Study Notes on Numerical Solutions of the Wave Equation with the Finite Difference Method[1]

Artur B. Adib[2]
Advisor: Carlos A. S. de Almeida[3]

Departamento de Física, Universidade Federal do Ceará, Brazil

[1]Study supported by the PIBIC/CNPq undergraduate research program.
[2]e-mail: adib@fisica.ufc.br
[3]e-mail: carlos@fisica.ufc.br

Preface

The simulation of the dynamics of classical field theories is currently gaining some attention from the high-energy community, mainly in the context of statistical field theory. Recent papers show that, in some particular but important conditions, *classical* field theories are a very good approximation to the *quantum* evolution of fields at finite temperature (see, for instance, [1, 2]). Also, in the context of condensed-matter systems, the dynamics of effective classical fields has proven to be a very efficient tool in describing both the equilibrium and non-equilibrium properties of systems such as ferromagnets [3]. In order to simulate these theories, they must be first discretized and cast on a lattice, a job far from trivial to do in a consistent manner [4]. After the discretization, the differential equations of motion transform themselves into *finite difference* equations. Before doing any sort of useful calculation in the physical context above, one is supposed to understand the basic foundations of the numerical method, the study of which is the objective of this monograph.

Following this motivation, I will present, in an introductory way, the Finite Difference method for hyperbolic equations, focusing on a method which has second order precision both in time and space (the so-called *leap-frog* method) and applying it to the case of the 1d and 2d wave equation. A brief derivation of the energy and equation of motion of a wave is done before the numerical part in order to make the transition from the continuum to the lattice clearer.

To illustrate the extension of the method to more complex equations, I also add dissipative terms of the kind $-\eta\dot{u}$ into the equations. I also briefly discuss the *von Neumann* numerical stability analysis and the *Courant* criterion, two of the most popular in the literature. In the end I present some numerical results obtained with the leap-frog algorithm, illustrating the importance of the lattice resolution through energy plots.

I have tried to collect, in a concise way, the main steps necessary to have a stable algorithm to solve wave-like equations. More sophisticated versions of these equations should be handled with care, and accompanied of a rigorous study of convergence and stability which could be found in the references cited in the end of this work.

Contents

Chapter 1

The Wave Equation

1.1 Introduction

Partial Differential Equations (from now on simply *PDEs*) are divided in the literature basically in three kinds: parabolic, elliptic and hyperbolic (the criterion of classification of these equations can be found in [5, Chap. 8]). In this work we will be interested mainly on hyperbolic equations, of which the wave equation is the paradigm:

$$\nabla^2 u = \frac{1}{v^2}\frac{\partial^2 u}{\partial t^2}, \tag{1.1}$$

where $v^2 = \frac{\tau}{\lambda}$ is the square of the wave velocity in the medium, which in the case of a free string could be determined by the tension τ and the mass density per length unit λ.

From the strict numerical point of view, the distinction between these classes of PDEs isn't of much importance [6]. There is, however, another sort of classification of PDEs which is relevant for numerical purposes: the *initial value problems* (which include the case of the hyperbolic equations) and the *boundary condition problems* (which include, for instance, parabolic equations). In this work we will restrict ourselves to initial value problems. See reference [6] for a good introduction to boundary condition problems.

In the equation (1.1) we could still add a dissipative term proportional to the first power of the time derivative of u, i.e.,

$$\tau\nabla^2 u = \lambda\frac{\partial^2 u}{\partial t^2} + \eta\frac{\partial u}{\partial t}, \tag{1.2}$$

where η is the viscosity coefficient.

Our first step will be to derive the wave equation from a simple mechanical analysis of the free rope. Being this a well known problem in classical mechanics, we will go through only the main steps of it (for a more complete treatment of the problem of the free string, see, for example, [7, Chaps. 8 and 9]). Once we are done with the 1-d wave equation, we will proceed further to the 2-d case, which isn't as abundant in the literature as the 1-d case.

1.2 Waves in 1-dimension (the free string)

1.2.1 Equation of Motion

Figure 1.1 gives us an idea of a mass element dm with linear dimension dx subject to tension forces. We are interested on the vertical displacement of this mass element, so, for this direction, we could write the resulting force:

$$dF_u = \vec{\tau}\cdot\hat{u}\big|_{x+dx} - \vec{\tau}\cdot\hat{u}\big|_x\,, \tag{1.3}$$

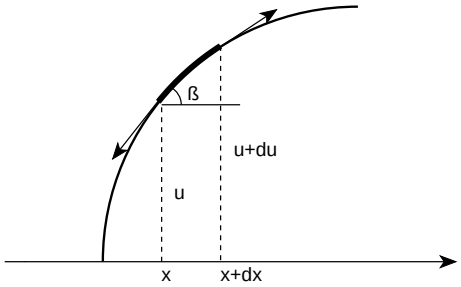

Figure 1.1: Representing the tension forces acting on an infinitesimal element of the rope.

where $\vec{\tau}$ is the tension and the unit vector $\hat{u}$ refers to the vertical direction. Within the domain of smooth deformations of the string (i.e., small β), we could write:

$$\tau_u \equiv \vec{\tau} \cdot \hat{u} = \tau \sin\beta \approx \tau \tan\beta = \tau \frac{\partial u}{\partial x} \tag{1.4}$$

We notice now that (1.3) could be written as:

$$dF_u = \frac{\tau_u|_{x+dx} - \tau_u|_x}{dx} dx = \frac{\partial \tau_u}{\partial x} dx = \frac{\partial}{\partial x}\left(\tau \frac{\partial u}{\partial x}\right) dx$$

We will now restrict ourselves to the case of constant tensions along the rope, so that:

$$dF_u = \tau \frac{\partial^2 u}{\partial x^2} dx \tag{1.5}$$

Equating this with Newton's second law

$$dF_u = dm \frac{\partial^2 u}{\partial t^2} = \lambda \frac{\partial^2 u}{\partial t^2} dx,$$

where λ is the linear mass density, we obtain then the wave equation for a free string:

$$\tau \frac{\partial^2 u}{\partial x^2} = \lambda \frac{\partial^2 u}{\partial t^2} \tag{1.6}$$

1.2.2 Energy

The kinetic energy could be evaluated in a straightforward manner, integrating the kinetic energy term for a representative mass element:

$$dT = \frac{1}{2} dm \cdot v_u^2,$$

with $dm = \lambda \cdot dx$ and $v_u = \frac{\partial u}{\partial t}$, in other words,

$$T = \int dT = \frac{\lambda}{2} \int_0^l \left(\frac{\partial u}{\partial t}\right)^2 dx \tag{1.7}$$

The potential energy can be obtained by calculating the work necessary to bring the string from a "trivial" configuration $u(x,0) = 0$ to the configuration at which we want to evaluate the potential energy $u(x,t)$. We will fix the boundary conditions $u(0,t) = u(l,t) = 0$ (string tied at the ends) and as a consequence of this, $\partial_t u(0,t) = \partial_t u(l,t) = 0$. The potential energy relative to the work necessary to change of δu the configuration of an element of the string in an interval of time dt is:

$$\delta V = -dF_u \cdot \delta u = -dF_u \cdot \left(\frac{\partial u}{\partial t}\right) dt$$

Therefore, the potential energy of the *whole* string in this *same interval of time* is:

$$dV = -dt \cdot \int dF_u \left(\frac{\partial u}{\partial t}\right)$$

Substituting (1.5) in the latter we get:

$$dV = -dt \cdot \int_0^l \tau \left(\frac{\partial^2 u}{\partial x^2}\right)\left(\frac{\partial u}{\partial t}\right) dx$$

We are however interested on $V\left[u(x,t)\right]$, so integrating in time and using the boundary conditions above, we have:

$$\begin{aligned}
V &= \int dV = -\tau \int_0^t dt \int_0^l dx \left(\frac{\partial u}{\partial t}\right)\left(\frac{\partial^2 u}{\partial x^2}\right) = \\
&= -\tau \int_0^t dt \left\{ \frac{\partial u}{\partial t}\frac{\partial u}{\partial x}\bigg|_0^l - \int_0^l \frac{\partial u}{\partial x}\frac{\partial^2 u}{\partial x \partial t} dx \right\} = \\
&= \tau \int_0^t dt \int_0^l \frac{\partial u}{\partial x}\frac{\partial^2 u}{\partial x \partial t} dx = \\
&= \tau \int_0^t dt \frac{1}{2}\frac{\partial}{\partial t}\int_0^l \left(\frac{\partial u}{\partial x}\right)^2 dx = \\
&= \frac{\tau}{2}\int_0^l \left(\frac{\partial u}{\partial x}\right)^2 dx\bigg|_0^t \Rightarrow
\end{aligned}$$

$$V = \frac{\tau}{2}\int_0^l \left(\frac{\partial u}{\partial x}\right)^2 dx \tag{1.8}$$

With (1.7) and (1.8) we have finally the total energy of the rope:

$$E = \frac{\lambda}{2}\int_0^l \left(\frac{\partial u}{\partial t}\right)^2 dx + \frac{\tau}{2}\int_0^l \left(\frac{\partial u}{\partial x}\right)^2 dx \tag{1.9}$$

In our applications we will take $\lambda = \tau$ such that $v^2 = 1$ and (1.9) assumes the simple form:

$$E = \frac{1}{2}\int_0^l \left[\left(\frac{\partial u}{\partial t}\right)^2 + \left(\frac{\partial u}{\partial x}\right)^2\right] dx \tag{1.10}$$

This energy equation will be very useful to test our algorithms through an analysis of conservation (or dissipation) during the dynamical evolution of the system.

1.3 Waves in 2-dimensions (the free membrane)

1.3.1 Equation of Motion

In its two dimensional version, the wave equation could be describing a membrane, a liquid surface, or some "coarse-grained" field in the surface physics, to cite a few. In the case of the membrane or other elastic surface, the oscillations are also constrained to be small (analogously to the 1-d string).

We notice now that the additional dimension forces us to define the tension "per unit length":

$$f = \frac{\tau}{l} \tag{1.11}$$

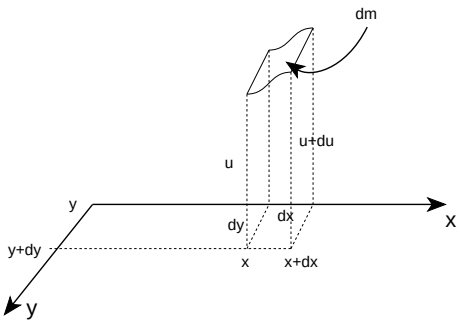

Figure 1.2: Representing a mass element dm of dimensions $dxdy$. The element is subject to tension forces on each side (analogous to the borders of the mass element in the 1-d case), being these forces orthogonal to the axes of the sides. Only the variation with respect to x of u is drawn (i.e., u and $u + du$ in the figure are displacements of $u(x, y)$ keeping y constant and varying x).

This force per unit length could be understood with a simple example: stretch a tape of width l from its extremities with force τ. We can't ask the force on a *point* of the tape, but only on some element of some definite length and width (of course, this element could be differential, playing the same role of a linear differential element in the case of the string). Formula (1.11), times the length of the element, then gives you the resulting force (tension) on the element. In this way, we extend the equation (1.3) to two dimensions:

$$dF_u = \left[\vec{f}_x \cdot \hat{u}\Big|_{x+dx,y} dy - \vec{f}_x \cdot \hat{u}\Big|_{x,y} dy \right] + \left[\vec{f}_y \cdot \hat{u}\Big|_{x,y+dy} dx - \vec{f}_y \cdot \hat{u}\Big|_{x,y} dx \right], \tag{1.12}$$

where $\vec{f}_x \cdot \hat{u}\Big|_{x,y} dy \equiv f_{x,u}|_{x,y}\, dy$ is the x component of the tension in the direction $\hat{u}$ acting on the side defined by the points (x, y) e $(x, y + dy)$, and so on. We are going to suppose that the forces on the sides of the elements are orthogonal to their axes, which is the same as decomposing the tension force on dm into four components, one for each side (notice, however, that we have effectively only *two* resulting components, to wit $\hat{x}$ and $\hat{y}$). Doing this we won't need to emphasize the tension components along x or y, $\vec{f}_x \cdot \hat{u}\Big|_{x,y}$ becoming simply $f_u|_{x,y}$ and so on. Nevertheless it *is* still important, for what we said above, to know *what* side we are talking about. So, in the regime of small vibrations (i.e., small angles of deformation), we could find f_u analogously to the string case

$$f_u dy = f \frac{\partial u}{\partial x} dy \tag{1.13}$$

$$f_u dx = f \frac{\partial u}{\partial y} dx, \tag{1.14}$$

where $f_u dy$ and $f_u dx$ are the tensions in the direction $\hat{u}$ on a side of length dy and dx along y and x, respectively. We emphasize that, with this notation plus the knowledge of the *point* where we are going to evaluate the derivatives, we have a complete specification of the side on which the tension acts[1]. With all this in hands, Eq. (1.12) becomes:

$$dF_u = \left[f_u|_{x+dx,y}\, dy - f_u|_{x,y}\, dy \right] + \left[f_u|_{x,y+dy}\, dx - f_u|_{x,y}\, dx \right]$$

With calculations analogous to those of the previous section, we have:

$$dF_u = \frac{f_u|_{x+dx,y} - f_u|_{x,y}}{dx} dxdy + \frac{f_u|_{x,y+dy} - f_u|_{x,y}}{dy} dydx = \tag{1.15}$$

[1] Indeed, once specified the *beginning* of the side with the pair (x, y), specifying the length with dx or dy furnishes us with the *direction* of the side in question. This is sufficient to localize it, since the z coordinate is unambiguously determined via $z = u(x, y)$.

$$= \frac{\partial f_u}{\partial x}dxdy + \frac{\partial f_u}{\partial y}dxdy = \tag{1.16}$$

$$= f\frac{\partial^2 u}{\partial x^2}dxdy + f\frac{\partial^2 u}{\partial y^2}dxdy \tag{1.17}$$

With Newton's second law we obtain:

$$\begin{aligned} f\frac{\partial^2 u}{\partial x^2}dxdy + f\frac{\partial^2 u}{\partial y^2}dxdy &= dm\frac{\partial^2 u}{\partial t^2} \Rightarrow \\ f\frac{\partial^2 u}{\partial x^2}dxdy + f\frac{\partial^2 u}{\partial y^2}dxdy &= \sigma dxdy\frac{\partial^2 u}{\partial t^2} \Rightarrow \\ f\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) &= \sigma\frac{\partial^2 u}{\partial t^2} \Rightarrow \end{aligned}$$

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \frac{1}{v^2}\frac{\partial^2 u}{\partial t^2}, \tag{1.18}$$

which is the desired wave equation for two dimensions, with $v^2 = \frac{f}{\sigma}$ and σ the surface mass density.

1.3.2 Energy

The derivation of the total energy is done in the same manner as the 1d case. We will consider a surface $z = u(x, y, t)$ with support of dimension $l \times l$, subject to the boundary conditions $u|_{boundary} \equiv u(0, y, t) = u(l, y, t) = u(x, 0, t) = u(x, l, t) = 0$ and $\dot{u}|_{boundary} \equiv \dot{u}(0, y, t) = \dot{u}(l, y, t) = \dot{u}(x, 0, t) = \dot{u}(x, l, t) = 0$, where $\dot{u} \equiv \partial u/\partial t$. Let us begin with the kinetic term:

$$dT = \frac{1}{2}dm \cdot \dot{u}^2 = \frac{\sigma}{2}\dot{u}^2 dxdy \Rightarrow \tag{1.19}$$

$$T = \frac{\sigma}{2}\int_0^l \int_0^l \left(\frac{\partial u}{\partial t}\right)^2 dxdy, \tag{1.20}$$

where σ is the surface mass density.

The potential energy is obtained in an analogous way to the Section (1.2):

$$\begin{aligned} dV &= -dt \cdot \int dF_u \left(\frac{\partial u}{\partial t}\right) = \\ &= -dt \cdot \int_0^l \int_0^l f\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right)\left(\frac{\partial u}{\partial t}\right) dxdy \Rightarrow \\ V &= -f\int_0^t dt\left\{\int_0^l dy\int_0^l dx\left(\frac{\partial^2 u}{\partial x^2}\frac{\partial u}{\partial t}\right) + \int_0^l dx\int_0^l dy\left(\frac{\partial^2 u}{\partial y^2}\frac{\partial u}{\partial t}\right)\right\} = \\ &= f\int_0^t dt\left\{\int_0^l dy\frac{1}{2}\frac{\partial}{\partial t}\int_0^l \left(\frac{\partial u}{\partial x}\right)^2 dx + \int_0^l dx\frac{1}{2}\frac{\partial}{\partial t}\int_0^l \left(\frac{\partial u}{\partial y}\right)^2 dy\right\} = \\ &= \frac{f}{2}\int_0^t dt\frac{\partial}{\partial t}\left\{\int_0^l \int_0^l \left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2\right] dxdy\right\} = \\ &= \frac{f}{2}\int_0^l \int_0^l \left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2\right] dxdy\Bigg|_0^t \Rightarrow \end{aligned}$$

$$V = \frac{f}{2}\int_0^l \int_0^l \left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2\right] dxdy \tag{1.21}$$

Then the total energy for our usual condition $f = \sigma \Rightarrow v^2 = 1$ is:

$$E = \frac{1}{2}\int_0^l \int_0^l \left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial u}{\partial t}\right)^2 \right] dxdy$$

or, in a more general way (we are going to take as granted the result for more than two dimensions),

$$E = \int d^n\vec{r} \left[\frac{1}{2}(\vec{\nabla} u)^2 + \frac{1}{2}\dot{u}^2 \right]$$

Once again I emphasize that these results are important for the verification of the stability of our numerical analysis. This derivation is shown here not only as an exercise, but also because I couldn't find the 2-d version in any textbook.

Chapter 2

Finite Differences

2.1 Introduction

Differently from the non-approximate analytical solutions of PDEs in the continuum (for instance, those obtained through variable separation and subsequent integration), numerical solutions obtained in a computer have limited precision[1]. It is due to the way in which computers store data and also because of their limited memory. After all, how could we write in decimal notation (or in any other base) an irrational number like $\sqrt{2}$ making use of a finite number of digits? In this work we won't stick with rigorous derivations of the theorems nor of most of the results presented. The references listed in the end should be considered for this end.

The central idea of numerical methods is quite simple: to give finite precision ("the discrete") to those objects endowed with infinite precision ("the continuum"). By *discretize* we understand to transform continuum variables like $x, y, .., z$ into a set of discrete values $\{x_i\}, \{y_i\}, ..., \{z_i\}$, where i runs over a finite number of values, thus sampling the wholeness of the original variables. As a consequence of this discretization, integrals become sums and derivatives turns out to mere differences of finite quantities (hence the name "finite differences"). I illustrate below these ideas:

$$\int f(x)dx = \lim_{\delta x \to 0} \sum_n f(n\delta x)\delta x \to \sum_n f(n\Delta x)\Delta x \tag{2.1}$$

$$\frac{df(x)}{dx} = \lim_{\delta x \to 0} \frac{f(x+\delta x) - f(x)}{\delta x} \to \frac{f(x+\Delta x) - f(x)}{\Delta x}, \tag{2.2}$$

where δx is a variable with *infinite* precision (thus its value could be as small as we want) and $\Delta x \ll 1$ is a variable with *finite* precision, which under the computational point of view is the limiting case analogous to δx. We could naively expect that, the smaller the value of Δx, the closer we are to the continuum theory. This would be indeed true if computers didn't have finite precision! The closer your significant digits get to the limiting precision of the computer, the *worse* is your approximation, because it will introduce the well-known "round-off errors", which are basically truncation errors. The reference [8] has a somewhat lengthy discussion about computational issues like this.

[1] In this point it is worth mentioning that there are basically two ways of solving a mathematical problem with the aid of a computer: symbolically and numerically. Symbolic methods deal fundamentally with algebraic manipulations and do not involve explicit numerical calculations, giving us an analytical form (whenever possible) to the desired problem. It is, however, widely understood that non-linear theories hardly have a closed-form solution, and even if they do, it is often a lot complicated and requires an understanding of very sophisticated tools. Whenever this is the case, one often resorts to the numerical approach, which doesn't furnish us with an analytical closed-form solution, but could give very precise numerical estimates for the solution of the problem. It has been used since the very beggining of the computer era, and today it is sometimes the *only* tool people have to attack some problems, pervading its use in almost every discipline of science and technology.

2.2 Difference Equations

Difference equations are to a computer in the same way as *differential* equations are to a good mathematician. That is, if you have a problem in the form of a differential equation, the most straightforward way of solving it is to transform your derivatives into differences, so that you finish with an *algebraic* difference equation. This turns out to be necessary for what we said about the limitations of a computer[2].

As a trivial example, take the ordinary differential equation:

$$\frac{df}{dx} = g(x) \tag{2.3}$$

Using a first-order Taylor expansion (see Appendix A) for $f(x)$,

$$f(x+h) \approx f(x) + f'(x)h \Rightarrow$$

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

we obtain the *Euler form* for the Eq. (2.3):

$$\frac{f(x+h) - f(x)}{h} \approx g(x)$$

Notice that this equation involves only differences as we said above, and to solve it in a computer we shall need the following *iterative relation* obtained directly from the above equation:

$$f(x+h) = hg(x) + f(x)$$

or, in the traditional numerical notation:

$$f_{n+1} = hg_n + f_n \tag{2.4}$$

Technically, once provided both the initial condition (for instance, $f_0 = 0$) and the functional form of $g_n = g(x_n)$, we could solve Eq. (2.4) by iterating it in a program loop.

In spite of its simple form, Euler's approximation is far from being useful for realistic equations; it could give rise to a completely erroneous approximation. Higher order expansions are frequently used in order to obtain equations with reduced error (see again Appendix A for some of these expansions). However, these higher order approximations are *also* subject to serious problems, like the lack of stability or convergence, so the problem is ubiquitous and has been one of the most attacked problems in the so-called "numerical analysis", a relatively modern branch of mathematics. I will say a little bit more later about these issues on convergence and stability.

It has already been said that the relevance of the classification of PDEs lies in their "nature"; those of initial value have a completely different way of solving numerically from those of boundary values. The latter kind doesn't evolve in time. It's the classic case of the Poisson equation which could be describing a thermostatic or an electrostatic system:

$$\frac{\partial^2 u(x,y)}{\partial x^2} + \frac{\partial^2 u(x,y)}{\partial y^2} = f(x,y) \tag{2.5}$$

Using the expansions from Appendix A, we have:

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h^2} = f_{i,j}, \tag{2.6}$$

[2] There are, however, more sophisticated methods like Finite Elements, but the fact that one needs to get rid of differentials transcends these methods when we are talking about numerical solutions.

where we took h as the *lattice spacing* (also grid or net resolution) both for the coordinate x and y ($x \to x_n = nh$ and $y \to y_n = nh$). The indices of this equation then correspond to *sites* in this lattice (a.k.a. lattice points), and sweep from 0 to the number of sites N_i or N_j. The problem becomes then to solve the equations given by (2.6) simultaneously for the various $u_{i,j}$. There are very interesting methods to solve this sort of problem which could be found in the references [6, 8].

Our work, however, is directed towards initial value problems which, as the very name suggests, deal with temporal evolutions starting from certain "initial values" at the "zero" instant. It is the typical case of the wave equation already presented, or of the diffusion equation

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial u}{\partial t}. \tag{2.7}$$

Our focus goes even finer, since we will deal only with "explicit discretizations", which could be understood as those which could be solved iteratively, that is, we could solve the difference equation for $u(x, y, t + \Delta t)$ explicitly in terms of the other variables $u(x', y', t)$ at the instant t (for instance, Euler's equation above is an explicit method). Implicit methods need a different approach, which often involves the solution of linear systems by using matrices (again [6, 8] do very well in these matters).

2.3 The von Neumann Stability Analysis

How should we know, after transforming a differential equation into finite differences, if the calculated solution is a stable one? By *numerically stable solutions* we understand those in which the error z_m^n between the correct theoretical solution $u(x_m, t_n)$ and the numerical solution U_m^n does not diverge (i.e., is limited) as $n \to \infty$ ($t \to \infty$), in other words:

$$z_m^n \equiv u(x_m, t_n) - U_m^n < \epsilon, \tag{2.8}$$

for any n, where the lower indices are spatial and the upper ones are temporal, and ϵ is a *finite* real value. For instance, an unstable discretization describing a vibrating string could be easily detected watching the energy of the system for a while: a divergent energy would certainly arise. Fortunately, there is a useful tool to identify unstable finite difference equations prior to simulating it, known as *von Neumann stability analysis* [6, 9], which could be applied to a difference equation to preview its numerical behavior.

The von Neumann method consists essentially in expanding the numerical error z_m^n in a discrete harmonic Fourier series:

$$z_m^n = \sum_r a_r(t_n) e^{ik_r x_m} \tag{2.9}$$

and analyzing if $a_r(t_n)$ increases (or decreases) as $t \to \infty$ (technically, if $a_r(t)$ decreases when $t \to \infty$ we have a *numerical dissipation*, which is usually harmless). It is then easy to see that if $a_r(t_n)$ isn't divergent for any n and m we will have a stable solution. This analysis is somewhat simple, since it is sufficient to study the behavior of a *single general* term of the series, for if we prove that this general term of the series could have a certain pathological behavior (like diverging for $n \to \infty$), then the *whole* solution is compromised; otherwise, our solution is stable.

Mitchell and Griffiths [9] show that z_m^n given by (2.8) satisfy the very same difference equation for u_m^n. Hence, if we take a certain z_m^n such that $\left|z_m^0\right| = 1$ and put it into the difference equation, we could achieve the desired stability condition. One possible z_m^n satisfying the criteria above is:

$$z_m^n = e^{\alpha n \Delta t} e^{i\beta m \Delta x} \tag{2.10}$$

Indeed, notice that for $n = 0$ we have $\left|z_m^0\right| = 1$, and with α and β arbitrary values we satisfy the above discussion. With this expression, we could now write the stability condition for the von Neumann analysis:

$$|\xi^n| \leq 1, \tag{2.11}$$

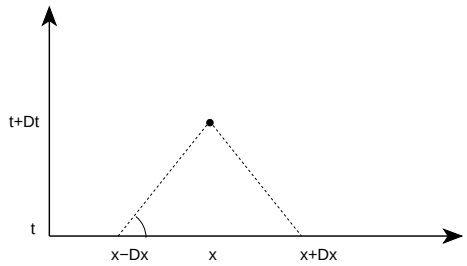

Figure 2.1: Representing an integration algorithm that needs the values $u(x - Dx, t)$ and $u(x + Dx, t)$ to obtain $u(x, t + Dt)$. The ratio Dx/Dt (which is the tangent of the angle in the base of the triangle) is then the "maximum speed" with which an information in the algorithm could propagate.

where $\xi = e^{\alpha \Delta t}$ is the *amplification factor*. In summary, putting the error given by

$$z_m^n = \xi^n e^{i\beta m \Delta x} \tag{2.12}$$

into the difference equation, together with (2.11), we get the necessary condition for stability.

2.4 The Courant Condition

Another important condition that we should pay some attention in initial value problems is related to the *speed* with which information could propagate in the difference equation. We could visualize the problem in the scheme of Figure 2.1.

It could be shown [6] that, applying von Neumann's condition for hyperbolic problems we arrive at the *Courant condition*: if the "physical" wave velocity $|v|$ in a differential equation is greater than the "algorithm speed" $\Delta x/\Delta t$, then the scheme is unstable. We have therefore the following expression for the Courant condition:

$$|v| \leq \frac{\Delta x}{\Delta t} \tag{2.13}$$

2.5 The *Leap-Frog* Algorithm and the Wave Equation

Consider the 1-d wave equation (to ease the notation, from now on we will take $v = \tau = \lambda = \sigma = 1$):

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

Using the second order expansion (A.9) from Appendix A for the derivatives above, we have:

$$\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} = \frac{u_i^{n+1} - 2u_i^n + u_i^{n-1}}{\Delta t^2}, \tag{2.14}$$

or, solving this algebraic equation for u_i^{n+1},

$$u_i^{n+1} = \rho \left(u_{j+1}^n + u_{j-1}^n\right) + 2(1-\rho)u_j^n - u_j^{n-1},$$

with $\rho = (\Delta t/\Delta x)^2$.

Notice that *this equation is explicit* and *has second order precision both in time and space* (we didn't write the error $\mathcal{O}(\Delta x^2, \Delta t^2)$, but you can easily track it when you do the above passage). We could also obtain it taking second order approximations for the characteristics of the wave equation (the definition of the characteristics of an equation could be found in [5, Cap. 8]),

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

so, with these approximations, to find u_m^{n+1} we need u_m^{n-1}, u_{m-1}^n and u_{m+1}^n. Then the name *leap-frog*, since, with respect to the time approximation for the derivative, we "leap", from $n-1$ to $n+1$, over the spatial derivative which involves only approximations at the instant n.

Now let us consider the case of the wave equation with dissipation:

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

where η is the viscosity coefficient. Using also a second order expansion for the new term we have:

$$\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} = \frac{u_i^{n+1} - 2u_i^n + u_i^{n-1}}{\Delta t^2} + \eta \frac{u_i^{n+1} - u_i^{n-1}}{2\Delta t},$$

from which we could solve for the term u_i^{n+1}:

$$u_i^{n+1} = \left[1 + \frac{\eta \Delta t}{2}\right]^{-1} \left\{\rho\left(u_{i+1}^n + u_{i-1}^n\right) + 2(1-\rho)u_i^n - \left[1 - \frac{\eta \Delta t}{2}\right] u_i^{n-1}\right\} \tag{2.15}$$

Let's consider now the 2d case with a symmetric spacing for the two coordinates ($\Delta x = \Delta y = \Delta l$) plus a viscosity term:

$$\frac{u_{i+1,j}^n - 2u_{i,j}^n + u_{i-1,j}^n}{\Delta l^2} + \frac{u_{i,j+1}^n - 2u_{i,j}^n + u_{i,j-1}^n}{\Delta l^2} = \frac{u_{i,j}^{n+1} - 2u_{i,j}^n + u_{i,j}^{n+1}}{\Delta t^2} + \eta \frac{u_{i,j}^{n+1} - u_{i,j}^{n-1}}{2\Delta t}$$

and, again solving for $u_{i,j}^{n+1}$,

$$\begin{aligned} u_{i,j}^{n+1} &= \left[1 + \tfrac{\eta \Delta t}{2}\right]^{-1} \left\{\rho\left[u_{i+1,j}^n + u_{i-1,j}^n + u_{i,j+1}^n + u_{i,j-1}^n - 4u_{i,j}^n\right] + \right. \\ &\quad \left. 2u_{i,j}^n - \left[1 - \tfrac{\eta \Delta t}{2}\right] u_{i,j}^{n-1}\right\} \end{aligned} \tag{2.16}$$

It could be shown that these difference equations *satisfy the von Neumann criterion when* $\Delta x/\Delta t$ *satisfy the Courant criterion* [6]. For the sake of completeness, let's now see how the above equations (i.e., the 1-d and 2-d discretizations) change when we add an "arbitrary" term[3] $F(u^n)$ which depends on $u(t)$ (it could be, for instance, the term $\delta V[\phi]/\delta\phi(\vec{x})$ which arises in classical field theories, where the $\lambda\phi^4$ is the paradigm). The differential equation for these cases are:

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2} - \eta \frac{\partial u}{\partial t} - F(u)$$

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} - \eta \frac{\partial u}{\partial t} - F(u),$$

whereas their discretizations are:

$$u_i^{n+1} = \left[1 + \frac{\eta \Delta t}{2}\right]^{-1} \left\{\rho\left(u_{i+1}^n + u_{i-1}^n\right) + 2(1-\rho)u_i^n - \left[1 - \frac{\eta \Delta t}{2}\right] u_i^{n-1} - \Delta t^2 F(u^n)\right\} \tag{2.17}$$

$$\begin{aligned} u_{i,j}^{n+1} &= \left[1 + \tfrac{\eta \Delta t}{2}\right]^{-1} \left\{\rho\left[u_{i+1,j}^n + u_{i-1,j}^n + u_{i,j+1}^n + u_{i,j-1}^n - 4u_{i,j}^n\right] + \right. \\ &\quad \left. 2u_{i,j}^n - \left[1 - \tfrac{\eta \Delta t}{2}\right] u_{i,j}^{n-1} - \Delta t^2 F(u^n)\right\} \end{aligned} \tag{2.18}$$

It may be worth mentioning that we need to define both the initial values and the boundary conditions in order to solve the above equations. For the applications which will be shown in Chapter 3, we shall use the following conditions:

[3] Notice that this term could *not* depend on time derivatives of $u(t)$ or any other "non-local" time dependence, that is, it should be defined completely in terms of u^n; otherwise we might not be able to solve the equation for u^{n+1} explicitly. Spatial derivatives are not a problem since they are defined locally in time.

$$\begin{cases} u(t)|_{boundaries} = 0 \\ \left.\frac{\partial u(t)}{\partial t}\right|_{boundaries} = 0 \\ u(x,y)|_{t=0} = C \exp\left[-\frac{(\vec{r}-\vec{r}_0)^2}{2\gamma}\right], \end{cases} \tag{2.19}$$

where $\vec{r} = x\hat{i} + y\hat{j}$, $\vec{r}_0 = \frac{l}{2}\hat{i} + \frac{l}{2}\hat{j}$, l is the lattice length, C is a normalization constant, and γ is a sufficiently small constant such that $u(\vec{r}) \to 0$ as $\vec{r} \to \textit{boundaries}$, that is, the initial condition is a gaussian sufficiently localized to make u continuous at the boundaries.

Chapter 3

Examples

3.1 The Free String (1D)

These simulations were executed in a PC of $350MHz$, for lattices of at most $N = 1000$. The integration time lay in the order of seconds.

Figure 3.1 shows some results for the conditions of the previous section (2.19) for various parameters. Notice the improvement of energy conservation for finer resolutions and the very good exponential fitting for $\eta = 1$. This exponential result is expected, since the vibrating string could be understood in terms of the Fourier space, where each mode behaves as a decoupled harmonic oscillator with a damping given by η.

3.2 The Membrane (2D)

These simulations were also executed in a PC of $350MHz$, and for lattices of $N = 500$ the integration time reached half an hour.

Figure 3.2 shows some results for $\eta = 0$ and $\eta = 1$. For a non-conservative system ($\eta = 1$), we see also the exponential fitting for $N = 200$.

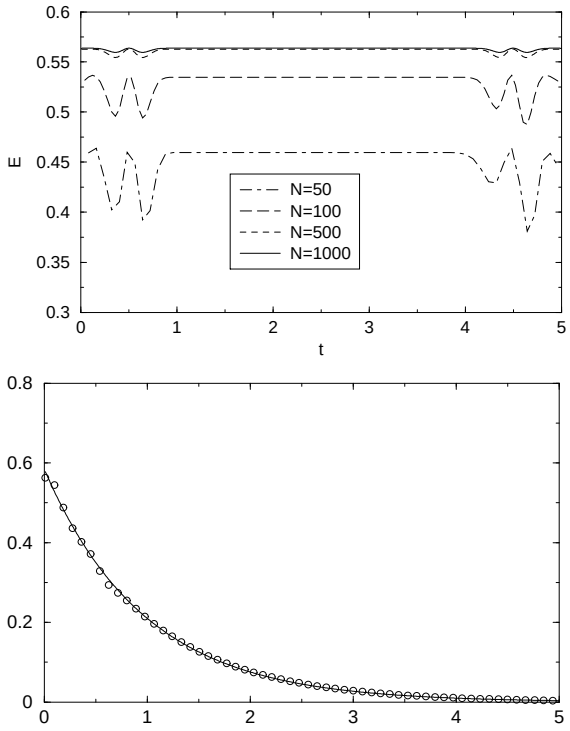

Figure 3.1: $E \times t$ for different lattice spacings and $\eta = 0$ (above) and $E \times t$ for $N = 1000$ and $\eta = 1$ (below). The linear dimension is fixed at $L = 1$.

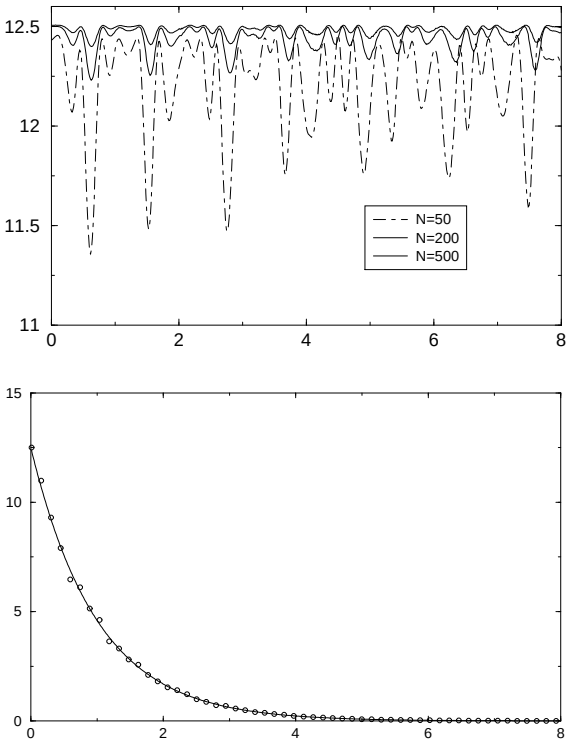

Figure 3.2: $E \times t$ for $\eta = 0$ and various N (above) and $E \times t$ for $\eta = 1$ and $N = 200$ (below). The dimensions of the membrane are $L \times L = 1$ for all runs.

Appendix A

Taylor's Theorem

A.1 Definitions

When we want to transform a differential equation into a difference equation, the Taylor expansion is often used:

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2}f''(x_0)(x - x_0)^2 + ... = \tag{A.1}$$

$$= \sum_{n=0}^{\infty} \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n, \tag{A.2}$$

where x_0 is the point around which we want to expand $f(x)$[1]. An alternative form for this expansion is achieved doing a simple variable change $x \to x + h$ and $x_0 \to x$:

$$f(x + h) = \sum_{n=0}^{\infty} \frac{f^{(n)}(x)}{n!}h^n \tag{A.3}$$

In the numerical case we will be interested in *truncating* the series, so that we finish with a finite number of terms. We could then write this expansion in the following form:

$$f(x + h) = \sum_{n=0}^{m} \frac{f^{(n)}(x)}{n!}h^n + \mathcal{O}(h^{m+1}), \tag{A.4}$$

where $\mathcal{O}(h^{m+1})$ corresponds to the truncated terms which powers of h are equal or higher than $m + 1$ (this term is frequently called the *error of order* $m + 1$). Notice that under the numerical point of view it is important to know the order of $\mathcal{O}$ in the discretization, since for $h \ll 1$, the greater the order of $\mathcal{O}$ the more negligible will be the error.

A.2 Useful Expansions

Some expansions that will be used throughout this text are shown below. All of them could be obtained from (A.4) by directly solving for the desired term or using more than one expansion to find higher order expansions for the derivative, and then solving the system. For instance:

$$\begin{cases} f(x + h) &= f(x) + f'(x)h + \frac{1}{2}f''(x)h^2 + ... \\ f(x - h) &= f(x) - f'(x)h + \frac{1}{2}f''(x)h^2 - ... \end{cases}$$

$$f'(x) = \frac{f(x + h) - f(x)}{h} - \mathcal{O}(h) \tag{A.5}$$

[1] For $x_0 = 0$ this expansion is also known as *Maclaurin expansion*.

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - 2\mathcal{O}(h^2) \tag{A.6}$$

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} - 2\mathcal{O}(h^2) \tag{A.7}$$

$$\frac{\partial f(x,y)}{\partial x} = \frac{f(x+h,y) - f(x,y)}{h} - \mathcal{O}(h) \tag{A.8}$$

$$\frac{\partial^2 f(x,y)}{\partial x^2} = \frac{f(x+h,y) - 2f(x,y) + f(x-h,y)}{h^2} - 2\mathcal{O}(h^2) \tag{A.9}$$

Notice that when we divide an error of order $\mathcal{O}(h^n)$ by h^r, automatically this error turns to order $n-r$, i.e., $\mathcal{O}(h^n)/h^r = \mathcal{O}(h^{n-r})$.

Bibliography

[1] M. Gleiser and R. O. Ramos, Phys. Rev. D**50**, 2441 (1994).

[2] G. Aarts and J. Smit, Nucl. Phys. B**555**, 355 (1999); Phys. Rev. D**61**, 025002 (2000).

[3] For a good review on the subject and its high-energy counterpart, see D. Boyanovsky and H. J. de Vega, *"Dynamics of Symmetry Breaking Out of Equilibrium: From Condensed Matter to QCD and the Early Universe"*, `hep-ph/9909372`.

[4] J. Borrill and M. Gleiser, Nucl. Phys. **B483**, 416 (1997)

[5] G. B. Arfken, H. J. Weber, *Mathematical Methods for Physicists,* 4th Ed.

[6] W. H. Press *et al., Numerical Recipes,* 2nd Ed.

[7] K. R. Symon, *Mechanics*, 3rd. Ed.

[8] W. Cheney e D. Kincaid, *Numerical Mathematics and Computing*, 3rd Ed.

[9] A. R. Mitchell e D. F. Griffiths, *The Finite Difference Method in Partial Differential Equations*