These notes draw heavily on *Numerical Recipes*, a valuable resource for entry-level understanding of numerical methods relevant to physics.

Partial differential equations (PDEs) are distinguished from ordinary differential equations by the presence of more than one independent variable, which may be time or space. Generally speaking, they are applicable to continuous systems: fluids, electromagnetic fields, gravity, general relativity, etc.

There are several types of PDEs, which can be classified as follows:

- Hyperbolic: second-order derivatives, with opposite signs, like the wave equation:

$$\frac{\partial^2 u}{\partial t^2} - v^2 \frac{\partial^2 u}{\partial x^2} = 0 \tag{1}$$

- Elliptic: second-order derivatives, with like signs, like the Laplace equation:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \tag{2}$$

- Parabolic: first-order and second-order mixed, like the diffusion equation:

$$\frac{\partial u}{\partial t} - \frac{\partial}{\partial x}\left(D \frac{\partial u}{\partial x}\right) = 0 \tag{3}$$

These sorts of equations are more complex than ODEs, mainly because you cannot reduce the equations to a single derivative of the state variable with the single independent variable. Furthermore, you cannot as easily set the boundary conditions; at least, there are many more ways to.

The simplest way to see this is the case of the Laplace equation. To solve Laplace for an electric potential within some region requires setting the potential at the surface of that region (or its derivative, or some combination). Clearly this is a bit complicated. Also, it is not at all obvious whether a solution with the desired boundary conditions is necessarily allowed by the equations (with Laplace, this is not really a problem of course, but it can be with more general equations).

The setting of the boundary conditions is an extremely important feature of any PDE problem, and usually defines the basic watersheds between numerical approaches. For example, you can imagine setting $u(x = 0, t)$ and using the wave or diffusion equations to integrate in time. These are *Cauchy* or *initial value* problems. But for Laplace, you need to specify *boundary values*, either of u (*Dirichlet*) or its derivative (*Neumann*).

1. Initial value problems

We will concern ourselves first with initial value problems. We will first consider the *flux-conservative initial value problems*. These are of the form:

$$\frac{\partial \vec{q}}{\partial t} = -\frac{\partial \vec{F}\left(\vec{q}\right)}{\partial x} \tag{4}$$

$\vec{F}$ is called the *conserved flux*.

For example, take the wave equation. It can be rewritten as:

$$\begin{aligned} \frac{\partial r}{\partial t} &= v\frac{\partial s}{\partial x} \\ \frac{\partial s}{\partial t} &= v\frac{\partial r}{\partial x} \end{aligned} \tag{5}$$

where:

$$\begin{aligned} r &= v\frac{\partial q}{\partial x} \\ s &= \frac{\partial q}{\partial t} \end{aligned} \tag{6}$$

In this case, we can write:

$$\vec{F}\left(\vec{q}\right) = \begin{pmatrix} 0 & -v \\ -v & 0 \end{pmatrix} \cdot \vec{q} \tag{7}$$

However, we will start with an even simpler example:

$$\frac{\partial q}{\partial t} = -v\frac{\partial q}{\partial x} \tag{8}$$

What is the general solution of this problem given some initial conditions $q_0(x)$?

2. A simple method that doesn't work

We will be looking at solving PDEs with finite difference related methods. These are not the only methods, but they are a good place to start.

To perform the finite differencing, we will take equal spaces in time and space:

$$\begin{aligned} x_j &= x_0 + j\Delta x \\ t_n &= t_0 + n\Delta t \end{aligned} \tag{9}$$

We can approximate the equation as:

$$\frac{q_j^{n+1} - q_j^n}{\Delta t} = -v\left(\frac{q_{j+1}^n - q_{j-1}^n}{2\Delta x}\right) \tag{10}$$

Then, given the initial conditions at $n = 0$ and all j, we can generate q at $n = 1$ explicitly from the current time step.

In the notebook, I implement this very simple method. It clearly doesn't work very well!!

3. Stability Analysis

To see why this happens, we need to perform a stability analysis. To do so using the *von Neumann stability analysis*, we will search for solutions of linearized equations under constant coefficients. For the advective equation, the right hand side is linear already, so we do not need to linearize it.

Explain to me what "linearizing" a nonlinear expression is?

In any case, under the conditions of linearity and constant coefficients, the solutions to the finite difference equation have the form:

$$q_j^n = \xi^n \exp\left(ikj\Delta x\right) \tag{11}$$

with some choice for k.

Under what conditions is this mode unstably growing in time?

For the difference question above, if you plug this in:

$$\begin{aligned}
\frac{q_j^{n+1} - q_j^n}{\Delta t} &= -v\left(\frac{q_{j+1}^n - q_{j-1}^n}{2\Delta x}\right) \\
\xi^n(\xi - 1)\exp\left(ikj\Delta x\right) &= -\frac{v\Delta t}{2\Delta x}\xi^n\left(\exp\left(ik(j+1)\Delta x\right) - \exp\left(ik(j-1)\Delta x\right)\right) \\
(\xi - 1) &= -\frac{v\Delta t}{2\Delta x}\left(\exp\left(ik\Delta x\right) - \exp\left(-ik\Delta x\right)\right) \\
\xi &= 1 - i\frac{v\Delta t}{\Delta x}\sin\left(k\Delta x\right)
\end{aligned} \tag{12}$$

What does this result mean for the stability of this method? Is it stable for any scale of fluctuation?

4. Lax Method

The Lax method is a stable alternative, which just reads:

$$\frac{q_j^{n+1} - \frac{1}{2}\left(q_{j+1}^n + q_{j-1}^n\right)}{\Delta t} = -v\left(\frac{q_{j+1}^n - q_{j-1}^n}{2\Delta x}\right) \tag{13}$$

Describe the difference between the Lax method and the original method in words.

We can evaluate the stability of this method as well:

$$\begin{aligned}
\frac{q_j^{n+1} - \frac{1}{2}\left(q_{j+1}^n + q_{j-1}^n\right)}{\Delta t} &= -v\left(\frac{q_{j+1}^n - q_{j-1}^n}{2\Delta x}\right) \\
\xi^n\left(\xi\exp(ikj\Delta x) - \frac{1}{2}\left(\exp(ik(j+1)\Delta x) + \exp\left(ik(j-1)\Delta x\right)\right)\right. &= -i\frac{v\Delta t}{\Delta x}\sin\left(k\Delta x\right)\xi^n\exp(ikj\Delta x) \\
\xi - \cos(k\Delta x) &= -i\frac{v\Delta t}{\Delta x}\sin\left(k\Delta x\right) \qquad (14)
\end{aligned}$$

and thus:

$$\xi = \cos(k\Delta x) - i\frac{v\Delta t}{\Delta x}\sin\left(k\Delta x\right) \qquad (15)$$

What does the stability requirement mean in this case?

Can you think of a heuristic interpretation of this requirement?

Things are actually a bit trickier, as we will see in the notebook. If I set $\Delta t = \Delta x/v$, we get exactly the right behavior. This is because in that case $\xi = 1$ exactly and so there is no increase — or decrease — in the signal. But, if the time step is a bit too large, we see the signal get unstable. If the time step is a bit too small, we see the signal damp away! We had better understand what is going on, because in real applications, where we don't know exactly what we should expect, we can easily be tricked.

Note that if we take a very broad Gaussian, it is less affected. What is going on here is that the small wavenumber k, large wavelength solutions are closer to actually being equal to zero in (**??**), and this leads to $\xi = 1$. So when the time step is too small, it tends to damp the high wavenumber, short wavelength modes, but not the longer ones. This causes the Gaussian to broaden (and also affects the fractional width of the narrow Gaussian more quickly than the broad one).

We can look directly at some sinusoidal waves to see this at work a little more cleanly, which the notebook does.

5. Numerical Viscosity

We can rewrite the numerical equation for Lax as follows:

$$\begin{aligned}
\frac{q_j^{n+1} - \frac{1}{2}\left(q_{j+1}^n + q_{j-1}^n\right)}{\Delta t} &= -v\left(\frac{q_{j+1}^n - q_{j-1}^n}{2\Delta x}\right) \\
\frac{q_j^{n+1} - q_j^n}{\Delta t} &= -\frac{1}{2}\frac{\left(q_{j+1}^n - 2q_j^n + q_{j-1}^n\right)}{\Delta t} - v\left(\frac{q_{j+1}^n - q_{j-1}^n}{2\Delta x}\right) \qquad (16)
\end{aligned}$$

This is an FTCS-type equation for:

$$\frac{\partial q}{\partial t} = -\frac{\partial q}{\partial x} + \frac{\Delta x^2}{2\Delta t}\frac{\partial^2 q}{\partial x^2} \qquad (17)$$

The second term here is like a viscosity. So although the advective equation itself has no viscosity, the numerical equations do have viscosity. This damps perturbations on small scales, like with regular viscosity.

Whether the viscosity matters or not is a practical question. It affects all scales at some rate, and the question is whether you are looking at sufficiently large scales that it does not matter.

6. Leapfrog Time Stepping

The Lax method is second order in space but first order in time. If we can store two time steps, we can achieve higher order in time. The basic idea is to calculate the space derivatives every step, and the time derivatives every other step.

In general this yields:

$$q_j^{n+1} - q_j^{n-1} = -\frac{\Delta t}{\Delta x}\left(F_{j+1}^n - F_{j-1}^n\right) \tag{18}$$

For the advective equation:

$$q_j^{n+1} - q_j^{n-1} = -\frac{v\Delta t}{\Delta x}\left(q_{j+1}^n - q_{j-1}^n\right) \tag{19}$$

If we evaluate this under the von Neumann stability analysis we find:

$$\begin{aligned} \xi^2 - 1 &= -\frac{v\Delta t}{\Delta x}\xi\left(\exp(ik\Delta x) - \exp(-ik\Delta x)\right) \\ \xi^2 - 1 &= -2i\xi\frac{v\Delta t}{\Delta x}\sin k\Delta x \end{aligned} \tag{20}$$

which then yields:

$$\xi = -i\frac{v\Delta t}{\Delta x}\sin k\Delta x \pm \sqrt{1 - \left(\frac{v\Delta t}{\Delta x}\sin k\Delta x\right)^2} \tag{21}$$

and so as long as the square root is of a real number:

$$|\xi|^2 = \left(\frac{v\Delta t}{\Delta x}\sin k\Delta x\right)^2 + 1 - \left(\frac{v\Delta t}{\Delta x}\sin k\Delta x\right)^2 = 1 \tag{22}$$

Then the conditions of stability for all k become:

$$\Delta t < \frac{\Delta x}{v} \tag{23}$$

i.e. the Courant condition again.

7. Central in Time Methods

Note that this is more stable

8. Diffusion Equation

Another typical type of equation is the diffusion equation, generally of the form:

$$\frac{\partial q}{\partial t} = \frac{\partial}{\partial x}\left(D\frac{\partial q}{\partial x}\right) \tag{24}$$

This too can be put in flux-conservative form where $F = -D\partial u/\partial x$.

Note you have actually already solved this problem in your homework, and for the simple, linear case we are discussing of course that is the preferred method.

Before actually discussing a specific method, it is worth asking what quantity is going to be relevant to the time step.

From dimensional analysis, how can you construct a quantity with units of time from the diffusion coefficient D and a chosen spatial grid size Δx?

We will define a unitless quantity:

$$\alpha = \frac{D\Delta t}{\Delta x^2} \tag{25}$$

We can perform the diffusion equation finite differencing with:

$$\frac{q_j^{n+1} - q_j^n}{\Delta t} = D\left[\frac{q_{j+1}^n - 2q_j^n + q_{j-1}^n}{\Delta x^2}\right] \tag{26}$$

which can be rearranged to:

$$q_j^{n+1} = q_j^n + \alpha\left[q_{j+1}^n - 2q_j^n + q_{j-1}^n\right] \tag{27}$$

Performing a stability analysis on this yields:

$$\begin{aligned}
\xi &= 1 + \alpha\left[e^{ik\Delta x} - 2 + e^{-ik\Delta x}\right] \\
&= 1 + 2\alpha\left[\cos(k\Delta x) - 1\right] \\
&= 1 + 2\alpha\left[1 - 2\sin^2\left(\frac{k\Delta x}{2}\right) - 1\right] \\
&= 1 - 4\alpha\sin^2\left(\frac{k\Delta x}{2}\right)
\end{aligned} \tag{28}$$

What does this make the condition on stability?

The problem in this case is that we are most likely interested in the diffusion on much larger scales than a single grid cell. If we are interested in some scale λ, then the number of time steps it will take to reach that scale is $N_T \sim (\lambda/\Delta x)^2$. That's fine for the diffusion equation in its simplest form: just take Δx to be comparable to the scales you care about. But if you are doing a problem numerically, usually it won't be a pure diffusion problem, and then there are other features in

the problem that need to be tracked at higher resolution. So we want a way to retain the spatial resolution but take longer time steps.

We can take larger time steps if we are willing to sacrifice accuracy at small scales. One version of this to construct an *implicit* differencing method:

$$\frac{q_j^{n+1} - q_j^n}{\Delta t} = D\left[\frac{q_{j+1}^{n+1} - 2q_j^{n+1} + q_{j-1}^{n+1}}{\Delta x^2}\right] \tag{29}$$

Here the right-hand side uses the future time step values!

How do we solve for q^{n+1} in this case?

We can check the stability of the system:

$$\xi - 1 = \alpha\xi\left(e^{ik\Delta x} - 2 + 2e^{ik\Delta x}\right) \tag{30}$$

and using the same trigonometry as before:

$$\xi\left[1 + 4\alpha\sin^2\left(\frac{k\Delta x}{2}\right)\right] = 1 \tag{31}$$

and so:

$$\xi = \frac{1}{1 + 4\alpha\sin^2\left(\frac{k\Delta x}{2}\right)} \tag{32}$$

This means that this is always stable. You can see how these implicit methods can end up being very stable, because you'll end up with terms in the denominator.

But it can't possibly be accurate at small scales for $\alpha > 1$). It tends to drive the solution to the solution of:

$$\frac{\partial^2 q}{\partial x^2} = 0 \tag{33}$$

which you can see by looking at the matrix equation above and setting $\alpha \gg 1$ — the "source" term goes to zero and you are left with the equilibrium solution. That's not a crazy result, but it does mean that if there is structure on small scales in your initial conditions, this dissipates, and perhaps not accurately.

We can achieve higher accuracy by using a second-order-in-time technique. This is the most commonly used technique, and is called the *Crank-Nicolson* method. It is simply the average of the two above methods.

$$\frac{q_j^{n+1} - q_j^n}{\Delta t} = \frac{\alpha}{2}\left[\left(q_{j+1}^{n+1} - 2q_j^{n+1} + q_{j-1}^{n+1}\right) + \left(q_{j+1}^n - 2q_j^n + q_{j-1}^n\right)\right] \tag{34}$$

This is second order in time because each side is centered on $n+1/2$ and therefore over the interval n to $n+1$ the first order error terms integrate out. If you work it out, it is also unconditionally stable, with:

$$\xi = \frac{1 - 2\alpha\sin^2\left(\frac{k\Delta x}{2}\right)}{1 + 2\alpha\sin^2\left(\frac{k\Delta x}{2}\right)} \tag{35}$$

How do we deal with generalizations of the diffusion equation, e.g. if D is a function of position, or is a function of q? In the first case, it is sometimes possible to define a change of variable $x(y)$ such that:

$$\mathrm{d}y = \frac{\mathrm{d}x}{D(x)} \tag{36}$$

In this case:

$$\frac{1}{D(x)}\frac{\partial}{\partial x} = \frac{\partial}{\partial y} \tag{37}$$

and then we just get:

$$\frac{\partial q}{\partial t} = \frac{1}{D(y)}\frac{\partial^2 q}{\partial x^2} \tag{38}$$

Note that in this (and most other nontrivial cases) the stability criterion is not the same across the whole grid. So:

$$\Delta t \leq \min\left[\frac{(\Delta y)^2}{2D(y)^{-1}}\right] \tag{39}$$

where there is *not* a typo in the denominator!

This isn't always possible but in this and in other situations one can use a finite differencing with D evaluated in the right places. The explicit form of the finite difference becomes, for example:

$$\frac{q_j^{n+1} - q_j^n}{\Delta t} = \frac{D_{j+1/2}\left(q_{j+1}^n - q_j^n\right) - D_{j-1/2}\left(q_j^n - q_{j-1}^n\right)}{(\Delta x)^2} \tag{40}$$

Again the stability criterion will be dependent on position and we have to pick the smallest time step (i.e. corresponding to highest D in our domain). We can do the same thing for fully implicit or Crank-Nicolson methods.

Nonlinear cases (where D is a function of q) are more difficult in general. Explicit methods can just evaluate D at the current time step:

$$D_{j+1/2} = \frac{1}{2}\left[D\left(q_{j+1}^n\right) + D\left(q_j^n\right)\right]. \tag{41}$$

But then if we plug this into an implicit scheme:

$$\frac{q_j^{n+1} - q_j^n}{\Delta t} = \frac{D_{j+1/2}\left(\vec{q}^{n+1}\right)\left(q_{j+1}^{n+1} - q_j^{n+1}\right) - D_{j-1/2}\left(\vec{q}^{n+1}\right)\left(q_j^{n+1} - q_{j-1}^{n+1}\right)}{(\Delta x)^2} \tag{42}$$

this isn't a simple tridiagonal linear system any more. The only way is if we can now change variables in the dependent variables to $z(q)$:

$$\mathrm{d}z = D(q)\,\mathrm{d}q, \tag{43}$$

which is nice because the RHS becomes:

$$\frac{\partial^2 z}{\partial x^2} \tag{44}$$

which has a straightforward finite difference. Great, but we are trying to evolve q. To do this we linearize:

$$z_j^{n+1} = z_j^n + \left(q_j^{n+1} - q_j^n\right) D\left(q_j^n\right) \tag{45}$$

Then plugging this into the spatial finite difference, we get an linear implicit scheme, where the nonlinear parts are basically calculated from step n.

9. Schrödinger Equation

In ordinary differential equations, there was an argument that sometimes one would trade decreased accuracy in some respects, for increased accuracy in other respects, like for energy conservation or phase space conservation. There are similar trade-offs in partial differential equations.

The Schrödinger equation is one example, where unitarity is important—if you are following an electron, the total probability of that electron existing *somewhere* really ought to stay near unity. It turns out that Crank-Nicolson, in addition to its nice properties above, *also* guarantees unitarity.

We'll strip out the mass and Planck's constant and just write:

$$i\frac{\partial \phi}{\partial t} = -\frac{\partial^2 \phi}{\partial x^2} + V(x)\phi. \tag{46}$$

The first-order explicit scheme for this method is unconditionally unstable. A first-order implicit scheme yields:

$$i\left[\frac{\phi_j^{n+1} - \phi_j^n}{\Delta t}\right] = -\left[\frac{\phi_{j+1}^{n+1} - 2\phi_j^{n+1} + \phi_{j-1}^{n+1}}{(\Delta x)^2}\right] + V_j \phi_j^{n+1} \tag{47}$$

This yields:

$$\begin{bmatrix} \beta_0 & \alpha & 0 & 0 & 0 & \dots \\ \alpha & \beta_1 & \alpha & 0 & 0 & \dots \\ 0 & \alpha & \beta_2 & \alpha & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & 0 & 0 & \alpha & \beta_{N-2} & \alpha \\ \dots & 0 & 0 & 0 & \alpha & \beta_{N-1} \end{bmatrix} \cdot \begin{bmatrix} \phi_0^{n+1} \\ \phi_1^{n+1} \\ \phi_2^{n+1} \\ \dots \\ \phi_{N-2}^{n+1} \\ \phi_{N-1}^{n+1} \end{bmatrix} = \begin{bmatrix} \phi_0^n \\ \phi_1^n \\ \phi_2^n \\ \dots \\ \phi_{N-2}^n \\ \phi_{N-1}^n \end{bmatrix} \tag{48}$$

where:

$$\begin{aligned} \alpha &= -\frac{\Delta t}{(\Delta x)^2} i \\ \beta_j &= \left[\frac{2\Delta t}{(\Delta x)^2} + (\Delta t) V_j\right] i + 1 \end{aligned} \tag{49}$$

Then von Neumann yields:

$$i\left[\frac{\xi - 1}{\Delta t}\right] = \xi\left[\frac{2\cos k\Delta x - 2}{(\Delta x)^2} - V_j\right] \tag{50}$$

and rearranging and using the same trig tricks as above:

$$\xi = \frac{1}{1 + i\left[\frac{4\Delta t}{(\Delta x)^2}\sin^2\left(\frac{k\Delta x}{2}\right) + V_j\Delta t\right]} \tag{51}$$

and this always is less than one because the modulus of the denominator is always greater than one.

But this method is not unitary in the sense of maintaining

$$\int_{-\infty}^{\infty} \mathrm{d}x\,|\phi(x)|^2 = 1 \tag{52}$$

This sort of issue was a nuisance for ODEs, which motivated symplectic methods, but in the case of Schrödinger it is extremely severe and makes the implicit methods useless, as I show in the notebooks.

A unitary form of the same first order approximation can be obtained as follows. Another way to think about the solutions to this equation across Δt is:

$$\vec{\phi}^{n+1} = \exp(-iH(\Delta t))\vec{\phi}^n \tag{53}$$

and the first-order-in-time explicit approximation of this is:

$$\vec{\phi}^{n+1} \approx (1 - iH(\Delta t))\vec{\phi}^n \tag{54}$$

and the first-order-in-time implicit approximation of this is:

$$\vec{\phi}^{n+1} \approx (1 + iH(\Delta t))^{-1}\,\vec{\phi}^n \tag{55}$$

where we evaluate H discretely. The norm of ϕ will evolve as:

$$\left[(1 + iH(\Delta t))^{-1}\right]^\dagger (1 + iH(\Delta t))^{-1} = (1 - iH(\Delta t))^{-1}\,(1 + iH(\Delta t))^{-1} \tag{56}$$

which for some eigenfunction with eigenvalue E will go as $1 - (\Delta t)^2E^2$, and thus will cause a decrease as we saw.

But at the same order of approximation we can write:

$$\vec{\phi}^{n+1} = \exp(iH(\Delta t))\vec{\phi}^n \approx (1 + iH(\Delta t)/2)^{-1}(1 - iH(\Delta t)/2)\vec{\phi}^n \tag{57}$$

which can be thought of as moving forward half a step explicitly and the next half step implicity. As noted above, this is just Crank-Nicolson. This is unconditionally stable and the norm is constant, since its adjoint operator is its inverse.