Lecture 9, Sci. Comp. for DPhil Students II

Nick Trefethen, Thursday 20.02.20

Last lecture

- V.7 Finite differencing in general grids
- V.8 Multiple space dimensions

Today

- V.9 Fourier spectral discretisation
- V.10 Fourier spectral discretisation via FFT

Handouts

- Gray-Scott equations from *PDE Coffee Table Book* (cf. Assmt. 3)
- 1D wave equation from the *PDE Coffee Table Book*
- nD wave equation from the *PDE Coffee Table Book*
- `m50_waveeq.m` - wave equation by finite diffs of order 2,4,6
- `m51_waveeqFourier.m` - wave equation by Fourier spectral method (matrix)
- `m52_waveeqFFT.m` - wave equation by Fourier spectral method (FFT)
- One-way wave equation from the *PDE Coffee Table Book*

Pass around: *Spectral Methods in MATLAB* (available online through Bodleian)

Assignment 3 is due next Tuesday.

V.9 Fourier spectral discretisation

Hand out 1D and nD pages from the *PDE Coffee Table Book*.

We've discussed how to derive finite difference approximations of high order: interpolate data in a largish number of points by a polynomial of suitable degree, then differentiate the interpolant.

Here's a code to illustrate such derivatives in action. It solves the wave equation

$$u_{tt} = u_{xx}, \quad x \in [-\pi, \pi], \quad \text{periodic BC's}$$

with spatial discretisation of order 2, 4 or 6.

[1D wave equation from the *PDE Coffee Table Book*]

[`m50_waveeq.m`]

The effect we see in these experiments is **dispersion**. One can quantify it beautifully. The wave equation admits solns

$$u(x,t) = e^{i(\omega t + \xi x)}$$

for ω and ξ related by the **dispersion relation**

$$\omega^2 = \xi^2, \quad \text{i.e.,} \quad \omega = \pm\xi,$$

but the 2nd-order leap frog discretisation replaces this by

$$\sin^2 \frac{\omega k}{2} = \frac{k^2}{h^2} \sin^2 \frac{\xi h}{2}$$

(sketch). From this one can study phase velocity, group velocity, etc. See Trefethen, "Group velocity in finite difference schemes", *SIAM Review* 24 (1982), 113–136.

These dispersive effects in finite-difference grids are analogous to such effects in crystals, which also have a regular lattice.

Now, what if we let the order of the finite difference formula approach infinity? We get **spectral methods**. The simplest flavours are:

- Periodic domains: **Fourier spectral methods**
- Non-periodic domains: **Chebyshev spectral methods**.

Today we'll discuss the former.

In the limit of infinite order, those finite differences approach the infinite **Laurent matrix** (or **Laurent operator**) with coefficients

$$D = h^{-1} \begin{pmatrix} \cdots & \frac{-2}{16} & \frac{2}{9} & \frac{-2}{4} & 2 & \frac{-\pi^2}{3} & 2 & \frac{-2}{4} & \frac{2}{9} & \frac{-2}{16} & \cdots \end{pmatrix}$$

where $-\pi^2/3$ is on the main diagonal. The structure here is that this is a doubly-infinite matrix that is constant along diagonals.

For a finite matrix with $h = 2\pi/N$, the formula is

$$D = \frac{1}{2} \begin{pmatrix} \cdots & \left[\frac{-2\pi^2}{3h^2} - 1/3\right] & \csc^2(h/2) & -\csc^2(2h/2) & \csc^2(3h/2) & \cdots \end{pmatrix}$$

where the cosecant is defined as always by $\csc(t) = 1/\sin(t)$. Again the term with π in it is on the main diagonal.

That is, suppose:

$$v = \text{vector of data on the periodic grid}$$

$$w = \text{vector of spectral approximations to } v' \text{ on the grid}$$

Then

$$w = Dv \qquad \text{(draw this matrix)}$$

D is a **spectral differentiation matrix**.

For derivations and details, see LNT, *Spectral Methods in MATLAB*, available online through Oxford e-books. The above matrix is on p. 23.

[Pass around *Spectral Methods in MATLAB.*]

Here's the idea that leads to such formulas, the fundamental idea of **spectral collocation methods**.

1. Interpolate data by a **global** interpolant (a periodic trigonometric polynomial)

$$p(x) = \sum_{j=-N/2}^{N/2} a_j e^{ijx}$$

2. Differentiate $p(x)$ and evaluate at the grid points.

Note that both notations ξ and j have appeared. The reason is that our interval has length 2π, so the wave numbers ξ that fit in it are the integers $0, \pm 1, \pm 2, \ldots$. On an interval of length $L \neq 2\pi$, we would need to use other wave numbers, generally not integers.

[`m51_waveeqFourier.m`]

V.10 Fourier spectral discretisation via FFT

FFT = Fast Fourier Transform, i.e., a fast algorithm for computing the discrete Fourier transform. (The FFT was discovered first by Gauss in 1805 and last by Cooley & Tukey in 1965. There were half a dozen discoverers in-between, including Runge and Lanczos.)

If $u(x) = e^{ijx}$, then $u'(x) = iju(x)$.

More generally, suppose $u(x)$ is a superposition of exponentials,

$$u(x) = \sum_j U_j e^{ijx}$$

($U = \hat{u}$ is the **discrete Fourier transform** of u.) Then

$$u'(x) = \sum_j ijU_j e^{ijx}, \qquad u''(x) = \sum_j -j^2 U_j e^{ijx},$$

and so on. Thus differentiation in space is equivalent to multiplication by ij in Fourier space. This suggests an alternative method for computing a Fourier 2nd spectral derivative:

1. Given u, compute its DFT `U = fft(u)` [MATLAB notation]
2. Multiply by $-j^2$: $W(j) = -j^2 U_j$
3. Take the inverse transform: `w = ifft(W)`.

Similar but fancier manipulation of Fourier transforms leads to the idea of **one-way wave equations** — see further handout from *PDE Coffee Table Book.*

[`m52_waveeqFFT.m`]

If time permits, explore the `trig` option in Chebfun (periodic functions represented by trigonometric, i.e. Fourier, interpolants).