Approximate Riemann problem solutions and second order methods

Stephen Millmore and Nikos Nikforakis

Laboratory for Scientific Computing, University of Cambridge

Outline

1. Approximate Riemann problem solvers

2. What does a numerical method need?

3. High-resolution methods - flux limiting

4. High-resolution methods - slope limiting

Outline

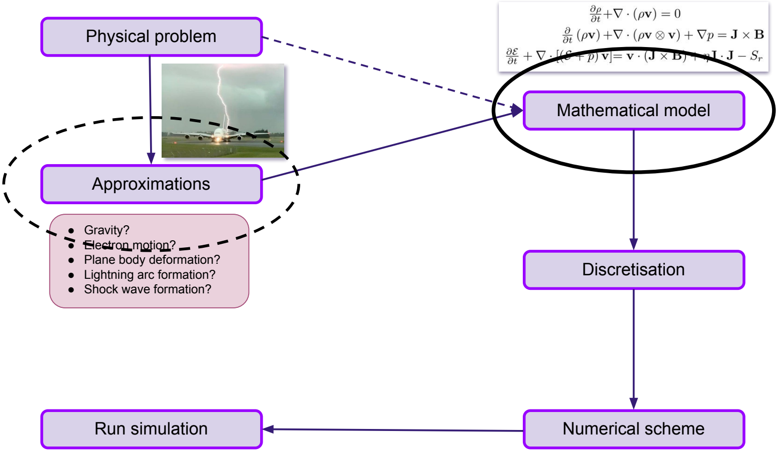
1. Approximate Riemann problem solvers

2. What does a numerical method need?

3. High-resolution methods - flux limiting

4. High-resolution methods - slope limiting

Approximate Riemann problem solvers



$$\frac{\partial \rho}{\partial t}+\nabla\cdot(\rho\mathbf{v})=0$$

$$\frac{\partial}{\partial t}(\rho\mathbf{v})+\nabla\cdot(\rho\mathbf{v}\otimes\mathbf{v})+\nabla p=\mathbf{J}\times\mathbf{B}$$

$$\frac{\partial \mathcal{E}}{\partial t}+\nabla\cdot[(\mathcal{E}+p)\,\mathbf{v}]=\mathbf{v}\cdot(\mathbf{J}\times\mathbf{B})+\eta\mathbf{J}\cdot\mathbf{J}-S_r$$

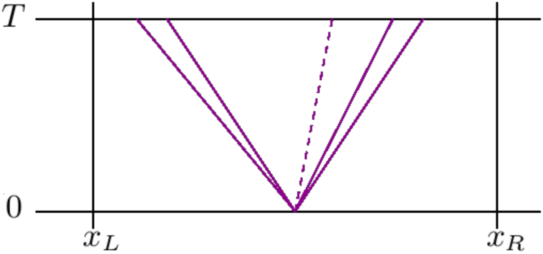
- Gravity?
- Electron motion?
- Plane body deformation?
- Lightning arc formation?
- Shock wave formation?

Why use an approximate solution?

- The iterative nature of the exact Riemann problem solver means that you may want to avoid this to improve computational efficiency
- This is an important concern if you are:
  1. Solving the Euler equations in the 1980's or 1990's, and computational power is limited
  2. **Extremely** concerned about run time (real-time simulations on a GPU)
  3. Solving something more complex than the Euler equations with an ideal gas, where even with current computational power, the solver is expensive
  4. Being told to do so by your lecturer or supervisor
- Originating with concerns about computing resources, much work has been done on **approximate Riemann solvers**
- These make approximations about wave types, speeds and even the number of waves, and sometimes compute fluxes directly, rather than from intermediate states
- Can these actually be accurate?

The consistency condition

- We consider a Riemann problem on a domain $x \in [x_L, x_R]$, $t \in [0, T]$, in which the wave solution does not pass through the boundaries x_L and x_R



- For this case, our conservation law, in integral form, states

$$\int_{x_L}^{x_R} \mathbf{u}\left(x, T\right) \mathrm{d}x = x_R \mathbf{u}_R - x_L \mathbf{u}_L + T\left(\mathbf{f}_L - \mathbf{f}_R\right)$$

The consistency condition

$$\int_{x_L}^{x_R} \mathbf{u}\left(x,T\right)\mathrm{d}x = x_R\mathbf{u}_R - x_L\mathbf{u}_L + T\left(\mathbf{f}_L - \mathbf{f}_R\right)$$

- An approximate Riemann solver is said to be **consistent** if the solution $\mathbf{u}\left(x,T\right)$ obeys this relationship
- An equivalent statement is that in the limit $\Delta x \to 0$ and $\Delta t \to 0$, the numerical method is equivalent to the underlying PDE being solved
- An approximate Riemann problem solution will obey the consistency condition, but will not attempt to reproduce the solution exactly

$$\int_{x_L}^{x_R} \mathbf{u}\left(x,T\right)\mathrm{d}x = \int_{x_L}^{x_R} \tilde{\mathbf{u}}^{\text{approx.}}\left(x,T\right)\mathrm{d}x$$

$$\mathbf{u}^{\text{approx.}}\left(x,T\right) \neq \mathbf{u}\left(x,T\right)$$

Features of an approximate Riemann problem solver

- If an approximate Riemann problem solver is used, it is clear that this must introduce some form of error
- This effects the absolute error in a numerical solution, but not the order of convergence (it may be more smeared, but will still approach the exact solution with resolution)
- It is important to note that the inter-cell flux, $\mathbf{f}_{i+1/2}$ **does not follow**

$$\mathbf{f}_{i+1/2}^{\text{approx.}} \neq \mathbf{f}\left(\mathbf{u}_{i+1/2}^{\text{approx.}}\right)$$

- Instead, the corresponding flux will need to be derived for an approximate Riemann problem solution
- Our approximations introduce a degree of freedom into the solution, and allowing for the expression above makes it straightforward to resolve

The HLL solver

- The Harten, Lax and van Leer (HLL) approximate Riemann solver is probably the simplest approximate solver
- It makes the assumption that there are just **two waves** in the solution, both these waves are **sharp**, and they move with speed S_L and S_R respectively
- We can write this solution as

$$\tilde{\mathbf{u}}\left(x,t\right) = \begin{cases} \mathbf{u}_L & \frac{x}{t} \leq S_L \\ \mathbf{u}^{\mathrm{HLL}} & S_L < \frac{x}{t} < S_R \\ \mathbf{u}_R & \frac{x}{t} \geq S_R \end{cases}$$

- Here $\tilde{\mathbf{u}}$ is used to denote an approximate solution
- If we know S_L and S_R, then it is possible to use the consistency condition to algebraically compute $\mathbf{u}^{\mathrm{HLL}}$

$$\mathbf{u}^{\mathrm{HLL}} = \frac{S_R \mathbf{u}_R - S_L \mathbf{u}_L + \mathbf{f}_L - \mathbf{f}_R}{S_R - S_L}$$

The HLL flux

- As previously stated, we cannot simply compute a flux at $x_{i+1/2}$ using $\mathbf{u}^{\mathrm{HLL}}(x_{i+1/2}, t)$ or even $\tilde{\mathbf{u}}(x_{i+1/2}, t)$
- Instead, since each wave in the HLL solver is sharp, it must obey **jump conditions**

$$\mathbf{f}^{\mathrm{HLL}} - \mathbf{f}_K = S_K \left(\mathbf{u}^{\mathrm{HLL}} - \mathbf{u}_K\right)$$

- Given that there is only a single intermediate state, we can use the left and right jump conditions to remove $\mathbf{u}^{\mathrm{HLL}}$ from this expression, giving

$$\mathbf{f}^{\mathrm{HLL}} = \frac{S_R \mathbf{f}_L - S_L \mathbf{f}_R + S_L S_R \left(\mathbf{u}_R - \mathbf{u}_L\right)}{S_R - S_L}$$

- Our approximate flux is then given by

$$\tilde{\mathbf{f}}_{i+1/2} = \begin{cases} \mathbf{f}_L & 0 \leq S_L \\ \mathbf{f}^{\mathrm{HLL}} & S_L < 0 < S_R \\ \mathbf{f}_R & 0 \geq S_R \end{cases}$$

- Note that if either both wave speeds are positive or negative, then the flux comes from either the left or right state as expected

HLL wave speeds

- The HLL scheme requires two wave speeds, S_L and S_R - we need to provide values for these
- Whilst we are free to provide any values that result in a consistent solution, we will introduce further error into our solution if we are not careful
- Computing an approximate solution from poorly approximated wave speeds is not going to go well...
- In this case, it is not an issue of numerical stability, the consistency condition on the approximate solver takes care of that
- However, what it doesn't say is that the $\mathbf{u}^{\text{HLL}}$ state is physical - negative density or energy are possible

HLL wave speeds

- Finding consistent wave speed estimates for guaranteeing physical solutions has proven to be surprisingly challenging
- However, for most applications, a ‘good enough’ estimate is fine; it is unlikely to cause trouble
- The relationships between the left- and right-moving waves to the sound speed provides some obvious (and commonly used) choices:

$$S_L = v_L - c_{s,L}\,, \qquad S_R = v_R + c_{s,R}$$

$$S_L = \min\left(v_L - c_{s,L}, v_R - c_{s,R}\right), \qquad S_R = \max\left(v_L + c_{s,L}, v_R + c_{s,R}\right)$$

$$S^+ = \max\left(|v_L| + c_{s,L}, |v_R| + c_{s,R}\right), \qquad S_L = -S^+, \qquad S_R = S^+$$

- The last of these is probably the best choice, with good accuracy properties - for even more choices and background, see Toro’s textbook and lectures

Improving the approximate Riemann problem solver

- The HLL solver is incredibly simple, and, unsurprisingly, this means it is not as accurate
- Because the solver ignores (amongst other things) the contact discontinuity, we can think of it consistently introducing errors into how the star states are distributed
- A result of this is that solutions with a contact discontinuity show **a lot** of smearing about this wave
- Do tackle this issue in particular, the HLLC solver was developed
- This introduces a three-wave approximate solution, including the contact discontinuity (the ‘C’ in the name)

The HLLC approximate Riemann problem solver

- The same principle applies as to the HLL solver, consider the consistency condition in the case of **three** sharp waves
- We now have wave speeds S_L, S^* and S_R
- In this case, we have two approximate intermediate states, $\mathbf{u}_K^{\mathrm{HLLC}}$, and we can show these are related to the HLL solution

$$\mathbf{u}^{\mathrm{HLL}} = \left(\frac{S^* - S_L}{S_R - S_L}\right)\mathbf{u}_L^{\mathrm{HLLC}} + \left(\frac{S_R - S^*}{S_R - S_L}\right)\mathbf{u}_R^{\mathrm{HLLC}}$$

- In other words, for a given S_L and S_R, the amount of material between these waves is the same for both HLL and HLLC
- Due to the two intermediate states, it takes a bit more work to express the HLLC states in terms of the initial left and right states only

The HLLC approximate Riemann problem solver

- We now have three sets of jump conditions,

$$\mathbf{f}_L^{\text{HLLC}} - \mathbf{f}_L = S_L \left(\mathbf{u}_L^{\text{HLLC}} - \mathbf{u}_L \right)$$

$$\mathbf{f}_R^{\text{HLLC}} - \mathbf{f}_L^{\text{HLLC}} = S_* \left(\mathbf{u}_R^{\text{HLLC}} - \mathbf{u}_L^{\text{HLLC}} \right)$$

$$\mathbf{f}_R - \mathbf{f}_R^{\text{HLLC}} = S_R \left(\mathbf{u}_R - \mathbf{u}_R^{\text{HLLC}} \right)$$

- We also assume that the central wave is indeed a contact discontinuity, i.e.

$$p_L^{\text{HLLC}} = p_R^{\text{HLLC}} = p_*$$

and

$$v_L^{\text{HLLC}} = v_R^{\text{HLLC}} = v_* = S_*$$

- Obtaining HLLC intermediate states is then a matter of algebra

The HLLC approximate Riemann problem solver

- For example, we can compute the intermediate wave speed in terms of known variables:

$$S_* = \frac{p_R - p_L + \rho_L v_L \left(S_L - v_L\right) - \rho_R v_R \left(S_R - v_R\right)}{\rho_L \left(S_L - v_L\right) - \rho_R \left(S_R - v_R\right)}$$

- Full evaluation of the algebra gives the HLLC states as

$$\mathbf{u}_K^{\text{HLLC}} = \rho_K \left(\frac{S_K - v_K}{S_K - S_*}\right) \begin{pmatrix} 1 \\ S_* \\ \frac{E_K}{\rho_K} + \left(S_* - v_K\right) \left[S_* + \frac{p_K}{\rho(S_* - v^*)}\right] \end{pmatrix}$$

- We can then write the HLLC fluxes as

$$\mathbf{f}_{i+1/2}^{\text{HLLC}} = \begin{cases} \mathbf{f}_L & 0 \leq S_L \\ \mathbf{f}_L + S_L \left(\mathbf{u}_L^{\text{HLLC}} - \mathbf{u}_L\right) & S_L < 0 \leq S_* \\ \mathbf{f}_R + S_R \left(\mathbf{u}_R^{\text{HLLC}} - \mathbf{u}_R\right) & S_* < 0 \leq S_R \\ \mathbf{f}_R & 0 > S_R \end{cases}$$

Using approximate Riemann solvers

- Recall Godunov's method:

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^n - \frac{\Delta t}{\Delta x}\left(\mathbf{f}(\mathbf{u}_{i+1/2}) - \mathbf{f}(\mathbf{u}_{i-1/2})\right)$$

- The quantity $\mathbf{f}_{i+1/2}^n = \mathbf{f}(\mathbf{u}_{i+1/2})$ is found my computing a Riemann problem solution between x_i and x_{i+1} to obtain $\mathbf{u}_{i+1/2}$, and plugging this into the flux function
- To use an approximate Riemann solver, we simply replace this, so $\mathbf{f}_{i+1/2}^n = \mathbf{f}_{i+1/2}^{\mathrm{HLL}}$ or $\mathbf{f}_{i+1/2}^n = \mathbf{f}_{i+1/2}^{\mathrm{HLLC}}$
- Note - sometimes people talk about using the HLLC solver as a numerical method, presumably meaning

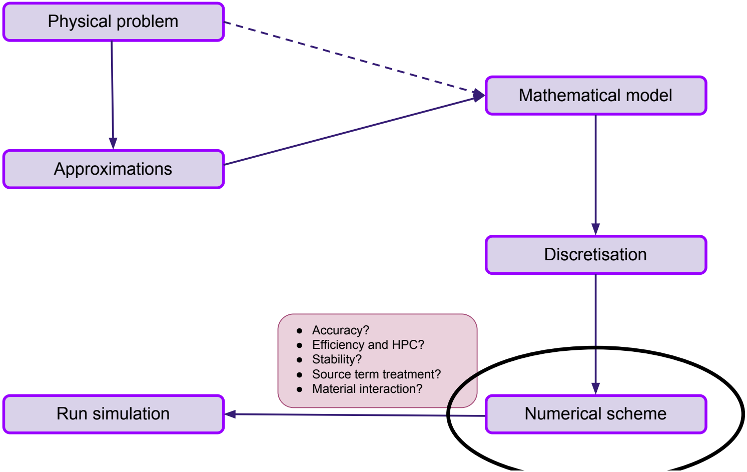
$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^n - \frac{\Delta t}{\Delta x}\left(\mathbf{f}_{i+1/2}^{\mathrm{HLLC}} - \mathbf{f}_{i-1/2}^{\mathrm{HLLC}}\right)$$

- This is Godunov's method with an HLLC (approximate Riemann problem) solver - Godunov's method is not the only one which uses Riemann problem solutions
- And any method which uses these can replace them with an approximate solver

Outline

1. Approximate Riemann problem solvers
2. What does a numerical method need?
3. High-resolution methods - flux limiting
4. High-resolution methods - slope limiting

What does a numerical method need?



- Accuracy?
- Efficiency and HPC?
- Stability?
- Source term treatment?
- Material interaction?

What makes a good numerical method?

- Reminder - we are considering hyperbolic equations only here
- **Conservative** - non-conservative methods fail at discontinuities
- **Consistent** - in the limit $\Delta t \to 0$ and $\Delta x \to 0$ the PDE should be recovered (mathematical property)
- **Convergent** - in the limit $\Delta t \to 0$ and $\Delta x \to 0$ the numerical method should tend towards a solution with increasing accuracy (algorithmic property)
- The **Lax-Wendroff theorem** states that if a numerical method, that is both consistent and conservative, converges to a solution, then this is guaranteed to be a weak solution of the underlying equation
- We then also need an **entropy condition** to obtain the correct weak solution

Consistency

- We mentioned consistency in terms of approximate Riemann problem solutions
- The Lax-Wendroff theorem relates to **consistency of the numerical flux**, $\mathbf{f}^n_{i+1/2}$
- Clearly, if this didn't recover $\mathbf{f}\left(\mathbf{u}\right)$ in the limit $\Delta x \to 0$, this would be bad
- Fortunately, it is a straightforward condition
- If we have a numerical flux $\mathbf{f}^n_{i+1/2}(\mathbf{u}_{i-p}, \ldots, \mathbf{u}_i, \ldots, \mathbf{u}_{i+q})$ then it is consistent if:

$$\mathbf{f}^n_{i+1/2}(\mathbf{u}, \ldots, \mathbf{u}, \ldots, \mathbf{u}) = \mathbf{f}(\mathbf{u})$$

Is satisfying Lax-Wendroff enough?

- Generally, obtaining the correct solution isn't enough for a good numerical method - it should also do so **quickly**
- We would like truncation error to decrease at greater than first order - we refer to such methods as **high-resolution numerical methods**
- We introduced a few of these, such as the Lax-Wendroff method, saw that they gave oscillations for the linear advection equation, and then hardly mentioned them again
- This is because developing higher-order methods which don't generate oscillations takes a bit of work

Monotone methods

- In order to ensure we can obtain a high-resolution numerical method, we need to consider what keeps a numerical method **stable**
- Unstable methods generate oscillations, and these will not be convergent (in general)
- Monotone methods are a class of numerical method that can be **proven** to be stable (see Toro's textbook for full details)
- We consider a (scalar) numerical method which can be written as

$$u_i^{n+1} = H\left(u_{i-k_L}^n, \dots, u_{i+k_R}^n\right)$$

  where k_L and k_R are two integers which determine the stencil of the method
- This is a **monotone scheme** if

$$\frac{\partial H}{\partial u_j^n} \geq 0 \quad \forall j$$

Monotone methods

$$\frac{\partial H}{\partial u_j^n} \geq 0 \quad \forall j$$

- An equivalent statement to this that for any two solutions to the underlying PDE, u and v

$$\text{if} \quad v_i^n \geq u_i^n \quad \forall i \quad \text{then} \quad v_i^{n+1} \geq u_i^{n+1}$$

- This means that the solution can never exceed its initial maximum, or drop below its initial minimum - oscillations cannot form
- For a conservation law written in terms of fluxes, the monotonicity requirement can be written

$$\frac{\partial}{\partial u_i^n} f_{i+1/2}\left(u_i^n, u_{i+1}^n\right) \geq 0 \quad \text{and} \quad \frac{\partial}{\partial u_{i+1}^n} f_{i+1/2}\left(u_i^n, u_{i+1}^n\right) \leq 0$$

- The stable numerical methods we have considered so far have been monotone methods

Are monotone methods the answer?

- Whilst a lack of oscillations is certainly a good thing, monotone methods do have problems
- They can often be diffusive - effectively the enforcement of monotonicity spreads out maxima, minima and sharp features
- This potentially introduces more error than small oscillatory behaviour (provided the oscillations remain small)
- However, a greater issue comes from **Godunov's theorem**:

Monotone methods are at most first order accurate

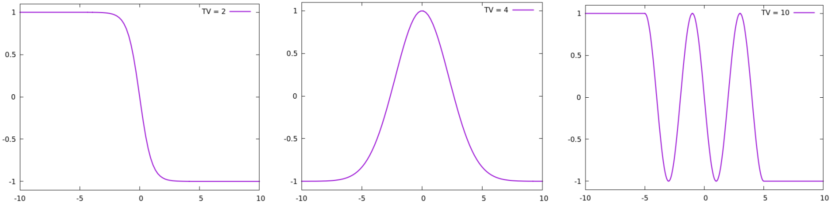
- We cannot use these methods if we want to have more accuracy (faster convergence) in our numerical methods

Total variation

- Fortunately, we don't have to do a lot to improve the capabilities of numerical methods
- A new class of numerical method is based around the behaviour of the **total variation** of the numerical solution

$$\mathrm{TV}\left(u\right) = \int_{-\infty}^{\infty} \left|u'\left(x\right)\right| \mathrm{d}x \quad \text{or} \quad \mathrm{TV}\left(u^n\right) = \sum_{i=-\infty}^{\infty} \left|u_{i+1}^n - u_i^n\right|$$

- This is effectively a measure of how much a function changes over a given range



Total variation diminishing

- A numerical scheme is said to be **total variation diminishing** (TVD) if

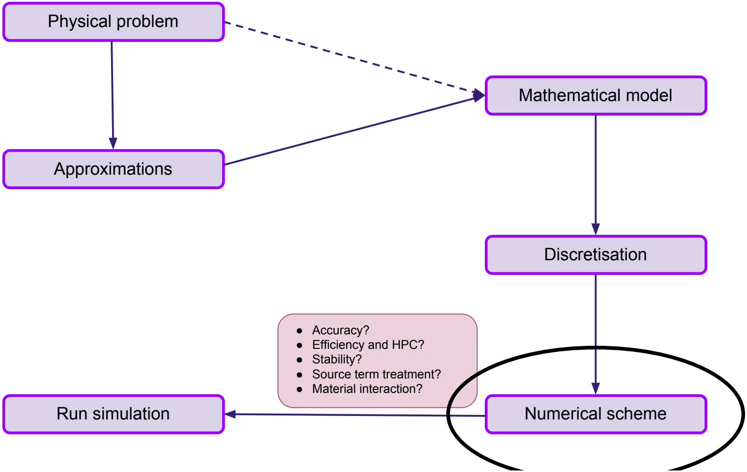
$$\mathrm{TV}\left(u^{n+1}\right) \leq \mathrm{TV}\left(u^{n}\right) \quad \forall n$$

- This is a very similar statement to that of a monotonic scheme - in general we do not expect new maxima or minima to occur
- It differs subtly - if a maximum and minimum both increase by the same amount, this is non-monotonic behaviour, but total variation has not changed
- It is possible to make a **non-linear combination** of a (non-TVD) **high-order** numerical method with first-order monotone method which results in a **high-resolution TVD method**

Outline

1. Approximate Riemann problem solvers
2. What does a numerical method need?
3. High-resolution methods - flux limiting
4. High-resolution methods - slope limiting

Flux limiting methods



- Accuracy?
- Efficiency and HPC?
- Stability?
- Source term treatment?
- Material interaction?

Concept of a high-resolution method

- The principle behind high-resolution methods is fairly simple
- Second-order accurate methods (and higher) are accurate in smooth regions, but suffer from oscillations near discontinuities (non-monotonicity)
- First-order methods are diffusive in general, but do capture discontinuities well
- It is reasonable to assume that discontinuities (which are zero-width by definition) do not occupy much of your numerical domain (even if they are slightly smeared by a numerical method)
- If only a small region of the domain is solved using first-order monotone methods, but the rest using higher-order methods, then we can still have **global high-order convergence** (under some measures...)
- A high-resolution method is high-order **except** near discontinuities, where it reverts to a first-order method

Types of high-resolution method

- There are several techniques available to us to obtain a high resolution method:
  1. **Flux limiting** - combine different flux methods (one of which must be a monotone scheme) to create an overall TVD flux
  2. **Slope limiting** - Use monotone schemes, but move beyond the piecewise-constant approximation for the data within a cell
  3. **Wave-based weighted schemes** - more complex use of slope limiting; can consider this as limiting on the Riemann problem solution
- Note - flux limiting and slope limiting use a very similar technique to achieve stability; some sources are lazy with keeping these name distinct
- Flux limiting tends to require centred schemes (where a high-order scheme is well defined), slope limiting works for centred and Riemann problem-based schemes and wave-based schemes inherently use Riemann problem-based methods

Mathematical concept - flux limiting

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{\Delta x}\left(f_{i+1/2} - f_{i-1/2}\right)$$

- To introduce higher-order methods, we return (temporarily) to scalar conservation laws
- Suppose we have two methods for computing the flux:
  1. f^{LO}: a first-order monotone scheme
  2. f^{HI}: a high-order
- A high-resolution TVD method defines its flux through

$$f_{i+1/2}^{\mathrm{TVD}} = f_{i+1/2}^{\mathrm{LO}} + \phi_{i+1/2}\left(f_{i+1/2}^{\mathrm{HI}} - f_{i+1/2}^{\mathrm{LO}}\right)$$

- $\phi_{i+1/2}$ is a weighting, or a **limiter function**

Limiter functions

$$f_{i+1/2}^{\mathrm{TVD}} = f_{i+1/2}^{\mathrm{LO}} + \phi_{i+1/2}\left(f_{i+1/2}^{\mathrm{HI}} - f_{i+1/2}^{\mathrm{LO}}\right)$$

- Limiter functions are function of the underlying conserved (or primitive) variables, $\phi_{i+1/2} = \phi_{i+1/2}\left(u_{i-k_L}, \ldots, u_{i+k_R}\right)$
- This makes them a **nonlinear combination** of the two flux methods, f^{LO} and f^{HI}
- The flux limiter is bounded, $0 \leq \phi_{i+1/2} \leq 1$
- For a smooth region, $\phi_{i+1/2} = 1$, and hence $f_{i+1/2}^{\mathrm{TVD}} = f_{i+1/2}^{\mathrm{HI}}$
- For a rapidly-changing region, $\phi_{i+1/2} = 0$, and hence $f_{i+1/2}^{\mathrm{TVD}} = f_{i+1/2}^{\mathrm{LO}}$
- The choice of how $\phi_{i+1/2}$ varies between these limits is the tricky part - limiters must be carefully calculated to ensure a method is TVD
- We shall not go into this, see Toro's book for more details

Flux limiters

$$f_{i+1/2}^{\text{TVD}} = f_{i+1/2}^{\text{LO}} + \phi_{i+1/2} \left(f_{i+1/2}^{\text{HI}} - f_{i+1/2}^{\text{LO}} \right)$$

- We have seen a form of flux limited method already
- If f^{LO} is the Law-Friedrichs flux, f^{HI} the Richtmyer flux and $\phi = 1/2$, we recover the FORCE scheme
- This is not a high-resolution scheme, and simply uses a **linear** weighting of two flux methods
- We have stated that a high-resolution scheme is first order near discontinuities $(\phi = 0)$ and second order elsewhere $(\phi = 1)$
- Perhaps unsurprisingly, the (numerical) gradient of the solution is useful for computing the limiter

Changing slope

- Gradient alone isn't sufficient to identify a discontinuity - smooth features can have high gradient, and approximations to gradient of a sharp feature are not always reliable (approximating an 'infinite' gradient)
- Instead, it is better to think of a discontinuity as a sudden change from smooth(ish) to rapidly changing data
- We introduce a slope ratio, $r = r(u_{i-1}, u_i, u_{i+1})$, and define a function $\phi_{i+1/2} = \phi_{i+1/2}(r)$
- At this point, it is important to note that choices for flux (and slope) limiting are **not unique**
- We will provide some examples, following Toro's textbook, which provide suitable combinations of r and ϕ shown to work well in practice

Flux Limiter Centred scheme (FLIC)

- This scheme makes a particular choice of high- and low-order schemes

$$\mathbf{f}^{\mathrm{LO}} = \mathbf{f}^{\mathrm{FORCE}}, \qquad \mathbf{f}^{\mathrm{HI}} = \mathbf{f}^{\mathrm{RI}}$$

- We make this distinction before we consider flux limiters because the choice of flux method can influence the definitions of the flux limiter
- This is, perhaps, not surprising - the FORCE flux already contains a Richtmyer flux contribution
- We would expect differences in obtaining a TVD flux if we replaced $\mathbf{f}^{\mathrm{LO}}$ with the Lax-Friedrichs $\mathbf{f}^{\mathrm{LF}}$ flux

Flux limiters for scalar equations

- For scalar PDEs, we have a single wave associated with the solution - we can always define an **upwind** and a **downwind** direction
- This is reflected in how we define our change of slope

$$r_i = \frac{\Delta u_{\text{upw.}}}{\Delta u_{\text{downw.}}} = \frac{\Delta u_{i-s/2}}{\Delta u_{i+s/2}}$$

$$\Delta u_{i+1/2} = u_{i+1} - u_i, \qquad s = \begin{cases} 1 & a_i > 0 \\ -1 & a_i \leq 0 \end{cases}$$

- Here a_i is the speed of information travel for u_i ($a_i = a$ for the advection equation, $a_i = u_i$ for Burgers' equation)
- In our computation of slope, we implicitly assume that Δx is constant across the domain, hence it does not appear in r_i

Some common flux limiters

- Initial definitions:

$$\phi_g = \frac{1 - C_{\max}}{1 + C_{\max}}, \qquad C_{\max} = C_{\mathrm{CFL}}$$

Superbee:

$$\phi(r) = \begin{cases} 0 & r \leq 0 \\ 2r & 0 < r \leq \frac{1}{2} \\ 1 & \frac{1}{2} < r \leq 1 \\ \min\left(2, \phi_g + (1 - \phi_g) r\right) & r > 1 \end{cases}$$

Van-Leer:

$$\phi(r) = \begin{cases} 0 & r \leq 0 \\ \frac{2r}{1+r} & 0 < r \leq 1 \\ \phi_g + \frac{2(1-\phi_g)r}{1+r} & r > 1 \end{cases}$$

Van-Albada:

$$\phi(r) = \begin{cases} 0 & r \leq 0 \\ \frac{r(1+r)}{1+r^2} & 0 < r \leq 1 \\ \phi_g + \frac{(1-\phi_g)r(1+r)}{1+r^2} & r > 1 \end{cases}$$

Minbee (or minmod):

$$\phi(r) = \begin{cases} 0 & r \leq 0 \\ r & 0 < r \leq 1 \\ 1 & r > 1 \end{cases}$$

FLIC for the Euler equations

- There are two important considerations for applying FLIC to the Euler equations - what needs changing?
- Firstly, we have more than one variable - do we use $\phi\left(\mathbf{u}\right)$ or $\boldsymbol{\phi}\left(\mathbf{u}\right)$?
- Secondly, we have more than one wave - the concept of upwind and downwind slopes no longer applies
- It is perhaps not surprising that there is not a unique means to tackle this
- We shall answer the second point first - we assume we have a discrete variable, q_i, that we shall use for limiting
- Toro recommends defining:

$$r^L_{i+1/2} = \frac{\Delta q_{i-1/2}}{\Delta q_{i+1/2}}, \qquad r^R_{i+1/2} = \frac{\Delta q_{i+3/2}}{\Delta q_{i+1/2}}$$

$$\phi_{i+1/2} = \phi^{\mathrm{LR}} = \min\left(\phi(r^L_{i+1/2}), \phi(r^R_{i+1/2})\right)$$

- **Note - the definition of $\phi(r)$ does not change**

Options for flux limiting on a vector quantity

1. Choose a single quantity to limit on
   - For FLIC, Toro recommends total energy, i.e. $q = E$
   - In general, this quantity **must** jump across a contact discontinuity
   - **All** variables are limited according to the slope change in this variable
2. For k variables, compute $\phi_{i+1/2,k}$ for each variable, q_k
   - A single flux limiter value is then chosen

$$\phi_{i+1/2} = \min_k \left(\phi_{i+1/2,k}\right)$$

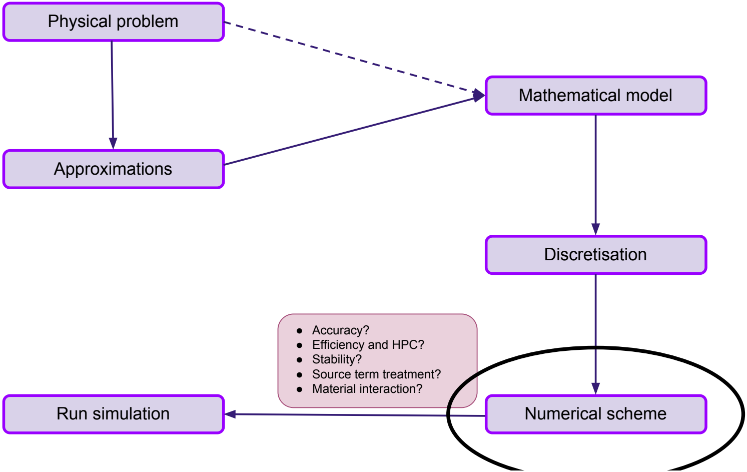
3. For k variables, compute $\phi_{i+1/2,k}$ for each variable, q_k
   - Each variable is now limited according to its own $\phi_{i+1/2,k}$

- Flux (and slope) limiting can be an art as much as a science - fortunately for most numerical methods, someone has done the hard work of working out the best technique for you!

Outline

1. Approximate Riemann problem solvers
2. What does a numerical method need?
3. High-resolution methods - flux limiting
4. High-resolution methods - slope limiting

Slope limiting methods



- Accuracy?
- Efficiency and HPC?
- Stability?
- Source term treatment?
- Material interaction?

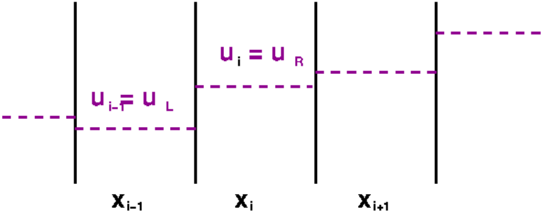
Slope limiting methods

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^n - \frac{\Delta t}{\Delta x}\left(\mathbf{f}(\mathbf{u}_{i+1/2}) - \mathbf{f}(\mathbf{u}_{i-1/2})\right)$$

- Recall that with Godunov's method (and an exact Riemann problem solver), the only approximation came into how the data was distributed in cells x_i and x_{i+1} for computing $\mathbf{u}_{i+1/2}$
- The piece-wise constant approximation is first-order accurate, hence Godunov's method is first-order accurate
- If we made a second-order accurate linear reconstruction (or an even higher order reconstruction) of the data within each cell, we could make the method have a higher order of accuracy
- But, as Godunov's theorem states, this would not be monotone (nor would it be TVD)
- This is because reconstruction of states is based on gradients, which don't exist at discontinuities - what if we only reconstruct **away from discontinuities**?

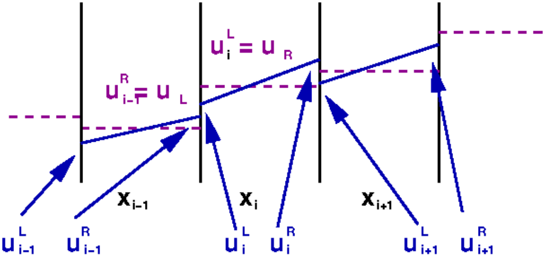
Data reconstruction

- Slope limiting starts with the piecewise-constant finite volume representation
- In a first-order method, the flux across $x_{i-1/2}$ is calculated using the two states, $\mathbf{u}_L$ and $\mathbf{u}_R$ - in this case, these are the piecewise-constant values
- The first idea to introduce is slope reconstruction - since we know the values of a cell's neighbours, we can make a more accurate estimation of the value at the cell boundaries



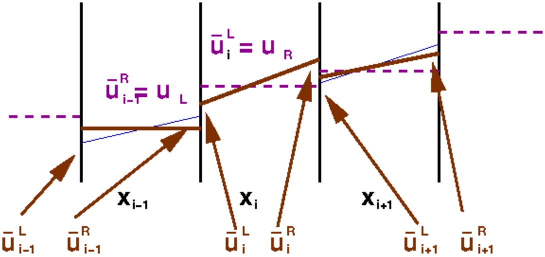
Data reconstruction

- Here we show a first-order reconstruction - using derivative approximations, we can replace the piecewise-constant approximations (whilst retaining the correct volume within the cell)
- This now introduces two interface values for each cell, e.g. $\mathbf{u}_i^L$ and $\mathbf{u}_i^R$
- Note terminology gets confusing now - the right state of a cell is the left state of an interface problem
- Are there any issues with this assumption?



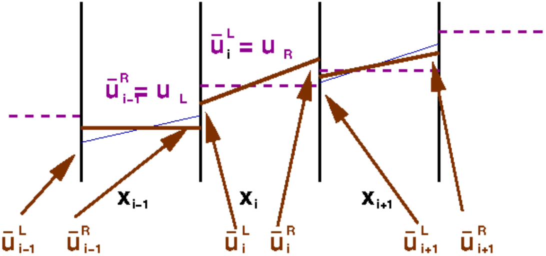
Data reconstruction

- Using the derivatives to make data reconstruction requires that the derivatives exist - they do not at discontinuities
- As with other higher-order methods, this leads to unstable behaviour
- In much the same way as fluxes can be limited based on the change in slope of a variable, reconstructed data can be limited too
- This is **slope limiting**



Slope limiting

- Much like flux limiting, slope limiting has two extremes
- In cases of high change in slope (or maxima/minima), the reconstruction is fully limited, and reverts to piecewise constant (cell x_{i-1})
- In the case of an appropriately smooth change in slope, the reconstructed data is used without modification (cell x_i)
- Intermediate changes result in some limiting taking place (cell x_{i+1})



Reconstruction - mathematically

- The reconstruction of cell $\mathbf{u}_i$ is now a function of x
- A linear reconstruction can be given by

$$\mathbf{u}_i\left(x\right) = \mathbf{u}_i^n + \left(x - x_i\right)\frac{\mathbf{\Delta}_i}{\Delta x}$$

- The quantity $\mathbf{\Delta}_i$ is a measure of the slope,

$$\mathbf{\Delta}_i = \frac{1}{2}\left(1+\omega\right)\mathbf{\Delta}_{i-1/2} + \frac{1}{2}\left(1-\omega\right)\mathbf{\Delta}_{i+1/2}$$

$$\mathbf{\Delta}_{i-1/2} = \mathbf{u}_i^n - \mathbf{u}_{i-1}^n, \qquad \mathbf{\Delta}_{i+1/2} = \mathbf{u}_{i+1}^n - \mathbf{u}_i^n, \qquad \omega \in \left[-1, 1\right]$$

- For the Euler equations, $\omega = 0$ is a safe choice, for scalar equations, knowledge of the direction of flow can make for better choices
- We can now define the cell-boundary values $\mathbf{u}_i^L$ and $\mathbf{u}_i^R$

$$\mathbf{u}_i^L = \mathbf{u}_i^n - \frac{1}{2}\mathbf{\Delta}_i, \qquad \mathbf{u}_i^R = \mathbf{u}_i^n + \frac{1}{2}\mathbf{\Delta}_i$$

Slope limiting - mathematically

- Much like flux limiting, slope limiting involves a quantity that allows us to switch between first-order and higher-order slopes
- Quantities $\mathbf{u}_i^L$ and $\mathbf{u}_i^R$ are the reconstructed states with no limiting or other treatment applied
- Quantities $\bar{\mathbf{u}}_i^L$ and $\bar{\mathbf{u}}_i^R$ are the **limited slopes**

$$\bar{\mathbf{u}}_i^L = \mathbf{u}_i^n - \frac{1}{2}\xi(r)\boldsymbol{\Delta}_i \qquad \bar{\mathbf{u}}_i^R = \mathbf{u}_i^n + \frac{1}{2}\xi(r)\boldsymbol{\Delta}_i$$

- If $\xi(r) = 0$, then $\bar{\mathbf{u}}_i^L = \bar{\mathbf{u}}_i^R = \mathbf{u}_i^n$ and we have a first order (constant) reconstruction
- If $\xi(r) = 1$, then $\bar{\mathbf{u}}_i^L = \mathbf{u}_i^L$ and $\bar{\mathbf{u}}_i^R = \mathbf{u}_i^R$ and we have fully reconstructed slopes

Slope limiting - mathematically

$$\bar{\mathbf{u}}_i^L = \mathbf{u}_i^n - \frac{1}{2}\bar{\mathbf{\Delta}}_i = \mathbf{u}_i^n - \frac{1}{2}\xi(r)\mathbf{\Delta}_i$$

$$\bar{\mathbf{u}}_i^R = \mathbf{u}_i^n + \frac{1}{2}\bar{\mathbf{\Delta}}_i = \mathbf{u}_i^n + \frac{1}{2}\xi(r)\mathbf{\Delta}_i$$

- As with flux limiting, $\xi(r)$ is a function which acts on a slope ratio, r
- We shall only consider slope limiters for the Euler equations, for which upwind/downwind values are ignored - as ever, Toro's textbook is a good resource for slope limiting scalar equations
- We shall assume we know a quantity, $\mathbf{q}$, on which we want to apply the limiter
- As with flux limiting methods, there are a variety of options; here limiting each variable on itself often works well
- Limiting based on characteristics works even better, but is more complicated

Slope vs. flux limiting

- There are obvious similarities between slope and flux limiting, but important, and sometimes subtle differences
- Flux limiters act after fluxes have been calculated; to ensure convergence they need to make sure the flux across a cell boundary is the same on both sides
- As a result, the limiters are **vertex-centred**
- Slope limiters act before fluxes are calculated; to ensure convergence, they must make sure the amount of material in each cell doesn't change
- As a result, the limiters are **cell-centred**

Some common slope limiters

- Initial definitions:

$$r = \frac{\Delta_{i-1/2}}{\Delta_{i+1/2}} = \frac{q_i - q_{i-1}}{q_{i+1} - q_i}, \qquad \xi_R(r) = \frac{2}{1+r}$$

Superbee:

$$\xi(r) = \begin{cases} 0 & r \leq 0 \\ 2r & 0 < r \leq \frac{1}{2} \\ 1 & \frac{1}{2} < r \leq 1 \\ \min\left(r, \xi_R(r), 2\right) & r > 1 \end{cases}$$

Van-Leer:

$$\xi(r) = \begin{cases} 0 & r \leq 0 \\ \min\left(\frac{2r}{1+r}, \xi_R(r)\right) & r > 0 \end{cases}$$

Van-Albada:

$$\xi(r) = \begin{cases} 0 & r \leq 0 \\ \min\left(\frac{r(1+r)}{1+r^2}, \xi_R(r)\right) & r > 0 \end{cases}$$

Minbee (or minmod):

$$\xi(r) = \begin{cases} 0 & r \leq 0 \\ r & 0 < r \leq 1 \\ \min\left(1, \xi_R(r)\right) & r > 1 \end{cases}$$

- Strictly speaking, these are analogous to the named flux limiters, not actually equivalent

Slope-limited schemes

- Slope-limited schemes follow three steps:

1. **Data reconstruction** - this uses all the slope-limiting techniques we have just seen
2. **Half-time step evolution** - a **local** evolution for the boundary-reconstructed variables is performed
3. **First-order update scheme** - this is used on the evolved (and reconstructed) states, which overall provides better-than-first-order convergence of a slope-limited method

- Slope-limited schemes can be used with both centred and Riemann problem based methods
- It is only the third part of the technique which changes, based on the method used
- We shall consider both methods

Half-time step evolution

- After performing data reconstruction, for every cell we have two boundary states, $\bar{\mathbf{u}}_i^L$ and $\bar{\mathbf{u}}_i^R$
- These are updated by a half-time step $(\Delta t/2)$ according to

$$\bar{\mathbf{u}}_i^{L,n+1/2} = \bar{\mathbf{u}}_i^L - \frac{1}{2}\frac{\Delta t}{\Delta x}\left(\mathbf{f}\left(\bar{\mathbf{u}}_i^R\right) - \mathbf{f}\left(\bar{\mathbf{u}}_i^L\right)\right)$$

$$\bar{\mathbf{u}}_i^{R,n+1/2} = \bar{\mathbf{u}}_i^R - \frac{1}{2}\frac{\Delta t}{\Delta x}\left(\mathbf{f}\left(\bar{\mathbf{u}}_i^R\right) - \mathbf{f}\left(\bar{\mathbf{u}}_i^L\right)\right)$$

- This evolution uses **only** data that currently exists for cell x_i, hence it is referred to as **local evolution**
- Actually... that is not quite true, since reconstruction uses neighbouring values
- This process is required such that these methods can be shown to be TVD, and also means they are **second order accurate in time**, though some schemes still work without this step

Slope Limited Centred scheme (SLIC)

- Recall the FORCE flux,

$$\mathbf{f}^{\mathrm{FORCE}}_{i+1/2}(\mathbf{u}^n_i, \mathbf{u}^n_{i+1}) = \frac{1}{2}\left(\mathbf{f}^{\mathrm{LF}}_{i+1/2}(\mathbf{u}^n_i, \mathbf{u}^n_{i+1}) + \mathbf{f}^{\mathrm{RI}}_{i+1/2}(\mathbf{u}^n_i, \mathbf{u}^n_{i+1})\right)$$

- SLIC takes our half-time step values, $\bar{\mathbf{u}}_i^{L,n+1/2}$ and $\bar{\mathbf{u}}_i^{R,n+1/2}$, and uses these the left and right states of the FORCE flux

$$\mathbf{f}^{\mathrm{SLIC}}_{i+1/2} = \mathbf{f}^{\mathrm{FORCE}}_{i+1/2}\left(\bar{\mathbf{u}}_i^{R,n+1/2}, \bar{\mathbf{u}}_{i+1}^{L,n+1/2}\right)$$

- The overall update is then given exactly as we have seen previously

$$\mathbf{u}^{n+1}_i = \mathbf{u}^n_i - \frac{\Delta t}{\Delta x}\left(\mathbf{f}^{\mathrm{SLIC}}_{i+1/2} - \mathbf{f}^{\mathrm{SLIC}}_{i-1/2}\right)$$

- If you have a first-order scheme already implemented, slope limited schemes are a relatively easy modification - they don't change the method you have implemented, they just alter the variables on which it is called

The MUSCL-Hancock scheme

- This is a version of a **M**onotone **U**pstream-centred **S**cheme for **C**onservation **L**aws (MUSCL) attributed to Hancock
- This uses Godunov's method as a means to update the half-time step data

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^n - \frac{\Delta t}{\Delta x}\left(\mathbf{f}(\mathbf{u}_{i+1/2}) - \mathbf{f}(\mathbf{u}_{i-1/2})\right)$$

- For the MUSCL-Hancock scheme, we have $\mathbf{u}_{i+1/2} = \mathbf{u}_{i+1/2}\left(\bar{\mathbf{u}}_i^{R,n+1/2}, \bar{\mathbf{u}}_{i+1}^{L,n+1/2}\right)$, where, as before we obtain the solution to the Riemann problem between these two states along $x = x_{i+1/2}$ - the cell boundary
- The Riemann problem can either use the exact solver, or one of the approximate solvers we have considered - provided it starts from the two states $\bar{\mathbf{u}}_i^{L,n+1/2}$ and $\bar{\mathbf{u}}_i^{R,n+1/2}$, we have a high-resolution method

Some notes on limiters

- We have so far seen four limiters, for both flux and slope limiting
- These are not the only four - why do we need so many?
- Each one has been selected to ensure that the numerical method remains TVD - this involves them satisfying several inequalities
- However, there is still freedom here - how does the limiter know if it has a very steep rarefaction or a true discontinuity?
- Depending on how this happens, different methods revert to first-order accurate at different rates
- This property is known as the **compressiveness** of the limiter
- Too compressive - errors around rarefactions, but discontinuities captured very accurately
- Not compressive enough - rarefactions captured well, but diffusion around discontinuities
- Van-Leer and Minbee are normally good choices which deal with most cases

Some notes on non-linear systems

- For linear PDEs, numerical methods have a lot of easy-to-determine properties e.g. convergence, stability
- This leads to methods which are then applied to non-linear systems - the same behaviour tends to hold **provided the solution doesn't change too much over a time step**
- Unfortunately, non-linear systems are good at generating discontinuities, which lead to rapid changes over a time step - this is especially true if you have taken steps to make sure your solution is as accurate as possible, i.e. you are using a **high-resolution method**
- As a result, you cannot always guarantee a numerical method will work as expected, if you use stability criteria obtained from linear theory
- In such cases, the best approach is to 'stretch out' the time over which this behaviour occurs (use a less compressive limiter)
- Note - this does not **alter the overall order of accuracy**

Dealing with non-linear behaviour

- Several options are available

1. Lowering the CFL number - it is standard to use a CFL number of 0.9-0.95 when solving the Euler equations
    - This can also be necessary if your half-time step update in a slope limited method has a higher wave speed than the initial state
    - Lowering the CFL number further introduces even more smearing in the solution, but will apply this even to relatively smooth regions
2. Using a less compressive limiter - this will (hopefully) leave smoother regions unaffected, but help around discontinuities
3. Using a very low CFL number ($C_{\text{CFL}} = 0.2$) for the first few time steps - often the most severe discontinuities are in the initial data, or the immediate evolution of these initial states

Don't despair...

- It may seem like numerical methods were a lot of effort for something that ended up not being too reliable
- Running through some of the standard tests may even reinforce this view, but...
- In real applications, you will rarely find your method fails for these reasons
- Standard tests exist for a reason - they want to find the weaknesses of a numerical method
- They are there to help you chose a technique that will work first time for your real application, depending whether you expect near-vacuum conditions (Toro's test 2) or very strong shock waves (Toro's tests 3 and 4)

Even more numerical methods

- This course has covered a few of the high-resolution methods available
- Further TVD methods exist, which use even more information from the Riemann problem solution, e.g. the **W**eighted **A**verage **F**lux scheme (WAF)
- Slope limited schemes exist where the slope approximation goes beyond linear (e.g. the **P**iecewise **P**arabolic **M**ethod (PPM))
- By relaxing the requirements for stability even further, we can produce **E**ssentially **N**on-**O**scillatory schemes (ENO) and **W**eighted **E**ssentially **N**on-**O**scillatory schemes (WENO) - these are very easy to extend beyond second order
- However, the underlying principles, those of finite volume schemes, are consistent throughout these techniques