Prof. A. C. Hansen

Mathematical Tripos Part II: Michaelmas Term 2024

Numerical Analysis – Lecture 1

1 The Poisson equation

Problem 1.1 (Approximation of ∇^2) Our goal is to solve the *Poisson equation*

$$\nabla^2 u = f \qquad (x,y) \in \Omega, \tag{1.1}$$

where $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ is the Laplace operator and Ω is an open connected domain of $\mathbb{R}^2$ with a Jordan boundary, specified together with the *Dirichlet boundary condition*

$$u(x,y) = \phi(x,y) \qquad (x,y) \in \partial\Omega. \tag{1.2}$$

(You may assume that $f \in C(\Omega)$, $\phi \in C^2(\partial\Omega)$, but this can be relaxed by an approach outside the scope of this course.) To this end we impose on Ω a square grid with uniform spacing of $h > 0$ and replace (1.1) by a *finite-difference* formula. For simplicity, we require for the time being that $\partial\Omega$ 'fits' into the grid: if a grid point lies inside Ω then all its neighbours are in $\mathrm{cl}\,\Omega$. We will discuss briefly in the sequel grids that fail this condition.

Remark 1.2 Finite differences are neither the only nor, arguably, the best means of solving partial differential equations. Other methods abound: finite elements, boundary elements, spectral and pseudospectral methods, finite-volume methods, vorticity methods, particle methods, meshless methods, gas-lattice methods and, in the important special case of the Poisson equation (1.1), fast multipole methods. Yet, finite differences are the simplest. The only additional ones that will feature in this lecture course are spectral methods in Chapter 3.

Since $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$, we need to consider a finite-difference approximation of second derivatives.

Proposition 1.3 *Let $g \in C^4[a,b]$ and $x \in (a+h, b-h)$. Then*

$$\Delta_h^2\, g(x) := g(x-h) - 2g(x) + g(x+h) = h^2 g''(x) + \tfrac{1}{12} h^4 g^{(4)} + \mathcal{O}(h^6). \tag{1.3}$$

Proof. Expanding into Taylor series,

$$\begin{aligned} g(x+h) - g(x) &= hg'(x) + \tfrac{1}{2!}h^2 g''(x) + \tfrac{1}{3!}h^3 g'''(x) + \cdots \\ g(x-h) - g(x) &= -hg'(x) + \tfrac{1}{2!}h^2 g''(x) - \tfrac{1}{3!}h^3 g'''(x) + \cdots \end{aligned}$$

and adding two expressions, we see that the terms with odd derivatives vanish, and the LHS of (1.3) is equal to $\sum_{k=1}^{m} \frac{2}{(2k)!} h^{2k} g^{(2k)}(x) + \mathcal{O}(h^{2m+2})$, where we took $m = 2$. □

Remark 1.4 In approximation of the second derivative g'' by the second central difference $\Delta_h^* g(x) = g(x-h) - 2g(x) + g(x+h)$, it is sometimes useful to know the terms of higher order:

$$\frac{1}{h^2}[g(x-h) - 2g(x) + g(x+h)] = g''(x) + \frac{1}{12}h^2 g^{(\mathrm{iv})} + \frac{1}{360}h^4 g^{(\mathrm{vi})} + \mathcal{O}(h^6).$$

Corollary 1.5 *The approximation*

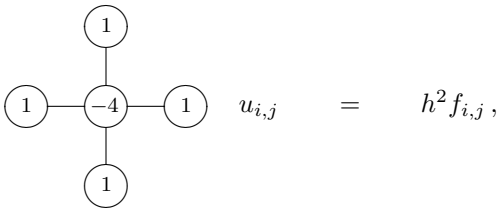
$$\begin{aligned} (\Delta_{h,x}^2 + \Delta_{h,y}^2)\, u(x,y) &= u(x-h,y) + u(x+h,y) + u(x,y-h) + u(x,y+h) - 4u(x,y) \\ &\approx h^2 \nabla^2 u(x,y) \end{aligned}$$

produces a local error of $\mathcal{O}(h^4)$.

Approximation 1.6 The aforementioned analysis justifies the *five-point method*

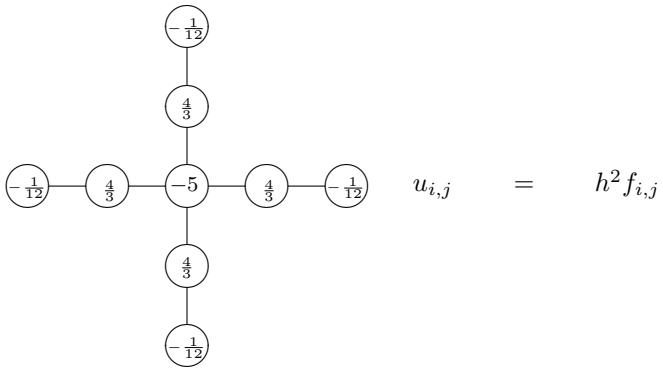
$$u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - 4u_{i,j} = h^2 f_{i,j}, \qquad (ih, jh) \in \Omega, \tag{1.4}$$

where $f_{i,j} = f(ih, jh)$ are given, and $u_{i,j} \approx u(ih, jh)$ is an approximation to the exact solution. It is usually denoted by the following *computational stencil*



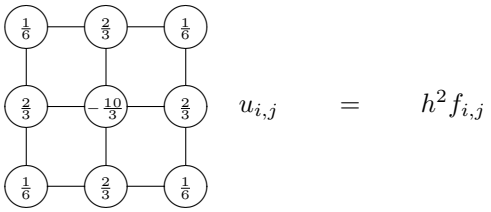
Whenever $(ih, jh) \in \partial\Omega$, we substitute appropriate Dirichlet boundary values. Note that the outcome of our procedure is a set of linear algebraic equations whose solution approximates the solution of the Poisson equation (1.1) at the grid points.

Approximation 1.7 It is easy (but laborious) to produce higher-order methods. You may verify, for example, that the stencil



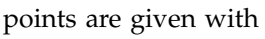
produces a local error of $\mathcal{O}(h^6)$. (This scheme is just a linear combination of two five-point methods with steps h and $2h$, respectively.) Needless to say, the implementation of this method is more complicated, since we might be 'missing' points near the boundary. Moreover, the set of algebraic equations that needs to be solved is less sparse than for the 5-point method, hence its solution is more expensive.

It is considerably easier to implement the *nine-point method*



but, as such, it again produces error of $\mathcal{O}(h^4)$. However, this can be remedied by a clever trick of adding the term $\frac{1}{12}h^4\nabla^2 f$ to the right-hand side, with the 5-point approximation to $h^2\nabla^2 f$, which increases the order to $\mathcal{O}(h^6)$ (see Exercise 1).

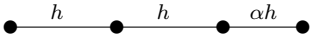
Problem 1.8 (Non-equispaced grids) Since the boundary often fails to fit exactly into a square grid, we sometimes need to approximate ∇^2 using non-equispaced points. Clearly, it is enough to be able to approximate a second directional derivative w.r.t. each variable and subsequently 'synthetize' an approximation to ∇^2.

For example, suppose that grid points are given with the spacing 

, where $0 < \alpha \leq 1$. It is easy to verify by a Taylor expansion that

$$\frac{2}{\alpha+1}g(x-h) - \frac{2}{\alpha}g(x) + \frac{2}{\alpha(\alpha+1)}g(x+\alpha h) = g''(x)h^2 + \frac{1}{3}(\alpha-1)g'''(x)h^3 + \mathcal{O}(h^4),$$

with error of $\mathcal{O}(h^3)$ (note that $\alpha = 1$ gives, as expected, $\mathcal{O}(h^4)$).

Better approximation can be obtained by taking two equispaced points on the 'interior' side, i.e. 

as follows:

$$\frac{\alpha-1}{\alpha+2}g(x-2h) - \frac{2(\alpha-2)}{\alpha+1}g(x-h) + \frac{\alpha-3}{\alpha}g(x) + \frac{6}{\alpha(\alpha+1)(\alpha+2)}g(x+\alpha h) = h^2 g''(x) + \mathcal{O}(h^4).$$

Prof. A. C. Hansen

Mathematical Tripos Part II: Michaelmas Term 2024

Numerical Analysis – Lecture 2

Finite-difference discretization of $\nabla^2 u = f$ replaces the PDE by a large system of linear equations. In the sequel we pay special attention to the *five-point formula,* which results in the approximation

$$h^2\nabla^2 u(x,y) \approx u(x-h,y) + u(x+h,y) + u(x,y-h) + u(x,y+h) - 4u(x,y)\,. \tag{1.5}$$

For the sake of simplicity, we restrict our attention to the important case of Ω being a *unit square*, where $h = \frac{1}{m+1}$ for some positive integer m. Thus, we estimate the m^2 unknown function values $u(ih,jh)_{i,j=1}^m$ (where $(ih,jh) \in \Omega$) by letting the right-hand side of (1.5) equal $h^2 f(ih,jh)$ at each value of i and j. This yields an $n \times n$ system of linear equations with $n = m^2$ unknowns $u_{i,j}$:

$$u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - 4u_{i,j} = h^2 f(ih,jh)\,. \tag{1.6}$$

(Note that when i or j is equal to 1 or m, then the values $u_{0,j}$, $u_{i,0}$ or $u_{i,m+1}$, $u_{m+1,j}$ are known boundary values and they should be moved to the right-hand side, thus leaving fewer unknowns on the left.) Having ordered grid points, we can write (1.6) as a linear system, say

$$A\boldsymbol{u} = \boldsymbol{b}\,.$$

Our present concern is to prove that, as $h \to 0$, the numerical solution (1.6) tends to the exact solution of the Poisson equation $\nabla^2 u = f$ (with appropriate Dirichlet boundary conditions).

Example 1.8 (Natural ordering) The way the matrix A of this system looks depends of course on the way how the grid points (ih,jh) are being assembled in the one-dimensional array. In the *natural ordering*, when the grid points are arranged by columns, A is the following block tridiagonal matrix:

$$A = \begin{bmatrix} B & I & & \\ I & B & I & \\ & \ddots & \ddots & \ddots \\ & & I & B & I \\ & & & I & B \end{bmatrix}, \qquad B = \begin{bmatrix} -4 & 1 & & \\ 1 & -4 & 1 & \\ & \ddots & \ddots & \ddots \\ & & 1 & -4 & 1 \\ & & & 1 & -4 \end{bmatrix}.$$

Before heading on let us prove the following simple but useful theorem whose importance will become apparent in the course of the lecture.

Theorem 1.9 (Gershgorin theorem) *All eigenvalues of an $n \times n$ matrix A are contained in the union of the Gershgorin discs in the complex plane:*

$$\sigma(A) \subset \cup_{i=1}^n \Gamma_i\,, \qquad \Gamma_i := \{z \in \mathbb{C} : |z - a_{ii}| \le r_i\}, \qquad r_i := \textstyle\sum_{j\neq i} |a_{ij}|\,.$$

Proof. For any matrix A, if $A\boldsymbol{x} = \lambda\boldsymbol{x}$ and $|x_i| = \max|x_j|$, then the ith equation of the relation $A\boldsymbol{x} = \lambda\boldsymbol{x}$ gives

$$|\lambda - a_{ii}| \cdot |x_i| = \Big|\sum_{j\neq i} a_{ij}x_j\Big| \le \sum_{j\neq i} |a_{ij}||x_j| \le |x_i| \sum_{j\neq i} |a_{ij}| =: |x_i|\, r_i\,,$$

and after dividing by $|x_i|$ we obtain $|\lambda - a_{ii}| \le r_i$. So, for any eigenvalue λ of A, the inequality $|\lambda - a_{ii}| \le r_i$ is valid for at least one value of i, hence the theorem. □

Lemma 1.10 *For any ordering of the grid points, the matrix A of the system (1.6) is symmetric and negative definite.*

Proof. Equation (1.6) implies that if $a_{ij} \neq 0$ for $i \neq j$, then the i-th and j-th points of the grid (ph, qh), are nearest neighbours. Hence $a_{ij} \neq 0$ implies $a_{ij} = a_{ji} = 1$, which proves the symmetry of A. Therefore A has real eigenvalues and eigenvectors.

It remains to prove that all the eigenvalues are negative. The arguments are parallel to the proof of Gershgorin theorem. Let $A\boldsymbol{x} = \lambda\boldsymbol{x}$, and let i be an integer such that $|x_i| = \max |x_j|$. With such an i we address the following identity (which is a reordering of the equation $(A\boldsymbol{x})_i = \lambda x_i$):

$$\underbrace{\big|(\lambda - a_{ii})\, x_i\big|}_{|\lambda+4|\,|x_i|} = \underbrace{\big|\textstyle\sum_{j\neq i}^{n} a_{ij} x_j\big|}_{\leq 4\,|x_i|} . \tag{1.7}$$

Here $a_{ii} = -4$ and $a_{ij} \in \{0, 1\}$ for $j \neq i$, with at most four nonzero elements on the right-hand side. It is seen that the case $\lambda > 0$ is impossible. Assuming $\lambda = 0$, we obtain $|x_j| = |x_i|$ whenever $a_{ij} = 1$, so we can alter the value of i in (1.7) to any of such j and repeat the same arguments. Thus, the modulus of every component of $\boldsymbol{x}$ would be $|x_i|$, but then the equations (1.7) that occur at the boundary of the grid and have fewer than four off-diagonal terms (see (1.6)) could not be true. Hence, $\lambda = 0$ is impossible too, hence $\lambda < 0$ which proves that A is negative definite. □

Proposition 1.11 *The eigenvalues of the matrix A are*

$$\lambda_{k,\ell} = -4\Big(\sin^2\frac{k\pi h}{2} + \sin^2\frac{\ell\pi h}{2}\Big), \qquad h = \frac{1}{m+1}\,, \qquad k, \ell = 1...m.$$

Proof. Let us show that, for every pair (k, ℓ), the vectors

$$v = (v_{i,j}), \quad v_{i,j} = \sin ix \sin jy, \quad \text{where} \quad x = k\pi h, \quad y = \ell\pi h,$$

are the eigenvectors of A. Indeed, for $i, j = 1...m$, we have

$$\begin{aligned}(Av)_{i,j} &= \sin(jy)\,\big[\sin(ix - x) - 2\sin(ix) + \sin(ix + x)\big] \\ &\quad + \sin(ix)\,\big[\sin(jy - y) - 2\sin(jy) + \sin(jy + y)\big] \\ &= \sin(jy)\sin(ix)\big[2\cos x - 2\big] + \sin(ix)\sin(jy)\big[2\cos y - 2\big] \;=\; \lambda v_{i,j}\,.\end{aligned}$$

Note that the terms $u_{i\pm1,j}, u_{i,j\pm1}$ do not appear in (1.6) for $i, j = 1$ or $i, j = m$, respectively, therefore (for such i, j) we should have dropped the corresponding components from above equation, but they are equal to zero because $\sin(i - 1)x = 0$ for $i = 1$, while $\sin(i + 1)x = 0$ for $i = m$, since $x = \frac{k\pi}{m+1}$. Thus, the eigenvalues are

$$\lambda_{k,\ell} = \big[2\cos x - 2\big] + \big[2\cos y - 2\big] = -4\Big(\sin^2\frac{x}{2} + \sin^2\frac{y}{2}\Big) = -4\Big(\sin^2\frac{k\pi h}{2} + \sin^2\frac{\ell\pi h}{2}\Big). \quad \square$$

Remark 1.12 As a matter of independent mathematical interest, note that for $1 \leq k, \ell \ll m$ we have $\sin x \approx x$, hence the eigenvalues for the discretized Laplacian ∇_h^2 are

$$\frac{\lambda_{k,\ell}}{h^2} \approx -\frac{4}{h^2}\left[\frac{k^2\pi^2h^2}{4} + \frac{\ell^2\pi^2h^2}{4}\right] = -(k^2 + \ell^2)\pi^2\,.$$

Now, recall (e.g. from the solution of the Poisson equation in a square by separation of variables in Maths Methods) that the *exact* eigenvalues of ∇^2 (in the unit square) are $-(k^2 + \ell^2)\pi^2$, $k, \ell \in \mathbb{N}$, with the corresponding eigenfunctions $V_{k,\ell}(x, y) = \sin k\pi x \sin \ell\pi y$. So, the eigenvectors of the discretized ∇_h^2 are the values of $V_{k,\ell}(x, y)$ on the grid-points, and the eigenvalues of ∇_h^2 approximate those for continuous case.

Prof. A. C. Hansen

Mathematical Tripos Part II: Michaelmas Term 2024

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 $\boldsymbol{e} = (e_{i,j}) \in \mathbb{R}^n$ where $n = m^2$, and for $\boldsymbol{x} \in \mathbb{R}^n$ let $\|\boldsymbol{x}\| = \|\boldsymbol{x}\|_{\ell_2}$ be the Eucledian 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 *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, for the exact solution, we have

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

Subtracting this from numerical approximation (1.6), 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 $|\eta_{i,j}|^2 < c^2h^8$, where $h = \frac{1}{m+1}$, and there are m^2 components, we have

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

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,

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

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 $\boldsymbol{c}_1, \dots, \boldsymbol{c}_m$, the portion $\widehat{\boldsymbol{c}}_2$ out of the second components and so on. This results in new system

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

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

(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

$$(Q\boldsymbol{y})_\ell = \sum_{j=1}^{m} \sin \frac{\pi j\ell}{m+1} y_j = \operatorname{Im} \sum_{j=0}^{m} \exp \frac{\mathrm{i}\pi j\ell}{m+1} y_j = \operatorname{Im} \sum_{j=0}^{2m+1} \exp \frac{2\mathrm{i}\pi j\ell}{2m+2} y_j, \quad \ell = 1...m, \tag{1.7}$$

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

Definition 1.15 (The discrete Fourier transform (DFT)) 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 (1.7) 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!

Prof. A. C. Hansen

Mathematical Tripos Part II: Michaelmas Term 2024

Numerical Analysis – Lecture 4

Algorithm 1.15 (The fast Fourier transform (FFT)) We assume that n is a power of 2, i.e. $n = 2m = 2^p$, and for $\boldsymbol{y} \in \Pi_{2m}$, denote by

$$\boldsymbol{y}^{(\mathrm{E})} = \{y_{2j}\}_{j\in\mathbb{Z}} \qquad \text{and} \qquad \boldsymbol{y}^{(\mathrm{O})} = \{y_{2j+1}\}_{j\in\mathbb{Z}}$$

the even and odd portions of $\boldsymbol{y}$, respectively. Note that $\boldsymbol{y}^{(\mathrm{E})}, \boldsymbol{y}^{(\mathrm{O})} \in \Pi_m$.

Suppose that we already know the inverse DFT of both 'short' sequences,

$$\boldsymbol{x}^{(\mathrm{E})} = \mathcal{F}_m^{-1}\boldsymbol{y}^{(\mathrm{E})}, \qquad \boldsymbol{x}^{(\mathrm{O})} = \mathcal{F}_m^{-1}\boldsymbol{y}^{(\mathrm{O})}.$$

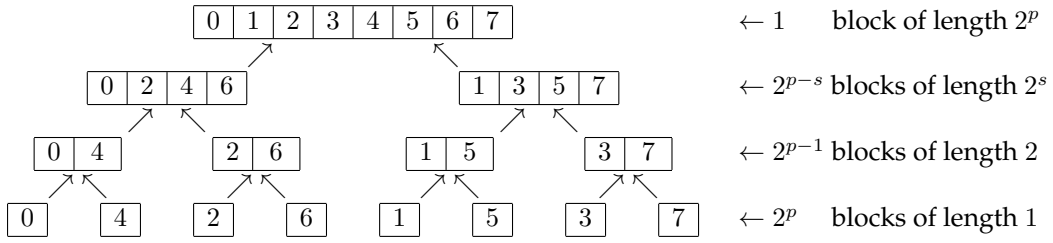
It is then possible to assemble $\boldsymbol{x} = \mathcal{F}_{2m}^{-1}\boldsymbol{y}$ in a small number of operations. Since $\omega_{2m}^{2m} = 1$, we obtain $\omega_{2m}^2 = \omega_m$, and

$$\begin{aligned} x_\ell = \sum_{j=0}^{2m-1} \omega_{2m}^{j\ell} y_j &= \sum_{j=0}^{m-1} \omega_{2m}^{2j\ell} y_{2j} + \sum_{j=0}^{m-1} \omega_{2m}^{(2j+1)\ell} y_{2j+1} \\ &= \sum_{j=0}^{m-1} \omega_m^{j\ell} y_j^{(\mathrm{E})} + \omega_{2m}^{\ell} \sum_{j=0}^{m-1} \omega_m^{j\ell} y_j^{(\mathrm{O})} = x_\ell^{(\mathrm{E})} + \omega_{2m}^{\ell} x_\ell^{(\mathrm{O})}, \qquad \ell = 0,\dots,m-1. \end{aligned}$$

Therefore, it costs just m products to evaluate the first half of $\boldsymbol{x}$, provided that $\boldsymbol{x}^{(\mathrm{E})}$ and $\boldsymbol{x}^{(\mathrm{O})}$ are known. It actually costs nothing to evaluate the second half, since

$$\omega_m^{j(m+\ell)} = \omega_m^{j\ell}, \qquad \omega_{2m}^{m+\ell} = -\omega_{2m}^{\ell} \qquad \Rightarrow \qquad x_{m+\ell} = x_\ell^{(\mathrm{E})} - \omega_{2m}^{\ell} x_\ell^{(\mathrm{O})}, \qquad \ell = 0,\dots,m-1.$$

To execute FFT, we start from vectors of unit length and in each s-th stage, $s = 1...p$, assemble 2^{p-s} vectors of length 2^s from vectors of length 2^{s-1}: this costs $2^{p-s}2^{s-1} = 2^{p-1}$ products. Altogether, the cost of FFT is $p2^{p-1} = \frac{1}{2}n\log_2 n$ products.



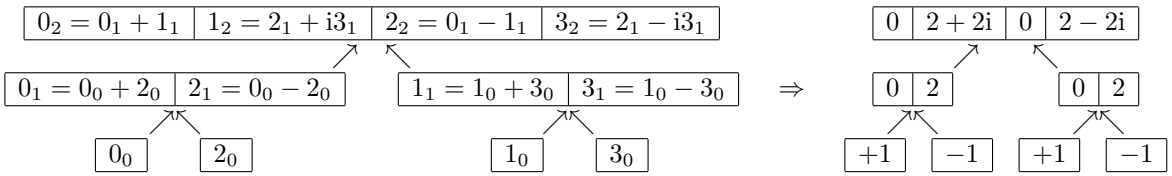
For $n = 1024 = 2^{10}$, say, the cost is $\approx 5 \times 10^3$ products, compared to $\approx 10^6$ for naive matrix multiplication! For $n = 2^{20}$ the respective numbers are $\approx 1.05 \times 10^7$ and $\approx 1.1 \times 10^{12}$, which represents a saving by a factor of more than 10^5.

Matlab demo: Check out the online animation for computing the FFT at

`http://www.damtp.cam.ac.uk/user/hf323/M21-II-NA/demos/fft_gui/fft_gui.html`

and download the Matlab GUI from there to follow the computation of each single FFT term.

Example 1.16 Computation of FFT for $n = 4$ in general, and for the vector $\boldsymbol{y} = (1, 1, -1, -1)$ in particular.



2 Partial differential equations of evolution

Method 2.1 We consider the solution of the *diffusion equation*

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}\,, \qquad 0 \le x \le 1, \quad t \ge 0,$$

with *initial conditions* $u(x,0) = u_0(x)$ for $t = 0$ and Dirichlet *boundary conditions* $u(0,t) = \phi_0(t)$ at $x = 0$ and $u(1,t) = \phi_1(t)$ at $x = 1$. By Taylor's expansion

$$\begin{aligned} \tfrac{\partial u(x,t)}{\partial t} &= \tfrac{1}{k}\big[u(x,t+k) - u(x,t)\big] + \mathcal{O}(k), && k = \Delta t\,, \\ \tfrac{\partial^2 u(x,t)}{\partial x^2} &= \tfrac{1}{h^2}\big[u(x-h,t) - 2u(x,t) + u(x+h,t)\big] + \mathcal{O}(h^2), && h = \Delta x\,, \end{aligned}$$

so that, for the true solution, we obtain

$$u(x,t+k) = u(x,t) + \tfrac{k}{h^2}\big[u(x-h,t) - 2u(x,t) + u(x+h,t)\big] + \mathcal{O}(k^2 + kh^2)\,. \tag{2.1}$$

That motivates the numerical scheme for approximation $u_m^n \approx u(x_m, t_n)$ on the rectangular mesh $(x_m, t_n) = (mh, nk)$:

$$u_m^{n+1} = u_m^n + \mu\left(u_{m-1}^n - 2u_m^n + u_{m+1}^n\right), \qquad m = 1...M\,. \tag{2.2}$$

Here $h = \frac{1}{M+1}$ and $\mu = \frac{k}{h^2} = \frac{\Delta t}{(\Delta x)^2}$ is the so-called *Courant number.* With μ being fixed, we have $k = \mu h^2$, so that the local truncation error of the scheme is $\mathcal{O}(h^4)$. Substituting whenever necessary initial conditions u_m^0 and boundary conditions u_0^n and u_{M+1}^n, we possess enough information to advance in (2.2) from $\boldsymbol{u}^n := [u_1^n, \ldots, u_M^n]$ to $\boldsymbol{u}^{n+1} := [u_1^{n+1}, \ldots, u_M^{n+1}]$.

Similarly to ODEs or Poisson equation, we say that the method is *convergent* if, for a fixed μ, and for every $T > 0$, we have

$$\lim_{h\to 0} |u_m^n - u(x_m, t_n)| = 0 \quad \text{uniformly for} \quad (x_m, t_n) \in [0,1]\times[0,T]\,.$$

In the present case, however, a method has an extra parameter μ, and it is entirely possible for a method to converge for some choice of μ and diverge otherwise.

Theorem 2.2 *If $\mu \le \frac{1}{2}$, then method (2.2) converges.*

Proof. Let $e_m^n := u_m^n - u(mh, nk)$ be the error of approximation, and let $\boldsymbol{e}^n = [e_1^n, \ldots, e_M^n]$ with $\|\boldsymbol{e}^n\| := \max_m |e_m^n|$. Convergence is equivalent to

$$\lim_{h\to 0} \max_{1\le n\le T/k} \|\boldsymbol{e}^n\| = 0$$

for every constant $T > 0$. Subtracting (2.1) from (2.2), we obtain

$$\begin{aligned} e_m^{n+1} &= e_m^n + \mu(e_{m-1}^n - 2e_m^n + e_{m+1}^n) + \mathcal{O}(h^4) \\ &= \mu e_{m-1}^n + (1-2\mu)e_m^n + \mu e_{m+1}^n + \mathcal{O}(h^4). \end{aligned}$$

Then

$$\|\boldsymbol{e}^{n+1}\| = \max_m |e_m^{n+1}| \le (2\mu + |1-2\mu|)\,\|\boldsymbol{e}^n\| + ch^4 = \|e^n\| + ch^4,$$

by virtue of $\mu \le \frac{1}{2}$. Since $\|\boldsymbol{e}^0\| = 0$, induction yields

$$\|\boldsymbol{e}^n\| \le cnh^4 \le \tfrac{cT}{k}\,h^4 = \tfrac{cT}{\mu}\,h^2 \to 0 \qquad (h \to 0)$$ □

Discussion 2.3 In practice we wish to choose h and k of comparable size, therefore $\mu = k/h^2$ is likely to be large. Consequently, the restriction of the last theorem is disappointing: unless we are willing to advance with tiny time step k, the method (2.2) is of limited practical interest. The situation is similar to stiff ODEs: like the Euler method, the scheme (2.2) is simple, plausible, explicit, easy to execute and analyse – but of very limited utility. . . .

Matlab demo: Download the Matlab GUI for *Stability of 1D PDEs* from `http://www.damtp.cam.ac.uk/user/hf323/M21-II-NA/demos/pde_stability/pde_stability.html` and solve the diffusion equation in the interval $[0,1]$ with method (2.2) and $\mu = 0.51 > \frac{1}{2}$. Using (as preset) 100 grid points to discretise $[0,1]$ will then require the time steps to be $5.1 \cdot 10^{-5}$. The solution will evolve very slowly, but wait long enough to see what happens!

Mathematical Tripos Part II: Michaelmas Term 2024

Numerical Analysis – Lecture 5

2 Partial differential equations of evolution

Method 2.1 We consider the solution of the *diffusion equation*

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}\,, \qquad 0 \le x \le 1, \quad t \ge 0,$$

with *initial conditions* $u(x,0) = u_0(x)$ for $t = 0$ and Dirichlet *boundary conditions* $u(0,t) = \phi_0(t)$ at $x = 0$ and $u(1,t) = \phi_1(t)$ at $x = 1$. By Taylor's expansion

$$\begin{aligned} \tfrac{\partial u(x,t)}{\partial t} &= \tfrac{1}{k}\big[u(x,t+k) - u(x,t)\big] + \mathcal{O}(k), && k = \Delta t\,, \\ \tfrac{\partial^2 u(x,t)}{\partial x^2} &= \tfrac{1}{h^2}\big[u(x-h,t) - 2u(x,t) + u(x+h,t)\big] + \mathcal{O}(h^2), && h = \Delta x\,, \end{aligned}$$

so that, for the true solution, we obtain

$$u(x,t+k) = u(x,t) + \tfrac{k}{h^2}\big[u(x-h,t) - 2u(x,t) + u(x+h,t)\big] + \mathcal{O}(k^2 + kh^2)\,. \tag{2.1}$$

That motivates the numerical scheme for approximation $u_m^n \approx u(x_m, t_n)$ on the rectangular mesh $(x_m, t_n) = (mh, nk)$:

$$u_m^{n+1} = u_m^n + \mu\left(u_{m-1}^n - 2u_m^n + u_{m+1}^n\right), \qquad m = 1...M\,. \tag{2.2}$$

Here $h = \frac{1}{M+1}$ and $\mu = \frac{k}{h^2} = \frac{\Delta t}{(\Delta x)^2}$ is the so-called *Courant number*. With μ being fixed, we have $k = \mu h^2$, so that the local truncation error of the scheme is $\mathcal{O}(h^4)$. Substituting whenever necessary initial conditions u_m^0 and boundary conditions u_0^n and u_{M+1}^n, we possess enough information to advance in (2.2) from $\boldsymbol{u}^n := [u_1^n, \ldots, u_M^n]$ to $\boldsymbol{u}^{n+1} := [u_1^{n+1}, \ldots, u_M^{n+1}]$.

Similarly to ODEs or Poisson equation, we say that the method is *convergent* if, for a fixed μ, and for every $T > 0$, we have

$$\lim_{h\to 0} |u_m^n - u(x_m, t_n)| = 0 \quad \text{uniformly for} \quad (x_m, t_n) \in [0,1]\times[0,T]\,.$$

In the present case, however, a method has an extra parameter μ, and it is entirely possible for a method to converge for some choice of μ and diverge otherwise.

Stability, consistency and the Lax equivalence theorem Suppose that a numerical method for a partial differential equation of evolution can be written in the form[1]

$$\boldsymbol{u}^{n+1} = A_h \boldsymbol{u}^n,$$

where $\boldsymbol{u}^n \in \mathbb{R}^M$, $A_h \in \mathbb{R}^{M\times M}$ is a matrix, and $h = \frac{1}{M+1}$. Fix a norm $\|\cdot\|$ on $\mathbb{R}^M$, and let $\|A_h\| = \sup \frac{\|A_h \boldsymbol{x}\|}{\|\boldsymbol{x}\|}$ be the corresponding induced matrix norm. If we define *stability* as preserving the boundedness of $\boldsymbol{u}^n$ with respect to the norm $\|\cdot\|$, then since

$$\|\boldsymbol{u}^n\| \le \|A_h^n \boldsymbol{u}^0\| \le \|A_h\|^n \|\boldsymbol{u}^0\|,$$

we get:

$$\|A_h\| \le 1 \text{ as } h \to 0 \quad \Rightarrow \quad \text{the method is stable.}$$

If we denote the exact solution of the PDE by $\widehat{u}(x,t)$ and let $\widehat{\boldsymbol{u}}^n = (\widehat{u}(mk, nt))_{1\le m\le M}$, then we have $\widehat{\boldsymbol{u}}^{n+1} = A_h \widehat{\boldsymbol{u}}^n + \boldsymbol{\eta}^n$ where $\boldsymbol{\eta}^n$ is the local truncation error. The error vector $\boldsymbol{e}^n = \widehat{\boldsymbol{u}}^n - \boldsymbol{u}^n$ satisfies

$$\boldsymbol{e}^{n+1} = A_h \boldsymbol{e}^n + \boldsymbol{\eta}^n.$$

[1] Assuming zero boundary conditions

Using $\|A_h\| \leq 1$ and assuming $\|\boldsymbol{e}^0\| = 0$, we get $\|\boldsymbol{e}^n\| \leq \|\boldsymbol{\eta}^{n-1}\| + \cdots + \|\boldsymbol{\eta}^0\|$. If *consistency* holds, i.e., $\|\boldsymbol{\eta}^n\| = O(k^2)$, then we see that $\|\boldsymbol{e}^n\| \leq nck^2$ for some constant $c > 0$. Since $n \leq T/k$ we end up with $\|\boldsymbol{e}^n\| \leq cTk$, and so $\|\boldsymbol{e}^n\| \to 0$ as $k \to 0$ uniformly in $n \in [1, T/k]$. This shows convergence.

We have thus arrived at the *Lax equivalence theorem*:

Theorem 2.2 *"consistency + stability = convergence"*

(more precisely what we have proved here is the implication $\implies$).

Norms The discussion above involves a choice of norm on $\mathbb{R}^M$. There are two standard choices of norms:

- *Sup-norm.* Here, we choose
$$\|\boldsymbol{u}\| = \|\boldsymbol{u}\|_\infty = \max_{i=1,\ldots,M} |u_i|.$$
It can be easily shown that the corresponding induced norm for a matrix $A \in \mathbb{R}^{M\times M}$ is given by:
$$\|A\|_{\infty\to\infty} := \sup_{\boldsymbol{x}} \frac{\|A\boldsymbol{x}\|_\infty}{\|\boldsymbol{x}\|_\infty} = \max_{i=1,\ldots,M} \sum_{j=1}^{M} |A_{ij}|.$$
This the choice of norm we implicitly used in the convergence proof of Theorem 2.1 (Lecture 4). The matrix in this case was
$$A_h = \begin{bmatrix} 1-2\mu & \mu & & \\ \mu & \ddots & \ddots & \\ & \ddots & \ddots & \mu \\ & & \mu & 1-2\mu \end{bmatrix},$$
for which we get $\|A_h\|_{\infty\to\infty} = |1-2\mu| + 2\mu \leq 1$ if $\mu \leq 1/2$.
- *Normalized Euclidean norm.* Another common of choice of norm is the normalized Euclidean length, namely,
$$\|\boldsymbol{u}\| := \sqrt{\tfrac{1}{M}\textstyle\sum_{i=1}^{M} |u_i|^2}.$$
The reason for the factor $\frac{1}{M}$ is to ensure that, because of the convergence of Riemann sums, we obtain
$$\|\boldsymbol{u}\| := \left[\tfrac{1}{M}\textstyle\sum_{i=1}^{M} |u_i|^2\right]^{1/2} \to \left[\int_0^1 |u(x)|^2 \mathrm{d}x\right]^{1/2} =: \|u\|_{L_2} \qquad (h = 1/(M+1) \to 0),$$
The induced matrix norm in this case is the *spectral norm* (or the *operator norm*) and is denoted $\|A\|_2$:[2]
$$\|A\|_2 := \sup_{\boldsymbol{x}} \frac{\|A\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2}.$$
The spectral norm of A is equal to the largest singular value of A. Equivalently, we can write $\|A\|_2 = [\rho(AA^T)]^{1/2}$ where ρ is the spectral radius:
$$\rho(M) := \max\{|\lambda| : \lambda \text{ eigenvalue of } M\}.$$
For certain matrices, such as normal matrices, one can show that $\|A\|_2 = \rho(A)$.

Problem 2.3 (Stability of (2.2)) Although we can deduce from the theorem that $\mu \leq \frac{1}{2}$ implies stability, we will prove directly that stability $\Leftrightarrow \mu \leq \frac{1}{2}$. Let $\boldsymbol{u}^n = [u_1^n, \ldots, u_M^n]^T$. We can express the recurrence (2.2)
$$u_m^{n+1} = u_m^n + \mu\left(u_{m-1}^n - 2u_m^n + u_{m+1}^n\right), \qquad m = 1...M,$$

[2]Note that if $\|\cdot\|$ is the normalized Euclidean norm, then $\|A\boldsymbol{x}\|/\|\boldsymbol{x}\| = \|A\boldsymbol{x}\|_2/\|\boldsymbol{x}\|_2$ where $\|x\|_2 = (\sum_i |x_i|^2)^{1/2}$ is the usual (unnormalized) Euclidean norm

in the matrix form

$$\boldsymbol{u}_h^{n+1} = A_h \boldsymbol{u}_h^n, \qquad A_h = I + \mu A_*, \qquad A_* = \begin{bmatrix} -2 & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & -2 \end{bmatrix}_{M\times M}.$$

Here A_* is TST, with $\lambda_\ell(A_*) = -4\sin^2\frac{\pi\ell h}{2}$, hence $\lambda_\ell(A_h) = 1 - 4\mu\sin^2\frac{\pi\ell h}{2}$, so that its spectrum lies within the interval $[\lambda_M, \lambda_1] = [1 - 4\mu\cos^2\frac{\pi h}{2}, 1 - 4\mu\sin^2\frac{\pi h}{2}]$. Since A_h is symmetric, we have

$$\|A_h\|_2 = \rho(A_h) = \begin{cases} |1 - 4\mu\sin^2\frac{\pi h}{2}| \le 1\,, & \mu \le \frac{1}{2}, \\ |1 - 4\mu\cos^2\frac{\pi h}{2}| > 1, & \mu > \frac{1}{2} \quad (h \le h_\mu)\,. \end{cases}$$

We distinguish between two cases.

1) $\mu \le \frac{1}{2}$: $\|\boldsymbol{u}^n\| \le \|A\| \cdot \|\boldsymbol{u}^{n-1}\| \le \cdots \le \|A\|^n \|\boldsymbol{u}^0\| \le \|\boldsymbol{u}^0\|$ as $n \to \infty$, for every $\boldsymbol{u}^0$.

2) $\mu > \frac{1}{2}$: Choose $\boldsymbol{u}^0$ as the eigenvector corresponding to the largest (in modulus) eigenvalue, $|\lambda| > 1$. Then $\boldsymbol{u}^n = \lambda^n \boldsymbol{u}^0$, becoming unbounded as $n \to \infty$.

Technique 2.4 (Semidiscretization) Let $u_m(t) = u(mh, t)$, $m = 1...M$, $t \ge 0$. Approximating $\partial^2/\partial x^2$ as before, we deduce from the PDE that the *semidiscretization*

$$\frac{du_m}{dt} = \frac{1}{h^2}(u_{m-1} - 2u_m + u_{m+1}), \qquad m = 1...M \tag{2.3}$$

carries an error of $\mathcal{O}(h^2)$. This is an ODE system, and we can solve it by any ODE solver. Thus, Euler's method yields (2.2), while backward Euler results in

$$u_m^{n+1} - \mu(u_{m-1}^{n+1} - 2u_m^{n+1} + u_{m+1}^{n+1}) = u_m^n.$$

This approach is commonly known as *the method of lines*. Much (although not all!) of the theory of finite-difference methods for PDEs of evolution can be presented as a two-stage task: first semidiscretize, getting rid of space variables, then use an ODE solver. Typically, each stage is conceptually easier than the process of discretizing in unison in both time and in space (so-called *full discretization*).

Method 2.5 (The Crank–Nicolson scheme) Discretizing the ODE (2.3) with the trapezoidal rule, we obtain

$$u_m^{n+1} - \tfrac{1}{2}\mu(u_{m-1}^{n+1} - 2u_m^{n+1} + u_{m+1}^{n+1}) = u_m^n + \tfrac{1}{2}\mu(u_{m-1}^n - 2u_m^n + u_{m+1}^n), \qquad m = 1...M. \tag{2.4}$$

Thus, each step requires the solution of an $M \times M$ TST system. The error of the scheme is $\mathcal{O}(k^3 + kh^2)$, so basically the same as with Euler's method. However, as we will see, Crank–Nicolson enjoys superior stability features, as compared with the method (2.2).

Note further that (2.4) is an *implicit* method: advancing each time step requires to solve a linear algebraic system. However, the matrix of the system is TST and its solution by sparse Cholesky factorization can be done in $\mathcal{O}(M)$ operations.

Mathematical Tripos Part II: Michaelmas Term 2024

Numerical Analysis – Lecture 6

Definition 2.4 (Normal matrices) We say that a matrix A is *normal* if $A = QD\bar{Q}^T$, where D is a (complex) diagonal matrix and Q is a unitary matrix (such that $Q\bar{Q}^T = I$, where the bar in $\bar{Q}$ means complex conjugation). In other words, a matrix is normal if it has a complete set of orthonormal eigenvectors.

Examples of the real normal matrices, besides the familiar symmetric matrices ($A = A^T$), inlclude also the matrices which are skew-symmetric ($A = -A^T$), and more generally the matrices with skew-symmetric off-diagonal part.

Proposition 2.5 *If A is normal, then $\|A\| = \rho(A)$.*

Proof. Let $\boldsymbol{u}$ be any vector (complex-valued as well). We can expand it in the basis of the orthonormal eigenvectors $\boldsymbol{u} = \sum_{i=1}^n a_i \boldsymbol{q}_i$. Then $A\boldsymbol{u} = \sum_{i=1}^n \lambda_i a_i \boldsymbol{q}_i$, and since $\boldsymbol{q}_i$ are orthonormal, we obtain

$$\|A\|_2 := \sup_{\boldsymbol{u}} \frac{\|A\boldsymbol{u}\|_2}{\|\boldsymbol{u}\|_2} = \sup_{a_i} \frac{\{\sum_{i=1}^n |\lambda_i a_i|^2\}^{1/2}}{\{\sum_{i=1}^n |a_i|^2\}^{1/2}} = |\lambda_{\max}| \,.$$

Remark 2.6 More generally, one can prove that, for any matrix A, we have $\|A\|_2 = [\rho(A\bar{A}^T)]^{1/2}$, and the previous result for normal matrices can be deduced from that formula.

Example 2.7 (Crank–Nicolson method for diffusion equation) Let

$$u_m^{n+1} - \tfrac{1}{2}\mu(u_{m-1}^{n+1} - 2u_m^{n+1} + u_{m+1}^{n+1}) = u_m^n + \tfrac{1}{2}\mu(u_{m-1}^n - 2u_m^n + u_{m+1}^n), \qquad m = 1...M.$$

Then $B\boldsymbol{u}^{n+1} = C\boldsymbol{u}^n$, where the matrices B and C are Toeplitz symmetric tridiagonal (TST),

$$\boldsymbol{u}^{n+1} = B^{-1}C\boldsymbol{u}^n, \qquad \begin{matrix} B = I - \frac{1}{2}\mu A_* \,, \\ C = I + \frac{1}{2}\mu A_* \,, \end{matrix} \qquad A_* = \begin{bmatrix} -2 & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & -2 \end{bmatrix}_{M\times M} .$$

All $M \times M$ TST matrices share the same eigenvectors, hence so does $B^{-1}C$. Moreover, these eigenvectors are orthogonal. Therefore, also $A = B^{-1}C$ is normal and its eigenvalues are

$$\lambda_k(A) = \frac{\lambda_k(C)}{\lambda_k(B)} = \frac{1 - 2\mu\sin^2\frac{1}{2}\pi kh}{1 + 2\mu\sin^2\frac{1}{2}\pi kh} \quad \Rightarrow \quad |\lambda_k(A)| \le 1, \qquad k = 1...M.$$

Consequently Crank–Nicolson is stable for all $\mu > 0$.
Matlab demo: Download the Matlab GUI for *Stability of 1D PDEs* from `http://www.damtp.cam.ac.uk/user/hf323/M21-II-NA/demos/pde_stability/pde_stability.html` and solve the diffusion equation in the interval $[0, 1]$ with the Euler method and with Crank–Nicolson. See the effect of unconditional stability!

Example 2.8 (Convergence of the Crank-Nicolson method for diffusion equation) It is not difficult to verify that the local error of the Crank-Nicolson scheme is $\eta_m^n = \mathcal{O}(k^3 + kh^2)$, where $\mathcal{O}(k^3)$ is inherited from the trapezoidal rule (compared to $\mathcal{O}(k^2)$ for the Euler method). We also have

$$\|\boldsymbol{\eta}^n\| = \{h\textstyle\sum_{m=1}^M |\eta_m^n|^2\}^{1/2} = \mathcal{O}(k^3 + kh^2) \,.$$

Hence, for the error vectors $\boldsymbol{e}^n$ we have

$$B\boldsymbol{e}^{n+1} = C\boldsymbol{e}^n + \boldsymbol{\eta}^n \quad \Rightarrow \quad \|\boldsymbol{e}^{n+1}\| \le \|B^{-1}C\| \cdot \|\boldsymbol{e}^n\| + \|B^{-1}\| \cdot \|\boldsymbol{\eta}^n\| \,.$$

We have just proved that $\|B^{-1}C\| \le 1$, and we also have $\|B^{-1}\| \le 1$, because all the eigenvalues of B are greater than 1 (by Gershgorin's theorem). Therefore, $\|\boldsymbol{e}^{n+1}\| \le \|\boldsymbol{e}^n\| + \|\boldsymbol{\eta}^n\|$, and

$$\|\boldsymbol{e}^n\| \le \|\boldsymbol{e}^0\| + n\|\boldsymbol{\eta}\| = n\|\boldsymbol{\eta}\| \le \tfrac{cT}{k}(k^3 + kh^2) = cT(k^2 + h^2).$$

Thus, taking $k = \alpha h$ will result in $\mathcal{O}(h^2)$ error of approximation.

We consider the solution of the *advection equation*

$$\frac{\partial u}{\partial t} = \frac{\partial u}{\partial x}\,, \qquad 0 \le x \le 1, \quad t \ge 0,$$

with *initial conditions* $u(x,0) = u_0(x)$ for $t = 0$ and Dirichlet *boundary conditions* $u(0,t) = \phi_0(t)$ at $x = 0$ and $u(1,t) = \phi_1(t)$ at $x = 1$.

Example 2.9 (Crank–Nicolson for advection equation) Let

$$u_m^{n+1} - u_m^n = \tfrac{1}{4}\mu(u_{m+1}^{n+1} - u_{m-1}^{n+1}) + \tfrac{1}{4}\mu(u_{m+1}^n - u_{m-1}^n), \qquad m = 1...M\,.$$

(This is the trapezoidal rule applied to the semidiscretization of advection equation $\frac{\partial u}{\partial t} = \frac{\partial u}{\partial x}$). In this case, $\boldsymbol{u}^{n+1} = B^{-1}C\boldsymbol{u}^n$, where the matrices B and C are Toeplitz antisymmetric tridiagonal,

$$B = \begin{bmatrix} 1 & -\frac{1}{4}\mu & & \\ \frac{1}{4}\mu & 1 & \ddots & \\ & \ddots & \ddots & -\frac{1}{4}\mu \\ & & \frac{1}{4}\mu & 1 \end{bmatrix}, \qquad C = \begin{bmatrix} 1 & \frac{1}{4}\mu & & \\ -\frac{1}{4}\mu & 1 & \ddots & \\ & \ddots & \ddots & \frac{1}{4}\mu \\ & & -\frac{1}{4}\mu & 1 \end{bmatrix}.$$

Similarly to Exercise 4, the eigenvalues and eigenvectors of the matrix

$$S = \begin{bmatrix} \alpha & \beta & & \\ -\beta & \alpha & \ddots & \\ & \ddots & \ddots & \beta \\ & & -\beta & \alpha \end{bmatrix},$$

are given by $\lambda_k = \alpha + 2\,\mathrm{i}\,\beta\cos kx$, and $\boldsymbol{w}_k = (\mathrm{i}^m \sin kmx)_{m=1}^M$, where $x = \pi h = \frac{\pi}{M+1}$. So, all such S are normal and share the same eigenvectors, hence so does $A = B^{-1}C$, hence A is normal and

$$\lambda_k(A) = \frac{\lambda_k(C)}{\lambda_k(B)} = \frac{1 + \frac{1}{2}\,\mathrm{i}\,\mu\cos kx}{1 - \frac{1}{2}\,\mathrm{i}\,\mu\cos kx} \quad \Rightarrow \quad |\lambda_k(A)| = 1, \qquad k = 1...M.$$

So, Crank–Nicolson is again stable for all $\mu > 0$.

Example 2.10 (Euler for advection equation) Finally, consider the Euler method for advection equation

$$u_m^{n+1} - u_m^n = \mu(u_{m+1}^n - u_m^n), \qquad m = 1...M\,.$$

We have $\boldsymbol{u}^{n+1} = A\boldsymbol{u}^n$, where

$$A = \begin{bmatrix} 1-\mu & \mu & & \\ & 1-\mu & \ddots & \\ & & \ddots & \mu \\ & & & 1-\mu \end{bmatrix},$$

but A is *not* normal, and although its eigenvalues are bounded by 1 for $\mu \le 2$ (note $1 - \mu$ is the only eigenvalue of A), it is the matrix induced norm of A that matters. For this example, it is easier to work with $\|A\|_{\infty\to\infty}$ which we see is given by $|1 - \mu| + \mu$ (by the formula in Lecture 5), and this is smaller than 1 precisely when $\mu \le 1$.

Mathematical Tripos Part II: Michaelmas Term 2024

Numerical Analysis – Lecture 7

Technique 2.11 (Fourier analysis of stability) Let us now assume a recurrence of the form

$$\sum_{k=r}^{s} a_k u_{m+k}^{n+1} = \sum_{k=r}^{s} b_k u_{m+k}^{n}, \qquad n \in \mathbb{Z}^+, \tag{2.5}$$

where m ranges over $\mathbb{Z}$. (Within our framework of discretizing PDEs of evolution, this corresponds to $-\infty < x < \infty$ in the undelying PDE and so there are no explicit boundary conditions, but the initial condition must be square-integrable in $(-\infty, \infty)$: this is known as a *Cauchy problem*.) The coefficients a_k and b_k are independent of m, n, but typically depend upon μ. We investigate stability by *Fourier analysis*. [Note that it doesn't matter what is the underlying PDE: numerical stability is a feature of algebraic recurrences, not of PDEs!]

Let $\boldsymbol{v} = (v_m)_{m\in\mathbb{Z}} \in \ell_2[\mathbb{Z}]$. Its *Fourier transform* is the function

$$\widehat{v}(\theta) = \sum_{m\in\mathbb{Z}} \mathrm{e}^{-\mathrm{i}m\theta} v_m, \qquad -\pi \le \theta \le \pi.$$

We equip sequences and functions with the norms

$$\|\boldsymbol{v}\| = \Big\{ \sum_{m\in\mathbb{Z}} |v_m|^2 \Big\}^{\frac{1}{2}} \qquad \text{and} \qquad \|\widehat{v}\|_* = \left\{ \frac{1}{2\pi} \int_{-\pi}^{\pi} |\widehat{v}(\theta)|^2 d\theta \right\}^{\frac{1}{2}}.$$

Lemma 2.12 (Parseval's identity) *For any $\boldsymbol{v} \in \ell_2[\mathbb{Z}]$, we have $\|\boldsymbol{v}\| = \|\widehat{v}\|_*$.*

Proof. By definition,

$$\begin{aligned} \|\widehat{v}\|_*^2 &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \Big| \sum_{m\in\mathbb{Z}} \mathrm{e}^{-\mathrm{i}m\theta} v_m \Big|^2 d\theta = \frac{1}{2\pi} \int_{-\pi}^{\pi} \sum_{m\in\mathbb{Z}} \sum_{k\in\mathbb{Z}} v_m \bar{v}_k \mathrm{e}^{-\mathrm{i}(m-k)\theta} d\theta \\ &= \frac{1}{2\pi} \sum_{m\in\mathbb{Z}} \sum_{k\in\mathbb{Z}} v_m \bar{v}_k \int_{-\pi}^{\pi} \mathrm{e}^{-\mathrm{i}(m-k)\theta} d\theta \overset{(*)}{=} \sum_{m\in\mathbb{Z}} \sum_{k\in\mathbb{Z}} v_m \bar{v}_k \delta_{m-k} = \|\boldsymbol{v}\|^2, \end{aligned}$$

where equality $(*)$ is due to the fact that

$$\int_{-\pi}^{\pi} \mathrm{e}^{-\mathrm{i}\ell\theta} d\theta = \begin{cases} 2\pi, & \ell = 0, \\ 0, & \ell \in \mathbb{Z} \setminus \{0\}, \end{cases}$$

□

The implication of the lemma is that the Fourier transform is an *isometry* of the Euclidean norm. This is an important reason underlying its many applications in mathematics and beyond.

Analysis 2.13 (Fourier analysis of stability) For $\theta \in [-\pi, \pi]$, let $\widehat{u}^n(\theta) = \sum_{m\in\mathbb{Z}} \mathrm{e}^{-\mathrm{i}m\theta} u_m^n$ be the Fourier transform of the sequence $\boldsymbol{u}^n \in \ell_2[\mathbb{Z}]$. We multiply the discretized equations (2.5) by $\mathrm{e}^{-\mathrm{i}m\theta}$ and sum up for $m \in \mathbb{Z}$. Thus, the left-hand side yields

$$\begin{aligned} \sum_{m=-\infty}^{\infty} \mathrm{e}^{-\mathrm{i}m\theta} \sum_{k=r}^{s} a_k u_{m+k}^{n+1} &= \sum_{k=r}^{s} a_k \sum_{m=-\infty}^{\infty} \mathrm{e}^{-\mathrm{i}m\theta} u_{m+k}^{n+1} \\ &= \sum_{k=r}^{s} a_k \sum_{m=-\infty}^{\infty} \mathrm{e}^{-\mathrm{i}(m-k)\theta} u_m^{n+1} = \left(\sum_{k=r}^{s} a_k \mathrm{e}^{\mathrm{i}k\theta} \right) \widehat{u}^{n+1}(\theta). \end{aligned}$$

Similarly manipulating the right-hand side, we deduce that

$$\widehat{u}^{n+1}(\theta) = H(\theta)\widehat{u}^n(\theta), \quad \text{where} \quad H(\theta) = \frac{\sum_{k=r}^{s} b_k \mathrm{e}^{\mathrm{i}k\theta}}{\sum_{k=r}^{s} a_k \mathrm{e}^{\mathrm{i}k\theta}}. \tag{2.6}$$

The function H is sometimes called the *amplification factor* of the recurrence (2.5)

Theorem 2.14 *The method (2.5) is stable* $\Leftrightarrow$ $|H(\theta)| \le 1$ *for all* $\theta \in [-\pi, \pi]$.

Proof. The definition of stability is equivalent to the statement that there exists $c > 0$ such that $\|\boldsymbol{u}^n\| \leq c$ for all $n \in \mathbb{Z}^+$. [Because we are solving a Cauchy problem, equations are identical for all $h = \Delta x$, and this simplifies our analysis and eliminates a major difficulty: there is no need to insist explicitly that $\|\boldsymbol{u}^n\|$ remains uniformly bounded when $h \to 0$]. The Fourier transform being an isometry, stability is thus equivalent to $\|\widehat{u}^n\|_* \leq c$ for all $n \in \mathbb{Z}^+$. Iterating (2.6), we obtain

$$\widehat{u}^n(\theta) = [H(\theta)]^n \widehat{u}^0(\theta), \qquad |\theta| \leq \pi, \quad n \in \mathbb{Z}^+. \tag{2.7}$$

1) Assume first that $|H(\theta)| \leq 1$ for all $|\theta| \leq \pi$. Then, by (2.7),

$$|\widehat{u}^n(\theta)| \leq |\widehat{u}^0(\theta)| \qquad \Rightarrow \qquad \|\widehat{u}^n\|_*^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} |\widehat{u}^n(\theta)|^2 \mathrm{d}\theta \leq \frac{1}{2\pi}\int_{-\pi}^{\pi} |\widehat{u}^0(\theta)|^2 \mathrm{d}\theta = \|\widehat{u}^0\|_*^2.$$

Hence stability.

2) Suppose, on the other hand, that there exists $\theta_0 \in [-\pi, \pi]$ such that $|H(\theta_0)| = 1 + 2\epsilon > 1$, say. Since H is continuous, there exist $-\pi \leq \theta_1 < \theta_2 \leq \pi$ such that $|H(\theta)| \geq 1 + \epsilon$ for all $\theta \in [\theta_1, \theta_2]$. We set $\eta = \theta_2 - \theta_1$ and choose as our initial condition the function (or the $\ell_2[\mathbb{Z}]$-sequence)

$$\widehat{u}^0(\theta) = \begin{cases} \sqrt{\frac{2\pi}{\eta}}, & \theta_1 \leq \theta \leq \theta_2, \\ 0, & \text{otherwise}, \end{cases}$$

Then

$$\begin{aligned} \|\widehat{u}^n\|_*^2 &= \tfrac{1}{2\pi}\int_{-\pi}^{\pi} |H(\theta)|^{2n} |\widehat{u}^0(\theta)|^2 \mathrm{d}\theta = \tfrac{1}{2\pi}\int_{\theta_1}^{\theta_2} |H(\theta)|^{2n} |\widehat{u}^0(\theta)|^2 \mathrm{d}\theta \\ &\geq \tfrac{1}{2\pi}(1+\epsilon)^{2n}\int_{\theta_1}^{\theta_2} \tfrac{2\pi}{\eta}\mathrm{d}\theta = (1+\epsilon)^{2n} \to \infty \quad (n \to \infty). \end{aligned}$$

We deduce that the method is unstable. □

Example 2.15 Consider the Cauchy problem for the diffusion equation.

1) For the Euler method

$$u_m^{n+1} = u_m^n + \mu(u_{m-1}^n - 2u_m^n + u_{m+1}^n)\,,$$

we obtain

$$H(\theta) = 1 + \mu\left(\mathrm{e}^{-\mathrm{i}\theta} - 2 + \mathrm{e}^{\mathrm{i}\theta}\right) = 1 - 4\mu\sin^2\tfrac{\theta}{2} \;\in\; [1-4\mu, 1]\,,$$

thus the method is stable iff $\mu \leq \frac{1}{2}$.

2) For the backward Euler method

$$u_m^{n+1} - \mu(u_{m-1}^{n+1} - 2u_m^{n+1} + u_{m+1}^{n+1}) = u_m^n\,,$$

we have

$$H(\theta) = \left[1 - \mu\left(\mathrm{e}^{-\mathrm{i}\theta} - 2 + \mathrm{e}^{\mathrm{i}\theta}\right)\right]^{-1} = \left[1 + 4\mu\sin^2\tfrac{\theta}{2}\right]^{-1} \in (0, 1]\,.$$

thus stability for all μ.

3) The Crank–Nicolson scheme

$$u_m^{n+1} - \tfrac{1}{2}\mu(u_{m-1}^{n+1} - 2u_m^{n+1} + u_{m+1}^{n+1}) = u_m^n + \tfrac{1}{2}\mu(u_{m-1}^n - 2u_m^n + u_{m+1}^n),$$

results in

$$H(\theta) = \frac{1 + \frac{1}{2}\mu(\mathrm{e}^{-\mathrm{i}\theta} - 2 + \mathrm{e}^{\mathrm{i}\theta})}{1 - \frac{1}{2}\mu(\mathrm{e}^{-\mathrm{i}\theta} - 2 + \mathrm{e}^{\mathrm{i}\theta})} = \frac{1 - 2\mu\sin^2\frac{\theta}{2}}{1 + 2\mu\sin^2\frac{\theta}{2}} \in (-1, 1]$$

Hence stability for all $\mu > 0$.

Mathematical Tripos Part II: Michaelmas Term 2024

Numerical Analysis – Lecture 8

Problem 2.18 (The advection equation) We look at the *advection equation* which we already considered in Lecture 6.

$$u_t = u_x, \qquad t \geq 0, \tag{2.6}$$

where $u = u(x,t)$. It is given with the initial condition $u(x,0) = \varphi(x)$. The exact solution of (2.6) is simply $u(x,t) = \varphi(x+t)$, a unilateral shift leftwards. This, however, does not mean that its numerical modelling is easy.

Example 2.19 (Downwind instability) 1) *Downwind instability*: Consider the discretization $\frac{\partial u_m(t)}{\partial x} \approx \frac{1}{2h}\left[u_m(t) - u_{m-1}(t)\right]$, so coming to the ODE $u'_m(t) = \frac{1}{2h}\left[u_m(t) - u_{m-1}(t)\right]$. For the Euler method, the outcome is

$$u_m^{n+1} = u_m^n + \mu(u_m^n - u_{m-1}^n), \quad n \in \mathbb{Z}_+.$$

We can analyze the stability of this method using Fourier analysis. The amplification factor is

$$H(\theta) = 1 + \mu - \mu e^{-\mathrm{i}\theta}.$$

We see that for $\theta = \pi/2$, $|H(\theta)|^2 = (1+\mu)^2 + \mu^2 > 1$, and so the method is unstable for all $\mu > 0$.

Method 2.20 (Upwind method) *Upwind scheme*: If we semidiscretize $\frac{\partial u_m(t)}{\partial x} \approx \frac{1}{h}\left[u_{m+1}(t) - u_m(t)\right]$, and solve the ODE again by Euler's method, then the result is

$$u_m^{n+1} = u_m^n + \mu(u_{m+1}^n - u_m^n), \quad n \in \mathbb{Z}_+ \tag{2.7}$$

The local error is $\mathcal{O}(k^2+kh)$ which is $\mathcal{O}(h^2)$ for a fixed μ, hence convergence if the method is stable. We can again use Fourier analysis to analyze stability. The amplification factor is

$$H(\theta) = 1 - \mu + \mu e^{\mathrm{i}\theta}$$

and we see that $|H(\theta)| = |1 - \mu + \mu e^{\mathrm{i}\theta}| \leq |1-\mu| + \mu = 1$ for $\mu \in [0,1]$. Hence we have stability for $\mu \leq 1$. If $\mu > 1$, then note that $|H(\pi)| = |1 - 2\mu| > 1$, and so we have instability for $\mu > 1$.

Matlab demo: Download the Matlab GUI for *Solving the Advection Equation, Upwinding and Stability* from `https://www.damtp.cam.ac.uk/user/hf323/M21-II-NA/demos/index.html` and solve the advection equation (2.6) with the different methods provided in the demonstration. Experience what can go wrong when "winding" in the wrong direction!

What about the case when $0 \leq x \leq 1$ (bounded domain)? Recall from Lecture 6 when we considered the Euler method for the advection equation

$$u_m^{n+1} - u_m^n = \mu(u_{m+1}^n - u_m^n), \qquad m = 1...M\,.$$

We have $\boldsymbol{u}^{n+1} = A\boldsymbol{u}^n$, where

$$A = \begin{bmatrix} 1-\mu & \mu & & \\ & 1-\mu & \ddots & \\ & & \ddots & \mu \\ & & & 1-\mu \end{bmatrix},$$

but A is *not* normal, and although its eigenvalues are bounded by 1 for $\mu \leq 2$ (note $1-\mu$ is the only eigenvalue of A), it is the matrix induced norm of A that matters. For this example, it is easier to work with $\|A\|_{\infty\to\infty}$ which we see is given by $|1-\mu|+\mu$ (by the formula in Lecture 5), and this is smaller than 1 precisely when $\mu \leq 1$.

Method 2.21 (The leapfrog method) *Leap-frog method*: We semidicretize (2.6) as $\frac{\partial u_m(t)}{\partial x} \approx \frac{1}{2h}\left[u_{m+1}(t) - u_{m-1}(t)\right]$, but now solve the ODE with the second-order *midpoint rule*

$$\boldsymbol{y}_{n+1} = \boldsymbol{y}_{n-1} + 2k\boldsymbol{f}(t_n, \boldsymbol{y}_n), \qquad n \in \mathbb{Z}_+\,.$$

The outcome is the two-step *leapfrog* method

$$u_m^{n+1} = \mu\,(u_{m+1}^n - u_{m-1}^n) + u_m^{n-1}. \tag{2.8}$$

The local error is now $\mathcal{O}(k^3+kh^2) = \mathcal{O}(h^3)$.

We analyse stability by the Fourier technique, assuming that we are solving a Cauchy problem. Thus, proceeding as before,

$$\widehat{u}^{n+1}(\theta) = \mu\left(\mathrm{e}^{\mathrm{i}\theta} - \mathrm{e}^{-\mathrm{i}\theta}\right)\widehat{u}^n(\theta) + \widehat{u}^{n-1}(\theta) \tag{2.9}$$

whence

$$\widehat{u}^{n+1}(\theta) - 2\mathrm{i}\mu\,\sin\theta\,\widehat{u}^n(\theta) - \widehat{u}^{n-1}(\theta) = 0, \qquad n \in \mathbb{Z}_+\,,$$

and our goal is to determine values of μ such that $|\widehat{u}^n(\theta)|$ is uniformly bounded for all n, θ.

This is a difference equation $w_{n+1} + bw_n + cw_{n-1} = 0$ with the general solution $w_n = c_1\lambda_1^n + c_2\lambda_2^n$, where λ_1, λ_2 are the roots of the characteristic equation $\lambda^2+b\lambda+c = 0$, and c_1, c_2 are constants, dependent on the initial values w_0 and w_1. If $\lambda_1 = \lambda_2$, then solution is $w_n = (c_1 + c_2 n)\lambda^n$. In our case, we obtain

$$\lambda_{1,2}(\theta) = \mathrm{i}\mu\sin\theta \pm \sqrt{1 - \mu^2\sin^2\theta}\,.$$

Stability is equivalent to $|\lambda_{1,2}(\theta)| \le 1$ for all θ and this is true if and only if $\mu \le 1$.

Problem 2.22 (The wave equation) Consider the wave equation

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2} \qquad t \ge 0,$$

given with initial conditions $u(x,0)$ and $u_t(x,0) = \frac{\partial u}{\partial t}(x,0)$. The usual approximation looks as follows

$$u_m^{n+1} - 2u_m^n + u_m^{n-1} = \mu(u_{m+1}^n - 2u_m^n + u_{m-1}^n)\,,$$

with the Courant number being now $\mu = k^2/h^2$.

The Fourier analysis (for Cauchy problem) provides

$$\widehat{u}^{n+1}(\theta) - 2\widehat{u}^n(\theta) + \widehat{u}^{n-1}(\theta) = -4\mu\sin^2\tfrac{\theta}{2}\,\widehat{u}^n(\theta)\,,$$

with the characteristic equation $\lambda^2 - 2(1 - 2\mu\sin^2\frac{\theta}{2})\lambda + 1 = 0$. The product of the roots is one, therefore stability (that requires the moduli of both λ to be at most one) is equivalent to the roots being complex conjugate, so we require

$$(1 - 2\mu\sin^2\tfrac{\theta}{2})^2 \le 1.$$

This condition is achieved if and only if $\mu = k^2/h^2 \le 1$.

Mathematical Tripos Part II: Michaelmas Term 2024

Numerical Analysis – Lecture 9

Problem 2.25 (The diffusion equation in two space dimensions) We are solving

$$\frac{\partial u}{\partial t} = \nabla^2 u, \qquad 0 \le x, y \le 1, \quad t \ge 0, \tag{2.11}$$

where $u = u(x, y, t)$, together with initial conditions at $t = 0$ and Dirichlet boundary conditions at $\partial\Omega$, where $\Omega = [0, 1]^2 \times [0, \infty)$. It is straightforward to generalize our derivation of numerical algorithms, e.g. by the method of lines. Thus, let $u_{\ell,m}(t) \approx u(\ell h, mh, t)$, where $h = \Delta x = \Delta y$, and let $u^n_{\ell,m} \approx u_{\ell,m}(nk)$ where $k = \Delta t$. The five-point formula results in

$$u'_{\ell,m} = \tfrac{1}{h^2}(u_{\ell-1,m} + u_{\ell+1,m} + u_{\ell,m-1} + u_{\ell,m+1} - 4u_{\ell,m}),$$

or in the matrix form

$$\boldsymbol{u}' = \tfrac{1}{h^2} A_* \boldsymbol{u}, \qquad \boldsymbol{u} = (u_{\ell,m}) \in \mathbb{R}^N, \tag{2.12}$$

where A_* is the block TST matrix of the five-point scheme:

$$A_* = \begin{bmatrix} H & I & & \\ I & \ddots & \ddots & \\ & \ddots & \ddots & I \\ & & I & H \end{bmatrix}, \quad H = \begin{bmatrix} -4 & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & -4 \end{bmatrix}.$$

Thus, the Euler method yields

$$u^{n+1}_{\ell,m} = u^n_{\ell,m} + \mu(u^n_{\ell-1,m} + u^n_{\ell+1,m} + u^n_{\ell,m-1} + u^n_{\ell,m+1} - 4u^n_{\ell,m}), \tag{2.13}$$

or in the matrix form

$$\boldsymbol{u}^{n+1} = A\boldsymbol{u}^n, \qquad A = I + \mu A_*$$

where, as before, $\mu = \frac{k}{h^2} = \frac{\Delta t}{(\Delta x)^2}$. The local error is $\boldsymbol{\eta} = \mathcal{O}(k^2 + kh^2) = \mathcal{O}(h^4)$. To analyse stability, we notice that A is symmetric, hence normal, and its eigenvalues are related to those of A_* by the rule

$$\lambda_{k,\ell}(A) = 1 + \mu\lambda_{k,\ell}(A_*) \overset{\text{Prop. 1.12}}{=} 1 - 4\mu\left(\sin^2\tfrac{\pi k h}{2} + \sin^2\tfrac{\pi \ell h}{2}\right).$$

Consequently,

$$\sup_{h>0} \rho(A) = \max\{1, |1 - 8\mu|\}, \qquad \text{hence} \qquad \mu \le \tfrac{1}{4} \quad \Leftrightarrow \quad \text{stability}.$$

Method 2.26 (Fourier analysis) Fourier analysis generalizes to two dimensions: of course, we now need to extend the range of (x, y) in (2.11) from $0 \le x, y \le 1$ to $x, y \in \mathbb{R}$. A 2D Fourier transform reads

$$\widehat{u}(\theta, \psi) = \sum_{\ell,m\in\mathbb{Z}} u_{\ell,m} \mathrm{e}^{-\mathrm{i}(\ell\theta + m\psi)}$$

and all our results readily generalize. In particular, the Fourier transform is an isometry from $\ell_2[\mathbb{Z}^2]$ to $L_2([-\pi, \pi]^2)$, i.e.

$$\Big(\sum_{\ell,m\in\mathbb{Z}} |u_{\ell,m}|^2\Big)^{1/2} =: \|\boldsymbol{u}\| = \|\widehat{u}\|_* := \Big(\frac{1}{4\pi^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi} |\widehat{u}(\theta, \psi)|^2 \, d\theta \, d\psi\Big)^{1/2},$$

and the method is stable iff $|H(\theta, \psi)| \le 1$ for all $\theta, \psi \in [-\pi, \pi]$. The proofs are an easy elaboration on the one-dimensional theory. Insofar as the Euler method (2.13) is concerned,

$$H(\theta, \psi) = 1 + \mu\left(\mathrm{e}^{-\mathrm{i}\theta} + \mathrm{e}^{\mathrm{i}\theta} + \mathrm{e}^{-\mathrm{i}\psi} + \mathrm{e}^{\mathrm{i}\psi} - 4\right) = 1 - 4\mu\big(\sin^2\tfrac{\theta}{2} + \sin^2\tfrac{\psi}{2}\big),$$

and we again deduce stability if and only if $\mu \le \frac{1}{4}$.

Method 2.27 (Crank-Nicolson for 2D) Applying the trapezoidal rule to our semi-dicretization (2.12) we obtain the two-dimensional Crank-Nicolson method:

$$(I - \tfrac{1}{2}\mu A_*)\,\boldsymbol{u}^{n+1} = (I + \tfrac{1}{2}\mu A_*)\,\boldsymbol{u}^n\,, \tag{2.14}$$

in which we move from the n-th to the $(n+1)$-st level by solving the system of linear equations $B\boldsymbol{u}^{n+1} = C\boldsymbol{u}^n$, or $\boldsymbol{u}^{n+1} = B^{-1}C\boldsymbol{u}^n$. For stability, similarly to the one-dimensional case, the eigenvalue analysis implies that $A = B^{-1}C$ is normal and shares the same eigenvectors with B and C, hence

$$\lambda(A) = \frac{\lambda(C)}{\lambda(B)} = \frac{1 + \frac{1}{2}\mu\lambda(A_*)}{1 - \frac{1}{2}\mu\lambda(A_*)} \quad \Rightarrow \quad |\lambda(A)| < 1 \text{ as } \lambda(A_*) < 0$$

and the method is stable for all μ. The same result can be obtained through the Fourier analysis.

Matlab demo: Download the Matlab GUI for *Solving the Wave and Diffusion Equations in 2D* from `http://www.damtp.cam.ac.uk/user/hf323/M21-II-NA/demos/pdes_2d/pdes_2d.html` and solve the diffusion equation (2.11) for different initial conditions. For the numerical solution of the equation you can choose from the Euler method and the Crank-Nicolson scheme. The GUI allows you to solve the wave equation as well. Compare the behaviour of solutions!

Technique 2.28 (Splitting) In all the examples of semi-discretization we have seen so far, we always reach a linear system of ODE of the form:

$$\boldsymbol{u}' = A\boldsymbol{u}, \qquad \boldsymbol{u}(0) = \boldsymbol{u}_0. \tag{2.15}$$

The solution of this linear system of ODE is given by

$$\boldsymbol{u}(t) = \mathrm{e}^{tA}\boldsymbol{u}_0 \tag{2.16}$$

where the *matrix exponential* function is defined by $\mathrm{e}^B := \sum_{k=0}^{\infty} \frac{1}{k!}B^k$. It is easily verified that $\mathrm{d}\mathrm{e}^{tA}/\mathrm{d}t = A\mathrm{e}^{tA}$, therefore (2.16) is indeed a solution of (2.15).

If A can be diagonalized $A = VDV^{-1}$, then $\mathrm{e}^{tA} = V\mathrm{e}^{tD}V^{-1}$ where e^{tD} is the diagonal matrix consisting $\mathrm{diag}\,(\mathrm{e}^{tD_{ii}})$. As such one can compute the solution of (2.15) exactly. However computing an eigenvalue decomposition can be costly, and so one would like to consider more efficient methods, based on the solution of sparse linear systems instead.

Observe that one-step methods for solving (2.15) are approximating a matrix exponential. Indeed, with $k = \Delta t$, we have:

Euler:	$\boldsymbol{u}^{n+1} = (I + kA)\boldsymbol{u}^n,$	$\mathrm{e}^z = 1 + z + \mathcal{O}(z^2);$
Implicit Euler:	$\boldsymbol{u}^{n+1} = (I - kA)^{-1}\boldsymbol{u}^n,$	$\mathrm{e}^z = (1 - z)^{-1} + \mathcal{O}(z^2);$
Trapezoidal:	$\boldsymbol{u}^{n+1} = \left(I - \frac{1}{2}kA\right)^{-1}\left(I + \frac{1}{2}kA\right)\boldsymbol{u}^n,$	$\mathrm{e}^z = \frac{1+\frac{1}{2}z}{1-\frac{1}{2}z} + \mathcal{O}(z^3).$

In practice the matrix A is very sparse, and this can be exploited when solving linear systems e.g., for the implicit Euler or Trapezoidal Rule.

In many cases, the matrix A is naturally expressed as a *sum of two matrices*, $A = B + C$. For example, when discretizing the diffusion equation in 2D with zero boundary conditions, we have $A = \frac{1}{h^2}(A_x + A_y)$ where $\frac{1}{h^2}A_x \in \mathbb{R}^{M^2\times M^2}$ corresponds to the 3-point discretization of $\frac{\partial^2}{\partial x^2}$, and $\frac{1}{h^2}A_y \in \mathbb{R}^{M^2\times M^2}$ corresponds to the 3-point discretization of $\frac{\partial^2}{\partial y^2}$. In matrix notations, if the grid points are ordered by columns, then we have:

$$A_x = \begin{bmatrix} -2I & I & & \\ I & \ddots & \ddots & \\ & \ddots & \ddots & I \\ & & I & -2I \end{bmatrix}, \; A_y = \begin{bmatrix} G & & & \\ & G & & \\ & & \ddots & \\ & & & G \end{bmatrix}, \; G = \begin{bmatrix} -2 & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & -2 \end{bmatrix} \in \mathbb{R}^{M\times M}. \tag{2.17}$$

Remark: It is convenient to note that $A_x = G \otimes I$ and $A_y = I \otimes G$, where $\otimes$ is the Kronecker product of matrices (`kron` in Matlab) defined by

$$A \otimes B = \begin{bmatrix} A_{11}B & A_{12}B & \dots & A_{1m_A}B \\ A_{21}B & A_{22}B & \dots & A_{2m_A}B \\ \vdots & & & \\ A_{n_A1}B & \dots & \dots & A_{n_Am_A}B \end{bmatrix} \in \mathbb{R}^{n_An_B \times m_Am_B}$$

where $A \in \mathbb{R}^{n_A \times m_A}$ and $B \in \mathbb{R}^{n_B \times m_B}$.

In general, $\exp(t(B+C)) \neq \exp(tB)\exp(tC)$. Equality holds however when B and C commute.

Proposition 2.29 *For any matrices B, C,*

$$\mathrm{e}^{t(B+C)} = \mathrm{e}^{tB}\mathrm{e}^{tC} + \tfrac{1}{2}t^2(CB - BC) + \mathcal{O}(t^3). \tag{2.18}$$

If B and C commute, then $\mathrm{e}^{B+C} = \mathrm{e}^B\mathrm{e}^C$.

Proof. We Taylor-expand both expressions $\mathrm{e}^{tB}\mathrm{e}^{tC}$ and $\mathrm{e}^{t(B+C)}$:

$$\begin{aligned} \mathrm{e}^{tB}\mathrm{e}^{tC} &= (I + tB + t^2B^2/2 + \mathcal{O}(t^3))(I + tC + t^2C^2/2 + \mathcal{O}(t^3)) \\ &= I + t(B+C) + \tfrac{t^2}{2}(B^2 + C^2 + 2BC) + \mathcal{O}(t^3) \end{aligned}$$

and

$$\begin{aligned} \mathrm{e}^{t(B+C)} &= I + t(B+C) + \tfrac{t^2}{2}(B+C)^2 + \mathcal{O}(t^3) \\ &= I + t(B+C) + \tfrac{t^2}{2}(B^2 + C^2 + BC + CB) + \mathcal{O}(t^3). \end{aligned}$$

Equation (2.18) follows.

When B and C commute, we can write:

$$\exp(B+C) = \sum_{n=0}^{\infty} \frac{1}{n!}(B+C)^n = \sum_{n=0}^{\infty} \frac{1}{n!}\left(\sum_{k=0}^{n} \binom{n}{k} B^{n-k}C^k\right) = \sum_{k=0}^{\infty}\sum_{n=k}^{\infty} \frac{1}{n!}\binom{n}{k} B^{n-k}C^k.$$

Recall that $\binom{n}{k} = \dfrac{n!}{k!(n-k)!}$, so

$$\exp(B+C) = \sum_{k=0}^{\infty}\sum_{n=k}^{\infty} \frac{1}{k!(n-k)!} B^{n-k}C^k = \sum_{k=0}^{\infty}\sum_{m=0}^{\infty} \frac{1}{k!m!} B^m C^k = \mathrm{e}^B\mathrm{e}^C.$$

□

Mathematical Tripos Part II: Michaelmas Term 2024

Numerical Analysis – Lecture 10

Technique 2.31 (Splitting for the 2D diffusion equation) Recall that for the 2D diffusion equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial x^2}$$

using the five-point discretisation scheme for the Laplacian yields the following ODE

$$\frac{d\boldsymbol{u}}{dt} = \frac{1}{h^2}(A_x + A_y)\boldsymbol{u}$$

where the matrices A_x and A_y are expressed as $A_x = G \otimes I$ and $A_y = I \otimes G$, where $\otimes$ is the Kronecker product, and G is the $M \times M$ tridiagonal matrix

$$G = \begin{bmatrix} -2 & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & -2 \end{bmatrix} \in \mathbb{R}^{M\times M}.$$

It is straightforward to verify that A_x and A_y commute; namely $A_xA_y = A_yA_x = G\otimes G$ (check out the basic rules of multiplication with the kronecker product https://en.wikipedia.org/wiki/Kronecker_product). This should not come as a suprise since the operators $\partial^2/\partial x^2$ and $\partial^2/\partial y^2$, which A_x/h^2 and A_y/h^2 approximate, are known to commute. So we can write

$$\mathrm{e}^{k(A_x+A_y)/h^2} = \mathrm{e}^{kA_x/h^2}\mathrm{e}^{kA_y/h^2}.$$

This means that the solution of the semi-discretized diffusion equation in 2D, with zero boundary conditions, satisfies

$$\boldsymbol{u}^{n+1} = \mathrm{e}^{kA_x/h^2}\mathrm{e}^{kA_y/h^2}\boldsymbol{u}^n. \tag{2.17}$$

The split Crank-Nicolson scheme: In the split Crank-Nicolson scheme, we approximate each exponential map in (2.17) by the rational function

$$r(z) = (1 + z/2)(1 - z/2)^{-1},$$

which leads to

$$\boldsymbol{u}^{n+1} = (I + \tfrac{\mu}{2}A_x)(I - \tfrac{\mu}{2}A_x)^{-1}(I + \tfrac{\mu}{2}A_y)(I - \tfrac{\mu}{2}A_y)^{-1}\boldsymbol{u}^n. \tag{2.18}$$

Note that computing $\boldsymbol{u}^{n+1/2} = (I + \frac{\mu}{2}A_y)(I - \frac{\mu}{2}A_y)^{-1}\boldsymbol{u}^n$ can be done efficiently in $\mathcal{O}(M^2)$ time as A_y is block-diagonal, and the matrices G are tridiagonal (each tridiagonal solve requires $\mathcal{O}(M)$ time, and we have M of these). Computing $\boldsymbol{u}^{n+1} = (I+\frac{\mu}{2}A_x)(I-\frac{\mu}{2}A_x)^{-1}\boldsymbol{u}^{n+1/2}$ can also be done in $\mathcal{O}(M^2)$ time, since A_x is also block-diagonal provided we appropriately permute the rows and columns so that the grid ordering is by rows instead of columns. This means that the update step (2.18) of Split-Crank-Nicolson can be performed in time $\mathcal{O}(M^2)$ and only requires tridiagonal matrix solves (no FFT needed).

Stability: One can easily verify stability of the split Crank-Nicolson scheme. Indeed, we can write

$$\|r(\mu A_x)r(\mu A_y)\|_2 \leq \|r(\mu A_x)\|_2\|r(\mu A_y)\|_2 \leq 1$$

since, as seen in previous lectures, $\|r(\mu A_x)\|_2 = \|(I + \frac{\mu}{2}A_x)(I - \frac{\mu}{2}A_x)^{-1}\|_2 \leq 1$ since A_x is symmetric and its eigenvalues are ≤ 0. (Same for $\|r(\mu A_y)\|_2$.)

Exercise: Check the consistency of the scheme

$$\boldsymbol{u}^{n+1} = r(\mu A_x)r(\mu A_y)\boldsymbol{u}^n.$$

In particular, show that split Crank-Nicolson has the 'same' local error as the classical Crank-Nicolson scheme. That is the local error is $\mathcal{O}(k^3 + kh^2)$.

Example 2.32 Consider the general diffusion equation

$$\frac{\partial u}{\partial t} = \nabla^\top \left(a(x,y)\nabla u\right) + f(x,y) = \frac{\partial}{\partial x}\left(a(x,y)\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(a(x,y)\frac{\partial u}{\partial y}\right) + f(x,y), \tag{2.19}$$

where $a(x,y) > \alpha > 0$ and $f(x,y)$ are given, together with initial conditions on $[0,1]^2$ and Dirichlet boundary conditions along $\partial[0,1]^2 \times [0,\infty)$. Replace each space derivative by *central differences* at midpoints,

$$\frac{\mathrm{d}g(\xi)}{\mathrm{d}\xi} \approx \frac{g(\xi + \frac{1}{2}h) - g(\xi - \frac{1}{2}h)}{h}\,,$$

resulting in the ODE system

$$\begin{aligned} u'_{\ell,m} \quad = \quad & \tfrac{1}{h^2}\Big[a_{\ell-\frac{1}{2},m}u_{\ell-1,m} + a_{\ell+\frac{1}{2},m}u_{\ell+1,m} + a_{\ell,m-\frac{1}{2}}u_{\ell,m-1} + a_{\ell,m+\frac{1}{2}}u_{\ell,m+1} \\ & - \big(a_{\ell-\frac{1}{2},m} + a_{\ell+\frac{1}{2},m} + a_{\ell,m-\frac{1}{2}} + a_{\ell,m+\frac{1}{2}}\big)u_{\ell,m}\Big] + f_{\ell,m}. \end{aligned} \tag{2.20}$$

Assuming zero boundary conditions, we have a system $\boldsymbol{u}' = A\boldsymbol{u}$, and the matrix A can be split as $A = \frac{1}{h^2}(A_x + A_y)$. Here, A_x and A_y are again constructed from the contribution of discretizations in the x- and y-directions respectively, namely A_x includes all the $a_{\ell\pm\frac{1}{2},m}$ terms, and A_y consists of the remaining $a_{\ell,m\pm\frac{1}{2}}$ components. The resulting operators A_x and A_y do not necessarily commute, and so the splitting scheme

$$\boldsymbol{u}^{n+1} = \mathrm{e}^{kA_x/h^2}\mathrm{e}^{kA_y/h^2}\boldsymbol{u}^n$$

will carry an error of $\mathcal{O}(k^2)$.

Strang splitting : One can obtain better splitting approximations of $\mathrm{e}^{t(B+C)}$. For example it is not hard to prove that $\mathrm{e}^{\frac{1}{2}tB}\mathrm{e}^{tC}\mathrm{e}^{\frac{1}{2}tB}$ gives a $\mathcal{O}(t^3)$ approximation of $\mathrm{e}^{t(B+C)}$, i.e.,

$$\mathrm{e}^{t(B+C)} = \mathrm{e}^{\frac{1}{2}tB}\mathrm{e}^{tC}\mathrm{e}^{\frac{1}{2}tB} + \mathcal{O}(t^3). \tag{2.21}$$

Technique 2.33 (Splitting methods) Recall that, for $z_1, z_2 \in \mathbb{C}$, we have $\mathrm{e}^{z_1+z_2} = \mathrm{e}^{z_1}\mathrm{e}^{z_2}$ and had this been true for the matrices, i.e. that $\mathrm{e}^{tA} = \mathrm{e}^{t(B+C)} = \mathrm{e}^{tB}\mathrm{e}^{tC}$, we could have approximated each component of the exponent of $A = A_x + A_y$ with the trapezoidal rule, say, to produce

$$\boldsymbol{u}^{n+1} = \big(I - \tfrac{1}{2}\mu A_x\big)^{-1}\big(I + \tfrac{1}{2}\mu A_x\big)\big(I - \tfrac{1}{2}\mu A_y\big)^{-1}\big(I + \tfrac{1}{2}\mu A_y\big)\boldsymbol{u}^n\,, \qquad \mu = k/h^2\,, \tag{2.22}$$

and since both $I - \frac{1}{2}\mu A_x$ and $I - \frac{1}{2}\mu A_y$ are tridiagonal, this system can be solved very cheaply.

Unfortunately, the assumption that $\mathrm{e}^{t(B+C)} = \mathrm{e}^{tB}\mathrm{e}^{tC}$ is, in general, false. Not all hope is lost, though, and we will demonstrate that, suitably implemented, splitting is a powerful technique to reduce drastically the expense of numerical solution.

Method 2.34 (Splitting of inhomogeneous systems) Our exposition so far has been limited to the case of zero boundary conditions. In general, the linear ODE system is of the form

$$\boldsymbol{u}' = A\boldsymbol{u} + \boldsymbol{b}, \qquad \boldsymbol{u}(0) = \boldsymbol{u}^0, \tag{2.23}$$

where $\boldsymbol{b}$ originates in boundary conditions (and, possibly, in a forcing term $f(x,y)$ in the original PDE (2.19)). Note that our analysis should accommodate $\boldsymbol{b} = \boldsymbol{b}(t)$, since boundary conditions might vary in time! The *exact* solution of (2.23) is provided by the *variation of constants* formula

$$\boldsymbol{u}(t) = \mathrm{e}^{tA}\boldsymbol{u}(0) + \int_0^t \mathrm{e}^{(t-s)A}\boldsymbol{b}(s)\,\mathrm{d}s, \qquad t \geq 0,$$

therefore

$$\boldsymbol{u}(t_{n+1}) = \mathrm{e}^{kA}\boldsymbol{u}(t_n) + \int_{t_n}^{t_{n+1}} \mathrm{e}^{(t_{n+1}-s)A}\boldsymbol{b}(s)\,\mathrm{d}s\,.$$

The integral on the right-hand side can be evaluated using quadrature.

For example, the trapezoidal rule $\int_0^k g(\tau)\,\mathrm{d}\tau = \frac{1}{2}k[g(0)+g(k)] + \mathcal{O}(k^3)$ gives

$$\boldsymbol{u}(t_{n+1}) \;\approx\; \mathrm{e}^{kA}\boldsymbol{u}(t_n) + \tfrac{1}{2}k[\mathrm{e}^{kA}\boldsymbol{b}(t_n) + \boldsymbol{b}(t_{n+1})],$$

with a local error of $\mathcal{O}(k^3)$. We can now replace exponentials with their splittings. For example, Strang's splitting (2.21), together with the rational function approximation $r(z) = (1+z/2)/(1-z/2)$ of the exponential map, results in

$$\boldsymbol{u}^{n+1} \;=\; r\big(\tfrac{1}{2}kB\big)\, r\big(kC\big)\, r\big(\tfrac{1}{2}kB\big)\big[\boldsymbol{u}^n + \tfrac{1}{2}k\boldsymbol{b}^n\big] + \tfrac{1}{2}k\boldsymbol{b}^{n+1}.$$

As before, everything reduces to (inexpensive) solution of tridiagonal systems.

Prof. A. C. Hansen

Mathematical Tripos Part II: Michaelmas Term 2024

Numerical Analysis – Lecture 11

3 Spectral Methods

General idea of spectral methods. The basic idea of spectral methods is simple. Consider a PDE of the form

$$\mathcal{L}u = f \tag{3.1}$$

where $\mathcal{L}$ is a differential operator (e.g., $\mathcal{L} = \frac{\partial^2}{\partial x^2}$, or $\mathcal{L} = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$, etc.) and f is a right-hand side function. We consider a finite-dimensional subspace of functions V spanned by a basis $\psi_1, \ldots, \psi_N$. A typical choice for V is a space of (trigonometric) polynomials of finite degree. We seek an approximate solution to the PDE by a linear combination of the ψ_n, i.e., $u_N(x) = \sum_{n=1}^N c_n \psi_n(x)$. Plugging $u_N(x)$ in the PDE we get the following linear equation in the unknowns (c_n):

$$\sum_{n=1}^N c_n \mathcal{L}\psi_n = f. \tag{3.2}$$

In general the equation will not have a solution, as there is no reason to expect that the original PDE has a solution in the subspace V. However, we can seek to satisfy equation (3.2) approximately. Assume that the $(\psi_n)_{1\le n\le N}$ are an orthonormal family of functions, with respect to some inner product $\langle\cdot,\cdot\rangle$. Then instead of looking for (c_n) that satisfy (3.2), we will require only that the projection of $\mathcal{L}u_N - f$ on the subspace V is zero. This is the same as requiring that

$$\sum_{n=1}^N c_n \langle \mathcal{L}\psi_n, \psi_m\rangle = \langle f, \psi_m\rangle \qquad \forall m = 1, \ldots, N. \tag{3.3}$$

If we call A the matrix $A_{m,n} = \langle \mathcal{L}\psi_n, \psi_m\rangle$, we end up with a $N \times N$ linear system $Ac = \tilde{f}$, where $\tilde{f}_m = \langle f, \psi_m\rangle$.

Discussion 3.1 (Large matrices versus small matrices) Finite difference schemes rest upon the replacement of derivatives by a linear combination of function values. This leads to the solution of a system of algebraic equations, which on the one hand tends to be large (due to the slow convergence properties of the approximation) but on the other hand is highly structured and sparse, leading itself to effective algorithms for its solution. We will get to know some of these algorithms in Section 4.

However, an enticing alternative to this strategy are methods that produce small matrices in the first place. Although, these matrices will usually not be sparse anymore, the much smaller the size of the matrices renders its solution affordable. The key point for such approximations are better convergence properties requiring much smaller number of parameters.

Problem 3.2 (Fourier approximation of functions) We consider the *truncated Fourier approximation* of a function f on the interval $[-1, 1]$:

$$f(x) \approx \phi_N(x) = \sum_{n=-N/2+1}^{N/2} \widehat{f}_n e^{i\pi n x}, \quad x \in [-1, 1], \tag{3.4}$$

where here and elsewhere in this section $N \ge 2$ is an even integer and

$$\widehat{f}_n = \frac{1}{2}\int_{-1}^{1} f(t) e^{-i\pi n t}\, dt, \quad n \in \mathbb{Z}$$

are the (Fourier) coefficients of this approximation. We want to analyse the approximation properties of (3.4).

Theorem 3.3 (The de la Valleé Poussin theorem) *If the function f is Riemann integrable and $\widehat{f}_n = \mathcal{O}(n^{-1})$ for $|n| \gg 1$, then $\phi_N(x) = f(x) + \mathcal{O}(N^{-1})$ as $N \to \infty$ for every point $x \in (-1,1)$ where f is Lipschitz.*

Remark 3.4 (The Gibbs effect at the end points) Note that if f is smoothly differentiable then, integrating by parts,

$$\widehat{f}_n = \frac{(-1)^{n+1}}{2\pi i n}[f(1) - f(-1)] + \frac{1}{\pi i n}\widehat{f'_n} = \mathcal{O}(n^{-1}) \text{ for } |n| \gg 1.$$

Since such an f is Lipschitz on $(-1,1)$, we deduce from Theorem 3.3 that ϕ_N converges to f there with speed $\mathcal{O}(N^{-1})$. However, convergence with speed $\mathcal{O}(N^{-1})$ is very slow and moreover, we cannot guarantee convergence at the endpoints -1 and 1. In fact, it is possible to show that

$$\phi_N(\pm 1) \to \frac{1}{2}[f(-1) + f(1)] \text{ as } n \to \infty$$

and hence, unless f is periodic we fail to converge.

Method 3.5 (Fourier approximation for periodic functions) Suppose f is an analytic function in $[-1,1]$, that can be extended analytically to a closed complex domain Ω. In addition let f be periodic with period 2. In particular, $f^{(m)}(-1) = f^{(m)}(1)$ for all $m \in \mathbb{Z}_+$. Then, by multiple integration by parts, we get

$$\widehat{f}_n = \frac{1}{\pi i n}\widehat{f'_n} = \frac{1}{(\pi i n)^2}\widehat{f''_n} = \frac{1}{(\pi i n)^3}\widehat{f'''_n} = \dots.$$

Thus, we have

$$\widehat{f}_n = \frac{1}{(\pi i n)^m}\widehat{f_n^{(m)}}, \quad m = 0, 1, \dots. \tag{3.5}$$

But, how large is $\big|\widehat{f_n^{(m)}}\big|$? To answer this question we use Cauchy's theorem of complex analysis, which states that

$$f^{(m)}(x) = \frac{m!}{2\pi i}\int_\gamma \frac{f(z)\,dz}{(z-x)^{m+1}}, \quad x \in [-1,1],$$

where γ is the positively oriented boundary of Ω. Therefore, with $\alpha^{-1} > 0$ being the minimal distance between γ and $[-1,1]$ and $M = \max\{|f(z)| : z \in \gamma\} < \infty$, it follows that

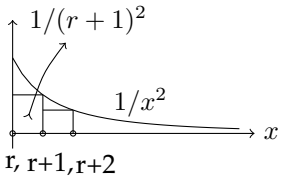
$$|f^{(m)}(x)| \le \frac{m!}{2\pi}\int_\gamma \frac{|f(z)|\,|dz|}{|z-x|^{m+1}} \le \frac{M \text{ length } \gamma}{2\pi} m!\,\alpha^{m+1},$$

and hence, we can bound $\big|\widehat{f_n^{(m)}}\big| \le c\,m!\,\alpha^{m+1}$ for some $c > 0$. Now, using (3.5) and the above upper bound,

$$\begin{aligned} |\phi_N(x) - f(x)| &= \Big|\sum_{n=-N/2+1}^{N/2} \widehat{f}_n e^{i\pi n x} - \sum_{n=-\infty}^{\infty} \widehat{f}_n e^{i\pi n x}\Big| \\ &\le \sum_{|n|\ge N/2} |\widehat{f}_n| = \sum_{|n|\ge N/2} \frac{|\widehat{f_n^{(m)}}|}{|\pi n|^m} \le \frac{c m! \alpha^{m+1}}{\pi^m} \sum_{n=N/2}^{+\infty} \frac{1}{n^m}. \end{aligned}$$

Using, that for any $r \in \mathbb{N}$, and $m > 1$

$$\sum_{n=r+1}^{+\infty} \frac{1}{n^m} \le \int_r^\infty \frac{dt}{t^m} = \frac{1}{m-1} r^{-m+1},$$



we deduce that

$$|\phi_N(x) - f(x)| \leq c' m! \left(\frac{\alpha}{\pi N}\right)^{m-1}, \quad m \geq 2.$$

Finally, we have a competition between $(\alpha/(\pi N))^{m-1}$ and $m!$ for large m. Because of Stirling's formula

$$m! \approx \sqrt{2\pi}\, m^{m+1/2} e^{-m}$$

we have

$$m!\Big(\frac{\alpha}{\pi N}\Big)^{m-1} \approx \sqrt{2\pi m}\, \frac{m}{e}\Big(\frac{\alpha m}{\pi e N}\Big)^{m-1}$$

which becomes very small for large N. Hence, $|\phi_N - f| = \mathcal{O}(N^{-p})$ for any $p \in \mathbb{N}$ and we deduce that the Fourier approximation of an analytic periodic function is of infinite order.

Definition 3.6 (Convergence at spectral speed) An N-term approximation ϕ_N of a function f converges to f at *spectral* speed if $\|\phi_N - f\|$ decays faster than $\mathcal{O}(N^{-p})$ for any $p = 1, 2, \ldots$.

Remark 3.7 It is possible to prove that there exist constants $c_1, w > 0$ such that $\|\phi_N - f\| \leq c_1 e^{-wN}$ for all $N \in \mathbb{N}$ uniformly in $[-1, 1]$. Thus, convergence is at least at an exponential rate.

Prof. A. C. Hansen

Mathematical Tripos Part II: Michaelmas Term 2024

Numerical Analysis – Lecture 12

Method 3.8 (The algebra of Fourier expansions) *Let $\mathcal{A}$ be the set of all functions $f : [-1,1] \to C$, which are analytic in $[-1,1]$, periodic with period 2, and that can be extended analytically into the complex plane.* Then $\mathcal{A}$ is a linear space, i.e., $f, g \in \mathcal{A}$ and $\alpha \in \mathbb{C}$ then $f+g \in \mathcal{A}$ and $af \in \mathcal{A}$. In particular, with f and g expressed in its Fourier series, i.e.,

$$f(x) = \sum_{n=-\infty}^{\infty} \widehat{f}_n e^{i\pi nx}, \quad g(x) = \sum_{n=-\infty}^{\infty} \widehat{g}_n e^{i\pi nx}$$

we have

$$f(x) + g(x) = \sum_{n=-\infty}^{\infty} (\widehat{f}_n + \widehat{g}_n) e^{i\pi nx}, \quad \alpha f(x) = \sum_{n=-\infty}^{\infty} \alpha \widehat{f}_n e^{i\pi nx} \tag{3.3}$$

and

$$f(x) \cdot g(x) = \sum_{n=-\infty}^{\infty} \left(\sum_{m=-\infty}^{\infty} \widehat{f}_{n-m} \widehat{g}_m \right) e^{i\pi nx} = \sum_{n=-\infty}^{\infty} \left(\widehat{f} * \widehat{g}\right)_n e^{i\pi nx}, \tag{3.4}$$

where $*$ denotes the convolution operator, hence $\widehat{(f \cdot g)}_n = (\widehat{f} * \widehat{g})_n$. Moreover, if $f \in \mathcal{A}$ then $f' \in \mathcal{A}$ and

$$f'(x) = i\pi \sum_{n=-\infty}^{\infty} n \cdot \widehat{f}_n e^{i\pi nx}. \tag{3.5}$$

Since $\{\widehat{f}_n\}$ decays faster than $\mathcal{O}(n^{-p})$ for any $p \in \mathbb{N}$, this provides that all derivatives of f have rapidly convergent Fourier expansions.

Example 3.9 (Application to differential equations) Consider the two-point boundary value problem: $y = y(x)$, $-1 \le x \le 1$, solves

$$y'' + a(x)y' + b(x)y = f(x), \quad y(-1) = y(1), \tag{3.6}$$

where $a, b, f \in \mathcal{A}$ and we seek a *periodic solution* $y \in \mathcal{A}$ for (3.6). Substituting y, a, b and f by their Fourier series and using (3.3)-(3.5) we obtain an infinite dimensional system of linear equations for the Fourier coefficients $\widehat{y}_n$:

$$-\pi^2 n^2 \widehat{y}_n + i\pi \sum_{m=-\infty}^{\infty} m \widehat{a}_{n-m} \widehat{y}_m + \sum_{m=-\infty}^{\infty} \widehat{b}_{n-m} \widehat{y}_m = \widehat{f}_n, \quad n \in \mathbb{Z}. \tag{3.7}$$

Since $a, b, f \in \mathcal{A}$, their Fourier coefficients decrease rapidly, like $\mathcal{O}(n^{-p})$ for every $p \in \mathbb{N}$. Hence, we can truncate (3.7) into the N-dimensional system

$$-\pi^2 n^2 \widehat{y}_n + i\pi \sum_{m=-N/2+1}^{N/2} m \widehat{a}_{n-m} \widehat{y}_m + \sum_{m=-N/2+1}^{N/2} \widehat{b}_{n-m} \widehat{y}_m = \widehat{f}_n, \qquad n = -N/2+1, \ldots, N/2. \tag{3.8}$$

Remark 3.10 The matrix of (3.8) is in general dense, but our theory predicts that fairly small values of N, hence very small matrices, are sufficient for high accuracy. For instance: choosing $a(x) = f(x) = \cos \pi x$, $b(x) = \sin 2\pi x$ (which incidentally even leads to a sparse matrix) we get

$N = 16$	error of size 10^{-10}
$N = 22$	error of size 10^{-15} (which is already hitting the accuracy of computer arithmetic)

Method 3.11 (Computation of Fourier coefficients (DFT)) We have to compute

$$\widehat{f}_n = \frac{1}{2}\int_{-1}^{1} f(t)e^{-i\pi nt}\,dt, \quad n \in \mathbb{Z}. \tag{3.9}$$

For this, suppose we wish to compute the integral on $[-1,1]$ of a function $h \in \mathcal{A}$ by means of the Riemann sums on the uniform partition

$$\int_{-1}^{1} h(t)\,dt \approx \frac{2}{N}\sum_{k=-N/2+1}^{N/2} h\left(\frac{2k}{N}\right). \tag{3.10}$$

This is known as a *rectangle rule*. We want to know how good this approximation is. As in Definition 1.18, let $\omega_N = e^{2\pi i/N}$. Then we have

$$\frac{2}{N}\sum_{k=-N/2+1}^{N/2} h\left(\frac{2k}{N}\right) = \frac{2}{N}\sum_{k=-N/2+1}^{N/2}\sum_{n=-\infty}^{\infty}\widehat{h}_n e^{2\pi ink/N} = \frac{2}{N}\sum_{n=-\infty}^{\infty}\widehat{h}_n\sum_{k=-N/2+1}^{N/2}\omega_N^{nk}.$$

Since $\omega_N^N = 1$ we have

$$\sum_{k=-N/2+1}^{N/2}\omega_N^{nk} = \omega_N^{-n(N/2-1)}\sum_{k=0}^{N-1}\omega_N^{nk} = \begin{cases} N, & n \equiv 0 \pmod N,\\ 0, & n \not\equiv 0 \pmod N,\end{cases}$$

and we deduce that

$$\frac{2}{N}\sum_{k=-N/2+1}^{N/2} h\left(\frac{2k}{N}\right) = 2\sum_{r=-\infty}^{\infty}\widehat{h}_{Nr}.$$

Hence, the error committed by the Riemann approximation is

$$\begin{aligned} e_N(h) &:= \frac{2}{N}\sum_{k=-N/2+1}^{N/2} h\left(\frac{2k}{N}\right) - \int_{-1}^{1} h(t)\,dt = 2\sum_{r=-\infty}^{\infty}\widehat{h}_{Nr} - 2\widehat{h}_0\\ &= 2\sum_{r=1}^{\infty}\left(\widehat{h}_{Nr} + \widehat{h}_{-Nr}\right).\end{aligned}$$

Since $h \in \mathcal{A}$, its Fourier coefficients decay at spectral rate, namely $\widehat{h}_{Nr} = \mathcal{O}((Nr)^{-p})$, and hence the error of the Riemann sums approximation (3.10) decays spectrally as a function of N,

$$e_N(h) = \mathcal{O}(N^{-p}) \quad \forall p \in \mathbb{N}.$$

Going back to the computation of the Fourier coefficients (3.9), we see that we may compute the integral of $h(x) = \frac{1}{2}f(x)e^{-i\pi nx}$ by means of the Riemann sums, and this gives a spectral method for calculating the Fourier coefficients of f:

$$\widehat{f}_n \approx \frac{1}{N}\sum_{k=-N/2+1}^{N/2} f\left(\frac{2k}{N}\right)\omega_N^{-nk}, \qquad n = -N/2+1,\dots,N/2. \tag{3.11}$$

Remark 3.12 One can recognise that formula (3.11) is the *discrete Fourier transform (DFT)* of the sequence $(y_k) = \left(f(\frac{2k}{N})\right)$, see previous definition, hence not only have we a spectral rate of convergence, but also a fast algorithm (FFT) of computing the Fourier coefficients.

Revision 3.13 (The fast Fourier transform (FFT)) The *fast Fourier transform (FFT)* is a computational algorithm, which computes the leading N Fourier coefficients of a function in just $\mathcal{O}(N\log_2 N)$ operations (cf. Algorithm 1.19). We assume that N is a power of 2, i.e. $N = 2m = 2^p$, and for $\boldsymbol{y} \in \Pi_{2m}$, denote by

$$\boldsymbol{y}^{(\mathrm{E})} = \{y_{2j}\}_{j\in\mathbb{Z}} \qquad \text{and} \qquad \boldsymbol{y}^{(\mathrm{O})} = \{y_{2j+1}\}_{j\in\mathbb{Z}}$$

the even and odd portions of $\boldsymbol{y}$, respectively. Note that $\boldsymbol{y}^{(\mathrm{E})}, \boldsymbol{y}^{(\mathrm{O})} \in \Pi_m$. To execute FFT, we start from vectors of unit length and in each s-th stage, $s = 1...p$, assemble 2^{p-s} vectors of length 2^s from vectors of length 2^{s-1} with

$$x_\ell = x_\ell^{(\mathrm{E})} + \omega_{2^s}^\ell x_\ell^{(\mathrm{O})}, \qquad \ell = 0, \ldots, 2^{s-1}-1. \tag{3.12}$$

Therefore, it costs just s products to evaluate the first half of $\boldsymbol{x}$, provided that $\boldsymbol{x}^{(\mathrm{E})}$ and $\boldsymbol{x}^{(\mathrm{O})}$ are known. It actually costs nothing to evaluate the second half, since

$$x_{2^{s-1}+\ell} = x_\ell^{(\mathrm{E})} - \omega_{2^s}^\ell x_\ell^{(\mathrm{O})}, \qquad \ell = 0, \ldots, 2^{s-1}-1.$$

Altogether, the cost of FFT is $p2^{p-1} = \frac{1}{2} N \log_2 N$ products.

Prof. A. C. Hansen

Mathematical Tripos Part II: Michaelmas Term 2024

Numerical Analysis – Lecture 13

Problem 3.13 (The Poisson equation) We consider the *Poisson equation*

$$\nabla^2 u = f, \quad -1 \leq x, y \leq 1, \tag{3.11}$$

where f is analytic and obeys the periodic boundary conditions

$$f(-1, y) = f(1, y), \quad -1 \leq y \leq 1, \qquad f(x, -1) = f(x, 1), \quad -1 \leq x \leq 1.$$

Moreover, we add to (3.11) the following *periodic boundary conditions*

$$\begin{aligned} &u(-1, y) = u(1, y), \quad u_x(-1, y) = u_x(1, y), \quad -1 \leq y \leq 1 \\ &u(x, -1) = u(x, 1), \quad u_y(x, -1) = u_y(x, 1), \quad -1 \leq x \leq 1. \end{aligned} \tag{3.12}$$

With these boundary conditions alone, a solution of (3.11) is only defined up to an additive constant. Hence, we add a *normalisation condition* to fix the constant:

$$\int_{-1}^{1} \int_{-1}^{1} u(x, y)\, dx\, dy = 0. \tag{3.13}$$

We have the spectrally convergent Fourier expansion

$$f(x, y) = \sum_{k,l=-\infty}^{\infty} \widehat{f}_{k,\ell} e^{i\pi(kx+\ell y)}$$

and seek the Fourier expansion of u

$$u(x, y) = \sum_{k,\ell=-\infty}^{\infty} \widehat{u}_{k,\ell} e^{i\pi(kx+\ell y)}.$$

Since

$$0 = \int_{-1}^{1} \int_{-1}^{1} u(x, y)\, dx\, dy = \sum_{k,\ell=-\infty}^{\infty} \widehat{u}_{k,\ell} \int_{-1}^{1} \int_{-1}^{1} e^{i\pi(kx+\ell y)}\, dx\, dy = \widehat{u}_{0,0},$$

and

$$\nabla^2 u(x, y) = -\pi^2 \sum_{k,\ell=-\infty}^{\infty} (k^2 + \ell^2) \widehat{u}_{k,\ell} e^{i\pi(kx+\ell y)},$$

together with (3.11), we have

$$\begin{cases} \widehat{u}_{k,\ell} = -\dfrac{1}{(k^2 + \ell^2)\pi^2} \widehat{f}_{k,\ell}, \quad k, \ell \in \mathbb{Z},\ (k, \ell) \neq (0, 0) \\ \widehat{u}_{0,0} = 0. \end{cases}$$

Remark 3.14 Applying a spectral method to the Poisson equation is not representative for its application to other PDEs. The reason is the special structure of the Poisson equation. In fact, $\phi_{k,\ell} = e^{i\pi(kx+\ell y)}$ are the eigenfunctions of the Laplace operator with

$$\nabla^2 \phi_{k,\ell} = -\pi^2 (k^2 + \ell^2) \phi_{k,\ell},$$

and they obey periodic boundary conditions.

Problem 3.15 (General second-order linear elliptic PDE) We consider the more general second-order linear elliptic PDE

$$\nabla^\top (a\nabla u) = f, \quad -1 \le x, y \le 1,$$

with $a(x,y) > 0$, and a and f periodic. We again impose the periodic boundary conditions (3.12) and the normalisation condition (3.13). We rewrite

$$\nabla^\top (a\nabla u) = \frac{\partial}{\partial x}(au_x) + \frac{\partial}{\partial y}(au_y) = f\,,$$

and use the Fourier expansions

$$g(x,y) = \sum_{k,\ell\in\mathbb{Z}} \widehat{g}_{k,\ell}\phi_{k,\ell}(x,y), \qquad h(x,y) = \sum_{m,n\in\mathbb{Z}} \widehat{h}_{m,n}\phi_{m,n}(x,y),$$

together with the bivariate versions of (3.4)-(3.5)

$$\widehat{(g\cdot h)}_{k,\ell} = \sum_{m,n\in\mathbb{Z}} \widehat{g}_{k-m,\ell-n}\widehat{h}_{m,n}, \qquad \widehat{(g_x)}_{k,\ell} = i\pi k\,\widehat{g}_{k,\ell}\,, \qquad \widehat{(g_y)}_{k,\ell} = i\pi \ell\,\widehat{g}_{k,\ell}\,,$$

$$\widehat{(h_x)}_{m,n} = i\pi m\,\widehat{h}_{m,n}\,, \qquad \widehat{(h_y)}_{m,n} = i\pi n\,\widehat{h}_{m,n}\,.$$

This gives

$$-\pi^2 \sum_{k,\ell\in\mathbb{Z}}\sum_{m,n\in\mathbb{Z}} (km+\ell n)\,\widehat{a}_{k-m,\ell-n}\widehat{u}_{m,n}\phi_{k,\ell}(x,y) = \sum_{k,\ell\in\mathbb{Z}} \widehat{f}_{k,\ell}\phi_{k,\ell}(x,y)\,.$$

In the next steps, we truncate the expansions to $-N/2+1 \le k,\ell,m,n \le N/2$ and impose the normalisation condition $\widehat{u}_{0,0} = 0$. This results in a system of N^2-1 linear algebraic equations in the unknowns $\widehat{u}_{m,n}$, where $m,n = -N/2+1 ... N/2$, and $(m,n) \neq (0,0)$:

$$\sum_{m,n=-N/2+1}^{N/2} (km+\ell n)\,\widehat{a}_{k-m,\ell-n}\,\widehat{u}_{m,n} = -\frac{1}{\pi^2}\,\widehat{f}_{k,\ell}\,, \qquad k,\ell = -N/2+1 ... N/2\,.$$

Discussion 3.16 (Analyticity and periodicity) The fast convergence of spectral methods rests on two properties of the underlying problem: analyticity and periodicity. If one is not satisfied the rate of convergence in general drops to polynomial. However, to a certain extent, we can relax these two assumptions while still retaining the substantive advantages of Fourier expansions.

- *Relaxing analyticity*: In general, the speed of convergence of the truncated Fourier series of a function f depends on the smoothness of the function. In fact, the smoother the function the faster the truncated series converges, i.e., for $f \in C^p(-1,1)$ we receive an $\mathcal{O}(N^{-p})$ order of convergence.

  Spectral convergence can be recovered, once analyticity is replaced by the requirement that $f \in C^\infty(-1,1)$, i.e., $f^{(m)}(x)$ exists for all $x \in (-1,1)$ and $m = 0,1,2,\ldots$. Consider, for instance, $f(x) = e^{-1/(1-x^2)}$. Then, $f \in C^\infty(-1,1)$ but cannot be extended analytically because of essential singularities at ± 1. Nevertheless, one can show that $|\widehat{f}_n| \sim \mathcal{O}(e^{-cn^\alpha})$, where $c > 0$ and $\alpha \approx 0.44$. While this is slower than exponential convergence in the analytic case (cf. Remark 3.7), it is still faster than $\mathcal{O}(n^{-m})$ for any integer m and hence, we have spectral convergence.

- *Relaxing periodicity*: Disappointingly, periodicity is necessary for spectral convergence. Once this condition is dropped, we are back to the setting of Theorem 3.3, i.e., Fourier series converge as $\mathcal{O}(N^{-1})$ unless $f(-1) = f(1)$. One way around this is to change our set of basis functions, e.g., to Chebyshev polynomials.

Prof. A. C. Hansen

Mathematical Tripos Part II: Michaelmas Term 2024

Numerical Analysis – Lecture 14

Revision 3.17 (Chebyshev polynomials) The Chebyshev polynomial of degree n is defined as

$$T_n(x) := \cos n \arccos x, \quad x \in [-1, 1],$$

or, in a more instructive form,

$$T_n(x) := \cos n\theta, \quad x = \cos\theta, \quad \theta \in [0, \pi]. \tag{3.14}$$

1) The sequence (T_n) obeys the three-term recurrence relation

$$\begin{aligned} &T_0(x) \equiv 1, \quad T_1(x) = x, \\ &T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \quad n \geq 1, \end{aligned}$$

in particular, T_n is indeed an algebraic polynomial of degree n, with the leading coefficient 2^{n-1}. (The recurrence is due to the equality $\cos(n+1)\theta + \cos(n-1)\theta = 2\cos\theta\cos n\theta$ via substitution $x = \cos\theta$, expressions for T_0 and T_1 are straightforward.)

2) Also, (T_n) form a sequence of orthogonal polynomials with respect to the inner product $(f, g)_w := \int_{-1}^{1} f(x)g(x)w(x)dx$, with the weight function $w(x) := (1 - x^2)^{-1/2}$. Namely, we have

$$(T_n, T_m)_w = \int_{-1}^{1} T_m(x)T_n(x)\,\frac{dx}{\sqrt{1-x^2}} = \int_0^{\pi} \cos m\theta \cos n\theta\, d\theta = \begin{cases} \pi, & m = n = 0\,, \\ \frac{\pi}{2}, & m = n \geq 1\,, \\ 0, & m \neq n\,. \end{cases} \tag{3.15}$$

Method 3.18 (Chebyshev expansion) Since $(T_n)_{n=0}^{\infty}$ form an orthogonal sequence, a function f such that $\int_{-1}^{1} |f(x)|^2 w(x)\,dx < \infty$ can be expanded in the series

$$f(x) = \sum_{n=0}^{\infty} \breve{f}_n T_n(x),$$

with the Chebyshev coefficients $\breve{f}_n$. Making inner product of both sides with T_n and using orthogonality yields

$$(f, T_n)_w = \breve{f}_n (T_n, T_n)_w \quad \Rightarrow \quad \breve{f}_n = \frac{(f, T_n)_w}{(T_n, T_n)_w} = \frac{c_n}{\pi} \int_{-1}^{1} f(x)T_n(x)\,\frac{dx}{\sqrt{1-x^2}}\,, \tag{3.16}$$

where $c_0 = 1$ and $c_n = 2$ for $n \geq 1$.

Connection to the Fourier expansion. Letting $x = \cos\theta$ and $g(\theta) = f(\cos\theta)$, we obtain

$$\int_{-1}^{1} f(x)T_n(x)\,\frac{dx}{\sqrt{1-x^2}} = \int_0^{\pi} f(\cos\theta)T_n(\cos\theta)\,d\theta = \frac{1}{2}\int_{-\pi}^{\pi} g(\theta)\cos n\theta\,d\theta\,. \tag{3.17}$$

Given that $\cos n\theta = \frac{1}{2}(e^{in\theta} + e^{-in\theta})$, and using the Fourier expansion of the 2π-periodic function g,

$$g(\theta) = \sum_{n\in\mathbb{Z}} \widehat{g}_n \mathrm{e}^{in\theta}, \quad \text{where} \quad \widehat{g}_n = \frac{1}{2\pi}\int_{-\pi}^{\pi} g(t)e^{-int}\,dt, \quad n \in \mathbb{Z}\,,$$

we continue (3.17) as

$$\int_{-1}^{1} f(x)T_n(x)\frac{dx}{\sqrt{1-x^2}} = \frac{\pi}{2}(\widehat{g}_{-n} + \widehat{g}_n)\,,$$

and from (3.16) we deduce that

$$\breve{f}_n = \begin{cases} \widehat{g}_0, & n = 0\,, \\ \widehat{g}_{-n} + \widehat{g}_n, & n \geq 1\,. \end{cases}$$

Discussion 3.19 (Properties of the Chebyshev expansion) As we have seen, for a general integrable function f, the computation of its Chebyshev expansion is equivalent to the Fourier expansion of the function $g(\theta) = f(\cos\theta)$. Since the latter is periodic with period 2π, we can use a discrete Fourier transform (DFT) to compute the Chebyshev coefficients $\breve{f}_n$. [Actually, based on this connection, one can perform a direct fast Chebyshev transform].

Also, if f can be analytically extended from $[-1,1]$ (to the so-called Bernstein ellipse), then $\breve{f}_n$ decays spectrally fast for $n \gg 1$ (with the rate depending on the size of the ellipse). Hence, the Chebyshev expansion inherits the rapid convergence of spectral methods without assuming that f is periodic.

Method 3.20 (The algebra of Chebyshev expansions) Let $\mathcal{B}$ be the set of analytic functions in $[-1,1]$ that can be extended analytically into the complex plane. We identify each such function with its Chebyshev expansion. Like the set $\mathcal{A}$, the set $\mathcal{B}$ is a linear space and is closed under multiplication. In particular, we have

$$\begin{aligned} T_m(x)T_n(x) &= \cos(m\theta)\cos(n\theta) \\ &= \tfrac{1}{2}\left[\cos((m-n)\theta) + \cos((m+n)\theta)\right] \\ &= \tfrac{1}{2}\left[T_{|m-n|}(x) + T_{m+n}(x)\right] \end{aligned}$$

and hence,

$$\begin{aligned} f(x)g(x) &= \sum_{m=0}^{\infty} \breve{f}_m T_m(x) \cdot \sum_{n=0}^{\infty} \breve{g}_n T_n(x) = \frac{1}{2}\sum_{m,n=0}^{\infty} \breve{f}_m \breve{g}_n \left[T_{|m-n|}(x) + T_{m+n}(x)\right] \\ &= \frac{1}{2}\sum_{m,n=0}^{\infty} \breve{f}_m \Big(\breve{g}_{|m-n|} + \breve{g}_{m+n}\Big) T_n(x). \end{aligned}$$

Lemma 3.21 (Derivatives of Chebyshev polynomials) *We can express derivatives T_n' in terms of (T_k) as follows,*

$$T_{2n}'(x) = (2n)\cdot 2\sum_{k=1}^{n} T_{2k-1}(x), \tag{3.18}$$

$$T_{2n+1}'(x) = (2n+1)\big[T_0(x) + 2\sum_{k=1}^{n} T_{2k}(x)\big]. \tag{3.19}$$

Proof. From (3.14), we deduce

$$T_m(x) = \cos m\theta \quad \Rightarrow \quad T_m'(x) = \frac{m\sin m\theta}{\sin\theta} \qquad x = \cos\theta\,.$$

So, for $m = 2n$, (3.18) follows from the identity $\frac{\sin 2n\theta}{\sin\theta} = 2\sum_{k=1}^{n}\cos(2k-1)\theta$, which is verified as

$$2\sin\theta\sum_{k=1}^{n}\cos(2k-1)\theta = \sum_{k=1}^{n} 2\cos(2k-1)\theta\sin\theta = \sum_{k=1}^{n}\big[\sin 2k\theta - \sin(2k-1)\theta\big] = \sin 2n\theta.$$

For $m = 2n+1$, (3.19) turns into identity $\frac{\sin(2n+1)\theta}{\sin\theta} = 1 + 2\sum_{k=1}^{n}\cos 2k\theta$, and that follows from

$$\sin\theta\Big(1 + 2\sum_{k=1}^{n}\cos 2k\theta\Big) = \sin\theta + \sum_{k=1}^{n}\big[\sin(2k+1)\theta - \sin(2k-1)\theta\big] = \sin(2n+1)\theta.$$

□

Remark 3.22 (Application to PDEs) With Lemma 3.21 all derivatives of u can be expressed in an explicit form as a Chebyshev expansion (cf. Exercise 19 on Example Sheets). For the computation of the Chebyshev coefficients the function f has to be sampled at the so-called Chebyshev points $\cos(2\pi k/N)$, $k = -N/2+1,\ldots,N/2$. This results into a grid, which is denser towards the edges. For elliptic problems this is not problematic, however for initial value PDEs such grids can cause numerical instabilities.

Mathematical Tripos Part II: Michaelmas Term 2024

Numerical Analysis – Lecture 15

Remark 3.23 (Chebyshev expansion for the derivatives) For an analytic function u, the coefficients $\breve{u}_n^{(k)}$ of the Chebyshev expansion for its derivatives are given by the following recursion,

$$\breve{u}_n^{(k)} = c_n \sum_{\substack{m=n+1 \\ n+m \text{ odd}}}^{\infty} m\, \breve{u}_m^{(k-1)}, \quad \forall\, k \geq 1,$$

where $c_0 = 1$ and $c_n = 2$ for $n \geq 1$. This can be derived from Lemma 3.21 (the case $m = 1$ is the topic of Ex. 19 on the Example Sheets).

Method 3.24 (The spectral method for evolutionary PDEs) We consider the problem

$$\begin{cases} \dfrac{\partial u(x,t)}{\partial t} = \mathcal{L}u(x,t), & x \in [-1,1], \quad t \geq 0\,, \\ u(x,0) = g(x), & x \in [-1,1], \end{cases} \tag{3.20}$$

with appropriate boundary conditions on $\{-1,1\} \times \mathbb{R}_+$ and where $\mathcal{L}$ is a linear operator (acting on x), e.g., a differential operator. We want to solve this problem by the method of lines (semi-discretization), using a spectral method for the approximation of u and its derivatives in the spatial variable x. Then, in a general spectral method, we seek solutions $u_N(x,t)$ with

$$u_N(x,t) = \sum_{\#\{n\}=N} c_n(t)\, \varphi_n(x), \tag{3.21}$$

where $c_n(t)$ are expansion coefficients and φ_n are basis functions chosen according to the specific structure of (3.20). For example, we may take

1) the *Fourier expansion* with $c_n(t) = \widehat{u}_n(t)$, $\varphi_n(x) = e^{i\pi n x}$ for periodic boundary conditions,

2) a polynomial expansion such as the *Chebyshev expansion* with $c_n(t) = \breve{u}_n(t)$, $\varphi_n(x) = T_n(x)$ for other boundary conditions.

The spectral approximation in space (3.21) results into a $N \times N$ system of ODEs for the expansion coefficients $\{c_n(t)\}$:

$$\boldsymbol{c}' = B\boldsymbol{c}\,, \tag{3.22}$$

where $B \in \mathbb{R}^{N\times N}$, and $\boldsymbol{c} = \{c_n(t)\} \in \mathbb{R}^N$. We can solve it with standard ODE solvers (Euler, Crank-Nikolson, etc.) which as we have seen are approximations to the matrix exponent in the exact solution $\boldsymbol{c}(t) = \mathrm{e}^{tB}\boldsymbol{c}(0)$.

Example 3.25 (The diffusion equation) Consider the diffusion equation for a function $u = u(x,t)$,

$$\begin{cases} u_t = u_{xx}, & (x,t) \in [-1,1] \times \mathbb{R}_+\,, \\ u(x,0) = g(x), & x \in [-1,1]\,. \end{cases} \tag{3.23}$$

with the periodic boundary conditions $u(-1,t) = u(1,t)$, $u_x(-1,t) = u_x(1,t)$, and standard normalisation $\int_{-1}^{1} u(x,t)\,dx = 0$, both imposed for all values $t \geq 0$.

For each t, we approximate $u(x,t)$ by its N-th order partial Fourier sum in x,

$$u(x,t) \approx u_N(x,t) \;= \sum_{n\in\Gamma_N} \widehat{u}_n(t)\, e^{i\pi n x}\,, \qquad \Gamma_N := \{-N/2+1, ..., N/2\}\,.$$

Then, from (3.23), we see that each coefficient $\widehat{u}_n$ fulfills the ODE

$$\widehat{u}_n'(t) = -\pi^2 n^2 \widehat{u}_n(t)\,. \qquad n \in \Gamma_N \tag{3.24}$$

Its exact solution is $\widehat{u}_n(t) = \mathrm{e}^{-\pi^2 n^2 t}\,\widehat{g}_n$ for $n \neq 0$ and we set $\widehat{u}_0(t) = 0$ due to the normalisation condition, so that

$$u_N(x,t) = \sum_{n\in\Gamma_N} \widehat{g}_n\, \mathrm{e}^{-\pi^2 n^2 t}\, \mathrm{e}^{i\pi n x}\,,$$

which is the exact solution truncated to N terms.

Here, we were able to find the exact solution without solving ODE numerically due to the special structure of the Laplacian. However, for more general PDE we will need a numerical method, and thus the issue of stability arises, so we consider this issue on that simplified example.

Analysis 3.26 (Stability analysis) The system (3.24) has the form

$$\widehat{\boldsymbol{u}}' = B\widehat{\boldsymbol{u}}\,, \qquad B = \operatorname{diag}\{-\pi^2 n^2\}\,, \quad n \in \Gamma_N\,,$$

and we note that (a) all the eigenvalues of B are negative, and that (b) they consist of the eigenvalues $\lambda_n^{(2)}$ of the second order differentiation operator, with $\max|\lambda_n^{(2)}| = (\frac{N}{2})^2$.

If we approximate this system with the Euler method:

$$\widehat{\boldsymbol{u}}^{k+1} = (I + \tau B)\widehat{\boldsymbol{u}}^k, \qquad \tau := \Delta t,$$

then we see that, for stability condition $\|I+\tau B\| \le 1$, we need to scale teh time step $\tau = \Delta t \sim N^{-2}$.

Note that, for the Crank-Nikolson scheme, since the spectrum of B is negative, we get stability for any time step $\tau > 0$.

For general linear operator $\mathcal{L}$ in (3.20) with constant coefficients, the matix B is again diagonal (hence normal), and provided that it spectrum is negative, for stability we must scale the time step $\tau \sim N^{-m}$, where m is the maximal order of differentiation.

The scaling $\tau \sim N^{-2}$ may seem similar to the scaling $k \sim h^2$ in difference methods which we viewed as a disadvantage, however in spectral methods we can take N, the order of partial Fourier or Chebyshev sums to achieve a good appoximation, rather small. (We may still need to choose τ small enough to get a desired accuracy.)

Example 3.27 (The diffusion equation with non-constant coefficient) We want to solve the diffusion equation with a non-constant coefficient $a(x) > 0$ for a function $u = u(x,t)$

$$\begin{cases} u_t = (a(x)u_x)_x, & (x,t) \in [-1,1] \times \mathbb{R}_+\,, \\ u(x,0) = g(x), & x \in [-1,1]\,, \end{cases} \tag{3.25}$$

with boundary and normalization conditions as before. Approximating u by its partial Fourier sum results in the following system of ODEs for the coefficients $\widehat{u}_n$

$$\widehat{u}_n'(t) = -\pi^2 \sum_{m \in \Gamma_N} mn\, \widehat{a}_{n-m}\, \widehat{u}_m(t), \qquad n \in \Gamma_N\,.$$

For the discretization in time we may apply the Euler method, this gives

$$\widehat{u}_n^{k+1} = \widehat{u}_n^k - \tau\, \pi^2 \sum_{m \in \Gamma_N} mn\, \widehat{a}_{n-m}\, \widehat{u}_m^k\,, \qquad \tau = \Delta t\,,$$

or in the vector form

$$\widehat{\boldsymbol{u}}^{k+1} = (I + \tau B)\widehat{\boldsymbol{u}}^k,$$

where $B = (b_{m,n}) = (-\pi^2 mn\, \widehat{a}_{n-m})$. For stability of Euler method, we again need $\|I + \tau B\| \le 1$, but analysis here is less straightforward.

Remark 3.28 (Chebyshev methods for evolutionary problems) In general, the boundary conditions for the considered PDEs have to be implemented in the Chebyshev expansion. If the boundary conditions are to be imposed exactly, either the basis functions have to be slightly modified, e.g., to $T_n(x) - 1$ instead of $T_n(x)$ for the boundary condition $u(1) = 0$, or we get additional conditions on the expansion coefficients $\breve{u}_n$ (cf. Exercise 20 from the Example Sheets). While the exact imposition is in general not a problem for the numerical treatment of elliptic PDEs, as soon as the boundary conditions depend on time we may run into serious stability issues. One way around this is the use of penalty methods in which the boundary conditions is added to the scheme later as a penalty term.

Prof. A. C. Hansen

Mathematical Tripos Part II: Michaelmas Term 2024

Numerical Analysis – Lecture 16

4 Iterative methods for linear algebraic systems

The general *iterative* method for solving $Ax = b$ is a rule $\boldsymbol{x}^{k+1} = f_k(\boldsymbol{x}^0, \boldsymbol{x}^1, \ldots, \boldsymbol{x}^k)$. We will consider the simplest ones: *linear, one-step, stationary* iterative schemes:

$$\boldsymbol{x}^{k+1} = H\boldsymbol{x}^k + \boldsymbol{v}, \qquad \boldsymbol{x}^0, \boldsymbol{v} \in \mathbb{R}^n. \tag{4.1}$$

Here one chooses H and v so that x^*, a solution of $A\boldsymbol{x} = \boldsymbol{b}$, satisfies $\boldsymbol{x}^* = H\boldsymbol{x}^* + \boldsymbol{v}$, i.e. it is the fixed point of the iteration (4.1) (if the scheme converges). Standard terminology:

the *iteration matrix* H, the *error* $\boldsymbol{e}^k := \boldsymbol{x}^* - \boldsymbol{x}^k$, the *residual* $\boldsymbol{r}^k := A\boldsymbol{e}^k = \boldsymbol{b} - A\boldsymbol{x}^k$.

For a given class of matrices A (e.g. positive definite matrices, or even a single particular matrix), we are interested in *convergent* methods, i.e. the methods such that $\boldsymbol{x}^k \to \boldsymbol{x}^* = A^{-1}\boldsymbol{b}$ for every starting value $\boldsymbol{x}^0$. Subtracting $\boldsymbol{x}^* = H\boldsymbol{x}^* + \boldsymbol{v}$ from (4.1) we obtain

$$\boldsymbol{e}^{k+1} = H\boldsymbol{e}^k = \cdots = H^{k+1}\boldsymbol{e}^0, \tag{4.2}$$

i.e., a method is convergent if $\boldsymbol{e}^k = H^k\boldsymbol{e}^0 \to 0$ for any $\boldsymbol{e}^0 \in \mathbb{R}^n$.

Scheme 4.1 (Iterative refinement) This is the scheme

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - S(A\boldsymbol{x}^k - \boldsymbol{b})\,.$$

If $S = A^{-1}$, then $\boldsymbol{x}^{k+1} = A^{-1}\boldsymbol{b} = \boldsymbol{x}^*$, so it is suggestive to choose S as an approximation to A^{-1}. The iteration matrix for this scheme is $H_S = I - SA$.

Scheme 4.2 (Splitting) This is the scheme

$$(A - B)\boldsymbol{x}^{k+1} = -B\boldsymbol{x}^k + \boldsymbol{b}\,,$$

with the iteration matrix $H = -(A - B)^{-1}B$. Any splitting can be viewed as an iterative refinement (and vice versa) because

$$\begin{aligned}(A - B)\boldsymbol{x}^{k+1} = -B\boldsymbol{x}^k + \boldsymbol{b} \quad &\Leftrightarrow \quad (A - B)\boldsymbol{x}^{k+1} = (A - B)\boldsymbol{x}^k - (A\boldsymbol{x}^k - \boldsymbol{b}) \\ &\Leftrightarrow \quad \boldsymbol{x}^{k+1} = \boldsymbol{x}^k - (A - B)^{-1}(A\boldsymbol{x}^k - \boldsymbol{b}),\end{aligned}$$

so we should seek a splitting such that $S = (A - B)^{-1}$ approximates A^{-1}.

Theorem 4.3 *Let $H \in \mathbb{R}^{n\times n}$. Then $\lim_{k\to\infty} H^k\boldsymbol{z} = 0$ for any $\boldsymbol{z} \in \mathbb{R}^n$ if and only if $\rho(H) < 1$.*

Proof. 1) Let λ be an eigenvalue of (the real) H, real or complex, such that $|\lambda| = \rho(H) \geq 1$, and let $\boldsymbol{w}$ be a corresponding eigenvector, i.e., $H\boldsymbol{w} = \lambda\boldsymbol{w}$. Then $H^k\boldsymbol{w} = \lambda^k\boldsymbol{w}$, and

$$\|H^k\boldsymbol{w}\|_\infty = |\lambda|^k\|\boldsymbol{w}\|_\infty \geq \|\boldsymbol{w}\|_\infty =: \gamma > 0. \tag{4.3}$$

If $\boldsymbol{w}$ is real, we choose $\boldsymbol{z} = w$, hence $\|H^k\boldsymbol{z}\|_\infty \geq \gamma$, and this cannot tend to zero.

If $\boldsymbol{w}$ is complex, then $\boldsymbol{w} = \boldsymbol{u} + \mathrm{i}\boldsymbol{v}$ with some real vectors $\boldsymbol{u}, \boldsymbol{v}$. But then at least one of the sequences $(H^k\boldsymbol{u}), (H^k\boldsymbol{v})$ does not tend to zero. For if both do, then also $H^k\boldsymbol{w} = H^k\boldsymbol{u} + \mathrm{i}H^k\boldsymbol{v} \to 0$, and this contradicts (4.3).

2) Now, let $\rho(H) < 1$, and assume for simplicity that H possesses n linearly independent eigenvectors $(\boldsymbol{w}_j)$ such that $H\boldsymbol{w}_j = \lambda_j\boldsymbol{w}_j$. Linear independence means that every $\boldsymbol{z} \in \mathbb{R}^n$ can be expressed as a linear combination of the eigenvectors, i.e., there exist $(c_j) \in \mathbb{C}$ such that $\boldsymbol{z} = \sum_{j=1}^n c_j\boldsymbol{w}_j$. Thus,

$$H^k\boldsymbol{z} = \textstyle\sum_{j=1}^n c_j\lambda_j^k\boldsymbol{w}_j\,,$$

and since $|\lambda_j| \leq \rho(H) < 1$ we have $\lim_{k\to\infty} H^k\boldsymbol{z} = 0$, as required. □

Remark 4.4 The complete proof of case (2) of Theorem 4.3 exploits the so-called Jordan normal form of the matrix H, namely $H = SJS^{-1}$, where J is a block diagonal matrix consisting of the Jordan blocks,

$$J = \begin{bmatrix} \boxed{J_1} & & & \\ & \boxed{J_2} & & \\ & & \ddots & \\ & & & \boxed{J_r} \end{bmatrix}, \qquad J_i = \begin{bmatrix} \lambda_i & 1 & & \\ & \lambda_i & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i \end{bmatrix}, \qquad J_i \in \mathbb{R}^{n_i \times n_i}, \qquad \sum_i n_i = n\,.$$

To prove that $J_i^k \to 0$ if $|\lambda_i| < 1$ one should split $J_i = \lambda_i I + P$, notice that $P^m = 0$ for $m \geq n_i$, and evaluate the terms of expansion $(\lambda_i I + P)^k = \sum_{m=0}^{n_i - 1} \binom{k}{m} \lambda_i^{k-m} P^m$.

Applying Theorem 4.3 to the error estimate (4.2), we arrive at the following statement.

Theorem 4.5 *Let $\boldsymbol{x}^*$, a solution of $A\boldsymbol{x} = \boldsymbol{b}$, satisfy $\boldsymbol{x}^* = H\boldsymbol{x}^* + \boldsymbol{v}$ and we are given the scheme*

$$\boldsymbol{x}^{k+1} = H\boldsymbol{x}^k + \boldsymbol{v}, \qquad \boldsymbol{x}^0, \boldsymbol{v} \in \mathbb{R}^n. \tag{4.4}$$

Then $\boldsymbol{x}^k \to \boldsymbol{x}^*$ for any choice of $\boldsymbol{x}^0$ if and only if $\rho(H) < 1$.

Note: Of course, we would like to know not just convergence but the rate of it. For example, we achieve convergence with

$$H = \begin{bmatrix} 0.99 & 10^6 \\ 0 & 0.99 \end{bmatrix},$$

but it will take quite a long time. We will discuss this topic briefly later on.

Method 4.6 (Jacobi and Gauss–Seidel) Both of these methods are versions of splitting which can be applied to any A with nonzero diagonal elements. We write A as the sum of three matrices $L_0 + D + U_0$: subdiagonal (strictly lower-triangular), diagonal and superdiagonal (strictly upper-triangular) portions of A, respectively.

1) *Jacobi method*. We set $A - B = D$, the diagonal part of A, and we obtain the next iteration by solving the diagonal system

$$D\boldsymbol{x}^{(k+1)} = -(L_0 + U_0)\boldsymbol{x}^{(k)} + \boldsymbol{b}, \qquad H_{\mathrm{J}} = -D^{-1}(L_0 + U_0)\,.$$

2) *Gauss–Seidel method*. We take $A - B = L_0 + D = L$, the lower-triangular part of A, and we generate the sequence $(\boldsymbol{x}^{(k)})$ by solving the triangular system

$$(L_0 + D)\,\boldsymbol{x}^{(k+1)} = -U_0\boldsymbol{x}^{(k)} + \boldsymbol{b}, \qquad H_{\mathrm{GS}} = -(L_0 + D)^{-1}U_0\,.$$

There is no need to invert $(L_0 + D)$, we calculate the components of $\boldsymbol{x}^{(k+1)}$ in sequence by forward substitution:

$$a_{ii}x_i^{(k+1)} = -\sum_{j<i} a_{ij}x_j^{(k+1)} - \sum_{j>i} a_{ij}x_j^{(k)} + b_i, \qquad i = 1..n.$$

As we mentioned above, the sequence $\boldsymbol{x}^{(k)}$ converges to solution of $A\boldsymbol{x} = \boldsymbol{b}$ if the spectral radius of the iteration matrix, $H_{\mathrm{J}} = -D^{-1}(L_0 + U_0)$ or $H_{\mathrm{GS}} = -(L_0 + D)^{-1}U_0$, respectively, is less than one. Our next goal is to prove that this is the case for two important classes of matrices A:

a) diagonally dominant and b) positive definite matrices.

We start with recalling the simple, but very useful Gershgorin theorem.

Revision 4.7 (Gershgorin theorem) *All eigenvalues of an $n \times n$ matrix A are contained in the union of the Gershgorin discs in the complex plane:*

$$\sigma(A) \subset \cup_{i=1}^n \Gamma_i\,, \qquad \Gamma_i := \{z \in \mathbb{C} : |z - a_{ii}| \leq r_i\}, \qquad r_i := \sum_{j \neq i} |a_{ij}|\,.$$

Prof. A. C. Hansen

Mathematical Tripos Part II: Michaelmas Term 2024

Numerical Analysis – Lecture 17

Definition 4.10 (Strictly diagonally dominant matrices) A matrix A is called strictly diagonally dominant by rows (resp. by columns) if

$$|a_{ii}| > \sum_{j\neq i} |a_{ij}|, \quad i = 1..n \qquad (\text{resp.} \quad |a_{jj}| > \sum_{i\neq j} |a_{ij}|, \quad j = 1..n\,).$$

From Gershgorin theorem, it follows that strictly diagonally dominant matrices are nonsingular.

Theorem 4.11 *If A is strictly diagonally dominant, then both the Jacobi and the Gauss-Seidel methods converge.*

Proof. For the Gauss-Seidel method, the eigenvalues of the iteration matrix $H_{\rm GS} = -(L_0+D)^{-1}U_0$ satisfy the equation

$$\det[H_{\rm GS} - \lambda I] = \det[-(L_0 + D)^{-1}U_0 - \lambda I] = 0 \quad \Rightarrow \quad \det[A_\lambda] := \det[U_0 + \lambda D + \lambda L_0] = 0$$

It is easy to see that if $A = L_0 + D + U_0$ is strictly diagonally dominant, then for $|\lambda| \geq 1$ the matrix $A_\lambda = \lambda L_0 + \lambda D + U_0$ is strictly diagonally dominant too, hence it is nonsingular, and therefore the equality $\det[A_\lambda] = 0$ is impossible. Thus $|\lambda| < 1$, hence convergence. The proof for the Jacobi method is the same. □

Theorem 4.12 (The Householder–John theorem) *If A and B are real matrices such that both A and $A-B-B^T$ are symmetric positive definite, then the spectral radius of $H = -(A - B)^{-1}B$ is strictly less than one.*

Proof. Let λ be an eigenvalue of H, so $H\boldsymbol{w} = \lambda\boldsymbol{w}$ holds, where $\boldsymbol{w} \neq \mathbf{0}$ is an eigenvector. (Note that both λ and $\boldsymbol{w}$ may have nonzero imaginary parts when H is not symmetric, e.g. in the Gauss–Seidel method.) The definition of H provides equality $-B\boldsymbol{w} = \lambda(A-B)\boldsymbol{w}$, and we note that $\lambda \neq 1$ since otherwise A would be singular (which it is not). Thus, we deduce

$$\overline{\boldsymbol{w}}^T B\boldsymbol{w} = \frac{\lambda}{\lambda - 1}\overline{\boldsymbol{w}}^T A\boldsymbol{w}, \tag{4.3}$$

where the bar means complex conjugation. Moreover, writing $\boldsymbol{w} = \boldsymbol{u}+i\boldsymbol{v}$, where $\boldsymbol{u}$ and $\boldsymbol{v}$ are real, we find (for $C = C^T$) the identity $\overline{\boldsymbol{w}}^T C\boldsymbol{w} = \boldsymbol{u}^T C\boldsymbol{u} + \boldsymbol{v}^T C\boldsymbol{v}$, so symmetric positive definiteness in the assumption implies $\overline{\boldsymbol{w}}^T A\boldsymbol{w} > 0$ and $\overline{\boldsymbol{w}}^T(A - B - B^T)\boldsymbol{w} > 0$. In the latter inequality, we use relation (4.3) and its conjugate transpose to obtain

$$0 < \overline{\boldsymbol{w}}^T A\boldsymbol{w} - \overline{\boldsymbol{w}}^T B\boldsymbol{w} - \overline{\boldsymbol{w}}^T B^T\boldsymbol{w} = \left(1 - \frac{\lambda}{\lambda - 1} - \frac{\overline{\lambda}}{\overline{\lambda} - 1}\right)\overline{\boldsymbol{w}}^T A\boldsymbol{w} = \frac{1 - |\lambda|^2}{|\lambda - 1|^2}\,\overline{\boldsymbol{w}}^T A\boldsymbol{w}.$$

Now $\lambda \neq 1$ implies $|\lambda - 1|^2 > 0$. Hence, recalling that $\overline{\boldsymbol{w}}^T A\boldsymbol{w} > 0$, we see that $1 - |\lambda|^2$ is positive. Therefore $|\lambda| < 1$ occurs for every eigenvalue of H as required. □

Corollary 4.13 1) *If A is symmetric positive definite, then the Gauss-Seidel method converges.*
2) *If both A and $2D-A$ are symmetric positive definite, then the Jacobi method converges.*

Proof. 1) For the Gauss-Seidel method, B is the superdiagonal part of symmetric A, hence $A-B-B^T$ is equal to D, the diagonal part of A, and if A is positive definite, then D is positive definite too (this is the first part of the Exercise 23 from Example Sheets).

2) For the Jacobi method, we have $B = A-D$, and if A is symmetric, then $A-B-B^T = 2D-A$. (The latter matrix is the same as A except that the signs of the off-diagonal elements are reversed.) □

Example 4.14 (Poisson's equation on a square) As we have seen in the previous sections linear systems $A\boldsymbol{x} = \boldsymbol{b}$, where A is a real symmetric positive (negative) definite matrix, frequently occur in numerical methods for solving elliptic partial differential equations. A typical example we already encountered is Poisson's equation on a square where the *five-point formula* approximation yields an $n \times n$ system of linear equations with $n = m^2$ unknowns $u_{p,q}$:

$$u_{p-1,q} + u_{p+1,q} + u_{p,q-1} + u_{p,q+1} - 4u_{p,q} = h^2 f(ph, qh) \tag{4.4}$$

(Note that when p or q is equal to 1 or m, then the values $u_{0,q}$, $u_{p,0}$ or $u_{p,m+1}$, $u_{m+1,q}$ are known boundary values and they should be moved to the right-hand side, thus leaving fewer unknowns on the left.)

For any ordering of the grid points (ph, qh) we have shown in Lemma 1.11 that the matrix A of this linear system is symmetric and negative definite.

Corollary 4.15 *For linear system (4.4), for any ordering of the grid, both Jacobi and Gauss-Seidel methods converge.*

Proof. By Lemma 1.11, A is symmetric and negative definite, hence convergence of Gauss-Seidel. To prove convergence of the Jacobi method, we need negative definiteness of the matrix $2D - A$, and that follows by the same arguments as in Lemma 1.11: recall that the proof operates with the modulus of the off-diagonal elements and does not depend on their sign. □

Method 4.16 (Relaxation) It is often possible to improve the efficiency of the splitting method by *relaxation*. Specifically, instead of letting $(A - B)\boldsymbol{x}^{(k+1)} = -B\boldsymbol{x}^{(k)} + \boldsymbol{b}$, we let

$$(A - B)\widehat{\boldsymbol{x}}^{(k+1)} = -B\boldsymbol{x}^{(k)} + \boldsymbol{b}, \quad \text{and then} \quad \boldsymbol{x}^{(k+1)} = \omega\widehat{\boldsymbol{x}}^{(k+1)} + (1-\omega)\boldsymbol{x}^{(k)} \qquad k = 0, 1, \ldots,$$

where ω is a real constant called the *relaxation parameter*. (Note that $\omega = 1$ corresponds to the standard "unrelaxed" iteration.) Good choice of ω leads to a smaller spectral radius of the iteration matrix (compared with the "unrelaxed" method), and the smaller the spectral radius, the faster the iteration converges. To this end, let us express the relaxation iteration matrix H_ω in terms of $H = -(A - B)^{-1}B$. We have

$$\widehat{\boldsymbol{x}}^{(k+1)} = H\boldsymbol{x}^{(k)} + \boldsymbol{v} \quad \Rightarrow \quad \boldsymbol{x}^{(k+1)} = \omega\widehat{\boldsymbol{x}}^{(k+1)} + (1-\omega)\boldsymbol{x}^{(k)} = \omega H\boldsymbol{x}^{(k)} + (1-\omega)\boldsymbol{x}^{(k)} + \omega\boldsymbol{v}$$

hence

$$H_\omega = \omega H + (1-\omega)I.$$

It follows that the spectra of H_ω and H are related by the rule $\lambda_\omega = \omega\lambda + (1-\omega)$, therefore one may try to choose $\omega \in \mathbb{R}$ to minimize

$$\rho(H_\omega) = \max\{|\omega\lambda + (1-\omega)| : \lambda \in \sigma(H)\}.$$

In general, $\sigma(H)$ is unknown, but often we have some information about it which can be utilized to find a "good" (rather than "best") value of ω. For example, suppose that it is known that $\sigma(H)$ is real and resides in the interval $[\alpha, \beta]$ where $-1 < \alpha < \beta < 1$. In that case we seek ω to minimize

$$\max\{|\omega\lambda + (1-\omega)| : \lambda \in [\alpha, \beta]\}.$$

It is readily seen that, for a fixed $\lambda < 1$, the function $f(\omega) = \omega\lambda + (1-\omega)$ is decreasing, therefore, as ω increases (decreases) from 1 the spectrum of H_ω moves to the left (to the right) of the spectrum of H. It is clear that the optimal location of the spectrum $\sigma(H_\omega)$ (or of the interval $[\alpha_\omega, \beta_\omega]$ that contains $\sigma(H_\omega)$) is the one which is centralized around the origin:

$$-[\omega\alpha + (1-\omega)] = \omega\beta + (1-\omega) \quad \Rightarrow \quad \omega_{\text{opt}} = \tfrac{2}{2-(\alpha+\beta)}, \quad -\alpha_{\omega_{\text{opt}}} = \beta_{\omega_{\text{opt}}} = \tfrac{\beta-\alpha}{2-(\alpha+\beta)}.$$

Mathematical Tripos Part II: Michaelmas Term 2024

Numerical Analysis – Lecture 18

Approach 4.20 (Minimization of quadratic function) The methods we considered so far for solving $A\boldsymbol{x} = \boldsymbol{b}$, namely Jacobi, Gauss-Seidel, and those with relaxation, fit into the scheme

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + c_k \boldsymbol{d}^{(k)} ,$$

where we were aimed at getting $\rho(H) < 1$ for the iteration matix H. Say, for Jacobi with relaxation, we set $c_k = \omega$ and $\boldsymbol{d}^{(k)} = D^{-1}(\boldsymbol{b} - A\boldsymbol{x}^{(k)})$.

For solving $A\boldsymbol{x} = \boldsymbol{b}$ with a (positive definite) matrix $A > 0$, there is a different approach to constructing good iterative methods. It is based on succesive minimization of the quadratic function

$$F(\boldsymbol{x}^{(k)}) := \|\boldsymbol{x}^* - \boldsymbol{x}^{(k)}\|_A^2 = \|\boldsymbol{e}^{(k)}\|_A^2 ,$$

since the minimizer is clearly the exact solution. Here, $\|\boldsymbol{y}\|_A := (A\boldsymbol{y}, \boldsymbol{y})^{1/2} := \sqrt{\boldsymbol{y}^T A \boldsymbol{y}}$ is a Euclidean-type distance which is well-defined for $A > 0$. So, at each step k, we are decreasing the A-distance between $\boldsymbol{x}^{(k)}$ and the exact solution $\boldsymbol{x}^*$. Thus, for a symmetric positive definite $A > 0$, we choose an iterative method that provides the descent condition

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + c_k \boldsymbol{d}^{(k)} \quad \Rightarrow \quad F(\boldsymbol{x}^{(k+1)}) < F(\boldsymbol{x}^{(k)}) . \tag{4.5}$$

An equivalent approach is to minimize the quadratic function

$$F_1(\boldsymbol{x}) = \tfrac{1}{2}\boldsymbol{x}^T A \boldsymbol{x} - \boldsymbol{x}^T \boldsymbol{b} ,$$

which attains its minimum when $\nabla F_1(\boldsymbol{x}) = A\boldsymbol{x} - \boldsymbol{b} = 0$, and which does not involve the unknown $\boldsymbol{x}^*$. It is easy to check that $F_1(\boldsymbol{x}) = \frac{1}{2}F(\boldsymbol{x}) - \frac{1}{2}c$, where $c = \boldsymbol{x}^{*T} A \boldsymbol{x}^*$ is a constant independent of k, hence equivalence.

Example 4.21 Both the Jacobi and the Gauss–Seidel methods satisfy (4.5), precisely

$$(A\boldsymbol{e}^{(k+1)}, e^{(k+1)}) = (A\boldsymbol{e}^{(k)}, \boldsymbol{e}^{(k)}) - (C\boldsymbol{y}^{(k)}, \boldsymbol{y}^{(k)}) < (A\boldsymbol{e}^{(k)}, \boldsymbol{e}^{(k)}) ,$$

$$\begin{aligned} \text{where for Gauss-Seidel:} \quad & C = D > 0, && \boldsymbol{y}^{(k)} := (L_0 + D)^{-1} A\boldsymbol{e}^{(k)}; \\ \text{and for Jacobi:} \quad & C = 2D - A > 0, && \boldsymbol{y}^{(k)} := D^{-1} A\boldsymbol{e}^{(k)}. \end{aligned}$$

Method 4.22 (A-orthogonal projection) Next, we strengthen the descent condition (4.5), namely given $\boldsymbol{x}^{(k)}$ and some $\boldsymbol{d}^{(k)}$ (called a *search direction*), we will seek $\boldsymbol{x}^{(k+1)}$ from the set of vectors on the line $\ell = \{\boldsymbol{x}^{(k)} + \alpha \boldsymbol{d}^{(k)}\}_{\alpha \in \mathbb{R}}$ such that it makes the value of $F(\boldsymbol{x}^{(k+1)})$ not just smaller than $F(\boldsymbol{x}^{(k)})$, but as small as possible (with respect to this set), namely

$$\boldsymbol{x}^{(k+1)} := \arg\min_{\alpha} F(\boldsymbol{x}^{(k)} + \alpha \boldsymbol{d}^{(k)}) . \tag{4.6}$$

Lemma 4.23 *The minimizer in (4.6) is given by the formula*

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{d}^{(k)} , \qquad \alpha_k = \frac{(\boldsymbol{r}^{(k)}, \boldsymbol{d}^{(k)})}{(A\boldsymbol{d}^{(k)}, \boldsymbol{d}^{(k)})} . \tag{4.7}$$

(This choice of α_k is referred to as exact line search.)

Proof. From the definition of F, it follows that in (4.6) we should choose the point $\boldsymbol{x}^{(k+1)} \in \ell$ that minimizes the A-distance between $\boldsymbol{x}^*$ and the points $\boldsymbol{y} \in \ell$. Geometrically, it is clear that the minimum occurs when $\boldsymbol{x}^{(k+1)}$ is the A-orthogonal projection of $\boldsymbol{x}^*$ onto the line $\ell = \{\boldsymbol{x}^{(k)} + \alpha \boldsymbol{d}^{(k)}\}$, i.e., when

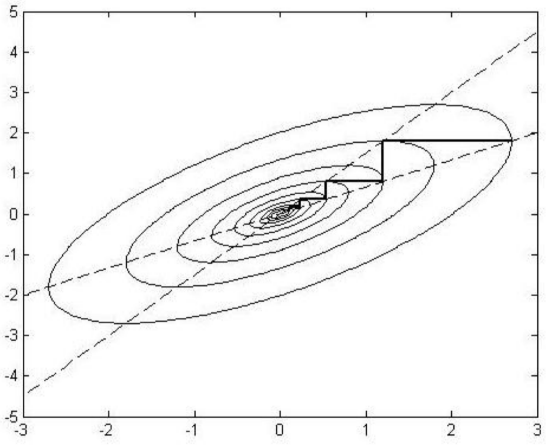
$$\boldsymbol{x}^* - \boldsymbol{x}^{(k+1)} \perp_A \boldsymbol{d}^{(k)} \quad \Rightarrow \quad A(\boldsymbol{x}^* - \boldsymbol{x}^{(k+1)}) \perp \boldsymbol{d}^{(k)} \quad \Rightarrow \quad \boldsymbol{r}^{(k+1)} = \boldsymbol{r}^{(k)} - \alpha_k A\boldsymbol{d}^{(k)} \perp \boldsymbol{d}^{(k)} .$$

This gives expression for α_k in (4.7). □

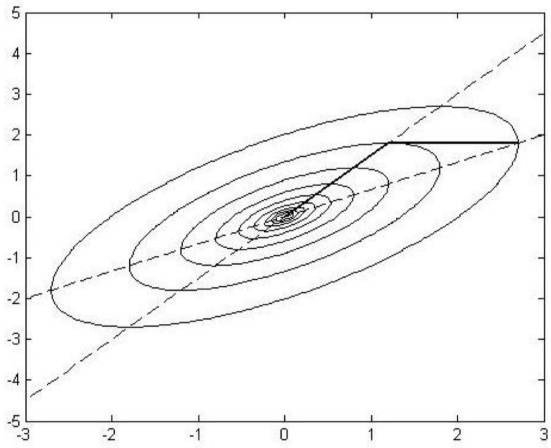
Method 4.24 (The steepest descent method) This method takes $\boldsymbol{d}^{(k)} = -\nabla F_1(\boldsymbol{x}^{(k)}) = \boldsymbol{b} - A\boldsymbol{x}^{(k)}$ for every k, the reason being that, locally, the negative gradient of a quadratic function shows the direction of the (locally) steepest descent at a given point. Thus, the iterations have the form

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_k(\boldsymbol{b} - A\boldsymbol{x}^{(k)}), \qquad k \geq 0\,. \tag{4.8}$$

It can be proved that the sequence $(\boldsymbol{x}^{(k)})$ converges to the solution $\boldsymbol{x}^*$ of the system $A\boldsymbol{x} = \boldsymbol{b}$ as required, but usually the speed of convergence is rather slow. The reason is that the iteration (4.8) decreases the value of $F(\boldsymbol{x}^{(k+1)})$ locally, relatively to $F(\boldsymbol{x}^{(k)})$, but the global decrease, with respect to $F(\boldsymbol{x}^{(0)})$, is often not that large. The use of *conjugate directions* provides a method with a global minimization property.



(a) Worst case scenario of steepest descent



(b) Conjugate gradient method applied to the same problem as in (a)

Conjugate directions Let's revisit equation (4.7) for a general direction $\boldsymbol{d}$ (i.e., not necessarily equal to the negative gradient). Assume $\boldsymbol{x} = \boldsymbol{x}^{(k)}$, and let $\boldsymbol{e}^{(k)} = \boldsymbol{x}^* - \boldsymbol{x}^{(k)}$ be the error and $\boldsymbol{r}^{(k)} = \boldsymbol{b} - A\boldsymbol{x}^{(k)} = A\boldsymbol{e}^{(k)}$ be the residual. Then we can write $\langle \boldsymbol{r}^{(k)}, \boldsymbol{d}\rangle = \langle \boldsymbol{e}^{(k)}, \boldsymbol{d}\rangle_A$, and so for a general search direction $\boldsymbol{d}$ with an exact line search, the iterate takes the form $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \frac{\langle \boldsymbol{e}^{(k)}, \boldsymbol{d}\rangle_A}{\langle \boldsymbol{d}, \boldsymbol{d}\rangle_A}\boldsymbol{d}$. By substracting $\boldsymbol{x}^*$, the iterates in terms of the error $\boldsymbol{e}^{(k+1)}$ are given by:

$$\boldsymbol{e}^{(k+1)} = \boldsymbol{e}^{(k)} - \frac{\langle \boldsymbol{e}^{(k)}, \boldsymbol{d}\rangle_A}{\langle \boldsymbol{d}, \boldsymbol{d}\rangle_A}\boldsymbol{d}. \tag{4.9}$$

Geometrically, this means that $\boldsymbol{e}^{(k+1)}$ is the projection of $\boldsymbol{e}^{(k)}$ onto the hyperplane that is A-orthogonal to $\boldsymbol{d}$, i.e., we have

$$\langle \boldsymbol{e}^{(k+1)}, \boldsymbol{d}\rangle_A = 0. \tag{4.10}$$

Definition 4.25 (Conjugate directions) The vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^n$ are *conjugate* with respect to a symmetric positive definite matrix A if they are nonzero and A-orthogonal: $\langle \boldsymbol{u}, \boldsymbol{v}\rangle_A := \langle \boldsymbol{u}, A\boldsymbol{v}\rangle = 0$.

The observation above allows us to prove the following important result.

Theorem 4.26 *Let $\boldsymbol{d}^{(0)}, \boldsymbol{d}^{(1)}, \ldots, \boldsymbol{d}^{(n-1)}$ be n nonzero pairwise conjugate directions, and consider the sequence of iterates*

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{d}^{(k)}, \qquad \alpha_k = \frac{\langle \boldsymbol{r}^{(k)}, \boldsymbol{d}^{(k)}\rangle}{\langle \boldsymbol{d}^{(k)}, A\boldsymbol{d}^{(k)}\rangle}.$$

Let $\boldsymbol{r}^{(k)} = \boldsymbol{b} - A\boldsymbol{x}^{(k)}$ be the residual. Then for each $k = 1, \ldots, n$, $\boldsymbol{r}^{(k)}$ is orthogonal to $\operatorname{span}\{\boldsymbol{d}^{(0)}, \ldots, \boldsymbol{d}^{(k-1)}\}$. In particular $\boldsymbol{r}^{(n)} = 0$.

Proof. Since $\boldsymbol{r}^{(k)} = A\boldsymbol{e}^{(k)}$, it suffices to show that $\boldsymbol{e}^{(k)}$ is A-orthogonal to $\operatorname{span}\{\boldsymbol{d}^{(0)}, \ldots, \boldsymbol{d}^{(k-1)}\}$. The proof is by induction on k. For $k = 0$ there is nothing to prove. Assume the statement is true for $k \geq 0$, and consider the equation (4.9) (with $\boldsymbol{d} = \boldsymbol{d}^{(k)}$). From the induction hypothesis, and the fact that the $\boldsymbol{d}^{(i)}$ are pairwise conjugate directions, we see that $\boldsymbol{e}^{(k+1)}$ is A-orthogonal to $\boldsymbol{d}^{(0)}, \ldots, \boldsymbol{d}^{(k-1)}$. Furthermore, we have already seen in (4.10) that $\langle \boldsymbol{e}^{(k+1)}, \boldsymbol{d}^{(k)} \rangle_A = 0$. Thus this shows that $\boldsymbol{e}^{(k+1)}$ is A-orthogonal to $\boldsymbol{d}^{(0)}, \ldots, \boldsymbol{d}^{(k)}$ as desired. □

So, if a sequence $(\boldsymbol{d}^{(k)})$ of conjugate directions is at hands, we have an iterative procedure with good approximation properties.

The (A-orthogonal) basis of conjugate directions is constructed by A-orthogonalization of the sequence $\{\boldsymbol{r}_0, A\boldsymbol{r}_0, A^2\boldsymbol{r}_0, ..., A^{n-1}\boldsymbol{r}_0\}$ with $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$. This is done in the way similar to orthogonalization of the monomial sequence $\{1, x, x^2, ..., x^{n-1}\}$ using a recurrence relation.

Remark 4.27 It is possible to extend the methods for solving $A\boldsymbol{x} = \boldsymbol{b}$ with symmetric positive definite A to any other matrices by a simple trick. Suppose we want to solve $B\boldsymbol{x} = \boldsymbol{c}$, where $B \in \mathbb{R}^{n\times n}$ is nonsingular. We can convert the above system to the symmetric and positive definite setting by defining $A = B^TB$, $\boldsymbol{b} = B^T\boldsymbol{c}$ and then solving $A\boldsymbol{x} = \boldsymbol{b}$ with the conjugate gradient algorithm (or any other method for positive definite A).

Prof. A. C. Hansen

Mathematical Tripos Part II: Michaelmas Term 2024

Numerical Analysis – Lecture 19

Algorithm 4.26 (The conjugate gradient method) Here it is.

(A) For any initial vector $\boldsymbol{x}^{(0)}$, set $\boldsymbol{d}^{(0)} = \boldsymbol{r}^{(0)} = \boldsymbol{b} - A\boldsymbol{x}^{(0)}$;

(B) For $k \geq 0$, calculate $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{d}^{(k)}$ and the residual

$$\boldsymbol{r}^{(k+1)} = \boldsymbol{r}^{(k)} - \alpha_k A\boldsymbol{d}^{(k)}, \quad \text{with} \quad \alpha_k := \{\, \boldsymbol{r}^{(k+1)} \perp \boldsymbol{d}^{(k)}\} = \frac{(\boldsymbol{r}^{(k)}, \boldsymbol{d}^{(k)})}{(A\boldsymbol{d}^{(k)}, \boldsymbol{d}^{(k)})}\,, \quad k \geq 0\,. \tag{4.8}$$

(C) For the same k, the next conjugate direction is the vector

$$\boldsymbol{d}^{(k+1)} = \boldsymbol{r}^{(k+1)} + \beta_k \boldsymbol{d}^{(k)}, \quad \text{with} \quad \beta_k := \{\, \boldsymbol{d}^{(k+1)} \perp A\boldsymbol{d}^{(k)}\} = -\frac{(\boldsymbol{r}^{(k+1)}, A\boldsymbol{d}^{(k)})}{(\boldsymbol{d}^{(k)}, A\boldsymbol{d}^{(k)})}, \quad k \geq 0\,. \tag{4.9}$$

Theorem 4.27 (Properties of CGM) *For every $m \geq 0$, the conjugate gradient method has the following properties.*

(1) *The linear space spanned by the residuals $\{\boldsymbol{r}^{(i)}\}$ is the same as the linear space spanned by the conjugate directions $\{\boldsymbol{d}^{(i)}\}$ and it coincides with the space spanned by $\{A^i\boldsymbol{r}^{(0)}\}$:*

$$\operatorname{span}\{\boldsymbol{r}^{(i)}\}_{i=0}^{m} = \operatorname{span}\{\boldsymbol{d}^{(i)}\}_{i=0}^{m} = \operatorname{span}\{A^i\boldsymbol{r}^{(0)}\}_{i=0}^{m}\,.$$

(2) *The residuals satisfy the orthogonality conditions: $(\boldsymbol{r}^{(m)}, \boldsymbol{r}^{(i)}) = (\boldsymbol{r}^{(m)}, \boldsymbol{d}^{(i)}) = 0$ for $i < m$.*

(3) *The directions are conjugate (A-orthogonal): $(\boldsymbol{d}^{(m)}, \boldsymbol{d}^{(i)})_A = (\boldsymbol{d}^{(m)}, A\boldsymbol{d}^{(i)}) = 0$ for $i < m$.*

Proof. We use induction on $m \geq 0$, the assertions being trivial for $m = 0$, since $\boldsymbol{d}^{(0)} = \boldsymbol{r}^{(0)}$ and (2)-(3) are void. Therefore, assuming that the assertions are true for some $m = k$, we ask if they remain true when $m = k + 1$.

(1) Formula (4.9)

$$\boldsymbol{d}^{(k+1)} = \boldsymbol{r}^{(k+1)} + \beta_k \boldsymbol{d}^{(k)}$$

readily implies that equivalence of the spaces spanned by $(\boldsymbol{r}^{(i)})_0^k$ and $(\boldsymbol{d}^{(i)})_0^k$, is preserved when k is increased to $k + 1$. Similarly, from $\boldsymbol{r}^{(k+1)} = \boldsymbol{r}^{(k)} - \alpha_k A\boldsymbol{d}^{(k)}$ in (4.8), and from the inductive assumption $\boldsymbol{r}^{(k)}, \boldsymbol{d}^{(k)} \in \operatorname{span}\{A^i\boldsymbol{r}^{(0)}\}_{i=0}^{k}$, it follows that $\boldsymbol{r}^{(k+1)} \in \operatorname{span}\{A^i\boldsymbol{r}^{(0)}\}_{i=0}^{k+1}$.

(2) Turning to assertion (2), we need $\boldsymbol{r}^{(k+1)} \perp \boldsymbol{r}^{(i)}$ for $i \leq k$, which by (1) is equivalent to

$$\boldsymbol{r}^{(k+1)} \perp \boldsymbol{d}^{(i)} \quad \text{for} \quad i \leq k\,.$$

We have $\boldsymbol{r}^{(k+1)} \perp \boldsymbol{d}^{(k)}$ by the definition of α_k in (4.8), so we need

$$\boldsymbol{r}^{(k+1)} \overset{(4.8)}{=} \boldsymbol{r}^{(k)} - \alpha_k A\boldsymbol{d}^{(k)} \perp \boldsymbol{d}^{(i)} \quad \text{for} \quad i < k\,,$$

and this follow from the inductive assumptions $\boldsymbol{r}^{(k)} \perp \boldsymbol{d}^{(i)}$ and $A\boldsymbol{d}^{(k)} \perp \boldsymbol{d}^{(i)}$.

(3) It remains to justify (3), namely that $\boldsymbol{d}^{(k+1)}$ defined in (4.9) satisfies

$$\boldsymbol{d}^{(k+1)} \perp A\boldsymbol{d}^{(i)} \quad \text{for} \quad i \leq k\,.$$

The value of β_k in (4.9) is defined to give $\boldsymbol{d}^{(k+1)} \perp A\boldsymbol{d}^{(k)}$, so we need

$$\boldsymbol{d}^{(k+1)} \overset{(4.9)}{=} \boldsymbol{r}^{(k+1)} + \beta_k \boldsymbol{d}^{(k)} \perp A\boldsymbol{d}^{(i)} \quad \text{for} \quad i < k\,.$$

By the inductive hypothesis $\boldsymbol{d}^{(k)} \perp A\boldsymbol{d}^{(i)}$, hence it remains to establish that $\boldsymbol{r}^{(k+1)} \perp A\boldsymbol{d}^{(i)}$ for $i < k$. Now, the formula (4.8) yields $A\boldsymbol{d}^{(i)} = (\boldsymbol{r}^{(i)} - \boldsymbol{r}^{(i+1)})/\alpha_i$, therefore we require the conditions $\boldsymbol{r}^{(k+1)} \perp (\boldsymbol{r}^{(i)} - \boldsymbol{r}^{(i+1)})$ for $i < k$, and they are a consequence of the assertion (2) for $m = k + 1$ obtained previously. □

Corollary 4.28 (A termination property) *If the conjugate gradient method is applied in exact arithmetic, then, for any $\boldsymbol{x}^{(0)} \in \mathbb{R}^n$, termination occurs after at most n iterations. More precisely, termination occurs after at most s iterations, where $s = \dim \operatorname{span}\{A^i \boldsymbol{r}_0\}_{i=0}^{n-1}$ (which can be smaller than n).*

Proof. Assertion (2) of Theorem 4.27 states that residuals $(\boldsymbol{r}^{(k)})_{k\geq 0}$ form a sequence of mutually orthogonal vectors in $\mathbb{R}^n$, therefore at most n of them can be nonzero. Since they also belong to the space $\operatorname{span}\{A^i \boldsymbol{r}_0\}_{i=0}^{n-1}$, their number is bounded by the dimension of that space. □

Definition 4.29 (The Krylov subspaces) Let A be an $n \times n$ matrix, $\boldsymbol{v} \in \mathbb{R}^n$ nonzero, and $m \in \mathbb{N}$. The linear space $K_m(A, \boldsymbol{v}) := \operatorname{span}\{A^i \boldsymbol{v}\}_{i=0}^{m-1}$ is called the *m-th Krylov subspace* of $\mathbb{R}^n$.

Theorem 4.30 (Number of iterations in CGM) *Let $A > 0$, and let s be the number of its distinct eigenvalues. Then, for any $\boldsymbol{v}$,*

$$\dim K_m(A, \boldsymbol{v}) \leq s \quad \forall\, m\,. \tag{4.10}$$

Hence, for any $A > 0$, the number of iterations of the CGM for solving $A\boldsymbol{x} = \boldsymbol{b}$ is bounded by the number of distinct eigenvalues of A.

Proof. Inequality (4.10) is true not just for positive definite $A > 0$, but for any A with n linearly independent eigenvectors $(\boldsymbol{u}_i)$. Indeed, in that case one can expand $\boldsymbol{v} = \sum_{i=1}^n a_i \boldsymbol{u}_i$, and then group together eigenvectors with the same eigenvalues: for each λ_ν we set $\boldsymbol{w}_\nu = \sum_{k=1}^{m_\nu} a_{i_k} \boldsymbol{u}_{i_k}$ if $A\boldsymbol{u}_{i_k} = \lambda_\nu \boldsymbol{u}_{i_k}$. Then

$$\boldsymbol{v} = \textstyle\sum_{\nu=1}^s c_\nu \boldsymbol{w}_\nu\,, \qquad c_\nu \in \{0, 1\}\,,$$

hence $A^i \boldsymbol{v} = \sum_{\nu=1}^s c_\nu \lambda_\nu^i \boldsymbol{w}_\nu$, thus for any m we get $K_m(A, \boldsymbol{v}) \subseteq \operatorname{span}\{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_s\}$, and that proves (4.10). By Corollary 4.28, the number of iteration in CGM is bounded by $\dim K_m(A, \boldsymbol{r}^{(0)})$, hence the final conclusion. □

Remark 4.31 Theorem 4.30 shows that, unlike other iterative schemes, the conjugate gradient method is both iterative and direct: each iteration produces a reasonable approximation to the exact solution, and the exact solution itself will be recovered after n iterations at most.

We now simplify and reformulate Algorithm 4.26.
Firstly, we rewrite expressions for the parameters α_k and β_k in (4.8)-(4.9) as follows:

$$\begin{aligned}
\alpha_k &= \frac{(\boldsymbol{r}^{(k)}, \boldsymbol{d}^{(k)})}{(\boldsymbol{d}^{(k)}, A\boldsymbol{d}^{(k)})} \overset{(c)}{=} \frac{\|\boldsymbol{r}^{(k)}\|^2}{(\boldsymbol{d}^{(k)}, A\boldsymbol{d}^{(k)})} > 0\,,\\
\beta_k &= -\frac{(\boldsymbol{r}^{(k+1)}, A\boldsymbol{d}^{(k)})}{(\boldsymbol{d}^{(k)}, A\boldsymbol{d}^{(k)})} \overset{(a)}{=} -\frac{(\boldsymbol{r}^{(k+1)}, \boldsymbol{r}^{(k+1)} - \boldsymbol{r}^{(k)})}{(\boldsymbol{d}^{(k)}, \boldsymbol{r}^{(k+1)} - \boldsymbol{r}^{(k)})} \overset{(b)}{=} \frac{\|\boldsymbol{r}^{(k+1)}\|^2}{(\boldsymbol{d}^{(k)}, \boldsymbol{r}^{(k)})} \overset{(c)}{=} \frac{\|\boldsymbol{r}^{(k+1)}\|^2}{\|\boldsymbol{r}^{(k)}\|^2} > 0\,.
\end{aligned}$$

Here, for β, we used in (a) the fact that $A\boldsymbol{d}^{(k)}$ is a multiple of $\boldsymbol{r}^{(k+1)} - \boldsymbol{r}^{(k)}$ by (4.8), and in (b) orthogonality of $\boldsymbol{r}^{(k+1)}$ to both $\boldsymbol{r}^{(k)}, \boldsymbol{d}^{(k)}$ proved in Theorem 4.27(2). Then, for both β and α, we used in (c) the property $(\boldsymbol{d}^{(k)}, \boldsymbol{r}^{(k)}) = \|\boldsymbol{r}^{(k)}\|^2$ which follows from (4.9) with index $k+1$, taking in account orthogonality $\boldsymbol{r}^{(k+1)} \perp \boldsymbol{d}^{(k)}$.

Secondly, we let $\boldsymbol{x}^{(0)}$ be the zero vector.

Algorithm 4.32 (Standard form of the conjugate gradient method) Here it is.

(1) Set $k = 0$, $\boldsymbol{x}^{(0)} = 0$, $\boldsymbol{r}^{(0)} = \boldsymbol{b}$, and $\boldsymbol{d}^{(0)} = \boldsymbol{r}^{(0)}$;
(2) Calculate the matrix-vector product $\boldsymbol{v}^{(k)} = A\boldsymbol{d}^{(k)}$ and $\alpha_k = \|\boldsymbol{r}^{(k)}\|^2 / (\boldsymbol{d}^{(k)}, \boldsymbol{v}^{(k)}) > 0$;
(3) Apply the formulae $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{d}^{(k)}$ and $\boldsymbol{r}^{(k+1)} = \boldsymbol{r}^{(k)} - \alpha_k \boldsymbol{v}^{(k)}$;
(4) Stop if $\|\boldsymbol{r}^{(k+1)}\|$ is acceptably small;
(5) Set $\boldsymbol{d}^{(k+1)} = \boldsymbol{r}^{(k+1)} + \beta_k \boldsymbol{d}^{(k)}$, where $\beta_k = \|\boldsymbol{r}^{(k+1)}\|^2 / \|\boldsymbol{r}^{(k)}\|^2 > 0$;
(6) Increase $k \to k+1$ and go back to (2).

The total work is dominated by the number of iterations, multiplied by the time it takes to compute $\boldsymbol{v}^{(k)} = A\boldsymbol{d}^{(k)}$. Thus the conjugate gradient algorithm is highly suitable when most of the elements of A are zero, i.e. when A is *sparse*.

Prof. A. C. Hansen

Mathematical Tripos Part II: Michaelmas Term 2024

Numerical Analysis – Lecture 20

Technique 4.33 (Preconditioning) In $A\boldsymbol{x} = \boldsymbol{b}$, we change variables, $\boldsymbol{x} = P^T\widehat{\boldsymbol{x}}$, where P is a non-singular $n \times n$ matrix, and multiply both sides with P. Thus, instead of $A\boldsymbol{x} = \boldsymbol{b}$, we are solving the linear system

$$PAP^T\widehat{\boldsymbol{x}} = P\boldsymbol{b} \quad \Leftrightarrow \quad \widehat{A}\widehat{x} = \widehat{\boldsymbol{b}}\,. \tag{4.11}$$

Note that symmetry and positive definiteness of A imply that $\widehat{A} = PAP^T$ is also symmetric and positive definite since $(\widehat{A}\boldsymbol{y}, \boldsymbol{y}) = (PAP^T\boldsymbol{y}, \boldsymbol{y}) = (AP^T\boldsymbol{y}, P^T\boldsymbol{y}) > 0$. Therefore, we can apply conjugate gradients to the new system. This results in the solution $\widehat{\boldsymbol{x}}$, hence $\boldsymbol{x} = P^T\widehat{\boldsymbol{x}}$. This procedure is called the *preconditioned conjugate gradient method* and the matrix P is called the *preconditioner*.

The *condition number* of a matrix A is the value $\kappa(A) := \|A\|\cdot\|A^{-1}\|$, so for a symmetric positive definite matrix A it is the ratio between its largest and smallest eigenvalues,

$$\kappa(A) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)} \geq 1\,.$$

The closer is this number to 1, the faster is convergence of CGM. More precisely, for the rate of convergnce of CGM, we have the uppper estimate

$$\|\boldsymbol{e}^{(k)}\|_A \leq 2\rho^k\,\|\boldsymbol{e}^{(0)}\|_A\,, \qquad \rho = \rho_A = \frac{\sqrt{\kappa(A)}-1}{\sqrt{\kappa(A)}+1} < 1\,. \tag{4.12}$$

The main idea of preconditioning is to pick P in (4.11) so that $\kappa(\widehat{A})$ is much smaller than $\kappa(A)$, thus accelerating convergence.

To this end, we note that the similarity transform $B \to C^{-1}BC$ preserves spectrum, hence

$$\kappa(\widehat{A}) = \kappa(PAP^T) = \kappa(P^{-1}[PAP^T]P) = \kappa(AP^TP)\,,$$

and if we set

$$S^{-1} := P^TP =: (QQ^T)^{-1},$$

then it is suggestive to choose S as an approximation to A which is easy to Cholesky-factorize, i.e., $S = QQ^T$ (or already in this form), and then take $P = Q^{-1}$. Then $AP^TP = AS^{-1}$ is close to identity, hence

$$\kappa(\widehat{A}) = \kappa(AP^TP) \approx \kappa(I) = 1 \quad \Rightarrow \quad \kappa(\widehat{A}) \ll \kappa(A)\,,$$

and the preconditioned system (4.11) will be solved much faster because of (4.12).

Each step in the CGM for solving $A\boldsymbol{x} = \boldsymbol{b}$ requires one matrix-vector product $A\boldsymbol{y}$, so with $P = Q^{-1}$, additional expense in each step of the CGM for the preconditioned system (4.11) while computing $\widehat{A}\boldsymbol{y} = PAP^T\boldsymbol{y}$ is two additional computations

$$\boldsymbol{u} = P^T\boldsymbol{y} = Q^{-T}\boldsymbol{y}, \qquad \boldsymbol{v} = P\boldsymbol{z} = Q^{-1}\boldsymbol{z},$$

for some $\boldsymbol{y}, \boldsymbol{z} \in \mathbb{R}^n$, but note that computing $Q^{-1}\boldsymbol{z}$ is the same as solving the linear system $Q\boldsymbol{v} = \boldsymbol{z}$, which is cheap (via forward substitution) as Q is a lower triangular matrix.

Example 4.34 1) The simplest choice of S is $D = \operatorname{diag} A$, then $P = D^{-1/2}$ in (4.11).

2) Another possibility is to choose S as a band matrix with small bandwidth. For example, solving the Poisson equation with the five-point formula, we may take S to be the tridiagonal part of A.

3) One can also take $P = L^{-1}$, where L is the lower triangular part of A (maybe imposing some changes). For example, for the Poisson equation, with $m = 20$ hence dealing with 400×400 system, we take P^{-1} as the lower triangular part of A, but change the diagonal elements from 4 to $\frac{5}{2}$. Then we get a computer precision after just 30 iterations.

Example 4.35 For the tridiagonal system $A\boldsymbol{x} = \boldsymbol{b}$ below, we choose the preconditioner as follows.

$$A = \begin{bmatrix} 2 & -1 & & \\ -1 & 2 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix}, \qquad Q = \begin{bmatrix} 1 & & & \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix}, \qquad S = QQ^T = \begin{bmatrix} 1 & -1 & & \\ -1 & 2 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix}.$$

The matrix S coincides with A except at the $(1,1)$-entry. The matrix $\widehat{A} = Q^{-1}AQ^{-T}$ for the preconditioned CGM has just two distinct eigenvalues, and we recover the exact solution just in two steps. To see the latter, note that $\widehat{A}$ is similar to $Q^{-T}Q^{-1}A = S^{-1}A$, hence it has the same spectrum. Since $A = S + \boldsymbol{e}_1\boldsymbol{e}_1^T$, we have $S^{-1}A = I + \boldsymbol{u}\boldsymbol{e}_1^T$, a rank-1 perturbation of the identity matrix, with all eigenvalues but one equal 1 (the remaining one equal $1 + u_1$).

Remark 4.36 (Rate of convergence of CGM) Here, we prove (4.12). As we have seen, every direction $\boldsymbol{d}^{(i)}$ in CGM is a linear combination of the vectors $(A^s\boldsymbol{r}^{(0)})_{s=0}^{i}$, therefore, any vector of the form $\widehat{\boldsymbol{x}}^{(k)} = \boldsymbol{x}^{(0)} + \sum_{i=0}^{k-1} a_i\boldsymbol{d}^{(i)}$ can be represented as

$$\widehat{\boldsymbol{x}}^{(k)} = \boldsymbol{x}^{(0)} + \textstyle\sum_{i=0}^{k-1} c_i A^i \boldsymbol{r}^{(0)} \,. \tag{4.13}$$

Approximation of this kind also arises from various iterative methods of the form

$$\widehat{\boldsymbol{x}}^{(k+1)} = \widehat{\boldsymbol{x}}^{(k)} - \tau_k(A\widehat{\boldsymbol{x}}^{(k)} - \boldsymbol{b}) \,,$$

in particular for the steepest descent method.

Subtracting both parts of (4.13) from the exact solution $\boldsymbol{x}^*$ we obtain $\widehat{\boldsymbol{e}}^{(k)} = \boldsymbol{e}^{(0)} - \sum_{i=0}^{k-1} c_i A^i \boldsymbol{r}^{(0)}$, and since $\boldsymbol{r}^{(0)} = A\boldsymbol{e}^{(0)}$, we can express the error $\widehat{\boldsymbol{e}}^{(k)} = \boldsymbol{x}^* - \widehat{\boldsymbol{x}}^{(k)}$ as

$$\widehat{\boldsymbol{e}}^{(k)} = (I - \textstyle\sum_{i=1}^{k} c_i A^i)\, \boldsymbol{e}^{(0)} = P_k(A)\, \boldsymbol{e}^{(0)}, \tag{4.14}$$

where P_k is a polynomial of degree $\leq k$, which satisfies $P_k(0) = 1$.

Now we make use of the following.

Theorem 4.37 (Non-examinable) *Given $A \in \mathbb{R}^{n\times n}$, $A > 0$, let $\{\boldsymbol{d}^{(k)}\}_{k=0}^{m-1}$ be a set of the conjugate directions, i.e., $(A\boldsymbol{d}^{(k)}, \boldsymbol{d}^{(i)}) = 0$ for $i < k$, and consider*

$$F(\boldsymbol{x}^{(k)}) := \|\boldsymbol{x}^* - \boldsymbol{x}^{(k)}\|_A^2 = \|\boldsymbol{e}^{(k)}\|_A^2 \,.$$

Then the value of $F(\boldsymbol{x}^{(m+1)})$ obtained through the CGM coincides with the minimum of $F(\boldsymbol{y})$ taken over all $\boldsymbol{y} = \boldsymbol{x}^{(0)} + \sum_{k=0}^{m} c_k\boldsymbol{d}^{(k)}$ simultaneously, namely

$$\arg \min_{c_0,\ldots,c_m} F(\boldsymbol{y}) = \boldsymbol{x}^{(m+1)} = \boldsymbol{x}^{(0)} + \sum_{k=0}^{m} \alpha_k \boldsymbol{d}^{(k)} \,.$$

Hence, at the k-th stage, the CGM produces the vector $\boldsymbol{x}^{(k)}$ that minimizes the functional

$$F(\widehat{\boldsymbol{x}}^{(k)}) = \|\widehat{\boldsymbol{e}}^{(k)}\|_A^2 = (A\widehat{\boldsymbol{e}}^{(k)}, \widehat{\boldsymbol{e}}^{(k)})$$

over all vectors $\widehat{\boldsymbol{x}}^{(k)}$ of the form $\widehat{\boldsymbol{x}}^{(k)} = \boldsymbol{x}^{(0)} + \sum_{i=0}^{k-1} a_i\boldsymbol{d}^{(i)}$, hence over all $\widehat{\boldsymbol{e}}^{(k)}$ of the form (4.14). Expressing $\boldsymbol{e}^{(0)}$ as $\boldsymbol{e}^{(0)} = \sum \gamma_i \boldsymbol{w}_i$, where $(\boldsymbol{w}_i)$ are orthonormal eigenvectors of A, we find from (4.14) that $\widehat{\boldsymbol{e}}^{(k)} = \sum_i \gamma_i P_k(\lambda_i)\boldsymbol{w}_i$, and $A\widehat{\boldsymbol{e}}^{(k)} = \sum_i \gamma_i P_k(\lambda_i)\lambda_i\boldsymbol{w}_i$, and respectively

$$\|\widehat{\boldsymbol{e}}^{(k)}\|_A^2 = \textstyle\sum_i [P_k(\lambda_i)]^2 \lambda_i \gamma_i^2 \leq \max_{\lambda\in\sigma(A)} [P_k(\lambda)]^2 \, \|\boldsymbol{e}^{(0)}\|_A^2 \,.$$

Hence, because of the minimization property of CGM,

$$\|\boldsymbol{e}^{(k)}\|_A = \min_{P_k} \|\widehat{\boldsymbol{e}}^{(k)}\|_A \leq \min_{P_k} \max_{\lambda\in\sigma(A)} |P_k(\lambda)| \, \|\boldsymbol{e}^{(0)}\|_A \,.$$

Now, assume that, for the spectrum $\sigma(A)$, we know the largest and the smallest eigenvalues, or some lower and upper bounds, say, $0 < m \le \lambda \le M$. Then the following minimization problem, on the class of polynomials of degree k, arises:

$$P_k(0) = 1, \quad \max_{x\in[m,M]} |P_k(x)| \to \min .$$

This problem has a classical solution $P_k^* = T_k^*$, where T_k^* is the Chebyshev polynomial on the interval $[m, M]$, which is obtained by dilation and translation of the standard Chebyshev polynomial T_k given on the interval $[-1, 1]$:

$$T_k(x) = \cos k\theta, \qquad x = \cos\theta, \qquad \theta \in [0, \pi] .$$

One can show that $|T_k^*(x)| \le 2\rho^k$ on the interval $[m, M]$, hence the rate of convergence of CGM admits the following estimate:

$$\|\boldsymbol{e}^{(k)}\|_A \le 2\rho^k \|\boldsymbol{e}^{(0)}\|_A, \qquad \rho = \frac{\sqrt{M}-\sqrt{m}}{\sqrt{M}+\sqrt{m}} < 1, \qquad \sigma(A) \in [m, M] .$$

Prof. A. C. Hansen

Mathematical Tripos Part II: Michaelmas Term 2024

Numerical Analysis – Lecture 21

5 Eigenvalues and eigenvectors

Remark 5.1 (Introduction to matrix eigenvalue calculations) Let A be a real $n \times n$ matrix. The eigenvalue equation is $A\boldsymbol{w} = \lambda \boldsymbol{w}$, where λ is a scalar, which may be complex if A is not symmetric. There exists a nonzero vector $\boldsymbol{w} \in \mathbb{R}^n$ satisfying this equation if and only if $\det(A - \lambda I) = 0$. The function $p(\lambda) = \det(A - \lambda I)$, $\lambda \in \mathbb{C}$, is a polynomial of degree n, but calculating the eigenvalues by finding the roots of p is a disaster area because of loss of accuracy due to rounding errors.

If the polynomial has some multiple roots and if A is not symmetric, then the number of linearly independent eigenvectors may be fewer than n, but there are always n mutually orthogonal real eigenvectors in the symmetric case. We assume in all cases, however, that the eigenvalue equations $A\boldsymbol{w}_i = \lambda_i \boldsymbol{w}_i$, $i = 1..n$, are satisfied by eigenvectors $\boldsymbol{w}_i$ that are linearly independent, which can be achieved by making an arbitrarily small change to A if necessary.

Method 5.2 (The power method) The iterative algorithms that will be studied for the calculation of eigenvalues and eigenvectors are all closely related to the power method, which has the following basic form for generating a single eigenvalue and eigenvector of A.

We pick a nonzero vector $\boldsymbol{x}^{(0)}$ in $\mathbb{R}^n$. Then, for $k = 0, 1, 2, \ldots$, we let $\boldsymbol{x}^{(k+1)}$ be a nonzero multiple of $A\boldsymbol{x}^{(k)}$, typically to satisfy $\|\boldsymbol{x}^{(k+1)}\| = 1$ so that

$$\boldsymbol{x}^{(k+1)} = A\boldsymbol{x}^{(k)} / \|A\boldsymbol{x}^{(k)}\|, \qquad k = 0, 1, 2, \ldots$$

This method is oriented on finding an eigenvector corresponding to the largest eigenvalue as the the following theorem shows.

Theorem 5.3 *Let $A\boldsymbol{w}_i = \lambda_i \boldsymbol{w}_i$, where the eigenvalues of A satisfy $|\lambda_1| \leq \cdots \leq |\lambda_{n-1}| < |\lambda_n|$ and the eigenvectors are of the unit length $\|\boldsymbol{w}_i\| = 1$. Assume $\boldsymbol{x}^{(0)} = \sum_{i=1}^n c_i \boldsymbol{w}_i$ with $c_n \neq 0$. Then $\boldsymbol{x}^{(k)} \to \pm \boldsymbol{w}_n$ as $k \to \infty$.*

Proof. Given $x^{(0)}$ as in the assumption, $\boldsymbol{x}^{(k)}$ is a multiple of

$$A^k \boldsymbol{x}^{(0)} = \textstyle\sum_{i=1}^n c_i \lambda_i^k \boldsymbol{w}_i = c_n \lambda_n^k \Big(\boldsymbol{w}_n + \sum_{i=1}^{n-1} \frac{c_i}{c_n} \big(\frac{\lambda_i}{\lambda_n} \big)^k \boldsymbol{w}_i \Big).$$

Since $\|\boldsymbol{x}^{(k)}\| = \|\boldsymbol{w}_n\| = 1$, we conclude that $\boldsymbol{x}^{(k)} = \pm \boldsymbol{w}_n + \mathcal{O}(\rho^k)$, where the sign is that of $c_n \lambda_n^k$ and the ratio $\rho = \frac{|\lambda_{n-1}|}{|\lambda_n|} < 1$ characterizes the rate of convergence. □

Here are the details of an implementation of the procedure.

0. Pick $\boldsymbol{x}^{(0)} \in \mathbb{R}^n$ satisfying $\|\boldsymbol{x}^{(0)}\| = 1$. Let ε be a small positive tolerance. Set $k = 0$.

1. Calculate $\widetilde{\boldsymbol{x}}^{(k+1)} = A\boldsymbol{x}^{(k)}$ and set $\lambda = \frac{\boldsymbol{x}^{(k)T} A \boldsymbol{x}^{(k)}}{\boldsymbol{x}^{(k)T} \boldsymbol{x}^{(k)}}$.
(This λ is called the *Raleigh quotient* and it minimizes $f(\mu) = \|\widetilde{\boldsymbol{x}}^{(k+1)} - \mu \boldsymbol{x}^{(k)}\|$ over μ.

2. If $f(\lambda) \leq \varepsilon$, accept λ as an eigenvalue and $\boldsymbol{x}^{(k)}$ as the corresponding eigenvector.

3. Otherwise, let $\boldsymbol{x}^{(k+1)} = \widetilde{\boldsymbol{x}}^{(k+1)} / \|\widetilde{\boldsymbol{x}}^{(k+1)}\|$, increase k by one and go back to **1.**

The termination occurs because, by the previous theorem, we have

$$\begin{aligned} \|\widetilde{\boldsymbol{x}}^{(k+1)} - \lambda \boldsymbol{x}^{(k)}\| &= \min_{\mu} \|\widetilde{\boldsymbol{x}}^{(k+1)} - \mu \boldsymbol{x}^{(k)}\| \leq \|\widetilde{\boldsymbol{x}}^{(k+1)} - \lambda_n \boldsymbol{x}^{(k)}\| \\ &= \|A\boldsymbol{x}^{(k)} - \lambda_n \boldsymbol{x}^{(k)}\| = \|A\boldsymbol{w}_n - \lambda_n \boldsymbol{w}_n\| + \mathcal{O}(\rho^k) = \mathcal{O}(\rho^k) \to 0. \end{aligned}$$

Discussion 5.4 (Deficiencies of the power method) The power method may perform adequately if $c_n \neq 0$ and $|\lambda_{n-1}| < |\lambda_n|$, where we are using the notation of Theorem 5.3, but often it is unacceptably slow. The difficulty of $c_n = 0$ is that, theoretically, in this case the method should find

an eigenvector $\boldsymbol{w}_m$ with the largest m such that $c_m \neq 0$, but practically computer rounding errors can introduce a small nonzero component of $\boldsymbol{w}_n$ into the sequence $\boldsymbol{x}^{(k)}$, and then $\boldsymbol{w}_n$ may be found eventually, but one has to wait for the small component to grow. Moreover, $|\lambda_{n-1}| = |\lambda_n|$ is not uncommon when A is real and nonsymmetric, because the spectral radius of A may be due to a complex conjugate pair of eigenvalues. Next, we will study the inverse iterations (with *shifts*), because they can be highly useful, particularly in the more efficient methods for eigenvalue calculations that will be considered later.

Method 5.5 (Inverse iteration) This method is highly useful in practice. It is similar to the power method 5.2, except that, instead of $\boldsymbol{x}^{(k+1)}$ being a multiple of $A\boldsymbol{x}^{(k)}$, we make the choice

$$(A - sI)\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)}, \qquad k = 0, 1, \ldots, \tag{5.1}$$

where s is a scalar that may depend on k and $\|x^{(k)}\| = 1$. Therefore the calculation of $\boldsymbol{x}^{(k+1)}$ from $\boldsymbol{x}^{(k)}$ requires the solution of an $n \times n$ system of linear equations whose matrix is $(A - sI)$. Further, if s is a constant and if $A - sI$ is nonsingular, we deduce from (5.1) that $\boldsymbol{x}^{(k)}$ is a multiple of $(A - sI)^{-k}\boldsymbol{x}^{(0)}$.

We again let $\boldsymbol{x}^{(0)} = \sum_{i=1}^{n} c_i \boldsymbol{w}_i$, as in the proof of Theorem 5.3, assuming that $\boldsymbol{w}_i$, $i = 1..n$, are linearly independent eigenvectors of A that satisfy $A\boldsymbol{w}_i = \lambda_i \boldsymbol{w}_i$. Therefore we note that the eigenvalue equation implies $(A - sI)\boldsymbol{w}_i = (\lambda_i - s)\boldsymbol{w}_i$, which in turn implies $(A - sI)^{-1}\boldsymbol{w}_i = (\lambda_i - s)^{-1}\boldsymbol{w}_i$. It follows that $\boldsymbol{x}^{(k)}$ is a multiple of

$$(A - sI)^{-k}\boldsymbol{x}^{(0)} = \sum_{i=1}^{n} c_i (A - sI)^{-k}\boldsymbol{w}_i = \sum_{i=1}^{n} c_i (\lambda_i - s)^{-k}\boldsymbol{w}_i.$$

Thus, if the m-th number in the set $\{|\lambda_i - s|\}$ is the smallest and if c_m is nonzero, then $\boldsymbol{x}^{(k)}$ tends to be a multiple of $\boldsymbol{w}_m$ as $k \to \infty$. We see that the speed of convergence can be excellent if s is very close to λ_m. Further, it can be made even faster by adjusting s during the calculation. Typical details are given in the following implementation.

Algorithm 5.6 (Typical implementation of inverse iteration)

0. Set s to an estimate of an eigenvalue of A. Prescribe $\boldsymbol{x}^{(0)} \neq 0$, let $0 < \varepsilon \ll 1$ and set $k = 0$.

1. Calculate (with pivoting if necessary) the LU factorization of $A - sI$.

2. Stop if U is singular because then s is an eigenvalue of A, while its eigenvector is any vector in the null space of U: it can be found easily, U being upper triangular.

3. Calculate $\boldsymbol{x}^{(k+1)}$ by solving $(A - sI)\boldsymbol{x}^{(k+1)} = LU\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)}$ using the LU factorization from **1.**

4. Set η to the number that minimizes $f(\mu) = \|\boldsymbol{x}^{(k)} - \mu\boldsymbol{x}^{(k+1)}\|$.

5. Stop if $f(\eta) \leq \varepsilon\|\boldsymbol{x}^{(k+1)}\|$. Since $f(\eta) = \|A\boldsymbol{x}^{(k+1)} - (s + \eta)\boldsymbol{x}^{(k+1)}\|$, we let $s + \eta$ be the calculated eigenvalue of A and $\boldsymbol{x}^{(k+1)}/\|\boldsymbol{x}^{(k+1)}\|$ be its eigenvector.

6. Otherwise, replace $\boldsymbol{x}^{(k+1)}$ by $\boldsymbol{x}^{(k+1)}/\|\boldsymbol{x}^{(k+1)}\|$, increase k by one, and either return to **3** without changing s or to **1** after replacing s by $s + \eta$.

Remark 5.7 (Further on inverse iteration) Algorithm 5.6 is very efficient if A is an *upper Hessenberg matrix:* every element of A under the first subdiagonal is zero (i.e. $a_{ij} = 0$ if $j < i - 1$). In this case the LU factorization in **1** requires just $\mathcal{O}(n^2)$ or $\mathcal{O}(n)$ operations when A is nonsymmetric or symmetric, respectively. Thus the replacement of s by $s + \eta$ in **6** need not be expensive, so fast convergence can often be achieved easily. There are standard ways of giving A this convenient form which will be considered later.

Prof. A. C. Hansen

Mathematical Tripos Part II: Michaelmas Term 2024

Numerical Analysis – Lecture 22

Theorem 5.8 *Let A and S be $n \times n$ matrices, S being nonsingular. Then $\boldsymbol{w}$ is an eigenvector of A with eigenvalue λ if and only if $\widehat{\boldsymbol{w}} = S\boldsymbol{w}$ is an eigenvector of $\widehat{A} = SAS^{-1}$ with the same eigenvalue.*

Proof. $A\boldsymbol{w} = \lambda\boldsymbol{w} \quad \Leftrightarrow \quad AS^{-1}(S\boldsymbol{w}) = \lambda\boldsymbol{w} \quad \Leftrightarrow \quad (SAS^{-1})(S\boldsymbol{w}) = \lambda(S\boldsymbol{w}).$ □

Definition 5.9 (Deflation) Suppose that we have found one solution of the eigenvector equation $A\boldsymbol{w} = \lambda\boldsymbol{w}$, where A is again $n \times n$. Then *deflation* is the task of constructing an $(n-1) \times (n-1)$ matrix, B say, whose eigenvalues are the other eigenvalues of A. Specifically, we apply a similarity transformation S to A such that the first column of $\widehat{A} = SAS^{-1}$ is λ times the first coordinate vector $\boldsymbol{e}_1$, because it follows from the characteristic equation for eigenvalues and from Theorem 5.8 that we can let B be the bottom right $(n-1) \times (n-1)$ submatrix of $\widehat{A} = SAS^{-1}$.

We write the condition on S as $(SAS^{-1})\boldsymbol{e}_1 = \lambda\boldsymbol{e}_1$. Then the last equation in the proof of Theorem 5.8 shows that it is sufficient if S has the property $S\boldsymbol{w} = c\boldsymbol{e}_1$, where c is any nonzero scalar.

Technique 5.10 (Algorithm for deflation for symmetric A) Suppose that A is symmetric and $\boldsymbol{w} \in \mathbb{R}^n$, $\lambda \in \mathbb{R}$ are given so that $A\boldsymbol{w} = \lambda\boldsymbol{w}$. We seek a nonsingular matrix S such that $S\boldsymbol{w} = c\boldsymbol{e}_1$ and such that SAS^{-1} is also symmetric. The last condition holds if S is orthogonal, since then $S^{-1} = S^T$. It is suitable to pick a *Householder reflection*, which means that S has the form

$$H_u = I - 2\boldsymbol{u}\boldsymbol{u}^T/\|\boldsymbol{u}\|^2, \quad \text{where} \quad \boldsymbol{u} \in \mathbb{R}^n.$$

Specifically, we recall from the Numerical Analysis IB course that Householder reflections are orthogonal and that, because $H_u\boldsymbol{u} = -\boldsymbol{u}$ and $H_u\boldsymbol{v} = \boldsymbol{v}$ if $\boldsymbol{u}^T\boldsymbol{v} = 0$, they reflect any vector in $\mathbb{R}^n$ with respect to the $(n-1)$-dimensional hyperplane orthogonal to $\boldsymbol{u}$. So, for any two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ of equal lengths,

$$H_{\boldsymbol{u}}\boldsymbol{x} = \boldsymbol{y}, \quad \text{where} \quad \boldsymbol{u} = \boldsymbol{x} - \boldsymbol{y}.$$

Hence,

$$\left(I - 2\frac{\boldsymbol{u}\boldsymbol{u}^T}{\|\boldsymbol{u}\|^2}\right)\boldsymbol{w} = \pm\|\boldsymbol{w}\|\boldsymbol{e}_1\,, \quad \text{where} \quad \boldsymbol{u} = \boldsymbol{w} \mp \|\boldsymbol{w}\|\boldsymbol{e}_1\,.$$

Since the bottom $n-1$ components of $\boldsymbol{u}$ and $\boldsymbol{w}$ coincide, the calculation of $\boldsymbol{u}$ requires only $\mathcal{O}(n)$ computer operations. Further, the calculation of SAS^{-1} can be done in only $\mathcal{O}(n^2)$ operations, taking advantage of the form $S = I - 2\boldsymbol{u}\boldsymbol{u}^T/\|\boldsymbol{u}\|^2$, even if all the elements of A are nonzero.

After deflation, we may find an eigenvector, $\widehat{\boldsymbol{w}}$ say, of SAS^{-1}. Then the new eigenvector of A, according to Theorem 5.8, is $S^{-1}\widehat{\boldsymbol{w}} = S\widehat{\boldsymbol{w}}$, because Householder matrices, like all symmetric orthogonal matrices, are *involutions*: $S^2 = I$.

Revision 5.11 (Givens rotations) The notation $\Omega^{[i,j]}$ denotes the following $n \times n$ matrix

$$\Omega^{[i,j]} = \begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & c & & s & & \\ & & & \ddots & & & \\ & & -s & & c & & \\ & & & & & \ddots & \\ & & & & & & 1 \\ & & \uparrow & & \uparrow & & \\ & & i & & j & & \end{bmatrix}, \quad c^2 + s^2 = 1.$$

Generally, for any vector $\boldsymbol{a}_k \in \mathbb{R}^n$, we can find a matrix $\Omega^{[i,j]}$ such that

$$\Omega^{[i,j]}\boldsymbol{a} = \begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & c & & s & & \\ & & & \ddots & & & \\ & & -s & & c & & \\ & & & & & \ddots & \\ & & & & & & 1 \\ & & \uparrow & & \uparrow & & \\ & & i & & j & & \end{bmatrix} \begin{bmatrix} a_{1k} \\ \vdots \\ a_{ik} \\ \vdots \\ a_{jk} \\ \vdots \\ a_{nk} \end{bmatrix} = \begin{bmatrix} a_{1k} \\ \vdots \\ r \\ \vdots \\ 0 \\ \vdots \\ a_{nk} \end{bmatrix} \begin{matrix} \\ \\ \leftarrow i \\ \\ \leftarrow j \\ \\ \\ \end{matrix} \qquad \begin{aligned} c &= \tfrac{a_{ik}}{\sqrt{a_{ik}^2 + a_{jk}^2}}\,, \\ s &= \tfrac{a_{jk}}{\sqrt{a_{ik}^2 + a_{jk}^2}}\,, \\ r &= \sqrt{a_{ik}^2 + a_{jk}^2}\,. \end{aligned}$$

1) We can choose $\Omega^{[i,j]}$ so that any prescribed element $\widetilde{a}_{jk}$ in the j-th row of $\widetilde{A} = \Omega^{[i,j]} \times A$ is zero.

2) The rows of $\widetilde{A} = \Omega^{[i,j]} \times A$ are the same as the rows of A, except that the i-th and j-th rows of the product are linear combinations of the i-th and j-th rows of A.

3) The columns of $\widehat{A} = \widetilde{A} \times \Omega^{[i,j]T}$ are the same as the columns of $\widetilde{A}$, except that the i-th and j-th columns of $\widehat{A}$ are linear combinations of the i-th and j-th columns of $\widetilde{A}$.

4) $\Omega^{[i,j]}$ is an orthogonal matrix, thus $\widehat{A} = \Omega^{[i,j]} A \Omega^{[i,j]T}$ inherits the eigenvalues of A.

5) If A is symmetric, then so is $\widehat{A}$.

Method 5.12 (Transformation to an upper Hessenberg form) We replace A by $\widehat{A} = SAS^{-1}$, where S is a product of Givens rotations $\Omega^{[i,j]}$ chosen to annihilate subsubdiagonal elements $a_{j,i-1}$ in the $(i$–$1)$-st column:

$$\begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \overset{\Omega^{[2,3]}\times}{\to} \begin{bmatrix} * & * & * & * \\ \bullet & \bullet & \bullet & \bullet \\ 0 & \bullet & \bullet & \bullet \\ * & * & * & * \end{bmatrix} \overset{\times\Omega^{[2,3]T}}{\to} \begin{bmatrix} * & \bullet & \bullet & * \\ * & \bullet & \bullet & * \\ 0 & \bullet & \bullet & * \\ * & \bullet & \bullet & * \end{bmatrix} \overset{\Omega^{[2,4]}\times}{\to} \begin{bmatrix} * & * & * & * \\ \bullet & \bullet & \bullet & \bullet \\ 0 & * & * & * \\ 0 & \bullet & \bullet & \bullet \end{bmatrix} \overset{\times\Omega^{[2,4]T}}{\to} \begin{bmatrix} * & \bullet & * & \bullet \\ * & \bullet & * & \bullet \\ 0 & \bullet & * & \bullet \\ 0 & \bullet & * & \bullet \end{bmatrix} \overset{\Omega^{[3,4]}\times}{\to} \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ 0 & \bullet & \bullet & \bullet \\ 0 & 0 & \bullet & \bullet \end{bmatrix} \overset{\times\Omega^{[3,4]T}}{\to} \begin{bmatrix} * & * & \bullet & \bullet \\ * & * & \bullet & \bullet \\ 0 & * & \bullet & \bullet \\ 0 & 0 & \bullet & \bullet \end{bmatrix}$$

The $\bullet$-elements have changed through a single transformation while the $*$-elements remained the same. It is seen that every element that we have set to zero remains zero, and the final outcome is indeed an upper Hessenberg matrix. If A is symmetric then so will be the outcome of the calculation, hence it will be tridiagonal. In general, the cost of this procedure is $\mathcal{O}(n^3)$.

Alternatively, we can transform A to upper Hessenberg using *Householder reflections*, rather than Givens rotations. In that case we deal with a column at a time, taking $\boldsymbol{u}$ such that, with $H_u = I - 2\boldsymbol{u}\boldsymbol{u}^T/\|\boldsymbol{u}\|^2$, the i-th column of $\widetilde{B} = H_u B$ is consistent with the upper Hessenberg form. Such a $\boldsymbol{u}$ has its first i coordinates vanishing, therefore $\widehat{B} = \widetilde{B} H_u^T$ has the first i columns unchanged, and all new and old zeros (which are in the first i columns) stay untouched.

$$\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix} \overset{H_1\times}{\to} \begin{bmatrix} * & * & * & * & * \\ \bullet & \bullet & \bullet & \bullet & \bullet \\ 0 & \bullet & \bullet & \bullet & \bullet \\ 0 & \bullet & \bullet & \bullet & \bullet \\ 0 & \bullet & \bullet & \bullet & \bullet \end{bmatrix} \overset{\times H_1^T}{\to} \begin{bmatrix} * & \bullet & \bullet & \bullet & \bullet \\ * & \bullet & \bullet & \bullet & \bullet \\ 0 & \bullet & \bullet & \bullet & \bullet \\ 0 & \bullet & \bullet & \bullet & \bullet \\ 0 & \bullet & \bullet & \bullet & \bullet \end{bmatrix} \overset{H_2\times}{\to} \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & \bullet & \bullet & \bullet & \bullet \\ 0 & 0 & \bullet & \bullet & \bullet \\ 0 & 0 & \bullet & \bullet & \bullet \end{bmatrix} \overset{\times H_2^T}{\to} \begin{bmatrix} * & * & \bullet & \bullet & \bullet \\ * & * & \bullet & \bullet & \bullet \\ 0 & * & \bullet & \bullet & \bullet \\ 0 & 0 & \bullet & \bullet & \bullet \\ 0 & 0 & \bullet & \bullet & \bullet \end{bmatrix} \overset{H_3\times}{\to} \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & \bullet & \bullet & \bullet \\ 0 & 0 & \bullet & \bullet & \bullet \end{bmatrix} \overset{\times H_3^T}{\to} \begin{bmatrix} * & * & * & \bullet & \bullet \\ * & * & * & \bullet & \bullet \\ 0 & * & * & \bullet & \bullet \\ 0 & 0 & * & \bullet & \bullet \\ 0 & 0 & \bullet & \bullet & \bullet \end{bmatrix}$$

Algorithm 5.13 (The QR algorithm) The "plain vanilla" version of the QR algorithm is as follows. Set $A_0 = A$. For $k = 0, 1, \ldots$ calculate the QR factorization $A_k = Q_k R_k$ (here Q_k is $n \times n$ orthogonal and R_k is $n \times n$ upper triangular) and set $A_{k+1} = R_k Q_k$.

The eigenvalues of A_{k+1} are the same as the eigenvalues of A_k, since we have

$$A_{k+1} = R_k Q_k = Q_k^{-1}(Q_k R_k) Q_k = Q_k^{-1} A_k Q_k, \tag{5.2}$$

a similarity transformation. Moreover, $Q_k^{-1} = Q_k^T$, therefore if A_k is symmetric, then so is A_{k+1}.

If for some $k \geq 0$ the matrix A_{k+1} can be regarded as "deflated", i.e. it has the block form

$$A_{k+1} = \begin{bmatrix} B & C \\ D & E \end{bmatrix},$$

where B, E are square and $D \approx O$, then we calculate the eigenvalues of B and E separately (again, with QR, except that there is nothing to calculate for 1×1 and 2×2 blocks). As it turns out, such a "deflation" occurs surprisingly often.

Technique 5.14 (The QR iteration for upper Hessenberg matrices) If A_k is upper Hessenberg, then its QR factorization by means of the Givens rotations produces the matrix

$$R_k = Q_k^T A_k = \Omega^{[n-1,n]} \cdots \Omega^{[2,3]} \Omega^{[1,2]} A_k\,,$$

which is upper triangular. The QR iteration sets $A_{k+1} = R_k Q_k = R_k \Omega^{[1,2]T} \Omega^{[2,3]T} \cdots \Omega^{[n-1,n]T}$, and it follows that A_{k+1} is also upper Hessenberg, because

$$\begin{bmatrix} * & * & * & * \\ 0 & * & * & * \\ 0 & 0 & * & * \\ 0 & 0 & 0 & * \end{bmatrix} \overset{\times\Omega^{[1,2]T}}{\to} \begin{bmatrix} \bullet & \bullet & * & * \\ \bullet & \bullet & * & * \\ 0 & 0 & * & * \\ 0 & 0 & 0 & * \end{bmatrix} \overset{\times\Omega^{[2,3]T}}{\to} \begin{bmatrix} * & \bullet & \bullet & * \\ * & \bullet & \bullet & * \\ 0 & \bullet & \bullet & * \\ 0 & 0 & 0 & * \end{bmatrix} \overset{\times\Omega^{[3,4]T}}{\to} \begin{bmatrix} * & * & \bullet & \bullet \\ * & * & \bullet & \bullet \\ 0 & * & \bullet & \bullet \\ 0 & 0 & \bullet & \bullet \end{bmatrix}$$

Thus a strong advantage of bringing A to the upper Hessenberg form initially is that then, in every iteration in QR algorithm, Q_k is a product of just $n-1$ Givens rotations. Hence each iteration of the QR algorithm requires just $\mathcal{O}(n^2)$ operations.

Prof. A. C. Hansen

Mathematical Tripos Part II: Michaelmas Term 2024

Numerical Analysis – Lecture 23

Technique 5.15 (The QR iteration for symmetric matrices) We bring A to the upper Hessenberg form, so that QR algorithm commences from a symmetric tridiagonal matrix A_0, and then Technique 5.14 is applied for every k as before. Since both the upper Hessenberg structure and symmetry is retained, each A_{k+1} is also *symmetric tridiagonal* too. It follows that, whenever a Givens rotation $\Omega^{[i,j]}$ combines either two adjacent rows or two adjacent columns of a matrix, the total number of nonzero elements in the new combination of rows or columns is at most five. Thus there is a bound on the work of each rotation that is independent of n. Hence each QR iteration requires just $\mathcal{O}(n)$ operations.

Notation 5.16 To analyse the matrices A_k that occur in the QR algorithm 5.13, we introduce

$$\bar{Q}_k = Q_0 Q_1 \cdots Q_k, \qquad \bar{R}_k = R_k R_{k-1} \cdots R_0, \qquad k = 0, 1, \ldots . \tag{5.3}$$

Note that $\bar{Q}_k$ is orthogonal and $\bar{R}_k$ upper triangular.

Lemma 5.17 (Fundamental properties of $\bar{Q}_k$ and $\bar{R}_k$) *A_{k+1} is related to the original matrix A by the similarity transformation $A_{k+1} = \bar{Q}_k^T A \bar{Q}_k$. Further, $\bar{Q}_k \bar{R}_k$ is the QR factorization of A^{k+1}.*

Proof. We prove the first assertion by induction. By (5.2), we have $A_1 = Q_0^T A_0 Q_0 = \bar{Q}_0^T A \bar{Q}_0$. Assuming $A_k = \bar{Q}_{k-1}^T A \bar{Q}_{k-1}$, equations (5.2)-(5.3) provide the first indentity

$$A_{k+1} = Q_k^T A_k Q_k = Q_k^T (\bar{Q}_{k-1}^T A \bar{Q}_{k-1}) Q_k = \bar{Q}_k^T A \bar{Q}_k \, .$$

The second assertion is true for $k = 0$, since $\bar{Q}_0 \bar{R}_0 = Q_0 R_0 = A_0 = A$. Again, we use induction, assuming $\bar{Q}_{k-1} \bar{R}_{k-1} = A^k$. Thus, using the definition (5.3) and the first statement of the lemma, we deduce that

$$\begin{aligned} \bar{Q}_k \bar{R}_k &= (\bar{Q}_{k-1} Q_k)(R_k \bar{R}_{k-1}) = \bar{Q}_{k-1} A_k \bar{R}_{k-1} = \bar{Q}_{k-1} (\bar{Q}_{k-1}^T A \bar{Q}_{k-1}) \bar{R}_{k-1} \\ &= A \bar{Q}_{k-1} \bar{R}_{k-1} = A \cdot A^k = A^{k+1} \end{aligned}$$

and the lemma is true. □

Remark 5.18 (Relation between QR and the power method) Assume that the eigenvalues of A have different magnitudes,

$$|\lambda_1| < |\lambda_2| < \cdots < |\lambda_n|, \quad \text{and let} \quad \boldsymbol{e}_1 = \textstyle\sum_{i=1}^n c_i \boldsymbol{w}_i = \sum_{i=1}^m c_i \boldsymbol{w}_i \tag{5.4}$$

be the expansion of the first coordinate vector in terms of the normalized eigenvectors of A, where m is the greatest integer such that $c_m \neq 0$.

Consider the first columns of both sides of the matrix equation $A^{k+1} = \bar{Q}_k \bar{R}_k$.

By the power method arguments, the vector $A^{k+1} \boldsymbol{e}_1$ is a multiple of $\sum_{i=1}^m c_i (\lambda_i / \lambda_m)^{k+1} \boldsymbol{w}_i$, so the first column of A^{k+1} tends to be a multiple of $\boldsymbol{w}_m$ for $k \gg 1$. On the other hand, if $\boldsymbol{q}_k$ is the first column of $\bar{Q}_k$, then, since $\bar{R}_k$ is upper triangular, the first column of $\bar{Q}_k \bar{R}_k$ is a multiple of $\boldsymbol{q}_k$.

Therefore $\boldsymbol{q}_k$ tends to be a multiple of $\boldsymbol{w}_m$. Further, because both $\boldsymbol{q}_k$ and $\boldsymbol{w}_m$ have unit length, we deduce that $\boldsymbol{q}_k = \pm \boldsymbol{w}_m + \boldsymbol{h}_k$, where $\boldsymbol{h}_k$ tends to zero as $k \to \infty$. Therefore,

$$A \boldsymbol{q}_k = \lambda_m \boldsymbol{q}_k + o(\mathbf{1}) \, , \quad k \to \infty \, . \tag{5.5}$$

Theorem 5.19 (The first column of A_k) *Let conditions (5.4) be satisfied. Then, as $k \to \infty$, the first column of A_k tends to $\lambda_m \boldsymbol{e}_1$, making A_k suitable for deflation.*

Proof. By Lemma 5.17, the first column of A_{k+1} is $\bar{Q}_k^T A \bar{Q}_k \boldsymbol{e}_1$, and, using (5.5), we deduce that

$$A_{k+1}\boldsymbol{e}_1 = \bar{Q}_k^T A \bar{Q}_k \boldsymbol{e}_1 = \bar{Q}_k^T A \boldsymbol{q}_k \overset{(5.5))}{=} \bar{Q}_k^T [\lambda_m \boldsymbol{q}_k + o(\mathbf{1})] \overset{(*)}{=} \lambda_m \boldsymbol{e}_1 + o(\mathbf{1}),$$

where in $(*)$ we used that $\bar{Q}_k^T \boldsymbol{q}_k = \boldsymbol{e}_1$ by orthogonality of $\bar{Q}$, and that $\bar{Q}_k \boldsymbol{x} = \mathcal{O}(\boldsymbol{x})$ because orthogonal mapping is isometry. □

Remark 5.20 (Relation between QR and inverse iteration) In practice, the statement of Theorem 5.19 is hardly ever important, because usually, as $k \to \infty$, the off-diagonal elements in the bottom row of A_{k+1} tend to zero *much faster* than the off-diagonal elements in the first column. The reason is that, besides the connection with the power method in Remark 5.18, the QR algorithm also enjoys a close relation with *inverse iteration* (Method 5.5).

Let again

$$|\lambda_1| < |\lambda_2| < \cdots < |\lambda_n|, \quad \text{and let} \quad \boldsymbol{e}_n^T = \textstyle\sum_{i=1}^n c_i \boldsymbol{v}_i^T = \sum_{i=s}^n c_i \boldsymbol{v}_i^T \tag{5.6}$$

be the expansion of the last coordinate row vector $\boldsymbol{e}_n^T$ in the basis of normalized *left eigenvectors* of A, i.e. $\boldsymbol{v}_i^T A = \lambda_i \boldsymbol{v}_i^T$, where s is the least integer such that $c_s \neq 0$.

Assuming that A is nonsingular, we can write the equation $A^{k+1} = \bar{Q}_k \bar{R}_k$ in the form $A^{-(k+1)} = \bar{R}_k^{-1} \bar{Q}_k^T$. Consider the bottom rows of both sides of this equation: $\boldsymbol{e}_n^T A^{-(k+1)} = (\boldsymbol{e}_n^T \bar{R}_k^{-1}) \bar{Q}_k^T$.

By the inverse iteration arguments, the vector $\boldsymbol{e}_n^T A^{-(k+1)}$ is a multiple of $\sum_{i=s}^n c_i (\lambda_s/\lambda_i)^{k+1} \boldsymbol{v}_i^T$, so the bottom row of $A^{-(k+1)}$ tends to be multiple of $\boldsymbol{v}_s^T$. On the other hand, let $\boldsymbol{p}_k^T$ be the bottom row of $\bar{Q}_k^T$. Since $\bar{R}_k$ is upper triangular, its inverse $\bar{R}_k^{-1}$ is upper triangular too, hence the bottom row of $\bar{R}_k^{-1} \bar{Q}_k^T$, is a multiple of $\boldsymbol{p}_k^T$.

Therefore, $\boldsymbol{p}_k^T$ tends to a multiple of $\boldsymbol{v}_s^T$, and, because of their unit lengths, we have $\boldsymbol{p}_k^T = \pm \boldsymbol{v}_s^T + \boldsymbol{h}_k^T$, where $\boldsymbol{h}_k \to 0$, i.e.,

$$\boldsymbol{p}_k^T A = \lambda_s \boldsymbol{p}_k^T + o(\mathbf{1}), \quad k \to \infty. \tag{5.7}$$

Theorem 5.21 (The bottom row of A_k) *Let conditions (5.6) be satisfied. Then, as $k \to \infty$, the bottom row of A_k tends to $\lambda_s \boldsymbol{e}_n^T$, making A_k suitable for deflation.*

Proof. By Lemma 5.17, the bottom row of A_{k+1} is $\boldsymbol{e}_n^T \bar{Q}_k^T A \bar{Q}_k$, and similarly to the previous proof we obtain

$$\boldsymbol{e}_n^T A_{k+1} = \boldsymbol{e}_n^T \bar{Q}_k^T A \bar{Q}_k = \boldsymbol{p}_k^T A \bar{Q}_k \overset{(5.7)}{=} [\lambda_s \boldsymbol{p}_k^T + o(\mathbf{1})] \bar{Q}_k = \lambda_s \boldsymbol{e}_n^T + o(\mathbf{1}). \tag{5.8}$$

the last equality by orthogonality of $\bar{Q}_k$. □

Technique 5.22 (Single shifts) As we saw in Method 5.5, there is a huge difference between power iteration and inverse iteration: the latter can be accelerated arbitrarily through the use of shifts. The better we can estimate $s_k \approx \lambda_s$, the more we can accomplish by a step of inverse iteration with the shifted matrix $A_k - s_k I$. Theorem 5.21 shows that the bottom right element $(A_k)_{nn}$ becomes a good estimate of λ_s. So, in the *single shift technique*, the matrix A_k is replaced by $A_k - s_k I$, where $s_k = (A_k)_{nn}$, before the QR factorization:

$$\begin{aligned} A_k - s_k I &= Q_k R_k, \\ A_{k+1} &= R_k Q_k + s_k I. \end{aligned}$$

A good approximation $s_k = (A_k)_{nn}$ to the eigenvalue λ_s generates even better approximation of $s_{k+1} = (A_{k+1})_{nn}$ to λ_s, and convergence is accelerating at a higher and higher rate (it will be the so-called cubic convergence $|\lambda_s - s_{k+1}| \leq \gamma |\lambda_s - s_k|^3$). Note that, similarly to the original QR iteration, we have

$$A_{k+1} = Q_k^T (Q_k R_k + s_k I) Q_k = Q_k^T A_k Q_k,$$

hence $A_{k+1} = \bar{Q}_k^T A \bar{Q}_k$, but note also that $\bar{Q}_k \bar{R}_k \neq A^{k+1}$, but we have instead

$$\bar{Q}_k \bar{R}_k = \textstyle\prod_{m=0}^k (A - s_m I)$$

Prof. A. C. Hansen

Mathematical Tripos Part II: Michaelmas Term 2024

Numerical Analysis – Lecture 24

Multigrid methods The speed of convergence of some iterative methods (Jacobi with relaxation, Gauss–Seidel, etc.) can be increased drastically when the linear system originates in the discretization of PDEs, using *multigrid methods*. Here we look at the system $A\boldsymbol{u} = \boldsymbol{b}$ originating from the 3-point formula for the Poisson equation on an m-grid $\Omega_h = \{ih : 1 \leq i \leq m\}$, $h = 1/(m+1)$, being solved by the weighted Jacobi iteration.

Recall that the matrix A in this case is given by

$$A = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & & \ddots & & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix} \in \mathbb{R}^{m\times m}.$$

The diagonal part of A is $D = 2I$. Thus the weighted Jacobi iterations takes the form:

$$\boldsymbol{u}^{(\nu+1)} = H_\omega \boldsymbol{u}^{(\nu)} + (\omega/2)\boldsymbol{b}$$

where $\nu = 0, 1, \ldots$ is the iteration count, $H = I - D^{-1}A = I - \frac{1}{2}A$, and $H_\omega = \omega H + (1-\omega)I = I - \frac{\omega}{2}A$. The error decay is expressed in terms of the iteration matrix H_ω:

$$\boldsymbol{e}^{(\nu)} = H_\omega^\nu \boldsymbol{e}^{(0)}.$$

We know from the results of Lecture 2 that the eigenvectors and the eigenvalues of H_ω are

$$\boldsymbol{w}^k = \left[\sin i\tfrac{k\pi}{m+1}\right]_{i=1,\ldots,m}, \qquad \lambda_k(\omega) = 1 - 2\omega \sin^2 \tfrac{k\pi}{2(m+1)} \qquad (k = 1, \ldots, m).$$

Consider the choice $\omega = 1/2$; then the eigenvalues of H_ω are $\lambda_k = 1 - \sin^2 \frac{k\pi}{2(m+1)} = \cos^2 \frac{k\pi}{2(m+1)}$. With this choice, the eigenvalues are all positive and decreasing with k, see Figure below.

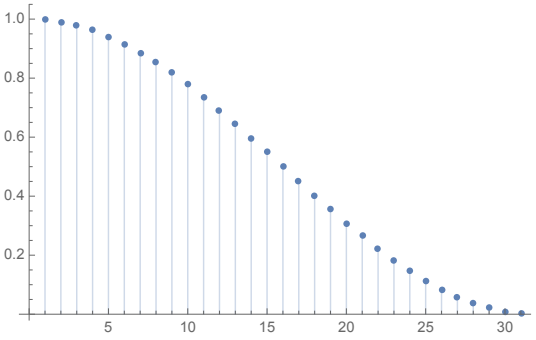


Figure 1: Eigenvalues of H_ω for $\omega = 1/2$ ($m = 31$).

In particular $\rho(H_\omega) = \lambda_1 = \cos^2 \frac{\pi}{2(m+1)} \approx 1 - \frac{\pi^2}{4m^2} < 1$, guaranteeing convergence, although a very slow one when m is large! However, expanding the error with respect to the (orthogonal) eigenvectors we obtain

$$\boldsymbol{e}^{(\nu)} = \sum_{k=1}^{m} a_k^{(\nu)} \boldsymbol{w}^k, \qquad \boldsymbol{e}^{(\nu)} = H_\omega^\nu \boldsymbol{e}^{(0)} \quad \Rightarrow \quad |a_k^{(\nu)}| = |\lambda_k|^\nu |a_k^{(0)}|,$$

i.e. the components of $\boldsymbol{e}^{(\nu)}$ (with respect to the basis of eigenvectors) decay at a different rate for different frequencies $k = 1, \ldots, m$. More precisely, the high frequencies, where k is close to m, will decay faster than the low frequencies, where k is closer to 1. Let us say that $k \in (0, m+1) = (0, \frac{1}{h})$ is *high frequency* (HF) with respect to the grid Ω_h if $kh \geq 1/2$ (i.e., $\frac{m+1}{2} \leq k \leq m$). Then the decay rate for the high frequency components of the error $\boldsymbol{e}$ is at least:

$$\mu_* = |\lambda_{(m+1)/2}| = 1 - \sin^2(\pi/4) = 1/2.$$

Therefore, for the coefficients at the HF components of $\boldsymbol{e}^{(\nu)}$ we obtain

$$|a_k^{(\nu)}| \leq |\mu_*|^\nu |a_k^{(0)}| = \left(\tfrac{1}{2}\right)^\nu |a_k^{(0)}| \ll |a_k^{(0)}|\,,$$

i.e. the Jacobi method converges fast for high frequencies.

The main observation of the multigrid is to note that the low frequencies $k \in (\frac{1}{4h}, \frac{1}{2h})$ with respect to the grid Ω_h become high frequencies for the *coarser grid* Ω_{2h} with step $2h$; indeed for such k we have $k(2h) \geq 1/2$.

The idea of the multigrid method then is that, although the global error may decrease slowly by iteration, its components with high frequencies relative to Ω_h are suppressed very quickly, and that dealing with the remaining components (with low frequencies relative to Ω_h) we can move to the coarse grid Ω_{2h}, where these components (in part) would be of high frequencies, and thus they can be suppressed in a similar way. Therefore, we cover the domain $[0, 1]$ by a range of nested grids, of increasing coarseness, say,

$$\Omega_h \subset \Omega_{2h} \subset \Omega_{4h} \subset \cdots \subset \Omega_{2^j h}\,.$$

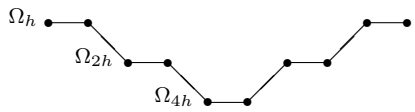
At every Ω_{h_i}, the iterations (Jacobi, or Gauss-Seidel) remove the high frequencies relative to this grid, and we move to Ω_{2h_i}. On the coarsest grid, where the number of variables is small, we can afford to solve the equations with a direct method, by Cholesky, say.

A typical multigrid method can be summarized by the following routine **MGV**, which gives an approximate solution to the linear system $A\boldsymbol{u} = b$, starting from the initial guess $\boldsymbol{u}^0$. We assume below that the size of the linear system is $m = 1/h - 1 = 2^\ell - 1$ for some integer ℓ.

MGV(A,b,$\boldsymbol{u}^0$)

1. If size of A is small enough, use a direct method to solve $A\boldsymbol{u} = b$ and exit. Else:
2. Presmoothing: Perform a small number (typically ≤ 5) of Jacobi or Gauss-Seidel iterations on $A\boldsymbol{u} = \boldsymbol{b}$ starting from $\boldsymbol{u}^0$.
3. Let $\boldsymbol{r} = \boldsymbol{b} - A\boldsymbol{u}$ be the residual, with $\boldsymbol{u}$ from the previous step.
4. Let $I_{2h}^h : \mathbb{R}^{\frac{m+1}{2}-1} \to \mathbb{R}^m$ be an *interpolation operator* that interpolates vectors on the coarse grid Ω_{2h} to vectors on the fine grid Ω_h; and let $R_h^{2h} : \mathbb{R}^m \to \mathbb{R}^{(m+1)/2-1}$ be a *restriction operator* that restricts vectors on the fine grid Ω_h to vectors on the coarse grid Ω_{2h}.
5. Let $\tilde{A} = R_h^{2h} A I_{2h}^h$ which is of size $\approx m/2 \times m/2$.
6. Recurse: let $\tilde{\boldsymbol{\delta}} = \mathbf{MGV}(\tilde{A}, R_h^{2h}\boldsymbol{r}, 0)$ (approximate solution to the residual equation $A\boldsymbol{\delta} = \boldsymbol{r}$ on the coarse grid)
7. Let $\boldsymbol{u} = \boldsymbol{u} + I_{2h}^h \tilde{\boldsymbol{\delta}}$
8. Postsmoothing: apply a few Jacobi or Gauss-Seidel iterations starting from $\boldsymbol{u}$ on $A_h\boldsymbol{u} = \boldsymbol{b}$
9. Return $\boldsymbol{u}$

Remark 5.15 *If we follow the recursive procedure outlined above, then we see that the algorithm starts at the finest grid, travels to the coarsest (where we apply a direct solver), and back to the finest:*



For this reason, the algorithm above describes what is known as a V-cycle, hence the name of the routine.

To make the algorithm above complete, one needs to define the interpolation and restriction operators. A common choice for the interpolation operator is linear interpolation, i.e.,

$$(I_{2h}^h \boldsymbol{v})_{2i} = \boldsymbol{v}_i \quad \text{and} \quad (I_{2h}^h \boldsymbol{v})_{2i+1} = (\boldsymbol{v}_i + \boldsymbol{v}_{i+1})/2$$

for $i \in [0, \frac{1}{2h})$. Assuming zero boundary conditions, this takes the following matrix form:

$$I_{2h}^h = 1/2 \begin{bmatrix} 1 & & & \\ 2 & & & \\ 1 & 1 & & \\ & 2 & & \\ & 1 & 1 & \\ & & \vdots & \\ & & \vdots & \\ & & 1 & 1 \\ & & & 2 \\ & & & 1 \end{bmatrix}.$$

For the restriction operator R_h^{2h}, we take the following averaging operator

$$(R_h^{2h} \boldsymbol{v})_i = \frac{1}{4}(\boldsymbol{v}_{2i-1} + 2\boldsymbol{v}_{2i} + \boldsymbol{v}_{2i+1}) \qquad i \in (0, \tfrac{1}{2h})$$

which corresponds, up to scaling, to the transpose of the linear interpolation operator above; namely $R_h^{2h} = \frac{1}{2}(I_{2h}^h)^T$.

In practice, one may need to apply the routine **MGV** multiple times to improve the accuracy, each time starting from the previously obtained solution.

Matlab demo: Download the Matlab GUI for *Multigrid Methods* from `https://www.damtp.cam.ac.uk/user/hf323/M21-II-NA/demos/multigrid/multigrid.html` and see the effect of multigrid (in comparison with Jacobi and Gauss-Seidel) for solving the Poisson equation with a forcing term f that possesses multiple frequencies.