Numerical Methods for the Solution of Hyperbolic Partial Differential Equations

Lecture Notes

Luciano Rezzolla

SISSA, International School for Advanced Studies, Trieste, Italy

Available also online at `www.sissa.it/~rezzolla`

April 26, 2004

Contents

Chapter 1

Introduction

Let us consider a quasi-linear partial differential equation (PDE) of second-order, which we can write generically as

$$a_{11}\frac{\partial^2 u}{\partial x^2} + 2a_{12}\frac{\partial^2 u}{\partial x \partial y} + a_{22}\frac{\partial^2 u}{\partial y^2} + f(x, y, u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}) = 0 \,, \tag{1.1}$$

where x, y are not necessarily all spatial coordinates and where we will assume the coefficients a_{ij} to be constant. The traditional classification of partial differential equations is then based on the sign of the determinant $\Delta \equiv a_{11}a_{22} - a_{12}^2$ that we can build with the coefficients of equation (1.1) and distinguishes three types of such equations. More specifically, equation (1.1) will be (strictly) *hyperbolic* if $\Delta = 0$ has roots that are real (and distinct), *parabolic* if $\Delta = 0$ has real but zero roots, while it will be *elliptic* if $\Delta = 0$ has complex roots (see Table 1.1).

Elliptic equations, on the other hand, describe *boundary value* problems, or **BVP**, since the space of relevant solutions Ω depends on the value that the solution takes on its boundaries $d\Omega$. Elliptic equations are easily recognizable by the fact the solution does not depend on time coordinate t and a prototype elliptic equation is in fact given by *Poisson equation* (cf. Table 1.1).

Hyperbolic and parabolic equations describe *initial value* problems, or **IVP**, since the space of relevant solutions Ω depends on the value that the solution L (which we assume with compact support) takes on some initial time (see Fig. 1.1). In practice, IVP problems are easily recognizable by the fact that the solution will depend on the time coordinate t. Very simple and useful examples of hyperbolic and parabolic equations are given by the *wave equation* and by the *diffusion equation*, respectively (cf. Table 1.1). An important and physically-based difference between hyperbolic and parabolic equations becomes apparent by considering the "characteristic velocities" associated to them. These represent the velocities at which perturbations are propagated and have *finite* speeds in the case of hyperbolic

Type	*Condition*	*Example*	
Hyperbolic	$a_{11}a_{22} - a_{12}^2 < 0$	Wave equation:	$\dfrac{\partial^2 u}{\partial t^2} = v^2 \dfrac{\partial^2 u}{\partial x^2}$
Parabolic	$a_{11}a_{22} - a_{12}^2 = 0$	Diffusion equation:	$\dfrac{\partial u}{\partial t} = \dfrac{\partial}{\partial x}\left(D\dfrac{\partial u}{\partial x}\right)$
Elliptic	$a_{11}a_{22} - a_{12}^2 > 0$	Poisson equation:	$\dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2} = \rho(x,y)$

Table 1.1: Schematic classification of a quasi-linear partial differential equation of second-order. For each class, a prototype equation is presented.

equations, while these speeds are *infinite* in the case of parabolic equations. In this way it is not difficult to appreciate that while both hyperbolic and parabolic equations describe time-dependent equations, the domain of dependence in a finite time for the two classes of equations can either be finite (as in the case of hyperbolic equations), or infinite (as in the case of parabolic equations).

In the following Sections 2–7 we will concentrate on partial differential equations of hyperbolic type. Before doing that, however, it is useful to discretize the continuum space of solutions (a "spacetime" in the case of IVPs) in spatial foliations such that the time coordinate t is constant on each slice. As shown in Fig. 1.2, each point $\mathcal{P}(x_j, t^n)$ in this discretized spacetime will have spatial and time coordinate defined as

$$x_j = x_0 + j\Delta x \;, \qquad\qquad j = 0, \pm 1, \ldots, \pm J \;,$$

$$t^n = t^0 + n\Delta t \;, \qquad\qquad n = 0, \pm 1, \ldots, \pm N \;, \tag{1.2}$$

where Δt and Δx are the increments between two spacelike and timelike foliations, respectively. In this way we can associate a generic solution $u(x,t)$ in the continuum spacetime to a set of discretized solutions $u_i^m \equiv \mathbf{u}(x_i, t^m)$ with $i = \pm I, \ldots, \pm 1, 0$ and $m = \pm M, \ldots, \pm 1, 0$ and $I \leq J;\ M \leq N$. Clearly, the number of discrete solutions to be associated to $u(x,t)$ will depend on the properties of the discretized spacetime (i.e. on the increments Δt and Δx) which will also determine the *truncation error* introduced by the discretization.

Once a discretization of the spacetime is introduced, *finite difference* techniques offer a very natural way to express a partial derivative (and hence a partial differential equation). The basic idea behind these techniques is that the solution of the differential equation $u(x_j, t^n + \Delta t)$ at a given position x_j and at a given time t^n can be Taylor-expanded in the vicinity of

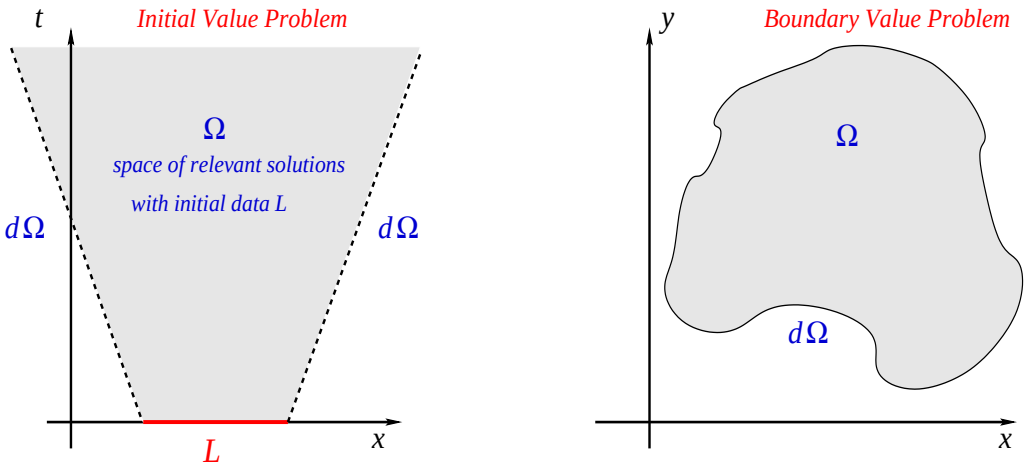

Figure 1.1: Schematic distinction between IVPs and BVPs.

(x, t^n). Under this simple (and most often reasonable assumption), differential operators can be substituted by properly weighted differences of the solution evaluated at different points in the numerical grid. In the following Section we will discuss how different choices in the way the finite-differencing is made will lead to numerical algorithms with different properties.

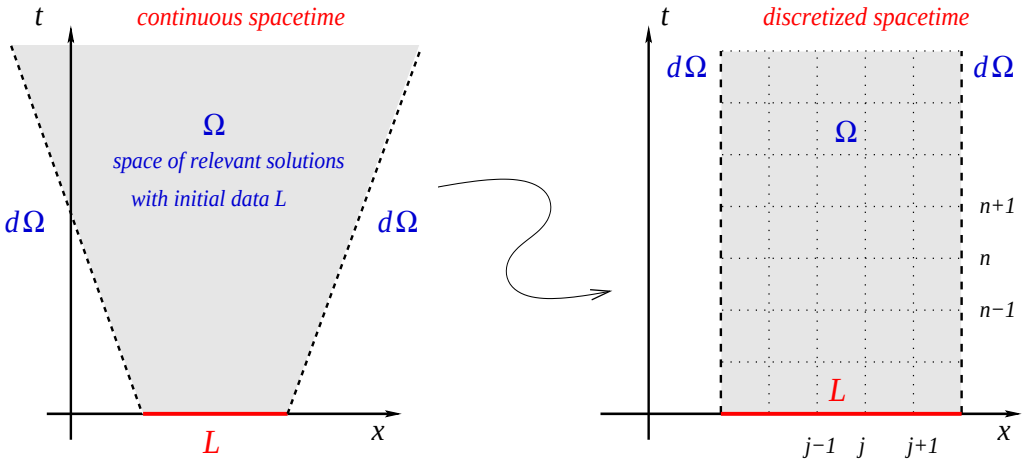

Figure 1.2: Schematic discretization of a hyperbolic IVP.

Chapter 2

Hyperbolic PDEs: Flux Conservative Formulation

It is often the case, when dealing with hyperbolic equations, that they can be formulated through conservation laws stating that a given quantity "u" is transported in space and time and is thus locally "conserved". The resulting "law of continuity" leads to equations which are called *conservative* and are of the type

$$\frac{\partial u}{\partial t} + \nabla \mathbf{F}(u) = 0 \,, \tag{2.1}$$

where $u(\mathbf{x}, t)$ is the *density* of the conserved quantity, $\mathbf{F}$ the density flux and $\mathbf{x}$ a vector of spatial coordinates. In most of the physically relevant cases, the flux density $\mathbf{F}$ will not depend explicitly on $\mathbf{x}$ and t, but only implicitly through the density $u(\mathbf{x}, t)$, i.e. $\mathbf{F} = \mathbf{F}(u(\mathbf{x}, t))$. The vector $\mathbf{F}$ is also called the *conserved flux* and takes this name from the fact that in the integral formulation of the conservation equation (2.1), the time variation of the integral of u over the volume $\mathcal{V}$ is indeed given by the net flux of u across the surface enclosing $\mathcal{V}$.

Generalizing expression (2.1), we can consider a vector of densities $\mathbf{U}$ and write a set of conservation equations in the form

$$\boxed{\frac{\partial \mathbf{U}}{\partial t} + \nabla \mathbf{F}(\mathbf{U}) = \mathbf{S}(\mathbf{U}) \,.} \tag{2.2}$$

Here, $\mathbf{S}(\mathbf{U})$ is a generic "source term" indicating the sources and sinks of the vector $\mathbf{U}$. The main property of the homogeneous equation (2.2) (i.e. when $\mathbf{S}(\mathbf{U}) = 0$) is that the knowledge of the state-vector $\mathbf{U}(x, t)$ at a given point x at time t allows to determine the rate of flow, or flux, of each state variable at (x, t).

Conservation laws of the form given by (2.1) can also be written as a quasi-linear form

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{A}(\mathbf{U}) \frac{\partial \mathbf{U}}{\partial x} = 0 \,, \tag{2.3}$$

where $\mathbf{A}(\mathbf{U}) \equiv \partial\mathbf{F}/\partial\mathbf{U}$ is the Jacobian of the flux vector $\mathbf{F}(\mathbf{U})$.

The use of a conservation form of the equations is particularly important when dealing with problems admitting shocks or other discontinuities in the solution, e.g. when solving the hydrodynamical equations. A non-conservative method, i.e. a method in which the equations are not written in a conservative form, might give a numerical solution which appears perfectly reasonable but then yields incorrect results. A well-known example is offered by Burger's equation, i.e. the momentum equation of an isothermal gas in which pressure gradients are neglected, and whose non-conservative representation fails dramatically in providing the correct shock speed if the initial conditions contain a discontinuity. Moreover, since the hydrodynamical equations follow from the physical principle of conservation of mass and energy-momentum, the most obvious choice for the set of variables to be evolved in time is that of the conserved quantities. It has been proved that non-conservative schemes do not converge to the correct solution if a shock wave is present in the flow, whereas conservative numerical methods, if convergent, do converge to the *weak solution* of the problem.

In the following, we will concentrate on numerical algorithms for the solution of hyperbolic partial differential equations written in the *conservative* form of equation (2.2). The advection and wave equations can be considered as prototypes of this class of equations in which with $\mathbf{S}(\mathbf{U}) = 0$ and will be used hereafter as our working examples.

Chapter 3

The advection equation in one spatial dimension (1D)

A special class of conservative hyperbolic equations are the so called *advection equations*, in which the time derivative of the conserved quantity is proportional to its spatial derivative. In these cases, **F**(**U**) is diagonal and given by

$$\mathbf{F}(\mathbf{U}) = v\mathbf{I} \cdot \mathbf{U} \,, \tag{3.1}$$

where **I** is the identity matrix.

Because in this case the finite-differencing is simpler and the resulting algorithms are easily extended to more complex equations, we will use it as our "working example". More specifically, the advection equation for u we will consider hereafter has, in 1D, the simple expression

$$\frac{\partial u}{\partial t} + v\frac{\partial u}{\partial x} = 0 \,, \tag{3.2}$$

and admits the general analytic solution $u = f(x - vt)$, representing a wave moving in the positive x-direction.

3.1 The 1D Upwind scheme: $\mathcal{O}(\Delta t, \Delta x)$

We will start making use of finite-difference techniques to derive a discrete representation of equation (3.2) by first considering the derivative in time. Taylor expanding the solution around $(x_j, t^n))$ we obtain

$$u(x_j, t^n + \Delta t) = u(x_j, t^n) + \frac{\partial u}{\partial t}(x_j, t^n)\Delta t + \mathcal{O}(\Delta t^2) \,, \tag{3.3}$$

or, equivalently,

$$u_j^{n+1} = u_j^n + \left.\frac{\partial u}{\partial t}\right|_j^n \Delta t + \mathcal{O}(\Delta t^2) \ . \tag{3.4}$$

Isolating the time derivative and dividing by Δt we obtain

$$\left.\frac{\partial u}{\partial t}\right|_j^n = \frac{u_j^{n+1} - u_j^n}{\Delta t} + \mathcal{O}(\Delta t) \ . \tag{3.5}$$

Adopting a standard convention, we will consider the finite-difference representation (FDR) of an m-th order *differential operator* $\partial^m/\partial x^m$ in the generic x-direction (where x could either be a time or a spatial coordinate) to be of order p if and only if $\partial^m u/\partial x^m = \mathrm{FDR} + \mathcal{O}(\Delta x^p)$. Of course, the time and spatial operators may have FDRs with different orders of accuracy and in this case the overall order of the equation is determined by the differential operator with the largest truncation error.
xNote also that while the truncation error is expressed for the differential operator, the numerical algorithms will not be expressed in terms of the differential operators and will therefore have different (usually smaller) truncation errors. This is clearly illustrated by the equations above, which show that the explicit solution (3.4) is of higher order than the finite-difference expression for the differential operator (3.5).

With this definition in mind, it is not difficult to realize that the finite-difference expression (3.5) for the time derivative is only first-order accurate in Δt. However, accuracy is not the most important requirement in numerical analysis and a first-order but stable scheme is greatly preferable to one which is higher order (i.e. has a smaller truncation error) but is unstable.

In way similar to what we have done in (3.5) for the time derivative, we can derive a first-order, finite-difference approximation to the space derivative as

$$\left.\frac{\partial u}{\partial x}\right|_j^n = \frac{u_j^n - u_{j-1}^n}{\Delta x} + \mathcal{O}(\Delta x) \ . \tag{3.6}$$

While formally similar, the approximation (3.6) suffers of the ambiguity, not present in expression (3.5), that the first-order term in the Taylor expansion can be equally expressed in terms of u_{j+1}^n and u_j^n, i.e.

$$\left.\frac{\partial u}{\partial x}\right|_j^n = \frac{u_{j+1}^n - u_j^n}{\Delta x} + \mathcal{O}(\Delta x) \ . \tag{3.7}$$

This ambiguity is the consequence of the first-order approximation which prevents a proper "centring" of the finite-difference stencil. However, and as long as we are concerned with an advection equation, this ambiguity is easily solved if we think that the differential equation will simply translate each point in the initial solution to the new position $x + v\Delta t$ over a time interval Δt. In this case, it is natural to select the points in the solution at the

time-level n that are "upwind" of the solution at the position j and at the time-level $n+1$, as these are the ones causally connected with u_j^{n+1}. Depending then on the direction in which the solution is translated, and hence on the value of the advection velocity v, two different finite-difference representations can be given of equation (3.2) and these are

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = -v \left(\frac{u_j^n - u_{j-1}^n}{\Delta x} \right) + \mathcal{O}(\Delta t, \Delta x) \,, \qquad \text{if } v > 0 \,, \tag{3.8}$$

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = -v \left(\frac{u_{j+1}^n - u_j^n}{\Delta x} \right) + \mathcal{O}(\Delta t, \Delta x) \,, \qquad \text{if } v < 0 \,, \tag{3.9}$$

respectively. As a result, the final finite-difference algorithms for determing the solution at the new time-level will have the form

$$u_j^{n+1} = u_j^n - \frac{v\Delta t}{\Delta x}(u_j^n - u_{j-1}^n) + \mathcal{O}(\Delta t^2, \Delta x) \,, \qquad \text{if } v > 0 \,, \tag{3.10}$$

$$u_j^{n+1} = u_j^n - \frac{v\Delta t}{\Delta x}(u_{j+1}^n - u_j^n) + \mathcal{O}(\Delta t^2, \Delta x) \,, \qquad \text{if } v < 0 \,. \tag{3.11}$$

More in general, for a system of linear hyperbolic equations with state vector $\mathbf{U}$ and flux-vector $\mathbf{F}$, the upwind scheme will take the form

$$\boxed{\mathbf{U}_j^{n+1} = \mathbf{U}_j^n \pm \frac{\Delta t}{\Delta x} \left[\mathbf{F}_{j\mp 1}^n - \mathbf{F}_j^n \right] + \mathcal{O}(\Delta t^2, \Delta x)} \,, \tag{3.12}$$

where the $\pm$ sign should be chosen according to whether $v > 0$ or $v < 0$. The logic behind the choice of the stencil in an upwind method is is illustrated in Fig. 1.2 where we have shown a schematic diagram for the two possible values of the advection velocity.

The upwind scheme (as well as all of the others we will consider here) is an example of an *explicit* scheme, that is of a scheme where the solution at the new time-level $n+1$ can be calculated explicitly from the quantities that are already known at the previous time-level n. This is to be contrasted with an *implicit* scheme in which the FDR of the differential equation has, on the right-hand-side, terms at the new time-level $n+1$. These methods require in general the solution of a number of coupled algebraic equations and will not be discussed further here.

The upwind scheme is a stable one in the sense that the solution will not have exponentially growing modes. This can be seen through a *von Neumann stability analysis*, a useful tool which allows a first simple validation of a given numerical scheme. It is important to underline that the von Neumann stability analysis is *local* in the sense that: *a)* it does not take into account boundary effects; *b)* it assumes that the coefficients of the finite difference equations are sufficiently slowly varying to be considered constant in time and space (this is

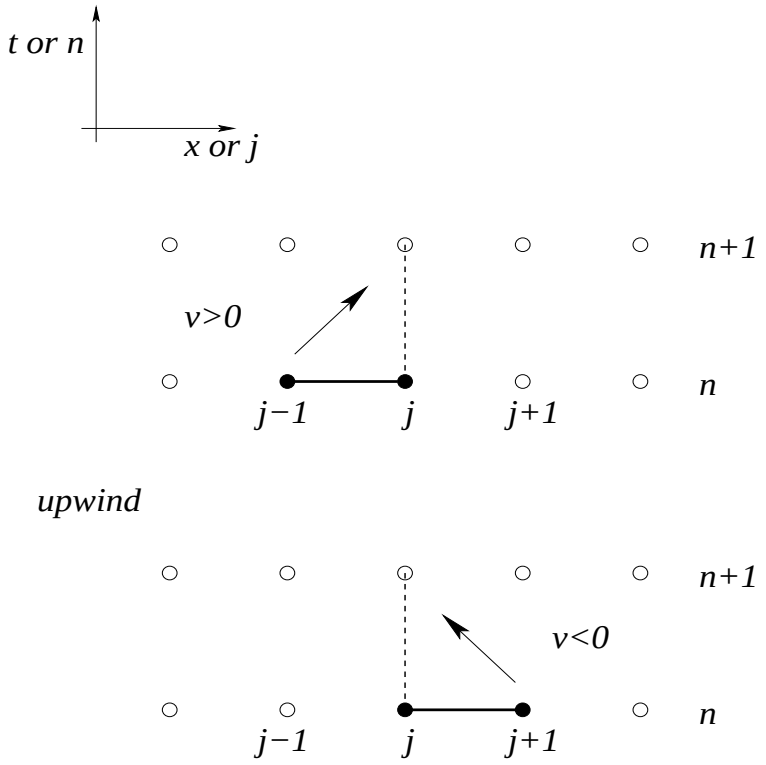

Figure 3.1: Schematic diagram of an UPWIND evolution scheme.

a reasonable assumptions if the equations are linear). Under these assumptions, the solution can be seen as a sum of eigenmodes which at each grid point have the form

$$u_j^n = \xi^n e^{ikx_j} \ , \tag{3.13}$$

where k is the spatial wave number and $\xi = \xi(k)$ is a *complex* number.

If we now consider the symbolic representation of the finite difference equation as

$$u_j^{n+1} = \mathcal{T}(\Delta t^p, \Delta x^q) u_j^n \ , \tag{3.14}$$

with $\mathcal{T}(\Delta t^p, \Delta x^q)$ being the evolution operator of order p in time and q in space, it then becomes clear from (3.13) and (3.14) that the time evolution of a single eigenmode is nothing but a succession of integer powers of the complex number ξ which is therefore named *amplification factor*. This naturally leads to a criterion of stability as the one for which the modulus of the amplfication factor is always less than 1, i.e.

$$|\xi|^2 = \xi\xi^* \leq 1 \ . \tag{3.15}$$

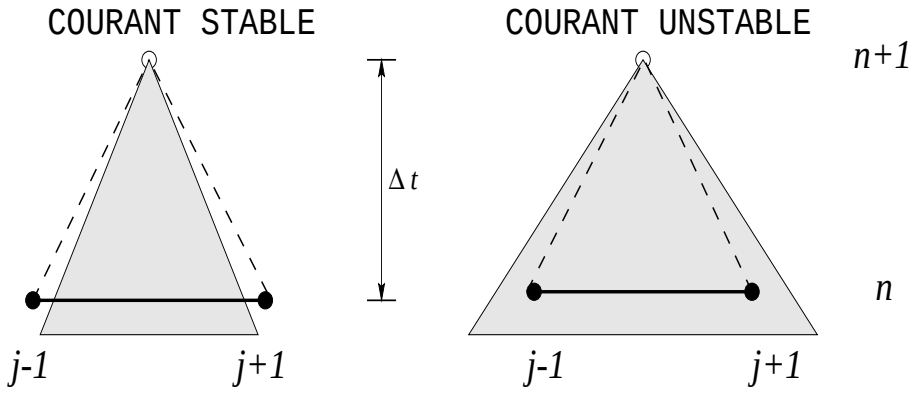

Figure 3.2: Schematic diagram of Courant stable and unstable choices of time-steps Δt. The two dashed lines limit the numerical domain of dependence of the solution at x_j^{n+1}, while the shaded area represents the physical domain of dependence. Stability is achieved when the first one is larger than the second one.

Using (3.13) in (3.10)–(3.11) we would obtain an amplification factor

$$\xi = 1 - \left|\frac{v\Delta t}{\Delta x}\right| (1 - \cos(k\Delta x)) - i\frac{v\Delta t}{\Delta x}\sin(k\Delta x) \,, \tag{3.16}$$

whose squared modulus is

$$|\xi|^2 = 1 - 2\left|\frac{v\Delta t}{\Delta x}\right| \left(1 - \left|\frac{v\Delta t}{\Delta x}\right|\right) (1 - \cos(k\Delta x)) \,, \tag{3.17}$$

which is less than one as long as the *Courant-Friedrichs-Löwy condition* (CFL condition)

$$\frac{|v|\Delta t}{\Delta x} \leq 1 \,, \tag{3.18}$$

is satisfied (condition (3.18) is sometimes referred to simply as the Courant condition.). Note that in practice, the CFL condition (3.18) is used to determine the time-step Δt once v is known and Δx has been chosen to achieve a certain accuracy, i.e.

$$\Delta t = c_{\text{CFL}} \frac{\Delta x}{|v|} \,, \tag{3.19}$$

with $c_{\text{CFL}} < 1$ being the CFL factor. Expression (3.19) also allows a useful interpretation of the CFL condition.

From a *mathematical* point of view, the condition ensures that the numerical domain of dependence of the solution is *larger* than the physical one. From a *physical* point of view, on the other hand, the condition ensures that the propagation speed of any physical

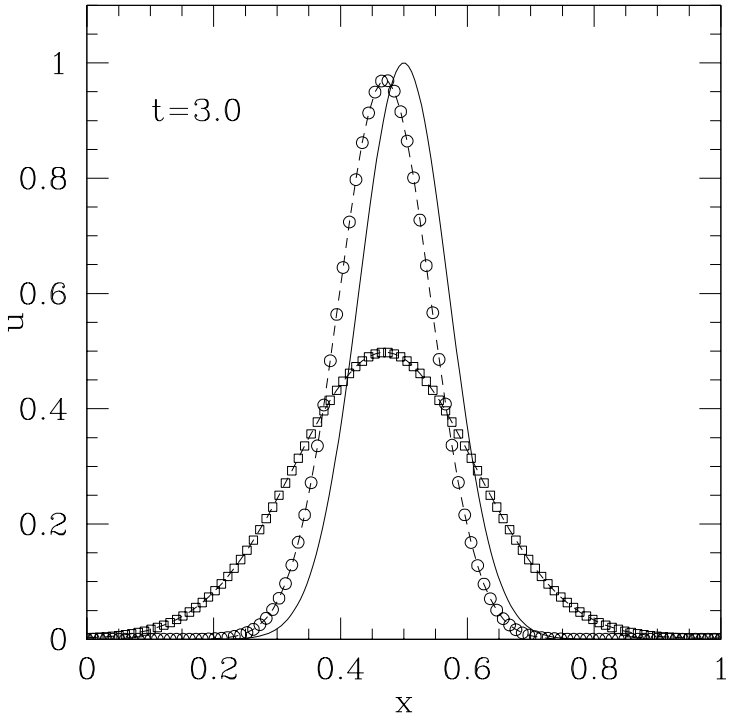

Figure 3.3: Time evolution of a Gaussian initially centred at $x = 0.5$ computed using an upwind scheme with $v = 10$ and 100 gridpoints. The analytic solution at time $t = 3$ is shown with a solid line the dashed lines are used to represent the numerical solution at the same time. Two different simulations are reported with the circles referring to a CFL factor $c_{\rm CFL} = 0.99$ and squares to a CFL factor $c_{\rm CFL} = 0.50$. Note how dissipation increases as the CFL is reduced.

perturbation (e.g. the sound speed, or the speed of light) is always smaller than the numerical one $v_{\rm N} \equiv \Delta x/\Delta t$, i.e.

$$|v| = c_{\rm CFL}\frac{\Delta x}{\Delta t} \leq v_{\rm N} \equiv \frac{\Delta x}{\Delta t}\,. \tag{3.20}$$

Equivalently, the CFL conditions prevents any physical signal to propagate for more than a fraction of a grid-zone during a single time-step (cf. Fig. 3.2)

As a final remark it should be noted that as described so far, the upwind method will yield satisfactory results only in the case in which the equations have an obvious transport character in one direction. However, in more general situations such as a wave equation, the upwind method will not be adequate and different expressions, based on finite-volume formulations of the equations will be needed [1, 4].

3.2 The 1D FTCS scheme: $\mathcal{O}(\Delta t, \Delta x^2)$

Let us consider again the advection equation (3.2) but we now finite difference with a more accurate approximation of the space derivative. To do this we can calculate the two Taylor expansions in $x_j \pm \Delta x$

$$\begin{aligned} u(x_j + \Delta x, t^n) &= u(x_j, t^n) + \frac{\partial u}{\partial x}(x_j, t^n)\Delta x + \frac{1}{2}\frac{\partial^2 u}{\partial x^2}(x_j, t^n)\Delta x^2 + \mathcal{O}(\Delta x^3) \, , \\ u(x_j - \Delta x, t^n) &= u(x_j, t^n) - \frac{\partial u}{\partial x}(x_j, t^n)\Delta x + \frac{1}{2}\frac{\partial^2 u}{\partial x^2}(x_j, t^n)\Delta x^2 + \mathcal{O}(\Delta x^3) \, , \end{aligned} \tag{3.21}$$

Subtracting now the two expressions and dividing by $2\Delta x$ we eliminate the first-order terms and obtain

$$\left.\frac{\partial u}{\partial x}\right|_j^n = \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} + \mathcal{O}(\Delta x^2) \, , \tag{3.22}$$

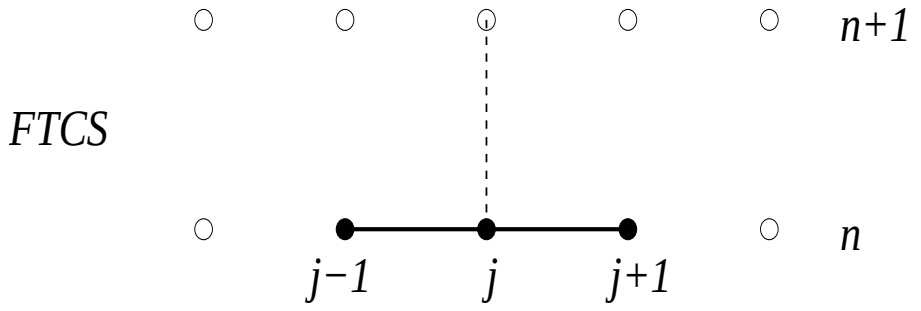

Figure 3.4: Schematic diagram of a FTCS evolution scheme.

Using now the second-order accurate operator (3.22) we can finite-difference equation (3.2) through the so called FTCS (Forward-Time-Centered-Space) scheme in which a first-order approximation is used for the time derivative, but a second order one for the spatial one. Using the a finite-difference notation, the FTCS is then expressed as

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = -v\left(\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x}\right) + \mathcal{O}(\Delta t, \Delta x^2) \, , \tag{3.23}$$

so that

$$u_j^{n+1} = u_j^n - \frac{v\Delta t}{2\Delta x}(u_{j+1}^n - u_{j-1}^n) + \mathcal{O}(\Delta t^2, \Delta x^2) \, , \tag{3.24}$$

or more generically, for a system of linear hyperbolic equations

$$\boxed{\mathbf{U}_j^{n+1} = \mathbf{U}_j^n - \frac{\Delta t}{2\Delta x}\left[\mathbf{F}_{j+1}^n - \mathbf{F}_{j-1}^n\right] + \mathcal{O}(\Delta t^2, \Delta x^2)}\,, \tag{3.25}$$

The stencil for the finite- differencing (3.24) is shown symbolically in Fig. 3.4.

Disappointingly, the FTCS scheme is *unconditionally unstable*: i.e., the numerical solution will be destroyed by numerical errors which will be certainly produced and grow exponentially. This is shown in Fig. 3.5 where we show the time evolution of a Gaussian using an FTCS scheme 100 gridpoints. The analytic solution at time $t = 0.3$ is shown with a solid line the dashed lines are used to represent the numerical solution at the same time. Note that the solution plotted here refers to a time which is 10 times smaller than the one in Fig. 3.3. Soon after $t \simeq 0.3$ the exponentially growing modes appear, rapidly destroying the solution.

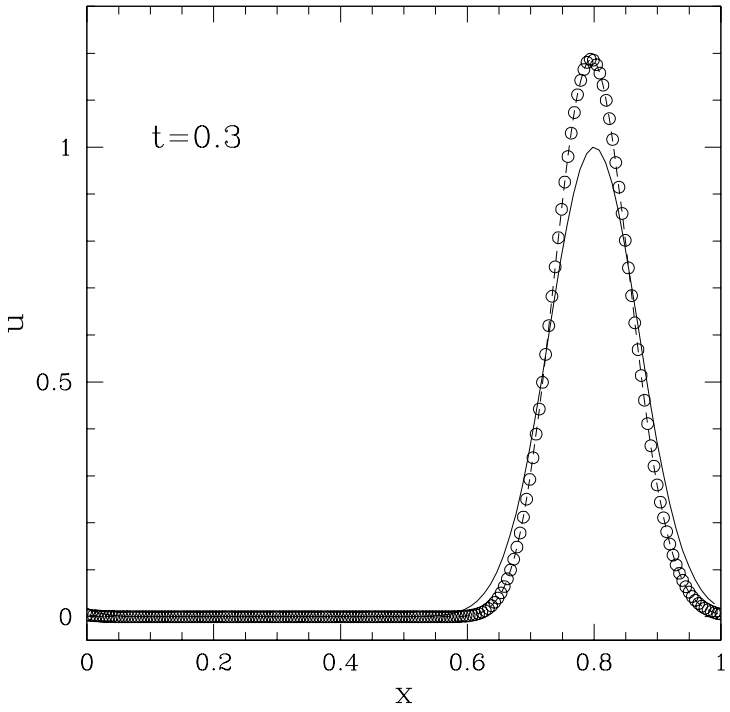

Figure 3.5: Time evolution of a Gaussian using an FTCS scheme with $v = 1$ and 100 gridpoints. The analytic solution at time $t = 0.3$ is shown with a solid line, while the dashed line is the numerical solution at the same time. Soon after $t \simeq 0.3$ the exponentially growing modes appear, rapidly destroying the solution.

Applying the definition (3.13) to equation (3.23) and few algebraic steps lead to an amplification factor

$$\xi(\Delta t, \Delta x, k) = 1 - i\frac{v\Delta t}{\Delta x}\sin(k\Delta x) \; . \tag{3.26}$$

whose squared modulus is

$$|\xi|^2 = 1 + \left[\frac{v\Delta t}{\Delta x}\sin(k\Delta x)\right]^2 > 1 \; , \tag{3.27}$$

thus proving the unconditional instability of the FTCS scheme. Because of this, the FTCS scheme is rarely used and will not produce satisfactory results but for a very short timescale as compared to the typical crossing time of the physical problem under investigation.

A final aspect of the von Neumann stability worth noticing is that it is a *necessary* but *not sufficient* condition for stability. In other words, a numerical scheme that appears stable with respect to a von Neumann stability analysis might still be unstable.

3.3 The 1D Lax-Friedrichs scheme: $\mathcal{O}(\Delta t, \Delta x^2)$

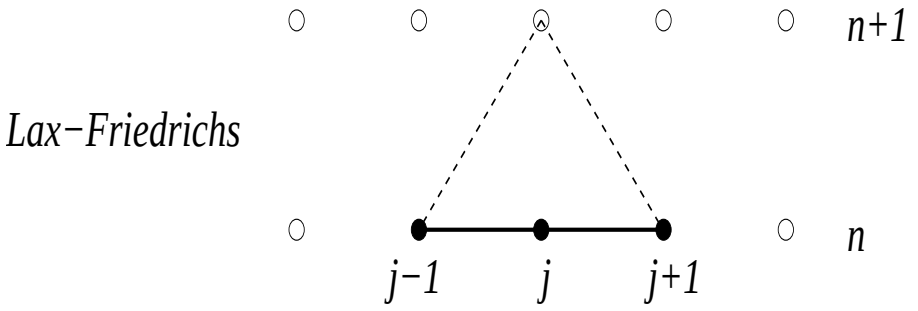

Figure 3.6: Schematic diagram of a Lax-Friedrichs evolution scheme.

A solution to the stability problems offered by the FTCS scheme was proposed by Lax and Friedrichs. The basic idea is very simple and is based on replacing, in the FTCS formula (3.23), the term u_j^n with its spatial average, i.e. $u_j^n = (u_{j+1}^n + u_{j-1}^n)/2$, so as to obtain for an advection equation

$$u_j^{n+1} = \frac{1}{2}(u_{j+1}^n + u_{j-1}^n) - \frac{v\Delta t}{2\Delta x}(u_{j+1}^n - u_{j-1}^n) + \mathcal{O}(\Delta t^2, \Delta x^2) \; , \tag{3.28}$$

and, for a system of linear hyperbolic equations

$$\boxed{\mathbf{U}_j^{n+1} = \frac{1}{2}(\mathbf{U}_{j+1}^n + \mathbf{U}_{j-1}^n) - \frac{\Delta t}{2\Delta x}\left[\mathbf{F}_{j+1}^n - \mathbf{F}_{j-1}^n\right] + \mathcal{O}(\Delta t^2, \Delta x^2)} \; . \tag{3.29}$$

The schematic diagram of a Lax-Friedrichs evolution scheme is shown in Fig. 3.6. Perhaps surprisingly, the algorithm (3.29) is now *conditionally stable* as can be verified through a von Neumann stability analysis. Proceeding as done for the FTCS scheme and using (3.13) in (3.29) we would obtain an amplification factor whose modulus squared is

$$|\xi|^2 = 1 - \sin^2(k\Delta x)\left[1 - \left(\frac{v\Delta t}{\Delta x}\right)^2\right] , \tag{3.30}$$

which is less than one as long as the CFL condition is satisfied.

Although not obvious, the correction introduced by the Lax-Friedrichs scheme is equivalent to the introduction of a *numerical dissipation* (viscosity). To see this, we rewrite (3.29) so that it clearly appears as a correction to (3.23):

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = -v\left(\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x}\right) + \frac{1}{2}\left(\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta t}\right) . \tag{3.31}$$

This is exactly the finite-difference representation of the equation

$$\frac{\partial u}{\partial t} + v\frac{\partial u}{\partial x} = \frac{1}{2}\left(\frac{\Delta x^2}{\Delta t}\right)\frac{\partial^2 u}{\partial x^2} , \tag{3.32}$$

where a diffusion term, $\propto \partial^2 u/\partial x^2$, has appeared on the right hand side. To prove this we recall that from the Taylor expansion we have:

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_j^n = \frac{1}{\Delta x}\left[\left.\frac{\partial u}{\partial x}\right|_{j+1}^n - \left.\frac{\partial u}{\partial x}\right|_j^n\right] + \mathcal{O}(\Delta x) \tag{3.33}$$

and with a first-order representation of the spatial derivative:

$$\left.\frac{\partial u}{\partial x}\right|_j^n = \frac{u_j^n - u_{j-1}^n}{\Delta x} + \mathcal{O}(\Delta x) \tag{3.34}$$

we eventually obtain

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_j^n = \frac{1}{\Delta x^2}[u_{j+1}^n - 2u_j^n + u_{j-1}^n] + \mathcal{O}(\Delta x) . \tag{3.35}$$

Note that although the expression for the second derivative in (3.35) is only $\mathcal{O}(\Delta x)$, it is multiplied by $(\Delta x^2)/\Delta t = \mathcal{O}(\Delta x)$ in equation (3.32), thus making the expression $\mathcal{O}(\Delta x^2)$ overall.

A reasonable objection could be made for the fact that the Lax-Friedrichs scheme has changed the equation whose solution one is interested in [i.e. eq. (3.2)] into a new equation,

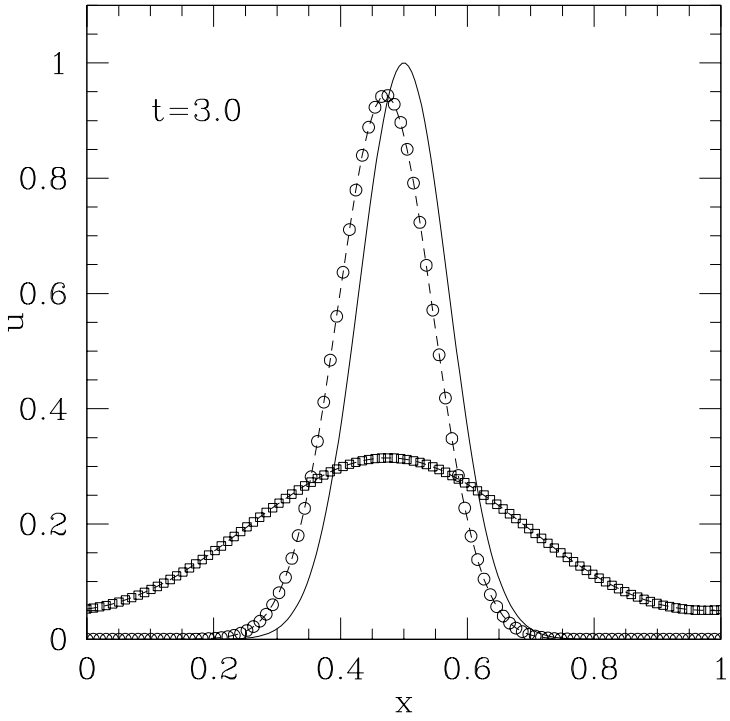

Figure 3.7: This is the same as in Fig. 3.3 but for a Lax-Friedrichs scheme. Note how the scheme is stable but also suffers from a considerable dissipation.

in which a spurious numerical dissipation has been introduced [i.e. eq. (3.32)]. Unless $|v|\Delta t = \Delta x$, $|\xi| < 1$ and the amplitude of the wave is doomed to decrease (see Fig. 3.7).

However, such objection can be easily circumvented. The key aspect in any numerical representation of a physical phenomenon is the determination of the length scale over which we need to achieve an accurate description. In a finite difference approach, this length scale must necessarily encompass many grid points and for which $k\Delta x \ll 1$. In this case, expression (3.30) clearly shows that the amplification factor is very close to 1 and the effects of dissipation are therefore small. Note that this is true also for the FTCS scheme so that on these scales the stable and unstable schemes are equally accurate. On the very small scales however, which we are not of interest to us, $k\Delta x \sim 1$ and the stable and unstable schemes are radically different. The first one will be simply inaccurate, the second one will have exponentially growing errors which will rapidly destroy the whole solution. It is rather obvious that stability and inaccuracy are by far preferable to instability, especially if the accuracy is lost over wavelengths that are not of interest or when it can be recovered easily by using

more refined grids.

3.4 The 1D Leapfrog scheme: $\mathcal{O}(\Delta t^2, \Delta x^2)$

Both the FTCS and the Lax-Friedrichs are "one-level" schemes with first-order approximation for the time derivative and a second-order approximation for the spatial derivative. In those circumstances $v\Delta t$ should be taken significantly smaller than Δx (to achieve the desired accuracy), well below the limit imposed by the Courant condition.

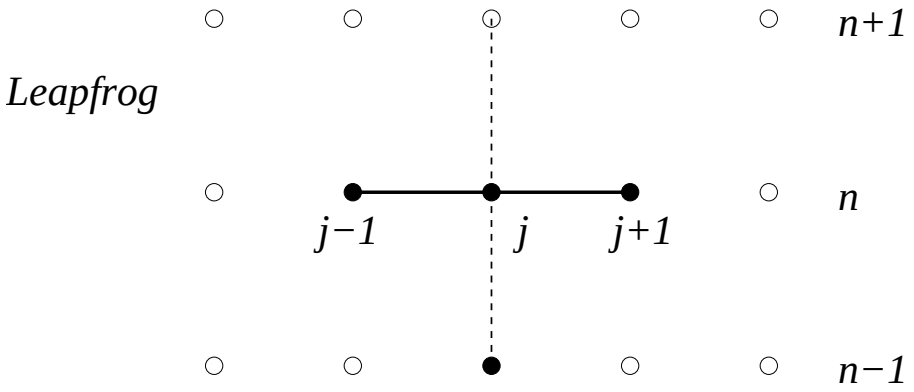

Figure 3.8: Schematic diagram of a Leapfrog evolution scheme.

Second-order accuracy in time can be obtained if we insert

$$\left.\frac{\partial u}{\partial t}\right|_j^n = \frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t} + \mathcal{O}(\Delta t^2) \,, \tag{3.36}$$

in the FTCS scheme, to find the *Leapfrog* scheme

$$u_j^{n+1} = u_j^{n-1} - \frac{\Delta t}{\Delta x}\left[u_{j+1}^n - u_{j-1}^n\right] + \mathcal{O}(\Delta t^3, \Delta x^2) \,, \tag{3.37}$$

where it should be noted that the factor 2 in Δx cancels the equivalent factor 2 in Δt.

For a set of linear equations, the Leapfrog scheme simply becomes

$$\boxed{\mathbf{U}_j^{n+1} = \mathbf{U}_j^{n-1} - \frac{\Delta t}{\Delta x}\left[\mathbf{F}_{j+1}^n - \mathbf{F}_{j-1}^n\right] + \mathcal{O}(\Delta t^3, \Delta x^2)}, \tag{3.38}$$

and the schematic diagram of a Leapfrog evolution scheme is shown in Fig. 3.8.

Also for the case of a Leapfrog scheme there are a number of aspects that should be noticed:

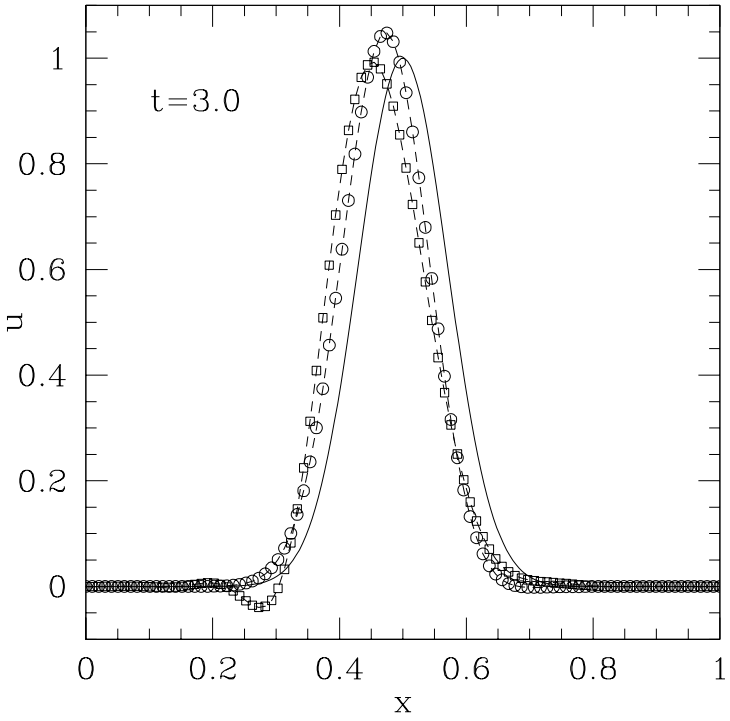

Figure 3.9: This is the same as in Fig. 3.3 but for a Leapfrog scheme. Note how the scheme is stable and does not suffers from a considerable dissipation even for low CFL factors. However, the presence of a little "dip" in the tail of the Gaussian for the case of $c_{\rm CFL} = 0.5$ is the result of the dispersive nature of the numerical scheme.

- In a Leapfrog scheme that is Courant stable, there is no amplitude dissipation (i.e. $|\xi|^2 = 1$). In fact, a von Neumann stability analysis yields

$$\xi = -i\alpha \sin(k\Delta x) \pm \sqrt{1 - [\alpha \sin(k\Delta x)]^2}\,, \tag{3.39}$$

where

$$\alpha \equiv \frac{v\Delta t}{\Delta x}\,, \tag{3.40}$$

and so that

$$|\xi|^2 = \alpha^2 \sin^2(k\Delta x) + \{1 - [\alpha \sin(k\Delta x)]^2\} = 1 \qquad \forall\ \alpha \leq 1\,. \tag{3.41}$$

As a result, the squared modulus of amplification factor is always 1, provided the CFL condition is satisfied (cf. Fig. 3.11).

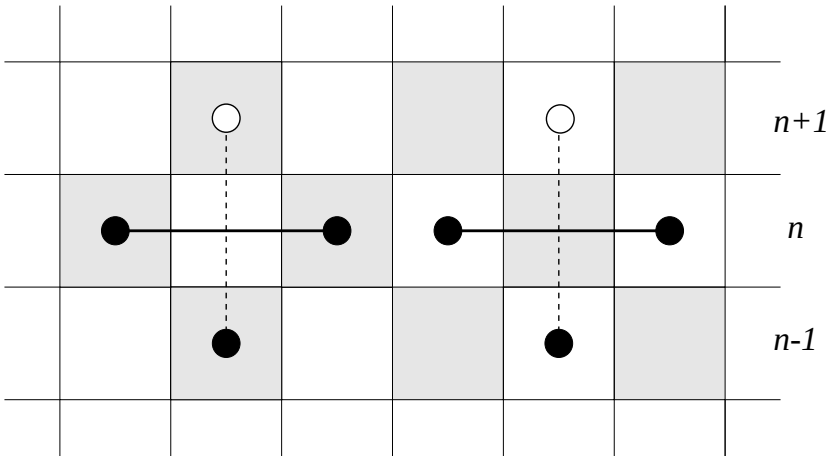

Figure 3.10: Schematic diagram of the decoupled grids in a Leapfrog evolution scheme.

- The Leapfrog scheme is a two-level scheme, requiring records of values at time-steps n and $n-1$ to get values at time-step $n+1$. This is clear from expression (4.24) and cannot be avoided by means of algebraic manipulations.

- The major disadvantage of this scheme is that odd and even mesh points are completely decoupled (see Fig. 8).

  In principle, the solutions on the black and white squares are identical. In practice, however, their differences increase as the time progresses. This effect, which becomes evident only on timescales much longer then the crossing timescale, can be cured either by discarding one of the solutions or by adding a dissipative term of the type

$$\ldots + \epsilon(u^n_{j+1} - 2u^n_{j+1} + u^n_{j+1}) \, , \tag{3.42}$$

  in the right-hand-side of (4.17), where $\epsilon \ll 1$ is an adjustable coefficient.

3.5 The 1D Lax-Wendroff scheme: $\mathcal{O}(\Delta t^2, \Delta x^2)$

The Lax-Wendroff scheme is the second-order accurate extension of the Lax-Friedrichs scheme. As for the case of the Leapfrog scheme, in this case too we need two time-levels to obtain the solution at the new time-level.

There are a number of different ways of deriving the Lax-Wendroff scheme but it is probably useful to look at it as to a combination of the Lax-Friedrichs scheme and of the Leapfrog scheme. In particular a Lax-Wendroff scheme can be obtained as

1. A Lax-Friedrichs scheme with half step:

$$\mathbf{U}^{n+\frac{1}{2}}_{j+\frac{1}{2}} = \frac{1}{2}\left[\mathbf{U}^n_{j+1} + \mathbf{U}^n_j\right] - \frac{\Delta t}{2\Delta x}\left[\mathbf{F}^n_{j+1} - \mathbf{F}^n_j\right] + \mathcal{O}(\Delta t^3, \Delta x^2) \ ,$$

$$\mathbf{U}^{n+\frac{1}{2}}_{j-\frac{1}{2}} = \frac{1}{2}\left[\mathbf{U}^n_j + \mathbf{U}^n_{j-1}\right] - \frac{\Delta t}{2\Delta x}\left[\mathbf{F}^n_j - \mathbf{F}^n_{j-1}\right] + \mathcal{O}(\Delta t^3, \Delta x^2) \ ,$$

   where $\Delta t/(2\Delta x)$ comes from having used a timestep $\Delta t/2$;

2. The evaluation of the fluxes $\mathbf{F}^{n+\frac{1}{2}}_{j\pm\frac{1}{2}}$ from the values of $\mathbf{U}^{n+\frac{1}{2}}_{j\pm\frac{1}{2}}$

3. A Leapfrog "half-step":

$$\boxed{\mathbf{U}^{n+1}_j = \mathbf{U}^n_j - \frac{\Delta t}{\Delta x}\left[\mathbf{F}^{n+\frac{1}{2}}_{j+\frac{1}{2}} - \mathbf{F}^{n+\frac{1}{2}}_{j-\frac{1}{2}}\right] + \mathcal{O}(\Delta t^3, \Delta x^2)}\,. \tag{3.43}$$

The schematic diagram of a Lax-Wendroff evolution scheme is shown in Fig. 3.11 and the application of this scheme to the advection equation (3.2) is straightforward. More specifically, the "half-step" values can be calculated as

$$u^{n+1/2}_{j\pm1/2} = \frac{1}{2}\left(u^n_j + u^u_{j\pm1}\right) \mp \frac{v\Delta t}{2\Delta x}\left(u^n_{j\pm1} - u^n_j\right) + \mathcal{O}(\Delta t^3, \Delta x^2) \ , \tag{3.44}$$

so that the solution at the new time-level will then be

$$u^{n+1}_j = u^n_j - \frac{v\Delta t}{2\Delta x}\left(u^n_{j+1} - u^n_{j-1}\right) + \frac{v^2\Delta t^2}{2\Delta x^2}\left(u^n_{j+1} - 2u^n_j + u^n_{j-1}\right) + \mathcal{O}(\Delta t^3, \Delta x^2) \ . \tag{3.45}$$

Aspects of a Lax-Wendroff scheme worth noticing are:

- In the Lax-Wendroff scheme there might be some amplitude dissipation. In fact, a von Neumann stability analysis yields

$$\xi = 1 - i\alpha\sin(k\Delta x) - \alpha^2\left[1 - \cos(k\Delta x)\right] \ , \tag{3.46}$$

  so that the squared modulus of the amplification factor is

$$|\xi|^2 = 1 - \alpha^2(1-\alpha^2)\left[1 - \cos^2(k\Delta x)\right] \ . \tag{3.47}$$

  As a result, the von Neumann stability criterion $|\xi|^2 \leq 1$ is satisfied as long as $\alpha^2 \leq 1$, or equivalently, as long as the CFL condition is satisfied. (cf. Fig. 10). It should be noticed, however, that unless $\alpha^2 = 1$, then $|\xi|^2 < 1$ and some amplitude dissipation is present. In this respect, the dissipative properties of the Lax-Friedrichs scheme are not completely lost in the Lax-Wendroff scheme but are much less severe (cf. Figs. 5 and 10).

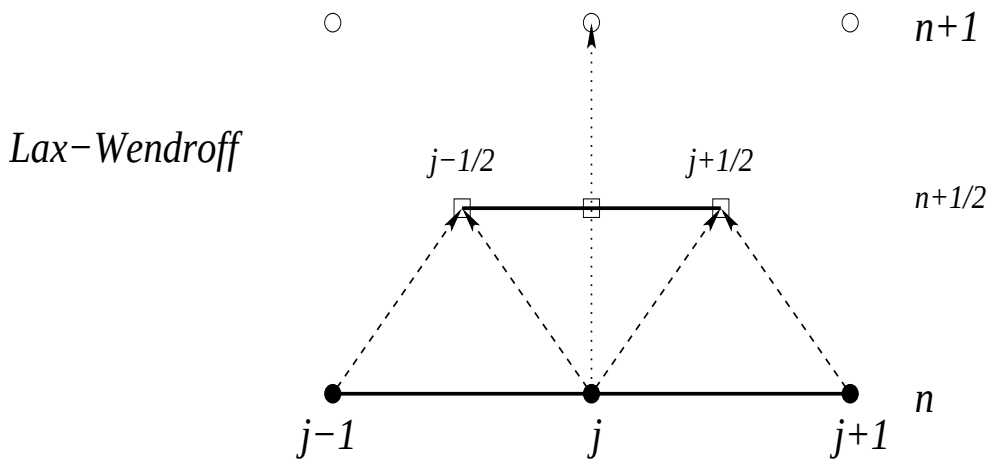

Figure 3.11: Schematic diagram of a Lax-Wendroff evolution scheme.

- The Lax-Wendroff scheme is a two-level scheme, but can be recast in a one-level form by means of algebraic manipulations. This is clear from expressions (4.23) and (4.24) where quantities at time-levels n and $n+1$ only appear.

3.6 The 1D ITCN scheme: $\mathcal{O}(\Delta t^2, \Delta x^2)$

The idea behind the iterative Crank-Nicholson scheme (ITCN) is that of transforming a stable implicit method (i.e. the Crank-Nicholson scheme) into an explicit one through a series of iteration. To see how to do this in practice, consider differencing the advection equation (3.2) having a centred space derivative but with the time derivative being backward centred, i.e.

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \frac{u_{j+1}^{n+1} - u_{j-1}^{n+1}}{2\Delta x} \,. \tag{3.48}$$

This scheme is also known as "backward in time, centred in space" or BTCS and has amplification factor

$$\xi = \frac{1}{1 + i\alpha \sin k\Delta x} \,, \tag{3.49}$$

so that $|\xi|^2 < 1$ for any choice of α, thus making the method unconditionally stable. The Crank-Nicholson (CN) scheme is a second-order accurate method obtained by averaging a BTCS and a FTCS method or, in other words, equations (3.23) and (3.48). Doing so one then finds

$$\xi = \frac{1 + i\alpha \sin k\Delta x/2}{1 - i\alpha \sin k\Delta x/2} \,. \tag{3.50}$$

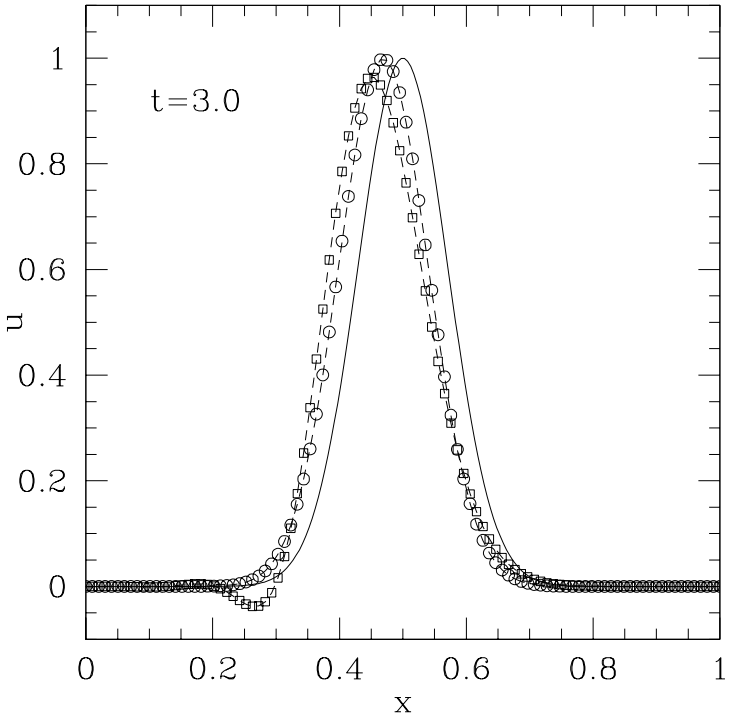

Figure 3.12: This is the same as in Fig. 3.3 but for a Lax-Wendroff scheme. Note how the scheme is stable and does not suffers from a considerable dissipation even for low CFL factors. However, the presence of a little "dip" in the tail of the Gaussian for the case of $c_{\text{CFL}} = 0.5$ is the result of the dispersive nature of the numerical scheme.

so that the method is stable. Note that although one averages between an explicit and an explicit scheme, terms containing u^{n+1} survive on the right hand side of equation (3.48), thus making the CN scheme implicit.

The first iteration of iterated Crank-Nicholson starts by calculating an intermediate variable ${}^{(1)}\tilde{u}$ using equation (3.23):

$$\frac{{}^{(1)}\tilde{u}_j^{n+1} - u_j^n}{\Delta t} = \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} \ . \tag{3.51}$$

Then another intermediate variable ${}^{(1)}\bar{u}$ is formed by averaging:

$${}^{(1)}\bar{u}_j^{n+1/2} = \frac{1}{2}\left({}^{(1)}\tilde{u}_j^{n+1} + u_j^n\right) \ . \tag{3.52}$$

Finally the timestep is completed by using equation (3.23) again with $\bar{u}$ on the right-hand

side:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \frac{{}^{(1)}\bar{u}_{j+1}^{n+1/2} - {}^{(1)}\bar{u}_{j-1}^{n+1/2}}{2\Delta x} \,. \tag{3.53}$$

Iterated Crank-Nicholson with two iterations is carried out in the same way. After steps (3.51) and (3.52), we calculate

$$\frac{{}^{(2)}\tilde{u}_j^{n+1} - u_j^n}{\Delta t} = \frac{{}^{(1)}\bar{u}_{j+1}^{n+1/2} - {}^{(1)}\bar{u}_{j-1}^{n+1/2}}{2\Delta x}, \tag{3.54}$$

$${}^{(2)}\bar{u}_j^{n+1/2} = \tfrac{1}{2}\big({}^{(2)}\tilde{u}_j^{n+1} + u_j^n\big). \tag{3.55}$$

Then the final step is computed analogously to equation (3.53):

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \frac{{}^{(2)}\bar{u}_{j+1}^{n+1/2} - {}^{(2)}\bar{u}_{j-1}^{n+1/2}}{2\Delta x}. \tag{3.56}$$

Further iterations can be carried out following the same logic.

To investigate the stability of these iterated schemes we compute the amplification factors relative to the different iterations to be

$${}^{(0)}\xi = 1 + 2i\beta \,, \tag{3.57}$$

$${}^{(1)}\xi = 1 + 2i\beta - 2\beta^2 \,, \tag{3.58}$$

$${}^{(2)}\xi = 1 + 2i\beta - 2\beta^2 - 2i\beta^3 \,, \tag{3.59}$$

$${}^{(3)}\xi = 1 + 2i\beta - 2\beta^2 - 2i\beta^3 + 2\beta^4 \,, \tag{3.60}$$

where $\beta \equiv (\alpha/2)\sin k\Delta x$, and ${}^{(0)}\xi$ corresponds to the FTCS scheme. Note that the amplification factors (3.57) correspond to those one would obtain by expanding equation (3.50) in powers of β.

Computing the squared moduli of (3.57) one encounters an alternating and recursive pattern. In particular, iterations 0 and 1 are unstable ($|\xi|^2 > 1$); iterations 2 and 3 are stable ($|\xi|^2 < 1$) provided $\beta^2 \leq 1$; iterations 4 and 5 are also unstable; iterations 6 and 7 are stable provided $\beta^2 \leq 1$; and so on. Imposing the stability for all wavenumbers k, we obtain $\alpha^2/4 \leq 1$, or $\Delta t \leq 2\Delta x$ which is just the CFL condition [the factor 2 is inherited by the factor 2 in equation (3.23)].

In other words, while the magnitude of the amplification factor for iterated Crank-Nicholson does approach 1 as the number of iterations becomes infinite, the convergence is not monotonic. The magnitude oscillates above and below 1 with ever decreasing oscillations. All the iterations leading to $|\xi|^2$ above 1 are unstable, although the instability might be very slowly growing as the number of iterations increases. Because the truncation error is not modified by the number of iterations and is always $\mathcal{O}(\Delta t^2, \Delta x^2)$, a number of iterations larger than two is never useful; three iterations, in fact, would simply amount to a larger computational cost.

Chapter 4

The Wave Equation in 1D

The wave equation in 1D has the general form:

$$\frac{\partial^2 u}{\partial t^2} = v^2 \frac{\partial^2 u}{\partial x^2} \,, \tag{4.1}$$

where, for simplicity, we will assume that v is constant (i.e. $v \neq v(x)$), thus restricting our attention to linear problems. It is much more convenient to rewrite (4.1) as a system of coupled first-order conservative partial differential equations (PDE). For this we set

$$r = v \frac{\partial u}{\partial x} \,, \tag{4.2}$$

$$s = \frac{\partial u}{\partial t} \,, \tag{4.3}$$

so that (4.1) can be rewritten as a system of 3 coupled, first-order differential equations

$$\begin{cases} \dfrac{\partial r}{\partial t} = v \dfrac{\partial s}{\partial x} \,, \\ \dfrac{\partial s}{\partial t} = v \dfrac{\partial r}{\partial x} \,, \\ \dfrac{\partial u}{\partial t} = s \,, \end{cases} \tag{4.4}$$

where it should be noted that the equations have the time derivative of *one* variable that is proportional to the space derivative of the *other* variable. This breaks the advective nature of the equation discussed in the previous Chapter and will prevent, for instance, the use of an upwind scheme.

In vector notation the system (4.4) can be symbolically written as

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} = 0 \,, \tag{4.5}$$

where

$$\mathbf{U} = \begin{pmatrix} r \\ s \end{pmatrix} \,, \qquad \text{and} \qquad \mathbf{F}(\mathbf{U}) = \begin{pmatrix} 0 & -v \\ -v & 0 \end{pmatrix} \mathbf{U} \,. \tag{4.6}$$

4.1 The FTCS Scheme

As mentioned in the previous Chapter, the upwind method cannot be applied to the solution of the wave equation and the simplest, first-order in time method we can use for the solution of the wave equation is therefore given by the FTCS scheme. Applying it to the first-order system (4.4) and obtain

$$r_j^{n+1} = r_j^n + \frac{v\Delta t}{2\Delta x}(s_{j+1}^n - s_{j-1}^n) + \mathcal{O}(\Delta t^2, \Delta x^2) \,, \tag{4.7}$$

$$s_j^{n+1} = s_j^n + \frac{v\Delta t}{2\Delta x}(r_{j+1}^n - r_{j-1}^n) + \mathcal{O}(\Delta t^2, \Delta x^2) \,, \tag{4.8}$$

Once the value of s_j^{n+1} has been calculated, the value of u can be integrated in time according to equation (4.3) so that

$$u_j^{n+1} = u_j^n + \Delta t s_j^n + \mathcal{O}(\Delta t^2, \Delta x^2) \,, \tag{4.9}$$

where it should be noted that u^{n+1} has the same truncation error of r^{n+1} and s^{n+1}.

Of course, we do not expect that the FTCS scheme applied to the solution of the wave equation will provide a stable evolution and this is clearly shown in Fig. 4.1 which reports the solution of equations (4.7), (4.7) and (4.9) having as initial conditions a Gaussian centered at $x = 5$ with unit variance. Different lines show the solution at different times and is apparent how the initial profile splits in two part propagating in two opposite directions. During the evolution, however, the error grows (note that the peaks of the two packets increase with time) and in about one crossing time the short wavelength noise appears (this is shown by the small sharp peaks produced when the wave has left the numerical grid). When this happens, the numerical errors have grown to be comparable with the solution, which will be rapidly destroyed.

4.2 The Lax-Friedrichs Scheme

As done in the previous Section, we can apply the Lax-Friedrichs scheme to the solution of the wave equation through the first-order system (4.4) and easily obtain

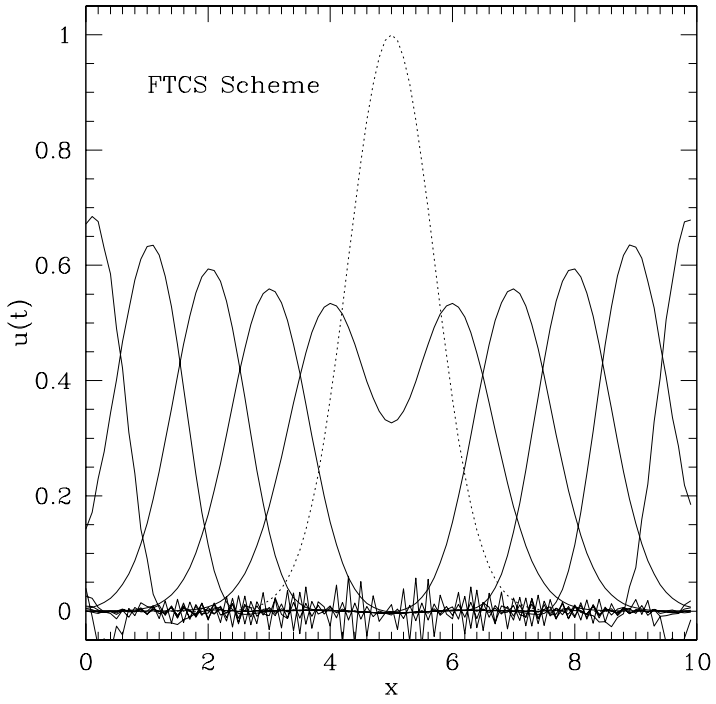

Figure 4.1: Plot of the time evolution of the wave equation when the FTCS scheme is used. The initial conditions were given by a Gaussian centered at $x = 5$ with unit variance and are shown with the dotted line. Note the growth of the wave crests and the appearance of short wavelength noise. When this happens, the numerical errors have grown to be comparable with the solution which will be rapidly destroyed.

$$r_j^{n+1} = \frac{1}{2}(r_{j+1}^n + r_{j-1}^n) + \frac{v\Delta t}{2\Delta x}(s_{j+1}^n - s_{j-1}^n) + \mathcal{O}(\Delta t^2, \Delta x^2)\,, \tag{4.10}$$

$$s_j^{n+1} = \frac{1}{2}(s_{j+1}^n + s_{j-1}^n) + \frac{v\Delta t}{2\Delta x}(r_{j+1}^n - r_{j-1}^n) + \mathcal{O}(\Delta t^2, \Delta x^2)\,, \tag{4.11}$$

Also in this case, once the value of s_j^{n+1} has been calculated, the value for u_j^{n+1} can be computed according to (4.9).

The solution of equations (4.10), (4.10) and (4.9) with the same initial data used in Fig. 4.1 is shown in Fig. 4.2. Note that we encounter here the same behaviour found in the solution of the advection equation and in particular it is apparent the progressive diffusion of the two travelling packets which spread over the numerical grid as they propagate.

As expected, the evolution is not stable and no error growth is visible many crossing times after the wave has left the numerical grid.

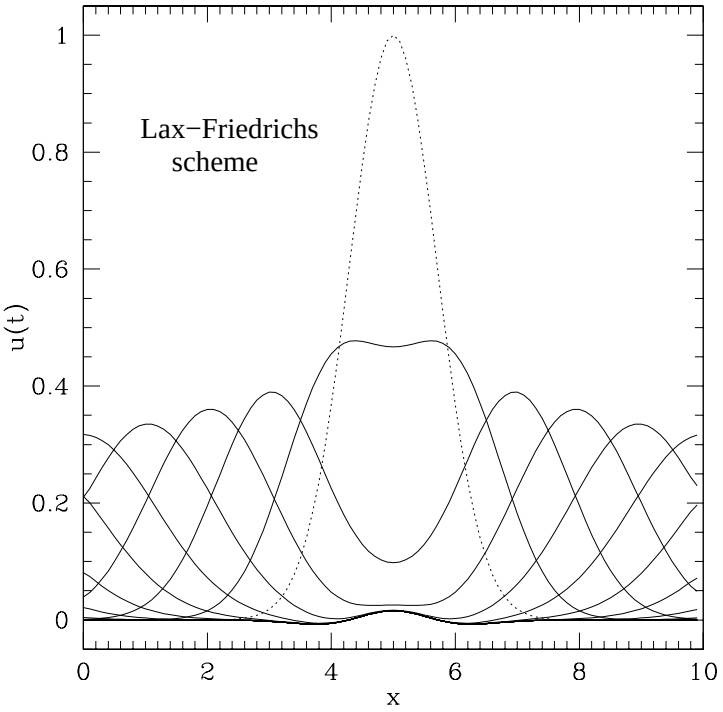

Figure 4.2: The same as in Fig. 4.1 but when the Lax-Friedrichs scheme is used. Note the absence of the late time instabilities but also the effects of the numerical diffusion that widens and lowers the wave fronts.

4.3 The Leapfrog Scheme

We can adapt the Leapfrog scheme to equations (4.4) for the solution of the wave equation in one dimension, centering variables on appropriate half-mesh points

$$r^n_{j+\frac{1}{2}} \equiv v\frac{\partial u}{\partial x}\Big|^n_{j+\frac{1}{2}} = v\frac{u^n_{j+1}-u^n_j}{\Delta x}+\mathcal{O}(\Delta x)\,, \tag{4.12}$$

$$s^{n+\frac{1}{2}}_j \equiv \frac{\partial u}{\partial t}\Big|^{n+\frac{1}{2}}_j = \frac{u^{n+1}_j-u^n_j}{\Delta t}+\mathcal{O}(\Delta t)\,, \tag{4.13}$$

and then considering the Leapfrog representation of equations (4.4)

$$r^{n+1}_{j+\frac{1}{2}} = r^n_{j+\frac{1}{2}}+v\frac{\Delta t}{\Delta x}\left(s^{n+\frac{1}{2}}_{j+1}-s^{n+\frac{1}{2}}_j\right)+\mathcal{O}(\Delta t^2,\Delta x)\,, \tag{4.14}$$

$$s^{n+\frac{1}{2}}_j = s^{n-\frac{1}{2}}_j+v\frac{\Delta t}{\Delta x}\left(r^n_{j+\frac{1}{2}}-r^n_{j-\frac{1}{2}}\right)+\mathcal{O}(\Delta t^2,\Delta x)\,, \tag{4.15}$$

A simple substitution of (4.12) and (4.13) into (4.14) and (4.15) shows how the Leapfrog representation of the wave equation is nothing but its second-order differencing:

$$\frac{u^{n+1}_j-2u^n_j+u^{n-1}_j}{\Delta t^2} = v^2\left(\frac{u^n_{j+1}-2u^n_j+u^n_{j-1}}{\Delta x^2}\right)+\mathcal{O}(\Delta t,\Delta x)\,, \tag{4.16}$$

so that the solution at the new time-level is

$$u^{n+1}_j = \alpha^2u^n_{j+1}+2u^n_j\left(1-\alpha^2\right)+\alpha^2u^n_{j-1}-u^{n-1}_j+\mathcal{O}(\Delta t^3,\Delta x^2)\,, \tag{4.17}$$

where

$$\alpha \equiv v\frac{\Delta t}{\Delta x}\,. \tag{4.18}$$

Note that as formulated in (4.17), the Leapfrog scheme has been effectively recast into a "one-level" scheme. As in the previous examples, the new value for the wave variable u is finally computed after the integration in time of s. Here however, to preserve the second-order accuracy in time it is necessary to average the time derivative s between n and $n+1$ to obtain

$$u^{n+1}_j = u^n_j+\frac{\Delta t}{2}(s^{n+1}_j+s^n_j)+\mathcal{O}(\Delta t^3,\Delta x^2)\,. \tag{4.19}$$

The solution of equations (4.17) and (4.19) with the same initial data used in Fig. 4.1 is shown in Fig. 4.3. Note that we do not encounter here a significant amount of diffusion for the two travelling wave packets. As expected, the evolution is stable and no error growth is visible many crossing times after the wave has left the numerical grid.

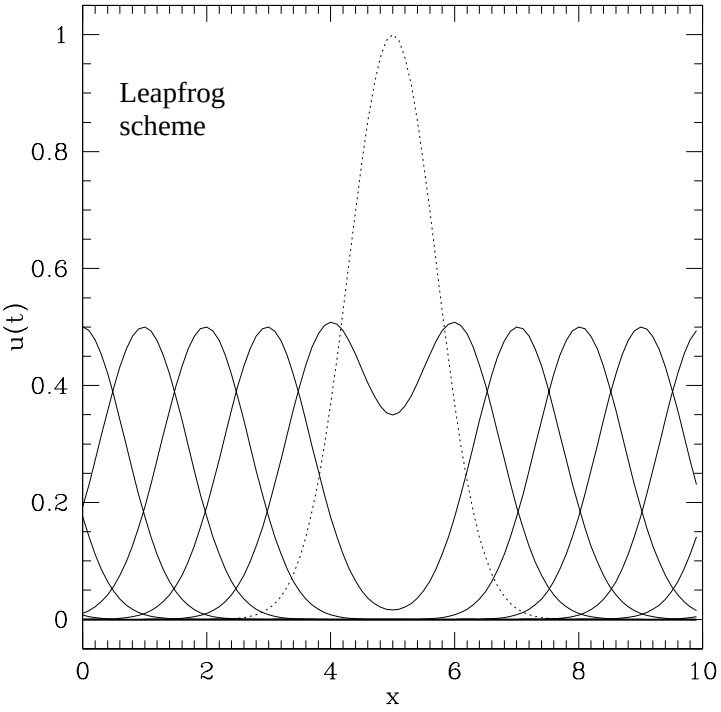

Figure 4.3: The same as in Fig. 4.1 but when the Leapfrog scheme is used. Note the absence of the late time instabilities and of the effects of the numerical diffusion.

4.4 The Lax-Wendroff Scheme

Also in the case, the application of this scheme to our system of equations (4.4) is straightforward. We can start with the time evolution of the variable r to obtain

$$r_j^{n+1} = r_j^n + v\frac{\Delta t}{\Delta x}\left(s_{j+1/2}^{n+1/2} - s_{j-1/2}^{n+1/2}\right) + \mathcal{O}(\Delta t^3, \Delta x^2)\,, \tag{4.20}$$

where the terms in the spatial derivatives are computed as

$$s_{j+1/2}^{n+1/2} = \frac{1}{2}\left(s_j^n + s_{j-1}^n\right) - \frac{v\Delta t}{2\Delta x}\left(r_{j+1}^n - r_j^n\right) + \mathcal{O}(\Delta t^3, \Delta x^2)\,, \tag{4.21}$$

$$s_{j-1/2}^{n+1/2} = \frac{1}{2}\left(s_j^n + s_{j+1}^n\right) - \frac{v\Delta t}{2\Delta x}\left(r_j^n - r_{j-1}^n\right) + \mathcal{O}(\Delta t^3, \Delta x^2)\,. \tag{4.22}$$

As done for the advection equation, it is convenient not to use equations (4.20) and (4.21) as two coupled but distinct equations and rather to combine them into two "one-level" evolution equations for r and s

$$r_j^{n+1} = r_j^n + v\frac{\Delta t}{\Delta x}\left[\frac{1}{2}(s_{j+1}^n - s_{j-1}^n) + \frac{v\Delta t}{2\Delta x}(r_{j+1}^n - 2r_j^n + r_{j-1}^n)\right] + \mathcal{O}(\Delta t^3, \Delta x^2)\,, \tag{4.23}$$

$$s_j^{n+1} = s_j^n + v\frac{\Delta t}{\Delta x}\left[\frac{1}{2}(r_{j+1}^n - r_{j-1}^n) + \frac{v\Delta t}{2\Delta x}(s_{j+1}^n - 2s_j^n + s_{j-1}^n)\right] + \mathcal{O}(\Delta t^3, \Delta x^2)\,. \tag{4.24}$$

The solution of equations (4.23), (4.24) and (4.19) with the same initial data used in Fig. 4.1 is shown in Fig. 4.4. Note that we do not encounter here a significant amount of diffusion for the two travelling wave packets. As expected, the evolution is stable and no error growth is visible many crossing times after the wave has left the numerical grid.

A final remark should be made on the last three evolution schemes considered. We have seen that all of them are stable and the last two, in particular, also show very little sign of numerical diffusion. We need at this point a way of measuring the "quality" of the solution and, more precisely, a way of estimating the overall truncation error, the dissipation and the stability of the solution. A systematic discussion of this will be presented in Section 6.2.

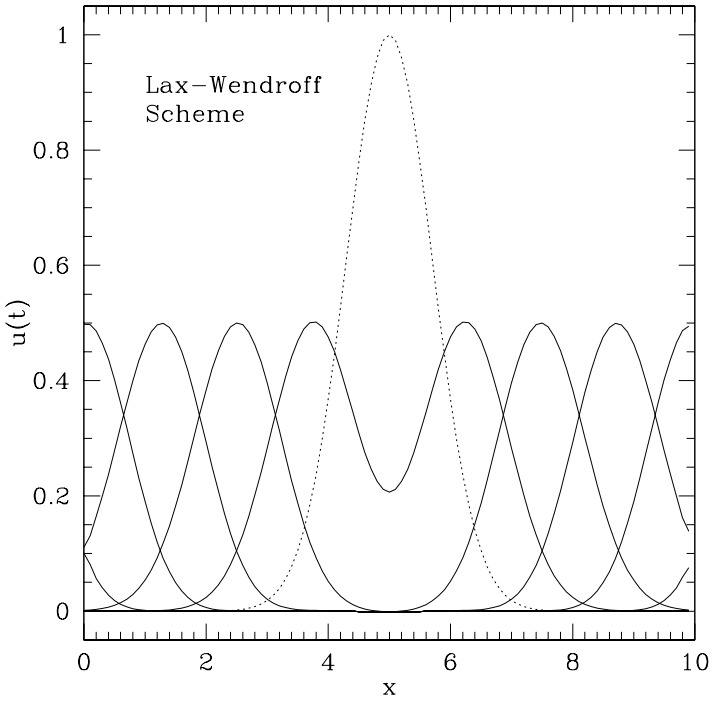

Figure 4.4: The same as in Fig. 4.1 but when the Lax-Wendroff scheme is used. Note the absence of the late time instabilities and of the effects of the numerical diffusion.

Chapter 5

Boundary Conditions

Unavoidable and common to all the numerical schemes discussed so far is the problem of treating the solution on the boundaries of the spatial grid as the time evolution proceeds. Let 1 be the first gridpoint and J the last one. It is clear from equations (3.23), (4.16), (4.23) and (4.24) that the new solution at the boundaries of the spatial grid (i.e. u_1^{n+1},u_J^{n+1}) is undetermined as it requires the values u_0^n, u_{J+1}^n. The most natural boundary conditions for the evolution of a wave equation are the so called *Sommerfeld boundary conditions* (or *radiative boundary conditions*) which will be discussed in the following Section. Other boundary conditions of general interest are:

- *Dirichlet* boundary conditions: values of the relevant quantity are imposed at the boundaries of the numerical grid. These values can be either functions of time or be held constant (cf. boundary conditions for boundary value problems);

  ◇ *"Periodic"* boundary conditions: assume that the numerical domain is topologically connected in a given direction; this is often used in cosmological simulations (and "videogames"!).

- *von Neumann* boundary conditions: values of the derivatives of the relevant quantity are imposed at the boundaries of the numerical grid. As for Dirichlet, these values can be either functions of time or be held constant (cf. boundary conditions for boundary value problems);

  ◇ *"Reflecting"* boundary conditions: mimic the presence of a reflecting boundary, i.e. of a boundary with zero transmission coefficient;

  ◇ *"Absorbing"* boundary conditions: mimic the presence of an absorbing boundary, i.e. of a boundary with unit transmission coefficient;

5.1 Outgoing Wave BCs: the outer edge

A scalar wave outgoing in the positive x-direction is described by the advection equation:

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0 \tag{5.1}$$

A finite-difference, first-order accurate representation of equation (5.1) which is centered in both time (at $n + \frac{1}{2}$) and in space (at $j + \frac{1}{2}$) is given by (see Fig. 3.11)

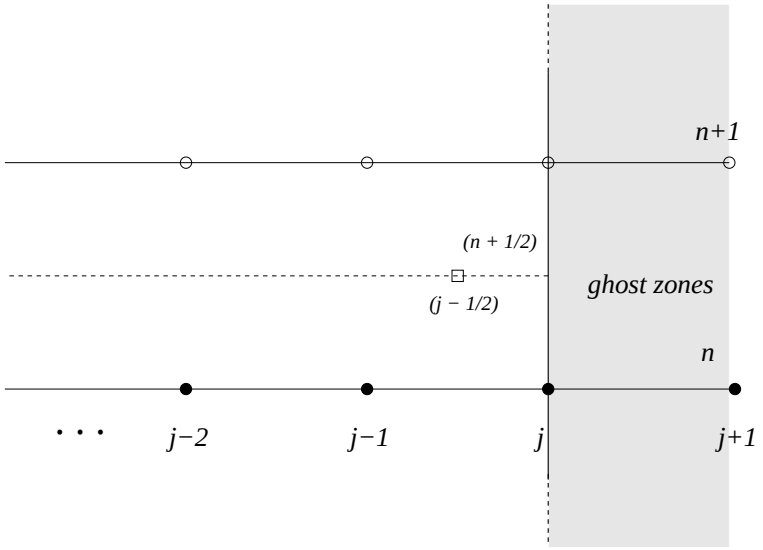

Figure 5.1: Schematic representation of the centering for a first-order, outgoing-wave Sommerfeld boundary conditions. An equivalent one can be drawn for an ingoing-wave.

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

and which leads to

$$u_{j+1}^{n+1}\left(1 + \frac{\Delta t}{\Delta x}\right) = u_j^{n+1}\left(-1 + \frac{\Delta t}{\Delta x}\right) + u_{j+1}^n\left(1 - \frac{\Delta t}{\Delta x}\right) + u_j^n\left(1 + \frac{\Delta t}{\Delta x}\right) \tag{5.2}$$

Expression (5.2) can also be written as

$$\boxed{u_{j+1}^{n+1} = u_j^n + u_j^{n+1}Q - u_{j+1}^n Q} \tag{5.3}$$

where

$$Q = \frac{1 - \Delta x/\Delta t}{1 + \Delta x/\Delta t} \,. \tag{5.4}$$

The use of expression (5.3) for the outermost grid point where the wave is outgoing will provide first-order accurate and stable boundary conditions. Note, however, that (5.3) is a discrete representation of a physical condition which would transmit the wave without reflection. In practice, however, a certain amount of reflection is always produced (the transmission coefficient is never exactly one); the residual wave is then transmitted back in the numerical box. A few reflections are usually sufficient to reduce the wave content to values below the machine accuracy.

5.2 Ingoing Wave BCs: the inner edge

Similarly, a scalar wave outgoing in the negative x-direction (or ingoing in the positive one) is described by the advection equation:

$$\frac{\partial u}{\partial t} - \frac{\partial u}{\partial x} = 0 \tag{5.5}$$

Following the same procedure discussed before, the algorithm becomes:

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

Then

$$\boxed{u_j^{n+1} = u_{j+1}^n + u_{j+1}^{n+1}Q - u_j^n Q} \tag{5.6}$$

where Q is the same quantity as for the out-going wave. If we use equations (5.3) and (5.6) to evolve the solution at time-step $n+1$ at the boundary of our spatial grid, we are guaranteed that our profile will be completely transported away, whatever integration scheme we are adopting (Leapfrog, Lax-Wendroff etc.).

5.3 Periodic Boundary Conditions

These are very simple to impose and if j is between 1 and J, they are given simply by

$$\boxed{u_1^{n+1} = u_{J-1}^{n+1}, \qquad u_J^{n+1} = u_2^{n+1}\,,} \tag{5.7}$$

In the case of a Gaussian leaving the center of the numerical grid, these boundary conditions effectively produce a reflection. The boundary conditions (5.7) force to break the

algorithm for the update scheme excluding the first and last points that need to be computed separately. An alternative procedures consists of introducing a number of *"ghost"* gridpoints outside the computational domain of interest so that the solution is calculated using always the *same stencil* for $j = 1, 2, \ldots, J$ and exploiting the knowledge of the solution also at the ghost gridpoints, e.g. 0 and $J + 1$.

In the case there is only one ghost gridpoint at either edge of the 1D grid, the boundary conditions are simply given by

$$\boxed{u_0^{n+1} = u_J^{n+1}, \qquad u_{J+1}^{n+1} = u_1^{n+1} \ .} \tag{5.8}$$

Chapter 6

Dissipation, Dispersion and Stability

We will here discuss a number of problems that often emerge when using finite-difference techniques for the solution of hyperbolic partial differential equations. In stable numerical schemes the impact of many of these problems can be suitably reduced by going to sufficiently high resolutions, but it is nevertheless important to have a simple and yet clear idea of what are the most common sources of these problems.

6.1 On the Origin of Dissipation and Dispersion

We have already seen in Chapter 3 how the Lax-Friedrichs scheme applied to a linear advection equation (3.2) yields the finite-difference expression

$$u_j^{n+1} = \frac{1}{2}(u_{j+1}^n + u_{j-1}^n) - \frac{v\Delta t}{2\Delta x}(u_{j+1}^n - u_{j-1}^n) + \mathcal{O}(\Delta t^2, \Delta x^2) \; . \tag{6.1}$$

We have also mentioned how expression (6.1) can be rewritten as

$$u_j^{n+1} = u_j^n - \frac{v\Delta t}{2\Delta x}(u_{j+1}^n - u_{j-1}^n) + \frac{1}{2}(u_{j+1}^n - 2u_j^n + u_{j-1}^n) + \mathcal{O}(\Delta t^2, \Delta x^2) \; , \tag{6.2}$$

to underline how the Lax-Friedrichs scheme effectively provides a first-order finite-difference representation of a non-conservative equation

$$\frac{\partial u}{\partial t} + v\frac{\partial u}{\partial x} = \epsilon_{\scriptscriptstyle \rm LF} \frac{\partial^2 u}{\partial x^2} \; , \tag{6.3}$$

that is an advection-diffusion equation in which a dissipative term $\epsilon_{\scriptscriptstyle \rm LF} = \Delta x^2/(2\Delta t)$ is present. Given a computational domain of length L, this scheme will therefore have a typical diffusion timescale $\tau \simeq L^2/\epsilon_{\scriptscriptstyle \rm LF}$. Clearly, the larger the diffusion coefficient, the faster will the solution be completely smeared over the computational domain.

In a similar way, it is not difficult to realize that the upwind scheme

$$u_j^{n+1} = u_j^n - \frac{v\Delta t}{\Delta x}\left(u_j^n - u_{j-1}^n\right) + \mathcal{O}(\Delta t^2, \Delta x)\,, \tag{6.4}$$

provides a first-order accurate approximation to equation (3.2), but a second-order approximation to equation

$$\frac{\partial u}{\partial t} + v\frac{\partial u}{\partial x} = \epsilon_{\rm UW}\frac{\partial^2 u}{\partial x^2}\,, \tag{6.5}$$

where

$$\epsilon_{\rm UW} = \frac{v\Delta x}{2}\left(1 - \frac{v\Delta t}{\Delta x}\right) = \frac{v\Delta x}{2}\left(1 - c_{\rm CFL}\right)\,. \tag{6.6}$$

Stated differently, also the upwind method reproduces at higher-order an advection-diffusion equation with a dissipative term which is responsible for the gradual dissipation of the advected quantity u. This is shown in Fig. 6.2 for a wave packet (i.e. a periodic function embedded in a Gaussian) propagating to the right and where it is important to notice how the different peaks in the packet are advected at the correct speed, although their amplitude is considerably diminished.

In Courant-limited implementations, $c_{\rm CFL} = |v|\Delta t/\Delta x$ so that the ratio of the dissipation coefficients can be written as

$$\frac{\epsilon_{\rm LF}}{\epsilon_{\rm UW}} = \frac{1}{c_{\rm CFL}(1 - c_{\rm CFL})} \geq 4\,, \qquad \text{for} \quad c_{\rm CFL} \in [0, 1]\,. \tag{6.7}$$

In other words, while the upwind and the Lax-Friedrichs methods are both dissipative, the latter is generically more dissipative despite being more accurate in space. This can be easily appreciated by comparing Figs. 3.3 and 3.7 but also provides an important rule: *a more accurate numerical scheme is not necessarily a preferable one.*

A bit of patience and a few lines of algebra would also show that the Lax-Wendroff scheme for the advection equation (3.2)

$$u_j^{n+1} = u_j^n - \frac{v\Delta t}{2\Delta x}\left(u_{j+1}^n - u_{j-1}^n\right) + \frac{v^2\Delta t^2}{2\Delta x^2}\left(u_{j+1}^n - 2u_j^n + u_{j-1}^n\right) + \mathcal{O}(\Delta t^3, \Delta x^2)\,. \tag{6.8}$$

provides a first-order accurate approximation to equation (3.2), a second-order approximation to an advection-diffusion equation with dissipation coefficient $\epsilon_{\rm LW}$, and a third-order approximation to equation

$$\frac{\partial u}{\partial t} + v\frac{\partial u}{\partial x} = \epsilon_{\rm LW}\frac{\partial^2 u}{\partial x^2} + \beta_{\rm LW}\frac{\partial^3 u}{\partial x^3}\,, \tag{6.9}$$

where

$$\beta_{\rm LW} = -\frac{v\Delta x^2}{6}\left[1 - \left(\frac{v\Delta t}{\Delta x}\right)^2\right]\,. \tag{6.10}$$

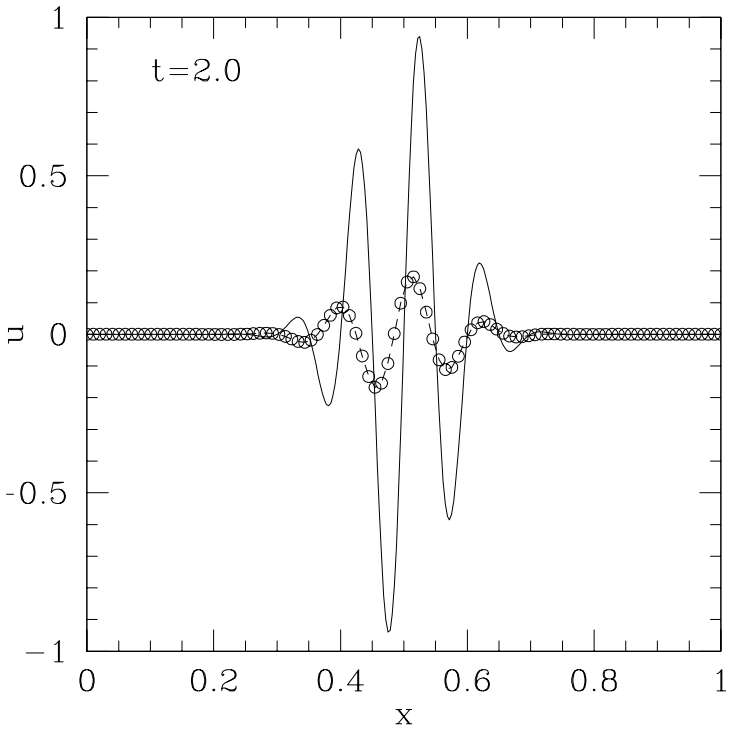

Figure 6.1: Time evolution of a wave-packet initially centred at $x = 0.5$ computed using a Lax-Friedrichs scheme with $C_{\rm CFL} = 0.75$. The analytic solution at time $t = 2$ is shown with a solid line the dashed lines are used to represent the numerical solution at the same time. Note how dissipation reduces the amplitude of the wave-packet but does not change sensibly the propagation of the wave-packet.

As mentioned in Section 3, the Lax-Wendroff scheme retains some of the dissipative nature of the originating Lax-Friedrichs scheme and this is incorporated in the dissipative term proportional to $\epsilon_{\rm LW}$. Using expression (6.8), it is easy to deduce the magnitude of this dissipation and compare it with the equivalent one produced with the Lax-Friedrichs scheme. A couple of lines of algebra show that

$$\epsilon_{\rm LW} = \frac{c_{\rm CFL} v \Delta x}{2} = c_{\rm CFL}^2 \epsilon_{\rm LF} < \epsilon_{\rm LF} \ , \tag{6.11}$$

thus emphasizing that the Lax-Wendroff scheme is considerably less dissipative.

The simplest way of quantifying the effects introduced by the right-hand-sides of equations (6.3), (6.5), and (6.9) is by using a single Fourier mode with angular frequency ω and wavenumber k, propagating in the positive x-direction, i.e.

$$u(t, x) = e^{i(kx - \omega t)} \ . \tag{6.12}$$

It is then easy to verify that

$$\frac{\partial u}{\partial t} = -i\omega u \,, \qquad \frac{\partial u}{\partial x} = iku \,, \qquad \frac{\partial^2 u}{\partial x^2} = -k^2 u \,, \qquad \frac{\partial^3 u}{\partial x^3} = -ik^3 u \,. \tag{6.13}$$

In the case in which the finite difference scheme provides an accurate approximation to a purely advection equation, the relations (6.13) lead to the obvious dispersion relation $\omega = vk$, so that the *numerical* mode $u_{_N}(t,x)$ will have a solution

$$u_{_N}(t,x) = e^{ik(x-vt)} \,, \tag{6.14}$$

representing a mode propagating with *phase velocity* $c_{\rm p} \equiv \omega/k = v$, which coincides with the *group velocity* $c_{\rm g} \equiv \partial\omega/\partial k = v$.

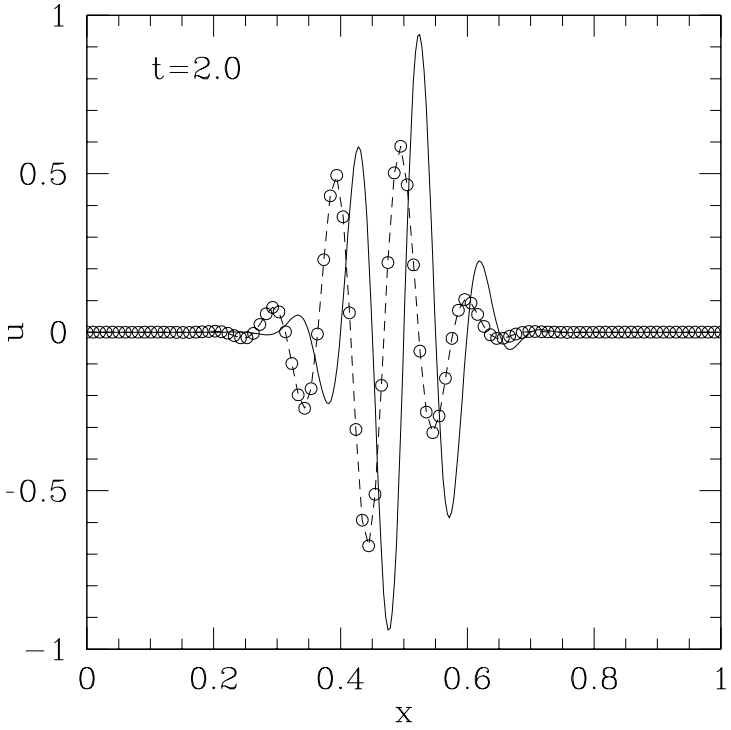

Figure 6.2: Time evolution of a wave-packet initially centred at $x = 0.5$ computed using a Lax-Wendroff scheme with $C_{\rm CFL} = 0.75$. The analytic solution at time $t = 2$ is shown with a solid line the dashed lines are used to represent the numerical solution at the same time. Note how the amplitude of the wave-packet is not drastically reduced but the group velocity suffers from a considerable error.

However, it is simple to verify that the advection-diffusion equation approximated by the Lax-Friedrichs scheme (6.3), will have a corresponding solution

$$u_{_N}(t,x) = e^{-\epsilon_{\rm LF} k^2 t} e^{ik(x-vt)} \ , \tag{6.15}$$

thus having, besides the advective term, also an exponentially decaying mode. Similarly, a few lines of algebra are sufficient to realize that the dissipative term does not couple with the advective one and, as a result, the phase and group velocities remain the same and $c_{\rm p} = c_{\rm g} = v$. This is clearly shown in Fig. 6.1 which shows how the wave packet is sensibly dissipated but, overall, maintains the correct group velocity.

Finally, it is possible to verify that the advection-diffusion equation approximated by the Lax-Wendroff scheme (6.9), will have a solution given by

$$u_{_N}(t,x) = e^{-\epsilon_{\rm LW} k^2 t} e^{ik\left[x-\left(v+\beta_{\rm LW} k^2\right)t\right]} \ , \tag{6.16}$$

where, together with the advective and (smaller) exponentially decaying modes already encountered before, there appears also a dispersive term $\sim x-(v+\beta_{\rm LW} k^2)t$ producing different propagation speeds for modes with different wavenumbers. This becomes apparent after calculating the phase and group velocities which are given by

$$c_{\rm p} = v + \beta_{\rm LW} k^2 \ , \qquad \text{and} \qquad c_{\rm g} = v + 3\beta_{\rm LW} k^2 \ , \tag{6.17}$$

and provides a simple interpretation of the results shown in Fig. 6.2.

6.2 Measuring Dissipation and Stability

From what discussed so far it appears clear that one is often in the need of tools that allow a rapid comparison among different evolution schemes. One might be interested, for instance, in estimating which of two methods is less dissipative or whether an evolution scheme which is apparently stable will eventually turn out to be unstable. A very useful tool that can be used in this context is the calculation of the *"norms"* of the quantity we are interested in. These are quantities that are computationally inexpensive and can be calculated at each time-level n as

$$\mathrm{L1-norm} :: \qquad ||u||(t^n) = \sum_{j=1}^{J} |u_j^n| \ , \tag{6.18}$$

$$\mathrm{L2-norm} :: \qquad ||u||^2(t^n) = \sum_{j=1}^{J} (u_j^n)^2 \ , \tag{6.19}$$

$$\mathrm{L-infinity-norm} :: \qquad ||u||_\infty(t^n) = \max_{j=1,\ldots J}(|u_j^n|) \ . \tag{6.20}$$

In the case of a scalar wave equation, the 2-norm has a physical interpretation and could be associated to the amount of energy contained in the numerical domain; its conservation is therefore a clear signature of a non-dissipative numerical scheme.

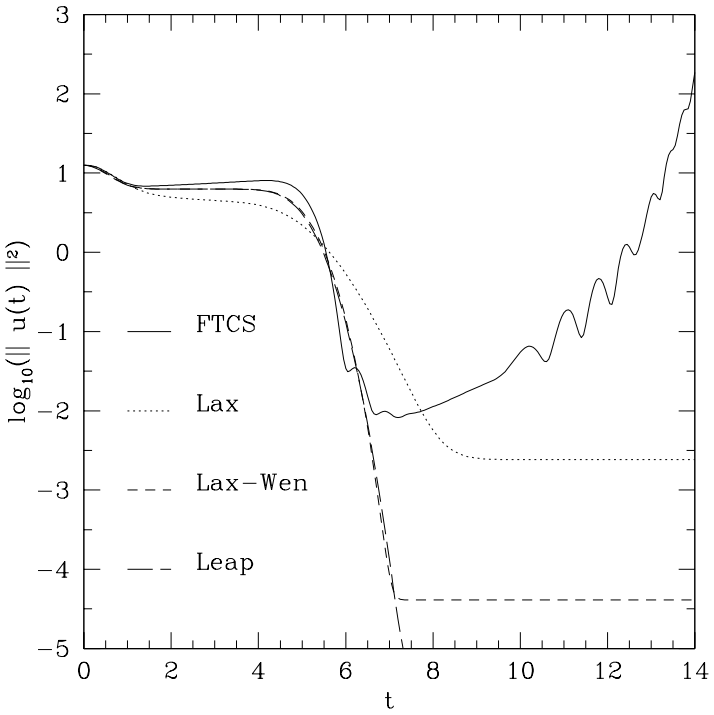

Figure 6.3: Time evolution of the logarithm of the L2 norms for the different numerical schemes discussed so far. Sommerfeld outgoing boundary conditions were used in this example.

Fig. 6.2 compares the $\mathbf{L}^2$ norms for the different numerical schemes discussed so far and in the case in which Sommerfeld outgoing boundary conditions were used. Note how the FTCS scheme is unstable and that the errors are already comparable with the solution well before a crossing time. Similarly, it is evident that the use of Sommerfeld boundary conditions allows a smooth evacuation of the energy in the wave from the numerical grid after $t \sim 6$. Finally, Fig. 11 clearly shows the dissipative features of a Lax-Friedrichs scheme when compared to a Leapfrog or Lax-Wendroff schemes.

6.3 Consistency, Convergence and Stability

To be written...

Chapter 7

The wave equation in two spatial dimensions (2D)

We will now extend the procedures studied so far to the case of a wave equation in two dimensions

$$\frac{\partial^2 u}{\partial t^2} = v^2 \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) . \tag{7.1}$$

As for the one-dimensional case, also in this case the wave equation can be reduced to the solution of a set of three first-order advection equations

$$\frac{\partial r}{\partial t} = v \frac{\partial s}{\partial x} , \tag{7.2}$$

$$\frac{\partial l}{\partial t} = v \frac{\partial s}{\partial y} , \tag{7.3}$$

$$\frac{\partial s}{\partial t} = v \left(\frac{\partial r}{\partial x} + \frac{\partial l}{\partial y} \right) , \tag{7.4}$$

once the following definitions have been made

$$r = v \frac{\partial u}{\partial x} , \tag{7.5}$$

$$l = v \frac{\partial u}{\partial y} , \tag{7.6}$$

$$s = \frac{\partial u}{\partial t} . \tag{7.7}$$

In vector notation the system can again be written as

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \mathbf{F}(\mathbf{U}) = 0 \,, \tag{7.8}$$

where

$$\mathbf{U} = \begin{pmatrix} r \\ l \\ s \end{pmatrix} \,, \qquad \text{and} \qquad \mathbf{F}(\mathbf{U}) = \begin{pmatrix} -v & 0 & 0 \\ 0 & -v & 0 \\ 0 & 0 & -v \end{pmatrix} \mathbf{U} = -v \begin{pmatrix} r \\ l \\ s \end{pmatrix} \,, \tag{7.9}$$

provided we define

$$\nabla \equiv \begin{pmatrix} 0 & 0 & \dfrac{\partial}{\partial x} \\ 0 & 0 & \dfrac{\partial}{\partial y} \\ \dfrac{\partial}{\partial x} & \dfrac{\partial}{\partial y} & 0 \end{pmatrix} \,. \tag{7.10}$$

The finite-difference notation should also be extended to account for the two spatial dimension and we will then assume that $u^n_{i,j} \equiv u(x_i, y_j, t^n)$.

7.1 The Lax-Friedrichs Scheme

We can look at the system of equations (7.2) and (7.3) as a set of two equations to be integrated with the procedures so far developed in one-dimension. Furthermore, we need to solve for eq. (7.4) which can be written as

$$\frac{\partial s}{\partial t} = -\frac{\partial F_{\rm x}}{\partial x} - \frac{\partial F_{\rm y}}{\partial y} \tag{7.11}$$

once we identify $F_{\rm x}$ with $-vr$ and $F_{\rm y}$ with $-vl$.

The Lax-Friedrichs scheme for this equation is just the generalization of the 1D expressions discussed so far and yields

$$\begin{aligned} s^{n+1}_{i,j} &= \frac{1}{4}\left[s^n_{i+1,j} + s^n_{i-1,j} + s^n_{i,j+1} + s^n_{i,j-1}\right] - \frac{\Delta t}{2\Delta x}\left[(F^n_{\rm x})_{i+1,j} - (F^n_{\rm x})_{i-1,j}\right] + \\ &\qquad - \frac{\Delta t}{2\Delta y}\left[(F^n_{\rm y})_{i,j+1} - (F^n_{\rm y})_{i,j-1}\right] \,, \\ &= \frac{1}{4}\left[s^n_{i+1,j} + s^n_{i-1,j} + s^n_{i,j+1} + s^n_{i,j-1}\right] - \frac{\Delta t}{2}\left[\frac{r^n_{i+1,j} - r^n_{i-1,j}}{\Delta x} + \frac{l^n_{i,j+1} - l^n_{i,j-1}}{\Delta y}\right] \,, \end{aligned} \tag{7.12}$$

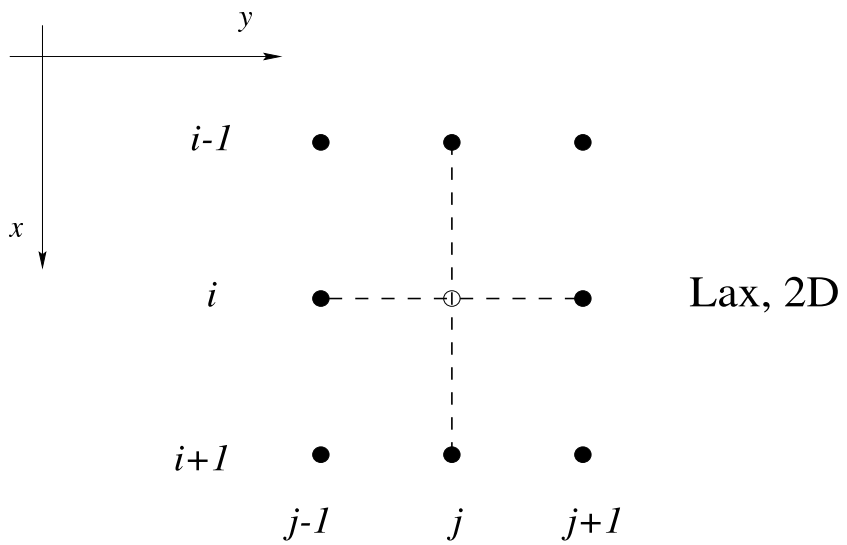

Figure 7.1: Schematic diagram of a Lax-Friedrichs evolution scheme in two dimensions. Note that the center of the cross-like stencil is not used in this case.

with the corresponding stencil being shown in Fig. 7.1 and where it should be noted that the center of the cross-like stencil is not used. A von Neumann stability analysis can be performed also in 2D and it yields

$$\xi = \frac{1}{2}[\cos(k_{\rm x}\Delta x) + \cos(k_{\rm y}\Delta y)] - i[\alpha_{\rm x}\sin(k_{\rm x}\Delta x) + \alpha_{\rm y}\sin(k_{\rm y}\Delta y)]\,, \tag{7.13}$$

where

$$\alpha_{\rm x} \equiv \frac{v_{\rm x}\Delta t}{\Delta x}\,, \qquad\qquad \alpha_{\rm y} \equiv \frac{v_{\rm y}\Delta t}{\Delta x}\,. \tag{7.14}$$

Stability is therefore obtained if

$$\frac{1}{2} - (\alpha_{\rm y}^2 + \alpha_{\rm y}^2) \geq 0\,, \tag{7.15}$$

or, equally, if

$$\Delta t \leq \frac{\Delta x}{\sqrt{2(v_{\rm x}^2 + v_{\rm y}^2)}}\,, \tag{7.16}$$

Expression (7.16) represents the 2D extension of the CFL stability condition. In general, for a N dimensional space, the CFL stability condition can be expressed as

$$\Delta t \leq \min\left(\frac{\Delta x_i}{\sqrt{N}|v|}\right)\,, \tag{7.17}$$

where $i = 1, ...N$ and $|v| \equiv (\sum_{i=1}^{N} v_i^2)^{1/2}$. Note, in 2D, the appearence of an averaging coefficient $1/4$ multiplying the value of the function at the time-level n.

7.2 The Lax-Wendroff Scheme

The 2D generalization of the one-dimensional scheme (3.43) is also straightforward and can be described as follows

1. Compute r and l at the half-time using a half-step Lax-Friedrichs scheme

$$\begin{aligned} r_{i,j}^{n+\frac{1}{2}} &= \frac{1}{4}\left(r_{i+1,j}^{n} + r_{i,j+1}^{n} + r_{i-1,j}^{n} + r_{i,j-1}^{n}\right) + \frac{\alpha}{2}\left(s_{i+1,j}^{n} - s_{i-1,j}^{n}\right) , \qquad (7.18) \\ l_{i,j}^{n+\frac{1}{2}} &= \frac{1}{4}\left(l_{i+1,j}^{n} + l_{i,j+1}^{n} + l_{i-1,j}^{n} + l_{i,j-1}^{n}\right) + \frac{\alpha}{2}\left(s_{i,j+1}^{n} - s_{i,j-1}^{n}\right) , \qquad (7.19) \end{aligned}$$

   where $\alpha \equiv v\Delta t/\Delta x$.

2. Evolve s to the time-level $n+1$ using a half-step Leapfrog scheme

$$s_{i,j}^{n+1} = s_{i,j}^{n} + \frac{\alpha}{2}\left(r_{i+1,j}^{n+\frac{1}{2}} - r_{i-1,j}^{n+\frac{1}{2}}\right) + \frac{\alpha}{2}\left(l_{i,j+1}^{n+\frac{1}{2}} - r_{i,j-1}^{n+\frac{1}{2}}\right) . \qquad (7.20)$$

3. Update u to the time-level $n+1$, i.e.

$$u_{i,j}^{n+1} = u_{i,j}^{n} + \frac{\Delta t}{2}\left(s_{i,j}^{n+1} + s_{i,j}^{n}\right) . \qquad (7.21)$$

4. Evolve r and l to the time-level $n+1$, i.e.

$$\begin{aligned} r_{i,j}^{n+1} &= \frac{1}{4}\left(r_{i+1,j}^{n} + r_{i,j+1}^{n} + r_{i-1,j}^{n} + r_{i,j-1}^{n}\right) + \\ &\qquad \frac{\alpha}{2}\left[\frac{1}{2}\left(s_{i+1,j}^{n} + s_{i+1,j}^{n+1}\right) - \frac{1}{2}\left(s_{i-1,j}^{n} + s_{i-1,j}^{n+1}\right)\right] , \qquad (7.22) \\ l_{i,j}^{n+1} &= \frac{1}{4}\left(l_{i+1,j}^{n} + l_{i,j+1} + l_{i-1,j}^{n} + l_{i,j-1}^{n}\right) + \\ &\qquad \frac{\alpha}{2}\left[\frac{1}{2}\left(s_{i,j+1}^{n} + s_{i,j+1}^{n+1}\right) - \frac{1}{2}\left(s_{i,j-1}^{n} + s_{i,j-1}^{n+1}\right)\right] . \qquad (7.23) \end{aligned}$$

7.3 The Leapfrog Scheme

The 2D generalization of the one-dimensional scheme (4.16) is less straightforward, but not particularly difficult. As in one dimension, we can start by rewriting directly the finite-difference form of the wave equation as

$$\frac{u_{i,j}^{n+1} - 2u_{i,j}^{n} + u_{i,j}^{n-1}}{\Delta t^2} = v^2 \left(\frac{u_{i+1,j}^{n} - 2u_{i,j}^{n} + u_{i-1,j}^{n}}{\Delta x^2} \right) + v^2 \left(\frac{u_{i,j+1}^{n} - 2u_{i,j}^{n} + u_{i,j-1}^{n}}{\Delta y^2} \right)$$

so that, after some algebra, we obtain the explicit form

$$u_{i,j}^{n+1} = \alpha^2 \left[u_{i+1,j}^{n} + u_{i-1,j}^{n} + u_{i,j+1}^{n} + u_{i,j-1}^{n} \right] + 2u_{i,j}^{n}(1 - 2\alpha^2) - u_{i,j}^{n-1} \, . \tag{7.24}$$

The stencil relative to the algorithm (7.24) is illustrated in Fig. 7.2.

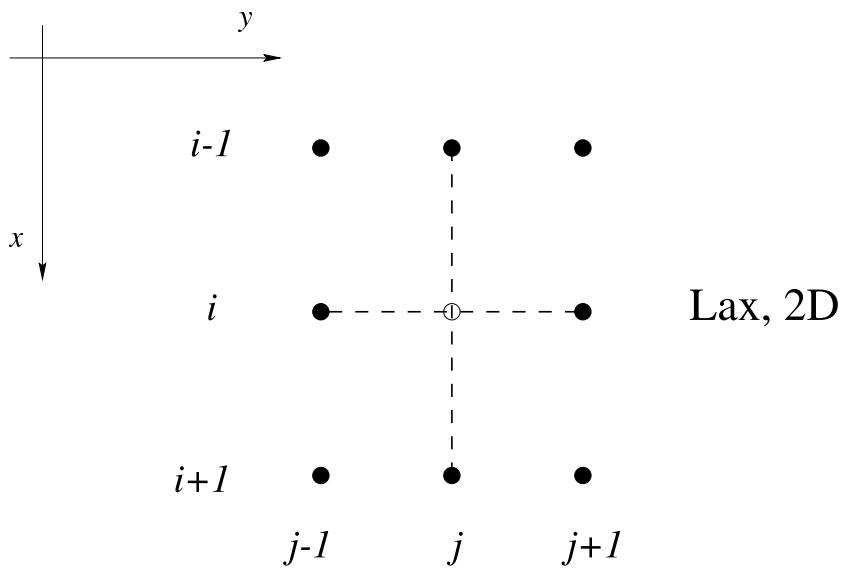

Figure 7.2: Schematic diagram of a Leapfrog evolution scheme in two dimensions. Note that the center of the cross-like stencil is used in this case both at the time-level n (fi lled circle) and at the time level $n + 1$ (fi lled square).

Figs. 7.3 and 7.4 show the solution of the wave equation in 2D using the scheme (7.24) and imposing Sommerfeld outogoing-wave boundary conditions at the edges of the numerical grid.

Radically different appears the evolution when reflective boundary conditions are imposed, as it is illustrated in Figs 4. Note that the initial evolution (i.e. for which the effects of the boundaries are negligible) is extremely similar to the one shown in Figs. 4, but becomes radically different when the wavefront has reached the outer boundary. As a result of the high (but not perfect!) reflectivity of the outer boundaries, the wave is "trapped" inside the numerical grid and bounces back and forth producing the characteristic interference patterns.

A synthetic representation of this different behaviour is presented with Fig. 16 which shows the time evolution of $\mathbf{L}^2$ norm for the two cases.

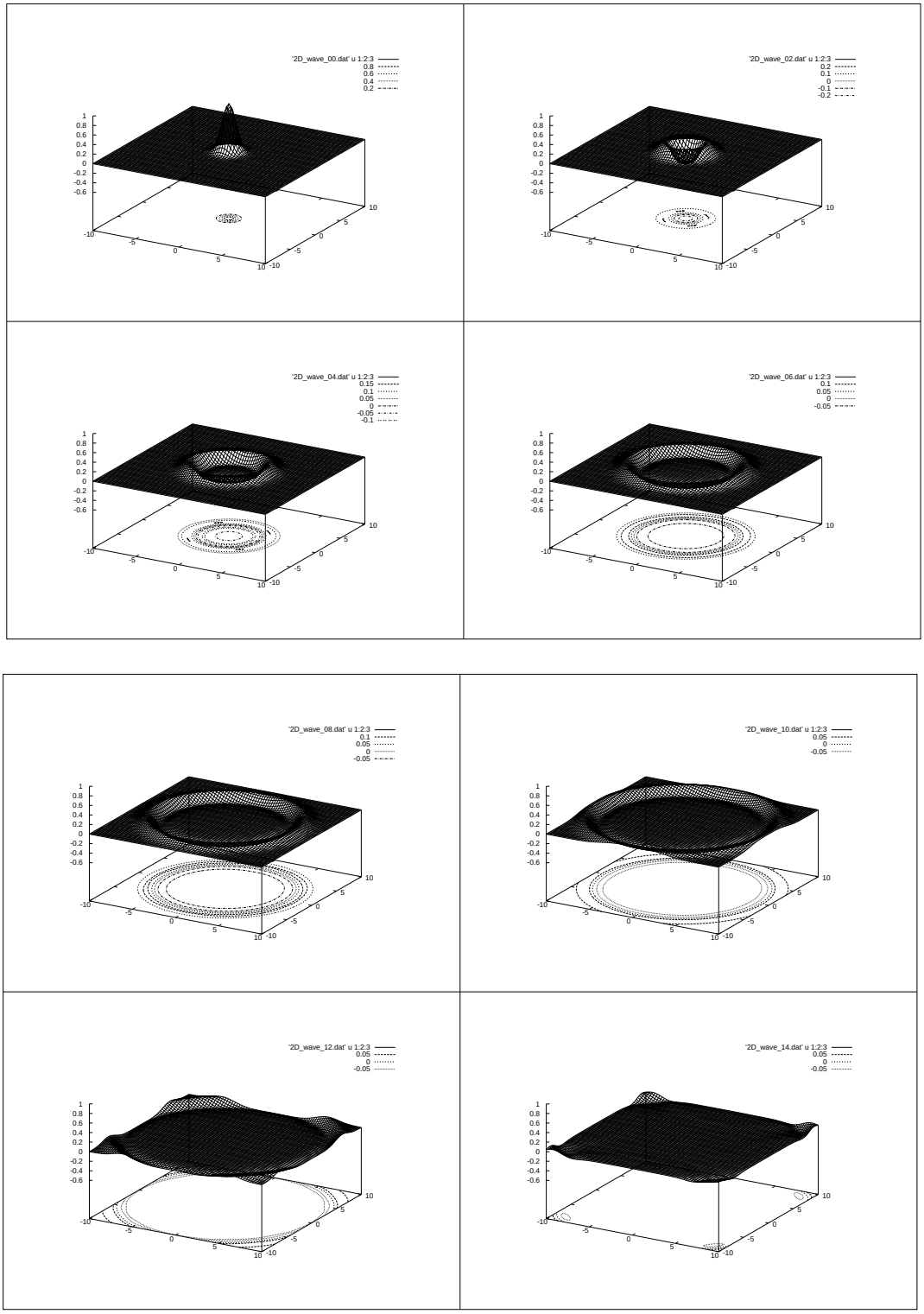

Figure 7.3: Plot of the time evolution of the wave equation when the Leapfrog scheme in 2D is used and Sommerfeld boundary conditions are imposed. Snapshots at increasing times are illustrated in a clockwise sequence.

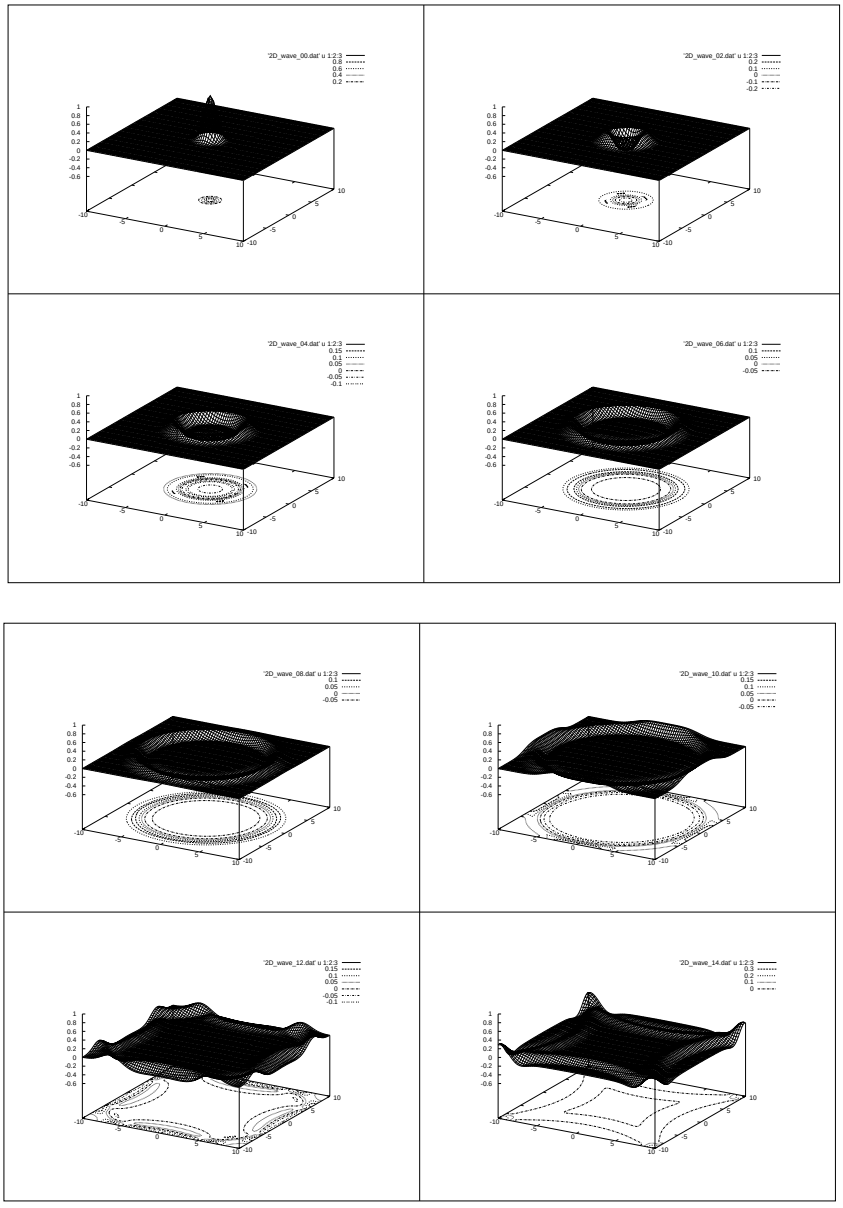

Figure 7.4: Plot of the time evolution of the wave equation when the Leapfrog scheme in 2D is used and Reflecting boundary conditions are applied. Snapshots at increasing times are illustrated in a clockwise sequence.

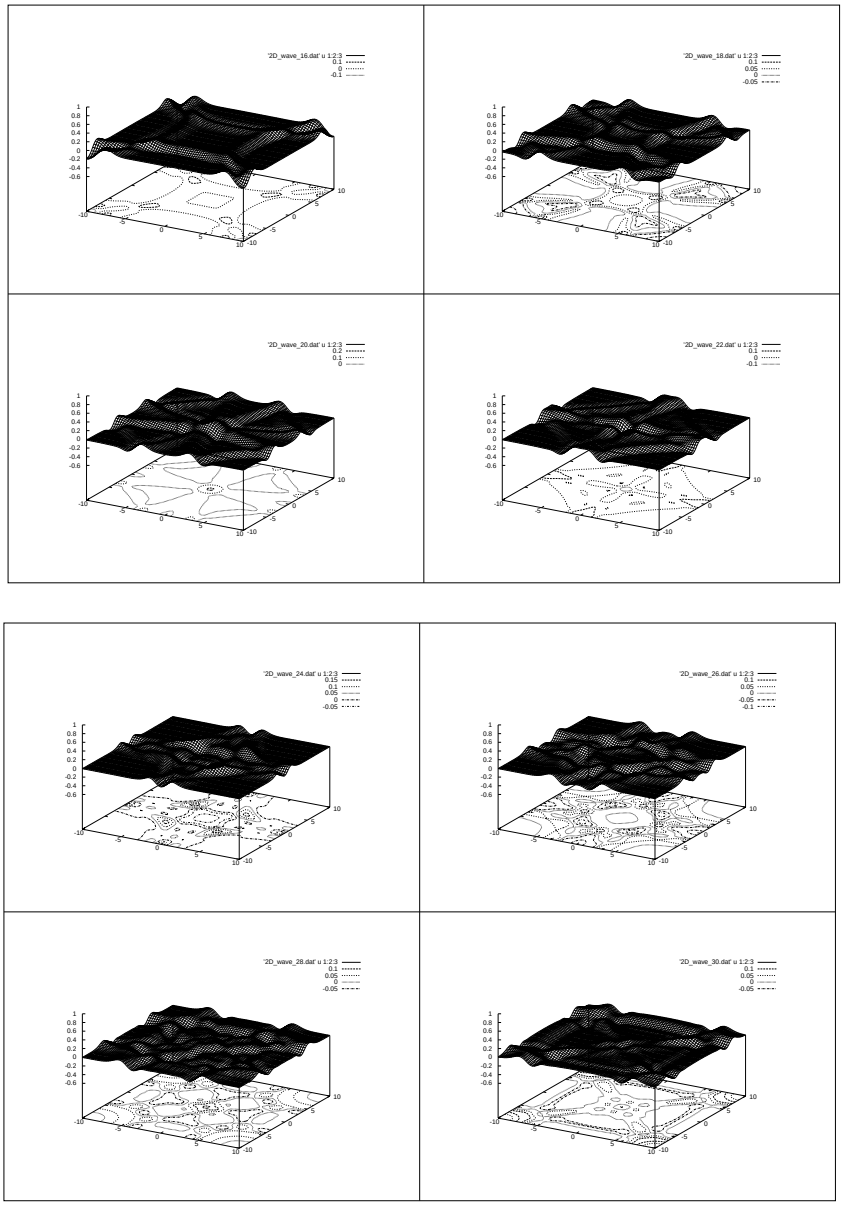

Figure 7.5: Plot of the time evolution of the wave equation when the Leapfrog scheme in 2D is used and Reflecting boundary conditions are applied.

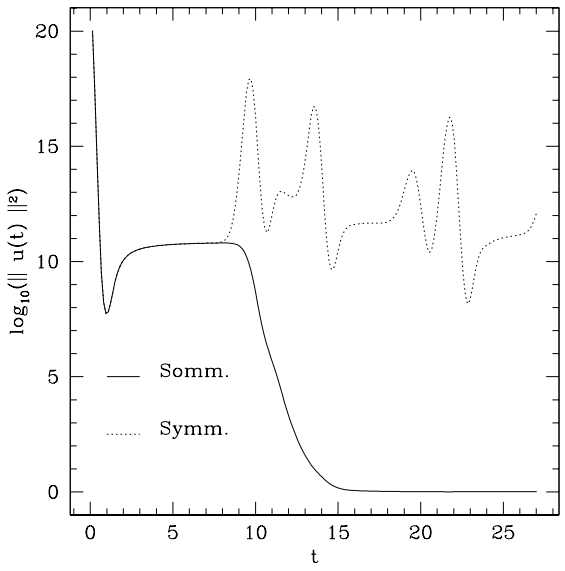

Figure 7.6: Plot of the time evolution of the $\mathbf{L}^2$ norm when the Leapfrog scheme in 2D is used. Note the radically different behaviour between Sommerfeld and reflecting boundary conditions.

Bibliography

[1] LEVEQUE, R. J. 2002, *Finite Volume Methods for Hyperbolic Problems*, Cambridge University Press, Cambridge, UK.

[2] POTTER, D 1973, *Computational Physics*, Wiley, New York, USA

[3] PRESS, W. H. ET AL., D 1992, *Numerical Recipes*, Cambridge University Press, Cambridge, UK.

[4] TORO, E. F. 1997, *Riemann Solvers and Numerical Methods for Fluid Dynamics*, Springer.

[5] VESELY, F. J. 1994, *Computational Physics: An Introduction*, Plenum, New York, USA