Dr J Lellmann

Mathematical Tripos Part II: Michaelmas Term 2014

Numerical Analysis – Lecture 3

Let $\widehat{u}_{i,j} = u(ih, jh)$ be the grid values of the exact solution of the Poisson equation, and let $e_{i,j} = u_{i,j} - \widehat{u}_{i,j}$ be the pointwise error of the 5-point formula. Set $\boldsymbol{e} = (e_{i,j}) \in \mathbb{R}^{m^2}$.

Theorem 1.14 *Subject to sufficient smoothness of the function f and of the boundary conditions, there exists a number $c > 0$, independent of $h = \frac{1}{m+1}$, such that*

$$\|\boldsymbol{e}\| \le ch\,.$$

Proof. 1) We already know (having constructed the 5-point formula by matching Taylor expansions) that

$$\widehat{u}_{i-1,j} + \widehat{u}_{i+1,j} + \widehat{u}_{i,j-1} + \widehat{u}_{i,j+1} - 4\widehat{u}_{i,j} = h^2 f_{i,j} + \eta_{i,j}, \qquad \eta_{i,j} = \mathcal{O}(h^4).$$

Subtracting this from (1.5), 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 1.15 (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

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

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

Method 1.16 (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

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

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 1.17 (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 $q_{\ell j} = \gamma_m \sin \frac{\pi \ell j}{m+1}$. 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

$$(Q\boldsymbol{y})_\ell = \sum_{j=1}^{m} \sin \frac{\pi \ell j}{m+1} y_j = \operatorname{Im} \sum_{j=0}^{m} \exp \frac{\mathrm{i}\pi \ell j}{m+1} y_j = \operatorname{Im} \sum_{j=0}^{2m+1} \exp \frac{2\mathrm{i}\pi \ell j}{2m+2} y_j, \quad \ell = 1, \ldots, m, \tag{1.8}$$

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

Problem 1.18 (The discrete Fourier transform) Let Π_n be the space of all *bi-infinite complex* n-*periodic sequences* $\boldsymbol{x} = \{x_\ell\}_{\ell \in \mathbb{Z}}$ (such that $x_{\ell+n} = x_\ell$). Set $\omega_n = \exp \frac{2\pi \mathrm{i}}{n}$, the primitive root of unity of degree n. The *discrete Fourier transform (DFT)* of $\boldsymbol{x}$ is

$$\mathcal{F}_n : \Pi_n \to \Pi_n \qquad \text{such that} \qquad \boldsymbol{y} = \mathcal{F}_n \boldsymbol{x}, \quad \text{where} \quad y_j = \frac{1}{n} \sum_{\ell=0}^{n-1} \omega_n^{-j\ell} x_\ell, \quad j = 0...n-1.$$

Trivial exercise: You can easily prove that $\mathcal{F}_n$ is an isomorphism of Π_n onto itself and that

$$\boldsymbol{x} = \mathcal{F}_n^{-1} \boldsymbol{y}, \qquad \text{where} \qquad x_\ell = \sum_{j=0}^{n-1} \omega_n^{j\ell} y_j, \quad \ell = 0...n-1.$$

An important observation: Thus, multiplication by Q in (1.8) can be reduced to calculating an inverse of DFT.

Since we need to evaluate DFT (or its inverse) only in a single period, we can do so by multiplying a vector by a matrix, at the cost of $\mathcal{O}(n^2)$ operations. This, however, is suboptimal and the cost of calculation can be lowered a great deal!