2006 Spring School on
Numerical Methods in Gravitation and Astrophysics

Introduction to Numerical Relativity
Lecture 2

Basic Finite Difference Techniques for Time Dependent PDEs

March 15, 2006

(adapted from Taller de Verano 1999 notes, MWC, July 1999)

`http://bh0.phas.ubc.ca/korea06/2006-korea-mwc-2.pdf`

Matthew W. Choptuik
CIAR Cosmology & Gravity Program
Dept of Physics & Astronomy
University of British Columbia
Vancouver, BC, Canada

Contents

1 Basic Finite Difference Techniques for Time Dependent PDEs

There are several good reference texts for this material. Among my personal favorites are [1]-[4].

1.1 Preliminaries

We can divide time-dependent PDEs into two broad classes:

1. **Initial-value Problems (Cauchy Problems)**, spatial domain has no boundaries (either infinite or "closed"—e.g. "periodic boundary conditions"

2. **Initial-Boundary-Value Problems**, spatial domain *finite*, need to specify boundary conditions

Note: Even if *physical* problem is really of Type 1, finite computational resources $\longrightarrow$ finite spatial domain $\longrightarrow$ approximate as Type 2; will hereafter loosely refer to either type as an IVP.

Working Definition: **Initial Value Problem**

- State of physical system arbitrarily (usually) specified at some initial time $t = t_0$.

- Solution exists for $t \geq t_0$; uniquely determined by equations of motion (EOM) and boundary conditions (BCs).

Issues in Finite Difference (FD) Approximation of IVPs

- Discretization (Derivation of FDA's)

- Solution of algebraic systems resulting from discretization

- Consistency

- Accuracy

- Stability

- Converegence

- Dispersion / Dissipation

- Treatment of Non-linearities

- Computational cost—expect $O(N)$ work ($N \equiv$ number of "grid points" (discrete events at which approximate solution is computed)

1.2 Types of IVPs (by example)

In the following three examples, u is always a function of one space and one time variable, i.e. $u \equiv u(x,t)$. Such a problem is often referred to as "1-d" by numericists, the time dimension being implicit in this nomenclature. I will also use the subscript notation for partial differentiation, e.g. $u_t \equiv \partial_t u$.

1.2.1 Wave and "Wave-Like" ("Hyperbolic"): The 1-d Wave Equation

$$\begin{aligned} u_{tt} &= c^2 u_{xx} \qquad c \in \mathbf{R}, \\ u(x,0) &= u_0(x) \\ u_t(x,0) &= v_0(x) \end{aligned} \tag{1}$$

1.2.2 Diffusion ("Parabolic"): The 1-d Diffusion Equation

$$\begin{aligned} u_t &= \sigma u_{xx} \qquad \sigma \in \mathbf{R}, \quad \sigma > 0. \\ u(x,0) &= u_0(x) \end{aligned} \tag{2}$$

1.2.3 Schrödinger: The 1-d Schrödinger Equation

$$\begin{aligned} i\psi_t &= -\frac{\hbar}{2m}\psi_{xx} + V(x,t)\psi \qquad \psi \in \mathbf{C} \\ \psi(x,0) &= \psi_0(x) \end{aligned} \tag{3}$$

Note: Although $\psi(x,t)$ is *complex* in this case, we can rewrite 3 as a *system* of 2 coupled scalar, real-valued equations.

1.3 Some Basic Concepts, Definitions and Techniques

We will be considering the finite-difference approximation (FDA) of PDEs, and as such, will generally we interested in the continuum limit, where the *mesh spacing*, or *grid spacing*, usually denoted h, tends to 0. Because any specific calculation must necessarily be performed at some specific, *finite* value of h, we will also be (extremely!) interested in the way that our discrete solution varies as a function of h. In fact, we will *always* view h as the basic "control" parameter of a typical FDA. Fundamentally, for sensibly constructed FDAs, we expect the error in the approximation to go to 0, as h goes to 0.

Let

$$Lu = f \tag{4}$$

denote a general *differential* system. For simplicity and concreteness, you can think of $u = u(x,t)$ as a single function of one space variable and time, but the discussion in this section applies to cases in more independent variables ($u(x,y,t)$, $u(x,y,z,t)$ $\cdots$ etc.), as well as multiple *dependent* variables ($u = \mathbf{u} = [u_1, u_2, \cdots, u_n]$). In (4), L is some differential operator (such as $\partial_{tt} - \partial_{xx}$) in our wave equation example), u is the unknown, and f is some specified function (frequently called a *source* function) of the independent variables.

Here and in section 1.9 it will be convenient to adopt a notation where a superscript h on a symbol indicates that it is discrete, or associated with the FDA, rather than the continuum. (Note, however, that for simplicity of presentation, we will *not* adopt this notation in much of the development below). With this notation, we will generically denote an FDA of (4) by

$$L^h u^h = f^h \tag{5}$$

where u^h is the discrete solution, f^h is the specified function evaluated on the finite-difference mesh, and L^h is the finite-difference approximation of L.

1.3.1 Residual

Note that another way of writing our FDA is

$$L^h u^h - f^h = 0 \tag{6}$$

It is often useful to view FDAs in this form for the following reason. First, we have a canonical view of what it means to solve the FDA—"drive the left-hand side to 0". Furthermore, for iterative approaches to the solution of the FDA (which are common, since it may be too expensive to solve the algebraic equations directly), we are naturally lead to the concept of a *residual*. The residual is simply the level of "non-satisfaction" of our FDA (and, indeed, of any algebraic expression). Specifically, if $\tilde{u}^h$ is some approximation to the true solution of the FDA, u^h, then the residual, r^h, associated with $\tilde{u}^h$ is just

$$r^h \equiv L^h \tilde{u}^h - f^h \tag{7}$$

This leads to the view of a convergent, iterative process as being one which "drives the residual to 0".

1.3.2 Truncation Error

The *truncation error*, τ^h, of an FDA is defined by

$$\tau^h \equiv L^h u - f^h \tag{8}$$

where u satisfies the continuum PDE (4). We note that the *form* of the truncation error can always be computed (typically using Taylor series) from the finite difference approximation and the differential equations.

1.3.3 Convergence

Assuming that our FDA is characterized by a *single* discretization scale, h, we say that the approximation *converges* iff

$$u^h \to u \quad \text{as} \quad h \to 0. \tag{9}$$

Operationally (i.e. in practice), convergence is clearly our chief concern as numerical analysts, particularly if there is reason to suspect that the solutions of our PDEs are good models for real phenomena. We note that this is believed to be the case for many interesting problems in general relativistic astrophysics—the two black hole problem being an excellent example.

1.3.4 Consistency

Assuming that the FDA with truncation error τ^h is characterized by a single discretization scale, h, we say that the FDA is *consistent* if

$$\tau^h \to 0 \quad \text{as} \quad h \to 0. \tag{10}$$

Consistency is obviously a necessary condition for convergence.

1.3.5 Order of an FDA

Assuming that the FDA is characterized by a single discretization scale, h, we say that the FDA is *p-th order accurate* or simply *p-th order* if

$$\lim_{h\to 0} \tau^h = O(h^p) \qquad \text{for some integer } p \tag{11}$$

1.3.6 Solution Error

The solution error, e^h, associated with an FDA is defined by

$$e^h \equiv u - u^h \tag{12}$$

1.3.7 Relation Between Truncation Error and Solution Error

Is is common to tacitly assume that

$$\tau^h = O(h^p) \qquad \longrightarrow \qquad e^h = O(h^p)$$

This assumption is often warranted, but it is extremely instructive to consider *why* it is warranted and to investigate (following Richardson 1910 (!) [5]) in some detail the *nature* of the solution error. We will return to this issue in more detail in section 1.9.

1.3.8 Deriving Finite Difference Formulae

The essence of finite-difference approximation of a PDE is the replacement of the continuum by a discrete lattice of grid points, and the replacement of derivatives/differential operators by finite-difference expressions. These finite-difference expressions (finite-difference quotients) approximate the derivatives of functions at grid points, using the grid values themselves. All of the operators and expressions we need can easily be worked out using Taylor series techniques. For example, let us consider the task of approximating the first derivative $u_x(x)$ of a function $u(x)$, given a discrete set of values $u_j \equiv u(jh)$ as shown in Figure 1. As it turns out, given the three values $u(x_j - h), u(x_j)$ and $u(x_j + h)$, which we will denote u_{j-1}, u_j, and u_{j+1} respectively, we can compute an $O(h^2)$ approximation to $u_x(x_j) \equiv (u_x)_j$ as follows. Taylor expanding, we have

$$\begin{aligned}
u_{j-1} &= u_j - h(u_x)_j + \frac{1}{2}h^2(u_{xx})_j - \frac{1}{6}h^3(u_{xxx})_j + \frac{1}{24}h^4(u_{xxxx})_j + O(h^5) \\
u_j &= u_j \\
u_{j+1} &= u_j + h(u_x)_j + \frac{1}{2}h^2(u_{xx})_j + \frac{1}{6}h^3(u_{xxx})_j + \frac{1}{24}h^4(u_{xxxx})_j + O(h^5)
\end{aligned}$$

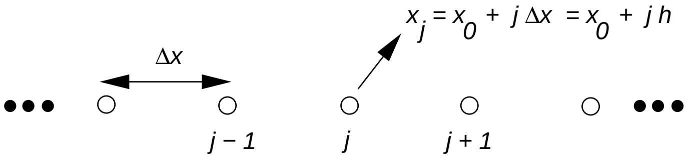

Figure 1: A one-dimensional, uniform finite difference mesh. Note that the spacing, $\triangle x = h$, between adjacent mesh points is *constant*. In the text we tacitly assume that the origin, x_0, of our coordinate system is $x_0 = 0$.

We now seek a linear combination of u_{j-1}, u_j, and u_{j+1} which yields $(u_x)_j$ to $O(h^2)$ accuracy, i.e. we seek c_-, c_0 and c_+ such that

$$c_- \, u_{j-1} + c_0 \, u_j + c_+ \, u_{j+1} = (u_x)_j + O(h^2)$$

This results in a system of three linear equations for u_{j-1}, u_j, and u_{j+1}:

$$\begin{aligned} c_- \; + \; c_0 \; + \; c_+ \; &= \; 0 \\ -hc_- \; + \; hc_+ \; &= \; 1 \\ \frac{1}{2}h^2 c_- \; + \; \frac{1}{2}h^2 c_+ \; &= \; 0 \end{aligned}$$

which has the solution

$$\begin{aligned} c_- &= -\frac{1}{2h} \\ c_0 &= 0 \\ c_+ &= +\frac{1}{2h} \end{aligned}$$

Thus our $O(h^2)$ finite difference approximation for the first derivative is

$$\frac{u(x+h) - u(x-h)}{2h} = u_x(x) + O(h^2) \qquad (13)$$

Note that it may not be obvious to you *a priori*, that the truncation error of this approximation *is* $O(h^2)$, since a naive consideration of the number of terms in the Taylor series expansion which can be eliminated using 2 values (namely $u(x+h)$ and $u(x-h)$) suggests that the error might be $O(h)$. The fact that the $O(h)$ term “drops out” is a consequence of the *symmetry*, or *centering* of the stencil, and is a common theme in such FDAs (which, naturally enough, are called centred difference approximations).

Using the same technique, we can easily generate the $O(h^2)$ expression for the *second* derivative, which uses the same difference stencil as the above approximation for the first derivative.

$$\frac{u(x+h) - 2u(x) + u(x-h)}{h^2} = u_{xx}(x) + O(h^2) \qquad (14)$$

Exercise: Compute the precise form of the $O(h^2)$ terms in expressions (13) and (14).

1.4 Sample Discretizations / FDAs

1.4.1 1-d Wave equation with fixed (Dirichlet) boundary conditions

$$\begin{aligned} u_{tt} &= u_{xx} \qquad (c=1) \qquad 0 \le x \le 1; \quad t \ge 0 \qquad (15) \\ u(x,0) &= u_0(x) \\ u_t(x,0) &= v_0(x) \\ u(0,t) &= u(1,t) = 0 \end{aligned}$$

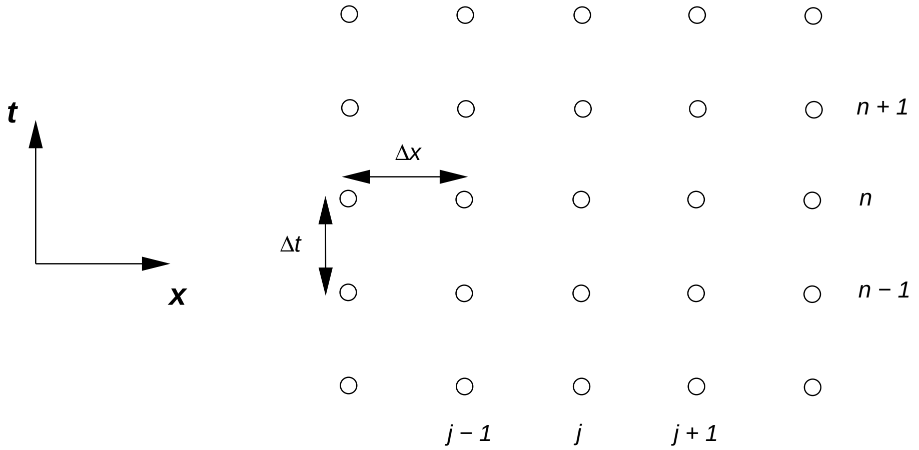

Figure 2: Portion of uniform finite-difference mesh (grid) for 1-d time-dependent problem. Note that the spacings in both the spatial *and* temporal directions are constant

We now introduce a discrete domain (uniform grid) (x_j, t^n)—part of which is shown in Figure 2.

$$
\begin{aligned}
t^n &\equiv n \triangle t\,, \qquad n = 0, 1, 2, \cdots \\
x_j &\equiv (j-1)\,\triangle x\,, \qquad j = 1, 2, \cdots J \\
u^n_j &\equiv u(n\,\triangle t\,, (j-1)\,\triangle x\,) \\
\triangle x &= (J-1)^{-1} \\
\triangle t &= \lambda\,\triangle x \qquad \lambda \equiv \text{“Courant number”}
\end{aligned}
$$

Note: When solving wave equations using FDAs, we will typically keep λ constant when we vary $\triangle x$. Thus, our FDA will always be characterized by *single* discretization scale, h.

$$
\begin{aligned}
\triangle x &\equiv h \\
\triangle t &\equiv \lambda h
\end{aligned}
$$

(Also note the `Fortran`-style indexing of the spatial grid index ($j = 1, 2, \cdots$) and the `C`-style indexing of the temporal one ($n = 0, 1, \cdots$). This is a particular convention which I, as a predominantly `Fortran` programmer, find convenient.)

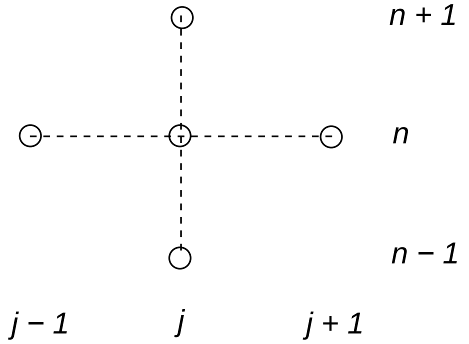

Figure 3: Stencil (molecule/star) for “standard” $O(h^2)$ approximation of (15).

FDA: “standard $O(h^2)$”

Discretized Interior equation:

$$\begin{aligned}
(\triangle t)^{-2}\left(u_j^{n+1} - 2u_j^n + u_j^{n-1}\right) &= (u_{tt})_j^n + \frac{1}{12}\triangle t^2 (u_{tttt})_j^n + O(\triangle t^4) \\
&= (u_{tt})_j^n + O(h^2) \\
(\triangle x)^{-2}\left(u_{j+1}^n - 2u_j^n + u_{j-1}^n\right) &= (u_{xx})_j^n + \frac{1}{12}\triangle x^2 (u_{xxxx})_j^n + O(\triangle x^4) \\
&= (u_{xx})_j^n + O(h^2)
\end{aligned}$$

Putting these two together, we get the $O(h^2)$ approximation

$$\frac{u_j^{n+1} - 2u_j^n + u_j^{n-1}}{\triangle t^2} = \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\triangle x^2} \qquad j = 2, 3, \cdots, J-1 \tag{16}$$

Note that a scheme such as (16) is often called a *three level scheme* since it couples *three "time levels"* of data (i.e. unknowns at three distinct, disrete times t^{n-1}, t^n, t^{n+1}.

Discretized Boundary conditions:

$$u_1^{n+1} = u_J^{n+1} = 0$$

Discretized Initial conditions:
We need to specify *two* "time levels" of data (effectively $u(x,0)$ and $u_t(x,0)$), i.e. we must specify

$$\begin{aligned}
u_j^0 \quad &, \quad j = 1, 2, \cdots, J \\
u_j^1 \quad &, \quad j = 1, 2, \cdots, J
\end{aligned}$$

ensuring that the initial values are compatible with the boundary conditions.

Note that we can solve (16) *explicitly* for u_j^{n+1}:

$$u_j^{n+1} = 2u_j^n - u_j^{n-1} + \lambda^2\left(u_{j+1}^n - 2u_j^n + u_j^{n-1}\right) \tag{17}$$

Also note that (17) is actually a *linear system* for the unknowns $u_j^{n+1},\ j = 1, 2, \cdots, J$; in combination with the discrete boundary conditions we can write

$$\mathbf{A}\ \mathbf{u}^{n+1} = \mathbf{b} \tag{18}$$

where $\mathbf{A}$ is a *diagonal* $J \times J$ matrix and $\mathbf{u}^{n+1}$ and $\mathbf{b}$ are vectors of length J. Such a difference scheme for an IVP is called an *explicit* scheme.

1.4.2 1-d Diffusion equation with Dirichlet boundary conditions

$$\begin{aligned}
u_t &= u_{xx} \qquad (\sigma = 1) \qquad 0 \le x \le 1; \quad t \ge 0 \\
u(x,0) &= u_0(x) \\
u(0,t) &= u_(1,t) = 0
\end{aligned} \tag{19}$$

We will use same discrete domain (grid) as for the 1-d wave equation.

FDA: Crank-Nicholson
This scheme illustrates a useful "rule of thumb": *Keep difference scheme "centred"*

- centred in time, centred in space

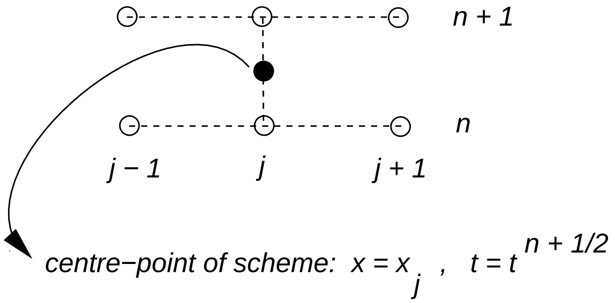

Figure 4: Stencil (molecule/star) for $O(h^2)$ Crank-Nicholson approximation of (19).

- minimizes truncation error for given h
- tends to minimize instabilities

Discretization of time derivative:

$$\begin{aligned}\triangle t^{-1}\left(u_j^{n+1} - u_j^{n}\right) &= (u_t)_j^{n+\frac{1}{2}} + \frac{1}{24}\triangle t^{2}\,(u_{ttt})_j^{n+\frac{1}{2}} + O(\triangle t^{4}) \qquad (20)\\ &= (u_t)_j^{n+\frac{1}{2}} + O(\triangle t^{2})\end{aligned}$$

$O(h^2)$ second-derivative operator:

$$\begin{aligned}D_{xx}\,u_j^n &\equiv \triangle x^{-2}\left(u_{j+1}^n - 2u_j^n + u_{j+1}^n\right) \qquad (21)\\ D_{xx} &= \partial_{xx} + \frac{1}{12}\triangle x^{2}\,\partial_{xxxx} + O(\triangle x^{4}) \qquad (22)\end{aligned}$$

(Forward) Time-averaging operator, μ_t:

$$\begin{aligned}\mu_t\,u_j^n &\equiv \frac{1}{2}\left(u_j^{n+1} + u_j^{n}\right) = u_j^{n+\frac{1}{2}} + \frac{1}{8}\triangle t^{2}\,(u_{tt})_j^{n+\frac{1}{2}} + O(\triangle t^{4}) \qquad (23)\\ \mu_t &= \left[I + \frac{1}{8}\triangle t^{2}\,\partial_{tt} + O(\triangle t^{4})\right]_{t=t^{n+1/2}} \qquad (24)\end{aligned}$$

where I is the identity operator. Assuming that $\triangle t = O(\triangle x) = O(h)$, is is easy to show (*exercise*) that

$$\mu_t\left[D_{xx}\,u_j^n\right] = (u_{xx})_j^{n+\frac{1}{2}} + O(h^2)$$

Putting the above results together, we are lead to the ($O(h^2)$) Crank-Nicholson approximation of (19):

$$\frac{u_j^{n+1} - u_j^{n}}{\triangle t} = \mu_t\left[D_{xx}\,u_j^n\right] \qquad (25)$$

Written out in full, this is

$$\frac{u_j^{n+1} - u_j^{n}}{\triangle t} = \frac{1}{2}\left[\frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{\triangle x^{2}} + \frac{u_{j+1}^{n} - 2u_j^{n} + u_{j-1}^{n}}{\triangle x^{2}}\right] \qquad j = 2, 3, \cdots, J-1 \qquad (26)$$

We can rewrite (26) in the form

$$a_+\,u_{j+1}^{n+1} + a_0\,u_j^{n+1} + a_-\,u_{j-1}^{n+1} = b_j \qquad j = 2, 3, \cdots, J-1 \qquad (27)$$

where

$$
\begin{aligned}
a_+ &\equiv -\frac{1}{2}\triangle x^{-2} \\
a_0 &\equiv \triangle t^{-1} + \triangle x^{-2} \\
a_- &\equiv -\frac{1}{2}\triangle x^{-2} \\
b_j &\equiv \left(\triangle t^{-1} - \triangle x^{-2}\right) u^n_j + \frac{1}{2}\triangle x^{-2}\left(u^n_{j+1} + u^n_{j-1}\right)
\end{aligned}
$$

which, along with the BCs ($u_1^{n+1} = u_J^{n+1} = 0$), is again a linear system of the form

$$\mathbf{A}\ \mathbf{u}^{n+1} = \mathbf{b}$$

for the "unknown vector" $\mathbf{u}^{n+1}$. This time, however, the matrix $\mathbf{A}$, is *not* diagonal, and the scheme is called *implicit*—i.e. the scheme *couples* unknowns at the *advanced* time level, $t = t^{n+1}$.

Note that $\mathbf{A}$ is a *tridiagonal* matrix: all elements A_{ij} for which $j \neq i+1, i$ or $i-1$ vanish. The solution of tridiagonal systems can be performed very efficiently using special purpose routines (such as `DGTSV` in `LAPACK` [6]): specifically, the operation count for solution of (26) is $O(J)$.

Also note that we can immediately write down the analogous scheme for the Schrödinger equation (3):

$$i\,\frac{\psi^{n+1}_j - \psi^n_j}{\triangle t} = -\frac{\hbar}{2m}\mu_t\left[D_{xx}\,\psi^n_j\right] + V(x_j)\,\mu_t\psi^n_j \qquad (28)$$

In this case we get a *complex* tridiagonal system, which can also be solved in $O(J)$ time, using, for example, the `LAPACK` routine `ZGTSV`.

1.5 The 1-D Wave Equation in More Detail

Recall our "standard" $O(h^2)$ discretization:

$$
\begin{aligned}
u^{n+1}_j &= 2u^n_j - u^{n-1}_j + \lambda^2\left(u^n_{j+1} - 2u^n_j + u^n_{j-1}\right), \qquad j = 2, 3, \cdots, J-1 \\
u^{n+1}_1 &= u^{n+1}_J = 0
\end{aligned}
$$

As we have discussed, to initialize the scheme, we need to specify u_j^0 and u_j^1, which is equivalent (in the limit $h \to 0$) to specifying $u(x,0)$ and $u_t(x,0)$.

Before proceeding to a discussion of a "proper initialization", let us briefly digress and consider the continuum case, and, for the sake of presentation, assume that we are considering a true IVP on an unbounded domain; i.e. we wish to solve

$$u_{tt} = u_{xx} \qquad -\infty < x < \infty \quad , \quad t \geq 0 \qquad (29)$$

As is well known, the general solution of (29) is the superposition of an arbitrary *left-moving* profile ($v = -c = -1$), and an arbitrary *right-moving* profile ($v = +c = +1$); i.e.

$$u(x,t) = \ell(x+t) + r(x-t) \qquad (30)$$

where (see Figure 5)

ℓ : constant along "left-directed" characteristics

r : constant along "right-directed" characteristics

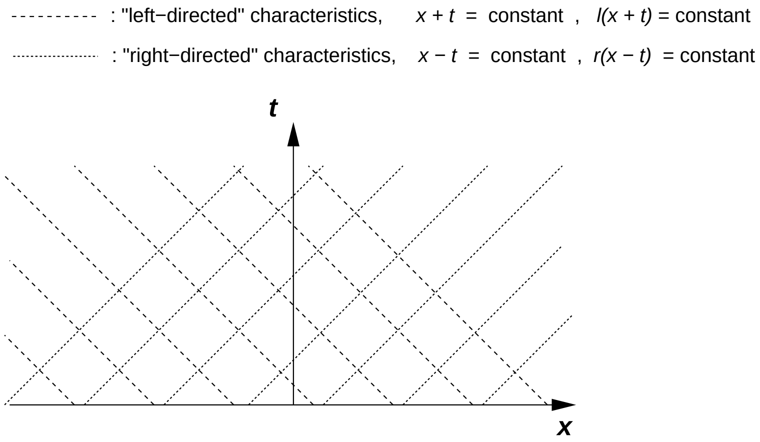
: "left−directed" characteristics, x + t = constant , l(x + t) = constant
: "right−directed" characteristics, x − t = constant , r(x − t) = constant
t
x

Figure 5: Characteristics of the wave equation: $u_{xx} = u_{tt}$. Signals (disturbances) travel along the characteristics (dashed and dotted lines.)

This observation provides us with an alternative way of specifying initial values, which is often quite convenient in practice. Rather than specifying $u(x,0)$ and $u_t(x,0)$ directly, we can specify *initial* left-moving and right-moving parts of the solution, $\ell(x)$ and $r(x)$. Specifically, we set

$$\begin{aligned} u(x,0) &= \ell(x) + r(x) && (31) \\ u_t(x,0) &= \ell'(x) - r'(x) \equiv \frac{d\ell}{dx}(x) - \frac{dr}{dx}(x) && (32) \end{aligned}$$

Returning now to the solution of the finite-differenced version of the wave equation, it is clear that given the initial data (31–32), we can trivially initialize u_j^0 with *exact* values, but that we can only approximately initialize u_j^1. The question then arises: *How accurately must we initialize the advanced values so as to ensure second order ($O(h^2)$) accuracy of the difference scheme?*

A brief, heuristic answer to this question (which can be more rigorously justified) is as follows. We have $\triangle x = O(h)$, $\triangle t = O(h)$ and the FDA is $O(h^2)$. Since the scheme is $O(h^2)$, we expect that

$$u_{\text{exact}}(x,t) - u_{\text{FD}}(x,t) = O(h^2)$$

for arbitrary, *fixed, FINITE t*. However, the number of time steps required to integrate to time t is $O(\triangle t^{-1}) = O(h^{-1})$. Thus, the per-time-step error must be $O(h^2)/O(h^{-1}) = O(h^3)$, and, therefore, we require

$$(u_{\text{FD}})_j^1 = (u_{\text{exact}})_j^1 + O(h^3)$$

We can readily accomplish this using (1) Taylor series and (2) the equation of motion to rewrite higher time derivatives in terms of spatial derivatives:

$$\begin{aligned} u_j^1 &= u_j^0 + \triangle t\, (u_t)_j^0 + \frac{1}{2}\triangle t^2\, (u_{tt})_j^0 + O(\triangle t^3) && (33) \\ &= u_j^0 + \triangle t\, (u_t) + \frac{1}{2}\triangle t^2\, (u_{xx})_j^0 + O(\triangle t^3) && (34) \end{aligned}$$

which, using results from above, can be written as

$$u_j^1 = (\ell + r)_j + \triangle t\, (\ell' - r')_j + \frac{1}{2}\triangle t^2\, (\ell'' + r'')_j \qquad (35)$$

1.6 Stability Analysis

One of the most frustrating—yet fascinating—features of FD solutions of time dependent problems, is that the discrete solutions often "blow up"—e.g. floating-point overflows are generated at some point in the evolution. Although "blow-ups" can sometimes be caused by legitimate (!) "bugs"—i.e. an incorrect implementation—at other times it is simply the *nature of the FD scheme* which causes problems. We are thus lead to consider the *stability* of solutions of difference equations (as well as their differential-equation progenitors).

Let us again consider the 1-d wave equation (15) and let us now remark that this is a *linear, non-dispersive* wave equation, a consequence of which is the fact that the "size" of the solution does *not* change with time:

$$\|u(x,t)\| \sim \|u(x,0)\| \,, \tag{36}$$

where $\|\cdot\|$ is an suitable norm, such as the L_2 norm:

$$\|u(x,t)\| \equiv \left(\int_0^1 u(x,t)^2 \, dx \right)^{1/2} \,. \tag{37}$$

We will use the property captured by (36) as our working definition of stability. In particular, if you believe (36) is true for the wave equation, then you believe the wave equation is stable.

Fundamentally, if our FDA approximation *converges*, then we expect the same behaviour for the difference solution:

$$\|u^n_j\| \sim \|u^0_j\| \,. \tag{38}$$

Now, we construct our FD solution by *iterating in time*, generating

$$u^0_j ,\ u^1_j ,\ u^2_j ,\ u^3_j ,\ u^4_j ,\ \cdots$$

in succession, using the FD equation

$$u^{n+1}_j = 2u^n_j - u^{n-1}_j + \lambda^2 \left(u^n_{j+1} - 2u^n_j + u^n_{j-1} \right) \,.$$

As it turns out, we are *not* guaranteed that (38) holds for all values of $\lambda \equiv \triangle t \,/\, \triangle x$. In fact, for certain λ (all $\lambda > 1$, as we shall see), we have

$$\|u^n_j\| \gg \|u^0_j\| \,,$$

and for those λ, $\|u^n\|$ *diverges* from u, even (especially!) as $h \to 0$—that is, the difference scheme is *unstable.*

In fact, for many wave problems (including all linear problems), given that a FD scheme is *consistent* (i.e. so that $\hat{\tau} \to 0$ as $h \to 0$), stability is the necessary and sufficient condition for convergence (Lax's theorem).

1.6.1 Heuristic Stability Analysis

Let us write a general time-dependent FDA in the form

$$\mathbf{u}^{n+1} = \mathbf{G}[\mathbf{u}^n] \,, \tag{39}$$

where $\mathbf{G}$ is some *update operator* (linear in our example problem), and $\mathbf{u}$ is a column vector containing sufficient unknowns to write the problem in first-order-in-time form. For example, if we introduce a new, auxiliary set of unknowns, v^n_j, defined by

$$v^n_j = u^{n-1}_j \,,$$

then we can rewrite the differenced-wave-equation (16) as

$$\begin{aligned} u^{n+1}_j &= 2u^n_j - v^n_j + \lambda^2 \left(u^n_{j+1} - 2u^n_j + u^n_{j-1} \right) \,, & (40) \\ v^{n+1}_j &= u^n_j \,, & (41) \end{aligned}$$

so with

$$\mathbf{u}^n = [u_1^n, v_1^n,\ u_2^n, v_2^n,\ \cdots\ u_J^n, v_J^n]\,,$$

(for example), (40-41) is clearly of the form (39). Equation (39) provides us with a compact way of describing the solution of the FDA. Given initial data, $\mathbf{u}^0$, the solution after n time-steps is

$$\mathbf{u}^n = \mathbf{G}^n \mathbf{u}^0\,, \tag{42}$$

where $\mathbf{G}^n$ is the n-th power of the matrix $\mathbf{G}$. Now, assume that $\mathbf{G}$ has a complete set of orthonormal eigenvectors

$$\mathbf{e}_k, \quad k = 1, 2, \cdots J\,,$$

and corresponding eigenvalues

$$\mu_k, \quad k = 1, 2, \cdots J\,,$$

so that

$$\mathbf{G}\,\mathbf{e}_k = \mu_k\,\mathbf{e}_k, \quad k = 1, 2, \cdots J\,.$$

We can then write the initial data as (spectral decomposition):

$$\mathbf{u}^0 = \sum_{k=1}^{J} c_k^0\,\mathbf{e}_k\,,$$

where the c_k^0 are coefficients. Using (42), the solution at time-step n is then

$$\begin{aligned} \mathbf{u}^n &= \mathbf{G}^n \left(\sum_{k=1}^{J} c_k^0\,\mathbf{e}_k \right) &(43)\\ &= \sum_{k=1}^{J} c_k^0\,(\mu_k)^n\,\mathbf{e}_k\,. &(44) \end{aligned}$$

Clearly, if the difference scheme is to be stable, we must have

$$|\mu_k| \le 1 \qquad k = 1, 2, \cdots J \tag{45}$$

(Note: μ_k will be complex in general, so $|\mu|$ denotes complex modulus, $|\mu| \equiv \sqrt{\mu\mu^\star}$).

Geometrically, then, the eigenvalues of the update matrix must lie on or within the unit circle (see Figure 6).

1.6.2 Von-Neumann (Fourier) Stability Analysis

Von-Neumann stability analysis is based on the ideas sketched above, but additionally assumes that the difference equation is linear with constant coefficients, and that the boundary conditions are periodic. We can then use Fourier analysis, which has the same benefits in the discrete domain—difference operators in real-space variable $x \longrightarrow$ algebraic operations in Fourier-space variable k—as it does in the continuum Schematically, instead of writing

$$\mathbf{u}^{n+1}(x) = \mathbf{G}[\mathbf{u}^n(x)]\,,$$

we consider the Fourier-domain equivalent:

$$\tilde{\mathbf{u}}^{n+1}(k) = \tilde{\mathbf{G}}[\tilde{\mathbf{u}}^n(k)]\,,$$

where k is the wave-number (Fourier-space variable) and $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{G}}$ are the Fourier-transforms of $\mathbf{u}$ and $\mathbf{G}$, respectively. Specifically, we define the Fourier-transformed grid function via

$$\tilde{\mathbf{u}}^n(k) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{-ikx}\,\mathbf{u}^n(x)\,dx\,. \tag{46}$$

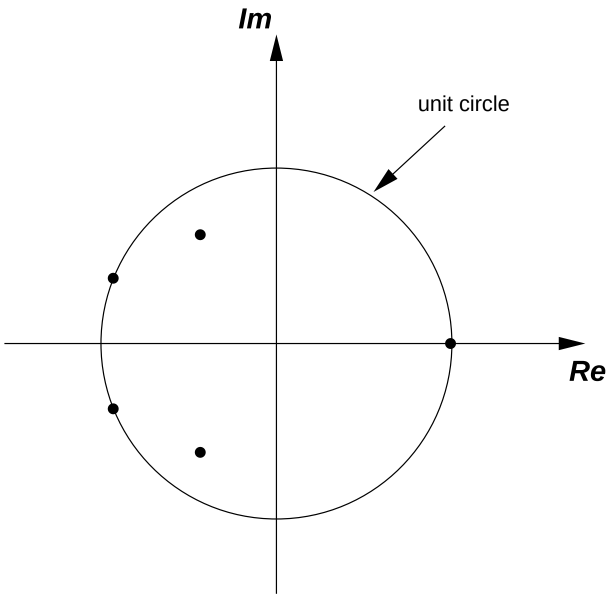

Figure 6: Schematic illustration of location in complex plane of eigenvalues of update matrix **G**. In this case, all eigenvalues (dots) lie on or within the unit circle, indicating that the corresponding finite difference scheme is stable.

For a general difference scheme, we will find that

$$\tilde{\mathbf{u}}^{n+1}(k) = \tilde{\mathbf{G}}(\xi)\,\tilde{\mathbf{u}}^{n}(k)\,,$$

where $\xi \equiv kh$, and we will have to show that $\tilde{\mathbf{G}}(\xi)$'s eigenvalues lie within or on the unit circle for all conceivable ξ. The appropriate range for ξ is

$$-\pi \le \xi \le \pi\,,$$

since the shortest wavelength representable on a uniform mesh with spacing h is $\lambda = 2h$ (Nyquist limit), corresponding to a maximum wave number $k = (2\pi)/\lambda = \pm\pi/h$.

Let us consider the application of the Von-Neumann stability analysis to our current model problem. We first define a (non-divided) difference operator D^2 as follows:

$$D^2\,u(x) = u(x+h) - 2u(x) + u(x-h)\,.$$

Then, suppressing the spatial grid index, we can write the first-order form of the difference equation (40-41) as

$$\begin{aligned} u^{n+1} &= 2u^{n} - v^{n} + \lambda^2\,D^2\,u^{n}\,, \\ v^{n+1} &= u^{n}\,, \end{aligned}$$

or

$$\begin{bmatrix} u \\ v \end{bmatrix}^{n+1} = \begin{bmatrix} 2+\lambda^2\,D^2 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}^{n}. \tag{47}$$

In order to perform the Fourier transform, we need to know the action of D^2 in Fourier-space. Using the transform inverse to (46) we have

$$u(x) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{+\infty} e^{ikx}\,\tilde{u}(k)\,dk\,,$$

so

$$\begin{aligned} D^2\,u(x) = u(x+h) - 2u(x) + u(x-h) &= \int_{-\infty}^{+\infty} \left(e^{ikh} - 2 + e^{-ikh}\right) e^{ikx}\,\tilde{u}(k)\,dk \\ &= \int_{-\infty}^{+\infty} \left(e^{i\xi} - 2 + e^{-i\xi}\right) e^{ikx}\,\tilde{u}(k)\,dk\,. \end{aligned}$$

Now consider the quantity $-4\sin^2(\xi/2)$:

$$\begin{aligned} -4\sin^2\frac{\xi}{2} &= -4\left(\frac{e^{i\xi/2} - e^{-i\xi/2}}{2i}\right)^2 \\ &= \left(e^{i\xi/2} - e^{-i\xi/2}\right)^2 = e^{i\xi} - 2 + e^{-i\xi}\,, \end{aligned}$$

so

$$D^2\,u(x) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{+\infty}\left(-4\sin^2\frac{\xi}{2}\right) e^{ikx}\,\tilde{u}(k)\,dk\,.$$

In summary, under Fourier transformation, we have

$$\begin{aligned} \mathbf{u}(x) &\longrightarrow \tilde{\mathbf{u}}(k)\,, \\ D^2\mathbf{u}(x) &\longrightarrow -4\sin^2\frac{\xi}{2}\tilde{\mathbf{u}}(k)\,. \end{aligned}$$

Using this result in the Fourier transform of (47), we see that we need to compute the eigenvalues of

$$\left[\begin{array}{cc} 2 - 4\lambda^2\sin^2(\xi/2) & -1 \\ 1 & 0 \end{array}\right],$$

and determine the conditions under which the eigenvalues lie on or within the unit circle. The characteristic equation (whose roots are the eigenvalues) is

$$\left|\begin{array}{cc} 2 - 4\lambda^2\sin^2(\xi/2) - \mu & -1 \\ 1 & -\mu\,, \end{array}\right| = 0$$

or

$$\mu^2 + \left(4\lambda^2\sin^2\frac{\xi}{2} - 2\right)\mu + 1 = 0\,.$$

This equation has roots

$$\mu(\xi) = \left(1 - 2\lambda^2\sin^2\frac{\xi}{2}\right) \pm \left(\left(1 - 2\lambda^2\sin^2\frac{\xi}{2}\right) - 1\right)^{1/2}\,.$$

We now need to find sufficient conditions for

$$|\mu(\xi)| \le 1,$$

or equivalently

$$|\mu(\xi)|^2 \le 1.$$

To this end, we note that we can write

$$\mu(\xi) = (1 - Q) \pm ((1-Q)^2 - 1)^{1/2}\,,$$

where the quantity, Q

$$Q \equiv 2\lambda\sin^2\frac{\xi}{2}\,,$$

is *real* and *non-negative* ($Q \ge 0$). There are now two cases to consider:

1. $(1-Q)^2 - 1 \le 0$,
2. $(1-Q)^2 - 1 > 0$.

In the first case, $((1-Q)^2-1)^{1/2}$ is purely imaginary, so we have

$$|\mu(\xi)|^2 = (1-Q)^2 + (1-(1-Q)^2) = 1\,.$$

In the second case, $(1-Q)^2 - 1 > 0 \longrightarrow (1-Q)^2 > 1 \longrightarrow Q > 2$, and then we have

$$1 - Q - ((1-Q^2)-1)^{1/2} \ < \ -1\,,$$

so, in this case, our stability criterion will *always* be violated. We thus conclude that a necessary condition for Von-Neumann stability is

$$(1-Q)^2 - 1 \le 0 \longrightarrow (1-Q)^2 \le 1 \longrightarrow Q \le 2\,.$$

Since $Q \equiv 2\lambda \sin^2(\xi/2)$ and $\sin^2(\xi/2) \le 1$, we must therefore have

$$\lambda \equiv \frac{\triangle t}{\triangle x} \le 1\,,$$

for stability of our scheme (16). This condition is often called the CFL condition—after Courant, Friedrichs and Lewy who derived it in 1928 (the ratio $\lambda = \triangle x \,/\, \triangle t$ is also frequently called the *Courant number*). In practical terms, we must limit the time-discretization scale , $\triangle t$, to values no larger than the space-discretization scale, $\triangle x$. Furthermore, this type of instability has a "physical" interpretation, often summarized by the statement *the numerical domain of dependence of an explicit difference scheme must contain the physical domain of dependence.*

1.7 Dispersion and Dissipation

Let us now consider an even simpler model "wave equation" than (1), the so-called *advection*, or *color* equation:

$$\begin{aligned} u_t &= a\,u_x \quad (a>0) \qquad -\infty < x < \infty \quad , \quad t \ge 0 \\ u(x,0) &= u_0(x) \end{aligned} \tag{48}$$

which has the exact solution

$$u(x,t) = u_0(x+at) \tag{49}$$

Equation (48) is another example of a non-disspative, non-dispersive partial differential equation.

To remind ourselves what "non-dispersive" means, let us note that (48) admits "normal mode" solutions:

$$u(x,t) \sim e^{ik(x+at)} \equiv e^{i(kx+\omega t)}$$

where $\omega \equiv ka$; in general, of course, $\omega \equiv \omega(k)$ is known as the *dispersion relation*, and

$$\frac{d\omega}{dk} \ \equiv \ \text{speed of propagation of mode with wave number } k$$

In the current case, we have

$$\frac{d\omega}{dk} = a = \text{constant}$$

which means that all modes propagate at the same speed, which is precisely what is meant by "non-dispersive". Further, if we consider resolving the general initial profile, $u_0(x)$ into "normal-mode" (Fourier) components, we find that the magnitudes of the components are preserved in time, i.e. equation (48) is also *non-dissipative.*

Ostensibly, we would like our finite-difference solutions to have the same properties—i.e. to be dissipationless and dispersionless, but, in general, this will not be (completely) possible. We will return to the issue of dissipation and dispersion in FDAs of wave problems below.

1.8 The Leap-Frog Scheme

First note that (48) is a good prototype for the general hyperbolic *system*:

$$\mathbf{u}_t = \mathbf{A}\mathbf{u}_x \tag{50}$$

where $\mathbf{u}$(x,t) is the n-component *solution vector*:

$$\mathbf{u}(x,t) = [u_1(x,t),\, u_2(x,t),\, \cdots\, u_n(x,t)] \tag{51}$$

and the $n \times n$ matrix $\mathbf{A}$ has distinct real eigenvalues

$$\lambda_1,\, \lambda_2,\, \cdots\, \lambda_n$$

so that, for example, there exists a similarity transformation $\mathbf{S}$ such that

$$\mathbf{S}\mathbf{A}\mathbf{S}^{-1} = \mathrm{diag}(\lambda_1,\, \lambda_2,\, \cdots\, \lambda_n)$$

The leap-frog scheme is a commonly used finite-difference approximation for hyperbolic systems. In the context of our simple scalar ($n = 1$) advection problem (48):

$$u_t = a\, u_x$$

an appropriate stencil is shown in Figure 7. Applying the usual $O(h^2)$ approximations to ∂_x and ∂_t, our leap-frog (LF) scheme is

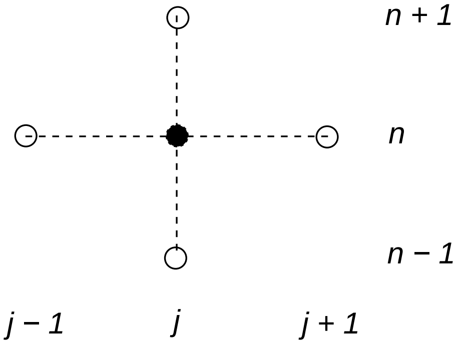

Figure 7: Stencil (molecule/star) for leap-frog scheme as applied to (48). Note that the central grid point has been filled in this figure to emphasize that the corresponding unknown, u^n_j, does not appear in the local discrete equation at that grid point (hence the term "leap-frog")

$$\frac{u^{n+1}_j - u^{n-1}_j}{2\,\triangle t} = a\,\frac{u^n_{j+1} - u^n_{j-1}}{2\,\triangle x} \tag{52}$$

or explicitly

$$u^{n+1}_j = u^{n-1}_j + a\lambda\left(u^n_{j+1} - u^n_{j-1}\right) \tag{53}$$

where

$$\lambda \equiv \frac{\triangle t}{\triangle x}$$

is the *Courant number* as previously.

Exercise: Perform a von Neumann stability analysis of (52) thus showing that $a\lambda \le 1$ (which, you should note, is just the CFL condition) is necessary for stability.

Observe that the LF scheme (52) is a *three-level* scheme. As in our treatment of the wave equation, $u_{tt} = u_{xx}$ using the "standard scheme", we need to specify

$$u_j^0 \quad , \quad u_j^1 \qquad j = 1, 2, \cdots J$$

to "get the scheme going"—that is, we need to specify *two* numbers per spatial grid point. This should be contrasted to the continuum where we need to specify only *one* number per x_j, namely $u_0(x_j)$. Again, the initialization of the u_j^0 is trivial, given the (continuum) initial data $u_0(x)$, and, again, we need u_j^1 to $O(\triangle t^3) = O(h^3)$ accuracy for $O(h^2)$ global accuracy. Two possible approaches are as follows.

Taylor Series: The development here is parallel to that for the wave equation. We have

$$u_j^1 = u_j^0 + \triangle t \, (u_t)_j^0 + \frac{1}{2} \triangle t^2 \, (u_{tt})_j^0 + O(\triangle t^2)$$

also, from the equation of motion $u_t = a u_x$, we get

$$u_{tt} = (u_t)_t = (a u_x)_t = a \, (u_t)_x = a^2 u_{xx}.$$

so we have our desired initialization formula:

$$u_j^1 = u_j^0 + \triangle t \, (u_0')_j^0 + \frac{1}{2} \triangle t^2 \left(a^2 u_0''\right)_j^0 + O(\triangle t^3) \tag{54}$$

Self-Consistent Iterative Approach: The idea here is to initialize the u_j^1 from the u_j^0 and a version of the discrete equations of motion which introduces a "ficticious" half-time-level—see Figure 8.

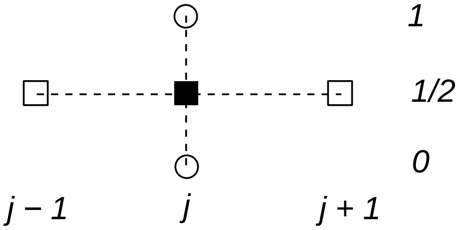

Figure 8: Stencil for initialization of leap-frog scheme for to (48). Note the introduction of the "fictitious" half-time level $t = t^{1/2}$ (squares).

Applying the leap-frog scheme on the stencil in the Figure, we have

$$\frac{u_j^1 - u_j^0}{\triangle t} = a \, \frac{u_{j+1}^{\frac{1}{2}} - u_{j-1}^{\frac{1}{2}}}{2 \triangle x}$$

or, explicitly solving for u_j^1:

$$u_j^1 = u_j^0 + \frac{1}{2} \lambda \left(u_{j+1}^{\frac{1}{2}} - u_{j-1}^{\frac{1}{2}} \right)$$

It is a straightforward exercise to show that in order to retain $O(h^2)$ accuracy of the difference scheme, we need "fictitious-time" values, $u_j^{1/2}$ which are accurate to $O(h^2)$ (i.e. we can neglect terms which are of $O(h^2)$). In particular, if we *define* $u_j^{1/2}$, via

$$u_j^{\frac{1}{2}} = \frac{u_j^1 + u_j^0}{2}$$

which amounts to defining the half-time values via linear interpolation in the advanced and retarded unknowns, we will retain second-order accuracy.

We are thus led to the following initialization algorithm which is perhaps best expressed in pseudo-code (note, all loops over `j` are implicit:)

```
u[0,j] := u_0(x_j)
u[1,j] := u_0(x_j)
DO
   usave[j] := u[1,j]
   u[1/2,j] := (u[1,j] + u[0,j]) / 2

   u[1,j]   := u[0,j] + (lambda / 2) * (u[1/2,j+1] - u[1/2,j-1])

UNTIL  norm(usave[j] - u[1,j]) < epsilon
```

1.9 Error Analysis and Convergence Tests

As a side remark, we note that the discussion in this section applies to essentially *any* continuum problem which is solved using FDAs on a *uniform* mesh structure. In particular, the discussion applies to the treatment of ODEs and elliptic problems, where, in fact, convergence is often easier to achieve due to the fact that the FDAs are typically intrinsically stable (i.e. we have an easier time constructing stable FDAs for these types of problems). We also note that departures from non-uniformity in the mesh do not, in general, complete destroy the picture, but, rather, tend to distort it in ways which are beyond the scope of these notes. In any case, my colleagues have been known to paraphrase my whole approach to this subject as

Convergence!, Convergence!, Convergence!

1.9.1 Sample Analysis: The Advection Equation

We again consider the solution of the advection equation, but this time we impose periodic boundary conditions on our spatial domain, which we take to be $0 \le x \le 1$ with $x = 0$ and $x = 1$ identified (i.e. we solve the wave equation on $\mathbf{S}^1 \times \mathbf{R}$):

$$\begin{aligned} u_t &= a\, u_x \quad (a > 0) \qquad 0 \le x \le 1, \quad t \ge 0 \\ u(x,0) &= u_0(x) \end{aligned} \tag{55}$$

Note that the initial conditions $u_0(x)$ must be compatible with periodicity, i.e. we must specify periodic initial data.

Again, given the initial data, $u_0(x)$, we can immediately write down the full solution

$$u(x,t) = u_0(x + a\,t \bmod 1) \tag{56}$$

where mod is the usual modulus function which "wraps" $x + a\ t$, $t > 0$ onto the unit circle. As we shall see, because of the simplicity and solubility of this problem, one can perform a rather complete "analytic" (i.e. closed form) treatment of the convergence of simple FDAs of (55). The point of the exercise, however, is *not* to advocate parallel "analytic" treatments for more complicated problems. Rather, the key idea to be extracted from the following is that, in principle (always), and in practice (almost always, i.e. I've never seen a case where it *didn't* work, but then there's a lot of computations I haven't seen):

> *The error, e^h, of an FDA is no less computable than the solution, u^h itself.*

This has widespread ramifications, one of which is that there's really no excuse for publishing solutions of FDAs without error bars, or their equivalents!

Proceeding with our sample error analysis of the leap-frog scheme applied to the advection equation, we first introduce some difference operators for the usual $O(h^2)$ centred approximations of ∂_x and ∂_t:

$$D_x\, u^n_j \equiv \frac{u^n_{j+1} - u^n_{j-1}}{2\,\triangle x} \tag{57}$$

$$D_t\, u^n_j \equiv \frac{u^{n+1}_j - u^{n-1}_j}{2\,\triangle t} \tag{58}$$

We again take

$$\triangle x \equiv h \qquad \triangle t \equiv \lambda \, \triangle x = \lambda \, h$$

and will hold λ fixed as h varies, so that, as usual, our FDA is characterized by the single scale parameter, h.

The idea behind our error analysis is that we want to view the solution of the FDA as a *continuum* problem, and hence we will express both the difference operators and the FDA solution as asymptotic series (in h) of differential operators, and continuum functions, respectively. We have the following expansions for D_x and D_t:

$$\begin{aligned} D_x &= \partial_x + \frac{1}{6} h^2 \, \partial_{xxx} + O(h^4) && (59) \\ D_t &= \partial_t + \frac{1}{6} \lambda^2 h^2 \, \partial_{ttt} + O(h^4) && (60) \end{aligned}$$

Now, in terms of the general, abstract formulation of (1.3), we have:

$$\begin{aligned} L\,u - f = 0 \quad &\Longleftrightarrow \quad (\partial_t - a\,\partial_x)\,u = 0 && (61) \\ L^h u^h - f^h = 0 \quad &\Longleftrightarrow \quad (D_t - a\,D_x)\,u^h = 0 && (62) \\ L^h u - f^h \equiv \tau^h \quad &\Longleftrightarrow \quad (D_t - a\,D_x)\,u \equiv \tau^h = \frac{1}{6} h^2 \left(\lambda^2 \partial_{ttt} - a\,\partial_{xxx}\right) u + O(h^4) = O(h^2) && (63) \end{aligned}$$

The Richardson ansatz: The key to our analysis is L.F. Richardson's old observation (*ansatz*) [5], that the solution, u^h, of *any* FDA which (1) uses a uniform mesh structure with scale parameter h, and (2) is completely centred, should have the following expansion in the limit $h \to 0$:

$$u^h(x,t) = u(x,t) + h^2 e_2(x,t) + h^4 e_4(x,t) + \cdots \qquad (64)$$

Here u is the continuum solution, while e_2, e_4, $\cdots$ are (continuum) *error functions* which *do not depend on* h. In a very real sense (64), is *the* key expression from which all error analysis of FDAs derives. We note that in the case that the FDA is *not* completely centred, we will have to modify the *ansatz*. In particular, for first order schemes (which are more common in relativistic astrophysics than one might expect!), we will have

$$u^h(x,t) = u(x,t) + he_1(x,t) + h^2 e_x(x,t) + h^3 e_3(x,t) + \cdots \qquad (65)$$

Note that the Richardson *ansatz* (64) is completely compatible with the assertion discussed in (1.3.7), namely that

$$\tau^h = O(h^2) \quad \longrightarrow \quad e^h \equiv u - u^h = O(h^2) \qquad (66)$$

However, the Richardson form (64) contains much more information than "second-order truncation error should imply second-order solution error", telling us the precise form of the h dependence of u^h.

Given the Richardson expansion, we can now proceed with our error analysis. We start from the FDA, $L^h u^h - f^h = 0$, and replace both L^h and u^h with continuum expansions:

$$\begin{aligned} L^h u^h = 0 \quad &\longrightarrow \quad (D_t - a\,D_x)\left(u + h^2 e_2 + \cdots\right) = 0 \\ &\longrightarrow \quad \left(\partial_t + \frac{1}{6}\lambda^2 h^2 \partial_{ttt} - a\,\partial_x - \frac{1}{6} a h^2 \, \partial_{xxx} + \cdots\right)\left(u + h^2 e_2 + \cdots\right) = 0 && (67) \end{aligned}$$

We now demand that terms in (67) vanish order-by-order in h. At $O(1)$ (zeroth-order), we have

$$(\partial_t - a\,\partial_x)\,u = 0 \qquad (68)$$

which is simply a statement of the *consistency* of the difference approximation. More interestingly, at $O(h^2)$ (second-order), we find

$$(\partial_t - a\,\partial_x)\,e_2 = \frac{1}{6}\left(a\partial_{xxx} - \lambda^2 \partial_{ttt}\right) u \qquad (69)$$

which, assuming that we view u as a "known" function, is simply a PDE for the leading order error function, e_2. Moreover, the PDE governing e_2 is of *precisely* the same nature as the original PDE (48).

In fact, we can *solve* (69) for e_2. Given the "natural" initial conditions

$$e_2(x, 0) = 0$$

(i.e. we initialize the FDA with the exact solution so that $u^h = u$ at $t = 0$), and defining $q(x + at)$:

$$q(x + at) \equiv \frac{1}{6} a \left(1 - \lambda^2 a^2\right) \partial_{xxx} u(x, t)$$

we have

$$e_2(x, t) = t\, q(x + at \bmod 1) \tag{70}$$

We note that, as is typical for leap-frog, we have *linear* growth of the finite difference error with time (to leading order in h). We also note that we can obviously push this analysis to higher order in h—what results, then, is an entire *hierarchy* of differential equations for u and the error functions e_2, e_4, e_6, $\cdots$. Indeed, it is useful to keep this view in mind:

> When one solves an FDA of a PDE, one is *not* solving some system which is "simplified" realtive to the PDE, rather, one is solving a much *richer* system consisting of an (infinite) hierarchy of PDEs, one for each function appearing in the Richardson expansion (64).

In the general case, of course, we will not be able to solve the PDE governing u, let alone that governing e_2—otherwise we wouldn't be considering the FDA in the first place! But it is precisely in this instance where the true power of Richardson's observation is evident. The key observation is that starting from (64), and computing FD solutions using the same initial data, but with differing values of h, we can learn a great deal about the error in our FD approximations. The whole game of investigating the manner in which a particular FDA converges or doesn't (i.e. looking at what happens as one varies h) is known as *convergence testing*. It is important to realize that there are no hard and fast rules for convergence testing; rather, one tends to tailor the tests to the specifics of the problem at hand, and, being largely an empirical approach, one gains experience and intuition as one works through more and more problems. That said, I emphasize again that the Richardson expansion, in some form or other, *always* underlies convergence analysis of FDAs.

A simple example of a convergence test, and the one I use most often in practice is the following. We compute three distinct FD solutions u^h, u^{2h}, u^{4h} at resolutions h, $2h$ and $4h$ respectively, but using the same initial data (as naturally expressed on the 3 distinct FD meshes). We also assume that the finite difference meshes "line up", i.e. that the $4h$ grid points are a subset of the $2h$ points which are a subset of the h points, so that, in particular, the $4h$ points constitute a common set of events (x_j, t^n) at which specific grid function values can be directly (i.e. no interpolation required) and meaningfully compared to one another. From the Richardson *ansatz* (64), we expect:

$$\begin{aligned}
u^h &= u + h^2 e_2 + h^4 e_4 + \cdots \\
u^{2h} &= u + (2h)^2 e_2 + (2h)^4 e_4 + \cdots \\
u^{4h} &= u + (4h)^2 e_2 + (4h)^4 e_4 + \cdots
\end{aligned}$$

We then compute a quantity $Q(t)$, which I will call a *convergence factor*, as follows:

$$Q(t) \equiv \frac{\|u^{4h} - u^{2h}\|_x}{\|u^{2h} - u^h\|_x} \tag{71}$$

where $\|\cdot\|_x$ is any suitable discrete spatial norm, such as the ℓ_2 norm, $\|\cdot\|_2$:

$$\|u^h\|_2 = \left(J^{-1} \sum_{j=1}^{J} \left(u_j^h\right)^2 \right)^{1/2} \tag{72}$$

and, for concreteness, the subtractions in (71) can be taken to involve the sets of mesh points which are common between u^{4h} and u^{2h}, and between u^{2h} and u^h. It is a simple exercise to show that, if our finite difference scheme is converging, then we should find:

$$\lim_{h\to 0} Q(t) = 4. \tag{73}$$

In practice, one can use additional levels of discretization, $8h$, $16h$, etc. to extend this test to look for "trends" in $Q(t)$ and, in short, to convince oneself (and, with luck, others), that the FDA really *is* converging. Moreover, once convergence of an FDA has been established, then a point-wise subtraction of any two solutions computed at different resolutions, will immediately provide an estimate of the level of error in both. For example, if we have u^h and u^{2h}, then, again by the Richardson *ansatz* we have

$$u^{2h} - u^h = \left(\left(u + (2h)^2 e_2 + \cdots\right) - \left(u + h^2 e_2 + \cdots\right)\right) = 3h^2 e_2 + O(h^4) \sim 3e^h \sim \frac{3}{4}e^{2h} \tag{74}$$

Richardson extrapolation: Richardson's observation (64) also provides the basis for all the techniques of *Richardson extrapolation*, where solutions computed at different resolutions are linearly combined so as to *eliminate* leading order error terms, and hence provide more accurate solutions. As an example, given u^h and u^{2h} which satisfy (64), we can take the linear combination, $\bar{u}^h$:

$$\bar{u}^h \equiv \frac{4u^h - u^{2h}}{3} \tag{75}$$

which, by (64), is easily seen to be $O(h^4)$, i.e. *fourth*-order accurate!

$$\begin{aligned} \bar{u}^h &\equiv \frac{4u^h - u^{2h}}{3} = \frac{4\left(u + h^2 e_2 + h^4 e_4 + \cdots\right) - \left(u + 4h^2 e_2 + 16h^4 e_4 + \cdots\right)}{3} \\ &= -4h^4 e_4 + O(h^6) = O(h^4) \end{aligned} \tag{76}$$

When it works, Richardson extrapolation has an almost magical quality about it, but one generally has to start with fairly accurate (on the order of a few %) solutions in order to see the dramatic improvement in accuracy suggested by (76). Partly because it is still a struggle to achieve that sort of accuracy (i.e. a few %) for *any* computation in many areas of numerical relativity/astrophysics, techniques based on Richardson extrapolation have not had a major impact in this context.

Independent Residual Evaluation A question which often arises in discussions of convergence testing is the following:

> "OK, you've established that u^h is converging as $h \to 0$, but how do you know you're converging to u, the solution of the continuum problem?"

Here, the notion of an independent residual evaluation is very useful. The idea is as follows: we have our continuum PDE

$$Lu - f = 0 \tag{77}$$

and our FDA

$$L^h u^h - f^h = 0 \tag{78}$$

We have demonstrated that u^h is apparently converging by, for example, computing the convergence factor (71) and verifying that it tends to 4 as h tends to 0. However, we do not know if we have derived and/or implemented our discrete operator L^h correctly. Note that implicit in the "implementation" is the fact that, particularly for multi-dimensional and/or implicit and/or multi-component FDAs, considerable "work" (i.e. analysis and coding) may be involved in setting up and solving the algebraic equations for u^h. As a check that we *are* converging to u, we consider a *distinct* (i.e. independent) discretization of the PDE:

$$\tilde{L}^h \tilde{u}^h - f^h = 0 \tag{79}$$

The only thing we need from this FDA for the purposes of the independent residual test is the new FD operator $\tilde{L}^h$. As with L^h, we can expand $\tilde{L}^h$ in powers of the mesh spacing:

$$\tilde{L}^h = L + h^2 E_2 + h^4 E_4 + \cdots \tag{80}$$

where E_2, E_4, $\cdots$ are higher order (involve higher order derivatives than L) differential operators. We then simply apply the new operator $\tilde{L}^h$ to our FDA u^h and investigate what happens as $h \to 0$. If u^h *is* converging to the continuum solution, u, we will have

$$u^h = u + h^2 e_2 + O(h^4) \tag{81}$$

and we will compute

$$\tilde{L}^h u^h = \left(L + h^2 E_2 + O(h^4)\right)\left(u + h^2 e_2 + O(h^4)\right) = Lu + h^2 (E_2\, u + L\, e_2) = O(h^2) \tag{82}$$

i.e., $\tilde{L}^h u^h$ will be a residual-like quantity which converges quadratically as $h \to 0$. Conversely, if we have goofed in our derivation and/or implementation of $L^h u^h = f^h = 0$, but we still see convergence; i.e. we have, for example, $u^{2h} - u^h \to 0$ as $h \to 0$, then we must have something like

$$u^h = u + e_0 + h e_1 + h^2 e_2 + \cdots \tag{83}$$

where the crucial fact is that the error must have an $O(1)$ component, e_0. In this case, we will compute

$$\tilde{L}^h u^h = \left(L + h^2 E_2 + O(h^4)\right)\left(u + e_0 + h e_1 + h^2 e_2 + O(h^4)\right) = Lu + Le_0 + hLe_1 + O(h^2) = Le_0 + O(h) \tag{84}$$

and, unless we are *extraordinarily* lucky, and $L\, e_0$ vanishes, we will *not* observe the expected convergence, rather, we will see $\tilde{L}^h u^h - f^h$ tending to a *finite* ($O(1)$) value—a sure sign that something is wrong.

There is of course, the problem that we might have slipped up in our implementation of the "independent residual evaluator", $\tilde{L}^h$, in which case the results from our test will be ambigous at best! However, a key point here is that because $\tilde{L}^h$ is only used *a posteriori* on a computed solution (we never use it to compute $\tilde{u}^h$, for example) it is a relatively easy matter to ensure that $\tilde{L}^h$ has been implemented in an error-free fashion (perhaps using symbolic manipulation facilities). Furthermore, many of the restrictions commonly placed on the "real" discretization (such as stability and the ease of solution of the resulting algebraic equations) do not apply to $\tilde{L}^h$.

1.10 Dispersion and Dissipation in FDAs

We again consider the advection model problem, $u_t = a\, u_x$, but now discretize only in space (semi-discretization) using the usual $O(h^2)$ centred difference approximation:

$$u_t = a\, D_x\, u \equiv a\, \frac{u_{j+1} - u_{j-1}}{2\, \triangle x} \tag{85}$$

We now look for normal-mode solutions to (85) of the form

$$u = e^{ik\,(x + a'\, t)}$$

where the "discrete phase speed", a', is to be determined. Substitution of this *ansatz* in (85) yields

$$ika'u = \frac{a\,(2i \sin(k \triangle x\,))}{2\,\triangle x} u$$

or, solving for the discrete phase speed, a'

$$a' = a\frac{\sin(k \triangle x\,)}{k\,\triangle x} = a\frac{\sin \xi}{\xi}$$

where we have defined the dimensionless wave number, ξ:

$$\xi \equiv k \triangle x$$

In the *low frequency* limit, $\xi \to 0$, we have the expected result:

$$a' = a\frac{\sin\xi}{\xi} \to a$$

so that low frequency components propagate with the correct phase speed, a. However, in the *high frequency* limit, $\xi \to \pi$, we have

$$a' = a\frac{\sin\xi}{\xi} \to 0 \quad !!$$

i.e. the highest frequency components of the solution don't propagate at all! This is typical of FDAs of wave equations, particularly for relatively low-order schemes. The propagation of high frequency components of the difference solution is essentially completely wrong. Arguably then, there can be little harm in attenuating (dissipating) these components, and, in fact, since high frequency components are potentially troublesome (particularly *vis a vis* non-linearities and the treatment of boundaries), it is often *advantageous* to use a dissipative difference scheme.

Some FDAs are naturally dissipative (the Lax-Wendroff scheme, for example), while others, such as leap-frog, are not. In the case of a leap-frog-based scheme, the idea is to add dissipative terms to the method, but in such a way as to retain $O(h^2)$ accuracy of the scheme. Consider, for example, the leap-frog scheme as applied to the advection model problem:

$$u^{n+1}_{j} = u^{n-1}_{j} + a\lambda\left(u^{n}_{j+1} - u^{n}_{j-1}\right)$$

We add dissipation to the scheme by modifying it as follows:

$$u^{n+1}_{j} = u^{n-1}_{j} + a\lambda\left(u^{n}_{j+1} - u^{n}_{j-1}\right) - \frac{\epsilon}{16}\left(u^{n-1}_{j+2} - 4u^{n-1}_{j+1} + 6u^{n-1}_{j} - 4u^{n-1}_{j-1} + u^{n-1}_{j-2}\right) \tag{86}$$

where ϵ is an adjustable, non-negative parameter. Note that

$$\begin{aligned} u^{n-1}_{j+2} - 4u^{n-1}_{j+1} + 6u^{n-1}_{j} - 4u^{n-1}_{j-1} + u^{n-1}_{j-2} &= \triangle x^4 (u_{xxxx})^{n-1}_{j} + O(h^6) \\ &= \triangle x^4 (u_{xxxx})^{n}_{j} + O(h^5) = O(h^4) \end{aligned}$$

so that the term which is added, does not change the leading order truncation error, which in the form we have written the equation, is $O(\triangle t^3) = O(h^3)$ (local/one-step truncation error).

A Von Neumann analysis of the modified scheme shows that, in addtion to the CFL condition $\lambda \le 1$, we must have $\epsilon < 1$ for stability, and, further, that the per-step amplification factor for a mode with wave number ξ is, to leading order

$$1 - \epsilon\sin^4\frac{\xi}{2}$$

Thus the addition of the dissipation term is analagous to the use of an explicit "high frequency filter" (low-pass filter), which has a fairly sharp rollover as $\xi \to \pi$.

We note that an advantage to the use of explicit dissipation techniques (versus, for example, the use of an intrinsically dissipative scheme) is that the amount of dissipation can be controlled by tuning the dissipation parameter.

References

[1] Mitchell, A. R., and D. F. Griffiths, **The Finite Difference Method in Partial Differential Equations**, New York: Wiley (1980)

[2] Richtmeyer, R. D., and Morton, K. W., **Difference Methods for Initial-Value Problems**, New York: Interscience (1967)

[3] H.-O. Kreiss and J. Oliger, **Methods for the Approximate Solution of Time Dependent Problems**, GARP Publications Series No. 10, (1973)

[4] Gustatsson, B., H. Kreiss and J. Oliger, **Time-Dependent Problems and Difference Methods**, New York: Wiley (1995)

[5] Richardson, L. F., "The Approximate Arithmetical Solution by Finite Differences of Physical Problems involving Differential Equations, with an Application to the Stresses in a Masonry Dam", *Phil. Trans. Roy. Soc.*, **210**, (1910) 307–357.

[6] Anderson, E. em et al "Lapack Users' Guide - Release 2.0", (1994)
`http://www.netlib.org/lapack/lug/lapack_lug.html`