From Finite-Volume Methods to Discontinuous Galerkin Schemes

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

[1]Princeton Plasma Physics Laboratory, Princeton, NJ

Princeton University, Course AST560, Spring 2021

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.

Godunov's Theorem: Workarounds

- To prove Godunov's Theorem we assumed that the solution at the next time-step was a *linear* combination of the old time-step. However, this leads to monotonicity violations.
- Hence, there may be hope that a *nonlinear* combination might allow constructing a monotonic (shock-capturing) scheme.
- Several ways to do this: we know that first-order upwind preserves monotonicity. Hence, perhaps combine a high-order recovery with first-order upwind in "non-smooth" regions.
- Use a *nonlinear recovery* for computing the interface value. This leads to the concept of *limiters*.
- Use symmetric diffusion free scheme and add controlled *hyper-diffusion* depending on *local* solution.

Diffusion, Hyperdiffusion

Consider adding a diffusion term to the RHS of the advection equation

$$\frac{\partial f}{\partial t} + \frac{\partial f}{\partial x} = \nu_2 \frac{\partial^2 f}{\partial x^2}$$

The diffusion term will damp modes as $-\nu_2 k^2$. Unfortunately, even long-wavelength modes will be damped: Not good! How about

$$\frac{\partial f}{\partial t} + \frac{\partial f}{\partial x} = -\nu_4 \frac{\partial^4 f}{\partial x^4}$$

This will damp modes as $-\nu_4 k^4$. Even higher-order hyper-diffusion can be added.

Diffusion, Hyperdiffusion

To discretize the (hyper)-diffusion use *symmetric recovery* based scheme:

$$\frac{\partial f_j}{\partial t} + \frac{f_{j+1/2} - f_{j-1/2}}{\Delta x} = \frac{\nu_2}{\Delta x}\left(\frac{\partial f_{j+1/2}}{\partial x} - \frac{\partial f_{j-1/2}}{\partial x}\right)$$

For example, use 4-cell symmetric recovery across each interface and use its *first-derivative* to compute gradients at each edge.

- The same recovery polynomial need not be used to compute the hyperbolic terms and diffusion term: use upwind biased for hyperbolic, for example, and symmetric recovery for (hyper)-diffusion.
- For hyper-diffusion a *wider* stencil may be needed if order is to be maintained.

Numerical Flux Function: Lax flux

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

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

  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.

Numerical Flux Function: Nonlinear equations

- For nonlinear scalar Burgers' equation ($F(f) = f^2/2$) we can use Lax-fluxes

$$G(f_L, f_R) = \frac{1}{4}(f_L^2 + f_R^2) - \frac{|s|}{2}(f_R - f_L)$$

  where now we compute the speed $s = (f_R + f_L)/2$. Note that the speed s is that of a shock with left/right values f_R and f_L.
- For more complicated system of equations (Euler, ideal MHD) one can use Lax fluxes (good first choice). Or, use a Roe solver (JCP, **43**, 357-372 (1981)).
- Roe solvers work by computing the flux Jacobian using suitable "Roe averages" and using these to compute eigenvalues and left/right eigenvectors and using the flux function for linear system of equations.

Roe averaging and Roe solver for numerical flux

- The Roe solver works by local linearization at an interface in which the nonlinear equation is written as

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

  where $\mathbf{A} = \mathbf{A}(\mathbf{Q}_L, \mathbf{Q}_R)$ is the flux Jacobian computed using left/right states at that interface.

- For conservation one must ensure the *flux-jump* condition

$$\mathbf{A}(\mathbf{Q}_L, \mathbf{Q}_R)(\mathbf{Q}_L - \mathbf{Q}_R) = \mathbf{F}(\mathbf{Q}_L) - \mathbf{F}(\mathbf{Q}_R).$$

  This puts serious constraints on how $\mathbf{A}$ is constructed. Roe realized a special way in which this could be done for Euler equations. Probably the most important advance in numerical methods for shock-dominanted/supersonic flow problems.

- Vast (majority?) number of fluid and MHD codes in CFD, astrophysics, (general-)relativistic hydro, use Roe solver. Huge number of variants and "fixes".

Generalizing recovery: path to discontinous Galerkin schemes

- In FV scheme we used *cell-averages* to recover interface values for use in numerical fluxes
- What if we store more than just cell-averages? One can imagine in addition, mean-slope, mean-quadratic moments. Lead naturally to the concept of discontinous Galerkin schemes.

The key connection is the concept of *weak-equality*. Consider an interval I and select a finite-dimensional function space on it, spanned by basis functions ψ_k, $k = 1, \ldots, N$. Choose an inner product, for example

$$(f, g) \equiv \int_I f(x) g(x)\, dx.$$

Weak-equality

Definition (Weak equality)

Two functions, f and g are said to be *weakly equal* if

$$(\psi_k, f - g) = 0$$

for all $k = 1, \ldots, N$. We denote weak equality by

$$f \doteq g.$$

- When we recovered polynomials across an interface in FV scheme we effectively choose a function space with only *one* basis function!
- In DG we can choose as many as we like: allows significant flexibility in designing accurate and compact schemes; suprisingly accurate for some problems.