2}(u_{i,j} + |u_{i,j}|))\frac{\Delta t}{\Delta x} \\
c_w^- &= \frac{1}{2}(\frac{1}{2}(u_{i-1,j} - |u_{i-1,j}|) + \frac{1}{2}(u_{i,j} - |u_{i,j}|))\frac{\Delta t}{\Delta x} \\
c_e^+ &= \frac{1}{2}(\frac{1}{2}(u_{i,j} + |u_{i,j}|) + \frac{1}{2}(u_{i+1,j} + |u_{i+1,j}|))\frac{\Delta t}{\Delta x} \\
c_e^- &= \frac{1}{2}(\frac{1}{2}(u_{i,j} - |u_{i,j}|) + \frac{1}{2}(u_{i+1,j} - |u_{i+1,j}|))\frac{\Delta t}{\Delta x} \\
c_s^+ &= \frac{1}{2}(\frac{1}{2}(v_{i,j-1} + |v_{i,j-1}|) + \frac{1}{2}(v_{i+1,j-1} + |v_{i+1,j-1}|))\frac{\Delta t}{\Delta y} \\
c_s^- &= \frac{1}{2}(\frac{1}{2}(v_{i,j-1} - |v_{i,j-1}|) + \frac{1}{2}(v_{i+1,j-1} - |v_{i+1,j-1}|))\frac{\Delta t}{\Delta y} \\
c_n^+ &= \frac{1}{2}(\frac{1}{2}(v_{i,j} + |v_{i,j}|) + \frac{1}{2}(v_{i+1,j} + |v_{i+1,j}|))\frac{\Delta t}{\Delta y} \\
c_n^- &= \frac{1}{2}((v_{i,j} - |v_{i,j}|) + \frac{1}{2}(v_{i+1,j} - |v_{i+1,j}|))\frac{\Delta t}{\Delta y}
\end{aligned}\right\} \tag{3.6}$$

$$\left.\begin{aligned}
v_w^+ &= v_{i-1,j} + \frac{1}{2}\Psi(r_w^+)(1 - C_w^+)(v_{i,j} - v_{i-1,j}) \\
v_w^- &= v_{i,j} - \frac{1}{2}\Psi(r_w^-)(1 + C_w^-)(v_{i,j} - v_{i-1,j}) \\
v_e^+ &= v_{i,j} + \frac{1}{2}\Psi(r_e^+)(1 - C_e^+)(v_{i+1,j} - v_{i,j}) \\
v_e^- &= v_{i+1,j} - \frac{1}{2}\Psi(r_e^-)(1 + C_e^-)(v_{i+1,j} - v_{i,j}) \\
v_s^+ &= v_{i,j-1} + \frac{1}{2}\Psi(r_s^+)(1 - C_s^+)(v_{i,j} - v_{i,j-1}) \\
v_s^- &= v_{i,j} - \frac{1}{2}\Psi(r_s^-)(1 + C_s^-)(v_{i,j} - v_{i,j-1}) \\
v_n^+ &= v_{i,j} + \frac{1}{2}\Psi(r_n^+)(1 - C_n^+)(v_{i,j+1} - v_{i,j}) \\
v_n^- &= v_{i,j+1} - \frac{1}{2}\Psi(r_n^-)(1 + C_n^-)(v_{i,j+1} - v_{i,j})
\end{aligned}\right\} \tag{3.7}$$

$$\left.\begin{aligned} r_w^+ &= \frac{v_{i-1,j} - v_{i-2,j}}{v_{i,j} - v_{i-1,j}} \\ r_w^- &= \frac{v_{i+1,j} - v_{i,j}}{v_{i,j} - v_{i-1,j}} \\ r_e^+ &= \frac{v_{i,j} - v_{i-1,j}}{v_{i+1,j} - v_{i,j}} \\ r_e^- &= \frac{v_{i+2,j} - v_{i+1,j}}{v_{i+1,j} - v_{i,j}} \\ r_s^+ &= \frac{v_{i,j-1} - v_{i,j-2}}{v_{i,j} - v_{i,j-1}} \\ r_s^- &= \frac{v_{i,j+1} - v_{i,j}}{v_{i,j} - v_{i,j-1}} \\ r_n^+ &= \frac{v_{i,j} - v_{i,j-1}}{v_{i,j+1} - v_{i,j}} \\ r_n^- &= \frac{v_{i,j+2} - v_{i,j+1}}{v_{i,j+1} - v_{i,j}} \end{aligned}\right\} \tag{3.8}$$

Ph term

Looking at the y-axis component in the momentum equation, the Ph is defined as equation 3.9:

Splitting each velocity component, a similar expression is obtained discretized according to the vy direction.

$$\tilde{P_h} = -\Delta t \nabla \cdot (v\vec{u}) = c_w^+ h_w^+ + c_w^- h_w^- - c_e^+ h_e^+ - c_e^- h_e^- + c_s^+ h_s^+ + c_s^- h_s^- - c_n^+ h_n^+ - c_n^- h_n^- \tag{3.9}$$

$$\left.\begin{aligned} c_w^+ &= \frac{1}{2}(u_{i-1,j} + |u_{i-1,j}|)\frac{\Delta t}{\Delta x} \\ c_w^- &= \frac{1}{2}(u_{i-1,j} - |u_{i-1,j}|)\frac{\Delta t}{\Delta x} \\ c_e^+ &= \frac{1}{2}(u_{i,j} + |u_{i,j}|\frac{\Delta t}{\Delta x} \\ c_e^- &= \frac{1}{2}(u_{i,j} - |u_{i,j}|)\frac{\Delta t}{\Delta x} \\ c_s^+ &= \frac{1}{2}(v_{i,j-1} + |v_{i,j-1}|)\frac{\Delta t}{\Delta y} \\ c_s^- &= \frac{1}{2}(v_{i,j-1} - |v_{i,j-1}|)\frac{\Delta t}{\Delta y} \\ c_n^+ &= \frac{1}{2}(v_{i,j} + |v_{i,j}|)\frac{\Delta t}{\Delta y} \\ c_n^- &= \frac{1}{2}(v_{i,j} - |v_{i,j}|)\frac{\Delta t}{\Delta y} \end{aligned}\right\} \tag{3.10}$$

$$\left.\begin{aligned}
h_w^+ &= h_{i-1,j} + \frac{1}{2}\Psi(r_w^+)(1-C_w^+)(h_{i,j}-h_{i-1,j})\\
h_w^- &= h_{i,j} - \frac{1}{2}\Psi(r_w^-)(1+C_w^-)(h_{i,j}-h_{i-1,j})\\
h_e^+ &= h_{i,j} + \frac{1}{2}\Psi(r_e^+)(1-C_e^+)(h_{i+1,j}-h_{i,j})\\
h_e^- &= h_{i+1,j} - \frac{1}{2}\Psi(r_e^-)(1+C_e^-)(h_{i+1,j}-h_{i,j})\\
h_s^+ &= h_{i,j-1} + \frac{1}{2}\Psi(r_s^+)(1-C_s^+)(h_{i,j}-h_{i,j-1})\\
h_s^- &= h_{i,j} - \frac{1}{2}\Psi(r_s^-)(1+C_s^-)(h_{i,j}-h_{i,j-1})\\
h_n^+ &= h_{i,j} + \frac{1}{2}\Psi(r_n^+)(1-C_n^+)(h_{i,j+1}-h_{i,j})\\
h_n^- &= h_{i,j+1} - \frac{1}{2}\Psi(r_n^-)(1+C_n^-)(h_{i,j+1}-h_{i,j})
\end{aligned}\right\} \tag{3.11}$$

$$\left.\begin{aligned}
r_w^+ &= \frac{h_{i-1,j}-h_{i-2,j}}{h_{i,j}-h_{i-1,j}}\\
r_w^- &= \frac{h_{i+1,j}-h_{i,j}}{h_{i,j}-h_{i-1,j}}\\
r_e^+ &= \frac{h_{i,j}-h_{i-1,j}}{h_{i+1,j}-h_{i,j}}\\
r_e^- &= \frac{h_{i+2,j}-h_{i+1,j}}{h_{i+1,j}-h_{i,j}}\\
r_s^+ &= \frac{h_{i,j-1}-h_{i,j-2}}{h_{i,j}-h_{i,j-1}}\\
r_s^- &= \frac{h_{i,j+1}-h_{i,j}}{h_{i,j}-h_{i,j-1}}\\
r_n^+ &= \frac{h_{i,j}-h_{i,j-1}}{h_{i,j+1}-h_{i,j}}\\
r_n^- &= \frac{h_{i,j+2}-h_{i,j+1}}{h_{i,j+1}-h_{i,j}}
\end{aligned}\right\} \tag{3.12}$$

P2u Advective term

P2 term is more straightforward to compute than the P1 and Ph terms.

It will be used a finite differences scheme, splitting the components depending on which term is computed (P2u o P2v)

$$P2u = u(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}) \tag{3.13}$$

Applying a finite differences method, a numerical expression can be obtained,but the staggered mesh requires velocity interpolation depending on the component u. In the case of v, component interpolation results in a more complicated expression. Because velocity v is not at the same place of the u velocity, two interpolations are needed to be done. First, the average value at the top and bottom of the control volume is done.

$$\frac{\partial u}{\partial x} = \frac{u_{i+1,j}-u_{i-1,j}}{2\Delta x} \tag{3.14}$$

$$v_{top} = \frac{(v_{i,j} + v_{i+1,j})}{2} \tag{3.15}$$

$$v_{bottom} = \frac{(v_{i,j-1} + v_{i+1,j-1})}{2} \tag{3.16}$$

But on the other hand, v velocity component interpolation results in a less easier expression. Since velocity v is not at the same place of the u velocity, two interpolations needs to be done. Firstly, the average value at the top and bottom of the control volume is done.

And then is computed as:

$$\frac{\partial v}{\partial y} = \frac{v_{top} - v_{bottom}}{\Delta y} \tag{3.17}$$

Obtaining the following expression (Eq.3.18) for the whole term:

$$\tilde{P_{2u}} = u_{i,j}(\frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} + \frac{(v_{i,j} + v_{i+1,j}) - (v_{i,j-1} + v_{i+1,j-1})}{2\Delta y}) \tag{3.18}$$

P2v Advective term

This term uses the same logic as the previous one (P2u), only changing the u and v discretization

$$P2v = v(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}) \tag{3.19}$$

So, the equations rearrange as:

$$\frac{\partial v}{\partial y} = \frac{u_{i,j+1} - u_{i,j-1}}{2\Delta y} \tag{3.20}$$

$$u_{top} = \frac{(u_{i,j} + u_{i,j+1})}{2} \tag{3.21}$$

$$u_{bottom} = \frac{(u_{i-1,j+1} + u_{i-1,j})}{2} \tag{3.22}$$

And then is computed as:

$$\frac{\partial u}{\partial x} = \frac{u_{top} - u_{bottom}}{\Delta x} \tag{3.23}$$

Obtaining the following expression (Eq.3.24) for the whole term:

$$\tilde{P_{2v}} = v_{i,j}(\frac{(u_{i,j} + u_{i,j+1}) - (u_{i-1,j+1} + u_{i-1,j})}{2\Delta x} + \frac{v_{i,j+1} - v_{i,j-1}}{2\Delta y}) \tag{3.24}$$

Ppu term

The pressure term for the u component is computed with a finite differences scheme, resulting in equation 3.25

$$\tilde{P_{pu}} = -g\frac{\partial \eta}{\partial x} = -g\frac{\eta_{i+1,j} - \eta_{i,j}}{\Delta x} \tag{3.25}$$

Ppv term

The pressure term for v is computed in a similar way to the previous term (Ppu) and it is computed like in equation 3.26

$$\tilde{P_{pv}} = -g\frac{\partial \eta}{\partial y} = -g\frac{\eta_{i,j+1} - \eta_{i,j}}{\Delta y} \tag{3.26}$$

3.2 Flux limiter

Upwind schemes are used in computational physics to solve hyperbolic equations as the advection terms.

Is numerically solver for u_j using the immediate u_{i-1} "upwind" value for the lowest order scheme or several "upwind" values, u_{i-1} i u_{i-2} , if the scheme is not superior order.

Non-linear schemes can be defined combining different explicit Euler methods in order to apply correctly the TVD method. The next expression is used to obtain the fluxes on the different faces on the control volume.

$$\phi_e = \phi^L_{i,j} + \frac{1}{2}\Psi(r)\left(\phi^H_{i+1,j} - \phi^L_{i,j}\right) \tag{3.27}$$

In which $\phi_{i,j}$, is the transported flux variable (u or v for instants) and ϕ^L and ϕ^H are related to low order or high order upwind advection variable.

The Ψ function is the flux limiter which is the responsible on the local control of the flux solution. If there are local extremes in the solution, the function tends to zero, while when the solution is smooth, the flux limiter grows to 1 in order to improve the precision order.

The r parameter, named as upwind/downwind, is very relevant in order to define the value of the flux limiter.

$$r = \frac{\phi_{i,j} - \phi_{i-1,j}}{\phi_{i+1,j} - \phi_{i,j}} \tag{3.28}$$

If r_j is near or equal to u, while for $r_j < 0$, it appears a local extreme. For this, depending on the value of r, Ψ will change its value. This makes possible to establish the TVD condition , the derivation of which can be obtained with:

$$0 \leq C + \frac{C(1-C)}{2}\left[\frac{\Psi_i}{r} - \Psi_{i-1}\right] \leq 1 \tag{3.29}$$

In the Shallow Water model, the Superbee scheme is used for low to moderate resolution, while MUSCL is used in all the high resolution simulations.

$$Superbee \rightarrow \psi(r) = max[0, (2r, 1), min(r, 2)] \tag{3.30}$$

$$MUSCL \rightarrow \psi(r) = max[0, min(2, 2r, \frac{1+r}{2})] \tag{3.31}$$

In the code it is chosen the Superbee scheme since it is a bit easier to compute and it doesn't make a big differences in our simulations.

3.3 Numerical integration

The Shallow water equations can be divided en two different terms: the terms with the temporal derivative on the left side of the expression , and the advection and the other terms on the right hand side. The first equation will be rewritten like this:

$$\frac{\partial u}{\partial t} = \frac{P_{1u}}{\Delta t} + P_{2u} + P_{pu} \tag{3.32}$$

It is important to say that the Coriolis parameters had been taken out from the expression, because this parameters need a different treatment after the updated velocities are updated.

When the advective term and the gradients of pressure are computed, the numerical integration will be performed before applying Coriolis.

In order to integrate the velocities, it will be used a third order Adams-Bashforth scheme.

$$u^{n+1} = u^n + \Delta t(\frac{23}{12}f_1^n - \frac{4}{3}f_1^{n-1} + \frac{5}{12}f_1^{n-2}) \tag{3.33}$$

$$v^{n+1} = v^n + \Delta t(\frac{23}{12}f_2^n - \frac{4}{3}f_2^{n-1} + \frac{5}{12}f_2^{n-2}) \tag{3.34}$$

$$\eta^{n+1} = \eta^n + \Delta t(\frac{23}{12}f_3^n - \frac{4}{3}f_3^{n-1} + \frac{5}{12}f_3^{n-2}) \tag{3.35}$$

On u^{n+1}, v^{n+1} and η^{n+1} are the time update horizontal velocities and free surface perturbation u^n, v^n and

η^n are their values a time step before and the f^n, f^{n-1} and f^{n-2} are the f_1, f_2 and f_3 already computed three time steps before (n, n-1, n-2).

Because the expression of the numerical integration, involves function terms up to two previous time steps, it requires the use of Euler's method for the first two iterations ($n = 0$ i $n = 1$), this will be computed like this:

It is important to say that the time step (Δ_t) is considered like a constant

3.4 Coriolis integration

It will be used an explicit Euler method for the temporal numerical integration

$$\frac{\mathrm{d}u}{\mathrm{d}t} - fv = 0 \rightarrow \frac{\widetilde{u}^{n+1} - \widetilde{u}^n}{\Delta t} = f\widetilde{v}^n \tag{3.36}$$

$$\frac{\mathrm{d}v}{\mathrm{d}t} + fu = 0 \rightarrow \frac{\widetilde{v}^{n+1} - \widetilde{v}^n}{\Delta t} = f\widetilde{u}^n \tag{3.37}$$

If we do the module on the velocity, this will increase its value over time

$$\left\| \overrightarrow{\widetilde{u}} \right\|^2 = (\widetilde{u}^n)^2 + (\widetilde{v}^n)^2 = (1 + f^2\Delta t^2)^n \left[(\widetilde{u}^0)^2 + (\widetilde{v}^0)^2\right] \tag{3.38}$$

We will rewrite all now as:

$$\frac{\mathrm{d}u}{\mathrm{d}t} - fv = 0 \rightarrow \frac{\widetilde{u}^{n+1} - \widetilde{u}^n}{\Delta t} = f\widetilde{v}^{n+1} \tag{3.39}$$

$$\frac{\mathrm{d}v}{\mathrm{d}t} + fu = 0 \rightarrow \frac{\widetilde{v}^{n+1} - \widetilde{v}^n}{\Delta t} = f\widetilde{u}^{n+1} \tag{3.40}$$

Now with a semi-implicit method, will result the following expressions:

$$\frac{\widetilde{u}^{n+1} - \widetilde{u}^n}{\Delta t} = f\left[(1-\alpha)\widetilde{v}^n + \alpha\widetilde{v}^{n+1}\right] \tag{3.41}$$

$$\frac{\widetilde{v}^{n+1} - \widetilde{v}^n}{\Delta t} = f\left[(1-\alpha)\widetilde{u}^n + \alpha\widetilde{u}^{n+1}\right] \tag{3.42}$$

Where $0 \leq \alpha \leq 1$, Cushman-Roisin and Beckers show that:

$$(\widetilde{u}^n)^2 + (\widetilde{v}^n)^2 = 1$$

If $\alpha = 0.5$, then $\|(\widetilde{u}^n)\|^2 = \left\|(\widetilde{u}^0)\right\|^2$, and therefore the kinetic energy is conserved.

For this, the velocities after Coriolis are computed like:

$$u^{n+1} = u^* + 0.5f(v^n + v^{n+1}\alpha) \tag{3.43}$$

$$v^{n+1} = v^* + 0.5f(u^n + u^{n+1}\alpha) \tag{3.44}$$

The velocity at the current instant, is denoted by the velocity obtained from the Adams-Bashforth integrationand it is named u^* and u^n and v^n are the old velocities before being correcte by Coriolis

Moreover, the velocities u^n and v^n need to be interpolated due the staggering of the mesh

$$u^n_{avg}(i,j) = 0.25(u^n_{i+1,j-1} + u^n_{i+1,j} + u^n_{i,j} + u^n_{i,j-1}) \tag{3.45}$$

$$v^n_{avg}(i,j) = 0.25(v^n_{i,j} + v^n_{i,j+1} + v^n_{i-1,j} + v^n_{i-1,j+1}) \tag{3.46}$$

If we apply all of this in the equation , we obtain the following expressions for the velocities:

$$u^{n+1}_{i,j} = \frac{u^*_{i,j} - (\alpha f_{i,j}\Delta t)^2 u^n_{i,j} + 2\alpha f_{i,j}\Delta t v^n_{avg}(i,j)}{1 + (\alpha f_{i,j}\Delta t)^2} \tag{3.47}$$

$$v^{n+1}_{i,j} = \frac{v^*_{i,j} - (\alpha f_{i,j}\Delta t)^2 v^n_{i,j} + 2\alpha f_{i,j}\Delta t u^n_{avg}(i,j)}{1 + (\alpha f_{i,j}\Delta t)^2} \tag{3.48}$$

3.5 Boundary conditions

To find the solutions of the equations of the Shallow water model Eq.2.14, we must impose some limits in the numerical domain that will be called boundary conditions. The most common are Double Periodic and Full-Slip conditions

First we will focus in Double Periodic Conditions (DPC), are very useful for code verification purposes and in testing ordinary cases. They remain the same expression for u, v and η. For u have this expression:

- Bottom
  u[:,1]=u[:,N_y+1]
  u[:,2]=u[:,N_y+2]
- Top
  u[:,N_y +3]=u[:,3]
  u[:,N_y +4]=u[:,4]
- Left
  u[1,:]=u[N_x+1,:]

u[2,:]=u[N_y+2,:]

- Right

  u[N_y +3,:]=u[3,:]

  u[N_y +4,:]=u[4,:]

The other condition (Full-Slip) is interesting in cases with frictional effects of flow running parallel to coastlines, which are typically computed by implementing a velocity shear near the coast.

In this case the normal component is assumed to be zero, as if there was not flow penetration through boundaries, and for the different variables it changes its expression.

For u have this expression:

- Bottom

  u[:,1]=u[:,3]

  u[:,2]=u[:,3]

- Top

  u[:,N_y +3]=u[:,N_y +2]

  u[:,N_y +4]=u[:,N_y +2]

- Left

  u[1,:]=0

  u[2,:]=0

- Right

  u[N_y +2,:]=0

  u[N_y +3,:]=0

  u[N_y +4,:]=0

For v have this expression:

- Bottom

  v[:,1]=0

  v[:,2]=0

- Top

  v[:,N_y +2]=0

  v[:,N_y +3]=0

  v[:,N_y +4]=0

- Left

  v[1,:]=v[2,:]

  v[2,:]=v[3,:]

- Right

u[N_y +3,:]=u[N_y +2,:]

u[N_y +4,:]=u[N_y +3,:]

And in the case of η's halo, must be adapted, establishing a 0 in the hole halo position

3.6 MMS Validation method

In order to validate the functions used for the code, a validation method must be chosen.

The most common methods for code validation are the method of exact solutions (MES) and the method of the manufactures solutions (MMS).

If we consider the following partial diferential equation (PDE):

$$\frac{\mathrm{D}\vec{u}}{\mathrm{D}t} = g \tag{3.49}$$

where $\vec{u}$ is the solution and g is the source term, the MES method chooses the function g and the solves the equation using mathematical tools as variable separation. In the other hand, the method of the manufactured solutions (MMS), applies the same procedure in inverted order, this meaning that first it is chosen the function $\vec{u}$ and the the total derivative (D/Dt) is applied to obtain the source term g.

For the purpose of this work the method of manufactured solutions (MMS) will be the one used for the code validation.

In the Shallow water model, the advective terms can be validated for separated like $Df = 0$, where D is the total derivative and f stands for the advective term to be validated.

So, an arbritary function called f_M (the manufactured solution is labeled with the subscript M) is chosen and after applying the material derivative, it appears a source term in the form of $Df_M = F_M$

Taking this into account, is logical then than the code must validate the following relation:

$$Df - F_m = 0 \tag{3.50}$$

When the numerical solution is computed, the error between the two functions in all the domain is evaluated as:

$$\|f - F_M\|_\infty = max\,|f - F_M| \tag{3.51}$$

For discretized domains, the equation 3.51 is proportional to a constant value named K and to the grid's size labeled as h, obtaining the following expression:

$$\|f - f_M\| = Kh^p \tag{3.52}$$

where p stands for the accuracy order of the function.

In the validation of the Shallow water model for periodic boundary conditions in x and y it is convenient to use periodic functions, like the following:

$$u_M = sin(2\pi x)cos(2\pi y)t \tag{3.53}$$

$$v_M = cos(2\pi x)sin(2\pi y)t \tag{3.54}$$

$$h_M = 0.01cos(2\pi x)cos(2\pi y)cos(t) + D \tag{3.55}$$

where t is considered to be a constant value set different to zero in any time instant, and the value D, must always follow the aspect ratio hypothesis ($D << L$)

The validation process consists in the evaluation of the advective terms at each point, always taking into consideration the staggering. In order to simplify the analysis, it has been taken an squared domain, with the spatial discretization (,) remaining constant. This process should give a second order precision for the advective terms and a third order for the numerical integration.

Advection Terms validation. P1

As it has been described in section 3.1, the term can be expressed as:

$$P1u = \nabla \cdot (u\vec{u}) = \frac{\partial\,(uu)}{\partial x} + \frac{\partial (uv)}{\partial y} \tag{3.56}$$

$$P1v = \nabla \cdot (v\vec{u}) = \frac{\partial\,(vu)}{\partial x} + \frac{\partial (vv)}{\partial y} \tag{3.57}$$

And the error calculated as:

$$e = \left\| P1u - \tilde{P1}u/\Delta t \right\|_\infty \tag{3.58}$$

$$e = \left\| P1v - \tilde{P1}v/\Delta t \right\|_\infty \tag{3.59}$$

If we apply the MMS validation to the P1 term, we can see that the error obtained is reduced when the density of the mesh increases. Because the condition of the Courant number must be always fulfilled, Δx and Δt can not have arbitrary values.

The parameters are set as: $\Delta x = [50, 100, 200, 500]$ and $\Delta t = 10^{-6}$

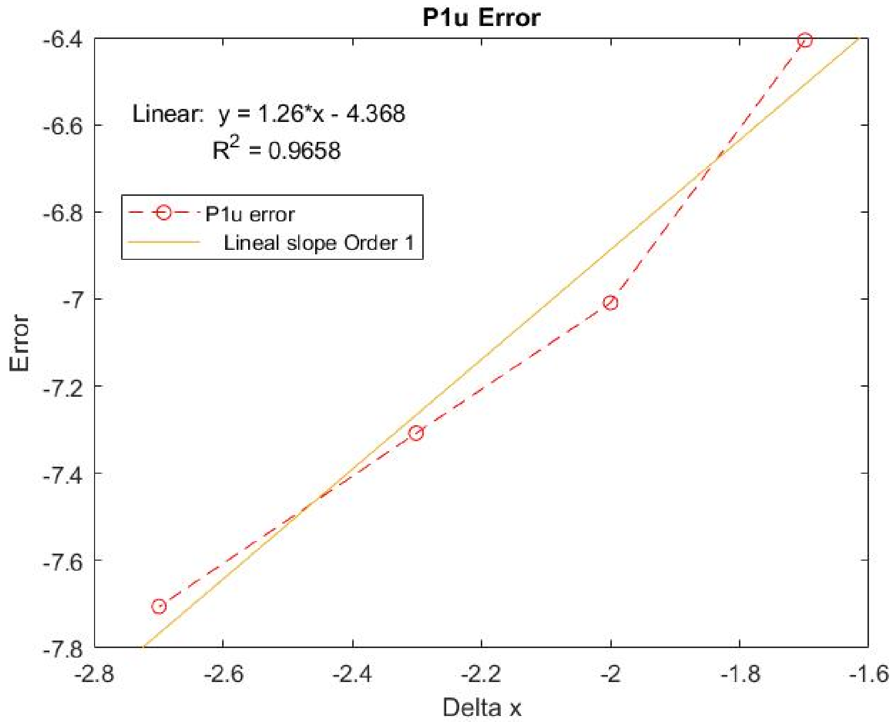

Figure 3.2: $\tilde{P1u}$ Validation error (Source: Own)

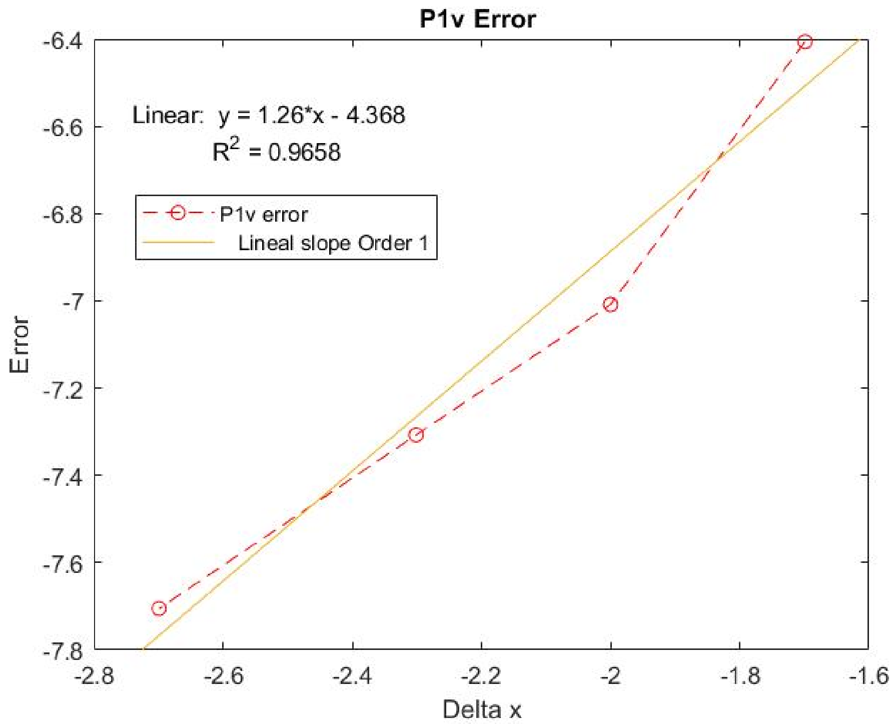

Figure 3.3: $\tilde{P1v}$ Validation error (Source: Own)

If we take a look on the results on the figures 3.2 and 3.3, it may looks strange that the precision is less than the expected, but this is in account of the TVD scheme switches between a first and second order schemes.

Also, when the density of the mesh is not enough, the error of the term $\tilde{P1}$ has some instabilities.

The same behaviour has been found in previous studies, so results can be considered acceptable.

Advection Terms validation. P2

As it has been described in section 3.1, the term can be expressed as:

$$P2u = u(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}) \tag{3.60}$$

$$P2v = v(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}) \tag{3.61}$$

And the error calculated as:

$$e = \left\| P2u - \tilde{P2u} \right\|_\infty \tag{3.62}$$

$$e = \left\| P2v - \tilde{P2v} \right\|_\infty \tag{3.63}$$

$$\tag{3.64}$$

The process followed for the validation is similar to the one expressed for the P1 term, with the parameters are set as: $\Delta x = [50, 100, 200, 500]$

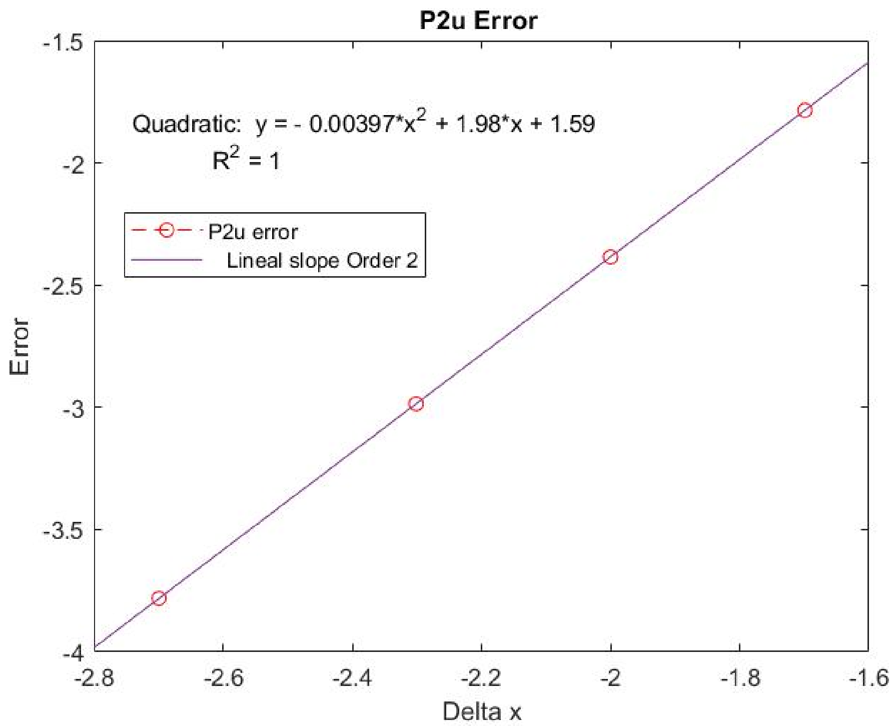

Figure 3.4: $\tilde{P2u}$ Validation error (Source: Own)

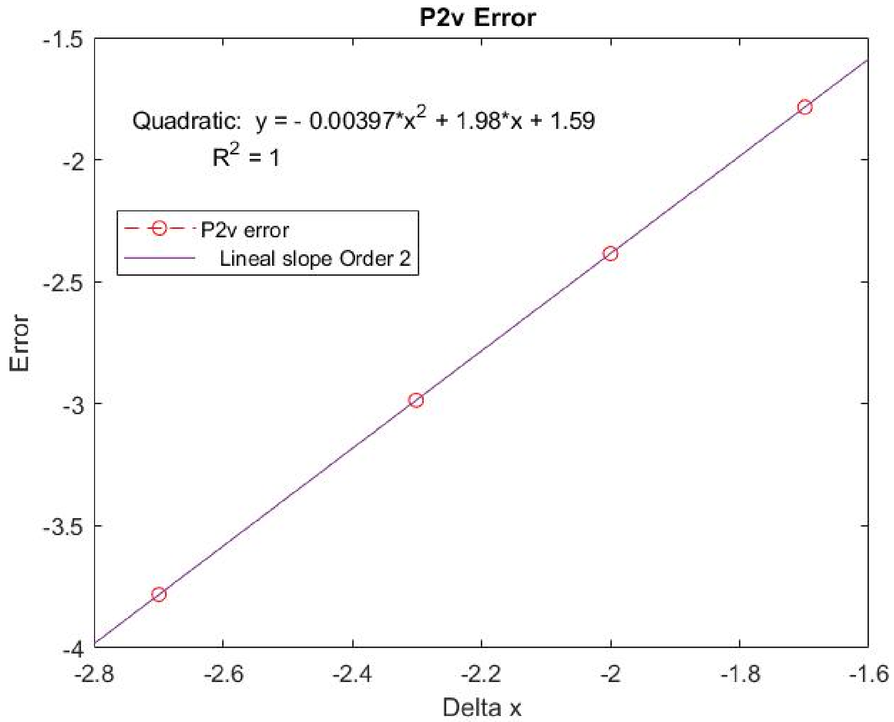

Figure 3.5: $\tilde{P2v}$ Validation error (Source: Own)

If we take a look on the results on the figures 3.4 and 3.5 it can be seen the second order accuracy.

Advection Terms validation. Ph i Pp

The term of the continuity equation (Ph) and the pressure contribution (Pp)can be written as:

$$Ph = \nabla \cdot (h\vec{u}) = \frac{\partial (hu)}{\partial x} + \frac{\partial (hv)}{\partial y} \tag{3.65}$$

$$Ppu = -g\frac{\partial \eta}{\partial x} \tag{3.66}$$

$$Ppv = -g\frac{\partial \eta}{\partial y} \tag{3.67}$$

The term Ph is time dependent like P1 and Pp follows the same trend as the P2 term.

And the error calculated as:

$$e = \left\| Ph - \tilde{Ph}/\Delta t \right\|_\infty \tag{3.68}$$

$$e = \left\| Ppu - \tilde{Ppu} \right\|_\infty \tag{3.69}$$

$$e = \left\| Ppv - \tilde{Ppv} \right\|_\infty \tag{3.70}$$

The process followed for the validation is similar to the ones expressed for the P1 and P2 terms, with the parameters are set as: $\Delta x = [50, 100, 200, 500]$ and $\Delta t = 10^{-6}$

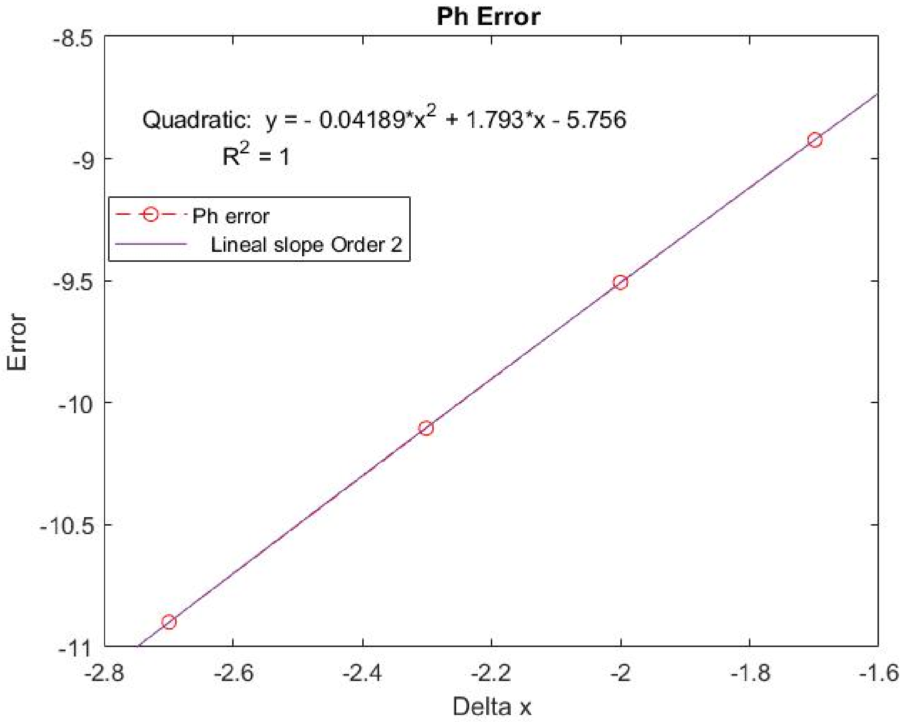

Figure 3.6: $\tilde{Ph}$ Validation error (Source: xxxx)

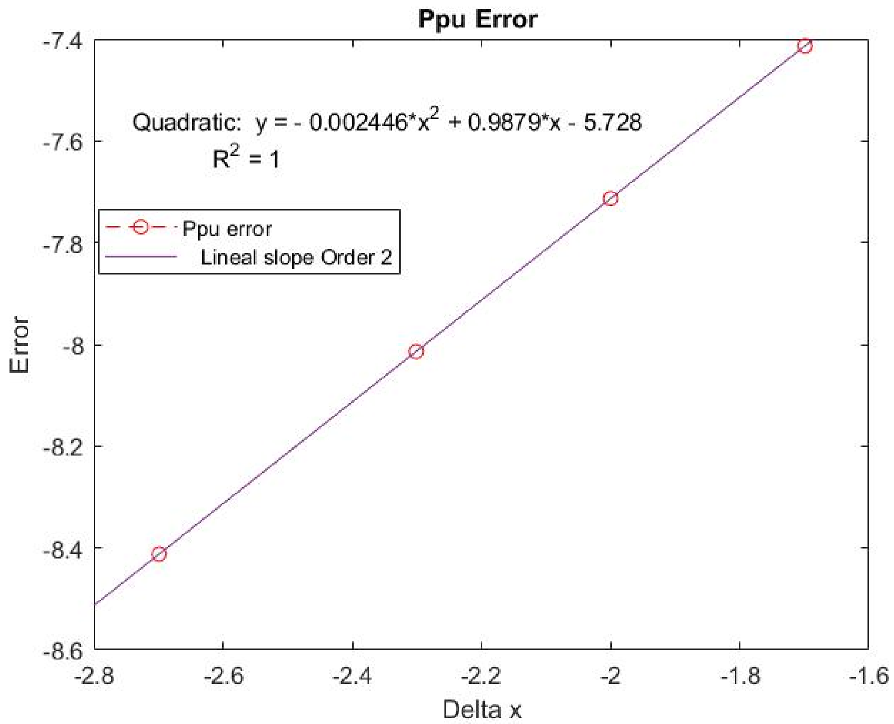

Figure 3.7: $\tilde{Ppu}$ Validation error (Source: Own)

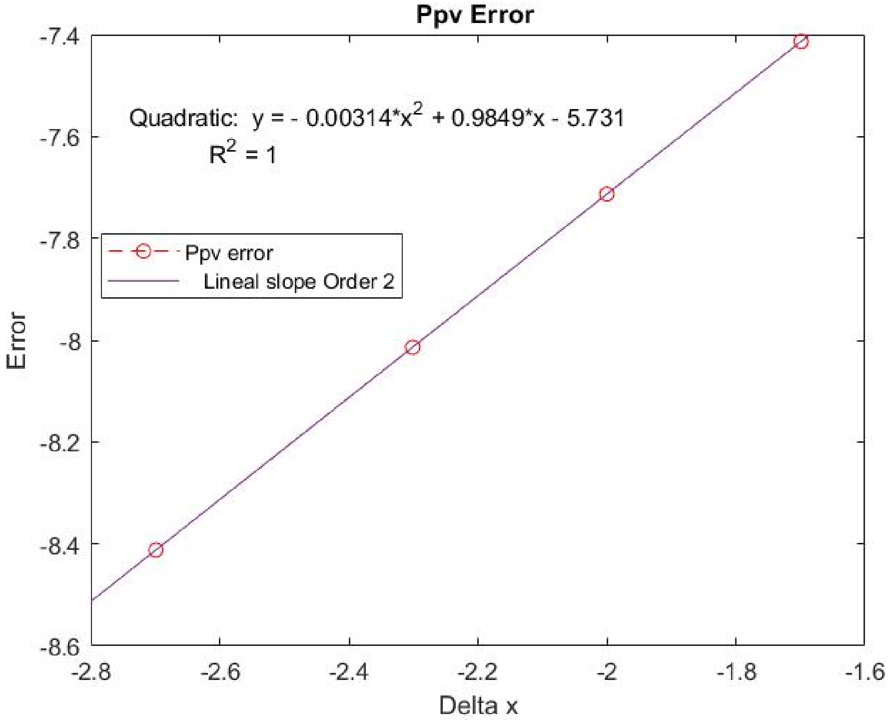

Figure 3.8: $\tilde{Ppv}$ Validation error (Source: Own)

Time integration Validation

The time integration as it is explained in section 3.3, is performed with a third order Adams-Bashforth scheme.

In order to test it, it has been calculated the integration of function an comparing the evaluation of the analytical solution with the solution obtained the function this gives an exact solution, as it is a third order approximation.

Chapter 4

Results and discussion

In order to test the code, some practical cases will are intended to be conducted in this chapter

4.1 Gravity wave simulation

In this section it will be represented the propagation of gravity waves with an initial perturbation. This simulation will help to test if the model works as expected in an homogeneous and non-rotating layer of fluid, and its dynamics.

The perturbation is modelled using Gaussian function that has this expression:

$$\eta(x, y) = A \cdot exp\left[(-\frac{(x - x_0)^2}{2\theta_x^2}) - (-\frac{(y - y_0)^2}{2\theta_y^2})\right] \tag{4.1}$$

,where A is the amplitude of the function, x0 and y0 is where the perturbation in located in the domain, and θ_x and θ_y are the values of the standard deviation.

The values to perform the simulation can not be decided arbitrary, and must always validate the Courant condition

$$C = \frac{U\Delta t}{\Delta x} = 0.5 \tag{4.2}$$

For discretion, in a constant number of square cells N = 100, the time increment can be estimated considering that the phase speed propagation of a gravity wave in a the system for a value of D = 5m is $U = \sqrt{gD} = 7m/s$, obtaining a $\Delta t < 0.14$, but in order to have a better solution, is computed $\delta t = 0.05$

The domain is an square with 200m of length and a fluid depth of D=5 m.

The parameters of the perturbation are:

A[m]	θ_x	θ_y	x_0 [m]	y_0 [m]
1	4	4	100	100

Table 4.1: Parameters Gaussian perturbation

Figures/heightevol.jpg

Figure 4.1: Surface height evolution (Source: Own)

4.2 Cartesian Vortex simulation

Once the gravity wave is successfully implemented the next step is to introduce a vortex into the model. By means of this simulation the integration of the Coriolis parameter can be tested.

Geostrophic balance

If *Ro* << 1, material derivatives of u and v can be neglected in front of Coriolis acceleration, resulting the momentum equations as:

$$-fv = -g\frac{\partial \eta}{\partial x} \quad (4.3)$$

$$fu = -g\frac{\partial \eta}{\partial y} \quad (4.4)$$

If the geostrophic wind is exclusively zonal ($v = 0, u = U(y)$), η can be only dependent on y. In this case, U(y) can be obtained as:

$$\eta_z(y) = -\int \frac{f(y)}{g(y)} U(y)dy + constant \quad (4.5)$$

Given U(y), the model should be initialized by computing z(y) ,

xing it as the initial topography of the free surface, and introducing U(y) as the velocity initial condition (u0 = U; v0 = 0).

Zonal winds

It is assumed free surface topography which can be decomposed into $\eta = \eta' + \eta_z$, where η' represents the pertorbation added to the zonal wind topography, called η_z

In this case, the resulting equation for v is:

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + fu = -g\frac{\partial \eta^{'}}{\partial y} - g\frac{\partial \eta_z}{\partial y} = -g\frac{\partial \eta^{'}}{\partial y} + fU \tag{4.6}$$

And can be rearranged as:

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + f(u - U) = -g\frac{\partial \eta^{'}}{\partial y} \tag{4.7}$$

If now the velocity u is written as the sum of the velocity of the pertorbation plus the zonal wind ($u = u' + U$), results in equation 4.8:

$$\frac{\partial v}{\partial t} + (u^{'} + U)\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + fu^{'} = -g\frac{\partial n^{'}}{\partial y} \tag{4.8}$$

If the same procedure is applied to the equation for u, it will be obtained:

$$\frac{\partial u^{'}}{\partial t} + (u^{'} + u)\frac{\partial u^{'}}{\partial x} + v\frac{\partial (u^{'} + U)}{\partial y} - fv = -g\frac{\partial r^{'}}{\partial x} \tag{4.9}$$

And the mass conservation equation remains as:

$$\frac{\partial \eta}{\partial t} + \frac{\partial (h(u^{'} + U))}{\partial x} + \frac{\partial hv}{\partial y} = 0 \tag{4.10}$$

To sum up, adding the zonal wind perturbation or the topography in the free surface makes the same effect. The condition we must always impose is that h must integrate the undisturbed thickness D. The basic change numerically is that the zonal wind must be added to the perturbation and not to the Coriolis term.

Introduction of the perturbation

The vortex is created with a combination of a Gaussian function and the Coriolis effect. The perturbation will have the same expression as the gravity wave simulation, but the domain an parameters will be adjusted to the simulation goal.

$$S(x, y) = A \cdot exp\left[(-\frac{(x - x_0)^2}{2\theta_x^2}) - (-\frac{(y - y_0)^2}{2\theta_y^2})\right] \tag{4.11}$$

The perturbation expressed in Eq.4.11 will be introduced progressively into the simulation in order to prevent large gravity moves to make a disturbance that shocks the model and delays the stabilisation of the vortex.

The goal is to increment the vortex height on every iteration for a few simulation days, and when the full amplitude of the perturbation is reached, let the perturbation drop and evolve over time

Figures/height.jpg

Figure 4.2: Surface height evolution (Source: Own)

It is rellevant to mention that the dissipation effect that creates que decreasing in the vortex is due the elevated time step used in the simulation ($\Delta t = 30s$)

The parameters for the simulation will be:

	A	$R_{max}*$	θ_x	θ_y
Cartesian Domain	3000 m	3500 m	1400	1400

Table 4.2: Vortex perturbation parameters

, where R stands for the maximum radii allowed fro the perturbation, A for the maximum amplitude and θ values for the standard deviation

The vortex rotation will be displayed using the potential vorticity, which is conserved over time, so it can be used as a tracer.

The potential vorticity is computed like:

$$\Pi = q = \frac{f + \omega}{h} \tag{4.12}$$

where f is the Coriolis parameter, w is the relative vorticity and h is the total height of the fluid.

Because vorticity is computed at each up-right cell corner, like it is shown in figure h requires an interpolation for η, leaving equation 4.13 to compute it

$$h = D + \frac{\eta_{i,j} + \eta_{i+1,j} + \eta_{i,j+1} + \eta_{i+1,j+1}}{4} \tag{4.13}$$

And finally, relative vorticity (ω) is computed without interpolation as:

$$\omega = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \tag{4.14}$$

4.3 Amsterdam island simulation

4.3.1 Orographic gravity waves

In stable environments, strong winds blowing against an object (like a mountain) are forced to raise on the windward side and descend along the downwind slope.
The distributed airflow begins to oscillate in a series of waves as it moves downstream, generating mountain waves

Mountain (lee) waves in Rotating Shallow Water

We will now consider a situation where topograpy is localised: a configuration with a single mountain, and consider perturbations (mountain waves) created by this mountain in a mean flow.

In fist place it is computed the most simpler case: horitzonatal uniform. The one-layer RSW equations on the f plane in the presence

$$u_t + (u+F)u_x + vu_y - (Bu)^{-0.5}v + h_x = 0 \tag{4.15}$$

$$v_t + (u+F)v_x + vv_y + (Bu)^{-0.5}u + h_y = 0 \tag{4.16}$$

$$h_t + ((u-F)(h-Mb))_x + (v(h-Mb))_y = 0 \tag{4.17}$$

Figures/islandgauss.jpg

Figure 4.3: Surface height evolution (Source: Own)

Chapter 5

Budget summary and/or economic feasibility study

The total budget cost is 6301.08 €and it is presented in table 5.1

Item	Description	Unitary cost	Total units	Total
1	Hours of work	8€/h	600 h	4800 €
2	Electricity	0.0045 €/kWh	50 kW · 570 h	128.25 €
3	Software licences	800 €	1	800
4	Contingency			572.83 €
Total				6301.08 €

Table 5.1: Total cost break down

Chapter 6

Analysis and assessment of environmental and social implications

A brief analysis and assessment of how the design and/or study that the thesis focuses on considers environmental and social considerations or has a positive impact in these dimensions. If we take into account the carbon dioxide generated by the main source of work, it is obtained that for each kWh consumed by the computer, it is generated, being this a big number.

If we look to the social implications, climate and atmospheric prediction have a huge part in the prediction and prevention natural disasters, with unfortunately with the climate crisis are becoming more frequent and violent.

Also, the simulations of the atmospheres regarding other planets, can help us to understand better our own and its phenomena

Bibliography

1. ENRIQUE GARCIA-MELENDO, Sergio Coll Ibars. *INTRODUCTION TO THE NUMERICAL SIMULATION OF ENVIRONMENTAL AND ATMOSPHERIC FLOWS*. 2021.
2. LEONARDO ANDRÉS MOYA ACOSTA Enrique García, Manel Soria. *High performance computing simulation of planetary atmospheres*. 2020.
3. GEORGES, Jean-Yves. *Amsterdam Island* [`http://jygeorges.free.fr/english/ams01b.html`]. [N.d.]. Consulted: 15/03/2023.