# Pre class assignment day 13

*Ref. Toro Riemann Solver Ed. 3*

**1. Finish the in class assignment 12:**

Implement the Lax-Friedrich scheme in conservative form, see Toro p185

Simple algebraic manipulations of this expression lead to a conservative version of the Lax–Friedrichs scheme for systems

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n + \frac{\Delta t}{\Delta x}\left[\mathbf{F}_{i-\frac{1}{2}} - \mathbf{F}_{i+\frac{1}{2}}\right] , \qquad (5.76)$$

with the Lax–Friedrichs intercell flux given by

$$\mathbf{F}_{i+\frac{1}{2}}^{\mathbf{LF}} = \frac{1}{2}(\mathbf{F}_i^n + \mathbf{F}_{i+1}^n) + \frac{1}{2}\frac{\Delta x}{\Delta t}(\mathbf{U}_i^n - \mathbf{U}_{i+1}^n) . \qquad (5.77)$$

Start with your existing code and make sure to write generic code, i.e., implement a function for F^LF_i+/-1/2 and one function for the F^n_i. Before starting to code, discuss with your group how array indices/slices are linked to the equations.

Now solve the inviscid Burgers equation, see below.

196 5 Notions on Numerical Methods

**5.5.2 The Inviscid Burgers Equation**

Our Test 3 consists of the inviscid Burgers equation

$$u_t + f(u)_x = 0 , \quad f(u) = \frac{1}{2}u^2 \qquad (5.112)$$

in the domain $[0, \frac{3}{2}]$ with initial conditions

$$u(x,0) = \begin{cases} -\frac{1}{2} & \text{if } x \le \frac{1}{2} , \\ 1 & \text{if } \frac{1}{2} \le x \le 1 , \\ 0 & \text{if } x \ge 1 . \end{cases} \qquad (5.113)$$

We solve this problem numerically on a domain of length $L = 1.5$ discretised by $M = 75$ equally spaced cells of width $\Delta x = 0.02$; the CFL coefficient used is 0.8. Fig. 5.15 shows computed results (symbol) along with the exact (line) solution, for the Godunov and Lax–Friedrichs schemes at time $t = 0.5$ units (32 time steps). Two new features are now present in solving non–linear PDEs.

*Hint (as announced in class):* A possible solution to calculating the fluxes could look like:

```
# Toro (5.77)
FLF = 0.5*(getFlux(u[:-1]) + getFlux(u[1:])) + 0.5*dx/dt*(u[:-1] - u[1:])
# Toro (5.76)
u[1:-1] += dt/dx * (FLF[:-1] - FLF[1:])
```

where getFlux(u) returns 0.5*u^2 for Burgers equation.

Plot and describe your results. How does your result compare to the plot that Toro shows (Fig. 5.15).

***Please submit your solution in an jupyter notebook.***

## 2. Reading

As discussed in class, one problem in solving hyperbolic PDEs with higher-order (e.g., second order Lax Wendroff) methods are spurious oscillation at discontinuities.

In class 13, we will address this issue by making use of so called flux limiters that restrict the fluxes in your numerical solution to remain within physically reasonable restrictions (e.g., while the total amount of cars in the traffic jam analogy is bound to remain constant by using a conservative scheme, there cannot be any negative amount of cars).

In preparation for this class, please read the following Sections in Toro (focus on the fundamental concepts described and less about the mathematical details and derivations. It could also be helpful to briefly look at other [potentially less detailed] resources on those topics, e.g., Wikipedia):

- Foreword of chapter 13 (Higher-order and TVD methods for scalar eqn)
- Section 13.1 Introduction
- Section 13.2 Basic properties of selected schemes
- Section 13.6 Total Variation Diminishing (TVD) Methods
- Section 13.7.2 The general flux limiter approach
- Section 13.7.3 TVD Upwind flux limiter schemes

and answer the following questions in your own words:

- What the basic idea behind TVD?
- What is a “TVD region”?
- What is the meaning of the r value in the flux limiters?

- Do you see a pitfall implementing eq. (13.139) in your code?

Finally, please write down two questions that you have related to the reading (and as usual any other question you have pertaining to the general topic).

***Please copy these question to the bottom of the notebook you used above and answer them in the notebook.***

***The deadline for this assignment is 2 hours before class, i.e., 11 Oct 1pm.***