Dr H. Fawzi

Mathematical Tripos Part II: Michaelmas Term 2023

Numerical Analysis – Lecture 9

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 semi-discretization (also known as 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, assuming zero Dirichlet boundary conditions

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

where A_* is the block TST (Toeplitz Symmetric Tridiagonal) matrix of the five-point scheme:

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

1) 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 $\boldsymbol{\eta} = \mathcal{O}(k^2 + kh^2) = \mathcal{O}(k^2)$. 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,

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

We could also have analyzed the stability of the discretization scheme using Fourier analysis, assuming we 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 $\mu \le \frac{1}{4}$.

2) Crank-Nicolson in 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.

Implementing the Crank-Nicolson method requires solving the linear system $B\boldsymbol{u}^{n+1} = C\boldsymbol{u}^n$ at each step. The matrix $B = I - \frac{1}{2}\mu A_*$ has a structure similar to that of A_*, so we may apply the fast Poisson solver seen in Lectures 3 and 4. The total computational cost per iteration is $\mathcal{O}(M^2 \log M)$ for a $M \times M$ discretization grid.

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!

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

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.

Splitting 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:

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

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 2.25 *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:

$$\mathrm{e}^{B+C} = \sum_{k=0}^{\infty} \frac{1}{k!}(B + C)^k = \sum_{k=0}^{\infty} \frac{1}{k!} \sum_{i+j=k} \binom{k}{i} B^iC^j = \sum_{i,j=0}^{\infty} \frac{1}{i!j!} B^iC^j = \mathrm{e}^B\mathrm{e}^C$$

where in the second step we used the fact that B and C commute. □