Lecture 6, Sci. Comp. for DPhil Students II

Nick Trefethen, Tuesday 11.02.20

Last lecture

- V.1 PDEs in science and engineering
- V.2 Explicit 1D finite differences

Today

- V.3 Dispersion relations and numerical instability
- V.4 Implicit 1D finite differences

Assignment 2 due now

Please do the quiz.

Handouts:

- Solutions to Assmt. 2
- PDE quiz
- `m40_fourthorderdiffusion.m` & `m41_implicit.m` - 4th-order diffusion, explicit & implicit
- Kuramoto-Sivashinsky equation from the *PDE Coffee Table Book*
- `m42_kuramotosivashinsky`.m and `m42chebfun.m` — Kuramoto-Sivashinsky eq.

Pass around Iserles and LeVeque books (again)

V.3 Dispersion relations and numerical instability

Last lecture we looked at the heat equation, $u_t = u_{xx}$.

Now let's consider a 4th-order diffusion equation:

$$u_t = -u_{xxxx}, \qquad u(-1) = u(1) = u'(-1) = u'(1) = 0.$$

The minus sign is appropriate to ensure positive diffusion, as one can show by Fourier analysis. We ask, what if at time t we have a sine wave $u(x,t) = \exp(i\xi x)$ for some wave number ξ? (I use the term "sine" loosely.) Then $u_{xx} = -\xi^2 u$ and $u_{xxxx} = \xi^4 u$. In other words, for the 2nd and 4th order diffusion equations we will have solutions $u(x,t) = \exp(i\xi x + \sigma t)$ with $\sigma = -\xi^2$ and $\sigma = \xi^4$, repectively. Such a relation between σ and ξ is called a **dispersion relation**. (More commonly, instead of σ, one would work with the frequency $i\sigma$.)

The simplest finite difference approximation to a 4th derivative looks like this:

$$u_{xxxx} = (u_{xx})_{xx} \approx \left(\frac{v_{j+2} - 2v_{j+1} + v_j}{h^2} - 2\frac{v_{j+1} - 2v_j + v_{j-1}}{h^2} + \frac{v_j - 2v_{j-1} + v_{j-2}}{h^2} \right) \Big/ h^2,$$

which simplifies to

$$\frac{v_{j+2} - 4v_{j+1} + 6v_j - 4v_{j-1} + v_{j-2}}{h^4}.$$

This suggests the simple explicit finite difference formula

$$v_j^{n+1} = v_j^n - \frac{k}{h^4}(v_{j+2}^n - 4v_{j+1}^n + 6v_j^n - 4v_{j-1}^n + v_{j-2}^n).$$

At a linear algebra level, this takes the form

$$v^{n+1} = Av^n$$

where A is a pentadiagonal Toeplitz matrix (i.e., constant along diagonals).

Here's a program like `m36_heat.m`, but modified for this 4th-order eq.:

[`m40_fourthorderdiffusion.m`]

We try different time steps and discover it's unstable unless k is extremely small — less than around 4.8×10^{-8}.

How can we explain the need for such a small time step?

The trick is **von Neumann analysis** of the finite difference formula, also known as **discrete Fourier analysis**, developed in the late 1940s.

We ask, what if at step n we have a sine wave

$$v_j^n = \exp(i\xi x_j) = \exp(i\xi jh)$$

for some wave number ξ? At the next time step, it will have been multiplied by some constant g, the **amplification factor** for this ξ. We compute $g = g(\xi)$ by plugging into the formula:

$$g = g(\xi) = 1 - \frac{k}{h^4}\left(e^{2i\xi h} - 4e^{i\xi h} + 6 - 4e^{-i\xi h} + e^{-2i\xi h}\right)$$

which can be simplified to

$$g(\xi) = 1 - \frac{k}{h^4}\left(e^{i\xi h/2} - e^{-i\xi h/2}\right)^4,$$

that is,

$$g(\xi) = 1 - \frac{k}{h^4}\left(2i\sin(\xi h/2)\right)^4 = 1 - \frac{16k}{h^4}\left(\sin(\xi h/2)\right)^4.$$

As ξ ranges over all possible values, the fourth power ranges over $[0,1]$. So we have

$$1 - 16k/h^4 \leq g(\xi) \leq 1.$$

A mode will blow up if $|g(\xi)| > 1$. Thus for stability we want to ensure $|g(\xi)| \leq 1$ for all ξ, i.e.

$$1 - \frac{16k}{h^4} \geq -1, \qquad \text{i.e.,} \qquad 2 \geq \frac{16k}{h^4},$$

or in other words

$$k \le \frac{h^4}{8}.$$

That's a very tight stability restriction! For $h = 0.025$, as in `m39_fourthorderdiffusion.m`, it gives

$$k \le 4.883 \times 10^{-8}.$$

This matches our experiment convincingly, but confirms that this finite difference formula will be expensive in practice.

This PDE is **stiff** — widely different time scales are present. That's why forward differencing is no good. We'll say more about this ODE/stiffness point of view next lecture.

V.4 Implicit 1D finite differences

It's surprisingly easy to cure the instability, at least for this PDE. We need an **implicit** formula, coupling adjacent values with respect to x at time step $n+1$:

$$v_j^{n+1} = v_j^n - \frac{k}{h^4}(v_{j+2}^{n+1} - 4v_{j+1}^{n+1} + 6v_j^{n+1} - 4v_{j-1}^{n+1} + v_{j-2}^{n+1})$$

almost exactly as before, but with the crucial difference $n \to n+1$. Now the amplification factor becomes

$$g = g(\xi) = \frac{1}{1 + (16k/h^4)(\sin(\xi h/2))^4},$$

which is ≤ 1 for all ξ.

At the linear algebra level, we now have

$$Bv^{n+1} = v^n, \quad \text{i.e.,} \quad v^{n+1} = B^{-1}v^n.$$

The modification of the program is easy:

[`m41_implicit.m`]

Now any time step will work! (Of course, if k is too big the accuracy will suffer.)

Nonlinear example: **Kuramoto-Sivashinsky equation**

[page from *PDE Coffee Table Book*]

$$u_t = -u_{xx} - u_{xxxx} - (u^2/2)_x.$$

This physics/mathematics of this equation is fascinating. Consider the linear part of the equation,

$$u_t = -u_{xx} - u_{xxxx}$$

and insert the ansatz $u(x,t) = \exp(i\xi x + \sigma t)$, You get the dispersion relation

$$\sigma = \xi^2 - \xi^4.$$

For $|\xi| < 1$, $\sigma > 0$: exponential growth as t increases.

For $|\xi| > 1$, $\sigma < 0$: exponential decay.

The fascinating thing is that the nonlinear term moves energy from small ξ to large ξ. So we have an engine: energy is amplified at low wave numbers, converted nonlinearly to high wave numbers, and then absorbed. The result is chaotic evolution dominated by structures with $\xi = O(1)$.

To solve the equation numerically we try discretizing the linear terms by backward Euler and the nonlinear term by Euler. It works! — and is chaotic.

[`m42_kuramotosivashinsky.m`]

We can also use the `spin` command in Chebfun: `m42chebfun.m`

Chebfun makes this available via a built-in demo. Try

```
spin('ks')
```

or to run the same demo on your own,

```
S = spinop('ks')
spin(S,256,.01)
```