NUMERICAL METHODS

Professor Antonio Navarra
by Niccolò Zanotti
University of Bologna
Updated October 18, 2024

2024/2025

Contents

0.1 Introduction, general remarks about the grid point method 2
0.1.1 **Historical introduction** 2
0.1.2 **Methods for the numerical solution of the equations of motion** 3
0.1.3 **Basic elements of the grid point method** 3
0.1.4 **Finite difference schemes** 5
0.1.5 **Convergence** 8
0.1.6 **Stability** 9
0.2 Time Differencing Schemes 14
0.2.1 **Definitions of some schemes** 14
0.2.2 **Properties of schemes applied to the oscillation equation** . 17
0.2.3 **Properties of schemes applied to the friction equation** . . . 30
0.2.4 **A combination of schemes** 32

0.1 Introduction, general remarks about the grid point method

In this chapter, following a short historical introduction on the development and use of numerical methods in atmospheric models, methods available for numerical solution of the differential equations governing the atmosphere will be briefly reviewed. Then, basic elements of the finite difference method for solving these equations will be introduced. Finally, the concept of stability of finite difference equations, and methods for testing the stability of such equations, will be discussed at some length.

0.1.1 Historical introduction

It is considered that Wilhelm Bjerknes (1904) was the first to point out that the future state of the atmosphere can in principle be obtained by an integration of differential equations which govern the behaviour of the atmosphere, using as initial values fields describing an observed state of the atmosphere. Such an integration performed using numerical methods is called numerical weather prediction. When, however, a numerical integration is performed starting from fictitious initial fields, it is called numerical simulation.

A first practical attempt at a numerical weather prediction was made by Richardson. After very tedious and time-consuming computations, carried out mostly during the First World War, Richardson obtained a totally unacceptable result. Despite this, he described his method and results in a book (Richardson, 1922), and this is today one of the most famous in meteorology.

The wrong result obtained by Richardson, and his estimate that 64,000 men are necessary to advance the calculations as fast as the weather itself is advancing, left some doubt as to whether the method would be of practical use. A number of developments that followed, however, improved the situation. Courant, Friedrichs and Lewy (1928) found that space and time increments in integrations of this type have to meet a certain stability criterion. Mainly due to the work of Rossby in the late 1930's, it became understood that even a rather simple equation, that describing the conservation of absolute vorticity following the motion of air particles, suffices for an approximate description of large-scale motions of the atmosphere. Finally, in 1945, the first electronic computer ENIAC (Electronic Numerical Integrator and Computer) was constructed. The absolute vorticity conservation equation, and this first electronic computer, were used by Charney, Fjortoft and von Neumann in the late 1940's for the first successful numerical forecast (Charney et al., 1950).

Much faster computers, and improved understanding of computational problems, now also enable long-term integrations of the basic primitive equations. It is generally considered that integration of the primitive equations enables easier incorporation of various physical processes than the integration of modified equations, that is, integration of the divergence and vorticity equations. Thus, it is mostly the primitive equations that are used today for practical numerical forecasting by meteorological services. Charts obtained by numerical forecasting are used by synopticians in these services as the principal basis for decisions on forecasts issued for public use.

A number of research groups have been actively engaged for more than a decade in development of models for the numerical simulation of the general circulation of the atmosphere. In such simulations starting from a fictitious initial state, e.g. an isothermal and motionless atmosphere, is often considered to be an advantage for the experiments. It enables a test of the ability of the computational and physical schemes of the model to simulate an atmosphere with statistical properties similar to those of the real atmosphere, with no, or not much, prior information on these properties.

Numerical models are also very frequently developed for studies of some smaller-scale atmospheric phenomena. Foremost among these are studies of the cumulus convection problem, and simulation of processes within the planetary boundary layer. In this text, however, we shall primarily have in mind the application of numerical methods to prediction and simulation of large-scale atmospheric motions.

0.1.2 Methods for the numerical solution of the equations of motion

Numerical solution of the equations of motion today in most cases is performed using the *grid point method*. In this method a set of points is introduced in the region of interest and dependent variables are initially defined and subsequently computed at these points. This set of points is called the *grid*. The words *mesh or lattice* are also used. It is necessary to have the grid points at fixed locations in the horizontal. This means that, according to the Eulerian system of equations, space and time coordinates are chosen as independent variables.

A number of attempts have been made to develop atmospheric models using an approach which is at least partly Lagrangian. Serious difficulties are encountered when a straightforward numerical integration of the Lagrangian system of equations is undertaken. However, it is possible to construct methods with some Lagrangianproperties ; for example, to have some or all of the computation points moving with the fluid. In hydrodynamics a number of such methods have proved to be very useful, especially for some problems which are not amenable to treatment by a strictly Eulerian technique (e.g. Harlow and Amsden, 1971). However, in meteorology the performance of Lagrangian or semi-Lagrangian models that have so far been developed has not been quite satisfactory. A discussion of one way of constructing a Lagrangian model, and a review of earlier attempts, can be found in a paper by Mesinger (1971).

Another possible approach is to express the spatial dependence of the variables in terms of a series of orthogonal functions, and then substitute this into the governing equations. In this way the equations reduce to a set of ordinary differential equations, so that the coefficients of the series can be computed as functions of time. This is the *spectral method* of solving the governing equations. Until relatively recently it was considered that in effiÂciency the spectral method could not be competitive with the grid point method. But the use of the fast Fourier transform has completely changed the situation and investigation of spectral methods is now the subject of intensive research.

In the following we shall consider the technique of using the grid point method, and the problems associated with it, using grid of computation points fixed in space. This is the most direct way of solving the equations of motion numerically. Furthermore, knowledge of this method is necessary for the investigation and understanding of the relative merits of other alternatives mentioned in this section.

0.1.3 Basic elements of the grid point method

With the grid point method, the most common way of solving the governing equations is to find approximate expressions for derivatives appearing in the equations. These approximate expressions are defined using only values of the dependent variables at the grid points, and at discrete time intervals. Thus, they are formed using differences of dependent variables over finite space and time intervals; for that reason this approach is called the *finite difference method*. The approximations for derivatives are then used to construct a system of algebraic equations that approximates the governing partial differential equations. This algebraic system is considered valid at each of the interior grid points of the computation region. For the initial time and at space boundary points, additional constraints or equations are defined that approximate the initial and boundary conditions as required by the physics of the problem. The set of algebraic equations obtained in this way is then solved, usually using an electronic computer, by a suitable step-wise procedure.

We shall now consider some basic elements of the finite difference method. For simplicity, we start by considering a function of one independent variable

$$u = u(x)$$

The function u is a solution to a differential equation that we are interested in. We want to find an approximation to this solution in a bounded region R of the independent variable, having a length L. The simplest way of introducing a set of grid points is to require that they divide the region R into an integer number of intervals of equal length

Δx. This length Δx is called the *grid interval*, or *grid length*. Let us denote the number of grid intervals by J. It is convenient to locate the origin of the x axis at the left-hand end of the region R. Thus, we are looking for approximations to $u(\S)$ at discrete points $\S = |\Delta\S$, where j takes integer values 0,1,2,... J. These approximate values we shall denote by

$$u_j = u_j(j\Delta x)$$

Thus, we are interested in finding $J + 1$ values u_j

Knowledge of a discrete set of values u_j, even if the approximations were perfect, offers, obviously, less information than knowledge of the function $u\ (x)$. Let us briefly consider the situation in that respect. We shall very often find it convenient to think of the function $u\ (x)$ as being formed by a sum of its Fourier components, that is

$$u(x) = \frac{a_0}{2} + \sum_{n\geq 1}(\left[a_n \cos\left(2\pi n\frac{x}{L}\right) + b_n \sin\left(2\pi n\frac{x}{L}\right)\right)$$

Now, the available $J + 1$ values u_j do not enable the computation of all of the coefficients $a_n b_n$; rather, they can be used to compute only $J + 1$ different coefficients. A natural choice is to assume that the $J + 1$ values u_j define the near value a_0 and as many as possible of the coefficients of the Fourier components at the long wave length end of the series, that is, coefficients for n = 1,2,3,..., $J/\ 2$. Of these components, the one with the shortest wavelength will have n=J/2, with the wave length

$$\frac{L}{n} = \frac{2L}{j} = \frac{2L}{\frac{L}{\Delta\S}} = 2\Delta\S$$

Having made that choice, we can say that with values u_j at discrete points $\S = j\Delta\S$ it is not possible to resolve waves with wave length shorter than $2\Delta x$.

Now let us consider the differences between values u} that will be used to construct approximations to derivatives. These differences are called *finite differences*. They can be calculated over one or more of the intervals Ax. Depending on the relation of the points from which the values are taken to the point where the derivative is required, they can be *centered or uncentered*. An un-centered difference is, for example, the *forward* difference

$$\Delta u_{\mathrm{i}} = u_{j+1} - u_j$$

More often centered (or *central*) differences are used, such as

$$\hat{\mathrm{I}}'\mathrm{u}_{j+\frac{1}{2}} = u_{i+1} - u_j$$

In a centered difference the difference is between values symmetrical about the point where the difference is being calculated.

One way to construct an approximation to a differential equation is to simply replace the derivatives by appropriate *finite difference quotients*. For example, for the first derivative one can use the approximation

$$\left(\frac{du}{dx}\right)_j \rightarrow \frac{u_{j+1} - u_j}{\Delta x}$$

The finite difference quotient here is, of course, only one of many possible approximations to the first derivative at point j.

If a finite difference quotient, or a more complex expression, is to be used as an approximation to a derivative, it is required, above all, that this approximation be consistent. This means that the approximation should approach the derivative when the grid interval approaches zero. The quotient `3.1` , obviously, has that property.

Important information is obatined when the true solution $u(j\Delta x)$ is substituted into an approximation to the derivative in place of the grid point values u_j, and $u(j\Delta x)$ is expanded in a Taylor series about the central point. For the quotient `3.1` this procedure gives

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

The difference between this expression and the derivative

$$\left(\frac{\mathrm{du}}{\mathrm{dx}}\right)_j$$

being approximated. In this case

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

is called the truncation error of the approximation to the derivative. These are terms that were "truncated off" to form the approximation. The truncation error gives a measure of how accurately the difference quotient approximates the derivative for small values of Δx.

The usual measure of this is the order of accuracy of an approximation. This is the lowest power of Ax that appears in the truncation error. Thus, approximation (3.1) is of the first order of accuracy. We can write

$$\varepsilon = 0(\Delta x)$$

For an approximation to the derivative to be consistent it must, obviously, be at least first order accurate.

0.1.4 Finite difference schemes

The algebraic equation obtained when derivatives in a differential equation are replaced by appropriate finite difference approximations is called a finite difference approximation to that differential equation, or a finite difference scheme. In this section we shall introduce the concepts of consistency, truncation error, and accuracy, for a finite difference scheme.

As an example, we shall use the linear advection equation

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

c is a positive constant.

It describes advection of the variable u at a constant velocity c in the direction of the x axis. The solution to this simple equation can, of course, also be obtained by an analytic method. It will be useful to obtain the analytic solution first, in order to investigate properties of numerical solutions by comparing them against known properties of the true solution.

It is convenient to this end to change from variables x, t to variables, ξ, t with the substitution $\xi = x - ct$.

Using the notation

$$u^{(x,t)} = U(\xi, t)$$

we obtain

$$\frac{\partial u}{\partial t} = \frac{\partial U}{\partial \xi}\frac{\partial \xi}{\partial y} + \frac{\partial U}{\partial t}\frac{\partial t}{\partial t} = -c\frac{\partial U}{\partial \xi} + \frac{\partial U}{\partial t}$$

$$\frac{\partial u}{\partial x} = \frac{\partial U}{\partial \xi}\frac{\partial \xi}{\partial X} + \frac{\partial U}{\partial t}\frac{\partial t}{\partial x}\frac{\partial U}{\partial \xi}$$

Substitution of these expressions into `g4.1` gives

$$\frac{\partial}{\partial t}U(\xi, t) = 0$$

Thus, it is seen that U cannot be a function of t, but can be an arbitrary function of ξ. A solution of g4.1 is, therefore,

$$u = f(x - ct)$$

where f is an arbitrary function. This, we see, is the general solution of the advection equation g4.1, since it can satisfy an arbitrary initial condition

$$u(x, 0) = F(x)$$

Thus,

$$u = F(x - ct)$$

is the solution of g4.1 satisfying the initial condition g4.3.

For a physical interpretation, it is often convenient to consider the solution in the x, t plane. In the present case, we see that the solution takes constant values along the straight lines

$$x - ct = const.$$

These lines are the *characteristics* of the advection equation ; one of them is shown in figg:1. We can say that the solution propagates along the characteristics.

Let us now construct a scheme for finding an approximate solution to g4.1 using the grid point method.

We are now looking only for an approximate solution at the discrete points in the (x, t) plane formed by the grid shown in Fig. figg:1. The approximate solution at a point $(j\Delta x, n\Delta t))$ is denoted by u_j^n.

The behaviour of the true solution, which propagates along characteristics in the (x,t) plane, suggests constructing the approximate equation by replacing the time derivative by a forward difference quotient, and the space derivative by a backward difference quotient. In this way we obtain the scheme

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

This scheme could be called a forward and upstream scheme, the latter word indicating the position of the point $j - 1$ relative to the advection velocity. It is, of course, only one of many possible consistent finite difference schemes for the differential equation. There are many schemes which approach the differential equation when the increments $\Delta x, \Delta t$ approach zero.

Since for small values of $\Delta x, \Delta t$ a finite difference equation approximates the corresponding differential equation, we can expect that its solution will be an approximation to the solution of that equation. We shall call solutions given by finite difference schemes numerical solutions. There are, of course, both approximate and numerical solutions obtained by other methods which will not be considered in this publication. It is most convenient to study the properties of numerical solutions when they can be compared with known solutions of the original differential equation, which we shall refer to as true solutions. The difference between the numerical and the true solution

$$u_j^n - u(j\Delta x, n\Delta t)$$

is the error of the numerical solution.

For obvious reasons, we cannot often expect to know the error of the numerical solution. However, we can always find a measure of the accuracy of the scheme by substituting the true solution $u\,(j\Delta x, n\Delta t)$ of the equation, into the numerical scheme. Since the true solution will not satisfy the numerical equations exactly, we will have to add an additional term to keep the equation valid. Let us denote this term by ε . For example, in the case of scheme g4.5 this procedure gives

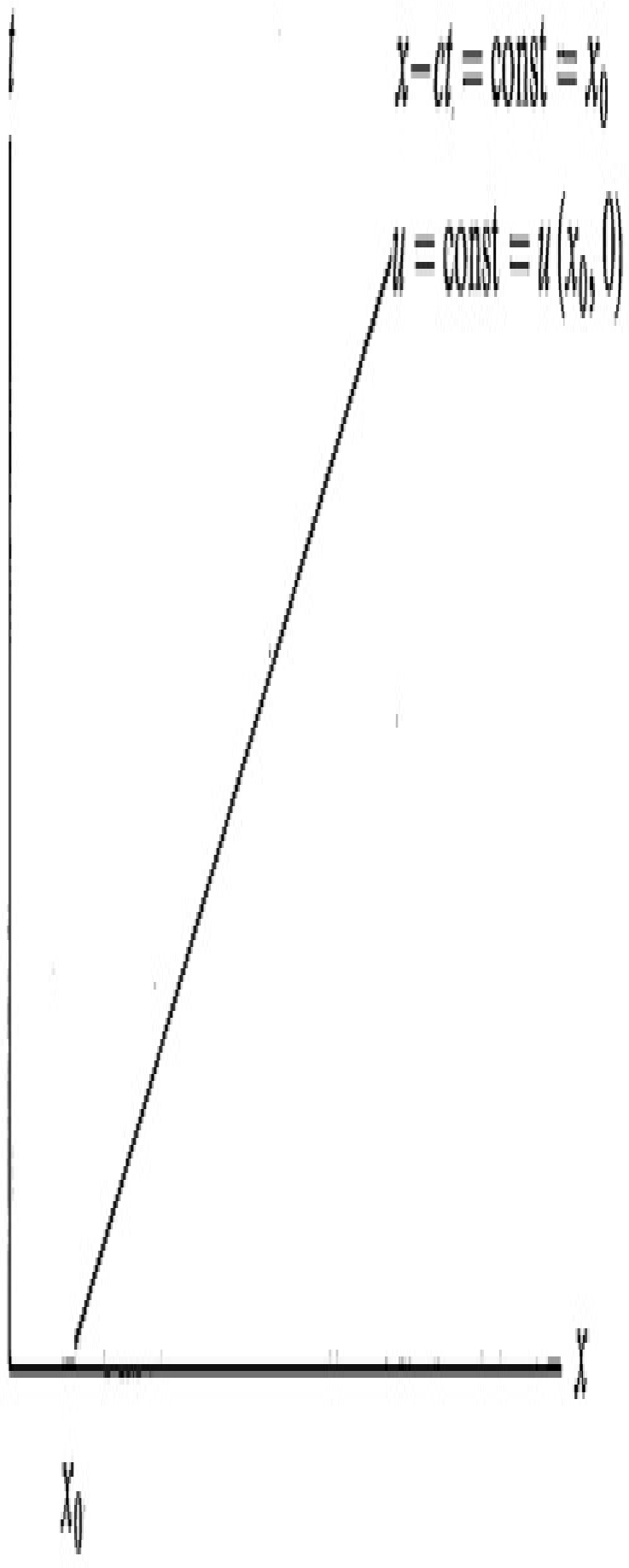


Figure 1: One of the characteristics of the linear advection equation `g4.1`

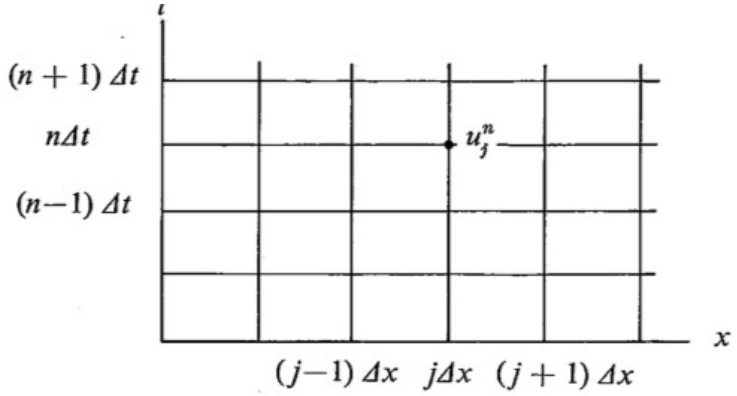


Figure 2:

$$\frac{u\left(j\Delta x,(n+1)\Delta t\right)-u\left(j\Delta x,\Delta \mathrm{nt}\right)}{\Delta t}+c\frac{u\left(j\Delta x,n\Delta t\right)-u\left((j-1)\,\Delta x,\Delta t\right)}{\Delta x}=\varepsilon$$

The term e we shall call the truncation error of the finite difference scheme. It shows how closely the true solution satisfies the equation of the scheme, and, thus, gives a measure of the accuracy of the scheme.

We can obtain a more useful form for the expression for the truncation error by performing a Taylor series expansion of the true solution about the central space and time point. Using the original differential equation to eliminate the leading term we obtain the truncation error `g4.7` as

$$\varepsilon=\frac{1}{2}\frac{\partial^{2u}}{\partial t^{2}}\Delta t+\frac{1}{6}\frac{\partial^{3}u}{\partial t^{3}}\left(\Delta t\right)^{2}+\ldots-c(\frac{1}{2}\frac{\partial^{2}u}{\partial x^{2}}\Delta x-\frac{1}{6}\frac{\partial^{3}u}{\partial t^{3}}\left(\Delta x\right)^{2}+\ldots)$$

As before, these are the terms that were "truncated off" to make the differential equation reduce to our finite difference scheme.

In the same way as for an approximation to the derivative, the order of accuracy of a finite difference scheme is the lowest power of Ax and At that appears in the truncation error. Thus, scheme `g4.5` is first order accurate. We can write

$$\varepsilon=O\left(\Delta t\right)+O\left(\Delta x\right)$$

or

$$\varepsilon=O\left(\Delta x,\Delta t\right).$$

It is useful to make a distinction between orders of accuracy in space and in time, especially when the lowest powers of Δx and Δt are not the same. As before, a necessary condition for consistency of a scheme is that it be at least of the first order of accuracy.

0.1.5 Convergence

The truncation error of a consistent scheme can be made arbitrarily small by a sufficient reduction of the increments Δx and Δt. Unfortunately, we cannot be sure that this will also result in a reduction of the error of the numerical solution. For that reason, we return to consideration of the error

$$u_{j}^{n}-u\left(j\Delta x,n\Delta t\right).$$

Following Richtmyer and Morton (1967) we ask two questions:

(a) What is the behavior of the error $u_{j}^{n}-u\left(j\Delta x,n\Delta t\right)$ when, for a fixed total time $n\Delta t$ the increments $\Delta x,\Delta t$ approach zero ?

(b) What is the behavior of the error $u_{j}^{n}-u\left(j\Delta x,n\Delta t\right)$ when, for fixed values of $\Delta x,\Delta t$, the number of time steps n increases?

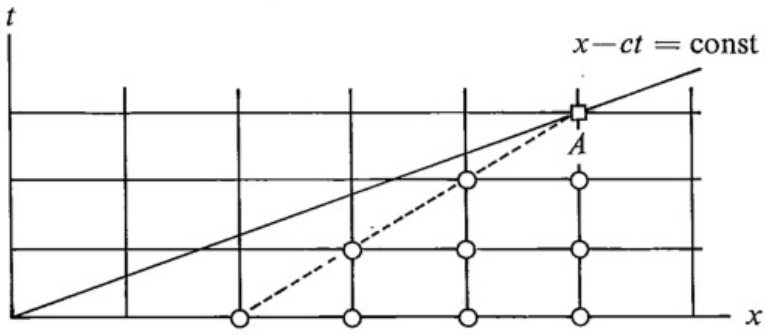


Figure 3:

The answer to the first of these questions depends on the convergence of the numerical solution : if the error approaches zero as the grid is refined (as $\Delta x, \Delta t \to 0$) the solution is called convergent. If a scheme gives a convergent solution for any initial conditions, then the scheme also is called convergent.

Consistency of a scheme does not guarantee convergence. We shall illustrate this by a simple example. We still consider the scheme `g4.5`; its truncation error `g4.8` approaches zero as the grid is refined, and, therefore, this is a consistent scheme. But consider the numerical solution, when the grid lines and characteristics are as shown in `figg:3`. The characteristic passing through the grid point taken as the origin in this example passes through another grid point, A, denoted by a square. Thus, the true solution at A, is equal to the initial value at the origin. However the numerical solution given by `g4.5` A is computed using the values at points denoted by circles. The shaded domain, including all of these points, is called the domain of dependence of the numerical scheme. The grid point at the origin is outside that domain, and, thus, cannot affect the numerical solution at A_0. Therefore, the error can be arbitrarily great. If the space and time steps were reduced by the same relative amount, say to one half of their values in the figure, the domain of dependence would still remain the same, and this situation would not change. Thus, as long as the ratio of the steps Δx and Δt remains the same, refinement of the grid cannot bring about a reduction in the error of the numerical solution.

$$x - ct = const$$

A necessary condition for convergence of a scheme is, obviously, that the characteristic defining the true solution at a grid point is inside the domain of dependence of the numerical solution at that point. In our example, this will happen when the slope of the characteristics is greater than the slope of the dashed line bounding the domain of dependence, that is, when

$$c\Delta t \leq \Delta x$$

Thus, this is a necessary condition for convergence of `g4.5`.

0.1.6 Stability

The answer to the second question raised at the beginning of the Section `Section1.5` depends on the stability of the numerical solution. A rigorous definition of stability employs the concepts of functional analysis, and refers to the boundedness of the numerical solution only (e.g. Richt-myer and Morton, 1967). The difficulties in defining stability are caused by the fact that the true solution, in general, does not have to be bounded. However, when we know that the true solution is bounded, as in the equations we are interested in here, we can use a definition referring to the boundedness of the error $u_j^n - u\left(j\Delta x, n\Delta t\right)$. We sat that a solution u_j^n is stable if this error remains bounded as n increases, for fixed values of $\Delta x, \Delta t$. As before, we say that a finite difference scheme is stable if it gives a stable solution for any initial conditions.

Stability of a scheme is a property of great practical significance. There are consistent schemes, of a high order of accuracy, that still give solutions diverging unacceptably fast from the true solution. Thus, conditions for stability, if any, should be known. There are three methods that can be used to investigate the stability of a scheme, and we shall give an example of each of these methods. We shall do this by considering again the forward and upstream scheme `g4.5`).

Direct method. Since we know that the true solution is bounded, it suffices to test the boundedness of the numerical solution. The scheme `g4.5` can be written as

$$u_j^{n+1} = (1-\mu)\, u_j^n + \mu u_{j-1}^n$$

where

$$\mu \equiv C\Delta t/\Delta x.$$

If $1-\mu \geq 0$, which happens to be also the necessary condition for convergence, we will have

$$\left|u_j^{n+1}\right| \leq (1-\mu)\left|u_j^n\right| + \mu\left|u_{j-1}^n\right|.$$

We can apply this at the point where at time level *n + 1* , $\left|u_j^{n+1}\right|$ is a maximum, $Max_{(j)}\left|u_j^{n+1}\right|$. The right side of `g6.2` can only be increased by replacing $\left|u_j^n\right|$ and $\left|u_{j-1}^n\right|$ by the maximum value at level n, $Max_{(j)}\left|u_j^n\right|$.The two terms on the right side can then be added, and we obtain

$$Max_{(j)}\left|u_j^{n+1}\right| \leq Max_{(j)}\left|u_j^n\right|$$

This proves the boundedness of the numerical solution. Hence, $1-\mu \geq 0$ is seen to be a sufficient condition for stability of `g6.1`.

This direct testing of the stability is simple. UnfortunÂately, as might be anticipated from the argument, it is successful only for a rather limited number of schemes.

Energy method. This method is of a much wider appliÂcability, and can be used even for nonlinear equations. If we know that the true solution is bounded, we test whether

$$\sum_j \left(u_j^n\right)^2$$

is also bounded.

If it is, then every value u_j^n must be bounded, and the stability of the scheme has been proved. The method is called the energy method since in physical applications u^2 is often proporÂtional to some form of energy.

Of course, there are examples when this is not so.

Squaring `g6.1` and summing over j we obtain

$$\sum_j \left(u_j^{n+1}\right)^2 = \sum_j \left[(1-\mu)^2\left(u_j^n\right)^2 + 2\mu(1-\mu)\, u_j^n u_{j-1}^n + \mu^2\left(u_{j-1}^n\right)^2\right]$$

We shall assume a cyclic boundary condition, for example $u_{-1} \equiv u_j$, then

$$\sum_j \left(u_{j-1}^n\right)^2 = \sum_j \left(u_j^n\right)^2$$

Now, using Schwarz inequality

$$\sum ab \leq \sqrt{\sum a^2}\sqrt{\sum b^2}$$

and `g6.4`, we can write

$$\sum_j u_j^n u_{j-1}^n \leq \sqrt{\sum_j \left(u_j^n\right)^2} \sqrt{\sum_j \left(u_{j-1}^n\right)^2} = \sum_j \left(u_j^n\right)^2$$

Using g6.4 and g6.5 we see that, if $1-\mu \geq 0$, g6.3 gives the inequality

$$\sum_j \left(u_j^{n+1}\right)^2 \leq \left[(1+\mu)^2 + 2\mu\,(1-\mu) + \mu^2\right] \sum_j \left(u_j^n\right)^2$$

or

$$\sum_j \left(u_j^{n+1}\right)^2 \leq \sum \left(u_j^n\right)^2$$

Thus, $1-\mu \leq 0$, coupled with the cyclic boundary condition, is proved to be a sufficient condition for stability of g6.1.

Von Neumann's method. Von Neumann's, or the Fourier series method is the most frequently used method.

We will usually not be able to use it to test the stability of nonlinear equations, and will have to resort to the analysis of their linearized versions. A solution to a linear equation, however, can be expressed in form of a Fourier series, where each harmonic component is also a solution. Thus, we can test the stability of a single harmonic solution ; stability of all admissible harmonics will then be a necessary condition for stability of the scheme.

For an illustration of this method, it is useful first to obtain an analytic solution of the equation g4.1

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

in the form of a single harmonic

$$u\,(x,t) = Re\left\lfloor U\,(t)\, e^{\mathrm{ikx}}\right\rfloor$$

Here *U(t)* is the wave amplitude, and k the wave number. Substituting this into the preceding equation we obtain

$$\frac{\mathrm{dU}}{dt} + ikU = 0$$

Thus, the problem of solving a partial differential equation has been reduced to that of solving this ordinary differential equation. Its solution is

$$U(t) = U(0)e^{-ikct}$$

where U(0) is the initial value of the amplitude. Hence, the desired harmonic solution is

$$u\,(x,t) = Re\left[U\,(0)\, e^{\mathrm{ik}(x-ct)}\right]$$

Each wave component is, thus, advected at a constant velocity c along the x axis with no change in amplitude.

Returning to the von Neumann method, we now look for an analogous solution of the finite difference equation g6.1. Into this equation we substitute a solution of the form

$$u_j^n = Re\left[U^{(u)} e^{ikj\Delta x}\right]$$

Here U^{n} is the amplitude at time level n. This substitution shows that g6.8 is a solution provided that

$$U^{n+1} = (1-\mu)\, U^{(n)} + \mu U^{(n)} e^{-ik\Delta x}$$

An equation of this kind enables analysis of the behavior of the amplitude U^{n} as n increases.

To this end we define an *amplification factor* $|\lambda|$ by

$$U^{n+1} \equiv \lambda U^n$$

This gives

$$|U^{n+1}| = |\lambda||U^{(n)}|.$$

For each harmonic solution g6.8 to be stable it is required that

$$|U^n| = |\lambda|^n |U^{(0)}| \leq B$$

where B is a finite number. This gives

$$n \log |\lambda| \leq \log \frac{B}{U^0} \equiv B'$$

where B' is a new constant. Since $n = \frac{t}{\Delta t}$, the necessary condition for stability becomes

$$\log |\lambda| \leq \frac{B'}{t} \Delta t$$

Now, suppose that we require boundedness of the solution for a finite time t. Condition g6.11 can then be written as

$$\log |\lambda| \leq 0(\Delta t)$$

If we now define

$$|\lambda| \equiv 1 + \delta$$

we see, in view of the power series expansion of $\log(1+\delta)$, that the stability condition obtained is equivalent to

$$\delta \leq 0\,(\Delta t)$$

or

$$|\lambda| \leq 1 + O(\Delta t)$$

This is the *von Neumann necessary condition for stability.*

The von Neumann condition allows an exponential, but no faster, growth of the solution. This, of course, is needed to analyze cases when the true solution grows exponentially. However, when we know that the true solution does not grow, as in our example g6.7, it is customary to replace g6.12 by a sufficient condition

$$|\lambda| \leq 1$$

This condition is much less generous than that required by the original definition of stability. Returning to our example, substitution of g6.10 into g6.9 gives

$$\lambda = 1 - \mu + \mu e^{-ik\Delta x}$$

From this we obtain

$$\lambda^2 = 1 - 2\mu\,(1-\mu)\,(1 - \cos k\Delta x)$$

and, therefore, $1 - \mu \geq 0$ is again found to be a sufficient condition for stability of g6.1.

An equation such as g6.15 gives further information about the behavior of the numerical solution. This can be obtained by studying the variation of $|\lambda|$ with μ for various

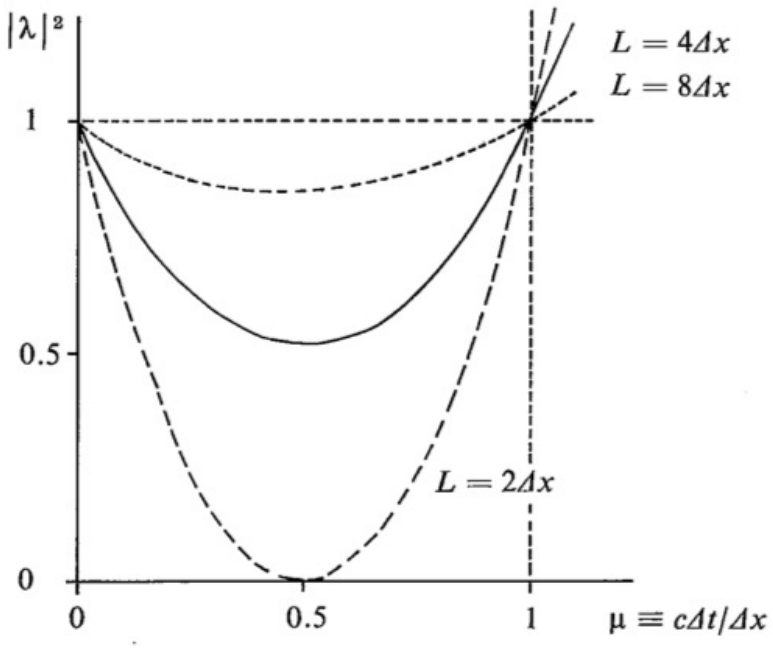


Figure 4:

fixed values of $k\Delta x$. To this end we plot the $|\lambda|^2$ curves ; `g6.15` shows that in the present case all of these curves are parabolas. Furthermore, recall that the minimum resolvable wave length is $2\Delta x$. Thus, the maximum value that wave number k can take is $\frac{\pi}{\Delta x}$. We thus plot the $|\lambda|^2$ curves for this maximum value $k = \frac{\pi}{\Delta x}$ (or wave length $L = 2\Delta x$, and for half this value, $k = \frac{\pi}{2\Delta x(L-4\Delta x)}$, and a quarter of this value, $k = \frac{\pi}{4\Delta x(L-8\Delta x)}$.

The first derivative

$$\frac{d|\lambda|^2}{d\mu} = -2(1-2\mu)(1-\cos k\Delta x)$$

shows that all the $|\lambda|^2$ curves have minima at $\mu = \frac{1}{2}$. This information, in addition to calculation of the ordiÂnales of `g6.15` at $\mu = \frac{0.1}{2}$ and 1, suffices for sketching

the graphs of the $|\lambda|^2$ curves as shown in `figg:1`. In general, as the wave length L increases, that is, as k approaches zero, the amplification factor approaches unity for any value of the parameter μ.

The figure shows that within the stable region the scheme is damping for all values $\mu \leq 1$. The damping increases as the wave length decreases. Since the true solution has a constant amplitude, this damping reveals an error due to finite differencing. We see that this error increases as the wave length decreases. At the shortest resolvable wave length, $L = 2\Delta x$, the error may be very great unless At is extremely small. It is even possible for this wave to be completely removed after only a single time step ! The dependence of the error on wave length, as seen here, might have been anticipated by considering representation of harmonics of various wave lengths by the finite difference grid. The shortest resolvable wave, with only two data points per wave length, is very poorly represented ; as the wave length increases, the representaÂtion by a finite difference grid improves, and approaches the continuous representation as the wave length tends to infinity.

There exists a wealth of more precise definitions of stability and convergence, as well as stability criteria. For a further discussion of these subjects, and of the relation between the properties of stability and converÂgence, the interested reader is referred to the book by Richtmyer and Morton (1967) and to the publication by Kreiss and Oliger (1973). However, for application of numerical methods to atmospheric models, it is more important to discuss other problems than to refine the stability and convergence concepts beyond the outline given here. These numerical problems, such as phase speed errors

and computational dispersion, nonlinear instability, effect of the space-time grid on the properties of the numerical solution, and, also the ideas behind and properties of the great variety of schemes that are currently being used in atmospheric models, will be discussed in the remaining chapters of this publication.

0.2 Time Differencing Schemes

In this chapter we consider ordinary differential equaÂtions with one dependent and one independent variable. Although atmospheric models are essentially always models for solving a complex set of partial differential equations, in some formulations the numerical solution of ordinary differential equations forms an important part of the computational procedure. For instance in spectral models the governing partial differential equaÂtions reduce to a set of ordinary differential equations for the expansion coefficients as dependent variables. A set of ordinary differential equations will also be obtained if a Lagrangian method is used, in which the computaÂtional points move with the fluid. But, most of all, schemes for solving ordinary differential equations are of interest here since they are often used without modiÂfication to construct approximations to the time deriÂvative terms in the governing partial differential equaÂtions. Knowledge of the properties of schemes for solving ordinary differential equations will then be used in invesÂtigating the properties of more complex schemes for solving the partial differential equations.

With that in mind, we shall here first define some of the schemes that will be interesting to analyze. Then we shall investigate the behaviour of numerical solutions obtained when these schemes are used for two specific ordinary differential equations: the oscillation (or frequency) equation, and the friction equation. These equations will serve as prototypes for later extension of the results to advection, gravity-inertia wave, and diffusion processes within the atmospheric primitive equations.

0.2.1 Definitions of some schemes

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

Another reason for not requiring a scheme of high accuracy for approximations to the time derivative terms is that, in order to meet a stability requirement of the type discussed in the preceding chapter, it is usually necessary to choose a time step significantly smaller than that required for adequate accuracy. With the time step usually chosen, other errors, for example in the space differencing, are much greater than those due to the time differencing. Thus, computational effort is better spent in reducing these other errors, and not in increasing the accuracy of the time differencing. This, of course, does not mean that it is not necessary to consider carefully the properties of various possible time differencing schemes. Accuracy, is only one important consideration in choosing a scheme.

To define some schemes, we consider the equation

$$\frac{\mathrm{dU}}{\mathrm{dt}} = f(U,t) \quad U = U(t)$$

The independent variable t is here called time. We divide the time axis into segments of equal length Δt. We shall denote by $U^{(n)}$ the approximate value of U at time $n\Delta t$. We assume that we know at least the first of the values $U^{(n)}$, $U^{(n-1)}$ ... and we want to construct a scheme for computation of an approximate value $U^{(n+1)}$. These are many possibilities.

Two level schemes

These are schemes that relate values of the dependent variable at two time levels : n and n + 1. Only a two level scheme can be used to advance an integration over the first time step, when just a single initial condition is available. With such a scheme we want to approximate the exact formula

$$U^{(n+1)} = U^{(n)} + \int_{n\Delta t}^{(n+1)\Delta t} f\,(U,t)\,\mathrm{dt}$$

We shall first list several schemes which do not use an iterative procedure.

Euler (or forward) scheme This is the scheme

$$U^{(n+1)} = U^{(n)} + \Delta t \bullet f^{(n)}$$

where

$$f^{(n)} \equiv f\left(U^{(n)}, n\Delta t\right)$$

The truncation error of this scheme is $O\,(\Delta t)$. Thus, this is a first order accurate scheme. For the integrand in `h1.2` we have here taken a constant value equal to that at the lower boundary of the time interval. Thus, f in `h1.3` is not centered in time, and the scheme is said to be uncentered. In general, uncentered schemes will be found to be of the first order of accuracy, and simple centered schemes to be of the second order of accuracy.

Backward scheme We can also take a constant value of f equal to that at the upper boundary of the time interval. We then obtain

$$U^{(n+1)} = U^{(n)} + \Delta t f^{(n+1)}$$

If, as here, a value of f depending on $U^{(n+1)}$ appears in the difference equation, the scheme is called implicit. For an ordinary differential equation, it may be simple to solve such a difference equation for the desired value $U^{(n+1)}$. But, for partial differential equations, this will require solving a set of simultaneous equations, with one equation for each of the grid points of the computation region. If a value of f depending on $U^{(n+1)}$ does not appear in the difference equation the scheme is called explicit. The truncation error of `h1.4` is also is $0\,(\Delta t)$.

Trapezoidal scheme If we approximate f in `h1.2` by an average of the values at the beginning and the end of the time interval, we obtain the trapezoidal scheme

$$U^{(n+1)} = U^{(n)} + \frac{1}{2}\Delta t\left(f^{(n)} + f^{(n+1)}\right)$$

This is also an implicit scheme. Its truncation error, however, is $O\left[\left(\Delta t^2\right)\right]$.

To increase the accuracy or for other reasons we can also construct iterative schemes. Two schemes that we will now define are constructed in the same way as :eq: h1.4 and :eq: h1.5, except that an iterative procedure is used to make them explicit.

Matsuno (or Euler-backward) scheme With this scheme a step is made first using the Euler scheme ; the value of U obtained for time level n + 1 is then used for an approximation to $f^{(n+1)}$, and this approximate value $f^{*(n+1)}$ is used to make a backward step. Thus,

$$U^{*(n+1)} = U^{(n)} + \Delta t f^{(n)}$$
$$U^{n+1} = U^{(n)} + \Delta t f^{*(n+1)}$$

where

$$f^{*(n+1)} \equiv f\left(U^{*(n+1)}, (n+1)\,\Delta t\right)$$

This is an explicit scheme, of the first order of accuracy.

Heun scheme Here, in much the same way, an approximation is constructed to the trapezoidal scheme. Thus,

$$U^{*(n+1)} = U^{(n)} + \Delta t f^{(n)}$$

$$U^{(n+1)} = U^{(n)} + \frac{1}{2}\Delta t\left(f^{(n)} + f^{*(n+1)}\right)$$

Thus, this is also an explicit scheme. It is of the second order of accuracy.

Three level schemes

Except at the first step, one can store the value $U^{(n-1)}$, and construct schemes taking advantage of this additional information.

These are three level schemes. They may approximate the formula

$$U^{(n+1)} = U^{(n-1)} + \int_{(n-1)\Delta t}^{(n+1)\Delta t} f\,(U,t)\,\mathrm{dt}$$

or, they can use the additional value $U^{(n-1)}$ to make a better approximation to f in `h1.2`.

Leapfrog scheme The simplest way of making a centered evaluation of the integral in `h1.8` is to take for f a constant value equal to that at the middle of the interval $2\Delta t$. This gives the leapfrog scheme

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

Its truncation error is $O\left[\left(\Delta t^2\right)\right]$. This is probably the scheme most widely used at present in atmospheric models. It has also been called the "mid-point rule", or "step-over" scheme.

Adams-Bashforth scheme The scheme that is usually called the Adams-Bashforth scheme in the atmoÂspheric sciences is, in fact, a simplified version of the original Adams-Bashforth scheme, which is of the fourth order of accuracy. The simplified version is obtained when f in `h1.2` is approximated by a value obtained at the centre of the interval Δt by a linear extrapolation using values $f^{(n-1)}$ and $f^{(n)}$. This gives

$$U^{(n+1)} = U^{(n)} + \Delta t\left(\frac{3}{2}f^{(n)} - \frac{1}{2}f^{(n-1)}\right)$$

This also is a second order accurate scheme.

There are many other rather obvious possibilities. For example, one can approximate the integral in `h1.8` using Simpson's rule, that is, by fitting a parabola to the values $\mathrm{f}^{(n-1)}$, $f^{(n)}$ and $\mathrm{f}^{(n+1)}$.

The implicit scheme obtained in this way is called the Milne-Simpson scheme. To illustrate the wealth of possible alternatives we note that in a paper by Young (1968) properties of 13 different schemes have been studied. Furthermore, when we are solving a more complicated partial differential equation, or a system of such equations, time (or space-time) differencing schemes can be constructed which are more complex than those which can be defined using the simple equation `h1.1`. Such schemes are widely used in atmospheric models, and some of them will be described in later chapters of this publication.

0.2.2 Properties of schemes applied to the oscillation equation

The stability and other important properties of the time differencing schemes defined in section `Section2.1` depend on the form of the function $f(U,t)$. Thus, in order to discuss these properties we have to prescribe this function. For applications in atmospheric models it is of particular interest to consider the case

$$f \equiv i\omega U$$

that is, the equation

$$\frac{dU}{dt} = i\omega U, U = U(T)$$

Equation `h2.1` we shall call the oscillation equation. The word frequency equation is also used. We allow U to be complex; then `h2.1` can be thought of as representing a system of two equations. The parameter ω is real, and is called the frequency.

It is easy to give some justification for our interest in the equation `h2.1`. As an example, recall that the harmonic component

$u(x,t) = R\ \mathrm{e}\left[U(t)\, e^{\mathrm{ikx}}\right]$

is a solution of the linear wave equation

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial t} = 0$$
$$c = const.$$

provided that

$$\frac{\mathrm{dU}}{\mathrm{dT}} + ikcU = 0$$

This ordinary differential equation reduces to `h2.1` if we substitute $\omega = -kc$

As another simple example we can consider the accelÂeration and Coriolis terms of the horizontal component of the equation of motion of the atmosphere, that is

$$\frac{\mathrm{du}}{\mathrm{dt}} = fv, \frac{\mathrm{dv}}{\mathrm{dt}} = -fu$$

If we define

$$U \equiv u + iv$$

we can write these two equations as

$$\frac{\mathrm{dU}}{\mathrm{dt}} = -ifU$$

This again reduces to `h2.1`, this time if we substitute $\omega = -f$.

Since there are many more important types of wave motion, we can hope that results obtained by a study of `h2.1` will be much more general. It can, indeed, be shown (e.g. Young, 1968) that the equation `h2.1` can be obtained from a rather general linearized system of governing equations, describing a number of types of wave motion in the atmosphere.

The general solution of `h2.1` is

$U(t) = U(0)\, e^{i\omega t}$
or, for discrete values $t = n\Delta t$

$$U(n\Delta t) = U(0)e^{in\omega\Delta t}$$

Thus, considering the solution in a complex plane, its argument rotates by $\omega\Delta t$ in each time step and Δt there is no change in amplitude.

The properties of various schemes when applied to `h2.1` are conveniently analyzed using the von Neumann method. This method, as we have seen, involves defining a variable λ by

$$U^{n+1} \equiv \lambda U^{(n)}$$

We also write

$$\lambda \equiv |\lambda| e^{i\theta}$$

Thus, the numerical solution can formally be written as

$$U^{(n)} = |\lambda|^n U^{(0)} e^{in\theta}$$

We see that θ represents the change in argument (or phase change) of the numerical solution in each time step.

Since we know that the amplitude of the true solution does not change, we shall require $|\lambda| \leq 1$ for stability.

In accordance with this and `h2.5`, we shall say that a scheme is

It will also be instructive to compare the phase change of the numerical solution per time step, θ, with that of the true solution, $\omega\Delta t$. The ratio of these changes, $\frac{\theta}{(\omega\Delta t)}$, is the relative phase change of the numerical solution. Obviously, we can say that a scheme is

For accuracy, therefore, it is desirable to have both the amplification factor and the relative phase speed close to unity. Exceptions to this are so-called "computaÂtional modes", which, as we shall see later, can appear as false solutions superposed on the physical solution. These are solutions that do not approach the true solution as the space and time steps approach zero. If such solutions exist they will each have their own value of the amplification factor. Since they are not an approximaÂtion to the true solution, it is desirable to have their amplitudes as small as possible, that is, to have their amplification factors less than unity.

We shall now discuss the properties of the schemes that have been defined in the preceding section.

Two level schemes

The three non-iterative two level schemes can be described by a single finite difference equation

$$U^{(n+1)} = U^{(n)} + \Delta t \left(\alpha f^n + \beta f^{(n+1)}\right)$$

with a consistency requirement

$$\alpha + \beta = 1$$

Obviously, $\alpha = 1$, $\beta = 1$ for the Euler scheme, $\alpha = 0$, $\beta = 1$ for the backward scheme, and $\alpha = \frac{1}{2}$, $\beta = \frac{1}{2}$ for the trapezoidal scheme.

Applied to the oscillation equation `h2.6` gives

$$U^{(n+1)} = U^{(n)} + i\omega\Delta t \left(\alpha U^n + \beta U^{(n+1)}\right)$$

In order to evaluate λ.we must solve this equation for $U^{(n+1)}$ Denoting, for brevity,

$$p \equiv \omega \Delta t$$

we obtain

$$U^{(n+1)} = \frac{1 + i\alpha p}{1 - i\beta p} U^{(n)}$$

Therefore,

$$\lambda = \frac{1 + i\alpha p}{1 - i\beta p}$$

or,

$$\lambda = \frac{1}{1 + \beta^2 p^2}(1 - \alpha\beta p^2 + ip)$$

Substituting for α and β allows us to investigate the effect of particular schemes. For the Euler scheme we have

$$\lambda = 1 + ip$$

for the backward scheme

$$\lambda = \frac{1}{1 + p^2}(1 + ip)$$

and, for the trapezoidal scheme,

$$\lambda = \frac{1}{1 + \frac{1}{4}p^2}\left(1 - \frac{1}{4}p^2 + ip\right)$$

To test for stability we need to know $|\lambda|$. Since the modulus of a ratio of two complex numbers is equaf to the ratio of their moduli, we can obtain the values of $|\lambda|$ directly from `h2.10`. For the Euler scheme we have

$$|\lambda| = \left(1 + p^2\right)^{\frac{1}{2}}$$

The Euler scheme is, thus, always unstable. It is interestÂing to note that, if Δt is chosen so as to make p relatively small, we have

$$|\lambda| = 1 + \frac{1}{2}p^2 + \ldots$$

This shows that $|\lambda| = 1 + O\left[(\Delta t)^2\right]$ that is, $|\lambda| - 1$ is an order of magnitude less than the maximum allowed by the von Neumann necessary condition for stability. However, experience shows that an indiscriminate use of the Euler scheme for solution of the atmospheric equations leads to amplification at a quite unacceptable rate.

For the backward scheme we obtain

$$|\lambda| = \left(1 + p^2\right)^{-\frac{1}{2}}$$

The backward scheme is, thus, stable no matter what value of is Δt chosen. Thus, it is an unconditionally stable scheme. We can, furthermore, notice that it is damping, and that the amount of damping increases as the frequency ω increases. This is often considered to be a desirable property of a scheme. For instance, we can think of a system in which a number of frequencies are present at the same time ; for example, solving a system of equations of the type `h2.1`. This situation is similar to that existing in the real atmosphere. It would appear to be necessary to maintain the amplitudes of motions of different frequencies in the correct ratio. However, in numerical integrations, high frequency motions are often excited to unrealistically large amplitudes through errors in the initial data. It may then be desirable to reduce the amplitudes of high frequency motions by a selective damping in the time differencing scheme. In other words, a scheme

with frequency dependent damping properties can be used to filter out undesirable high frequency motions.

For the trapezoidal scheme we find

$$|\lambda| = 1$$

The trapezoidal scheme is, thus, always neutral. The amplitude of the numerical solution remains constant, just as does that of the true solution. It is useful to note that both the implicit schemes considered here were stable no matter how large a value of Δt was chosen.

The iterative two level schemes can also be described by a single equation in the same way as `h2.6`. Thus, we write

$$U^{(n+1)^*} = U^{(n)} + U^{(n)} + \Delta t f^{(n)}$$

$$U^{(n+1)} = U^{(n)} + \Delta t \left(\alpha f^{(n)} + \beta f^{*(n+1)} \right)$$

$$\alpha + \beta = 1$$

Now, $a = 1$, $\beta = 1$ for the Matsuno scheme, and, $a = \frac{1}{2}$, $\beta = \frac{1}{2}$ for the Heun scheme.

Applied to the oscillation equation `h2.18` gives

$$U^{(n+1)^*} = U^{(n)} + i\omega \Delta t U^n$$

$$U^{n+1} = U^{(n)} + i\omega \Delta t \left(\alpha U^{(n)} + \beta U^{(n+1)^*} \right)$$

Eliminating $U^{(n+1)^*}$ we obtain, again using `h2.8`,

$$U^{n+1} = \left(1 - \beta p^2 + ip \right) U^{(n)}$$

Thus,

$$\lambda = 1 - \beta p^2 + ip$$

Substituting the appropriate values of β we now obtain the values of λ, for the two schemes. Hence, for the Matsuno scheme

$$\lambda = 1 - p^2 + ip$$

and for the Heun scheme

$$\lambda = 1 - \frac{1}{2} p^2 + ip$$

To test for stability we evaluate $|\lambda|$. For the Matsuno scheme we obtain

$$|\lambda| = \left(1 - p^2 + p^4 \right)^{\frac{1}{2}}$$

Thus, the Matsuno scheme is stable if

$$|p| \leq 1$$

In other words, to achieve stability we have to choose Δt sufficiently small so that

$$\Delta t \leq \frac{1}{|\omega|}$$

The Matsuno scheme, thus, is conditionally stable. The higher the frequency, the more restrictive is the stability condition.

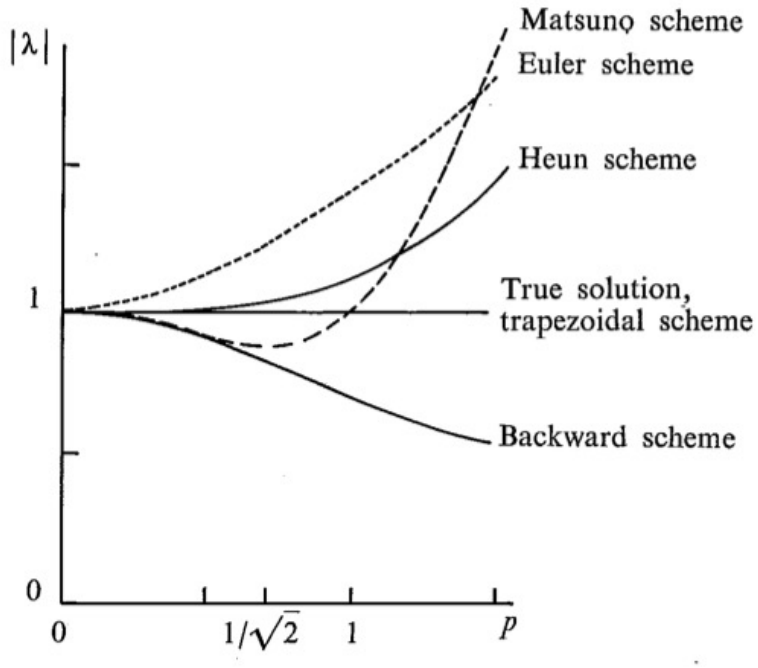


Figure 5:

Differentiating h2.23 we find that

$$\frac{d|\lambda|}{dp} = \frac{p}{(1-p^2+p^4)^{\frac{1}{2}}}(1-2p^2)$$

Hence, the amplification factor of the Matsuno scheme has a minimum for $p = 1/\sqrt{2}$. Therefore, as pointed out by Matsuno (1966a) when dealing with a system with a number of frequencies we can choose a time step so as to have $0 \leq p \leq 1/\sqrt{2}$ for all the frequencies present, and then, in the same way as backward implicit scheme, this scheme will reduce the relative amplitudes of high frequencies. This technique has recently become very popular for initialization of atmospheric models, where it is used to damp the spurious high frequency noise generated by the assimilation of the observed data. As shown by Matsuno (1966b) higher order accuracy schemes with similar filtering characteristics can be conÂstructed.

For the Heun scheme h2.22 gives

$$|\lambda| = \left(1 + \frac{1}{4}p^4\right)^{\frac{1}{2}}$$

This is always greater than unity. Thus, the Heun scheme is always unstable, like the Euler scheme. However, instead of :eq:2.15, for small p we now have Â

$$|\lambda| = 1 + \frac{1}{8}p^4 + \ldots$$

that is, $\left|\hat{\text{I}}\text{»}\right| = 1 + 0\left[(\Delta t)^4\right]$. This instability is quite weak. Experience shows that it can be tolerated when we can choose a relatively small value of Δt. (Note that, whenever the amplification rate is less than that allowed by the von Neumann necessary condition, the total amplification in a given time is reduced as the time step is reduced.)

Figure figg:5 Summarizes the results obtained for the five schemes considered so far. For all of these schemes the amplification factors were found to be even functions of p, so the amplification factor curves are shown only for $p \geq 0$.

It is also of interest to consider the phase change per time step, θ and the relative phase change per time step, θp.

Using the notation

$$\lambda \equiv \lambda_{\rm re} + i\lambda_{im}$$

we have, using h2.4,

$$\theta = \arctan \frac{\lambda_{\rm im}}{\lambda_{\rm re}}$$

or

$$\frac{\theta}{p} = \frac{1}{p} \arctan \frac{\lambda_{\rm im}}{\lambda_{\rm re}}$$

For the Euler and the backward schemes, using h2.11 and h2.12 we obtain

$$\frac{\theta}{p} = \frac{1}{p} \arctan \mathrm{p}$$

Since the right-hand side is always less than unity, we can see that these two schemes are decelerating. For $p = 1$ we have $\theta p = \pi 4$.

In other cases the effect may not be so obvious. For the Matsuno scheme, for example, h2.21 gives

$$\frac{\theta}{p} = \frac{1}{p} \arctan \frac{p}{1 - p^2}$$

It is not obvious whether the right-hand side here is greater or less than unity. However, the behaviour of h2.31 for all p is of no practical interest, since we already know that p'$must be chosen less than unity in order to ensure stability, and rather small for frequencies for which we want the$ eq : 'h2.31 for small p; we obtain

$$\frac{\theta}{p} = 1 + \frac{2}{3}p^2 + \ldots$$

The Matsuno scheme, therefore, is seen to be accelerating. For the special value $p = 1$ this can be seen directly from h2.13, since then $\frac{\theta}{p} = \frac{\pi}{2}$.

Analysis of phase errors of schemes applied to the oscillation equation is not so important as analysis of the amplification factor. Phase errors do not affect stability, and when these schemes are used to solve the partial differential equations of motion additional phase errors due to space differencing will appear. We will then be interested only in the total phase error, and it will be found that the error due to space differencing is usually dominant.

Three level schemes and computational modes

We consider first the leapfrog scheme h1.9. Applied to the oscillation equation it gives

$$U^{n+1} = U^{(n+1)} + i2\omega\Delta U^{(n)}$$

A problem with all three or more level schemes includÂing this is that they require more than one initial condition to start the computation. From a physical standpoint a single initial condition $U^{(0)}$ should have been sufficient. However, in addition to the physical initial condition, three level schemes require a computational initial condition $U^{(1)}$). This value cannot be calculated by a three level scheme, and, therefore, it will usually have to be obtained using one of the two level schemes.

According to h2.3 we also have

$$U^{(n)} = \lambda U^{(n-1)}, U^{(n+1)} = \lambda^2 U^{(n-1)}$$

When these relations are substituted into h2.32 we obtain

$$\lambda^2 - i2p\lambda - 1 = 0$$

a second degree equation for λ. It has solutions

$$\lambda_1 = \sqrt{1+p^2} + ip$$
$$\lambda_2 = -\sqrt{1-p^2} + ip$$

Thus, there are *two solutions* of the form $U^{n+1} = \lambda U^{(n)}$. This necessarily follows from the fact that we are considering a three level scheme; substitution of `h2.33` into the difference equation given by these schemes will always give a second degree equation for λ. In general, an m level scheme will give $m-1$ solutions of the form $U^{n+1} = \lambda U^{(n)}$. A solution of this type corresponding to a single value of λ is called a *mode*.

Consider now the two values that have been obtained for λ. If a solution of the form $U^{n+1} = \lambda U^{(n)}$ is to represent an approximation to the true solution, then we must have $\lambda \to 1$ as $\Delta \to 0$. For the values `h2.34`, as $p \equiv \omega \Delta t \to 0$ we do have $\lambda_1 \to 1$, however at the same time $\lambda_2 \to -1$. Solutions like that associated with λ_2 are usually called *physical modes* because we are always solving equations describing physical processes. SoluÂtions like that associated with λ_2 are not approximations to the true solution, and are called *computational modes*.

To clarify this situation we consider the simple case $\omega = 0$, that is, the equation

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

with the true solution

$$U = const$$

The leapfrog scheme, applied to `h2.35`, gives

$$U^{(n+1)} = U^{(n-1)}$$

For a given physical initial condition $U^{(0)}$, we consider two special choices of $U^{(1)}$.

A. Suppose calculating of $U^{(1)}$ happened to give the true value $U^{(0)}$, `h2.37` then gives, for all n,

$$U^{(n+1)} = U^{(n)}$$

or, since p = 0,

$$U^{(n+1)} = \lambda_1 U^{(n)}$$

Thus, we obtain a numerical solution that is equal to the true solution `h2.36`, and consists of the physical mode only.

B. Suppose calculating $U^{(1)}$ gives $U^{(1)} = -U^{(0)}$.

Then we obtain, for all n,

$$U^{(n+1)} = -U^{(n)}$$

or

$$U^{(n+1)} = \lambda_2 U^{(n)}$$

The numerical solution now consists entirely of the computational mode. Hence, it would appear that a good choice of the computational initial condition is of vital importance for obtaining a satisfactory numerical solution.

In general, since `h2.31` is a linear equation, its solution will be a linear combination of the two solutions

$$U_1^{(n)} = \lambda_1^n U_1^{(0)}$$
$$U_2^{(n)} = \lambda_2^n U_2^{(0)}$$

Therefore, we can write

$$U^{(n)} = a\lambda_1^n U_1^{|0|} + b\lambda_2^n U_2^{(0)}$$

where a and b are constants. Now this has to satisfy the physical and the computational initial condition; we obtain

$$U^{(0)} = aU_1^{|0|} + bU_2^{(0)}$$
$$U^{(1)} = a\lambda_1 U_1^{(0)} + bU_2^{(0)}$$

These equations can be solved for a $U_1^{(0)}$ and $bU_2^{(0)}$, and the results substituted into `h2.38`.

In this way we find

$$U^{(n)} = \frac{1}{\lambda_1 - \lambda_2}\left[\lambda_1^n\left(U^{(1)} - \lambda_2 U^{(0)}\right) - \lambda_2^n\left(U^1 - \lambda_1 U^{(0)}\right)\right]$$

Therefore, the amplitudes of the physical and of the computational modes are seen to be proportional to, respectively,

$$|U^{(1)} - \lambda_2 U^{(0)}| \quad and \quad |U^{(1)} - \lambda_1 U^{(0)}|$$

These are seen to depend on $U^{(1)}$. If, for example, we are able to choose $U^{(1)} = \lambda_1 U_1^{(0)}$, the numerical solution will consist of the physical mode only. If, on the other hand, the choice of $U^{(1)}$ is so unsuccessful as to have $U^{(1)} = \lambda_2 U^{(0)}$, the solution will consist entirely of the computational mode.

While this analysis illustrates the importance of a careful choice of $U^{(1)}$, it is not always possible to calculate $U^{(1)} = \lambda_1 U^{(0)}$ so as to eliminate the computational mode. Numerical methods are used in practice to solve equations that cannot be solved by analytical methods, and are more complex than the simple oscillation equaÂtion `h2.1`. In these cases we will not know the exact values of λ_1 and λ_2.

Thus $U^{(1)}$„ is usually computed using one of the two level schemes. The simplest method is to use the Euler scheme, or, a more refined procedure could be used, for example the Heun scheme. Using `h2.39` it can be shown that the latter alternative will give a smaller amplitude of the computational mode.

We also note that even if we did know the exact value of λ_1 this would still not allow the computational mode to be eliminated in a practical numerical calculation. The numerical solution which we calculate is not an exact solution of the finite difference equations, since the arithmetical operations are performed in practice only to a finite number of significant digits. The error produced in this way is called *round off error*, though in electronic computers results of arithmetic operations are sometimes truncated to a given number of digits, instead of being rounded off. With round off errors present, permanent elimination of the computational mode is not possible in principle, since the computational mode would appear in the course of integration in any case even if were absent initially. However, it is usually found that round off errors are of little importance in atmospheric models, and in solving partial differential equations in general.

Proceeding now to the stability analysis, in view of `h2.38` and our inability to eliminate the computational mode completely, we will have to require for stability that neither of the two amplification factors is greater than unity. It is convenient to consider three special cases.

$|p| \leq 1$. In `h2.34` $1 - p^2$ is positive, and we obtain $|\lambda_1| = |\lambda_2| = 1$

Thus, in this case both modes are stable and neutral. For the phase change, using `h2.28`

$$\theta_1 = arctan\left(\frac{p}{\sqrt{1-p^2}}\right)$$

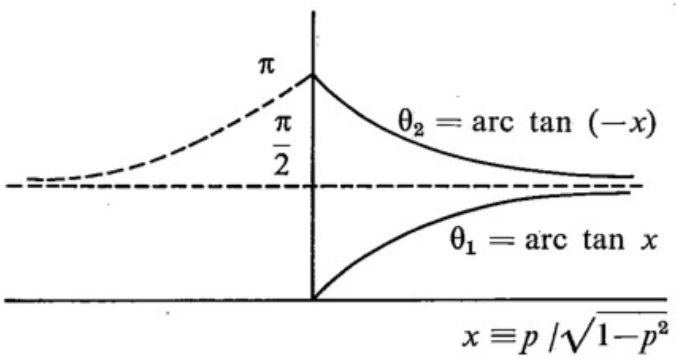


Figure 6:

$$\theta_2 = arctan\left(\frac{-p}{\sqrt{1-p^2}}\right)$$

It is instructive to consider the behaviour of θ, as a function of p, especially as $p \to 0$. We consider first the case $p > 0$.. Since for both modes $\lambda_{\text{im}} = \left|\hat{\text{I}}\text{»}\right| \sin\theta = p$‘$wehave$: $math$: ‘$0 < \theta < \pi$. Considering the signs of λ_{re}‘$wefindthat,$: $math$: ‘$0 < \theta < \frac{\pi}{2}$‘$and$: $math$:frac{pi}{2} < theta{2} < pi‘. To illustrate these results, the phase changes `h2.41` are plotted in `figg:6`. We see that, for all p,

$$\theta_2 = \pi - \theta_1$$

Specifically, as $p \to 0, \theta_1 \to p$ while $\theta_2 \to \pi - p$

Thus, for small Δt the physical mode is seen to approximate the true solution, while the behaviour of the computaÂtional mode is quite different. For the case $p < 0$, we obtain in the same way

$$\pm\theta_2 = -\pi - \theta_1$$

Thus, for $p \gtrless 0$

$$\theta_2 = \pm\pi - \theta_1$$

For accuracy of the physical mode, θ_1 should closely approximate the phase change of the true solution, p. For small p `h2.41` gives

$$\theta_1 = p + \frac{1}{6}p^3 + \ldots$$

Thus, the leapfrog scheme is accelerating. The acceleraÂtion, though, is four times less than that of the Matsuno scheme. It is instructive to note that schemes of different orders of accuracy can still have the same order of leading term in power series expansions of either the amplification factors or the phase changes.

Differentiating the first equation in `h2.41` we find

$$\frac{d\theta_1}{\text{dp}} = \frac{1}{\sqrt{1-p^2}}$$

The phase error, thus, is seen to increase sharply as $p \to 1.$, when $\frac{\theta_1}{p} \to \frac{\pi}{2}$

It may be useful to illustrate the behavior of the two modes obtained

$$U_1^{(n)} = U_1^{(0)} e^{in\theta 1}$$
$$U_2^{(n)} = U_2^{|0|} e^{\text{in}(\pm\pi - \theta_1)}$$

in the complex plane. For simplicity, we consider the case $\theta_1 = \frac{\pi}{8}$ and assume that the imaginary part of the solution is equal to zero at the initial moment. The physical mode,

as seen in h2.43, rotates in the positive sense by an angle θ_1 in each time step Δt, while at the same time the computational mode, in the case $p > 0$, rotates by an angle $\pi - \theta_1$. Therefore, the two modes can be represented graphically as in figg:7.

A detailed knowledge of the behaviour of the computational mode may be helpful in recognizing its excessive presence in an integration. Thus, we plot the real and imaginary parts of the computational mode as functions of time. This can be done by using an alternative form of the second equation in h2.43

$$U_2^{(n)} = (-1)^n U_2^{(0)} (\cos n\theta_1 - i \sin n\theta_2)$$

or directly from figg:7. We obtain diagrams as shown in figg:8. Because of the factor $(-1)^n$, both real and imaginary parts oscillate between time steps.

$|p| = 1$ This is a limiting case of the solutions considered for |p| < 1. h2.34 shows that the values of λ. are now equal,

$$\lambda_1 = 2 = ip$$

Therefore

$$|\lambda_1| = |\lambda_2| = 1$$

Thus, both modes are still neutral. Since neither of them has a real part, we obtain, for p = Â± 1,

$$\theta_1 = \theta_2 = \pm\frac{\pi}{2}$$

Therefore, the two modes can be written in the form

$$U^{(n)} = U^{(0)} e^{\pm in\frac{\pi}{2}}$$

In a complex plane, they rotate by an angle of $\frac{\pm\pi}{2}$ in each time step, while the true solution rotates by an angle of Â± 1 only. The phase error, thus, is large.

$|p| > 1$ Both values of λ in h2.34 still have imaginary parts only, so that

$$\lambda_1 = i\left(p + \sqrt{p^2 - 1}\right)$$

$$\lambda_2 = i\left(p - \sqrt{p^2 - 1}\right)$$

where the expressions in parentheses are real. Therefore,

$$|\lambda_1| = \left|p + \sqrt{p^2 - 1}\right|$$

$$|\lambda_2| = \left|p - \sqrt{p^2 - 1}\right|$$

Thus, for $p > 1$ we have $|\lambda_1| > 1$, and for $p < 1$ $|\lambda_2 > 1|$. Therefore, for $|p > 1|$ the leapfrog scheme is unstable. The instability increases sharply as |p| increases beyond 1 ; we can see this, for example for $p > 1$, because

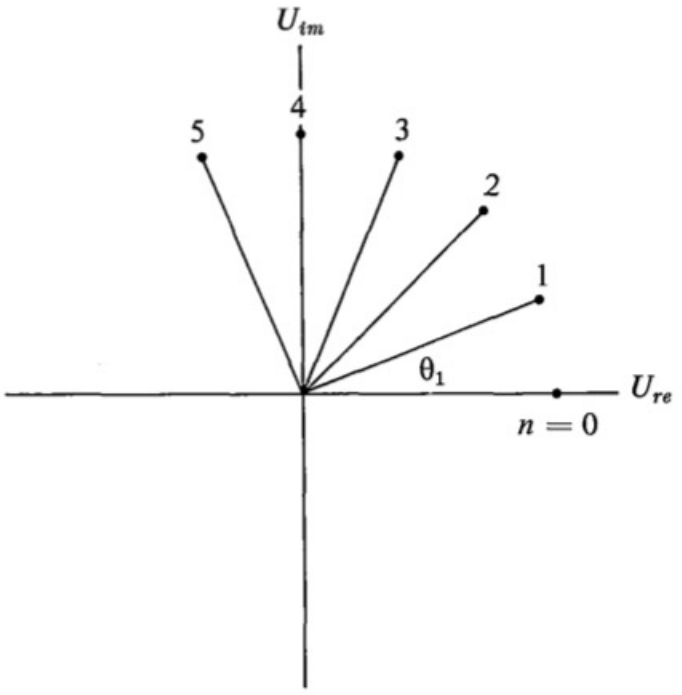
$$\frac{d\,|\lambda|_1}{\mathrm{dp}} = 1 + \frac{p}{\sqrt{p^2 - 1}}$$

which is unbounded as $p \to 1$

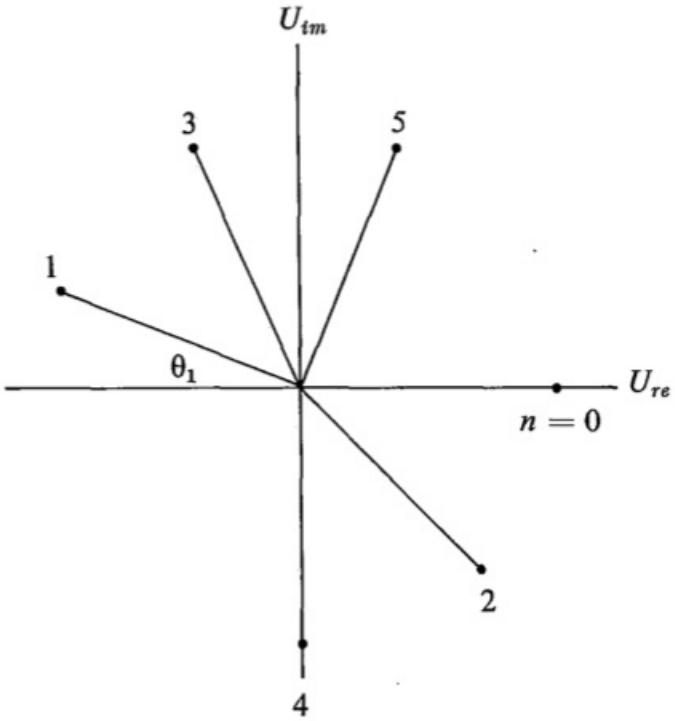
Since the two values of λ still have no real parts, we again have

$$\theta_1 = \theta_2 = \pm\frac{\pi}{2}$$

The two modes for $p \gtrless 1$, can thus be written as



Physical mode



Computational mode

Figure 7:

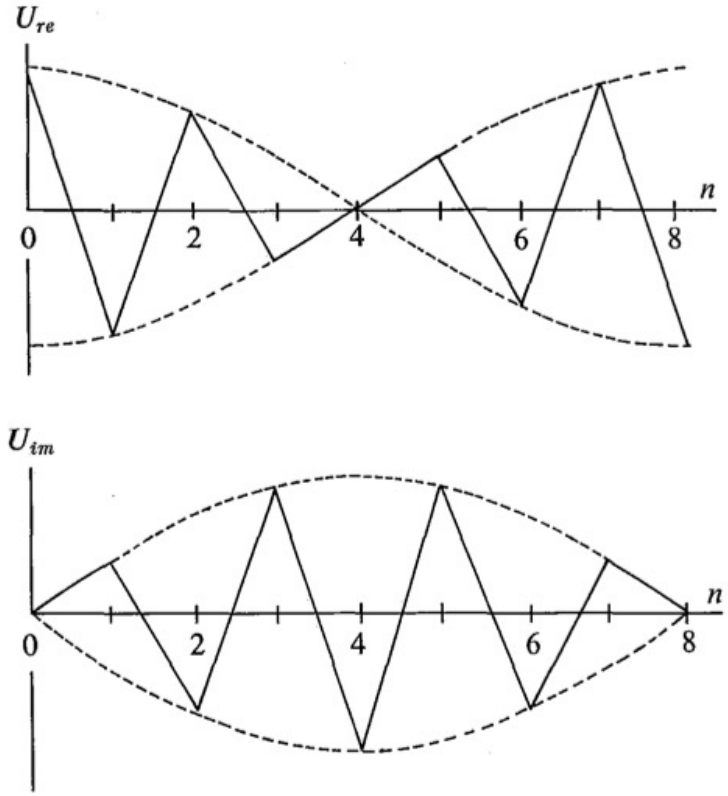


Figure 8:

$$U_1^{(n)} = \left|p + \sqrt{p^2 - 1}\right|^n U_1^{(0)} e^{\pm in\frac{\pi}{2}}$$

$$U_2^{(n)} = \left|p - \sqrt{p^2 - 1}\right|^n U_2^{(0)} e^{\pm in\frac{\pi}{2}}$$

In the complex plane, both modes again rotate by an angle of $\frac{\pm\pi}{2}$ in each time step. However, this time the amplitude of one of the modes increases, and that of the other decreases with time. The real part of the unstable mode can, for instance, be represented as a function of time by a graph like that in Fig. 2.5. Because of `h2.48` the period of the unstable oscillation is always $4\Delta t$. This can be used to diagnose the instability: if the results appear unsatisfactory, it is a good idea to

check for the presence of growing oscillations of that period.

To sum up, advantages of the leapfrog scheme are that it is a very simple scheme, of second order accuracy, and neutral within the stability range $|\omega\Delta t \leq 1|$. A disadvantage of the leapfrog scheme is the presence of a neutral computational mode. With nonlinear equations there is a tendency for a slow amplification of the compuÂtational mode. An example of this growth can be seen, for example, in one figure of a paper by Lilly (1965). The usual method used for suppressing this instability is the occasional insertion of a step made by a two level scheme, which eliminates the computational mode. A multi-level scheme that damps the computational mode could also be used for this purpose.

When solving the system of gravity wave equations, as will be shown in Chapter `Chapter4`, it is possible to construct grids and/or finite difference schemes which have essenÂtially the same properties as the leapfrog scheme, but in which the computational mode is absent. These methods calculate the physical mode only, and at the same time require only one half of the computation time needed for the regular leapfrog scheme as described here.

We consider, finally, stability and other properties of the Adams-Bashforth scheme `h1.10`. Applied to the oscillation equation it gives

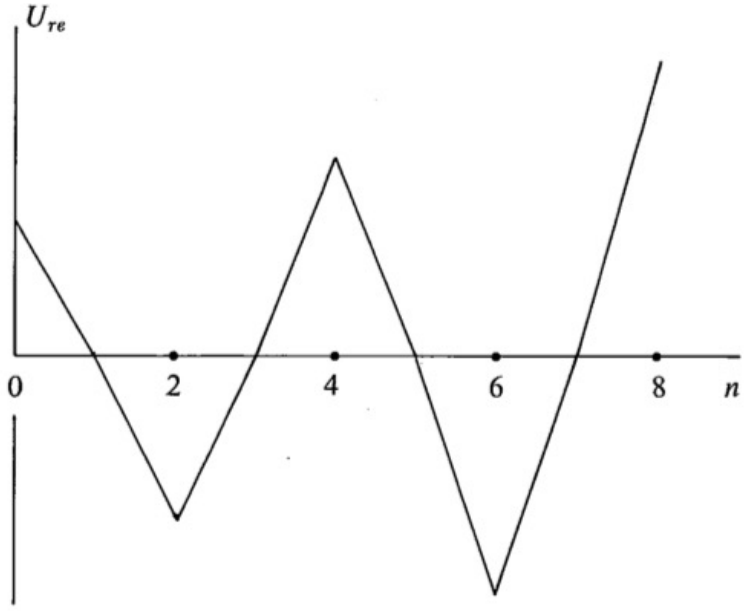


Figure 9:

$$U^{(n+1)} = U^{(n)} + i\omega\Delta t\left(\frac{3}{2}U^{(n)} - \frac{1}{2}U^{(n-1)}\right)$$

Substituting the relations `h2.33` we find

$$\lambda^2 - \left(1 + i\frac{3}{2}p\right)\lambda + i\frac{1}{2}p = 0$$

We have, of course, again obtained a second degree equation for λ.
It has the solutions

$$\lambda_1 = \frac{1}{2}\left(1 + i\frac{3}{2}p + \sqrt{1 - \frac{9}{4}p^2 + ip}\right)$$

$$\lambda_2 = \frac{1}{2}\left(1 + i\frac{3}{2}p - \sqrt{1 - \frac{9}{4}p^2 + ip}\right)$$

Thus, as $p \to 0$, $\lambda_1 \to 1$ while $\lambda_2 \to 0$. We see that the solution associated with λ_2 again represents a physical mode, and that associated with λ_2 a computational mode. However, while for the leapfrog scheme the computational mode was found to be neutral, here it is seen to be damped. This is a very useful property of the Adams-Bashforth scheme as the computational mode cannot cause inconveniences.

The exact analysis of the amplification factors here is more difficult because of the presence of square roots in `h2.51`. However, since for reasons of accuracy we have to choose a relatively small value of p in any case, it will suffice to consider amplification factors for small values of p only. The power series expansion of `h2.51` then gives

$$\lambda_1 = 1 + ip - \frac{1}{2}p^2 + i\frac{1}{4}p^3 - \frac{1}{8}p^4 + \ldots$$

$$\lambda_2 = i\frac{1}{2}p^2 + i\frac{1}{2}p^2 - \frac{1}{4}p^3 + \frac{1}{8}p^4 \ldots$$

Now, after rearranging the terms these series can be written as

$$\lambda_1 = \left(1 - \frac{1}{2}p^2 - \frac{1}{8}p^4 - \ldots\right) + i\left(p + \frac{1}{4}p^3 + \ldots\right)$$

$$\lambda_2 = \left(\frac{1}{2}p^2 + \frac{1}{8}p^4 + \ldots\right) + i\left(\frac{1}{2}p - \frac{1}{4}p^3 - \ldots\right)$$

which can be used to obtain the amplification factors

$$|\lambda_1| = \left(1 + \frac{1}{2}p^4 + \ldots\right)^{\frac{1}{2}}$$

$$|\lambda_2| = \left(\frac{1}{4}p^2 + \ldots\right)^{\frac{1}{2}}$$

The higher order terms have been omitted. A final expansion gives

$$|\lambda_1| = 1 + \frac{1}{4}p^4 + \ldots$$

$$|\lambda_2| = \frac{1}{2}p + \ldots$$

Expressions `h2.52` and/or `h2.53` show that the physical mode of the Adams-Bashforth scheme is always unstable. However, as for the Heun scheme, the amplification is only by a fourth order term, and it can be tolerated when a sufficiently small value of Δt is chosen. Note that the amplification given by `h2.53` is twice that given by `h2.26` for the Heun scheme. Since the amplification is proporÂtional to $(\Delta t)^4$, however, a small reduction in time step would compensate for that difference. Thus, the Adams-Bashforth scheme, with only one evaluation of the right hand side per time step, can still be considered much more economical. It has been fairly frequently used in meteorological numerical studies. For example, it is being used by Deardorff in his numerical simulations of the planetary boundary layer (e.g. Deardorff, 1974).

Analyses of the properties of some other schemes, applied to the oscillation equation, can be found in papers by Lilly (1965), Kurihara (1965) and Young (1968). In practice the choice of a scheme will depend not only on the properties considered here, but also on some practical considerations. For example, we might expect that the three level schemes, since they use more informaÂtion, would generally give better results than the two level schemes. Our findings agree with that conjecture ; for example, for second order accuracy the explicit three level schemes required only one evaluation of the right hand side per time step, while the two level schemes required two evaluations. As another example, if we want to damp high frequency motions with three level schemes we can linearly extrapolate the derivative beyond the centre of the interval $(n\Delta t, (n+1)\,\Delta t)$, and thus obtain a scheme that will perform such a damping in a more selective and more economical way than the Mat-suno scheme (Mesinger, 1971). However, three level schemes generally require more core storage space in the computer than two level schemes and this may affect our our decision.

0.2.3 Properties of schemes applied to the friction equation

We shall now consider the properties of schemes when applied to the equation

$$\frac{dU}{dt} = -\kappa U, \qquad U = U\,(t)\,, \qquad \kappa > 0$$

We shall call this equation the friction equation.

Again it is easy to justify our interest in this equation. For example, if we define $U \equiv u + iv$, it describes the effect of friction proportional to the velocity vector, as is often assumed for motions near the ground. As another example, note that when seeking a solution of the heat transfer, or Fick's diffusion equation

$$\frac{\partial u}{\partial t} = \sigma \frac{\partial^2 u}{\partial x^2}, \qquad \sigma > 0$$

in the form of a single harmonic component

$$u\,(x,t) = Re\left[\left(U\,(t)\,e^{\mathrm{ikx}}\right)\right]$$

we obtain

$$\frac{\mathrm{dU}}{\mathrm{dt}} = -\sigma k^2 U$$

This is equivalent to `k3.1` **if we substitute** $x \equiv \sigma k^2$.

The general solution of `k3.1` is

$$U(t) = U(0)\, e^{-\kappa t}$$

Thus, both the real and the imaginary part decrease exponentially with time.

The properties of schemes applied to `k3.1` will again be analyzed using the von Neumann method. As in the previous section, we consider first the non-iterative two level scheme `h2.6`. Applied to the friction equation, `h2.6` gives

$$U^{(n+1)} = U^n - \kappa \Delta t \left(\alpha U^{(n)} + \beta U^{(n+1)} \right)$$

Writing

$$K \equiv \kappa \Delta t$$

we obtain, rearranging the terms in `k3.3`,

$$U^{(n+1)} = \frac{1 - \alpha K}{1 + \beta K} U^{(n)}$$

For the Euler scheme $\alpha = 1$‘$and : math$:beta $= 0$‘; thus, `k3.5` shows that the Euler scheme is now stable if $|1 - K \leq 1|$, that is, if

$$0 < K \leq 2$$

Thus we see that the stability criteria of particular schemes do not have to be the same when they are applied to different equations. In the case of `k3.6`, one will normally be more demanding in the choice of Δt. For example, we will want $K < 1$, to prevent the solution `k3.5` oscillating from time step to time step.

For the backward scheme $\alpha = 0$‘$and : math$:beta $= 1$‘; it is always stable if $K > 0$. The solution does not oscillate in sign.

For the trapezoidal scheme $\alpha = \frac{1}{2}$‘$and : math$:beta $=$ frac{1}{2}‘; the scheme is again always stable for $K > 0$. The solution does not oscillate if $K < 2$.

Considering the iterative two level scheme `h2.18` we obtain

$$U^{(n+1)} = \left(1 - K + \beta K^2\right) U^{(n)}$$

Therefore, both the Matsuno and the Heun scheme are stable for sufficiently small values of K.

It is instructive to consider in some detail the behaviour of the numerical solution obtained using the leapfrog scheme. Applied to `k3.1` it gives

$$U^{(n+1)} = U^{(n-1)} - 2\Delta t U^{(n)}$$

The equation for the amplification factor is

$$\lambda^2 + 2K\lambda - 1 = 0$$

giving the solutions

$$\lambda_1 = -K + \sqrt{1 + K^2}$$
$$\lambda_2 = -K - \sqrt{1 + K^2}$$

As $k \to 0, \quad \lambda \to 1$, while $\lambda_2 \to -1$ thus, the solution associated with λ_1 again represents the physical mode, and that associated with λ_2 the computational mode. For

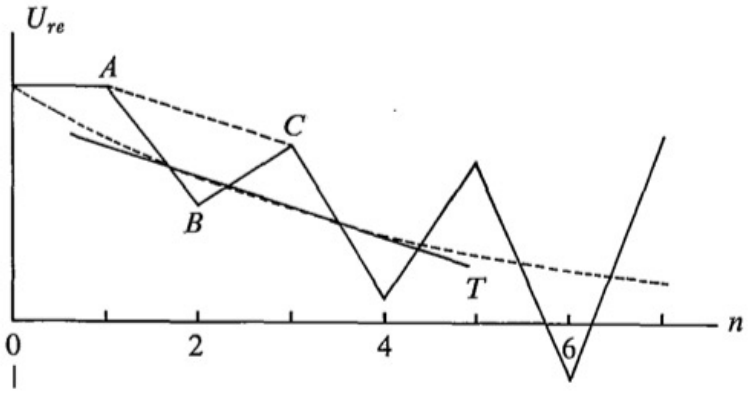


Figure 10:

$K > 0$, that is, for the normal case of a forward integration in time, we have $\lambda_2 < -1$; hence, the compuÂtational mode is always unstable. It changes sign from time step to time step, and its magnitude increases. As before, we cannot hope to eliminate the computaÂtional mode completely. This amplification is not negligible, and the leapfrog scheme is therefore not suitable for numerical integration of the friction equation.

A simple example can be given to illustrate the inÂstability of the leapfrog scheme. Let U have only a real part, and suppose we have set $U^{(1)} = U^{(0)}$, as shown in `figg:10` Furthermore, let the dashed curve in the figure represent the true solution satisfying the given initial condition $U^{(0)}$. Knowing $U^{(0)}$, $U^{(1)}$, and the true solution it is possible to construct a graph of the numerical solution, using the fact that $\frac{\mathrm{dU}}{dt=-\kappa U}$ is equal to the slope of the line tangent to the true solution at the approÂpriate value of U. In this way we obtain the numerical solution shown by the full line. In this method, the deriÂvative is calculated as a function of the current value of $U^{(n)}$, and the increment due to this derivative is added to the preceding value. This is seen to result in an unbounded growth of the difference between consecutive values of $U^{(n)}$, even when this difference is equal to zero initially.

Finally, for the Adams-Bashforth scheme we obtain

$$\lambda = \frac{1}{2}\left(1 - \frac{3}{2}K \pm \sqrt{1 - K\frac{9}{4}K^2}\right)$$

The Adams-Bashforth scheme, thus, is stable for suffiÂciently small values of K. The computational mode is damped.

0.2.4 A combination of schemes

A natural question to ask at this point is what can we do if, for example, the equation contains both the oscillaÂtion and the friction term, that is

$$\frac{\mathrm{dU}}{\mathrm{Dt}} = i\omega U - \kappa U$$

Here we might like to use the leapfrog scheme because of the oscillation term $i\omega U$, but we know that it cannot be used for the friction term $-\kappa U$. In this and similar situations we can use different schemes for the different terms; for example, we might use the leapfrog scheme for the oscillation term and the forward scheme for the friction term. We then obtain

$$U^{(n+1)} = U^{(n-1)} + 2\Delta t\left(i\omega U^{(n)} - \kappa U^{(n-1)}\right)$$

Other combinations, of course, are also possible.