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 12

Topics : Numerical fluxes for non-linear equations, non-conservative schemes, Riemann problems for Burgers' equation

Discussion in the tutorials of the week 26 – 30 January 2026

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 12.1: Programming: the Burgers' equation

For this exercise, use the template `FV_Burgers.py`

We consider Burgers' equation

$$u_t + \left(\frac{1}{2}u^2\right)_x = 0$$

in the domain $(-1, 3)$ with transmissive boundary conditions[1] and with the initial conditions

- Shock:
$$\text{(S)} \qquad u_0(x) = \begin{cases} 1 & \text{if } x < 0, \\ 0, & \text{otherwise} \end{cases} \qquad T_{\text{end}} = 2,$$
- Rarefaction:
$$\text{(R)} \qquad u_0(x) = \begin{cases} 0 & \text{if } x < 0, \\ 1, & \text{otherwise,} \end{cases} \qquad T_{\text{end}} = 1.5,$$
- Rarefaction and shock:
$$\text{(RS)} \quad u_0(x) = \begin{cases} 1 & \text{if } 0 < x < 1, \\ 0, & \text{otherwise.} \end{cases} \qquad T_{\text{end}} = 1.5,$$

[1]The transmissive boundary condition ($\partial u/\partial x = 0$ at the boundaries) is already implemented in the code. Note how this differs from the periodic boundary conditions from the previous homework.

Our goal is to solve the three problems (S), (R), (RS) for $T \leq 2$ numerically using

- *Godunov method,*
- *Lax-Friedrichs scheme (LF-FV version),*
- *local Lax-Friedrichs scheme.*

The Godunov scheme is already implemented. For all schemes we will use a CFL number of 0.8.

(a) Implement the LF-FV scheme and the LLF scheme. You can reuse your implementation of LF-FV of last homework if it was correct.

(b) Observe how the solution evolves in time for all three cases. Plot the exact and computed solution at final time for the case (RS) with all three methods and $N = 80$. Identify the shock and rarefaction waves in the plots.

(c) Write a function `main_graph` which calls the `my_driver` routine for the three methods and draw the solution for all four methods in the same plot. Note that the computed solution and the mesh are outputs argument to `my_driver`. Plot the solution for cases (S), (R) and (RS) for $N = 40$. How do the schemes compare concerning the level of numerical diffusion?

(d) We compare the error in both the L^1 and the L^∞ norm for the cases (R) and (S). Use $N = 40$ and Nrefine=6. Compare the three fluxes. What convergence rates do you observe? Do you observe differences between the convergence rates for the cases (R) and (S)?

(e) We now consider the Burgers' equation in the domain $(0, 1)$ with periodic boundary conditions and with smooth initial conditions

$$u_0(x) = \sin(2\pi x).$$

Extend the code to this new case with the name `testfunction="sine"`. Show the computed solution at the final time $T_{\text{end}} = 0.25$ using either the LLF, LF-FV or Godunov flux. Is the solution at final time still smooth? What is the difference with the linear advection equation with similar initial and boundary conditions?

Hint: Define a new `testfunction` in `get_testproblem` and adapt the boundary conditions based on the choice of `testfunction`. You don't need to compute the exact solution.

Exercise 12.2: A non-conservative flux

For this exercise, use the template `non_conservative.py`

We consider the same problem as in Exercise 12.1: Burgers' equation in the domain $(-1, 3)$ with transmissive boundary conditions and with the initial conditions (S), (R) and (RS). We solve the problem with the following non-conservative scheme

$$U_j^{n+1} = U_j^n - \frac{\Delta t}{\Delta x} U_j^n \left\{ U_j^n - U_{j-1}^n \right\}. \tag{1}$$

This scheme seems like a reasonable extension of the upwind scheme to Burgers, considering that the transport velocity for Burgers is given by u. However, one can show that the scheme does *not conserve mass*: there holds

$$\Delta x \sum_j U_j^{n+1} = \Delta x \sum_j U_j^n + \mathcal{O}(\Delta x) \sum_j (U_j^n - U_{j-1}^n)^2. \tag{2}$$

(a) Implement in `my_driver` the non-conservative scheme (1).

(b) Plot the exact and computed solution at final time for all three cases and $N = 80$. Does the scheme produce correct results? In particular, is the shock speed correct?

(c) Show (with pen and paper) that the equation (2) holds.

Exercise 12.3: Riemann problems for Burgers' equation

We consider Burgers equation

$$u_t + \left(\frac{1}{2}u^2\right)_x = 0.$$

Solve the equation (i.e. find the entropy solution) for the following initial data

(a)

$$u(x,0) = \begin{cases} -\frac{1}{2} & \text{for } x < 0, \\ 1 & \text{for } x \geq 0. \end{cases}$$

(b)

$$u(x,0) = \begin{cases} 1 & \text{for } x < 0, \\ -\frac{1}{2} & \text{for } x \geq 0. \end{cases}$$