PDEs

Introduction to Numerical Analysis

Riccardo Mannella

February 16, 2008

Basic

From a formal point of view there are three main different categories of PDE, depending on their characteristics (curve of information propagation):

- **hyperbolic** an example is the wave equation, $\frac{\partial^2 u}{\partial t^2} - v^2 \frac{\partial^2 u}{\partial x^2} = 0$
- **parabolic** an example is the diffusion equation $\frac{\partial P}{\partial t} + \frac{\partial}{\partial x}\left(D \frac{\partial P}{\partial x}\right) = 0$
- **elliptic** an example is Poisson equation $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \rho(x, y)$

Computationally the difference is whether we have an *initial value* (evolution) or a *boundary value* (static) problem

Basic

Initial value problems, point to evaluate:

- Variable to propagate?
- Evolution equation for each variable?
- Highest time derivative?
- Boundaries? (Dirichlet/Neumann)

Main concern: *stability*!

Basic

Boundary value problems, point to evaluate:

- Relevant variables?
- Equations to satisfy in the "bulk"?
- Boundaries? (Dirichlet/Neumann)

Main concern: *efficiency*. Tipically, we need to solve a large number of algebraic equations.

A worked example: boundaries

Start with $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \rho(x, y)$, introduce a grid spacing, Δ, write

$$x_j = x_0 + j\Delta \quad j = 0, ..., J \qquad y_l = y_0 + l\Delta \quad l = 0, ..., L$$

write $u_{j,l} = u(x_j, y_l)$ etc. and the equation becomes

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

or alternatively

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

A worked example: boundaries

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

Introduce $i = j(L+1) + l$, and the equation becomes

$$u_{i+L+1} + u_{i-(L+1)} + u_{i+1} + u_{i-1} - 4u_i = \Delta^2 \rho_i$$

Bringing the "boundary conditions" to the r.h.s., we can write the equation in the form

$$\mathbf{Au} = \mathbf{b}$$

where $\mathbf{A}$ has the form

A worked example: boundaries

$$u_{i+L+1} + u_{i-(L+1)} + u_{i+1} + u_{i-1} - 4u_i \equiv \mathbf{Au} = \Delta^2 \rho_i$$

$$\mathbf{A} = \left[\begin{array}{ccc|ccc|ccc}
-4 & 1 & & 1 & & & & & \\
1 & -4 & 1 & & 1 & & & & \\
 & 1 & -4 & & & 1 & & & \\
\hline
1 & & & -4 & 1 & & 1 & & \\
 & 1 & & 1 & -4 & 1 & & 1 & \\
 & & 1 & & 1 & -4 & & & 1 \\
\hline
 & & & 1 & & & -4 & 1 & \\
 & & & & 1 & & 1 & -4 & 1 \\
 & & & & & 1 & & 1 & -4
\end{array}\right]$$

A worked example: boundaries

How do we solve $\mathbf{Au} = \mathbf{b}$?

- Fourier (fast) methods, really applicable only when $\mathbf{A}$ is constant (see later)
- Directly, using for instance conjugate gradient (very accurate, not very efficient when $\mathbf{A}$ is large because it is sparse)
- Relaxation, recommended when $\mathbf{A}$ is large. Write $\mathbf{A} = \mathbf{E} - \mathbf{F}$ where $\mathbf{E}$ is easily invertible, start with some guessed $\mathbf{u}^{(0)}$, and iterate to solve
$$\mathbf{Eu}^{(r+1)} = \mathbf{b} + \mathbf{Fu}^{(r)}$$

Flux-Conservative

Many PDE are in the form of a flux conserved equation,

$$\frac{\partial \mathbf{u}}{\partial t} = -\frac{\partial \mathbf{F}(\mathbf{u})}{\partial x}$$

For instance $\frac{\partial^2 u}{\partial t^2} = v^2 \frac{\partial^2 u}{\partial x^2}$ can be written

$$\frac{\partial r}{\partial t} = v \frac{\partial s}{\partial x}$$

$$\frac{\partial s}{\partial t} = v \frac{\partial r}{\partial x}$$

where $r \equiv v \frac{\partial u}{\partial x}$ and $s \equiv \frac{\partial u}{\partial t}$

Flux-Conservative

Consider $\frac{\partial u}{\partial t} = -v\frac{\partial u}{\partial x}$. Using $x_j = x_0 + j\Delta x \;\; j = 0, .., J$, $t_n = t_0 + n\Delta t \;\; n = 0, ..., N$, defining $u_j^n \equiv u(t_n, x_j)$, the equation is discretised as

$$\left.\frac{\partial u}{\partial t}\right|_{j,n} = \frac{u_j^{n+1} - u_j^n}{\Delta t} + O(\Delta t)$$

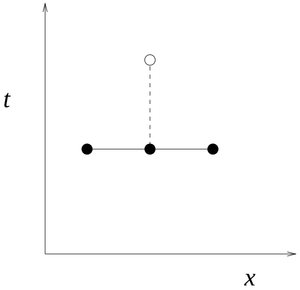
$$\left.\frac{\partial u}{\partial x}\right|_{j,n} = \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} + O(\Delta x^2)$$

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = -v\left(\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x}\right)$$

Flux-Conservative

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = -v \left(\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} \right)$$

$$u_j^{n+1} = u_j^n - v\Delta t \left(\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} \right)$$



Forward Time Centered Space, FTCS: *explicit* scheme

Flux-Conservative

Stability Analysis (von Neumann): the idea is to study the linear stability of a wave of k vector. Assuming $u_j^n = \xi^n e^{ikj\Delta x}$ and substituting in

$$u_j^{n+1} = u_j^n - v\Delta t \left(\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} \right)$$

one obtains (ξ^n is the amplitude at t_n)

$$\xi = 1 - i\frac{v\Delta t}{\Delta x} \sin k\Delta x$$

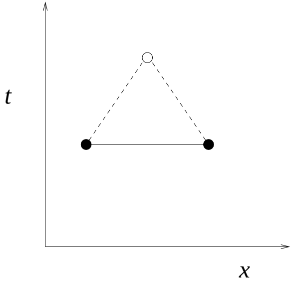
Clearly, $|\xi| > 1$ for any choice of Δx and Δt, hence the method is unstable for all k's.

Flux-Conservative

Lax method: use the replacement $u_j^n = \frac{u_{j+1}^n + u_{j-1}^n}{2}$, this leads to

$$u_j^{n+1} = \frac{u_{j+1}^n + u_{j-1}^n}{2} - v\Delta t \left(\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} \right)$$

LAX method



Flux-Conservative

Lax method, stability.

$$u_j^{n+1} = \frac{u_{j+1}^n + u_{j-1}^n}{2} - v\Delta t \left(\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} \right)$$

as before, introduce $u_j^n = \xi^n e^{ikj\Delta x}$

$$\xi = \cos k\Delta x - i \frac{v\Delta t}{\Delta x} \sin k\Delta x$$

and the stability condition ($|\xi| < 1$) is satisfied if

$$\frac{|v|\Delta t}{\Delta x} < 1$$

Flux-Conservative

A little magic: how comes Lax work?

$$u_j^{n+1} = \frac{u_{j+1}^n + u_{j-1}^n}{2} - v\Delta t \left(\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} \right)$$

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = -v \left(\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} \right) + \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{2\Delta t}$$

This is the FTCS of the equation

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

Flux-Conservative

A little magic: how comes Lax work? (2)

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

Lax introduce a “numerical” dissipation. Is this good? In practice, $k\Delta x << 1$: Lax will damp modes such that $k\Delta x \approx 1$ which is OK. It is better to have a stable method which dumps the short wavelengths, rather than an unstable method. Both FTCS and Lax are inaccurate for short wavelengths, but the inaccuracy is tolerable for stable schemes.

Flux-Conservative

Many dimensions: given

$$\frac{\partial}{\partial t}\left[\begin{array}{c} r \\ s \end{array}\right] = -\frac{\partial}{\partial x}\left[\begin{array}{c} vs \\ vr \end{array}\right]$$

Lax becomes

$$r_j^{n+1} = \frac{1}{2}(r_{j+1}^n + r_{j-1}^n) + \frac{v\Delta t}{2\Delta x}(s_{j+1}^n - s_{j-1}^n)$$

$$s_j^{n+1} = \frac{1}{2}(s_{j+1}^n + s_{j-1}^n) + \frac{v\Delta t}{2\Delta x}(r_{j+1}^n - r_{j-1}^n)$$

Flux-Conservative

Many dimensions: stability?

$$\begin{bmatrix} r_j^n \\ s_j^n \end{bmatrix} = \xi^n e^{ikj\Delta x} \begin{bmatrix} r^0 \\ s^0 \end{bmatrix}$$

Substitute in the scheme, and the resulting equation for ξ reads

$$\xi = \cos k\Delta x \pm i \frac{v\Delta t}{\Delta x} \sin k\Delta x$$

Again, stable as long as $|v|\Delta t < \Delta x$

Flux-Conservative

Other possible errors, beside amplitude errors, are:

Dispersion $\xi = \exp(-ik\Delta x)$, it means that there is phase mixing

Nonlinear instability a nonlinear term can move energy in Fourier space (Navier-Stokes): in other words, cases when the short time scales are enhanced by the physics

Transport errors Lax is symmetric, but there may be cases when there is transport in one direction, hence the scheme should reflect this symmetry

Flux-Conservative

An example for the latter case is the following

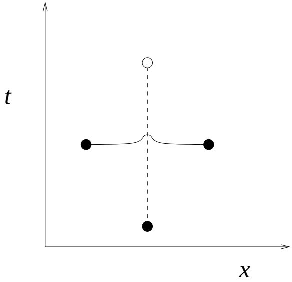
$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = -v_j^n \begin{cases} \frac{u_j^n - u_{j-1}^n}{\Delta x}, & v_j^n > 0 \\ \frac{u_{j+1}^n - u_j^n}{\Delta x}, & v_j^n < 0 \end{cases}$$

Although only first order in time and space, this method is superior to the Lax method (first order in time and second order in space) for asymmetric cases. The important message here is that "fidelity" is more important than "accuracy". It is interesting to work out the stability range (exercise!).

Flux-Conservative

The Lax method is only first order in time: this means that the limiting factor is really that $v\Delta t$ must be fairly smaller than Δx. Can we do better, ie use methods which are also second order in time? A way is staggered leap-frog:

$$u_j^{n+1} - u_j^{n-1} = -\frac{v\Delta t}{\Delta x}(u_{j+1}^n - u_{j-1}^n)$$



Flux-Conservative

Stability for the Staggered Leap Frog:

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

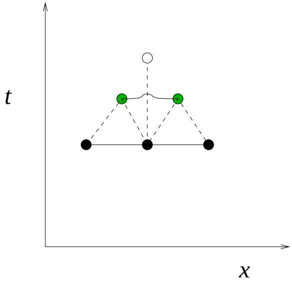
which yields $\xi = 1$ as long as $|v|\Delta t < \Delta x$. The method has no amplitude dissipation. There is only a snag, the method couples points like in a chessboard, so it is often necessary to add a small diffusion by hand.

Flux-Conservative

Two-Step Lax-Wendroff scheme: done using half steps

$$u_{j+1/2}^{n+1/2} = \frac{1}{2}(u_{j+1}^n - u_j^n) - \frac{\Delta t}{2\Delta x}(F_{j+1}^n - F_j^n)$$

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x}(F_{j+1/2}^{n+1/2} - F_{j-1/2}^{n+1/2})$$



Flux-Conservative

Stability for Lax-Wendroff: with the usual method, we find ($\alpha = v\Delta t/\Delta x$)

$$|\xi|^2 = 1 - \alpha^2(1-\alpha^2)(1-\cos k\Delta x)^2$$

which leads to $|\xi| \leq 1$ as long as $\alpha^2 \leq 1$. The damping here is vary small, though:

$$|\xi|^2 \approx 1 - \alpha^2(1-\alpha^2)\frac{(k\Delta x)^4}{4} + \ldots$$

whereas for Lax method we had

$$|\xi|^2 \approx 1 - (1-\alpha^2)(k\Delta x)^2 + \ldots$$

Schroedinger and split operators

A special case is Schroedinger equation (perhaps nonlinear one)

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

A possibility would be to use "standard" diffusion schemes, for instance

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

which is unconditionally stable. However, it is not unitary!

Schroedinger and split operators

Formally from $i\frac{\partial\psi}{\partial t} = H\psi$ we have

$$\psi(x,t) = e^{-iHt}\psi(x,0)$$

Use Cayley's form for the finite-difference representation of e^{-iHt}

$$e^{-iHt} \approx \frac{1 - iH\Delta t/2}{1 + iH\Delta t/2}$$

which yields the scheme

$$(1 + \frac{1}{2}iH\Delta t)\psi_j^{n+1} = (1 - \frac{1}{2}iH\Delta t)\psi_j^n$$

stable, unitary, 2nd order in space and time!

Schroedinger and split operators

A real case: 3+1 GPE. The GPE is NSE-like equation

$$i\frac{\partial\psi}{\partial t} = \left(-\frac{\nabla^2}{2m} + V(x) + g|\psi|^2\right)\psi$$

where the potential has the structure

$$V(x) = \frac{1}{2}m\omega_0^2\left(\lambda_1 x^2 + \lambda_2 y^2 + \lambda_3 z^2\right)$$

Schroedinger and split operators

We need a propagator which is stable, accurate, and keeps the modulus constant. From the Cayley form, we could use

$$(1 + \frac{1}{2} iH\Delta t)\psi_j^{n+1} = (1 - \frac{1}{2} iH\Delta t)\psi_j^{n}$$

this keep the modulus constant up to $O(h^3)$. But the problem is ∇^2! Having N bins in each direction implies solving a $N^3 \times N^3$ complex (non)linear system at each integration time step, although only $7N - 2D$ elements differs from zero.

Schroedinger and split operators

Use a split operator technique: break up the H into bits which are easily integrated and arrange things in such a way that the commutators are correct up to second order in the integration time step:

$$i\frac{\partial}{\partial t}\psi(\vec{r},t) = \left(H_x(\vec{r},t) + H_y(\vec{r},t) + H_z(\vec{r},t)\right)\psi(\vec{r},t)$$

where

$$H_i(\vec{r},t) \equiv -\frac{1}{2m}\frac{\partial^2}{\partial x_i^2} + V(x_i) + \frac{1}{3}g|\psi(\vec{r},t)|^2$$

Define $A_i(t) \equiv i\delta H_i(\vec{r},t)$

Schroedinger and split operators

A smart way to rearrange things is:

$$
\begin{aligned}
\psi(\vec{r}, t+\delta) \quad = \quad & \frac{1}{1 + A_y(t)/2}\,(1 - A_x(t)/2) \times \\
& \frac{1}{1 + A_z(t)/2}\,(1 - A_z(t)/2) \times \\
& \frac{1}{1 + A_x(t)/2}\,(1 - A_y(t)/2)\,\psi(\vec{r}, t).
\end{aligned}
$$

Use a PC (leap frog) to take care of the nonlinear term: at each step, solve a $N \times N$ tridiagonal system: the CPU is shorter by a factor $1/N^5$.

Diffusive Initial Value Problems

Start with the prototype equation

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

A difference scheme could be (FTCS)

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = D\left[\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x^2}\right]$$

This is OK, because there is implicit diffusion, in these problems. Of course $D > 0$ otherwise the method does not make sense.

Diffusive Initial Value Problems

Studying the stability we find

$$\xi = 1 - \frac{4D\Delta t}{\Delta x^2} \sin^2 \frac{k\Delta x}{2} \approx 1 - Dk^2\Delta t$$

which leads to the criterion

$$\frac{2D\Delta t}{\Delta x^2} \leq 1 \qquad \left(\frac{Dk^2\Delta t}{2} \leq 1 \quad k \to 0 \right)$$

Physically, the diffusion scale must be smaller than the cell spacing. There is a problem because we are interested in scales $\lambda >> \Delta x$, and the stability condition requires a step which typically is by far too small

Diffusive Initial Value Problems

So, we take typically a largish integration time step, knowing that the short scales (large k) will be poorly integrated, using some tricks to cure this. The tricks are twofold:

- Integration schemes which damp the short scales (fully implicit schemes, but only first order in Δt)
- Integration schemes which preserve the small scales amplitude, so that the large scales are integrated with a superimposed spurious "fluctuation" (Crank-Nicholson, second order in Δt)

The latter methods are typically ended with a few integrations of the former, to damp the short (spurious) scales.

Diffusive Initial Value Problems

Implicit schemes: start again from $\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2}$ and introduce the scheme

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = D\left[\frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{\Delta x^2}\right]$$

This is a fully implicit method, and it requires to solve a tridiagonal matrix equation

$$-\alpha u_{j-1}^{n+1} + (1+2\alpha)u_j^{n+1} - \alpha u_{j+1}^{n+1} = u_j^n \quad j = 1, ..., J-1$$

where $\alpha = D\Delta t/\Delta x^2$.

Diffusive Initial Value Problems

Implicit schemes

$$-\alpha u_{j-1}^{n+1} + (1+2\alpha)u_j^{n+1} - \alpha u_{j+1}^{n+1} = u_j^n \quad \alpha = D\Delta t/\Delta x^2$$

The amplification factor is

$$\xi = \frac{1}{1 + 4\alpha \sin^2 \frac{k\Delta x}{2}}$$

so the algorithm is unconditionally stable: wrong short-scale evolution particularly as Δt becomes larger, but correct equilibrium distribution

Diffusive Initial Value Problems

Second order The second approach is based on method which combine implicit (for stability) and explicit (to get 2nd order in time and space) schemes. Take the average of an implicit and an explicit FTCS scheme:

$$
\begin{aligned}
\frac{u_j^{n+1} - u_j^n}{\Delta t} &= \frac{D}{2}\left[\frac{(u_{j+1}^n - 2u_j^n + u_{j-1}^n)}{\Delta x^2}\right. \\
&+ \left.\frac{(u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1})}{\Delta x^2}\right]
\end{aligned}
$$

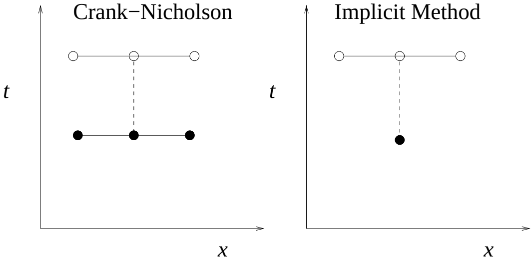
being centred at the timestep $n + 1/2$, the method is second order in time and space.

Diffusive Initial Value Problems

The amplification factor in this case is

$$\xi = \frac{1 - 2\alpha \sin^2 \frac{k\Delta x}{2}}{1 + 2\alpha \sin \frac{k\Delta x}{2}}$$

which means that the method is stable whatever the timestep.



The schemes in cartoons

Diffusive Initial Value Problems

A more general case is

$$\partial_t u = \partial_x D(x) \partial_x u$$

An obvius generalization of the FTCS is

$$\begin{aligned} \frac{u_j^{n+1} - u_j^n}{\Delta t} &= \frac{D_{j+1/2} \partial_x u_{j+1/2}^n - D_{j-1/2} \partial_x u_{j-1/2}^n}{\Delta x} \\ &= \frac{D_{j+1/2}(u_{j+1}^n - u_j^n) - D_{j-1/2}(u_j^n - u_{j-1}^n)}{\Delta x^2} \end{aligned}$$

with the definition $D_{j+1/2} = D(x_{j+1/2})$ etc.. Stable if

$$\Delta t \leq \min_j \Delta x^2 / (2 D_{j+1/2})$$

Diffusive Initial Value Problems

The Crank-Nicholson for the case

$$\partial_t u = \partial_x D(x) \partial_x u$$

is

$$\begin{aligned} \frac{u_j^{n+1} - u_j^n}{\Delta t} &= \frac{D_{j+1/2}(u_{j+1}^n - u_j^n) - D_{j-1/2}(u_j^n - u_{j-1}^n)}{2\Delta x^2} \\ &+ \frac{D_{j+1/2}(u_{j+1}^{n+1} - u_j^{n+1}) - D_{j-1/2}(u_j^{n+1} - u_{j-1}^{n+1})}{2\Delta x^2} \end{aligned}$$

with the definition $D_{j+1/2} = D(x_{j+1/2})$ etc.. Stable for any Δt.

Diffusive Initial Value Problems

Finally, the case when $D = D(u)$. Explicit methods simply replace

$$D_{j+1/2} = \frac{1}{2}\left[D(u^n_{j+1}) + D(u^n_j)\right]$$

Implicit methods are more involved, but typically the replacement

$$D(u_j^{n+1}) \approx D(u_j^n) + (u_j^{n+1} - u_j^n)\left.\frac{\partial D}{\partial u}\right|_{j,n}$$

is OK.

Multidimensions

More magic is needed.... let us see examples.

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

Taking $\Delta = \Delta x = \Delta y$ Lax becomes

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

Multidimensions

Stability: assume $F_\beta = v_\beta u$, the amplitude equation becomes

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

with $\alpha_\beta = v_\beta \Delta t/\Delta$. After some manipulation,

$$\Delta t \leq \frac{\Delta}{\sqrt{2}\sqrt{v_x^2 + v_y^2}}$$

$$\Delta t \leq \Delta/(\sqrt{N}|v|)$$

Multidimensions

Diffusion equations:

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

Crank-Nicholson is possible ($\alpha \equiv D\Delta t/\Delta^2$)

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

with

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

Multidimensions

Another possibility is to use a trick:

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

becomes

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

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

which is an example of split operator technique.

Fourier Methods

Start from

$$u_{jl} = \frac{1}{JL} \sum_m \sum_n \hat{u}_{mn} e^{-2\pi ijm/J} e^{-2\pi iln/L}$$

In the problem

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

we have after some algebra

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

Cyclic Reduction

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + b(y)\frac{\partial u}{\partial y} + c(y)u = g(x.y)$$

Writing the finite difference

$$\mathbf{u}_{j-1} + \mathbf{T}\mathbf{u}_j + \mathbf{u}_{j+1} = \Delta^2 \mathbf{g}_j$$

j is the differentiation along x, the one along y is in vector form, and we have

$$\mathbf{T} = \mathbf{B} - 2\mathbf{I}$$

Example is

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

Cyclic Reduction

Write three lines of CR

$$\mathbf{u}_{j-2} + \mathbf{T}\mathbf{u}_{j-1} + \mathbf{u}_j = \Delta^2 \mathbf{g}_{j-1}$$

$$\mathbf{u}_{j-1} + \mathbf{T}\mathbf{u}_j + \mathbf{u}_{j+1} = \Delta^2 \mathbf{g}_j$$

$$\mathbf{u}_j + \mathbf{T}\mathbf{u}_{j+1} + \mathbf{u}_{j+2} = \Delta^2 \mathbf{g}_{j+1}$$

Multiply the second by $-\mathbf{T}$ and add, result is

$$\mathbf{u}_{j-2} + \mathbf{T}^{(1)}\mathbf{u}_j + \mathbf{u}_{j+2} = \Delta^2 \mathbf{g}_j^{(1)}$$

and this can be iterated. What are $\mathbf{T}^{(1)}$ and $\mathbf{g}_j^{(1)}$?

Cyclic Reduction

From the algebra,

$$\mathbf{T}^{(1)} = 2\mathbf{I} - \mathbf{T}^2$$

$$\mathbf{g}_j^{(1)} = \mathbf{g}_{j-1} - \mathbf{T}\mathbf{g}_j + \mathbf{g}_{j+1}$$

The procedure is iterated, until we have

$$\mathbf{T}^{(f)}\mathbf{u}_{J/2} = \Delta^2 \mathbf{g}_{J/2}^{(f)} - \mathbf{u}_0 - \mathbf{u}_J$$

where $\mathbf{u}_0$ and $\mathbf{u}_J$ are the boundaries. Now, the system is solved for $\mathbf{u}_{J/2}$, then with $\mathbf{u}_{J/2}$ and $\mathbf{u}_0$, we can find $\mathbf{u}_{J/4}$ etc.

Relaxation Methods redux

A nice way to solve $\mathcal{L}u = \rho$ is to think of these methods as

$$\frac{\partial u}{\partial t} = \mathcal{L}u - \rho$$

All the methods we studied for diffusion problems can be applied. For instance, take the system $\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \rho(x, y)$ using FTCS, and taking the largest possible time step ($=\Delta^2/4$) we have (Jacobi)

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

Relaxation Methods redux

Another possibility is Gauss-Seidel,

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

Both methods are slow. Write $\mathbf{Ax} = \mathbf{b}$, and split $\mathbf{A} = \mathbf{D} + \mathbf{U} + \mathbf{L}$, Jacobi means

$$\mathbf{Dx}^r = -(\mathbf{L} + \mathbf{U})\mathbf{x}^{r-1} + \mathbf{b}$$

The largest eigenvalue of $-\mathbf{D}^{-1}(\mathbf{L} + \mathbf{U})$ control the convergency. Typically, the number of iterations goes like J^2.

Relaxation Methods redux

Gauss-Seidel is also slow

$$(\mathbf{L} + \mathbf{D})\mathbf{x}^r = -(\mathbf{U})\mathbf{x}^{r-1} + \mathbf{b}$$

The largest eigenvalue of $-(\mathbf{D} + \mathbf{L})^{-1}(\mathbf{U})$ control the convergency. Typically, the number of iterations goes like J^2, and it is only approximatively half of the number of steps in Jacobi.

Relaxation Methods redux

SOR:

$$\mathbf{x}^r = \mathbf{x}^{r-1} - (\mathbf{L} + \mathbf{D})^{-1}[(\mathbf{L} + \mathbf{D} + \mathbf{U})\mathbf{x}^{r-1} - \mathbf{b}]$$

This can be written as (residual vector ξ^{r-1})

$$\mathbf{x}^r = \mathbf{x}^{r-1} - (\mathbf{L} + \mathbf{D})^{-1}\xi^{r-1}$$

Iterate then using

$$\mathbf{x}^r = \mathbf{x}^{r-1} - \omega(\mathbf{L} + \mathbf{D})^{-1}\xi^{r-1}$$

and make $\omega > 1$, so that the iterations converge more quickly.