Lecture 19

Partial differential equations

- The Schrödinger equation
- More than 1+1.
- Operator splitting.
- Elliptic equations: Fourier methods.
- Elliptic equations: relaxation methods.

The Schrödinger equation:

We have driven home the message that accuracy is not everything, one wants stability as well. In the previous examples we also noted that even that is not enough: one wanted to be able to take appropriately large timesteps. In many physical applications there are *further* additional restrictions that need to be met. An example of this is Schrödinger's equation.

$$i\frac{\partial \psi}{\partial t} = -\frac{\partial^2 \psi}{\partial x^2} + V(x)\psi$$

This is a parabolic equation, so we could try one of the schemes we discussed before, for instance, an implicit scheme for stability reasons,

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

A von Neumann analysis of this scheme seems to give the green light,

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

But there is a problem. The scheme is not *unitary*. That is, we know that Schroedinger's equation can be written as,

$$i\frac{\partial\psi}{\partial t} = H\psi \quad \text{with} \quad H = -\frac{\partial^2}{\partial x^2} + V(x)$$

Which can be formally integrated as, $\psi(x,t) = e^{-iHt}\psi(x,0)$

This implies that if one starts with initial data such that,

$$\int_{-\infty}^{\infty} |\psi|^2 dx = 1$$

This condition is preserved in time.

What is happening is that the schemes we write are tantamount to (say, for an explicit FTCS unstable scheme),

$$\psi_j^{n+1} = (1 - iH\Delta t)\psi_j^n$$

With H appropriately discretized as a centered difference. For the implicit scheme one has,

$$\psi_j^{n+1} = (1 + iH\Delta t)^{-1}\psi_j^n$$

The evolution operator so constructed is first order accurate in time but is not unitary. The correct way to handle this is to use Cayley's representation of the evolution operator,

$$e^{-iHt} \simeq \frac{1 - \frac{1}{2}iH\Delta t}{1 + \frac{1}{2}iH\Delta t}$$ Or, $$\left(1 + \tfrac{1}{2}iH\Delta t\right)\psi_j^{n+1} = \left(1 - \tfrac{1}{2}iH\Delta t\right)\psi_j^n$$

Which is both second order accurate and unitary.

More than 1+1:

Our discussion was confined for simplicity reasons to 1+1 dimensional equations. The techniques we discussed are, however, of more general applicability. For instance, von Neumann stability can be tested in any number of dimensions, one just assumes a single amplitude that grows with n and harmonic dependence (with different k's) in the spatial coordinates.

The computational cost of going to higher dimensions grows dramatically. Stability of a code, however, can in general be assessed with very small grids. Sometimes additional instabilities can appear when the grid is enlarged, but instabilities seen for smaller grids never go away when the grid is enlarged (unless there are special features that justify this). Some misguided people think "my code will stabilize with more points" confusing accuracy and stability.

Example: Lax method for a flux conservative equation in 2D

$$\frac{\partial u}{\partial t} = -\nabla \cdot \mathbf{F} = -\left(\frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y}\right)$$

Use a spatial grid with

$$x_j = x_0 + j\Delta$$

$$y_l = y_0 + l\Delta$$

Same Δ in both for simplicity.

Lax scheme:

$$u_{j,l}^{n+1} = \frac{1}{4}(u_{j+1,l}^n + u_{j-1,l}^n + u_{j,l+1}^n + u_{j,l-1}^n) - \frac{\Delta t}{2\Delta}(F_{j+1,l}^n - F_{j-1,l}^n + F_{j,l+1}^n - F_{j,l-1}^n)$$

Let's study stability in the particular case $F_x = v_x u, \qquad F_y = v_y u$

Mode: $u_{j,l}^n = \xi^n e^{ik_x j\Delta} e^{ik_y l\Delta}$

$$\xi = \frac{1}{2}(\cos k_x\Delta + \cos k_y\Delta) - i\alpha_x \sin k_x\Delta - i\alpha_y \sin k_y\Delta$$

where

$$\alpha_x = \frac{v_x \Delta t}{\Delta}, \qquad \alpha_y = \frac{v_y \Delta t}{\Delta}$$

$|\xi|^2 \leq 1$ becomes

$$\frac{1}{2} - (\alpha_x^2 + \alpha_y^2) \geq 0$$

$$\Delta t \leq \frac{\Delta}{\sqrt{2}(v_x^2 + v_y^2)^{1/2}}$$

More generally:

$$\Delta t \leq \frac{\Delta}{\sqrt{N}|v|}$$

Maximum propagation speed.

Diffusion equation in multidimensions:

$$\frac{\partial u}{\partial t} = D\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right)$$

Crank-Nicholson:

$$u_{j,l}^{n+1} = u_{j,l}^{n} + \frac{1}{2}\alpha\left(\delta_x^2 u_{j,l}^{n+1} + \delta_x^2 u_{j,l}^{n} + \delta_y^2 u_{j,l}^{n+1} + \delta_y^2 u_{j,l}^{n}\right)$$

$$\alpha \equiv \frac{D\Delta t}{\Delta^2} \qquad \Delta \equiv \Delta x = \Delta y$$

$$\delta_x^2 u_{j,l}^{n} \equiv u_{j+1,l}^{n} - 2u_{j,l}^{n} + u_{j-1,l}^{n}$$

Problem: matrix is not tridiagonal anymore. It is still sparse, though.

Alternative approach: divide your stepsize in time into two stepsizes of half the length and update each dimension in each substep:

$$u_{j,l}^{n+1/2} = u_{j,l}^{n} + \frac{1}{2}\alpha\left(\delta_x^2 u_{j,l}^{n+1/2} + \delta_y^2 u_{j,l}^{n}\right)$$
$$u_{j,l}^{n+1} = u_{j,l}^{n+1/2} + \frac{1}{2}\alpha\left(\delta_x^2 u_{j,l}^{n+1/2} + \delta_y^2 u_{j,l}^{n+1}\right)$$

Each substep now only requires solving a tridiagonal system.

This is a particular example of a more general technique known as *Operator Splitting*.

Operator splitting: Sometimes one is faced with equations like the advection-diffusion equation,

Where one knows different schemes for treating the two terms in the RHS.

$$\frac{\partial u}{\partial t} = -v\frac{\partial u}{\partial x} + D\frac{\partial^2 u}{\partial x^2}$$

One possibility is to combine these schemes in time: take a step to the midpoint assuming the RHS has only the first term. Then take another step to the final point assuming only the second term is present.

The advantage of this technique is that we can use discretizations tailored to each term (for instance explicit for the advection term, Crank-Nicholson for the diffusion term). It is clear that the resulting scheme reproduces the equation in the continuum limit.

Details will matter. This approach sometimes works. Of course no general statements can be made about it.

Elliptic equations: boundary value problems

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

Fourier methods:

We expand both u and rho in multidimensional Fourier series. We will discuss routines for doing Fourier tranform later in the course,

$$u_{jl} = \frac{1}{JL} \sum_{m=0}^{J-1} \sum_{n=0}^{L-1} \widehat{u}_{mn} e^{-2\pi ijm/J} e^{-2\pi iln/L} \qquad \rho_{jl} = \frac{1}{JL} \sum_{m=0}^{J-1} \sum_{n=0}^{L-1} \widehat{\rho}_{mn} e^{-2\pi ijm/J} e^{-2\pi iln/L}$$

We now consider the discretized version of the above equation,

$$\frac{u_{j+1,l} - 2u_{j,l} + u_{j-1,l}}{\Delta^2} + \frac{u_{j,l+1} - 2u_{j,l} + u_{j,l-1}}{\Delta^2} = \rho_{j,l}$$

Or,

$$u_{j+1,l} + u_{j-1,l} + u_{j,l+1} + u_{j,l-1} - 4u_{j,l} = \Delta^2 \rho_{j,l}$$

We get,

$$\widehat{u}_{mn}\left(e^{2\pi im/J}+e^{-2\pi im/J}+e^{2\pi in/L}+e^{-2\pi in/L}-4\right)=\widehat{\rho}_{mn}\Delta^2$$

Or,

$$\widehat{u}_{mn}=\frac{\widehat{\rho}_{mn}\Delta^2}{2\left(\cos\frac{2\pi m}{J}+\cos\frac{2\pi n}{L}-2\right)}$$

The general strategy for solution is therefore:
a) Compute the Fourier transform of the source rho.
b) Use the above equation to find hat u.
c) Find the desired solution via inverse Fourier transform.

However, we need to take into account some details. In particular, boundary conditions. Careful examination shows that the solution we just constructed satisfies periodic boundary conditions.

If one is faced with a Dirichlet boundary condition u=0 at the boundary, one would use a sine Fourier expansion,

$$u_{jl} = \frac{2}{J}\frac{2}{L}\sum_{m=1}^{J-1}\sum_{n=1}^{L-1}\widehat{u}_{mn}\sin\frac{\pi jm}{J}\sin\frac{\pi ln}{L}$$

And similarly for rho and for the inverse transform.

If one has u=0 on all boundaries except one where one has u=f(y) at x=JΔ, one has to take the above solution for the u=0 problem in all boundaries and add a solution of the homogeneous equation that satisfies the boundary condition,

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=0 \qquad u^H_{jl}=\frac{2}{L}\sum_{n=1}^{L-1}A_n\sinh\frac{\pi nj}{J}\sin\frac{\pi nl}{L}$$

$$A_n = \frac{1}{\sinh \pi n}\sum_{l=1}^{L-1} f_l \sin\frac{\pi nl}{L} \qquad f(y=l\Delta)\equiv f_l$$

Another way of viewing the boundary conditions is to encode them in the right hand side of the equation.

We formally write for the solution $$u = u' + u^B$$

Where u'=0 on the boundary and u^B=0 everywhere except on the boundary.

The equation then becomes $$\nabla^2 u' = -\nabla^2 u^B + \rho$$

And discretized,

$$u'_{j+1,l} + u'_{j-1,l} + u'_{j,l+1} + u'_{j,l-1} - 4u'_{j,l} = - (u^B_{j+1,l} + u^B_{j-1,l} + u^B_{j,l+1} + u^B_{j,l-1} - 4u^B_{j,l}) + \Delta^2 \rho_{j,l}$$

The u^B terms vanish everywhere except (in the example before) at j=J-1, where,

$$u'_{J,l} + u'_{J-2,l} + u'_{J-1,l+1} + u'_{J-1,l-1} - 4u'_{J-1,l} = -f_l + \Delta^2 \rho_{J-1,l}$$

So in matrix form, the linear equation to be handled is exactly the same as before with the exception of one row.

Relaxation methods:

We already discussed relaxation methods in the context of two-point boundary value problems. The methods consisted in finding a scheme that updated the values of the function across the grid approximating in each iteration the desired solution better and better.

There is an alternative, more "physical" way to think of relaxation. Consider an elliptic problem, $\mathcal{L}u = \rho$

Now consider the associated diffusive equation, $\dfrac{\partial u}{\partial t} = \mathcal{L}u - \rho$

As the solution of the parabolic equation reaches its asymptotic value, the time derivatives vanish and one is effectively solving the elliptic equation we started with!

We can therefore just translate the machinery we developed for diffusive IVPs to construct relaxation algorithms for elliptic problems!

Let us consider as example the Poisson equation,

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} - \rho$$

And try an FTCS scheme,

$$u_{j,l}^{n+1} = u_{j,l}^{n} + \frac{\Delta t}{\Delta^2}\left(u_{j+1,l}^{n} + u_{j-1,l}^{n} + u_{j,l+1}^{n} + u_{j,l-1}^{n} - 4u_{j,l}^{n}\right) - \rho_{j,l}\Delta t$$

Recall that FTCS was stable only if $\Delta t/\Delta^2 \leq \frac{1}{2}$

(we derived this in 1+1 dimensions, in 2+1 dimensions the RHS is 1/4)

Taking the largest possible timestep, we get

$$u_{j,l}^{n+1} = \frac{1}{4}\left(u_{j+1,l}^{n} + u_{j-1,l}^{n} + u_{j,l+1}^{n} + u_{j,l-1}^{n}\right) - \frac{\Delta^2}{4}\rho_{j,l}$$

This equation has a simple interpretation: the solution is given by the average of the four nearest neighbors (plus a source term). It is a classical method called Jabobi method.

Jacobi's method might be classical but is not very useful. We learnt earlier that FTCS works for diffusion equations in the sense of not blowing up, but that the stepsizes required for stability were too small. Here we have a reflection of the same problem: Jacobi's method converges, but too slowly.

Another classical (and also useless in practice) method is the Gauss-Seidel method. It is the same as Jacobi's method but instead of performing the average with the values at the previous "time" we use new values as they become available, that is,

$$u_{j,l}^{n+1} = \frac{1}{4}\left(u_{j+1,l}^{n} + u_{j-1,l}^{n+1} + u_{j,l+1}^{n} + u_{j,l-1}^{n+1}\right) - \frac{\Delta^2}{4}\rho_{j,l}$$

(if we are incrementing j for fixed l).

To analyze a bit the rates of convergence (a full discussion exceeds the scope of this course) we will analyze the matrix equations resulting from both methods.

Summary

- Schrödinger equation and other equations with conservation laws may require special discretizations.
- Going to more than 1D is in principle straightforward, costly in practice.
- For elliptical equations one uses Fourier methods or relaxation techniques.