Numerical Methods for the Solution of Partial Differential Equations

Lecture Notes for the COMPSTAR School on Computational Astrophysics, 8-13/02/10, Caen, France

Luciano Rezzolla

Albert Einstein Institute, Max-Planck-Institute for Gravitational Physics, Potsdam, Germany

Available also online at `www.aei.mpg.de/~rezzolla`

February 20, 2010

Contents

1 Introduction **3**
- 1.1 Discretization of differential operators and variables 4
- 1.2 Errors 5
  - 1.2.1 Machine-precision error 5
  - 1.2.2 Round-off error 6
  - 1.2.3 Truncation error 6

2 Hyperbolic PDEs: Flux Conservative Formulation **9**

3 The advection equation in one dimension (1D) **11**
- 3.1 The 1D Upwind scheme: $\mathcal{O}(\Delta t, \Delta x)$ 11
- 3.2 The 1D FTCS scheme: $\mathcal{O}(\Delta t, \Delta x^2)$ 16
- 3.3 The 1D Lax-Friedrichs scheme: $\mathcal{O}(\Delta t, \Delta x^2)$ 18
- 3.4 The 1D Leapfrog scheme: $\mathcal{O}(\Delta t^2, \Delta x^2)$ 21
- 3.5 The 1D Lax-Wendroff scheme: $\mathcal{O}(\Delta t^2, \Delta x^2)$ 23
- 3.6 The 1D ICN scheme: $\mathcal{O}(\Delta t^2, \Delta x^2)$ 25
  - 3.6.1 ICN as a θ-method 27
  - 3.6.2 Summary 31

4 Dissipation, Dispersion and Convergence **33**
- 4.1 On the Origin of Dissipation and Dispersion 33
- 4.2 Measuring Dissipation and Convergence 37
  - 4.2.1 The summarising power of norms 37
  - 4.2.2 Consistency and Convergence 38
  - 4.2.3 Convergence and Stability 41

5 The Wave Equation in 1D **43**
- 5.1 The FTCS Scheme 44
- 5.2 The Lax-Friedrichs Scheme 45
- 5.3 The Leapfrog Scheme 46
- 5.4 The Lax-Wendroff Scheme 48

6 Boundary Conditions **51**
6.1 Outgoing Wave BCs: the outer edge 51
6.2 Ingoing Wave BCs: the inner edge 53
6.3 Periodic Boundary Conditions 53

7 The wave equation in two spatial dimensions (2D) **55**
7.1 The Lax-Friedrichs Scheme 56
7.2 The Lax-Wendroff Scheme 57
7.3 The Leapfrog Scheme 58

8 Parabolic PDEs **65**
8.1 Diffusive problems 65
8.2 The diffusion equation in 1D 65
8.3 Explicit updating schemes 66
8.3.1 The FTCS method 66
8.3.2 The Du Fort-Frankel method and the θ-method 66
8.3.3 ICN as a $\boldsymbol{\theta}$-method 68
8.4 Implicit updating schemes 70
8.4.1 The BTCS method 70
8.4.2 The Crank-Nicolson method 71

A Semi-analytical solution of the model parabolic equation **73**
A.1 Homogeneous Dirichlet boundary conditions 73
A.2 Homogeneous Neumann boundary conditions 76

Acknowledgments

I am indebted to the several students who have helped me with the typing of the lectures notes into at TEXformat. They are too numerous to be reported here but my special thanks go to Olindo Zanotti for his help with the hyperbolic equations and to Gregor Leiler for his help with the parabolic equations and Chapter 4.

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

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

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

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 boundary* problems, or **IVBP**, 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 upper panel of Fig. 1.1). In practice, IVBP 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 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).

1.1 Discretization of differential operators and variables

Consider, for simplicity, a generic one-dimensional IVBP that could be written as

$$\mathcal{L}(u) - f = 0 \ , \tag{1.2}$$

where $u = u(x,t)$ and $\mathcal{L}$ is a differential operator in the two variables x and t acting on u. One of the most used methods for the solution of such a problem is by means of *finite differences*. It consists in two "discretization steps":

- *Variables discretization*: replace the function $u(x,t)$ with a discrete set of values $\{u_j^n\}$ that should approximate the pointwise values of u, *i.e.*, $u_j^n \approx u(x_j, t_n)$;
- *Operator discretization*: replace the continuous differential operator $\mathcal{L}$ with a discretized one, $\mathcal{L}_\Delta$, that when applied to the set $\{u_j^n\}$, gives an approximation to $\mathcal{L}(u)$ in terms of differences between the various u_j^n.

The set of values $\tilde{u} \equiv \{u_j^n,\ j = 1, \ldots, J,\ n = 1, \ldots, N\}$ (J and N are the number of points considered for the space and time variable respectively) is called the *grid function* and will be denoted by $\tilde{u}$. After this discretization process, the problem (1.2) is replaced by

$$\mathcal{L}_\Delta(\tilde{u}) - \tilde{f} = 0 + \epsilon_{\scriptscriptstyle T} \ , \tag{1.3}$$

that is, a discrete representation of *both* the differential operator $\mathcal{L}$ and of the variable u. The above equation is the *discrete representation* of the problem (1.2). Note that the righ-hand-side of (1.3) is not exactly zero and it differs from it by the *truncation error* $\epsilon_{\scriptscriptstyle T}$, which will be introduced in Sect. 1.2.3

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 IVBPs) in spatial foliations such that the time coordinate t is constant on each slice. As shown in the lower panel of Fig. 1.1, each point $\mathcal{P}(x_j, t^n)$ in this discretized spacetime will have spatial and time coordinate defined as

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

$$t^n = t^0 + n\Delta t \ , \qquad\qquad n = 0, \pm 1, \ldots, \pm N \ , \tag{1.4}$$

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 (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.

1.2 Errors

Errors are a natural and inevitable heritage of numerical analysis and their presence is not a nuisance as long their origing is well determined and under control. Three main errors will be discussed repeatedly in these notes and we briefly discuss them below.

1.2.1 Machine-precision error

The *machine-precision error* reflects the precision of the machine used and can be expressed in terms of the equality

$$\mathrm{fp}\,(1.0) = \mathrm{fp}\,(1.0) + \epsilon_{\mathrm{M}} \ , \tag{1.5}$$

where $\mathrm{fp}\,(1.0)$ is the floating-point description of the number 1. Stated differently, the machine-precision error reflects the ability of the machine to distinguish two floating point numbers and is therefore related to the number of significant figures used in the mantissa.

1.2.2 Round-off error

The *round-off error* is the accumulation of machine-precision errors as a result of N floating point operations. Because of the random nature in which machine-precision errors add-up, this error can be estimated to be

$$\epsilon_{\mathrm{RO}} \approx \sqrt{N}\epsilon_{\mathrm{M}} \ . \tag{1.6}$$

Clearly, when performing a numerical computation one should restrict the number of operations such that ϵ_{RO} is below the error at which the results needs to be determined.

1.2.3 Truncation error

The *truncation error* is fundamentally different from the previous two types of errors in that it is not dependent on the machine used but it reflects the human decision made in discretizing the continuum problem. Mathematically it can therefore be expressed as

$$\mathcal{L}(u) - f = \mathcal{L}_{\Delta}(\tilde{u}) - \tilde{f} + \epsilon_{\mathrm{T}} \ . \tag{1.7}$$

Since the truncation error is totally under the human judgment, its measure is essential to guarantee that the discretization operation has been made properly and that the discretized problem is therefore a faithful representation of the continuum one, modulo the truncation error.

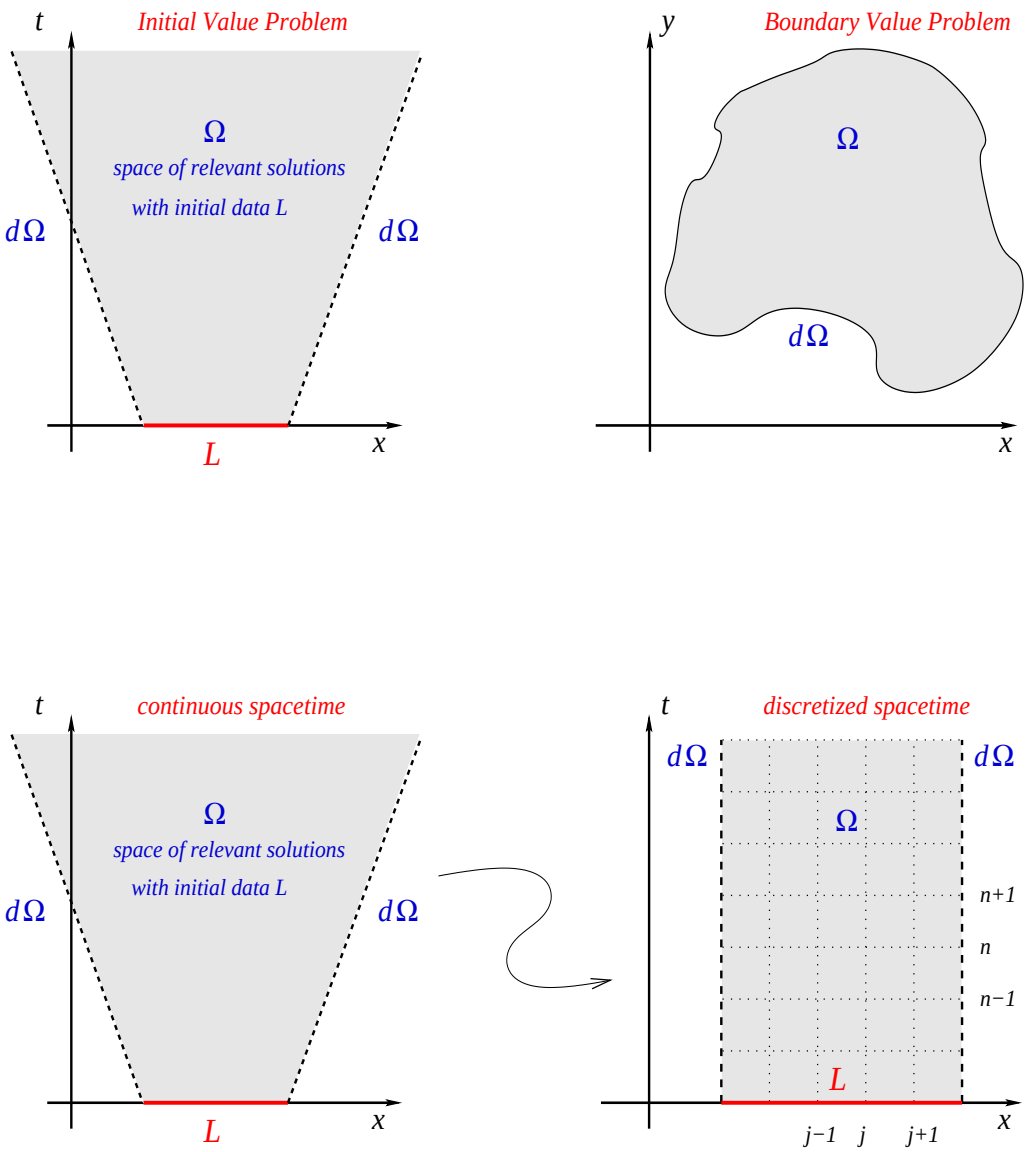

Figure 1.1: *Upper panel:* Schematic distinction between IVBPs and BVPs. *Lower Panel:* Schematic discretization of a hyperbolic IVBP

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

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

where $u(\boldsymbol{x}, t)$ is the *density* of the conserved quantity, $\boldsymbol{F}$ the density flux and $\boldsymbol{x}$ a vector of spatial coordinates. In most of the physically relevant cases, the flux density $\boldsymbol{F}$ will not depend explicitly on $\boldsymbol{x}$ and t, but only implicitly through the density $u(\boldsymbol{x}, t)$, *i.e.*, $\boldsymbol{F} = \boldsymbol{F}(u(\boldsymbol{x}, t))$. The vector $\boldsymbol{F}$ 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 $\boldsymbol{U}$ and write a set of conservation equations in the form

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

Here, $\boldsymbol{S}(\boldsymbol{U})$ is a generic "source term" indicating the sources and sinks of the vector $\boldsymbol{U}$. The main property of the homogeneous equation (2.2) (*i.e.*, when $\boldsymbol{S}(\boldsymbol{U}) = 0$) is that the knowledge of the state-vector $\boldsymbol{U}(x, t)$ 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

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

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

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 $\boldsymbol{S}(\boldsymbol{U}) = 0$ and will be used hereafter as our working examples.

Chapter 3

The advection equation in one 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, $\boldsymbol{F}(\boldsymbol{U})$ is diagonal and given by

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

where $\mathbf{I}$ 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 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

$$\frac{\partial^m u}{\partial x^m} = \mathcal{L}_\Delta(u) + \mathcal{O}(\Delta x^p) \; . \tag{3.6}$$

Of course, the time and spatial operators may have finite-difference representations 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.
Note 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

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

While formally similar, the approximation (3.7) 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.*,

$$\left.\frac{\partial u}{\partial x}\right|_j^n = \frac{u_{j+1}^n - u_j^n}{\Delta x} + \mathcal{O}(\Delta x) \; . \tag{3.8}$$

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

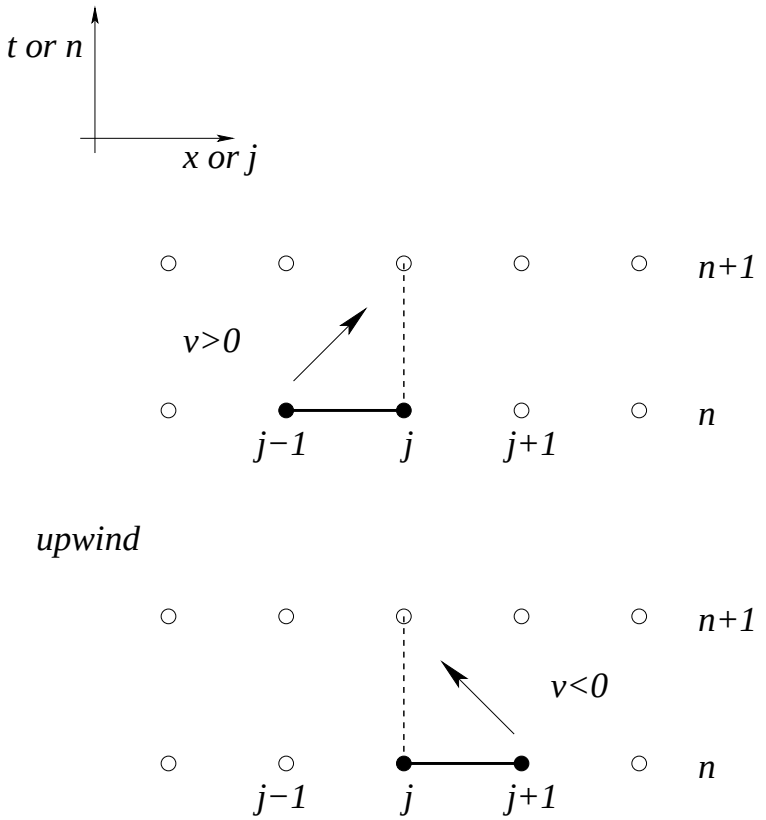

Figure 3.1: Schematic diagram of an UPWIND evolution scheme.

on the value of the advection velocity v, two different finite-difference representations can be given of equation (3.2) and these are

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

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

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

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

$$u_j^{n+1} = u_j^n - \frac{v\Delta t}{\Delta x}(u_{j+1}^n - u_j^n) + \mathcal{O}(\Delta t^2, \Delta x \Delta t) \,, \qquad \text{if } v < 0 \,. \tag{3.12}$$

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

$$\boldsymbol{U}_j^{n+1} = \boldsymbol{U}_j^n \pm \frac{\Delta t}{\Delta x}\left[\boldsymbol{F}_{j\mp 1}^n - \boldsymbol{F}_j^n\right] + \mathcal{O}(\Delta t^2, \Delta x \Delta t) \ , \tag{3.13}$$

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. 1.1 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 finite-difference representations 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 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

$$u_j^n = \xi^n e^{ikx_j} \ , \tag{3.14}$$

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

$$u_j^{n+1} = \mathcal{T}(\Delta t^p, \Delta x^q) u_j^n \ , \tag{3.15}$$

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 (3.14) and (3.15) 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.*,

$$|\xi|^2 = \xi\xi^* \leq 1 \ . \tag{3.16}$$

Using (3.14) in (3.11)–(3.12) we would obtain an amplification factor

$$\xi = 1 - |\alpha|\left(1 - \cos(k\Delta x)\right) - \mathrm{i}\alpha \sin(k\Delta x) \ , \tag{3.17}$$

where

$$\alpha \equiv \frac{v\Delta t}{\Delta x} \ . \tag{3.18}$$

Its quared modulus $|\xi|^2 \equiv \xi\xi^*$ is then

$$|\xi|^2 = 1 - 2\,|\alpha|\,(1 - |\alpha|)\,(1 - \cos(k\Delta x)) \ , \tag{3.19}$$

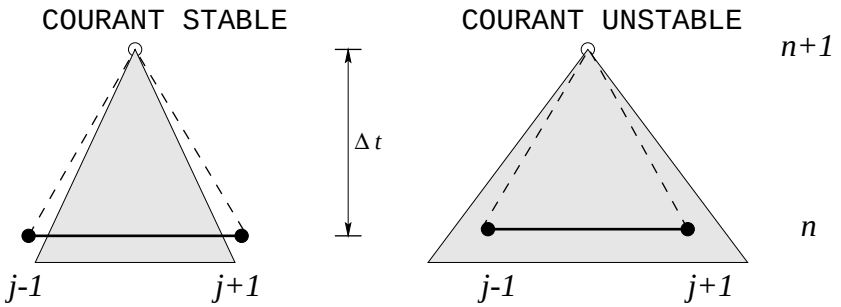

Figure 3.2: Schematic diagram of Courant stable and unstable choices of time-steps Δt. The two dashed lines limit the numerical domain of dependence of the solution at x_j^{n+1}, while the shaded area represents the physical domain of dependence. Stability is achieved when the first one is larger than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

so that the amplification factor (3.19) is less than one as long as the *Courant-Friedrichs-Löwy condition* (CFL condition)

$$|\alpha| \leq 1 \,, \tag{3.20}$$

is satisfied (condition (3.20) is sometimes referred to simply as the Courant condition.). Note that in practice, the CFL condition (3.20) is used to determine the time-step Δt once v is known and Δx has been chosen to achieve a certain accuracy, *i.e.*,

$$\Delta t = c_{\rm CFL} \frac{\Delta x}{|v|} \,, \tag{3.21}$$

with $c_{\rm CFL} < 1$ being the CFL factor. Expression (3.21) 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 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.*,

$$|v| = c_{\rm CFL} \frac{\Delta x}{\Delta t} \leq v_{\rm N} \equiv \frac{\Delta x}{\Delta t} \,. \tag{3.22}$$

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].

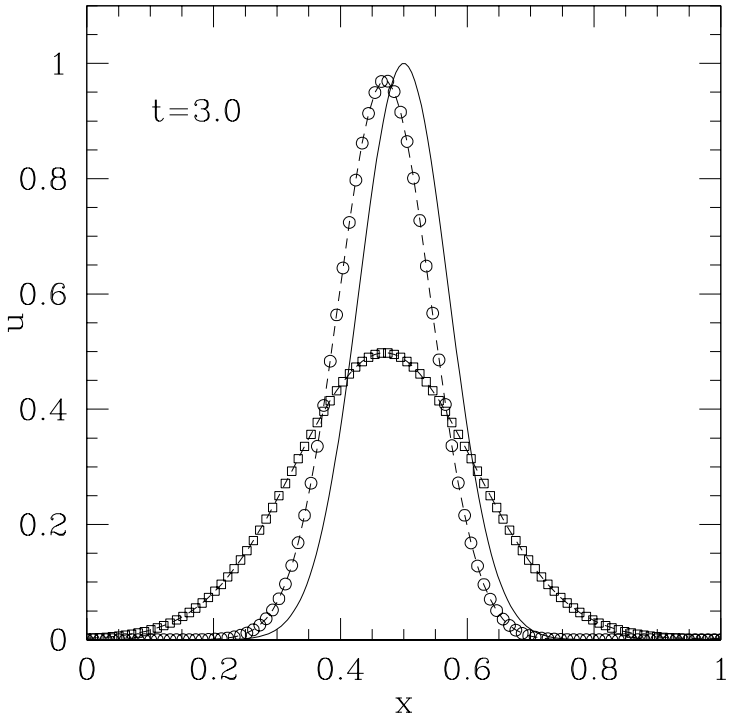

Figure 3.3: Time evolution of a Gaussian initially centred at $x = 0.5$ computed using an upwind scheme with $v = 10$ and 100 gridpoints. The analytic solution at time $t = 3$ is shown with a solid line the dashed lines are used to represent the numerical solution at the same time. Two different simulations 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.

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 $x_j \pm \Delta x$

$$u(x_j + \Delta x, t^n) = u(x_j, t^n) + \frac{\partial u}{\partial x}(x_j, t^n)\Delta x + \frac{1}{2}\frac{\partial^2 u}{\partial x^2}(x_j, t^n)\Delta x^2 + \mathcal{O}(\Delta x^3) \,, \tag{3.23}$$

$$u(x_j - \Delta x, t^n) = u(x_j, t^n) - \frac{\partial u}{\partial x}(x_j, t^n)\Delta x + \frac{1}{2}\frac{\partial^2 u}{\partial x^2}(x_j, t^n)\Delta x^2 + \mathcal{O}(\Delta x^3) \,, \tag{3.24}$$

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

$$\left.\frac{\partial u}{\partial x}\right|_j^n = \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} + \mathcal{O}(\Delta x^2) \,, \tag{3.25}$$

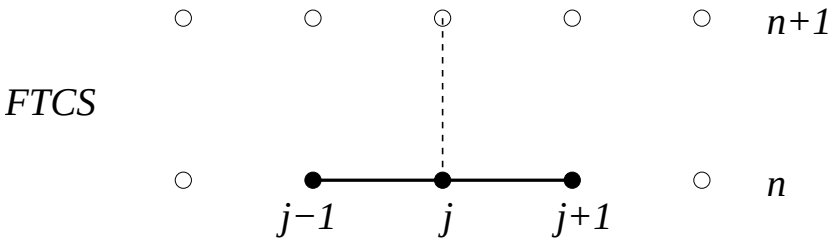

Figure 3.4: Schematic diagram of a FTCS evolution scheme.

Using now the second-order accurate operator (3.25) 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

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = -v \left(\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} \right) + \mathcal{O}(\Delta t, \Delta x^2) \,, \tag{3.26}$$

so that

$$u_j^{n+1} = u_j^n - \frac{\alpha}{2}(u_{j+1}^n - u_{j-1}^n) + \mathcal{O}(\Delta t^2, \Delta x^2 \Delta t) \,, \tag{3.27}$$

or more generically, for a system of linear hyperbolic equations

$$\boldsymbol{U}_j^{n+1} = \boldsymbol{U}_j^n - \frac{\Delta t}{2\Delta x} \left[\boldsymbol{F}_{j+1}^n - \boldsymbol{F}_{j-1}^n \right] + \mathcal{O}(\Delta t^2, \Delta x^2 \Delta t) \,, \tag{3.28}$$

The stencil for the finite- differencing (3.27) 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.

Applying the definition (3.14) to equation (3.26) and few algebraic steps lead to an amplification factor

$$\xi = 1 - \mathrm{i}\alpha \sin(k\Delta x) \,. \tag{3.29}$$

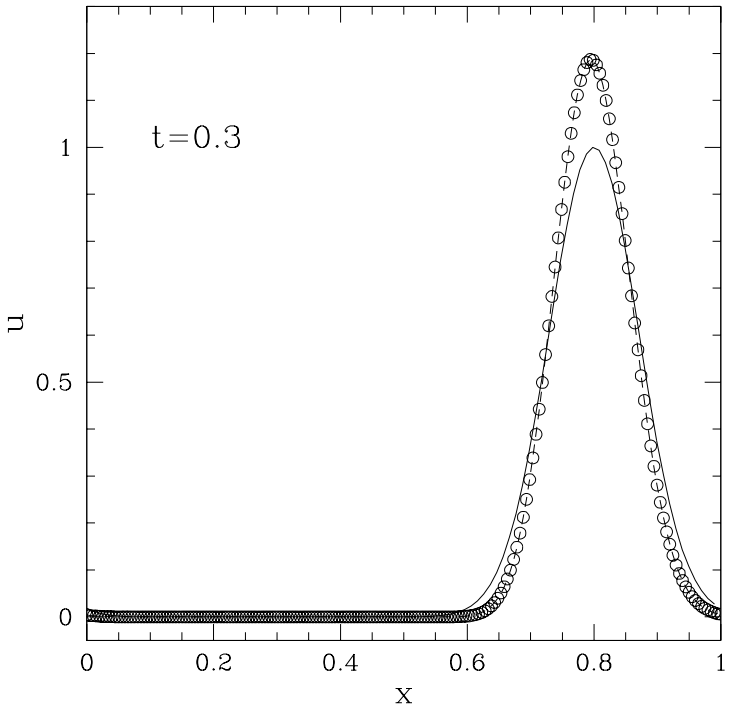

Figure 3.5: Time evolution of a Gaussian using an FTCS scheme with $v = 1$ and 100 gridpoints. The analytic solution at time $t = 0.3$ is shown with a solid line, while the dashed line is the numerical solution at the same time. Soon after $t \simeq 0.3$ the exponentially growing modes appear, rapidly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

whose squared modulus is

$$|\xi|^2 = 1 + (\alpha \sin(k\Delta x))^2 > 1 \,, \tag{3.30}$$

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: $\mathcal{O}(\Delta t, \Delta x^2)$

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

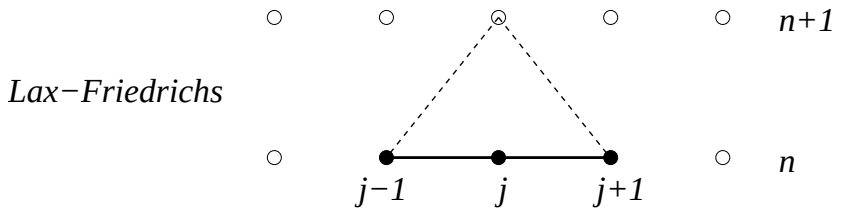

Figure 3.6: Schematic diagram of a Lax-Friedrichs evolution scheme.

formula (3.26), 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

$$u_j^{n+1} = \frac{1}{2}(u_{j+1}^n + u_{j-1}^n) - \frac{\alpha}{2}(u_{j+1}^n - u_{j-1}^n) + \mathcal{O}(\Delta x^2) \,, \tag{3.31}$$

and, for a system of linear hyperbolic equations

$$\boldsymbol{U}_j^{n+1} = \frac{1}{2}(\boldsymbol{U}_{j+1}^n + \boldsymbol{U}_{j-1}^n) - \frac{\Delta t}{2\Delta x}\left[\boldsymbol{F}_{j+1}^n - \boldsymbol{F}_{j-1}^n\right] + \mathcal{O}(\Delta x^2) \,. \tag{3.32}$$

Note that the truncation error in equations (3.31) and (3.32) is reported to be $\mathcal{O}(\Delta x^2)$ and not $\mathcal{O}(\Delta t^2, \Delta x^2 \Delta t)$ because we are assuming that the CFL condition is satisfied and hence $\Delta t = \mathcal{O}(\Delta x)$. We will maintain this assumption hereafter.

The schematic diagram of a Lax-Friedrichs evolution scheme is shown in Fig. 3.6. Perhaps surprisingly, the algorithm (3.32) is now *conditionally stable* as can be verified through a von Neumann stability analysis. Proceeding as done for the FTCS scheme and using (3.14) in (3.32) we would obtain an amplification factor whose modulus squared is

$$|\xi|^2 = 1 - \sin^2(k\Delta x)\left(1 - \alpha^2\right) \,, \tag{3.33}$$

which is < 1 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 (3.32) so that it clearly appears as a correction to (3.26):

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = -v\left(\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x}\right) + \frac{1}{2}\left(\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta t}\right) \,. \tag{3.34}$$

This is exactly the finite-difference representation of the equation

$$\frac{\partial u}{\partial t} + v\frac{\partial u}{\partial x} = \frac{1}{2}\left(\frac{\Delta x^2}{\Delta t}\right)\frac{\partial^2 u}{\partial x^2} \,, \tag{3.35}$$

where a diffusion term, $\propto \partial^2 u/\partial x^2$, has appeared on the right hand side. To prove this we sum the two Taylor expansions (3.23)–(3.24) around x_j to eliminate the first-order derivatives and obtain

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

where the sum has allowed us to cancel both the terms $\mathcal{O}(\Delta x)$ and $\mathcal{O}(\Delta x^3)$. Note that since the expression for the second derivative in (3.36) is $\mathcal{O}(\Delta x^2)$, it is appears multiplied by $\Delta x^2/\Delta t = \mathcal{O}(\Delta x)$ in equation (3.35), thus making the right-hand-side $\mathcal{O}(\Delta x^3)$ overall. The left-hand-side, on the other hand, is only $\mathcal{O}(\Delta x)$ (the time derivative is $\mathcal{O}(\Delta x)$, while the spatial derivative is $\mathcal{O}(\Delta x^2)$). As a result, the dissipative term goes to zero more rapidly than the intrinsic truncation error of the Lax-Friedrichs scheme, thus guaranteeing that the in the continuum limit the algorithm will converge to the correct solution of the advection equation.

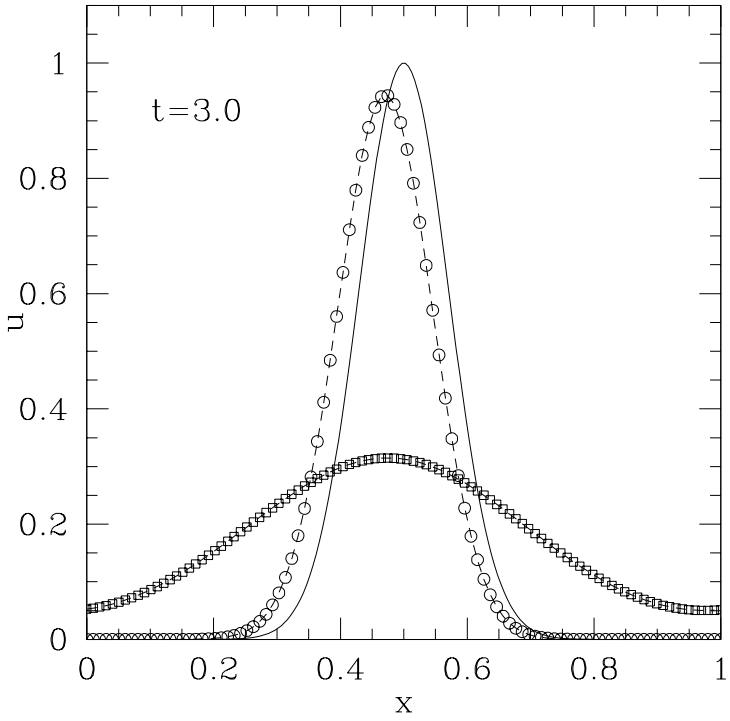

Figure 3.7: This is the same as in Fig. 3.3 but for a Lax-Friedrichs scheme. Note how the scheme is stable but also suffers from a considerable dissipation.

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, in which a spurious numerical dissipation has been introduced [*i.e.*, eq. (3.35)]. 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. As mentioned above, the dissipative term is always smaller than the truncation error thus guaranteeing the convergence to the correct solution. Furthermore, it is useful to bear in mind that 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 (3.33) 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. This is called *"consistency"* of the discretized operator and will be discussed in detail in Sect. 4.2.2.

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.

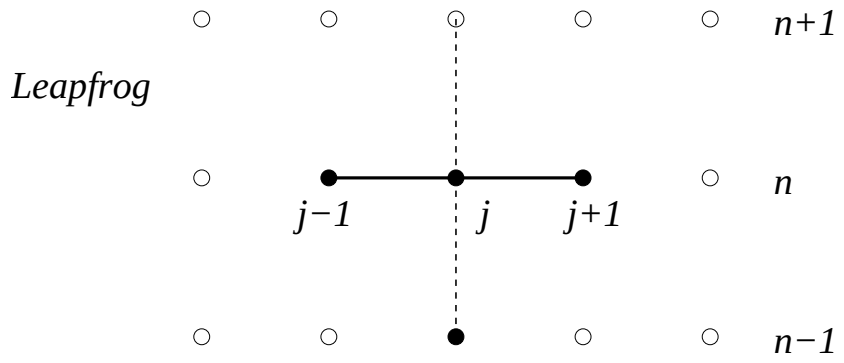

Figure 3.8: Schematic diagram of a Leapfrog evolution scheme.

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

$$\left.\frac{\partial u}{\partial t}\right|_j^n = \frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t} + \mathcal{O}(\Delta t^2)\,, \tag{3.37}$$

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

$$u_j^{n+1} = u_j^{n-1} - \alpha\left(u_{j+1}^n - u_{j-1}^n\right) + \mathcal{O}(\Delta x^2)\,, \tag{3.38}$$

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

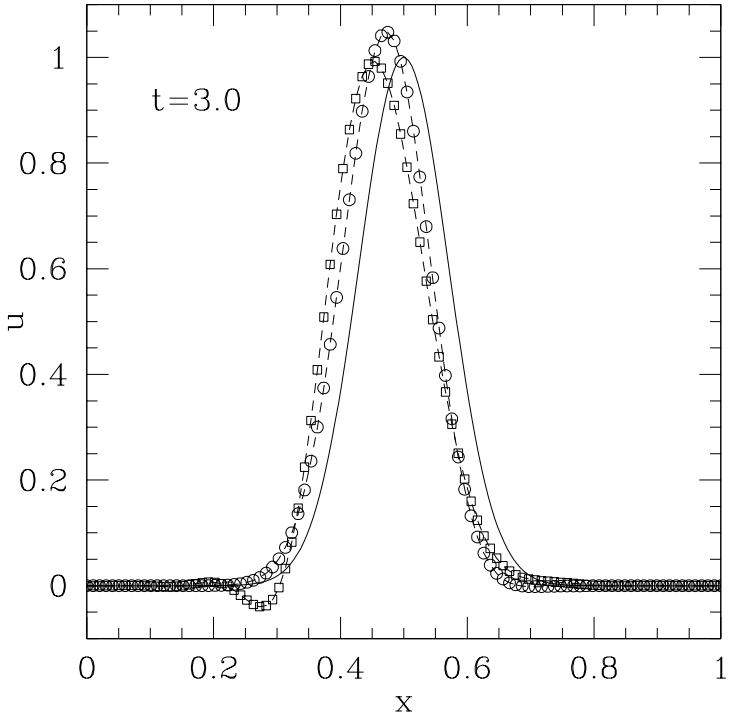

Figure 3.9: This is the same as in Fig. 3.3 but for a Leapfrog scheme. Note how the scheme is stable and does not suffers from a considerable dissipation even for low CFL factors. However, the presence of a little "dip" in the tail of the Gaussian for the case of $c_{\rm CFL} = 0.5$ is the result of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

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

$$\boldsymbol{U}_j^{n+1} = \boldsymbol{U}_j^{n-1} - \frac{\Delta t}{\Delta x}\left[\boldsymbol{F}_{j+1}^n - \boldsymbol{F}_{j-1}^n\right] + \mathcal{O}(\Delta x^2)\,, \tag{3.39}$$

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:

- 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

$$\xi = -\mathrm{i}\alpha\sin(k\Delta x) \pm \sqrt{1 - \left[\alpha\sin(k\Delta x)\right]^2}\,, \tag{3.40}$$

  and so that

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

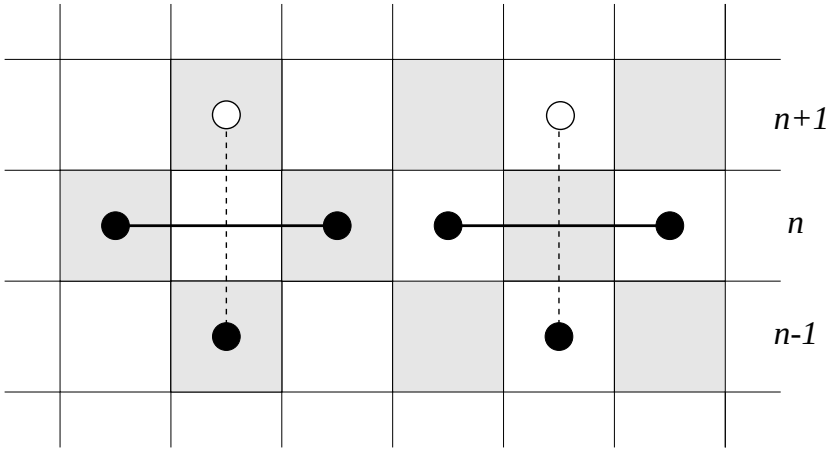

Figure 3.10: Schematic diagram of the decoupled grids in a Leapfrog evolution scheme.

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

- 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 (5.22) 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 (5.17), 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:

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

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

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

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

3. A Leapfrog "half-step":

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

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

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

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

$$u_{j}^{n+1} = u_{j}^{n} - \alpha\left(u_{j+1/2}^{n+1/2} - u_{j-1/2}^{n+1/2}\right) + \mathcal{O}(\Delta x^2) \tag{3.45}$$

$$= u_{j}^{n} - \frac{\alpha}{2}\left(u_{j+1}^{n} - u_{j-1}^{n}\right) + \frac{\alpha^2}{2}\left(u_{j+1}^{n} - 2u_{j}^{n} + u_{j-1}^{n}\right) + \mathcal{O}(\Delta x^2)\,. \tag{3.46}$$

where expression (3.46) has been obtained after substituting (3.44) in (3.45).

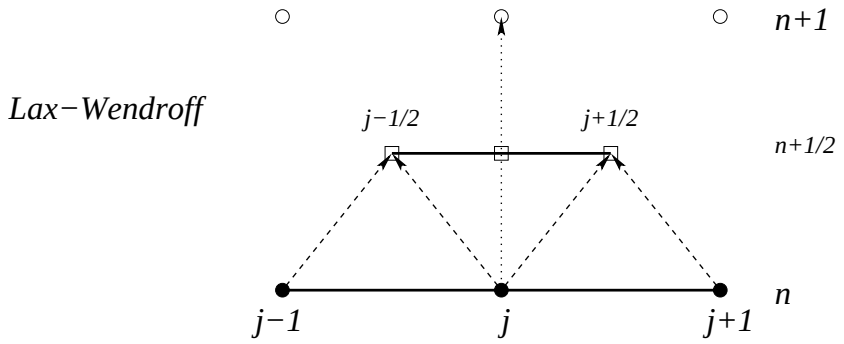

Figure 3.11: Schematic diagram of a Lax-Wendroff evolution scheme.

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

$$\xi = 1 - \mathrm{i}\alpha \sin(k\Delta x) - \alpha^2 \left[1 - \cos(k\Delta x)\right] \ , \tag{3.47}$$

  so that the squared modulus of the amplification factor is

$$|\xi|^2 = 1 - \alpha^2(1 - \alpha^2) \left[1 - \cos^2(k\Delta x)\right] \ . \tag{3.48}$$

  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).

- 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 (3.46) where quantities at time-levels n and $n + 1$ only appear.

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

The idea behind the *iterative Crank-Nicolson* (ICN) scheme is that of transforming a stable implicit method, *i.e.*, the Crank-Nicolson (CN) scheme (see Sect. 8.4.2) into an explicit one through a series of iterations. 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.*,

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = -v \left(\frac{u_{j+1}^{n+1} - u_{j-1}^{n+1}}{2\Delta x} \right) \ . \tag{3.49}$$

This scheme is also known as "backward in time, centred in space" or BTCS (see Sect. 8.4.1) and has amplification factor

$$\xi = \frac{1}{1 + \mathrm{i}\alpha \sin k\Delta x} \ , \tag{3.50}$$

so that $|\xi|^2 < 1$ for any choice of α, thus making the method unconditionally stable.

The *Crank-Nicolson* (CN) scheme, instead, is a second-order accurate method obtained by averaging a BTCS and a FTCS method or, in other words, equations (3.26) and (3.49). Doing so one then finds

$$\xi = \frac{1 + \mathrm{i}\alpha \sin k\Delta x/2}{1 - \mathrm{i}\alpha \sin k\Delta x/2} \ . \tag{3.51}$$

so that the method is stable. Note that although one averages between an explicit and an implicit scheme, terms containing u^{n+1} survive on the right hand side of equation (3.49), thus making the CN scheme implicit.

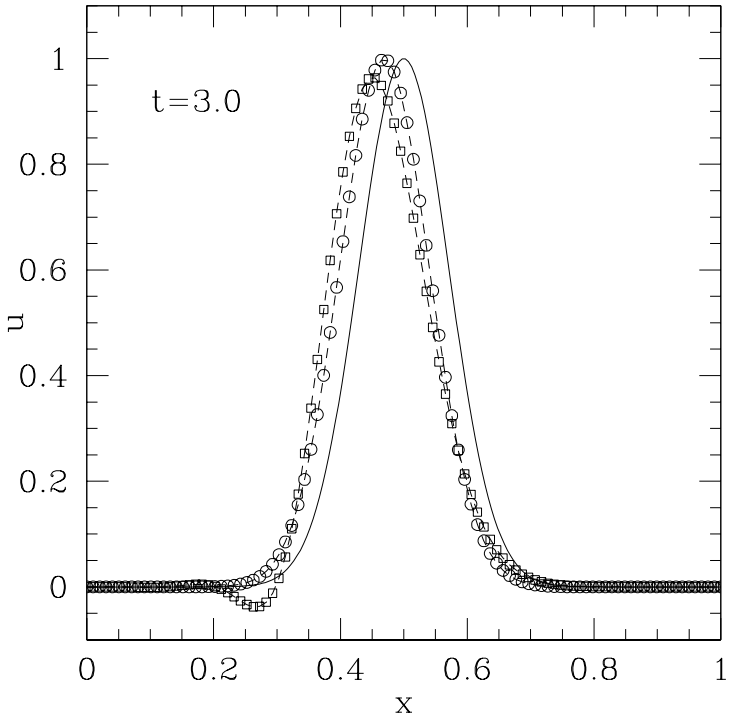

Figure 3.12: This is the same as in Fig. 3.3 but for a Lax-Wendroff scheme. Note how the scheme is stable and does not suffers from a considerable dissipation even for low CFL factors. However, the presence of a little "dip" in the tail of the Gaussian for the case of $c_{\rm CFL} = 0.5$ is the result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

The first iteration of iterative Crank-Nicolson starts by calculating an intermediate variable ${}^{(1)}\tilde{u}$ using equation (3.26):

$$\frac{{}^{(1)}\tilde{u}_j^{n+1} - u_j^n}{\Delta t} = -v \left(\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} \right) . \tag{3.52}$$

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

$${}^{(1)}\bar{u}_j^{n+1/2} \equiv \frac{1}{2} \left({}^{(1)}\tilde{u}_j^{n+1} + u_j^n \right) . \tag{3.53}$$

Finally the timestep is completed by using equation (3.26) again with $\bar{u}$ on the right-hand side:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = -v \left(\frac{{}^{(1)}\bar{u}_{j+1}^{n+1/2} - {}^{(1)}\bar{u}_{j-1}^{n+1/2}}{2\Delta x} \right) . \tag{3.54}$$

Iterated Crank-Nicolson with *two iterations* is carried out in much the same way. After steps (3.52) and (3.53), we calculate

$$\frac{{}^{(2)}\tilde{u}_j^{n+1} - u_j^n}{\Delta t} = -v\left(\frac{{}^{(1)}\bar{u}_{j+1}^{n+1/2} - {}^{(1)}\bar{u}_{j-1}^{n+1/2}}{2\Delta x}\right), \tag{3.55}$$

$${}^{(2)}\bar{u}_j^{n+1/2} \equiv \frac{1}{2}\left({}^{(2)}\tilde{u}_j^{n+1} + u_j^n\right). \tag{3.56}$$

Then the final step is computed analogously to equation (3.54):

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = -v\left(\frac{{}^{(2)}\bar{u}_{j+1}^{n+1/2} - {}^{(2)}\bar{u}_{j-1}^{n+1/2}}{2\Delta x}\right). \tag{3.57}$$

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

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

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

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

$${}^{(4)}\xi = 1 + 2\mathrm{i}\beta - 2\beta^2 - 2\mathrm{i}\beta^3 + 2\beta^4\,, \tag{3.61}$$

where $\beta \equiv (\alpha/2)\sin(k\Delta x)$, and ${}^{(1)}\xi$ corresponds to the FTCS scheme. Note that the amplification factors (3.58) correspond to those one would obtain by expanding equation (3.51) in powers of β.

Computing the squared moduli of (3.58) one encounters an alternating and recursive pattern. In particular, iterations 1 and 2 are unstable ($|\xi|^2 > 1$); iterations 3 and 4 are stable ($|\xi|^2 < 1$) provided $\beta^2 \leq 1$; iterations 5 and 6 are also unstable; iterations 7 and 8 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 (3.26)].

In other words, while the magnitude of the amplification factor for iterated Crank-Nicolson 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.

3.6.1 ICN as a θ-method

In the ICN method the M-th average is made weighting equally the newly predicted solution ${}^{(M)}\tilde{u}_j^{n+1}$ and the solution at the "old" timelevel" u^n. This, however, can be seen as the special case of a more generic averaging of the type

$${}^{(M)}\bar{u}^{n+1/2} = \theta\,{}^{(M)}\tilde{u}^{n+1} + (1-\theta)u^n\,, \tag{3.62}$$

where $0 < \theta < 1$ is a constant coefficient. Predictor-corrector schemes using this type of averaging are part of a large class of algorithms named θ-*methods* [10], and we refer to the ICN generalized in this way as to the "θ-ICN" method.

A different and novel generalization of the θ-ICN can be obtained by *swapping* the averages between two subsequent corrector steps, so that in the M-th corrector step

$$ {}^{(M)}\bar{u}^{n+1/2} = (1-\theta)\, {}^{(M)}\tilde{u}^{n+1} + \theta u^n \ , \tag{3.63} $$

while in the $(M+1)$-th corrector step

$$ {}^{(M+1)}\bar{u}^{n+1/2} = \theta\, {}^{(M+1)}\tilde{u}^{n+1} + (1-\theta) u^n \ . \tag{3.64} $$

Note that as long as the number of iterations is even, the sequence in which the averages are computed is irrelevant. Indeed, the weights θ and $1-\theta$ in eqs. (3.63)–(3.64) could be inverted and all of the relations discussed hereafter for the swapped weighted averages would continue to hold after the transformation $\theta \to 1-\theta$.

Constant Arithmetic Averages

Using a von Neumann stability analysis, Teukolsky has shown that for a hyperbolic equation the ICN scheme with M iterations has an amplification factor [13]

$$ {}^{(M)}\xi = 1 + 2\sum_{n=1}^{M} (-\mathrm{i}\beta)^n \ , \tag{3.65} $$

where $\beta \equiv v[\Delta t/(2\Delta x)]\sin(k\Delta x)$ [1]. More specifically, zero and one iterations yield an unconditionally unstable scheme, while two and three iterations a stable one provided that $\beta^2 \leq 1$; four and five iterations lead again to an unstable scheme and so on. Furthermore, because the scheme is second-order accurate from the first iteration on, Teukolsky's suggestion when using the ICN method for hyperbolic equations was that two iterations should be used *and no more* [13]. This is the number of iterations we will consider hereafter.

Constant Weighted Averages

Performing the same stability analysis for a θ-ICN is only slightly more complicated and truncating at two iterations the amplification factor is found to be

$$ \xi = 1 - 2\mathrm{i}\beta - 4\beta^2\theta + 8\mathrm{i}\beta^3\theta^2 \ , \tag{3.66} $$

where ξ is a shorthand for ${}^{(2)}\xi$. The stability condition in this case translates into requiring that

$$ 16\beta^4\theta^4 - 4\beta^2\theta^2 - 2\theta + 1 \leq 0 \ , \tag{3.67} $$

or, equivalently, that for $\theta > 3/8$

$$ \frac{\sqrt{\frac{1}{2} - \sqrt{2\theta - \frac{3}{4}}}}{2\theta} \leq \beta \leq \frac{\sqrt{\frac{1}{2} + \sqrt{2\theta - \frac{3}{4}}}}{2\theta} \ , \tag{3.68} $$

[1] Note that we define β to have the opposite sign of the corresponding quantity defined in ref. [13]

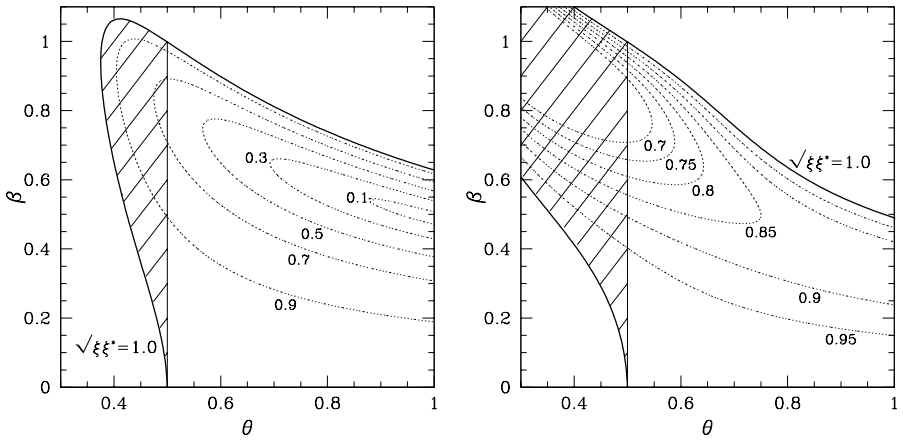

Figure 3.13: *Left panel:* stability region in the (θ, β) plane for the two-iterations θ-ICN for the advection equation (3.2). Thick solid lines mark the limit at which $|\xi| = 1$, while the dotted contours indicate the values of the amplification factor in the stable region. The shaded area for $\theta < 1/2$ refers to solutions that are linearly unstable [15]. *Right panel:* same as in the left panel but when the averages between two corrections are swapped. Note that the amplification factor in this case is less sensitive on θ and always larger than the corresponding amplification factor in the left panel.

which reduces to $\beta^2 \leq 1$ when $\theta = 1/2$. Because the condition (3.68) must hold for every wavenumber k, we consider hereafter $\beta \equiv v\Delta t/(2\Delta x)$ and show in the left panel of Fig. 3.13 the region of stability in the (θ, β) plane. The thick solid lines mark the limit at which $|\xi| = 1$, while the dotted contours indicate the different values of the amplification factor in the stable region.

A number of comments are worth making. Firstly, although the condition (3.68) allows for weighting coefficients $\theta < 1/2$, the θ-ICN is stable only if $\theta \geq 1/2$. This is a known property of the weighted Crank-Nicolson scheme [10] and inherited by the θ-ICN. In essence, when $\theta \neq 1/2$ spurious solutions appear in the method [16] and these solutions are linearly unstable if $\theta < 1/2$, while they are stable for $\theta > 1/2$ [15]. For this reason we have shaded the area with $\theta < 1/2$ in the left panel of Fig. 3.13 to exclude it from the stability region. Secondly, the use of a weighting coefficient $\theta > 1/2$ will still lead to a stable scheme provided that the timestep (*i.e.*, β) is suitably decreased. Finally, as the contour lines in the left panel of Fig. 3.13 clearly show, the amplification factor can be very sensitive on θ.

Swapped weighted averages

The calculation of the stability of the θ-ICN when the weighted averages are swapped as in eqs. (3.63) and (3.64) is somewhat more involved; after some lengthy but straight-

forward algebra we find the amplification factor to be

$$\xi \quad = \quad 1 - 2\mathrm{i}\beta - 4\beta^2\theta + 8\mathrm{i}\beta^3\theta(1-\theta) \; , \tag{3.69}$$

which differs from (3.66) only in that the θ^2 coefficient of the $\mathcal{O}(\beta^3)$ term is replaced by $\theta(1-\theta)$. The stability requirement $|\xi| \leq 1$ is now expressed as

$$16\beta^4\theta^2(1-\theta)^2 - 4\beta^2\theta(2-3\theta) - 2\theta + 1 \leq 0 \; . \tag{3.70}$$

Solving the condition (3.70) with respect to β amounts then to requiring that

$$\beta \geq \frac{\sqrt{2-3\theta-\sqrt{4\theta-11\theta^2+8\theta^3}}}{2(1-\theta)\sqrt{2\theta}} \; , \tag{3.71a}$$

$$\beta \leq \frac{\sqrt{2-3\theta+\sqrt{4\theta-11\theta^2+8\theta^3}}}{2(1-\theta)\sqrt{2\theta}} \; , \tag{3.71b}$$

which is again equivalent to $\beta^2 \leq 1$ when $\theta = 1/2$. The corresponding region of stability is shown in right panel of Fig. 3.13 and should be compared with left panel of the same Figure. Note that the average-swapping has now considerably increased the amplification factor, which is always larger than the corresponding one for the θ-ICN in the relevant region of stability (*i.e.*, for $1/2 \leq \theta \leq 1$ [2]).

[2] Of course, when the order of the swapped averages is inverted from the one shown in eqs. (3.63)–(3.64) the stability region will change into $0 \leq \theta \leq 1/2$.

3.6.2 Summary

In what follow I summarize the most salient aspects of the different finite-difference operators discussed so far and report, for each of them, the truncation error ϵ_{T}, the amplification factor $|\xi|^2$ and the finite-difference representation of the advection equation 3.2.

Method	ϵ_{T}	$\lvert\xi\rvert^2$ for $(k\Delta x \ll 1)$	finite-difference form
Upwind	$\mathcal{O}(\Delta t, \Delta x)$	$1 - 2\lvert\alpha\rvert(1 - \lvert\alpha\rvert)\cos(k\Delta x)$	$u_j^{n+1} = u_j^n \mp \alpha(u_{j\pm1}^n - u_j^n)$
FTCS	$\mathcal{O}(\Delta t, \Delta x^2)$	$1 + \sin^2(k\Delta x)\alpha^2$	$u_j^{n+1} = u_j^n - \alpha(u_{j+1}^n - u_{j-1}^n)$
Lax Friedrichs	$\mathcal{O}(\Delta t, \Delta x^2)$	$1 - \sin^2(k\Delta x)(1 - \alpha^2)$	$u_j^{n+1} = \frac{1}{2}(u_{j+1}^n + u_{j-1}^n) - \alpha(u_{j+1}^n - u_{j-1}^n)$
Lepafrog	$\mathcal{O}(\Delta t^2, \Delta x^2)$	1	$u_j^{n+1} = u_j^{n-1} - \alpha(u_{j+1}^n - u_{j-1}^n)$
Lax Wendroff	$\mathcal{O}(\Delta t^2, \Delta x^2)$	$1 - \alpha^2(1 - \alpha^2)\sin^2(k\Delta x)$	$u_j^{n+1} = u_j^n - \frac{1}{2}\alpha(u_{j+1}^n - u_{j-1}^n) - \frac{1}{2}\alpha^2(u_{j+1}^n - 2u_j^n + u_{j-1}^n)$

Table 3.1: Schematic summary of the finite-difference operators discussed so far.

Chapter 4

Dissipation, Dispersion and Convergence

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.

4.1 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

$$u_j^{n+1} = \frac{1}{2}(u_{j+1}^n + u_{j-1}^n) - \frac{\alpha}{2}(u_{j+1}^n - u_{j-1}^n) + \mathcal{O}(\Delta x^2) \, . \tag{4.1}$$

We have also mentioned how expression (4.1) can be rewritten as

$$u_j^{n+1} = u_j^n - \frac{\alpha}{2}(u_{j+1}^n - u_{j-1}^n) + \frac{1}{2}(u_{j+1}^n - 2u_j^n + u_{j-1}^n) + \mathcal{O}(\Delta x^2) \, , \tag{4.2}$$

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

$$\frac{\partial u}{\partial t} + v\frac{\partial u}{\partial x} = \varepsilon_{\text{LF}} \frac{\partial^2 u}{\partial x^2} \, , \tag{4.3}$$

that is an advection-diffusion equation in which a dissipative term

$$\varepsilon_{\text{LF}} \equiv v\frac{\Delta x^2}{2\Delta t} \, , \tag{4.4}$$

is present. Given a computational domain of length L, this scheme will therefore have a typical diffusion timescale $\tau \simeq L^2/\varepsilon_{\text{LF}}$. 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

$$u_j^{n+1} = u_j^n - \alpha \left(u_j^n - u_{j-1}^n\right) + \mathcal{O}(\Delta x^2) \,, \tag{4.5}$$

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

$$\frac{\partial u}{\partial t} + v\frac{\partial u}{\partial x} = \varepsilon_{\rm UW}\frac{\partial^2 u}{\partial x^2} \,, \tag{4.6}$$

where

$$\varepsilon_{\rm UW} \equiv \frac{v\Delta x}{2} \,. \tag{4.7}$$

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. 4.2 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, $\alpha = |v|\Delta t/\Delta x < 1$ so that the ratio of the dissipation coefficients can be written as

$$\frac{\varepsilon_{\rm LF}}{\varepsilon_{\rm UW}} = \frac{1}{\alpha} \geq 1 \,, \qquad \text{for} \quad \alpha \in [0,1] \,. \tag{4.8}$$

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) [*cf.* eq. (3.46)]

$$u_j^{n+1} = u_j^n - \alpha \left(u_{j+1}^n - u_{j-1}^n\right) + \frac{\alpha^2}{2}\left(u_{j+1}^n - 2u_j^n + u_{j-1}^n\right) + \mathcal{O}(\Delta x^2) \,. \tag{4.9}$$

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

$$\frac{\partial u}{\partial t} + v\frac{\partial u}{\partial x} = \varepsilon_{\rm LW}\frac{\partial^2 u}{\partial x^2} + \beta_{\rm LW}\frac{\partial^3 u}{\partial x^3} \,, \tag{4.10}$$

where

$$\varepsilon_{\rm LW} \equiv \frac{\alpha v\Delta x}{2} \,, \qquad \beta_{\rm LW} \equiv -\frac{v\Delta x^2}{6}\left(1-\alpha^2\right) \,. \tag{4.11}$$

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 $\varepsilon_{\rm LW}$. Using expression (4.9), it is easy to deduce the

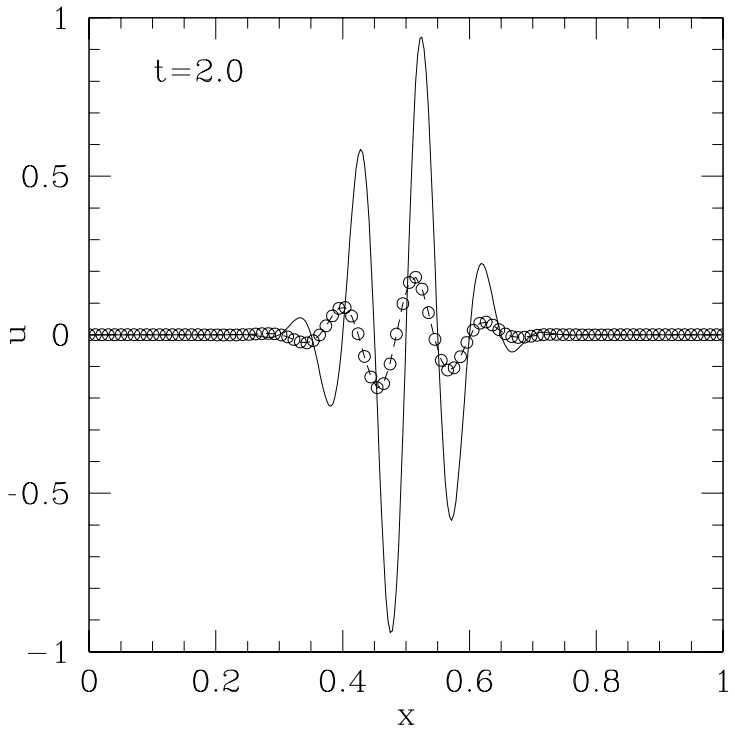

Figure 4.1: Time evolution of a wave-packet initially centred at $x = 0.5$ computed using a Lax-Friedrichs scheme with $C_{\rm CFL} = 0.75$. The analytic solution at time $t = 2$ is shown with a solid line the dashed lines are used to represent the numerical solution at the same time. Note how dissipation reduces the amplitude of the wave-packet but does not change sensibly the propagation of the wave-packet.

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

$$\varepsilon_{\rm LW} = \alpha^2 \varepsilon_{\rm LF} \ll \varepsilon_{\rm LF} \ , \tag{4.12}$$

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

The simplest way of quantifying the effects introduced by the right-hand-sides of equations (4.3), (4.6), and (4.10) is by using a single Fourier mode with angular frequency ω and wavenumber k, propagating in the positive x-direction, *i.e.*,

$$u(x,t) = e^{{\rm i}(kx-\omega t)} \ . \tag{4.13}$$

It is then easy to verify that in the continuum limit

$$\frac{\partial u}{\partial t} = -{\rm i}\omega u \ , \qquad \frac{\partial u}{\partial x} = {\rm i}ku \ , \qquad \frac{\partial^2 u}{\partial x^2} = -k^2 u \ , \qquad \frac{\partial^3 u}{\partial x^3} = -{\rm i}k^3 u \ . \tag{4.14}$$

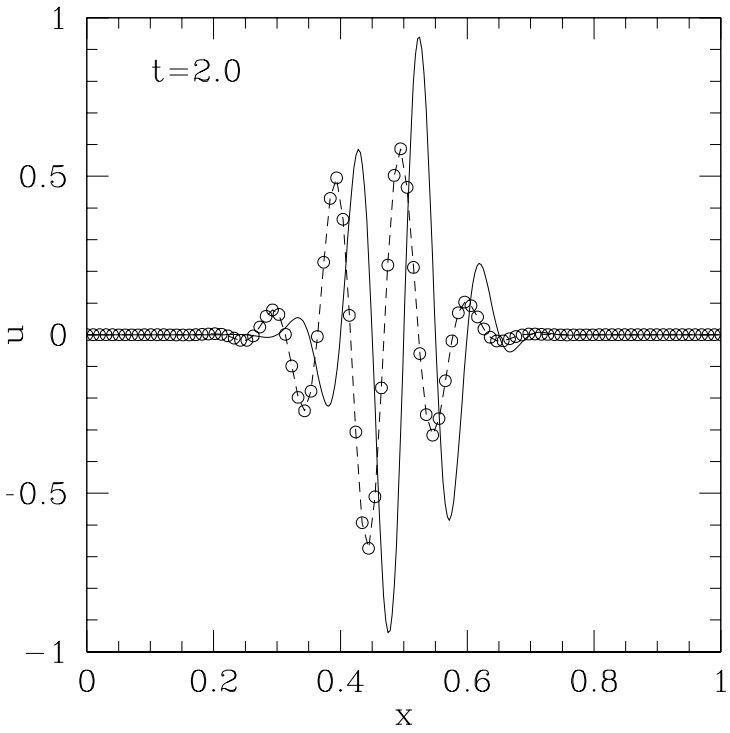

Figure 4.2: Time evolution of a wave-packet initially centred at $x = 0.5$ computed using a Lax-Wendroff scheme with $C_{\rm CFL} = 0.75$. The analytic solution at time $t = 2$ is shown with a solid line the dashed lines are used to represent the numerical solution at the same time. Note how the amplitude of the wave-packet is not drastically reduced but the group velocity suffers from a considerable error.

In the case in which the finite difference scheme provides an accurate approximation to a purely advection equation, the relations (4.14) lead to the obvious dispersion relation $\omega = vk$, so that the *numerical* mode $\tilde{u}(x,t)$ will have a solution

$$\tilde{u}(x,t) = e^{{\rm i}k(x-vt)} \ , \tag{4.15}$$

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

However, it is simple to verify that the advection-diffusion equation approximated by the Lax-Friedrichs scheme (4.3), will have a corresponding solution

$$\tilde{u}(x,t) = e^{-\varepsilon_{\rm LF} k^2 t} e^{{\rm i}k(x-vt)} \ , \tag{4.16}$$

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. 4.1 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 (4.10), will have a solution given by

$$\tilde{u}(x,t) = e^{-\varepsilon_{\rm LW} k^2 t} e^{{\rm i}k\left[x-\left(v+\beta_{\rm LW} k^2\right)t\right]} , \tag{4.17}$$

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

$$c_{\rm p} = \frac{\omega}{k} = v + \beta_{\rm LW} k^2 , \qquad \text{and} \qquad c_{\rm g} = \frac{\partial \omega}{\partial k} = v + 3\beta_{\rm LW} k^2 , \tag{4.18}$$

and provides a simple interpretation of the results shown in Fig. 4.2.

4.2 Measuring Dissipation and Convergence

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. In what follows we discuss some of these tools and how they can be used to ascertain a fundamental property of the numerical solution: its convergence

4.2.1 The summarising power of norms

A very useful tool that can be used in this context is the calculation of the *"norms"* of the quantity we are interested in. In the continuum limit the *p-norm* is defined as

$$\|u\|_p = \frac{1}{(b-a)} \left(\int_a^b |u(x,t)|^p {\rm d}x \right)^{1/p} . \tag{4.19}$$

and has the same dimensions of the originating quantity $u(x,t)$. The extension of this concept to a discretised space and time is straightforward and yields the commonly used norms

$$1-{\rm norm} :: \qquad ||u||(t^n) = \frac{1}{N} \sum_{j=1}^{N} |u_j^n| , \tag{4.20}$$

$$2-{\rm norm} :: \qquad ||u||^2(t^n) = \frac{1}{N} \left(\sum_{j=1}^{N} (u_j^n)^2 \right)^{1/2} , \tag{4.21}$$

$$p-{\rm norm} :: \qquad ||u||^p(t^n) = \frac{1}{N} \left(\sum_{j=1}^{N} (u_j^n)^p \right)^{1/p} , \tag{4.22}$$

$$\text{infinity} - \text{norm} :: \qquad ||u||_\infty(t^n) = \max_{j=1,\ldots N}(|u_j^n|) . \tag{4.23}$$

In the case of a scalar wave equation (see Sect. 5 for a discussion), 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.

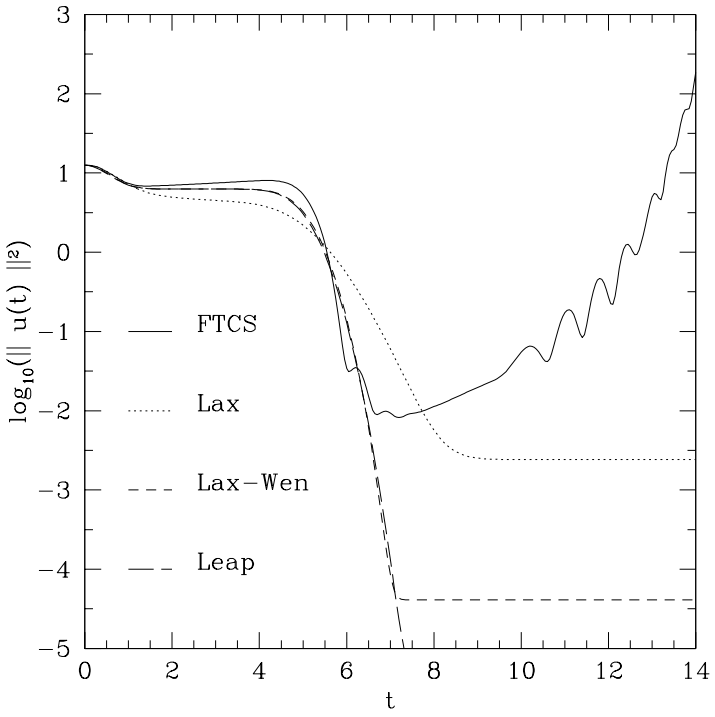

Figure 4.3: Time evolution of the logarithm of the 2-norms for the different numerical schemes discussed so far. Sommerfeld outgoing boundary conditions were used in this example.

Fig. 4.2 compares the 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 $t \sim 6$.

4.2.2 Consistency and Convergence

Consider therefore a PDE of the type

$$\mathcal{L}(u) - f = 0 \,, \tag{4.24}$$

where $\mathcal{L}$ is a second-order differential quasi-linear operator [*cf.* eq. (1.1)]. Let also $\mathcal{L}_\Delta$ be the discretized representation of such continuum differential operator and $\epsilon = \mathcal{O}(\Delta x^p, \Delta t^q)$ the associated truncation error, *i.e.*,

$$\mathcal{L}_\Delta(u_i^n) - f_i^n = 0 + \mathcal{O}(\Delta x^p, \Delta t^q) \; . \tag{4.25}$$

For compactness let us assume that largest contribution to the truncation error can be expressed simply as $\epsilon = Ch^p = \mathcal{O}(h^p)$ where h corresponds to either the spatial or time discretization and C is a real constant coefficient. The finite-difference representation $\mathcal{L}_\Delta$ is said to be *consistent* if

$$\lim_{h \to 0} \epsilon = 0 \; , \tag{4.26}$$

Let $u(x,t)$ represent the exact solution to a PDE and $\tilde{u}$ the exact solution of the finite-difference equation that approximates the PDE with a truncation error $\mathcal{O}(\Delta x^p, \Delta t^q)$. The finite-difference equation is said to be *convergent* when the truncation error tends to zero as a power of p in Δx and a power of q in Δt, namely

$$\lim_{h \to 0} \epsilon = \Delta x^p + \Delta t^q \; , \tag{4.27}$$

Note that this condition is much more severe that the simple requirement that the truncation error will tend to zero as Δx and Δt tend to zero. The latter condition, in fact, does not ensure that the numerical solution is approaching the exact one at the *expected* rate, that is the rate determined by the truncation error and consequent to the choice of the given finite-difference representation of the continuum differential operator.

Since checking convergence essentially amounts to measuring how the truncation error changes with resolution, it is useful to define a *local* (*i.e.*, pointwise) deviation from the exact solution u at $x = x_i$ as

$$\epsilon_j(h) = u_j^{(h)} - u(x_j) \tag{4.28}$$

be the magnitude of the *largest* truncation error (and which could be either in space or in time) associated to the numerical solution $u_j^{(h)}$ obtained with grid spacing h. If the numerical method used is *p*-th order accurate, then

$$\epsilon_j(h) = Ch^p \; + \mathcal{O}(h^{p+1}) \; , \tag{4.29}$$

where C is a constant real coefficient. A different solution computed with a grid spacing k will have at the same spatial position x_j a corresponding truncation $\epsilon_j(k)$ error, so that *error ratio* will be

$$R_j(h,k) \equiv \frac{\epsilon_j(h)}{\epsilon_j(k)} \; , \tag{4.30}$$

and the "numerical" local convergence order, that is the order of convergence as measured from the two numerical solutions at x_j will be

$$\tilde{p} \equiv \frac{\log R_j(h,k)}{\log(h/k)} \; . \tag{4.31}$$

In the rather common case in which $k = h/2$, expressions (4.30) reduces to

$$R_j(h, h/2) = 2^{\tilde{p}} ,$$

and the overall order of accuracy is measured numerically as $\tilde{p} = \log_2(R)$. As we will discuss in the following Section, *the discrete representation of the continuum equations is said to be convergent if and only if* $\tilde{p} = p$, *i.e.,* if

$$\lim_{h\to 0} \tilde{p} \equiv \frac{\log(\epsilon)}{\log(Ch)} = p . \tag{4.32}$$

Stated differently, convergence requires not only that the error is decreasing and thus that the method is consistent (see Sect. 4.2.3) but that it is decreasing at the *expected rate*.

In general there will be a minimun resolution, say $h_{\min}$, below which the truncation error will dominate over the others, *e.g.,* round-off error. Clearly, one should expect convergence only for $h < h_{\min}$ and the solution in this case is said to be in a *convergent regime*.

What discussed so far assumes the knowledge of the exact solution, which, in general, is not available. This does not represent a major obstacle and the convergence test can still be performed by simply employing a third numerical evaluation of the solution. This is referred to as a *"self-convergence"* test and exploits the fact that the difference between two numerical solutions does not depend on the actual exact solution

$$u_j^{(h)} - u_j^{(k)} = \Big(\epsilon_j(h) - u(x_j)\Big) - \Big(\epsilon_j(k) - u(x_j)\Big) = \epsilon_j(h) - \epsilon_j(k) ,$$

where of course the two solutions $u_j^{(h)}$ and $u_j^{(k)}$ should be evaluated at the same grid-point x_j. If one of the numerical solutions is not availble at such a point (*e.g.,* because the spacing used is not uniform) a suitable interpolation is needed and attention must be paid that the error it introduces is much smaller than either $\epsilon_j(h)$ or $\epsilon_j(k)$ in order not to spoil the convergence test.

With (4.29) in mind and using three different numerical solutions $u_j^{(h)}$, $u_j^{(k)}$, $u_j^{(l)}$ with grid spacings such that $h > k > l$, the numerical error ratio is then defined as

$$R_j(h, k; l) \equiv \frac{u_j^{(h)} - u_j^{(l)}}{u_j^{(k)} - u_j^{(l)}} = \frac{\epsilon_j(h) - \epsilon_j(l)}{\epsilon_j(k) - \epsilon_j(l)} = \frac{h^{\tilde{p}} - l^{\tilde{p}}}{k^{\tilde{p}} - l^{\tilde{p}}} , \tag{4.33}$$

where the numerical solution $u_j^{(l)}$ with the associated error $\epsilon_j(l)$ has the role of "reference" solution since it is the one with the smallest error. In the common case in which $k = h/2$ and $l = k/2 = h/4$, the error ratio assumes the simple expresion

$$R(h, h/2; h/4) = 2^{\tilde{p}} - 1 ,$$

so that the computed overall accuracy order is $\tilde{p} = \log_2(R + 1)$.

As a final comment we note that all what discussed so far for a local convergence analysis can be extended to a *global* evaluation of the truncation error and this amounts to essentially replacing all the error estimates discussed above with the corresponding p-norms.

4.2.3 Convergence and Stability

We conclude this Chapter with an important theorem that brings together many of the different concepts exposed so far and provides a unique interpretation for the interplay between consistency, convergence and stability. We have seen in the previous Section that The finite-difference representation is said to be *consistent* if

$$\lim_{h\to 0} \epsilon = 0 \,, \tag{4.34}$$

and it will be said to be *convergent* if

$$\lim_{h\to 0} \tilde{p} \equiv \frac{\log(\epsilon)}{\log(Ch)} = p \,. \tag{4.35}$$

Clearly, *also* for a convergent solution $\epsilon \to 0$ as $h \to 0$; however, conditions (4.27) and (4.32) underline that while a convergent solution is *also* consistent, the latter is not necessarily true. Stated differently, while there are infinite consistent representation of the differential operator, only one will be convergent.

There are numerous ways in which a consistent representation of a differential operator may not be convergent and in large majority of the cases the lack of convergence is related to a programming error (or "bug"). Because of this, convergence tests represent the most efficient if not the only way of validating that the discrete form of the equations represents a faithful representation of the continuum ones (and hence of picking out bugs!).

The knowledge of convergence has also another rewarding aspect and this is beautifully summarised in the following theorem:

> **Theorem** *Given a properly posed initial-value problem and a finite difference approximation to it that satisfies the consistency condition, stability is the necessary and sufficient condition for convergence.*

This theorem, known as the *"Lax equivalence theorem"*, is very powerful as it shows that for an initial-value problem which has been discretised with a consistent finite-difference operator, the concept of stability and convergence are interchangable. In general, therefore, proving that the numerical solution is convergent will not only validate that the discrete form of the equations represents a faithful representation of the continuum ones, but also that the solution will be bounded at all times.

Chapter 5

The Wave Equation in 1D

The numerical solution of the wave equation offers a good example of how a higher-order (in space and time) PDE can be easily solved numerically through the solution of a system of coupled 1st-order PDEs.

In one spatial dimenion (1D) the wave equation has the general form:

$$\frac{\partial^2 u}{\partial t^2} = v^2 \frac{\partial^2 u}{\partial x^2} \,, \tag{5.1}$$

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 (5.1) as a system of coupled first-order conservative PDE. For this we set

$$r = v\frac{\partial u}{\partial x} \,, \tag{5.2}$$

$$s = \frac{\partial u}{\partial t} \,, \tag{5.3}$$

so that (5.1) can be rewritten as a system of 3 coupled, first-order differential equations

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

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.

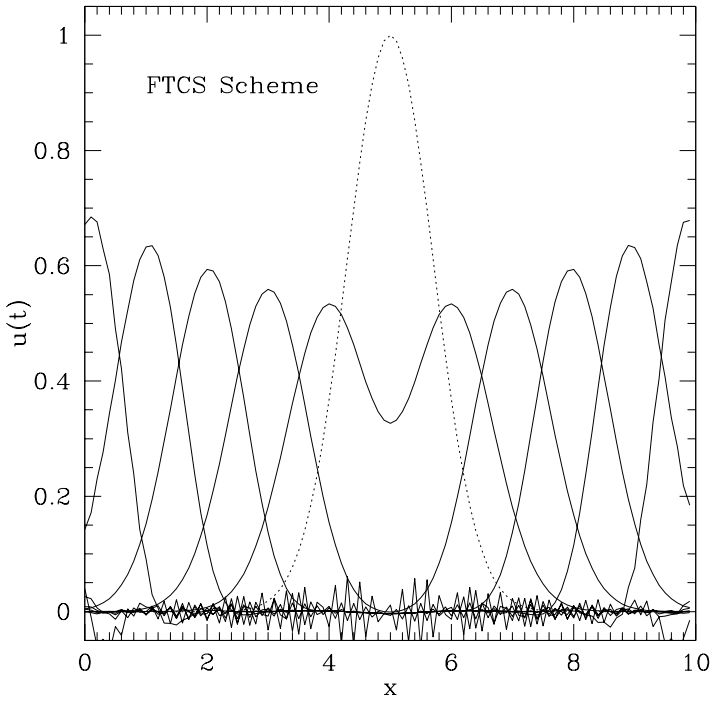

Figure 5.1: Plot of the time evolution of the wave equation when the FTCS scheme is used. The initial conditions were given by a Gaussian centered at $x = 5$ with unit variance and are shown with the dotted line. Note the growth of the wave crests and the appearance of short wavelength noise. When this happens, the numerical errors have grown to be comparable with the solution which will be rapidly destroyed.

In vector notation the system (5.4) can be symbolically written as

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{F}(\boldsymbol{U})}{\partial x} = 0 \,, \tag{5.4}$$

where

$$\boldsymbol{U} = \begin{pmatrix} r \\ s \end{pmatrix} \,, \qquad \text{and} \qquad \boldsymbol{F}(\boldsymbol{U}) = \begin{pmatrix} 0 & -v \\ -v & 0 \end{pmatrix} \boldsymbol{U} \,. \tag{5.5}$$

5.1 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 (5.4) and obtain

$$r_j^{n+1} = r_j^n + \frac{\alpha}{2}(s_{j+1}^n - s_{j-1}^n) + \mathcal{O}(\Delta x^2) \,, \tag{5.6}$$

$$s_j^{n+1} = s_j^n + \frac{\alpha}{2}(r_{j+1}^n - r_{j-1}^n) + \mathcal{O}(\Delta x^2) \,, \tag{5.7}$$

Once the value of s_j^{n+1} has been calculated, the value of u can be integrated in time according to equation (5.3) so that

$$u_j^{n+1} = u_j^n + \Delta t s_j^n + \mathcal{O}(\Delta x^2) \,, \tag{5.8}$$

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. 5.1 which reports the solution of equations (5.6), (5.6) and (5.8) 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.

5.2 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 (5.4) and easily obtain

$$r_j^{n+1} = \frac{1}{2}(r_{j+1}^n + r_{j-1}^n) + \frac{\alpha}{2}(s_{j+1}^n - s_{j-1}^n) + \mathcal{O}(\Delta x^2) \,, \tag{5.9}$$

$$s_j^{n+1} = \frac{1}{2}(s_{j+1}^n + s_{j-1}^n) + \frac{\alpha}{2}(r_{j+1}^n - r_{j-1}^n) + \mathcal{O}(\Delta x^2) \,, \tag{5.10}$$

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 (5.8).

The solution of equations (5.9), (5.9) and (5.8) with the same initial data used in Fig. 5.1 is shown in Fig. 5.2. 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.

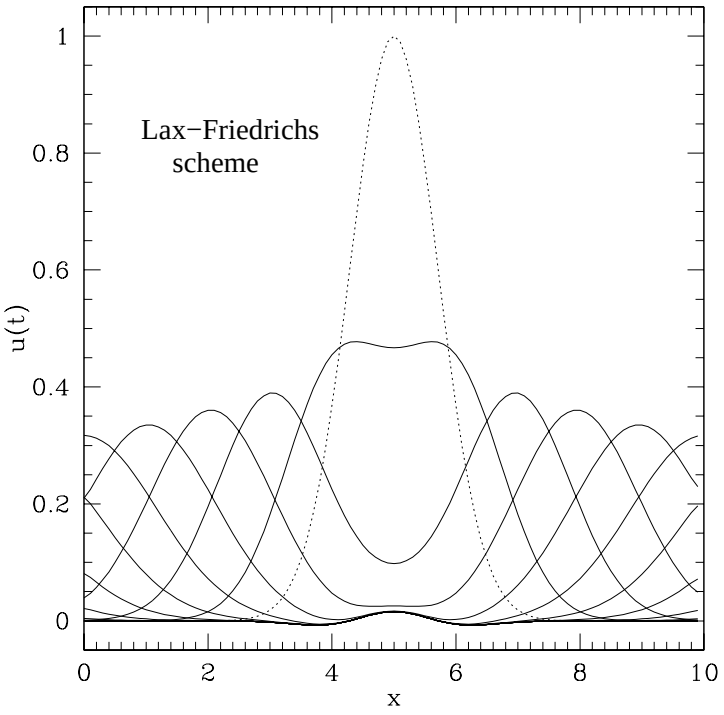

Figure 5.2: The same as in Fig. 5.1 but when the Lax-Friedrichs scheme is used. Note the absence of the late time instabilities but also the effects of the numerical diffusion that widens and lowers the wave fronts.

5.3 The Leapfrog Scheme

We can adapt the Leapfrog scheme to equations (5.4) for the solution of the wave equation in one dimension, centering variables on appropriate half-mesh points

$$r^n_{j+\frac{1}{2}} \equiv v\frac{\partial u}{\partial x}\bigg|^n_{j+\frac{1}{2}} = v\frac{u^n_{j+1}-u^n_j}{\Delta x}+\mathcal{O}(\Delta x)\,, \tag{5.11}$$

$$s^{n+\frac{1}{2}}_j \equiv \frac{\partial u}{\partial t}\bigg|^{n+\frac{1}{2}}_j = \frac{u^{n+1}_j-u^n_j}{\Delta t}+\mathcal{O}(\Delta t)\,, \tag{5.12}$$

and then considering the Leapfrog representation of equations (5.4)

$$r_{j+\frac{1}{2}}^{n+1} = r_{j+\frac{1}{2}}^{n} + \alpha\left(s_{j+1}^{n+\frac{1}{2}} - s_{j}^{n+\frac{1}{2}}\right) + \mathcal{O}(\Delta x^2)\,, \tag{5.13}$$

$$s_{j}^{n+\frac{1}{2}} = s_{j}^{n-\frac{1}{2}} + \alpha\left(r_{j+\frac{1}{2}}^{n} - r_{j-\frac{1}{2}}^{n}\right) + \mathcal{O}(\Delta x^2)\,, \tag{5.14}$$

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

$$u_j^{n+1} = u_j^n + \frac{\Delta t}{2}(s_j^{n+1} + s_j^n) + \mathcal{O}(\Delta x^2) = u_j^n + \frac{\Delta t}{2} s_j^{n+1/2} + \mathcal{O}(\Delta x^2)\,. \tag{5.15}$$

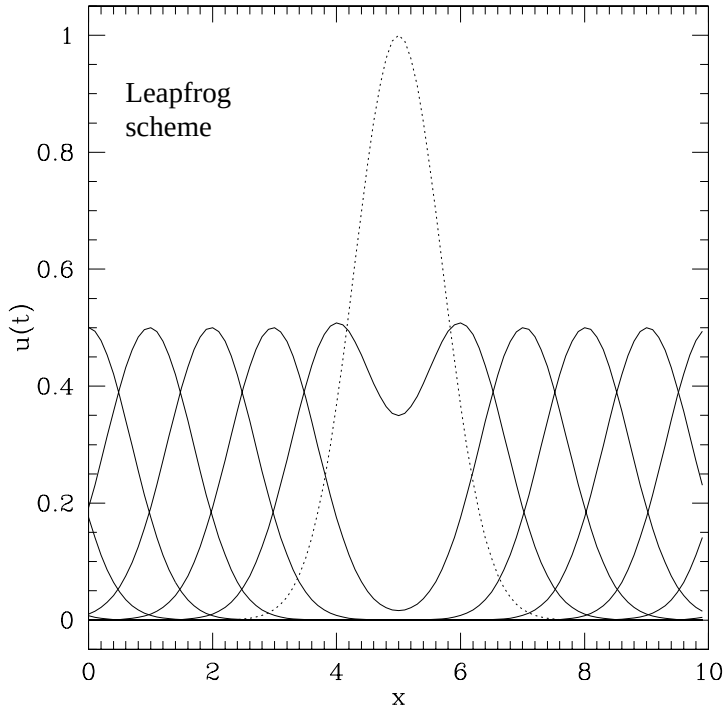

Figure 5.3: The same as in Fig. 5.1 but when the Leapfrog scheme is used. Note the absence of the late time instabilities and of the effects of the numerical diffusion.

A simple substitution of (5.11) and (5.12) into (5.13) and (5.14) shows how the

Leapfrog representation of the wave equation is nothing but its second-order differencing:

$$\frac{u_j^{n+1} - 2u_j^n + u_j^{n-1}}{\Delta t^2} = v^2 \left(\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x^2} \right) + \mathcal{O}(\Delta t^2, \Delta x^2) \,, \tag{5.16}$$

so that the solution at the new time-level is

$$u_j^{n+1} = \alpha^2 u_{j+1}^n + 2u_j^n \left(1 - \alpha^2\right) + \alpha^2 u_{j-1}^n - u_j^{n-1} + \mathcal{O}(\Delta x^4) \,. \tag{5.17}$$

Note that as formulated in (5.17), the Leapfrog scheme has been effectively recast into a "one-level" scheme.

The solution of equations (5.17) and (5.15) with the same initial data used in Fig. 5.1 is shown in Fig. 5.3. 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.

5.4 The Lax-Wendroff Scheme

Also in the case, the application of this scheme to our system of equations (5.4) is straightforward. We can start with the time evolution of the variable r to obtain

$$r_j^{n+1} = r_j^n + \alpha \left(s_{j+1/2}^{n+1/2} - s_{j-1/2}^{n+1/2} \right) + \mathcal{O}(\Delta x^2) \,, \tag{5.18}$$

where the terms in the spatial derivatives are computed as

$$s_{j+1/2}^{n+1/2} = \frac{1}{2}\left(s_j^n + s_{j+1}^n\right) + \alpha \left(r_{j+1}^n - r_j^n\right) + \mathcal{O}(\Delta x^2) \,, \tag{5.19}$$

$$s_{j-1/2}^{n+1/2} = \frac{1}{2}\left(s_j^n + s_{j-1}^n\right) + \alpha \left(r_j^n - r_{j-1}^n\right) + \mathcal{O}(\Delta x^2) \,. \tag{5.20}$$

As done for the advection equation, it is convenient not to use equations (5.18) and (5.19) as two coupled but distinct equations and rather to combine them into two "one-level" evolution equations for r and s

$$r_j^{n+1} = r_j^n + \alpha \left[\frac{1}{2}(s_{j+1}^n - s_{j-1}^n) + \frac{\alpha}{2}(r_{j+1}^n - 2r_j^n + r_{j-1}^n) \right] + \mathcal{O}(\Delta x^2) \,, \tag{5.21}$$

$$s_j^{n+1} = s_j^n + \alpha \left[\frac{1}{2}(r_{j+1}^n - r_{j-1}^n) + \frac{\alpha}{2}(s_{j+1}^n - 2s_j^n + s_{j-1}^n) \right] + \mathcal{O}(\Delta x^2) \,. \tag{5.22}$$

The solution of equations (5.21), (5.22) and (5.15) with the same initial data used in Fig. 5.1 is shown in Fig. 5.4. 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.

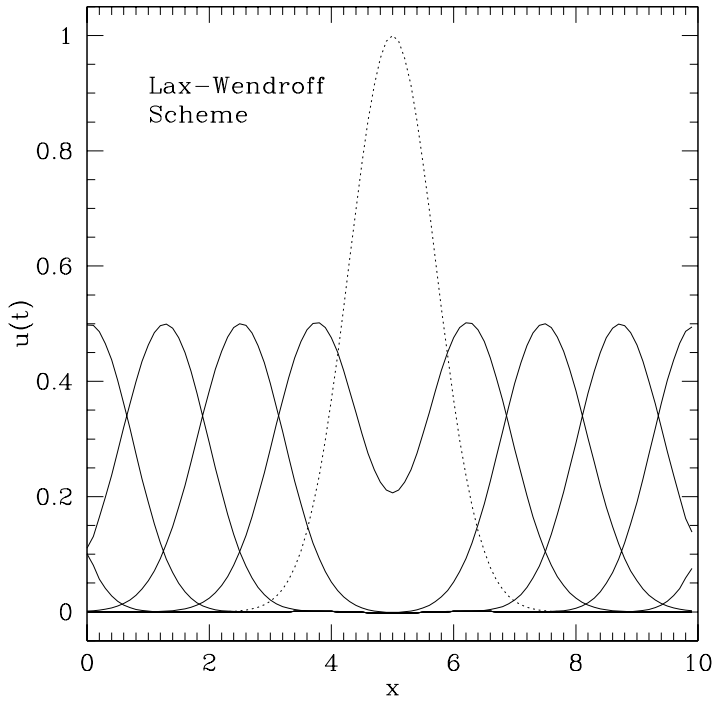

Figure 5.4: The same as in Fig. 5.1 but when the Lax-Wendroff scheme is used. Note the absence of the late time instabilities and of the effects of the numerical diffusion.

Chapter 6

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 (3.26), (5.16), (5.21) and (5.22) 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-type*** 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-type*** 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;

6.1 Outgoing Wave BCs: the outer edge

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

$$\frac{\partial u}{\partial t} + v\frac{\partial u}{\partial x} = 0 \tag{6.1}$$

A finite-difference, first-order accurate representation of equation (6.1) 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)

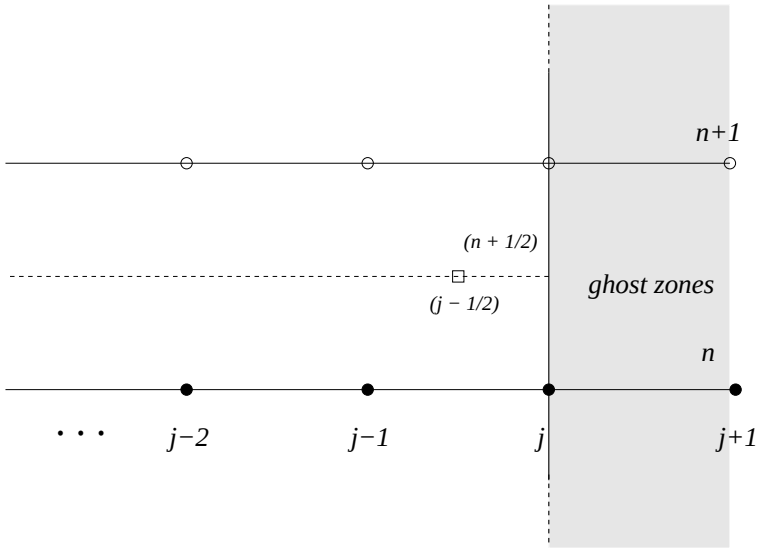

Figure 6.1: Schematic representation of the centering for a first-order, outgoing-wave Sommerfeld boundary conditions. An equivalent one can be drawn for an ingoing-wave.

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

and which leads to

$$u_{j+1}^{n+1}\,(1+\alpha)=u_j^{n+1}\,(-1+\alpha)+u_{j+1}^{n}\,(1-\alpha)+u_j^{n}\,(1+\alpha) \tag{6.2}$$

Expression (6.2) can also be written as

$$u_{j+1}^{n+1}=u_j^{n}-u_j^{n+1}Q+u_{j+1}^{n}Q\;, \tag{6.3}$$

where

$$Q\equiv\frac{1-\alpha}{1+\alpha}\;. \tag{6.4}$$

The use of expression (6.3) for the outermost grid point where the wave is outgoing will provide first-order accurate and stable boundary conditions. Note, however, that (6.3) 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.

6.2 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:

$$\frac{\partial u}{\partial t} - \frac{\partial u}{\partial x} = 0 \tag{6.5}$$

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

$$u_j^{n+1} = u_{j+1}^n - u_{j+1}^{n+1}Q + u_j^n Q\,, \tag{6.6}$$

where Q is the same quantity as for the out-going wave. If we use equations (6.3) and (6.6) 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.).

6.3 Periodic Boundary Conditions

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

$$u_1^{n+1} = u_{J-1}^{n+1}, \qquad u_J^{n+1} = u_2^{n+1}\,, \tag{6.7}$$

In the case of a Gaussian leaving the center of the numerical grid, these boundary conditions effectively produce a reflection. The boundary conditions (6.7) 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

$$u_0^{n+1} = u_J^{n+1}, \qquad u_{J+1}^{n+1} = u_1^{n+1}\,. \tag{6.8}$$

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

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

where

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

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

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

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

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

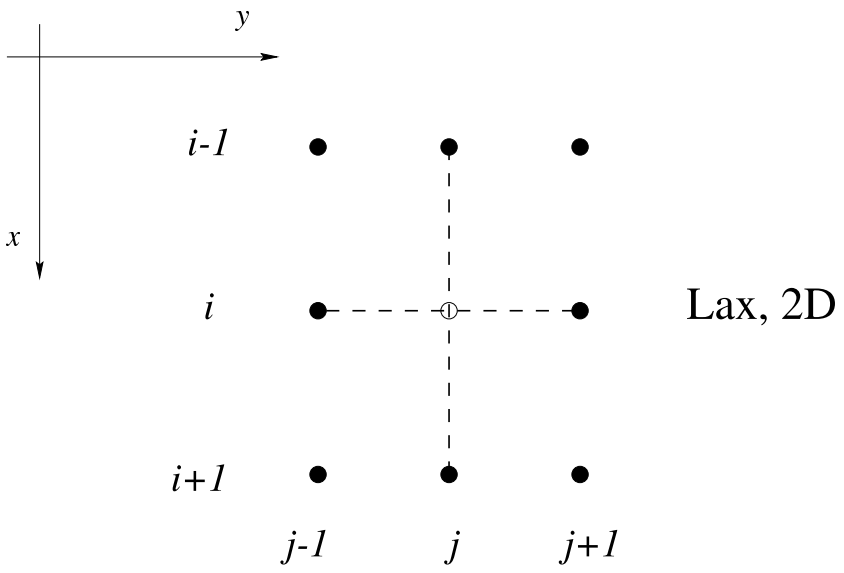

Figure 7.1: Schematic diagram of a Lax-Friedrichs evolution scheme in two dimensions. Note that the center of the cross-like stencil is not used in this case.

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

$$r_{i,j}^{n+\frac{1}{2}} = \frac{1}{4}\left(r_{i+1,j}^{n} + r_{i,j+1}^{n} + r_{i-1,j}^{n} + r_{i,j-1}^{n}\right) + \frac{\alpha}{4}\left(s_{i+1,j}^{n} - s_{i-1,j}^{n}\right) , \tag{7.18}$$

$$l_{i,j}^{n+\frac{1}{2}} = \frac{1}{4}\left(l_{i+1,j}^{n} + l_{i,j+1}^{n} + l_{i-1,j}^{n} + l_{i,j-1}^{n}\right) + \frac{\alpha}{4}\left(s_{i,j+1}^{n} - s_{i,j-1}^{n}\right) , \tag{7.19}$$

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

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

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

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

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

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

$$r_{i,j}^{n+1} = \frac{1}{4}\left(r_{i+1,j}^{n} + r_{i,j+1}^{n} + r_{i-1,j}^{n} + r_{i,j-1}^{n}\right) + \frac{\alpha}{2}\left[\frac{1}{2}\left(s_{i+1,j}^{n} + s_{i+1,j}^{n+1}\right) - \frac{1}{2}\left(s_{i-1,j}^{n} + s_{i-1,j}^{n+1}\right)\right] , \tag{7.22}$$

$$l_{i,j}^{n+1} = \frac{1}{4}\left(l_{i+1,j}^{n} + l_{i,j+1} + l_{i-1,j}^{n} + l_{i,j-1}^{n}\right) + \frac{\alpha}{2}\left[\frac{1}{2}\left(s_{i,j+1}^{n} + s_{i,j+1}^{n+1}\right) - \frac{1}{2}\left(s_{i,j-1}^{n} + s_{i,j-1}^{n+1}\right)\right] . \tag{7.23}$$

7.3 The Leapfrog Scheme

The 2D generalization of the one-dimensional scheme (5.16) 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.

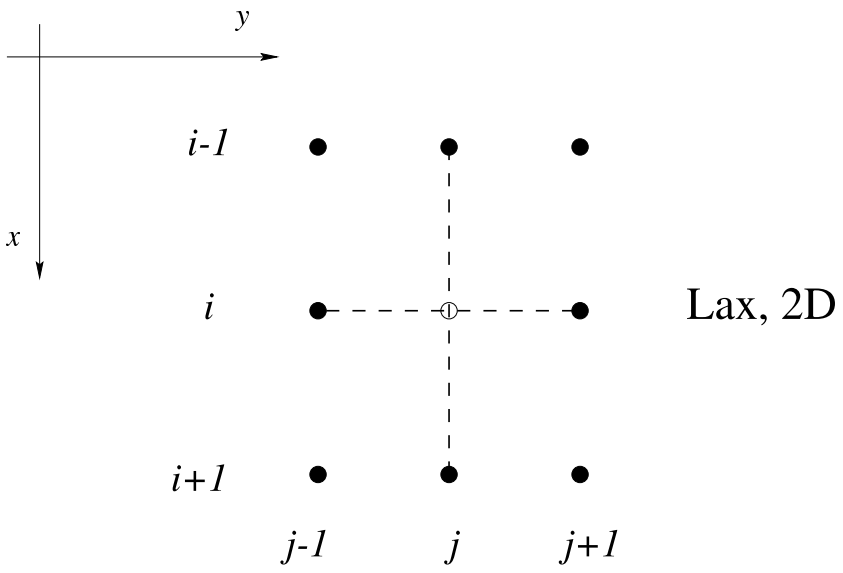

Figure 7.2: Schematic diagram of a Leapfrog evolution scheme in two dimensions. Note that the center of the cross-like stencil is used in this case both at the time-level n (filled circle) and at the time level $n + 1$ (filled square).

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.

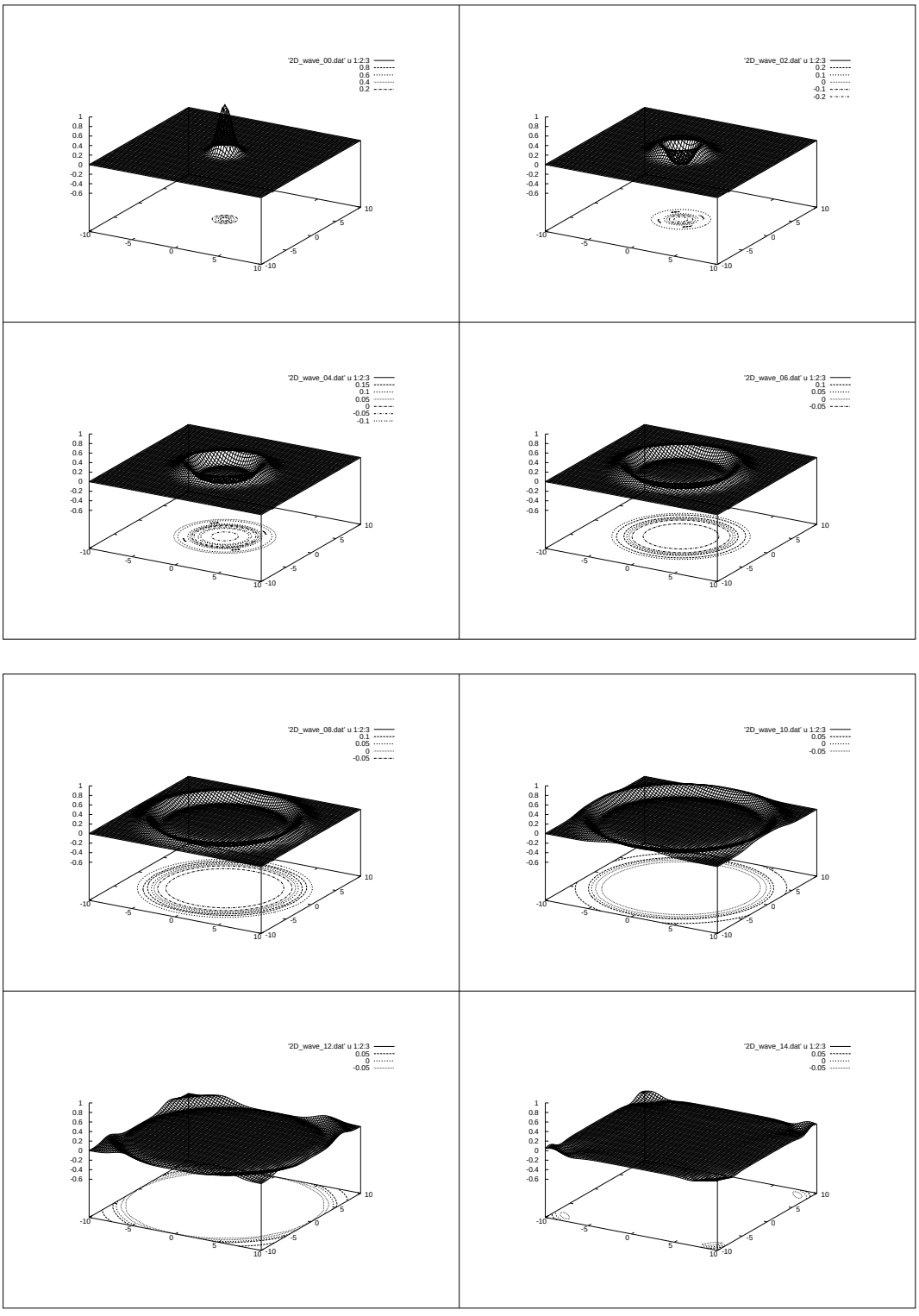

Figure 7.3: Plot of the time evolution of the wave equation when the Leapfrog scheme in 2D is used and Sommerfeld boundary conditions are imposed. Snapshots at increasing times are illustrated in a clockwise sequence.

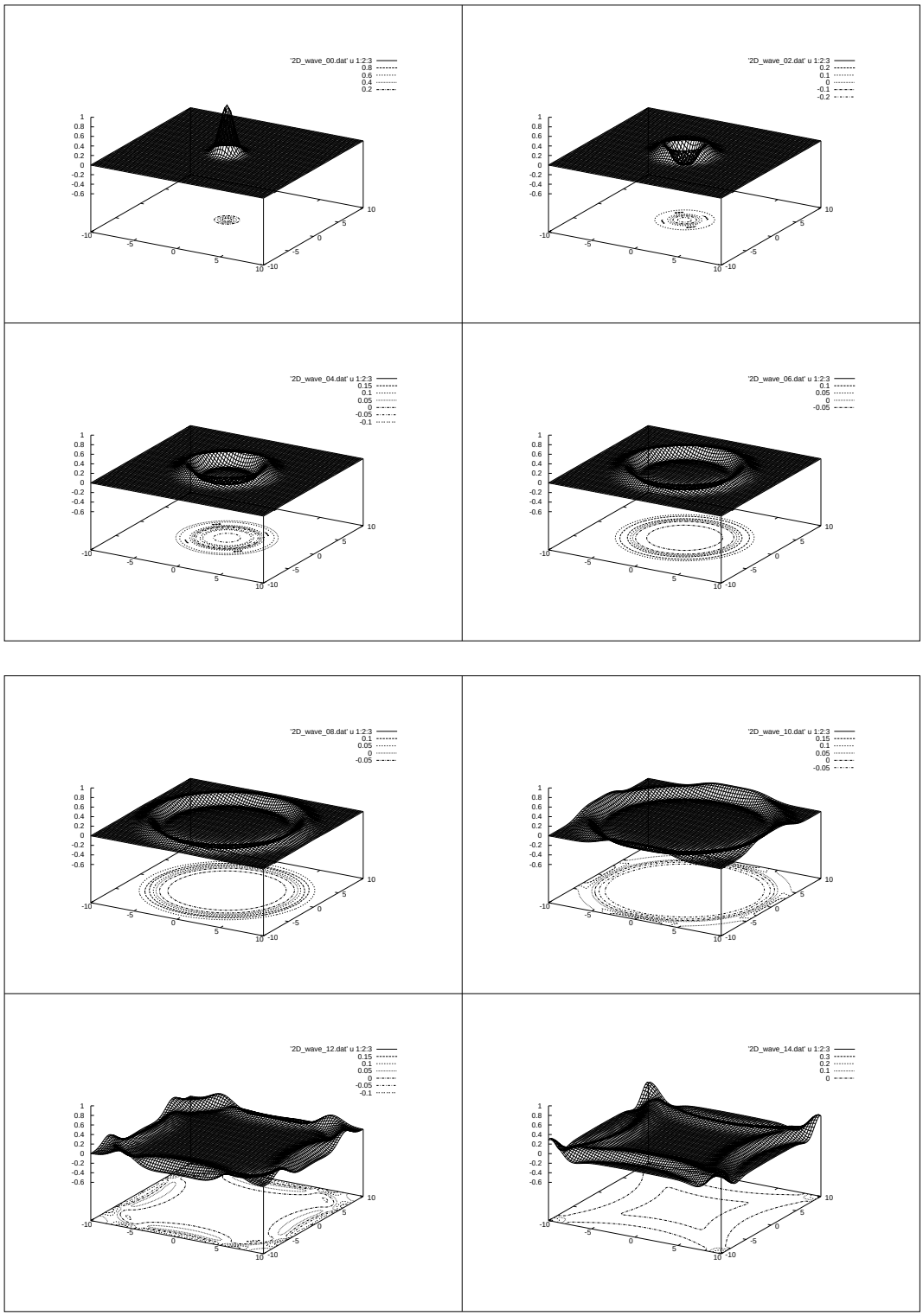

Figure 7.4: Plot of the time evolution of the wave equation when the Leapfrog scheme in 2D is used and Reflecting boundary conditions are applied. Snapshots at increasing times are illustrated in a clockwise sequence.

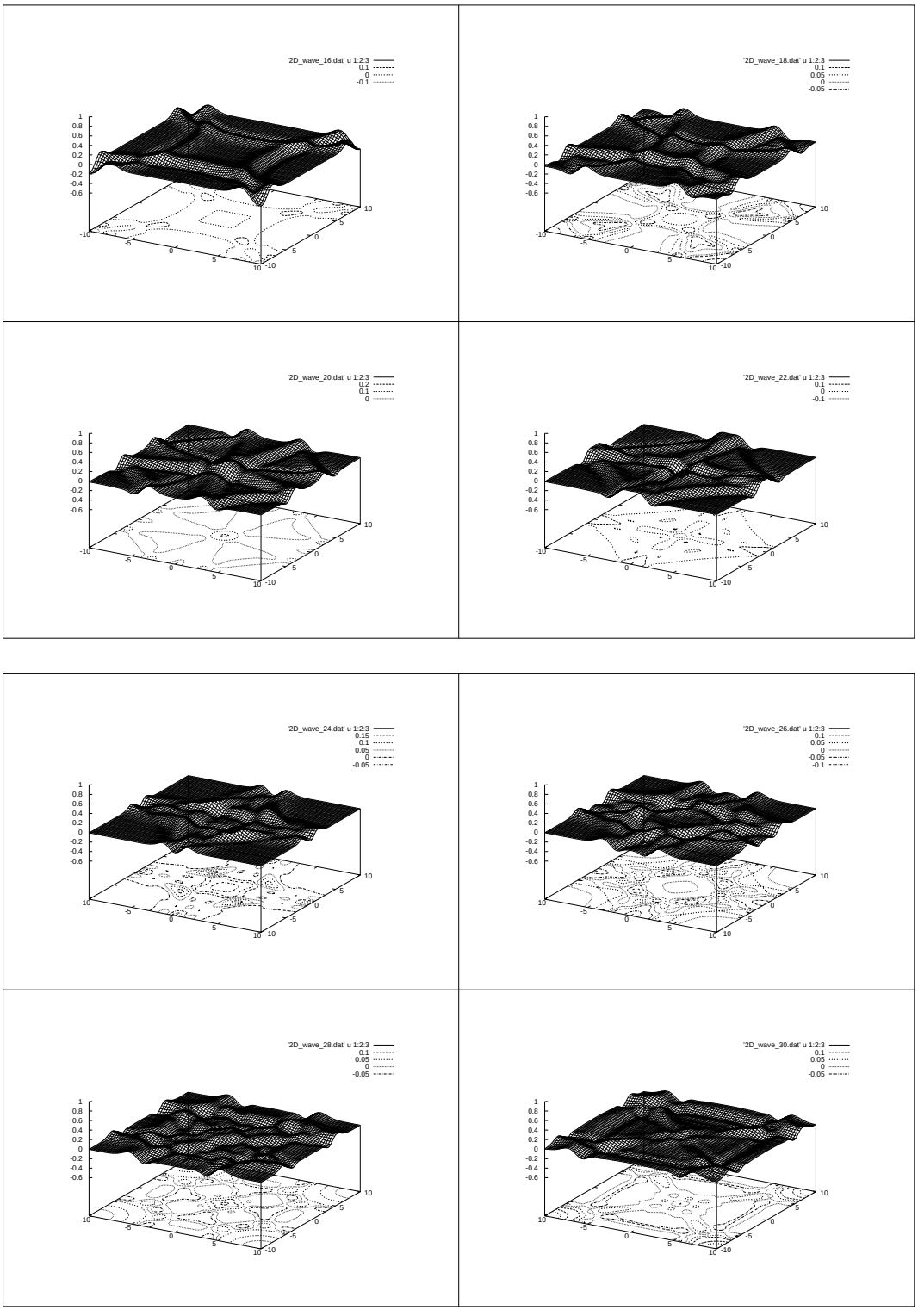

Figure 7.5: Plot of the time evolution of the wave equation when the Leapfrog scheme in 2D is used and Reflecting boundary conditions are applied.

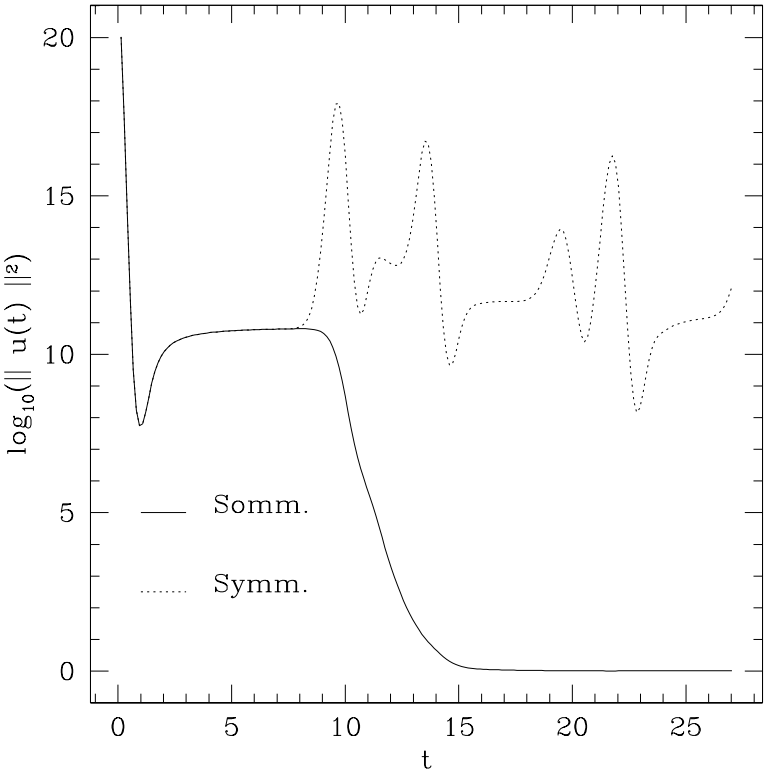

Figure 7.6: Plot of the time evolution of the 2-norm when the Leapfrog scheme in 2D is used. Note the radically different behaviour between Sommerfeld and reflecting boundary conditions.

Chapter 8

Parabolic PDEs

8.1 Diffusive problems

The inclusion of viscosity in the description of a fluid leads to non trivial complications in the numerical solution of the hydrodynamic equations. From an analytical point of view, the resulting equations are no longer purely hyperbolic PDE's but rather mixed hyperbolic-parabolic PDE's. This means that the numerical method used to solve them must necessarily be able to cope with the parabolic part of the equations. It is therefore convenient to fully understand the prototypical parabolic equation, the one-dimensional diffusion equation, both analytically and numerically, before attempting to solve any mixed hyperbolic-parabolic PDE.

8.2 The diffusion equation in 1D

The description of processes like the heat conduction in a solid body or the spread of a dye in a motionless fluid is given by the one-dimensional *diffusion equation*

$$\frac{\partial u(x,t)}{\partial t} = D\frac{\partial^2 u(x,t)}{\partial x^2} \; . \tag{8.1}$$

Here D is a constant coefficient that determines the magnitude of the "diffusion" in the process under investigation (being given by the thermal conductivity and dye diffusion coefficient respectively in the above mentioned examples).

A complete description of some particular process will clearly be possible only once the initial value (*i.e.*, $u(x,0) = h(x)$ with $x \in [0, L]$) and the boundary conditions are specified. The most common boundary conditions (BCs) are such to prescribe the value of the function $u(x,t)$ at the boundaries, $u(0,t) = u_0(t)$ and $u(L,t) = u_L(t)$, if the boundaries of the physical domain are modeled to be in the origin and at a distance L from the origin. This type of BCs are called *Dirichlet boundary conditions* (DBC).

On the other hand, it is possible that the physics of the problem requires the BCs to be specified in terms of the derivatives of $u(x,t)$. This is the case for instance when 1-D heat conduction in a bar is investigated and the boundaries of the bar are

completely insulated so that no heat flux is present outside the body. More generally, if $q(x,t) \equiv \partial u(x,t)/\partial x$, the so called *Neumann boundary conditions* (NBC) are written as $q(0,t) = q_0(t)$ and $q(L,t) = q_L(t)$. It should be noted however that Dirichlet BCs and Neumann BCs are not the only possible BCs.

In what follows, first the analytic solution to a simple diffusive problem will be given and then some numerical methods to solve it will be examined.

8.3 Explicit updating schemes

8.3.1 The FTCS method

The most straightforward way to finite difference equation (8.1) is by the FTCS method, *i.e.*,

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = D\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x^2} + \mathcal{O}(\Delta t, \Delta x^2) \ , \tag{8.2}$$

Unlike for a hyperbolic equation, where the FTCS method leads to un unconditionally unstable method, the presence of a second space derivative in the model parabolic equation (8.1) allows the FTCS method to be conditionally stable [9]. A von Neumann stability analysis leads in fact to the stability criterion

$$\gamma \equiv 2D\frac{\Delta t}{\Delta x^2} \leq 1 \ , \tag{8.3}$$

that lends itself to a physical interpretation: the maximum time step is, up to a numerical factor, the diffusion time across a cell of width Δx. This stability condition poses a serious limit in the use of the above scheme since the typical time scales of interest will require a number of timesteps which could be prohibitive in multidimensional calculations. The additional fact that the overall scheme is first-order accurate in time only strengthens the need for a different method.

8.3.2 The Du Fort-Frankel method and the θ-method

With this objective in mind, it is not difficult to think of a way to avoid the reduced accuracy due to the forward-time finite differencing approach used in FTCS. A simple time-centered finite differencing

$$\frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t} = D\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x^2} \tag{8.4}$$

should grant second-order accuracy. Unfortunately, this method is unconditionally unstable. To overcome the stability problem, Du Fort and Frankel [11] suggested the following scheme

$$\frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t} = D\frac{u_{j+1}^n - u_j^{n+1} - u_j^{n-1} + u_{j-1}^n}{\Delta x^2} \ , \tag{8.5}$$

which is obtained from (8.4) with the substitution of u_j^n with $\frac{1}{2}(u_j^{n+1} + u_j^{n-1})$, that is, by taking the average of u_j^{n+1} and u_j^{n-1}, *i.e.*,

$$u_j^{n+1} = \left(\frac{1-\gamma}{1+\gamma}\right) u_j^{n-1} + \left(\frac{\gamma}{1+\gamma}\right)\left(u_{j+1}^n + u_{j-1}^n\right) + \mathcal{O}(\Delta x^2)\ . \tag{8.6}$$

With this substitution, the method is still explicit and becomes unconditionally stable, but not without a price. A consistency analysis shows, in fact, that the Du Fort-Frankel method could be inconsistent. The local truncation error is [8]

$$\epsilon = \frac{\Delta t^2}{6}\frac{\partial^3 u}{\partial t^3}\bigg|_{j,n} - D\frac{\Delta x^2}{12}\frac{\partial^4 u}{\partial x^4}\bigg|_{j,n} + \left(\frac{\Delta t}{\Delta x}\right)^2 \frac{\partial^2 u}{\partial t^2}\bigg|_{j,n} + \ldots \tag{8.7}$$

$$= \mathcal{O}\left(\Delta t^2, \Delta x^2, \left(\frac{\Delta t}{\Delta x}\right)^2\right)\ , \tag{8.8}$$

which shows that if Δt and Δx tend to zero at the same rate, *i.e.*, $\Delta t = k\Delta x$ with k being a constant, then the truncation error does not vanish for $\Delta t \to 0$ and $\Delta x \to 0$. Indeed, the solution obtained with this method will effectively be the solution to equation

$$\frac{\partial u(x,t)}{\partial t} + k^2\frac{\partial^2 u(x,t)}{\partial t^2} = D\frac{\partial^2 u(x,t)}{\partial x^2}\ , \tag{8.9}$$

and not the solution of (8.1). On the other hand, it is also clear from (8.7) that having a timestep $\Delta t = k\Delta x^{1+\varepsilon}$ with $\varepsilon > 0$ will assure the consistency of the method. Of course, the closer is ε to 1, the smaller will have to be Δx in order to achieve consistency. Moreover, accuracy requirements pose an additional constraint on ε. For a first order-method it is necessary to have $\varepsilon = 1/2$ while to achieve second-order accuracy the requirement is $\varepsilon = 1$. It would be pointless and computationally inefficient to set $\varepsilon > 1$ since in this case the dominant contribution to the truncation error would be determined by the term $O\left(\Delta x^2\right)$ which acts as an upper limit to the overall accuracy order. This means that ε is constrained to be in the interval $1/2 \leq \varepsilon \leq 1$.

The advantages of the Du Fort-Frankel method over the FTCS scheme should now be easily seen. To achieve first-order accuracy, a timestep $\Delta t = (\Delta x)^{3/2}$ is needed with the former while the latter requires $\Delta t \approx (\Delta x)^2$. On the other hand, if a timestep $\Delta t = (\Delta x)^2$ is used the Du Fort-Frankel method gains, second-order accuracy. Finally, any desired accuracy between first and second order could be achieved with a timestep that is independent of the diffusion coefficient D. The only minor drawback of the Du Fort-Frankel scheme lies in the requirement of keeping track of an additional time level.

A generalization of the Du Fort-Frankel scheme is also straightforward. In particular, when averaging u_j^{n+1} and u_j^{n-1}, instead of weighting them equally, it is possible to average them with different weights. The resulting update scheme is therfore

$$\frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t} = D\frac{u_{j+1}^n - 2\left(\theta u_j^{n+1} - (1-\theta)u_j^{n-1}\right) + u_{j-1}^n}{\Delta x^2}\ , \tag{8.10}$$

where θ is a variable parameter. In [8] it is shown that the local truncation error for this scheme is

$$\frac{\Delta t^2}{6}\frac{\partial^3 u}{\partial t^3}\bigg|_{j,n} - D\frac{\Delta x^2}{12}\frac{\partial^4 u}{\partial x^4}\bigg|_{j,n} + (2\theta-1)\frac{2\Delta t}{\Delta x^2}\frac{\partial u}{\partial t}\bigg|_{j,n} + \tag{8.11}$$

$$\frac{\Delta t^2}{\Delta x^2}\frac{\partial^2 u}{\partial t^2}\bigg|_{j,n} + \mathcal{O}\left(\frac{\Delta t^3}{\Delta x^2}, \Delta t^4, \Delta x^4\right) , \tag{8.12}$$

which clearly shows that consistency could be achieved for any value of θ if $\Delta t = k\Delta x^{2+\varepsilon}$ with ε and k being positive real numbers. If $\theta = 1/2$, on the other hand, the scheme is actually the Du Fort-Frankel scheme [*cf.* expression (8.7)] with the consistency constraints already outlined above. It is therefore clear that, when solving equation (8.1), timestep considerations show that the only viable θ-scheme is the $\theta = 1/2$ scheme, *i.e.*, the Du Fort-Frankel scheme.

8.3.3 ICN as a θ-method

We next extend the stability analysis of the θ-ICN discussed in Sect. 3.6.1 to the a parabolic partial differential equation and use as model equation the one-dimensional diffusion equation (8.1). Parabolic equations are commonly solved using implicit methods such as the Crank-Nicolson, which is unconditionally stable and thus removes the constraints on the timestep [*i.e.*, $\Delta t \approx \mathcal{O}(\Delta x^2)$] imposed by explicit schemes [9]. In multidimensional calculations, however, or when the set of equations is of mixed hyperbolic-parabolic type, implicit schemes can be cumbersome to implement since the resulting system of algebraic equations does no longer have simple and tridiagonal matrices of coefficients. In this case, the most conveniente choice may be to use an explicit method such as the ICN.

Also in this case, the first step in our analysis is the derivation of a finite-difference representation of the right-hand-side of eq. (8.1) which, at second-order, has the form

$$\mathcal{L}_\Delta(u^n_{j,j\pm1}) = \frac{u^n_{j+1} - 2u^n_j + u^n_{j-1}}{\Delta x^2} + \mathcal{O}(\Delta x^2) . \tag{8.13}$$

Constant Arithmetic Averages

Next, we consider first the case with constant arithmetic averages (*i.e.*, $\theta = 1/2$) and the expression for the amplification factor after M-iterations is then purely real and given by

$${}^{(M)}\xi = 1 + 2\sum_{n=1}^{M}(-\gamma)^n \ , \tag{8.14}$$

where $\gamma \equiv (2D\Delta t/\Delta x^2)\sin^2(k\Delta x/2)$. Requiring now for stability that $\sqrt{\xi^2} \leq 1$ and bearing in mind that

$$-1 \leq \sum_{n=0}^{M}(-\gamma)^{n+1} \leq 0 , \qquad \text{for } \gamma \leq 1 , \tag{8.15}$$

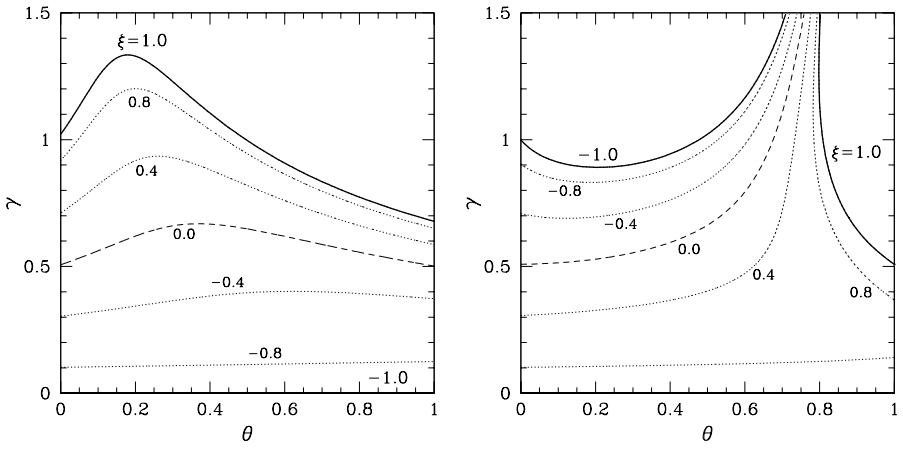

Figure 8.1: *Left panel:* stability region in the (θ, γ) plane for the two-iterations θ-ICN for the diffusion equation (8.1). Thick solid lines mark the limit at which $\xi^2 = 1$, while the dotted contours indicate the values of the amplification factor in the stable region. *Right panel:* same as in the left panel but with swapping the averages between two corrections.

we find that the scheme is stable for *any* number of iterations provided that $\gamma \leq 1$. Furthermore, because the scheme is second-order accurate from the first iteration on, our suggestion when using the ICN method for parabolic equations is that one iteration should be used *and no more*. In this case, in particular, the ICN method coincides with a FTCS scheme [9].

Note that the stability condition $\gamma \leq 1$ introduces again a constraint on the timestep that must be $\Delta t \leq \Delta x^2/(2D)$ and thus $\mathcal{O}(\Delta x^2)$. As a result and at least in this respect, the ICN method does not seem to offer any advantage over other explicit methods for the solution of a parabolic equation [1].

Constant Weighted Averages

We next consider the stability of the θ-ICN method but focus our attention on a two-iterations scheme since this is the number of iterations needed in the solution of the parabolic part in a mixed hyperbolic-parabolic equation when, for instance, operator-splitting techniques are adopted [9]. In this case, the amplification factor is again purely real and given by

$$\xi \quad = \quad 1 - 2\gamma + 4\gamma^2\theta - 8\gamma^3\theta^2 \; , \tag{8.16}$$

so that stability is achieved if

$$0 \leq \gamma \left(1 - 2\theta\gamma + 4\theta^2\gamma^2\right) \leq 1 \; . \tag{8.17}$$

[1] Note that also the Dufort-Frankel method [11], usually described as unconditionally stable, does not escape the timestep constraint $\Delta t \approx \mathcal{O}(\Delta x^2)$ when a consistent second-order accurate solution is needed [8].

Since $\gamma > 0$ by definition, the left inequality is always satisfied, while the right one is true provided that, for $\gamma < 4/3$,

$$\frac{\gamma - \sqrt{\gamma(4-3\gamma)}}{4\gamma^2} \leq \theta \leq \frac{\gamma + \sqrt{\gamma(4-3\gamma)}}{4\gamma^2} \ . \tag{8.18}$$

The stability region described by the condition (8.18) is shown in the left panel of Fig. 8.1 for $\sin k\Delta x = 1$ and illustrates that the scheme is stable for any value $0 \leq \theta \leq 1$, and also that slightly larger timesteps can be taken when $\theta \simeq 0.2$.

Swapped Weighted Averages

After some lengthy algebra the calculation of the amplification factor for the θ-ICN method with swapped weighted averages yields

$$\xi = 1 - 2\gamma + 4\gamma^2\theta - 8\gamma^3\theta(1-\theta) \ , \tag{8.19}$$

and stability is then given by

$$-1 \leq 1 - 2\gamma + 4\gamma^2\theta - 8\gamma^3\theta(1-\theta) \leq 1 \ . \tag{8.20}$$

Note that none of the two inequalities is always true and in order to obtain analytical expressions for the stable region we solve the condition (8.20) with respect to θ and obtain

$$\theta \leq \frac{2\gamma - 1 + \sqrt{4\gamma^2 - 4\gamma + 5}}{4\gamma} \ , \tag{8.21a}$$

$$\theta \leq \frac{\gamma(2\gamma-1) - \sqrt{\gamma\,(4\gamma^3 - 4\gamma^2 + 5\gamma - 4)}}{4\gamma^2} \ , \tag{8.21b}$$

$$\theta \geq \frac{\gamma(2\gamma-1) + \sqrt{\gamma\,(4\gamma^3 - 4\gamma^2 + 5\gamma - 4)}}{4\gamma^2} \ . \tag{8.21c}$$

The resulting stable region for $\sin k\Delta x = 1$ is plotted in the right panel of Fig. 8.1 and seems to suggest that arbitrarily large values of γ could be considered when $\theta \gtrsim 0.6$ It should be noted, however, that the amplification factor is also severely reduced as larger values of γ are used and indeed it is essentially zero in the limit $\theta \to 1$.

8.4 Implicit updating schemes

8.4.1 The BTCS method

It is common for explicit schemes to be only conditionally stable and in this respect the Du Fort-Frankel method is somewhat unusual. Implicit methods, on the other hand, do not share this property being typically unconditionally stable. This suggests to apply an implicit finite differencing to equation (8.1) in the form of a "backward-time centered-space" (BTCS) scheme and obtain

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = D\frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{\Delta x^2} + \mathcal{O}(\Delta t, \Delta x^2) \ . \tag{8.22}$$

As a von Neumann stability analysis shows [9], the differencing (8.22) is unconditionally stable. This method is also called *backward time*. Rearranging the terms it is easy to obtain

$$-\gamma u_{j-1}^{n+1} + 2(1+\gamma)u_j^{n+1} - \gamma u_{j+1}^{n+1} = 2u_j^n \ , \tag{8.23}$$

which shows that to obtain u at time level $n+1$ is necessary to solve a system of linear equations. Luckily, the system is *tridiagonal*, *i.e.*, only the nearest neighbors of the diagonal term are non zero, which allows the use of *sparse matrix* techniques (a matrix is called sparse if the number of non zero elements is small compared to the number of all the elements). The main disadvantage of this scheme, besides that of requiring the simultaneous solution of N algebraic equations, is that it is only first-order accurate in time.

8.4.2 The Crank-Nicolson method

Combining the stability of an implicit method with the accuracy of a method that is second-order both in space and in time is possible and is achieved by averaging explicit FTCS and implicit BTCS schemes:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \frac{D}{2}\left[\frac{(u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}) + (u_{j+1}^n - 2u_j^n + u_{j-1}^n)}{\Delta x^2}\right] + \mathcal{O}(\Delta t^2, \Delta x^2) \ . \tag{8.24}$$

This scheme is called *Crank-Nicolson* and is second-order in time since both the left hand side and the right hand side are centered in $n + 1/2$. As the fully implicit scheme, the CN scheme is unconditionally stable and is the best choice for the solution of simple one dimensional diffusive problems.

The disadvantage of this scheme with respect to an explicit scheme like the Du Fort-Frankel scheme lies in the fact that in more than one dimension the system of linear equation will no longer be tridiagonal, although it will still be sparse. The extension of the Du Fort-Frankel scheme, on the other hand, is straightforward and with the same constraints as in the one dimensional case. Because of this and other problems which emerge in multidimensional applications, more powerful methods, like the *Alternating Direction Implicit* (ADI) have been developed. ADI embodies the powerful concept of *operator splitting* or *time splitting*, which requires a more detailed explanation and will not be given in these notes.

Appendix A

Semi-analytical solution of the model parabolic equation

In this appendix we present details on the derivation of the semi-analytic solution to equation

$$\frac{\partial u(x,t)}{\partial t} = D\frac{\partial^2 u(x,t)}{\partial x^2} \ , \tag{A.1}$$

where D is a constant coefficient. We will first consider homogeneous Dirichlet and then homogeneous Neumann boundary conditions. Because the initial value $u(x,0) = h(x)$ is also needed, we will consider two different initial profiles for the two cases. The solutions we will obtain are to be considered semi-analytical in the sense that it is usually necessary to evaluate them numerically. This is so because infinite series and integrals that could not always be evaluated analytically are involved.

A.1 Homogeneous Dirichlet boundary conditions

Consider a generic problem for which equation (8.1) holds over a domain $[0, L]$. Suppose also that the boundary conditions could be written as *homogeneous* DBC, *i.e.*, $u(0,t) = u(L,t) = 0$, and that at time $t_0 = 0$ the distribution of $u(x,t)$ is that shown in Figure A.1, which could be written as

$$h(x) \equiv u(x,0) = \begin{cases} 2x/L & \text{if} \quad 0 \leq x \leq L/2 \\ \\ -2x/L + 2 & \text{if} \quad L/2 < x \leq L \end{cases} \tag{A.2}$$

while the boundary conditions are $u(0,t) = u(L,t) = 0$.

The equation could be solved by means of the separation of variables technique, *i.e.*, by searching for a solution of the form

$$u(x,t) = f(x)g(t) \ , \tag{A.3}$$

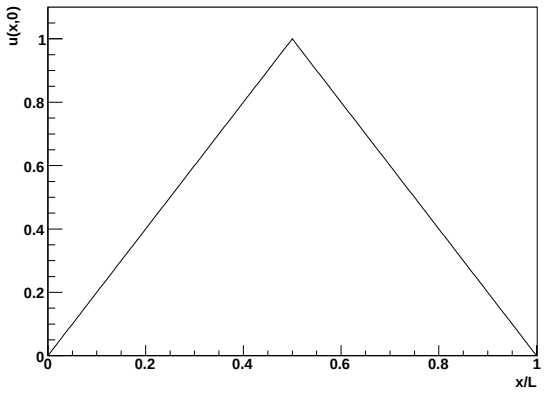

Figure A.1: Initial value for the diffusive problem (8.1).

which allows to write equation (A.1) as

$$f\frac{\partial g}{\partial t} = Dg\frac{\partial^2 f}{\partial x^2} \ . \tag{A.4}$$

Multiplying both sides by $1/(fg)$ the result is

$$\frac{1}{g}\frac{\partial g}{\partial t} = D\frac{1}{f}\frac{\partial^2 f}{\partial x^2} \ . \tag{A.5}$$

The left hand side of (A.5) is a function of t only while the right hand side depends only on x. Because of that, their common value can only be a constant, with this constant being a negative number because otherwise $g \to \infty$ (and therefore $u \to \infty$) as $t \to \infty$. Thus the common value could be denoted as $-\lambda$ with $\lambda > 0$ and so (A.5) becomes

$$\frac{1}{g}\frac{\partial g}{\partial t} = -\lambda = D\frac{1}{f}\frac{\partial^2 f}{\partial x^2} \ . \tag{A.6}$$

Recaling that the initial condition has been written as $h(x)$ it is possible to write the solution as

$$u(x,t) = h(x)e^{-\lambda t} \ , \tag{A.7}$$

with the requirement that

$$-D\frac{\partial^2 f}{\partial x^2} = \lambda f \ . \tag{A.8}$$

The problem (A.8) is an *eigenvalue problem* for the differential operator $-D\,\partial^2/\partial x^2$ with *eigenvalue* λ and *eigenfunction* $f(x)$. The eigenfunctions and eigenvalues will be determined imposing the boundary conditions.

The general solution to (A.8) can be written as

$$f(x) = Ae^{-\mathrm{i}kx} + Be^{\mathrm{i}kx} \ , \tag{A.9}$$

with $k \equiv \sqrt{\lambda/D}$, A and B are constants to be determined through the boundary conditions. Requiring that $f(0) = 0$ it is easily found that $B = -A$ and thus

$$f(x) = A\left(e^{-\mathrm{i}kx} - e^{\mathrm{i}kx}\right) = -2iA\sin kx\ . \tag{A.10}$$

The second boundary condition $f(L) = 0$ allows to find the eigenvalues and the eignenfunctions (and the trivial solution $f(x) = 0$ as well). In fact $\sin(kL) = 0$ as soon as

$$kL = \sqrt{\frac{\lambda}{D}} = m\pi\ , \quad m = 0, \pm1, \pm2, \pm3, \ldots \tag{A.11}$$

so that the eigenvalues and the eigenfunctions are

$$\lambda_m = D\left(\frac{m\pi}{L}\right)^2\ , \qquad f_m(x) = \sin\left(\frac{m\pi}{L}x\right)\ . \tag{A.12}$$

The solution to (A.8) will therefore be a linear superposition of the eigenfunctions $f_m(x)$,

$$u(x,t) = \sum_{m=1}^{\infty} a_m \sin\left(\frac{m\pi}{L}x\right)\exp\left[D\left(\frac{m\pi}{L}\right)^2 t\right]\ . \tag{A.13}$$

One last condition is still not satisfied, the initial value condition. And is exactly this condition that allows to find the coefficients a_m such that

$$u(x,0) = \sum_{m=1}^{\infty} a_m \sin\left(\frac{m\pi}{L}x\right) = h(x)\ . \tag{A.14}$$

This is a Fourier series on the interval $[0, L]$ of the initial value $h(x)$ and its coefficients may easily be evaluated keeping in mind the orthogonality property of the eigenfunctions. It is not difficult to show that

$$\int_0^L \sin\left(\frac{m\pi}{L}x\right)\sin\left(\frac{k\pi}{L}x\right)\mathrm{d}x = \begin{cases} 0 & \text{if} \quad k \neq m,\ k = m = 0\ , \\ L/2 & \text{if} \quad k = m\ , \end{cases} \tag{A.15}$$

which allows to compute the coefficients a_m as

$$a_m = \frac{2}{L}\int_0^L h(x)\sin\left(\frac{m\pi}{L}x\right)\mathrm{d}x\ . \tag{A.16}$$

With $h(x)$ as defined in (A.2), the above computation leads to the final solution which therefore is

$$u(x,t) = \sum_{m=1}^{\infty} a_m \sin\left(\frac{m\pi}{L}x\right)\exp\left[-D\left(\frac{m\pi}{L}\right)^2 t\right], \qquad a_m = 8\frac{\sin(m\pi/2)}{m^2\pi^2}\ . \tag{A.17}$$

A.2 Homogeneous Neumann boundary conditions

Once equation (A.1) has been solved for homogenous Dirichlet boundary conditions it is straightforward to solve it with homogeneous Neumann boundary conditions. In fact, the same procedure could be carried over to yield the correct solution.

Once again, let the mathematical domain be $x \in [0, L]$ for $t > 0$ and if $q(x,t) \equiv \partial u/\partial x$ the homogeneous Neumann boundary conditions are written as $q(0,t) = q(L,t) = 0$. Since the boundary conditions require the derivative to vanish, the initial condition is chosen so that this condition is satisfied at $t = 0$ as well. The initial condition will then be

$$h(x) \equiv u(x,0) = 1 + 2\left(\frac{x}{L}\right)^3 - 3\left(\frac{x}{L}\right)^2 . \tag{A.18}$$

Everything that has been said in the previous case up to (A.9) still holds. The boundary conditions now require that

$$f'(x) \equiv \frac{\mathrm{d}f}{\mathrm{d}x} = \mathrm{i}k\left(Ae^{\mathrm{i}kx} - Be^{-ikx}\ ,\right) \tag{A.19}$$

vanishes at the boundaries of the domain. From $f'(0) = 0$ follows that $A = B$ while $f'(L) = 0$ leads to the same eigenvalue $\lambda_m = D\left(m\pi/L\right)^2$ as in the previous case. The eigenfunction on the other hand changes since the general solution could be now written as

$$f(x) = A\left(e^{\imath kx} + e^{-\imath kx}\right) = 2A\cos\left(kx\right) \tag{A.20}$$

so that the eigenvalue and the eigenfunction in this case are

$$\lambda_m = D\left(\frac{m\pi}{L}\right)^2, \qquad f_m(x) = \cos\left(\frac{m\pi}{L}x\right) . \tag{A.21}$$

To satisfy the initial condition it is necessary that

$$u(x,0) = \sum_{m=0}^{\infty} a_m \cos\left(\frac{m\pi}{L}x\right) = h(x) \tag{A.22}$$

where the sum now extends from 0 to ∞. This is because the orthogonality property of the eigenfunctions, which still holds and could once again be used to compute the coefficients a_m, now reads

$$\int_0^L \cos\left(\frac{m\pi}{L}x\right)\cos\left(\frac{k\pi}{L}x\right)\mathrm{d}x = \tag{A.23}$$

Because of this, the initial condition could be written as

$$h(x) = 1+2\left(\frac{x}{L}\right)^3 - 3\left(\frac{x}{L}\right)^2 = \frac{1}{2} + \sum_{m=1}^{\infty} a_m \cos\left(\frac{m\pi}{L}x\right) , \quad a_m = 24\frac{1-\cos\left(m\pi\right)}{m^4\pi^4} , \tag{A.24}$$

so that the complete solution is

$$u(x,t) = \frac{1}{2} + \sum_{m=1}^{\infty} a_m \cos\left(\frac{m\pi}{L}x\right)\exp\left[-D\left(\frac{m\pi}{L}\right)^2 t\right], \qquad a_m = 24\frac{1-\cos\left(m\pi\right)}{m^4\pi^4} . \tag{A.25}$$

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

[6] E. C. Zachmanoglou and D. W. Thoe. *Introduction to Partial Differential Equations with Applications*. Dover Publications, Inc, 1986.

[7] A. Iserles. *A First Course in the Numerical Analysis of Differential Equations*. Cambridge University Press, 1996.

[8] G. D. Smith. *Numerical Solution of Partial Differential Equations: Finite Difference Methods*. Oxford Univesity Press, third edition, 1986.

[9] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in Fortran 77 - The Art of Scientific Computing*, volume One. Cambridge Univesity Press, second edition, 1997.

[10] R. D. Richtmyer and K. W. Morton. *Difference Methods for Initial-Value Problems*. Interscience - a division of John Wiley & Sons, second edition, 1967.

[11] E. C. Du Fort and S. P. Frankel. Stability conditions in the numerical treatment of parabolic differential equations. *Mathematical Tables and Other Aids to Computation*, 7(43):135–152, July 1953.

[12] R. J. LeVeque. *Finite Difference Methods for Differential Equations - Lecture Notes*. URL = ftp://amath.washington.edu/pub/rjl/papers/amath58X.ps.gz.

[13] S. A. Teukolsky. Stability of the iterated Crank-Nicolson method in numerical relativity. *Physical Review D*, 61(087501), 2000.

[14] J. Crank and P. Nicolson. A practical method for the numerical evaluation of solutions of partial differential equations of the heat-conduction type. *Proc. Camb. Philos. Soc.*, 43:50–67, 1947.

[15] G. J. Barclay, D. F. Griffiths, and D. J. Higham. Theta method dynamics. *LMS Journal of Computation and Mathematics*, 3:27–43, 2000.

[16] A. M. Stuart and A. T. Peplow. The dynamics of the theta method. *SIAM Journal on Scientific and Statistical Computing*, 12(6):1351–1372, 1991.

[17] R. J. LeVeque. *Numerical Methods for Conservation Laws*. Birkhäuser-Verlag, Basel, Switzerland, 1992.