PYU33C01, S. Hutzler, 2025

Assignment 3: Numerical solution of an ordinary differential equation

The aim of this exercise is to study numerical solutions of the ordinary differential equation (ODE)

$$f(x,t) = dx/dt = (1+t)x + 1 - 3t + t^2, \tag{1}$$

using three numerical schemes of different accuracy.

The solutions of such an ODE can be understood by plotting the so-called *direction field*. This consists of small arrows, of slope dx/dt (as given by the above ODE), plotted onto grid points (t,x) in the x-t plane. (See for example Chapter 3 in "From Calculus to Chaos", D. Acheson, Oxford University Press 1997.)

1. Produce such a direction field for t ranging from 0 to 5 and x from -3 to +3 by evaluating dx/dt at 25×25 grid points. Python commands np.meshgrid(,) and ax.quiver(,,,) are useful for this. (You can google how they work.)

2. Pick as a starting value $x(t = 0) = 0.0655$ and solve eqn.(1) using the simple Euler method with step size 0.04. (Feel free to use some of the python code presented in the lecture, and available on Blackboard, for solving ODEs.) Plot your numerical solution, together with the direction field. The point $x(0) = 0.0655$ was deliberately chosen to be close to the critical value of $x_c = 0.065923...$ which separates the solutions which eventually tend to $+\infty$ ($x(0) > x_c$), from those which eventually tend to $-\infty$ ($x(0) < x_c$). How does your simple Euler solution behave?

3. Repeat this calculation using both improved Euler method and Runge-Kutta method (step size 0.04 in both cases). (Again feel free to use part of the code provided on Blackboard.) How do your new numerical solutions behave? Reduce the step size to 0.02, what do you observe now when you run the three different integration methods? What is the benefit in using accurate integration schemes?

4. Using your Runge-Kutta method for a step size of 0.04, find a starting value, $x(t = 0)$, so that $-2.0 \leq x(t = 5) \leq -1.9$.

Report submission (as pdf) **via Blackboard**. Briefly introduce the numerical methods, then detail your findings using relevant graphs. Submit your Python code as a *.py* file. It should also print out your name and student number onto the screen when executed.