MAE 4990: CFD Training - Module A.1

1$^{\text{st}}$–order FV for Linear Advection Equation

Module Goals

By the end of this module, you will be able to:

1. Solve the 1D linear advection equation using a **first-order** finite-volume method.
2. Implement the concept of a **numerical flux** and use the **Local Lax–Friedrichs (LLF)** flux.
3. Choose a stable timestep using a **CFL** condition on both **uniform** and **nonuniform** grids.
4. Verify results by comparing to the **exact solution** and reporting an error at a final time.

1 Governing Equation and Exact Solution

We consider the 1D linear advection equation

$$u_t + a\,u_x = 0, \tag{1}$$

where a is a constant advection speed.

For a given initial condition $u(x, 0) = u_0(x)$, the exact solution is a translation:

$$u(x, t) = u_0(x - at). \tag{2}$$

If periodic boundary conditions are used on $x \in [0, L]$, interpret $x - at$ modulo L.

2 Grid and Data Locations (Uniform and Nonuniform)

This module uses the 1D FV indexing:

$$\text{cell } i: \quad [x_{i-1/2},\, x_{i+1/2}], \qquad \text{cell center: } x_i.$$

Cell widths may vary (nonuniform grid):

$$\Delta x_i = x_{i+1/2} - x_{i-1/2}, \qquad i = 1, \ldots, N_x. \tag{3}$$

Uniform grid

A uniform grid has constant Δx:

$$x_{i+1/2} = i\,\Delta x, \quad i = 0, \ldots, N_x, \qquad \Delta x = \frac{L}{N_x}.$$

Nonuniform grid

A nonuniform grid has variable Δx_i but still satisfies

$$0 = x_{1/2} < x_{3/2} < \cdots < x_{N_x+1/2} = L.$$

Your solver must use Δx_i in the update and use $\min_i \Delta x_i$ in the CFL condition.

3 Finite-Volume Semi-Discrete Form

Let $\bar{u}_i(t)$ denote the cell average in cell i. The conservative finite-volume semi-discrete update is

$$\frac{d\bar{u}_i}{dt} \;=\; -\frac{1}{\Delta x_i}\left(\hat{f}_{i+1/2} - \hat{f}_{i-1/2}\right), \qquad i = 1, \ldots, N_x, \tag{4}$$

where $\hat{f}_{i\pm1/2}$ are **numerical fluxes** at faces.

3.1 Numerical flux concept

The physical flux for linear advection is

$$f(u) = au. \tag{5}$$

In a finite-volume method, the face flux must be computed from left/right states:

$$\hat{f}_{i+1/2} = \hat{f}(u_L, u_R).$$

In this module (first order), use piecewise-constant face states:

$$u_L = \bar{u}_i, \qquad u_R = \bar{u}_{i+1}. \tag{6}$$

3.2 Local Lax–Friedrichs (LLF) flux

The LLF flux is

$$\hat{f}(u_L, u_R) = \frac{1}{2}\Big(f(u_L) + f(u_R)\Big) - \frac{1}{2}\alpha_{i+1/2}(u_R - u_L), \tag{7}$$

where $\alpha_{i+1/2}$ is an estimate of the maximum wave speed at the face. For linear advection with constant a, use

$$\alpha_{i+1/2} = |a|. \tag{8}$$

4 Time Integration and CFL Condition

You will advance the semi-discrete ODE system using your SSPRK2 or SSPRK3 routines from Module 0.

4.1 CFL timestep

Choose a timestep Δt using

$$\Delta t \;=\; \text{CFL}\, \min_i \frac{\Delta x_i}{|a|}, \tag{9}$$

where $\text{CFL} \in (0, 1]$ is user-selected (typical values: 0.2–0.9).

5 Boundary Conditions

Implement **periodic** boundary conditions:

$$\bar{u}_0 \equiv \bar{u}_{N_x}, \qquad \bar{u}_{N_x+1} \equiv \bar{u}_1,$$

so that face fluxes at the domain boundaries can be computed using wrapped indices. Here, $\bar{u}_0$ denotes the cell-averaged value of the ghost cell located just outside of to the left boundary at $x_{\frac{1}{2}}$ while $\bar{u}_{N_x+1}$ denotes the cell-averaged value of the ghost cell located just outside of to the right boundary at $x_{N_x+\frac{1}{2}}$. For example:

$$\begin{aligned} \hat{f}_{\frac{1}{2}} &= \hat{f}(\bar{u}_0, \bar{u}_1) = \hat{f}(\bar{u}_{N_x}, \bar{u}_1) \\ \hat{f}_{N_x+\frac{1}{2}} &= \hat{f}(\bar{u}_{N_x}, \bar{u}_{N_x+1}) = \hat{f}(\bar{u}_{N_x}, \bar{u}_1) \end{aligned} \tag{10}$$

and for all other interfaces:

$$\hat{f}_{i+\frac{1}{2}} = \hat{f}(\bar{u}_i, \bar{u}_{i+1})$$

6 Required MATLAB Interfaces

Use the time stepping infrastructure from Module 0. Implement a 1D advection RHS:

- `R = rhs_advection_1d(U, t, geom, params)`

where, in this example, `geom` contains at least `dx` (cell widths) and `params` contains `a` and `CFL`. It is up to you, how you actually set this structure. It is fine as long as your implementation works (and simple!).

Inside `rhs_advection_1d`, assemble $\mathcal{L}(\bar{u}, t)$ using

$$\frac{d\bar{u}_i}{dt} = -\frac{1}{\Delta x_i}\left(\hat{f}_{i+1/2} - \hat{f}_{i-1/2}\right),$$

with LLF flux and first-order face states $u_L = \bar{u}_i$, $u_R = \bar{u}_{i+1}$.

7 What to Run and What to Report

Use $x \in [0, 1]$, periodic boundary conditions, and $a = 1$.

Case 1

Use a smooth initial condition

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

Case 2

Use a non-smooth top-hat (square) pulse as a prototype discontinuity (shock-like feature):

$$u_0(x) = \begin{cases} 1, & 0.25 \le x \le 0.5, \\ 0, & \text{otherwise.} \end{cases}$$

Final time

Try running all cases up to $T \in [0.25, 1.5]$. For example, use $T = 0.5$ (half period) and $T = 1$ (full period) for $a = 1$ on $x \in [0, 1]$. It would be instructive if you run for different $T \in [0.25, 1.5]$ and plot the solution. So, try different intermediate values for T.

Uniform vs nonuniform grids

Run the smooth case on:

- a uniform grid with $N_x = 2^{i+3}$ for $i = 1, 2, \ldots, 9$,
- a nonuniform grid with the same number of cells $N_x = 2^{i+3}$ for $i = 1, 2, \ldots, 9$.

Run the discontinuous case on $N_x = 100, 500, 1000, 2500, 5000$ for both uniform and nonuniform grids.

Outputs for your report

Include:

- Plots of $u(x, 0)$, $u(x, T)$ and $u_{exact}(x, T) = u_0(x - aT)$ overlaid (use cell centers for plotting).
- For the smooth case on uniform grid: the final-time error norm (choose one and state it), e.g.

$$\|e\|_2 \approx \left(\sum_{i=1}^{N_x} \left(\bar{u}_i(T) - u_{\text{exact}}(x_i, T) \right)^2 \Delta x_i \right)^{1/2}.$$

  use this to create an error table (like you did in MAE 3100)

- A brief statement on how the FV method captures smooth solution as the number of cells are increased.
- A brief statement on how the FV captures discontinuities and what happens as the number of cells are increased.
- For each case, try $CFL = 1.5$. What happens in this case?

8 Minimal MATLAB Snippets (reference)

Uniform and nonuniform grids (face coordinates)

Store face locations $x_{i+1/2}$ for $i = 0, \ldots, N_x$.

```
% Domain and parameters
L = 1.0;
Nx = 100; % number of cells
a = 1.0;
CFL = 0.5;
%
uniform = true; % false for nonuniform

if(uniform)
```

```
  % ---- uniform faces ----
  xf = linspace(0,L,Nx+1).'; % faces: x_{1/2}..x_{Nx+1/2}
else
  % ---- example nonuniform faces (monotone mapping) ----
  s = linspace(0,1,Nx+1).';
  beta = 2.0; % stretching strength
  %
  xf = L*(exp(beta*s)-1)/(exp(beta)-1); % faces: x_{1/2}..x_{Nx+1/2}
  %
end
%
dx = diff(xf); % dx_i
%
% cell centers for plotting
xc = 0.5*(xf(1:end-1)+xf(2:end));
% calculate the initial condition using xc.
```

CFL timestep

```
dt = CFL*min(dx)/abs(a);
```

9 Submission

Submit a short, well-formatted **PDF report** that includes your results (figures, brief captions, and any requested error values). This report will become part of your end-of-semester compilation.