**Assignment #2. Design of ODE solver and Time-dependent partial differential equations.**

In this assignment we will consider the design of an ode solver, together with the approximation and analysis of time-dependent partial differential equations, that is, equations containing both temporal and spacial partial derivatives.

We will begin with the analysis of purely parabolic (diffusion) and purely hyperbolic (advection) equations. Then, we will consider an equation of a mixed type, namely, the viscous Burger's equation, which is a model non-linear advection-diffusion (mixed) equation. Partial differential equations of the latter type appear often in engineering disciplines, most famous example being the system of Navier–Stokes equations, which universally utilized for modeling of fluid flows at most scales and regimes.

### Exercise 1: Design an ODE solver

We wish to design a RK-solver in Python/Matlab that can solve systems of first order ODE's in the form of the general initial value problem

$$y' = f(t, y), \quad y(t_0) = y_0.$$

The solution method should be able to change step size, using a combined absolute and relative error tolerance `tol` that is given by

$$\texttt{tol} = \texttt{reps}\ ||y||\ + \texttt{aeps}$$

The stepsize control must be based on keeping an estimate of the local error smaller than `tol`. Some guidelines for devlopment of aa RK scheme, e.g. see Chapter 5.5 in the notes by Engsig-Karup & Thomasen (2018).

1) For testing we use a model for flame propagation (see the mathworks blog that contains a walkthrough) that is given by

   $$y' = y^2 - y^3, \quad y_0 = \delta, \quad 0 \le t \le 2/\delta$$

   where $y(t)$ represents the radius of the ball and $\delta$ is a small value. Try $\delta = 0.02$ and se if you can reduce it to 0.0001 within a relative tolerance of $10^{-4}$ (you may decrease this, but this requires more work). Results can be compared to a ODE solver provided in Python/Matlab. Can you apply the Picard-Lindeloff theorem to understand if a unique solution exists?

2) Choose a pair of method of order 2 and 3 and find the formula for the error estimator

   $$E = h||\sum_{j=1}^{3} \hat{b}_j f(t_j, k_j)||$$

   Use this formulation to construct a (simple) step-size control that attempts to satisfy the condition

   $$E < \texttt{tol}$$

   The method should be tested for suitable chosen values of `reps` and `aeps`. You can measure the efficiency of the strategy by counting the number of times it is necessary to evaluate the right-hand side function or the number of steps used.

**Exercise 2: Purely parabolic model problem**

Consider a one-dimensional unsteady heat diffusion problem with Dirichlet boundary conditions:

$$\begin{aligned} \partial_t u(x,t) &= \epsilon \partial_{xx} u(x,t), \qquad -1 < x < 1, \quad t > 0, \\ u(-1,t) &= g_L(t), \qquad t > 0, \\ u(1,t) &= g_R(t), \qquad t > 0, \\ u(x,0) &= \eta(x), \qquad -1 \le x \le 1. \end{aligned} \tag{1}$$

One can check that a function of the form

$$u(x,t) = \sum_{i=1}^{N} \exp(-\epsilon \alpha_i^2 t)[a_i \cos(\alpha_i x) + b_i \sin(\alpha_i x)] \tag{2}$$

for an arbitrary natural number $N$ and real constants $\alpha_i$, $a_i$, $b_i$, $i = 1, \ldots, N$, verifies the equations (1) for appropriately chosen $g_L$, $g_R$, and $\eta$, which may be used for the verification purposes. You can take $N = 3$, $\alpha_1 = 1$, $\alpha_2 = 4$, $\alpha_3 = 16$, $a_i = 1$, $b_i = 0$. The diffusion constant $\epsilon$ can be set to a value of 0.1.

1) Approximate the temporal derivative using the forward finite differencing [equation (1.1) in LeVeque], and the spacial derivative using the central differencing [equation (1.13) in LeVeque]; the resulting scheme is known as Forward Time and Central Space (FTCS). You may assume that the grid points are uniformly distributed as $x_i = -1 + hi$, $i = 0, 1, ..., N$ in space with grid increment $h = \frac{2}{N}$ and $t_j = 0 + kj$, $i = 0, 1, ..., M$ in time with time step size $k$.

2) Derive a criterion for $h$, $k$ in terms of $\varepsilon$ for the time-discretization method to be absolutely stable (See Section 9.3 in LeVeque).

3) Derive a criterion for $h$, $k$ in terms of $\varepsilon$ for the discrete maximum principle to hold. Compare it with the criterion derived in 2).

4) Using *either* the maximum principle or Lax-Richtmyer theorem, establish the convergence of this scheme as $k \to 0$ when $h(k)$ satisfies the stability requirements in 2) or 3).

5) Write a Python/Matlab program implementing FTCS scheme and demonstrate convergence of your solver with the rate predicted by the theory.

**Exercise 3: Purely hyperbolic model problem**

Consider a one-dimensional unsteady advection problem

$$\begin{aligned} \partial_t u(x,t) + a \partial_x u(x,t) &= 0, \qquad -1 \le x \le 1, \quad t > 0, \\ u(-1,t) &= u(1,t) \qquad t > 0, \\ u(x,0) &= \eta(x), \qquad -1 \le x \le 1. \end{aligned} \tag{3}$$

where $a$ is a constant advection speed. Owing to the constant advection speed and the periodic boundary conditions the problem admits solutions of the form $u(x,t) = f(x - at)$ where $f(\cdot)$ is a (differentiable) function determined by the initial conditions. For the purpose of verifying the correctness of your code you can take $f(y) = \sin(2\pi y)$ and $a = 0.5$.

1) It can be shown that the FTCS discretization scheme for equations (3) is unconditionally unstable for approximating solutions to this PDE with periodic boundaries. Show that by instead approximating the spatial derivative using information upwind to the propagation direction (a Forward Time Backward Space (FTBS) scheme) we obtain a conditionally stable numerical scheme.

2) Write a Python/Matlab program implementing the FTBS scheme and demonstrate convergence of your solver with the expected order of accuracy.

3) Determine the phase (dispersion) error and amplitude (diffusion) error per time step using the proposed numerical scheme. Then, predict the amount of diffusion and dispersion which will be experienced when the numerical scheme is employed to solve the advection equation with initial condition $u(x,t) = \sin(2\pi x)$ on a periodic domain with $c_r = a\frac{\Delta t}{\Delta x} = 0.8$ after 40 wave periods. It can be assumed that the wave can be resolved with 100 points per wave length. Confirm the prediction by visualization of your computed results.

**Exercise 4: Non-linear advection-diffusion equation**

In this part of the assignment we are going to numerically solve an equation, which has parabolic and hyperbolic parts, namely, the viscous Burger's equation:

$$\begin{aligned} \partial_t u(x,t) + \tfrac{1}{2}\partial_x (u(x,t)^2) &= \epsilon \partial_{xx} u(x,t), \qquad -1 < x < 1, \quad t > 0, \\ u(-1,t) &= g_L(t), \qquad t > 0, \\ u(1,t) &= g_R(t), \qquad t > 0, \\ u(x,0) &= \eta(x), \qquad -1 \le x \le 1. \end{aligned} \tag{4}$$

It may be assumed that the solution $u(x,t)$ is differentiable and that the diffusion coefficient $\varepsilon$ is a constant.

For appropriate initial and boundary conditions, the following function is a solution of the initial-boundary value problem (4) (check this!):

$$u(x,t) = -\tanh\left(\frac{x + 0.5 - t}{2\epsilon}\right) + 1, \tag{5}$$

and therefore can be used for verification of numerical schemes.

1) Based on your experience with Exercises 1 and 2, try to come up with a stable numerical scheme for the numerical approximation of the initial-boundary value problem (4).

2) Write a Python/Matlab program to solve the model problem and demonstrate convergence of your solver on the problem corresponding to the exact solution (5).

3) Now set up a different initial/boundary value problem, with homogeneous boundary conditions $g_L(t) = g_R(t) = 0$, and initial condition $\eta(x) = -\sin(\pi x)$. Solve the problem for $\epsilon = 0.01/\pi$ on a uniform mesh with the objective to estimate the value $\partial_x u|_{x=0} = -152.00516$ of the $x$-derivative at time $t = 1.6037/\pi$.

4) Compared to the uniform grid solution determined in 3), demonstrate that it is possible to reduce the number of mesh points while maintaining accuracy in your solution for a certain level of accuracy, e.g., by employing non-uniform grids and/or high-order accurate discretization methods.

5) Make sure to include relevant plots of your final solution and results.

**Exercise 5: Physics-Informed Neural Networks (PINNs)**

In this part, we will consider a new tool from the scientific machine learning toolbox, namely, Physics-Informed Neural Networks (PINNs). Solve at least one of the three PDEs from Exercises 2 (parabolic), 3 (hyperbolic) and 4 (mixed) using a physics-informed Neural Network. You can use an existing framework in Tensorflow/Keras, PyTorch, Julia, etc. to complete this exercise. A python-based implementation in Jupyter notebook (guidelines for installation is provided in a separate note) is recommended. Some details needed for this exercises are given in lecture slides of the course.

1) For the parabolic equation problem in exercise 2, make plots of the solution. What is the neural network architecture used to solve the problem? Can you produce a solution with same accuracy as the numerical solution produced using finite differences? ( Hint: experiment with hyperparameters of the neural network, e.g. number of neurons, hidden nodes, etc.)

2) For the hyperbolic equation problem in exercise 3, formulate the loss function to take into account the boundary condition and the residual of the PDE and train inside the time windows. Discuss if it is possible to find a solution outside the time window used for training.

3) For the nonlinear advection-diffusion equation problem in exercise 4, solve the problem using PINN and estimate the gradient as accurately as possible.

4) Make sure to include relevant plots of your final solutions and results as well as detail what hyperparameters were used to solve the problems.

Communicate your results in a small written report. Make sure to describe and comment on important details and findings. Include sufficiently details for the reader to both reproduce and understand your reported results and conclusions. The assignment handin should include the code produced (as a zip) and a pdf containing the written report.

**Deadline for this assignment is Sunday, May 18, 2025, 23:59.**