

1 Partial differential equations

In this chapter we discuss initial value problems for partial differential equations. We will in fact at first only consider the maybe simplest, non-trivial PDE we might think of,

$$\frac{\partial u}{\partial t} + V \frac{\partial u}{\partial x} = 0, \quad u(x, t = 0) = u^0(x). \quad (1.1)$$

Before we start to develop numerical schemes to treat this problem, let us shortly summarize the analytic properties of the solutions. In fact, this equation can be solved analytically, so it deals as a reference case.

The general solutions of this PDE are of the form $u(x, t) = u(x - Vt)$. Thus, this PDE couples temporal to spatial phenomena, but space and time may only appear in the combination $x - Vt$ in the solution (This allows for example to rewrite the equation as an ODE in terms of a new variable $\xi = x - Vt$). Once we know the initial condition $u^0(x)$, we immediately find $u(x, t) = u^0(x - Vt)$. All that Eq. (1.1) does, is to translate the initial distribution $u^0(x)$ in time by the distance Vt . The shape of u is completely preserved. For $V = c$, we might think of Eq. (1.1) as a wave equation for right-traveling electromagnetic waves in vacuum. In fact, the second order wave-equation

$$\frac{\partial^2 E}{\partial t^2} - c^2 \frac{\partial^2 E}{\partial x^2} = 0 \quad (1.2)$$

we know from electrodynamics is closely related to Eq. (1.1), since the operator in (1.2) can also be written in the form

$$\frac{\partial^2}{\partial t^2} - c^2 \frac{\partial^2}{\partial x^2} = \left(\frac{\partial}{\partial t} - c \frac{\partial}{\partial x} \right) \left(\frac{\partial}{\partial t} + c \frac{\partial}{\partial x} \right), \quad (1.3)$$

which leads to the conclusion, that Eq. (1.2) supports right- and left-traveling waves.

Let us now start to develop numerical methods to solve Eq. (1.1). When we discussed ODEs, we found that we have to introduce approximations to the differential operators in order to develop numerical schemes. It is the same here, but now we have to approximate the operators $\partial/\partial x$ and $\partial/\partial t$.

Let us start with a general observation. Assume we decided on a way how to approximate the spatial derivative in Eq. (1.1). This of course means that we now implicitly made the step from a continuous function $u(x, t)$ to a set of discrete values $u(x_i, t)$. We arrive at a system of ODEs,

$$\frac{\partial u_i}{\partial t} = -V \left(\frac{\partial u}{\partial x} \right)_i, \quad i = 1, \dots, N. \quad (1.4)$$

Note, that since the approximation of $(\partial u / \partial x)_i$ involves neighboring positions $\dots, i - 1, i + 1, \dots$, this is a system of coupled ODEs. In principle we already know how to proceed from here. We might think of applying the methods we discussed earlier to solve the system ODEs. Maybe use our standard work-horse Runge-Kutta algorithm? Well, there might be a few surprises for us. Let us start with a very simple scheme to see which points we will have to pay attention to.

1.1 Upwind differencing

The most simple way to approximate the differential operators in Eq. (1.1) is using first-order approximations. Using the notation $u_i^n = u(x_i, t_n) = u(i\Delta x, n\Delta t)$, we get

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = -V \frac{u_i^n - u_{i-1}^n}{\Delta x}. \quad (1.5)$$

On the left side we recognize a forward Euler step in time. The spatial derivative is a left-sided difference. We might also use a right-sided difference here. Both are of order Δx . However, we choose the left-sided difference, since we know from the analytical discussion, that the solution will strictly propagate from left to right. This means, that if we update the value at a position i , we should only take into account the prior value at i and to the left of it.

We rearrange terms and arrive at the *upwind* scheme,

$$u_i^{n+1} = -\frac{V\Delta t}{\Delta x} (u_i^n - u_{i-1}^n) + u_i^n. \quad (1.6)$$

The fraction on the right side is also known as Courant-Friedrichs-Lax number, so we abbreviate it as

$$C = \frac{V\Delta t}{\Delta x}. \quad (1.7)$$

In Eq. (1.6) appear the spatial and the temporal step sizes Δx and Δt , respectively. Usually we first choose Δx (by choosing the number of nodes N), such that we sample our function good enough in space. How do we choose Δt ? From where do we know what is a sufficiently small, but still as large as possible, choice? How can we infer, that we made a bad choice?

1.2 The Courant-Friedrichs-Lax criterion

The Courant-Friedrichs-Lax criterion (short CFL criterion) gives us a first idea how we have to choose Δt , once we chose Δx .

Suppose we integrate the evolution of the initial condition $u^0(x)$, and we are at time $t_n = n\Delta t$, at position $x_j = j\Delta x$. From the analytic solution we know, that the value is $u_j^n = u^0(x_j - Vt_n)$. At time t_0 this value was the distance Vt_n away (to the left) from our position x_j . Our numerical scheme of course does not know about this. But

1.3. Neumann stability

we have to make sure, that it is possible for the scheme to bring the information from $(x_j - Vt_n, 0)$ to (x_j, t_n) . If our scheme transports information to the right, but at a velocity that is too small, we can not expect to get the correct value for u_j^n . On the other hand, if information is transported too fast, the correct value has already passed our position.

Obviously, the ideal case is that the algorithm transports all information with the correct velocity V . We can achieve this by choosing $C = 1$, i.e. $\Delta t = \Delta x/V$, because then (1.6) becomes

$$u_i^{n+1} = u_{i-1}^n, \quad (1.8)$$

from which we deduce

$$u_i^n = u_{i-n}^0, \quad (1.9)$$

which means that we get the analytically correct value, even though we use discrete values of u .

1.3 Neumann stability

A Neumann analysis is a method to determine the stability of algorithms for PDEs. A stable algorithm guarantees, that we get finite values from finite input values. To investigate this, we make use of the linearity of the equation itself and the method we employ to solve it. Since everything is linear, we might think of u as being composed out of Fourier modes. Once we know what our algorithm does to each Fourier mode, we know what it does to our initial values.

To do a Neumann stability analysis, we assume that $u_j^0 = \exp(ik\Delta x j)$, i.e. that our initial value is a Fourier mode with wave-number k . That there are only discrete wave-numbers available for k can go unattended here, we can specify this later, for now k is completely free.

When we insert u^0 into the upwind scheme, we will obtain ξu^0 as a result (due to linearity). The (in general complex) factor ξ is called amplification factor. After n steps, we will have $\xi^n u^0$. If $|\xi| > 1$ we get exponential growth, so the method would be unstable. The ideal case would be $|\xi| = 1$, and for $|\xi| < 1$ we have a damping of the Fourier-mode.

We insert $u_j^n = \xi^n \exp(ik\Delta x j)$ into (1.6) and obtain

$$\xi(k) = 1 - C + C \exp(-ik\Delta x). \quad (1.10)$$

For $C = 1$ we have $|\xi| = 1$ for all k . This is again the ideal case, as we already found out in the section above. If $C > 1$, i.e. $V\Delta t/\Delta x > 1$, we have $|\xi| > 1$ for some k . The Fourier-modes for these k numbers will grow exponentially. We will investigate later which modes these are exactly. For $C < 1$, we have $|\xi| < 1$ for all modes, except the constant mode $k = 0$. This means we expect a damping of the solution. Note that $\xi(k)$ is a complex number. The magnitude of ξ decides about the growth of the mode, the phase of ξ describes phase-errors that the Fourier-mode will experience.

1.4 Forward time centered space

Can the upwind method be improved by using better approximations to the derivatives? The FTCS (forward time centered space) algorithm is the first try to improve the upwind scheme by using second order, centered differences in x . The approximation of the time-derivative still is a first order forward Euler.

The FCTS scheme for Eq. (1.1) is

$$u_i^{n+1} = -\frac{C}{2} (u_{i+1}^n - u_{i-1}^n) + u_i^n. \quad (1.11)$$

The Neumann analysis yields the amplification factor

$$\xi(k) = 1 - iC \sin(k\Delta x). \quad (1.12)$$

Obviously, this scheme is unstable for all values of C , since $|\xi| > 1$ for all $k \neq 0$.

1.4.1 Lax stabilization of FTCS

Lax found, that it is possible to stabilize FTCS by replacing $u_i^n \rightarrow \frac{1}{2}(u_{i+1}^n + u_{i-1}^n)$ in the approximation of the time-derivative. This modifies (1.11) to

$$u_i^{n+1} = -\frac{C}{2} (u_{i+1}^n - u_{i-1}^n) + \frac{1}{2}(u_{i+1}^n + u_{i-1}^n). \quad (1.13)$$

The amplification factor for this method is

$$\xi(k) = \cos(k\Delta x) - iC \sin(k\Delta x), \quad (1.14)$$

which implies $|\xi| \leq 1$ for $C \leq 1$. Here we find, that the method is stable, provided we choose the correct value for C . For $C = 1$, the u_{i+1}^n part cancels out, and we recover the upwind method.

It is interesting to re-engineer Eq. (1.13) and find the equation of which (1.13) is a discretization. We find

$$\frac{\partial u}{\partial t} = -V \frac{\partial u}{\partial x} + \frac{\Delta x^2}{2\Delta t} \frac{\partial^2 u}{\partial x^2}, \quad (1.15)$$

which is the original equation plus an additional diffusive term with the coefficient $\Delta x^2/(2\Delta t)$. Thus, Lax stabilization introduces artificial diffusion to prevent instability. The amount of diffusion depends on the prefactor.

1.5 Staggered Leap Frog

The leap frog method uses second order derivatives in time and in space. The spatial derivative is approximated at the central time over which the temporal step leaps (like a frog). The leap frog scheme for (1.1) is

$$\frac{u_i^{n+1} - u_i^{n-1}}{2\Delta t} = -V \frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x}. \quad (1.16)$$

1.6. Lax-Wendroff method

Now, u^{n+1} is a function of u^n and u^{n-1} ,

$$u_i^{n+1} = -C (u_{i+1}^n - u_{i-1}^n) + u_i^{n-1}. \quad (1.17)$$

The second order approximation of the time-derivative introduced a new problem. The original problem (1.1) is a first order initial value problem in t , i.e. we have to specify one initial condition in time. From Eq. (1.17) we see, that now, we need two initial conditions in time! Of course they can not be independent of each other, otherwise we would start with inconsistent initial conditions.

To solve the problem with the second initial conditions, we can use the following idea. The time integration is accurate up to second order in Δt . This means, we only have to get the second initial conditions correct up to second order in t . Assume we know u^0 and we additionally need u^{-1} to start the integration. For u^{-1} we can write a Taylor's series

$$u^{-1} = u(x, -\Delta t) = u(x, 0) - \Delta t \frac{\partial u}{\partial t} \Big|_{t=0} + \frac{\Delta t^2}{2} \frac{\partial^2 u}{\partial t^2} \Big|_{t=0} + \mathcal{O}(\Delta t^3). \quad (1.18)$$

In general we only know $u^0 = u(x, 0)$. We can make use of the PDE (1.1) itself to replace all temporal derivatives with spatial derivatives. From (1.1) it follows, that

$$\frac{\partial u}{\partial t} \Big|_{t=0} = -V \frac{\partial u}{\partial x} \Big|_{t=0}, \quad (1.19)$$

$$\frac{\partial^2 u}{\partial t^2} \Big|_{t=0} = V^2 \frac{\partial^2 u}{\partial x^2} \Big|_{t=0}. \quad (1.20)$$

Replacing these terms in (1.18) allows us to obtain the second initial condition, consistent up to second order in Δt .

The Neumann stability analysis for (1.17) yields the amplification factor

$$\xi(k) = -iC \sin(k\Delta x) \pm \sqrt{1 - C^2 \sin^2(k\Delta x)}. \quad (1.21)$$

The magnitude of $\xi(k)$ is 1 for all values of $C \leq 1$, which means that the leap frog method is stable and has no damping of the amplitude.

Unfortunately, there is again a potential spoiler. Since both derivatives are approximated by central differences, the temporal evolution of all values at grid nodes with even index is purely determined by values at other nodes with even index. The even grid points decouple completely from the uneven points and vice versa. This problem is known as mesh-drifting instability.

1.6 Lax-Wendroff method

A second order scheme which avoids mesh-drifting is the Lax-Wendroff method. This scheme first advances the solution from u^n to $u^{n+1/2}$ via

$$u_{i+1/2}^{n+1/2} = \frac{1}{2} (u_{i+1}^n + u_i^n) - \frac{V\Delta t}{2\Delta x} (u_{i+1}^n - u_i^n). \quad (1.22)$$

1. Partial differential equations

We note that these intermediate values in time are defined at intermediate points in space. Using these values, we may step from u^n to u^{n+1} ,

$$u_i^{n+1} = u_i^n - C \left(u_{i+1/2}^{n+1/2} - u_{i-1/2}^{n+1/2} \right). \quad (1.23)$$

Substituting (1.22) into (1.23), we get

$$u_i^{n+1} = u_i^n - C \left[\frac{1}{2} (u_{i+1}^n + u_i^n) - \frac{C}{2} (u_{i+1}^n - u_i^n) - \frac{1}{2} (u_i^n + u_{i-1}^n) + \frac{C}{2} (u_i^n - u_{i-1}^n) \right]. \quad (1.24)$$

From here we can see the reason, why mesh-drifting should not occur anymore. The temporal evolution of the value u_i^n depends on u_{i-1}^n, u_{i+1}^n and u_i^n itself. Odd and even positions do not decouple any longer.

The Neumann stability analysis for (1.24) gives

$$\xi(k) = 1 - iC \sin(k\Delta x) - C^2(1 - \cos(k\Delta x)), \quad (1.25)$$

or

$$|\xi(k)|^2 = 1 - C^2(1 - C^2)(1 - \cos(k\Delta x))^2. \quad (1.26)$$

We find again, that the method is stable for $C \leq 1$. In case of $C < 1$ we have $|\xi| < 1$ and some damping, which mainly affects long wave-length oscillations.

1.7 Conclusion

In conclusion, we discussed several methods to solve (1.1) and found none of them free of problems. For this particular equation it turns out, that the CFL number C is closely related to stability. However, this has not to be true for other equations. We observed, that the discretizations in time and space should not be treated independently.

We will close this chapter with Fig. 1.1 which summarizes some of our findings. The plots show the amplification magnitudes $|\xi|$ versus $k\Delta x$ for different choices of C . By plotting $|\xi|$ versus $k\Delta x$ and not only versus k , we are independent of the spatial discretization, thus, the results are valid in a general context. For $C = 1$ we find all but the FTCS algorithms to be stable. If $C < 1$, all again all methods except FTCS are stable, but now only Leap frog provides undamped solutions. Upwind and Lax-Wendroff suffer from damping. Lax-Wendroff shows substantial damping only for larger values of $k\Delta x$: Thus it dampens shorter wave-lengths modes more than upwind, but at the same time has better accuracy for longer wave-length modes than upwind. And of course second order precision in time, which means reduced computational effort.

In case of $C > 1$ all our algorithms are unstable. Note that ξ has two roots with different magnitudes for Leap frog. Only one of the roots is plotted here. The result for the other root looks like the red line, but is mirrored along the $k\Delta x = 0$ line.

1.7. Conclusion

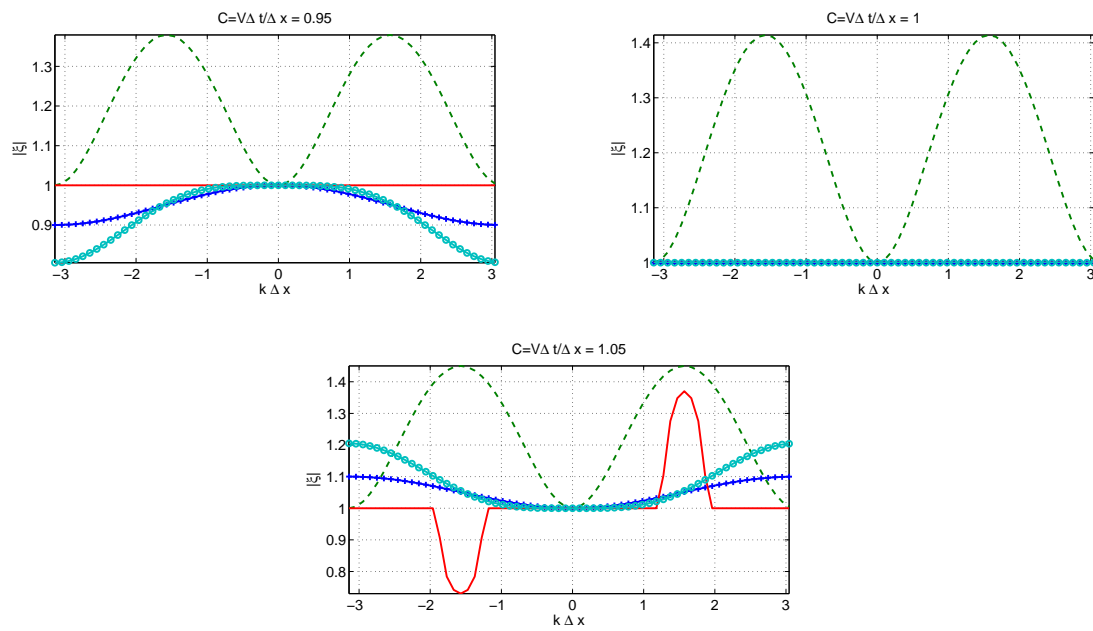


Figure 1.1: Comparison of amplification factors for Upwind (blue, crossed), FTCS (green, dashed), Leap Frog (red, solid) and Lax-Wendroff (blue, dotted). The three graphs show $|\xi|$ versus $k\Delta x$ for different values of C . Values of $|\xi|$ larger than 1 correspond to instability. Note that only one of the roots of $|\xi|$ is plotted for Leap-frog.