Dr H. Fawzi

Mathematical Tripos Part II: Michaelmas Term 2023

Numerical Analysis – Lecture 3

Let $\widehat{u}$ be the exact solution of the Poisson equation, and let $\widehat{u}_{i,j} = \widehat{u}(ih, jh)$ be its values on the grid. Let

$$e_{i,j} = \widehat{u}_{i,j} - u_{i,j} \tag{1.7}$$

be the pointwise error of the 5-point formula. Set $\boldsymbol{e} = (e_{i,j}) \in \mathbb{R}^N$ where $N = M^2$, and for $\boldsymbol{x} \in \mathbb{R}^N$ let $\|\boldsymbol{x}\|_2$ be the Euclidean norm of the vector $\boldsymbol{x}$:

$$\|\boldsymbol{x}\|_2^2 = \sum_{k=1}^{N} |x_k|^2 = \sum_{i=1}^{M}\sum_{j=1}^{M} |x_{i,j}|^2.$$

Theorem 1.11 *Assume the solution $\widehat{u}$ of Poisson's equation is C^4 and let*

$$c = \tfrac{1}{12} \max_{0<x,y<1} \left|\frac{\partial^4 \widehat{u}}{\partial x^4}(x,y)\right| + \left|\frac{\partial^4 \widehat{u}}{\partial y^4}(x,y)\right| > 0. \tag{1.8}$$

Then the error vector $\boldsymbol{e}$ *defined in* (1.7) *satisfies*

$$\|\boldsymbol{e}\|_2 \le (c/8)h\,.$$

Proof. For a C^4 univariate function $g : (a,b) \to \mathbb{R}$, the finite-difference approximation of $g''(x)$ for $x \in (a+h, b-h)$ satisfies

$$|g''(x) - (g(x+h) + g(x-h) - 2g(x))/h^2| \le \frac{h^2}{12} \max_{\xi \in (x-h,x+h)} |g^{(iv)}(\xi)|.$$

Applied to the Laplacian of a C^4 bivariate function $u(x,y)$ we get

$$\begin{aligned} &|\nabla^2 u(x,y) - (u(x+h,y) + u(x-h,y) + u(x,y+h) + u(x,y-h) - 4u(x,y))/h^2| \\ &\le \frac{h^2}{12} \max_{\substack{\xi\in(x-h,x+h)\\ \kappa\in(y-h,y+h)}} |\frac{\partial^4 u}{\partial x^4}(\xi,\kappa)| + |\frac{\partial^4 u}{\partial y^4}(\xi,\kappa)|. \end{aligned}$$

1) Since $\hat{u}$ is the exact solution of Poisson's equation, we know that $\nabla^2 \hat{u}(ih, jh) = f_{ij}$ for all $1 \le i,j \le M$. Replacing the left-hand side with the five-point approximation, and using the error bound above we can write:

$$\widehat{u}_{i-1,j} + \widehat{u}_{i+1,j} + \widehat{u}_{i,j-1} + \widehat{u}_{i,j+1} - 4\widehat{u}_{i,j} = h^2 f_{i,j} + \eta_{i,j}, \qquad |\eta_{i,j}| \le ch^4 \tag{1.9}$$

where c is as defined in (1.8).

The solution of the five-point method u satisfies, for all $1 \le i,j \le M$:

$$u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - 4u_{i,j} = h^2 f_{i,j}. \tag{1.10}$$

Subtracting (1.10) from (1.9), we obtain

$$e_{i-1,j} + e_{i+1,j} + e_{i,j-1} + e_{i,j+1} - 4e_{i,j} = \eta_{i,j}$$

or, in the matrix form, $A\boldsymbol{e} = \boldsymbol{\eta}$, where A is symmetric (negative definite). It follows that

$$A\boldsymbol{e} = \boldsymbol{\eta} \quad \Rightarrow \quad \boldsymbol{e} = A^{-1}\boldsymbol{\eta} \quad \Rightarrow \quad \|\boldsymbol{e}\|_2 \le \|A^{-1}\|\,\|\boldsymbol{\eta}\|_2,$$

where $\|A^{-1}\|$ is *operator norm* (also known as the *spectral norm*) of A^{-1} defined as $\|A^{-1}\| = \max_{\boldsymbol{x}\neq 0} \|A^{-1}\boldsymbol{x}\|_2/\|\boldsymbol{x}\|_2$.

2) Since every component of $\boldsymbol{\eta}$ satisfies $|\eta_{i,j}|^2 \le c^2 h^8$, where $h = \frac{1}{M+1}$, and there are M^2 components, we have

$$\|\boldsymbol{\eta}\|_2^2 = \sum_{i=1}^{M}\sum_{j=1}^{M} |\eta_{i,j}|^2 \le c^2 M^2 h^8 < c^2 \,\tfrac{1}{h^2}\, h^8 = c^2 h^6 \quad\Rightarrow\quad \|\boldsymbol{\eta}\|_2 \le c h^3.$$

3) The matrix A is symmetric, hence so is A^{-1} and therefore $\|A^{-1}\| = \rho(A^{-1})$. Here $\rho(A^{-1})$ is the spectral radius of A^{-1}, that is $\rho(A^{-1}) = \max_i |\lambda_i|$, where λ_i are the eigenvalues of A^{-1}. The eigenvalues of A^{-1} are the reciprocals of the eigenvalues of A, and the latter are given by Proposition 1.12. Thus, using the fact that $\sin(\pi h/2) \ge 1$ for $h \le 1$ we get

$$\|A^{-1}\| \;=\; \tfrac{1}{4} \max_{k,\ell=1\ldots m} \left(\sin^2 \tfrac{k\pi h}{2} + \sin^2 \tfrac{\ell \pi h}{2}\right)^{-1} \;=\; \frac{1}{8\sin^2(\frac{1}{2}\pi h)} \;\le\; \frac{1}{8h^2}.$$

Therefore $\|\boldsymbol{e}\|_2 \le \|A^{-1}\|\,\|\boldsymbol{\eta}\|_2 \le (c/8)h$ as desired. □

Fast Poisson solvers Suppose that we are solving $\nabla^2 u = f$ in a square $M \times M$ grid with the 5-point formula. Let the grid be enumerated in as before, i.e., by columns. Thus, the linear system $A\boldsymbol{u} = \boldsymbol{b}$ can be written explicitly in the block form

$$\underbrace{\begin{bmatrix} H & I & & \\ I & H & \ddots & \\ & \ddots & \ddots & I \\ & & I & H \end{bmatrix}}_{A} \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \\ \vdots \\ \boldsymbol{u}_M \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \vdots \\ \boldsymbol{b}_M \end{bmatrix}, \qquad H = \begin{bmatrix} -4 & 1 & & \\ 1 & -4 & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & -4 \end{bmatrix}_{M\times M},$$

where $\boldsymbol{u}_k, \boldsymbol{b}_k \in \mathbb{R}^M$ are portions of $\boldsymbol{u}$ and $\boldsymbol{b}$, respectively, and B is a TST-matrix which means *tridiagonal*, *symmetric* and *Toeplitz* (i.e., constant along diagonals). By Exercise 4, its eigenvalues and orthonormal eigenvectors are given as

$$H\boldsymbol{q}_\ell = \lambda_\ell \boldsymbol{q}_\ell, \qquad \lambda_\ell = -4 + 2\cos\tfrac{\ell\pi}{M+1}, \qquad \boldsymbol{q}_\ell = \gamma_M \big(\sin \tfrac{j\ell\pi}{M+1}\big)_{j=1}^{M}, \qquad \ell = 1..M,$$

where $\gamma_M = \sqrt{\frac{2}{M+1}}$ is the normalization factor. Hence $H = QDQ^{-1} = QDQ$, where $D = \mathrm{diag}\,(\lambda_\ell)$ and $Q = Q^T = (q_{j\ell})$. Note that all $M \times M$ TST matrices share the same full set of eigenvectors, hence they all commute!

Hockney method Set $\boldsymbol{v}_k = Q\boldsymbol{u}_k$, $\boldsymbol{c}_k = Q\boldsymbol{b}_k$, therefore our system becomes

$$\begin{bmatrix} D & I & & \\ I & D & \ddots & \\ & \ddots & \ddots & I \\ & & I & D \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \\ \vdots \\ \boldsymbol{v}_M \end{bmatrix} = \begin{bmatrix} \boldsymbol{c}_1 \\ \boldsymbol{c}_2 \\ \vdots \\ \boldsymbol{c}_M \end{bmatrix}.$$

Let us by this stage reorder the grid *by rows, instead of by columns.*. In other words, we permute $\boldsymbol{v} \mapsto \widehat{\boldsymbol{v}} = P\boldsymbol{v}$, $\boldsymbol{c} \mapsto \widehat{\boldsymbol{c}} = P\boldsymbol{c}$, so that the portion $\widehat{\boldsymbol{c}}_1$ is made out of the first components of the portions $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_M$, the portion $\widehat{\boldsymbol{c}}_2$ out of the second components and so on. This results in new system

$$\begin{bmatrix} \Lambda_1 & & & \\ & \Lambda_2 & & \\ & & \ddots & \\ & & & \Lambda_M \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{v}}_1 \\ \widehat{\boldsymbol{v}}_2 \\ \vdots \\ \widehat{\boldsymbol{v}}_M \end{bmatrix} = \begin{bmatrix} \widehat{\boldsymbol{c}}_1 \\ \widehat{\boldsymbol{c}}_2 \\ \vdots \\ \widehat{\boldsymbol{c}}_M \end{bmatrix}, \qquad \Lambda_k = \begin{bmatrix} \lambda_k & 1 & & \\ 1 & \lambda_k & 1 & \\ & \ddots & \ddots & \ddots \\ & & 1 & \lambda_k \end{bmatrix}_{M\times M}, \qquad k = 1...M.$$

These are M *uncoupled* systems, $\Lambda_k \widehat{\boldsymbol{v}}_k = \widehat{\boldsymbol{c}}_k$ for $k = 1...M$. Being *tridiagonal*, each such system can be solved fast, at the cost of $\mathcal{O}(M)$. Thus, the steps of the algorithm and their computational cost are as follows.

1. Form the products $\boldsymbol{c}_k = Q\boldsymbol{b}_k$, $\quad k = 1...M$ .... $\mathcal{O}(M^3)$
2. Solve $M \times M$ tridiagonal systems $\Lambda_k \widehat{\boldsymbol{v}}_k = \widehat{\boldsymbol{c}}_k$, $\quad k = 1...M$ .... $\mathcal{O}(M^2)$
3. Form the products $\boldsymbol{u}_k = Q\boldsymbol{v}_k$, $\quad k = 1...M$ .... $\mathcal{O}(M^3)$

(Permutations $\boldsymbol{c} \mapsto \widehat{\boldsymbol{c}}$ and $\widehat{\boldsymbol{v}} \mapsto \boldsymbol{v}$ are basically free.)

Improved Hockney algorithm We observe that the computational bottleneck is to be found in the $2M$ *matrix-vector products by the matrix* Q. Recall further that the elements of Q are $q_{j\ell} = \gamma_M \sin \frac{\pi j \ell}{M+1}$. This special form lends itself to a considerable speedup in matrix multiplication. Before making the problem simpler, however, let us make it more complicated! We write a typical product in the form

$$(Q\boldsymbol{y})_\ell = \sum_{j=1}^{M} \sin \frac{\pi j \ell}{M+1} y_j = \operatorname{Im} \sum_{j=0}^{M} \exp \frac{\mathrm{i}\pi j \ell}{M+1} y_j = \operatorname{Im} \sum_{j=0}^{2M+1} \exp \frac{2\mathrm{i}\pi j \ell}{2M+2} y_j, \quad \ell = 1...M, \tag{1.11}$$

where $y_{M+1} = \cdots = y_{2M+1} = 0$.

The discrete Fourier transform (DFT) The *discrete Fourier transform* of a vector $y \in \mathbb{C}^n$ is $x = \mathcal{F}_n y$ defined by

$$x_\ell = \sum_{j=0}^{n-1} \omega_n^{j\ell} y_j \quad \ell = 0, \ldots, n-1$$

where $\omega_n = \exp(2i\pi/n)$. (We assume in the above that vectors are indexed from 0 to $n-1$.) Thus, we see that multiplication by Q in (1.11) can be reduced to calculating a DFT. In the next lecture, we see how to compute the DFT of a vector y in $\mathcal{O}(n \log n)$ operations, instead of $\mathcal{O}(n^2)$.