Technische Universität Berlin Winter term 2025/26
Institute of Mathematics
Prof. Dr. T. Breiten, Prof. Dr. S. May, Dr. Juliette Dubois

Numerical Mathematics II for Engineers
Tutorial 3

Topics : Well-posedness of PDEs, 2D Finite Differences

Discussion in the tutorials of the week 10–14 November 2025

Disclaimers:

- To test your code, we provide tests. This should help you to locate precisely your errors, and code more efficiently. To run the test, you need to install pytest.
- Exercises should be solved in **fixed groups of 3 students**.
- Hand in the solution in **one folder** labeled **hw[hw_number]_group[group_number]** and containing:
  - **One pdf** for the theoretical questions and comments on the numerical results,
  - **One python file per programming exercise**.
  - Write the group number and all names of your members **in each file**.

Exercise 3.1: Well-posedness of PDEs: uniqueness

We define the domain $\Omega = (0,1) \times (0,1)$, and study two ill-posed PDEs on Ω.

(a) Let $\partial\Omega$ be the boundary of Ω, and $\mathbf{v}$ be the outer normal vector field. Show that if the 2D Poisson equation with Neumann boundary condition

$$-\Delta u(\mathbf{x}) = f(\mathbf{x}), \qquad \text{for } \mathbf{x} \in (0,1) \times (0,1), \tag{1}$$
$$\nabla u(\mathbf{x}) \cdot \mathbf{v}(\mathbf{x}) = g(\mathbf{x}) \qquad \text{for } \mathbf{x} \in \partial\Omega \tag{2}$$

admits a solution u to (1)-(2), then the solution is not unique.

Hint: Study the function $u + c$, where c is a constant.

(b) We assume that the 2D wave equation with no boundary conditions

$$u_{tt}(t,\mathbf{x}) - \Delta u(t,\mathbf{x}) = f(t,\mathbf{x}), \qquad \text{for } (t,\mathbf{x}) \in (0,1) \times (0,1) \times (0,1), \tag{3}$$

admits a solution u. Show that there exist solutions v of the form $v(t,x) = u(t,x) + h(t,x)$, where $h(t,x)$ is **not** constant, neither in t nor in $\mathbf{x}$.

Exercise 3.2: Well-posedness of PDEs: stability

In the previous exercise, we have studied the heat equation

$$u_t(t,x) - u_{xx}(t,x) = 0, \qquad \text{for } (t,x) \in (0,T) \times (0,1), \tag{4}$$

We study now the "backward heat equation"

$$u_t(t, x) + u_{xx}(t, x) = 0, \qquad \text{for } (t, x) \in (0, T) \times (0, 1), \tag{5}$$

with homogeneous Dirichlet conditions $u(t, 0) = 0, u(t, 1) = 0$. We investigate whether this PDE is well-posed in the sense of Hadamard. In particular, we focus on its *stability*, namely that the solution to (5) should depend continuously on the data.

Here, the data is the initial condition u_0. We consider u_0 and a perturbed initial condition $u_{0,\delta} = u_0 + \delta$, and denote by u (respectively u_δ) the solution to the PDE (5) with initial condition u_0 (respectively $u_{0,\delta}$). We say that the solution *depends continuously on the data* if there holds

$$\sup_{(x,t)\in(0,1)\times(0,1)} |u(t, x) - u_\delta(t, x)| < C \sup_{x\in(0,1)} |u_0(x) - u_{0,\delta}(x)| \tag{6}$$

for some C independent of $u_0, u_{0,\delta}$. In the following, we denote by E_u the perturbation in the solution and E_0 the perturbation in the initial data,

$$\begin{aligned} E_u &= \sup_{(x,t)\in(0,1)\times(0,1)} |u(t, x) - u_\delta(t, x)| \\ E_0 &= \sup_{x\in(0,1)} |u_0(x) - u_{0,\delta}(x)|. \end{aligned}$$

(a) Let the initial condition be

$$u(0, x) = u_0(x) = \frac{1}{n} \sin(n\pi x), \quad n \in \mathbb{N} \tag{7}$$

Show that $\sup_{x\in(0,1)} |u_0(x)| \to 0$ when $n \to \infty$.

(b) For the initial condition (7), the exact solution to (5) is given by

$$u(t, x) = \frac{1}{n} \sin(n\pi x) e^{n^2\pi^2 t}. \tag{8}$$

Compute the supremum of $|u_n(t, x)|$ over $(0, 1) \times (0, 1)$ for a fixed n, then its limit when $n \to \infty$.

(c) We choose as perturbed initial data $u_{0,\delta}(x) = u_0(x) + \sin(m\pi x)/m$ for $m \in \mathbb{N}, m \neq n$. What is the solution to (5) with this new initial data?

Hint: Use the fact that the PDE is linear.

(d) Compute the perturbation in the initial data E_0 and in the solution E_u.

(e) Show that the solution to (5) does not depend continuously on the data. Is the PDE well-posed?

Exercise 3.3: Finite Differences in 2D

We solve the 2D Poisson problem with Dirichlet boundary conditions

$$\begin{cases} -\Delta u_i(x, y) = f(x, y) & \text{in } \Omega = (0, 1)^2 \subset \mathbb{R}^2 \\ u_i(x, y) = 0 & \text{on } \Gamma = \partial\Omega \end{cases}$$

with finite differences. The template `2DPoisson_template.py` and the test `2DPoisson_test.py` are provided. We denote by $\bar{\Omega} = \Omega \cup \partial\Omega$ the closure of Ω.

(a) Determine the right-hand sides $f_1(x,y)$ and $f_2(x,y)$ such that the Poisson problem has the following solutions

$$u_1(x,y) = xy + x^3y^3 - x^3y - xy^3 \quad \text{and} \quad u_2(x,y) = \sin(3\pi x)\sin(\pi y).$$

(b) Using the 5-point difference stencil for the Laplace operator and using the lexicographical ordering, write the reduced discrete system for the Poisson problem.

(c) Complete the function `get_mesh(N, xL, xR, yL, yR)` which returns the meshsize `h`, the grid of inner points `X,Y` for the reduced system, and the grid `Xfull, Yfull` of the full domain $\bar{\Omega}$ for plotting. Note that we assume that Ω is a square, meaning that `xR-xL` and `yR-yL` should be equal.

(d) Complete the function `get_matrix_rhs` which returns the reduced matrix A and the right-hand side in lexicographic ordering.

Hint: The matrix A can be defined in a compact way using the Kronecker product. The function `scipy.sparse.kron(A,B)` returns $A \otimes B$, defined by

$$A \otimes B = \begin{pmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & & \vdots \\ a_{n1}B & \dots & a_{nn}B \end{pmatrix}.$$

The reduced matrix can then be written $A = (I_N \otimes S) + (S \otimes I_N)$, where S is the tridiagonal matrix $S = -\frac{1}{h^2}(1,-2,1) \in \mathbb{R}^{N\times N}$.

(e) Complete the function `my_driver` that solves the discrete Poisson problem. The array `solution_full` should include the boundary points and have the shape of the grid `(Xfull, Yfull)` since it is required for the plotting part.

To handle the boundary condition, complete the function `get_indexing` which return two arrays `inner, boundary`, each one having the shape `(Xfull, Yfull)`. The array `boundary` has the value `True` at boundary points and `False` elsewhere. The array `inner` has the value `False` at boundary points and `True` elsewhere.

Plot the solution of the Poisson problem for the right-hand side f_2 with $N = 40$.

Hint: To create and use the indexing function, can follow those steps:

- define a (lambda/handle) function `is_boundary(x,y)` which returns a boolean index with the same shape as (x, y). An entry should be `True` if the point is on the boundary and `False` otherwise.
- use the function `is_boundary` on the grid `(X,Y)` to obtain the boundary index as an `numpy.ndarray` of booleans. You can obtain the interior domain as a boolean index by using the complement operator ~ on the boundary index obtained previously.
- use the boolean indexing to assign the solution of the reduced problem to the interior domain, and the boundary condition to the boundary points.

(f) Plot the convergence curves for the right-hand side f_1 and f_2 using log-log curves and $N = \{40, 80, 160, 320\}$. Which order do you obtain?