Practical Session 2/3
PDE problems and Nievergelt's method

All of these problems are based on the slides provided during the morning sessions. Some additional source code (Python or Matlab) may be provided to help you with the implementation tasks. For each problem, you don't have to implement a parallel version of the PinT algorithm : you may use only one process that does the work of all parallel processes.

Problem 1 *We consider the one-dimensional heat equation,*

$$\partial_t u = \partial_{xx} u + f(x,t), \quad x \in [0,L],\ t \in [0,T] \quad u(0,t) = u(L,t) = 0,\ u(x,0) = u_0(x) \tag{1}$$

and the one-dimensional transport (advection) equation,

$$\partial_t u = -a \partial_x u, \quad x \in [0,L],\ t \in [0,T] \quad u(0,t) = u(L,t)\ u(x,0) = u_0(x) \tag{2}$$

We want to analyze the link between those PDE and the Dahlquist equation, and implement a solver to obtain a numerical approximation of their solution.

1. *Use the method of lines to transform each PDE into a system of ODEs of the form*

$$\frac{d\mathbf{u}}{dt} = A\mathbf{u}(t) + \mathbf{f}(t), \tag{3}$$

   with A a matrix depending on the PDE, and $\mathbf{u}(t)$ a vector containing the discretization of the PDE solution on a given spatial mesh at a given time. You may use second order finite differences to approximate ∂_{xx} and first order upwind for ∂_x. What is the exact solution of (3) ? Is it the exact solution of the PDE ?
2. *What are the eigenvalues of the two A matrices for the heat and transport equation ? In which basis are both matrices diagonal ? Can you comment on the link with the Dahlquist equation, in particular looking at the form of the exact solution of (3) ?*
3. *How can you compute the numerical stability of Backward and Forward Euler, when used to numerically solve (3) ? Retrieve the stability condition for each problem and numerical scheme using the numerical stability condition computed for the Dahlquist equation. Compare with the CFL conditions for the transport equation as described in the slides.*
4. *Implement two generic time integration solvers for (3), using Forward and Backward Euler. The matrix A, the initial condition $u_0(x)$, and the source term $\mathbf{f}(t)$ should be given in arguments of the solver.*
5. *Solve the heat equation using $u_0(x) = 20$ as initial condition, $f(x,t) = 0$, $T = 1/2$ and $L = 1$. Represent the solution for $t \in [1/8, 1/4, 1/2]$.*
6. *Solve the transport equation using $u_0(x) = \sin(x)$, $L = 2\pi$, $a = 1$ and $T = 2\pi$. Compare with the exact solution of the advection equation. Can you find two configurations where the error is dominated either by the space discretization or by the time discretization ?*

Problem 2 *Implementation of Nievergelt's method. We first consider the linear ODE*

$$\frac{du}{dt} = \cos(t)u(t), \quad t \in [0, 2\pi], \quad u(0) = 1, \tag{4}$$

and its numerical solution using Forward Euler.

1. *Compute the exact solution of (4). What is the required time-step size to get an accuracy of $1e^{-3}$ for the numerical approximation with Forward Euler ? Determine an appropriate partition of $[0, 2\pi]$ for the parallel processes.*

2. *Apply Nievergelt's method to solve (4). Use an accuracy of $1e^{-3}$ for the fine solver and $1e^{-1}$ for coarse solver, and the time sub-interval decomposition determined in the first question. Note that each propagator may be a function with specific arguments, depending on the way you implement Nievergelt's method. Also, you can choose the simplest strategy when determining the initial values for the accurate trajectories using $M_n = 2$, one point above, one point below. Do you get back the fine solution ?*

Then, we consider the following non linear ODE :

$$\frac{du}{dt} = \cos(t)e^{-u}, \quad t \in [0, 2\pi], \quad u(0) = \ln(2). \tag{5}$$

4. *Compute the exact solution of (5). What is the required time-step size to get an accuracy of $1e^{-3}$ for the numerical approximation with Forward Euler ? Determine an appropriate partition of $[0, 2\pi]$ for the parallel processes.*
5. *Apply Nievergelt's method to solve (5), using the same implementation as for the linear case. Use an accuracy of $1e^{-3}$ for the fine solver and $1e^{-1}$ for coarse solver, and the time sub-interval decomposition determined in the fourth question. What is the error compared with the fine solution ? How can you make it lower than $1e^{-3}$?*