Dr H. Fawzi

Mathematical Tripos Part II: Michaelmas Term 2022

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 $\|\boldsymbol{x}\| = \|\boldsymbol{x}\|_{\ell_2}$ be the Euclidean norm of the vector $\boldsymbol{x}$:

$$\|\boldsymbol{x}\|^2 = \sum_{k=1}^{n} |x_k|^2 = \sum_{i=1}^{m}\sum_{j=1}^{m} |x_{i,j}|^2 .$$

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 $\mathbf{e}$ *defined in* (1.7) *satisfies*

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

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

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

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

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

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,

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

Therefore $\|\boldsymbol{e}\| \le \|A^{-1}\|\,\|\boldsymbol{\eta}\| \le ch$ for some constant $c > 0$. □

Observation 1.12 (Special structure of 5-point equations) We wish to motivate and introduce a family of efficient solution methods for the 5-point equations: the *fast Poisson solvers.* Thus, suppose that we are solving $\nabla^2 u = f$ in a square $m \times m$ grid with the 5-point formula (all this can be generalized a great deal, e.g. to the nine-point formula). Let the grid be enumerated in *natural ordering,* i.e. by columns. Thus, the linear system $A\boldsymbol{u} = \boldsymbol{b}$ can be written explicitly in the block form

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

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

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

where $\gamma_m = \sqrt{\frac{2}{m+1}}$ is the normalization factor. Hence $B = QDQ^{-1} = QDQ$, where $D = \operatorname{diag}(\lambda_\ell)$ 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!

Method 1.13 (The 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.)

Method 1.14 (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

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

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 $\mathcal{O}(n^2)$.