Finite-Volume Methods

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

[1]Princeton Plasma Physics Laboratory, Princeton, NJ

PPPL Graduate Summer School, 2021

Hyperbolic PDEs: rigorous definition, no reliance on linearization

Consider a system of conservation laws written as

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

where Q is a vector of conserved quantities and F(Q) is a vector of fluxes. This system is called *hyperbolic* if the flux Jacobian

$$\mathrm{A} \equiv \frac{\partial \mathrm{F}}{\partial \mathrm{Q}}$$

has *real eigenvalues* and a *complete set of linearly independent* eigenvectors. In multiple dimensions if F_i are fluxes in direction i then we need to show that arbitrary linear combinations $\sum_i n_i \partial \mathrm{F}_i / \partial \mathrm{Q}$ have real eigenvalues and linearly independent set of eigenvectors.

The Riemann Problem for hyperbolic PDEs

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

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

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

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

where $\mathsf{Q}_{L,R}$ 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.

Weak-solutions and entropy conditions

At a shock the solution has a discontinuity. Hence, derivatives are not defined! Differential form of the equations break-down. We must use concept of weak-solutions in this case.

Let $\phi(x,t)$ is a compactly supported (i.e. zero outside some bounded region) smooth function (enough continuous derivatives). Then multiply conservation law

$$\int_0^\infty \int_{-\infty}^\infty \phi(x,t)\left[\frac{\partial \mathsf{Q}}{\partial t} + \frac{\partial \mathsf{F}}{\partial x}\right] dx\, dt = 0$$

by $\phi(x,t)$ and integrating by parts to get the *weak-form*

$$\int_0^\infty \int_{-\infty}^\infty \left[\frac{\partial \phi}{\partial t}\mathsf{Q} + \frac{\partial \phi}{\partial x}\mathsf{F}\right] dx\, dt = -\int_{-\infty}^\infty \phi(x,0)\mathsf{Q}(x,0)dx.$$

Definition (Weak-solution)

A function $\mathsf{Q}(x,t)$ is said to be a weak-solution if it satisfies the weak-form for all compact, smooth $\phi(x,t)$.

Weak-solutions and entropy conditions

Unfortunately, weak-solutions are not unique! Why does this happen?

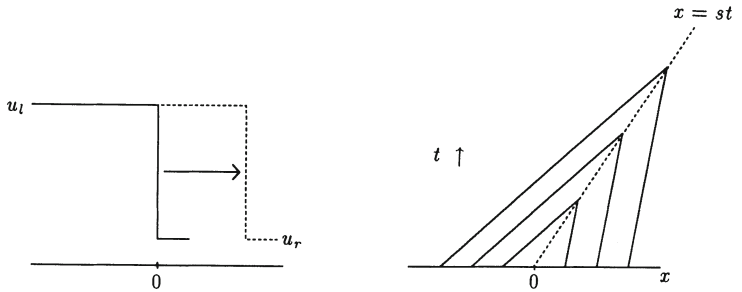
In physical problems there is always some non-ideal effects (viscosity, Landau damping etc) that does not allow a genuine discontinuity to form. However, this "viscous shock layer" can be extremely thin compared to system size. Also, we know entropy must increase in the physical universe.

This indicates we can recover uniqueness in two ways

- Add a viscous (diffusion) term and take limit of viscosity going to zero. (Generally not convenient for numerical work)
- Impose *entropy condition*: construct an *entropy* function such that it remains conserved for smooth solutions but *increases* across a shock. Entropy is naturally suggested in most physical problems.

Weak-solutions of Burgers' equation: shock

When characteristics *converge* a shock will form



Shock-speed is given by the Rankine-Hugoniot jump condition

Consider a discontinuity in the solution wirh left/right states Q_L and Q_R. Then the speed at which this discontinuity moves, s, is called the *shock-speed* and is determined by the Rankine-Hugoniot jump condition

$$s(\mathrm{Q}_R - \mathrm{Q}_L) = \mathrm{F}_R - \mathrm{F}_L$$

For Burgers's equation we simply have

$$s = \frac{1}{2}(u_L + u_R).$$

For linear systems of hyperbolic equations as $\mathrm{F} = \mathrm{AQ}$ we have

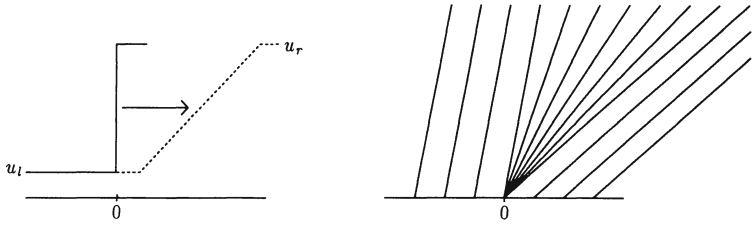
$$s(\mathrm{Q}_R - \mathrm{Q}_L) = \mathrm{A}(\mathrm{Q}_R - \mathrm{Q}_L)$$

which means the eigenvalues of A are the shock-speeds.

For general nonlinear hyperbolic systems only very specific jumps in which the jump in flux and jump in conserved variables are *linearly dependent* can be shocks.

Weak-solutions of Burgers' equation: rarefaction

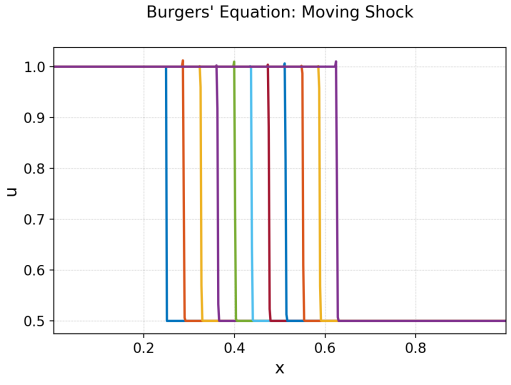
When characteristics *diverge* inifinite solutions to the weak-form! An entropy respecting solution is a *rarefaction*

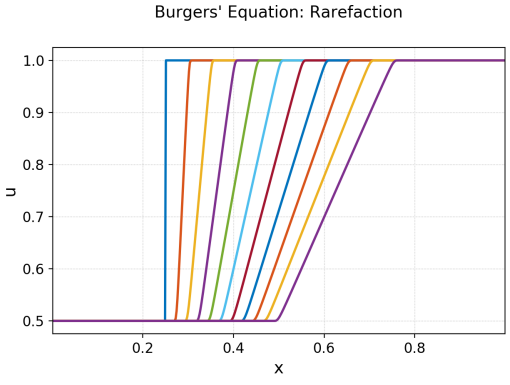


One possible defintion of entropy respecting shocks for Burgers' equation: *only* allow a discontinuity if $u_L > u_R$. So in the above case me must not allow a shock to form.

Weak-solutions: shock and rarefaction

When characteristics *diverge* (right plot below) the weak-solution is not unique. A false "shock" solution also is a weak-solution. Imposing *entropy condition* gives a *rarefaction* wave seen in the right plot.





Euler equations of invicid fluids

The Euler equations for invicid fluids are important in themselves, and form the basis of many other more complex equation systems (Navier-Stokes, multi-fluid plasma equations, MHD, ...)

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \qquad \text{Continuity}$$

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u} + p\mathbf{I}) = 0 \qquad \text{Momentum}$$

$$\frac{\partial \mathcal{E}}{\partial t} + \nabla \cdot [(\mathcal{E} + p)\mathbf{u}] = 0 \qquad \text{Energy}$$

where

$$\mathcal{E} = \underbrace{\frac{p}{\gamma - 1}}_{\text{IE}} + \underbrace{\frac{1}{2}\rho u^2}_{\text{KE}}.$$

is the total energy of the system. If we solve the system in this *conservative* form, then density, momentum and energy are conserved automatically, even locally.

Beyond hyperbolic PDEs: Source terms, non-ideal effects

In most physics applications one must add source terms and non-ideal effects to the underlying hyperbolic PDE, converting it into a PDE of *mixed* type. Typically we will have systems of the form

$$\frac{\partial \mathsf{Q}}{\partial t} + \frac{\partial \mathsf{F}}{\partial x} + \frac{\partial \mathsf{G}}{\partial x} + \ldots = \mathsf{S}$$

where $\mathsf{G}(\mathsf{Q}, \partial \mathsf{Q}/\partial x)$ are *viscous*/non-ideal fluxes that depend on *gradients* of Q (viscous stress-tensor in Navier-Stokes equations, heat-conducion etc) and $\mathsf{S}(\mathsf{Q}, x, t)$ are *source* terms.

The presence of non-ideal and source terms can *significantly* change the physics and required numerics.

Example: Ideal multifluid equations (five-moment)

Multi-fluid plasma equations are an important example. Ignoring non-ideal terms:

$$\frac{\partial \rho_s}{\partial t} + \nabla \cdot (\rho_s \mathsf{u}_s) = 0$$

$$\frac{\partial}{\partial t}(\rho_s \mathsf{u}_s) + \nabla \cdot (\rho_s \mathsf{u}_s \mathsf{u}_s + p_s \mathsf{I}) = \frac{q_s \rho_s}{m_s}(\mathsf{E} + \mathsf{u}_s \times \mathsf{B})$$

$$\frac{\partial \mathcal{E}_s}{\partial t} + \nabla \cdot [(\mathcal{E}_s + p_s)\mathsf{u}_s] = \frac{q_s \rho_s}{m_s}\mathsf{u}_s \cdot \mathsf{E}$$

for each plasma species s (electrons, ions, ...). These are coupled to Maxwell equations

$$\frac{\partial \mathsf{B}}{\partial t} + \nabla \times \mathsf{E} = 0$$

$$\epsilon_0 \mu_0 \frac{\partial \mathsf{E}}{\partial t} - \nabla \times \mathsf{B} = -\mu_0 \sum_s \frac{q_s \rho_s}{m_s}\mathsf{u}_s$$

Multifluid equations (five-moment): conservation properties

Note that in multifluid system total momentum (fluid+field) and total energy (fluid+field) is conserved. Hence, conservation properties are *indirect*: ensuring conservation of total momentum and total energy (specially locally) is non-trivial.

$$\frac{\partial}{\partial t}\left(\sum_s \rho_s \mathbf{u}_s + \epsilon_0 \mathbf{E}\times\mathbf{B}\right) + \nabla\cdot\left[\sum_s(\rho_s \mathbf{u}_s\mathbf{u}_s + p_s\mathbf{I}) + \left(\frac{\epsilon_0}{2}|\mathbf{E}|^2 + \frac{1}{2\mu_0}|\mathbf{B}|^2\right)\mathbf{I} - \left(\epsilon_0\mathbf{E}\mathbf{E} + \frac{1}{\mu_0}\mathbf{B}\mathbf{B}\right)\right] = 0$$

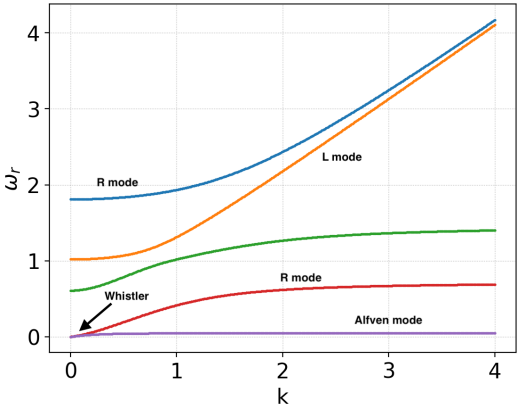
$$\frac{\partial}{\partial t}\left(\sum_s \mathcal{E}_s + \frac{\epsilon_0}{2}|\mathbf{E}|^2 + \frac{1}{2\mu_0}|\mathbf{B}|^2\right) + \nabla\cdot\left[\sum_s(\mathcal{E}_s + p_s)\mathbf{u}_s + \frac{1}{\mu_0}\mathbf{E}\times\mathbf{B}\right] = 0.$$

Multifluid equations (five-moment): eigensystem

The multifluid system is not hyperbolic! However, it has a very complicated eigenstructure (called "dispersion relations" when studying linear plasma problems)

- The presence of the Lorentz force terms add many new time-scales: plasma-frequency, electron/ion cyclotron frequencies ...
- Adding non-ideal terms adds even more scales: diffusion and viscous time-scales.

Understanding the frequencies in the system is critical to determine stable time-steps for explicit schemes. More on this later when we discuss time-stepping.



Essence of the finite-volume method

Consider a PDE of the form (non necessarily hyperbolic)

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

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:

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

where

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

are the *cell-averages* and

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

are *numerical fluxes* at cell interfaces.

Essence of the finite-volume method

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

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

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

- We only know cell-averages Q_j in each cell; we *do not* know the *cell-edge* values $\mathsf{Q}_{j\pm1/2}$ needed to compute the numerical flux $\mathsf{F}_{j\pm1/2}$.
- 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).

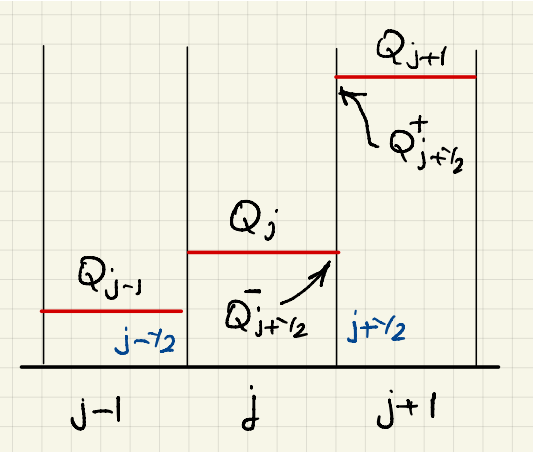
Essence of the finite-volume method

Instead of computing one edge value we will compute *two* values: one the left and one on right of cell. With this, the numerical-flux will then be

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

We must impose the consistency condition:

$$F_{j+1/2}(Q, Q) = F(Q).$$



Cell-averages v/s cell-center values

- Typically, finite-volume schemes evolve the cell-average values; finite-difference schemes evolve cell-center (or 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.

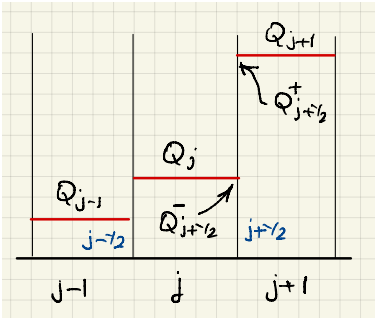
Essence of the finite-volume method

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*

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

with *consistency* condition

$$\lim_{Q_{L,R} \to Q} G(Q_L, Q_R) = F(Q)$$



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

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

Steps in constructing finite-volume method

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

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 $\mathrm{Q}^{\pm}$ 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:

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

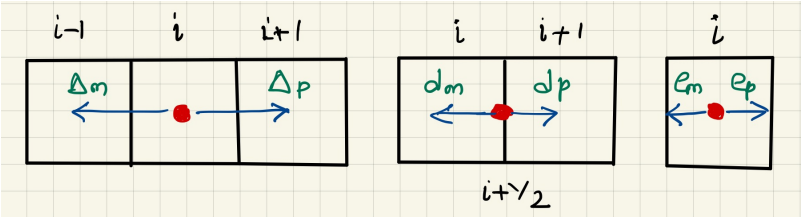
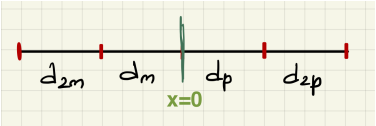


Figure: Basic indexing operators to move from cell to cell, face to cell and cell to face.

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:

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

- 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*

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

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!

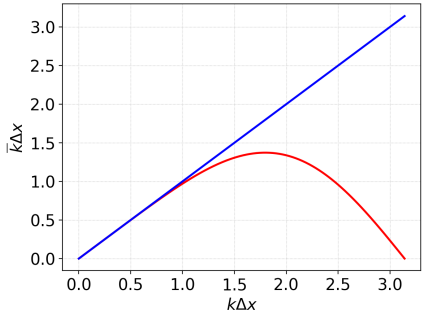


Figure: Real-part of numerical dispersion relation for four-cell recovery scheme. Notice the strong dispersion for higher-k modes

Numerical Flux Function

The numerical flux function computes a *consistent* flux at the cell-edge from the cell averages.

$$\lim_{Q_{L,R} \to Q} G(Q_L, Q_R) = F(Q).$$

Examples

- *Central Flux* in which we simply average the flux from the two states at the interface

$$G(Q_L, Q_R) = \frac{1}{2} \left(F(Q_L) + F(Q_R) \right).$$

- *Upwind Flux* in which we choose the edge on the "upwind" side to account for direction of information flow:

$$G(Q_L, Q_R) = F(Q_L)$$

if information is flowing from left-to-right, and

$$G(Q_L, Q_R) = F(Q_R)$$

if information is flowing from right-to-left. Begs the question: how to determine which direction information is flowing in? Answer: the eigensystem of the hyperbolic equation contains this!

Numerical Flux Function: Lax flux

- A good choice of the numerical flux function is the *local Lax* flux:

$$\mathsf{G}(\mathsf{Q}_L, \mathsf{Q}_R) = \frac{1}{2}\left(\mathsf{F}(\mathsf{Q}_L) + \mathsf{F}(\mathsf{Q}_R)\right) - \frac{|\lambda|}{2}(\mathsf{Q}_R - \mathsf{Q}_L)$$

  where $|\lambda|$ is an estimate of the (absolute) maximum of all eigenvalues at the interface.

- For advection equation this becomes

$$G(f_L, f_R) = \frac{1}{2}a(f_L + f_R) - \frac{|a|}{2}(f_R - f_L)$$

  This works for either sign of advection speed a, automatically giving upwinding.

- Note $|\lambda|$ is only a local (to the interface) *estimate*. You can use a global estimate too: orginal formulation by Peter Lax ("Lax fluxes").

Numerical Flux Function: Systems of equations

- Lax flux is a good "first" flux to use. However, notice it only takes into account a *single* piece of information: maximum eigenvalue.
- For a *linear system* of equations (Maxwell equation) or *locally linearized* nonlinear system we can instead do

$$G(Q_R, Q_L) = \frac{1}{2}\big(F(Q_R) + F(Q_L)\big) - \frac{1}{2}(A^+\Delta Q_{R,L} - A^-\Delta Q_{R,L})$$

  where the *fluctuations* $A^\pm \Delta Q$ are defined as

$$A^\pm \Delta Q_{R,L} \equiv \sum_p r^p \lambda_p^\pm (w_R^p - w_L^p) = \sum_p r^p \lambda_p^\pm l^p (Q_R - Q_L).$$

  where $\lambda_p^+ = \max(\lambda_p, 0)$ and $\lambda_p^- = \min(\lambda_p, 0)$.
- Additional care is needed for nonlinear equations like Euler or ideal MHD equations. More on this on Thursday.

Godunov's Theorem

- A very important theorem proved by Godunov is that there is **no** *linear scheme* that is "monotonicity preserving" (no new maxima/minima created) and **higher than first-order accurate**!
- Consider a general scheme for advection equation

$$f_j^{n+1} = \sum_k c_k f_{j+k}^n.$$

The discrete slope then is

$$f_{j+1}^{n+1} - f_j^{n+1} = \sum_k c_k \left(f_{j+k+1}^n - f_{j+k}^n \right).$$

Assume that all $f_{j+1}^n - f_j^n > 0$. To maintain monotonicity at next time-step hence one must have all $c_k \geq 0$.

Godunov's Theorem

- First order upwind scheme:

$$f_j^{n+1} = f_j^n - \frac{\Delta t}{\Delta x}(f_j^n - f_{j-1}^n)$$

  this satisfies monotonicity as long as $\Delta t/\Delta x \leq 1$.

- Second order symmetric scheme

$$f_j^{n+1} = f_j^n - \frac{\Delta t}{2\Delta x}(f_{j+1}^n - f_{j-1}^n)$$

  clearly this does not satisfy the condition of monotonicity.

- In general condition on Taylor series to ensure atleast second-order accuracy shows that at least *one* of the c_ks must be negative. Hence, by contradiction, *no such scheme exists*!

Godunov's Theorem: Unfortunate Consequences and Workarounds

- Godunov's Theorem is highly distressing: accurate discretization seems to preclude a scheme free from monotonicity violations
- One way around is to start with a linear scheme that is very accurate and then add some local diffusion to it to control the monotonicity.
- However, Godunov's theorem shows that this "diffusion" must be dependent on the local solution itself and can't be fixed *a priori*. This means a **monotonicity preserving scheme must be nonlinear**, even for linear hyperbolic equations.
- Leads to the concept of *nonlinear limiters* that control the monotonicity violations (adding diffusion to high-k modes). No free lunch: limiters must diffuse high-k modes but this will inevitably lead to issues like inability to capture, for example, high-k turbulence spectra correctly without huge grids.
- Major research project: interaction of shocks, boundary layers and turbulence in high-Reynolds number flows.

Nonlinear flux limiters: Getting around Godunov's Theorem

To get around Godunov's Theorem we need to construct a *nonlinear scheme*, even for linear equations. One apporach is to use nonlinear flux-limiters:

$$\mathsf{F}_{j+1/2} = \phi(r_{j+1})\mathsf{F}^{H}_{j+1/2} + \big(1 - \phi(r_{j+1})\big)\mathsf{F}^{L}_{j+1/2}$$

where $\phi(r) > 0$ is a *limiter* function: chooses between *high-order* and *low-order* flux.

- What are the low- and high-order fluxes? For high-order fluxes: use either symmetric or higher-order upwind-biased recovery to construct the flux. For low-order use first-order upwind fluxes.

The first-order upwind flux is "Total-Variation Diminishing" (TVD), $\mathrm{TV}(f^{n+1}) \leq \mathrm{TV}(f^{n})$ where "Total-Variation" is defined as:

$$\mathrm{TV}(f) = \sum_{j} |f_{j+1} - f_j|$$

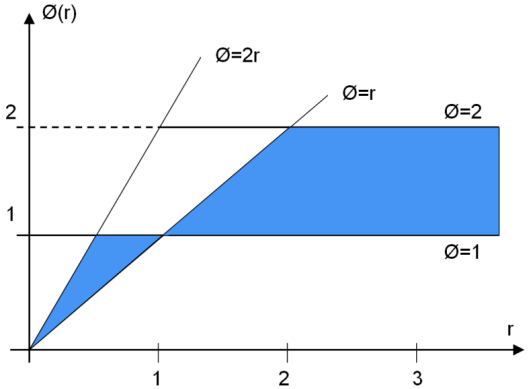
Nonlinear flux limiters

The limiter function $\phi(r)$ depends on an estimate of the *relative* slopes at an interface. For example, one choice is

$$r_{j+1/2} = \frac{\text{slope at upwind-edge}}{\text{slope at } j+1/2}$$

(For systems of equations one needs limit each eigenvector instead). With this, choose a function that maintains TVD property. Eg, min-mod limiter

$$\phi(r) = \max\big(0, \min(2r, (1+r)/2, 2)\big).$$

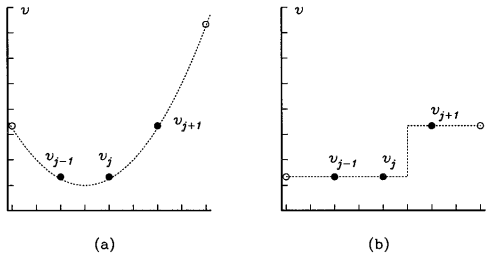


Admissible limiter region for second-order TVD schemes (Sweby, 1984)

See Wikipedia page https://en.wikipedia.org/wiki/Flux_limiter. (Not very high-quality but gives you a general idea).

Nonlinear flux limiters: No "Perfect" Limiter!

- Unfortunately, there is no perfect limiter (though some come close to perfection): depends on problem and best to implement many!
- Most limiters "chop off" genuine maxima/minima: notice that $\phi(r < 0) = 0$ which means that if there is a genuine maxima/minima then low-order flux is selected.
- Tricky to distinguish step-function from parabola! "Best" limiter (IMO): Suresh and Huynh, JCP **136**, 83-99 (1997). Not an easy paper to understand.



Some Parting Thoughts and All the Best!

Here are some personal parting thoughts on computational physics:

- Computational physics is a rapidly evolving field. Good field to be in!
- Strive for **technical excellence**. Do not settle for existing methods or tools and spend time in understanding deeply **both** the physics of the equations *and* the numerics used to solve them. Go beyond your classwork and thesis reasearch (make it a point to read arxiv physics.comp-ph and math.NA postings *every day*).
- Modern computational physics is moving to C and C++: please learn them. Use good software practices (write modular code, use version control, build systems, regression tests). Even for your thesis code!
- To become *really good* you must **apprentice yourself** to a genuine expert.
- If you become an expert at the (i) physics (ii) mathematics of the numerical methods (iii) programming and software techniques, you will be in a very strong position to contribute to development in many different fields. You will bring *unique skills* which few other people will be able to match.