17_PDEs_comments

April 17, 2019

1 17 PDEs: Miscellaneous comments on PDEs

1.1 Origin of the Jacobi algorithm in a diffusion problem

(see e.g., *Numerical Recipes in C*, Ch 19.5)

We want to solve the Laplace equation

$$\nabla^2 u = 0$$

with given boundary conditions.

Rewrite as a *diffusion equation* (with $D = 1$):

$$\nabla^2 u = \frac{\partial u}{\partial t}$$

The **equilibrium solution** of the diffusion equation (i.e., after very long t when relaxed to equilibrium)

$$\lim_{t \to +\infty} u(\mathbf{x}, t) \equiv u(\mathbf{x})$$

is the solution of the boundary value problem.

1.2 Jacobi algorithm rediscovered

Set up the finite difference scheme for the 2D diffusion equation, using $u(x, y, t) = u(i\Delta, j\Delta, n\Delta t) = u_{ij}^n$:

$$\frac{1}{\Delta^2}\Big[(u_{i+1,j}^n - 2u_{ij}^n + u_{i-1,j}^n) + (u_{i,j+1}^n - 2u_{ij}^n + u_{i,j-1}^n)\Big] \\ = \frac{1}{\Delta t}(u_{ij}^{n+1} - u_{ij}^n)$$

In 2D, this leap frog algorithm is stable for

$$\frac{\Delta t}{\Delta^2} \leq \frac{1}{4}$$

Let's assume that we can push our solution to the edge of stability and use $\frac{\Delta t}{\Delta^2} = \frac{1}{4}$. Then the finite difference scheme becomes

$$\frac{1}{4}\Big(u_{i+1,j}^n + u_{i-1,j}^n) + (u_{i,j+1}^n + u_{i,j-1}^n\Big) - u_{ij}^n = u_{ij}^{n+1} - u_{ij}^n$$

or when collecting terms, the **Jacobi algorithm** re-emerges

$$u_{ij}^{n+1} = \frac{1}{4}\left(u_{i+1,j}^{n} + u_{i-1,j}^{n} + u_{i,j+1}^{n} + u_{i,j-1}^{n}\right)$$

where you now interpret n as the step number in the iterations towards convergence.

1.3 Broad classes of physical behavior

(For real fields $u(\mathbf{x}, t)$.)

- elliptic PDE without time dependence: static boundary value problem (static, think $F = 0$)
- parabolic PDE with first time derivative: diffusion problem (friction, think $F = -bv$)
- hyperbolic PDE with second time derivative: wave problem (oscillator, think $F = ma$)

In [ ]: