Numerical methods
Revised March 2001

By **R. W. Riddaway (revised by M. Hortal)**

Table of contents

1. SOME INTRODUCTORY IDEAS

1.1 Introduction

The use of numerical models for weather prediction involves the solution of a set of coupled non-linear partial differential equations. In general these equations describe three important dynamical processes—advection, adjustment (how the mass and wind fields adjust to one another) and diffusion. In this note we will concentrate upon how to solve simple linear one-dimensional versions of the equations which describe each of these processes. These can be conveniently derived ftom the shallow-water equations in which

(a) the earth's rotation is ignored

(b) there is no motion in the y -direction

(c) there are no variations in the y -direction

The set of equations we are going to consider is then

$$\begin{cases} \dfrac{\partial u}{\partial t} = -u\dfrac{\partial u}{\partial x} - g\dfrac{\partial h}{\partial x} + \dfrac{\partial}{\partial x}\left(K\dfrac{\partial u}{\partial x}\right) \\[2ex] \dfrac{\partial h}{\partial t} = \underbrace{-u\dfrac{\partial h}{\partial x}}_{\text{advection}}\underbrace{-h\dfrac{\partial u}{\partial x}}_{\text{adjustment}} + \underbrace{\dfrac{\partial}{\partial x}\left(K\dfrac{\partial h}{\partial x}\right)}_{\text{diffusion}} \end{cases}$$

Linearising the equations about a basic state (u_0, H) constant in space and time gives

$$\frac{\partial u}{\partial t} + u_0\frac{\partial u}{\partial x} = -g\frac{\partial h}{\partial x} + \frac{\partial}{\partial x}\left(K\frac{\partial u}{\partial x}\right)$$

$$\frac{\partial h}{\partial t} + u_0\frac{\partial h}{\partial x} = -H\frac{\partial u}{\partial x} + \frac{\partial}{\partial x}\left(K\frac{\partial h}{\partial x}\right)$$

where u and h are the perturbations in the x -component of velocity and the height of the free surface. The parts of these equations describing the three main processes are as follows.

Advection

$$\frac{\partial u}{\partial t} + u_0\frac{\partial u}{\partial x} = 0$$

$$\frac{\partial h}{\partial t} + u_0\frac{\partial h}{\partial x} = 0$$

In general the one-dimensional linearised advection equation can be written as

$$\frac{\partial \varphi}{\partial t} + u_0\frac{\partial \varphi}{\partial x} = 0$$

As well as investigating the linear advection equation, it is necessary to consider the non-linear problem. For this we use the one-dimensional non-linear advection equation

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

Adjustment

$$\frac{\partial u}{\partial t} + g\frac{\partial h}{\partial x} = 0$$

$$\frac{\partial h}{\partial t} + H\frac{\partial u}{\partial x} = 0$$

These are often called the one-dimensional linearised gravity-wave equations.

Diffusion

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(K\frac{\partial u}{\partial x}\right)$$

$$\frac{\partial h}{\partial t} = \frac{\partial}{\partial x}\left(K\frac{\partial h}{\partial x}\right)$$

The general form of the one-dimensional diffusion equation (with constant eddy diffusivity K) is

$$\frac{\partial \varphi}{\partial t} = K\frac{\partial^2 \varphi}{\partial x^2}$$

Many of the ideas and techniques used to solve these simplified equations can be extended to deal with the full primitive equations.

Finite difference techniques were, historically, the most common approach to solving partial differential equations (PDE's) in meteorology but, since a number of years now, spectral techniques have become very useful in global models and local representations such as the finite elements or the local spectral method are becoming increasingly researched, mainly in connection with the massive parallel-processing machines.

1.2 Classification of PDE's

Most meteorological problems fall into one of three categories—these are referred to as boundary value problems, initial value problems and eigenvalue problems. In this note we will be mainly concerned with initial value problems.

1.2 (a) Boundary value problems.

The problem is to determine φ in a certain domain D, where the differential equation governing φ within D is $\mathrm{L}(\varphi) = f$, and $\mathrm{B}(\varphi) = g$ on the boundary; here L and B are differential operators.

$$\begin{aligned} \mathrm{L}(\varphi) &= f \quad \text{in solution domain D} \\ \mathrm{B}(\varphi) &= g \quad \text{on the boundary} \end{aligned}$$

Typical examples of this type of problem involve the solution of the Helmholtz or Poisson equations.

1.2 (b) Initial value problems.

These are propagation problems in which we want to predict the behaviour of a system given the initial conditions. This is done by solving the differential equation $\mathrm{L}(\varphi) = f$ within D where the initial condition is $\mathrm{I}(\varphi) = h$ and the prescribed conditions on the open boundaries are $\mathrm{B}(\varphi) = g$. Problems involving the solution of the advection equation, gravity wave equations and diffusion equation fall into this category.

1.2 (c) Eigenvalue problems.

The problem is to determine λ and φ such that $\mathrm{L}(\varphi) = \lambda\varphi$ is satisfied within domain D. Problems of this type occur in baroclinic instability studies.

An alternative method of classification has been devised for linear second order PDE's of the form

$$a\frac{\partial^2\varphi}{\partial\xi^2} + 2b\frac{\partial^2\varphi}{\partial\xi\partial\eta} + c\frac{\partial^2\varphi}{\partial\eta^2} + 2d\frac{\partial\varphi}{\partial\xi} + 2e\frac{\partial\varphi}{\partial\eta} + f\varphi = 0$$

The classification is based on the properties of the characteristics (not discussed here) of the equation. We find that there are three basic types of equation: hyperbolic, parabolic and elliptic. Hyperbolic and parabolic equations are initial value problems, whereas an elliptic equation is a boundary value problem.

TABLE 1. CHARACTERISTICS OF HYPERBOLIC, PARABOLIC AND ELLIPTIC PDES

Type	Characteristic directions	Condition	Example
hyperbolic	Real	$b^2 - ac > 0$	Wave equation
parabolic	Imaginary	$b^2 - ac = 0$	Diffusion equation
elliptic	non-existent	$b^2 - ac < 0$	Poisson equation

1.3 Existence and uniqueness

Let us consider an initial value problem for a real function of time only

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f(t, y); \quad y(t_0) = y_0 \tag{1}$$

where f is a known function of the two variables.

We could be unable to solve explicitly Eq. (1) and therefore we ask ourselves the following questions.

1) How are we to know that the initial value problem (1) actually has a solution?

2) How do we know that there is only one solution $y(t)$ of (1)?

3) Why bother asking the first two questions?

The answer to the third question is that our equation is just an approximation to the physical problem we want to solve and, therefore, if it has not one and only one solution it cannot be a good representation of the physical process; that is, the problem is not well posed. On the other hand, if the problem is well posed we can hope to get by some means a solution close enough to the real solution even if we are unable to find the exact solution or the exact solution is not an analytical one. The situation is then exactly the same as in the theory of limits where it is often possible to prove that a sequence of functions $y_u(t)$ has a limit without our having to know what this limit is, and we can use any member of the sequence from a place onwards to represent an approximation to the limit.

This suggests the following algorithm for proving the existence of a solution $y(t)$ of (1):

(a) Construct a sequence of functions $y_u(t)$ that come closer and closer to solving (1);

(b) Show that the sequence of functions $y_u(t)$ has a limit $y(t)$ on a suitable interval $t_0 \le t \le t_0 + \alpha$;

(c) Prove that $y(t)$ is a solution of (1) on this interval.

This is the so called successive approximations or Picard iterates. By this method, it is possible to show the following

Picard's Theorem:

Let f and $\dfrac{\partial f}{\partial y}$ be continuous in the rectangle R: $t_0 \le t \le t_0 + a$, $|y - y_0| \le b$. Then the initial-value problem

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f(t, y); \quad y(t_0) = y_0$$

has a unique solution $y(t)$ on the interval $t_0 \le t \le t_0 + a$.

Unfortunately, the equations involved in meteorology are not ordinary differential equations but partial differential equations and the proof of existence and uniqueness of its solution is not as straightforward as applying Picard's theorem. Nevertheless, the example serves to illustrate the importance or proving an existence and uniqueness theorem as a hunting license to go looking for this solution or for a close approximation to it.

For the linear equations we are dealing with in this set of lectures, we can find the general analytic solution to the equation and, therefore, do not need to prove the existence theorem. But it will still be nice to prove the uniqueness of it, given a suitable set of initial and boundary conditions. Nevertheless, this falls outside the scope of the course and we will only hope that such a uniqueness could be proven.

1.4 Discretization

The non-linear equations describing the evolution of the atmosphere do not have analytical solutions even if the problem is well posed. An analytical function is the most perfect way of representing a given physical field as it gives us the value of this field in any of the infinite number of points of space and at any instant in time.

If an analytical solution does not exist, we have to resort to numerical techniques to find a certain approximation to the true solution of the system of equations, that is, we have to use computers. But computers cannot deal with infinite amounts of numbers, so we have to represent our meteorological fields by a finite number of values. This is called the discretization process.

As a simple example consider the linear one-dimensional evolutionary problem

$$\frac{\partial \varphi}{\partial t} = \mathrm{H}(\varphi) \tag{2}$$

where H is a linear differential space operator (though the techniques considered can also be applied to non-linear problems). We will assume that φ is specified at $(N+1)$ gridpoints in our domain $(0 \le x \le L)$, and that there are suitable boundary conditions for φ. We now want to consider how we can numerically find $\partial\varphi/\partial t$, given the grid point values φ_j—that is we only consider the space discretization.

The common way of tackling this problem is to simply express the derivatives which occur on the right hand side of (2) in terms of the differences between the gridpoint values of φ. This is the finite difference technique, which will be discussed at length later. Note that when using this technique no assumption is made about how φ varies between the grid points.

An alternative approach is to expand φ in terms of a finite series of $(N+1)$ linearly independent functions e_m, where $m = m_1, \ldots m_2$,where $m_2 - m_1 = N$, so that

$$\varphi = \sum_{m=m_1}^{m_2} \varphi_m(t) e_m(x) \tag{3}$$

This series is only an exact solution of the original PDE in very special circumstances. Therefore, when (3) is substituted into (2) there will be a residual R

$$R = \sum_m \frac{\mathrm{d}\varphi_m}{dt} e_m - \sum_m \varphi_m H(e_m)$$

We now want to choose the time derivatives $\mathrm{d}\varphi_m/dt$ by minimising R in some way. One method for doing this is to use a least square approach—we then have to minimise

$$I = \int R \;\mathrm{d}x$$

with respect to the time derivatives. Carrying this out and rearranging gives:

$$\sum_l \frac{\mathrm{d}\varphi_l}{dt} \int e_m e_l \mathrm{d}x = \sum_l \varphi_l \int e_m H(e_l) \mathrm{d}x; \quad m = m_1, \ldots m_2 \tag{4}$$

This equation could also be derived using the Galerkin method in which we set

$$\int R \psi_i \mathrm{d}x = 0; \quad i = 1, 2, \ldots N+1$$

where the ψ_i can be any set of linearly independent test functions. If the expansion functions are used as test functions we get (4). Since the expansion functions are known (3) can be used to provide the expansion coefficients φ_m given the gridpoint values φ_j. Also the integrals

$$\int e_m e_l \mathrm{d}x \text{ and } \int e_m H(e_l) \mathrm{d}x$$

in (4) can be calculated exactly for all possible values of m and l. Therefore, (4) reduces to a set of coupled ordinary differential equations that can be solved for the $\mathrm{d}\varphi_m/dt$ given the φ_m. The complete solution is then

$$\frac{\partial\varphi}{\partial t} = \sum_m \frac{\mathrm{d}\varphi_m}{dt} e_m$$

This general approach is often referred to as the Galerkin technique.

For the case where the expansion functions are orthogonal we end up with $(N+1)$ uncoupled ordinary differential equations for the rate of change of the expansion coefficients

$$\frac{\mathrm{d}\varphi_m}{\mathrm{d}t} = \sum_l \varphi_l \int e_m H(e_l)\mathrm{d}x; \quad m = m_1, \ldots m_2$$

An example of this kind of approach is the spectral method in which a Fourier series is used. In this case (3) becomes

$$\varphi = \sum_{m=-M}^{M} \varphi_m(t)\exp\left\{\mathbf{i}\left(\frac{2\pi x}{L}\right)m\right\}$$

where the φ_m are complex Fourier coefficients and $M = N/2$.

With spherical geometry, it is natural to use spherical harmonics.

For the spectral method the expansion functions are global. An alternative approach is to use a set of expansion functions which are only locally non- zero; this is the basis of the finite element method. With this method we still have a set of nodes (i.e. grid points) with nodal values φ_j, but now we assume that the variation in φ within an element (i.e. a set of nodes) can be described by a low-order polynomial, with the requirement that there is continuity in φ between adjacent elements. The simplest case is to assume a linear variation in φ across an element which has only two nodes (the end points); i.e. a linear piecewise fit. Then (3) becomes

$$\varphi = \sum_{j=1}^{N+1} \varphi_j(t) e_j(x)$$

where the φ_j are the nodal values and the $e_j(x)$ are "hat" functions (sometimes called chapeau functions) as in Fig. 1 .

The expansion functions are not orthogonal, but they are nearly so; therefore the integrals which occur in (4) can be easily evaluated. The result of this process is to produce a set of coupled equations from which the time derivative can be determined.

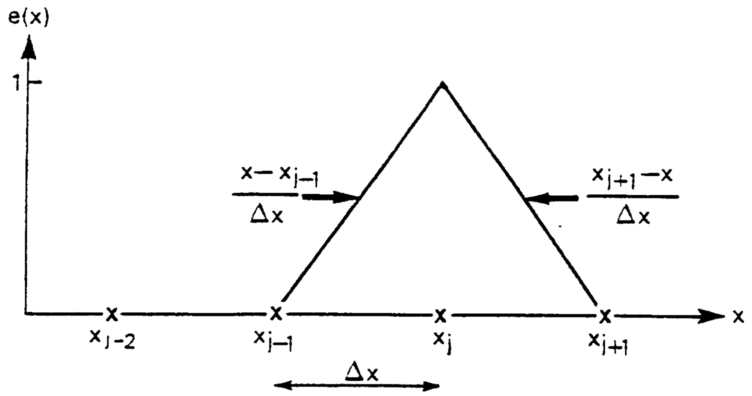

e(x)
1
$\frac{x - x_{j-1}}{\Delta x}$
$\frac{x_{j+1} - x}{\Delta x}$
x_{j-2}
x_{j-1}
x_j
x_{j+1}
x
Δx

Figure 1. Representation of a "hat" function or piecewise linear finite element.

An interesting feature of the Galerkin technique is that if the original equation (2) has a quadratic invariant (e.g. energy)

$$\frac{\partial E}{\partial t} = 0 \quad \text{with } E = \int_0^L \frac{\varphi^2}{2} dx$$

then this property is retained when a Galerkin approximation is made for the spatial variations (when finite differences are used there is no guarantee of this happening). However, note that quadratic invariance is lost when time stepping is introduced.

The spectral and finite element methods will be dealt with in Sections 6 and 7, but now we will concentrate upon the finite difference technique.

1.5 Convergence, consistency and stability

(a) Convergence: a discretized solution of a differential equation is said to be convergent if it approaches the solution of the continuous equation when the discretization becomes finer and finer (that is the distance between grid points in the finite difference technique becomes smaller, or the number of basis functions in the spectral or the finite element techniques becomes higher).

We would like to ensure convergence, but this is difficult to do. However there is a theorem which overcomes this problem, but before it can be stated we need to introduce two more definitions.

(b) Consistency: a discretization technique is consistent with a PDE if the truncation error of the discretized equation tends to zero as the discretization becomes finer and finer.
Note that consistency means only that the discretized equation is similar to the continuous equation but this does not guarantee by itself that the corresponding solutions are close to each other (convergence).
Consistency is easy to test. Suppose $\tilde{\varphi}_j^n$ is the true solution of the PDE (2) at position x_j and time t_n. This solution is now substituted into the finite difference equation and Taylor expansions used to express everything in terms of the behaviour of $\tilde{\varphi}$ at position x_j and time t_n. Rearranging the equation then gives:

$$\left.\frac{\partial\tilde{\varphi}}{\partial t}\right|_j^n = H(\varphi)|_j^n + E$$

If the truncation error E approaches zero as the grid length and time step approach zero, the scheme is consistent Hereafter consistency will be assumed without comment

(c) Stability: a discretization scheme is stable if their solutions are uniformly bounded functions of the initial state for any value of Δt small enough, that is if the numerical solution does not go to infinity as t increases.
There are various techniques for testing stability, three of which will be described later.

The consistency and stability of discretization schemes can be investigated; therefore, we can check if the scheme is convergent by making use of the following theorem.

<u>The Lax–Richtmeyer Theorem</u>

If a discretization scheme is consistent and stable, then it is convergent (the converse is also true).

2. Finite differences

2.1 Introduction

Suppose we have an interval L which is covered with $N+1$ equally spaced grid points. The gridlength is then $\Delta x = L/N$ and the grid points are at $x_j = (j-1)\Delta x,\ j = 1, 2, \ldots N+1$. Let the value of φ at x_j be represented by φ_j.

We are now going to derive expressions which can be used to give an approximate value of a derivative at a grid point in terms of grid-point values. In order to construct a finite difference approximation to the first derivative at point j, we have initially to derive expressions for φ_{j-1} and φ_{j+1} in terms of the behaviour of φ at point j. Using a Taylor expansion gives:

$$\varphi_{j+1} = \varphi(x_j + \Delta x) = \varphi_j + \varphi_j'\Delta x + \varphi_j''\left(\frac{\Delta x^2}{2!}\right) + \varphi'''_{j+\theta_1}\left(\frac{\Delta x^3}{3!}\right) \tag{5}$$

$$\varphi_{j-1} = \varphi(x_j – \Delta x) = \varphi_j – \varphi_j'\Delta x + \varphi_j''\left(\frac{\Delta x^2}{2!}\right) – \varphi'''_{j+\theta_2}\left(\frac{\Delta x^3}{3!}\right) \tag{6}$$

Solving (5) and (6) for φ_j' gives

$$\varphi_j' = \frac{\varphi_{j+1} - \varphi_j}{\Delta x} + E; \quad E = -\varphi_j''\left(\frac{\Delta x}{2!}\right) - \varphi'''_{j+\theta_1}\left(\frac{\Delta x^2}{3!}\right)$$

$$\varphi_j' = \frac{\varphi_j - \varphi_{j-1}}{\Delta x} + E; \quad E = \varphi_j''\left(\frac{\Delta x}{2!}\right) - \varphi'''_{j+\theta_2}\left(\frac{\Delta x^2}{3!}\right)$$

Alternatively, subtracting (6) from (5) leaves

$$\varphi_j' = \frac{\varphi_{j+1} - \varphi_{j-1}}{2\Delta x} + E; \quad E = \left(\frac{\Delta x^2}{3!2}\right)(\varphi'''_{j+\theta_1} + \varphi'''_{j+\theta_2}) \tag{7}$$

When E is omitted, these expressions give the forward, backward and centred finite difference approximations to the first derivative. The truncation error is given by E and the order of the approximation is defined by the lowest power of Δx in E. Therefore the forward and backward schemes are first order and the centred scheme is second order. The higher the order of the scheme, the greater is the accuracy of the finite difference approximation. All three schemes are consistent if the derivatives are bounded, because then the error approaches zero when Δx tends to zero.

A fourth order scheme can be derived by using (5) and (6) with expansions of $\varphi(x_j + 2\Delta x)$ and $\varphi(x_j - 2\Delta x)$. The result is:

$$\varphi_j' = \frac{4}{3}\left[\frac{\varphi_{j+1} - \varphi_{j-1}}{2\Delta x}\right] - \frac{1}{3}\left[\frac{\varphi_{j+2} - \varphi_{j-2}}{4\Delta x}\right] + \mathrm{O}(\Delta x^4) \tag{8}$$

The usual finite difference approximation to the second derivative, derived from (5) and (6), is

$$\varphi_j'' = \frac{\varphi_{j+1} - 2\varphi_j + \varphi_{j+1}}{\Delta x^2} + \mathrm{O}(\Delta x^2) \tag{9}$$

Finally it is worth introducing the notation that is often used for finite differences:

$$\delta_{mx}\varphi_j = \frac{\varphi_{j+m/2} - \varphi_{j-m/2}}{m\Delta x}$$

$$\overline{\varphi_j}^{mx} = \frac{\varphi_{j+m/2} + \varphi_{j-m/2}}{2}$$

Using this notation (7) and (9) become

$$\varphi_j' = \overline{\delta_x \varphi_j}^{x} = \delta_{2x}\varphi_j \quad \text{and} \quad \varphi_j'' = \delta_x^2 \varphi_j$$

2.2 The linear advection equation: Analytical solution

The one-dimensional linearised advection equation is

$$\frac{\partial\varphi}{\partial t} + u_0\frac{\partial\varphi}{\partial x} = 0; \quad \varphi = \varphi(x, t); \quad u_0 = \text{constant} \tag{10}$$

For convenience cyclic boundary conditions will be prescribed for φ at $x = 0$ and $x = L$.

$$\varphi(0, t) = \varphi(L, t)$$

The initial condition for φ is

$$\varphi(x, 0) = f(x) \quad 0 \le x \le L \quad \text{with} \quad f(x + L) = f(x)$$

In order to find an analytical solution for the linear advection equation we make use of the technique of separation of variables:

We look for a solution $\varphi(x, t)$ of the form

$$\varphi(x, t) = X(x)T(t)$$

substituting in the partial differential equation (10) we get

$$X(x)\frac{\mathrm{d}T}{\mathrm{d}t} + u_0 T(t)\frac{\mathrm{d}X}{\mathrm{d}x} = 0$$

dividing by $X(x)T(t)$ we get

$$\frac{1}{T}\frac{\mathrm{d}T}{\mathrm{d}t} = -u_0\frac{1}{X}\frac{\mathrm{d}X}{\mathrm{d}t}$$

the left-hand side is a function of t only while the right-hand side is a function of x only: therefore, they can be equal only if both of them are constant

$$\begin{aligned} \frac{1}{X}\frac{\mathrm{d}X}{\mathrm{d}x} &= \lambda & \qquad \frac{\mathrm{d}X}{\mathrm{d}x} &= \lambda X \\ \frac{1}{T}\frac{\mathrm{d}T}{\mathrm{d}t} &= -u_0\lambda & \qquad \frac{\mathrm{d}T}{\mathrm{d}t} &= -u_0\lambda T \end{aligned}$$

we have two "eigenvalue problems" for the operators $\mathrm{d}/\mathrm{d}x$ and $\mathrm{d}/\mathrm{d}t$, whose solutions are

$$\begin{aligned} X &= X_0 \exp[\lambda x] \\ T &= T_0 \exp[-u_0\lambda t] \end{aligned}$$

and the solution of the advection equation is

$$\varphi(x, t) = X_0 T_0 \exp[\lambda x - u_0\lambda t] = \varphi_0 \exp[\lambda \cdot (x - u_0 t)] = f(x - u_0 t) \qquad (11)$$

Therefore we get a function propagating without change of shape along the positive x axis with speed u_0 (phase speed).

If we have periodic boundary conditions, λ has only certain (imaginary) values, if they are sinusoidal with time, we must have $\lambda = \mathrm{i}k$ where k is the wave number and $ku_0 = \omega$ the frequency. Of course if φ is to represent a physical field, this field is the real part of the found solution.

As the advection equation is linear, any linear combination of solutions of the type found is also a solution of the equation. As all the component waves of a disturbance travel with the same speed there is no dispersion and the disturbance does not change shape with time.

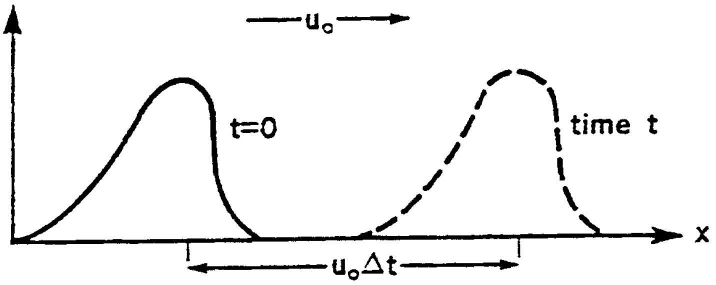


Figure 2. Representation of the solution of the analytical linear advection equation.

It is interesting to consider the "energy" defined by

$$E(t) = \frac{1}{2}\int_0^L \varphi^2 \mathrm{d}x \tag{12}$$

Multiplying the advection equation by φ and integrating with respect to x then gives

$$\frac{\partial E}{\partial t} = -\frac{u_0}{2}\int_0^L \frac{\partial \varphi^2}{\partial x} dx = -\frac{u_0}{2}[\varphi^2]_0^L = 0$$

Therefore the energy is conserved—as indeed it must be since there is no change in shape of the disturbance.

2.3 Space discretization: Dispersion and round-off error

Let us consider again the one-dimensional linear advection equation

$$\frac{\partial u}{\partial t} + C\frac{\partial u}{\partial x} = 0 \tag{13}$$

and represent the space derivative by means of centred finite differences

$$\frac{\partial u_j}{\partial t} = -C\frac{u_{j+1} - u_{j-1}}{2\Delta x} \tag{14}$$

To solve this space discretized equation we try a solution of similar form to the continuous equation, namely

$$u_j(t) = \Re e\{U(t)\exp(\mathrm{i}kj\Delta x)\} \tag{15}$$

Substituting in the discretized equation (14) we get

$$\frac{\mathrm{d}U}{\mathrm{d}t} + \mathrm{i}k\left(C\frac{\sin k\Delta x}{k\Delta x}\right)U = 0 \tag{16}$$

whose solution is

$$U = U_0 \exp(\mathrm{i}kC^* t) \tag{17}$$

and therefore the phase speed is

$$C^* = C\frac{\sin k\Delta x}{k\Delta x} = f(k) \tag{18}$$

A dispersion (phase speed dependent on wave number k) is introduced by the space discretization in the sense of decreasing the computed phase speed compared with the continuous solution.

The phase speed becomes 0 when $k\Delta x = \pi$ (wavelength $\lambda = 2\Delta x$)

The group velocity (at which energy is carried) is

$$C_\mathrm{g} = \frac{\mathrm{d}(kC)}{\mathrm{d}k} = C \text{ in the continuous equation}$$

$$C_g^* = \frac{\mathrm{d}(kC^*)}{\mathrm{d}k} = C\cos(k\Delta x) \text{ in the discretized equation}$$

which reaches a value of $-C$ (propagation in the wrong direction) for the shortest waves $k\Delta x = \pi$.

It is illuminating to see how accurately a finite difference approximation represents the derivative of a known function. Suppose $\varphi(x) = \Re e\{\exp(\mathrm{i}kx)\}$, where $k = 2\pi/L$ is the wavenumber and L is the wavelength. Substituting φ into (7) gives (dropping the $\Re e$ and ignoring E).

$$\begin{aligned}\varphi_j' &= \frac{\exp([\mathrm{i}k(x+\Delta x)] - \exp[\mathrm{i}k(x-\Delta x)])}{2\Delta x} = \frac{\exp[\mathrm{i}kx]}{2\Delta x}(\exp[\mathrm{i}k\Delta x] - \exp[-\mathrm{i}k\Delta x])\\ &= \mathrm{i}k\exp[\mathrm{i}kx]\left(\frac{\sin k\Delta x}{k\Delta x}\right)\end{aligned}$$

Therefore, the finite difference approximation is equal to the exact value multiplied by a correction factor F. If the wavelength consists of l grid lengths we have $k = 2\pi/l\Delta x$, and the correction factor becomes

$$F = \frac{\sin q}{q}; \quad q = \frac{2\pi}{l}$$

Similar calculation for the fourth order scheme shows that

$$f = \frac{4}{3}\frac{\sin q}{q} - \frac{1}{3}\frac{\sin 2q}{2q}$$

Plotting F against l for these schemes (see Fig. 4 later on) shows that about 10 grid lengths are required to describe accurately the behaviour of one wave and the shortest waves are badly mistreated. The plots also show that the fourth-order scheme is more accurate than the second-order scheme. This can be illustrated by examining the behaviour of F for the large wavelengths (l large, q small). Using series expansions we find that

$$\text{second order} \quad F = 1 - \frac{q^2}{6} = 1 + \mathrm{O}(q^2)$$

$$\text{fourth order} \quad F = 1 - \frac{q^4}{30} = 1 + \mathrm{O}(q^4)$$

Since the correct value of F is unity, this shows that second and fourth-order schemes have second and fourth-order errors. In general, if $F = 1 + \mathrm{O}(q^n)$ the scheme is said to be n th order.

2.4 Time discretization: Stability and computational mode

Finite differences can be used for time derivatives as well as space derivatives—that is we represent time derivatives in terms of values at discrete time levels. If Δt is the time interval (usually called the time step) then the time levels are given by $t_n = n\Delta t$ with $n = 0, 1, \dots$ Now the grid-point value of φ at position x_j at time t_n is denoted by φ_j^n.

Usually either forward or centred time differences are used:

$$(i) \qquad \text{forward} \left(\frac{\partial\varphi}{\partial t}\right)_j^n \to \frac{\varphi_j^{n+1} - \varphi_j^n}{\Delta t} + \mathrm{O}(\Delta t)$$

$$(ii) \qquad \text{centred} \left(\frac{\partial\varphi}{\partial t}\right)_j^n \to 2\frac{\varphi_j^{n+1} - \varphi_j^{n-1}}{2\Delta t} + \mathrm{O}(\Delta t^2)$$

Once again, centred differences are more accurate than forward time differences

In order to solve an initial value problem we must cast the PDE in finite difference form. The difference equation is then manipulated so as to give an algorithm which gives the grid-point value of φ at time level $(n+1)$ in terms of the values at earlier time levels.

As an example consider the advection equation with a forward time difference and backward (upstream) space difference

$$\frac{\partial\varphi}{\partial t} + u_0\frac{\partial\varphi}{\partial x} = 0 \to \frac{\varphi_j^{n+1} - \varphi_j^n}{\Delta t} + u_0\left(\frac{\varphi_j^n - \varphi_{j-1}^n}{\Delta x}\right) = 0; \qquad (u_0 > 0)$$

This scheme is described as being first order in time and space. Manipulation of the difference equation provides the following algorithm for solving the equation

$$\varphi_j^{n+1} = \varphi_j^n - \alpha(\varphi_j^n - \varphi_{j-1}^n); \qquad \alpha = \frac{u_0\Delta t}{\Delta x} \tag{19}$$

Knowing φ_j everywhere at time n allows us to calculate the new value at time $(n+1)$ grid point by grid point; this is an example of an explicit scheme.

Let’s try a solution of the form

$$\varphi_j^n = \varphi_0 \exp i(kj\Delta x - \omega n\Delta t)$$

Substituting in (19) we get

$$\exp(-i\omega t) = -\alpha i \sin(k\Delta x)$$

where $\omega = a + ib$ (complex number).

If b>0 φ_j^n is exponentially increasing with time (unstable)

if b<0 the solution is damped

if b=0 the solution is neutral (amplitude constant in time)

Also, as we have approximated the operator $\frac{\partial}{\partial t}$, we introduce yet another dispersion

Another scheme that arises is

$$\varphi_j^{n+1} + \frac{\alpha}{4}(\varphi_{j+1}^{n+1} - \varphi_{j-1}^{n+1}) = \varphi_j^n - \frac{\alpha}{4}(\varphi_{j+1}^n - \varphi_{j-1}^n)$$

Now we have a set of simultaneous equations which have to be solved for the φ^{n+1}; this is an example of an implicit scheme.

Both the above schemes are examples of two-time-level schemes. That is the finite difference equation only uses information from two time levels. Later we will come across examples of three-time-level schemes.

Just because we can produce an algorithm for solving an equation, it does not follow that its use will provide realistic solutions. For example, if we use a forward time difference and centred space difference in the advection equation we get

$$\varphi_j^{n+1} = \varphi_j^n - \frac{\alpha}{2}(\varphi_{j+1}^n - \varphi_{j-1}^n)$$

This is an explicit two-time-level scheme which is first order in time and second order in space. It appears to be a suitable algorithm for solving the equation. However it will be shown later that it has the property that the difference between its exact and numerical solution increases exponentially with time—the scheme is unstable.

The ratio α is called the C.F.L. number (after Courant, Freidrichs and Levy), or sometimes just the Courant number. We will see that it is of great significance when we consider the stability of numerical schemes.

In three-time-level schemes there is an extra complication. Let us consider the (leapfrog) scheme:

$$\varphi_j^{n+1} = \varphi_j^{n-1} - \alpha(\varphi_{j+1}^n - \varphi_{j-1}^n)$$

and try a solution $\varphi_j^n = \varphi_0 \lambda_k^n \exp(ikj\Delta x)$ where the superindex of λ means exponentiation. If the modulus of λ is greater than one, the solution is unstable. If it is smaller than one the solution is damped and if it is one the solution is neutral. Now substitute into the discretized equation and we get

$$\lambda^2 + 2ip\lambda - 1 = 0; \qquad \mathrm{p} \equiv -\alpha \sin(k\Delta x)$$

which has two solutions

$$\lambda = ip + \sqrt{1-p^2} \xrightarrow[\Delta x \to 0; \Delta t \to 0]{} 1 \qquad \text{physical mode}$$

$$\lambda = ip - \sqrt{1-p^2} \xrightarrow[\Delta x \to 0; \Delta t \to 0]{} -1 \qquad \text{computational mode}$$

It is now necessary to show that the scheme under consideration is convergent by making use of the Lax–Richtmeyer theorem. It can easily be shown that the above schemes are consistent, and so convergence is assured if they are stable. To do this we have to consider the behaviour of initial errors and examine if they grow exponentially. However, there are various ways in which this can be done. Here we will consider only three approaches.

(a) The energy method in which the scheme is considered unstable if the "energy" defined earlier increases with time.

(b) The von Neumann series method in which the behaviour of a single Fourier harmonic is studied; the stability of all admissible harmonics is a necessary condition for the stability of the scheme.

(c) The matrix method

2.5 Stability analysis of various schemes

2.5 (a) Methods of stability analysis .

(i) Energy method.

Earlier we found that, for the advection equation with periodic boundary conditions, the energy $E(t)$ was conserved. We now want to study an analogous quantity E^n given by

$$E^n = \frac{1}{2}\sum_{j=2}^{N+1}(\varphi_j{}^n)^2\Delta x$$

As an example of how to apply this method, we will study the stability of (19). The first step is to derive an expression for $(\varphi_j{}^{n+1})^2$. This is done by multiplying (19) by $(\varphi_j^{n+1}+\varphi_j^n)$ to give

$$(\varphi_j^{n+1})^2 - (\varphi_j^n)^2 = -\alpha(\varphi_j^{n+1}+\varphi_j^n)(\varphi_j^n - \varphi_{j-1}^n)$$

Substituting for φ_j^{n+1} in the RHS and rearranging

$$(\varphi_j^{n+1})^2 - (\varphi_j^n)^2 = -\alpha\{(\varphi_j^n)^2 - (\varphi_{j-1}^n)^2\} - \alpha(1-\alpha)(\varphi_j^n - \varphi_{j-1}^n)^2$$

Summing over all gridpoints and using the boundary condition $\varphi_1^n = \varphi_{N+1}^n$ leaves

$$E^{n+1} - E^n = -\alpha(1-\alpha)\sum_{j=2}^{N+1}(\varphi_j^n - \varphi_{j-1}^n)^2\Delta x$$

Therefore, in order to prevent the energy growing from step to step we require

(a) $\alpha \ge 0$ which implies $u_0 \ge 0$

(b) $(1-\alpha) \ge 0$ which implies $\alpha = \dfrac{u_0\Delta t}{\Delta x} \le 1$

This means that, having chosen the grid length Δx, we will only get a stable solution if the time step is chosen so that $\Delta t \le \Delta x / u_0$. But note that if we ensure stability by having $0 \le \alpha \le 1$, the energy is forced to decay from step to step.

The energy method is a quite general approach for analysing difference schemes and can be used for non-linear problems with complicated boundary conditions. However for most cases it requires considerable effort and ingenuity in order to derive practical stability criteria.

(ii) Fourier series method.

This was introduced by J. von Neumann and, by comparison with the energy method, it is simple to apply and provides considerable insight into the performance of different schemes.

Once again consider the original advection equation (10). If the initial condition is given by

$$\varphi(x, 0) = f(x) = C_k \exp[\mathrm{i}kx]; \quad k = \frac{2\pi}{L}m$$

where m is the number of waves, then we know that the true solution is

$$\varphi(x, t) = C_k \exp[\mathrm{i}k(x - u_0 t)] \tag{20}$$

Now consider the finite difference equation. The initial condition is

$$\varphi_j^0 = C_k \exp[\mathrm{i}kx_j]$$

and, in general, the solution is given by

$$\varphi_j^n = (\lambda_k)^n C_k \exp[\mathrm{i}kx_j] \tag{21}$$

where λ_k is a complex quantity which depends upon the finite difference scheme and the wavemunber k.

$$\text{If } \lambda_k = |\lambda_k| \exp[\mathrm{i}\theta] \text{ we have } \varphi_j^n = C_k |\lambda_k|^n \exp\left[\mathrm{i}k\left(x_j + \frac{n\theta}{k}\right)\right] \tag{22}$$

Therefore, $|\lambda_k|$ gives the fractional change in amplitude/timestep and θ provides information about the phase.

Comparing (22) with the analytic solution (20) shows the following.

(a) The stability of the finite difference scheme is assured if $|\lambda_k| \le 1$ for all k

(b) The numerical scheme has introduced a fictitious damping of $D = |\lambda_k|$ per time step; if $D = 1$ (no damping) the scheme is said to be neutral.

(c) The phase speed of the numerical solution is given by $c = -\theta / k\Delta t$; this is usually different from u_0 and so a phase error is introduced. A convenient measure of this is the relative phase speed $r = c/u_0$.

(d) Since the speed of the disturbance depends upon the wave number there is computational dispersion; this means that a disturbance made up of a variety of Fourier components will not keep its shape. In other words the group velocity $c_g = \partial(kc)/\partial k$ is not the same as the phase velocity.

For partial differential equations with constant coefficients, the stability criterion given in (a) is too stringent since a legitimate exponential growth of a physically realistic solution may be possible. Therefore the stability criterion should be

$$|\lambda_k| \le 1 + \mathrm{O}(\Delta t)$$

which allows an exponential, but not faster, growth of the solution. However, when we know that the true solution does not grow (as for the advection equation), it is customary to ensure that $|\lambda_k| \le 1$.

(iii) The matrix method.

Let $\mathbf{U}_n$ be a vector at time $n\Delta t$; if we can express $\mathbf{U}_{n+1}$ as

$$\mathbf{U}_{n+1} = \mathbf{A}\mathbf{U}_n \text{ (scheme of two time levels)}$$

where $\mathbf{A}$ is called the amplification matrix, the method runs as follows:

Let $\mathbf{V}_k$ be the eigenvectors of $\mathbf{A}$ corresponding to the eigenvalues λ_k

$$\mathbf{A}\mathbf{V}_k = \lambda_k \mathbf{V}_k$$

We project vectors $\mathbf{U}$ onto the space defined by these eigenvectors

$$\mathbf{U}_0 = \sum_k U_0^k \mathbf{V}_k$$

Therefore we obtain, by repeated multiplication by the amplification matrix

$$\mathbf{U}_n = \sum_k \mathbf{U}_0^k \lambda_k^{(n)} \mathbf{V}_k$$

where superindex (n) stands for the exponential operation.

This solution will be bounded when $n \to \infty$ for all $|\lambda_k| \le 1$ and, in this case, the scheme is stable.

This method is equivalent to the von Newman method when the Fourier basis functions are eigenvalues of the amplification matrix.

2.5 (b) Forward time schemes.

(i) Forward time differencing with non-centred space differencing.

This is the scheme introduced in Subsection 2.4 and may conveniently be written as

$$\varphi_j^{n+1} = \varphi_j^n - \alpha(\varphi_j^n - \varphi_{j-1}^n); \quad \alpha = \frac{u_0 \Delta t}{\Delta x}$$

This is an explicit two-time-level scheme which is first order in space and time. It is called upwind scheme if $u_0 > 0$ and downwind if $u_0 < 0$.

Substituting $\varphi_j^n = (\lambda_k)^n C \exp[\mathrm{i}kx_j]$ into the above algorithm yields

$$\begin{aligned}\lambda_k &= 1-\alpha\{1-\exp[-\mathrm{i}k\Delta x]\}\\ &=1-\alpha(1-\cos(k\Delta x+\mathrm{i}\sin k\Delta x))\end{aligned}$$

Since λ_k is complex we can express it as

$$\lambda_k = |\lambda_k|(\cos\theta+\mathrm{i}\sin\theta)$$

Substituting this expression in the above, and equating real and imaginary parts gives two equations for $|\lambda_k|$ and θ in terms of α and $k\Delta x$

$$\begin{aligned}|\lambda_k|\cos\theta &= 1-\alpha(1-\cos k\Delta x)\\ |\lambda_k|\sin\theta &= -\alpha\sin k\Delta x\end{aligned}$$

To study the stability we require an expression for $|\lambda_k|$. Squaring and adding then gives

$$|\lambda_k|^2 = 1-2\kappa(\alpha-1)(1-\cos k\Delta x)$$

Since $1-\cos k\Delta x \geq 0$ we can only satisfy the stability criterion $|\lambda_k|\leq 1$ if $\alpha((\alpha-1)\leq 0)$; therefore, we require $u_0\geq 0$ (upwind) and $u_0\Delta t/\Delta x\leq 1$ (CFL limit) (the same result as when the energy method was used). The scheme is said to be conditionally stable.

To study the damping and phase errors, it is often convenient to think in terms of wavelengths consisting of l grid lengths; we then replace k by $2\pi\Delta x/l$. It can then be shown that the damping per time step (D) and relative phase error (r) can be expressed as

$$D = [1+2\alpha(\alpha-1)(1-\cos q)]^{\frac{1}{2}}; \quad q = \frac{2\pi}{l} \tag{23}$$

$$r = -\frac{1}{\alpha q}\mathrm{atan}\left\{\frac{-\alpha\sin q}{1-\alpha(1-\cos q)}\right\} \tag{24}$$

The characteristics of a scheme can be conveniently displayed by plotting graphs of D and r against Il for various choices of α. However, to make comparisons between schemes easier, we will only consider values of D and r for $l = 2, 3, 4, 6$ and 10 with $\alpha = 0.5$. These are shown in Table 2. Clearly the upstream differencing scheme reproduces the phase speed very well (though there are phase errors when $\alpha\neq 1$: $r<1$ when $0<\alpha<1/2$ and $r>1$ when $1/2<\alpha<1$), but the damping is excessive.

(ii) Forward time differencing with centred space differencing (FTCS).

$$\frac{\varphi_j^{n+1}-\varphi_j^n}{\Delta t}+u_0\left(\frac{\varphi_{j+1}^n-\varphi_{j-1}^n}{2\Delta x}\right) = 0$$

Using the Fourier series method it is easy to show that

$$|\lambda_k|^2 = 1+\alpha^2(\sin k\Delta x)^2$$

Therefore, $|\lambda_k|^2\geq 1$ always and so the scheme is unstable for all values of α and k; the scheme is then said to be absolutely unstable, although the space discretization is more accurate than in the upwind scheme, which is conditionally stable.

(iii) Implicit Schemes.

Consider what happens when the space derivative is replaced by the average value of the centred space difference at time levels n and $n+1$. Using forward time differencing and the notation for spatial differences introduced in Subsection 1.4, we have

$$\frac{\varphi_j^{n+1}-\varphi_j^n}{\Delta t}+\frac{u_0}{2}(\delta_{2x}\varphi_j^n+\delta_{2x}\varphi_j^{n+1}) = 0$$

Rearranging yields

$$\varphi_j^{n+1}+\frac{\alpha}{4}(\varphi_{j+1}^{n+1}-\varphi_{j-1}^{n+1}) = \varphi_j^n-\frac{\alpha}{4}(\varphi_{j+1}^n-\varphi_{j-1}^n) \tag{25}$$

This is an implicit two-time-level scheme (the Crank–Nicolson scheme) which is second order in time and second order in space. Performing a stability analyses in the usual way we find that $|\lambda_k| = 1$. Therefore the scheme is absolutely stable and neutral (no damping), but further analysis shows that there are significant phase errors (see Table 2).

Note that the problem with using this type of scheme is that we cannot simply express the new value φ_j^{n+1} in terms of known values at previous times. Thus, we have a large number of simultaneous equations which have to be solved (i.e. a tridiagonal matrix has to be inverted). For simple cases this can be done exactly, but for more complicated problems expensive successive approximation methods have to be used.

This implicit approach can be generalised to

$$\frac{\varphi_j^{n+1}-\varphi_j^n}{\Delta t}+u_0(\beta_n\delta_{2x}\varphi_j^n+\beta_{n+1}\delta_{2x}\varphi_j^{n+1}) = 0$$

where β_n and β_{n+1} are weights such that $\beta_n+\beta_{n+1} = 1$.

There are three special cases which should be highlighted:

(a) $\beta_n = 1$ and $\beta_{n+1} = 0$ gives the absolutely unstable FTCS scheme.

(b) $\beta_n = 0$ and $\beta_{n+1} = 1$ results in the fully forward implicit scheme.

(c) $\beta_n = \beta_{n+1} = 1/2$ yields the scheme described above in which the derivatives at time levels n and $n+1$ are equally weighted.

A stability analysis of the general scheme shows that

$$|\lambda|^2 = \frac{1+\alpha^2\beta_n^2(\sin k\Delta x)^2}{1+\alpha^2\beta_{n+1}^2(\sin k\Delta x)^2}$$

Therefore there is absolute instability if the present values are weighted more heavily than the future ones $(\beta_n > 1/2,\ \beta_{n+1} < 1/2)$ whereas there is absolute stability if more or equal weight is given to the future values $(\beta_n \le 1/2,\ (\beta_{n+1} \ge 1/2))$

2.5 (c) The leapfrog scheme.

This is probably the most common scheme used for meteorological problems. The "leapfrog" refers to the centred time difference which is used in conjunction with centred space differences

$$\varphi_j^{n+1} = \varphi_j^{n-1} - \alpha(\varphi_{j+1}^n - \varphi_{j-1}^n) \tag{26}$$

This is an explicit three-time-level scheme which is second order in space and time. Using the Fourier series technique to test stability, we find that

$$\lambda^2 + 2\mathrm{i}p\lambda - 1 = 0; \quad p = -\alpha \sin k\Delta x$$

giving

$$\lambda = \mathrm{i}p \pm \sqrt{1-p^2}$$

Therefore there are two solutions for λ which is a consequence of using a three-time-level scheme (in general an m -time-level scheme will have $m-1$ solutions for λ with each solution being referred to as a mode).

It can be shown that for one of the modes $\lambda \to 1$ as $\Delta x, \Delta t \to 0$; this is referred to as the physical mode. The other mode has no physical significance and is called the computational mode (for this mode $\lambda \to -1$ as $\Delta x, \Delta t \to 0$). If $|\alpha| \le 1$ we have $|p| \le 1$ and so $\sqrt{1-p^2}$ is real. Consequently $|\lambda| = 1$ for both modes and so the scheme is conditionally stable and neutral. Further analysis shows that for the physical mode

$$r = -\frac{1}{\alpha q}\operatorname{atan}\left(\frac{-p}{\sqrt{1-p^2}}\right); \qquad \text{p=-}\alpha\sin q; \qquad \text{q=2}\frac{\pi}{l}, \tag{27}$$

whereas for the computational mode the phase speed is in the opposite direction to u_0 ($r = -1$) and the amplitude of the mode changes sign every time step. In general, the solution to the finite difference equation will be a combination of the physical and computational modes.

The tables of D and r against l (Table 2) for the physical mode reveal that the phase errors are worse than for the upstream difference scheme, but the leapfrog scheme has the important property that there is no damping for any choice of α.

The characteristics of the leapfrog scheme can be improved by using a fourth order finite difference scheme for the space derivative (see Subsection 2.1)—the scheme is then said to have fourth-order advection. Table 2 shows that this has no effect on the damping (the scheme remains neutral), but it does lead to an improvement in the phase speed. However the stability condition is now more restrictive since we require $\alpha \le 0.73$.

The leapfrog scheme is very popular because it is simple, second order and neutral; however there are still phase errors and computational dispersion. Also, the computational mode has to be contended with and the dependent variable has to be kept at two time levels.

To start the leap-frog scheme it is customary to use a forward time step and, in order to suppress separation of the solutions at odd and even time steps, it is usual to either

(i) use an occasional forward time step

(ii) use a weak time filter of the type

$$\tilde{\varphi}_j^{n-1} = \varphi_j^{n-1} + a(\tilde{\varphi}_j^{n-2} - 2\varphi_j^{n-1} + \varphi_j^n)$$

where the tilde denotes the filtered value (a is typically 0.005).

Another variant of the leapfrog scheme is the semi-momentum approximation. For this, the wind field is smoothed before multiplying by the derivative. Using the notation introduced in Subsection 2.1, the scheme becomes

$$\overline{\delta_t\varphi}^t = -\overline{\bar{u}^x\delta_x\varphi}^x$$

For constant u , this reduces to (26).

2.5 (d) The Lax–Wendroff scheme.

This is a useful scheme because it is second order in space and time. but (unlike the leapfrog scheme) it is only a two-time-level scheme and so has no computational mode.

The Lax–Wendroff scheme cannot be constructed by an independent choice of finite difference approximations for the space and time derivatives. It is derived from a second-order accurate Taylor series expansion

$$\varphi(x, t + \Delta t) = \varphi(x, t) + \Delta t\frac{\partial\varphi}{\partial t} + \frac{\Delta t^2}{2}\frac{\partial^2\varphi}{\partial t^2}$$

Using the advection equation this becomes

$$\varphi(x, t + \Delta t) = \varphi(x, t) + -u_0\Delta t\frac{\partial\varphi}{\partial x} + \frac{u_0^2\Delta t^2}{2}\frac{\partial^2\varphi}{\partial x^2} \tag{28}$$

Replacing the derivatives by second order accurate finite difference approximation gives

$$\varphi_j^{n+1} = \varphi_j^n - \frac{\alpha}{2}(\varphi_{j+1}^n - \varphi_{j-1}^n) + \frac{\alpha^2}{2}(\varphi_{j+1}^n - 2\varphi_j^n + \varphi_{j-1}^n) \tag{29}$$

This scheme can be replaced by one in which there are two steps:

(i) provisional values of $\varphi_{j+1/2}^{n+1/2}$ are calculated using a forward time step with centred space differencing

$$\varphi_{j+1/2}^{n+1/2} = \frac{1}{2}(\varphi_{j+1}^n + \varphi_j^n) - \frac{\alpha}{2}(\varphi_{j+1}^n - \varphi_j^n)$$

(ii) The φ_j^{n+1} are calculated from centred space and time differences using the provisional values $\varphi_{j+1/2}^{n+1/2}$

$$\varphi_j^{n+1} = \varphi_j^n - \alpha(\varphi_{j+1/2}^{n+1/2} - \varphi_{j-1/2}^{n+1/2})$$

The stability analysis shows that

$$D = [1-\alpha^2(1-\alpha^2)(1-\cos q)^2]^{\frac{1}{2}}; \quad q = \frac{2\pi}{l} \tag{30}$$

and so the scheme is stable provided $\alpha \le 1$. The ratio of the phase speed to the advection velocity is given by:

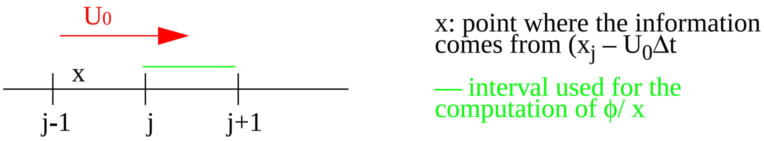
$$r - -\frac{1}{\alpha q}\mathrm{atan}\left\{\frac{-\alpha \sin q}{1-\alpha^2(1-\cos q)}\right\} \tag{31}$$

Table 3 shows that the characteristics of the Lax–Wendroff scheme fall between those of the upstream differencing and leapfrog schemes. The characteristics of the scheme can be improved by using fourth order advection.

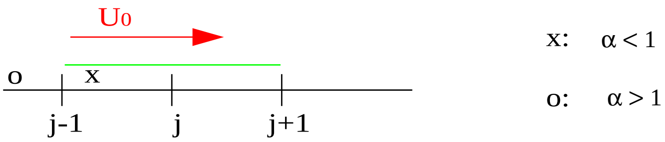
2.5 (e) Intuitive look at stability.

If the information for the future time step “comes from” inside the interval used for the computation of the space derivadive, the scheme is stable. Otherwise it is unstable. The CFL number α is the fraction of Δx travelled by an air parcel during Δt seconds.

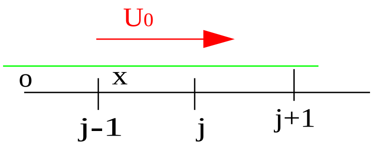
-Downwind scheme (unstable):


U0
x
j-1
j
j+1
x: point where the information comes from ($x_j - U_0\Delta t$
— interval used for the computation of ϕ/ x

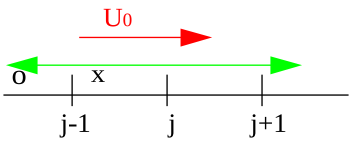
Upwind scheme (conditionally stable):



Leapfrog (conditionally stable)



Implicit (unconditionally stable). The interval covers the whole x-axis because we have to solve a coupled system of equations including all the points:



2.6 Group velocity

For a non-dispersive equation, plane wave solutions have the form $\exp[\mathrm{i}k(x-ct)]$, where the phase velocity c is independent of the wave number k. However, if there is dispersion the wave solutions have the same form, but now $c = c(k)$. Even if the original equation is non-dispersive, a discrete model will introduce dispersion.

In order to understand the effect of dispersion it is necessary to introduce the group velocity c_g given by

$$c_\mathrm{g} - \frac{\partial}{\partial k}(kc)$$

This represents the speed of propagation of the energy of wave number k and when there is dispersion we have $c = c(k)$ and $c_\mathrm{g} = c_\mathrm{g}(k)$. For a non dispersive medium $c_\mathrm{g} = c$.

For the linear advection equation we know that any disturbance should move without change of shape with the advecting velocity u_0 (which is independent of k). However, when the problem is solved numerically we find that $c = c(k)$ and dispersion is introduced. For example, the phase velocity from the leapfrog scheme is such that

$$r = \frac{c}{u_0} = \frac{1}{\alpha q}\mathrm{atan}\left\{\frac{-\alpha\sin q}{1-\alpha(1-\cos q)}\right\}; \quad q = \frac{2\pi}{l}$$

However the group velocity gives

$$r_\mathrm{g} = \frac{c_\mathrm{g}}{u_0} = \frac{\cos q}{[1-(\alpha\sin q)^2]^{1/2}}$$

Therefore, when $\alpha = 0.5$ we get the following (Table 2).

TABLE 2. RATIO OF THE RELATIVE PHASE ERROR (r) AND THE RELATIVE GROUP VELOCITY ERROR (r_g) FOR DIFFERENT WAVELENGTHS.

l	2	3	4	6	10
r	0.00	0.43	0.67	0.86	0.92
r_g	-1.00	-0.55	0.00	0.59	0.85

Note that the two gridlength waves ($l = 2$) travel in the wrong direction with speed u_0, whilst the longer waves move with a speed approaching the advecting velocity.

To illustrate the effect of computational dispersion consider three cases taken from Vichenevetsky and Bowles (1982). Each integration was carried out with the leapfrog scheme using $\alpha = 0.2$ (hence any effects are mainly due to the space discretization).

For the case shown in Fig. 3 (a), the long-wave components move with a group velocity of about u_0 ($r_g \approx 0$ for the long waves) whilst the two-gridlength waves travel upstream with speed $c_g(l = 2) = -u_0$; the four gridlength waves are stationary since $c_g(l = 4) = 0$. Therefore, during the integration the computational dispersion has caused a broadening of the disturbance (this is not caused by dissipation because the leapfrog scheme is neutral) and has generated parasitic short gridlength waves which travel upstream.

The disturbance shown in Fig. 3 (b) is dominated by waves with $l = 2$. Therefore the dominant feature of the integration is the upstream movement of the wave packets with a group velocity of about $-u_0$,

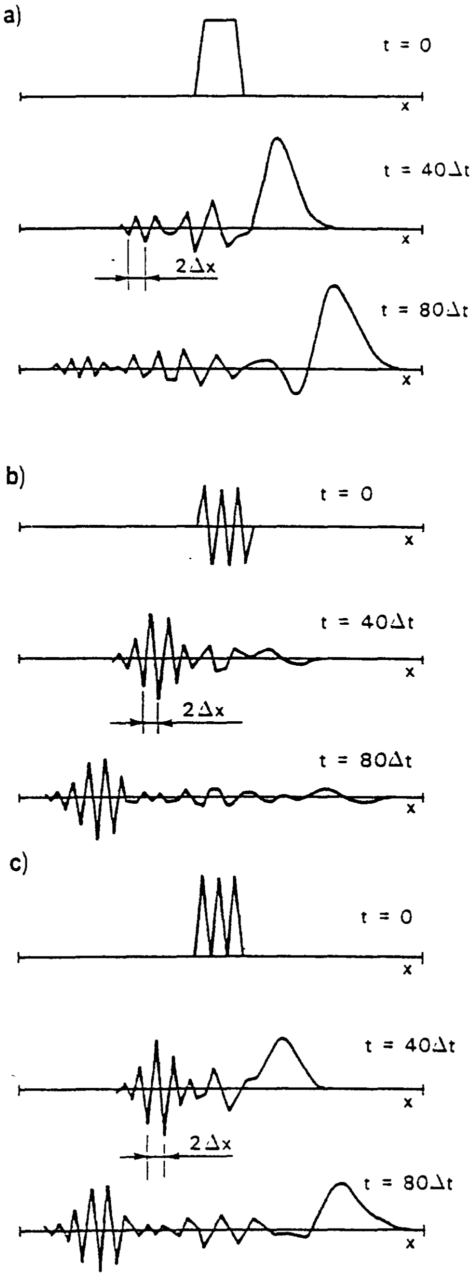


Figure 3. llustration of computational dispersion using the leapfrog scheme with $\alpha = 0$. Taken from Vichenevetsky and Bowles (1982).

In the last case, Fig. 3 (c), the initial disturbance consists of two-gridlength waves superimposed upon a broaderscale feature. Consequently, as in case (a), the two-gridlength waves move upstream, whilst the part of the initial disturbance composed of the larger wavelengths moves downstream with a group velocity of about u_0.

Numerical schemes should be examined for their computational dispersion. However, in practice the effects of computational dispersion are obscured because of the dissipation inherent in many numerical schemes or the explicit diffusion that is introduced to control the two-gridlength waves.

2.7 Choosing a scheme

There is a great variety of finite difference schemes and so it is worth considering what factors should be taken into account when choosing one.

(a) It is desirable to have high-order truncation errors for the space and time differences. In general centred differences are more accurate than one-sided differences.

(b) Ideally we would like the phase errors and damping to be small; however, it is usually necessary to compromise between these two. Plots of D and r against l are a convenient way of examining these aspects.

(c) The advantage of an explicit scheme is that it is easy to program, but it will only be conditionally stable and so the choice of time step is limited. Implicit schemes are absolutely stable; however the price we pay for this is that at every time step a system of simultaneous equations has to be solved.

(d) If the scheme has more than two time levels there will be computational modes and possibly separation of the solution at odd and even timesteps. Also more fields of the dependent variable have to be stored than for the a two-time-level scheme.

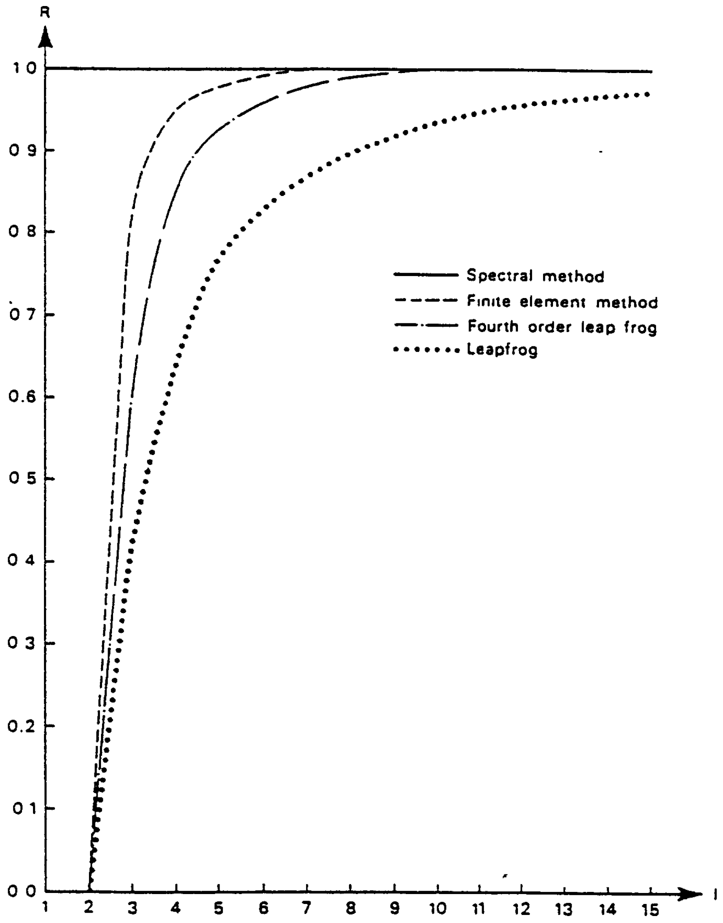


Figure 4. Response function (R) against number of gridlengths per wavelength (l) for various semi-discrete versions of the one-dimensional linear advection equation. Note that for the leapfrog scheme R is the same as the correction factor F introduced in Subsection 2.3.

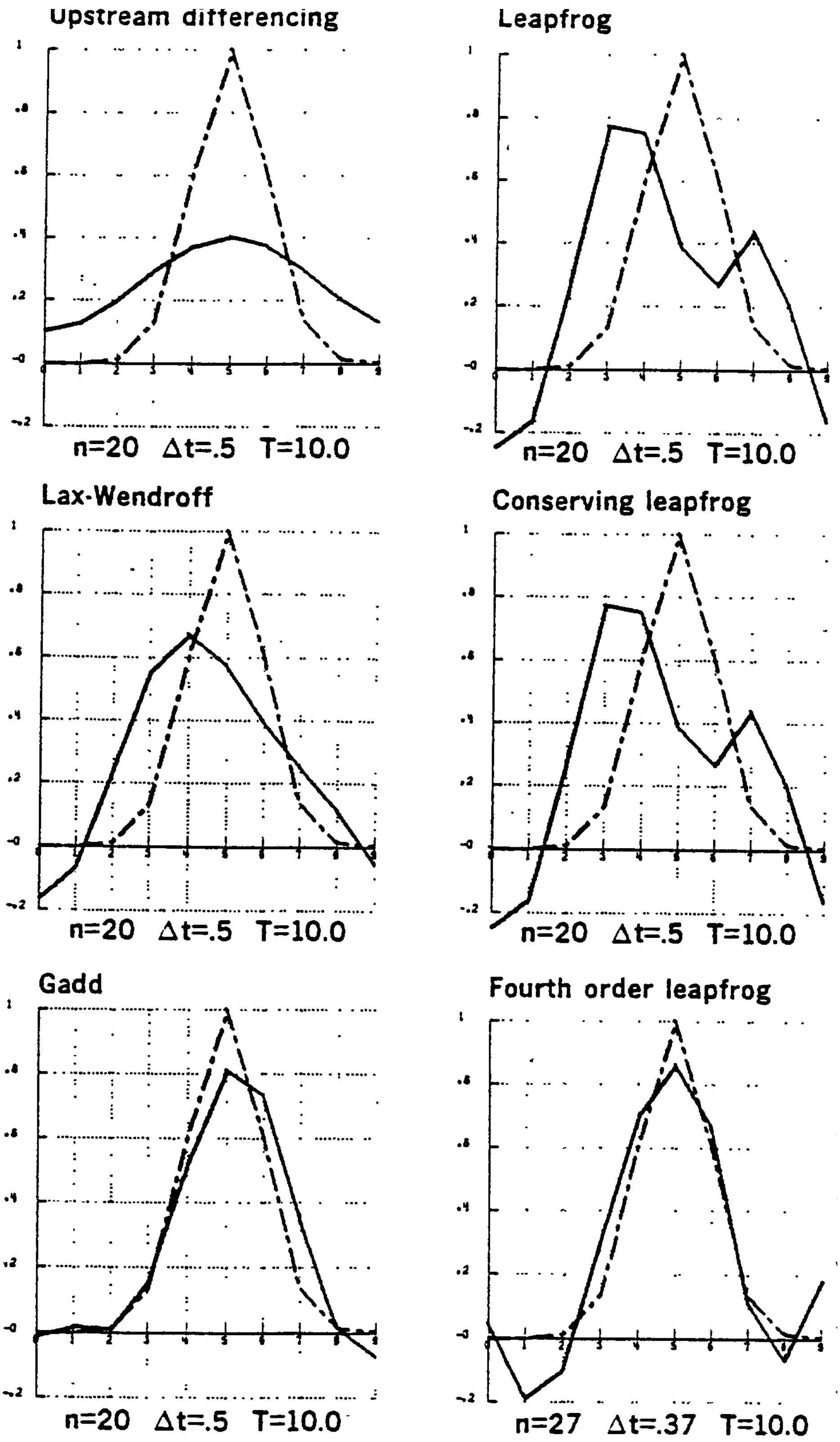


Figure 5. Solutions of the linear advection equation using various numerical methods for a Gaussian initial disturbance and a uniform wind. Full line:- numerical solution; dot-dashed line: exact solution.

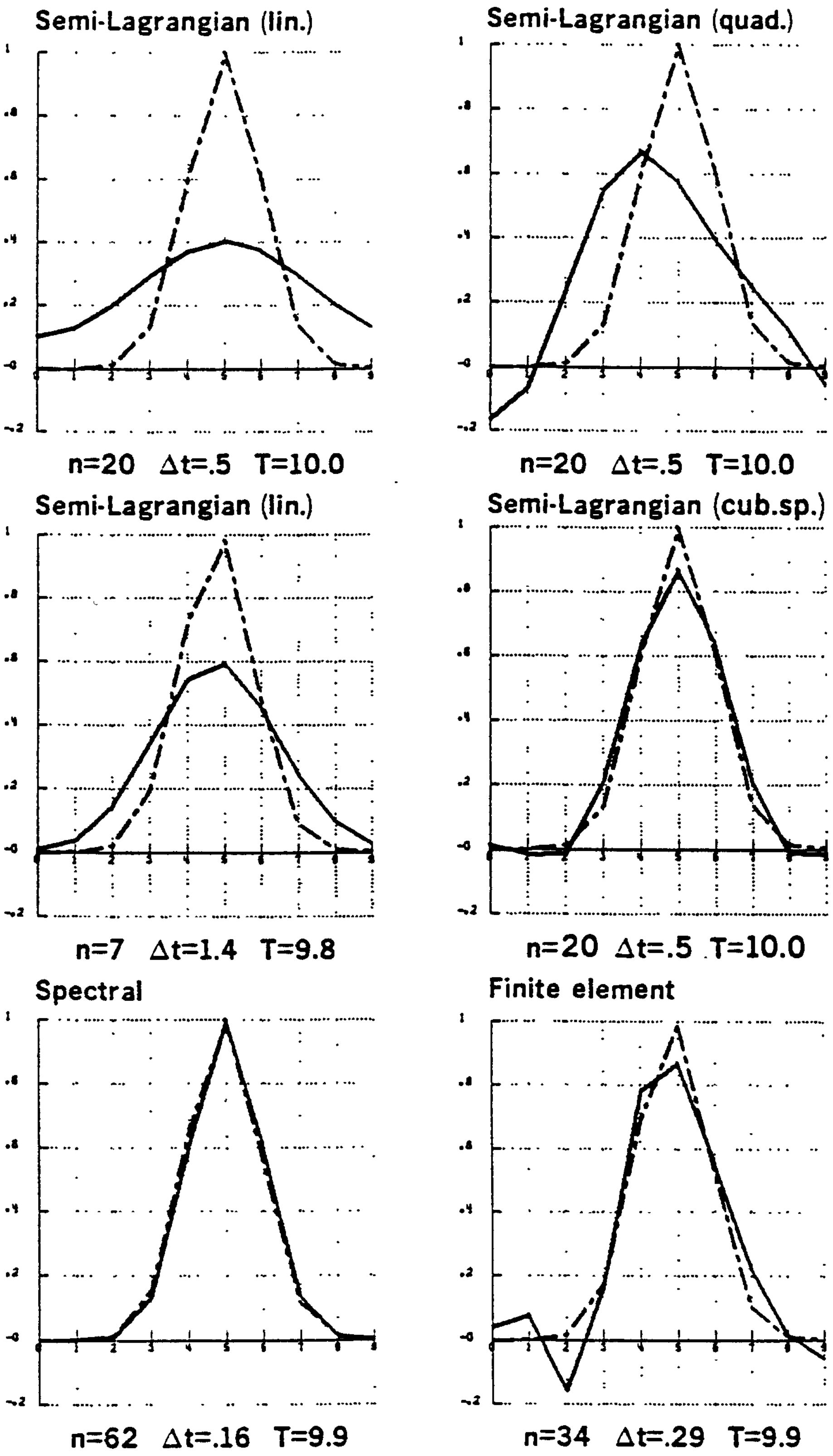
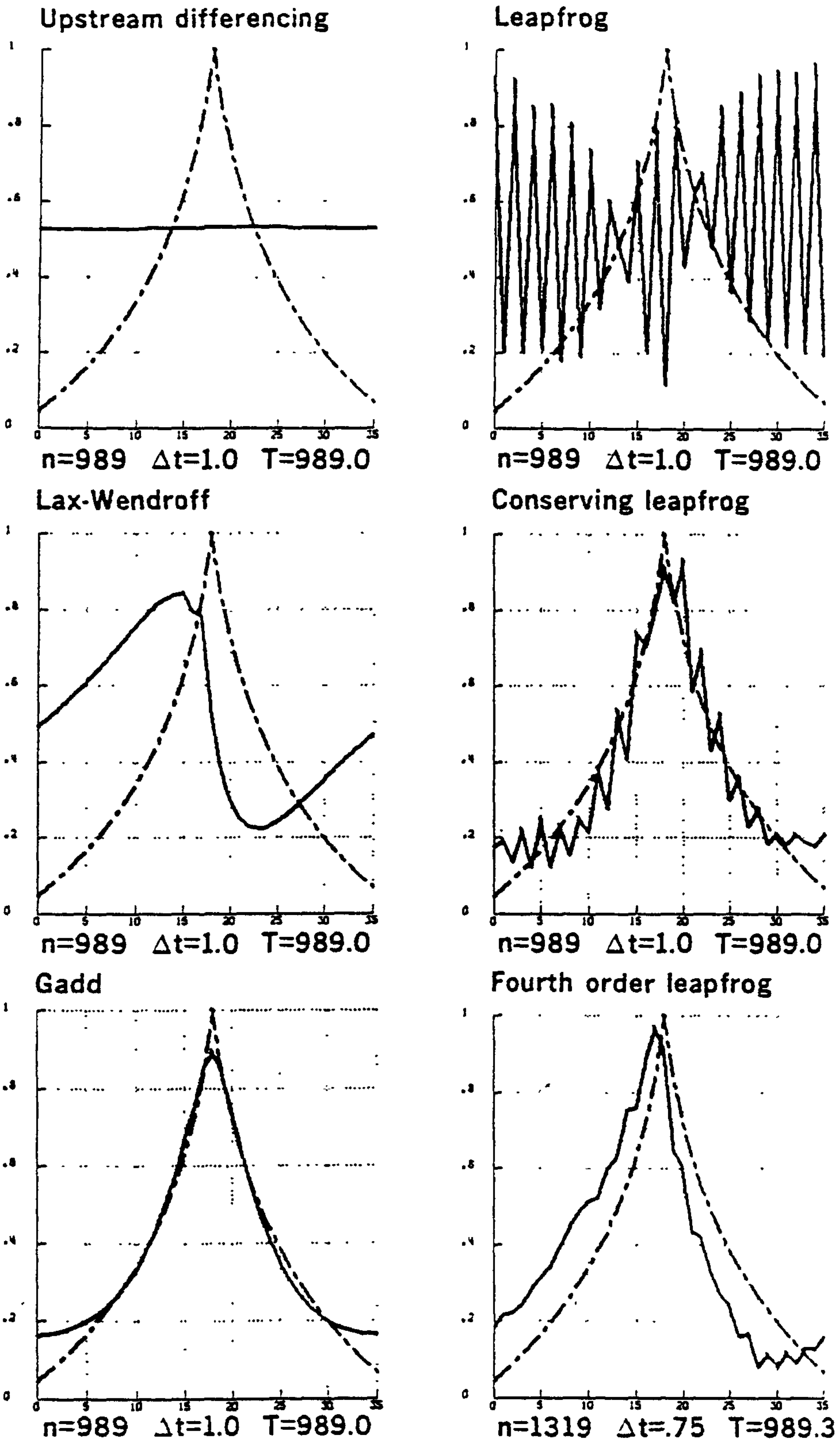


Figure 0. Continued



Upstream differencing

Leapfrog

Lax-Wendroff

Conserving leapfrog

Gadd

Fourth order leapfrog

Figure 6. Solutions of the linear advection equation using various numerical methods for the Crowley test . Full line:- numerical solution; dot-dashed line: exact solution..

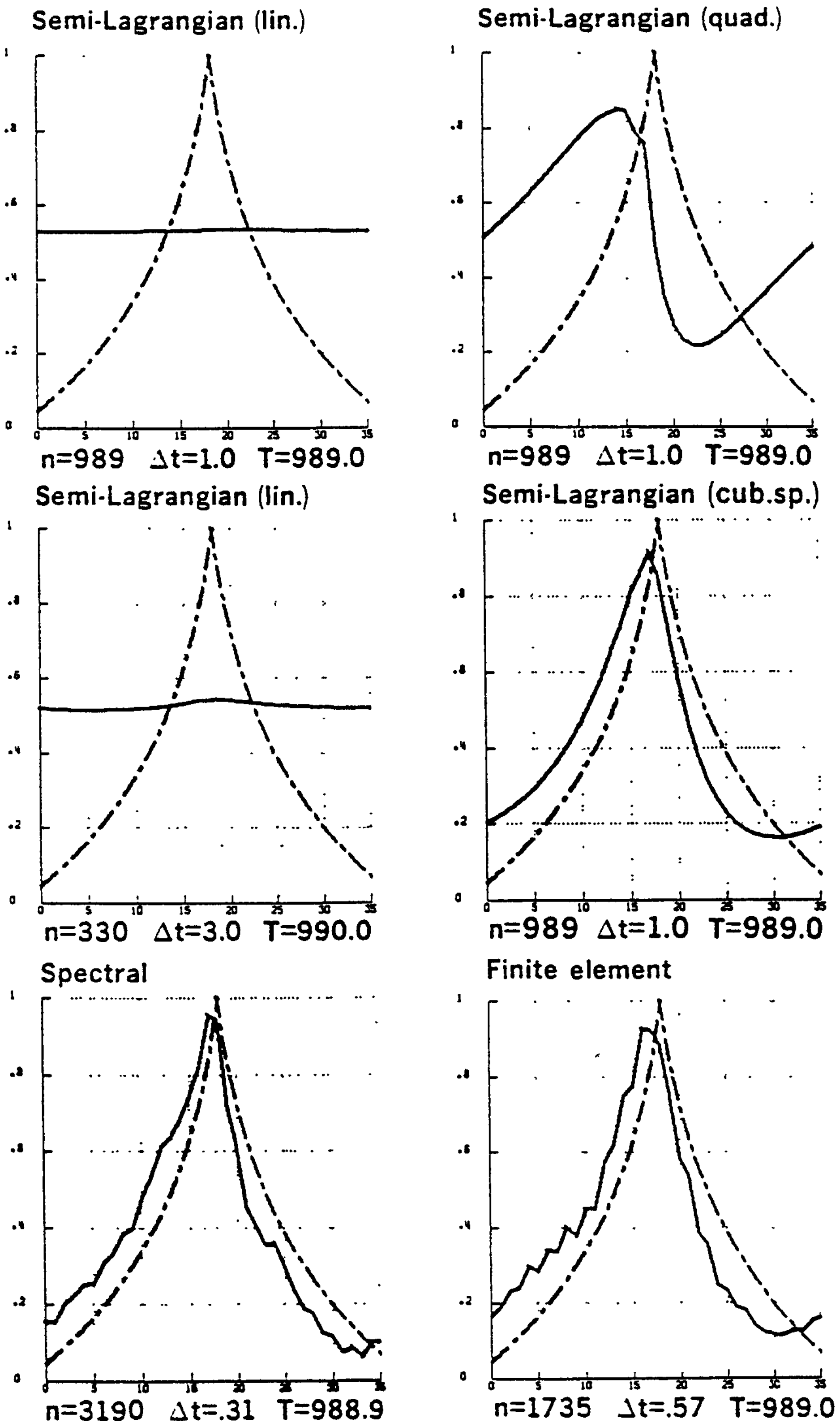


Figure 0. Continued.

A convenient way of comparing schemes is to consider their behaviour for the longer waves (l large so k and

$q = 2\pi/l$ small). For each scheme we can derive expressions for D and r in terms of l; these can then be expanded as power series under the assumption that q is small. If $D = 1 + O(q^n)$ the scheme is said to have n th order dissipation, whereas $r = 1 + O(q^n)$ indicates that there are n th order phase errors. The higher the order of accuracy of the amplitude and phase speed the better.

Sometimes it is interesting to examine just the effect of space discretization. Using a single Fourier component, the semi-discrete finite difference version of the linear advection equation may be expressed as:

$$\frac{\partial\varphi}{\partial t} = -\mathrm{i}ku_0R\varphi$$

where R is the response function. For the original PDE, $R = 1$ for all k and, ideally, our difference scheme should reproduce this. Fig. 4 shows R for second and fourth order space differencing as a function of l (this is the same as the correction factor F described in Subsection 2.3); also shown are the values for the spectral and finite element methods which are discussed later. These results suggest that the standard finite difference approximations for the advection are inferior to the spectral and finite element representations.

As well as examining the behaviour of schemes theoretically, it is often illuminating to actually solve the equation numerically using the various techniques. For example Gadd (1978) considered the behaviour of a Gaussian profile whilst Carpenter (1981) used a step function. Collins (1983) preferred a severe test first introduced by Crowley (1968). For this the advecting velocity u varies with x ($u = ax + b$ with a and b constant). It is then easy to show that if $\varphi(x, 0) = f[\ln(u)]$ then the analytical solution to the advection equation is $\varphi(x, t) = f[\ln(u - at)]$ The particular functions chosen by Collins are:

$$u = 0.9 - 1.6\frac{x}{L} \qquad 0 \le x \le \frac{L}{2} \qquad u = -0.7 + 1.6\frac{x}{L} \qquad \frac{L}{2} \le x \le L$$

along with $\varphi(x, 0) = \ln(u)$. It can be shown that the fluid particles will all repeat their relative positions after a time

$$T = \frac{2L}{1.6}\ln\left[\frac{u(0)}{u(L/2)}\right]$$

Fig. 5 shows the results of using various finite difference schemes to advect a Gaussian shaped disturbance with a constant wind (also shown are the results of using the semi-Langrangian, spectral and finite-element techniques discussed later). For these calculations we have used

$$N = \frac{1}{10} \qquad \mathrm{u}_0 = 1.0 \qquad \Delta x = 1.0 \qquad \Delta t = \frac{1}{2}\ \text{(maximum possible time step)}$$

and the integration has continued until the disturbance crosses the domain once. In Fig. 6 are the corresponding results for the Crowley test in which the initial disturbance was normalised so that it has a maximum value of unity. Examination of these results gives a clear indication of the characteristics of each of the schemes.

No matter what methods are used to select a finite difference scheme, there will inevitably be an element of compromise—the perfect scheme does not exist.

2.8 The two-dimensional advection equation .

Before leaving the advection equation it is worth considering the two-dimensional version

$$\frac{\partial\varphi}{\partial t}+u_0\frac{\partial\varphi}{\partial x}+v_0\frac{\partial\varphi}{\partial y} = 0$$

If this is put in finite difference form, the stability of the resulting difference equation can be examined by using

$$\varphi^n = \varphi_0\lambda^n\exp[i(kx+ly)]$$

For conditionally stable schemes, the stability criterion usually has the form

$$S = \Delta t\left|\frac{u_0\sin\alpha}{\Delta x}+\frac{v_0\sin\beta}{\Delta y}\right|\le 1$$

where $\alpha = k\Delta x$ and $\beta = l\Delta y$.

Let $u_0 = R\cos\theta$ and $v_0 = R\sin\theta$, and $R = (u_0^2+v_0^2)^{1/2}$. We then have

$$S(\alpha, \beta, \theta) = R\Delta t\left|\frac{\cos\theta\ \sin\alpha}{\Delta x}+\frac{\sin\theta\ \sin\beta}{\Delta y}\right|$$

If we maximise S with respect to α , β and θ we get

$$\sin\alpha = \sin\beta = 1 \quad\text{and}\quad \tan\theta = \frac{\Delta x\sin\beta}{\Delta y\sin\alpha}$$

Substituting for these in S gives the stability criterion

$$\Delta tR\left(\frac{1}{\Delta x^2}+\frac{1}{\Delta y^2}\right)^{1/2}\le 1$$

If $\Delta x = \Delta y = \Delta s$ this becomes $\Delta t\le\dfrac{\Delta s}{R\sqrt{2}}$

The appearance of the $\sqrt{2}$ is typical when going from one to two-dimensional problems. It means that the stability criterion is more restrictive than in the one-dimensional cases.

This problem can be overcome by the splitting technique discussed in Subsection 4.5 (a).

TABLE 3. DAMPING/TIME STEP (D) AND RELATIVE PHASE ERROR (r) FOR VARIOUS SCHEMES TO SOLVE THE ONE DIMENSIONAL LINEAR ADVECTION EQUATION WHEN $\alpha = 1/2\times$ THE C.F.L. STABILITY CRITERION ($\alpha = 1/2$ FOR ABSOLUTELY STABLE SCHEMES).

(a) Damping/time step(D)					
	$2\Delta x$	$3\Delta x$	$4\Delta x$	$6\Delta x$	$10\Delta x$
Upstream differencing	0.00	0.50	0.71	0.87	0.95
Crank–Nicholson	1.00	1.00	1.00	1.00	1.00
Lax–Wendroff	0.50	0.76	0.90	0.98	1.00
Gadd	0.13	0.79	0.95	0.99	1.00
Leapfrog	1.00	1.00	1.00	1.00	1.00
4th-order leapfrog	1.00	1.00	1.00	1.00	1.00

TABLE 3. DAMPING/TIME STEP (D) AND RELATIVE PHASE ERROR (r) FOR VARIOUS SCHEMES TO SOLVE THE ONE DIMENSIONAL LINEAR ADVECTION EQUATION WHEN $\alpha = 1/2\times$ THE C.F.L. STABILITY CRITERION ($\alpha = 1/2$ FOR ABSOLUTELY STABLE SCHEMES).

(a) Damping/time step(D)					
	$2\Delta x$	$3\Delta x$	$4\Delta x$	$6\Delta x$	$10\Delta x$
Spectral	1.00	1.00	1.00	1.00	1.00
Finite element	1.00	1.00	1.00	1.00	1.00
Semi-Lagrangian (p=0, linear interpolation)	0.00	0.50	0.71	0.87	0.95
Semi-Lagrangian (p=0, cubic spline)	0.00	0.88	0.97	1.00	1.00

TABLE 2. CONTINUED

(b) Relative phase error (r)					
	$2\Delta x$	$3\Delta x$	$4\Delta x$	$6\Delta x$	$10\Delta x$
Upstream differencing	1.00	1.00	1.00	1.00	1.00
Crank–Nicholson	0.00	0.41	0.63	0.81	0.93
Lax–Wendroff	0.00	0.58	0.75	0.88	0.95
Gadd	0.00	0.98	1.03	1.04	1.02
Leapfrog	0.00	0.43	0.67	0.86	0.95
4th-order leapfrog	0.00	0.65	0.89	0.99	1.00
Spectral	1.05	1.02	1.01	1.00	1.00
Finite element	0.00	0.87	0.99	1.01	1.00
Semi-Lagrangian (p=0, linear interpolation)	1.00	1.00	1.00	1.00	1.00
Semi-Lagrangian (p=0, cubic spline)	1.00	1.00	1.00	1.00	1.00

3. THE NON-LINEAR ADVECTION EQUATION

3.1 Introduction

An important property of the primitive equations is that the advective terms are non-linear. In this section we will consider the simple non-linear advection equation

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0 \qquad (32)$$

If initially $u = f(x)$ then the solution is $u = f(x - ut)$. However, unlike the linear advection, this is an implicit equation for the dependent variable and the solution no longer consists of the initial disturbance travelling with speed u without change of shape. As it is a non-linear equation, in general it does not have an analytical solution.

The properties of finite difference forms of the non-linear advection equation cannot be studied using the techniques introduced earlier for investigating the stability, phase errors and damping of the linear version of the equation. However we can use the integral properties of the non-linear advection equation to give guidance about suitable finite difference schemes.

3.2 Preservation of conservation properties

Multiplying (32) by u and integrating over the domain (assuming cyclic boundary conditions), we get

$$\frac{\partial E}{\partial t} = 0 \qquad E = \frac{1}{2}\int_0^L u^2 \mathrm{d}x$$

where E is the total kinetic energy. Hence E is conserved and it would be desirable that the finite difference form of the equations preserved this property.

Consider the semi-discrete form of the equation in which only the advection term has been discretised. For various schemes we will examine

$$\frac{\partial E'}{\partial t} \quad \text{where} \quad E' = \frac{1}{2}\sum_j u_j^2 \Delta x$$

and try to find schemes for which E' is conserved. The most obvious finite difference scheme is

$$\frac{\partial u_j}{\partial t} = -u_j\left(\frac{u_{j+1} - u_{j-1}}{2\Delta x}\right)$$

Multiplying by u_j and summing over all points gives

$$\frac{\partial E'}{\partial t} = -\frac{1}{2}\sum_j (u_j^2 u_{j+1} - u_j^2 u_{j-1})$$

Since the terms are not of the form $(A_{j+1} - A_j)$ there will not be cancellation of all the terms and so the energy E' is not conserved.

An alternative finite difference scheme can be derived by casting (32) in flux form

$$\frac{\partial u}{\partial t} = -\frac{\partial}{\partial x}\left(\frac{u^2}{2}\right)$$

and then using

$$\begin{aligned}\frac{\partial u_j}{\partial t} &= -\frac{1}{2}\left(\frac{u_{j+1}^2 - u_{j-1}^2}{2\Delta x}\right) \\ &= -\left(\frac{u_{j+1} + u_{j-1}}{2}\right)\left(\frac{u_{j+1} - u_{j-1}}{2\Delta x}\right)\end{aligned}$$

Analysis of this scheme reveals that once again energy is not conserved. However, the scheme

$$\frac{\partial u_j}{\partial t} = \; = -\left(\frac{u_{j+1}+u_j+u_{j-1}}{3}\right)\left(\frac{u_{j+1}-u_{j-1}}{2\Delta x}\right)$$

does conserve energy. Let us multiply both sides by $u_j\Delta x$ and add over all the points of our domain, we get:

$$\frac{\partial}{\partial t}E' = -\sum_j \frac{1}{6}(\underset{\uparrow}{u_{j+1}^2 u_j} + \overset{\downarrow}{u_j^2 u_{j+1}} - \underset{\uparrow}{u_j^2 u_{j-1}} - \overset{\downarrow}{u_{j-1}^2 u_j})$$

The terms joined by arrows cancel from consecutive grid-points and therefore the total sum is zero.

This suggests that suitable averaging can produce energy conserving schemes.

3.3 Aliasing

Aliasing occurs when the non-linear interactions in the advection term produce a wave which is too short to be represented on the grid; this wave is then falsely represented (aliased) as a wave with a larger wavelength.

Suppose we have a discrete mesh with $(N+1)$ grid points and grid length Δx, giving a domain $L = N\Delta x$. The shortest resolvable wave on this grid has a wavelength of $\lambda_{\mathrm{min}} = 2\Delta x$; therefore the maximum wavenumber l_{max} is given by

$$l_{\mathrm{max}} = \frac{L}{\lambda_{\mathrm{min}}} = \frac{N}{2}$$

Now consider how the non-linear product

$$A = u(x)\frac{\partial\varphi}{\partial x}$$

is represented on our grid. Suppose u and φ are single Fourier components with wave numbers l_1 and l_2 respectively.

$$u(x) = \sin\left(\frac{2\pi}{L}l_1 x\right); \qquad \varphi(x) = \sin\left(\frac{2\pi}{L}l_2 x\right); \qquad x_j = (j-1)\Delta x$$

Substitution in (32) gives

$$\begin{aligned} A &= \frac{2\pi}{L}l_2 \sin\left(\frac{2\pi}{L}l_1 x\right)\cos\left(\frac{2\pi}{L}l_2 x\right) \\ &= \frac{2\pi}{L}l_2\frac{1}{2}\left\{\sin\left(\left[\frac{2\pi}{L}(l_1+l_2)x\right] + \sin\left[\frac{2\pi}{L}(l_1-l_2)x\right]\right)\right\} \end{aligned}$$

and so A has contributions from wavenumbers (l_1+l_2) and (l_1-l_2). Now if the magnitudes of both these new wavenumbers are less than l_{max}, A can be correctly represented. However, if either $|l_1+l_2|$ or $|l_1-l_2|$ are greater than l_{max}, the non-linear product will be misrepresented on the grid.

Now consider what a wave with wave number $l > l_{\max}$ will look like on our grid. A little trigonometrical manipulation reveals that

$$\sin\left(\frac{2\pi}{L}lx_j\right) = -\sin\left[\frac{2\pi}{L}(2l_{\max}-l)x_j\right]$$

Therefore on the grid it is not possible to distinguish between wave numbers l and $l^* = 2l_{\max}-l$. This means that if the non-linear interaction leads to a wave number $l > l_{\max}$, then l is misrepresented as l^*—hence there is aliasing

As an example, suppose we have a wave with wavelength $\lambda = 4\Delta x/3$, which corresponds to wave number $l = L/\lambda = 3N/4$. Since $l \geq l_{\max} = N/2$, this wave number is represented as $l^* = N/4$, giving a wavelength $\lambda^* = L/l^* = 4\Delta x$. This is illustrated below.

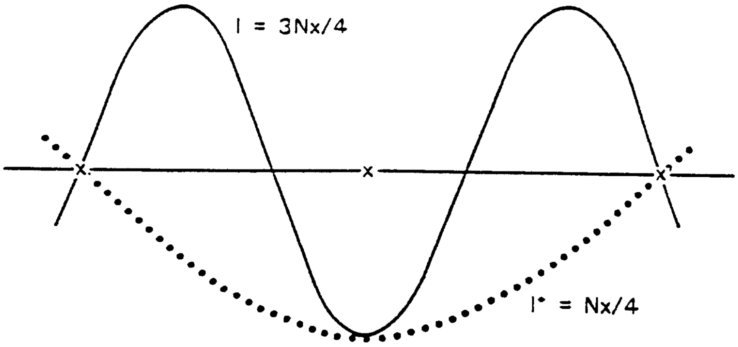


Figure 7. Graphical representation of aliasing.

3.4 Non-linear instability

As explained above, when two wave numbers l_1 and l_2 interact to give $(l_1 + l_2)$ which is greater than $l_{\max}$, the resulting wave is misrepresented as wave number $l^* = 2l_{\max} - (l_1 + l_2)$. Now if l^* is one of the original waves (l_2 say), then we have

$$l_1 = 2l_{\max} - (l_1 + l_2) \text{ giving } 2l_1 = 2l_{\max} - l_2 \tag{33}$$

To get the range of possible values of l_1 that can satisfy (33), we insert the maximum and minimum values that l_2 can have.

(i) The maximum value of l_2 is $l_{\max}$ which gives $l_1 = l_{\max}/2$—that is $\lambda_1 = 4\Delta x$.

(ii) The minimum values of l_2 is 0 which gives $l_1 = l_{\max}$—that is $\lambda_1 = 2\Delta x$.

Therefore, if one of the waves involved in the non-linear interaction has a wavelength less than $4\Delta x$ (i.e. $2\Delta x \leq \lambda_1 \leq 4\Delta x$), aliasing causes a channeling of energy towards the small wavelengths. The continuous feedback of energy leads to a catastrophic rise in the kinetic energy of wavelengths $2\Delta x$ to $4\Delta x$—this process is referred to as non-linear instability.

Note that even if wavelengths less than $4\Delta x$ are not initially present, non-linear interactions will eventually produce them.

3.5 A necessary condition for instability

Consider the semi-discrete case

$$\frac{\partial \varphi_j}{\partial t} = -u(x_j)\left(\frac{\varphi_{j+1} - \varphi_{j-1}}{2\Delta x}\right) \tag{34}$$

If $u(x_j) > 0$ everywhere, (34) can be rewritten as

$$\frac{1}{u_j}\frac{\partial \varphi_j}{\partial t} = -\left(\frac{\varphi_{j+1} - \varphi_{j-1}}{2\Delta x}\right)$$

Define the weighted energy E_w as

$$E_\mathrm{w} = \sum \frac{1}{u_j}\frac{\varphi_j^2}{2}$$

We then have

$$\begin{aligned}\frac{\partial E_\mathrm{w}}{\partial t} &= -\sum_j \varphi_j\left(\frac{\varphi_{j+1} - \varphi_{j-1}}{2\Delta x}\right)\\ &= -\frac{1}{2\Delta x}\left\{\sum_j \varphi_j\varphi_{j+1} - \sum_j \varphi_j\varphi_{j-1}\right\}\end{aligned}$$

The sums of the products are zero if there are cyclic boundary conditions. Therefore

$$\frac{\partial E_\mathrm{w}}{\partial t} = 0$$

and so E_w is conserved. This means that if initially the weighted energy is E, then at any time t we have

$$\sum_j \frac{1}{u_j}\frac{\varphi_j^2(t)}{2} = E$$

If $M = \mathrm{minimum}(1/u_j)$, then this gives

$$\sum_j \varphi_j^2(t) = \frac{2}{M}E,$$

which shows that the solution is bounded even if u is rough. Clearly it is necessary for the advecting velocity to change sign in order to obtain instability. But note that this no longer holds when time stepping is introduced.

3.6 Control of non-linear instability

(a) Eliminate the waves that cause non-linear instability by Fourier analysing the fields, discarding the wavelengths less than $4\Delta x$ and then reconstituting the field (in fact it is only necessary to discard wavelengths less than $3\Delta x$).

(b) Use a smoothing operator which reduces the amplitude of the short waves while having little effect on the meteorologically important waves.

(c) Introduce an explicit diffusion term.

(d) Use a time integration scheme with built-in diffusion (e.g. the Lax–Wendroff scheme).

(e) Introduce smoothing directly into the finite difference scheme in order to preserve integral constraints such as energy conservation. A classic example of this is the Arakawa scheme for the non-linear vorticity equation.

(f) Use a Galerkin technique (spectral or finite element). For these, the space discretization conserves quadratic invariants, though this property cannot be guaranteed when time discretization is introduced.

(g) Use a semi-Lagrangian scheme for advection.

4. TOWARDS THE PRIMITIVE EQUATIONS

4.1 Introduction

A major problem in numerical weather prediction is to have a proper representation of the geostrophic adjustment process—this is associated with gravity–inertia waves.

In the early days the adjustment process in numerical forecasts was taken care of by using the geostrophic approximation in the vorticity equation; the effect of this was to eliminate the gravity waves entirely. Later the primitive equations were used and then the treatment of the gravity–inertia waves became very important.

4.2 The one-dimensional gravity-wave equations

The one-dimensional linearised gravity-wave equations (derived from the shallow-water equations) are

$$\frac{\partial u}{\partial t} + g\frac{\partial h}{\partial t} = 0 \qquad \frac{\partial h}{\partial t} + H\frac{\partial u}{\partial x} = 0 \tag{35}$$

These equations can be easily manipulated into two separate wave equations for u and h, hence they form a system of hyperbolic equations. Taking the time derivative of the u-equation and the x-derivative of the h-equation we get, upon elimination of h,

$$\frac{\partial^2 u}{\partial t^2} + gH\frac{\partial^2 u}{\partial x^2} = 0$$

and similarly for h

If we seek solutions of the form

$$u = \hat{u}\exp[\mathrm{i}k(x-ct)] \qquad h = \hat{h}\exp[\mathrm{i}k(x-ct)] \tag{36}$$

we find that the phase speed of the waves is given by $c = \pm(gH)^{1/2}$. Therefore, there are two waves travelling in opposite directions along the x-axis.

We now consider ways of solving these equations using finite difference techniques. It is convenient to divide the schemes into two categories—explicit and implicit.

4.2 (a) Explicit schemes.

When (35) is put in finite difference form using centred space and time differences (leapfrog scheme), we have

$$\frac{u_j^{n+1}-u_j^{n-1}}{2\Delta t} = -g\delta h_j^n \tag{37}$$

$$\frac{h_j^{n+1}-h_j^{n-1}}{2\Delta t} = -H\delta u_j^n \tag{38}$$

where δ represents the centred finite difference operator corresponding to the first derivative. The stability of this scheme is determined by substituting the following into (37) and (38)

$$u_j^n = \hat{u}\lambda^n \exp[ikx_j] \qquad h_j^n = \hat{h}\lambda^n \exp[ikx_j]$$

and then finding the condition for which $|\lambda| \le 1$ This procedure gives

$$\lambda^2 + 2ip\lambda - 1 = 0 \qquad p = -\sqrt{gH}\frac{\Delta t}{\Delta x}\sin k\Delta x$$

Proceeding as in Subsection 2.5 when dealing with the leapfrog scheme for advection, it can be shown that there is linear computational stability provided.

$$\Delta t \le \frac{\Delta x}{(gH)^{1/2}},$$

and that this scheme is neutral. However, although there is no damping, there are phase errors and computational dispersion; also there is a computational mode since it is a three-time-level scheme.

When forward time differences are used with centred space differences, we find that

$$|\lambda|^2 = 1 + gH\left(\frac{\Delta t}{\Delta x}\right)^2(\sin k\Delta x)^2$$

therefore this scheme is absolutely unstable.

4.2 (b) Implicit schemes.

Consider what happens when the space derivatives are replaced by centred space differences averaged over time levels $n-1$ and $n+1$; centred differences will be used for the time derivatives.

$$\frac{u_j^{n+1}-u_j^{n-1}}{2\Delta t} = -g\left(\frac{\delta h_j^{n+1}+\delta h_j^{n-1}}{2}\right) \tag{39}$$

$$\frac{h_j^{n+1}-h_j^{n-1}}{2\Delta t} = -H\left(\frac{\delta u_j^{n+1}+\delta u_j^{n-1}}{2}\right) \tag{40}$$

where δ represents a centred finite difference operator corresponding to the first derivative. Applying δ to (39) we find δu_j^{n+1}, which is then substituted into (40) to give

$$gH(\Delta t)^2\delta^2 h_j^{n+1} - h_j^{n+1} = F(h^{n-1}, u^{n-1}) \tag{41}$$

Therefore, since the RHS is known, (41) is an elliptic equation which can be solved for h_j^{n+1}, given suitable boundary conditions; u_j^{n+1} can be found in a similar fashion.

It can be shown that this scheme is absolutely stable and so any time step can be used. However, a Helmholtz equation has to be solved every time step and this can be computationally expensive.

An implicit scheme using forward time differences can be constructed using the Crank–Nicolson approach in which

$$\frac{u_j^{n+1}-u_j^{n-1}}{2\Delta t} = -g(\beta_n\delta h_j^n + \beta_{n+1}\delta h_j^{n+1}) \tag{42}$$

$$\frac{h_j^{n+1}-h_j^{n-1}}{2\Delta t} = -H(\beta_n\delta u_j^n + \beta_{n+1}\delta u_j^{n+1}) \tag{43}$$

where β_n and β_{n+1} are weights such that $\beta_n + \beta_{n+1} = 1$ ($\beta_n = 1, \beta_{n+1} = 0$ corresponds to the forward time-centred space scheme, which is absolutely unstable).

A stability analysis of (42) and (43) shows that there is instability if $\beta_n > 1/2$ and absolute stability if $\beta_n \le 1/2$

4.3 Staggered grids

We now consider the best way of distributing the variables u and h on the grid.

Initially we might expect that u and h should be held at each grid point.

x	x	x	x	x
u, h	u, h	u, h	u, h	u, h

However careful examination shows that, if centred differences are used, we have two separate subgrids. This means that the solutions on the subgrids can become decoupled from one another.

Displacing the grid points which carry the h variable to the middle between the u points we get rid of this problem as now the centred space derivative uses successive points of the same variable.

x	o	x	o	x	o	x	o	x	o
u	h	u	h	u	h	u	h	u	h

This also has the effect of improving the dispersion characteristics of any scheme because the effective grid length if halved. These ideas can be extended to the two dimensional problem

$$\frac{\partial u}{\partial t} + g\frac{\partial h}{\partial x} = 0 \qquad \frac{\partial v}{\partial t} + g\frac{\partial h}{\partial y} = 0 \qquad \frac{\partial h}{\partial t} + H\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) = 0$$

There are various grids that can be used, they are known as Arakawa A–E grids and are shown below:

As well as space staggering it is often desirable to have time staggering. This is particularly useful for leapfrog schemes where the most common distribution of variables is known as the Eliassen grid. However grid E is the same as grid B tilted by 45°.

There is not a general consensus as to which grid has the best properties although grids A and D are known to be worst. Grid C was used in the grid-point model of ECMWF.

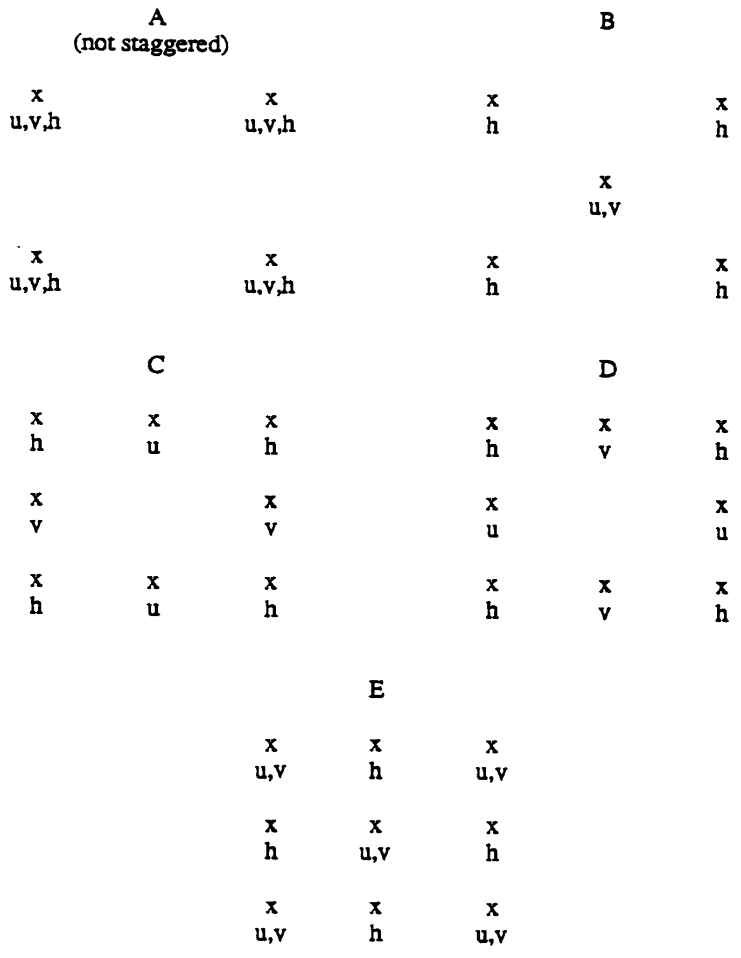


Figure 8. The arrangement of variables on the Arakawa A–E grids.

4.4 The shallow-water equations.

To make our equations more realistic we should include the advection. Therefore, sticking to the linear one-dimension case we get

$$\frac{\partial u}{\partial t}+u_0\frac{\partial u}{\partial x}+g\frac{\partial h}{\partial x} = 0$$
$$\frac{\partial h}{\partial t}+u_0\frac{\partial h}{\partial x}+H\frac{\partial u}{\partial x} = 0 \tag{44}$$

Substituting (36) into (44) gives the dispersion relationship

$$c = u_0 \pm (gH)^{1/2}$$

When the leapfrog scheme is used with centred space differencing, the stability criterion becomes

$$\Delta t \le \frac{\Delta x}{u_0+(gH)^{1/2}}$$

If $H = 10\ \text{km}$, the phase speed of the gravity waves is 313 m/s; now if $\Delta x = 10^5\ \text{m}$ and $u_0 = 100\ \text{m/s}$ the stability criterion becomes $\Delta t \le 4.0\ \text{min}$. Note that this criterion is mainly determined by the gravity wave speed.

The stability analysis of the shallow-water equations will be performed in two dimensions using the spectral method. The point we want to stress here is that the adjustment terms limit the upper size of the time step to, typically, one third of the one possible for the stable treatment of the advection terns.

4.5 Increasing the size of the time step

We saw in the former section that in an explicity treatment of the shallow water equations representing synoptic scale features only ($\Delta x \approx 100\ \text{km}$) the time step for stability is restricted to a value much lower than the typical time scale of such features, therefore increasing the amount of calculations to be performed much above what would be desirable.

Several ways of increasing the allowed time step have been devised but only the most successful ones will be revised here.

4.5 (a) The splitting method.

For the set of equations discussed in Subsection 4.4, there are clearly two different physical mechanisms acting. Therefore, it may be desirable to treat the advection and gravity parts separately. Marchuk devised the splitting technique which makes this possible.

The equations are split as follows:

$$\frac{\partial u}{\partial t}+u_0\frac{\partial u}{\partial x} = 0 \qquad \frac{\partial h}{\partial t}+u_0\frac{\partial h}{\partial x} = 0 \qquad \text{advection} \tag{45}$$

$$\frac{\partial u}{\partial t}+g\frac{\partial h}{\partial x} = 0 \qquad \frac{\partial h}{\partial t}+H\frac{\partial u}{\partial x} = 0 \qquad \text{adjustment} \tag{46}$$

The following procedure is then used.

(a) Use standard finite difference techniques to solve (45). If h^* and u^* denote new values after one time step we have

$$h^* = \lambda_{\text{adv}} h^n \qquad u^* = \lambda_{\text{adv}} u^n$$

(b) The new values are now used as the starting point for solving (46)

$$h^{n+1} = \lambda_{\text{adj}} h^* \qquad u^{n+1} = \lambda_{\text{adj}} u^*$$

Substituting for h^* and u^* gives

$$h^{n+1} = \lambda h^n \quad \text{and} \quad u^{n+1} = \lambda u^n \quad \text{where} \quad \lambda = \lambda_{\text{adj}} \lambda_{\text{adv}}$$

The complete scheme is stable provided $|\lambda| \le 1$. and this will be satisfied if $|\lambda_{\text{adj}}| \le 1$ and, therefore, there is stability if each of the separate steps is stable.

It is possible to exploit the fact that the same time step does not have to be used for each step. For example, the gravity wave speed is larger than the advection speed and so it appears reasonable to use a large time step for advection (Δt say) and a number of smaller time steps for the gravity wave equations (M steps of δt, with $\Delta t = M\delta t$). The two steps will be stable provided

$$u_0 \frac{\Delta t}{\Delta x} \le 1 \qquad c\frac{\delta t}{\Delta x} = c\frac{\Delta t}{M\delta t} \le 1$$

Typically c is about three times larger than u_0 and so it is appropriate to use $M = 3$ and to take three adjustment steps to each advection step. This approach has been used effectively by the UK Met. Office—see Gadd (1978).

4.5 (b) Forward–backward scheme.

Let us consider the adjustment terms of the one-dimensional shallow-water equations as given by (35). The procedure is to solve the second equation by means of a FTSC step and then to use the calculated values of the height for calculating new values of u using the first equation.

This can be stated as follows:

$$h_j^{n+1} = h_j^n - \frac{H}{2}\frac{\Delta t}{\Delta x}(u_{j+1}^n - u_{j-1}^n) \qquad \text{forward}$$

$$u_j^{n+1} = u_j^n - \frac{g}{2}\frac{\Delta t}{\Delta x}(h_{j+1}^{n+1} - h_{j-1}^{n+1}) \qquad \text{backward}$$

If we use the von Neumann method for analysing the stability of this scheme we find that

$$\Delta t \le \frac{2\Delta x}{(gH)^{1/2}}$$

which is twice the time step allowed by the leapfrog method. Furthermore, the scheme is neutral and, although the second equation looks similar to an implicit scheme, the set of equations is decoupled as in an explicit method and we don't have to solve a coupled system of simultaneous equations. .

4.5 (c) Pressure averaging.

A procedure somewhat similar to the forward–backward scheme is the pressure averaging technique. The name comes from the primitive equations using z as the vertical co-ordinate, where the adjustment tern for the momentum equations is given by the so-called pressure gradient term. As we are dealing here with the shallow-water equations, it would be more adequate to call it height or geopotential averaging.

The idea is to take as the height in the wind equation some average of the previous, present and future time values and using centred time derivatives.

Therefore the momentum equation reads

$$\frac{u_j^{n+1}-u_j^{n-1}}{2\Delta t} = -\frac{g}{2\Delta x}\{(1-2\varepsilon)(h_{j+1}^n - h_{j-1}^n) + \varepsilon[(h_{j+1}^{n+1} - h_{j-1}^{n+1}) + (h_{j+1}^{n-1} - h_{j-1}^{n-1})]\}$$

which reduces to the leapfrog scheme if $\varepsilon = 0$. If we take $\varepsilon = 1/4$ we get, from the von Neumann stability analysis, the condition

$$\Delta t \le \frac{2\Delta x}{(gH)^{1/2}},$$

which is the same as we got for the forward-backward scheme.

4.5 (d) Semi-implicit scheme.

It was stated in section Subsection 4.2 (b) that an implicit treatment of the gravity wave equation is absolutely stable for any size of the time step, therefore, we could try such a treatment for the adjustment terms in the shallow water equations while keeping an explicity formulation of the advection terms.

The disretized equations in two dimensions then read

$$\begin{aligned} u_j^{n+1} &= u_j^{n-1} - \Delta t V_j^n \cdot \vec{\nabla} u_j^n - \frac{g\Delta t}{2}\nabla_x(h_j^{n+1} + h_j^{n-1}) \\ v_j^{n+1} &= v_j^{n-1} - \Delta t V_j^n \cdot \vec{\nabla} v_j^n - \frac{g\Delta t}{2}\nabla_y(h_j^{n+1} + h_j^{n-1}) \\ h_j^{n+1} &= h_j^{n-1} - \Delta t V_j^n \cdot \vec{\nabla} h_j^n - \frac{H\Delta t}{2}\vec{\nabla} \cdot (\vec{\nabla}_j^{n+1} + \vec{\nabla}_j^{n-1}) \end{aligned} \tag{47}$$

where

$$V_j^n = (u_j^n, v_j^n) \qquad \vec{\nabla} = (\nabla_x, \nabla_y)$$

and ∇_x and ∇_y are the centred approximations to the x and y derivatives, respectively. Upon substitution of u_j^{n+1} and v_j^{n+1} from the first two equations into the third equation we get

$$\nabla^2 h_j^{n+1} - \frac{4(\Delta s)^2}{gH(\Delta t)^2} h_j^{n+1} = F^{n,n-1}$$

where $\Delta s = \Delta x = \Delta y$. This is a Helmholtz equation which has to be solved at every time step and, therefore, it is more expensive than the explicit method. Nevertheless, there are fast Helmholtz solvers which are described in chapter 8 and a stability analysis, which we will perform in Section 6 using the spectral approach shows that the time step size is no longer limited by the phase speed of the (fast) gravity waves, but by the speed of the more slow

Rossby modes.

Computer tests show that the increased size of the time step overcomes the higher amount of work needed at every time step, and so the semi-implicit time scheme is faster than the explicit one The advantage is most notable if we use the spectral technique with spherical harmonics as these are eigenfunctions of the Laplacian operator and, therefore, the set (47) becomes a decoupled set of equations, one for every spectral component of the height function.

4.6 Diffusion

The only terms not treated so far from the shallow-water equations in its linearized form are the diffusion terms. The linear diffusion equation for a function A in one dimension can be written as:

$$\frac{\partial A}{\partial t} = K\frac{\partial^2 A}{\partial x^2}; \qquad \mathrm{K} > 0 \tag{48}$$

This is a parabolic equation whose analytical solution, when we use periodic boundary conditions and a single wave of wave number k as the initial condition can be shown to be

$$A(x, t) = A_0 \sin(kx) \exp[-k^2 Kt]$$

which represents the initial disturbance with an amplitude decaying with time.

We will consider here only three time-stepping schemes combined with centred second-order space differencing in order to show that, as it was the case with the other terms, an explicit treatment is in general conditionally stable while an implicit treatment is normally stable.

4.6 (a) Explicit forward scheme.

$$\frac{A_j^{n+1} - A_j^n}{\Delta t} = K\frac{A_{j+1}^n - 2A_j^n + A_{j-1}^n}{(\Delta x)^2} \tag{49}$$

As usual, we consider the behaviour of a single harmonic and assume

$$A_j^n = \lambda^n C \exp[\mathrm{i}kx_j]$$

Substituting this into (49) gives

$$\lambda = 1 - 4\sigma\left(\sin\frac{k\Delta x}{2}\right)^2 \quad \text{with} \quad \sigma = \frac{K\Delta t}{(\Delta x)^2}$$

For stability we require $|\lambda| \le 1$ and this is satisfied for all wavelengths provided $\sigma \le 1/2$. However, though stability is ensured by using this condition, a value of σ in the range $1/4 \le \sigma \le 1/2$ gives a negative value of λ, which causes the amplitude of the wave to switch sign between successive time steps. This may be avoided by choosing $\sigma \le 1/4$.

In numerical models, a typical value of the eddy diffusivity is $K = 10^5\ \mathrm{m^2/s}$. With $\Delta x = 100\ \mathrm{km}$ the stability condition $\sigma \le 1/4$ is satisfied if $\Delta t \le 2 \times 10^6\ \mathrm{s}$. This is sufficiently large for it not to produce any problems. However, this is not the case in the vertical where a typical grid spacing of $1\ \text{km}$ leads to $\Delta t \leq 2\ \text{s}$.

4.6 (b) Classical implicit scheme.

In this scheme, the space derivative is evaluated at time level $n+1$. The scheme then reads:

$$\frac{A_j^{n+1} - A_j^n}{\Delta t} = K + \frac{A_{j+1}^{n+1} - 2A_j^{n+1} + A_{j-1}^{n+1}}{(\Delta x)^2}$$

and the usual stability analysis gives

$$\lambda = \frac{1}{1 + 4\sigma^2 \left(\sin \frac{k\Delta x}{2}\right)^2}$$

which has $|\lambda| \leq 1$ for all values of k and σ. Therefore, the scheme is absolutely stable.

4.6 (c) Crank–Nicholson scheme.

This is a mean between the two former schemes and the space derivative is evaluated at time level $n + 1/2$ by averaging over time levels n and $n+1$.

Like the classical implicit method, the Crank–Nicholson scheme is absolutely stable. However, the advantage of this scheme is that it is second-order accurate in time as opposed to first-order accuracy in time of both the explicit and the classical implicit methods.

It is interesting to generalize this approach by weighting the present and future values of the right hand side with weights β_n and β_{n+1}, subject to the condition $\beta_n + \beta_{n+1} = 1$. Some experiments suggest that values of $\beta_n = 1/4$, $\beta_{n+1} = 3/4$ give an accurate scheme with which long time steps can be used.

When the eddy diffusivity and the grid spacing vary, the continuous diffusion equation is

$$\frac{\partial A}{\partial t} = \frac{\partial}{\partial x}\left(K \frac{\partial A}{\partial x}\right)$$

and the generalized time stepping just described can be written

$$\frac{A_j^{n+1} - A_j^n}{\Delta t} = \frac{1}{\Delta x_j}\left\{\frac{K_{j+1/2}}{\Delta x_{j+1/2}}[\beta_n(A_{j+1}^n - A_j^n) + \beta_{n+1}(A_{j+1}^{n+1} - A_j^{n+1})]\right.$$
$$\left.\frac{K_{j-(1/2)}}{\Delta x_{j-(1/2)}}[\beta_n(A_j^n - A_{j-1}^n) + \beta_{n+1}(A_j^{n+1} - A_{j-1}^{n+1})]\right\}$$

where $\Delta x_j = x_{j+1/2} - x_{j-1/2}$, $\Delta x_{j+1/2} = x_{j+1} - x_j$ and $\Delta x_{j-1/2} = x_j - x_{j-1}$.

5. The semi-Lagrangian technique

5.1 Introduction

So far we have taken an Eulerian view and considered what was the evolution in time of a dependent variable at fixed points in space and in the spectral and finite elements we will consider what is the time evolution of some coefficients multiplying some basis functions also fixed in space; in other words, we used the partial time derivative $\partial/\partial t$.

A few years ago, several attempts were made to build 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.

Let us consider the one-dimensional advection equation

$$\left(\frac{\partial}{\partial t}+u\frac{\partial}{\partial x}\right)\varphi = 0 \tag{50}$$

where φ is the advected property and u is the advection velocity. This equation can be recast in the form

$$\frac{\mathrm{d}\varphi}{\mathrm{d}t} = 0 \tag{51}$$

where the left-hand side stands for the Lagrangian derivative and its meaning is the time evolution of a material volume and equation (51) could be read as: the property φ is conserved within an air parcel. The discretization can be written as

$$\frac{\varphi_A^{t+\Delta t}-\varphi_D^{t}}{\Delta t} = 0$$

where subindexes A and D indicate the arrival (at time instant $t+\Delta t$) and departure (at time instant t) points of the considered air parcel.

If we know the initial distribution of φ (defined, for example, on a regular array of points) then by tracking the fluid parcels we end up with information about the distribution of φ at some later time, but in general the points where we know the value of φ will not be uniformly distributed any more and this makes the procedure very difficult to apply.

The semi-Lagrangian technique overcomes this difficulty by considering the end points as consisting of a regular mesh and tracking back the origin of each parcel. The simplest method for finding the value of φ at gridpoint j at time level $n+1$ (φ_j^{n+1} say) consists in tracking back the air parcel over one time step to find where it was at time level n. Having located it origin we now find its φ value by interpolation from the values at the neighbouring grid points at time level n.

If the interpolated value is φ_*^n we have

$$\varphi_j^{n+1} = \varphi_*^n \tag{52}$$

5.2 Stability in one-dimension

Let us consider the linear advection equation

$$\frac{\mathrm{d}\varphi}{\mathrm{d}t} \equiv \frac{\partial\varphi}{\partial t} + C\frac{\partial\varphi}{\partial x} = 0 \tag{53}$$

The distance travelled during the last interval Δt by an air parcel arriving at point x_j is $C\Delta t$, therefore it comes from a point

$$x_* = x_j - C\Delta t \tag{54}$$

If this point lies between grid points $(j-p)$ and $(j-p-1)$, and we call α the fraction of grid length from point x_* to point x_{j-p} we have

$$C\Delta t = (p+\alpha)\Delta x \tag{55}$$

and using linear interpolation to find φ_*^n we get

$$\varphi_j^{n+1} = \varphi_*^n = (1-\alpha)\varphi_{\text{j-p}}^n + \alpha\varphi_{\text{j-p-1}}^n \tag{56}$$

(Note that when $p = 0$, $\alpha = C\Delta t/\Delta x$ and (56) becomes identical to the upstream differencing scheme). We study the stability using the von Neumann method and, therefore, assume a solution of the form

$$\varphi_{\mathrm{j}}^n = \varphi_0\lambda^n \exp[\mathrm{i}kx_j] \tag{57}$$

substituting we get

$$\lambda = \{1-\alpha(1-\exp[-\mathrm{i}k\Delta x])\}\exp[-\mathrm{i}pk\Delta x] \tag{58}$$

and

$$D \equiv |\lambda| = [1-2\alpha(1-\alpha)\{1-\cos(k\Delta x)\}]^{1/2}. \tag{59}$$

Therefore $|\lambda| \le 1$ as long as $\alpha(1-\alpha) \ge 0$, that is

$$0 \le \alpha \le 1 \tag{60}$$

the scheme is, therefore, stable if the interpolation points are the two nearest ones to the departure point, but it is neutral only if $\alpha = 0$ or $\alpha = 1$,that is to say when no interpolation is needed. We will come to this point later.

We find that heavy damping occurs for the shortest wavelengths (there is complete extinction when $l = 2$ and $\alpha = 0.5$). but the damping decreases as l increases. A strange feature of this scheme (peculiar to the case of 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 to the above can be carried out for quadratic interpolation. Once again the scheme is absolutely stable provided φ_*^n is computed by interpolation from the nearest three grid points. This scheme has less damping than the linear interpolation, but the phase representation is not improved. It is easy to show that when the departure point is within half a grid length from the grid point (i.e. $p = 0$), this scheme becomes identical to the Lax–Wendroff scheme.

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

5.3 Cubic spline interpolation

An accurate way of finding the value of φ at the departure point is to use cubic spline interpolation. The spline $S(x)$ is defined to be a cubic polynomial within any grid interval, where the coefficients are chosen so that

(i) $S(x_j) = \varphi_j$ at each gridpoint

(ii) the gradient of $S(x)$ is continuous

(iii) $\int_L (\mathrm{d}^2S/\mathrm{d}x^2)\mathrm{d}x$ is minimised

It can then be shown that, in the interval $x_{j-1} \le x \le x_j$, the spline is

$$\begin{aligned} S(x) = {} & \frac{D_{j-1}}{\Delta x^2}(x_j - x)^2(x - x_{j-1}) - \frac{D_j}{\Delta x^2}(x - x_{j-1})^2(x_j - x) \\ & + \frac{\varphi_{j-1}}{\Delta x^3}(x_j - x)^2\{2(x - x_{j-1}) + \Delta x\} + \frac{\varphi_j}{\Delta x^3}(x - x_{j-1})^2\{2(x_j - x) + \Delta x\} \end{aligned} \tag{61}$$

where φ_{j-1} and φ_j are the grid-point values of φ at $j-1$ and j, and D_{j-1} and D_j are the corresponding gradients of the splines derived from

$$\frac{D_{j-1} + 4D_j + D_{j+1}}{6} = \frac{\varphi_{j+1} - \varphi_{j-1}}{2\Delta x} \tag{62}$$

The implementation of this scheme requires two steps:

(a) The derivatives of the splines at each grid point j and at time level n (D_j^n say) are derived from the set of simultaneous equations defined by (62).

$$\frac{D_{j-1}^n + 4D_j^n + D_{j+1}^n}{6} = \frac{\varphi_{j+1}^n - \varphi_{j-1}^n}{2\Delta x}$$

(b) Having found the point x_* from which an air parcel originates, the value of φ_*^n is calculated from (61) using the values of φ and D at the two neighbouring grid points. If point x_* lies between grid points $m = j - p$ and $m - 1$ at a distance $\hat{\alpha}\Delta x$ from point m, then (61) gives an expression for φ_*^n in terms of φ_{m-1}^n, φ_m^n, D_{m-1}^n, D_m^n and $\hat{\alpha}$. The time stepping algorithm (52) then becomes

$$\begin{aligned} \varphi_j^{n+1} = {} & \varphi_m^n - \hat{\alpha}D_m^n\Delta x + \hat{\alpha}^2\{(D_{m-1}^n + 2D_m^n)\Delta x + 3(\varphi_{m-1}^n - \varphi_m^n)\} \\ & - \hat{\alpha}^3\{(D_{m-1}^n + D_m^n)\Delta x + 2(\varphi_{m-1}^n - \varphi_m^n)\} \end{aligned} \tag{63}$$

A corresponding expression can obviously be derived for the case when $u_0 < 0$.

Although cubic spline interpolation requires much more computation than a linear interpolation (compare (56) with (63)), the characteristics of the cubic spline scheme are far superior. Therefore, in choosing a scheme it is necessary to balance accuracy against computational expense.

Let us now turn to the non-linear advection equation. In this case the advecting velocity, and hence the displacement of every air parcel, is a function of x. We can still use the same method and estimate the displacement by $d_j = u_j\Delta t$ and, therefore, p and α will depend upon j.

A more accurate estimate of the displacement is found by using an advecting velocity from midway between the departure and arrival points; this could be estimated in many ways. This is the equivalent to the Crank–Nicholson scheme if we estimate the advecting velocity at time $t + \Delta t/2$, or to the centred time-differencing schemes if we use the estimate at time t and the departure point at time $t - \Delta t$.

5.4 Cubic Lagrang interpolation and shape preservation

Cubic spline interpolation is quite expensive and can be unusable in more than one domension. A simpler although not so accurate interpolation is provided by the cubic Lagrange polynomials defined as follows:

Q(x) is a cubic polynomial covering 4 consecutive gridpoints

Q(x_j)=φ_j at each of these four grid-points.

Then Q(x) can be expressed as

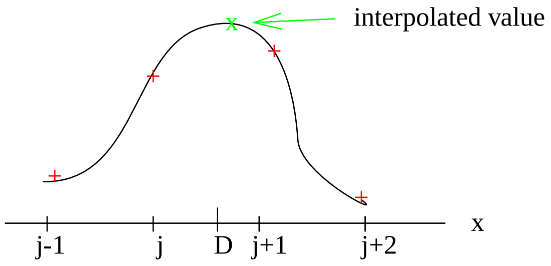
$$Q(x) = \sum_{i=1}^{4} C_i(x)\varphi_i$$

where the functions $C_i(x)$ can be computed as

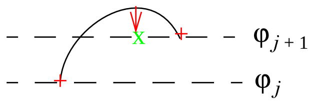
$$C_i(x) = \frac{\prod_{k \neq i}^{4}(x - x_k)}{\prod_{k \neq i}^{4}(x_i - x_k)}$$

Cubic Hermite interpolation is somewhat similar but the input data are the values and the derivatives at the two gridpoints surrounding the interpolation point.

Any high order interpolation can produce artificial maxima and minima not present in the original data. Supose we want to interpolate to point D, by means of a cubic polynomial, the function given at the 4 consecutive grid-points (j-1), j, (j+1) and (j+2)



As pure advection can not produce new maxima in the advected function, it is convenient to avoid possible overshooting in the cubic interpolations. If the interpolation was done by Hermite polynomials, appropriate modification of the derivatives at points j and j+1 can lead to the elimination of maxima in the interpolation interval. In the case of cubic Lagrange polynomials, the technique called quasi-monotone interpolation can be applied: after interpolation, the interpolated value is restricted to stay within the interval $\varphi_j \rightarrow \varphi_{j+1}$



5.5 Various quasi-Lagrangian schemes in 2D

We will consider here only schemes using centred time differences. The general form of the evolution equation for a given parameter $X(x, y, t)$ can be written

$$\frac{\partial X}{\partial t} + U\frac{\partial X}{\partial x} + V\frac{\partial X}{\partial y} = L \cdot X + N(X) \tag{64}$$

where $L \cdot X$ is the linear part of the equation and $N(X)$ the non-linear part.

The left-hand side is the Lagrangian total derivative

$$\frac{\mathrm{d}X}{\mathrm{d}t} \equiv \frac{\partial X}{\partial t} + U\frac{\partial X}{\partial x} + V\frac{\partial X}{\partial y}$$

5.5 (a) Method with interpolation (Robert 1982).

The evolution equation is discretized as follows:

$$\frac{X_{\mathrm{G}}^{t+\Delta t} - X_{\mathrm{O}}^{t-\Delta t}}{2\Delta t} = L \cdot \frac{X_{\mathrm{G}}^{t+\Delta t} + X_{\mathrm{O}}^{t-\Delta t}}{2} + \{N(X^t)\}_{\mathrm{I}}$$

X_{G} is the value of X at grid point G, X_{O} is the value of X at the point O where the particle comes from, X_{I} is the value of X at the mid-point between O and G. Superscripts t, $t - \Delta t$ and $t + \Delta t$ refer to time levels. This method needs the interpolation of $X^{t-\Delta t}$ at point O and X^t at point G.

5.5 (b) Method avoiding one interpolation (Ritchie 1986).

We define point O' as the closest grid point to O and I' as the mid-point between O' and G. We can write

$$U = U^* + U' \qquad V = V^* + V'$$

where $2U^*\Delta t$ and $2V^*\Delta t$ are the components of vector $\overrightarrow{\mathrm{O'G}}$.

The method consists in a semi-Lagrangian treatment of the advection by the wind (U^*, V^*), the advection by the residual wind (U', V') being incorporated into the non-linear part of the right-hand side. This discretization reads:

$$\frac{X_{\mathrm{G}}^{t+\Delta t} - X_{\mathrm{O'}}^{t-\Delta t}}{2\Delta t} = L \cdot \frac{X_{\mathrm{G}}^{t+\Delta t} + X_{\mathrm{O'}}^{t-\Delta t}}{2} + \{N(X^t)\}_{\mathrm{I'}} - \left(U'\frac{\partial X}{\partial x} + V'\frac{\partial X}{\partial y}\right)_{\mathrm{I'}}^{t}$$

This method avoids the interpolation at point O, and the residual interpolation at the point I' is very simple due to the three possible locations shown in Fig. 9 . The damping, on the other hand, is reduced due to the lack of interpolation at the departure point.

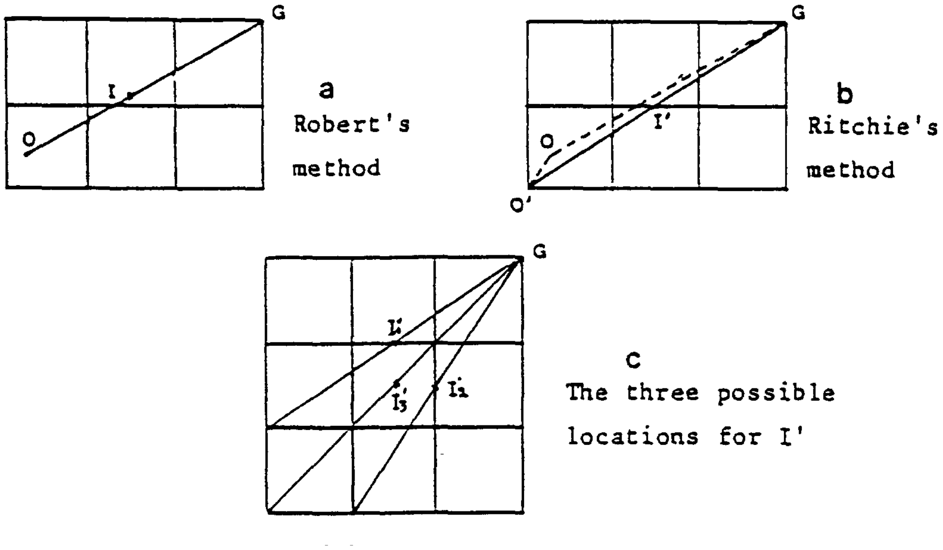


Figure 9. Location of the points where the interpolation is performed for quasi-Lagrangian techniques.

5.5 (c) Method without any interpolation interpolation .

One supplementary simplification can be achieved by evaluating the non-linear terms by taking the average at time t between their values at grid points G and O′.

$$\{N(X^t)\}_{\mathrm{I}'} - \left(U'\frac{\partial X}{\partial x} + V'\frac{\partial X}{\partial y}\right)^t_{\mathrm{I}'} = \frac{1}{2}\left[(N(X^t))_{\mathrm{G}} + (N(X^t))_{\mathrm{O}'} + -\left(U'\frac{\partial X}{\partial x} + V'\frac{\partial X}{\partial y}\right)^t_{\mathrm{G}} - \left(U'\frac{\partial X}{\partial x} + V'\frac{\partial X}{\partial y}\right)^t_{\mathrm{O}'}\right]$$

5.5 (d) Method used at ECMWF.

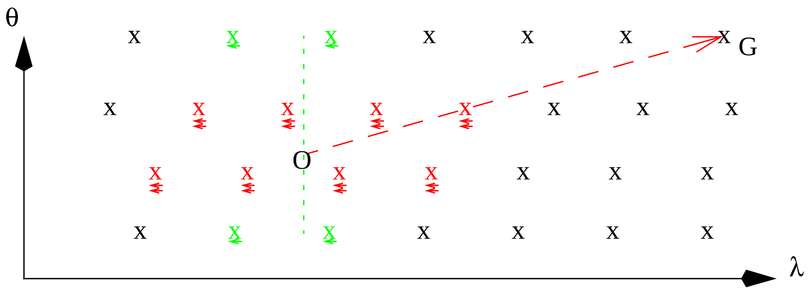


Figure 10: 12-point interpolation used in the horizontal at ECMWF

At ECMWF the method of Robert is used with cubic Lagrange polynomials and quasi-monotone limiter. In order to reduce the cost of the interpolation, the interpolation in longitude at the rows not immediate adjacent to the departure point O is done linearly (singly underlined points in Figure 10). The procedure is as follows and is valid for a reduced Gaussian grid to be described later. The longitude and latitude of the departure point O is found (see later). At each of the two rows of grid-points second nearest neighbours to the departure point, linear interpolations are performed to the longitude of the departure point. At the nearest neighboring grid point rows, cubic quasi-monotone interpolations are performed to the same longitude. Finally a quasi-monotone cubic interpolation in latitude is performed using the 4 interpolated values. In the vertical a similar procedure is followed: at each nearest neighboring level to the departure point a 12-point interpolation is performed and at the second nearest neighbouring levels a bilinear interpolation is done. Finally a quasi-monotone (or standard, depending of the variable to be interpolated) cubic interpolation is done in the vertical direction. A total of 32 points are used then for each three-dimensional interpolation.

5.6 Stability on the shallow water equations

We can perform the stability analyses of the three methods, as applied to the shallow water equations, in a form exactly similar to the way we did it in the Eulerian case. We are not going to follow the procedure again but instead we present the results on stability and dispersion characteristics of the three schemes.

(a) For the Robert scheme the stability criterion is

$$f^2 \Delta t^2 < 1$$

as long as the interpolation is done by using the grid point values around the origin point, and the adjustment terms are treated implicitly.

(b) The Ritchie scheme leads to a stability criterion for the advective part of

$$(MU' + NV')\Delta t \le 1$$

which is analogous to the one we obtained with the semi-implicit scheme replacing U_{O} and V_{O} by the residual velocity (U', V'). This relationship can be shown to be always true, due to the way in which the residual velocity was defined.

(c) The stability criterion of the fully non-interpolating scheme is completely analogous to the former one.

The dispersion $r \equiv \frac{\alpha_{\mathrm{numerical}}}{\alpha_{\mathrm{analytical}}}$ is given in Fig. 11 as a function of $2U_{\mathrm{O}}\frac{\Delta t}{\Delta x}$ for the analytical slow solution in the one dimensional case.

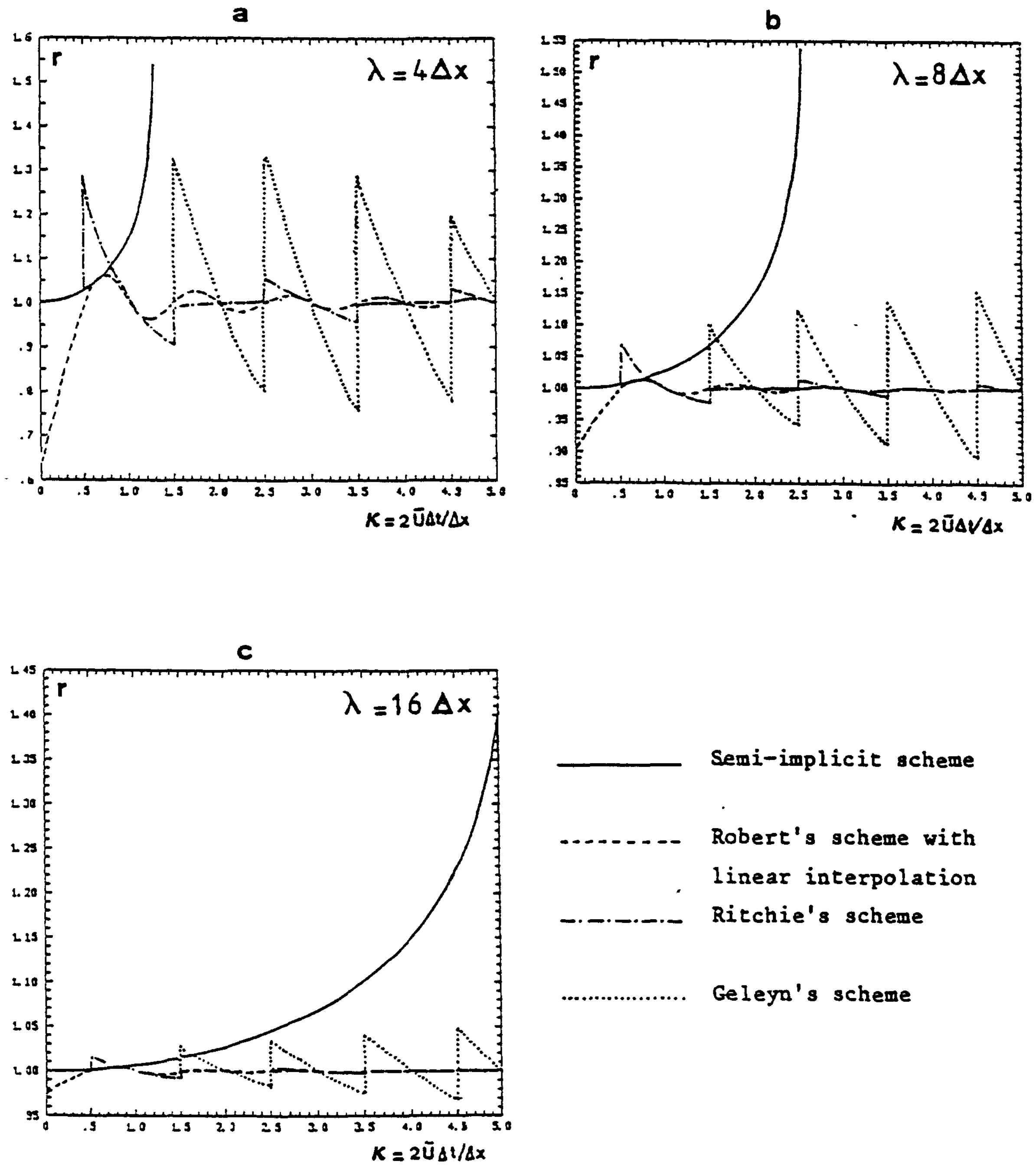


Figure 11. Effect of time integration on the slow wave for various values of the wavelength.

5.7 Computation of the trajectory

The computation of the departure point for a parcel of air arriving at a grid-point G at time $t + \Delta t$ can be done by solving the vector semi-Lagrangian equation defining the velocity of the parcel

$$\frac{d\mathbf{r}}{dt} = \mathbf{V}$$

The cedntered discretization of this equation in a three-time-level scheme is

$$\frac{\mathbf{r}^{\mathrm{t}+\Delta\mathrm{t}} - \mathbf{r}^{t-\Delta t}}{2\Delta t} = \mathbf{V}^t$$

where $\mathbf{r}^{t+\Delta t}$ is the arrival position vector of point G (the grid-point where the parcel arrives at time $t + \Delta t$), $\mathbf{r}^{t-\Delta t}$ is the position vector of the departure point O (where the parcel was at time $t - \Delta t$) and $\mathbf{V}^t$ is the velocity vector at the present time t at the middle of the trajectory. In plane geometry the trajectory is asumed to be a straight line (velocity constant during the interval $t - \Delta t \to t + \Delta t$). Now, the position of the middle of the trajectory depends on the position of the departure point, which is what we try to determine with this equation, therefore the equation is an implicit equation and has to be solved by an iterative method, depicted in Fig. 12

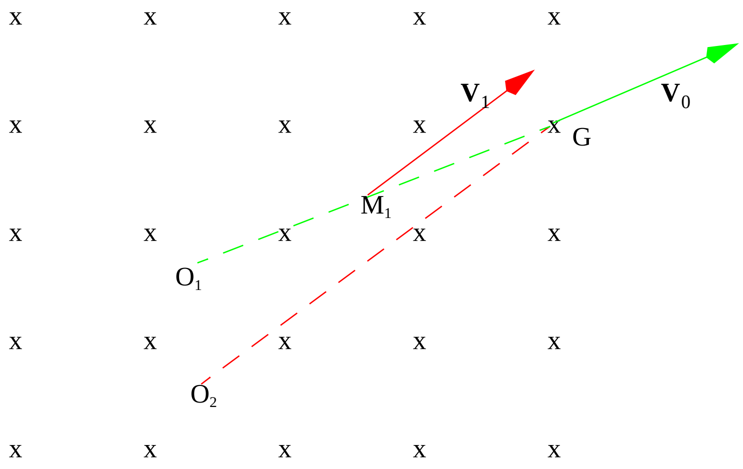


Figure 12: Iterative trajectory computation

In the first iteration, we take the velocity $\mathbf{V}_0$ at the arrival point G. Using this velocity we go backwards a distance $2\mathbf{V}_0\Delta t$ to reach point O$_1$, this is the first guess of our departure point. Then we take the point M$_1$ midway between points G and O$_1$ and interpolate the velocity at the present time to that point. Using that velocity $\mathbf{V}_1$ we go back from G a distance $2\mathbf{V}_1\Delta t$ to point O$_2$ and repeat the procedure until it converges. At ECMWF only three iterations are done and no test of convergence is performed.

In spherical geometry the trajectory is asumed to be an arc of a great circle instead of a straight line, which complicates somewhat the computations but the idea remains the same. Also in spherical geometry one has to take into account that the interpolated wind components refer to a local frame of reference pointing to the local North and East and, in order to use the interpolated values at grid point G, they have to be “rotated”.

In order to have an idea about the convergence of the iterative procedure just described, let us apply this procedure to the computation of the semi-Lagrangian trajectory in one dimension. For the shake of simplicity we will consider a two-time-level scheme and use the velocities only at the departure point instead of interpolating them at the middle of the trajectory. At the n’th iteration the departure point $\mathbf{r}_{n+1}$ is computed as

$$\mathbf{r}_{n+1} = G - \Delta t V_n$$

Now assume that V varies linearly between grid-points

$$V = a + b\mathbf{r} \Rightarrow b = \frac{\partial V}{\partial \mathbf{r}}$$

then

$$\mathbf{r}_{n+1} = G - a\Delta t - b\mathbf{r}_n \Delta t$$

For the iterative procedure to converge, this equation must have a solution of the form

$$\mathbf{r}_n = \lambda^n + K; \qquad (|\lambda| < 1)$$

Substituting we get

$$K = \frac{G - a\Delta t}{1 + b\Delta t}$$
$$\lambda = -b\Delta t$$

therefore for convergence we must have

$$\Delta t < \frac{1}{|b|}$$

This condition means that the parcels do not overtake eachother during the interval Δt and is much less restrictive than the CFL stability limit. Also it does not depend on the mesh size.

5.8 Two-time-level schemes

A centered discretization (second order accurate in space and time) of the general semi-Lagrangian equation

$$\frac{d\mathbf{X}}{dt} = \mathbf{R}$$

using only two time levels is

$$\frac{\mathbf{X}_A^{t+\Delta t} - \mathbf{X}_D^t}{\Delta t} = \mathbf{R}_M^{t+\frac{\Delta t}{2}}$$

where **R** has to be extrapolated in time before being interpolated to the middle point of the trajectory

$$\mathbf{R}^{t+\frac{\Delta t}{2}} \approx \frac{3}{2}\mathbf{R}^t - \frac{1}{2}\mathbf{R}^{t-\Delta t}$$

An alternative second-order accurate scheme can be developed from a Taylor series expansion in the semi-Lagrangian sense arround the departure point of the trajectory

$$\mathbf{X}_A^{t+\Delta t} = \mathbf{X}_D^t + \Delta t\left(\frac{d\mathbf{X}}{dt}\right)_D^t + \frac{(\Delta t)^2}{2}\left(\frac{d^2\mathbf{X}}{dt^2}\right)_{AV}$$

Notice that the time level and the position in the trajectory are consistent as requested in the Lagrangian point of view. Here subindex AV means some average value along the trajectory.

In the case of the computation of the trajectory, **X** is the position vector of the parcel of air and this equation is the equation of a uniformly accelerated movement with initial velocity $(d\mathbf{X}/dt)_D^t$ and acceleration $(d^2\mathbf{X}/dt^2)_{AV}$. The trajectory can not any more be considered as a straight line in this case and the middle point of the trajectory is not half way between the arrival and the departure points.

Now, substituting $(d\mathbf{X}/dt)_D^t$ by $\mathbf{R}_D^t$ and $(d^2\boldsymbol{X}/dt^2)_{AV}$ by $(d\mathbf{R}/dt)_{AV}$ we get

$$\mathbf{X}_A^{t+\Delta t} = \mathbf{X}_D^t + \Delta t\mathbf{R}_D^t + \frac{(\Delta t)^2}{2}\left(\frac{d\mathbf{R}}{dt}\right)_{AV} \qquad (65)$$

and $(d\mathbf{R}/dt)_{AV}$ needs to be evaluated. This is done at ECMWF as

$$\left(\frac{d\mathbf{R}}{dt}\right)_{AV} \approx \frac{\mathbf{R}_A^t - \mathbf{R}_D^{t-\Delta t}}{\Delta t}$$

which is not strictly compatible with the Lagrangian point of view because it uses values at time t at the arrival point of the trajectory and values at time $t - \Delta t$ at the departure point of the present trajectory which runs between times t and $t + \Delta t$. It is therefore only an approximation.

With this choice, Eq. (65) becomes

$$\mathbf{X}_A^{t+\Delta t} = \mathbf{X}_D^t + \frac{\Delta t}{2}(\mathbf{R}_A^t + \{2\mathbf{R}^t - \mathbf{R}^{t-\Delta t}\}_D)$$

and the computation of the trajectory

$$\mathbf{r}_A^{t+\Delta t} = \mathbf{r}_D^t + \frac{\Delta t}{2}(\mathbf{V}_A^t + \{2\mathbf{V}^t - \mathbf{V}^{t-\Delta t}\}_D)$$

6. THE SPECTRAL METHOD

6.1 Introduction

When using finite difference techniques for evolutionary problems, we only consider grid-point values of the dependent variables; no assumption is made about how the variables behave between grid points. An alternative approach is to expand the dependent variables in terms of a finite series of smooth orthogonal functions. The problem is then reduced to solving a set of ordinary differential equations which determine the behaviour in time of the expansion coefficients.

As an example consider the linear one-dimensional evolutionary problem

$$\frac{\partial\varphi}{\partial t} = \mathrm{H}(\varphi) \tag{66}$$

where H is a linear differential operator. Expanding φ in terns of a set of orthogonal functions $e_m(x)$, $m = m_1, \ldots m_2$ we have

$$\varphi = \sum_m \varphi_m(t) e_m(x) \tag{67}$$

The φ_m are the expansion coefficients whose behaviour we want to determine. We now use the procedure outlined earlier in Subsection 1.4—that is we minimise the integral of the square of the residual caused by using the approximate solution (67) in the original equation (66) (alternatively we could use the Galerkin method with the expansion functions as test functions). Since the expansion functions are orthonormal we have

$$\int_0^L e_l^* e_m \mathrm{d}x = \begin{cases} 1 & l = m \\ 0 & l \neq m \end{cases}$$

where e_m^* is the complex conjugate of e_m. Using this condition we get

$$\frac{\mathrm{d}\varphi_m}{\mathrm{d}t} = \sum_l \varphi_l \int_0^L e_m^* \mathrm{H}(e_l)\mathrm{d}x \qquad \text{for all } m \tag{68}$$

That is, we have a set of ordinary differential equations for the rate of change with time of the expansion coefficients.

It is now interesting to consider how our choice of expansion functions can greatly simplify the problem

(a) If the expansion functions are eigenfunctions of H we have $\mathrm{H}(e_l) = \lambda_l e_l$, where the λ_l are the eigenvalues; (68) then becomes

$$\frac{\mathrm{d}\varphi_l}{\mathrm{d}t} = \lambda_l \varphi_l$$

and the equations have become decoupled.

(b) If the original equation is

$$\mathrm{L}\left(\frac{\partial\varphi}{\partial t}\right) = \mathrm{H}(\varphi)$$

where L is a linear operator, then our problem is simplified by using expansion functions that are eigenfunctions of L with eigenvalues λ_m; we then have

$$\lambda_m \frac{\mathrm{d}\varphi_m}{\mathrm{d}t} = \sum_l \varphi_l \int_0^L e_m^* \mathrm{H}(e_l)\mathrm{d}x$$

6.2 The one-dimensional linear advection equation

It is convenient to write the advection equation in terms of the longitude $\lambda = 2\pi x / L$ and the angular velocity $\gamma = 2\pi u_0 / L$.

$$\frac{\partial\omega}{\partial t} + \gamma\frac{\partial\omega}{\partial\lambda} = 0 \qquad \omega = \frac{2\pi}{L}x \tag{69}$$

with boundary conditions: $\omega(\lambda, t) = \omega(\lambda + 2\pi p, t)$ for integer p, and initial conditions: $\omega(\lambda, 0) = f(\lambda)$. For any reasonable function $f(\lambda)$ the analytical solution to (69) is $\omega(\lambda, t) = f(\lambda - \gamma t)$

If we are going to use the approach outlined in Subsection 6.1, we must choose suitable expansion functions. The obvious choice is the finite Fourier series

$$\omega(\lambda, t) \approx \sum_{m = -M}^{M} \omega_m(t)\exp[\mathrm{i}m\lambda] \tag{70}$$

because the expansion functions are then eigenfunctions of the differential space operator. Here M is the maximum wave number and the ω_m are the complex expansion coefficients. Since $\omega_{-m}(t) = \omega^*_m(t)$ we need only be concerned with ω_m for $0 \le m \le M$, rather than the full set of expansion coefficients.

We should now use the Galerkin method, but for this simple problem it is sufficient to substitute (70) in (69) and equate coefficients of the expansion functions. This yields (as does the formal Galerkin method)

$$\frac{\mathrm{d}\omega_m}{\mathrm{d}t} + \mathrm{i}m\gamma\omega_m = 0 \qquad 0 \le m \le M \tag{71}$$

giving $2M + 1$ equations for the ω_m 's. For this particular case (71) can be integrated exactly to give

$$\omega_m(t) = \omega_m(0)\exp[\mathrm{i}m\gamma t] \tag{72}$$

If $f(\lambda)$ is also represented by a truncated Fourier series the complete solution is

$$\omega(\lambda, t) = \sum_m a_m \exp[\mathrm{i}m(\lambda - \gamma t)] \qquad \text{where } f(\lambda) = \sum_m a_m \exp[\mathrm{i}m\lambda]$$

which is the same as the exact solution. There is no dispersion due to the space discretization, unlike in the finite differences method. This fact is due to the space derivatives being computed analytically while they were approximated in the finite difference method.

The expression (72) can be represented graphically as a vector in the complex plane rotating anticlockwise with a constant angular velocity $m\gamma/2\pi$.

Scalarly multiplying Eq. (70) by each of the basis functions and using the orthogonality property of the Fourier basis we get at the initial time

$$\omega_m(0) = A_m \int_0^{2\pi} \omega(\lambda, 0)\exp[-im\lambda]d\lambda \tag{73}$$

where A_m are the normalization factors (which is known as the direct Fourier transform).

At any future time we can apply Eq. (70) to get the space distribution of the solution. This is normally known as inverse Fourier transform.

In the practice the initial conditions can be given in the form of grid-point data ($N+1$ points with spacing Δx say). Therefore, we think of the truncated Fourier series as representing an interpolating function which exactly fits the values of ω at the $N+1$ grid points. Eq. (73) then has to be computed as a discrete sum

$$\omega_m(0) = A'_m \sum_{i=1}^{K} \omega(\lambda_i)\exp[-im\lambda_i] \tag{74}$$

which is known as discrete direct Fourier transform. The corresponding discrete inverse Fourier transform is

$$\omega(\lambda_i, 0) = \sum_{m=-M}^{M} \omega_m(0)\exp[im\lambda_i] \tag{75}$$

Both of them can be computed with the Fast Fourier Transform (FFT) algorithm. It can be shown that, starting from the set of $\omega_m(0)$, going to the set $\omega(\lambda_i, 0);$ i=1, ...,K and returning to $\omega_m(0)$ we recover exactly the original values (the transforms are exact) as long as $K \ge 2M+1$ and the points are equally spaced in λ. This distribution of points with $K = 2M+1$ is known as the linear grid. On the other hand it can be shown also that the product of two functions can be computed without aliassing by the transform method of transforming both functions to grid-point space, multiplying together the functions at each grid-point and transforming back the product to Fourier space, as long as $K \ge 3M+1$. The distribution of points for which $K = 3M+1$ is known as the quadratic grid.

Having derived the initial conditions in terms of the spectral coefficients we must now integrate the ordinary differential equations for the expansion coefficients at some future time. Normally this has to be done using a time-stepping procedure such as the leapfrog scheme, i.e.

$$\frac{\mathrm{d}\omega_m}{\mathrm{d}t} = F_m \quad \text{becomes} \quad \omega_m^{n+1} = \omega_m^{n-1} + 2\Delta t F_m^n$$

This scheme is stable provided $|m\gamma\Delta t| \le 1$ for all m; but since the maximum value of m is M we require $|M\gamma\Delta t| \le 1$. In terms of the original grid, $L = 2M\Delta x$ giving $\gamma = \pi u_0 / M\Delta x$—hence there is stability provided $|\alpha| \le 1/\pi$. This shows that the stability criterion is more restrictive than for conventional explicit finite difference schemes. However, the spectral scheme has the great advantage that it has only very small phase errors which are not significant even for two gridlength waves.

Table 1 shows how D and r vary with l when $\alpha = 0.5 \times (1/\pi)$. The results of using the spectral method on the test problems described in Subsection 2.6 are given in Figs. 5 and 6. Note the impressive characteristics and results of the spectral model.

If we start the spectral method from a grid-point distribution and use the value of M which corresponds to the quadratic grid, Eq. (74) gives us a number of degrees of freedom smaller than the original number of degrees of freedom and therefore upon return to grid-point space by means of Eq. (75) we may not recover the original information. The resulting “fitted” function displays what is known as spectral ripples. This does not happend with the linear grid in which the number of degrees of freedom in Fourier space is the same as the number of degrees of freedom in grid-point space. To illustrate this point Fig. 13 shows a function composed of several abrupt steps and the result of transforming it to Fourier space and back to grid-point space using a spectral truncation for which the grid-point distribution corresponds either to the linear or the quadratic grid for that spectral truncation.

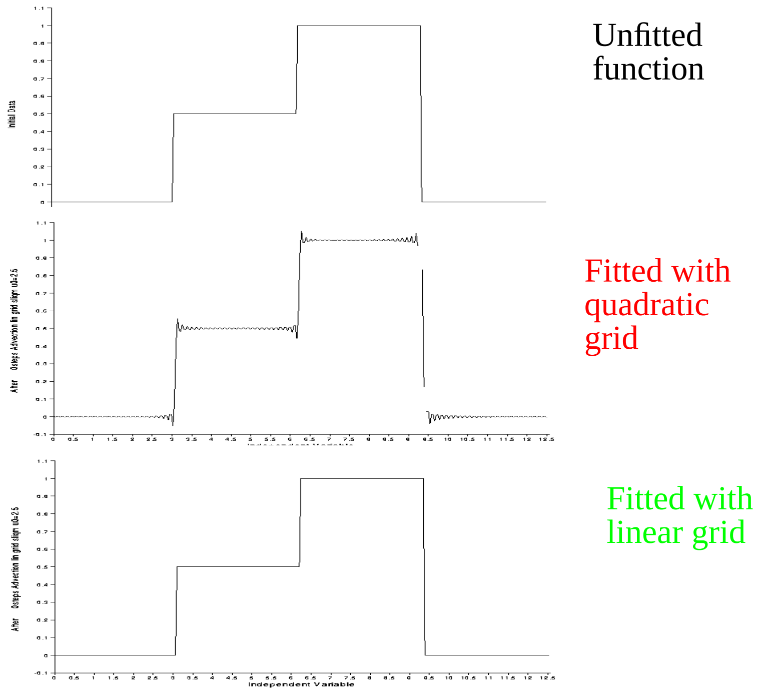


Figure 13: Step functions spectrally fitted using the quadratic and the linear grids

6.3 The non-linear advection equation

$$\frac{\partial\omega}{\partial t} = -\omega\frac{\partial\omega}{\partial\lambda} \tag{76}$$

If we again use the truncated Fourier series (70), the right-hand side of (76) becomes

$$F = \sum_{m=-2M}^{2M} F_m \exp[\mathrm{i}m\lambda] \quad \text{where} \quad F_m = -\mathrm{i}\sum_{m'=m-M}^{M} (m-m')\omega_{m'}\omega_{m-m'} \qquad \text{for} \quad m \geq 0$$

Similarly the left-hand side of (76) is written as

$$\frac{\partial\omega}{\partial t} = \sum_{m=-M}^{M} \frac{\mathrm{d}\omega_m}{\mathrm{d}t}\exp[\mathrm{i}m\lambda]$$

Since the series on either side of (76) are truncated at different wave numbers, there will always be a residual R. Using the Galerkin method (the least squares gives the same result) we now choose the time derivaties subject to the condition

$$\int_0^{2\pi} R\exp[-\mathrm{i}m\lambda]\mathrm{d}\lambda = 0 \quad \text{for all} \quad m$$

It can be shown that this yields

$$\frac{\mathrm{d}\omega_m}{\mathrm{d}t} = F_m \qquad -M \le m \le M \tag{77}$$

Thus, the Fourier components F_m with wave numbers larger than M are simply neglected. This means that there is no aliasing of small-scale components outside the original truncation and, hence, no non-linear instability.

In practice there are two approaches to the problem of calculating non-linear terms in the context of the spectral method—using interaction coefficients or the transform method

(a) Interaction coefficients.
An alternative way of expressing (77) is

$$\frac{\mathrm{d}\omega_m}{\mathrm{d}t} = -\sum_k\sum_l \mathrm{i}l\omega_k\omega_l\int \exp[\mathrm{i}k\lambda]\exp[\mathrm{i}l\lambda]\exp[-\mathrm{i}m\lambda]\mathrm{d}\lambda$$

$$= -\sum_k\sum_l \mathrm{i}\omega_k\omega_l I_{k,l,m} \quad \text{where} \quad I_{k,l,m} = \int l\exp[\mathrm{i}k\lambda]\exp[\mathrm{i}l\lambda]\exp[-\mathrm{i}m\lambda]\mathrm{d}\lambda$$

where the $I_{k,l,m}$ are the interaction coefficients. If there are only a small number of possible waves, then it is possible to calculate and store the interaction coefficients. However, for most problems this is not possible and so the transform method is used for calculating the non-linear terns.

(b) Transform method.
Using Fast Fourier Transforms (FFTs) it is easy to move from the spectral representation (spectral space) to a grid-point representation (physical space). Therefore, the essence of the transform method is to calculate derivatives in spectral space, but to transform to physical space using FFTs whenever a product is required. Once all the products have been computed at grid points, the spectral coefficients of this product field are calculated—that is we use FFTs to return to spectral space. Now, consider how we apply this to the non-linear advection equation.
Given the ω_m we want to compute the spectral coefficients of the non-linear term $-\omega\frac{\partial\omega}{\partial\lambda}$ (i.e. the F_m on the right-hand side of (77)). The following three steps are required to do this:
(i) Calculate ω and $D = \frac{\partial\omega}{\partial\lambda}$ at grid points λ_j by using the spectral coefficients

$$\omega(\lambda_j) = \sum_m \omega_m\exp[\mathrm{i}m\lambda_j] \qquad D(\lambda_j) = \sum_m \mathrm{i}m\omega_m\exp[\mathrm{i}m\lambda_j]$$

(ii) Calculate the advection term at each grid point in physical space

$$F(\lambda_j) = -\omega(\lambda_j)D(\lambda_j)$$

(iii) Return to spectral space by calculating the Fourier coefficients

$$F_m = \frac{1}{2\pi}\sum_j F(\lambda_j)\exp[-\mathrm{i}m\lambda_j]$$

In practice this procedure has to be employed to calculate the spectral coefficients of the non-linear term at every time level. As the product of the two functions is computed in grid-point space and not in spectral space, we get aliassing unless the number of grid-points corresponds to the quadratic grid. Even so, products of more than two functions will still have aliassing.

6.4 The one-dimensional gravity wave equations

Since these equations are linear, they can be dealt with in the same way as the linear advection equation described in Subsection 6.2. Writing the gravity wave equations in terms of the longitude $\lambda = 2\pi x / L$ gives

$$\frac{\partial \omega}{\partial t} + g\frac{\partial h}{\partial \lambda} = 0 \qquad \frac{\partial h}{\partial t} + H\frac{\partial \omega}{\partial \lambda} = 0$$

where ω is the angular velocity $2\pi u / L$. Using

$$\omega(\lambda, t) = \sum_{m=-M}^{M} \omega_m(t)\exp[\mathrm{i}m\lambda] \qquad h(\lambda, t) = \sum_{m=-M}^{M} h_m(t)\exp[\mathrm{i}m\lambda]$$

it is found that the Galerkin procedure gives

$$\frac{\mathrm{d}\omega_m}{\mathrm{d}t} + \mathrm{i}mgh_m = 0 \qquad \frac{\mathrm{d}h_m}{\mathrm{d}t} + \mathrm{i}mH\omega_m = 0$$

Therefore, with centred time differences, the time stepping algorithms for the Fourier coefficients are

$$\omega_m^{n+1} = \omega_m^{n-1} - 2\mathrm{i}m\Delta t g h_m^n$$
$$h_m^{n+1} = h_m^{n-1} - 2\mathrm{i}m\Delta t H \omega_m^n$$

Therefore, since our original equations were linear, the complete integration can be carried out in spectral space.

6.5 Stability of various time stepping schemes

6.5 (a) The forward time scheme.

(i) Linear advection equation

$$\frac{\varphi_m^{n+1} - \varphi_m^n}{\Delta t} = -U_0 \mathrm{i}m\varphi_m^n$$

Using von Neumann we find

$$\lambda = 1 - \mathrm{i}mU_0\Delta t$$

similar to the FTCS scheme and always unstable as $|\lambda| > 1$.

(ii) Gravity wave equations

$$\frac{u_m^{n+1}-u_m^n}{\Delta t} = -gimh_m^n$$

$$\frac{h_m^{n+1}-h_m^n}{\Delta t} = -Himu_m^n$$

$\lambda = 1 \pm \mathrm{i}\sqrt{gH}m\Delta t \quad \rightarrow \quad |\lambda| > 1$: always unstable.

6.5 (b) The leapfrog time scheme.

(i) Linear advection equation

$$\frac{\varphi_m^{n+1}-\varphi_m^{n-1}}{2\Delta t} = -U_0 im\varphi_m^n$$

$\lambda = \pm\sqrt{1-U_0^2m^2(\Delta t)^2} - \mathrm{i}U_0m\Delta t : |\lambda| = 1$ if $U_0m\Delta t \le 1$, but $|\lambda| > 1$ otherwise. Therefore the scheme is conditionally stable and neutral, but the stability criterion is more restrictive than using finite differences as already stated in Subsection 6.2.

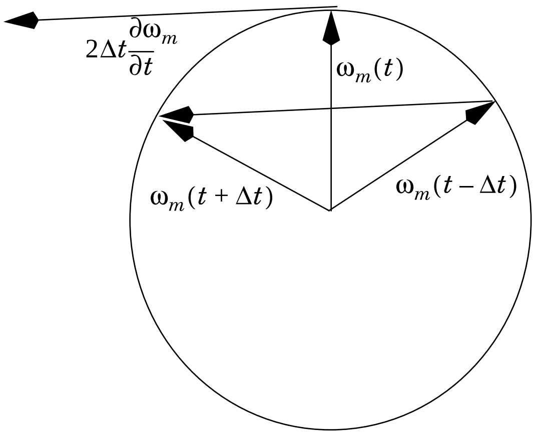
(ii) Gravity-wave equations

$$\frac{u_m^{n+1}-u_m^{n-1}}{2\Delta t} = -gimh_m^n$$

$$\frac{h_m^{n+1}-h_m^{n-1}}{2\Delta t} = -Himu_m^n$$

$\lambda = \pm\sqrt{1-gHm^2(\Delta t)^2} - \mathrm{i}\sqrt{gH}m\Delta t : |\lambda| = 1$ if $\sqrt{gH}m\Delta t \le 1$, but $|\lambda| > 1$ otherwise, and the stability condition for the scheme to be neutral is more restrictive than in finite differences.

The leapfrog scheme can be represented graphically as follows:



from which it is clear that if Δt is too large $\omega_m(t+\Delta t)$ can not stay in the circle and therefore its modulus will increase unlike in the analytical solution.

6.5 (c) Implicit centred scheme.

(i) Linear advection equation

$$\frac{\varphi_m^{n+1}-\varphi_m^{n-1}}{2\Delta t} = -\mathrm{i}m\frac{U_0}{2}(\varphi_m^{n+1}+\varphi_m^{n-1})$$

$$\lambda^2 = (1-\mathrm{i}mU_0\Delta t)/(1+\mathrm{i}mU_0\Delta t) \quad \rightarrow \quad |\lambda| = 1 : \text{always neutral}$$

(ii) Gravity wave equations

$$\frac{u_m^{n+1}-u_m^{n-1}}{2\Delta t} = -\mathrm{i}m\frac{g}{2}(h_m^{n+1}+h_m^{n-1})$$

$$\frac{h_m^{n+1}-h_m^{n-1}}{2\Delta t} = -\mathrm{i}m\frac{H}{2}(u_m^{n+1}+u_m^{n-1})$$

$$\lambda^2 = (1-\mathrm{i}m\Delta t\sqrt{gH})/(1+\mathrm{i}m\Delta t\sqrt{gH}) \quad \rightarrow \quad |\lambda| = 1 : \text{always neutral.}$$

6.5 (d) Shallow water equations.

(i) Explicit scheme

Non-linear equations	Linearized version
$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}-fv+\frac{\partial\varphi}{\partial x} = 0$	$\frac{\partial u'}{\partial t}+U_0\frac{\partial u'}{\partial x}+V_0\frac{\partial u'}{\partial y}-f_0v'+\frac{\partial\varphi'}{\partial x} = 0$
$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}+fu+\frac{\partial\varphi}{\partial y} = 0$	$\frac{\partial v'}{\partial t}+U_0\frac{\partial v'}{\partial x}+V_0\frac{\partial v'}{\partial y}+f_0u'+\frac{\partial\varphi'}{\partial x} = 0$
$\frac{\partial\varphi}{\partial t}+u\frac{\partial\varphi}{\partial x}+v\frac{\partial\varphi}{\partial y}+\varphi\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right) = 0$	$\frac{\partial\varphi'}{\partial t}+U_0\frac{\partial\varphi'}{\partial x}+V_0\frac{\partial\varphi'}{\partial y}+\Phi_0\left(\frac{\partial u'}{\partial x}+\frac{\partial v'}{\partial y}\right) = 0$

Asume a solution of the form

$$\begin{aligned} u' &= u_0\exp[\mathrm{i}(\alpha t+mx+ny)] \\ v' &= v_0\exp[\mathrm{i}(\alpha t+mx+ny)] \\ \varphi' &= \varphi_0\exp[\mathrm{i}(\alpha t+mx+ny)] \end{aligned}$$

Substituting we get

$$\begin{aligned} u_0\frac{\exp[\mathrm{i}\alpha\Delta t]-\exp[-\mathrm{i}\alpha\Delta t]}{2\Delta t}+\mathrm{i}mU_0u_0+\mathrm{i}nV_0u_0-f_0v_0+\mathrm{i}m\varphi_0 &= 0 \\ v_0\frac{\exp[\mathrm{i}\alpha\Delta t]-\exp[-\mathrm{i}\alpha\Delta t]}{2\Delta t}+\mathrm{i}mU_0v_0+\mathrm{i}nV_0v_0+f_0u_0+\mathrm{i}n\varphi_0 &= 0 \\ \varphi_0\frac{\exp[\mathrm{i}\alpha\Delta t]-\exp[-\mathrm{i}\alpha\Delta t]}{2\Delta t}+\mathrm{i}mU_0\varphi_0+\mathrm{i}nV_0\varphi_0+\mathrm{i}\Phi_0(mu_0+nv_0) &= 0 \end{aligned}$$

$$
\begin{aligned}
u_0\frac{1}{\Delta t}\sin(\alpha\Delta t) + mU_0u_0 + nV_0u_0 + \mathrm{i}f_0v_0 + m\varphi_0 &= 0\\
v_0\frac{1}{\Delta t}\sin(\alpha\Delta t) + mU_0v_0 + nV_0v_0 - \mathrm{i}f_0u_0 + n\varphi_0 &= 0\\
\varphi_0\frac{1}{\Delta t}\sin(\alpha\Delta t) + mU_0\varphi_0 + nV_0\varphi_0 + \Phi_0(mu_0 + nv_0) &= 0
\end{aligned}
$$

i.e.

$$\frac{1}{\Delta t}\sin(\alpha\Delta t)\mathbf{Z} + (mU_0 + nV_0)\mathbf{Z} + \mathbf{HZ} = 0$$

where

$$\mathbf{Z} = (u_0, v_0, \varphi_0) \quad \text{and} \quad \mathbf{H} = \begin{bmatrix} 0 & \mathbf{i}f_0 & m \\ -\mathbf{i}f_0 & 0 & n \\ \Phi_0 m & \Phi_0 n & 0 \end{bmatrix}$$

Projecting $\mathbf{Z}$ on the eigenvectors $\mathbf{X}$ of matrix $\mathbf{H}$, for which

$$\mathbf{HX} = \lambda\mathbf{X} \quad \Rightarrow \quad (\mathbf{H} - \mathbf{I}\lambda)\mathbf{X} = 0 \quad \Rightarrow \quad \lambda^3 - \Phi_0\gamma m^2 - \Phi_0\lambda n^2 - \lambda f_0^2 = 0$$

i.e.

$$
\begin{aligned}
\lambda_1 &= 0\\
\lambda^2 - \Phi_0(m^2 + n^2) - f_0^2 = 0 \quad &\Rightarrow \quad \lambda_{2,3} = \pm\sqrt{f_0^2 + \Phi_0(m^2 + n^2)}
\end{aligned}
$$

We obtain three vector equations

$$
\begin{aligned}
&\frac{1}{\Delta t}\sin(\alpha\Delta t)\mathbf{Y} + (U_0m + V_0n)\mathbf{Y} + \lambda_i\mathbf{Y} = 0\\
&\Rightarrow \quad |\sin(\alpha\Delta t)| = |-\Delta t(U_0m + V_0n + \lambda_i)| \le 1
\end{aligned}
$$

the most restrictive of the three is when $\lambda_i = +\sqrt{f_0^2 + \Phi_0(m^2 + n^2)}$
which gives the stability condition that

$$\Delta t \le \frac{1}{U_0M + V_0N + \sqrt{f_0^2 + \Phi_0(m^2 + n^2)}} \qquad M = \max(m) \quad N = \max(n)$$

The values for the atmosphere of these quantities are $\Phi_0 \approx 9 \cdot 10^4 \mathrm{m^2/s^2}$; $U_0 \approx 20\mathrm{m/s}$; $f_0 \approx 10^{-4}\mathrm{s^{-1}}$. For a model representing waves down to a wavelength of ~380 km, $M = N \sim 2.65 \times 10^{-6}$ $\mathrm{m^{-1}}$ which gives for Δt a value of ~4 min

(ii) Semi-implicit scheme

$$u_0\frac{\exp[\mathrm{i}\alpha\Delta t]-\exp[-\mathrm{i}\Delta t]}{2\Delta t}+\mathrm{i}mU_0u_0+\mathrm{i}nV_0u_0-f_0v_0+\mathrm{i}m\varphi_0\frac{\exp([\mathrm{i}\alpha\Delta t]+\exp[-\alpha\Delta t])}{2\Delta t} = 0$$
$$v_0\frac{\exp[\mathrm{i}\alpha\Delta t]-\exp[-\mathrm{i}\Delta t]}{2\Delta t}+\mathrm{i}mU_0v_0+\mathrm{i}nV_0v_0+f_0u_0+\mathrm{i}n\varphi_0\frac{\exp([\mathrm{i}\alpha\Delta t]+\exp[-\alpha\Delta t])}{2\Delta t} = 0$$
$$\varphi_0\frac{\exp[\mathrm{i}\alpha\Delta t]-\exp[-\mathrm{i}\Delta t]}{2\Delta t}+\mathrm{i}mU_0\varphi_0+\mathrm{i}nV_0\varphi_0+\mathrm{i}\Phi_0(mu_0+nv_0)\frac{\exp([\mathrm{i}\alpha\Delta t]+\exp[-\alpha\Delta t])}{2\Delta t} = 0$$

$$\frac{\mathrm{i}u_0}{\Delta t}\sin(\alpha\Delta t)+\mathrm{i}mU_0u_0+\mathrm{i}nV_0u_0-f_0v_0+\mathrm{i}m\varphi_0\cos(\alpha\Delta t) = 0$$
$$\frac{\mathrm{i}v_0}{\Delta t}\sin(\alpha\Delta t)+\mathrm{i}mU_0v_0+\mathrm{i}nV_0v_0+f_0v_0+\mathrm{i}n\varphi_0\cos(\alpha\Delta t) = 0$$
$$\frac{\mathrm{i}\varphi_0}{\Delta t}\sin(\alpha\Delta t)+\mathrm{i}mU_0\varphi_0+\mathrm{i}nV_0\varphi_0+\mathrm{i}\Phi_0(mu_0+nv_0)\cos(\alpha\Delta t) = 0$$

i.e.

$$\frac{\sin(\alpha\Delta t)}{\Delta t}\mathbf{Z}+(U_0m+V_0n)\mathbf{Z}+\mathbf{HZ} = 0$$

where

$$\mathbf{Z} = (u_0, v_0, \phi_0) \qquad \mathbf{H} = \begin{bmatrix} 0 & \mathrm{i}f_0 & m\cos(\alpha\Delta t) \\ -\mathrm{i}f_0 & 0 & n\cos(\alpha\Delta t) \\ m\Phi_0\cos(\alpha\Delta t) & n\Phi_0\cos(\alpha\Delta t) & 0 \end{bmatrix}$$

Continuing as above, the eigenvalues λ of $\mathbf{H}$ are given by

$$\lambda^3-\Phi_0\lambda(m^2+n^2)\cos^2(\alpha\Delta t)\ -\lambda f_0^2 = 0$$

i.e.

$$\lambda = 0$$
$$\lambda^2-\Phi_0(m^2+n^2)\cos^2(\alpha\Delta t)\ -f_0^2 = 0 \quad\Rightarrow\quad \lambda = \pm\sqrt{f_0^2+\Phi_0(m^2+n^2)\cos^2(\alpha\Delta t)}$$

Hence

$$\sin(\alpha\Delta t)+\Delta t\sqrt{f_0^2+\Phi_0(m^2+n^2)\cos^2(\alpha\Delta t)}\ +\Delta t(U_0m+V_0n) = 0$$

If $f_0 = 0$ this gives:

$$\sin(\alpha\Delta t)+\Delta t\cos(\alpha\Delta t)\ \sqrt{\Phi_0(m^2+n^2)} = -\Delta t(U_0m+V_0n)$$

The function on the left hand side has a maximum negative value when $\Delta t\sqrt{\Phi_0(m^2+n^2)}\le 1$, in which case there is a real solution for α

$$\Delta t(U_0 m + V_0 n) \le 1 \qquad \Rightarrow \qquad \Delta t \le \frac{1}{U_0 m + V_0 n}$$

If $\Delta t\sqrt{\Phi_0(m^2+n^2)} > 1$ the condition is less restrictive. The numerical phase speed is:

$$c_{\text{num}} = -\frac{\alpha}{(m^2+n^2)}$$

while the analytical one is given by the same formula, but with the frequency α_{anal} given by

$$\alpha_{\text{anal, 1}} = -(mU_0 + nV_0) \qquad \text{slow solution (or Rossby wave)}$$

$$\alpha_{\text{anal, 2.3}} = -(mU_0 + nV_0) \pm \sqrt{f_0^2 + \Phi_0(m^2+n^2)} \qquad \text{fast solution (inertia–gravity waves)}$$

We can therefore compute the dispersion error

$$r = \frac{c_{\text{num}}}{c_{\text{anal}}} = \frac{\alpha}{\alpha_{\text{anal}}}$$

6.6 The spherical harmonics

When using spherical geometry it is natural to expand any dependent variable φ in terms of a truncated series of spherical harmonics

$$\varphi(\lambda, \mu, t) = \sum_{m=-M}^{M} \left\{ \sum_{n=|m|}^{J} \varphi_n^m Y_n^m(\lambda, \mu) \right\} \tag{78}$$

where λ is the longitude and $\mu = \sin(\text{latitude})$. Again m is the zonal wave number, now n the total wavenumber and $n - |m|$ represents the effective meridional wave number. In (78) we can choose the truncation that we want.

(a) If $J = M$ the truncation is described as <u>triangular</u> (a model with this truncation and $m = 40$ is said to be a T40 model).

(b) For <u>rhomboidal</u> truncation $J = M + |m|$.

The reason for these descriptions becomes apparent when we plot a diagram of permissible values of n and m for fixed M; such diagrams for $M = 4$ are shown in Fig. 14 .

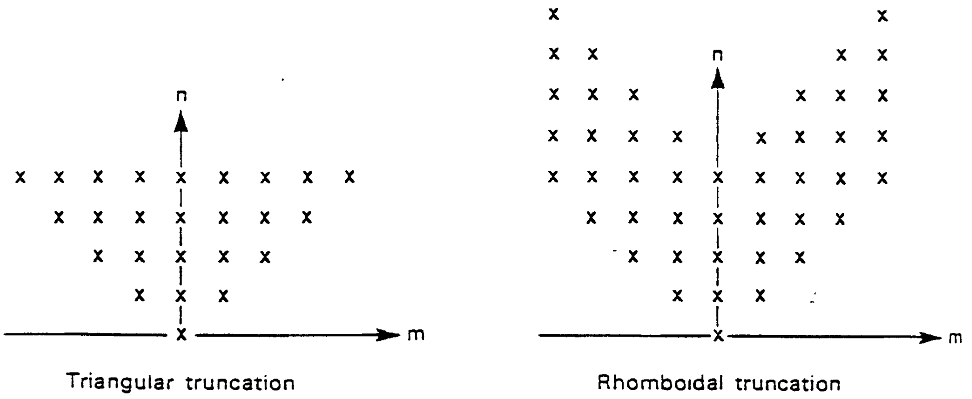


Figure 14. Permissible vales of m and n for triangular and rhomboidal truncation.

The spherical harmonics have the property that

$$\nabla^2 Y_n^m = -\frac{n(n+1)}{a^2} Y_n^m \tag{79}$$

where ∇^2 is the Laplacian in spherical coordinates and a is the earth's radius. Another property is that

$$Y_n^m(\lambda, \mu) = P_n^m(\mu)\exp[\mathrm{i}m\lambda]$$

where P_n^m is the associated Legendre polynomial of degree n and order m, which may be computed as

$$P_n^m(\mu)=\sqrt{(2n+1)\frac{(n\,\tilde{-}\,m)!}{(n+m)!}}\frac{1}{2^n n!}(1-\mu^2)^{\frac{m}{2}}\frac{d^{(n+m)}(\mu^2-1)^n}{d\mu^{n+m}}; \qquad \mathrm{m} \geq 0$$

$$P_n^{-m}(\mu)=P_n^m(\mu)$$

and are orthogonal:

$$\frac{1}{2}\int_{-1}^{1} P_n^m(\mu)P_s^m(\mu)d\mu = \delta_{n,s}$$

The space derivatives can be computed analytically as:

$$\frac{\partial}{\partial\lambda}Y_n^m = \mathrm{i}mY_n^m \qquad m \geq 0$$

and using the properties of the Legendre polynomial we have

$$(1-\mu^2)\frac{\partial}{\partial\mu}Y_n^m = -n\varepsilon_{n+1}^m Y_{n+1}^m + (n+1)\varepsilon_n^m Y_{n-1}^m \qquad m \geq 0$$

$$\varepsilon_n^m = \left(\frac{n^2-m^2}{4(n\ \ -1)}\right)^{1/2}$$

For $m > 0$ we use the fact that $Y_n^{-m} = (Y_n^m)^*$. With these relationships space derivatives can be calculated exactly leaving a set of ordinary differential equations for the time rate of change of the spherical harmonic coefficients φ_n^m.

Normally we have to deal with non-linear terms in which two spherical harmonics interact to produce a third. Unless the truncation is very severe the calculations are very time consuming. This problem can be overcome by the transform method introduced in section Subsection 6.3.

(a) Starting in spectral space, the spectral coefficients are used to calculate the dependent variables on a latitude–longitude grid (inverse spectral transform). For a regularly spaced longitude grid with at least $2M+1$ points and a specially chosen latitude grid (the Gaussian latitudes which are almost regularly spaced), the transformation can be done exactly.

(b) The non-linear dynamics and physical process terms of each prognostic equation are calculated in real space.

(c) The non-linear terms are transformed back to the spectral domain (direct spectral transform).

In order to perform the spectral transforms it is convenient to introduce the Fourier coefficients

$$\varphi_m(\mu, t) = \frac{1}{2\pi}\int_0^{2\pi}\varphi(\lambda, \mu, t)\exp[-im\lambda]d\lambda = \sum_{n=|m|}^{N}\varphi_n^m(t)P_n^m(\mu)$$

Scalarly multiplying Eq. (78) by each of the spherical harmonics and making use of the orthogonality properties of both the Fourier basis functions and the Legendre polynomials, we get

$$\varphi_n^m(t) = \frac{1}{4\pi}\int_0^1\int_0^{(2\pi)}\varphi(\lambda, \mu, t)P_n^m(\mu)\exp[-im\lambda]d\lambda d\mu$$

which is the direct spectral transform. This transform can be done by first performing the integral with respect to λ. This is a Fourier transform which will compute the Fourier coefficients. If the original function is given in a discrete set of longitude points, the transform is a discrete Fourier transform and, as discused earlier it is exact if the longitude points are equally spaced and its number is at least 2M+1.

The integral with respect to the latitude can be performed from the Fourier coefficients by means of a Gaussian quadrature formula and it can be shown that this integral is exact if the latitudes at which the input data are given are taken at the points where

$$P_{N_G}^0(\mu) = 0$$

(these are called the Gaussian latitudes) with $N_G \geq (2M+1)/2$. Furthermore products of two functions can be computed alias-free if the number of Gaussian latitudes is $N_G \geq (3M+1)/2$. The Gaussian latitudes are not equally spaced as the points to compute the discrete Fourier transforms but they are nearly so and therefore this spacing is approximately the same as the longitudinal spacing.

The distribution of points allowing exact transforms is called the linear Gaussian grid and it has at least (2M+1) longitude points equally spaced at each of at least (2M+1)/2 Gaussian latitude rows. Products of two functions can be computed alias-free if we use a quadratic Gaussian grid which is made of at least (3M+1) equally spaced points in each of at least (3M+1)/2 Gaussian latitudes.

The same distribution of grid-points of a Gaussian grid can represent a linear or a quadratic Gaussian grid depending on the spectral truncation used in conjunction with that grid. As an example, the quadratic grid corresponding to a spectral truncation of T213 coincides with the linear Gaussian grid corresponding to the spectral truncation T319.

Finally it should be noted that only true scalars should be represented by a series of spherical harmonics: hence when spectral methods are used, the primitive equations are put in their vorticity and divergence form, rather than in their momentum (u and v) form.

6.7 The reduced Gaussian grid

When using a regular Gaussian grid as described above, either a quadratic or a linear Gaussian grid, the number of longitude points per row of latitude is the same no matter how close we are to the pole. Therefore the geographycal distance between points of the same row decreases as we approach the poles and the resolution, which is nearly isotropic close to the equator becomes highly anisotropic close to the poles.

The triangular truncation in spectral space is isotropic because the shortest wavelength representable (wavenumber n=M) is independent of the wave direction (given by the value of the zonal wavenumber m). On the other hand the amplitude of the associated Legendre polynomials is very small when m is large and $|\mu|$ approaches 1. This suggest the possibility of ignoring some of the values of m in the Fourier transforms at Gaussian latitudes approaching the poles. The number of longitude points needed to represent properly the retained wavelengths is then smaller and the distance between points decreases less dramatically than with the regular (or full) grid, resulting in a more isotropic resolution.

The Gaussian grid resulting from these considerations is called the reduced Gaussian grid.

In spherical geometry, even using the reduced linear Gaussian grid, the number of degrees of freedom in grid-point space is larger than the number of degrees of freedom in spectral space and therefore, if we start with the representation of a field in grid-point space, go to spectral space and return to grid-point space, part of the degrees of freedom in the initial data are lost and spectral or “Gibbs” ripples appear as a consequence of the spectral fitting. Nevertheless, the problem is less noticeable when using the linear Gaussian grid than when using the quadratic Gaussian grid because in the former the ratio between the number of degrees of freedom in grid-point space and in spectral space is closer to 1 than in the latter.

6.8 Diffusion in spectral space

The linear diffusion equation in two dimensions for a variable A is

$$\frac{\partial A}{\partial t} = K\nabla^2 A; \qquad \mathrm{K} > 0$$

Transforming to spectral space and making use of the property of the spherical harmonics given by Eq. (79), we get

$$\frac{\partial A_n^m}{\partial t} = -K\frac{n(n+1)}{a^2}A_n^m$$

Applying the leapfrog time discretization we get two solutions, the physical solution which is unconditionally stable and a computational solution which is unconditionally unstable. If we apply a forward time-stepping scheme we get one solution which is conditionally stable. Finally if we apply a fully implicit (or backward) time-stepping scheme we get

$$\frac{A_n^m(t+\Delta t) - A_n^m(t)}{\Delta t} = -K\frac{n(n+1)}{a}A_n^m(t+\Delta t)$$

$$A_n^m(t+\Delta t) = \frac{A_n^m(t)}{1+\Delta t K n(n+1)/a^2}$$

which is a decoupled system of equations and the scheme is unconditionally stable.

There is no penalty for using an implicit time-stepping scheme because the basis functions are eigenfunctions of the equation operator. It is also straightforward to apply a superharmonic operator such as ∇^4, or even ∇^{2m} with any integer value of m. It sufices to substitute in the solution $n(n+1)/a^2$ by $(n(n+1)/a^2)^m$.

6.9 Advantages and disadvantages

6.9 (a) Advantages.

(i) Space derivatives calculated exactly.

(ii) Non-linear quadratic terns calculated without aliasing (if computed in spectral space or using the quadratic grid).

(iii) For a given accuracy fewer degrees of freedom are required than in a grid-point model.

(iv) Easy to construct semi-implicit schemes since spherical harmonics are eigenfunctions of the Helmholtz operator.

(v) On the sphere there is no pole problem.

(vi) Phase lag errors of mid-latitude synoptic disturbances are reduced.

(vii) The use of staggered grids is avoided.

6.9 (b) Disadvantages.

(i) The schemes appear complicated, though they are relatively easy to implement.

(ii) The calculation of the non-linear terms takes a long time unless the transform method is used.

(iii) Physical processes cannot be included unless the transform method is used.

(iv) As the horizontal resolution is refined, the number of arithmetic operations increases faster in spectral models than in grid-point models due to the Legendre transforms whose cost increases as N^3.

(v) Spherical harmonics are not suitable for limited-area models.

6.10 Further reading

The original version of this note is based mainly on a review article by Machenhauer on "The spectral method" which is Chapter 3 of GARP Publication Series No.17, Volume II. That article contains far more information than is in this note, except for the linear and the reduced Gaussian grids.

7. THE FINITE-ELEMENT TECHNIQUE

7.1 Introduction

As with the spectral method, the finite element technique approximates the field of a dependent variable by a finite series expansion in terms of linearly independent analytical functions. This means that the dependent variable is defined over the whole domain rather than just at discrete points as in the grid-point method. The difference between the spectral and finite-element techniques lies in the form of the expansion functions: for the spectral method these are global functions whereas for the finite elements they are only locally non zero (see Subsection 1.4).

There are two basic steps in the finite-element technique:

(a) expand the dependent variables in terms of a set of low-order polynomials (the basis functions) which are only locally non-zero.

(b) insert these expansions into the governing equations and orthogonalize the error with respect to some test functions.

As an example consider how we can represent a field φ in finite-element notation when we are given the values of φ at equally spaced points along the x -direction. Let the points be given by x_j (the nodes) and the values of the dependent variable by φ_j (the nodal value)—see Fig. 15 . Now suppose that φ varies linearly between the nodes—there is a piecewise linear fit. Therefore the behaviour of φ within an element (the region between the nodes) is determined by the nodal values. If we define a set of basis functions $e_j(x)$ given by the hat (chapeau) function (see Fig. 15), the field of φ can be represented by

$$\varphi = \sum_j \varphi_j e_j(x) \qquad (80)$$

An example of this is given in Fig. 15 . This approach is called collocation method.

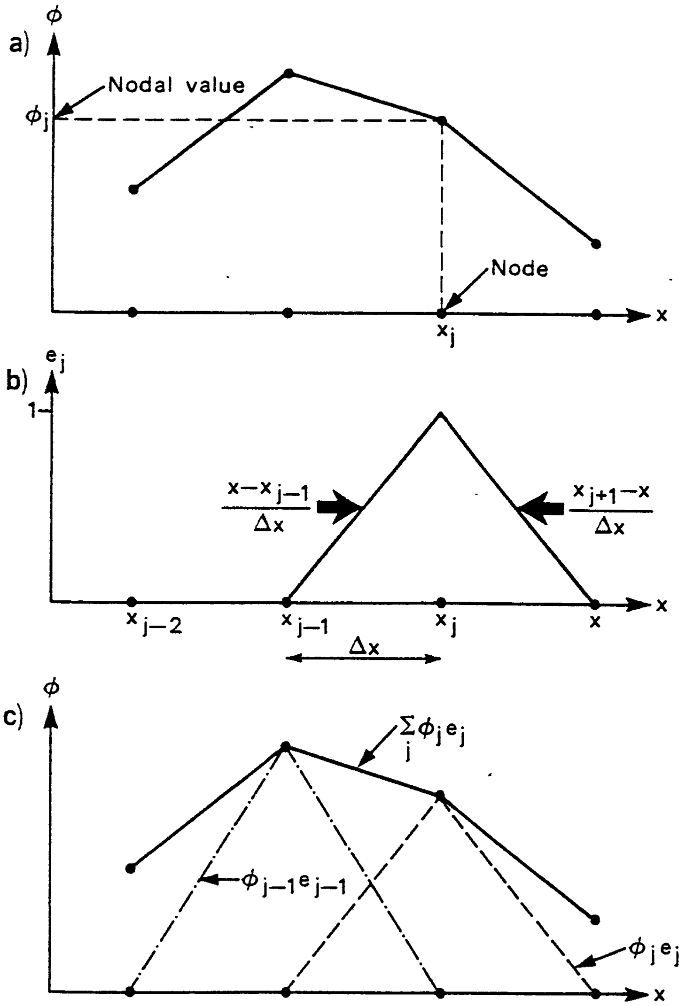


Figure 15. Illustrations of (a) linear piecewise fit, (b) linear basis functions and (c) of how a linear piecewise fit is made up of a linear combination of basis functions.

Another approach to the calculation of the expansion coefficients φ_j when we are given a continuous function is to minimize the distance between the continuous function φ and the discrete approximation $\sum \varphi_j e_j(x)$. In order to apply this approach, we need first to define a topology which is usually done by defining a scalar product (,) and the corresponding norm $\|\psi\|^2 = (\psi, \psi)$; therefore the space of functions is given the structure of a Hilbert space. It can be easily shown that this procedure gives the same result as the Galerkin approach of scalarly multiplying both sides of (80) by each of the basis functions $e_l(x)$

$$(\varphi, e_l(x)) = \sum_j \varphi_j(e_j(x), e_l(x)) \tag{81}$$

which is a system of simultaneous linear equations for the unknown coefficients φ_j.

7.2 Linear advection equation

7.2 (a) . Once again we consider the linear advection equation with periodic boundary conditions

$$\frac{\partial\varphi}{\partial t}+u_0\frac{\partial\varphi}{\partial x} = 0 \qquad \varphi(x+L,t) = \varphi(x,t) \qquad \varphi(x,0) = f(x)$$

Define a mesh of points $x_j = (j-1)\Delta x$, $j = 1, 2, \ldots N+1$ with $\Delta x = L/N$. We assume that the finite-element approximation to the exact solution has a piecewise linear representation using the x_j as the nodes.

$$\varphi(x,t) = \sum_{j=1}^{N+1}\varphi_j(t)e_j(x)$$

Substituting in the original equation gives

$$R = \sum_j\frac{\mathrm{d}\varphi_j}{\mathrm{d}t}e_j+u_0\sum_j\varphi_j\frac{\mathrm{d}e_j}{\mathrm{d}x} \tag{82}$$

where R is the residual. If we simply set $R = 0$ (point collocation) we have the problem that $\mathrm{d}e_j/\mathrm{d}x$ is not defined at the nodes. If this is overcome by making further approximations we end up with the usual centred difference approximation. However, the use of higher-order finite-element interpolation with point collocation does not lead to standard higher order difference schemes.

An alternative approach is to use the Galerkin method with the basis functions as the test functions (the least-squares method gives the same results). Therefore, we have

$$\int Re_i\mathrm{d}x = 0 \qquad i = 1, 2, \ldots N+1 \tag{83}$$

Substituting for R from (82) into (83) gives

$$\sum_j\frac{\mathrm{d}\varphi_j}{\mathrm{d}t}\int_0^L e_ie_j\mathrm{d}x+u_0\sum_j\varphi_j\int_0^L\frac{\mathrm{d}e_j}{\mathrm{d}x}e_i\mathrm{d}x = 0 \tag{84}$$

Since the basis functions are hat-functions, there are going to be many combinations of i and j for which the integrals are zero. In fact, for a given j, there will only be non-zero contributions for $i = j-1, j, j+1$ (that is $x_{j-1}\le x\le x_{j+1}$). It is easy to show that

$$\begin{aligned}
&\int e_{j+1}e_j\mathrm{d}x = \frac{1}{6}\Delta x &&\int e_j^2\mathrm{d}x = \frac{2}{3}\Delta x\\
&\int\frac{\mathrm{d}e_{j\pm1}}{\mathrm{d}x}e_j\mathrm{d}x = \pm\frac{1}{2} &&\int\frac{\mathrm{d}e_j}{\mathrm{d}x}e_j\mathrm{d}x = 0\\
&\int e_{j\pm p}e_j\mathrm{d}x = 0 &&\int\frac{\mathrm{d}e_{j\pm p}}{\mathrm{d}x}e_j\mathrm{d}x = 0 \qquad \mathrm{p}>1
\end{aligned}$$

Using these results in (84) gives

$$\frac{1}{6}\left(\frac{\mathrm{d}\varphi_{j+1}}{\mathrm{d}t}+4\frac{\mathrm{d}\varphi_j}{\mathrm{d}t}+\frac{\mathrm{d}\varphi_{j-1}}{\mathrm{d}t}\right)+u_0\left(\frac{\varphi_{j+1}-\varphi_{j-1}}{2\Delta x}\right) = 0 \tag{85}$$

We find that this implicit scheme has a slightly smaller truncation error for the space derivative than the usual fourth order scheme.

Now consider how the scheme defined by (85) is used. in practice. Let F_j^n represent the time derivative of φ at node j and time level n .

$$\frac{\mathrm{d}\varphi_j^n}{\mathrm{d}t} = F_j^n \tag{86}$$

Applying (85) at time level n yields

$$\frac{1}{6}(F_{j+1}^n + 4F_j^n + F_{j-1}^n) = -u_0\left(\frac{\varphi_{j+1} - \varphi_{j-1}}{2\Delta x}\right) \qquad \text{for all } j \tag{87}$$

Since the RHS of (87) is known, this set of simultaneous linear equations can be solved for all the F_j^n . The next step is to introduce a time stepping scheme. For example, if the leapfrog scheme is used (86) becomes

$$\varphi_j^{n+1} = \varphi_j^{n-1} + 2\Delta t F_j^n \tag{88}$$

To study the stability of this scheme we combine (87) and (88) to give the complete numerical algorithm

$$\varphi_{j-1}^{n+1} + 4\varphi_j^{n+1} + \varphi_{j-1}^{n+1} = \varphi_{j-1}^n(1 + 6\alpha) + 4\varphi_j^n + \varphi_{j+1}^n(1 - 6\alpha) \tag{89}$$

and then use the von Neumann series method in the usual way. For this scheme it can be shown that there is stability if $\alpha = u_0\Delta t/\Delta x \le \sqrt{3}$; this is more restrictive than for the corresponding finite difference scheme which is centred in space and time. Further analysis shows that the scheme is neutral $(D = 1)$, with the relative phase speed of the physical mode being given by

$$r = \frac{1}{\alpha q}\operatorname{atan}\left\{\frac{p}{(s^2 - p^2)^{1/2}}\right\}$$

$$p = -\alpha\sin q \qquad s = \frac{2 + \cos q}{3} \qquad q = \frac{2\pi}{l}$$

The variation of D and r with l for the finite element method is given in Table 3 for the case where $\alpha = 0.5 \times (1/\sqrt{3})$. Also the results of using this technique on the test problem given in Subsection 2.6 are shown in Figs. 5 and 6 . Comparison of these with the results from the fourth order leapfrog scheme shows that they appear to produce forecasts of a similar quality. The major disadvantage of this method is that it is implicit.

The scheme defined by (87) and (88) (or (89)) is a three-level scheme. If a two-level scheme is required (i.e. a forward time difference) we can take the Crank–Nicolson approach and use a weighted mean of the advection terms at time levels n and $n + 1$ with weights β_n and β_{n+1} . The expressions corresponding to (87) and (88) are then

$$\frac{1}{6}(F_{j+1}^n + 4F_j^n + F_{j-1}^n) = \beta_{n+1}A_j^{n+1} + \beta_n A_j^n$$

$$\varphi_{j-1}^{n+1} = \varphi_j^n + \Delta t F_j^n$$

which can be combined to give

$$\varphi_{j-1}^{n+1}(1 - 3\alpha\beta_{n+1}) + 4\varphi_j^{n+1} + \varphi_{j+1}^{n+1}(1 + 3\alpha\beta_{n+1}) = \varphi_{j-1}^n(1 + 3\alpha\beta_n) + 4\varphi_{j+1}^n(1 - 3\alpha\beta_n)$$

A stability analysis shows that for $\beta_n > 1/2$ there is instability, whereas for $\beta_n \le 1/2$ there is absolute stability with $\beta_n = \beta_{n+1} = 1/2$ giving a neutral scheme (i.e. no damping, though there are phase errors). As the explicit scheme gives a coupled system of equations, no penalty is paid in using an implicit approach which is absolutely stable if both systems can be solved using the same kind of solver.

7.2 (b) . In the piecewise linear element representation, the function is obliged to behave linearly between nodes. To improve this fit we can use second-order polynomials as the basis functions as the convolution polynomials represented in Fig. 16 . With this representation we get not only continuity of the function at the nodes as in the piecewise linear case but also continuity of the first derivative.

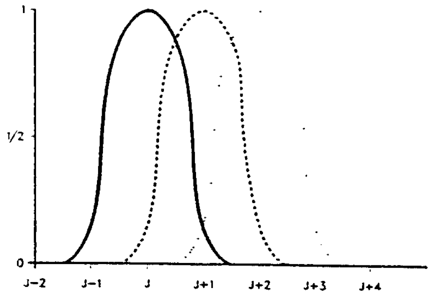


Figure 16. Second order polynomials as basis functions.

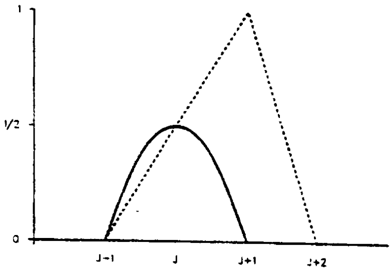


Figure 17. Linear together with quadratic elements.

An alternative is to use simultaneously linear and quadratic elements as the ones shown in Fig. 17 .

We don't automatically get continuity of the derivative at the nodes in this representation, but the fit of a given function between the nodes can be improved.

Now, if we apply the Galerkin approach to the linear advection equation, as was done in the piecewise linear representation, we get a similar system of simultaneous equations

$$\mathbf{A}\Psi^{n+1} = \mathbf{F}^n \tag{90}$$

but the matrix instead of being tridiagonal as it was in equations (89) is a less sparse matrix and therefore more expensive to solve.

7.3 Second-order derivatives

Let us now turn to the treatment by means of finite elements of an equation involving second-order space derivatives. As an example, we will show how finite-element techniques can be used to solve a simple Helmholtz equation

$$\frac{\partial^2\psi}{\partial x^2} - \alpha^2\psi = 0 \tag{91}$$

A first alternative is to use quadratic elements as the basis functions and use the Galerkin method as before

$$\sum_j \psi_j \left(\frac{\mathrm{d}^2 e_j}{\mathrm{d}x^2}, e_j\right) - \alpha^2 \sum_j \psi_j (e_j, e_i) = 0 \tag{92}$$

so that the second derivatives of the basis functions can be calculated analytically and then the five-diagonal system (92) solved.

A second alternative using linear elements is as follows:

Let us assume that we use the scalar product of space L, that is

$$(f, g) = \int_0^L fg \ \mathrm{d}\boldsymbol{x}$$

then (92) can be written as

$$\sum_j \psi_j \int \frac{\mathrm{d}^2 e_j}{\mathrm{d}x^2} e_i \mathrm{d}x - \alpha^2 \sum_j \psi_j \int e_j e_i \mathrm{d}x = 0$$

The first term can be integrated by parts

$$\int \frac{\mathrm{d}^2 e_j}{\mathrm{d}x^2} e_i \mathrm{d}x = \left[\frac{\mathrm{d}e_j}{\mathrm{d}x} e_i\right]_0^L - \int \frac{\mathrm{d}e_j}{\mathrm{d}x}\frac{\mathrm{d}e_i}{\mathrm{d}x}\mathrm{d}x$$

the first term of the RHS is zero for all $i \neq 1$ and $i \neq N + 1$, and now all the derivatives are first order and can be calculated analytically using linear elements.

The matrix of the resulting system of equations is tridiagonal except for the elements $(j = 1, i = 1)$, $(j = 2, i = 1)$, $(j = N + 1, i = N + 1)$, $(j = N, i = N + 1)$.

7.4 Boundaries, irregular grids and asymmetric algorithms

The finite-element method can easily cope with boundaries and irregular grids by choosing suitable basis functions. Also asymmetric algorithms can be derived by choosing test functions that are different from the basis functions. These aspects of the finite-element method will be illustrated by their application to the linear advection equation using a linear piecewise fit.

7.4 (a) Boundaries. Suppose we have boundaries at nodes $j = 1$ and $j = N + 1$. Making a linear piecewise fit it is easy to see that the basis functions for $2 \leq j \leq N$ are the usual hat functions, whereas the basis functions associated with the boundary nodes have a value of 1 at the boundary falling to 0 at the first internal node (see Fig. 18). The usual Galerkin procedure then gives

$$\begin{aligned}
&\frac{1}{3}\left(\frac{\mathrm{d}\varphi_2}{\mathrm{d}t}+2\frac{\mathrm{d}\varphi_1}{\mathrm{d}t}\right)+u_0\left(\frac{\varphi_1-\varphi_2}{\Delta x}\right) = 0 \\
&\frac{1}{6}\left(\frac{\mathrm{d}\varphi_{j-1}}{\mathrm{d}t}+4\frac{\mathrm{d}\varphi_j}{\mathrm{d}t}+\frac{\mathrm{d}\varphi_{j+1}}{\mathrm{d}t}\right)+u_0\left(\frac{\varphi_{j+1}-\varphi_j}{\Delta x}\right) = 0 \qquad 2\le j\le n \qquad (93)\\
&\frac{1}{3}\left(\frac{\mathrm{d}\varphi_N}{\mathrm{d}t}+2\frac{\mathrm{d}\varphi_{N+1}}{\mathrm{d}t}\right)+u_0\left(\frac{\varphi_{N+1}-\varphi_N}{\Delta x}\right) = 0
\end{aligned}$$

This set of equations can be solved for the $\mathrm{d}\varphi/\mathrm{d}t$ at all nodes, including the boundary nodes.

Now a paradox arises: we know that the linear advection equation has a unique solution given a suitable set of initial and boundary conditions, but the system (93) gives us, in principle, the values of $\mathrm{d}\varphi/\mathrm{d}t$ at all nodes and, therefore, does not allow us to specify any boundary condition. The same is true for the Helmholtz equation of Subsection 7.3.

The solution of this paradox is that either the matrix of the resulting system is singular and, therefore, the system of equations cannot be solved, or the system is over specified and the solution we get doesn't correspond to the boundary conditions.

The cure is then to scalarly multiply only by the interior elements that is, use in (84) or (92) only the values of $2\le i\le N$ and compute ψ_1 and ψ_{N+1} from the boundary conditions. The system then has $N-1$ equations and can be solved for the $N-1$ interior coefficients $\psi_j(2\le j\le N)$

7.4 (b) Irregular grids. Using the basis functions shown in Fig. 18 , it is straightforward to show that the finite-element formulation of the advection equation on an irregular grid is

$$\frac{1}{6}\left(\frac{\mathrm{d}\psi_{j-1}}{\mathrm{d}t}+2\frac{\mathrm{d}\psi_j}{\mathrm{d}t}\right)\Delta x_{j-1/2}+\frac{1}{6}\left(2\frac{\mathrm{d}\psi_j}{\mathrm{d}t}+\frac{\mathrm{d}\psi_{j+1}}{\mathrm{d}t}\right)\Delta x_{j+1/2}+u_0(\psi_{j+1}-\psi_{j-1}) = 0$$

Naturally this reduces to (85) when the grid is uniform, i.e. $\Delta x_{j-1} = \Delta x_{j+1} = \Delta x$.

7.4 (c) Asymmetric algorithms. So far the choice of linear basis functions has lead to symmetric algorithms. However, this symmetry can be broken by using asymmetric test functions. For example, the use of the basis and test functions illustrated in Fig. 15 in the advection equation produces an algorithm which has some of the characteristics of the upstream finite difference scheme.

$$\frac{1}{3}\left(\frac{\mathrm{d}\psi_{j-1}}{\mathrm{d}t}+2\frac{\mathrm{d}\psi_j}{\mathrm{d}t}\right)+u_0\left(\frac{\psi_j-\psi_{j-1}}{\Delta x}\right) = 0$$

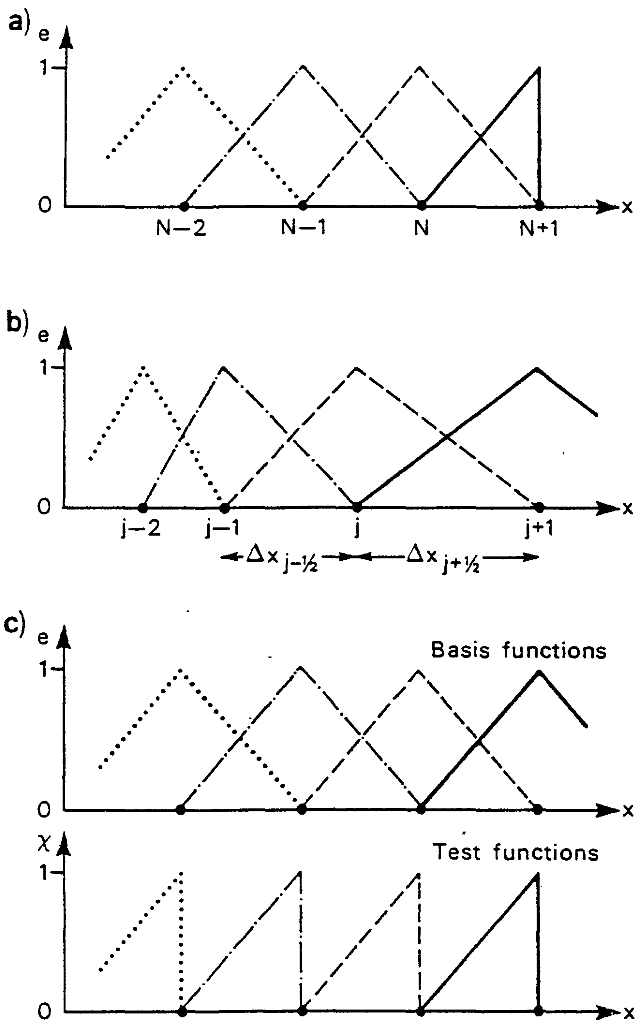


Figure 18. llustrations of (a) linear basis functions in the vicinity of a boundary, (b) linear basis functions for an irregular grid and (c) linear basis and test functions which would give asymmetric algorithms.

7.5 Treatment of non-linear terms

Consider the treatment of the non-linear tern in the one-dimensional advection equation

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

A straightforward one-step approach is to use

$$u = \sum_j u_j e_j \qquad D = \frac{\partial u}{\partial x} = \sum_j u_j \frac{\mathrm{d}e_j}{\mathrm{d}x}$$

Substitution in the non-linear equation and making the Galerkin assumption yields

$$\frac{1}{6}\left(\frac{\mathrm{d}u_{j+1}}{\mathrm{d}t}+4\frac{\mathrm{d}u_j}{\mathrm{d}t}+\frac{\mathrm{d}u_{j-1}}{\mathrm{d}t}\right) = -\frac{1}{2\Delta x}\left\{\frac{(u_{j+1}+2u_j)}{3}(u_{j+1}-u_j)+\frac{(u_{j-1}+2u_j)}{3}(u_j-u_{j-1})\right\}$$

Alternatively, a two-step method can be used. In this we first find the best piecewise approximation to D by using the Galerkin assumption

$$\frac{1}{6}(D_{j+1}+4D_j+D_{j-1}) = \frac{u_{j+1}-u_{j-1}}{2\Delta x}$$

Having solved this set of equations for the D_j, the second step is to find the best approximation to uD by again using the Galerkin assumption

$$\frac{1}{6}\left(\frac{\mathrm{d}u_{j+1}}{\mathrm{d}t}+4\frac{\mathrm{d}u_j}{\mathrm{d}t}+\frac{\mathrm{d}u_{j-1}}{\mathrm{d}t}\right) = -\frac{1}{12}\{u_{j-1}(D_{j-1}+D_j)+u_{j+1}(D_{j+1}+D_j)+u_j(D_{j+1}+6D_j+D_{j-1})\}$$

This is more accurate than the one-step procedure. but it has the disadvantage that an extra matrix inversion is required to find the D_j.

Finally, it is worth noting that finite element schemes do not appear to suffer from aliasing and non-linear instability. This happens because the interactions which normally give rise to aliasing are heavily smoothed in the finite-element method.

7.6 Staggered grids and two-dimensional elements

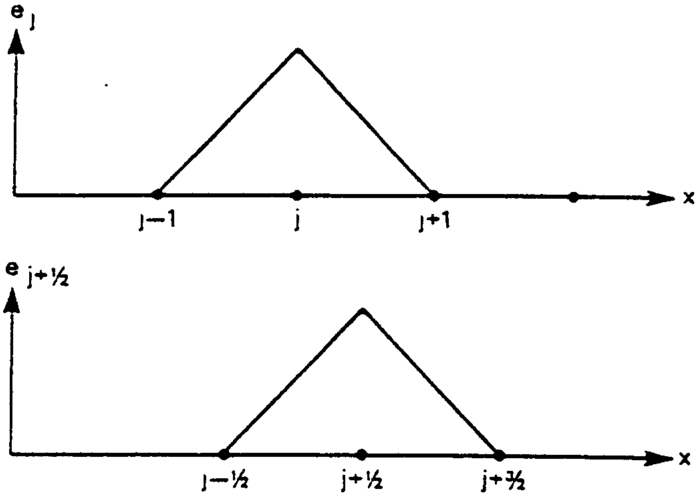


Figure 19. Staggered piecewise linear elements.

In Subsection 4.3 it was shown that it is natural to use a staggered grid when dealing with the gravity-wave equations. Therefore, we will now consider the finite-element approximations to these equations using linear basis functions and a staggered grid.

Define two sets of basis functions (e_j and $e_{j+1/2}$) shown in Fig. 19 and assume that

$$h = \sum_j h_j e_j \quad \text{and} \quad u = \sum_j u_{j+1/2} e_{j+1/2}$$

Substituting the expansions in the following equation

$$\frac{\partial h}{\partial t} + H\frac{\partial u}{\partial x} = 0$$

and using the Galerkin procedure with the e_j as test functions gives

$$\sum \frac{\mathrm{d}h_j}{\mathrm{d}t}\int e_i e_j \mathrm{d}x + H\sum u_{j+1/2}\int e_j \frac{\mathrm{d}e_{j+1/2}}{\mathrm{d}x}\mathrm{d}x$$

Calculation of the integrals leads to

$$\frac{1}{6}\left(\frac{\mathrm{d}h_{j-1}}{\mathrm{d}t} + 4\frac{\mathrm{d}h_j}{\mathrm{d}t} + \frac{\mathrm{d}h_{j+1}}{\mathrm{d}t}\right) + H\frac{(u_{j+1/2} - u_{j-1/2})}{\Delta x} = 0$$

The corresponding finite element approximation to the other equation

$$\frac{\partial u}{\partial t} + g\frac{\partial h}{\partial x} = 0$$

is the following

$$\frac{1}{6}\left(\frac{\mathrm{d}u_{j+3/2}}{\mathrm{d}t} + 4\frac{\mathrm{d}u_{j+1/2}}{\mathrm{d}t} + \frac{\mathrm{d}u_{j-1/2}}{\mathrm{d}t}\right) + g\frac{(h_{j+1} - h_{j-1})}{\Delta x} = 0$$

7.7 Two dimensional elements

With rectangular mesh we can define rectangular elements where the linear basis function $e_{ij}(x, y)$ associated with node (i, j) has a value of unity at this node and falls to zero at the 8 adjacent nodes (see Fig. 20). A variable φ can then be expanded in terms of these basis functions.

$$\varphi(x, y, t) = \sum_{i,j} \varphi_{ij} e_{ij}(x, y)$$

Substituting this in the original partial difference equation and using the usual Galerkin procedure to orthogonalize the error leads to a set of equations describing the behaviour of the expansion coefficients φ_{ij}.

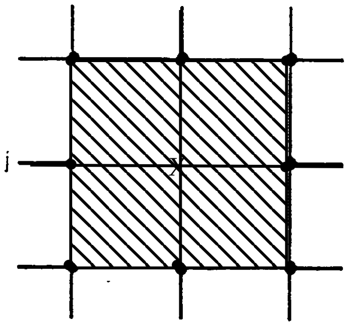


Figure 20. llustration of a two dimensional linear basis function for a rectangular grid. In the shaded area the basis function is non-zero; the basis function is zero at nodes marked by • and unity at the node marked X.

For solving equations with spherical geometry, it is possible to generate a grid of icosohedra, with each triangular face divided into equilateral triangles. Each element then has the form shown in Fig. 21 . To illustrate the kind of algorithms produced, just one example will be given. Using linear elements it can be shown that the finite element description of the derivative $D = \partial\varphi / \partial x$ is

$$\frac{1}{12}(D_1 + D_2 + D_3 + 6D_4 + D_5 + D_6 + D_7) = \frac{1}{6\Delta x}[(\varphi_2 - \varphi_1) + 2(\varphi_5 - \varphi_3) + (\varphi_7 - \varphi_6)]$$

Cullen (1974) has used this approach in a primitive equation model using spherical geometry.

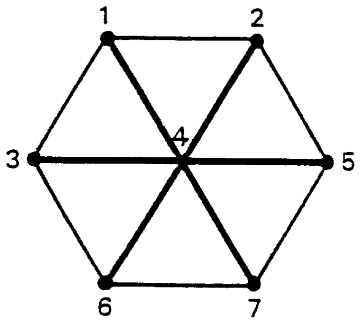


Figure 21. An element is made up of 6 triangles.

7.8 The local spectral technique

One of the advantages of the finite-element method is the possibility of using irregular grids while still maintaining a high degree of accuracy, as opposed to the finite difference technique. This allows us to define elements whose shape is adapted to the geometry of the domain in which we want to solve our equations. This possibility is the basis of the success that finite elements have had in engineering problems involving complicated structures. The main weakness of the method is that, inside each element the function is assumed to have a linear behaviour, or otherwise we end up with a system of equations whose matrix is not sparse and, therefore, is very expensive to solve both in terms of CPU time and storage memory.

A way round this problem is provided by the local spectral technique. In this approach we define a set of local domains, just as in the finite-element method, but we use inside each element a spectral representation, taking as basis functions a set of Lagrange interpolating polynomials and imposing continuity of the solution through the element boundaries. This gives us a system of equations with a blocked matrix, each block being diagonal and therefore very sparse.

The technique is very well suited for implementation on a parallel computer if the interior of each element is solved in a single processor, the communications being limited to passing a small quantity of information between nearest neighbours only.

7.9 Application for the computation of vertical integrals in the ECMWF model

In the semi-Lagrangian version of the ECMWF forecast model, the vertical discretization is needed only in order to compute the vertical integrals of the continuity and the hydrostatic equations. The quantities to be integrated are defined at “full” levels and the integration is performed by the mid-point rule, so that the integrals are in principle only available at “half” levels. Extrapolation or averaging to full levels compromise the second-order accuracy of the integration.

As an alternative, a finite-element scheme has been developed using cubic splines as basis functions. These B-splines differ from the cubic splines defined in Subsection 5.3. The B-splines are defined as piecewise cubic (at each interval) polynomials which are non-zero only over 4 grid intervals, whose zeroth, first and second derivatives are continuous and whose integral over the whole domain is prescribed.

These polynomials can be used as basis functions for the finite-element method. Unlike the case with the piecewise linear elements, we can not use the collocation method because the coefficients of the expansion of a function in terms of these basis functions are not the values of the function at the nodes.

Let’s compute the value of a vertical integral using this method:

$$F(x) = \int_0^x f(y)dy$$

Then we expand both F(x) and f(y) as a linear combination of B-splines (the basis fuctions chosen to expand both functions could be different. In our case they are the same except for the boundaries where they are modified to suit the appropriate boundary conditions)

$$\sum_{i=1}^{N} \Psi_i d_i(x) = \sum_{i=1}^{N} \psi_i \int_0^x e_i(y)dy$$

Now we apply the Galerkin procedure using some “test functions” $t_i(x)$

$$\sum_{i=1}^{N} \Psi_i \int_0^1 \mathrm{d}_i(x) t_j(x)dx = \sum_{i=1}^{N} \psi_i \int_0^1 \left[t_j(x) \int_0^x e_i(y)(dy) \right] dx$$

which can be expressed in matrix form

$$\utilde{A}\tilde{\Psi} = \utilde{B}\tilde{\psi} \quad => \quad \tilde{\Psi} = \utilde{A}^{-1}\utilde{B}\tilde{\psi} \qquad (94)$$

The initial information we get to perform the integral is the set of values of f(x), say $\tilde{f}$, at the “full levels” of the model and the final result we need is the value of F(x) also on the full levels of the model, say $\tilde{F}$. If we choose the number of basis functions the same as the number of degrees of freedom of f(x) (including appropriate boundary conditions) then transforming this set of values to the vector $\tilde{\psi}$ is simply a matrix multiplication by a square matrix, say $\underset{\sim}{S}$, and the projection from $\tilde{\Psi}$ to the values of F(x) is a multiplication by another matrix, say $\underset{\sim}{S}'^{-1}$. Therefore expression (94) can be written as

$$\tilde{F} = \underset{\sim}{S}'^{-1}\underset{\sim}{A}^{-1}\underset{\sim}{B}\underset{\sim}{S}\tilde{f}$$

and the matrix $\underset{\sim}{S}'^{-1}\underset{\sim}{A}^{-1}\underset{\sim}{B}\underset{\sim}{S}$ is our integration operator.

8. Solving the algebraic equations

8.1 Introduction

In all the methods we have seen for solving the partial differential equations of atmospheric motion we finally arrive at a set of simultaneous algebraic equations where the unknowns are the grid points or the coefficients at time step $t + \Delta t$ and we have to solve this system.

The spectral method leads to the simplest case where the matrix of the system to be solved

$$\mathbf{Ax} = \mathbf{B}$$

is diagonal due to the orthogonality of the basis functions chosen, and so the equations of the system are decoupled from one another. The solution then is straightforward, each equation having only one unknown. On the other hand, as we saw in the chapter on the spectral technique, the transformations to grid-point space and back to spectral space are very expensive in terms of computing, mainly when the number of degrees of freedom in the model is increased and so finite-difference and finite-element methods cannot be discarded, even in the horizontal discretization.

In these cases, the system of algebraic equations we arrive at is coupled, mostly in the form of tridiagonal or block tridiagonal matrices.

The simplest method of solving a system of simultaneous equations is by matrix inversion, so that if matrix $\mathbf{A}$ is non-singular, it has an inverse $\mathbf{A}^{-1}$ and the system can be transformed into

$$\mathbf{A}^{-1}\mathbf{Ax} = \mathbf{A}^{-1}\mathbf{B} \quad \Rightarrow \quad \mathbf{x} = \mathbf{A}^{-1}\mathbf{B}$$

The drawback of this method is that, with large matrices, the inversion operation is very expensive both in terms of memory and CPU time.

8.2 Gauss elimination

Let us assume we have a one-dimensional model treated by means of finite differences (centred) or finite elements (linear) of dimension n (the number of grid points or the number of elements). We end up with the following system of equations at every time step

$$
\begin{aligned}
a_{11}x_1 + a_{21}x_2 \quad & = b_1 \\
a_{12}x_1 + a_{22}x_2 + a_{32}x_3 \quad & = b_2 \\
a_{23}x_2 + a_{33}x_3 + a_{43}x_4 \quad & = b_3 \\
\\
\\
a_{n-1,n}x_{n-1} + a_{n,n}x_n \quad & = b_n
\end{aligned}
$$

or $\mathbf{Ax} = \mathbf{B}$ with $\mathbf{A}$ tridiagonal

$$
\mathbf{A} = \begin{bmatrix}
a_{11} & a_{21} & 0 & 0 & 0 & \dots & 0 \\
a_{12} & a_{22} & a_{32} & 0 & 0 & \dots & 0 \\
0 & a_{23} & a_{33} & a_{43} & 0 & \dots & 0 \\
0 & 0 & & & & & \\
0 & 0 & & & & & \\
. & . & . & & & & \\
. & . & . & & 0 & a_{n-1,n} & a_{n,n}
\end{bmatrix}
$$

The most used method for solving this system is the so-called Gauss elimination, or forward elimination and back substitution. The method is implemented in most scientific subroutine libraries and can, therefore, be used by a simple subroutine call. It runs as follows:

From the first equation, we extract x_1

$$x_1 = \frac{(b_1 - a_{21}x_2)}{a_{11}}$$

and substitute in the second equation

$$a_{12}\frac{(b_1 - a_{21}x_2)}{a_{11}} + a_{22}x_2 + a_{32}x_3 = b_2$$

now we extract x_2

$$x_2 = \frac{\left(b_2 - \frac{a_{21}b_1}{a_{11}} - a_{32}x_3\right)}{\left(a_{22} - \frac{a_{21}}{a_{11}}\right)}$$

and substitute in the third equation ... and so on

When we reach the last equation and substitute x_{n-1} from the last but one, we are left with an equation in a single unknown which can therefore be solved and the result substituted in the expression for x_{n-1} taken from the last but one equation, and so on until we arrive back at the expression for x_1.

The method works as long as the matrix of the system is not quasi-singular and the denominators (pivots) of the expressions are not too small. It is, therefore, useful to reorder the unknowns so that the pivots a_{11}, $a_{22} - a_{21}/a_{11}$

are as big as possible; this is always done in the scientific subroutines from well developed libraries.

In matrix form, the method is equivalent to decomposing the original matrix $\mathbf{A}$ in the form

$$\mathbf{A} \equiv \mathbf{M}\mathbf{K}\mathbf{M}^{\mathrm{T}}$$

where $\mathbf{K}$ is diagonal and

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & & & & & \\ m_1 & 1 & 0 & & & & & \\ 0 & m_2 & 1 & & & & & \\ & & & & & & & \\ & & & m_{j+1} & 1 & m_{j+2} & & \\ & & & & 0 & 1 & m_{j+3} & \\ & & & & & 0 & 1 & m_n \\ & & & & & & & 1 \end{bmatrix}$$

where $j = \mathrm{int}\left(\frac{n}{2}\right)$ and n is the rank of matrix $\mathbf{A}$ ($\mathbf{M}^{\mathrm{T}}$ is the transpose of M)

8.3 Iterative methods

When the matrix is not as sparse as in the previous example of Gauss elimination, a direct method of solution could be intractable due to memory and/or CPU limitations, large amounts of both resources being needed for inverting a large matrix. The most straightforward methods are then the iterative methods. We need to solve the system.

$$\mathbf{A}\mathbf{x} = \mathbf{B} \tag{95}$$

and start off with a guess x_0 for the solution x. This not being in general the true solution, we can calculate a residual.

$$\mathbf{R}_0 = \mathbf{A}\mathbf{x}_0 - \mathbf{B} \tag{96}$$

and use it to get a new estimate x_1 and a new residual

$$\mathbf{R}_1 = A\mathbf{x}_1 - \mathbf{B} \tag{97}$$

and so on. If the residuals $\mathbf{R}_n$ are smaller as n increases, the method converges and we stop when the residual becomes smaller than a pre-defined magnitude.

The general procedure for iterative methods can be expressed as follows

$$\begin{aligned} \mathbf{A}\mathbf{x} &= \mathbf{B} \\ \mathbf{Q}^{-1}\mathbf{A}\mathbf{x} &= \mathbf{Q}^{-1}\mathbf{B} \end{aligned} \tag{98}$$

where $\mathbf{Q}$ is known as the splitting, or preconditioning, matrix and could be simply the unity matrix. Then we add and subtract $\mathbf{I}\mathbf{x}$

$$\mathbf{x} + (\mathbf{Q}^{-1}\mathbf{A} - \mathbf{I})\mathbf{x} = \mathbf{Q}^{-1}\mathbf{B} \tag{99}$$

$$\mathbf{x} = (\mathbf{I} - \mathbf{Q}^{-1}\mathbf{A})\mathbf{x} + \mathbf{Q}^{-1}\mathbf{B} \tag{100}$$

we then obtain the $(n+1)$ th iterative estimate of x from the n th estimate by

$$\mathbf{x}^{n+1} = (\mathbf{I} - \mathbf{Q}^{-1}\mathbf{A})\mathbf{x}^{n} + \mathbf{Q}^{-1}\mathbf{B} \tag{101}$$

This equation can be viewed as the discrete analogue (with unit time step) of

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = -\mathbf{Q}^{-1}\mathbf{A}\mathbf{x} + \mathbf{Q}^{-1}\mathbf{B} \tag{102}$$

which has a stationary solution when the right hand side becomes zero.

The general solution of evolutionary problem (102) is

$$\mathbf{x} = \exp[\lambda t] - \mathbf{k} \tag{103}$$

where λ is an eigenvalue of matrix $-\mathbf{Q}^{-1}\mathbf{A}$ (the diagonal elements of the diagonalized $-\mathbf{Q}^{-1}\mathbf{A}$ matrix) and $\mathbf{k}$ is a constant vector. The solution approaches the stationary solution $\mathbf{x} = \mathbf{k}$ when the real part of λ is negative, that is to say, if all the eigenvalues of matrix $\mathbf{Q}^{-1}\mathbf{A}$ have real positive parts (elliptic problem). The solution can be made to converge quicker by multiplying the eigenvalues by a constant greater than 1 (successive over-relaxation).

The iteration procedure (101) is performed successively over each component of vector $\mathbf{x}$; if we use in the right-hand side of (101) always the components of x from the n th iteration the method is called Jacobi iteration; on the other hand, if we use on the right-hand side of the new iteration values of the components of x whenever they are available, the procedure is called Gauss–Seidel iteration, it cuts down the storage requirement on a computer as only one value of each component of x (either the n th iteration or the estimate) needs to be kept and it can be shown to converge quicker than the Jacobi method.

As an example, let us work out the iterative solution of the Helmholtz equation in centred finite-difference form

$$\nabla^2 x_{ij} - \lambda_{ij}^2 x_{ij} = F_{ij} \tag{104}$$

where ∇^2 is the discrete Laplace operator and λ_{ij} and F_{ij} are known. If x_{ij}^n is the n th iteration for this solution, we get the "residual" vector

$$\nabla^2 x_{i,j}^n - \lambda_{i,j}^2 x_{i,j} - F_{i,j} = R_{i,j}^n \tag{105}$$

or

$$x_{i-1,j}^n + x_{i+1,j}^n + x_{i,j-1}^n + x_{i,j+1}^n + (-4 - \lambda_{i,j}^2)x_{i,j}^n - F_{i,j} = R_{i,j}^n \tag{106}$$

and we take the $(n+1)$ th iteration of $x_{i,j}$ such that the new residual is zero

$$x_{i-1,j}^n + x_{i+1,j}^n + x_{i,j-1}^n + x_{i,j+1}^n + (-4 - \lambda_{i,j}^2)x_{i,j}^{n+1} - F_{i,j} = 0 \tag{107}$$

$$x_{i,j}^{n+1} = x_{i,j}^{n} + \frac{R_{i,j}^{n}}{4+\lambda_{i,j}^{2}} \tag{108}$$

This is the Jacobi iterative method.

If we proceed for the calculation of the new components in the sense of increasing sub-indexes i and j, we can use in (106) the already available values of $x_{i-1,j}^{n+1}$ and $x_{i,j-1}^{n+1}$ instead of the n th iteration values and we get the Guass–Seidel procedure.

If we multiply in (108) the fraction by a factor μ (the over-relaxation factor) before adding it to $x_{i,j}^{n}$ we get the successive overrelaxation method or SOR

$$x_{i,j}^{n+1} = x_{i,j}^{n} + \left(\frac{R_{i,j}^{n}}{4+\lambda_{i,j}^{2}}\right)\mu \tag{109}$$

It can be shown that, in an iterative method, the short-scale errors of the first guess with respect to the true solution converge very quickly towards zero. What makes iterative methods expensive in terms of computer time is the slow convergence of the long-range features of the initial error. This suggests the so-called multigrid methods in order to speed up the convergence of an iterative method to the point of making it competitive with direct methods, such as the ones which are described later.

If we chose a subset of grid points from the original grid, say one of every four points and solve the equation over this reduced grid, the long-scale features are seen from this grid as shorter scale because they cover a smaller number of grid points and, therefore, the convergence is faster. Once a solution is found on a coarse grid, we interpolate it to the finer grid and refine the solution in this grid. The procedure can run over a range of grid sizes and can be iterated forwards and backwards from the coarser to the finer grids.

A further refinement of the method is called the adaptive multigrid method. In this procedure we define the finer grid on which the solution from the coarse grid has to be refined only in the domain regions where we find that the truncation error of the completed solution exceeds a certain predefined threshold value.

8.4 Decoupling of the equations

The fields to be forecasted in P.E. models are three dimensional and, therefore, the matrices of the algebraic system to which we arrive when we discretize the partial differential equations are, in principle, six-dimensional and therefore too big to be treated directly by any direct or iterative means.

The general system of equations can be expressed as

$$\sum_{i}\sum_{j}\sum_{k} A_{ijk}^{lmn} x_{ijk} = B_{lmn} \tag{110}$$

8.4 (a) Separable case.

The simplest case would be that the matrix be factorizable, i.e.

$$A_{ijk}^{lmn} \equiv C_i^l D_j^m E_k^n \tag{111}$$

but this case is unfortunately very rare.

The procedure would then be as follows:

(i) Define

$$Z_i^{mn} = \sum_j \sum_k D_j^m E_k^n x_{ijk} = \sum_j D_j^m Y_{ij}^n \tag{112}$$

where

$$Y_{ij}^n = \sum_k E_k^n x_{ijk} \tag{113}$$

(ii) Solve

$$\sum_i C_i^m Z_i^{mn} = B_{lmn} \tag{114}$$

for each pair (m, n)

(iii) Solve

$$\sum_j D_j^m Y_{ij}^n = Z_i^{mn} \tag{115}$$

for each pair (m, i)

(iv) Finally solve

$$\sum_k E_k^n x_{ijk} = Y_{ij}^n \tag{116}$$

for each pair (i, j)

8.4 (b) Use of the eigenvector matrix.

Let us consider the case of Poisson equation in the three dimensions

$$\nabla_3^2 \varphi = F = \nabla_{\mathrm{h}}^2 \varphi + \frac{\partial^2 \varphi}{\partial z^2} \tag{117}$$

where ∇_{h}^2 is the horizontal Laplacian in Cartesian coordinates.

If we apply centred finite differences in the vertical we get the system of coupled equations

$$\nabla^2 \underset{\sim}{\varphi} + \mathbf{M} \underset{\sim}{\varphi} = \mathbf{F} \tag{118}$$

where $\underset{\sim}{\varphi}$ is the vector of fields φ at the different vertical levels and matrix $\mathbf{M}$ stands for

$$\mathbf{M} = \begin{bmatrix} -2 & 1 & 0 & \dots & & \\ 1 & -2 & 1 & 0 & \dots & \\ 0 & 1 & -2 & 1 & 0 & \dots \\ & & & & & \\ . & . & . & . & . & . \\ . & . & . & . & . & . \end{bmatrix} \text{ of rank } K \text{ (the number of levels)}$$

Let E_j $(j = 1, \dots K)$be the eigenvectors of this matrix which form the columns of matrix $\mathbf{E}$

Then the system (118) may be written as

$$\nabla^2\underset{\sim}{\varphi}' + \mathbf{E}^{-1}\mathbf{M}\mathbf{E}\underset{\sim}{\varphi}' = \mathbf{F}' \tag{119}$$

where $\underset{\sim}{\varphi}' = \mathbf{E}^{-1}\underset{\sim}{\varphi}$ and $\mathbf{F}' = \mathbf{E}^{-1}\mathbf{F}$.

Matrix $\mathbf{E}^{-1}\mathbf{ME}$ is a diagonal matrix made of the eigenvalues of $\mathbf{M}$ and therefore (119) is a system of K decoupled bi-dimensional equations.

8.4 (c) Fourier transform in a bi-dimensional roblem.

Let

$$x_{pj} = \frac{1}{M}\sum_{i=0}^{M} E^i x_{ij}\sin\left(\frac{ip\pi}{M}\right) \quad \text{and} \quad B^{pn} = \frac{1}{M}\sum_{m=0}^{M} E^m B^{mn}\sin\left(\frac{mp\pi}{M}\right) \tag{120}$$

where

$$E^i = \begin{cases} 1/2 & \text{if} \quad i = 0 \quad \text{or} \quad i = M \\ 1 & \text{otherwise} \end{cases} \quad \text{(direct Fourier transform)}$$

taking into account the orthogonality relationship

$$\sum_{k=0}^{M} \sin\left(\frac{km\pi}{M}\right)\sin\left(\frac{kp\pi}{M}\right) = \frac{1}{2}M\delta_{p,m} \tag{121}$$

for the Fourier functions, the original bidimensional system

$$\sum_i\sum_j A_i^{mn} x_{ij} = B^{mn} \tag{122}$$

is reduced to

$$\sum_j A_j^n(p)\hat{x}_{pj} = B^{mn} \tag{123}$$

which are $M + 1$ (p=0,...M) systems of one-dimensional equations from whose solution we can then find

$$x_{ij} = 2\sum_{p=0}^{M} E^p \hat{x}_{pj} \sin\left(\frac{ip\pi}{M}\right) \text{ (inverse Fourier transform)}$$

The reduction of the system to form (123) can be accomplished for matrices whose eigenvectors are the Fourier bases, such as the one for a Poisson or a Helmholtz equation. Let us consider the Poisson equation using centred second order finite differences

$$\nabla^2 \mathbf{U} = \mathbf{V} \quad \text{or} \quad \mathbf{BU} = \mathbf{V}$$

where

$$\mathbf{B} = \begin{bmatrix} \mathbf{A} & \mathbf{I} & \mathbf{0} & & \dots \\ \mathbf{I} & \mathbf{A} & \mathbf{I} & \mathbf{0} & \dots \\ \mathbf{0} & \mathbf{I} & \mathbf{A} & \mathbf{I} & \dots \\ . & . & . & . & \dots \end{bmatrix} \qquad \mathbf{U} = \begin{bmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \\ . \\ \mathbf{U}_N \end{bmatrix} \qquad \mathbf{V} = \begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \\ . \\ \mathbf{V}_N \end{bmatrix}$$

$\mathbf{U}_n$ being the grid point value of the unknown at row n

$\mathbf{V}_n$ being the grid point value of the second number at row n

$$\mathbf{A} = \begin{bmatrix} -4 & 1 & 0 & . & . & \dots \\ 1 & -4 & 1 & 0 & . & \dots \\ 0 & 1 & -4 & 1 & 0 & \dots \\ . & . & . & . & . & \dots \end{bmatrix} \qquad \mathbf{I} = \begin{bmatrix} 1 & 0 & 0 & \dots \\ 0 & 1 & 0 & \dots \\ 0 & 0 & 1 & \dots \\ . & . & . & \dots \end{bmatrix}$$

The eigenvalues of $\mathbf{A}$ are $\lambda_j = -4 + 2\cos(j\pi/M)$; $j = 1, \dots M$, and the corresponding eigenvectors

$$\mathbf{q}_j = \left[\sin\left(\frac{j\pi}{M}\right) \; \sin\left(\frac{2j\pi}{M}\right) \; \sin\left(\frac{3j\pi}{M}\right) \; . \; \dots \right]$$

The same holds for any matrix of the form

$$\mathbf{A}^* = \begin{bmatrix} a & b & 0 & . & \dots \\ b & a & b & 0 & \dots \\ 0 & b & a & b & \dots \\ . & . & . & . & \dots \\ . & . & . & . & \dots \end{bmatrix}$$

whose eigenvalues are $\lambda_j = a + 2b\cos(j\pi/M)$

This matrix appears in the finite difference discretization of a Helmholtz equation.

Calling

$$\mathbf{Q} = [\mathbf{q}_1 \; \mathbf{q}_2 \; \dots \; \mathbf{q}_M]$$

we have

$$\mathbf{QAQ}^{\mathrm{T}} = \mathrm{diag}(\lambda_j) \equiv \Lambda \quad \text{and} \quad \mathbf{Q}^{\mathrm{T}}\mathbf{Q} = \mathbf{I}$$

The original system may be written as

$$\mathbf{U}_{k+1} + \mathbf{AU}_k + \mathbf{U}_{k-1} = \mathbf{V}_k$$

Multiplication by $\mathbf{Q}$ in this system gives

$$\mathbf{QU}_{k+1} + \mathbf{QAQ}^{\mathrm{T}}\mathbf{QU}_k + \mathbf{QU}_{k-1} = \mathbf{QV}_k$$

The product of $\mathbf{Q}$ by a vector $\mathbf{U}_k$ is the discrete Fourier transform of the vector $\mathbf{U}$, namely $\tilde{\mathbf{U}}_k$, and therefore we get

$$\tilde{\mathbf{U}}_{k+1} + \Lambda\tilde{\mathbf{U}}_k + \tilde{\mathbf{U}}_{k-1} = \mathbf{V}_k$$

and the equation for Fourier component j is

$$U^j_{k+1} + \lambda_j U^j_k + U^j_{k-1} = V^j_k$$

which is a set of decoupled equations for the Fourier components. This method is therefore identical to the vertical decoupling of section Subsection 8.4 (b) but, when the eigenvectors are the Fourier basis functions, there is the advantage of using the Fast Fourier Transform algorithm in projecting onto the eigenvector space.

8.5 The Helmholtz equation

In many of the present forecast models, the equation of the semi-implicit time stepping scheme leads to a Helmholtz equation

$$(1 - \Gamma\nabla^2)\mathbf{x} = \mathbf{B} \tag{124}$$

where Γ is a matrix for the vertical coordinate and its dimension is the number of levels in the model.

By the method of vertical decoupling of Subsection 8.4 we can convert set (124) into a set of "horizontal" equations, one for each level. Nevertheless, one of the advantages of using the spectral technique on a global model based on the spherical harmonics is that these functions are eigenfunctions of the Laplacian operator so that effectively the set of equations (124) are already decoupled in the horizontal and the coupling is between the different vertical levels for each spectral coefficient of $\mathbf{x}$, that is

$$[(1 - \Gamma\nabla^2)\mathbf{x}]^m_n \equiv \left(1 + \frac{n(n+1)}{a^2}\Gamma\right)\mathbf{x}^m_n \tag{125}$$

the system

$$\left(1 + \frac{n(n+1)}{a^2}\Gamma\right)\mathbf{x}^m_n = B^m_n \tag{126}$$

is, for fixed (m, n), a system of N (N = number of vertical levels) equations, easily solved by simple matrix inversion.

In the case of finite differences or finite elements in the horizontal things are not as easy and we have to decouple the equations in the vertical to arrive at a set of horizontal Helmholtz equations

$$(1 - K\nabla^2)\mathbf{x} = \mathbf{B} \tag{127}$$

whose matrices are very large but sparse.

We can solve each system (127) by an iterative (expensive) method, by use the Fourier transform method (if we have the appropriate boundary conditions) or by a block reduction algorithm as follows:

Let the problem be to solve equation

$$\mathbf{D}\psi = \mathbf{G} \tag{128}$$

in two dimensions, where **D** is a block tridiagaonal matrix as found with centred second order finite differences or piecewise linear finite elements

$$\mathbf{D} = \begin{bmatrix} \mathbf{E} & -\mathbf{I} & \mathbf{0} & \mathbf{0} & \dots \\ -\mathbf{I} & \mathbf{E} & -\mathbf{I} & \mathbf{0} & \dots \\ \mathbf{0} & -\mathbf{I} & \mathbf{E} & -\mathbf{I} & \dots \\ . & . & . & . & \dots \end{bmatrix} \tag{129}$$

where $\mathbf{E}$ is a matrix (normally also tri-diagaonal) and $\mathbf{I}$ the unit matrix corresponding to one dimension. ,

Therefore, if we have discretized dimension x by M values and dimension y by N values, $\mathbf{E}$ and $\mathbf{I}$ are $M \times M$ matrices and $\mathbf{D}$ has $N \times N$ blocks.

Now multiply each even row by $\mathbf{E}$ and add the odd rows immediately above and below giving

$$\begin{bmatrix} \mathbf{E} & -\mathbf{I} & \mathbf{0} & \mathbf{0} & . & \dots \\ \mathbf{0} & \mathbf{E}^2 - 2\mathbf{I} & \mathbf{0} & -\mathbf{I} & \mathbf{0} & \dots \\ \mathbf{0} & -\mathbf{I} & \mathbf{E} & -\mathbf{I} & \mathbf{0} & \dots \\ . & . & . & . & . & \dots \end{bmatrix} \begin{bmatrix} \psi_1 \\ \psi_2 \\ . \\ \psi_N \end{bmatrix} = \begin{bmatrix} \mathbf{g}_1 \\ \mathbf{g}_1 + \mathbf{g}_3 + \mathbf{E}\mathbf{g}_2 \\ \mathbf{g}_3 \\ . \end{bmatrix} \tag{130}$$

the fourth block equation reads

$$(\mathbf{E}^2 - 2\mathbf{I})\psi_4 - \mathbf{I}(\psi_2 + \psi_6) = \mathbf{g}_3 + \mathbf{g}_5 + \mathbf{E}\mathbf{g}_4$$

which includes only even numbered ψ 's; we can therefore write down a system of equations for the even numbered ψ 's

$$\begin{bmatrix} \mathbf{E}^2-2\mathbf{I} & -\mathbf{I} & \mathbf{0} & . & \dots \\ -\mathbf{I} & \mathbf{E}^2-2\mathbf{I} & -\mathbf{I} & . & \dots \\ \mathbf{0} & -\mathbf{I} & \mathbf{E}^2-2\mathbf{I} & -\mathbf{I} & \dots \\ . & . & . & . & \dots \end{bmatrix} \begin{bmatrix} \psi_2 \\ \psi_4 \\ . \\ . \end{bmatrix} = \begin{bmatrix} \mathbf{g}_1+\mathbf{g}_3+\mathbf{E}\mathbf{g}_2 \\ \mathbf{g}_1+\mathbf{g}_5+\mathbf{E}\mathbf{g}_4 \\ . \\ . \end{bmatrix} \tag{131}$$

which is of the same form as the original system but with the dimension reduced and the method can be iterated until left with a single block

$$\mathbf{E}^{(n)}\psi_i = \mathbf{g}_i^{(n)} \tag{132}$$

$\mathbf{E}^{(n)}$ is of the form

$$\mathbf{E}^{(r+1)} = \{\mathbf{E}^{(r)}\}^2 - 2\mathbf{I} \tag{133}$$

In the trigometric identity $\cos 2\theta = 2(\cos\theta)^2 - 1$, let $\theta = 2^r\beta$ and we get

$$2\cos(2^{r+1}\beta) = \{2\cos(r^2\beta)\}^2 - 2 \tag{134}$$

Now $2\cos(2^r\beta)$ is the Chebyshev polynomial of order 2^r for which the zeros are

$$\alpha_j^{(r)} = 2\cos\left(\pi\frac{2^{j-1}}{2^{r+1}}\right) \qquad j = 1, 2, \dots 2^r \tag{135}$$

by analogy, $\mathbf{E}^{(r)}$ is expressible as a product of linear factors

$$\mathbf{E}^{(r)} = \prod_{j=1}^{2^r}\left[\mathbf{E} - 2\cos\left(\pi\frac{2^{j-1}}{2^{r+1}}\right)\mathbf{I}\right] \tag{136}$$

and (132) takes the form

$$(\mathbf{E}-\alpha_1\mathbf{I})(\mathbf{E}-\alpha_2\mathbf{I})\dots(\mathbf{E}-\alpha\mathbf{I})\psi_i = \mathbf{g}_i^{(n)}$$

The matrix is factorized and, therefore, the method of Subsection 8.3 can be applied to obtain, for instance ψ_2, and from (131) and the same method gives $\psi_4, \psi_6, \dots$

Having solved for the even numbered fields, the odd numbered ones are obtained from the original system as systems of $M \times M$ equations.

REFERENCES

(a) General

Kreiss, H. and J. Oliger, 1973: Methods for the approximate solution of time dependent problems. WMO/ICSU Joint Organising Committee, GARP Publications Series No. 10, 107 pp.

Mesinger, F. and A. Arakawa, 1976: Numerical methods used in atmospheric models. WMO/ISCU Joint Organising Committee, GARP Publications Series No. 17, Volumes I and II, pp 64 and 499.

(b) Specific

Bates, J. R. and A. McDonald, 1982: Multiply-upstream, semi-Lagrangian advective schemes: analysis and application to a multi-level primitive equation model. Mon. Wea. Rev., 10, 1831–1842.

Carpenter, K. M., 1981: The accuracy of Gadd's modified Lax–Wendroff algorithm for advection. Quart. J. R. Met. Soc., 107, 468–70.

Collins, W. G., 1983: An accuracy variation of the two-step Lax–Wendroff integration of horizontal advection. Quart. J. R. Met. Soc., 109. 255–261.

Crowley, W.P., 1968: Numerical advection experiments. Mon. Wea. Rev., 96, 1-11

Cullen, M. J. P., 1979: The finite element method. GARP Publication Series, No. 17, Vol. II. 302–337.

Gadd, A. J., 1978: A numerical advection scheme with small phase speed errors. Quart. J. R. Met. Soc., 104, 569–582.

Leslie, L. M. and B. J. McAvaney, 1973: Comparative test of direct and iterative methods for solving Helmholtz-type equations. Mon. Wea. Rev., 101, 235–239.

Machenhauer, B., 1979: The spectral method. GARP Publication Series No. 17, Vol. II, 124–275.

Pudykiewicz, J. and A. Staniforth, 1984: Some properties and comparative performance of the semi-Lagrangian method of Robert in the solution of the advection-diffusion equation. Atmosphere–Ocean, 22, 283–308.

Ritchie, H., 1986: Eliminating the interpolation associated with the semi-Lagrangian scheme. Mon. Wea. Rev., 114, 135-146.

Robert, A., 1981: A stable numerical integration scheme for the primitive meteorological equations. Atmosphere–Ocean, 19, 35-46.

Robert, A., 1982: A semi-Lagrangian and semi-implicit numerical integration scheme for the primitive meteorological equations. J. Meteor. Soc. Japan, 60, 319-325.

Strong, G. and G. J. Fix, 1973: An analysis of the finite element method. Prentice-Hall Series in Automatic Computation, Prentice-Hall, 306 pp.

Temperton, C., 1977: Direct methods for the solution of the discrete Poisson equation: some comparisons. ECMWF Research Dept. Internal Report No. 13.

Vichenevetsky, R. and J. B. Bowles, 1982: Fourier analysis of numerical approximations of hyperbolic equations. SIAM, Philadelphia.