Hyperbolic PDEs and Finite-Volume Methods III

Ammar H. Hakim (ammar@princeton.edu) [1]

[1]Princeton Plasma Physics Laboratory, Princeton, NJ

Princeton University, Course AST560, Spring 2021

The Riemann Problem for hyperbolic PDEs

The Riemann problem is a class of *initial value* problems for a hyperbolic PDE

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = 0.$$

on $x \in [-\infty, \infty]$ with initial conditions

$$\mathbf{Q}(x, 0) = \mathbf{Q}_R \quad x > 0$$
$$\mathbf{Q}(x, 0) = \mathbf{Q}_L \quad x < 0$$

where $\mathbf{Q}_{L,R}$ are *constant* initial states.

- Fundamental mathematical problem in theory of hyperbolic PDEs: brings out the key structure of the nonlinear solutions of the system.
- For some important systems like (relativisitic) Euler equations, ideal MHD the Riemann problem can be solved *exactly* (modulo some nonlinear root-finding).
- Good test for shock-capturing schemes as it tests ability to capture discontinuities and complex non-linear phenomena.

Essence of the finite-volume method

Consider a PDE of the form (non necessarily hyperbolic)

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = 0.$$

Now make a grid with cells $I_j = [x_{j-1}, x_{j+1/2}]$ and $\Delta x = x_{j+1/2} - x_{j-1/2}$. The finite-volume method *usually* evolves the cell-averages of the solution:

$$\frac{\partial \mathbf{Q}_j}{\partial t} + \frac{\mathbf{F}_{j+1/2} - \mathbf{F}_{j-1/2}}{\Delta x} = 0$$

where

$$\mathbf{Q}_j(t) \equiv \frac{1}{\Delta x} \int_{I_j} \mathbf{Q}(x,t)\, dx$$

are the *cell-averages* and

$$\mathbf{F}_{j\pm1/2} \equiv \mathbf{F}(\mathbf{Q}_{j\pm1/2})$$

are at cell interfaces.

Essence of the finite-volume method

The finite-volume method *usually* evolves the cell-averages of the solution:

$$\frac{\partial \mathbf{Q}_j}{\partial t} + \frac{\mathbf{F}_{j+1/2} - \mathbf{F}_{j-1/2}}{\Delta x} = 0$$

This equation is an *exact* evolution equation for the cell-averages.However, notice that

- We only know cell-averages $\mathbf{Q}_j$ in each cell; we *do not* know the *cell-edge* values $\mathbf{Q}_{j\pm1/2}$ needed to compute the flux $\mathbf{F}_{j\pm1/2}$.
- The finite-volume method consists of determining these *edge values* and *constructing a numerical-flux* so the cell-averages can be updated.
- Time-stepping can be done with a ODE solver (method-of-lines) or using a *single-step* method (fully discrete scheme).

Cell-averages v/s cell-center values

- Typically, finite-volume schemes evolve the cell-average values; finite-difference schemes evolve nodal values.
- For some low-order (first and some second-order) schemes the *forms* of the scheme may look superficially the same. However, this is not true in general and one must *very carefully* distinguish between cell-average and point-wise values. Otherwise incorrect schemes can result that "look okay" but do not achieve full accuracy.
- What we evolve (cell-average, nodal values or in DG moments or interior node values) is called the *solution representation*.

Remember Your Representation

When studying or designing numerical schemes **never** confuse one solution representation for another.

Finite-Volume method computes *mean* of flux gradient

To derive the basic form of the scheme we did

$$\frac{1}{\Delta x}\int_{I_j}\frac{\partial F}{\partial x}\,dx = \frac{F_{j+1/2}-F_{j-1/2}}{\Delta x}.$$

- Notice that the left-hand side is the *mean* of the flux gradient in the cell I_j
- Hence, in effect, the FV scheme is computing the *mean* of the flux gradient and not the flux gradient itself. This is then used to update *cell-average* of the solution.
- This is important to remember when computing source terms; making plots or computing diagnostics. (Remember Your Representation!).

Example: How to compute mean of *product* of values?

- Given cell-average values Q_j and V_j how can you compute cell-average value $(QV)_j$?
- Clearly, $(QV)_j$ is not the same as $Q_j V_j$.
- In general, depending on the order of the scheme one has to *recover* $Q(x)$ and $V(x)$ to sufficiently high order in a cell, multiply them and then compute the average of the product. Potential complications when solutions are not smooth enough.
- Almost never done! However, it may be important when trying to extract delicate information from simulations like turbulence spectra etc.

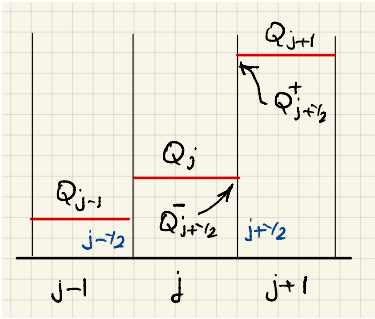
Essence of the finite-volume method

Instead of computing one edge value we will compute *two* values: one the left and one on right of cell-edge. We will next define a *numerical flux function*

$$\mathbf{G} = \mathbf{G}(\mathbf{Q}^-_{j+1/2}, \mathbf{Q}^+_{j+1/2})$$

with *consistency* condition

$$\lim_{\mathbf{Q}_{L,R} \to Q} \mathbf{G}(\mathbf{Q}_L, \mathbf{Q}_R) = \mathbf{F}(\mathbf{Q})$$



In terms of the numerical flux function the FV update formula becomes

$$\frac{\partial \mathbf{Q}_j}{\partial t} + \frac{\mathbf{G}(\mathbf{Q}^+_{j+1/2}, \mathbf{Q}^-_{j+1/2}) - \mathbf{G}(\mathbf{Q}^+_{j-1/2}, \mathbf{Q}^-_{j-1/2})}{\Delta x} = 0$$

Steps in constructing finite-volume method

$$\frac{\partial \mathbf{Q}_j}{\partial t} + \frac{\mathbf{G}(\mathbf{Q}^+_{j+1/2}, \mathbf{Q}^-_{j+1/2}) - \mathbf{G}(\mathbf{Q}^+_{j-1/2}, \mathbf{Q}^-_{j-1/2})}{\Delta x} = 0$$

Hence, to completely specify a finite-volume scheme we must design algorithms for each of the following three steps:

- **Step 1**: A **recovery scheme** (possibly with limiters) to compute the left/right interface values $\mathbf{Q}^{\pm}$ at each interface using a set of cell-average values around that interface,
- **Step 2**: A **numerical flux function** that takes the left/right values and returns a consistent approximation to the physical flux, and
- **Step 3**: A **time-stepping scheme** to advance the solution in time and compute the cell-averages at the next time-step.

Some notation for use in recovery stencils

Example: symmetric recovery across two cells can be written as

$$Q_{i+1/2} = \frac{1}{2}(Q_{i+1} + Q_j) = \frac{1}{2}(d_p + d_m)Q_{i+1/2}$$

Example: central difference scheme for second derivative:

$$\frac{\partial^2 Q_i}{\partial x^2} = \frac{1}{\Delta x^2}(Q_{i+1} - 2Q_j + Q_{i-1}) = \frac{1}{\Delta x^2}(\Delta_p - 2I + \Delta_m)Q_i$$

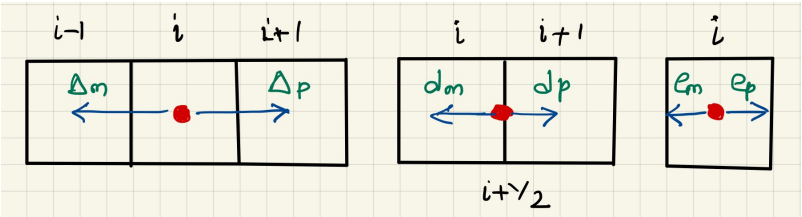
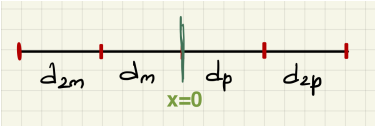


Figure: Basic indexing operators to move from cell to cell, face to cell and cell to face.

Recovery scheme: four-cell stencil, centered scheme



- To construct a four-cell symmetric stencil recovery across an interface we will use a four-cell stencil: $\{d_{2m}, d_m, d_p, d_{2p}\}$
- Setup a local coordinate system with $x = 0$ at the interface and assume a polynomial recovery

$$p(x) = p_0 + p_1 x + p_2 x^2 + p_3 x^3$$

- Match the cell-averages of $p(x)$ in each of the cells $\{d_{2m}, d_m, d_p, d_{2p}\}$ to get a system of linear equations. Solve this system to determine p_0, p_1, p_2, p_3.

Recovery scheme: four-cell stencil, centered scheme

Solving the system of four equations for the four coefficients p_i, $i = 0, \ldots, 3$ yields:

$$
\begin{aligned}
p_0 &= \frac{1}{12}(-d_{2m} + 7d_m + 7d_p - d_{2p})Q \\
p_1 &= \frac{1}{12\Delta x}(d_{2m} - 15d_m + 15d_p - d_{2p})Q \\
p_2 &= \frac{1}{4\Delta x^2}(d_{2m} - d_m - d_p + d_{2p})Q \\
p_4 &= \frac{1}{6\Delta x^3}(d_{2m} - 3d_m + 3d_p - d_{2p})Q.
\end{aligned}
$$

- Notice: stencils of the even coefficients are *symmetric* and the odd coefficients are *anti-symmetric*.
- To compute the interface value we do not really need all of these coefficients but only need to evaluate the recovery polynomial at $x = 0$, i.e we only need $p(0) = p_0$

Recovery scheme: four-cell stencil, centered scheme

To compute the interface value we do not really need all of these coefficients but only need to evaluate the recovery polynomial at $x = 0$, i.e we only need $p(0) = p_0$. Hence, the interface value can be computed from

$$Q^+ = Q^- = \frac{1}{12}(-d_{2m} + 7d_m + 7d_p - d_{2p})Q.$$

Note that due the symmetric nature of the stencil we have only a *single* value at the interface. This means that the numerical flux function at an interface is simply

$$G(Q, Q) = F(Q)$$

from consistency requirements.This completes the spatial finite-volume discretization! The scheme one gets from this is very accurate (even "structure preserving" for Maxwell equations), though not very robust in presence of sharp gradients. (No Free Lunch)

How accurate is any given scheme?

To fix ideas consider we wish to solve the advection equation

$$\frac{\partial f}{\partial t} + \frac{\partial f}{\partial x} = 0$$

Using the four-cell symmetric recovery scheme to compute interface values in the FV update formula we get the semi-discrete scheme *five-cell stencil* update formula:

$$\frac{\partial f_j}{\partial t} = -\frac{1}{\Delta x}\int_{x_{j-1/2}}^{x_{j+1//2}} \frac{\partial f}{\partial x}\, dx = -\frac{1}{12\Delta x}(f_{j-2} - 8f_{j-1} + 8f_{j+1} - f_{j+2})$$

How accurate is this scheme, or what is its order of convergence?

How accurate is any given scheme? Use Taylor series

- Take a Taylor series polynomial around the cell center of cell $I_j = [-\Delta x/2, \Delta x/2]$ locally at $x = 0$

$$T(x) = \sum_{n=0} \frac{T_n}{n!} x^n.$$

- Compute the cell average of this polynomial in each of the stencil cells $\{\Delta_{2m}, \Delta_m, \Delta_p, \Delta_{2p}\}$
- Substitute these averages in the update formula to compute the mean value of the flux gradient in the cell $I_j = [-\Delta x/2, \Delta x/2]$

$$\frac{1}{12\Delta x}(\Delta_{2m} - 8\Delta_m + 8\Delta_p - \Delta_{2p})T = T_1 + \frac{\Delta x^2}{24} T_3 - \frac{21\Delta x^4}{640} T_5 + \dots$$

- Subtract the exact cell average of the gradient of the Taylor polynomial in cell $I_j = [-\Delta x/2, \Delta x/2]$, i.e.

$$\frac{1}{\Delta x} \int_{-\Delta x/2}^{\Delta x/2} \frac{\partial T}{\partial x}\, dx = T_1 + \frac{\Delta x^2}{24} T_3 + \frac{\Delta x^4}{1920} T_5 + \dots$$

from the stencil computed value. The remainder term is the error of the scheme.

Symmetric four-cell recovery scheme is fourth-order accurate

The above procedure (needs use of a compute algebra system to simplify the computations) shows that the symmetric four-cell recovery scheme has error that goes like

$$\frac{\Delta x^4}{30} T_5 + O(\Delta x^6)$$

showing the scheme converges with *fourth-order* accuracy $O(\Delta x^4)$ for linear advection equation. (Reducing Δx by 2 reduces error by a factor of 16).

Accuracy is not everything: dispersion and diffusion

- High-order symmetric schemes like the one we derived are very accurate (even "structure preserving" for some problems) but not robust.
- Two other properties of the scheme are important to understand: *dispersion* and *diffusion*. For this we wil derive a *numerical dispersion relation* analogous to dispersion relation we derived for linearized systems.
- Consider a single mode $f(x) = e^{ikx}$ where k is the wavenumber. Compute the cell-average of the mode on each of the cells in the stencil, plug into the stencil formula to derive the *numerical dispersion relation*

$$i\bar{k} = \sum_{m=-N}^{M} c_m e^{ikm\Delta x}$$

where we have written the stencil in the generic form

$$\frac{1}{\Delta x} \sum_{m=-N}^{M} c_m f_{j+m}$$

Symmetric four-cell recovery scheme has no diffusion!

- Note that the numerical dispersion relation will in general give a *complex* effective wavenumber $\overline{k}$.
- The dispersion relation for a hyperbolic equation is $\omega = \lambda k$. Hence, the *real part* of $\overline{k}$ represents dispersion and *imaginary part* of $\overline{k}$ represents diffusion/growth. Obviously, we want imaginary part to be *negative* to avoid solution blow-up!
- The four-cell symmetric stencil has *no imaginary part* of $\overline{k}$. This related to the fact that it is *symmetric* (anti-symmetric stencil coefficients). This is not necessarily a good thing!

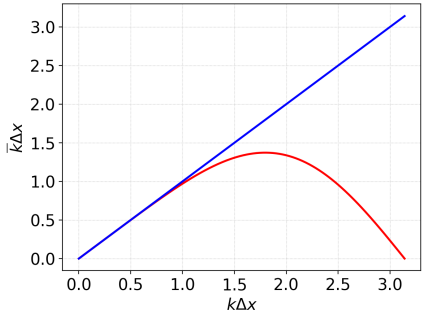


Figure: Real-part of numerical dispersion relation for four-cell recovery scheme. Notice the strong dispersion for higher-k modes