

Numerical Methods in Meteorology and Oceanography



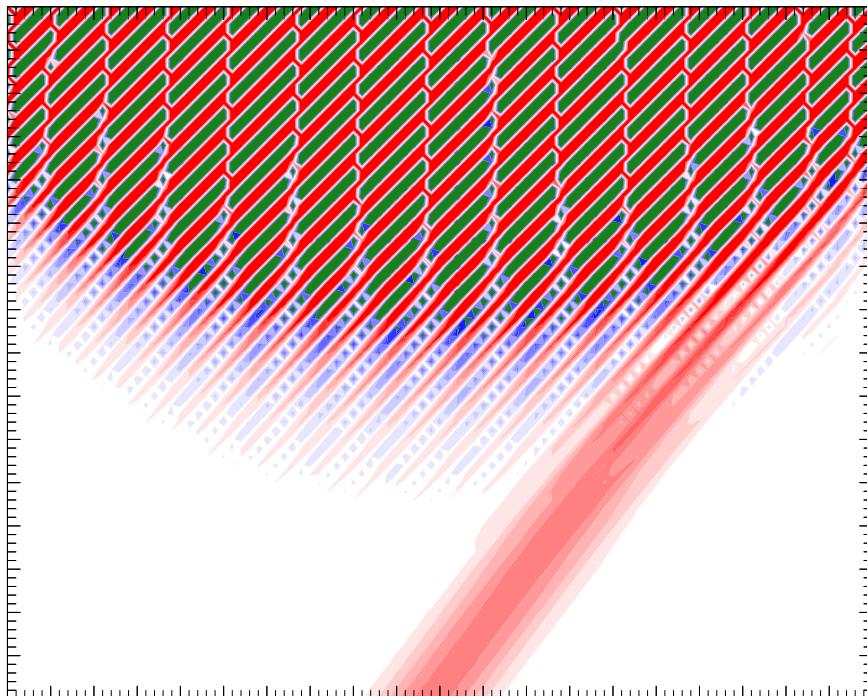
Kristofer Döös, Laurent Brodeau and Peter Lundberg

Department of Meteorology

Stockholm University

http://doos.misu.su.se/pub/numerical_methods.pdf

E-mail: doos@misu.su.se



January 11, 2017

Contents

1	Introduction	9
1.1	What is a numerical model of the circulation of the atmosphere or the ocean?	9
1.2	Brief historical background	9
2	Partial Differential Equations	13
2.1	Elliptic Equations	14
2.2	Parabolic Equations	15
2.3	Hyperbolic Equations	15
2.4	Overview	16
3	Finite differences	19
3.1	The grid-point method	19
3.2	Finite-difference schemes	20
3.3	Finite time differences	22
3.3.1	Two-level schemes	23
3.3.2	Three-level schemes	25
4	Numerical stability of the advection equation	27
4.1	The advection equation integrated using a leap-frog scheme	27
4.2	Initial and Boundary conditions	28
4.3	Stability of the the advection equation with leap-frog schemes	28
4.4	The advection equation with an Euler-forward scheme in time and a centred scheme in space	33
4.5	The upstream or upwind scheme	33
4.6	The advection equation with a scheme of fourth order	36
5	The numerical mode	41
5.1	The numerical mode of the three-level schemes	41
5.1.1	The computational initial condition	43
5.2	Suppression of the computational mode	43
5.2.1	Euler-forward or -backward at regular intervals	43
5.2.2	The Robert-Asselin filter	44
5.2.3	The Robert-Asselin-Williams filter	44

6 Accuracy of the numerical phase speed	47
6.1 Dispersion due to the spatial discretisation	47
6.2 Dispersion due to the time discretisation	47
6.3 Dispersion due to both spatial and temporal discretisation	49
7 The shallow-water equations	53
7.1 One-dimensional gravity waves with centred space differencing	53
7.2 2D shallow water equations	57
7.3 Two-dimensional gravity waves with centred space differencing	58
7.4 The shallow-water equations with leap-frog schemes	62
7.5 Boundary conditions	63
7.5.1 Closed boundary conditions	63
7.5.2 Open boundary conditions	63
7.5.3 Cyclic boundary conditions	66
7.6 Conservation of mass, energy and enstrophy	66
7.6.1 The shallow-water equations with non-linear advection terms	66
7.6.2 Discretisation	68
7.7 A shallow-water model	71
8 Diffusion and friction terms	73
8.1 Rayleigh friction	73
8.2 Laplacian friction	76
8.3 The advection-diffusion equation	79
9 Implicit and semi-implicit schemes	83
9.1 Implicit versus explicit schemes, a simple example	83
9.2 Semi-implicit schemes	84
9.2.1 The one-dimensional diffusion equation	85
9.2.2 Two-dimensional pure gravity waves	85
9.3 The semi-implicit method of Kwizak and Robert	86
10 The semi-Lagrangian technique	89
10.1 One-dimensional passive advection	89
10.2 Interpolation and Stability	91
11 Poisson and Laplace equations (elliptic)	93
11.1 Jacobi iteration	93
11.2 Gauss-Seidel iteration	93
11.3 Successive Over Relaxation (SOR)	94
12 Model Coordinates	97
12.1 Vertical ocean coordinates	97
12.1.1 Fixed height or depth coordinates	97

CONTENTS	5
12.2 Atmospheric vertical coordinates	98
12.2.1 Pressure coordinates	101
12.2.2 Atmospheric sigma coordinates	101
12.2.3 Hybrid coordinates	102
12.3 Horizontal coordinates	103
12.3.1 Finite differences	103
12.3.2 Finite elements	104
13 3D modelling	107
13.1 The Equations of motion	107
13.2 The tracer Equation	109
13.3 Discretised on a cartesian grid	109
13.4 Discretised on an orthogonal curvilinear grid	110
14 Spherical Harmonics	115
14.1 Spectral methods	115
14.2 Spherical harmonics	116
14.3 The spectral transform method	119
14.4 Application to the shallow water equations on a sphere	119
15 Practical computer exercises	121
15.1 Exercise 1 and 2	121
15.2 Theory	121
15.3 Technical aspects	121
15.4 Report	123
15.5 Theoretical exercises	124
15.6 Experimental exercises	124
15.7 1D Shallow water model exercises	128
15.8 2D Shallow water model exercises	129
A Exams at Stockholm University	137

Preface

The purpose of this book is to provide an introduction and an overview of numerical modelling of the ocean and atmosphere. Focus will be on numerical schemes for the most common equations in oceanography and meteorology as well as on the stability, precision and other basic numerical properties of these schemes. We will use as simple equations as possible that still capture the properties of the primitive equations used in the general circulation models. For simplicity the equations will often be referred to as the *hydrodynamic* equations since the numerical methods to be described here are valid for modelling both the ocean and the atmosphere. Due to the non-linearities of these equations, it is not possible to find analytical solutions. The equations are therefore instead solved numerically on a grid by discretisation, and the derivatives of the differential equations are replaced by finite differences. This is what constitutes a numerical model, which often is referred to as a general circulation model when it represents the 3D global circulation of the atmosphere or the ocean. These models are based on the Navier-Stokes equations (including the Coriolis effect) with atmospheric thermodynamic terms for various energy sources (radiation, latent heat) and a salt tracer equation for the ocean. A coupled atmospheric and oceanic general circulation model represents the core part of an Earth System climate model.

Kristofer Döös, professor of climate modelling at the Department of Meteorology, Stockholm University

Chapter 1

Introduction

1.1 What is a numerical model of the circulation of the atmosphere or the ocean?

A numerical model of the circulation of the atmosphere and/or the ocean is basically constituted by

- a) A grid covering the spatial domain in consideration
- b) Equations describing conservation of mass, heat, salt, moisture and momentum
- c) Open and/or solid boundaries

The schematic shown in Figure 1.1 illustrates this for a coupled ocean-atmosphere model with corresponding forcing and exchange between the two models.

1.2 Brief historical background

A short summary is here given of the historical background of the numerical modelling of the ocean and atmosphere.

[Bjerknes \(1904\)](#) was the first to formulate that it should be possible to predict and model the circulation of the atmosphere. For this he set up seven equations with seven unknown variables, which were

- Three equations of conservation of momentum for the three velocity components based on Newton's second law
- The continuity equation, i.e. the conservation of mass
- The equation of state for ideal gases
- The equation of conservation of energy based on the first law of thermodynamics
- A conservation equation for water mass in the atmosphere

These equations have become known as the "primitive equations", since they do not deal with any filtered quasi-geostrophic equation but go back to the original "primitive equations" set up by Bjerknes. These equations also require boundary conditions at the bottom and top of the atmosphere as well as initial state of

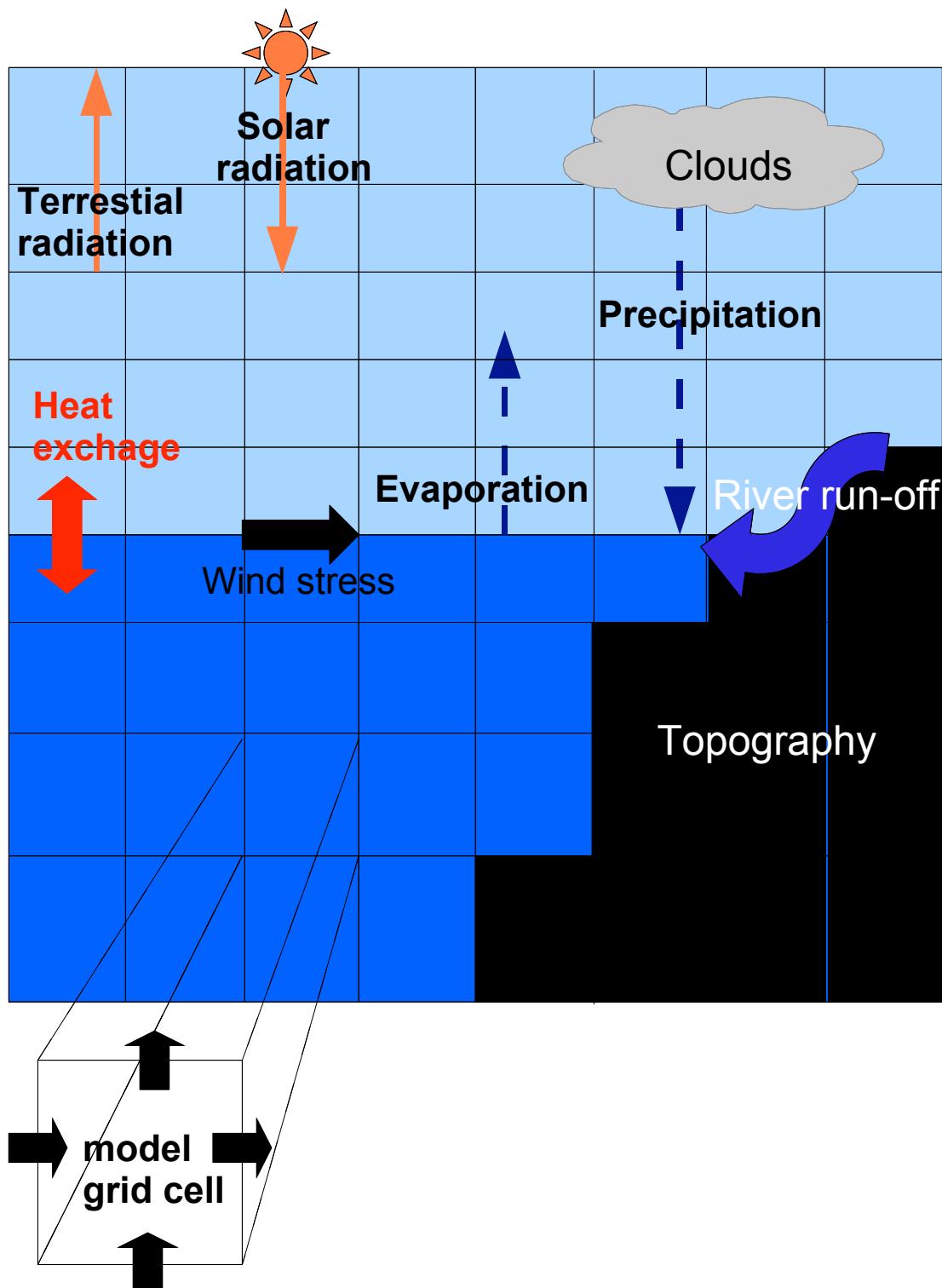


Figure 1.1: Schematic illustration of a numerical ocean-atmosphere circulation model

the atmosphere. The problem is that despite that we have as many equations as unknowns and we have the appropriate boundary and initial conditions, the equations can not be solved analytically.

The first attempt to solve the equations numerically was made by [Richardson \(1922\)](#) during the First World War despite that the digital computer had not yet been invented. In order to do so [Richardson \(1922\)](#) used finite differences of the equations Bjerknes had set up.

It was same L. F. Richardson that had been the first to propose a decade earlier the use of finite differences for approximating partial differential equations in [Richardson \(1911\)](#), which now was used for numerical weather prediction (NWP). He failed, however, to forecast realistically the weather for two reasons. 1) There were no computers available then and to do it manually would have required 64,000 persons to compute a 24 hour forecast in 24 hours. 2) Richardson was not aware of the numerical instabilities that arises from the finite differences, which gave Richardson's calculation erroneous results.

These numerical instabilities were independently investigated of [Richardson \(1922\)](#) work of only a few years later discovered by Courant, Friedrichs and Lewy ([Courant et al., 1928](#)), who had been working from a purely mathematically point of view to solve partial differential equations with finite difference, found independently of [Richardson \(1922\)](#) work that there are certain conditions in the choice of the numerical time step and grid size, which need to be respected for the n numerical solutions to be stable. This was later further developed by Charney, Fjørtoft and von Neumann ([Charney et al., 1950](#)), who were also the trio that made the first successful numerical forecast in the late 1940s, based on integration of the absolute vorticity conservation equation. This single equation approach of conservation of absolute vorticity following the motion of air columns instead of the full set of seven equations that [Bjerknes \(1904\)](#) came from Carl-Gustaf Rossby, who was also the driving force behind the first real time NWP made in two runs 23 - 24 March 1954 in Stockholm by Harold Bedient and Bo Döös ([Döös and Eaton, 1957](#); [Wiin-Nielsen, 1991](#); [Persson, 2005a](#)).

An ocean storm surge model based on the shallow water equations was set up by [Hansen \(1956\)](#) using finite differences for the three shallow water equations including the non-linear advection terms. Here, the main focus was the fast barotropic gravity waves instead of filtering them out as in the NWP models. Also [Fischer \(1959\)](#) constructed a numerical shallow water model for the same region but with the finite difference schemes well presented, which [Hansen \(1956\)](#) failed to provide.

The return to the full primitive equations for NWP models, which [Richardson \(1922\)](#) had used in the very beginning, was inevitable since the quasi-geostrophic equations, although very useful for understanding the large-scale extratropical dynamics of the atmosphere, were not accurate enough to allow continued progress in NWP. A complete Atmospheric General Circulation Model (AGCM) based the primitive equations was developed by [Smagorinsky \(1963\)](#) at the Geophysical Fluid Dynamics Laboratory (GFDL) in Princeton, where a great deal of the model development was made. The first Ocean General Circulation Model (OGCM) was developed only a few years later by [Bryan and Cox \(1967\)](#) also at GFDL. Syukuro Manabe and Kirk Bryan combined their models and made the first coupled Ocean-Atmosphere General Circulation Model (AOGCM) in [Manabe and Bryan \(1969\)](#), which they used for climate studies. The ocean and atmospheric GCMs have since then increased in numbers over the world and have improved in many aspects such as resolution with more powerful computers, better parameterisations, more observations to feed the models with. The basic numerics have also improved but the fundamental properties of the finite differences and the numerics remain. This is why, the focus in this book will be on the limitations of the

numerical schemes that to a large extent were discovered during the 20th century.

For more comprehensive historical descriptions there are several studies focusing on the NWP such as Persson (2005a,b,c) and Wiin-Nielsen (1991).

Chapter 2

Partial Differential Equations

Before going into the details of numerical procedures let us first classify the most usual types of Partial Differential Equations (PDEs) that occur in meteorology and oceanography. Partial differential equations are of vast importance in applied mathematics and engineering since so many real physical situations can be modelled by them. Second-order linear PDEs of the type needed for modelling the ocean and atmosphere circulation can be classified into three categories - hyperbolic, parabolic and elliptic. These partial differential equations are general linear homogeneous equations of second order:

$$a \frac{\partial^2 u}{\partial x^2} + b \frac{\partial^2 u}{\partial x \partial y} + c \frac{\partial^2 u}{\partial y^2} + d \frac{\partial u}{\partial x} + e \frac{\partial u}{\partial y} + f u + g = 0. \quad (2.1)$$

As you may note, this expression bears a resemblance to the equation for a conic section

$$ax^2 + bxy + cy^2 + dx + ey + f = 0, \quad (2.2)$$

where a, b, c, d, e and f are constants, Equation (2.2) represents an ellipse, parabola or a hyperbola depending whether $b^2 - 4ac$ is negative, equal to zero or positive, respectively. This indicates that one can classify the PDE according to Table 2.1.

Each type of system is associated with significantly different characteristic behaviour, and the solution scheme for each type of equation can also differ. All three classes of PDEs are represented among the most common equations in hydrodynamics and require the specification of different kinds of boundary conditions. We will now provide some examples of these equations.

Table 2.1: *Classification of the three PDE types and their different kinds of boundary conditions*

PDE	$b^2 - 4ac$	Boundary conditions
1. elliptic	$b^2 - 4ac < 0$	Dirichlet/Neumann/Robin
2. parabolic	$b^2 - 4ac = 0$	One initial+some boundary condition
3. hyperbolic	$b^2 - 4ac > 0$	Cauchy+some boundary condition

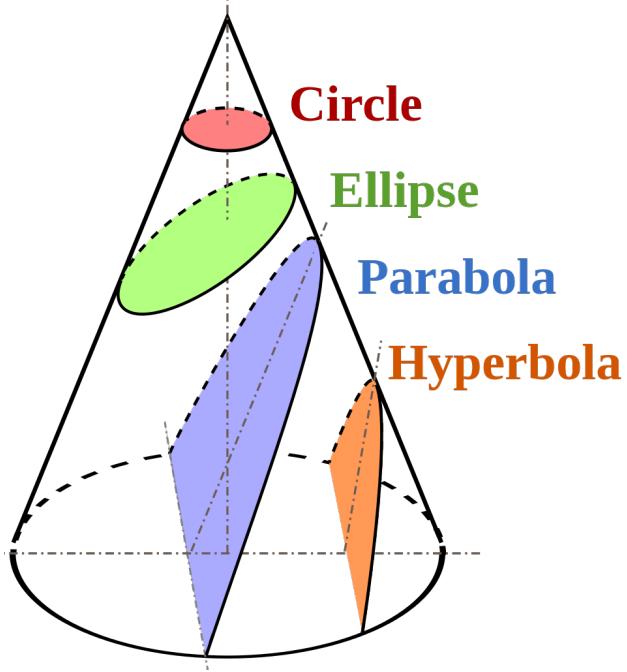


Figure 2.1: A diagram showing conic sections, including a circle, an ellipse, a parabola and a hyperbola. Note that these geometrical representations of Equation (2.2) are not solutions to the PDE Equation (2.1).

2.1 Elliptic Equations

The Laplace and Poisson equations are classical examples of elliptic equations, representing e.g. the steady-state temperature in a plate and the relationship between the stream function and vorticity.

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \text{ (or } g(u, x, y)) . \quad (2.3)$$

In conformity with Equation (2.1), it is found that $a = 1$, $b = 0$, $c = 1$ and $d = e = f = 0$, which leads to $b^2 - 4ac = -4 < 0$, hence the PDE must be elliptic.

Another example of an elliptic equation in hydrodynamics can be found from the simplest equations of motion in the atmosphere and the ocean, *viz.* the linearised shallow-water equations without friction:

$$\frac{\partial u}{\partial t} - fv = -g \frac{\partial h}{\partial x}, \quad (2.4a)$$

$$\frac{\partial v}{\partial t} + fu = -g \frac{\partial h}{\partial y}, \quad (2.4b)$$

$$\frac{\partial h}{\partial t} + H \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) = 0. \quad (2.4c)$$

Note that the independent variables x and y in Equation (2.1) do not have anything with x and y in these equations and must therefore be replaced in order to calculate $b^2 - 4ac$. From Equations (2.4) it can be

deduced that

$$\frac{\partial}{\partial t} \left[\frac{\partial^2 h}{\partial t^2} + f^2 - gH \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) h \right] = 0. \quad (2.5)$$

If we integrate equation in time we obtain

$$\frac{\partial^2 h}{\partial t^2} + f^2 h - gH \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) h = \xi(x, y), \quad (2.6)$$

where $\xi(x, y)$ is an integration constant independent of t . When the problem is stationary, the geostrophic relationship is obtained:

$$f^2 h - gH \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) h = \xi(x, y), \quad (2.7)$$

In analogy with Equation (2.1), we find that $a = -gH$, $b = 0$ and $c = -gH$ which leads to $b^2 - 4ac = -4g^2 H^2 < 0$, and hence this PDE must be elliptic.

There are three different types of possible boundary conditions for this class of PDEs:

- a) u specified on the boundary (Dirichlet),
- b) $\frac{\partial u}{\partial \vec{n}}$ specified on the boundary (Neumann), where the vector \vec{n} is perpendicular to the boundary
- c) $au + \frac{\partial u}{\partial \vec{n}}$ specified on the boundary (Cauchy),

2.2 Parabolic Equations

An example of a parabolic equation is the diffusion equation for the dependent variable T , corresponding to e.g. temperature, salinity, humidity or any passive tracer:

$$\frac{\partial T}{\partial t} = K \frac{\partial^2 T}{\partial x^2}, \quad \text{where } K > 0. \quad (2.8)$$

Based on Equation (2.1), $a = K$, $b = 0$ and $c = 0$, leading to $b^2 - 4ac = 0$, and hence the PDE is parabolic.

Assume that $T(x, t)$ is the temperature distribution along the x -axis as a function of time. To solve the equation over the interval $0 \leq x \leq L$ one must specify an initial condition $T(x, 0)$ on $0 \leq x \leq L$. The boundary conditions $T(0, t)$ and $T(L, t)$ must also be specified during the whole time period under consideration.

2.3 Hyperbolic Equations

This class of PDEs describes wave motion. Typical examples are the vibrating string and gravity waves in the ocean or atmosphere.

$$\frac{\partial^2 u}{\partial t^2} = c_0^2 \frac{\partial^2 u}{\partial x^2}. \quad (2.9)$$

The following first-order PDE, known as the advection equation, can also be classified as hyperbolic, since its solutions satisfy the wave equation:

$$\frac{\partial u}{\partial t} = -c_0 \frac{\partial u}{\partial x}. \quad (2.10)$$

By first differentiating the advection equation with respect to t and also with respect to x , it is possible to eliminate u_{tx} between the two equations and we obtain the wave equation. In analogy with Equation (2.1) it is found that $b^2 - 4ac = 4c^2_0 > 0$ and the advection equation is hence a hyperbolic PDE.

In order to obtain unique solutions for $0 \leq x \leq L$ we need boundary conditions at $x = 0$ and $x = L$, viz. the ends of the spatial domain, and two initial conditions.

- $u(0, t)$ and $u(L, t)$ specified on the boundaries $x = 0$ and $x = L$
- $\frac{\partial u(0,t)}{\partial x}$ and $\frac{\partial u(L,t)}{\partial x}$ as free or open boundaries at $x = 0$ and $x = L$
- $u(x, 0)$ specified initially on $0 \leq x \leq L$
- $\frac{\partial u(x,0)}{\partial t}$ specified initially on $0 \leq x \leq L$

2.4 Overview

Having specified these three types of PDEs, it is important to underline that the behaviour of the solutions, the proper initial and/or boundary conditions, and the numerical methods that can be used to find the solutions depend essentially on the type of PDE that we are dealing with. Although non-linear multidimensional PDEs in general cannot be reduced to these canonical forms, we need to study these prototypes in order to develop an understanding of their properties, and then apply similar methods to the more complicated equations governing the motion of the ocean and atmosphere.

Exercises:

- a) Solve the following equation by separation of variables:

$$\frac{\partial u}{\partial t} = -c \frac{\partial u}{\partial x}, \quad (2.11)$$

with the initial condition

$$u(x, 0) = -Ae^{ikx}. \quad (2.12)$$

- b) Show that the advection equation

$$\frac{\partial u}{\partial t} = -c \frac{\partial u}{\partial x} \quad (2.13)$$

has the general solution $u = f(x - ct)$, where f is an arbitrary continuously differentiable solution. Interpret the equation geometrically in the xt -plane. Solve the equation with the following initial condition: $u(x, 0) = g(x)$.

- c) Derive the wave equation from the system

$$\frac{\partial \vec{V}}{\partial t} = -g \nabla h, \quad (2.14)$$

$$\frac{\partial h}{\partial t} = -H \nabla \left(h \vec{V} \right), \quad (2.15)$$

where g and H are constants and \vec{V} the horizontal velocity vector (u, v) .

Chapter 3

Finite differences

3.1 The grid-point method

Let us study a function with one independent variable:

$$u = u(x).$$

Suppose we have an interval L , which is partitioned by $N + 1$ equally spaced grid points (including the two at the endpoints of the interval). The grid length is then $\Delta x = L/N$ and the grid points are located at $x_j = j\Delta x$ where $j = 0, 1, 2, \dots, N$ are integers. Let the value of u at x_j be represented by u_j .

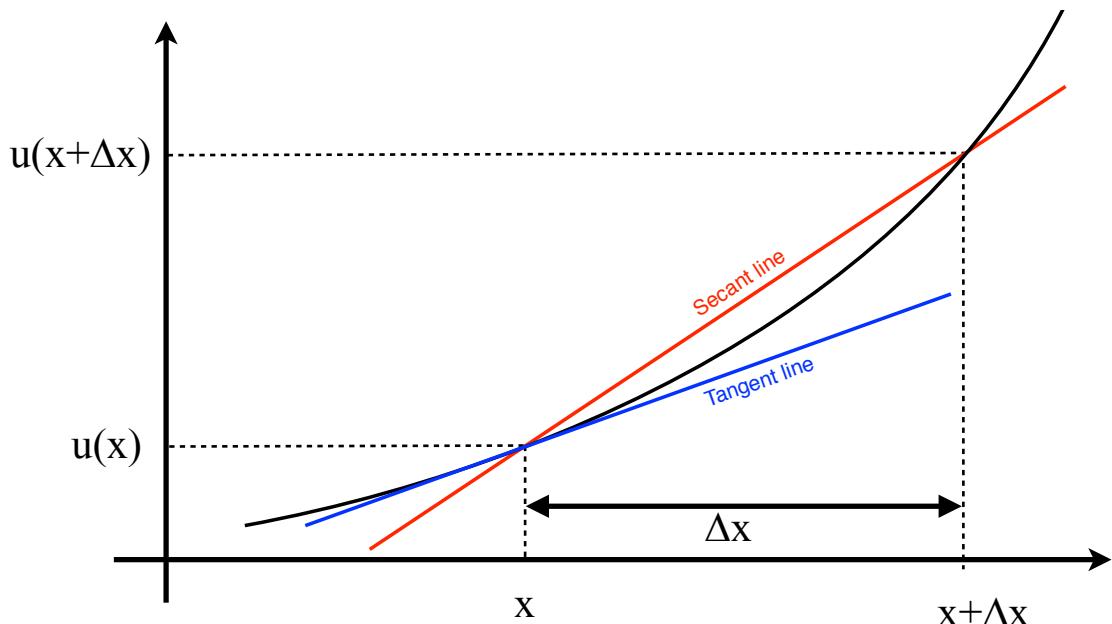


Figure 3.1: The red secant approaches the blue tangent, which is the derivative of u at the point x when $\Delta x \rightarrow 0$. A finite difference is when $\Delta x \neq 0$.

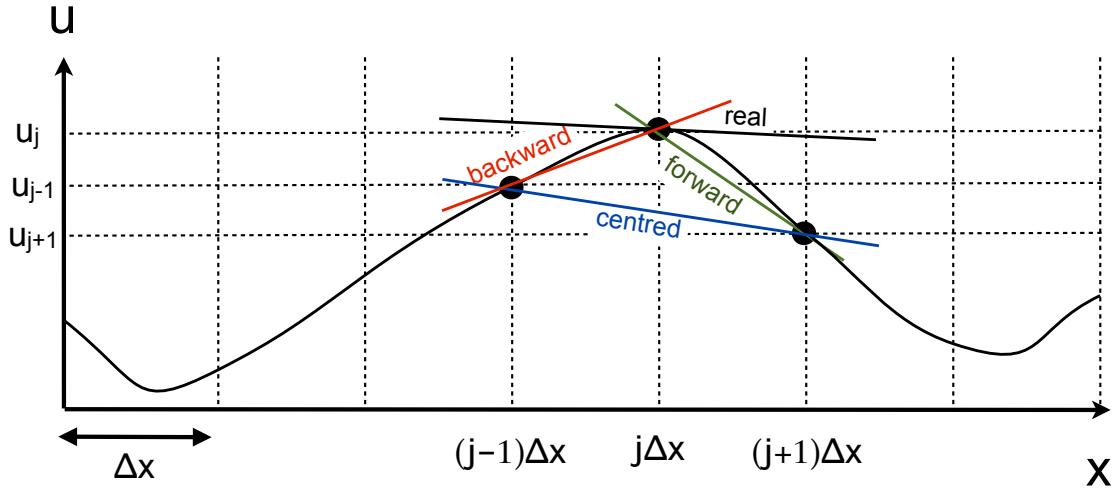


Figure 3.2: The backward, forward and centred finite differences of a function $u(x)$, defined at the grid points $x = j\Delta x$ so that $u_j = u_j(j\Delta x)$, where Δx is the grid length and $j = 0, 1, 2, \dots$ are integers.

3.2 Finite-difference schemes

The formal mathematical definition of the derivative of a function $u(x)$ is

$$\frac{du}{dx} = \lim_{\Delta x \rightarrow 0} \frac{u(x + \Delta x) - u(x)}{\Delta x}, \quad (3.1)$$

which is illustrated in Figure 3.1. The difference between a derivative and a finite difference is that in the latter case Δx will not tend to zero.

We will now derive expressions which can be used to give an approximate value of a derivative at a grid point in terms of grid-point values. The finite differences can be constructed between values of u_j over the grid length Δx . As illustrated in Figure 3.2, the first derivative of $u(x)$ can be approximated in three ways:

- forward difference: $(\frac{du}{dx})_j \approx \frac{u_{j+1} - u_j}{\Delta x}$
- centred difference: $(\frac{du}{dx})_j \approx \frac{u_{j+1} - u_{j-1}}{2\Delta x}$
- backward difference: $(\frac{du}{dx})_j \approx \frac{u_j - u_{j-1}}{\Delta x}$

These various finite difference schemes introduce errors that can be estimated by deriving the finite differences in a more rigorous way using a Taylor expansion. The Taylor series for $f(y)$ about $y = a$ is

$$f(y) = f(a) + (y - a) f'(a) + \frac{1}{2} (y - a)^2 f''(a) + \dots + \frac{1}{n!} (y - a)^n f^{(n)}(a). \quad (3.2)$$

Substituting $f(y)$ by $u(x)$, a by x_j and y by x_{j+1} , we obtain the Taylor expansion of the function $u(x)$ at

the point $j + 1$.

$$\begin{aligned} u_{j+1} = u_j + \Delta x \left(\frac{du}{dx} \right)_j + \frac{1}{2} (\Delta x)^2 \left(\frac{d^2 u}{dx^2} \right)_j + \frac{1}{6} (\Delta x)^3 \left(\frac{d^3 u}{dx^3} \right)_j + \\ \frac{1}{24} (\Delta x)^4 \left(\frac{d^4 u}{dx^4} \right)_j + \frac{1}{120} (\Delta x)^5 \left(\frac{d^5 u}{dx^5} \right)_j + \dots \quad (3.3) \end{aligned}$$

The forward difference can now be expressed as

$$\begin{aligned} \frac{u_{j+1} - u_j}{\Delta x} = \left(\frac{du}{dx} \right)_j + \frac{1}{2} (\Delta x) \left(\frac{d^2 u}{dx^2} \right)_j + \frac{1}{6} (\Delta x)^2 \left(\frac{d^3 u}{dx^3} \right)_j + \\ \frac{1}{24} (\Delta x)^3 \left(\frac{d^4 u}{dx^4} \right)_j + \frac{1}{120} (\Delta x)^4 \left(\frac{d^5 u}{dx^5} \right)_j + \dots \quad (3.4) \end{aligned}$$

The difference between this expression and the approximated derivative $\left(\frac{du}{dx} \right)_j$ is

$$\varepsilon = \frac{1}{2} (\Delta x) \left(\frac{d^2 u}{dx^2} \right)_j + \frac{1}{6} (\Delta x)^2 \left(\frac{d^3 u}{dx^3} \right)_j + \frac{1}{24} (\Delta x)^3 \left(\frac{d^4 u}{dx^4} \right)_j + \frac{1}{120} (\Delta x)^4 \left(\frac{d^5 u}{dx^5} \right)_j + \dots, \quad (3.5)$$

which is denoted the truncation error associated with approximation of the derivative. The terms that have been truncated, i.e. "cut-off", are represented by ε . Hence we have an accuracy of first order with

$$\varepsilon = O(\Delta x), \quad (3.6)$$

which is the lowest order of accuracy that is acceptable.

The accuracy of the centred difference can be obtained from Equation (3.3) and a Taylor expansion in x of the function $u(x)$ at the point $j - 1$

$$\begin{aligned} u_{j-1} = u_j - \Delta x \left(\frac{du}{dx} \right)_j + \frac{1}{2} (\Delta x)^2 \left(\frac{d^2 u}{dx^2} \right)_j - \frac{1}{6} (\Delta x)^3 \left(\frac{d^3 u}{dx^3} \right)_j + \\ \frac{1}{24} (\Delta x)^4 \left(\frac{d^4 u}{dx^4} \right)_j - \frac{1}{120} (\Delta x)^5 \left(\frac{d^5 u}{dx^5} \right)_j + \dots \quad (3.7) \end{aligned}$$

so that

$$\frac{u_{j+1} - u_{j-1}}{2\Delta x} \rightarrow \left(\frac{du}{dx} \right)_j + \frac{1}{6} (\Delta x)^2 \left(\frac{d^3 u}{dx^3} \right)_j + \frac{1}{120} (\Delta x)^4 \left(\frac{d^5 u}{dx^5} \right)_j + \dots$$

Here the truncation error is of second order:

$$\varepsilon = \frac{1}{6} (\Delta x)^2 \left(\frac{d^3 u}{dx^3} \right)_j + \dots = O((\Delta x)^2).$$

A scheme with fourth order accuracy can be obtained if we undertake a Taylor expansion for of u_{j+2} :

$$u_{j+2} = u_j + 2\Delta x \left(\frac{du}{dx} \right)_j + \frac{1}{2} (2\Delta x)^2 \left(\frac{d^2 u}{dx^2} \right)_j + \frac{1}{6} (2\Delta x)^3 \left(\frac{d^3 u}{dx^3} \right)_j + \frac{1}{24} (2\Delta x)^4 \left(\frac{d^4 u}{dx^4} \right)_j + \frac{1}{120} (2\Delta x)^5 \left(\frac{d^5 u}{dx^5} \right)_j + \dots \quad (3.8)$$

and

$$u_{j-2} = u_j - 2\Delta x \left(\frac{du}{dx} \right)_j + \frac{1}{2} (2\Delta x)^2 \left(\frac{d^2 u}{dx^2} \right)_j - \frac{1}{6} (2\Delta x)^3 \left(\frac{d^3 u}{dx^3} \right)_j + \frac{1}{24} (2\Delta x)^4 \left(\frac{d^4 u}{dx^4} \right)_j - \frac{1}{120} (2\Delta x)^5 \left(\frac{d^5 u}{dx^5} \right)_j + \dots \quad (3.9)$$

so that

$$\frac{u_{j+2} - u_{j-2}}{4\Delta x} \rightarrow \left(\frac{du}{dx} \right) + \frac{4}{6} (\Delta x)^2 \left(\frac{d^3 u}{dx^3} \right)_j + \frac{16}{120} (\Delta x)^4 \left(\frac{d^5 u}{dx^5} \right)_j + \dots$$

This scheme is, as the previous centred scheme, accurate to second order, and if we combine the two centred schemes so that

$$\frac{4}{3} \frac{u_{j+1} - u_{j-1}}{2\Delta x} - \frac{1}{3} \frac{u_{j+2} - u_{j-2}}{4\Delta x} \rightarrow \left(\frac{du}{dx} \right) - \frac{1}{30} (\Delta x)^4 \left(\frac{d^5 u}{dx^5} \right)_j + \dots \quad (3.10)$$

we find an accuracy of fourth order, i.e. $\varepsilon = O((\Delta x)^4)$.

3.3 Finite time differences

The time-derivative schemes that are used for PDEs are relatively simple, usually of second- and sometimes even only of first-order accuracy. There are several reasons for this. First, it is a general experience that schemes constructed to have a high order of accuracy are mostly not very successful when solving PDEs. This is in contrast to the experience with ordinary differential equations, where very accurate schemes, such as the Runge-Kutta method, are extremely successful. There is a basic reason for this difference. With an ordinary differential equation of first order, the equation and a single initial condition is all that is required for an exact solution. Thus, the error of the numerical solution is entirely due the degree of inadequacy of the scheme. With a PDE, the error of the numerical solution arises from both by the inadequacy of the scheme and by insufficient information about the initial conditions, which only are known only at discrete grid points. Thus, an increase in accuracy of the scheme applied improves only one of these two components, and the result is not too impressive.

Another reason for not requiring a scheme of high accuracy for approximations of the time-derivative terms is that, in order to meet a stability requirement of the type to be discussed in the next chapter, it is usually necessary to choose a time step significantly smaller than that required for adequate accuracy. Once a time step has been specified, other errors, e.g. in the spatial differencing, are much greater than those due to the time differencing. Thus, computational effort is better spent in reducing these errors, and not in increasing the accuracy of the time-differencing schemes. This, of course, does not mean that it is not

necessary to carefully consider the properties of various possible time-differencing schemes. Accuracy is only one important consideration when choosing a scheme.

To define some schemes, we consider a general first order differential equation:

$$\frac{du}{dt} = f(u, t), \quad (3.11)$$

where $u = u(t)$. The independent variable t is the time. f is a function of u and t , corresponding e.g. to the advection equation where $f = -c\frac{\partial u}{\partial x}$.

In order to discretise the equation we divide the time axis into segments of equal length Δt . The approximated value of $u(t)$ at time $t = n\Delta t$ is denoted u^n . In order to compute u^{n+1} we need to know at least u^n and often also u^{n-1} . A number of time differentiation schemes are available

3.3.1 Two-level schemes

These are schemes that use two different time levels: n and $n + 1$ so that the time integration yields

$$u^{n+1} = u^n + \int_{n\Delta t}^{(n+1)\Delta t} f(u, t) dt. \quad (3.12)$$

The problem now is that f only exists as discrete values f^n and f^{n+1} at times $n\Delta t$ and $(n + 1)\Delta t$ respectively.

Euler or forward scheme

This is defined as

$$u^{n+1} = u^n + \Delta t f^n. \quad (3.13)$$

Here the truncation error is $O(\Delta t)$, i.e. the scheme is accurate to first order. It is said to be *uncentred*, since the time derivative is at time level $n + 1/2$ and the function at time level n . In general, uncentred schemes are accurate to first order and centred schemes to second order.

Backward scheme

This is defined as

$$u^{n+1} = u^n + \Delta t f^{n+1}. \quad (3.14)$$

The backward scheme is uncentred in time and is accurate to $O(\Delta t)$. If, as here, a value of f is taken at time level $n + 1$ and f depends on u , i.e. u^{n+1} , the scheme is said to be *implicit*. For an ordinary differential equation, it may be straightforward matter to solve for u^{n+1} .

For a PDE it will, however, require solving a set of simultaneous equations, with one equation for each of the grid points of the computation region. If no value of f depends on u^{n+1} on the right hand side of the equation above, the scheme is said to be *explicit*.

In the very simple cases when f only depends on t , such as e.g. $du/dt = -\gamma u$, the discretised equation becomes $u^{n+1} = u^n + \Delta t (-\gamma u^{n+1})$, which can be rearranged as $u^{n+1} = u^n/(1 + \gamma\Delta t)$ so that there

are no terms at time level $n + 1$ in the right-hand membrum. In this case the discretised equation can be integrated despite being implicit.

Crank-Nicolson scheme

The Crank-Nicolson scheme is based on the trapezoidal rule. If we approximate f by an average between time levels n and $n + 1$ we obtain

$$u^{n+1} = u^n + \frac{1}{2}\Delta t (f^n + f^{n+1}). \quad (3.15)$$

This scheme is *implicit*, since it requires information from the future ($n + 1$). The truncation error can be found as before from the Taylor series in Equation (3.2). Substituting $f(y)$ by $u(t)$, a by t^n and y by t^{n+1} , we obtain the Taylor expansion of the function $u(t)$ at the time level $n + 1$:

$$u^{n+1} = u^n + \Delta t \left(\frac{du}{dt} \right)^n + \frac{1}{2} (\Delta t)^2 \left(\frac{d^2 u}{dt^2} \right)^n + \frac{1}{6} (\Delta t)^3 \left(\frac{d^3 u}{dt^3} \right)^n + \frac{1}{24} (\Delta t)^4 \left(\frac{d^4 u}{dt^4} \right)^n + \dots \quad (3.16)$$

Substituting $f(y)$ by $u(t)$, a by t^{n+1} and y by t^n , we obtain the Taylor expansion of the function $u(t)$ in the time level n .

$$\begin{aligned} u^n = u^{n+1} - \Delta t \left(\frac{du}{dt} \right)^{n+1} + \frac{1}{2} (\Delta t)^2 \left(\frac{d^2 u}{dt^2} \right)^{n+1} - \frac{1}{6} (\Delta t)^3 \left(\frac{d^3 u}{dt^3} \right)^{n+1} + \\ \frac{1}{24} (\Delta t)^4 \left(\frac{d^4 u}{dt^4} \right)^{n+1} + \dots \end{aligned} \quad (3.17)$$

Subtracting Equation (3.16) from Equation (3.17) and dividing by 2 we obtain

$$u^{n+1} = u^n + \frac{1}{2}\Delta t \left[\left(\frac{du}{dt} \right)^n + \left(\frac{du}{dt} \right)^{n+1} \right] + \frac{1}{24} (\Delta t)^3 \left[\left(\frac{d^3 u}{dt^3} \right)^n + \left(\frac{d^3 u}{dt^3} \right)^{n+1} \right] \dots, \quad (3.18)$$

which can also be expressed as the average time derivative between time level n and $n + 1$

$$\frac{1}{2} \left[\left(\frac{du}{dt} \right)^n + \left(\frac{du}{dt} \right)^{n+1} \right] = \frac{u^{n+1} - u^n}{\Delta t} + \epsilon, \quad (3.19)$$

where

$$\epsilon = \frac{1}{24} (\Delta t)^2 \left[\left(\frac{d^3 u}{dt^3} \right)^n + \left(\frac{d^3 u}{dt^3} \right)^{n+1} \right] + \dots = O[(\Delta t)^2]. \quad (3.20)$$

The truncation error ϵ is therefore of second order for the Crank-Nicolson scheme.

Matsuno or Euler-backward scheme

To increase the accuracy compared to the Euler-forward and -backward schemes we can construct iterative schemes such as the Matsuno scheme that is initiated by an Euler-forward time step.

$$u_*^{n+1} = u^n + \Delta t f^n. \quad (3.21)$$

In this case the value of the obtained u_*^{n+1} is used for an approximation of f^{n+1} , which hereafter is used to make a backward step to yield a final u^{n+1} .

$$u^{n+1} = u^n + \Delta t f_*^{n+1}, \quad (3.22)$$

where

$$f_*^{n+1} = f(u_*^{n+1}, (n+1)\Delta t). \quad (3.23)$$

This scheme is *explicit* and of accuracy $O(\Delta t)$.

Heun scheme

This is similar to the Matsuno scheme and is explicit but of second-order accuracy, since the second step is made using the Crank-Nicolson scheme:

$$u_*^{n+1} = u^n + \Delta t f^n, \quad (3.24)$$

$$u^{n+1} = u^n + \frac{1}{2}\Delta t (f^n + f_*^{n+1}). \quad (3.25)$$

3.3.2 Three-level schemes

These schemes use the time at three levels and the time integration becomes

$$u^{n+1} = u^{n-1} + \int_{(n-1)\Delta t}^{(n+1)\Delta t} f(u, t) dt. \quad (3.26)$$

The simplest three-level scheme is to assign f a constant value equal to that at the middle of the time interval of length $2\Delta t$, which yields a leapfrog scheme:

$$u^{n+1} = u^{n-1} + 2\Delta t f^n, \quad (3.27)$$

and is of accuracy order $(\Delta t)^2$. This is the most widely used scheme in atmospheric and oceanic models.

Exercises:

- a) Determine the order of accuracy of the centered discretisation of the advection scheme:

$$\frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t} + c \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} = 0.$$

- b) Determine the order of accuracy of the Euler forward scheme for the heat equation:

$$\frac{\partial u}{\partial t} = A \frac{\partial^2 u}{\partial x^2},$$

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = A \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{(\Delta x)^2},$$

where $A > 0$.

Chapter 4

Numerical stability of the advection equation

In this chapter we will study partial differential equations (PDEs) with one dependent and two independent variables. More specifically we shall consider various simplified forms of the advection equation, describing the advection of a dependent variable. In practice, this has proved to be the most important part of the hydrodynamic equations for the atmosphere and the ocean.

We will use the advection equation to investigate what is required of the numerical schemes to yield stable solutions to the PDE, i.e. solutions such that small perturbations do not grow in time, but rather decrease.

4.1 The advection equation integrated using a leap-frog scheme

This equation is

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0,$$

where $u = u(x, t)$ and c is the phase speed. Analytical solutions are of the form $u(x, t) = u_0 e^{ik(x-ct)}$, where $c \equiv \omega/k$.

Let us now consider one among many possible discretisations of this equation by using a centred difference both in time and in space:

$$\frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t} + c \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} = 0. \quad (4.1)$$

When used in model applications, this needs to be reformulated:

$$u_j^{n+1} = u_j^{n-1} - \frac{c\Delta t}{\Delta x} (u_{j+1}^n - u_{j-1}^n). \quad (4.2)$$

Note that all the grid points for the leap-frog scheme, which are shown by the green dots in Figure 4.1, are located around the "here and now" point j, n shown as the black dot in the graph.

4.2 Initial and Boundary conditions

We also need initial and boundary conditions in order to integrate Equation (4.2) forward in time on a grid such as that shown in Figure 4.1. The initial condition is that all the u_j^0 values (red dots) have to be prescribed, i.e. the variable u has to have specified values at the time step $n = 0$. Furthermore we need to be able to integrate a first time step, which is not possible with a three-level time scheme. We therefore use an Euler-forward scheme for the first time step and hereafter proceed using the leap-frog scheme for the rest of the integration.

The boundary conditions imply that values for all the u_0^n and u_{JX}^n (blue dots in Figure 4.1) have to be prescribed, i.e. the variable u has to have values at all time steps on the two walls located at $j = 0$ and $j = JX$, where JX is the total number of grid cells.

In the case of a periodic domain, i.e., where e.g. the values on the eastern boundary equal the values on the western boundary, which is the case for global models of the Earth, we can use cyclic boundary conditions. For practical purposes this boundary can e.g. be located at the Greenwich meridian where the longitude can be expressed as 0° or 360° . When u is computed in e.g. a Fortran code at the eastern ($j = 0$) and western ($j = JX$) boundaries one will need u -values for " $j = -1$ " and " $j = JX + 1$ ". This can easily be coded by introducing in the model code's j-loops $jm = j - 1$ and when $jm = -1$ then you set it to $jm = JX - 1$. The same procedure is used for the eastern boundary with $jp = j + 1$ and when $jp = JX + 1$, it is replaced by $jp = 1$.

4.3 Stability of the advection equation with leap-frog schemes

In order to study the stability we will use what is known as the von Neumann method, which in fact was first briefly introduced by [Crank and Nicolson \(1947\)](#) and later more rigorously by [Charney et al. \(1950\)](#), in which study von Neumann was the last author. This method is generally not possible to use for non-linear equations and one is therefore limited to study the linearised version of the equations in a numerical model. In general a solution of a linear equation can be expressed as a Fourier series, where each Fourier component is a solution. Thus, we can test the stability using solely one Fourier component of the form

$$u_j^n = u_0 e^{ik(j\Delta x - C_D n \Delta t)}.$$

Note that the phase speed C_D , with the index D representing finite differencing, which is an approximation of the phase speed c occurring in the differential equation. It is this C_D , obtained as a solution of Equation (4.1), which here will be investigated. We note that

$$u_j^{n+1} = u_0 e^{ik[j\Delta x - C_D(n+1)\Delta t]} = u_j^n e^{-ikC_D\Delta t} = u_j^n \lambda,$$

where $\lambda \equiv e^{-ikC_D\Delta t}$. We obtain similarly, for $n - 1$:

$$u_j^{n-1} = u_j^n \lambda^{-1}. \tag{4.3}$$

These results can be generalised as

$$u_j^{n+m} = u_j^n \lambda^m \tag{4.4}$$

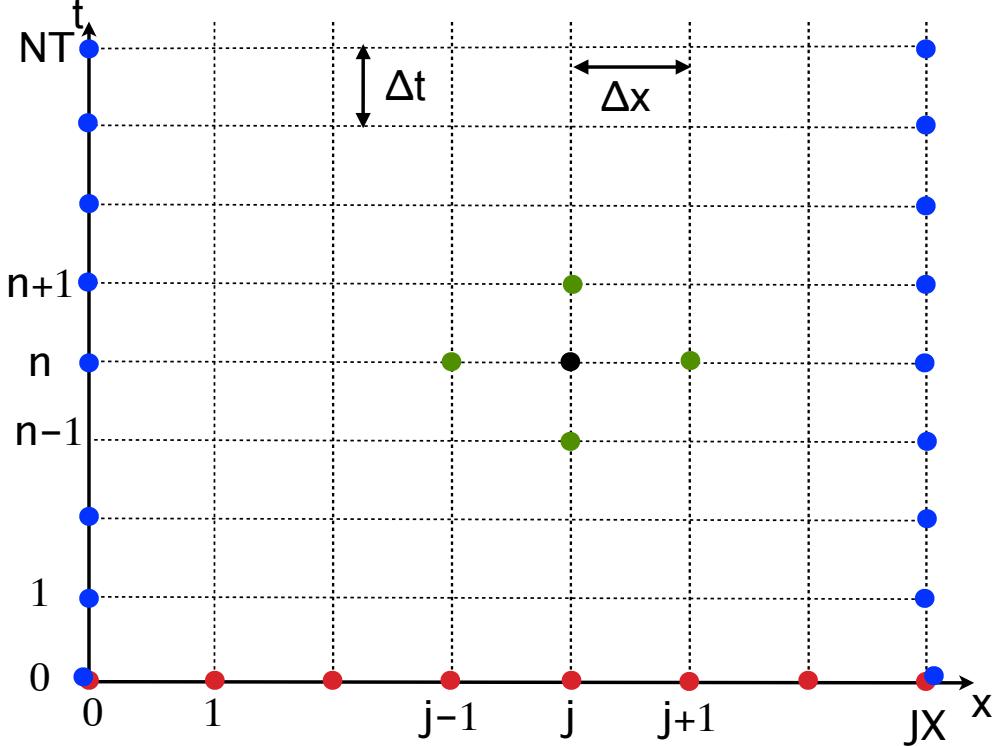


Figure 4.1: A finite-difference grid in time and space. The grid points used by the leap-frog scheme are shown as green dots. The red dots are the necessary initial condition points and the blue dots illustrate the two boundaries of the domain, where values of the independent variable must be prescribed.

and

$$u_j^n = \lambda^n u_0 e^{ikj\Delta x}. \quad (4.5)$$

From this we can deduce that if $|u|$ is not going to "blow up" when integrating in time, one needs to require that

$$|\lambda| = \left| e^{-ikC_D\Delta t} \right| \leq 1 \quad (4.6)$$

and, reversely, if $|\lambda| > 1$ the solution is unstable and "blows up". For the condition to be fulfilled, C_D must be real. This technique, based on examining the amplification factor λ , is called the von Neumann method.

We also need expressions for the Fourier components of the spatial derivatives:

$$u_{j+1}^n = u_0 e^{ik[(j+1)\Delta x - C_D n \Delta t]} = e^{ik\Delta x} u_j^n, \quad (4.7)$$

$$u_{j-1}^n = u_0 e^{ik[(j-1)\Delta x - C_D n \Delta t]} = e^{-ik\Delta x} u_j^n. \quad (4.8)$$

Let us now return to Equation (4.1) and introduce λ :

$$\lambda^2 + 2i \frac{c\Delta t}{\Delta x} \sin(k\Delta x) \lambda - 1 = 0. \quad (4.9)$$

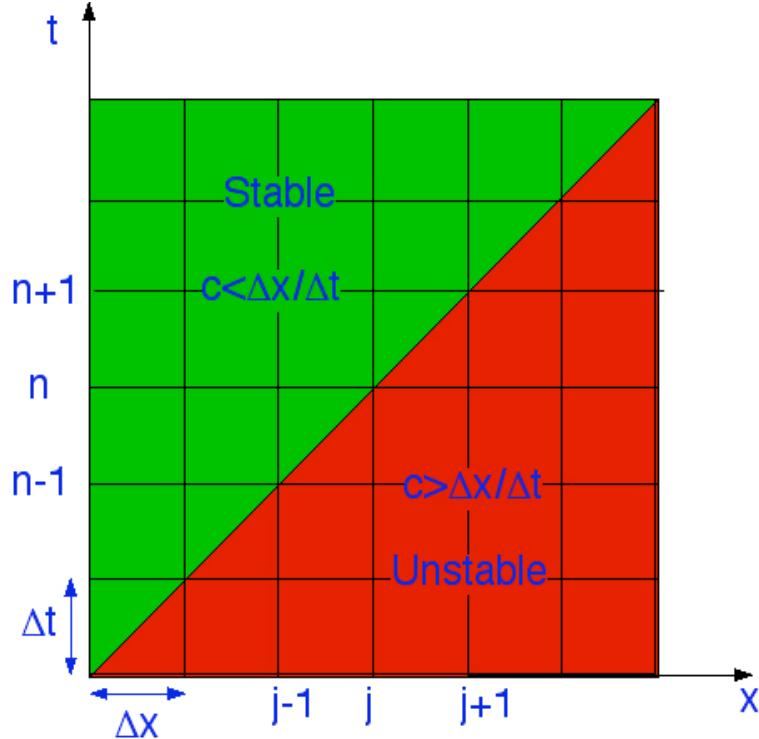


Figure 4.2: The Courant-Friedrichs-Lowy (CFL) stability criterion for centred schemes in time and space.

Since $e^{i\alpha} - e^{-i\alpha} = 2i \sin \alpha$ and $x^2 + \alpha x + \beta = 0 \Rightarrow x = -\alpha/2 \pm \sqrt{\alpha^2/4 - \beta}$. The quadratic Equation (4.9) has the solution,

$$\lambda = -i \frac{c\Delta t}{\Delta x} \sin(k\Delta x) \pm \sqrt{1 - \left(\frac{c\Delta t}{\Delta x} \sin(k\Delta x)\right)^2}. \quad (4.10)$$

Remember that the absolute value of the complex number $a + ib$ is $\sqrt{a^2 + b^2}$! If

$$\left[\frac{c\Delta t}{\Delta x} \sin(k\Delta x) \right]^2 \leq 1 \quad (4.11)$$

then

$$|\lambda|^2 = \left[\frac{c\Delta t}{\Delta x} \sin(k\Delta x) \right]^2 + \left\{ \sqrt{1 - \left[\frac{c\Delta t}{\Delta x} \sin(k\Delta x) \right]^2} \right\}^2 = 1, \quad (4.12)$$

i.e. the scheme we are presently examining is stable if Equation (4.11) holds true and can also be formulated as

$$\frac{c\Delta t}{\Delta x} |\sin(k\Delta x)| \leq 1. \quad (4.13)$$

Since $|\sin(k\Delta x)| \leq 1$, we have conditional stability if

$$c \leq \frac{\Delta x}{\Delta t} \quad (4.14)$$

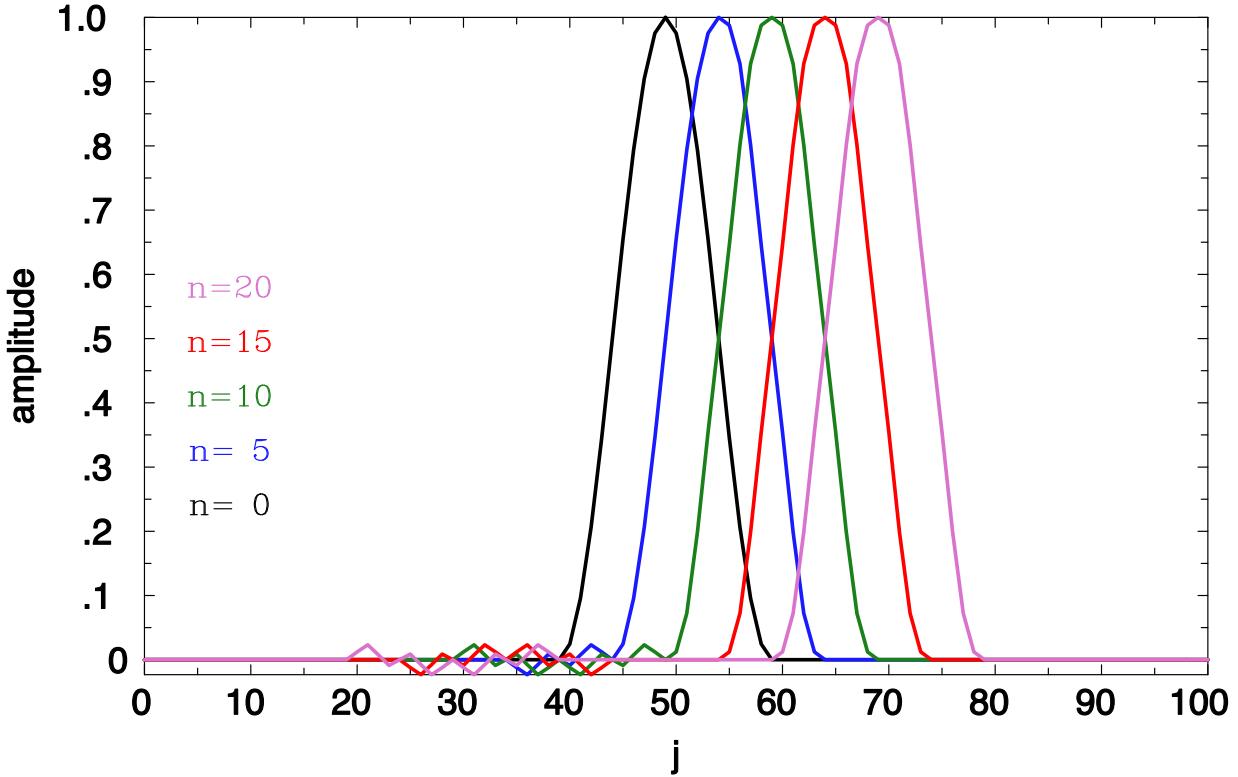


Figure 4.3: The advection equation integrated numerically with leap-frog and centred scheme in space Equation (4.1). The first time step has been integrated with an Euler (forward) scheme. The initial value is $u_j^{n=0} = \cos(\frac{\pi}{20}j)$ for $-10 \leq j \leq 10$ and $u_j^{n=0} = 0$ for the rest. The Courant number is for this scheme $\mu \equiv \frac{c\Delta t}{\Delta x} = 1$. Curves are plotted for the time steps $n=0, 5, 10, 15, 20$.

Note that this is because we need stability for all wave numbers k and we have hence chosen the "worst" case when $\sin(k\Delta x) = 1$.

The reverse case is when

$$\left[\frac{c\Delta t}{\Delta x} \sin(k\Delta x) \right]^2 > 1. \quad (4.15)$$

For simplicity we define

$$\gamma \equiv \frac{c\Delta t}{\Delta x} \sin(k\Delta x), \quad (4.16)$$

so that Equation (4.15) is reduced to

$$\gamma^2 > 1. \quad (4.17)$$

It is then preferable to rewrite Equation (4.10) as

$$\lambda = -i\gamma \pm i\sqrt{\gamma^2 - 1} = i \left(-\gamma \pm \sqrt{\gamma^2 - 1} \right) \quad (4.18)$$

or equivalently

$$|\lambda|^2 = \left(-\gamma \pm \sqrt{\gamma^2 - 1} \right)^2 = 2\gamma^2 \pm 2\gamma\sqrt{\gamma^2 - 1} - 1, \quad (4.19)$$

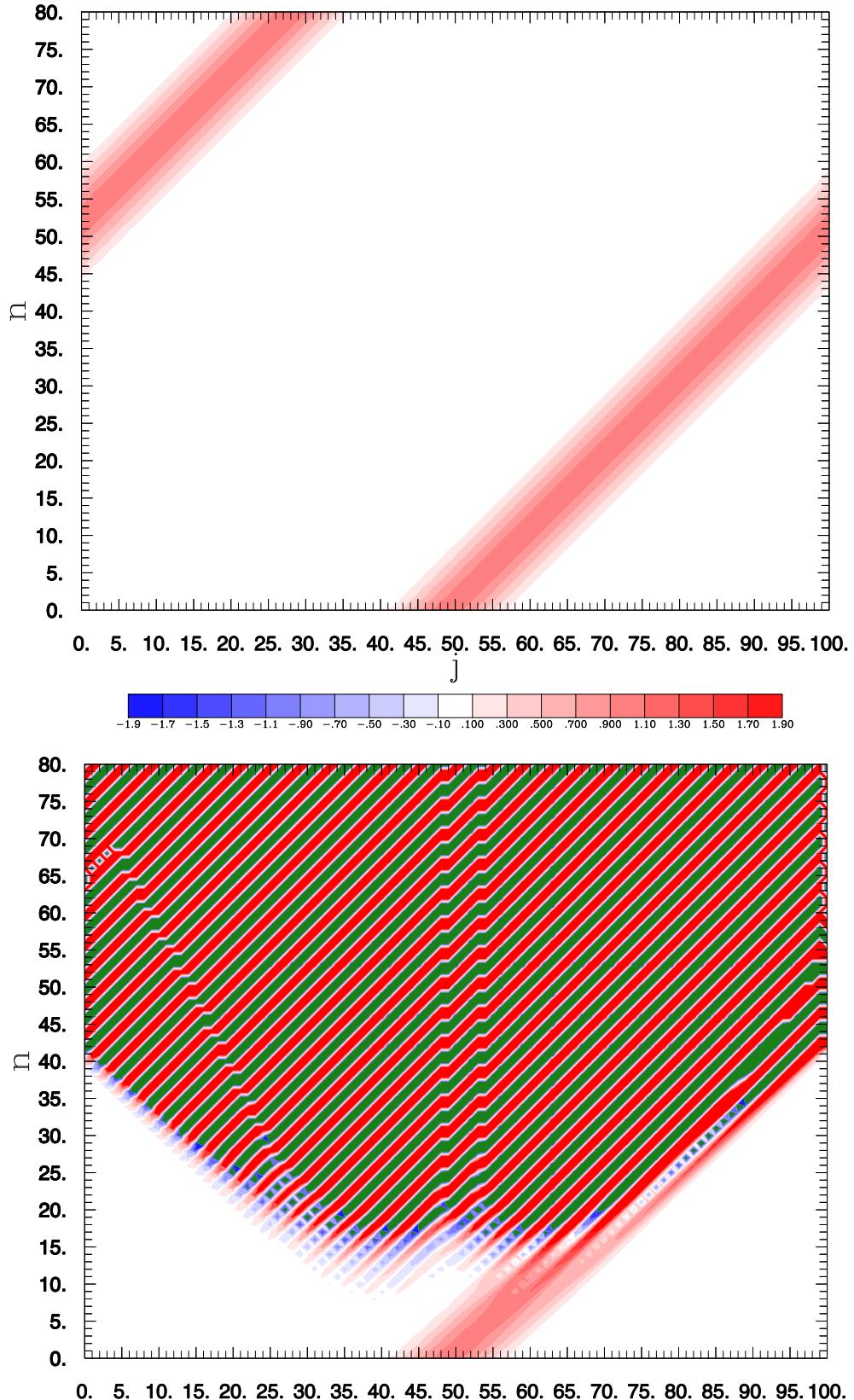


Figure 4.4: The advection equation integrated numerically with leap-frog and centred scheme in space Equation (4.1). The first time step has been integrated with an Euler (forward) scheme. The initial value is $u_j^{n=0} = \cos\left(\frac{\pi}{20}j\right)$ for $-10 \leq j \leq 10$ and $u_j^{n=0} = 0$ for the rest. The Courant number is $\mu \equiv \frac{c\Delta t}{\Delta x} = 1$ at the top panel, which gives a stable nice solution but the solution "blows up" with $\mu = 1.1$ in the bottom panel. Note also that cyclic boundary conditions have been applied.

4.4. THE ADVECTION EQUATION WITH AN EULER-FORWARD SCHEME IN TIME AND A CENTRED SCHEME IN SPACE

which has at least one root that is larger than one. Consequently when $c > \frac{\Delta x}{\Delta t}$, then $|\lambda| > 1$ and thus the solution is unstable.

The leap-frog scheme is thus *conditionally stable* and known as the *Courant-Fredrichs-Lowy (CFL)* criterion (Courant et al., 1928). For a given spatial resolution Δx we require a time step Δt not exceeding $\frac{\Delta x}{c}$ for Equation (4.14) to be valid for the fastest possible phase speed in the system. The *Courant number* (sometimes denoted the *CFL-number*) is defined as

$$\mu \equiv \frac{c\Delta t}{\Delta x}, \quad (4.20)$$

which should in this particular case of the advection equation with centred finite differences both in time and space not exceed 1 ($\mu \leq 1$) in order to satisfy the CFL stability criterion. The CFL criterion will as we will see later change with chosen finite differences and equations.

If the above advection equation with centres finite differences is applied to the ocean then we would for a chosen spatial grid resolution Δx adjust the time step Δt to satisfy $\mu \equiv \frac{c\Delta t}{\Delta x} \leq 1$. The phase speed is here the one for the long gravity waves ($c = \sqrt{gH}$), where H is the depth of the ocean. In order to guarantee numerical stability we need to choose the deepest depth H_{MAX} so that $\Delta t \leq \frac{\Delta x}{c} = \frac{\Delta x}{\sqrt{gH_{MAX}}}$.

4.4 The advection equation with an Euler-forward scheme in time and a centred scheme in space

Let us now use another numerical scheme for the advection equation, so that instead of Equation (4.1) we have

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + c \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} = 0, \quad (4.21)$$

which yields

$$\lambda = 1 - i \frac{c\Delta t}{\Delta x} \sin(k\Delta x). \quad (4.22)$$

As before the stability criterion is that $|\lambda| \leq 1$, and the absolute value of λ is now

$$|\lambda|^2 = 1 + \left[\frac{c\Delta t}{\Delta x} \sin(k\Delta x) \right]^2 \quad (4.23)$$

and thus $|\lambda| > 1$. The solution consequently grows with time, independently of how we choose the time step. This scheme is thus *unconditionally unstable*, which is simply referred to as *unstable*.

4.5 The upstream or upwind scheme

Let us now use yet another numerical scheme for the advection equation:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + c \frac{u_j^n - u_{j-1}^n}{\Delta x} = 0 \quad if c > 0, \quad (4.24)$$

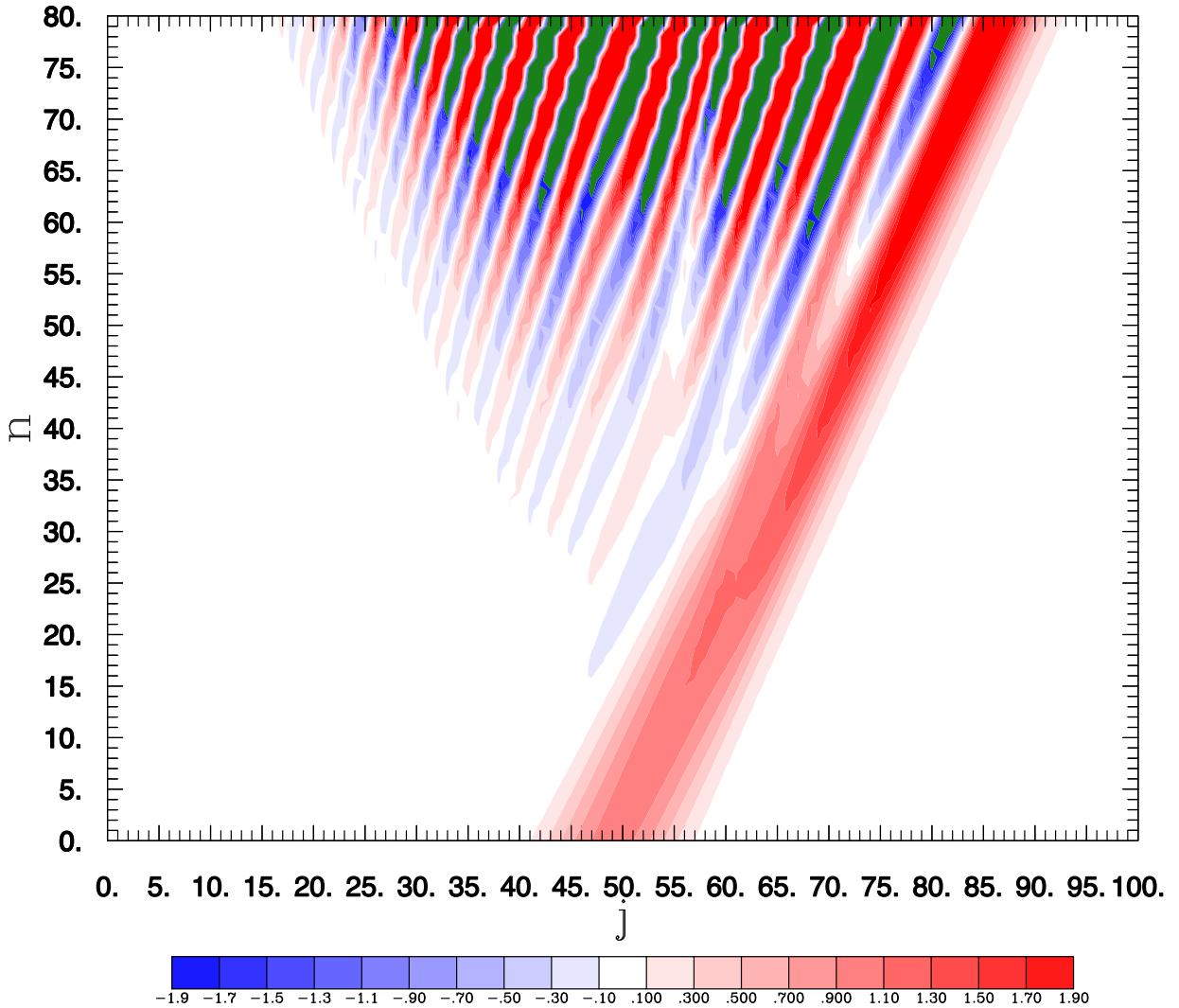


Figure 4.5: The advection equation integrated with Euler forward and a centred scheme in space. The Courant number is here $\mu \equiv \frac{c\Delta t}{\Delta x} = 0.5$.

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + c \frac{u_{j+1}^n - u_j^n}{\Delta x} = 0 \quad \text{if } c < 0.$$

This scheme is denoted upstream or upwind since it looks for information from where the wind or current comes at point $j - 1$. If $c > 0$ one should use a backward scheme in space in order to have an upstream scheme.

The von Neumann stability analysis yields

$$\lambda = 1 - \frac{c\Delta t}{\Delta x} [1 - \cos(k\Delta x) + i \sin(k\Delta x)]$$

and

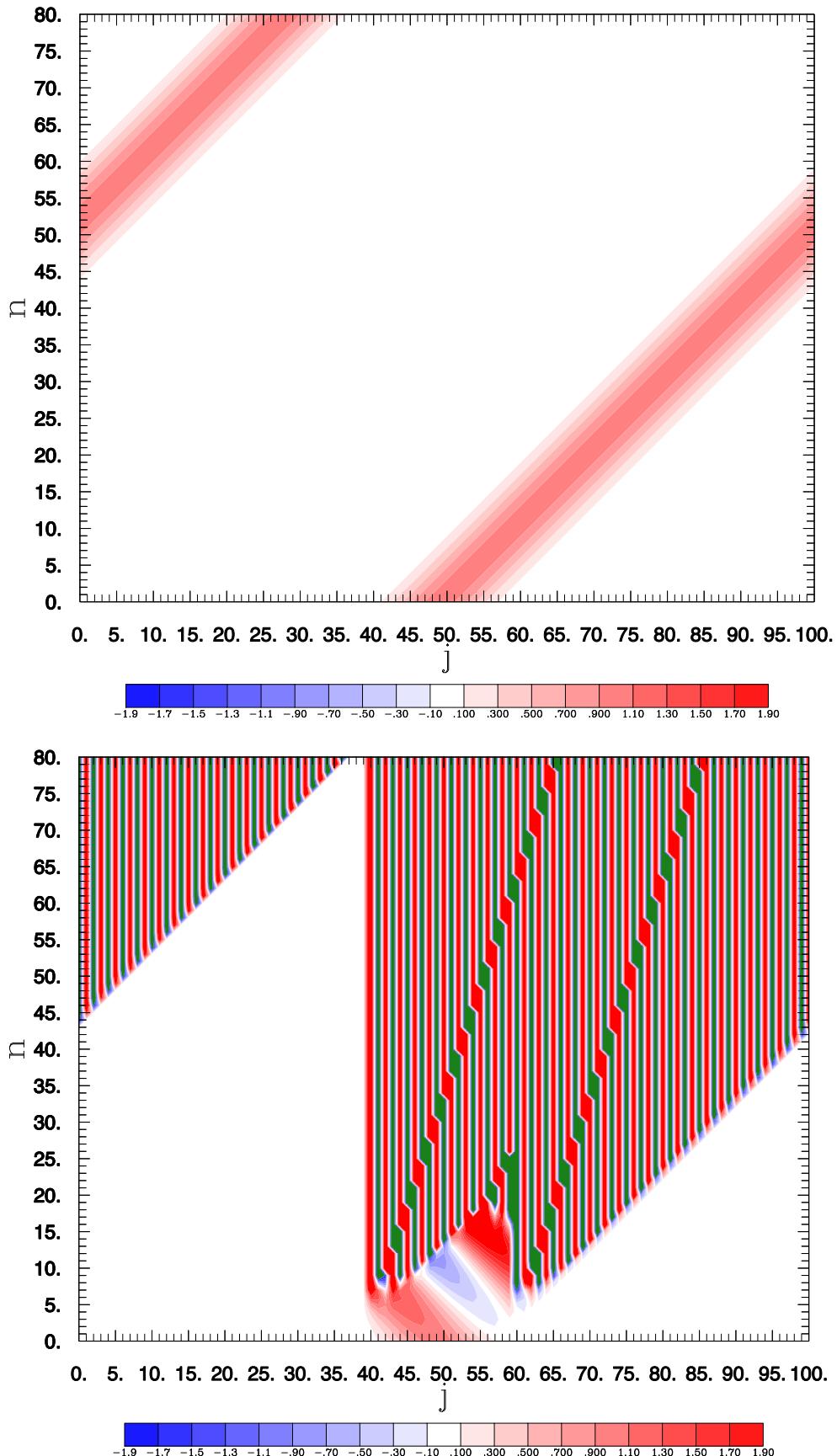


Figure 4.6: The advection Equation integrated numerically with the upstream scheme with Equation (4.24). Stable solution in the top panel where $\mu \equiv \frac{c\Delta t}{\Delta x} = 1$, but unstable in the bottom panel when $\mu = -1$, where a downstream scheme should instead be used.

$$|\lambda|^2 = \left\{ 1 - \frac{c\Delta t}{\Delta x} [1 - \cos(k\Delta x)] \right\}^2 + \left[\frac{c\Delta t}{\Delta x} \sin(k\Delta x) \right]^2 = 1 + 2\frac{c\Delta t}{\Delta x} [1 - \cos(k\Delta x)] \left(\frac{c\Delta t}{\Delta x} - 1 \right).$$

For $|\lambda|^2 < 1$ we require $0 \leq \frac{c\Delta t}{\Delta x} \leq 1$, which implies that c must be positive to ensure stability. The scheme is hence conditionally stable. The criterion is otherwise similar to the one for the leap-frog scheme.

The upwind advection scheme was used in early numerical weather models due to its good stability properties, and it still finds use in idealised ocean box models as well as in some general circulation models. When using an upwind scheme, however, one should realise that it is highly diffusive. It uses backward spatial differences if the velocity is in the positive x direction, and forward spatial differences for negative velocities. The term upwind denotes the use of upwind, or upstream, information in determining the form for the finite difference; downstream information is ignored.

4.6 The advection equation with a scheme of fourth order

Let us now study one advection scheme that uses the fourth-order accurate spatial scheme given by Equation (3.10):

$$\frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t} + c \left(\frac{4}{3} \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} - \frac{1}{3} \frac{u_{j+2}^n - u_{j-2}^n}{4\Delta x} \right) = 0. \quad (4.25)$$

The von Neumann method yields

$$\lambda^2 + i \frac{c\Delta t}{3\Delta x} [8 \sin(k\Delta x) - \sin(2k\Delta x)] \lambda - 1 = 0,$$

which has the solution

$$\lambda = -i \frac{c\Delta t}{6\Delta x} [8 \sin(k\Delta x) - \sin(2k\Delta x)] \pm \sqrt{1 - \left[\frac{c\Delta t}{6\Delta x} [8 \sin(k\Delta x) - \sin(2k\Delta x)] \right]^2}.$$

If

$$\left[\frac{c\Delta t}{6\Delta x} [8 \sin(k\Delta x) - \sin(2k\Delta x)] \right]^2 < 1 \quad (4.26)$$

then

$$|\lambda|^2 = \left[\frac{c\Delta t}{6\Delta x} [8 \sin(k\Delta x) - \sin(2k\Delta x)] \right]^2 + \left\{ \sqrt{1 - \left[\frac{c\Delta t}{6\Delta x} [8 \sin(k\Delta x) - \sin(2k\Delta x)] \right]^2} \right\}^2 = 1,$$

i.e. this scheme is stable if Equation (4.26) is fulfilled. This condition can also be expressed as

$$\left| \frac{c\Delta t}{6\Delta x} [8 \sin(k\Delta x) - \sin(2k\Delta x)] \right| < 1$$

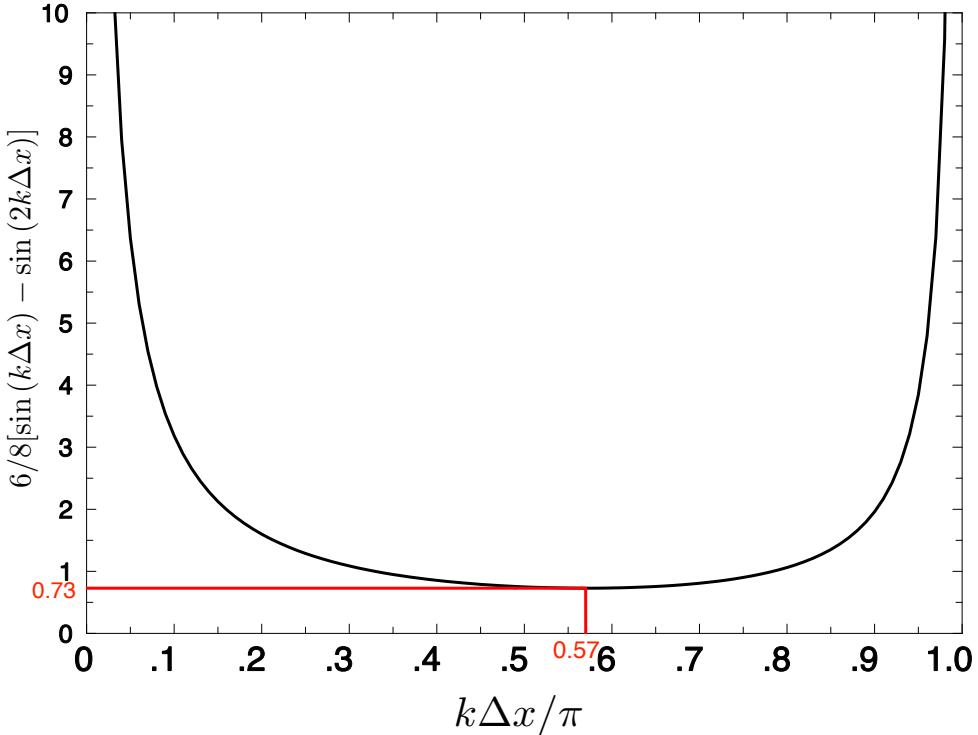


Figure 4.7: Right hand side of Equation (4.27) as a function of $k\Delta x/\pi$. The minimum value in red., which is the stability criterion with a Courant number $\mu \equiv \frac{c\Delta t}{\Delta x} \approx 0.73$.

or

$$\frac{c\Delta t}{\Delta x} < \frac{6}{8 \sin(k\Delta x) - \sin(2k\Delta x)}. \quad (4.27)$$

The exact stability criterion is found by searching for what $k\Delta x$ the right hand side of Equation (4.27) reaches its minimum as illustrated by Figure 4.7. This turns out to be $k\Delta x/\pi \approx 0.57$, which leads to that the scheme is stable when the Courant number $\mu \equiv \frac{c\Delta t}{\Delta x} \lesssim 0.73$. Figure 8.2 shows how the scheme evolves in time with a value just above and below the stability criterion.

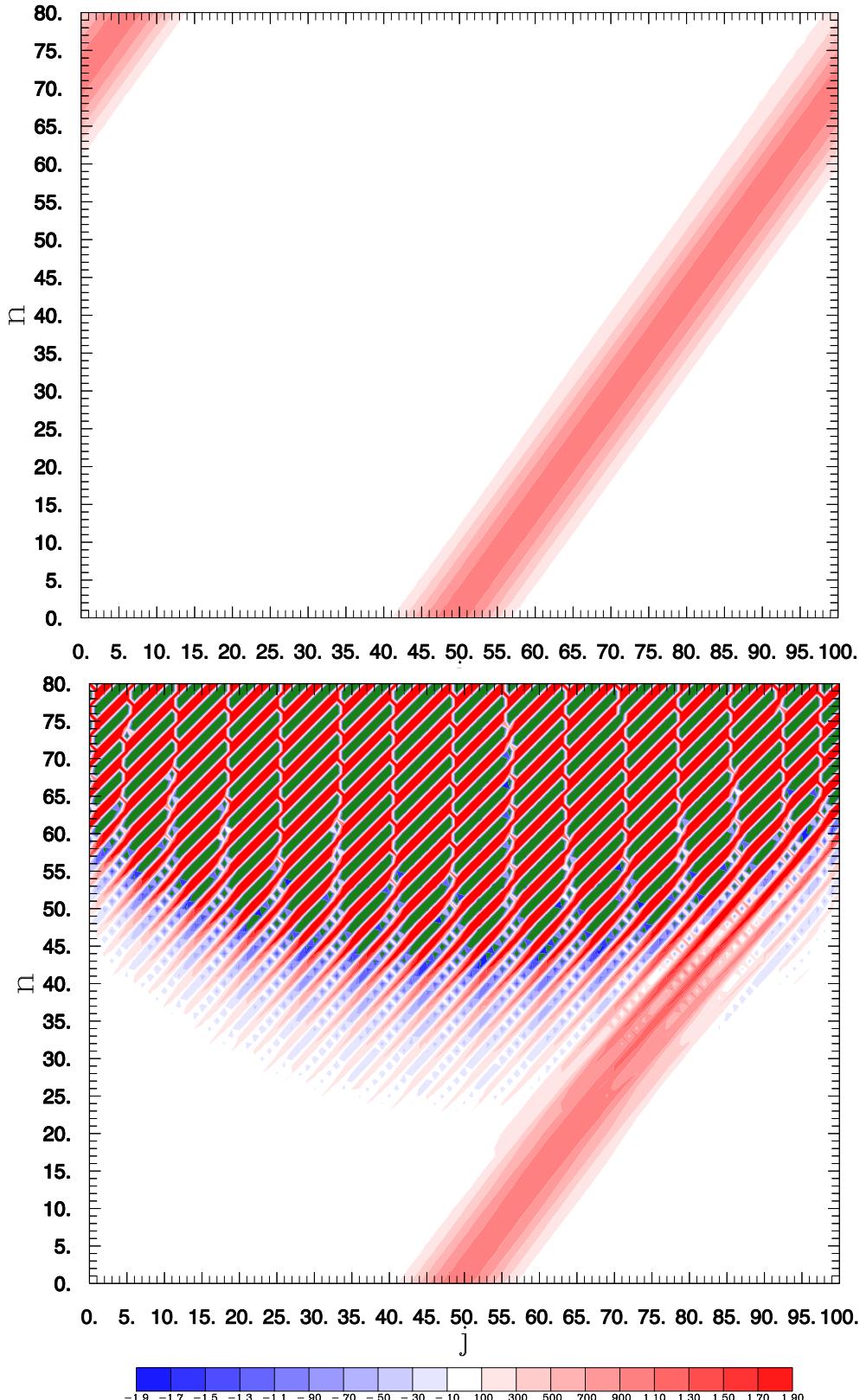


Figure 4.8: The advection equation integrated with the 4th order spatial numerical scheme Equation (4.25). Stable solution in the top panel where $\mu \equiv \frac{c\Delta t}{\Delta x} = 0.72$, but unstable in the bottom panel when $\mu = 0.74$.

Exercises:

- a) Consider the leapfrog scheme for the advection equation,

$$\frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t} + c \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} = 0.$$

Use

$$u_j^n = \lambda^n u_0 e^{ikj\Delta x}$$

and show that the amplification factor is

$$\lambda = -i \frac{c\Delta t}{\Delta x} \sin(k\Delta x) \pm \sqrt{1 - \left(\frac{c\Delta t}{\Delta x} \sin(k\Delta x) \right)^2}.$$

- b) Show that for $\frac{c\Delta t}{\Delta x} > 1$ in the previous exercise, we will have one of the solutions to the differential equation superior to one for at least some wave lengths, i.e. that the solution “blows up” .
- c) Discretise the advection equation with Euler forward in both time and space. Show that for $c > 0$ (down stream scheme), the amplitude of the solutions will grow in time (unstable). But for $c < 0$ (up stream scheme) the amplitude will decrease in time, i.e. stable.
- d) Make a stability analysis of the following discretisation of the advection equation

$$\frac{u_j^{n+1} - \frac{1}{2} (u_{j+1}^n + u_{j-1}^n)}{\Delta t} + c \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} = 0$$

Chapter 5

The numerical mode

In this chapter we shall examine some of the consequences of partial differential equations having been approximated with finite differences.

5.1 The numerical mode of the three-level schemes

One problem with a three-level scheme such as the leap-frog scheme is that it requires more than one initial condition to start the numerical integration. From a purely physical standpoint a single initial condition for $u^{n=0}$ should suffice. However, in addition to this physical initial condition, for computational purposes three-level schemes require an initial condition also for $u^{n=1}$. This value cannot be calculated by a three-level scheme, and, will usually have to be obtained using some type of two-level scheme.

Consider the oscillation equation:

$$\frac{du}{dt} = i\omega u; \text{ where } u = u(t)$$

which has the solution

$$u = u_0 e^{i\omega t}$$

The leap-frog scheme can be written as

$$u^{n+1} = u^{n-1} + 2i\omega \Delta t u^n \quad (5.1)$$

If we now study the amplification roots (cf. Chapter 4) we find

$$\lambda^2 - 2i\omega \Delta t \lambda - 1 = 0$$

which has the solution

$$\lambda_{1,2} = i\omega \Delta t \pm \sqrt{1 - (\omega \Delta t)^2}$$

Thus there are two solutions of the form $u^{n+1} = \lambda u^n$.

Since we are dealing with a linear equation, its solution will be a linear combination of the two solutions:

$$u_1^n = \lambda_1^n u_1^0 \quad \text{and} \quad u_2^n = \lambda_2^n u_2^0$$

so that

$$u^n = a\lambda_1^n u_1^0 + b\lambda_2^n u_2^0$$

byta till c1 $\circ\circ\circ\circ 7i6rt32e97$ where a and b are constants. If $u^{n+1} = \lambda u^n$ should represent the approximation of the true solution then $\lambda \rightarrow 1$ when $\Delta t \rightarrow 0$. λ_1 does this but $\lambda_2 \rightarrow -1$. λ_1 is called the *physical mode* and λ_2 is called the *computational mode*, which has been introduced by the numerical scheme . This computational mode changes sign for each even and odd n .

A straightforward way to illustrate the computational mode is to study the simple case when $\omega = 0$, viz.

$$\frac{du}{dt} = 0$$

which has the exact solution

$$u(t) = \text{const}$$

The leap-frog scheme yields

$$u^{n+1} = u^{n-1}$$

For a given physical initial condition $u^{n=0}$, we consider two special choices of $u^{n=1}$:

a) If calculating $u^{n=1}$ happened to yield the true value $u^{n=0}$, then for all n

$$u^{n+1} = u^n \tag{5.2}$$

or

$$u^{n+1} = \lambda_1 u^n$$

In this case we have obtained a numerical solution that is equal to the true solution and consists solely of the physical mode.

b) Suppose now instead that when calculating $u^{n=1}$ we obtain $u^{n=1} = -u^{n=0}$. Then for all n

$$u^{n+1} = -u^n \tag{5.3}$$

or

$$u^{n+1} = \lambda_2 u^n$$

The numerical solution now consists entirely of the computational mode.

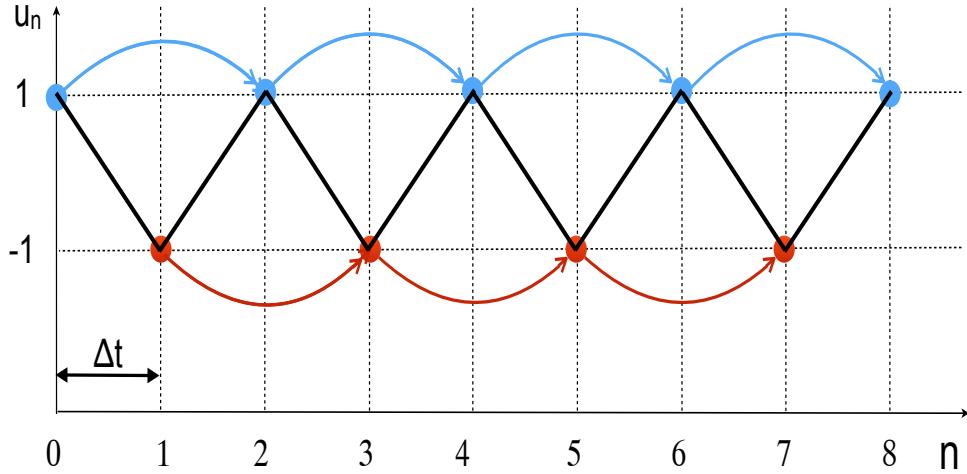


Figure 5.1: Illustration of the physical mode in blue from Equation (5.2) and the numerical (or computational) mode in red from Equation (5.3). The initial physical condition is set to $u^{n=0} = 1$ and the computation initial condition is set to $u^{n=1} = -1$. The resulting solution in black will thus change sign every time step since the two modes are uncoupled.

5.1.1 The computational initial condition

A good choice of the computational initial condition is of vital importance for obtaining a satisfactory numerical solution for short simulations, where the initial condition is crucial (which is the case for weather forecast but less so for long climate simulations). The computational initial condition ($u^{n=1}$), which is one time step ahead of the physical condition ($u^{n=0}$) can be computed with a single Euler-forward time step. Although this scheme is computationally unstable, it can be used for a single time step since many time steps are required before the solution grows and "blows up".

The computational initial condition for our academic case of the oscillation equation is then, with an Euler-forward time step:

$$u^{n=1} = u^{n=0} + i\omega\Delta t u^{n=0} \quad (5.4)$$

An alternative computational initial condition, when the solution is not so sensitive to the initial condition is to assign to the same value to both time steps, ($u^{n=1} = u^{n=0}$), but as shown above this can immediately trigger a computational mode.

5.2 Suppression of the computational mode

The computational mode introduced by the leap-frog scheme can be suppressed in two different ways:

5.2.1 Euler-forward or -backward at regular intervals

???JÄMFÖR MED TIDIGARFE NOATIONER

The problem can be solved by integrating with an Euler-forward or -backward scheme at regular intervals constituted by a certain number of time steps (50 time steps or so is often used). This would imply that

in the example of the previous paragraph ????? that $u^{n+1} = u^n$ every 50 time steps, which eliminates the computational mode so that $a = 1$ and $b = 0$.

5.2.2 The Robert-Asselin filter

Another way, which is the most common one used in atmospheric models, is to employ a Robert-Asselin filter (Robert, 1966; Asselin, 1972). First one applies a leap-frog integration to obtain the solution at time level $n + 1$:

$$u^{n+1} = u_f^{n-1} + 2\Delta t f(u^n)$$

whereafter the filter is applied as a time smoothing between the time levels $n - 1$, n and $n + 1$ so that

$$u_f^n = u^n + \gamma (u_f^{n-1} - 2u^n + u^{n+1}) \quad (5.5)$$

where the index f indicates the smoothed values and γ is the Asselin coefficient, which usually chosen to range be between 0.01 and 0.2. The next "frog jump" will be

$$u^{n+2} = u_f^n + 2\Delta t f(u^{n+1})$$

Note that the added term resembles smoothing in time; an approximation of an ideally time-centred smoother is

$$u_f^n = u^n + \gamma (u^{n-1} - 2u^n + u^{n+1}) \quad (5.6)$$

In our particular case of the discretised oscillation given by Equation (5.4) we can estimate the damping effect of the Robert-Asselin filter by introducing its discretised solution $u^n = u_0 e^{i\omega n \Delta t}$ into the smoother (Equation (5.6)), with the exception that u^{n-1} is taken as an unfiltered value. This results in

$$u_f^n = u^n [1 - 4\gamma \sin^2(\omega \Delta t / 2)] \quad (5.7)$$

The computational mode, the period of which is $2\Delta t$, is hence reduced by $(1 - 4\gamma)$ every time step. Because the field at $n - 1$ is replaced by an already filtered value, the Robert-Asselin filter introduces a slight difference compared to this simplified filter.

5.2.3 The Robert-Asselin-Williams filter

Williams (2009) showed that when used with the leap-frog scheme, the nonmean-conserving feature of the Robert-Asselin filter degrades the numerical accuracy. In the same study, the author tackled this problem by introducing an extra step in the filtering process in order to include the possibility of conserving the mean value. The resulting filter is implemented in leap-frog integrations as follows:

$$u^{n+1} = u_f^{n-1} + 2\Delta t f(u_f^n) \quad (5.8)$$

$$u_{ff}^n = u_f^n + \frac{\gamma\alpha}{2} (u^{n+1} - 2u_f^n + u_{ff}^{n-1}) \quad (5.9)$$

$$u_f^{n+1} = u^{n+1} - \frac{\gamma(1-\alpha)}{2} \left(u^{n+1} - 2u_f^n + u_{ff}^{n-1} \right) \quad (5.10)$$

The Robert-Asselin-Williams filter introduces an extra operation that is simple and does not represent a significant computational expense compared to the Robert-Asselin filter. It also introduces a new parameter, α , such that $0 < \alpha < 1$, where $\alpha = 1$ corresponds to the traditional Robert-Asselin filter. [Williams \(2009\)](#) showed that a value of $\alpha = 0.53$ minimises spurious numerical impacts on the physical solution and yields the closest match to the exact solution over a broad frequency range.

Chapter 6

Accuracy of the numerical phase speed

We shall now investigate the accuracy of the numerical phase speed associated with the discretisations in space as well in time.

6.1 Dispersion due to the spatial discretisation

Let us first examine the advection equation with a centred scheme in space:

$$\frac{\partial u}{\partial t} + c \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} = 0.$$

By substituting wave solutions

$$u_j(t) = u_0 e^{ik(j\Delta x - C_D t)},$$

we find

$$C_D = c \frac{\sin(k\Delta x)}{k\Delta x},$$

where C_D is the numerical phase speed and c the true phase speed. Their ratio should ideally be as close as possible to one, but is

$$\frac{C_D}{c} = \frac{\sin(k\Delta x)}{k\Delta x}.$$

The numerical group speed is

$$C_{Dg} = \frac{d(\omega_D)}{dk} = \frac{d(kC_D)}{dk} = c \cos(k\Delta x),$$

which is dispersive since it depends on the wave number k .

6.2 Dispersion due to the time discretisation

The effects of the centred scheme will be analysed in the same way as the effects of the centred scheme in space:

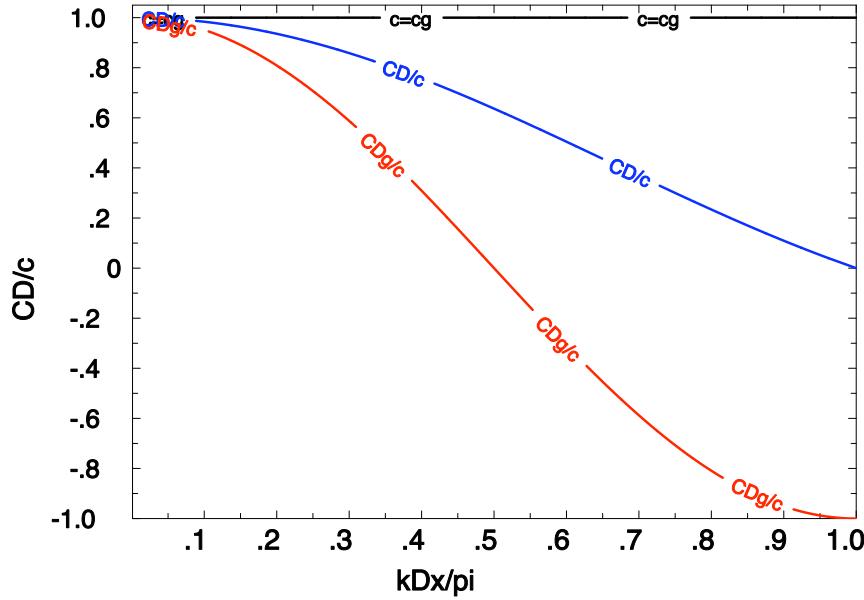


Figure 6.1: The numerical phase speed due to the centred finite difference in space compared to the analytical phase speed. The black line is the solution of the continuous equations which is the non dispersive analytical case, i.e. the phase speed is the same as the group velocity $c = c_g$. The blue line is the numerical phase speed normalised by dividing with c . Note that when the wave number increases (wave length decreases) then the numerical phase speed deviates from the analytical phase speed. The phase speed is clearly dispersive since the waves propagate at different speeds depending on their wave lengths. The red line shows the computational group velocity CD_g which at wave lengths shorter than 4 grid cells ($k\Delta x < \pi/2$) propagate in the wrong direction.

$$\frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t} + c \frac{\partial u}{\partial x} = 0$$

Substituting the wave solutions

$$u^n(x) = u_0 e^{ik(x - C_D n \Delta t)}$$

we get

$$C_D = \frac{\arcsin(\omega \Delta t)}{k \Delta t}$$

where C_D is the numerical phase speed and c the true phase speed. The ratio should ideally be as close as possible to one, but is

$$\frac{C_D}{c} = \frac{\arcsin(\omega \Delta t)}{\omega \Delta t}$$

The computational and physical group speeds also differ, so that

$$\frac{C_{Dg}}{c_g} = \frac{d(\omega_D)}{c_g dk} = \frac{d(kC_D)}{c_g dk} = \frac{1}{\sqrt{1 - (\omega \Delta t)^2}}$$

The computational phase speed and group velocity (both normalised with c) are presented in Figure 6.2, graphed as functions of $\omega\Delta t$.

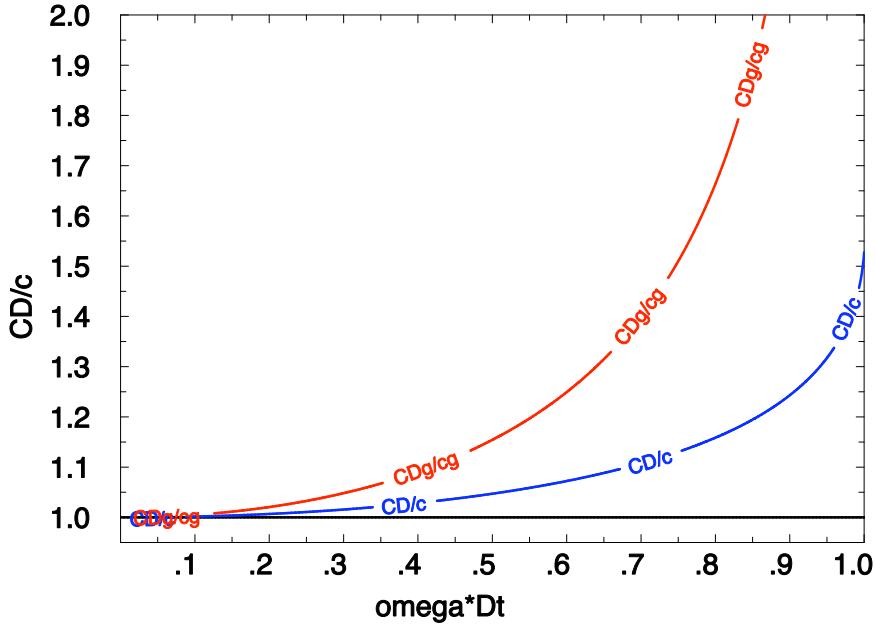


Figure 6.2: The numerical phase speed due to the centred finite difference in time compared to the analytical phase speed. The black line is the solution of the continuous equations which is the non dispersive analytical case, i.e. the phase speed is the same as the group velocity $c = c_g$. The blue line is the numerical phase speed normalised by dividing with c . Note that when the time step increases compared to the wave frequency (ω) then the numerical phase speed increases and deviates from the analytical phase speed. The red line shows the computational group velocity C_{Dg} .

6.3 Dispersion due to both spatial and temporal discretisation

Let us now investigate the effects of the "leap-frog" scheme on the phase and group velocities using the advection equation with centred schemes in both time and space:

$$\frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t} + c \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} = 0$$

By substituting wave solutions

$$u_j^n = u_0 e^{ik(j\Delta x - C_D n \Delta t)}$$

we obtain

$$C_D = \frac{1}{k\Delta t} \arcsin \left[\frac{c\Delta t}{\Delta x} \sin(k\Delta x) \right]$$

where C_D is the numerical phase speed and c the true phase speed. Their ratio should ideally be as close as possible to one, but is

$$\frac{C_D}{c} = \frac{1}{\mu k \Delta x} \arcsin [\mu \sin (k \Delta x)]$$

where the Courant number is defined as before as $\mu \equiv \frac{c \Delta t}{\Delta x}$.

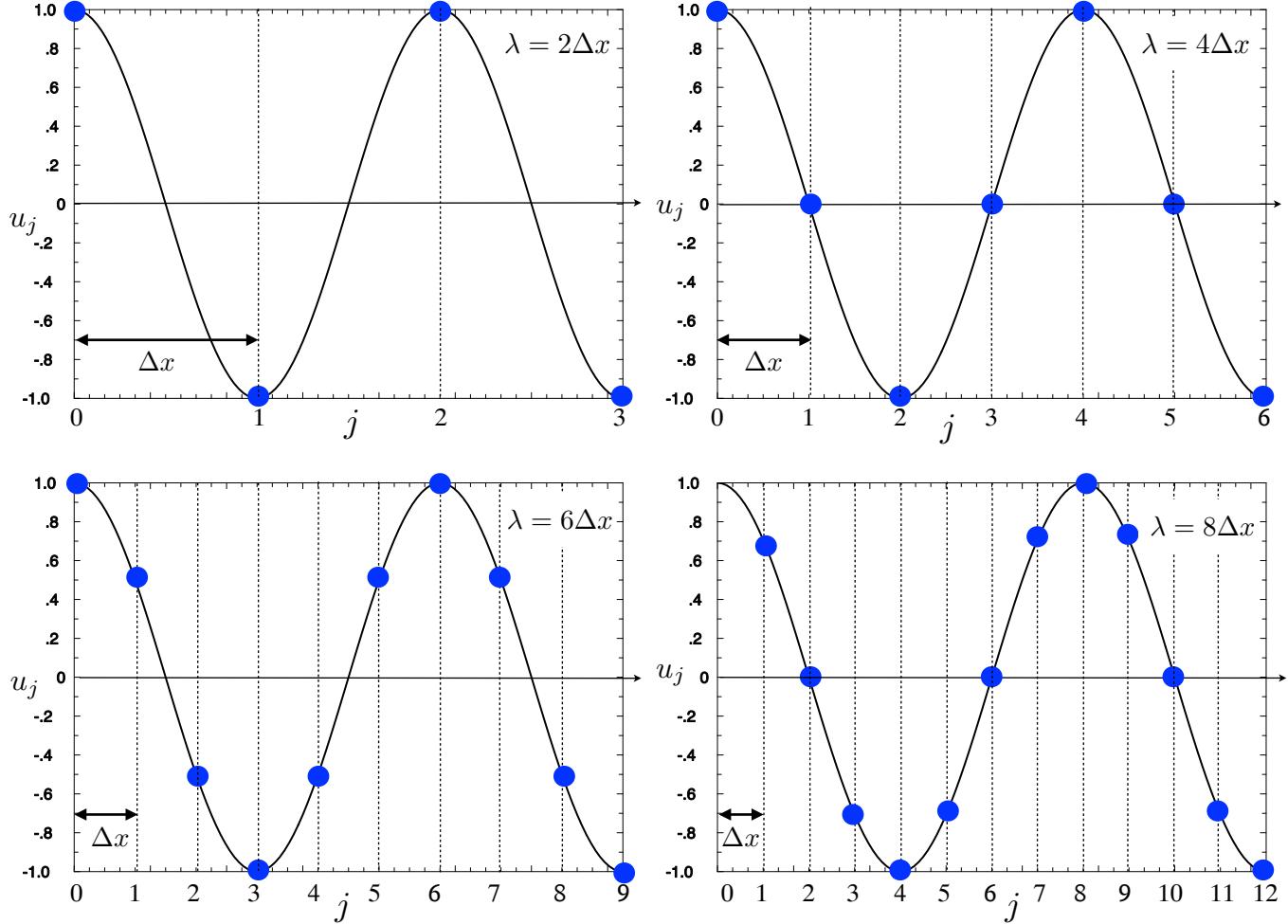


Figure 6.3: The two, three, four and 8 grid-interval wave with a wave length of $\lambda = 2, 3, 4$ and $8\Delta x$.

This phase speed is a function of the wave number k . The finite differencing in space thus causes a *computational dispersion*. As $k \Delta x$ increases, the computational phase speed C_D decreases from c to zero when $k \Delta x = \pi$, which corresponds to the shortest possible wave with a wave length of two-grid cells ($\lambda = 2\Delta x$). Thus, all waves propagate at a slower speed than the true phase speed c , with this decelerating effect increasing as the wave length decreases. The two grid-cell wave is stationary. Note that if $\mu = 1$, which is the limit of stability for the advection equation, the computation phase speed is the same as the analytical one ($C_D = c$).

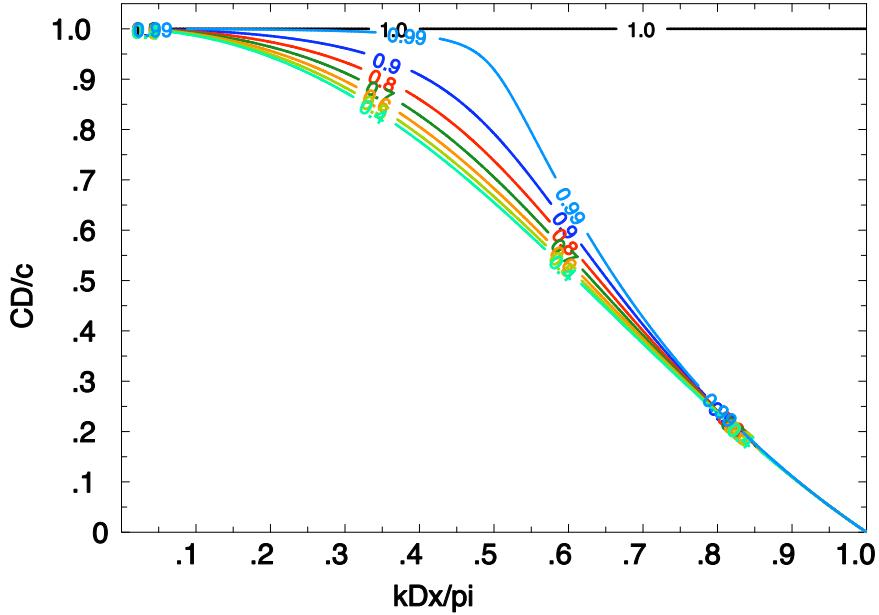


Figure 6.4: Numerical dispersion of the leap-frog scheme. The lines show the ratio C_D/c as a function of the normalised wave number $\frac{k\Delta x}{\pi}$. The different lines correspond to different Courant numbers ($\mu \equiv \frac{c\Delta t}{\Delta x}$), where these are indicated on the lines. The ideal solution is $C_D/c = 1$. Note that when the wave number increases (wave length decreases) then the numerical phase speed deviates from the analytical phase speed. The phase speed is clearly dispersive since the waves propagate at different speeds depending on their wave lengths.

The reason for the two grid-cell wave being stationary is obvious when we look at the wave illustrated in Figure 6.3. For this wave we have $u_{j+1} = u_{j-1}$ at all grid points, which yields $\frac{\partial u_j}{\partial t} = 0$ in the advection equation.

We have encountered two effects in this section:

- The advection speed is slower than the true advection speed.
- The advection speed changes with the wave number.

Exercises

Derive the numerical phase speed :

$$C_D = \frac{1}{k\Delta t} \arcsin \left[\frac{c\Delta t}{\Delta x} \sin(k\Delta x) \right]$$

Chapter 7

The shallow-water equations

In this chapter we will consider the equations describing the horizontal propagation of gravity and inertia-gravity waves. These equations are often referred to as the linearised shallow-water equations. Mathematically, this means that we will be dealing with a system of two or three partial differential equations of first order. Thus, we will now have two or three dependent variables (one or two velocities and pressure/free surface height). The system of equations will always be equivalent to a single differential equation, but of a higher order. This equation can be obtained from the system by elimination of dependent variables.

7.1 One-dimensional gravity waves with centred space differencing

For simplicity we shall start with the simplest case of gravity waves:

$$\frac{\partial u}{\partial t} = -g \frac{\partial h}{\partial x} \quad (7.1a)$$

$$\frac{\partial h}{\partial t} = -H \frac{\partial u}{\partial x}, \quad (7.1b)$$

where g is the gravity, h is the depth of the ocean or height of the atmosphere and H its position when the fluid is static, i.e. $u = 0$. We seek solutions of the form

$$u(x, t) = u_0 e^{i(kx - \omega t)} \quad (7.2a)$$

$$h(x, t) = h_0 e^{i(kx - \omega t)}, \quad (7.2b)$$

which yield the frequency equation

$$\omega^2 = gHk^2$$

and hence the phase speed is

$$c = \frac{\omega}{k} = \pm \sqrt{gH}$$

This shows that the gravity waves can propagate along the x-axis in both directions with a speed of \sqrt{gH} , which is not a function of the wave number and consequently there is no dispersion of the waves.

Consider now the differential-finite-difference equations with continuous time derivatives and centred

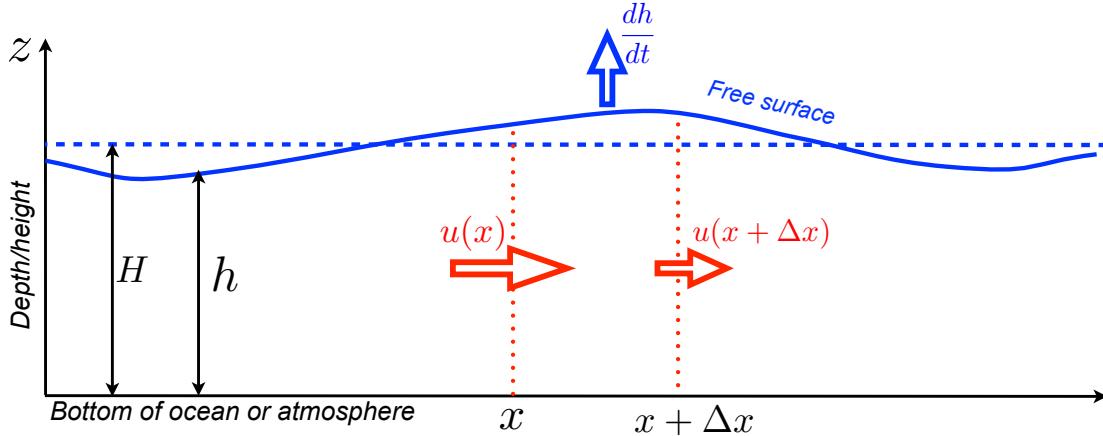


Figure 7.1: Schematic illustration of the shallow water equations in one dimension.

finite differences in space

$$\frac{\partial u_j}{\partial t} = -g \frac{h_{j+1} - h_{j-1}}{2\Delta x} \quad (7.3a)$$

$$\frac{\partial h_j}{\partial t} = -H \frac{u_{j+1} - u_{j-1}}{2\Delta x}, \quad (7.3b)$$

We seek solutions of the form

$$u_j = u_0 e^{i(jk\Delta x - \omega_D t)}, \\ h_j = h_0 e^{i(jk\Delta x - \omega_D t)},$$

which after insertion in Equation (7.3) yield the numerical phase speed

$$c_D = \frac{\omega_D}{k} = \pm \sqrt{gH} \frac{\sin(k\Delta x)}{k\Delta x}$$

and the numerical group speed

$$C_{Dg} = \frac{d(kC_D)}{dk} = \pm \sqrt{gH} \cos(k\Delta x).$$

Both numerical speeds are functions of the wave number, and thus we recognise that the space differencing again results in computational dispersion, viz. the same as obtained for the advection equation with centred schemes.

As illustrated by Figure 7.2, there are two types of possible grids for these types of equations. We can have the two dependent variables in the same points as in Equation (7.3) or we can alternate them in space.

$$\frac{\partial u_j}{\partial t} = -g \frac{h_{j+1} - h_j}{\Delta x}, \quad (7.5a)$$

$$\frac{\partial h_j}{\partial t} = -H \frac{u_j - u_{j-1}}{\Delta x}. \quad (7.5b)$$

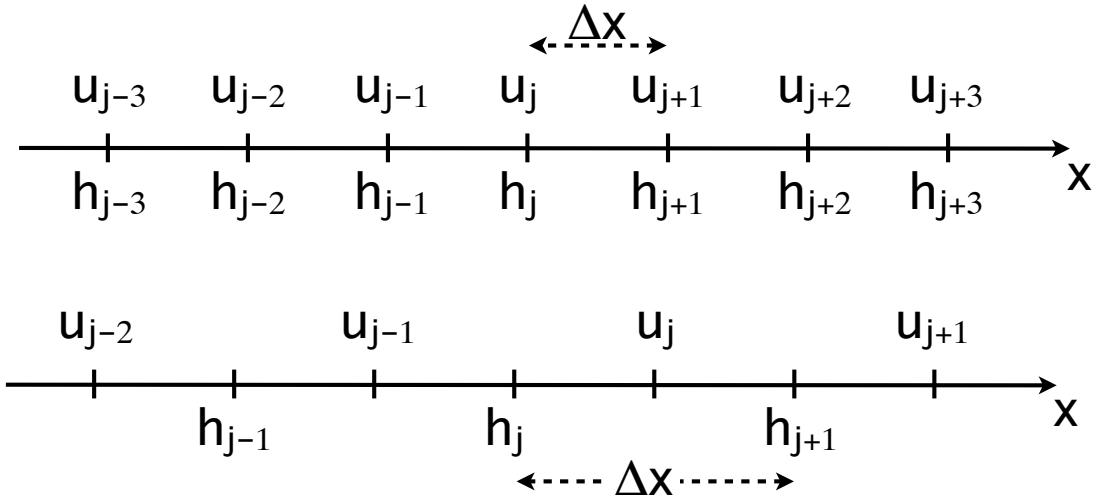


Figure 7.2: Top: unstaggered grid with two dependent variables that are both located at every grid point. Bottom: staggered grid with two dependent variables that are located at alternate grid points.

This is known as a staggered grid. The computational phase speed and group velocity now become

$$c_D = \frac{\omega_D}{k} = \pm \sqrt{gH} \frac{\sin\left(\frac{k\Delta x}{2}\right)}{\left(\frac{k\Delta x}{2}\right)}$$

and

$$C_{Dg} = \frac{d(\omega_D)}{dk} = \frac{d(kC_D)}{dk} = \pm \sqrt{gH} \cos\left(\frac{k\Delta x}{2}\right)$$

The staggered grid hence has the advantage that

- the computational time or the necessary number of grid points is halved
- the truncation error is halved with $\Delta x \rightarrow \Delta x/2$
- waves with $k\Delta x > \pi/2$, which are the waves shorter than 4 grid cells, are eliminated. They are the ones that have a large phase speed error.

Equations (7.5) also need to be discretised in time in order to be amenable to a numerical solution. The most straight-forward time difference is the three-level leapfrog scheme:

$$\frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t} = -g \frac{h_j^n - h_{j-1}^n}{\Delta x}, \quad (7.6a)$$

$$\frac{h_j^{n+1} - h_j^{n-1}}{2\Delta t} = -H \frac{u_{j+1}^n - u_j^n}{\Delta x}, \quad (7.6b)$$

which is centred in both space and time. The stability of these discretised Equations (7.6) can be obtained

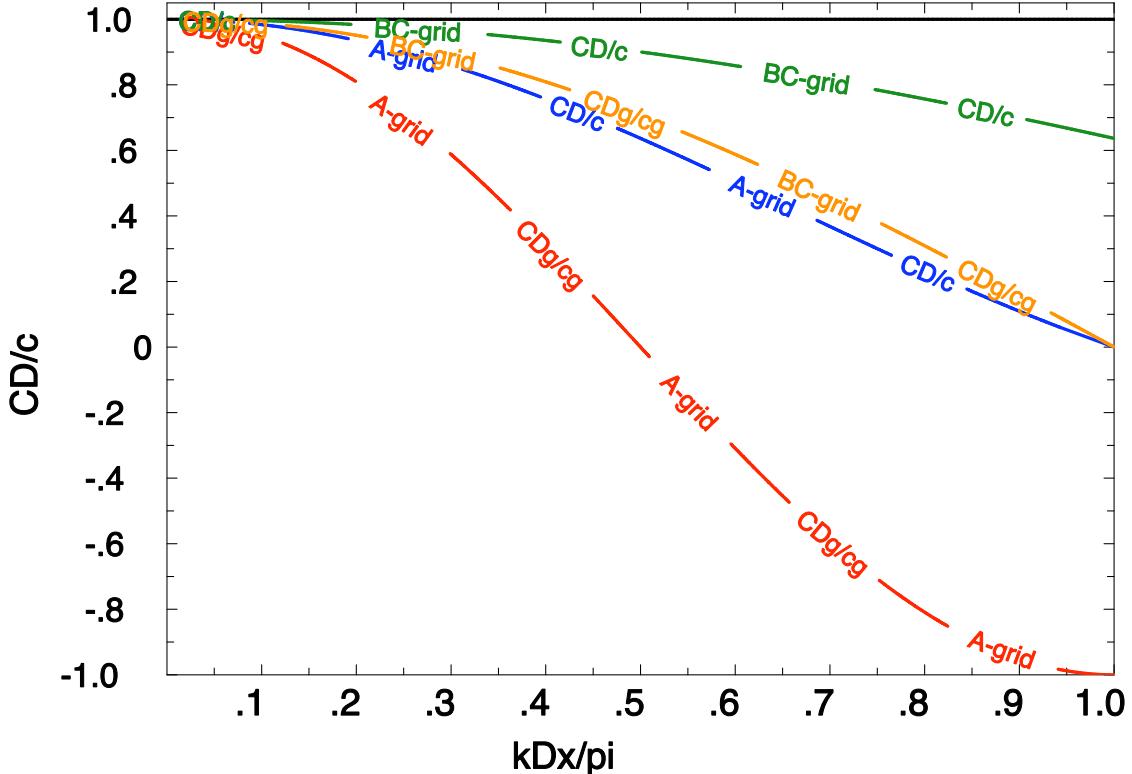


Figure 7.3: The numerical phase speed C_D due to the centred finite difference in space compared to the analytical phase speed c for one dimension shallow water Equations (7.3). The black line is the solution of the continuous equations which is the non dispersive analytical case, i.e. the phase speed is the same as the group velocity $c = c_g$. The blue line is the numerical phase speed normalised by dividing with c . Note that when the wave number increases (wave length decreases) then the numerical phase speed deviates from the analytical phase speed. The phase speed is clearly dispersive since the waves propagate at different speeds depending on their wave lengths. The red line shows the computational group velocity C_{Dg} which at wave lengths shorter than 4 grid cells ($k\Delta x < \pi/2$) propagate in the wrong direction. The green and orange lines are for the staggered BC-grid.

by the von Neumann stability method (cf. 4.3):

$$\frac{\lambda - \lambda^{-1}}{2\Delta t} u_0 = -g \frac{1 - e^{-ik\Delta x}}{\Delta x} h_0, \quad (7.7a)$$

$$\frac{\lambda - \lambda^{-1}}{2\Delta t} h_0 = -H \frac{e^{ik\Delta x} - 1}{\Delta x} u_0. \quad (7.7b)$$

We eliminate u_0 and h_0 between these two equations and find that

$$\left(\frac{\lambda - \lambda^{-1}}{2\Delta t} \right)^2 = \frac{gH}{(\Delta x)^2} \left(1 - e^{-ik\Delta x} \right) \left(e^{ik\Delta x} - 1 \right), \quad (7.8)$$

which results in two quadratic equations:

$$\lambda^2 \pm 2\alpha\lambda - 1 = 0 \quad (7.9)$$

where $\alpha \equiv \frac{2\Delta t \sqrt{gH}}{\Delta x} \sin\left(\frac{k\Delta x}{2}\right)$. The corresponding four roots are:

$$\lambda_{1,2,3,4} = \pm i\alpha \pm \sqrt{1 - \alpha^2}. \quad (7.10a)$$

The requirement for stability is that $|\lambda| \leq 1$, which is satisfied if $\alpha < 1$, which is the same as if

$$\frac{\Delta t \sqrt{gH}}{\Delta x} \sin\left(\frac{k\Delta x}{2}\right) \leq \frac{1}{2}. \quad (7.11)$$

To satisfy stability criterion for all wave lengths we require the Courant number μ is

$$\mu = \frac{\sqrt{gH}\Delta t}{\Delta x} \leq \frac{1}{2}. \quad (7.12)$$

Exercise: Show that stability criterion for the unstaggered case is that the Courant number must be $\mu \leq 1$.

7.2 2D shallow water equations

Let us now consider one of the simplest possible subsets of the Navier-Stokes equations in the atmosphere or the ocean, viz. the linearised shallow water equations (frequently denoted the gravity-inertia wave equations) in two dimensions:

$$\frac{\partial u}{\partial t} - fv = -g \frac{\partial h}{\partial x}, \quad (7.13a)$$

$$\frac{\partial v}{\partial t} + fu = -g \frac{\partial h}{\partial y}, \quad (7.13b)$$

$$\frac{\partial h}{\partial t} = -H \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right), \quad (7.13c)$$

where f is the Coriolis acceleration, here assumed to be constant. As before, we seek wave type solutions:

$$(u, v, h) = (u_0, v_0, h_0) e^{i(kx+ly-\omega t)} \quad (7.14)$$

When inserted into Equations (7.13), we obtain an equation for the frequency:

$$\omega^2 = f^2 + gH(k^2 + l^2),$$

which describes the dispersion relationship for Poicaré waves (inertia-gravity waves).

7.3 Two-dimensional gravity waves with centred space differencing

There are several possible grids known as the "Arakawa grids", which are usually identified with the letters A to E ([Mesinger and Arakawa, 1976](#)). The three most common ones are illustrated in Figure 7.4.

For each of the three grids we use the simplest centred approximations for the space derivative and the Coriolis terms. We do not need to study the time differencing since this has previously been examined and is the same as before.

A-grid:

$$\frac{\partial u_{i,j}}{\partial t} = -g \frac{h_{i+1,j} - h_{i-1,j}}{2\Delta x} + fv_{i,j}, \quad (7.15a)$$

$$\frac{\partial v_{i,j}}{\partial t} = -g \frac{h_{i,j+1} - h_{i,j-1}}{2\Delta y} - fu_{i,j}, \quad (7.15b)$$

$$\frac{\partial h_{i,j}}{\partial t} = -H \left(\frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} + \frac{v_{i,j+1} - v_{i,j-1}}{2\Delta y} \right). \quad (7.15c)$$

B-grid:

$$\frac{\partial u_{i,j}}{\partial t} = -g \frac{h_{i+1,j} + h_{i+1,j+1} - h_{i,j} - h_{i,j+1}}{2\Delta x} + fv_{i,j}, \quad (7.16a)$$

$$\frac{\partial v_{i,j}}{\partial t} = -g \frac{h_{i,j+1} + h_{i+1,j+1} - h_{i,j} - h_{i+1,j}}{2\Delta y} - fu_{i,j}, \quad (7.16b)$$

$$\frac{\partial h_{i,j}}{\partial t} = -H \left(\frac{u_{i,j} + u_{i,j-1} - u_{i-1,j} - u_{i-1,j-1}}{2\Delta x} + \frac{v_{i,j} + v_{i-1,j} - v_{i,j-1} - v_{i-1,j-1}}{2\Delta y} \right). \quad (7.16c)$$

C-grid:

$$\frac{\partial u_{i,j}}{\partial t} = -g \frac{h_{i+1,j} - h_{i,j}}{\Delta x} + \frac{f}{4} (v_{i,j} + v_{i+1,j} + v_{i+1,j-1} + v_{i,j-1}), \quad (7.17a)$$

$$\frac{\partial v_{i,j}}{\partial t} = -g \frac{h_{i,j+1} - h_{i,j}}{\Delta y} - \frac{f}{4} (u_{i,j} + u_{i,j+1} + u_{i-1,j+1} + u_{i-1,j}), \quad (7.17b)$$

$$\frac{\partial h_{i,j}}{\partial t} = -H \left(\frac{u_{i,j} - u_{i-1,j}}{\Delta x} + \frac{v_{i,j} - v_{i,j-1}}{\Delta y} \right). \quad (7.17c)$$

For simplicity we shall first study the quasi-one-dimensional case where u , v and h do not depend on y so

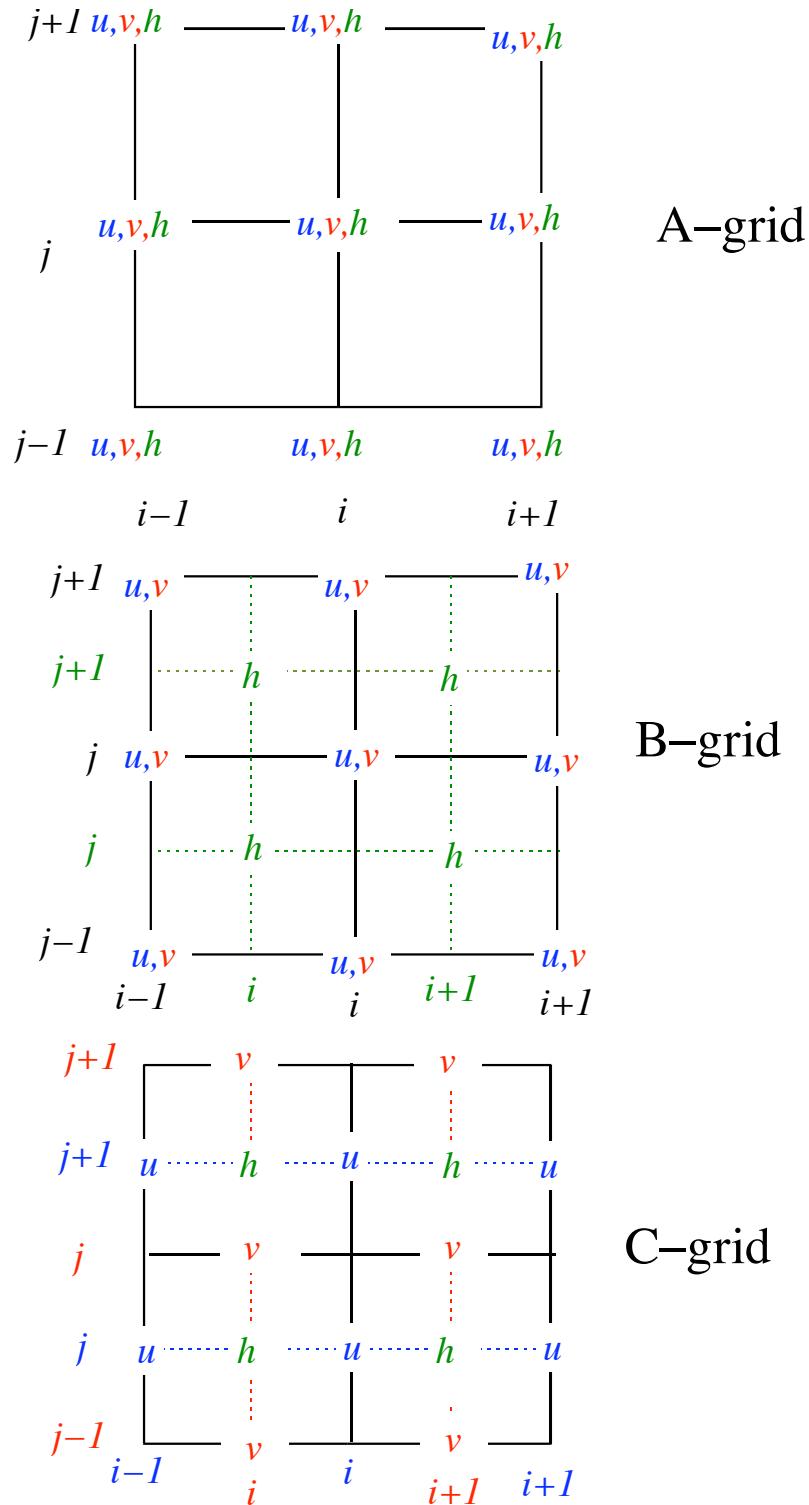


Figure 7.4: The three most common Arakawa grids: A, B and C.

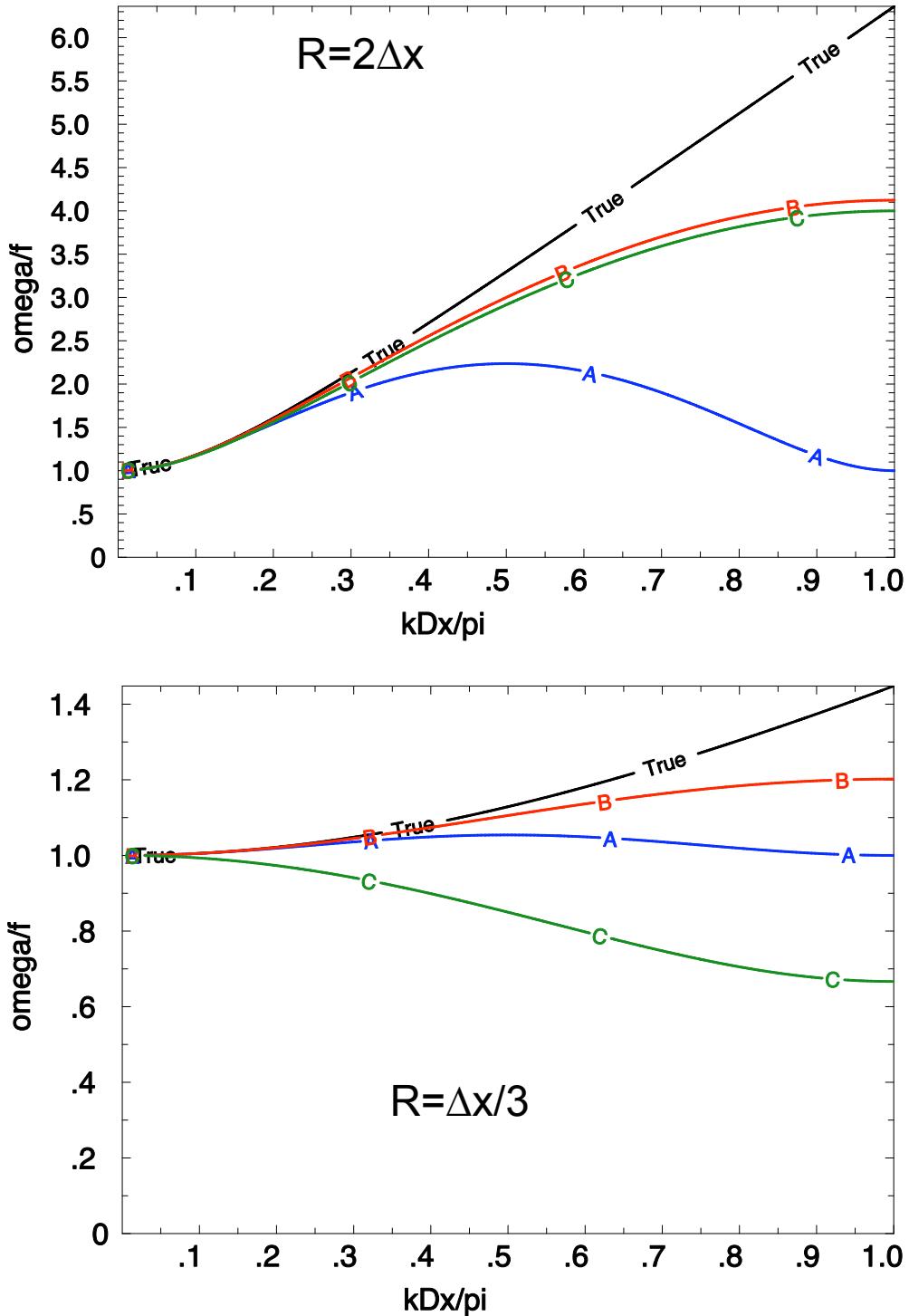


Figure 7.5: The function ω/f from (7.19) where at the top panel $gH/(f\Delta x)^2 = 4$, i.e. the Rossby radius set to two gridcells ($\sqrt{gH}/f = 2\Delta x$) and at the bottom panel $gH/(f\Delta x)^2 = 1/9$, i.e. the Rossby radius is set to a third of a grid cell ($\sqrt{gH}/f = \Delta x/3$). The black line corresponds to the true analytical solution, the blue line to the A-grid, the red line to the B-grid and the green line to the C-grid. NB: The B and C grid give similar results when the Rossby radius is well resolved but the C-grid degenerates when the grid resolution is coarse.

that the shallow-water Equations (7.13) reduced to

$$\frac{\partial u}{\partial t} - fv = -g \frac{\partial h}{\partial x}, \quad (7.18a)$$

$$\frac{\partial v}{\partial t} + fu = 0, \quad (7.18b)$$

$$\frac{\partial h}{\partial t} + H \frac{\partial u}{\partial x} = 0. \quad (7.18c)$$

Inserting the wave solutions in Equations (7.18) result in the frequency equation:

$$\left(\frac{\omega}{f}\right)^2 = 1 + \frac{gH}{f^2} k^2 \quad (7.19)$$

Let us now look at the effect of the finite differencing in space for this case. As the variables are assumed not to depend on y , Equations (7.15) for the A-grid reduce to

$$\frac{\partial u_{i,j}}{\partial t} = -g \frac{h_{i+1,j} - h_{i-1,j}}{2\Delta x} + fv_{i,j}, \quad (7.20a)$$

$$\frac{\partial v_{i,j}}{\partial t} = -fu_{i,j}, \quad (7.20b)$$

$$\frac{\partial h_{i,j}}{\partial t} = -\frac{H}{2\Delta x} (u_{i+1,j} - u_{i-1,j}), \quad (7.20c)$$

and for the B-grid:

$$\frac{\partial u_{i,j}}{\partial t} = -g \frac{h_{i+1,j} + h_{i+1,j+1} - h_{i,j} - h_{i,j+1}}{2\Delta x} + fv_{i,j}, \quad (7.21a)$$

$$\frac{\partial v_{i,j}}{\partial t} = -fu_{i,j}, \quad (7.21b)$$

$$\frac{\partial h_{i,j}}{\partial t} = -\frac{H}{2\Delta x} (u_{i,j} + u_{i,j-1} - u_{i-1,j} - u_{i-1,j-1}), \quad (7.21c)$$

and for the C-grid:

$$\frac{\partial u_{i,j}}{\partial t} = -g \frac{h_{i+1,j} - h_{i,j}}{\Delta x} + \frac{f}{4} (v_{i,j} + v_{i+1,j} + v_{i+1,j-1} + v_{i,j-1}), \quad (7.22a)$$

$$\frac{\partial v_{i,j}}{\partial t} = -\frac{f}{4} (u_{i,j} + u_{i,j+1} + u_{i-1,j+1} + u_{i-1,j}), \quad (7.22b)$$

$$\frac{\partial h_{i,j}}{\partial t} = -\frac{H}{\Delta x} (u_{i,j} - u_{i-1,j}). \quad (7.22c)$$

Inserting the wave solutions with no j -dependence $(u_i, v_i, h_i) = (u_0, v_0, h_0) e^{I(ik\Delta x - \omega_D t)}$, (where I is the Imaginary unit) into Equations (7.20-7.22) yields the following frequency equations:

$$\textbf{A-grid: } \left(\frac{\omega_D}{f}\right)^2 = 1 + \frac{gH}{f^2} \frac{\sin^2(k\Delta x)}{(\Delta x)^2}$$

$$\textbf{B-grid: } \left(\frac{\omega_D}{f}\right)^2 = 1 + \frac{gH}{f^2} \frac{\sin^2(k\Delta x/2)}{(\Delta x/2)^2}$$

$$\textbf{C-grid: } \left(\frac{\omega_D}{f}\right)^2 = \cos^2\left(\frac{k\Delta x}{2}\right) + \frac{gH}{f^2} \frac{\sin^2(k\Delta x/2)}{(\Delta x/2)^2}$$

The non-dimensional frequencies ω_D/f are now seen to depend on the two parameters $k\Delta x$ and gH/f^2 (Rossby radius squared) and are graphed in Figure 7.5, where they can be validated against results from the non-discretised solution of Equation (7.19).

One can summarise the pros and cons of the grids by

- Grid A: The frequency reaches a maximum at $k\Delta x = \pi/2$, i.e. a wave-length of 4 grid cells. The group velocity is thus zero for this wave length. If gravity-inertia waves of approximately this wave number are excited near a point inside the computational region, e.g. by non-linear effects or forcing through heating or ground topography, the wave energy remains near that point. Beyond this maximum value, for $\pi/2 < k\Delta x < \pi$, the frequency decreases as the wave number increases. For these waves the group velocity thus has the wrong sign. Finally, the two-grid-cell wave with $k\Delta x = \pi$ behaves like a pure inertial oscillation, and its group velocity is again zero.
- Grid B: The frequency increases monotonically over the range $0 < k\Delta x < \pi$. However it assumes a local maximum at the end of the range, and hence the group velocity is zero for the two-grid-cell wave with $k\Delta x = \pi$.
- Grid C: The frequency increases monotonically in a similar way as in the B-grid case if $gH/(f\Delta x)^2 > 1/4$, i.e. when the Rossby radius is larger than half a grid cell ($\sqrt{gH}/f > \frac{\Delta x}{2}$). If, however, the Rossby radius is exactly half a grid ($\sqrt{gH}/f = \frac{\Delta x}{2}$), the group velocity is zero and for smaller Rossby radii the frequency will decrease in an unrealistic way with increasing wave number over $0 < k\Delta x < \pi$. The advantage of the C-grid lies in that the velocities are normal to the grid-cell walls, which makes the differencing of the continuity equation as well as the scalar transport in the tracer equation a straightforward matter (c.f. section 13.2).

7.4 The shallow-water equations with leap-frog schemes

The discretised linearised inviscid shallow-water equations can now be written with centred finite differences in both time and space on a C-grid as

$$\frac{u_{i,j}^{n+1} - u_{i,j}^{n-1}}{2\Delta t} = -g \frac{h_{i+1,j}^n - h_{i,j}^n}{\Delta x} + \frac{f}{4} (v_{i,j}^n + v_{i+1,j}^n + v_{i+1,j-1}^n + v_{i,j-1}^n), \quad (7.23a)$$

$$\frac{v_{i,j}^{n+1} - v_{i,j}^{n-1}}{2\Delta t} = -g \frac{h_{i,j+1}^n - h_{i,j}^n}{\Delta y} - \frac{f}{4} (u_{i,j}^n + u_{i,j+1}^n + u_{i-1,j+1}^n + u_{i-1,j}^n), \quad (7.23b)$$

$$\frac{h_{i,j}^{n+1} - h_{i,j}^{n-1}}{2\Delta t} = -H \left(\frac{u_{i,j}^n - u_{i-1,j}^n}{\Delta x} + \frac{v_{i,j}^n - v_{i,j-1}^n}{\Delta y} \right). \quad (7.23c)$$

The von Neumann stability method can be applied to the non-rotating case, and hence Equation (7.23)

becomes

$$\frac{\lambda - \lambda^{-1}}{2\Delta t} u_0 = -g \frac{e^{Ik\Delta x} - 1}{\Delta x} h_0, \quad (7.24a)$$

$$\frac{\lambda - \lambda^{-1}}{2\Delta t} v_0 = -g \frac{e^{Il\Delta y} - 1}{\Delta y} h_0, \quad (7.24b)$$

$$\frac{\lambda - \lambda^{-1}}{2\Delta t} h_0 = -H \left(\frac{1 - e^{-Ik\Delta x}}{\Delta x} u_0 + \frac{1 - e^{-Il\Delta y}}{\Delta y} v_0 \right). \quad (7.24c)$$

We eliminate u_0 , v_0 and h_0 between these three equations and find that

$$(\lambda - \lambda^{-1})^2 = -16gH(\Delta t)^2 \left[\frac{\sin^2(k\Delta x/2)}{(\Delta x)^2} + \frac{\sin^2(l\Delta y/2)}{(\Delta y)^2} \right]. \quad (7.25)$$

The requirement for stability is that $|\lambda| \leq 1$, which is satisfied if

$$\sqrt{gH}\Delta t \sqrt{\frac{\sin^2(k\Delta x/2)}{(\Delta x)^2} + \frac{\sin^2(l\Delta y/2)}{(\Delta y)^2}} \leq \frac{1}{2}, \quad (7.26)$$

If we have assumed that $\Delta x = \Delta y$, then to satisfy stability criterion for all wave lengths we require

$$\frac{\sqrt{gH}\Delta t}{\Delta x} \leq \frac{1}{\sqrt{8}} \approx 0.35. \quad (7.27)$$

7.5 Boundary conditions

There are two types of boundary conditions: Closed boundary conditions, which correspond to the border between land points and ocean points. Open boundary conditions, which is where the model grid covering the domain under consideration ends but the real ocean/atmosphere continues.

7.5.1 Closed boundary conditions

The staggered B-grid is well adapted to no-slip boundary conditions, since the velocity points are located at the corners of the computational cell. Unlike the C-grid, there are no ambiguities in the way the dynamical boundary condition is imposed at "tips" of the adjacent land masses as shown in Figure 7.8.

The B-grid yields, however, a better dispersion relationship than the C-grid since it is better at resolving the Rossby radius at coarse resolutions (Batteen and Han, 1981), a feature that makes this staggering technique more suitable for coarsely-resolved models.

7.5.2 Open boundary conditions

An open boundary condition has two main purposes: It should permit waves to propagate out from the model domain without being reflected back. It should be possible to force the inner solution with external fields, which can e.g. be obtained from observations or models covering a larger domain.

Open boundary conditions also need to conserve mass so that the average of sea surface elevation (h) remains constant. The energy budget should also be treated accurately allowing the correct energy flux

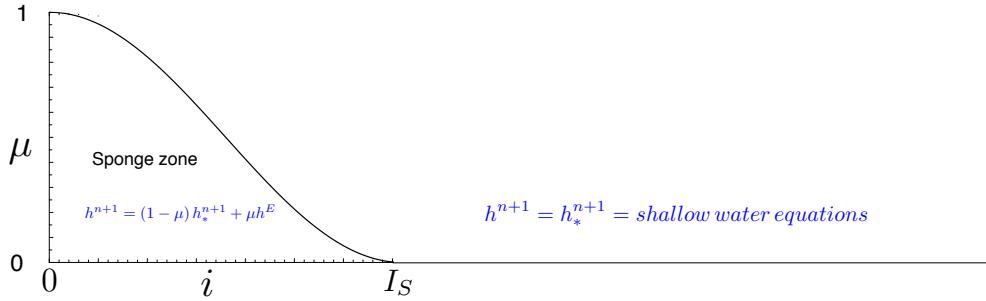


Figure 7.6: Schematic illustration of a sponge zone between the open boundary located at $i = 0$ and $i = I_S$.

through the open boundaries. ??? to balance the energy flux through the free surface...wind stress...lägg till diskussion om energi...kanske senare i Sadournys bit???

There are many different types of sophisticated radiative open-boundary conditions based on the wave equation. We will here, however, only present the simplest, which is the "sponge" boundary condition. Here all field variables are first updated using the standard interior leap-frog schemes. The field values in the sponge zone are then relaxed to the externally given values h^E according to

$$h^{n+1} = (1 - \nu) h_*^{n+1} + \nu h^E, \quad (7.28)$$

where h_*^{n+1} is obtained from the model equations, in the present case the shallow-water equations. The non-uniformity of h^E in the sponge zone is taken into account by letting the solution decay as we leave the boundary. This can be of an e-folding character or be a cosine-shaped relaxation factor such as

$$\nu = 0.5 \left[1 + \cos \left(\pi \frac{i}{I_S} \right) \right] \quad (7.29)$$

for the interval $0 \leq i \leq I_S$, where I_S is the number of grid points of the sponge zone, typically 10 to 30.

The externally given values h^E can originate from observations or from another model, which often has a coarser grid. This is the case for regional climate models as well as for most local numerical weather-prediction models, which are forced at their open boundaries by a global circulation model covering the entire Earth. Figure 7.7 shows schematically such a nested grid, where the light blue border can be treated as an open boundary for the finer interior grid, driven by the values obtained from the coarser-grid model. The nesting is hence a way to "zoom" in on a particular region by increasing the spatial resolution here. The nesting can be one-way, where only the interior values are influenced by the exterior values from the coarser surrounding grid or two-way, where also the coarser grid values are affected by the data from the fine grid.

Radiative boundary conditions such as those due to Orlanski (1976) have the advantage of letting the waves exit the domain without reflection and at the same time independently imposing the open boundary values h^E . Tests of different radiative boundary conditions can be found in e.g. Nycander and Döös (2003).

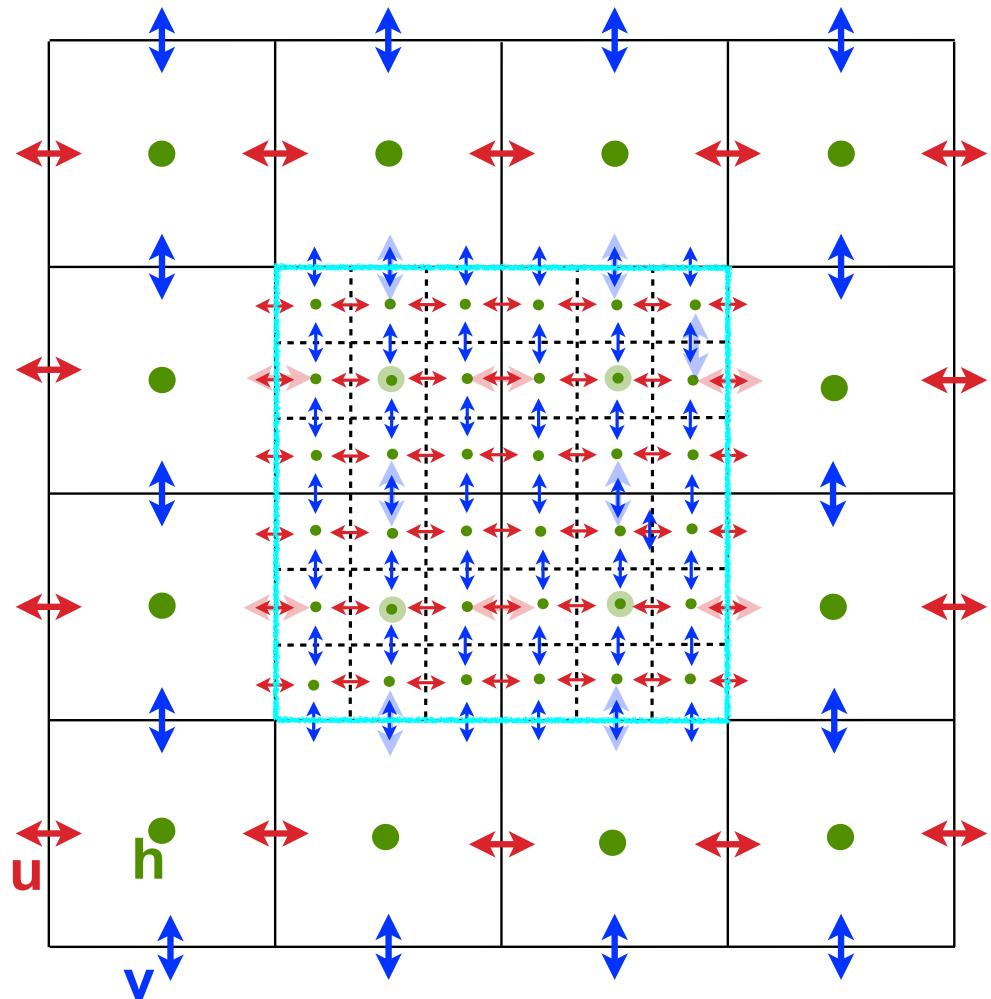


Figure 7.7: A nested C-grid with 3:1 grid size ratio. The solid lines denote coarse-grid cell boundaries and the dashed lines are the boundaries for each fine grid cell. The light blue blurry lines denote the boundaries between the fine and coarse grids, which can be treated as an open boundary for the fine grid.

7.5.3 Cyclic boundary conditions

Har inte detta gjorts tidigare ????

Cyclic boundary conditions are e.g. when the values on the eastern wall equal those on the western, which is the case for a global models of the Earth. This can then for instance be located at the Greenwich meridian where the longitude can be expressed as 0° or 360° . When computing u , v and h next to the eastern ($i = 0$) or western ($i = IX$) you will need values, which are for $i = 0$ and $i = IX + 1$. This can be easily be coded by introducing in the model code's i-loops $im = i - 1$ and when $im = -1$ then you set it to $im = IX - 1$. Same procedure for the eastern wall with $ip = i + 1$ and when $ip = IX + 1$ which is replaced by $ip = 1$.

7.6 Conservation of mass, energy and enstrophy

There are several reasons why numerical schemes for models are often formulated so as to respect conservation properties of the governing equations. An important practical consideration is that satisfying conservation properties helps to ensure the computational stability of a model. Apart from this, the direct physical realism of a conservation property may be a desirable feature. For example, ensuring conservation of mass prevents the surface pressure from drifting to unrealistic values in long-term integrations of atmospheric models. Advection schemes which satisfy an appropriate dynamical conservation property may help to ensure the realism of the simulated energy spectrum. There are, however, considerations other than conservation that might influence the choice of numerical scheme. Shape-preservation (avoidance of the generation of spurious maxima or minima) may be considered as an important feature of an advection scheme, and the economy of a method (especially the ability to accomodate long time steps) may be a critical factor. Indeed, semi-Lagrangian advection schemes, generally without formal conservation properties, are increasingly being developed for numerical weather prediction.

7.6.1 The shallow-water equations with non-linear advection terms

The shallow-water equations with non-linear advection terms will, following [Sadourny \(1975\)](#), now be presented. The momentum equations in vector form can in this case be written as

$$\frac{\partial \mathbf{V}}{\partial t} + \mathbf{V} \cdot \nabla \mathbf{V} + f \mathbf{k} \times \mathbf{V} = -g \nabla h, \quad (7.30)$$

which can also be expressed as

$$\frac{\partial \mathbf{V}}{\partial t} + \xi \mathbf{k} \times (h \mathbf{V}) = -\nabla \left(gh + \frac{1}{2} \mathbf{V} \cdot \mathbf{V} \right). \quad (7.31)$$

The continuity equation with non-linear terms is

$$\frac{\partial h}{\partial t} + \nabla \cdot (h \mathbf{V}) = 0, \quad (7.32)$$

where \mathbf{V} is the horizontal velocity vector, $f \equiv 2\Omega \sin(\Phi)$ the Coriolis parameter, $\xi \equiv \left(f + \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \right) / h$ the horizontal absolute potential vorticity, \mathbf{k} the unit vector normal to the plane domain S , and h the total

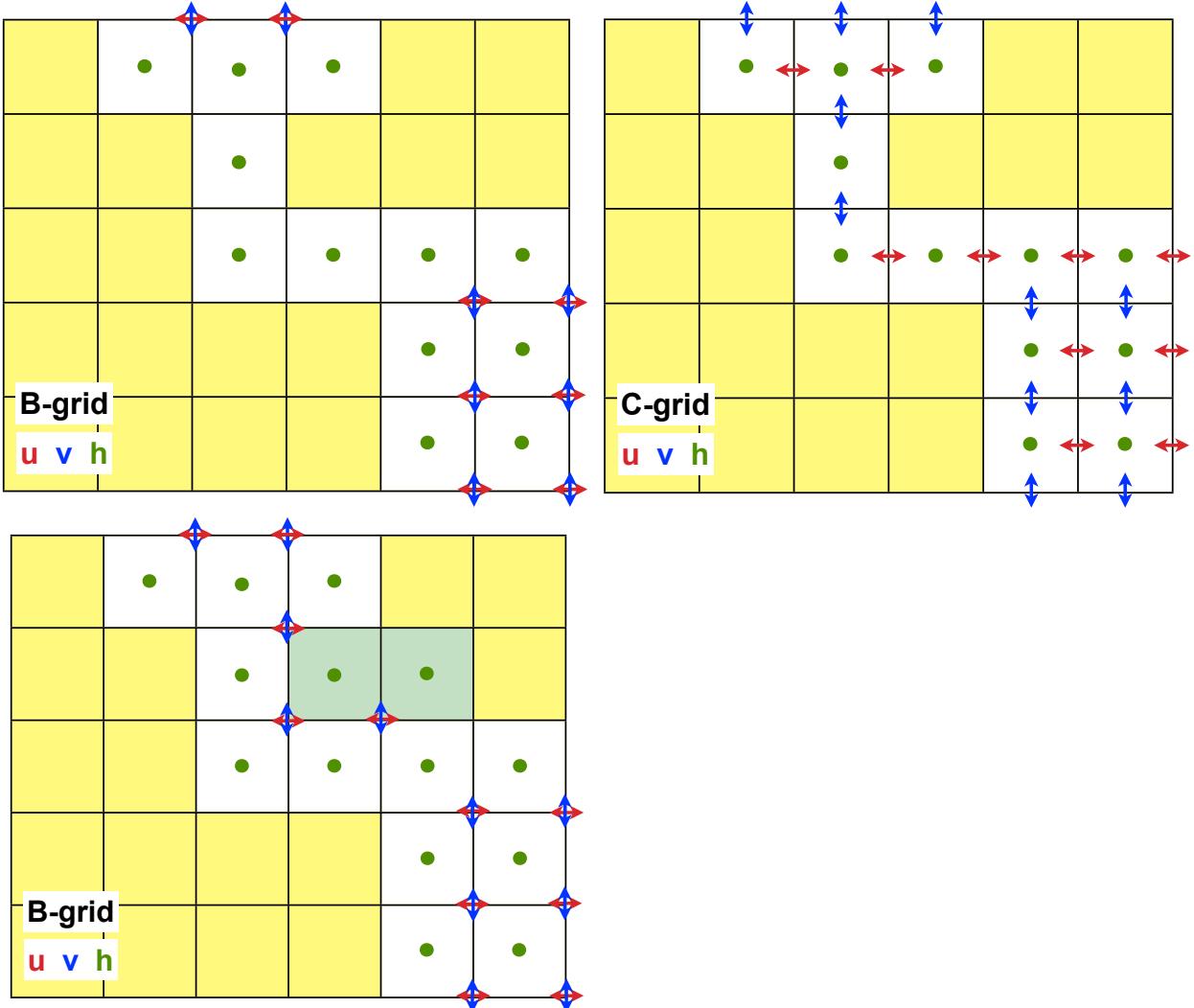


Figure 7.8: Illustration of how a narrow strait is resolved with a B-grid (top left panel) and C-grid (top right panel). Land cells in yellow, ocean cells in white with corresponding u , v and h points in colour. The only way to enable velocity points in this B-grid would be to "dig" out (two cells in green) so the strait is at least two grid cells wide (bottom left panel).

water- or air-column height.

The equations can also be written on scalar form

$$\frac{\partial u}{\partial t} - \xi h v = -\frac{\partial B}{\partial x} \quad (7.33a)$$

$$\frac{\partial v}{\partial t} + \xi h u = -\frac{\partial B}{\partial y} \quad (7.33b)$$

$$\frac{\partial h}{\partial t} + \frac{\partial (hu)}{\partial x} + \frac{\partial (hv)}{\partial y} = 0, \quad (7.33c)$$

where $B \equiv gh + \frac{1}{2}(u^2 + v^2) = gh + \frac{1}{2}\mathbf{V} \cdot \mathbf{V}$. It can be verified that these equations are such that the following properties are conserved:

- a) Total mass: $M = \int_S h dS$
- b) Total Energy: $E = \int_S \frac{1}{2}(gh + \mathbf{V} \cdot \mathbf{V}) h dS$
- c) Absolute potential enstrophy: $Z = \int_S \frac{1}{2}\xi^2 h dS$

Enstrophy is the integral of the vorticity and can be interpreted as a quantity directly related to the kinetic energy corresponding to dissipation effects.

7.6.2 Discretisation

The discretisation on a C-grid is illustrated in Figure 13.3. Fel referens?????

The spatial differencing operators are

$$\delta_x u = \frac{1}{\Delta x} (u_{i+1/2,j} - u_{i-1/2,j}) \rightarrow \frac{u_{i,j} - u_{i-1,j}}{\Delta x}$$

$$\delta_y v = \frac{1}{\Delta y} (v_{i,j+1/2} - v_{i,j-1/2}) \rightarrow \frac{v_{i,j} - v_{i,j-1}}{\Delta y}$$

$$\bar{u}^x = \frac{1}{2} (u_{i+1/2,j} + u_{i-1/2,j}) \rightarrow \frac{1}{2} (u_{i,j} + u_{i-1,j})$$

$$\bar{v}^y = \frac{1}{2} (v_{i,j+1/2} + v_{i,j-1/2}) \rightarrow \frac{1}{2} (v_{i,j} + v_{i,j-1})$$

The mass fluxes U and V are defined at the same points as the velocities u and v :

$$U_{i,j} \equiv \bar{h}^x u = u_{i,j} \frac{1}{2} (h_{i,j} + h_{i+1,j})$$

$$V_{i,j} \equiv \bar{h}^y v = v_{i,j} \frac{1}{2} (h_{i,j} + h_{i,j+1})$$

The gradient operator will act on the Bernoulli function B defined at the locations where h is defined:

$$B \equiv gh + \frac{1}{2} (\bar{u}^{2x} + \bar{v}^{2y}) = gh_{i,j} + \frac{1}{2} \left[\frac{1}{2} (u_{i,j}^2 + u_{i-1,j}^2) + \frac{1}{2} (v_{i,j}^2 + v_{i,j-1}^2) \right]$$

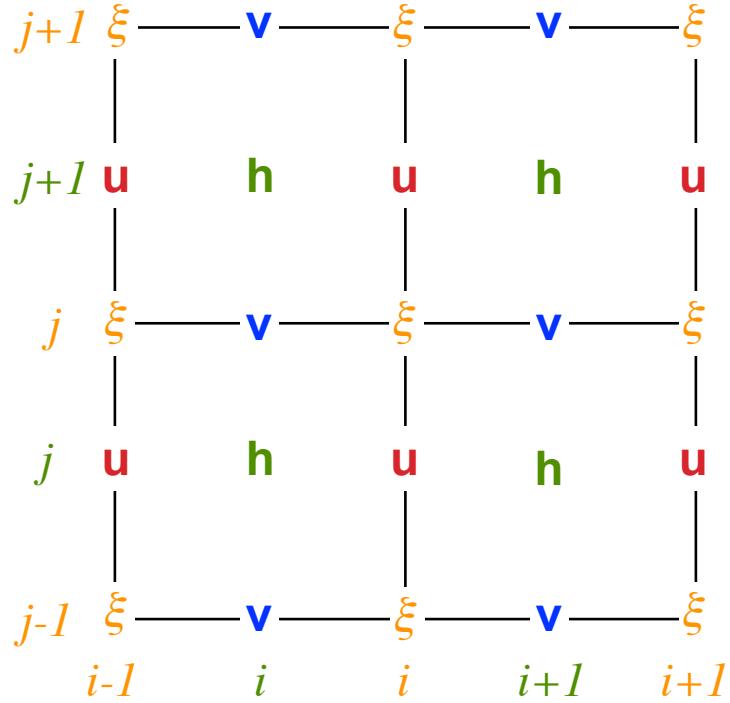


Figure 7.9: *C-grid with points for the zonal velocity u , meridional velocity v , water or air column height h and vorticity ξ .*

The potential absolute vorticity is redefined and located in the corners of the C-grid

$$\xi_{i,j} = \frac{f + \delta_x v - \delta_y u}{\bar{h}^{xy}} = \frac{f + \frac{v_{i+1,j} - v_{i,j}}{\Delta x} - \frac{u_{i,j+1} - u_{i,j}}{\Delta y}}{\frac{1}{4}(h_{i,j} + h_{i+1,j} + h_{i,j+1} + h_{i+1,j+1})}$$

Simple expressions are chosen for

- a) Total mass: $M = \sum h$
- b) Total energy: $E = \frac{1}{2} \sum (gh^2 + h\bar{u}^2 + h\bar{v}^2)$
- c) Absolute potential enstrophy: $Z = \frac{1}{2} \sum \xi^2 \bar{h}^{xy}$

Here the symbol \sum refers to a summation of the same species over all grid points. Note that due to symmetry

$$\sum ab^x = \sum b\bar{a}^x$$

and that due to skew-symmetry

$$\sum a\delta_x b = - \sum b\delta_x a$$

The time derivative of the total energy is

$$\frac{dE}{dt} = \sum \left(U \frac{\partial u}{\partial t} + V \frac{\partial v}{\partial t} + B \frac{\partial h}{\partial t} \right) \quad (7.34)$$

A simple energy-conserving model can be defined as

$$\begin{aligned} \frac{\partial u}{\partial t} - \bar{\xi} \bar{V}^{xy} + \delta_x B &= 0 \\ \frac{\partial v}{\partial t} + \bar{\xi} \bar{U}^{yx} + \delta_y B &= 0 \\ \frac{\partial h}{\partial t} + \delta_x U + \delta_y V &= 0 \end{aligned}$$

This can also be formulated in a longer way using indicial notation:

$$\frac{\partial u_{i,j}}{\partial t} = \frac{1}{2} \left[\xi_{i,j} \frac{1}{2} (V_{i,j} + V_{i+1,j}) + \xi_{i,j-1} \frac{1}{2} (V_{i,j-1} + V_{i+1,j-1}) \right] - \frac{B_{i+1,j} - B_{i,j}}{\Delta x} \quad (7.35a)$$

$$\frac{\partial v_{i,j}}{\partial t} = \frac{1}{2} \left[\xi_{i,j} \frac{1}{2} (U_{i,j} + U_{i,j+1}) + \xi_{i-1,j} \frac{1}{2} (U_{i-1,j} + U_{i-1,j+1}) \right] - \frac{B_{i,j+1} - B_{i,j}}{\Delta y} \quad (7.35b)$$

$$\frac{\partial h_{i,j}}{\partial t} = - \frac{U_{i+1,j} - U_{i,j}}{\Delta x} - \frac{V_{i,j+1} - V_{i,j}}{\Delta y} \quad (7.35c)$$

Energy conservation can be obtained from Equation (7.34), and we find that

$$\frac{dE}{dt} = \sum \left(U \bar{\xi} \bar{V}^{xy} - V \bar{\xi} \bar{U}^{yx} \right) + \sum (-U \delta_x B - B \delta_x U) + \sum (-V \delta_y B - B \delta_y V) = 0$$

where each of the three summations cancel out due to the symmetry or skew-symmetry of the operators.

An absolute potential enstrophy model can be defined as

$$\frac{\partial u}{\partial t} - \bar{\xi}^y \bar{V}^{xy} + \delta_x B = 0 \quad (7.36a)$$

$$\frac{\partial v}{\partial t} + \bar{\xi}^x \bar{U}^{yx} + \delta_y B = 0 \quad (7.36b)$$

$$\frac{\partial h}{\partial t} + \delta_x U + \delta_y V = 0 \quad (7.36c)$$

In the corresponding vorticity equation, the discretised gradient vanishes ???KOLLA en eller flera $\delta_x \delta_y = \delta_y \delta_x$, so that

$$\frac{\partial}{\partial t} \left(\bar{\xi} \bar{h}^{xy} \right) + \delta_x \left(\bar{\xi}^x \bar{U}^{yx} \right) + \delta_y \left(\bar{\xi}^y \bar{V}^{xy} \right) = 0$$

which when combined with the average continuity equation

$$\frac{\partial}{\partial t} \left(\bar{h}^{xy} \right) + \delta_x \left(\bar{U}^{yx} \right) + \delta_y \left(\bar{V}^{xy} \right) = 0$$

yields the conservative potential enstrophy equation

$$\frac{\partial}{\partial t} \left(\xi^2 \bar{h}^{xy} \right) + \delta_x \left(\xi^2 \bar{U}^{yx} \right) + \delta_y \left(\xi^2 \bar{V}^{xy} \right) = 0$$

???? menig till

7.7 A shallow-water model

We will here summarise the results above by formulating a model in the way it is programmed in computer code of the discretised shallow-water equations on a C-grid subject to the required conditions. This will hence be close to how a Fortran or C++ code is structured. The following steps will be taken:

- Set the initial condition of the fields for $n = 0$ over the entire model grid indices i and j so that

$$u_{i,j}^{n=0}, \quad v_{i,j}^{n=0}, \quad h_{i,j}^{n=0}$$

are known.

- Integrate the shallow-water equations a first time step with an Euler-forward scheme and loop over all the model grid indices i and j :

$$\begin{aligned} u_{i,j}^1 &= u_{i,j}^0 + \Delta t \left[-g \frac{h_{i+1,j}^0 - h_{i,j}^0}{\Delta x} + \frac{f}{4} (v_{i,j}^0 + v_{i+1,j}^0 + v_{i+1,j-1}^0 + v_{i,j-1}^0) \right] \\ v_{i,j}^1 &= v_{i,j}^0 + \Delta t \left[-g \frac{h_{i,j+1}^0 - h_{i,j}^0}{\Delta y} - \frac{f}{4} (u_{i,j}^0 + u_{i,j+1}^0 + u_{i-1,j+1}^0 + u_{i-1,j}^0) \right] \\ h_{i,j}^1 &= h_{i,j}^0 - \Delta t H \left(\frac{u_{i,j}^0 - u_{i-1,j}^0}{\Delta x} + \frac{v_{i,j}^0 - v_{i,j-1}^0}{\Delta y} \right) \end{aligned}$$

- Time integrate the model from $n = 1$ to $n = N_t$, where N_t is the total number of time steps to be computed, which are going to be computed so that the total time integration will be $N_t \Delta t$. Leap-frog the time step with loops over i and j .

$$\begin{aligned} u_{i,j}^{n+1} &= u_{i,j}^{n-1} + 2\Delta t \left[-g \frac{h_{i+1,j}^n - h_{i,j}^n}{\Delta x} + \frac{f}{4} (v_{i,j}^n + v_{i+1,j}^n + v_{i+1,j-1}^n + v_{i,j-1}^n) \right] \\ v_{i,j}^{n+1} &= v_{i,j}^{n-1} + 2\Delta t \left[-g \frac{h_{i,j+1}^n - h_{i,j}^n}{\Delta y} - \frac{f}{4} (u_{i,j}^n + u_{i,j+1}^n + u_{i-1,j+1}^n + u_{i-1,j}^n) \right] \\ h_{i,j}^{n+1} &= h_{i,j}^{n-1} - 2\Delta t H \left(\frac{u_{i,j}^n - u_{i-1,j}^n}{\Delta x} + \frac{v_{i,j}^n - v_{i,j-1}^n}{\Delta y} \right) \end{aligned}$$

- Apply a Robert-Asselin filter in order to suppress the computational mode:

$$\begin{aligned} u_{i,j}^n &= u_{i,j}^n + \gamma (u_{i,j}^{n-1} - 2u_{i,j}^n + u_{i,j}^{n+1}) \\ v_{i,j}^n &= v_{i,j}^n + \gamma (v_{i,j}^{n-1} - 2v_{i,j}^n + v_{i,j}^{n+1}) \\ h_{i,j}^n &= h_{i,j}^n + \gamma (h_{i,j}^{n-1} - 2h_{i,j}^n + h_{i,j}^{n+1}) \end{aligned}$$

- Store the resulting fields at regular time intervals and compute some statistics, e.g. the total volume

V , the kinetic energy E_K , and available potential energy E_P

$$\begin{aligned} V &= \sum_{i=1}^{N_X} \sum_{j=1}^{N_Y} h_{i,j}^n \Delta x \Delta y \\ E_P &= \frac{g}{2} \sum_{i=1}^{N_X} \sum_{j=1}^{N_Y} (h_{i,j}^n)^2 \Delta x \Delta y \\ E_K &= \frac{H}{2} \sum_{i=1}^{N_X} \sum_{j=1}^{N_Y} \left[(u_{i,j}^n)^2 + (v_{i,j}^n)^2 \right] \Delta x \Delta y \end{aligned}$$

- f) To economise disk space we switch the time steps (since we only store three time steps) before returning to the beginning of the time loop so that $n \rightarrow n - 1$ and $n + 1 \rightarrow n$ and

$$\begin{aligned} u_{i,j}^{n-1} &= u_{i,j}^n, & v_{i,j}^{n-1} &= v_{i,j}^n, & h_{i,j}^{n-1} &= h_{i,j}^n \\ u_{i,j}^n &= u_{i,j}^{n+1}, & v_{i,j}^n &= v_{i,j}^{n+1}, & h_{i,j}^n &= h_{i,j}^{n+1} \end{aligned}$$

- g) End the time loop of the model and the entire model code

This shallow-water model can be extended to include terms representing non-linear advection and friction/viscosity, both of which are to be derived in the what follows.

Chapter 8

Diffusion and friction terms

In this chapter we will investigate the discretisation of friction and diffusion terms and how this affects the stability of the solution. Diffusion and friction terms are included in most models from very simple ones based on the shallow-water equations to highly complex ocean-atmosphere general circulation models.

8.1 Rayleigh friction

We start by studying the simplest type of friction parameterisation, Rayleigh friction, where the retarding acceleration is directly proportional to the velocity. A straightforward example is given by

$$\frac{\partial u}{\partial t} = -\gamma u \quad (= f), \quad (8.1)$$

where $\gamma > 0$ and the solution is

$$u(t) = u_0 e^{-\gamma t}. \quad (8.2)$$

Using a centred time difference (i.e. a leap-frog scheme), which is the most common one employed in equations with advection terms, the discretisation of Equation (8.1) is

$$\frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t} = -\gamma u_j. \quad (8.3)$$

When a stability analysis is undertaken (in analogy with the one the advection equation was subjected to in the previous chapter) with $u_j^{n+m} = u_j^n \lambda^m$ one finds that

- a) if the right-hand side of Equation (8.3) is taken at time step n , i.e. $f^n = -\gamma u_j^n$, then $\lambda_{1,2} = -\gamma\Delta t \pm \sqrt{1 + (\gamma\Delta t)^2}$, which has at least one root that is always greater than one for any $\gamma\Delta t > 0$. The scheme is hence unconditionally unstable.
- b) if the right-hand side of Equation (8.3) is taken at time step $n-1$, i.e. $f^{n-1} = -\gamma u_j^{n-1}$, then $\lambda^2 = 1 - 2\gamma\Delta t$. The scheme is conditionally stable since $\lambda^2 \leq 1$ if $\gamma\Delta t \leq 1$. But since $\lambda^2 < 0$ for $1/2 < \gamma\Delta t < 1$, the roots of λ will be purely imaginary and the solution of u will oscillate and change sign for every second time step. If e.g. $\gamma\Delta t = 1$ then $\lambda_{1,2} = \pm i$ and $u^n = i^n = 1, 0, -1, 0, 1, \dots$

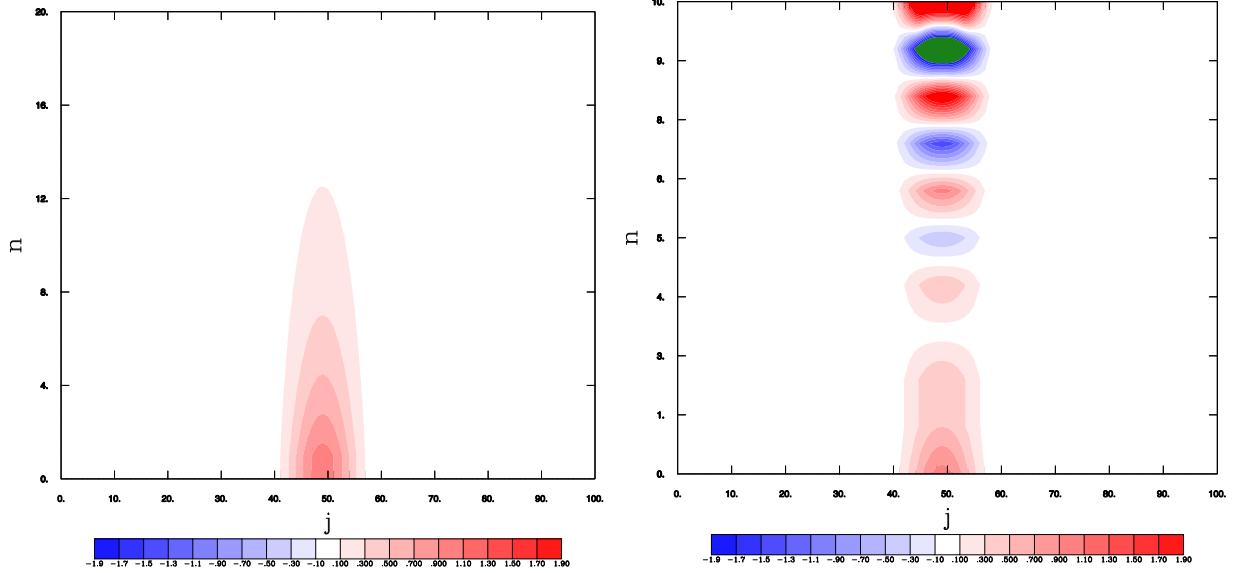


Figure 8.1: The Rayleigh friction Equation (8.3) integrated analytically (left figure) and numerically (right figure) with Equation (8.3) and the right-hand side at time step n (case 1) with $\gamma\Delta t = 0.5$, which clearly gives an unstable solution

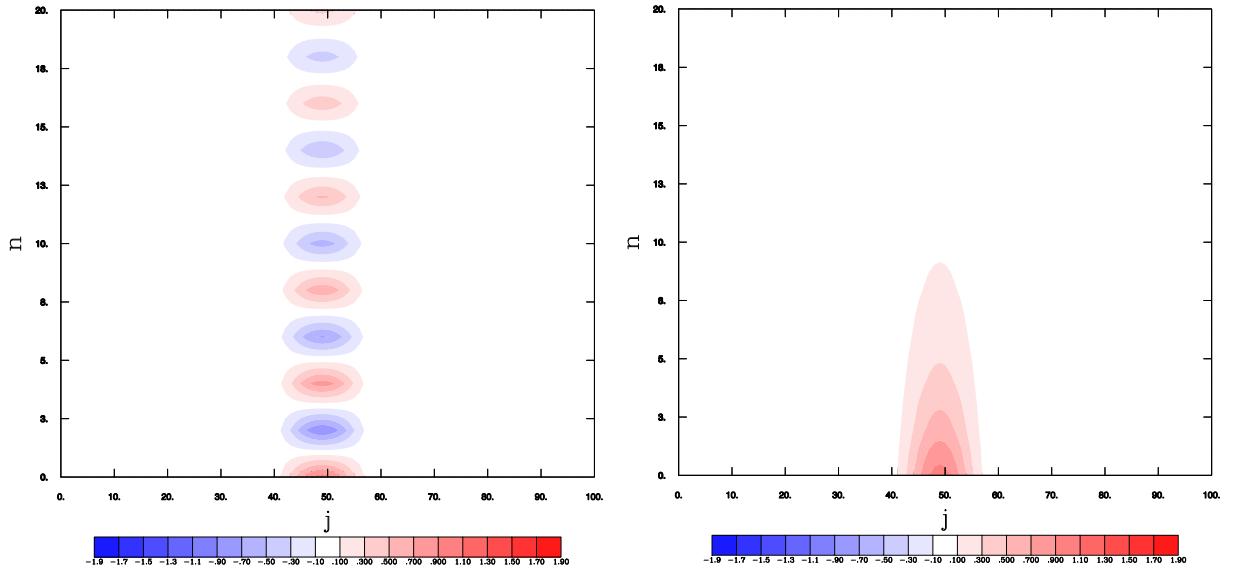


Figure 8.2: The Rayleigh friction Equation (8.3) integrated with (2) i.e. the right-hand side at time step $n - 1$. Left figure with $\gamma\Delta t = 0.95$ and right figure with $\gamma\Delta t = 0.2$.

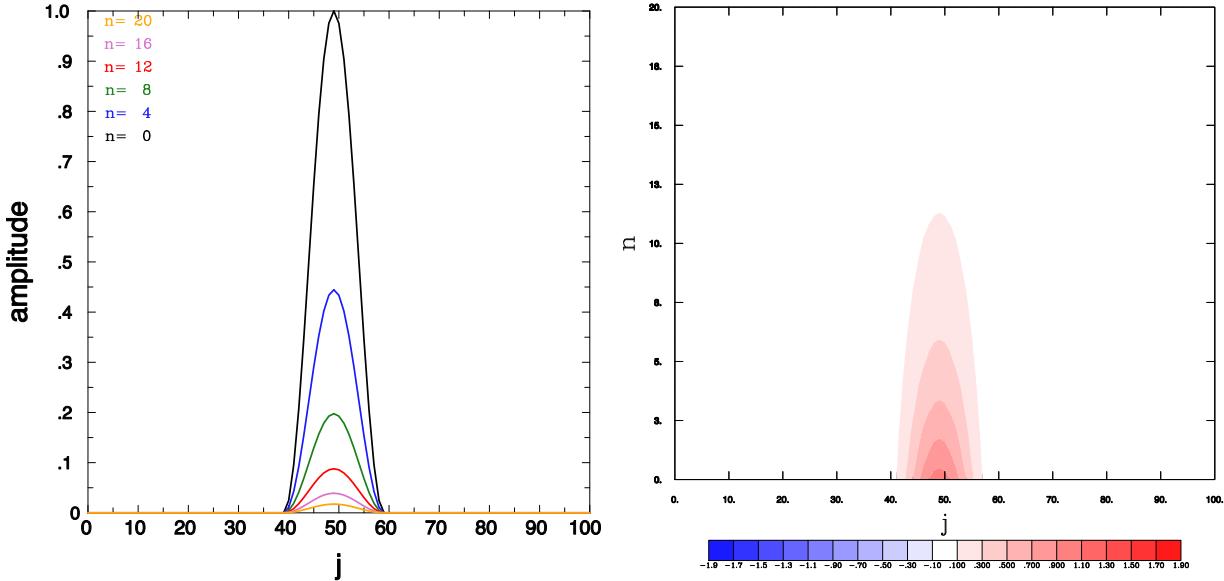


Figure 8.3: The Rayleigh-friction equation (8.3) integrated with the Crank-Nicholson scheme, i.e. the right-hand side taken at time steps $n - 1$ and $n + 1$ with $\gamma\Delta t = 0.2$.

For the more restricted condition $\gamma\Delta t < 1/2$, λ will be real and u will have a more realistic evolution in time with no numerical oscillations.

- c) if the right-hand side of Equation (8.3) is taken as an average of $n - 1$ and $n + 1$, the following finite-difference equation is obtained:

$$\frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t} = -\frac{\gamma}{2} \left(u_j^{n+1} + u_j^{n-1} \right).$$

This is known as the Crank-Nicholson scheme and is said to be implicit because it includes a term at time level $n + 1$ on the right-hand side of Equation (8.3). It yields the best approximation of Equation (8.2), and the stability analysis results in $\lambda^2 = \frac{1-\gamma\Delta t}{1+\gamma\Delta t} < 1$ for all values of $\gamma\Delta t$. The scheme is hence unconditionally stable. For the same reasons as above one requires $\gamma\Delta t < 1$ in order for a realistic evolution in time. Implicit schemes are often complicated to solve since they include values on both sides of the equation that need to be determined simultaneously. This is, however, not so in this particular case, since the right-hand side is evaluated at the same spatial grid point j as the left-hand side and the equation can be rearranged so that

$$u_j^{n+1} = \frac{1 - \gamma\Delta t}{1 + \gamma\Delta t} u_j^{n-1}.$$

When, as in this case, employing the Crank-Nicholson scheme for only a time integration, one should use a two-time-step integration with an Euler-forward scheme so that only two time steps are used

and the equation becomes becomes

$$u_j^{n+1} = \frac{1 - \gamma \Delta t / 2}{1 + \gamma \Delta t / 2} u_j^n.$$

The stability analysis above is strictly only valid for these discretisations of the very simple Rayleigh friction Equation (8.1). However, it turns out that one obtains approximately the same stability criteria, when a Rayleigh friction term is included in the momentum equations or the tracer equations in a GCM. It is, however, not possible then to perform a stability analysis on these more comprehensive equations.

8.2 Laplacian friction

A somewhat more realistic friction parameterisation is related to Laplacian diffusion, which is included in the heat equation:

$$\frac{\partial u}{\partial t} = A \frac{\partial^2 u}{\partial x^2}, \quad (8.4)$$

where A is the diffusion coefficient with the unit m^2/s . Often the letter K is used, when representing a diffusion coefficient and A when it is used in the momentum equations as viscosity coefficient. The letter A originates from the German word *Austausch*, which means "exchange", referring to the exchange of water parcels. It replaces the molecular viscosity with a much larger eddy viscosity in order to parameterise the sub-grid scales. Equation (8.4) has has the solution

$$u(x, t) = u_0 e^{\pm i k x - A k^2 t}. \quad (8.5)$$

The simplest way to construct a finite difference of a second-order derivative is to apply finite differencing to a finite difference. This is achieved by first postulating two finite differences centred on the intermediate positions $j + 1/2$ and $j - 1/2$ as illustrated by Fig. 8.4:

$$\left(\frac{du}{dx} \right)_{j-1/2} \approx \frac{u_j - u_{j-1}}{\Delta x}, \quad (8.6)$$

$$\left(\frac{du}{dx} \right)_{j+1/2} \approx \frac{u_{j+1} - u_j}{\Delta x}. \quad (8.7)$$

Since the second-order derivative is defined as the derivative of the derivative, we can similarly construct a further finite difference:

$$\left(\frac{d^2 u}{dx^2} \right)_j \equiv \left[\frac{d}{dx} \left(\frac{du}{dx} \right) \right]_j \approx \frac{\frac{u_{j+1} - u_j}{\Delta x} - \frac{u_j - u_{j-1}}{\Delta x}}{\Delta x} = \frac{u_{j+1} - 2u_j + u_{j-1}}{(\Delta x)^2}. \quad (8.8)$$

The advantage of this formulation is that it is straightforward and intuitively evident. The disadvantage is that it does not provide an estimate of the accuracy of the scheme. To obtain this we use the Taylor-series method previously employed in Section 3.2. A centred finite difference of the Laplace operator

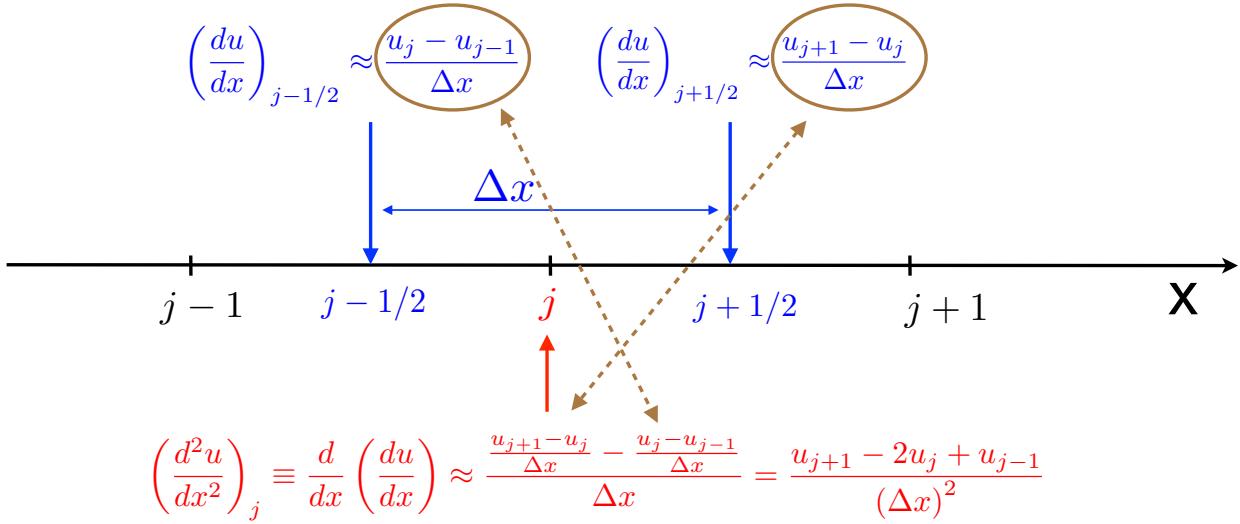


Figure 8.4: The second derivative is the derivative of the derivative. By first estimating the finite differences at $j + 1/2$ and $j - 1/2$ and then the finite difference of those two, one obtains the second finite difference at j .

corresponding to the second order derivative can hence be obtained by combining two Taylor series:

$$u_{j+1} = u_j + \Delta x \left(\frac{du}{dx}\right)_j + \frac{1}{2} (\Delta x)^2 \left(\frac{d^2u}{dx^2}\right)_j + \frac{1}{6} (\Delta x)^3 \left(\frac{d^3u}{dx^3}\right)_j + \frac{1}{24} (\Delta x)^4 \left(\frac{d^4u}{dx^4}\right)_j + \dots \quad (8.9)$$

$$u_{j-1} = u_j - \Delta x \left(\frac{du}{dx}\right)_j + \frac{1}{2} (\Delta x)^2 \left(\frac{d^2u}{dx^2}\right)_j - \frac{1}{6} (\Delta x)^3 \left(\frac{d^3u}{dx^3}\right)_j + \frac{1}{24} (\Delta x)^4 \left(\frac{d^4u}{dx^4}\right)_j - \dots \quad (8.10)$$

Summing these two equations and dividing by $(\Delta x)^2$ we obtain

$$\frac{u_{j+1} - 2u_j + u_{j-1}}{(\Delta x)^2} = \frac{d^2u}{dx^2} + \frac{1}{12} (\Delta x)^2 \left(\frac{d^4u}{dx^4}\right)_j + \dots \quad (8.11)$$

This finite-difference approximation of the second-order derivative is hence accurate to order $(\Delta x)^2$, which is the same as saying it has a second-order truncation error.

Equation (8.4) can now be approximated by integrating in time with a leap-frog scheme:

$$\frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t} = A \frac{u_{j+1} - 2u_j + u_{j-1}}{(\Delta x)^2}. \quad (8.12)$$

A stability analysis using the von Neumann method is undertaken by inserting $u_j^n = u_0 \lambda^n e^{ikj\Delta x}$ into this equation. Different numerical results are obtained depending on at which time step the right-hand side is chosen. Let us study the same three cases as we did in the Rayleigh-friction example:

- a) If the right-hand side of Equation (8.12) is taken at time step n , the equation for the amplification factor becomes

$$\lambda^2 + \frac{8A\Delta t}{(\Delta x)^2} \sin^2\left(\frac{k\Delta x}{2}\right) \lambda - 1 = 0,$$

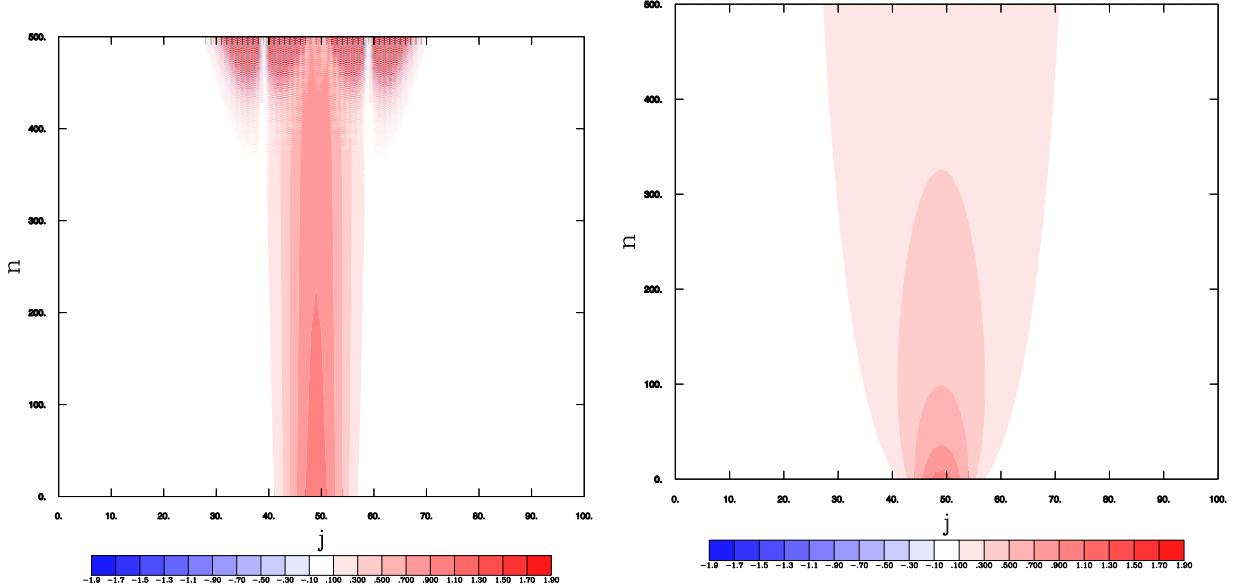


Figure 8.5: The heat equation integrated numerically using Equation (8.12) with the right-hand side at time step n and with $\frac{A\Delta t}{(\Delta x)^2} = 0.01$ (left figure) and in the right figure with the right-hand side at time step at $n-1$ and $\frac{A\Delta t}{(\Delta x)^2} = 0.125$.

which has the roots $\lambda_{1,2} = -a \pm \sqrt{1 + a^2}$, where $a \equiv \frac{4A\Delta t}{(\Delta x)^2} \sin^2\left(\frac{k\Delta x}{2}\right)$. The second root will be $\lambda_2 < -1$ for any $\frac{A\Delta t}{(\Delta x)^2} > 0$, implying that the scheme is unconditionally unstable.

- b) If the right-hand side of Equation (8.12) is taken at time step $n-1$, the amplification-factor equation becomes

$$\lambda^2 = 1 - \frac{8A\Delta t}{(\Delta x)^2} \sin^2\left(\frac{k\Delta x}{2}\right).$$

The scheme is stable when $-1 \leq \lambda^2 \leq 1$, which is the case when $\frac{A\Delta t}{(\Delta x)^2} < 1/4$, and thus the scheme is conditionally stable. However, for the same reasons as for the Rayleigh-friction equation, we recommend the stricter condition $\frac{A\Delta t}{(\Delta x)^2} < 1/8$, this in order to have $\lambda^2 > 0$ and hereby avoiding oscillations in time of the solution.

- c) If the right-hand side of Equation (8.12) is taken as an average of $n-1$ and $n+1$ (the Crank-Nicholson scheme) we have

$$\frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t} = \frac{A}{2} \left(\frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{(\Delta x)^2} + \frac{u_{j+1}^{n-1} - 2u_j^{n-1} + u_{j-1}^{n-1}}{(\Delta x)^2} \right). \quad (8.13)$$

This scheme is implicit as it includes terms at time step $n+1$ on the right-hand side of the equation, which, however, can not be solved as easily as in the Rayleigh-friction case, since the $n+1$ terms on the right-hand side occur at the spatial grid points $j-1, j, j+1$. It is, however, possible to use algorithms that eliminate these terms. We can nevertheless undertake a stability analysis and calculate

the amplification factor, which is found to be

$$\lambda^2 = \frac{1 - \frac{4A\Delta t}{(\Delta x)^2} \sin^2\left(\frac{k\Delta x}{2}\right)}{1 + \frac{4A\Delta t}{(\Delta x)^2} \sin^2\left(\frac{k\Delta x}{2}\right)}. \quad (8.14)$$

This is always smaller than one and the scheme is hence unconditionally stable. In order to avoid imaginary roots that lead to oscillating solutions one should, however, use $\frac{A\Delta t}{(\Delta x)^2} < 1/4$.

In most cases when modelling the atmosphere or the ocean, γ and A are of such magnitudes that the stability criterion in the present chapter permits Δt to be much larger (5-10 times) than that conforming to the CFL criterion $c < \frac{\Delta x}{\Delta t}$. The most common mistake when writing a code is, however, to use the unconditionally unstable scheme with the friction taken at time step n .

Note that the schemes in the two last cases discussed above are in fact two-level schemes, since we do not use any values at time step n , but only at $n - 1$ and $n + 1$. There is consequently no reason to use a leap-frog scheme here and we can instead use an Euler-forward scheme in time and replace all $n - 1$ by n . The stability analysis remains unaltered but, since the time step is halved, we should replace Δt by $\Delta t/2$. It is nevertheless easier to demonstrate the disparities between the three cases by using leap-frog schemes in all three cases.

In section 4.3 we saw that wave propagation with the discretised advection equation required restrictions on the Courant number $\frac{c\Delta t}{\Delta x}$. Here, we have seen that similar conditions arise for stability when Rayleigh friction and Laplacian diffusion are used. This leads to restrictions on the non-dimensional numbers $\gamma\Delta t$ and $\frac{A\Delta t}{(\Delta x)^2}$. The discretised momentum equation will need to satisfy all these stability criteria when friction parameterisations are included as well as the CFL criterion. The next section will examine how the advection and the diffusion will affect each other's stability criteria.

8.3 The advection-diffusion equation

Let us now examine an equation with both advection and diffusion terms:

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = A \frac{\partial^2 u}{\partial x^2}, \quad (8.15)$$

which has the analytical solution

$$u(x, t) = u_0 e^{\pm ik(x-ct)-Ak^2t}. \quad (8.16)$$

We have previously seen that a stable discretisation scheme for the advection equation is centred in time as well as in space, while for the diffusion equation the discretised Laplace operator must be taken at time step $n - 1$ in order to ensure stability. Let us now combine these schemes:

$$\frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t} + c \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} = A \frac{u_{j+1}^{n-1} - 2u_j^{n-1} + u_{j-1}^{n-1}}{(\Delta x)^2}. \quad (8.17)$$

We now undertake a stability analysis by inserting $u_j^n = \lambda^n e^{ikj\Delta x}$ into this Equation (8.17), which after some calculation yields

$$\lambda^2 + 2ia\lambda + b - 1 = 0, \quad (8.18)$$

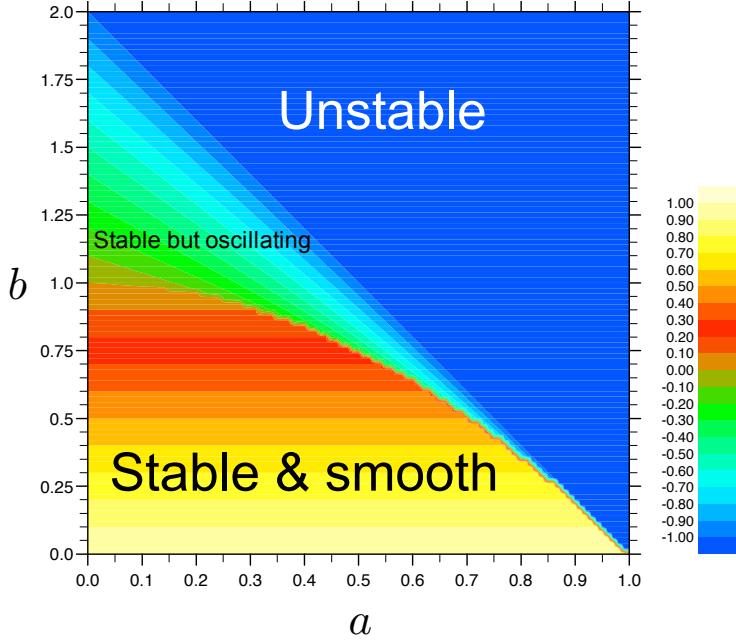


Figure 8.6: The stability function λ^2 as a function of $a \equiv \frac{c\Delta t}{\Delta x} \sin(k\Delta x)$ and $b \equiv \frac{8A\Delta t}{(\Delta x)^2} \sin^2\left(\frac{k\Delta x}{2}\right)$. The region in dark blue is where $\lambda^2 < -1$ and corresponds to unstable solutions. The green - light blue region where $-1 < |\lambda|^2 < 0$ is for stable solutions but oscillating ones. The yellow-red-brown region is for $0 < \lambda^2 < 1$, which correspond stable solutions and no oscillations.

with the coefficents $a \equiv \frac{c\Delta t}{\Delta x} \sin(k\Delta x)$ and $b \equiv \frac{8A\Delta t}{(\Delta x)^2} \sin^2\left(\frac{k\Delta x}{2}\right)$. Note that a is in fact the Courant number multiplied by $\sin(k\Delta x)$.

The solution of Equation (8.18) is

$$\lambda = -ia \pm \sqrt{1 - b - a^2}.$$

If $1 - b - a^2 > 0$ then $|\lambda|^2 = a^2 + 1 - b - a^2 = 1 - b < 1$, viz. the scheme is stable for this root.

If $1 - b - a^2 < 0$ then $\lambda = -i(a \mp \sqrt{a^2 + b - 1}) \Rightarrow \lambda^2 = -(2a^2 + b - 1 \mp \sqrt{a^2 + b - 1})$

It is not obviously discernible when this second root yields a stable solution, and thus we have graphed λ^2 as a function of a and b in Figure 8.6.

The above stability analysis can now be tested by integrating Equation (8.17):

$$u_j^{n+1} = u_j^{n-1} - \frac{c\Delta t}{\Delta x} (u_{j+1}^n - u_{j-1}^n) + \frac{2A\Delta t}{(\Delta x)^2} (u_{j+1}^{n-1} - 2u_j^{n-1} + u_{j-1}^{n-1}), \quad (8.19)$$

which we then integrate numerically for 100 time steps with the same initial condition as for the Rayleigh and diffusion equations in previous sections. The choice of the Courant number $\frac{c\Delta t}{\Delta x}$ and the Laplacian diffusion stability number $\frac{A\Delta t}{(\Delta x)^2}$ will determine the stability of the integration. From Figure 8.6 we can see that a stable and non-oscillating solution will require a Courant number $(\frac{c\Delta t}{\Delta x})$ below 1. If we choose e.g. $\frac{2A\Delta t}{(\Delta x)^2} = \frac{1}{8}$, then $b = \frac{1}{2}$ and we recognise from Figure 8.6 that $|\lambda|^2 < 1$ for a up to approximately 0.72. To

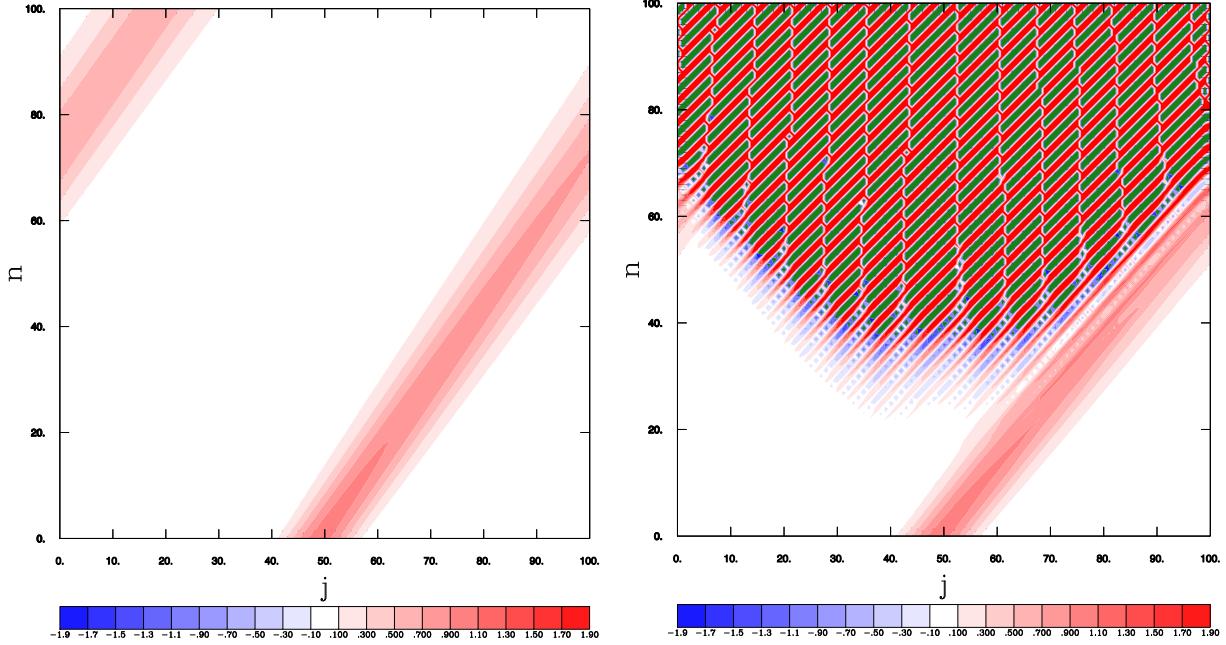


Figure 8.7: The heat-diffusion equation (8.19) integrated numerically using Equation (8.19) with $\frac{2A\Delta t}{(\Delta x)^2} = \frac{1}{8}$ and $\frac{c\Delta t}{\Delta x} = 0.7$ in the left-hand panel and $\frac{c\Delta t}{\Delta x} = 0.8$ in the right-hand panel.

test this we have integrated Equation (8.19) with a Courant number $\frac{c\Delta t}{\Delta x}$ above and below. Figure 8.7 shows these two integrations, with a stable solution obtained for $\frac{c\Delta t}{\Delta x} = 0.7$ and instability for $\frac{c\Delta t}{\Delta x} = 0.8$.

Exercises

- a) Perform a stability analysis for the Rayleigh friction equation with the right-hand side of Equation (8.3) taken at the time step n
- b) Same as in a) but taken at the time step $n - 1$
- c) Same as in a) but taken at the time step $n + 1$
- d) Calculate the stability criterion for

$$\frac{\partial u}{\partial t} = A \frac{\partial^2 u}{\partial x^2} \quad \text{where } A > 0$$

with

$$\frac{u_j^{n+1} - u_j^n}{2\Delta t} = A \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{(\Delta x)^2}$$

Estimate an upper limit for Δt for

- a) $A = 10^6 m^2/s$, $\Delta x = 400 km$ (large scale horizontal diffusion)
- b) $A = 1 m^2/s$, $\Delta x = 10 m$ (vertical diffusion in a boundary layer)
- e) The diffusion equation can be integrated using the Crank-Nicholson scheme:

$$\frac{T_j^{n+1} - T_j^n}{\Delta t} = \frac{A}{2} \left[\frac{T_{j+1}^n - 2T_j^n + T_{j-1}^n}{(\Delta x)^2} + \frac{T_{j+1}^{n+1} - 2T_j^{n+1} + T_{j-1}^{n+1}}{(\Delta x)^2} \right]$$

Examine the stability of this scheme.

Chapter 9

Implicit and semi-implicit schemes

The time step permitted by the economical explicit schemes, twice that satisfying the CFL criterion, is still considerably shorter than that required for accurate integration of the quasi-geostrophic equations. Thus we will here consider implicit schemes, which have the pleasing property of being stable for any choice of time step.

9.1 Implicit versus explicit schemes, a simple example

For implicit schemes, the spatial terms are evaluated, at least partially, at the unknown time level. Let us consider one of the simplest possible examples by examining the one-dimensional diffusion equation (also known as the heat equation):

$$\frac{\partial T}{\partial t} - \alpha \frac{\partial^2 T}{\partial x^2} = 0.$$

This equation is discretised with centred spatial finite differences on an unstaggered A-grid and integrated in time with an Euler-forward scheme. In traditional explicit form we obtain

$$T_i^{n+1} = T_i^n + \alpha \frac{\Delta t}{\Delta x^2} [T_{i-1}^n - 2T_i^n + T_{i+1}^n],$$

which simplifies into

$$T_i^{n+1} = sT_{i-1}^n + (1 - 2s)T_i^n + sT_{i+1}^n \quad ; \quad s = \alpha \frac{\Delta t}{\Delta x^2},$$

This equation is explicit in terms of T_i^{n+1} , which is the value at the unknown time level, and is hence possible to solve. Similarly, but evaluating the spatial term at the unknown time level $n + 1$, yields the fully implicit discretisation

$$T_i^{n+1} = T_i^n + \alpha \frac{\Delta t}{\Delta x^2} [T_{i-1}^{n+1} - 2T_i^{n+1} + T_{i+1}^{n+1}],$$

which simplifies into

$$-sT_{i-1}^{n+1} + (1+2s)T_i^{n+1} - sT_{i+1}^{n+1} = T_i^n \quad \text{with } s = \alpha \frac{\Delta t}{\Delta x^2}.$$

To solve this equation one needs to consider all grid points i . In the present case, when we are dealing with the linearised heat equation, it is possible to express the problem as a linear system of equations $\mathbf{A}\bar{X} = \bar{B}$, where \mathbf{A} is a matrix, \bar{X} a vector given by the unknown values of T at time $n+1$, and \bar{B} a vector given by the known values of T :

$$\begin{bmatrix} (1+2s) & -s & & & \\ -s & (1+2s) & -s & & \\ \dots & \dots & \dots & & \\ & \dots & \dots & \dots & \\ -s & (1+2s) & -s & & \\ \dots & \dots & \dots & & \\ & \dots & \dots & & \\ -s & (1+2s) & & & \end{bmatrix} \begin{bmatrix} T_2^{n+1} \\ T_3^{n+1} \\ \dots \\ \dots \\ T_i^{n+1} \\ \dots \\ \dots \\ T_{I-1}^{n+1} \end{bmatrix} = \begin{bmatrix} T_2^n + sT_1^{n+1} \\ T_3^n \\ \dots \\ \dots \\ T_i^n \\ \dots \\ \dots \\ T_{I-1}^n + sT_I^{n+1} \end{bmatrix} \quad (9.1)$$

T_1^{n+1} and T_I^{n+1} are known from the *Dirichlet* boundary conditions. The solution at time level $n+1$ is determined by solving this system of equations. The implicit method is consequently very computationally demanding compared to the explicit method, but since it is unconditionally stable (as will be shown in next section) one can use larger time steps that do not necessarily satisfy the CFL criterion. Depending on the character of the matrix, different algorithms can be used to solve the linear system of equations above. In the present case, the matrix is tridiagonal, which is advantageous from a computational standpoint.

9.2 Semi-implicit schemes

Semi-implicit schemes evaluate the spatial derivative at an average of the time levels n and $n+1$ instead of only at $n+1$ as in the fully-implicit case. If $F(x, y, t)$ is a term comprising spatial derivatives of a given scalar $T(x, y, t)$, we can consider the following general expression for a discretised version of the equation for the time evolution of $T_{i,j}$:

$$\frac{T_{i,j}^{n+1} - T_{i,j}^n}{\Delta t} = (1-\beta)F_{i,j}^n + \beta F_{i,j}^{n+1}.$$

- $\beta = 0 \rightarrow$ Explicit scheme
- $\beta = 1 \rightarrow$ Fully implicit scheme
- $0 < \beta < 1 \rightarrow$ Semi-implicit scheme

A commonly used semi-implicit method is the Crank-Nicolson scheme, in which $\beta = 0.5$ and the time derivative is expressed with the usual Euler-forward scheme. The term comprising spatial derivatives is therefore centred at time level $n+1/2$, which in fact turns this scheme into a trapezoidal implicit scheme in

time. By carrying out a Taylor expansion around $(i, n+1/2)$, one can verify that this scheme is characterised by a second-order accuracy in time, which is an appreciable improvement with regard to the first-order accuracy of the Euler-forward explicit scheme.

9.2.1 The one-dimensional diffusion equation

The one-dimensional diffusion Equation (8.4) is usually associated with centred differencing in space. When using a semi-implicit time scheme it becomes

$$\frac{T_i^{n+1} - T_i^n}{\Delta t} = (1 - \beta) \frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{(\Delta x)^2} + \beta \frac{T_{i+1}^{n+1} - 2T_i^{n+1} + T_{i-1}^{n+1}}{(\Delta x)^2}. \quad (9.2)$$

The fully implicit scheme case ($\beta = 0.5$) results in a numerical precision of second order both in time and space and hence the truncation error is of $O(\Delta t^2, \Delta x^2)$.

9.2.2 Two-dimensional pure gravity waves

Let us now discretise the equations for two-dimensional gravity waves, viz. Equations (7.13) without the Coriolis terms, using the Crank-Nicolson scheme on a C-grid and an Euler-forward time integration:

$$u^{n+1} = u^n - \frac{g\Delta t}{2} (\delta_x h^n + \delta_x h^{n+1}), \quad (9.3a)$$

$$v^{n+1} = v^n - \frac{g\Delta t}{2} (\delta_y h^n + \delta_y h^{n+1}), \quad (9.3b)$$

$$h^{n+1} = h^n - \frac{H\Delta t}{2} (\delta_x u^n + \delta_y v^n + \delta_x u^{n+1} + \delta_y v^{n+1}). \quad (9.3c)$$

Here δ_x represents the spatial differencing operator. Applied to a scalar f , it has the form

$$\delta_x f = \frac{1}{\Delta x} (f_{i+1/2} - f_{i-1/2}).$$

Inserting the wave solutions

$$(u^n, v^n, h^n) = (u_0, v_0, h_0) \lambda^n e^{i(k\Delta x + l\Delta y)},$$

we find

$$\begin{aligned} u_0 (1 - \lambda) &= ig\Delta t (1 + \lambda) \frac{\sin(k\Delta x/2)}{\Delta x} h_0, \\ v_0 (1 - \lambda) &= ig\Delta t (1 + \lambda) \frac{\sin(l\Delta y/2)}{\Delta y} h_0, \\ h_0 (1 - \lambda) &= iH\Delta t (1 + \lambda) \left[\frac{\sin(k\Delta x/2)}{\Delta x} u_0 + \frac{\sin(l\Delta y/2)}{\Delta y} v_0 \right]. \end{aligned}$$

Elminating u_0 , v_0 and h_0 , we obtain a cubic equation for λ . Here $\lambda_1 = 1$ is associated with the stationary solution and

$$\lambda_{2,3} = \frac{1 - A \pm 2i\sqrt{A}}{1 + A} \quad \text{where} \quad A \equiv gH\Delta t^2 \left[\frac{\sin^2(k\Delta x/2)}{\Delta x^2} + \frac{\sin^2(l\Delta y/2)}{\Delta y^2} \right].$$

Examination of these latter roots show that they always yield amplification factors satisfying $|\lambda| = 1$, and thus the scheme is unconditionally stable. The scheme is also said to be "neutrally stable" since $|\lambda| = 1$ is just on the edge of stability. This example of an application of the Crank-Nicolson scheme shows the power of semi-implicit methods as they both decrease the temporal truncation error from $O(\Delta t)$ to $O(\Delta t^2)$ and make the scheme unconditionally stable.

By now studying the one-dimensional case, viz. assuming that the waves propagate along the x -axis ($l \equiv 0$), and using

$$\lambda = |\lambda| e^{-i\omega\Delta t}$$

as well as Δt deduced from the expression for A above, it is possible to express ω or c as a functions of A , k and Δx . It is then possible to compare the phase speed of the non-stationary solutions in the present explicit case with the phase speed (Equation ??) found for the explicit spatially centred scheme used in section 7.1:

$$\left| \frac{c_D^{[impl]}}{c_D^{[expl]}} \right| \equiv \left| \frac{\omega_D^{[impl]}}{\omega_D^{[expl]}} \right| = \left| \frac{1}{2\sqrt{A}} \arctan \left(\frac{2\sqrt{A}}{1-A} \right) \right|$$

Since $A \geq 0$, the right-hand term is always ≤ 1 , and thus $c_D^{[impl]} \leq c_D^{[expl]} \leq \sqrt{gH}$. Semi-implicit time differencing is therefore seen to result in a slowing down of gravity-waves speed even larger than that due to explicit spatially centered differencing.

To solve the system of Equations (9.3a), as will be done in next section, the quantities $\delta_x u^{n+1}$ and $\delta_y v^{n+1}$ can be eliminated from the third equation by applying the operators δ_x and δ_y to the first and second of these equations, respectively and substituting the results into the third equation. This yields an equation for the height h which can be solved using a number of standard methods. As seen in the previous section, it is possible to express and solve this equation as a linear matrix system involving each grid point of the domain. Another possible option is to use an iterative technique, e.g. a *relaxation* procedure: ???UTVECKLA OCH SKRIV OM????

- a) make a first guess h^{n+1} which is usually h^n
- b) at each of the grid points the value of h^{n+1} has to satisfy the equation
- c) the preceding step is repeated as many times as needed to make the change at every point less than some pre-assigned small value

9.3 The semi-implicit method of Kwizak and Robert

There is no advantage in using an implicit method for the terms representing advective and *Coriolis* acceleration in the governing equations of atmospheric and oceanic models. They are associated with slower phase

speeds, and should not require excessively small time steps for linear stability when calculated explicitly. Since the trapezoidal implicit scheme is a two-level scheme like the forward-backward scheme, it is convenient to use the Adams-Basforth scheme (REFERENS????) for this purpose (????). Kwizak and Robert (1971) chose, however, to use the leap-frog scheme. The usual procedure for solving the semi-implicit difference system for variables at time level $n + 1$ will be illustrated for the shallow water equations. These equations can be written in a compact form:

$$\frac{\partial u}{\partial t} = -g \frac{\partial h}{\partial x} + A_u, \quad (9.4a)$$

$$\frac{\partial v}{\partial t} = -g \frac{\partial h}{\partial y} + A_v, \quad (9.4b)$$

$$\frac{\partial h}{\partial t} = -H \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) + A_h, \quad (9.4c)$$

where A_u , A_v and A_h represent terms that were omitted in Equations (7.13) describing the propagation of pure gravity waves. This time we apply implicit differencing over a time interval $2\Delta t$ centered around time n for the terms containing spatial derivatives (by using $\beta = 0.5$ with time steps $n - 1$ and $n + 1$ rather than n and $n + 1$ as previously. Second-order centered schemes are used for spatial derivatives and the leapfrog scheme for the time derivative, and hence the discretised system is

$$u^{n+1} = u^{n-1} - g\Delta t (\delta_x h^{n-1} + \delta_x h^{n+1}) + 2\Delta t A_u^n, \quad (9.5a)$$

$$v^{n+1} = v^{n-1} - g\Delta t (\delta_y h^{n-1} + \delta_y h^{n+1}) + 2\Delta t A_v^n, \quad (9.5b)$$

$$h^{n+1} = h^{n-1} - H\Delta t (\delta_x u^{n-1} + \delta_y v^{n-1} + \delta_x u^{n+1} + \delta_y v^{n+1}) + 2\Delta t A_h^n. \quad (9.5c)$$

We now apply the operator δ_x to the first and δ_y to the second of these equations, and add the results. By introducing the notation

$$\delta_{xx} h = \delta_x (\delta_x h) \quad \text{and} \quad \delta_{yy} h = \delta_y (\delta_y h)$$

we obtain

$$(\delta_x u + \delta_y v)^{n+1} = (\delta_x u + \delta_y v)^{n-1} - g\Delta t [(\delta_{xx} + \delta_{yy}) h^{n-1} + (\delta_{xx} + \delta_{yy}) h^{n+1}] + 2\Delta t (\delta_x A_u + \delta_y A_v)^n$$

Substituting the right-hand side into ???9C??? Eqs. (9.5a), and defining the “finite difference Laplacian” by

$$\nabla_*^2 \equiv \delta_{xx} + \delta_{yy}$$

we find that

$$h^{n+1} = h^{n-1} - 2H\Delta t (\delta_x u + \delta_y v)^{n-1} + gH\Delta t^2 (\nabla_*^2 h^{n-1} + \nabla_*^2 h^{n+1}) + 2\Delta t [A_h - H\Delta t (\delta_x A_u + \delta_y A_v)]^n$$

By, in addition, introducing the definitions

$$F^{n-1} \equiv h^{n-1} - 2H\Delta t (\delta_x u + \delta_y v)^{n-1} + gH\Delta t^2 \nabla_*^2 h^{n-1}$$

$$G^n \equiv 2\Delta t [A_h - H\Delta t (\delta_x A_u + \delta_y A_v)]^n$$

this can be formulated as

$$h^{n+1} - gH\Delta t^2 \nabla_*^2 h^{n+1} = F^{n-1} + G^n \quad (9.6)$$

here the terms have been arranged to show that at time level n , the right-hand side is known at all grid points. Once the equation has been solved for the values h^{n+1} then u^{n+1} and v^{n+1} can be obtained directly from the Eqs. (9.5a). a,b????? The algebraic system (9.6) is an elliptic PDE that resembles the *Helmholtz* equation:

$$\nabla^2 h + ah + b(x, y) = 0$$

Several methods (???REFERENSER???) are available to solve this standard equation.

Chapter 10

The semi-Lagrangian technique

In an Eulerian advection scheme an observer watches the world evolve around him at a fixed geographical point. Such schemes work well on regular Cartesian grids, but often lead to overly restrictive time steps due to considerations of computational stability. In a Lagrangian advection scheme observer watches the world evolve around himself as he travels with a fluid particle. Such schemes can often use much larger time steps than Eulerian ones, but have the disadvantage that an initially regularly spaced set of particles will generally evolve into a highly irregular-spaced set at later times, and important features of the flow may consequently not be well represented. The idea behind the semi-Lagrangian advection schemes is to try for the best of both worlds: the regular resolution of the Eulerian schemes and the enhanced stability of the Lagrangian ones. This is achieved by using a different set of particles at each time level, the particles being chosen such that they arrive exactly at the grid points of a regular Cartesian mesh at the end of the time step.

Several attempts have been made to construct stable-time integration schemes permitting large time steps. Robert (1981) proposed using the quasi-Lagrangian technique for the treatment of the advective part of the equations.

10.1 One-dimensional passive advection

To present the basic idea underlying the semi-Lagrangian method in its simplest context let us examine the one-dimensional advection equation

$$\frac{DF}{Dt} = \frac{\partial F}{\partial t} + \frac{\partial x}{\partial t} \frac{\partial F}{\partial x} = 0 \quad (10.1)$$

where F is the advected property and

$$\frac{\partial x}{\partial t} = U(x, t) \quad (10.2)$$

is a given advection velocity. Equation (10.1) states that the scalar F remains constant along a fluid path or trajectory. In Figure 10.1, the exact trajectory in the x - t plane of the fluid particle that arrives at mesh point x_m at time $t^n + \Delta t$ is represented by the solid curve AC , and an approximate straight-line trajectory by the dashed line $A'C$. Let us assume we know $F(x, t)$ at all mesh points x_m at times $t^n - \Delta t$ and t^n , and that we wish to obtain values at the same grid points at time $t^n + \Delta t$. The essence of semi-Lagrangian advection

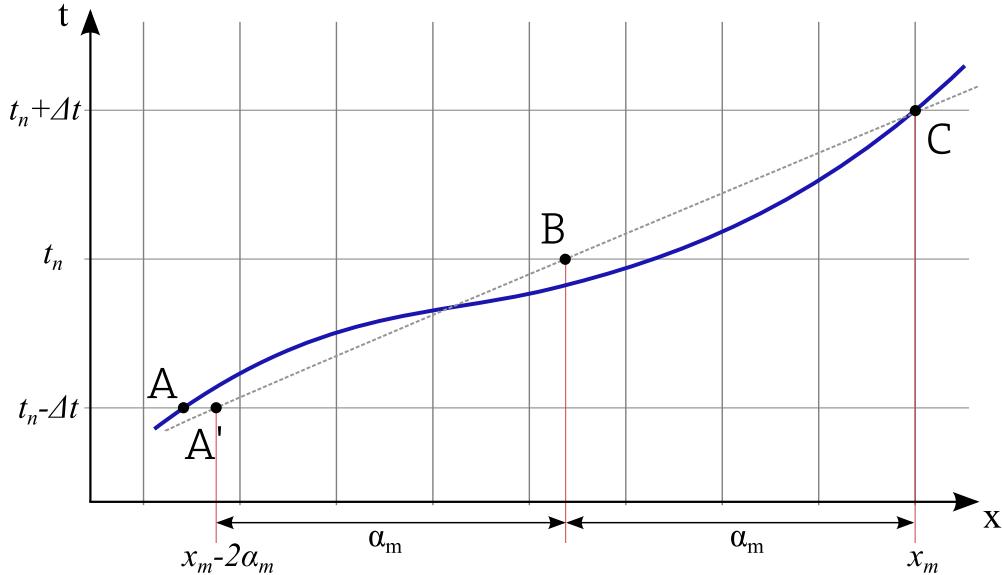


Figure 10.1: Schematic for three time level advection. Actual (solid curve) and approximate (dashed line) trajectories that arrive at mesh point x_m at time $t^n + \Delta t$. Here α_m is the distance the particle is displaced in x in time Δt .

is to "roughly integrate" Equation (10.1) along the approximated straight-line trajectory $A'C$. Thus,

$$\frac{F(x_m, t^n + \Delta t) - F(x_m - 2\alpha_m, t^n - \Delta t)}{2\Delta t} = 0 \quad (10.3)$$

where α_m is the distance BD var aer D någonstanms???? the particle travels in x in time Δt when following the approximated space-time trajectory $A'C$. If we know α_m , the value of F at the arrival point x_m at the time $t^n + \Delta t$ is just its value at the upstream point $x_m - 2\alpha_m$ at the time $t^n - \Delta t$. However, we have not as yet determined α_m ; even if we had, we would only know F at grid points, and generally it still remains to evaluate F somewhere between these.

To determine α_m , note that U evaluated at the point B of Figure 10.1 is just the inverse of the slope of the straight line $A'C$, which yields the following approximation of Equation (10.2):

$$\alpha_m = \Delta t U(x_m - \alpha_m, t^n) \quad (10.4)$$

This equation may be solved iteratively for the displacement α_m , e.g. by

$$\alpha_m^{k+1} = \Delta t U(x_m - \alpha_m^k, t^n) \quad (10.5)$$

with some initial guess for α_m^0 , provided that U can be evaluated between the grid points. To evaluate F and U b between grid points, spatial interpolation is used. A summary of the semi-Lagrangian algorithm for passive advection in one dimension is thus:

- Solve Equation (10.5) iteratively for the displacement α_m for all grid points x_m using some initial guess (usually the displacement at the previous time level) and an interpolation formula.

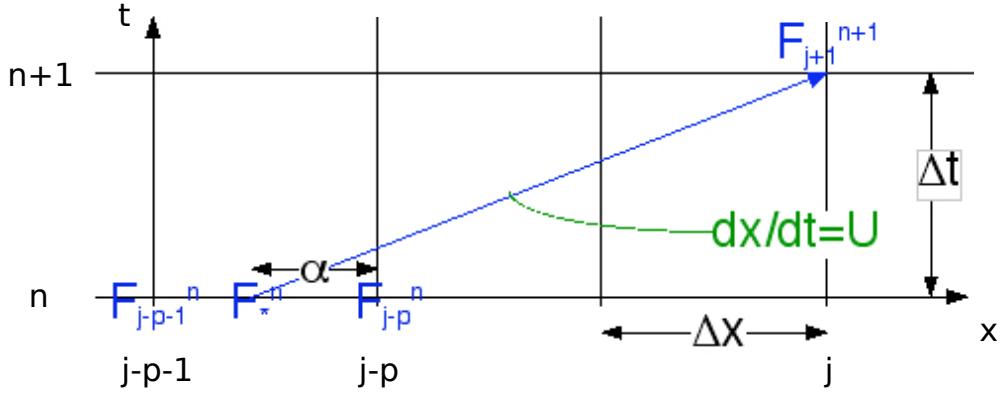


Figure 10.2: Schematic for the space interpolation.

- b) Evaluate F at upstream points $x_m - 2\alpha_m$ at time $t^n - \Delta t$ using an interpolation formula.
- c) Evaluate F at arrival points x_m at time $t^n + \Delta t$ using Equation (10.3).

10.2 Interpolation and Stability

Let us consider the linear advection equation

$$\frac{DF}{Dt} = \frac{\partial F}{\partial t} + U \frac{\partial F}{\partial x} = 0$$

The distance traveled during the previous time interval Δt by an air or water parcel arriving at point x_j is $U\Delta t$, and hence it comes from a point

$$x_* = x_j - U\Delta t$$

If this located between grid points $j - p$ and $j - p - 1$, and we define α as the fraction of grid length from x_* to x_{j-p} , we have

$$U\Delta t = (p + \alpha) \Delta x$$

and, using linear interpolation to find F_*^n , it is found that

$$F_j^{n+1} = F_*^n = (1 - \alpha) F_{j-p}^n + \alpha F_{j-p-1}^n \quad (10.6)$$

We determine the stability using the von Neumann method and hence assume a solution of the form

$$F_j^n = F_0 \lambda^n e^{ijk\Delta x},$$

insertion yields

$$\lambda = \left[1 - \alpha \left(1 - e^{-ik\Delta x} \right) \right] e^{-ipk\Delta x}$$

and

$$|\lambda|^2 = 1 - 2\alpha(1 - \alpha)[1 - \cos(k\Delta x)]$$

Thus $|\lambda| \leq 1$ as long as $\alpha(1 - \alpha) \geq 0$, or

$$0 \leq \alpha \leq 1$$

The scheme is consequently stable if the interpolation points are the two nearest ones to the departure point, but it is neutrally stable only if $\alpha = 0$ or $\alpha = 1$, that is to say when no interpolation is needed.

We find that heavy damping occurs for short wavelengths (i.e. large wave numbers) with complete extinction when

$$\cos(k\Delta x) = -1 \Leftrightarrow k\Delta x = (2m - 1)\pi \quad m = 1, 2, 3, \dots$$

in terms of wavelength extinction thus takes place when

$$L = \frac{2\Delta x}{2m - 1} \quad m = 1, 2, 3, \dots$$

The modes for $m > 1$ do not really make sense here, since it would be equivalent to considering waves with a wavelength shorter than the spatial increment Δx . Total damping thus occurs for the shortest waves that can be represented for a given grid resolution, i.e. when $L = 2\Delta x$ and $\alpha = 1/2$.

A noteworthy feature of this scheme (peculiar to the case of a constant wind) is that for a given α the phase errors and dissipation decrease as p increases. This happens because the departure point can be located precisely using only the wind at the arrival point.

A similar analysis as that above can be carried out for quadratic interpolation. Once again the scheme is absolutely stable provided F_*^n is computed by interpolation from the nearest three grid points. This scheme has less damping than the one based on linear interpolation, but the phase representation is not improved. It is easy to show that when the departure point lies within half a grid-cell length from the grid point (i.e. $p = 0$), this scheme becomes identical to the Lax-Wendroff scheme (referens????).

These ideas can be extended to two-dimensional flow. It has been found that bi-quadratic interpolation is absolutely stable for a constant flow (provided the nine grid points closest to the departure point are used for interpolation) and that the characteristics of this scheme are superior to those of the bilinear interpolation scheme.

Chapter 11

Poisson and Laplace equations (elliptic)

Consider Poisson's equation in two dimensions

$$\nabla^2 u = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) u = f(x, y) \quad (11.1)$$

If $f(x, y) = 0$ then (11.1) is called the Laplace Equation.

Equation (6.8) and can be discretised as

$$\frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{(\Delta x)^2} + \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{(\Delta y)^2} = f_{i,j} \quad (11.2)$$

If we consider a square grid in such a way that $\Delta x = \Delta y$, then Equation (11.2) simplifies to

$$u_{i,j} = \frac{1}{4} [u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - (\Delta x)^2 f_{i,j}] \quad (11.3)$$

If the boundary values for the domain are known then it is possible to solve this by iteration.

In iterative methods we need initial values at iteration level $u_{i,j}^m$ ($m = 0$ initially) and the purpose is to calculate $u_{i,j}^{m+1}$.

11.1 Jacobi iteration

Values from previous iteration level are used, which results in

$$u_{i,j}^{m+1} = \frac{1}{4} [u_{i-1,j}^m + u_{i+1,j}^m + u_{i,j-1}^m + u_{i,j+1}^m - (\Delta x)^2 f_{i,j}]$$

The method works but is inefficient and is not used in solving practical problems.

11.2 Gauss-Seidel iteration

A clear improvement in efficiency of iterative methods is obtained if we use the newly computed values in the iteration formula: iteration level $m + 1$ values are available for nodes $(i - 1, j)$ and $(i, j - 1)$ when calculating u for node (i, j) . Thus the Gauss-Seidel formula is:

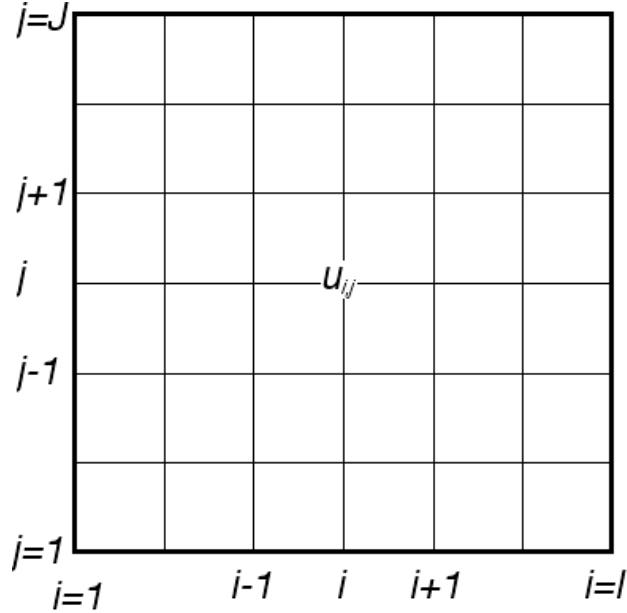


Figure 11.1: Grid for the Poisson and Laplace equations. Boundary values required for the four walls: $i = 1, j = 1, J; i = I, j = 1, J; i = 1, I, j = 1; i = 1, I, j = J$.

$$u_{i,j}^{m+1} = \frac{1}{4} [u_{i-1,j}^{m+1} + u_{i+1,j}^m + u_{i,j-1}^{m+1} + u_{i,j+1}^m - (\Delta x)^2 f_{i,j}] \quad (11.4)$$

The inclusion of the two newly computed values makes Gauss-Seidel iteration more efficient than Jacobi iteration.

11.3 Successive Over Relaxation (SOR)

Gauss-Seidel iteration method can be further improved by increasing the convergence rate using the method of SOR (Successive Over Relaxation). The change between two successive Gauss-Seidel iterations is called the residual c , which is defined as

$$c = u_{i,j}^{m+1} - u_{i,j}^m$$

In the method of SOR, the Gauss-Seidel residual is multiplied by a relaxation factor ω and new iteration value is obtained from

$$u_{i,j}^{m+1} = u_{i,j}^m + \omega c = u_{i,j}^m + \omega (\hat{u}_{i,j}^{m+1} - u_{i,j}^m) = (1 - \omega) u_{i,j}^m + \omega \hat{u}_{i,j}^{m+1} \quad (11.5)$$

where $\hat{u}_{i,j}^{m+1}$ denotes the new iteration value obtained from Gauss-Seidel method using Equation (11.4). It can be easily seen that if $\omega = 1$, SOR reduces to Gauss-Seidel iteration method. By substituting Equation (11.4) of the Gauss-Seidel iteration method to Equation (11.5), we obtain the equation used in the SOR method:

$$u_{i,j}^{m+1} = (1 - \omega) u_{i,j}^m + \frac{\omega}{4} [u_{i-1,j}^{m+1} + u_{i+1,j}^m + u_{i,j-1}^{m+1} + u_{i,j+1}^m - (\Delta x)^2 f_{i,j}] \quad (11.6)$$

Usually the numerical value of relaxation parameter can be obtained by trial and error and optimum value is around 1.5. In the case that $0 < \omega < 1$, the method is said to be "under relaxed". According to the selection of parameter ω , we either extrapolate $\omega > 1$ or $0 < \omega < 1$ interpolate between the old iteration value at level m and Gauss-Seidel value at level $m + 1$. If we extrapolate too much, i.e. ω is too high, the iteration starts to oscillate and probably collapses.

Exercises

- 1) Set up a numerical model with 10×10 points. Start with $u = 0$ in the interior and with $u = 1$ as boundary conditions. Test the convergence of the 3 different iterations schemes from this chapter with $f = 0$.

Chapter 12

Model Coordinates

12.1 Vertical ocean coordinates

The three most common vertical coordinate systems in the ocean circulation models are presented in Figure 12.1. z-star....

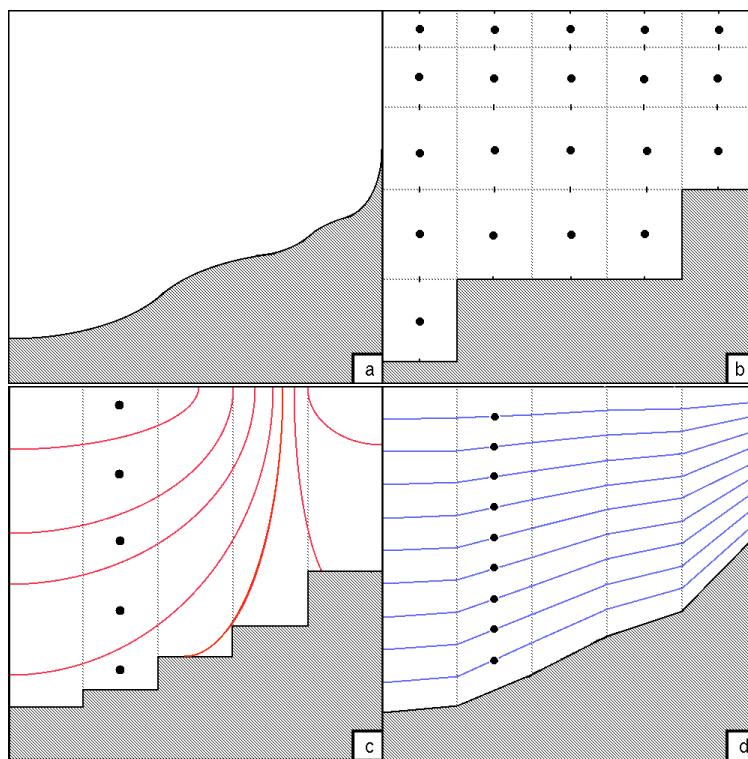


Figure 12.1: a) Ocean model vertical coordinates. The real topography, b) depth level coordinates, c) density coordinates, d) sigma coordinates.

12.1.1 Fixed height or depth coordinates

A simple example of a vertical discretisation is the one of the continuity equation, which is used in many OGCMs (Ocean General Circulation Model), which uses the B-grid and fixed depth levels as in Figure 12.2.

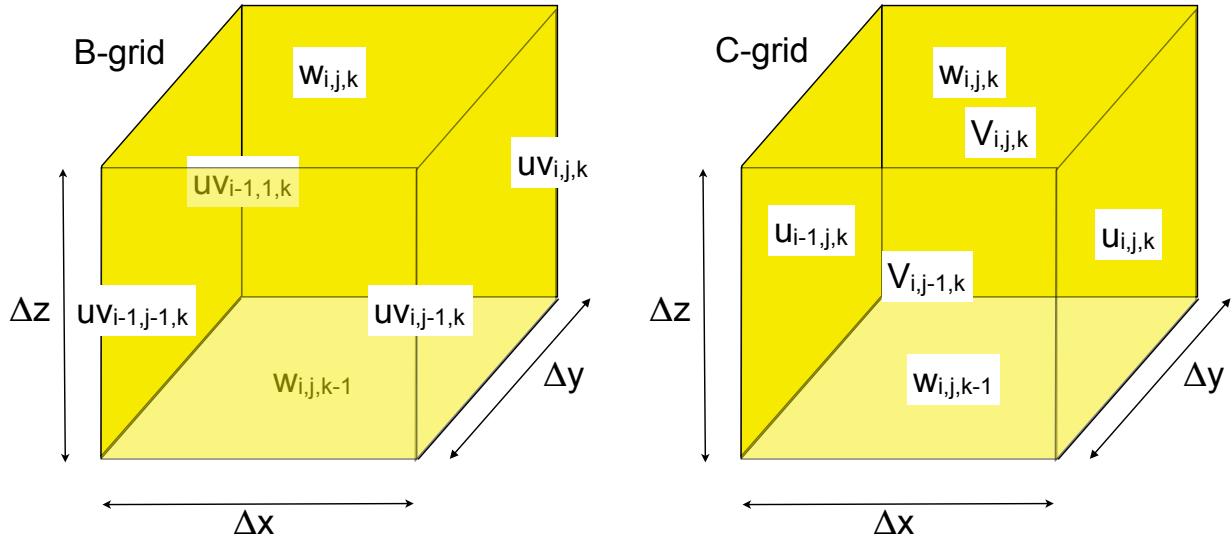


Figure 12.2: Finite difference boxes for the B-grid and the C-grid with fixed depth level coordinates.

The continuity equation then becomes:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \quad (12.1)$$

can be discretised on a B-grid to

$$w_{i,j,k} = w_{i,j,k-1} - \Delta z \left[\frac{(u_{i,j,k} + u_{i,j-1,k}) - (u_{i-1,j,k} + u_{i-1,j-1,k})}{2\Delta x} + \frac{(v_{i,j,k} + v_{i-1,j,k}) - (v_{i,j-1,k} + v_{i-1,j-1,k})}{2\Delta y} \right] \quad (12.2)$$

or on a C-grid to

$$w_{i,j,k} = w_{i,j,k-1} - \Delta z \left[\frac{u_{i,j,k} - u_{i-1,j,k}}{\Delta x} + \frac{v_{i,j,k} - v_{i,j-1,k}}{\Delta y} \right] \quad (12.3)$$

Equation (12.2) is integrated from the bottom and upwards with the boundary condition $w = 0$ where k increases with depth. Equation (12.2) can be explained by that the sum of all the volume fluxes in or out of the grid box is zero.

12.2 Atmospheric vertical coordinates

Instead of depth/height as vertical coordinate in our system of equations it is possible to use other quantities. The density varies with latitude and height/depth which make the equations sometimes less easy to use with than an alternative system which uses other quantities such as pressure, sigma or isentropic for the

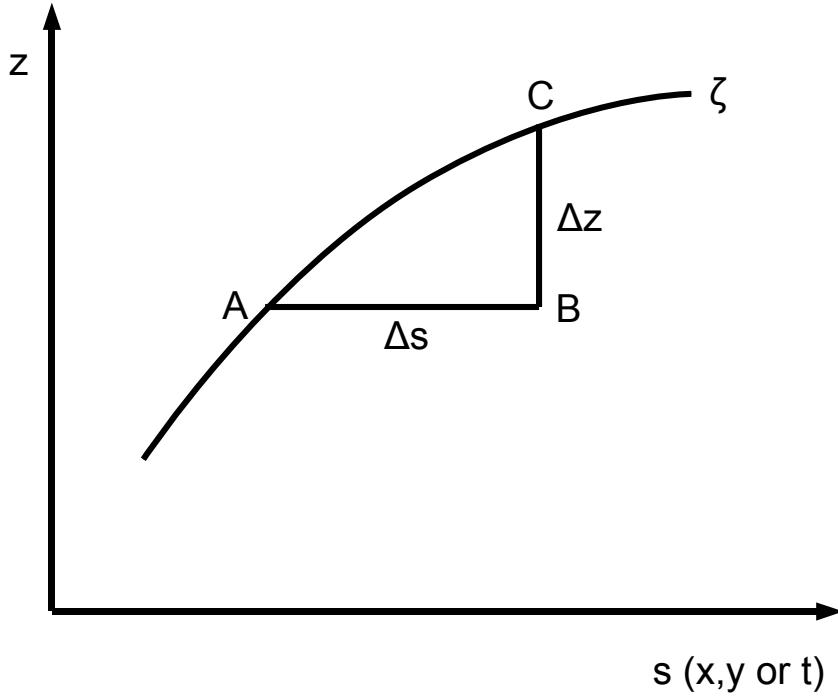


Figure 12.3: Schematic showing the vertical coordinate transformation.

atmosphere and density or sigma for the ocean as the vertical coordinate. These coordinates can be more useful for numerical techniques in solving the complete equations of motion. We can derive a system of equations for a generalised vertical coordinate which is assumed to be related to the height/depth by a single-valued monotonic function. When we transform the vertical coordinate a variable $u(x, y, z, t)$ becomes $a(x, y, \zeta(x, y, z, t), t)$. The horizontal coordinates remain the same. Let s represent x, y or t . From Figure 12.3 we see that

$$\frac{C - A}{\Delta s} = \frac{B - A}{\Delta s} + \frac{C - B}{\Delta z} \frac{\Delta z}{\Delta s} \quad (12.4)$$

so that

$$\left(\frac{\partial a}{\partial s} \right)_\zeta = \left(\frac{\partial a}{\partial s} \right)_z + \left(\frac{\partial a}{\partial z} \right)_s \left(\frac{\partial z}{\partial s} \right)_\zeta \quad (12.5)$$

where

$$\frac{\partial a}{\partial \zeta} = \frac{\partial a}{\partial z} \frac{\partial z}{\partial \zeta} \quad (12.6)$$

or

$$\frac{\partial a}{\partial z} = \frac{\partial a}{\partial \zeta} \frac{\partial \zeta}{\partial z} \quad (12.7)$$

Substituting Eq. 12.7 in Eq. 12.5, we obtain

$$\left(\frac{\partial a}{\partial s} \right)_\zeta = \left(\frac{\partial a}{\partial s} \right)_z + \frac{\partial a}{\partial \zeta} \frac{\partial \zeta}{\partial z} \left(\frac{\partial z}{\partial s} \right)_\zeta \quad (12.8)$$

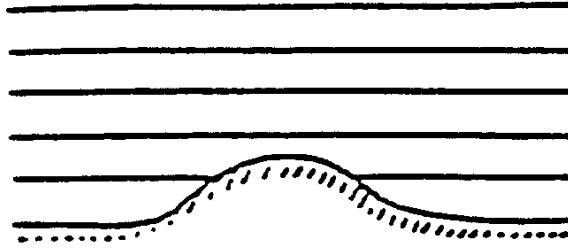
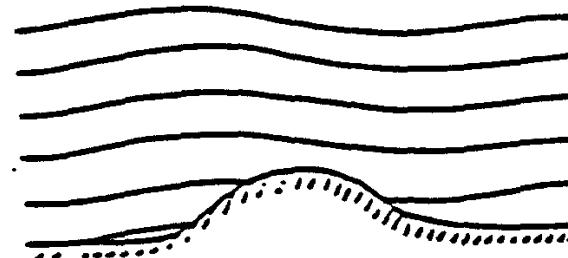
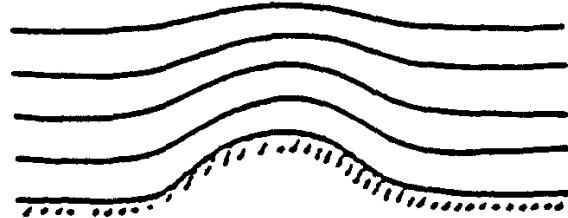
z -coordinates p -coordinates σ -coordinates

Figure 12.4: Schematic illustration of height, pressure and sigma coordinates in the atmosphere..

From this relationship, if $s = x, y$, we can get an equation for a horizontal gradient of the scalar a in ζ coordinates:

$$\nabla_\zeta a = \nabla_z a + \frac{\partial a}{\partial \zeta} \frac{\partial \zeta}{\partial z} \nabla_\zeta z \quad (12.9)$$

and for the horizontal divergence of a vector \mathbf{V} :

$$\nabla_\zeta \cdot \mathbf{V} = \nabla_z \cdot \mathbf{V} + \frac{\partial \mathbf{V}}{\partial \zeta} \cdot \frac{\partial \zeta}{\partial z} \nabla_\zeta z \quad (12.10)$$

The total derivative of $a(x, y, \zeta, t)$ becomes

$$\frac{da}{dt} = \left(\frac{\partial a}{\partial t} \right)_\zeta + \mathbf{V} \cdot \nabla_\zeta a + \dot{\zeta} \frac{\partial a}{\partial \zeta} \quad (12.11)$$

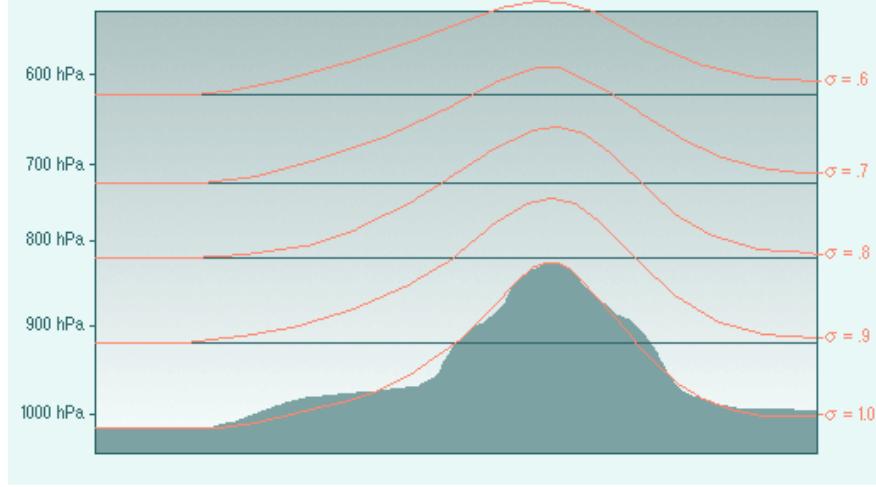


Figure 12.5: Atmospheric sigma coordinates.

12.2.1 Pressure coordinates

Pressure or isobaric coordinates can be used in atmosphere although not very often anymore in atmospheric GCMs. In pressure coordinates where $\partial p / \partial \zeta \equiv 1$, the total derivative (Eq. 12.11) is given by

$$\frac{da}{dt} = \frac{\partial a}{\partial t} + \mathbf{V} \cdot \nabla a + \omega \frac{\partial a}{\partial p} \quad (12.12)$$

where the vertical velocity in pressure coordinates is $\omega \equiv dp/dt$. The continuity equation 12.1 can now be written as

$$\nabla_p \cdot \mathbf{V} + \frac{\partial \omega}{\partial p} = 0 \quad (12.13)$$

Most Atmospheric GCMs today use however terrain-following vertical coordinates, which is described beneath.

12.2.2 Atmospheric sigma coordinates

The sigma coordinate system defines the base at the model's ground level. The surfaces in the sigma coordinate system follow the model terrain and are steeply sloped in the regions where terrain itself is steeply sloped. The sigma coordinate system defines the vertical position of a point in the atmosphere as a ratio of the pressure difference between that point and the top of the domain to that of the pressure difference between a fundamental base below the point and the top of the domain. Because it is pressure based and normalised, it is easy to mathematically cast governing equations of the atmosphere into a relatively simple form. The sigma coordinate is hence $\sigma = p/p_S$ where $p_S(x, y, z, t)$ is the pressure at the surface of the Earth. The boundary values are hence $\sigma = 0$ at the top of the atmosphere where $p = 0$ and $\sigma = 1$ at the surface of the Earth. The relationship between pressure and sigma coordinates is presented in Figure 12.5.

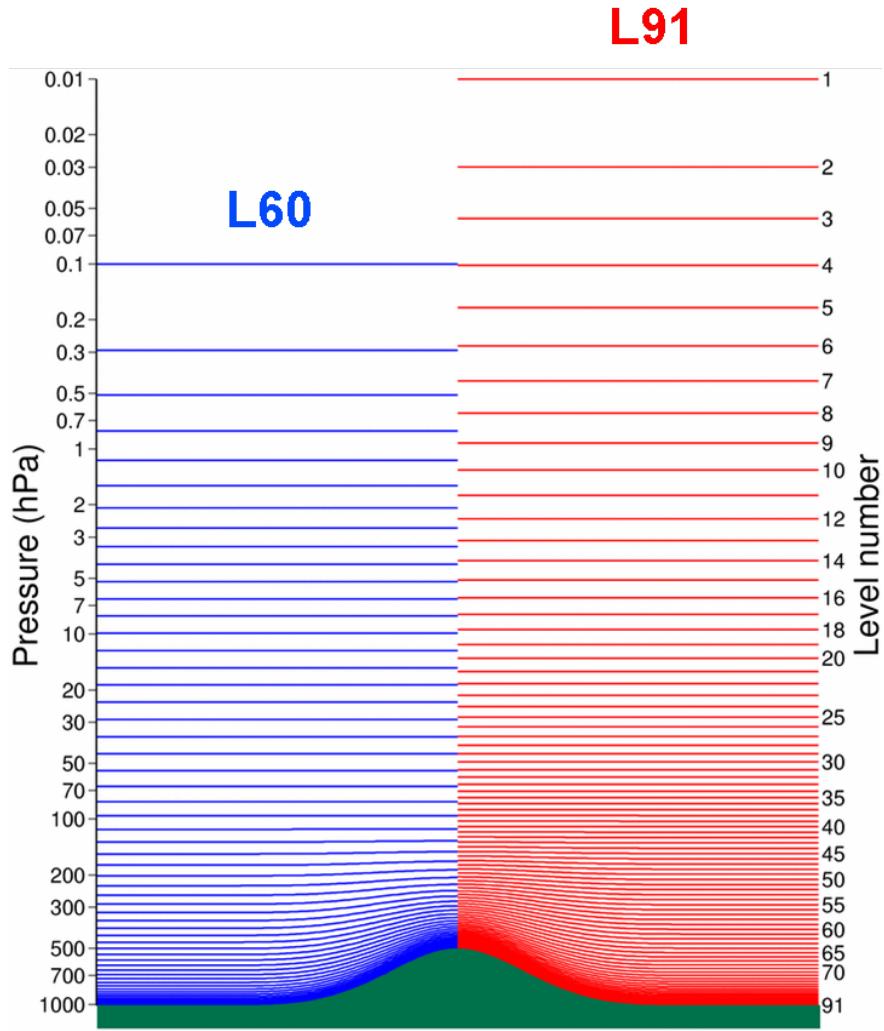


Figure 12.6: Two different vertical resolution of the hybrid-coordinate model at the ECMWF (The European Centre for Medium-Range Weather Forecasts).

12.2.3 Hybrid coordinates

The hybrid coordinate system has the properties of sigma coordinates in the lower atmosphere and pressure in the stratosphere.

Following [Simmons and Burridge \(1981\)](#) the atmosphere is divided into $NLEV$ layers, which are defined by the pressures at the interfaces between them and these pressures are given by

$$p_{k+1/2} = a_{k+1/2} + b_{k+1/2} p_S \quad (12.14)$$

for $k = 0, 1, \dots, NLEV$, with $k = 0$ at the top of the atmosphere and $k = NLEV$ at the Earth's surface. The $a_{k+1/2}$ and $b_{k+1/2}$ are constants, whose values effectively define the vertical coordinate and p_S is the surface pressure. The dependent variables, which are the zonal wind (u), the meridional wind (v), the temperature

```

=====
TOP OF MODEL ATMOSPHERE ===== i = 1      a (Pa)      b (Pa Pa-1)
----- model level ----- (data) ----- j = 1      a = 0.00000   b = 0.00000
===== interface ===== i = 2      a = 10.00000  b = 0.00000
----- model level ----- (data) ----- j = 2      a = 20.00000  b = 0.00000
===== interface ===== i = 3      a = 28.21708  b = 0.00000
                                         a = 38.42530  b = 0.00000

----- model level ----- (data) ----- j = 1
===== interface ===== i = 1
----- model level ----- (data) ----- j
===== interface ===== i
----- model level ----- (data) ----- j + 1

===== interface ===== i = 59      a = 7.36774  b = 0.99402
----- model level ----- (data) ----- j = 59      a = 3.68387  b = 0.99582
===== interface ===== i = 60      a = 0.00000  b = 0.99763
----- model level ----- (data) ----- j = 60 = J      a = 0.00000  b = 0.99881
===== MODEL SURFACE ===== i = 61 = I      a = 0.00000  b = 1.00000

```

Figure 12.7: Model levels for the 60 layers presented in Figure 12.6, with a_k and a_k at interfaces are highlighted in red.

(T) and the specific humidity (q) are defined in the middle of the layers, where the pressure is defined by

$$p_k = \frac{1}{2}(p_{k-1/2} + p_{k+1/2}) \quad (12.15)$$

for $k = 1, 2, \dots, NLEV$. The vertical coordinate is $\eta = \eta(p, p_S)$ and has the boundary value $\eta(0, p_S) = 0$ at the top of the atmosphere and $\eta(p_S, p_S) = 1$ at the Earth's surface. Two different vertical resolutions with hybrid coordinates are presented in Figure 12.6.

12.3 Horizontal coordinates

12.3.1 Finite differences

Most of the finite difference schemes in this book are made on cartesian grids, which are both orthogonal and of constant grid sizes as illustrated in Figure 12.8.

The drawback of the cartesian grids is that they do not change in space. Ocean and atmospheric circulation models do not have cartesian grids apart from some academic ones used in courses on numerical methods like in the present book. A typical grid will at least have a latitude dependence of Δx taking account that the distances between longitudes decrease with latitude. These grids are called curvilinear, but can still be orthogonal preserve right angles between the two coordinates at every point of interest in the grid. Figure

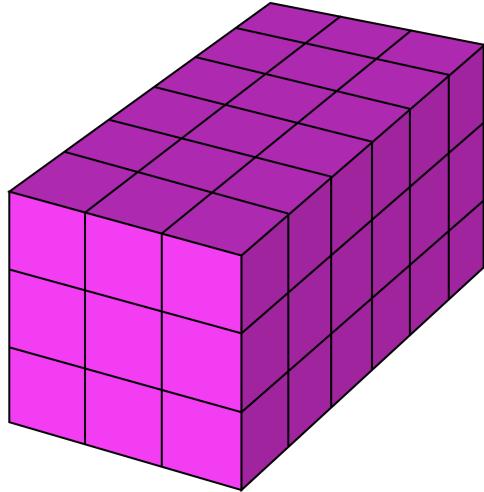


Figure 12.8: A *Cartesian grid*. Simplest possible grid.

[12.9](#) an example of this. The region bounded by two adjacent segments of one of the curvilinear coordinates and two adjacent segments of the other curvilinear coordinates will be transformable to a rectangle. An orthogonal curvilinear coordinate system permits the design of a grid system in a complicated region such as that bounded by a shoreline or "bending" the north pole to a land position. The vertical resolution Δz_k will typically have a thinner grid cells near the surface between the ocean and the atmosphere.

12.3.2 Finite elements

In designing a numerical weather prediction model, one of the most fundamental aspects is the choice of discretisation technique in each of the spatial dimensions. In the vertical, by far the most popular choice is the finite difference method; while in the horizontal, both finite-difference and (especially for global models) spectral methods are widely employed. A third possibility is the finite element method.

The essence of the finite element method can be seen by considering various ways of representing a function $u(x)$ on an interval $a \leq x \leq b$. In the finite-difference method the function is defined only on a set of grid points; i.e. $u(x_j)$ is defined for a set of x_j , but there is no explicit information about how the function behaves between the grid points. In the spectral method, on the other hand, the function is defined in terms of a finite set of basis functions:

$$u(x) = \sum_{k=0}^N a_k e_k \quad (12.16)$$

where the basis functions $e_k(x)$ are global (e.g. Fourier series, or spherical harmonics for two-dimensional functions on the surface of a sphere), and the a_k are the spectral coefficients. Equation (12.16) defines $u(x)$ everywhere on the interval, and the representation is independent of any set of grid points.

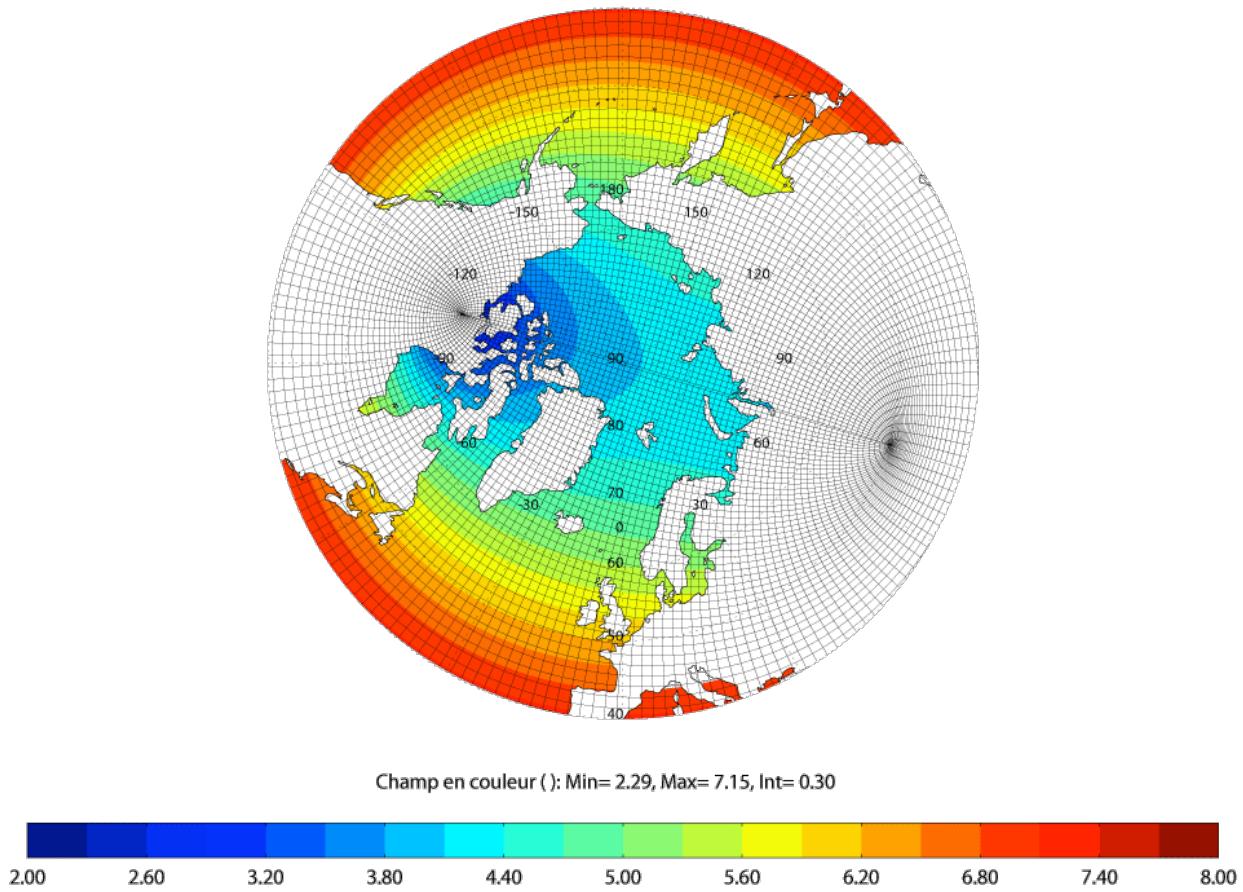


Figure 12.9: The orthogonal curvilinear ORCA12 ocean grid for the NEMO model, which is tripolar with two north poles in order to avoid the north pole to be an ocean point. The colour scale indicates the grid size in km.

In the finite-element method, the function is again in terms of a finite set of basis functions:

$$u(x) = \sum_{k=0}^N a_k e_k$$

but this time basis functions $e_k(x)$ are local, i.e. they are non-zero only on a small-sub-interval. As in the spectral method, the a_k are the coefficients of the basis functions, and $u(x)$ is defined everywhere; but as in the finite-difference method, there is an underlying mesh of grid points (nodes) involved in the representation.

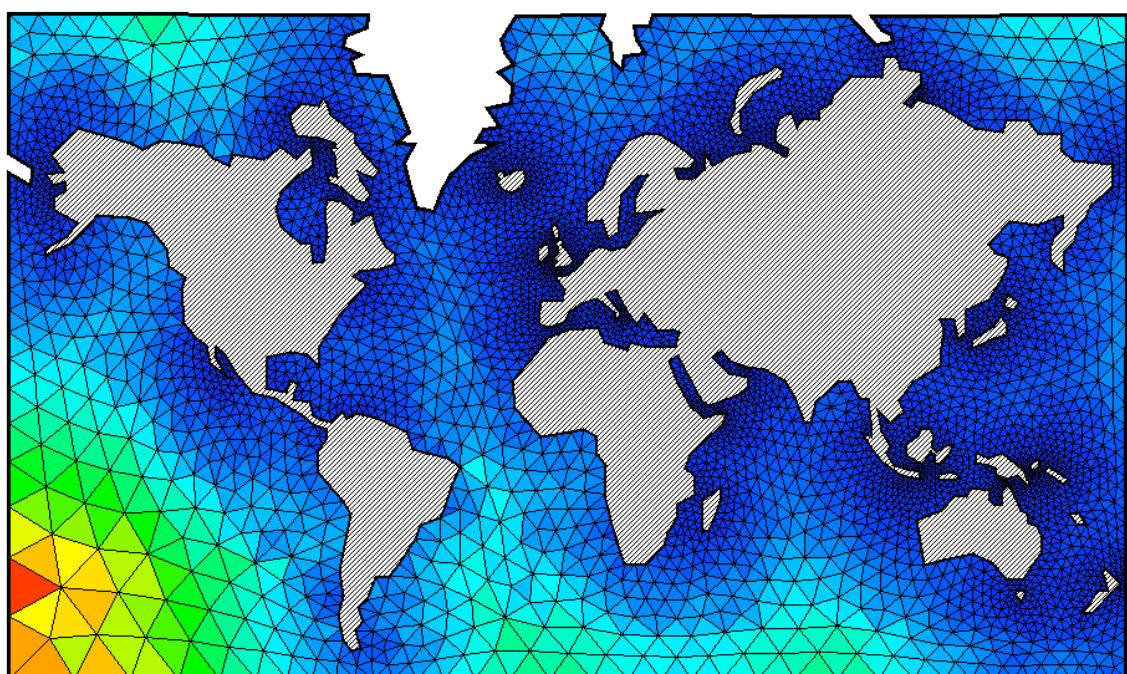


Figure 12.10: *Finite elements for an ocean general circulation model*

Chapter 13

3D modelling

The 3D models of the ocean circulation known today as Ocean General Circulation Models (OGCMs) are increasing in complexity as modellers improve them. The numerical improvements can be such as higher order numerical schemes, advanced model grids, etc. The improvement of the physics can be to replace the hydrostatic equations by the non-hydrostatic, to change the mixing parameterisations of the unresolved scales, etc. All these improvements are necessary in order to continue to make more realistic model integrations. We will in the present chapter present a 3D model of the ocean circulation as simple as possible but still with the core of the numerics close to the OGCMs. This in order to be able to "see" the core of the numerics of the 3D model. OGCMs have nearly all some sort of curvilinear coordinates and some depth dependent layer thicknesses, which make the discretised equations not so easy to understand. We have therefore discretised the equations on a rectangular cartesian C-grid as illustrated by Figures 13.3, 13.1 and 13.2. Our model is instead of on a spherical grid with a realistic bottom bathymetry on a rectangular ocean domain with a flat bottom. We have also replaced the two tracer equations of heat and salt by one tracer equation for the density. A real OGCM would instead compute the density with the equation of state from the temperature and salinity and depth (pressure).

Our simplified 3D-model makes otherwise use of the usual hypothesis such as

- Boussinesq hypothesis: density variations are neglected except in their contribution to the buoyancy force, which explains the ρ_0 instead of ρ in the horizontal momentum Equations 13.1 and 13.2.
- Hydrostatic hypothesis: the vertical momentum equation is reduced to a balance between the vertical pressure gradient and the buoyancy force, which removes convective processes from the equations. The convection is instead parameterised with an increased vertical diffusion.
- Incompressibility hypothesis: the 3D divergence of the velocity is approximated to be zero.
- turbulent closure hypothesis : the turbulent fluxes (which represent the effect of small scale processes on the large-scale) are expressed in terms of large-scale features, which in our case simple Laplacian diffusion and viscosity.

13.1 The Equations of motion

The equations of motion that will be used are

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} - fv = -\frac{1}{\rho_0} \frac{\partial p}{\partial x} + A_H \nabla^2 u + A_V \frac{\partial^2 u}{\partial z^2} + F^x \quad (13.1)$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} + fu = -\frac{1}{\rho_0} \frac{\partial p}{\partial y} + A_H \nabla^2 v + A_V \frac{\partial^2 v}{\partial z^2} + F^y \quad (13.2)$$

$$0 = -\frac{\partial p}{\partial z} + \rho g \quad (13.3)$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \quad (13.4)$$

$$\frac{\partial \rho}{\partial t} + \mathbf{V} \cdot \nabla \rho = K_H \nabla^2 \rho + K_V \frac{\partial^2 \rho}{\partial z^2} + C \quad (13.5)$$

Before discretising the above equations we need to rewrite them on the form

$$\frac{\partial u}{\partial t} = \xi v - w \frac{\partial u}{\partial z} - \frac{\partial E}{\partial x} + A_H \nabla^2 u + A_V \frac{\partial^2 u}{\partial z^2} + F^x \quad (13.6)$$

$$\frac{\partial v}{\partial t} = -\xi u - w \frac{\partial v}{\partial z} - \frac{\partial E}{\partial y} + A_H \nabla^2 v + A_V \frac{\partial^2 v}{\partial z^2} + F^y \quad (13.7)$$

$$0 = -\frac{\partial p}{\partial z} + \rho g \quad (13.8)$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \quad (13.9)$$

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\mathbf{V} \rho) + K_H \nabla^2 \rho + K_V \frac{\partial^2 \rho}{\partial z^2} + C \quad (13.10)$$

where the absolute vorticity is

$$\xi \equiv \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} + f \quad (13.11)$$

and the energy function is defined as

$$E \equiv \frac{p}{\rho_0} + \frac{1}{2} (u^2 + v^2) \quad (13.12)$$

A possible discretisation of these equations with centred finite difference are

$$u_{i,j,k}^{n+1} = u_{i,j,k}^{n-1} + 2\Delta t \left\{ \frac{1}{4} [\xi_{i,j,k}^n (v_{i,j,k}^n + v_{i+1,j,k}^n) + \xi_{i,j-1,k}^n (v_{i,j-1,k}^n + v_{i+1,j-1,k}^n)] - (w_{i,j,k}^n + w_{i+1,j,k}^n + w_{i,j,k-1}^n + w_{i+1,j,k-1}^n) \frac{u_{i,j,k-1}^n - u_{i,j,k+1}^n}{8\Delta z} - \frac{E_{i+1,j,k}^n - E_{i,j,k}^n}{\Delta x} \right\} \quad (13.13)$$

$$v_{i,j,k}^{n+1} = v_{i,j,k}^{n-1} + 2\Delta t \left\{ -\frac{1}{4} [\xi_{i,j,k}^n (u_{i,j,k}^n + u_{i,j+1,k}^n) + \xi_{i-1,j,k}^n (u_{i-1,j,k}^n + u_{i-1,j+1,k}^n)] - (w_{i,j,k}^n + w_{i,j+1,k}^n + w_{i,j,k-1}^n + w_{i,j+1,k-1}^n) \frac{v_{i,j,k-1}^n - v_{i,j,k+1}^n}{8\Delta z} - \frac{E_{i,j+1,k}^n - E_{i,j,k}^n}{\Delta y} \right\} \quad (13.14)$$

$$p_{i,j,k} = \sum_{k'=1}^{k-1} g\rho_{i,j,k'} \Delta z + g\rho_{i,j,k} \Delta z / 2 + g\rho_{i,j,1} \eta_{i,j} \quad (13.15)$$

$$w_{i,j,k-1} = w_{i,j,k} - \frac{U_{i,j} - U_{i-1,j}}{\Delta x} - \frac{V_{i,j} - V_{i,j-1}}{\Delta y} \quad (13.16)$$

$$\eta_{i,j}^{n+1} = \eta_{i,j}^{n-1} + 2\Delta t w_{i,j,0} \quad (13.17)$$

where the absolute vorticity is located between the corners of the T-boxes as illustrated by Figures 13.3 and 13.1.

$$\xi_{i,j,k} \equiv f + \frac{v_{i+1,j,k} - v_{i,j,k}}{\Delta x} - \frac{u_{i,j+1,k} - u_{i,j,k}}{\Delta y} \quad (13.18)$$

The fluxes U and V are defined in the same points as the velocity components u and v are located:

$$U_{i,j,k} \equiv u_{i,j,k} \frac{1}{2} (h_{i,j,k} + h_{i+1,j,k}) \quad (13.19)$$

$$V_{i,j,k} \equiv v_{i,j,k} \frac{1}{2} (h_{i,j,k} + h_{i,j+1,k}) \quad (13.20)$$

The grid cell thickness is constant at all depths except for the surface layer where the sea surface elevation is included

$$h_{i,j,k} = \Delta z + \eta_{i,j} \text{ for } k = 1 \quad (13.21)$$

$$h_{i,j,k} = \Delta z \text{ for } k \neq 1 \quad (13.22)$$

The gradient operator will act on the quantity E defined at the locations where h is defined:

$$E \equiv \frac{p_{i,j,k}}{\rho_0} + \frac{1}{2} \left[\frac{1}{2} (u_{i,j,k}^2 + u_{i-1,j,k}^2) + \frac{1}{2} (v_{i,j,k}^2 + v_{i,j-1,k}^2) \right] \quad (13.23)$$

13.2 The tracer Equation

The tracer equation describes the rate of change of a tracer such as e.g. potential temperature or salt in the ocean or the water vapour in the atmosphere or any other passive tracer that is advected and diffused in the ocean or atmosphere. This tracer equation with a simple parameterisation of the diffusion can be expressed as

$$\frac{\partial T}{\partial t} + \mathbf{V} \cdot \nabla T = K_H \nabla^2 T + K_V \frac{\partial^2 T}{\partial z^2} + C \quad (13.24)$$

where K_H and K_V are the horizontal and vertical diffusion coefficients and C a possible source term such as the heat flux between the atmosphere and the ocean. The continuity equation is in the case of incompressibility as in the ocean

$$\nabla \cdot \mathbf{V} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \quad (13.25)$$

The tracer Equation 13.24 can now be rewritten by incorporating the continuity Equation 13.25 to

$$\frac{\partial T}{\partial t} + \nabla \cdot (\mathbf{V} T) = K_H \nabla^2 T + K_V \frac{\partial^2 T}{\partial z^2} + C \quad (13.26)$$

13.3 Discretised on a cartesian grid

This tracer equation will now be discretised on a C-grid illustrated in Figure 13.4. Let us start with the term $\nabla \cdot (\mathbf{V} T)$ in Equation 13.26, which simply expresses the divergence of the tracer flux. The discretised

version of this term is simply the sum of all the tracer transports in and out of a grid box divided by the volume of the grid box. The tracer flux across the grid wall, where $u_{i,j,k}$ is located becomes

$$\begin{aligned} U_{i,j,k}^n &\equiv u_{i,j,k}^n \frac{1}{2} (T_{i,j,k}^n + T_{i+1,j,k}^n) \Delta y \Delta z \\ V_{i,j,k}^n &\equiv v_{i,j,k}^n \frac{1}{2} (T_{i,j,k}^n + T_{i,j+1,k}^n) \Delta x \Delta z \\ W_{i,j,k}^n &\equiv w_{i,j,k}^n \frac{1}{2} (T_{i,j,k}^n + T_{i,j,k+1}^n) \Delta x \Delta y \end{aligned} \quad (13.27)$$

which leads to

$$\nabla \cdot (\mathbf{V} T) \rightarrow \frac{(U_{i,j,k} - U_{i-1,j,k} + V_{i,j,k} - V_{i,j-1,k} + W_{i,j,k-1} - W_{i,j,k})}{\Delta x \Delta y \Delta z} \quad (13.28)$$

The discretised tracer equation becomes with a centred leap-frog time scheme and the diffusion terms at time step $n - 1$ as required by stability we get

$$\begin{aligned} T_{i,j,k}^{n+1} &= T_{i,j,k}^{n-1} + 2\Delta t \left[\right. \\ &\quad \left. - \frac{\left(U_{i,j,k}^n - U_{i-1,j,k}^n + V_{i,j,k}^n - V_{i,j-1,k}^n + W_{i,j,k-1}^n - W_{i,j,k}^n \right)}{\Delta x \Delta y \Delta z} \right. \\ &\quad \left. + K_H \frac{T_{i-1,j,k}^{n-1} - 2T_{i,j,k}^{n-1} + T_{i+1,j,k}^{n-1}}{(\Delta x)^2} + \frac{T_{i,j-1,k}^{n-1} - 2T_{i,j,k}^{n-1} + T_{i,j+1,k}^{n-1}}{(\Delta y)^2} \right. \\ &\quad \left. + K_V \frac{T_{i,j,k-1}^{n-1} - 2T_{i,j,k}^{n-1} + T_{i,j,k+1}^{n-1}}{(\Delta z)^2} \right] + C_{i,j,k}^n \end{aligned} \quad (13.29)$$

13.4 Discretised on an orthogonal curvilinear grid

A drawback of the above discretised tracer equation is that it requires cartesian grids, which do not change in space. Ocean and atmospheric circulation models do not have cartesian grids apart from some academic ones used in courses on numerical methods like in the present book.

It is therefore advantageously to make the finite differences directly on the diffusive tracer fluxes. Let U, V, W now instead be the sum of the advective and diffusive fluxes so that

$$\begin{aligned} U_{i,j,k}^n &\equiv \left[u_{i,j,k}^n \frac{1}{2} (T_{i,j,k}^n + T_{i+1,j,k}^n) - K_H \frac{T_{i+1,j,k}^{n-1} - T_{i,j,k}^{n-1}}{\Delta x} \right] \Delta y_i \Delta z_k \\ V_{i,j,k}^n &\equiv \left[v_{i,j,k}^n \frac{1}{2} (T_{i,j,k}^n + T_{i,j+1,k}^n) - K_H \frac{T_{i,j+1,k}^{n-1} - T_{i,j,k}^{n-1}}{\Delta y} \right] \Delta x_i \Delta z_k \\ W_{i,j,k}^n &\equiv \left[w_{i,j,k+1}^n \frac{1}{2} (T_{i,j,k}^n + T_{i,j,k+1}^n) - K_V \frac{T_{i,j,k}^{n-1} - T_{i,j,k+1}^{n-1}}{\Delta z_k} \right] \Delta x_i \Delta y_i \end{aligned} \quad (13.30)$$

Note that we have written $K_{V,k}$ with an index k , which indicated that we can let the vertical diffusion coefficient vary with depth. The vertical diffusion coefficient $K_{V,k}$ can be assumed to be either constant,

or a function of the local Richardson number ($\text{Ri} = N^2 / (\text{du}/\text{dz})^2$), or computed from a turbulent closure model (either TKE or KPP formulation).

The discretised tracer equation becomes now

$$T_{i,j,k}^{n+1} = T_{i,j,k}^{n-1} - 2\Delta t \left[\frac{U_{i,j,k}^n - U_{i-1,j,k}^n + V_{i,j,k}^n - V_{i,j-1,k}^n + W_{i,j,k-1}^n - W_{i,j,k}^n}{\Delta x_{i,j} \Delta y_{i,j} \Delta z_k} - C_{i,j,k} \right] \quad (13.31)$$

Note the horizontal indices on the grid lengths $\Delta x_{i,j}$ and $\Delta y_{i,j}$ in order to fit a curvilinear grid such as the one in Figure 12.9. The vertical grid thickness Δz_k will however have a vertical index since it only varies vertically with the exception of the bottom box that might vary in order to fit an exact depth of the ocean.

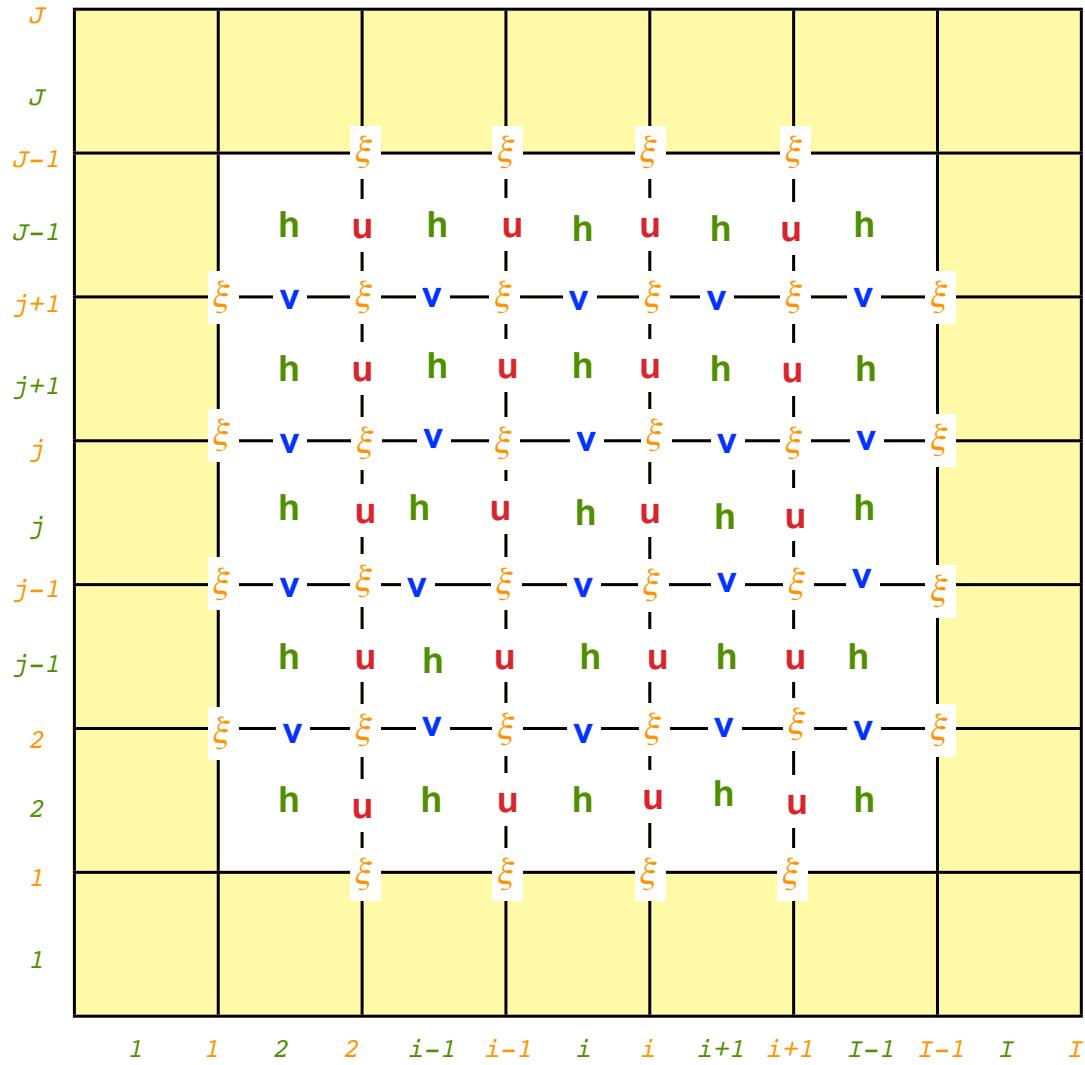


Figure 13.1: Horizontal view (longitude-latitude) of a possible rectangular model grid with land as yellow grid boxes. Only the non-zero variables are shown. I and J are the total number of grid boxes in the zonal and meridional direction respectively.

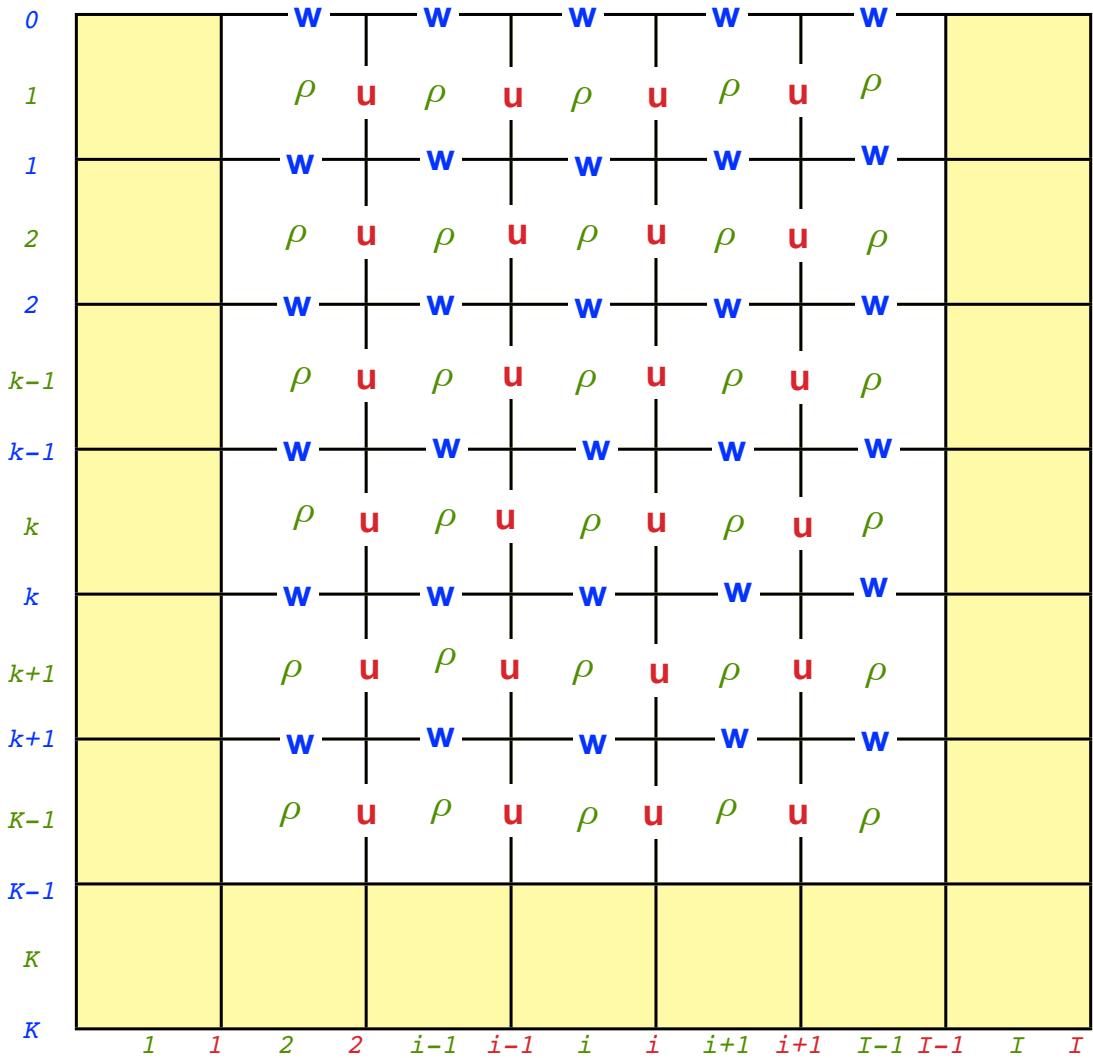


Figure 13.2: Zonal-vertical view of the model grid with land/bottom as yellow grid boxes. Only the non-zero variables are shown. K is the total number of vertical depth layers. Note that the vertical index k is increasing with depth and hence of opposite direction to the direction of the z -coordinate.

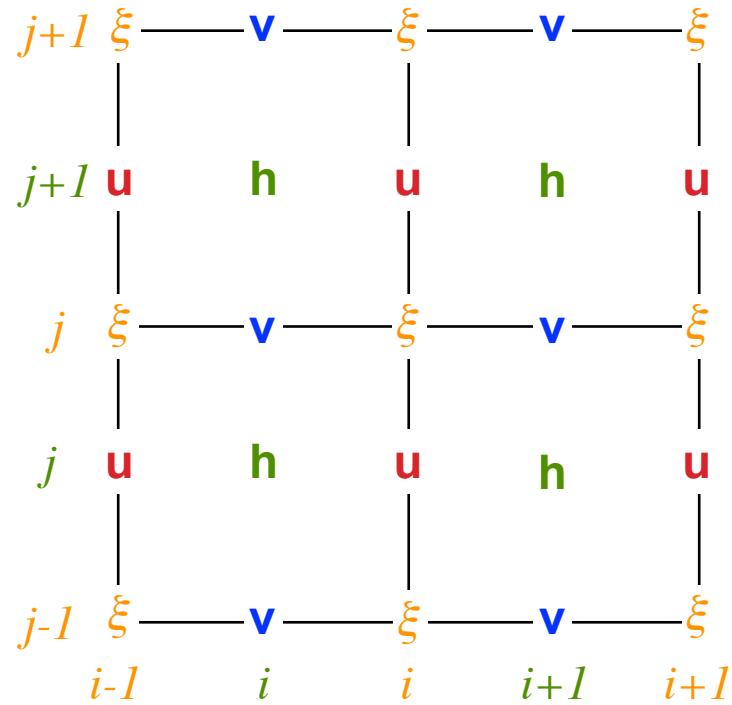


Figure 13.3: *C-grid with points for the zonal velocity u , meridional velocity v , water or air column height h and vorticity ξ .*

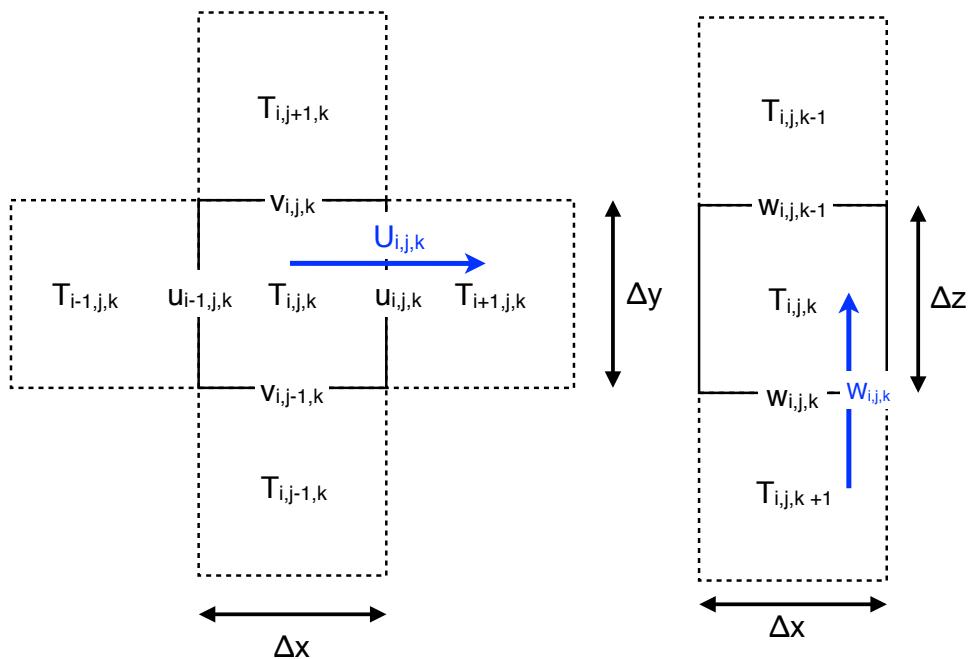


Figure 13.4: *Horizontal (left) and vertical (right) views of the tracer equation applied on a C-grid. The blue arrows illustrate the tracer flux $U_{i,j,k} = u_{i,j,k} \frac{1}{2} (T_{i,j,k} + T_{i+1,j,k}) \Delta y \Delta z$ and $W_{i,j,k} \equiv w_{i,j,k} \frac{1}{2} (T_{i,j,k} + T_{i,j,k+1}) \Delta x \Delta y$.*

Chapter 14

Spherical Harmonics

In modern atmospheric general circulation models (AGCMs), the horizontal spatial representation of scalar dynamic and thermodynamic fields is based on truncated series of spherical harmonic functions, the nature of the underlying two-dimensional horizontal physical grid, also known as a transform grid, is tightly coupled to the parameters of the spherical harmonic expansion itself.

14.1 Spectral methods

The numerical integration methods discussed thus far are based on the discrete representation of the data on a grid or mesh of points covering the space over which a prediction of the variables is desired. Then local time derivatives of the quantities to be predicted are determined by expressing the horizontal and vertical advection terms, sources etc., in finite difference form. Finally, the time extrapolation is achieved by one of many possible algorithms, for example leapfrog. The finite difference technique has a number of associated problems such as truncation error, linear and non-linear instability. Despite these difficulties, the finite difference method has been the most practical method of producing forecasts numerically from the dynamical equations. There is another approach called the spectral method which avoids some of the difficulties cited previously, in particular, non-linear instability; however the method is less versatile and the required computations are comparatively time consuming. In a general sense, the mode of representation of data depends on the nature of the data and the shape of the region over which the representation is desired. An alternative to depiction on a mesh or grid of discrete points is a representation in the form of a series of orthogonal functions. This requires the determination of the coefficients of these functions, and the representation is said to be spectral representation or a series expansion in wave number space. When such functions are used, the space derivatives can be evaluated analytically, eliminating the need for approximating them with finite differences.

The operational ECMWF forecast model uses a spectral technique for its horizontal discretisation. Over the past decade or so this technique has become the most widely used method of integrating the governing equations of numerical weather prediction over hemispheric or global domains. Following the development of efficient transform methods by Eliassen et al. (1970) and Orszag (1970), and the construction and testing of multi-level primitive-equation models (e.g. Bourke, 1974; Hoskins and Simmons, 1975; Daley et al., 1976), spectral models were introduced for operational forecasting in Australia and Canada during 1976.

The technique has been utilized operationally in the USA since 1980, in France since 1982, and in Japan and at ECMWF since 1983. The method is also extensively used by groups involved in climate modelling.

A comprehensive account of the technique has been given by Machenhauer (1979), and a further description of the method has been given by Jarraud and Simmons (1984). Reference should be made to these or other reviews for further discussion of most of the points covered below.

Associated Legendre polynomials

In mathematics, the associated Legendre functions are the canonical solutions of the general Legendre equation

$$\frac{d}{dx} \left[(1 - x^2) \frac{d}{dx} P_n^m(x) \right] = -n(n+1) P_n^m \quad (14.1)$$

where the indices n and m are referred to as the degree and order of the associated Legendre function respectively. The argument can be reparameterized in terms of angles, letting $x = \cos\theta$ and $\theta = \pi/2 - \varphi$, which is presented in Table 14.1 and in Figure 14.1.

Table 14.1: The associated Legendre polynomials P_n^m in $x = \cos\theta$, where $\theta = \pi/2 - \varphi$ is the colatitude.

	$m = 0$	$m = 1$	$m = 2$	$m = 3$
$n = 0$	1			
$n = 1$	$\cos\theta$	$-\sin\theta$		
$n = 2$	$\frac{1}{2}(3\cos^2\theta - 1)$	$-3\cos\theta\sin\theta$	$-3\sin^2\theta$	
$n = 3$	$\frac{1}{2}(5\cos^3\theta - 3\cos\theta)$	$-\frac{3}{2}(5\cos^2\theta - 1)\sin\theta$	$15\cos\theta\sin^2\theta$	$-15\sin^3\theta$

14.2 Spherical harmonics

Global atmospheric models use as basis functions spherical harmonics, which are the eigenfunctions of the Laplace equation on the sphere:

$$\nabla^2 Y_n^m = \frac{1}{a^2} \left[\frac{1}{\cos^2\varphi} \frac{\partial^2 Y_n^m}{\partial\lambda^2} + \frac{1}{\cos\varphi} \frac{\partial}{\partial\varphi} \left(\cos\varphi \frac{\partial Y_n^m}{\partial\varphi} \right) \right] = -\frac{n(n+1)}{a^2} Y_n^m \quad (14.2)$$

The spherical harmonics are products of Fourier series in longitude (λ) and associated Legendre polynomials in latitude (φ):

$$Y_n^m(\lambda, \varphi) = P_n^m(\mu) e^{im\lambda} \quad (14.3)$$

where $\mu = \sin\varphi$, m is the zonal wavenumber and n is the "total" wavenumber in spherical coordinates (as suggested by the Laplace equation).

In the usual application of the method, the basic prognostic variables are vorticity, divergence, temperature, a humidity variable, and the logarithm of surface pressure. Their horizontal representation is in terms

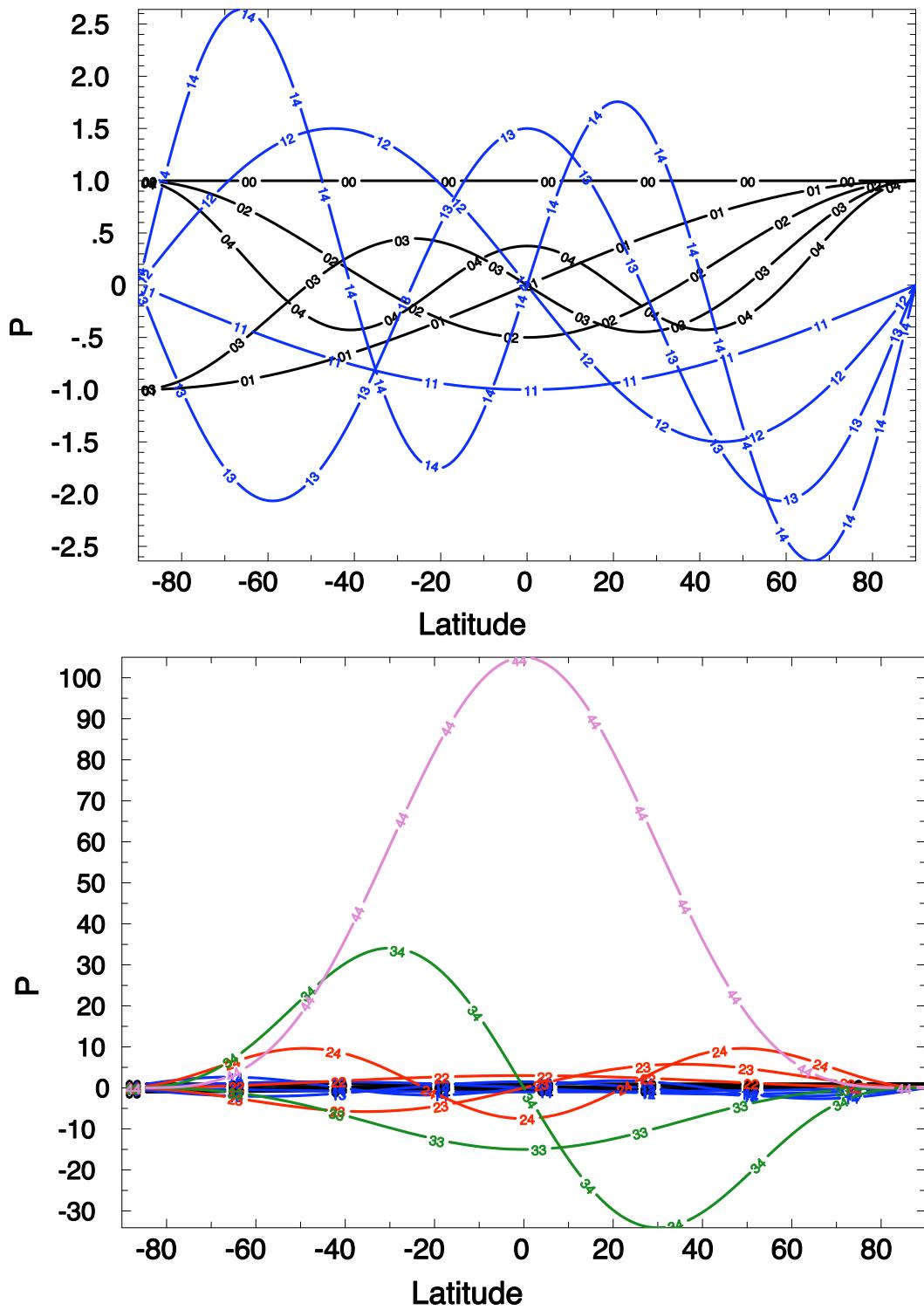


Figure 14.1: The first 15 associated Legendre polynomials P_n^m . The first index on the curves indicate the order of the polynomial and the second index the degree. The top figure shows $m = 1, 2$ for $n = 1, 2, 3, 4$. The bottom figure shows $m = 1, 2, 3, 4$ for $n = 1, 2, 3, 4$. Each colour correspond to a separate order (m) of the polynomial P_n^m .

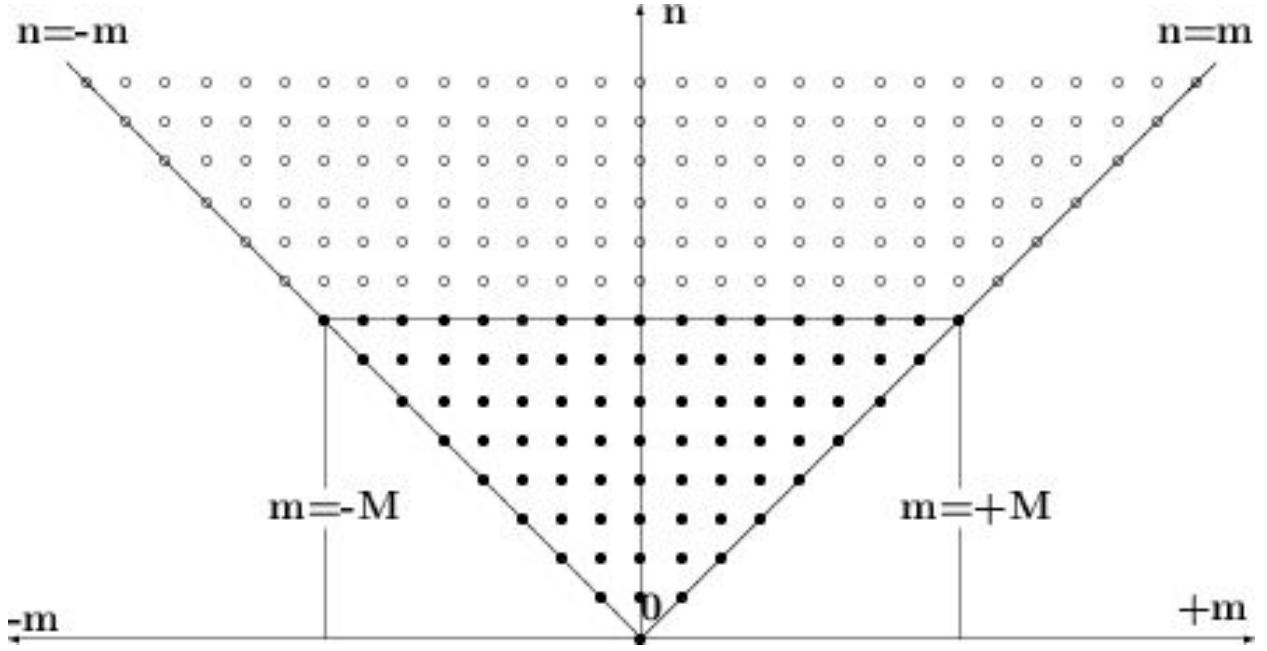


Figure 14.2: The triangular truncation in the (m, n) wavenumber space prescribes a triangular region of spherical harmonic modes indicated by the filled squares. Modes outside of this triangle are set to \circ (open circles).

of truncated series of spherical harmonic functions, whose variation is described by sines and cosines in the east-west and by associated Legendre functions in the north-south. The horizontal variation of a variable U is thus given by

$$U(\lambda, \varphi, t) = \sum_{n=0}^N \sum_{m=-n}^n U_n^m(t) Y_n^m(\lambda, \varphi) \quad (14.4)$$

the spatial resolution is uniform throughout the sphere. This has a major advantage over finite differences based on a latitude-longitude grid, where the convergence of the meridians at the poles requires very small time steps. Although there are solutions for this "pole problem" for finite differences, the natural approach to solve the pole problem for global models is the use of spherical harmonics.

It is becoming increasingly common for the so-called "triangular" truncation of the expansion to be used (Figure 14.2). This truncation is defined by $M = N = \text{constant}$, and gives uniform resolution over the sphere. The symbol "TN" is the usual way of defining the resolution of such a truncation; being the smallest total wave number retained in the expansion. The smallest resolved half-wavelength in any particular direction is then $\pi a/N$ (320 km for T63, 190 km for T106), although the corresponding lateral variation is of larger scale.

Derivatives of a spectral represented variable U are known analytically

$$\frac{\partial U}{\partial \lambda} = \sum_{n=0}^N \sum_{m=-n}^n i m U_n^m Y_n^m$$

and

$$\frac{\partial U}{\partial \varphi} = \sum_{n=0}^N \sum_{m=-n}^n U_n^m \frac{\partial P_n^m}{\partial \varphi} e^{im\lambda}$$

14.3 The spectral transform method

In a spectral model, a variable, $\xi(\lambda, \phi)$, is represented by a truncated series of spherical harmonic functions. This can be expressed as

$$\xi_n^m = \sum_{j=1}^J \xi^m(\phi_j) P_n^m(\phi_j), \quad (14.5)$$

where j is the latitudinal index, $\xi^m(\phi_j)$ is obtained by a fast fourier transform of $\xi(\lambda, \phi)$ and $P_n^m(\phi_j)$ is the associated Legendre functions.

The grid point values are obtained by the inverse transform

$$\xi^m(\phi) = \sum_{n=|m|}^{N(m)} \xi_n^m P_n^m(\phi), \quad (14.6)$$

followed by an inverse fast fourier transform to obtain $\xi(\lambda, \phi)$.

In a spectral model, the explicit time steps and horizontal gradients are performed in spectral space. The tendencies of the equations are however evaluated in grid point space. One benefit when representing the variables in spectral space is that horizontal derivatives are continuously represented, i.e. no finite differencing is needed to evaluate gradients. The methods used to perform the spherical harmonic transforms are however well out of the scope of this project. We have therefore chosen to use an already existing library to carry out the spectral transforms.

14.4 Application to the shallow water equations on a sphere

The momentum and mass continuity equations governing the motion of a rotating, homogenous, incompressible and hydrostatic fluid can be written on vector form as

$$\frac{d\mathbf{V}}{dt} = -f\mathbf{k} \times \mathbf{V} - \nabla\Phi + \nu\nabla^2\mathbf{V}, \quad (14.7)$$

$$\frac{d\Phi}{dt} = -\Phi\nabla \cdot \mathbf{V}, \quad (14.8)$$

where $\mathbf{V} = (u, v)$ is the horizontal velocity vector, Φ is the geopotential height, f is the coriolis parameter and ν is the horizontal diffusion coefficient. Furthermore,

$$\frac{d}{dt} = \frac{\partial}{\partial t} + \mathbf{V} \cdot \nabla, \quad (14.9)$$

and the ∇ operator is defined in spherical coordinates as

$$\nabla = \frac{1}{a \cos \phi} \frac{\partial}{\partial \lambda} + \frac{1}{a} \frac{\partial}{\partial \phi}, \quad (14.10)$$

where a is the earth radius, λ is the longitude coordinate and ϕ is the latitude coordinate.

The above equations describe the shallow-water equations in the u, v, Φ system. In the model, we use another form of these equations. By introducing the relative vorticity ζ and horizontal divergence δ , the equations can be transformed into the ζ, δ, Φ system. We do not derive these equations here since it is relatively straightforward.

By introducing

$$\xi = \mathbf{k} \cdot (\nabla \times \mathbf{V}), \quad (14.11)$$

and

$$\delta = \nabla \cdot \mathbf{V}, \quad (14.12)$$

one can obtain the following set of equations (with $\mu = \sin \phi$)

$$\frac{\partial \eta}{\partial t} = -\frac{1}{a(1-\mu^2)} \frac{\partial}{\partial \lambda} U\eta + \frac{1}{a} \frac{\partial}{\partial \mu} V\eta, \quad (14.13)$$

$$\frac{\partial \delta}{\partial t} = \frac{1}{a(1-\mu^2)} \frac{\partial}{\partial \lambda} U\eta - \frac{1}{a} \frac{\partial}{\partial \mu} V\eta + \nabla^2 \left(\Phi + \frac{U^2 + V^2}{2(1-\mu^2)} \right), \quad (14.14)$$

$$\frac{\partial \Phi}{\partial t} = -\frac{1}{a(1-\mu^2)} \frac{\partial}{\partial \lambda} U\Phi + \frac{1}{a} \frac{\partial}{\partial \mu} V\Phi, \quad (14.15)$$

where now $\eta = \xi + f$ (absolute vorticity including earth rotation) and $(U, V) = (u, v) \cos \phi$.

Chapter 15

Practical computer exercises

15.1 Exercise 1 and 2

The aim of the exercises is to concretise the theory presented in the lectures and to give a better understanding of basic numerical methods. The leapfrog scheme, the Euler forward scheme and the upwind scheme will be studied applied on the advection equation, diffusion equation, and shallow-water equations.

15.2 Theory

The solution interval is in all cases $0 \leq x < 1$, with the periodic boundary condition $u(0, t) = u(1, t)$. The relative error is defined as $\|u - v\|/\|v\|$, where $v(x, t)$ is the analytic solution, and the norm is defined by

$$\|f\| = \left(\Delta x \sum_{i=1}^N |f_i|^2 \right)^{1/2}.$$

The advection equation is

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0,$$

the diffusion equation is

$$\frac{\partial u}{\partial t} - \mu \frac{\partial^2 u}{\partial x^2} = 0,$$

and the shallow-water equations in the one-dimensional case, are

$$\begin{cases} \frac{\partial u}{\partial t} = -g \frac{\partial h}{\partial x}, \\ \frac{\partial h}{\partial t} = -H \frac{\partial u}{\partial x}. \end{cases}$$

15.3 Technical aspects

We will use Fortran 90/95 to solve the numerical exercises. Write your code in a plain text file with the extension `.f90`. To run it, the computers at MISU are equipped with the `gfortran` compiler. If your

code is in a file named `exercise1.f90`, you compile the code by opening a terminal and typing

```
gfortran exercise1.f90
```

Assuming there are no error messages from the code, this creates the program `a.out` which can be run by typing

```
./a.out
```

If you do get error messages, they can sometimes be quite puzzling themselves. To get better information about where the error occurred, you may add the flags `-g -traceback`. Another useful flag is `-fbounds-check` which checks your code for index related errors, for instance if you try to evaluate the 6th element in a vector with only 5 elements. Putting this together:

```
gfortran -g -traceback -fbounds-check exercise1.f90
```

Note: The error checking slows the computations down. However, these exercises should not be so demanding.

It is convenient for everyone if we all use similar notation. I therefore recommend this variable convention, or similar:

<code>u</code>	velocity in x-direction
<code>v</code>	velocity in y-direction
<code>h</code>	the height of the field
<code>i</code>	index in x-direction
<code>j</code>	index in y-direction
<code>n</code>	index in time
<code>nx</code>	number of points in x-direction
<code>ny</code>	number of points in y-direction
<code>nt</code>	number of time steps
<code>dx</code>	grid spacing in x-direction
<code>dy</code>	grid spacing in y-direction
<code>dt</code>	length of time step
<code>g</code>	gravity
<code>H</code>	mean depth

In the following examples, I use `ic` to denote the current i -point, and `ip = ic + 1`, and `im = ic`

- 1. Similar is applied for j and n -points.

A simple way of programming the advection equation discretized on a leapfrog scheme is as follows.

```
DO nc=2,NT-1
```

```
np = nc+1 ; nm = nc-1
```

```

DO ic=2,NX-1

    ip = ic+1 ; im = ic-1

    u(ic,np) = u(ic,nm) - cfl * (u(ip,nc) - u(im,nc))

END DO

END DO

```

Note that the boundary points are not included. These can be iterated separately by including

```

u(1,np) = u(1,nm) - cfl * (u(2,nc) - u(nx-1,nc));
u(nx,np)= u(nx,nm) - cfl * (u(2,nc) - u(nx-1,nc));

```

in the time loop. Another way is to add the IF-statement

```

IF (ic == 1) THEN
    im = NX
ELSE IF (ic == NX) THEN
    ip = 1
END IF

```

in your loop.

For plotting, you may use software of your choice. Computers at MISU are equipped with necessary software for MATLAB and Python. Appendix A contains a function to read a matrix from a Fortran file into MATLAB. It may be copied from this document and used. The function, and a similar for Python, can also be accessed from the course Mondo-page.

15.4 Report

Each student should write two reports. The first should cover Exercises 1–5 and the second Exercise 6. The numerical experiments should be documented with **selected** figures of u (and, for the shallow-water equations, h) as a function of x for appropriate fixed values of t . In some cases 3D plots with u as a function of x, t are also useful. It should be clear what the figures show. The results should be analysed and commented. There is no need to include any code in the report, but please explain what sort of numerical scheme is used for what and why. The reports may be written in either Swedish or English.

Please hand in the first report no later than Friday 9 September 2011 at 15:00, and the second no later than Monday 16 September 2011. If you decide to send the reports via e-mail, please use the .pdf format. Formatted Word (.docx), OpenOffice (.odt), or similar are not accepted.

15.5 Theoretical exercises

Exercise 1 – Leapfrog

Study the leapfrog scheme for the advection equation.

- a) Give the leapfrog scheme (centred scheme with 2nd order accuracy in space and time).
- b) Derive the stability criterion.
- c) Discuss the computational mode and how it can be avoided.

Exercise 2 – Upwind

Study the upwind scheme for the advection equation.

- a) Give the upwind scheme (uncentred scheme with 1st order accuracy in space and time).
- b) Derive the stability condition.

Exercise 3 – Euler forward

Study the Euler forward scheme for the diffusion equation.

- a) Derive the stability criterion of the Euler forward scheme for the diffusion equation:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \mu \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{(\Delta x)^2}.$$

- b) Discuss how the time step should be chosen. Why is it good to choose a smaller time step than that given by the stability criterion? Hint: Study how the amplification factor depends on the wavelength, especially the highest frequencies.

15.6 Experimental exercises

Exercise 4 – Advection

Study the advection equation using a simple numerical model. Write a program that can solve the advection equation with two different schemes: the leapfrog scheme and the upwind scheme.

- a) Run the program with the resolution $\Delta x = 0.1$ and the CFL-numbers 0.9, 1.0 and 1.1. Use a cosine wave as initial condition:

$$u(x, t = 0) = \cos(2\pi x).$$

Solve the problem both with the leapfrog scheme and the upwind scheme, initialising the leapfrog scheme with a single Euler forward step. Plot the results obtained with both schemes and the analytic solution in the same figure. Comment on both the phase error and the amplitude error. Also show how the relative error develops in time.

- b) Run the program with the resolution $\Delta x = 0.02$ and the CFL-number 0.9. Use a cosine pulse as initial condition:

$$u(x, t=0) = \begin{cases} \frac{1}{2} + \frac{1}{2} \cos [10\pi(x - 0.5)], & \text{for } 0.4 \leq x \leq 0.6 \\ 0, & \text{elsewhere} \end{cases}$$

Solve the problem both with the leapfrog scheme and the upwind scheme. Initialise the leapfrog scheme both with a single Euler forward step, and with a constant step: $u_j^2 = u_j^1$. Try to identify the computational mode. (For this, 3D plots of u as a function of x, t are useful.)

Exercise 5 – Diffusion

Study the diffusion equation using a simple numerical model. Write a program that can solve the diffusion equation using the Euler forward scheme. Set $\Delta x = 0.05$ and try with three different time steps: the critical value $(\Delta x)^2/2\mu$ permitted by the stability criterion, a value slightly larger than the critical one, and half the critical value.

Run the program with two different initial conditions:

- rectangular pulse:

$$u_i^1 = \begin{cases} 1, & \text{for } 0 \leq x \leq 0.5 \\ 0, & \text{for } 0.5 < x < 1 \end{cases}$$

- spike: $u = 1$ in a single grid point at $x = 0.5$, and $u = 0$ in all other grid points.

Exercise 6 – 1D Shallow-Water model

Study one-dimensional gravity waves. Gravity waves can be described by the shallow-water equations.

- a) Write a program that solves the 1D shallow water equations given in the Theory section. Discretise using the leapfrog scheme on an unstaggered grid. Set $H = g = 1$ (thus $c = \sqrt{gH} = 1$) to simplify the system. Run the program with $\Delta x = 0.025$ and the CFL-number 0.9. (The CFL-number is defined as $c \cdot \Delta t / \Delta x$.) Initialise the leapfrog scheme with a single Euler forward step, and use the following initial conditions which describes a travelling pulse:

$$\begin{aligned} h(x, t=0) &= \begin{cases} \frac{1}{2} + \frac{1}{2} \cos[10\pi(x - 0.5)], & \text{for } 0.4 \leq x \leq 0.6 \\ 0, & \text{elsewhere} \end{cases} \\ u(x, t=0) &= h(x, t=0) \end{aligned}$$

Interpret the results physically.

- b) Rewrite the program using an staggered grid, with h and u defined in different points. Run the program and try to find the stability limit for the time step. Then choose the time step 10 % smaller than the stability limit, and use the travelling pulse as initial condition. Use *both* the resolution of the previous subexercise and *half* of that resolution. Compare the accuracy of the solution with the previous solution obtained on an A-grid. Discuss the difference between the two results and the result from the previous subexercise. Note that Δx is the distance between two height points (and also two velocity points).

Exercise 7

The aim of this exercise is to solve the linearised shallow water equations, 1D and 2D, with some standard numerical methods, and study atmospheric and oceanographic processes.

Theory

The linearized shallow water equations in two dimensions including rotation and physical parameterisations are

$$\frac{\partial u}{\partial t} = fv - g \frac{\partial h}{\partial x} + F_u \quad (15.1)$$

$$\frac{\partial v}{\partial t} = -fu - g \frac{\partial h}{\partial y} + F_v \quad (15.2)$$

$$\frac{\partial h}{\partial t} = -H \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) + F_h, \quad (15.3)$$

Physical parameterisations (so called *model physics*), F_u , F_v and F_h may for instance be horizontal diffusion and/or drag forcing. Realistic values of g , H and f should be used in order to gain understanding

of real atmospheric and oceanographic problems. For instance, the North Atlantic ocean is 3000 km across and 4 km deep. Diffusion or drag is to be added to the velocity fields only. The 1D case is obtained by simply neglecting $\partial/\partial y$ -derivatives.

Technical aspects

The model is to be programmed in Fortran 90/95 standard. Try to make the code clear and well structured, and use common notations (u for zonal velocity, dt for time step, and so on). See Appendix A for hints.

For plotting you may use a technique of your choice, e.g. MATLAB/Octave, Python, or other. Included in Appendix B you will find a MATLAB function that reads Fortran unformatted files. However, the function may need to be tailored to fit your files.

Report

In order to complete the shallow water exercise each student should write a concise report. This report should live up to general MISU demands (self-contained, structured, etc.). Include selected figures and illustrations. Long mathematical derivations or parts of the code are not welcome in the report. Try to write the report so that someone knowledgeable about the course content, but has not taken the course, might understand.

Please hand in the report no later than 30 September 2011. If you choose to send it via e-mail, please use .pdf format, and not .docx, .odt, or similar.

15.7 1D Shallow water model exercises

Here you should develop a 1D shallow water model that includes rotation, relaxation zone, and Asselin filtering on a staggered grid.

- a) Write the code for the model. To do this you need to add an extra velocity variable to the 1D model developed earlier in the course and rewrite your schemes to a staggered form. Also, the model should have solid boundaries, and not periodic as before.
- b) Run the model without rotation ($f = 0$) and with the following initial conditions:

$$u(x, 0) = 0 \quad (15.4)$$

$$v(x, 0) = 0 \quad (15.5)$$

$$h(x, 0) = h_0 e^{-(x/L_W)^2}, \quad (15.6)$$

where $-L/2 \leq x \leq L/2$ and $L_W \approx L/7$. Describe your results.

- c) Study Rossby adjustment to geostrophic balance (Holton chapter 7.6) by running the same experiment with a realistic rotation (what should f be according to your total integration time, $T = N_T \Delta t$?).
- d) Change your initial conditions to:

$$u(x, 0) = 0 \quad (15.7)$$

$$v(x, 0) = 0 \quad (15.8)$$

$$h(x, 0) = \begin{cases} h_0 & \text{if } |x| \leq L_W \\ 0 & \text{if } |x| > L_W \end{cases} \quad (15.9)$$

What occurs? Explain the new phenomena. What happens at the boundaries?

- e) How do gravity waves affect the results? Try running the same experiment with a relaxation at the boundaries and explain the difference. The relaxation can be parameterised in the following way:

$$(u_d, v_d) = \left(1 - d \frac{\Delta t}{\tau}\right) (u, v) \quad (15.10)$$

where τ is a relaxation time-scale (s^{-1}) and d is the damping value. Construct the function so that $d = 0$ in the interior of the system, and d approaches 1 as you approach the boundaries. The function may be either trigonometric or linear. Here, it is sufficient to set $\tau = \Delta t$ for simplicity. Give an explanation of what the relaxation might represent. Why do limited area models need relaxation?

- f) Apply the Asselin-filter to the previous experiment. Do you see any difference?

15.8 2D Shallow water model exercises

Exercise 0: Fortran code of a 2D shallow water model

Write and program a linear 2D shallow water model with finite differences based on the following equations

$$\frac{\partial u}{\partial t} - f_0 v = -g \frac{\partial h}{\partial x} \quad (15.11)$$

$$\frac{\partial v}{\partial t} + f_0 u = -g \frac{\partial h}{\partial y} \quad (15.12)$$

$$\frac{\partial h}{\partial t} + D_0 \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) = 0 \quad (15.13)$$

Use Leap-frog time scheme but with an Euler forward for the first time step on an Arakawa C-grid and an Asselin filter (with appropriate value of γ). Let use scales similar to the world oceans $L \sim 5 \cdot 10^6$ m, $D_0 \sim 4000$ m, $g = 9.81$ m s $^{-2}$, $f_0 = 10^{-4}$ s $^{-1}$. Set fixed Δx and Δy , and let Δt be determined by a CFL number, phase speed, and $\max(\Delta x, \Delta y)$.

Let NX, NY be determined by L , and NT by T_{\max} , which is the simulation period.

This way you may alter the resolution without changing domain size (as done in a real GCM). Construct the main time loop so that the model does not store the fields for every time step. To simulate open boundaries you should implement a sponge (relaxation) zone to eliminate gravity waves.

Exercise 1: Geostrophic adjustment

The purpose of this lab is to understand what controls the evolution of a disturbance initially at rest on a rotating plane, more commonly known as geostrophic adjustment. To study this phenomena, you will use shallow water model from previous exercise.

- a) Implement an initial disturbance in the model, consisting of a discontinuity in h that is symmetric in y . Starting from the linearized shallow water equations, derive the final steady state of sea surface height and the velocities. Run the model until the system varies only very little in time (steady state), and verify your results with what you obtained analytically. Tweak the parameters, and discuss which are the most important.
- b) Write down the analytical expressions for the total, kinetic, and potential energies. Implement a Gaussian disturbance of the sea surface height instead of the discontinuity. Let the disturbance be smaller than the domain, but make sure it is larger than the resolution. Study how the energies (kinetic, potential, and total) vary in time. Study at least two cases, one shallow case ($D_0 \sim 500$ m) and one deep ($D_0 \sim 10000$ m). Give a physical explanation of the results.
- c) Run the model long enough so that the system comes to steady state. The system can be considered to be in steady state when the energy of the system changes very little in time. Repeat the experiments for other depths and sizes of the disturbance. How do the results change? Try to explain the changes and compare with theory.

Exercise 2: The role of solid boundaries

The aim of this assignment is to gain understanding of the influence solid boundaries have on geophysical flows. Although applicable to some situations in the atmosphere, the influence of lateral boundaries is more important in the world oceans. In this assignment you will use a simple shallow water model with solid boundaries, and look at how an initial disturbance evolves, and how a system initially at rest evolves as you add a forcing to it.

a) Evolution of an initial disturbance

Use a Gaussian disturbance of reasonable size of the h -field centered at one of the boundaries so that the disturbance is “cut” by the wall. Run the model.

- (a) Describe the results and connect to theory. Do any particular kind of waves show up? How can you identify those kinds of waves? (Phase speed, shape...)
- (b) Compare model results with theoretical results. Look at a cross section of the wave, and estimate its phase speed. Try different settings e.g. changing the resolution.

Try the experiments for two systems: North Atlantic ($L = 5 \cdot 10^7$ m, $H = 4000$ m), and the Baltic Sea ($L = 5 \cdot 10^6$ m, $H = 80$ m).

b) Coastal upwelling

Model how the sea surface height changes as you add wind forcing at the surface to a closed ocean basin. Use the shallow water model, and let the system be at rest initially. Implement a wind forcing. The wind should be parallel to the coast and constant, but be zero some distance from it. To find reasonable parameters, think of the Ekman number (in the momentum equations, the Ekman number is the fraction of the stress and the Coriolis terms. If that fraction is large, then wind stress is important.)

- (a) Look at the evolution of the velocity and height field on short time scales ($T \sim 1/f$). Can you identify a transport toward the coastline? Compare the results with theoretical predictions. Think about how the Ekman spiral is represented in a barotropic model.
- (b) Look at longer time scales. Describe and explain. Use theory. Does the model resemble a real ocean basin?

Exercise 3: Topography and β -plane

In the past two assignments we have used constant depth and constant Coriolis force. The aim of this assignment is to study and understand the kind of waves that are generated when $D(y)$ or $f(y)$ are sloping planes of the kind $D = D_0 + \alpha y$ and $f = f_0 + \beta y$. For this assignment you should use a model with periodic boundary conditions in x and open boundaries (sponge) in y . The initial disturbance should be in geostrophic balance to avoid gravity waves.

- a) Derive phase speed and group velocity for Rossby waves in this linear system.

- b) Define f and D . Run the model first with β -plane and constant D , and then with sloping topography and constant f . Use theory to show that they give similar end results.
- c) Pick one of the two cases above ($f = f_0 + \beta y$, $D = D_0$ or $f = f_0$, $D = D_0 + \alpha y$). Run the model for about 50 days. Describe and explain the evolution of the system. Connect to theory. What kind of waves develop? Do they have any distinguishing properties? In which direction and at what speed do the waves move? How does this compare with theory?

Try 4-5 different values of the wavenumber, and compare phase speed and group velocity to theoretical values. (Tip: To change wavenumber, change disturbance width).

- d) Try some other surface of D or f that might be more realistic or just interesting.

Exercise 4: Depth-Integrated Ocean Circulation

The aim of this assignment is to study Stommel's and/or Munk's theories for barotropic large scale ocean circulation. To do this, implement some features from the previous assignments and model the large scale circulation in a closed basin. The model should include a wind forcing, β -plane, and bottom friction. The wind forcing should be constant in x and sinusoidal in y . Use parameters suitable for a world ocean basin, e.g. North Atlantic, North Pacific

Run the model without any friction and comment on the results. What effect does $\beta = \partial f / \partial y$ have on the system? What effect does the wind forcing have? Do the results resemble theory? Why / why not? Model limitations?

- a) Use Rayleigh linear friction to parameterize bottom friction. Give a physical explanation of this kind of friction. Describe and explain the circulation using Stommel's theory.
- b) Change the friction parameter, Γ , and describe how the results change.
- c) Use Laplace friction to parameterize bottom friction. Give a physical explanation of this kind of friction. Describe and explain the circulation using Munk's theory.
- d) Change the friction parameter, A , and describe how the results change.

If you wish to make the model more realistic: Use a non-rectangular grid where each grid box has a certain resolution in $(\Delta\theta, \Delta\phi)$ (lon/lat) coordinates, and $\Delta x = \cos \phi \Delta\theta$, $\Delta y = \Delta\phi$. Remember that forcing (wind, friction, diffusion ...) should then be weighted to the grid box area. Also let $f = 2\Omega \sin \phi$.

General structure of the code

```

PROGRAM structure_of_code
!!-----
!!  

!!  

!!  

!!      General structure of a well written shallow water model in Fortran 90.  

!!      Features: Staggered grid, rotation, diffusion, and relaxation schemes.  

!!  

!!      Author: Joakim Kjellsson,  

!!      At:      Department of Meteorology, Stockholm University  

!!  

!!      Last change: 14 July 2011  

!!  

!!  

!!      PS. It is good and common practice to always write a  

!!      few lines about what the code does and how at the top.  

!!-----
```

```
IMPLICIT NONE
```

```
!!-----
```

```
!! Think about:  

!!      Explain each parameter in words and units.  

!!      Use common notation.  

!!      Do not write too much in one line.
```

```
!! Physical constants:
```

```
REAL*4,  PARAMETER      :: f  = -----, & ! Coriolis parameter [s-1]
&                      g  = ----- ! Gravity           [m s-2]
```

```
!! Model parameters:
```

```
REAL*4,  PARAMETER      :: H  = -----, & ! Mean depth [m]
&                      mu = ----- ! Diffusion coeff. [m2 s-1]
```

```
!! Grid
```

```
INTEGER*4, PARAMETER     :: NX = -----, & ! Number of i-points
&                      NY = -----, & ! Number of j-points
```

```

&                                NT = -----      ! Number of time steps

REAL*4,  PARAMETER             :: dx = -----, & ! Zonal grid spacing [m]
&                               dy = -----, & ! Merid. grid spacing [m]
&                               dt = ----- ! Time step [s]

!! Save data to file
CHARACTER*200                  :: outFile = ----- ! Name of output file

!! Work variables
INTEGER*4                        :: ic, ip, im, jc, jp, jm, nc, nm, np
REAL*4                            :: du, dv, dh

!! Data matrices
REAL*4, DIMENSION(NX,NY,NT) :: u, v, h

!!---
!! Initial condition
!!---

!! Main loop

!! You might want to indent the code inside the loop to make it
!! easier to see where the loop starts and ends.

DO nc=-----
    np = nc+1
    nm = nc-1

    DO jc=-----
        jp = jc+1
        jm = jc-1

        DO ic=-----
            ip = ic+1
            im = ic-1

```

```
! Reset
du = 0.
dv = 0.
dh = 0.

! Coriolis
du = du + -----
dv = dv + -----

! Sea surface height gradient
du = du + -----
dv = dv + -----

! Continuity
dh = dh + -----

! Diffusion
du = du + -----
dv = dv + -----

! Relaxation
du = du + -----
dv = dv + -----

! New time step
u(ic,jc,nt) = -----
v(ic,jc,nt) = -----
h(ic,jc,nt) = -----

! Asselin filtering
u(ic,jc,nc) = -----
v(ic,jc,nc) = -----
h(ic,jc,nc) = -----

! Update time index

END DO

! Don't forget to take care of boundary conditions,
! zonally periodic, no-slip, or other.
```

```
END DO
```

```
END DO
```

```
! !-----
```

```
!! Save the data to file
```

```
! !-----
```

```
END PROGRAM
```


Appendix A

Exams at Stockholm University

2005

Tentamen i numeriska metoder inom meteorologin och oceanografin, ME3580, torsdagen den 3 november 2005, kl. 9.00-15.00 av *Kristofer Döös och Erland Källén, Meteorologiska institutionen, Stockholms universitet*

Consider the partial differential equation (PDE):

$$\frac{\partial T}{\partial t} = A \nabla^2 T \quad \text{where } A > 0 \text{ and } \nabla^2 \equiv \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

- 1) Find a solution for $T(x, y, t)$ and describe what it could represent physically. (N.B. Nothing to do with numerical methods!) (2p)
- 2) Is the PDE elliptic, hyperbolic or parabolic? Show how you determine this. Clue: use only one dimension in space. (2p)
- 3) Discretise the time derivative with centred difference and the laplace with centred difference at time step n-1. (2p)
- 4) Derive the order of accuracy of the two schemes with help of Taylor series. (2p)
- 5) Make a stability analysis. (2p)
- 6) Is there a numerical mode? Why? How do you suppress it? (2p)

Consider the non-linear shallow water equations:

$$\frac{\partial u}{\partial t} - \xi h v = - \frac{\partial H}{\partial x} \tag{A.1}$$

$$\frac{\partial v}{\partial t} + \xi h u = - \frac{\partial H}{\partial y} \tag{A.2}$$

$$\frac{\partial h}{\partial t} + \frac{\partial (hu)}{\partial x} + \frac{\partial (hv)}{\partial y} = 0 \tag{A.3}$$

where $\xi \equiv \left(f + \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \right) / h$ and $H \equiv gh + \frac{1}{2} (u^2 + v^2)$

- 7) Discretise the equations [A.1-A.3](#) with centred schemes on a C-grid (See Figure [A.1](#) at bottom of the page). (4p).
- 8) Find a second way to discretise the terms ξhv and ξhu . (2p)
- 9) What type of initial condition is needed and how can one integrate the first time step? (2p)
- 10) What type of boundary conditions are needed? (2p)
- 11) Describe a numerical model code in matlab or fortran based on these discretised equations with IM grid cells in the x direction and JM in the y direction and how it is integrated in time for NT time steps. (2p)
- 12 a) Linearise eqs [A.1-A.3](#) around a state of rest with an average shallow water depth D . (3p)
- 12 b) Determine the types of wave motions that can be described with the linearised system given in 12a). If you cannot solve 12a) you can get the answer to 12a), but in this case you will not be given any points for 12a. (3p)

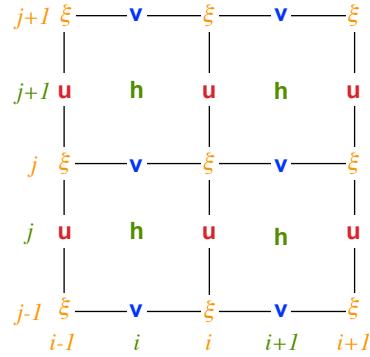


Figure A.1: C-grid.

Formulas: $e^{i\alpha} - e^{-i\alpha} = 2i \sin \alpha$ and $e^{i\alpha} + e^{-i\alpha} = 2 \cos \alpha$ and $2 \sin^2(\alpha/2) = 1 - \cos \alpha$

Taylor series for $f(x)$ about $x = a$ is $f(x) = f(a) + (x - a)f'(a) + \frac{1}{2}(x - a)^2f''(a) + \dots + \frac{1}{n!}(x - a)^n f^{(n)}(a)$
 $a \frac{\partial^2 u}{\partial x^2} + b \frac{\partial^2 u}{\partial x \partial y} + c \frac{\partial^2 u}{\partial y^2} + d \frac{\partial u}{\partial x} + e \frac{\partial u}{\partial y} + fu + g = 0$. If $b^2 - 4ac < 0$ then elliptic; if $= 0$ then parabolic; if > 0 then hyperbolic .

2007

Tentamen i numeriska metoder inom meteorologin och oceanografin, ME3580, torsdagen den 1 november 2007, kl. 9.00-15.00 av Kristofer Döös och Heiner Körnich, Meteorologiska institutionen, Stockholms universitet. Tentamen på totalt 30 p. För godkänt krävs 15 p. och väl godkänt 22,5 p.

Consider the shallow water equations in vector-form:

$$\frac{\partial}{\partial t} \vec{u}_h + \nabla \frac{\vec{u}_h \cdot \vec{u}_h}{2} + (\xi + f) \vec{k} \times \vec{u}_h = -g \nabla h \quad (\text{A.4})$$

$$\frac{\partial H}{\partial t} + \nabla \cdot (\vec{u}_h H) = 0 \quad (\text{A.5})$$

with the following definitions:

$$\text{the total depth } H = h - h_b, \quad (\text{A.6})$$

$$\text{the surface elevation } h_b = h_b(x, y), \quad (\text{A.7})$$

$$\text{the relative vorticity } \xi = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}, \quad (\text{A.8})$$

$$\text{and the horizontal velocity vector } \vec{u}_h = \begin{pmatrix} u \\ v \end{pmatrix} \quad (\text{A.9})$$

- 1) Linearise the equations (A.4) and (A.5) for small amplitude motion around a mean depth $D = D(x, y)$. Derive the solution for the time-independent linear motion. Show that the divergence of this motion is zero and discuss this motion for a spatially varying surface elevation h_b . (4p)
- 2) Use the linear equations to derive the tendency equation for the total energy $E_{tot} = D|\vec{u}_h|^2/2 + gh^2/2$. Assume a closed region with no normal flow across the boundaries: $\vec{u}_h \cdot \vec{n} = 0$, where the vector \vec{n} which is the orthogonal unity vector on the boundary contour. Argue why the area integrated total energy is conserved. Hint: Apply the divergence theorem (2p)

$$\iint_A dA \nabla \cdot \vec{v} = \oint \vec{v} \cdot \vec{n} dr. \quad (\text{A.10})$$

- 3) Discretise the linearised shallow water equations with a flat bottom on a C-grid (see Fig. ??) with Euler forward finite differences in time and centred finite differences in space. (3p)
- 4) What initial and boundary conditions are required. (1p)
- 5) Is there a numerical mode? How can you suppress a numerical mode? (1p)
- 6) Simplify the discretised equations from task 3 for the case when $f = 0$ and for no y dependency. i.e. when $\partial/\partial y = 0$. (1p)
- 7) Make a stability analysis on the equations from previous task. (3p).
- 8) Calculate the truncation error of the Euler forward finite difference. (2p)
- 9) Derive the frequency equation by substituting wave solutions into the linearised shallow water equations for the case when $\partial/\partial y = 0$. Note that this is for the continuous equations with no finite differences. (3p)
- 10) Same as previous task but for the discretised equations with finite differences in space on a C-grid. Use

continuous differences in time. Discuss the effects of discretisation on the phase speed. (3p)

11) Find two possible discretisations on the C-grid of the non-linear advection terms in the shallow water momentum equations (both the zonal and the meridional momentum equations). (3p)

12) In what way is a hydrostatic 3-D model (GCM) different from a shallow water model? What extra equations and terms are there? Discretise these extra terms and equations. (4p).

Formelsamling: $e^{i\alpha} - e^{-i\alpha} = 2i \sin \alpha$ and $e^{i\alpha} + e^{-i\alpha} = 2\cos \alpha$ and $2\sin^2(\alpha/2) = 1 - \cos \alpha$

Taylorserien för $f(x)$ runt $x = a$ är $f(x) = f(a) + (x - a)f'(a) + \frac{1}{2}(x - a)^2f''(a) + \dots + \frac{1}{n!}(x - a)^n f^{(n)}(a)$

2008

Tentamen i numeriska metoder inom meteorologin och oceanografin, ME3580, torsdagen den 30 oktober 2008, kl. 9.00-15.00 av *Kristofer Döös och Heiner Körnich, Meteorologiska institutionen, Stockholms universitet*. Tentamen på totalt 30 p. För godkänt krävs 15 p. och väl godkänt 22,5 p.

Consider the linearised shallow water equations in the quasi-one-dimensional case where u , v and h do not depend on y so that the shallow water equations are reduced to

$$\frac{\partial u}{\partial t} - fv = -g \frac{\partial h}{\partial x} \quad (\text{A.11})$$

$$\frac{\partial v}{\partial t} + fu = 0 \quad (\text{A.12})$$

$$\frac{\partial h}{\partial t} + H \frac{\partial u}{\partial x} = 0 \quad (\text{A.13})$$

- 1) Derive the dispersion relationship (frequency equation) by substituting wave solutions into [A.11- A.13](#). (2p)
- 2) Discretise Eqs. [A.11- A.13](#) in space on a B-grid (see Figure on next page) and keep the continuous derivatives in time. (1p)
- 3) Derive the numerical dispersion relationship for these discretised equations in space. (3p)
- 4) Describe the differences between these model waves and the analytical continuous waves (2p)
- 5) Discretise the equations in task 2 with centred finite differences in time (leap-frog). (1p)
- 6) Make a stability analysis on the equations in previous task. (5p).
- 7) Derive the numerical dispersion relationship from the disrcetised equation from task 5 (this time with finite differences in time). (4p)
- 8) What initial and boundary conditions are required? How can you start to integrate the first time step? (1p)
- 9) Is there a numerical mode? Why ? How can you suppress a numerical mode? (1p)
- 10) Calculate the truncation error of the finite difference in space of the continuity equation from task 2. (2p)
- 11) Consider the continuity equation with pressure as vertical coordinate

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial \omega}{\partial p} = 0 \quad (\text{A.14})$$

Discretise it on a B-grid and explain how it can be integrated. (2p)

- 12) Derive the linearised potential vorticity equation

$$\frac{\partial}{\partial t} \left[\xi - \frac{f}{H} h \right] = 0 \quad (\text{A.15})$$

from the equations [A.11- A.13](#). Discuss the relationship between the vorticity and the surface elevation implied by equation [A.15](#). (3p)

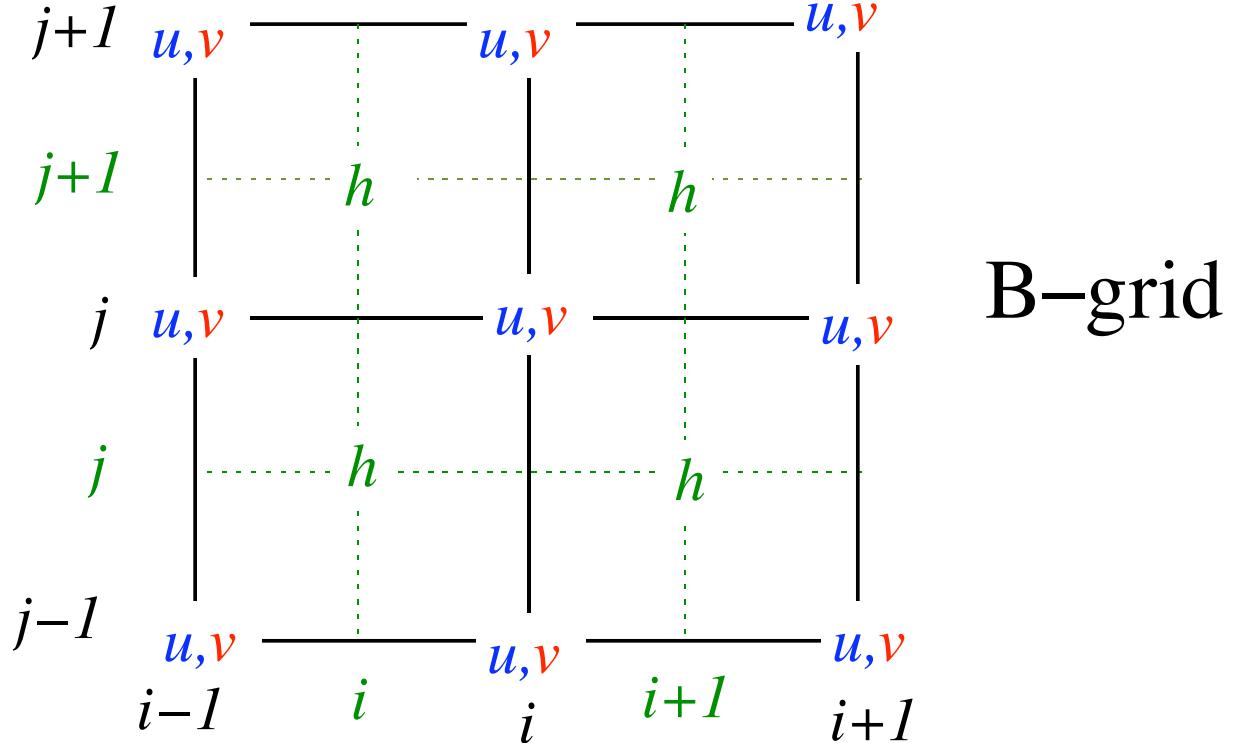
- 13) Assume that the real part of the wave solution for $\eta = h - H$ is given as

$$\eta = \eta_0 \cos(kx - \sigma t). \quad (\text{A.16})$$

Express the wave solution for the vorticity and the divergence in terms of the amplitude η_0 . Which motion type (divergence or vorticity) has a larger amplitude? Use the fact $\sigma > f$. (3p)

Formulas: $e^{i\alpha} - e^{-i\alpha} = 2i \sin \alpha$ and $e^{i\alpha} + e^{-i\alpha} = 2 \cos \alpha$ and $2 \sin^2(\alpha/2) = 1 - \cos \alpha$

The Taylor series for $f(x)$ around $x = a$ is $f(x) = f(a) + (x-a)f'(a) + \frac{1}{2}(x-a)^2f''(a) + \dots + \frac{1}{n!}(x-a)^n f^{(n)}(a)$



2011

Tentamen i numeriska metoder inom meteorologin och oceanografin, MO7004, tisdagen den 27 september 2011, kl. 9.00-15.00 av *Kristofer Döös, Meteorologiska institutionen, Stockholms universitet*. Tentamen på totalt 100 poäng. För betyg A krävs minst 90 p., B ges mellan 89 och 80p, C: 65-79, D:55-64, E:50-54, Fx:45-49 och under 45 p. ges betyg F.

Skriv på svenska eller engelska!

Consider the shallow water equations on the form:

$$\frac{\partial u}{\partial t} - fv = -g \frac{\partial h}{\partial x} + A \nabla^2 u \quad (\text{A.17})$$

$$\frac{\partial v}{\partial t} + fu = -g \frac{\partial h}{\partial y} + A \nabla^2 v \quad (\text{A.18})$$

$$\frac{\partial h}{\partial t} + D \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) = 0 \quad (\text{A.19})$$

where D är is the average depth.

- a) Discretise the equations on a C-grid. Choose the finite time differencing as you would in your favourite shallow water model. See the attached C-grid Figure on reverse page! (15p)
- b) What type of boundary conditions are needed for the discretised equations? (5p)
- c) What initial conditions are required for the discretised equations? How can you start to integrate the first time step? (5p)
- d) Derive the order of accuracy of the Laplacian diffusion terms. (10p)
- e) Make a stability analysis for the non-rotative ($f=0$), inviscid ($A=0$) and no y -variations ($\partial/\partial y = 0$) case. (15p)
- f) Make a stability analysis for the discretised Eq. A.18 when $f=0$ and there is no y -variation ($\partial/\partial y = 0$). Be careful in the choice of time step of the viscous term! (10p)
- g) Is there a numerical mode? Why? How do you suppress it? (5p)
- h) Derive the dispersion relationship (frequency equation) by substituting wave solutions into Eq. A.17 - A.19 for the inviscid ($A=0$) and no y -variations ($\partial/\partial y = 0$) case. (10p)
- i) Same as previous but for the discretised case in space. Use continuous derivatives in time. Discuss the differences with the previous non-discretised analytical case. (15p)
- j) Add the non-linear terms to Eq. A.17 - A.19 and apply them on the C-grid. Use the horizontal absolute potential vorticity $\xi \equiv \left(f + \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \right) / h$ and $H \equiv gh + \frac{1}{2} (u^2 + v^2)$ (10p)

Formelsamling:

$$e^{i\alpha} - e^{-i\alpha} = 2i \sin \alpha$$

$$e^{i\alpha} + e^{-i\alpha} = 2\cos \alpha$$

$$2 \sin^2(\alpha/2) = 1 - \cos \alpha$$

Taylorserien för $f(x)$ runt $x = a$ är $f(x) = f(a) + (x-a)f'(a) + \frac{1}{2}(x-a)^2f''(a) + \dots + \frac{1}{n!}(x-a)^n f^{(n)}(a)$

$$x^2 + \alpha x + \beta = 0 \Rightarrow x = -\alpha/2 \pm \sqrt{\alpha^2/4 - \beta}$$

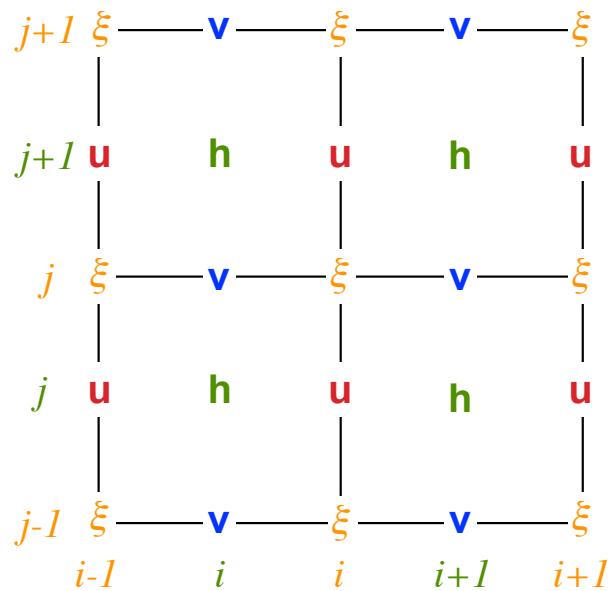


Figure A.2: The C-grid

2012

Tentamen i numeriska metoder inom meteorologin och oceanografin, MO7004, onsdag den 3 oktober 2012, kl. 9.00-15.00 av *Kristofer Döös, Meteorologiska institutionen, Stockholms universitet*. Tentamen på totalt 100 poäng. För betyg A krävs minst 90 p., B ges mellan 89 och 80p, C: 65-79, D:55-64, E:50-54, Fx:45-49 och under 45 p. ges betyg F.

The tracer Equation can under certain circumstances be written

$$\frac{\partial T}{\partial t} + \nabla \cdot (\mathbf{V}T) = K_H \nabla^2 T + K_V \frac{\partial^2 T}{\partial z^2} + C \quad (\text{A.20})$$

- a) Derive from where the second term (advection term) comes from! Which extra equation do you need and under what conditions is this extra equation valid? (14p)
- b) Discretise Equation A.20 with, centred finite differences in both time and space on a C-grid! (14p)
- c) Derive the order of accuracy of the Laplacian diffusion terms in Equation A.20! (14p)
- d) Simplify Equation A.20 by excluding the non-linear advection terms, no source term and only one dimension in space. Discretise with centred finite differences in both time and space. Perform a stability analysis! Note that this discretised equation should be completely centred! Is this a good choice? Any better choice of the time steps? (15p)
- e) The time derivative and the advection part of the equation can be simplified in one dimension if the velocity is set to a constant phase speed so

$$\frac{\partial T}{\partial t} + c \frac{\partial T}{\partial x} = 0 \quad (\text{A.21})$$

Derive an expression for the ratio between the phase speed c and the computational phase speed C_D . (14p)

- f) Adapt and discretise Equation A.20 so it would fit into your shallow water model you have constructed during this course. (14p)
- g) Discretise the Equation A.20 in such a way the size of the grid boxes can vary in space so that $\Delta x_{i,j}$, $\Delta y_{i,j}$ and Δz_k have grid indices. Hint: Write the tracer advection and tracer diffusion on flux form. (15p)

PTO (Please Turn Over)

Formelsamling:

$$e^{i\alpha} - e^{-i\alpha} = 2i \sin \alpha$$

$$e^{i\alpha} + e^{-i\alpha} = 2\cos \alpha$$

$$2 \sin^2 (\alpha/2) = 1 - \cos \alpha$$

Taylorserien för $f(x)$ runt $x = a$ är $f(x) = f(a) + (x-a)f'(a) + \frac{1}{2}(x-a)^2f''(a) + \dots + \frac{1}{n!}(x-a)^n f^{(n)}(a)$

$$x^2 + \alpha x + \beta = 0 \Rightarrow x = -\alpha/2 \pm \sqrt{\alpha^2/4 - \beta}$$

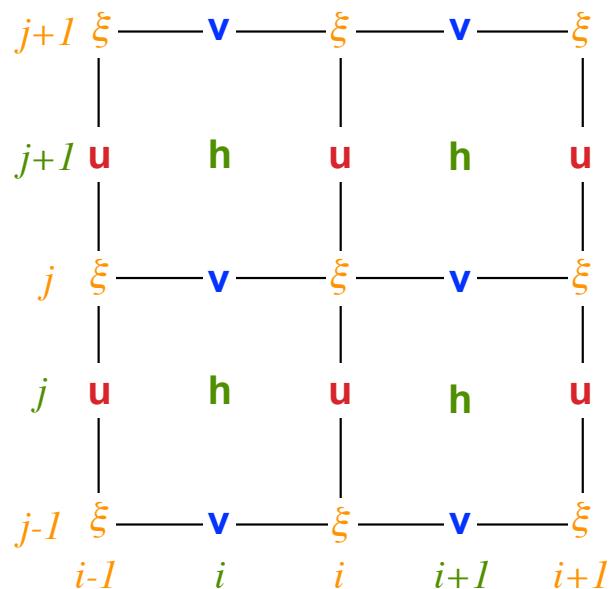


Figure A.3: The C-grid

2013

Tentamen i numeriska metoder inom meteorologin och oceanografin, MO7004, onsdag den 22 mars 2013, kl. 9.00-15.00 av *Kristofer Döös, Laurent Brodeau och Saeed Falahat, Meteorologiska institutionen, Stockholms universitet*. Tentamen på totalt 100 poäng. För betyg A krävs minst 90 p., B ges mellan 89 och 80p, C: 65-79, D:55-64, E:50-54, Fx:45-49 och under 45 p. ges betyg F.

Consider the shallow water equations on the form:

$$\frac{\partial u}{\partial t} - fv = -g \frac{\partial h}{\partial x} \quad (\text{A.22})$$

$$\frac{\partial v}{\partial t} + fu = -g \frac{\partial h}{\partial y} \quad (\text{A.23})$$

$$\frac{\partial h}{\partial t} + H \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) = 0 \quad (\text{A.24})$$

where H is the average depth and f is constant.

- a) Derive an Equation for h from Equations (1), (2) and (3). (10p)
- b) Discretise Equations (1), (2) and (3) with, centred finite differences in both time and space on a C-grid! (10p)
- c) Derive the dispersion relationship (frequency equation) from Equations (1),(2) and (3) by applying wave solutions. (15p)
- d) Same as previous but for the discretised case in space. Use continuous derivatives in time. Discuss the differences with the previous non-discretised analytical case. (15p)
- e) Derive the order of accuracy for the two finite differences of discretised Equation (1) (15p)
- f) Make a stability analysis for the discretised Equations when $f=0$. Note that you should keep the y -dependency. Set $\Delta x = \Delta y$ in the end to see how the stability criterion is simplified to. (15p)
- g) Discretise the Equations (1), (2) and (3) using a semi-implicit scheme **centred on time level n**. Terms containing **spatial partial derivatives** must be evaluated as the average of the $(n-1)$ th and $(n+1)$ th time levels. β the implicitness parameter is set to 0.5. For clarity, the use of operators such as δ_x , δ_{xx} and ∇_*^2 is strongly encouraged. Express the discretised version of Equation (3) in terms of h only and suggest a method to integrate the equations in time despite their implicitity. (10p)
- h) Make a stability analysis for the above discretised Equations when $f=0$ and there is no y -dependency. (10p)

Formelsamling:

$$e^{i\alpha} - e^{-i\alpha} = 2i \sin \alpha$$

$$e^{i\alpha} + e^{-i\alpha} = 2\cos \alpha$$

$$2 \sin^2 (\alpha/2) = 1 - \cos \alpha$$

Taylorserien för $f(x)$ runt $x = a$ är $f(x) = f(a) + (x-a)f'(a) + \frac{1}{2}(x-a)^2f''(a) + \dots + \frac{1}{n!}(x-a)^n f^{(n)}(a)$

$$x^2 + \alpha x + \beta = 0 \Rightarrow x = -\alpha/2 \pm \sqrt{\alpha^2/4 - \beta}$$

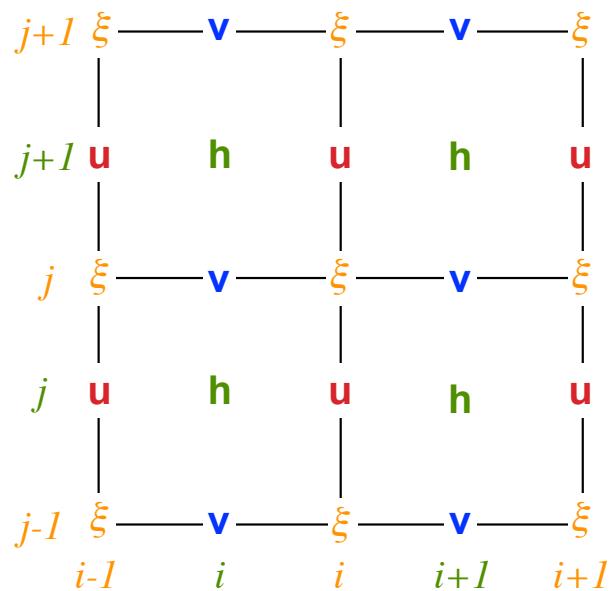


Figure A.4: The C-grid

2014

Tentamen i numeriska metoder inom meteorologin och oceanografin, MO7004, onsdag den 22 mars 2013, kl. 9.00-15.00 av *Kristofer Döös, Laurent Brodeau och Saeed Falahat, Meteorologiska institutionen, Stockholms universitet*. Tentamen på totalt 100 poäng. För betyg A krävs minst 90 p., B ges mellan 89 och 80p, C: 65-79, D:55-64, E:50-54, Fx:45-49 och under 45 p. ges betyg F.

Consider the shallow water equations on the form:

$$\frac{\partial u}{\partial t} = -g \frac{\partial h}{\partial x} \quad (\text{A.25})$$

$$\frac{\partial v}{\partial t} = -g \frac{\partial h}{\partial y} \quad (\text{A.26})$$

$$\frac{\partial h}{\partial t} + H \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) = 0 \quad (\text{A.27})$$

where H is the average depth.

- a) Derive an Equation for h from Equations (1), (2) and (3). (10p)
- b) Discretise Equations (1), (2) and (3) with, centred finite differences in both time and space on a B-grid! (10p)
- c) Derive the dispersion relationship (frequency equation) from Equations (1),(2) and (3) by applying wave solutions. (15p)
- d) Same as previous but for the discretised case in space. Use continuous derivatives in time. Discuss the differences with the previous non-discretised analytical case. (15p)
- e) Derive the order of accuracy for the two finite differences of discretised Equation (1) (15p)
- f) Make a stability analysis for the discretised Equations. Note that you should keep the y-dependency. Set $\Delta x = \Delta y$ in the end to see how the stability criterion is simplified to. (15p)
- g) Discretise the Equations (1), (2) and (3) using a semi-implicit scheme **centred on time level n**. Terms containing **spatial partial derivatives** must be evaluated as the average of the $(n - 1)$ th and $(n + 1)$ th time levels. β the implicitness parameter is set to 0.5. For clarity, the use of operators such as δ_x , δ_{xx} and ∇_*^2 is strongly encouraged. Express the discretised version of Equation (3) in terms of h only and suggest a method to integrate the equations in time despite their implicitity. (10p)
- h) Make a stability analysis for the above discretised Equations when $f=0$ and there is no y-dependency. (10p)

Formelsamling:

$$e^{i\alpha} - e^{-i\alpha} = 2i \sin \alpha$$

$$e^{i\alpha} + e^{-i\alpha} = 2\cos \alpha$$

$$2 \sin^2(\alpha/2) = 1 - \cos \alpha$$

Taylorserien för $f(x)$ runt $x = a$ är $f(x) = f(a) + (x-a)f'(a) + \frac{1}{2}(x-a)^2f''(a) + \dots + \frac{1}{n!}(x-a)^n f^{(n)}(a)$

$$x^2 + \alpha x + \beta = 0 \Rightarrow x = -\alpha/2 \pm \sqrt{\alpha^2/4 - \beta}$$

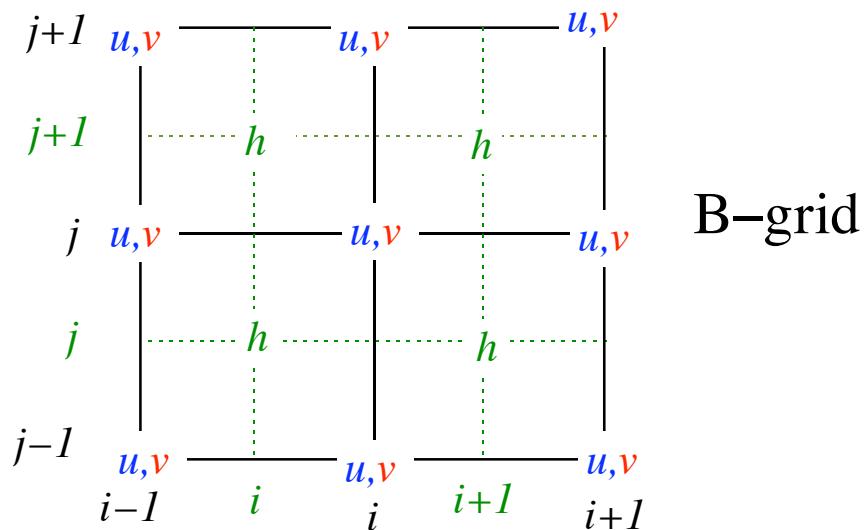


Figure A.5: The B-grid

2015

Tentamen i numeriska metoder inom meteorologin och oceanografin, MO7004, onsdag den 18 februari 2015, kl. 9.15-14.15 av *Kristofer Döös, Laurent Brodeau och Aitor Aldama Campino, Meteorologiska institutionen, Stockholms universitet*. Tentamen på totalt 100 poäng. För betyg A krävs minst 90 p., B ges mellan 89 och 80p, C: 65-79, D:55-64, E:50-54, Fx:45-49 och under 45 p. ges betyg F.

Consider the heat equation in 2D:

$$\frac{\partial T}{\partial t} + \mathbf{V} \cdot \nabla T = K \nabla^2 T \quad (\text{A.28})$$

where \mathbf{V} is the horizontal velocity vector and T the temperature, K the horizontal diffusion coefficient. Set the divergence of the velocity to zero as if the continuity equation was stationary ($\nabla \cdot \mathbf{V} = 0$).

- a) Discretise Equation A.28 with, centred finite differences in both time and space on a C-grid! (15p)
- b) Derive the order of accuracy of the Laplacian diffusion terms in Equation A.28! (15p)
- c) The time derivative and the advection part of the equation can be simplified in one dimension if the velocity is set to a constant phase speed so

$$\frac{\partial T}{\partial t} + c \frac{\partial T}{\partial x} = 0 \quad (\text{A.29})$$

Derive an expression for the ratio between the phase speed c and the computational phase speed C_D . (15p)

- d) In absence of advection the heat equation can instead be simplified in one dimension to

$$\frac{\partial T}{\partial t} = K \frac{\partial^2 T}{\partial x^2} \quad (\text{A.30})$$

Make a stability analysis with the same centred finite differences as in previously in task 1. Would you choose this discretisation or would you choose another discretisation? Argue and make a stability analysis if necessary on your choice of numerical schemes! (20p)

- e) Discretise the 1D version (along x) of Equation (1) based on a semi-implicit scheme, on the unstructured **A-grid**, in a way that the whole scheme is **centred on time level $n+1/2$** . Discretize the time derivative using the forward Euler scheme. Terms containing **spatial partial derivatives** must be evaluated as the average of the n^{th} and $(n + 1)^{\text{th}}$ time levels. This means that β , the implicitness parameter, is set to 0.5. Assume the velocity \mathbf{V} to be constant ($\mathbf{V} \equiv U_0$). Write down the discretized implicit solution of the problem in a simplified and clear way, with unknown terms (level $n + 1$) at the left hand side and known terms (level n) at the right hand side. Suggest at least a method to integrate this equation in time despite its implicitness. (15p)

- f) By assuming Dirichlet boundary conditions at the two extremities of the 1D domain $T(x = 0, t) = T_0$ and $T(x = L, t) = T_L$, write down the linear system that must be solved at each time step to solve the equation you obtained in the former question. The linear system should be written in the following form:

$$\mathbf{A} \cdot X = B$$

where \mathbf{A} is a square matrix, X the vector containing the unknown values of T at time $n + 1$ and B the vector containing known values of T . (20p)

Formelsamling:

$$e^{i\alpha} - e^{-i\alpha} = 2i \sin \alpha$$

$$e^{i\alpha} + e^{-i\alpha} = 2\cos \alpha$$

$$2\sin^2(\alpha/2) = 1 - \cos \alpha$$

Taylorserien för $f(x)$ runt $x = a$ är $f(x) = f(a) + (x - a)f'(a) + \frac{1}{2}(x - a)^2f''(a) + \dots + \frac{1}{n!}(x - a)^n f^{(n)}(a)$

$$x^2 + \alpha x + \beta = 0 \Rightarrow x = -\alpha/2 \pm \sqrt{\alpha^2/4 - \beta}$$

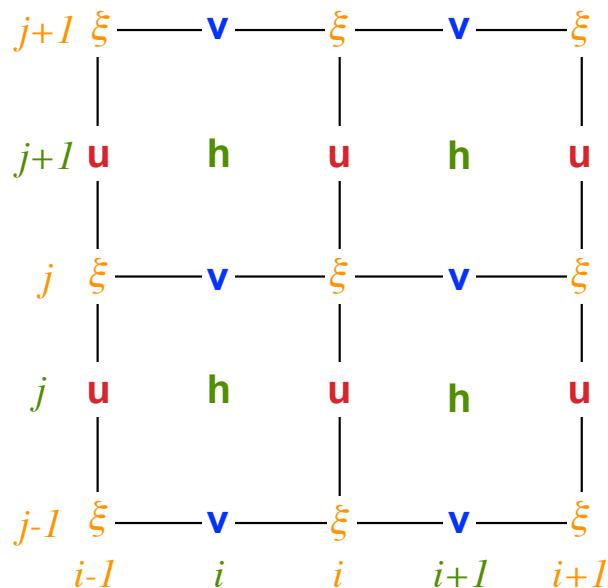


Figure A.6: The C-grid

2016

Bibliography

- Asselin, R.: Frequency filter for time integrations, Monthly Weather Review, 100, 487–490, 1972. [44](#)
- Batteen, M. L. and Han, Y.-J.: On the computational noise of finite-difference schemes used in ocean models, Tellus, 33, 387–396, 1981. [63](#)
- Bjerknes, V.: Das Problem der Wettervorhersage: betrachtet vom Standpunkte der Mechanik und der Physik, 1904. [9](#), [11](#)
- Bryan, K. and Cox, M. D.: A numerical investigation of the oceanic general circulation, Tellus A, 19, 1967. [11](#)
- Charney, J. G., Fjörtoft, R., and Neumann, J. v.: Numerical integration of the barotropic vorticity equation, Tellus, 2, 237–254, 1950. [11](#), [28](#)
- Courant, R., Friedrichs, K., and Lewy, H.: Über die partiellen Differenzengleichungen der mathematischen Physik, Mathematische Annalen, 100, 32–74, 1928. [11](#), [33](#)
- Crank, J. and Nicolson, P.: A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type, in: Mathematical Proceedings of the Cambridge Philosophical Society, vol. 43, pp. 50–67, Cambridge Univ Press, 1947. [28](#)
- Döös, B. R. and Eaton, M. A.: Upper-Air Analysis over Ocean Areas1, Tellus, 9, 184–194, 1957. [11](#)
- Fischer, G.: Ein numerisches Verfahren zur Errechnung von Windstau und Gezeiten in Randmeeren1, Tellus, 11, 60–76, 1959. [11](#)
- Hansen, W.: Theorie zur Errechnung des Wasserstandes und der Strömungen in Randmeeren nebst Anwendungen1, Tellus, 8, 287–300, 1956. [11](#)
- Manabe, S. and Bryan, K.: Climate calculations with a combined ocean-atmosphere model, Journal of the Atmospheric Sciences, 26, 786–789, 1969. [11](#)
- Mesinger, F. and Arakawa, A.: Numerical methods used in atmospheric models, GARP technical report 17, WMO/ICSU. Geneva, Switzerland, 1, 17, URL http://twister.ou.edu/CFD2003/Mesinger_ArakawaGARP.pdf, 1976. [58](#)
- Nycander, J. and Döös, K.: Open boundary conditions for barotropic waves, Journal of Geophysical Research: Oceans, 108, 2003. [64](#)

- Orlanski, I.: A simple boundary condition for unbounded hyperbolic flows, *Journal of Computational Physics*, 21, 251 – 269, doi:[http://dx.doi.org/10.1016/0021-9991\(76\)90023-1](http://dx.doi.org/10.1016/0021-9991(76)90023-1), URL <http://www.sciencedirect.com/science/article/pii/0021999176900231>, 1976. [64](#)
- Persson, A.: Early operational Numerical Weather Prediction outside the USA: an historical Introduction. Part 1: Internationalism and engineering NWP in Sweden, 1952–69, *Meteorological Applications*, 12, 135–159, 2005a. [11](#), [12](#)
- Persson, A.: Early operational Numerical Weather Prediction outside the USA: an historical introduction: Part II: Twenty countries around the world, *Meteorological Applications*, 12, 269–289, 2005b. [12](#)
- Persson, A.: Early operational Numerical Weather Prediction outside the USA: an historical introduction Part III: Endurance and mathematics–British NWP, 1948–1965, *Meteorological Applications*, 12, 381–413, 2005c. [12](#)
- Richardson, L. F.: The Approximate Arithmetical Solution by Finite Differences of Physical Problems Involving Differential Equations, with an Application to the Stresses in a Masonry Dam, *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 210, 307–357, doi:[10.1098/rsta.1911.0009](https://doi.org/10.1098/rsta.1911.0009), URL <http://rsta.royalsocietypublishing.org/content/210/459-470/307>, 1911. [11](#)
- Richardson, L. F.: Weather prediction by numerical process, Cambridge University Press, 1922. [11](#)
- Robert, A. J.: The integration of a low order spectral form of the primitive meteorological equations(Spherical harmonics integration of low order spectral form of primitive meteorological equations), *METEOROLOGICAL SOCIETY OF JAPAN, JOURNAL*, 44, 237–245, 1966. [44](#)
- Sadourny, R.: The dynamics of finite-difference models of the shallow-water equations, *Journal of the Atmospheric Sciences*, 32, 680–689, 1975. [66](#)
- Simmons, A. J. and Burridge, D. M.: An Energy and Angular-Momentum Conserving Vertical Finite-Difference Scheme and Hybrid Vertical Coordinates, *Monthly Weather Review*, 109, 758–766, 1981. [102](#)
- Smagorinsky, J.: General circulation experiments with the primitive equations: I. the basic experiment*, *Monthly weather review*, 91, 99–164, 1963. [11](#)
- Wiin-Nielsen, A.: The birth of numerical weather prediction, *Tellus A*, 43, 36–52, 1991. [11](#), [12](#)
- Williams, P. D.: A proposed modification to the robert-asselin time filter*, *Monthly Weather Review*, 137, 2538–2546, 2009. [44](#), [45](#)