High order finite volume schemes for hyperbolic systems

Gabriella Puppo*

Dipartimento di Matematica
la Sapienza Università di Roma, Italy

Structure Preserving Numerical Methods for Hyperbolic Equations
Würzburg (and everywhere!), Sept-Dec 2020

---

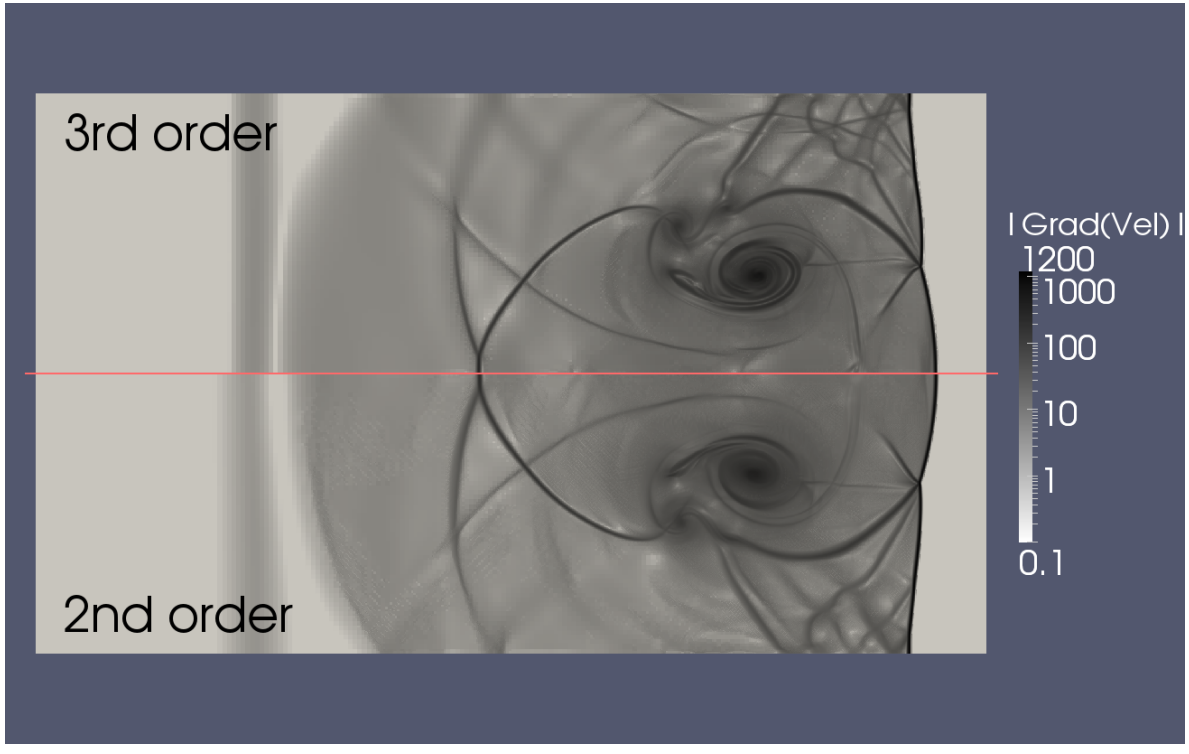
*With: **Isabella Cravero** (Università di Torino, Torino, Italy) **Matteo Semplice** (Università dell'Insubria, Como, Italy), **Giuseppe Visconti** (La Sapienza Università di Roma, Italy), Rosamaria Pidatella e Giovanni Russo (Università di Catania, Catania, Italy)

Outline

SAPIENZA
UNIVERSITÀ DI ROMA

Introduction

High order schemes



(from Coco, Russo, Semplice, JSC 2016)

SAPIENZA
UNIVERSITÀ DI ROMA

Central WENO

Finite volume methods

Consider a hyperbolic system of balance laws of the form

$$\partial_t \boldsymbol{u} + \nabla_x \cdot \boldsymbol{f}(\boldsymbol{u}) = \boldsymbol{s}(\boldsymbol{u}).$$

To integrate the system, one covers the computational domain with N elements $\Omega_j, j = 1, \ldots, N$. Define the cell average of the unknown

$$\boldsymbol{u}_j = \frac{1}{|\Omega_j|} \int_{\Omega_j} \boldsymbol{u} \, \mathrm{d}x.$$

Finite volume methods

Consider a hyperbolic system of balance laws of the form

$$\partial_t \boldsymbol{u} + \nabla_x \cdot \boldsymbol{f}(\boldsymbol{u}) = \boldsymbol{s}(\boldsymbol{u}).$$

To integrate the system, one covers the computational domain with N elements $\Omega_j, j = 1, \ldots, N$. Define the cell average of the unknown

$$\boldsymbol{u}_j = \frac{1}{|\Omega_j|} \int_{\Omega_j} \boldsymbol{u} \, \mathrm{d}x.$$

Integrating the PDEs on each element, one finds the evolution equation for the cell averages as

$$\frac{\mathrm{d}\boldsymbol{u}_j}{\mathrm{d}t} = -\frac{1}{|\Omega_j|} \int_{\partial\Omega_j} \boldsymbol{f} \cdot \boldsymbol{n} \, \mathrm{d}s + \frac{1}{|\Omega_j|} \int_{\Omega_j} \boldsymbol{s}(\boldsymbol{u}) \, \mathrm{d}x.$$

Finite volume methods

Integrating the PDEs on each element, one finds the evolution equation for the cell averages as

$$\frac{\mathrm{d}\boldsymbol{u}_j}{\mathrm{d}t} = -\frac{1}{|\Omega_j|}\int_{\partial\Omega_j} \boldsymbol{f}\cdot\boldsymbol{n}\,\mathrm{d}s + \frac{1}{|\Omega_j|}\int_{\Omega_j} \boldsymbol{s}(\boldsymbol{u})\,\mathrm{d}x.$$

- Quadrature rules to approximate the line and volume integrals.
- High order reconstruction algorithm, to estimate the point values of $\boldsymbol{u}$ along $\partial\Omega_j$, and within Ω_j, from the cell averages.
- Approximation of the fluxes along $\partial\Omega_j$ accounting for intercell communication (approximate Riemann solvers).
- Approximate integration in time.

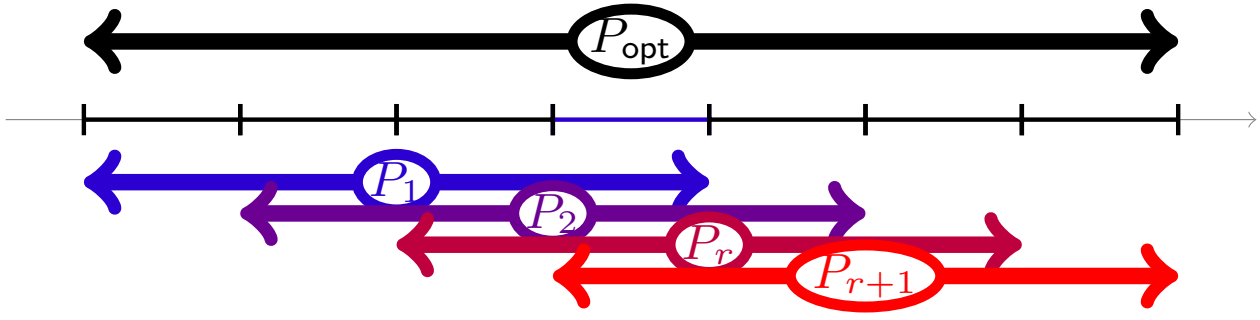
Reconstructions

The key point in finite volume schemes is the reconstruction, which provides from the cell averages $\boldsymbol{u}_j$ the point values along the boundary of Ω_j, and at the interior quadrature nodes.
The reconstruction must be

- fast to compute: use polynomials to approximate the data;
- high order accurate: choose a high degree interpolation polynomial, which is based on a stencil, i.e. a set of cells around the cell Ω_j;
- non oscillatory: choose only information coming from cells which do not contain discontinuities: non linear algorithm;
- efficient: recycle computations as much as possible.

Weighted essentially non-oscillatory reconstructions (1D)

Given the cell averages $\overline{u}_{j-r}, \ldots, \overline{u}_{j+r}$ of a bounded function $u(x)$,



$$(P_{\text{opt}})_j \text{ s.t. } \forall i = -r, \ldots, r: \qquad \frac{1}{|\Omega_{j+i}|} \int_{\Omega_{j+i}} P_{\text{opt}}(x)\mathrm{d}x = \overline{u}_{j+i}$$

- If $\mathcal{R}_j = (P_{\text{opt}})_j$, the accuracy is $O(h^{2r+1})$ in smooth regions.
- However $(P_{\text{opt}})_j$ is oscillatory if a discontinuity is present in its stencil.
- Thus, downgrade, if needed, to a lower accuracy non-oscillatory alternative, $\mathcal{R}_j = P_k$, s.t. P_k contains no discontinuities*.

*Shu, 1997

Example: WENO3 reconstruction in 1D

Third order linear reconstruction algorithm: $\mathcal{R}(x)$

- stencil of 3 cells: $\Omega_{j-1}, \Omega_j, \Omega_{j+1}$;
- $\exists! P_{\text{opt}} \in \mathbb{P}_2 : \int_{\Omega_i} P_{\text{opt}} \mathrm{d}x = |\Omega_i| \overline{u}_i$ for $i = j-1, j, j+1$.

Instead, for each reconstruction point ξ:

- find a convex combination: $P_{\text{opt}}(\xi) = d_L(\xi) P_L(\xi) + d_R(\xi) P_R(\xi)$;
- compute nonlinear weights ω_L and ω_R such that

  $\Rightarrow$ on smooth data: $\omega_j \approx d_j$ and $\mathcal{R}_j(\xi) \approx P_{\text{opt}}(\xi)$

  $\Rightarrow$ otherwise either $\omega_R \approx 0$ and $\mathcal{R}_j(\xi) \approx P_L(\xi)$; or $\omega_L \approx 0$ and $\mathcal{R}_j(\xi) \approx P_R(\xi)$;

- set $\mathcal{R}_j(\xi) := \omega_L(\xi) P_L(\xi) + \omega_R(\xi) P_R(\xi)$

Example: WENO3 reconstruction in 1D

Third order linear reconstruction algorithm: $\mathcal{R}(x)$

- stencil of 3 cells: $\Omega_{j-1}, \Omega_j, \Omega_{j+1}$;
- $\exists! P_{\text{opt}} \in \mathbb{P}_2 : \int_{\Omega_i} P_{\text{opt}} \mathrm{d}x = |\Omega_i| \overline{u}_i$ for $i = j-1, j, j+1$.
- Choosing $\mathcal{R} = P_{\text{opt}}$ would be
  - ➔ third order accurate on smooth data,
  - ➔ oscillatory in the presence of discontinuities.

Instead, for each reconstruction point ξ:

- find a convex combination: $P_{\text{opt}}(\xi) = d_L(\xi) P_L(\xi) + d_R(\xi) P_R(\xi)$;
- compute nonlinear weights ω_L and ω_R such that
  - $\Rightarrow$ on smooth data: $\omega_j \approx d_j$ and $\mathcal{R}_j(\xi) \approx P_{\text{opt}}(\xi)$
  - $\Rightarrow$ otherwise either $\omega_R \approx 0$ and $\mathcal{R}_j(\xi) \approx P_L(\xi)$ or $\omega_L \approx 0$ and $\mathcal{R}_j(\xi) \approx P_R(\xi)$;
- set $\mathcal{R}_j(\xi) := \omega_L(\xi) P_L(\xi) + \omega_R(\xi) P_R(\xi)$

Example: WENO3 reconstruction in 1D

Third order linear reconstruction algorithm: $\mathcal{R}(x)$

- stencil of 3 cells: $\Omega_{j-1}, \Omega_j, \Omega_{j+1}$;
- $\exists! P_{\text{opt}} \in \mathbb{P}_2 : \int_{\Omega_i} P_{\text{opt}} \mathrm{d}x = |\Omega_i| \overline{u}_i$ for $i = j-1, j, j+1$.

Instead, for each reconstruction point ξ:

- consider $P_L \in \mathbb{P}_1$ interpolating $\overline{u}_j$ and $\overline{u}_{j-1}$;
- consider $P_R \in \mathbb{P}_1$ interpolating $\overline{u}_j$ and $\overline{u}_{j+1}$;
- find a convex combination: $P_{\text{opt}}(\xi) = d_L(\xi) P_L(\xi) + d_R(\xi) P_R(\xi)$;
- compute nonlinear weights ω_L and ω_R such that

  $\Rightarrow$ on smooth data: $\omega_j \approx d_j$ and $\mathcal{R}_j(\xi) \approx P_{\text{opt}}(\xi)$

  $\Rightarrow$ otherwise either $\omega_R \approx 0$ and $\mathcal{R}_j(\xi) \approx P_L(\xi)$ or $\omega_L \approx 0$ and $\mathcal{R}_j(\xi) \approx P_R(\xi)$;
- set $\mathcal{R}_j(\xi) := \omega_L(\xi) P_L(\xi) + \omega_R(\xi) P_R(\xi)$

Example: WENO3 reconstruction in 1D

Third order linear reconstruction algorithm: $\mathcal{R}(x)$

- stencil of 3 cells: $\Omega_{j-1}, \Omega_j, \Omega_{j+1}$;
- $\exists! P_{\text{opt}} \in \mathbb{P}_2 : \int_{\Omega_i} P_{\text{opt}} \mathrm{d}x = |\Omega_i| \overline{u}_i$ for $i = j-1, j, j+1$.

Instead, for each reconstruction point ξ:

- find a convex combination: $P_{\text{opt}}(\xi) = d_L(\xi) P_L(\xi) + d_R(\xi) P_R(\xi)$;
- compute nonlinear weights ω_L and ω_R such that
  - $\Rightarrow$ on smooth data: $\omega_j \approx d_j$ and $\mathcal{R}_j(\xi) \approx P_{\text{opt}}(\xi)$
  - $\Rightarrow$ otherwise either $\omega_R \approx 0$ and $\mathcal{R}_j(\xi) \approx P_L(\xi)$; or $\omega_L \approx 0$ and $\mathcal{R}_j(\xi) \approx P_R(\xi)$;
- set $\mathcal{R}_j(\xi) := \omega_L(\xi) P_L(\xi) + \omega_R(\xi) P_R(\xi)$

Example: WENO3 reconstruction in 1D

Third order linear reconstruction algorithm: $\mathcal{R}(x)$

- stencil of 3 cells: $\Omega_{j-1}, \Omega_j, \Omega_{j+1}$;
- $\exists! P_{\text{opt}} \in \mathbb{P}_2 : \int_{\Omega_i} P_{\text{opt}} \mathrm{d}x = |\Omega_i| \overline{u}_i$ for $i = j-1, j, j+1$.

Instead, for each reconstruction point ξ:

- find a convex combination: $P_{\text{opt}}(\xi) = d_L(\xi) P_L(\xi) + d_R(\xi) P_R(\xi)$;
- compute nonlinear weights ω_L and ω_R such that

  $\Rightarrow$ on smooth data: $\omega_j \approx d_j$ and $\mathcal{R}_j(\xi) \approx P_{\text{opt}}(\xi)$

  $\Rightarrow$ otherwise either $\omega_R \approx 0$ and $\mathcal{R}_j(\xi) \approx P_L(\xi)$; or $\omega_L \approx 0$ and $\mathcal{R}_j(\xi) \approx P_R(\xi)$;

- set $\mathcal{R}_j(\xi) := \omega_L(\xi) P_L(\xi) + \omega_R(\xi) P_R(\xi)$

Summing up

WENO reconstructions are very popular and effective. The main ingredients can be summarized as follows.

- They are based on an optimal polynomial P_{opt} which guarantees maximum accuracy but is actually not directly computed.
- The idea is to recover P_{opt} when the flow is smooth, from lower degree polynomials, but this can be achieved only at one reconstruction point at a time.
- Since $\mathcal{R} = P_{\text{opt}}$ when the flow is smooth, the reconstruction algorithm becomes linear on smooth flows.
- The presence of discontinuities triggers the non linearities of the scheme, choosing lower degree polynomials, based on smooth stencils.

The pain of several reconstruction points

SAPIENZA UNIVERSITÀ DI ROMA

For a FV scheme in 2D, several reconstruction points are needed to update a single cell. With WENO, the reconstruction must be repeated at each point.

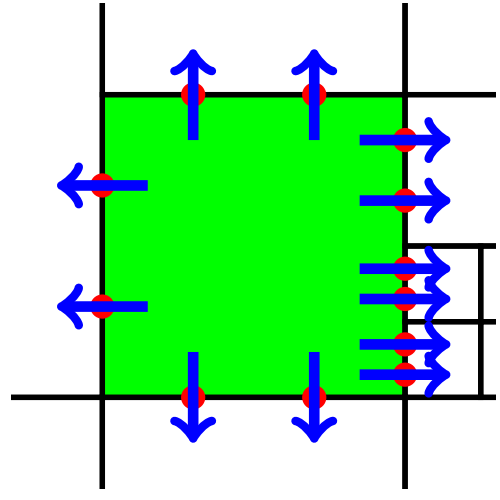
- Things can only get worse on nonuniform grids, as for a mesh created by an adaptive algorithm, such as AMR.

The pain of several reconstruction points

For a FV scheme in 2D, several reconstruction points are needed to update a single cell. With WENO, the reconstruction must be repeated at each point.

- Things can only get worse on nonuniform grids, as for a mesh created by an adaptive algorithm, such as AMR.

- On non uniform grids, you must reconstruct point values at many locations on $\partial\Omega_j$, and the coefficients d_k may not always be positive.

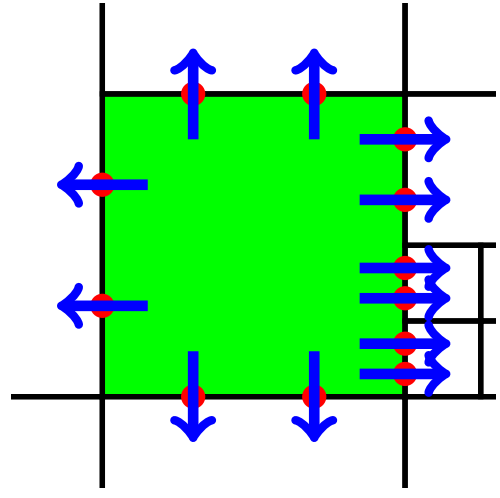


The pain of several reconstruction points

For a FV scheme in 2D, several reconstruction points are needed to update a single cell. With WENO, the reconstruction must be repeated at each point.

- Things can only get worse on nonuniform grids, as for a mesh created by an adaptive algorithm, such as AMR.

- On non uniform grids, you must reconstruct point values at many locations on $\partial\Omega_j$, and the coefficients d_k may not always be positive.



- Moreover in AMR, the mesh topology, and therefore the quadrature nodes, change continuously in time.

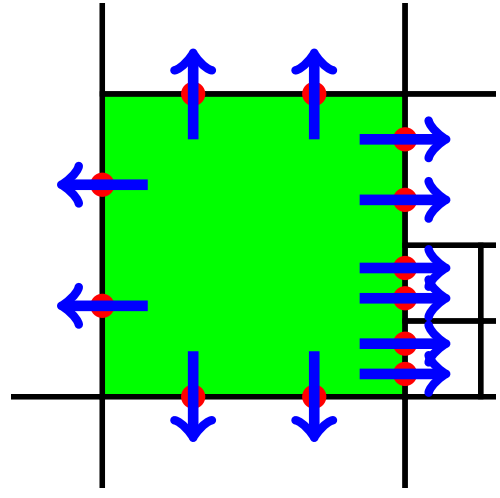
You need a reconstruction which is not based on a single point.

The pain of several reconstruction points

For a FV scheme in 2D, several reconstruction points are needed to update a single cell. With WENO, the reconstruction must be repeated at each point.

- Things can only get worse on nonuniform grids, as for a mesh created by an adaptive algorithm, such as AMR.

- On non uniform grids, you must reconstruct point values at many locations on $\partial\Omega_j$, and the coefficients d_k may not always be positive.



- Moreover in AMR, the mesh topology, and therefore the quadrature nodes, change continuously in time.

You need a reconstruction which is not based on a single point.

A single reconstruction for all points

Recall, WENO3:

$$\text{Given } \hat{x} \in \Omega,\ \mathcal{R}(\hat{x}) = d_L(\hat{x})P_L(\hat{x}) + d_R(\hat{x})P_R(\hat{x}) \qquad \text{(WENO3)}$$

is replaced by

$$\forall x:\ \mathcal{R}(x) = d_0\ P_0(x) + d_L P_L(x) + d_R P_R(x) \qquad \text{(CWENO3)}$$

how? $$P_0(x) := \frac{1}{d_0}\left(P_{\text{opt}}(x) - d_L P_L(x) - d_R P_R(x)\right)$$

why? d_k do not depend on the reconstruction point
$\Rightarrow$ no dependence on mesh topology,
not even in 2d/3d, AMR, ...

Levy, P., Russo
SISC (2000)

A single reconstruction for all points

Given $\hat{x} \in \Omega$, $\mathcal{R}(\hat{x}) = d_L(\hat{x})P_L(\hat{x}) + d_R(\hat{x})P_R(\hat{x})$ (WENO3)

is replaced by

$$\boxed{\forall x : \mathcal{R}(x) = d_0\ P_0(x) + d_L P_L(x) + d_R P_R(x)} \quad \text{(CWENO3)}$$

how? $$P_0(x) := \frac{1}{d_0}\left(P_{\text{opt}}(x) - d_L P_L(x) - d_R P_R(x)\right)$$

why? d_k do not depend on the reconstruction point

$\Rightarrow$ no dependence on mesh topology,

not even in 2d/3d, AMR, ...

Levy, P., Russo

SISC (2000)

A single reconstruction for all points

Given $\hat{x} \in \Omega$, $\mathcal{R}(\hat{x}) = d_L(\hat{x})P_L(\hat{x}) + d_R(\hat{x})P_R(\hat{x})$ (WENO3)

is replaced by

$$\boxed{\forall x : \mathcal{R}(x) = d_0\ P_0(x) + d_L P_L(x) + d_R P_R(x)} \quad \text{(CWENO3)}$$

how? $$\boxed{P_0(x) := \frac{1}{d_0}\left(P_{\text{opt}}(x) - d_L P_L(x) - d_R P_R(x)\right)}$$

why? d_k do not depend on the reconstruction point
$\Rightarrow$ no dependence on mesh topology,
not even in 2d/3d, AMR, ...

Levy, P., Russo
SISC (2000)

A single reconstruction for all points

~~Given $\hat{x} \in \Omega$, $\mathcal{R}(\hat{x}) = d_L(\hat{x})P_L(\hat{x}) + d_R(\hat{x})P_R(\hat{x})$~~ (WENO3)

is replaced by

$$\forall x : \mathcal{R}(x) = d_0\ P_0(x) + d_L P_L(x) + d_R P_R(x) \qquad \text{(CWENO3)}$$

how?
$$P_0(x) := \frac{1}{d_0}\left(P_{\text{opt}}(x) - d_L P_L(x) - d_R P_R(x)\right)$$

why? d_k do not depend on the reconstruction point
$\Rightarrow$ no dependence on mesh topology,
not even in 2d/3d, AMR, ...

Levy, P., Russo
SISC (2000)

CWENO, the general case

Let $p = 2r + 1$ be the required accuracy, where r is the degree of the $r + 1$ low order polynomials P_k forming the standard WENO reconstruction. Now,

1. choose $d_0, d_1, d_{r+1} \in (0, 1)$ such that $\sum_{k=0}^{r+1} d_k = 1$;
2. compute $P_0(x) := \frac{1}{d_0}\left(P_{\text{opt}}(x) - \sum_{k=1}^{r+1} d_k P_k(x)\right)$;
3. compute WENO-style nonlinear weights $d_k \rightsquigarrow \omega_k$; (no x dependence!)
4. compute the reconstruction polynomial (unif. accurate in the cell!)

$$\mathcal{R}(x) = \sum_{k=0}^{r+1} \omega_k P_k(x) \; = u(x) + O(h)^p; \qquad \forall x \in \text{cell}$$

5. evaluate $\mathcal{R}(x)$ on each reconstruction point needed.

Cravero, P., Semplice, Visconti
Math. Comp. (2018)

Background


Shu, C.W.
Lecture Notes in Math., Springer, (1998).


Capdeville
JCP (2008)


Coco, Russo, Semplice,
J. Sci. Comp. (2016)


Balsara, Garain, Shu
JCP (2016)


Castro, Semplice.
Int. J. Num. Meth. Fluids (2019)


Boscheri, Semplice, Dumbser
CiCp (2019)


Busto, Chiocchetti, Dumbser, Gaburro, Peshkov
Frontiers Phys. (2020)

SAPIENZA
UNIVERSITÀ DI ROMA

Spectral properties

Spectral properties

A more efficient reconstruction, but...

The CWENO reconstruction we have proposed is more efficient than standard WENO, but the natural question is:

- does CWENO maintain the good properties of standard WENO?
- One way to do it is to compare the spectral properties of the two reconstructions, which means to study the discrete evolution of Fourier modes of the form $u_k(x,t) = \hat{u}_k(t)e^{ikx}$ in the linear advection equation.
- This brought us to introduce the new concepts of distortion and temperature for a numerical scheme for conservation laws.

A more efficient reconstruction, but...

The CWENO reconstruction we have proposed is more efficient than standard WENO, but the natural question is:

- does CWENO maintain the good properties of standard WENO?
- One way to do it is to compare the spectral properties of the two reconstructions, which means to study the discrete evolution of Fourier modes of the form $u_k(x,t) = \hat{u}_k(t)e^{ikx}$ in the linear advection equation.
- This brought us to introduce the new concepts of distortion and temperature for a numerical scheme for conservation laws.

Von Neumann analysis

Consider the linear advection equation $u_t + a u_x = 0$, with periodic initial and boundary conditions on $(0, 2\pi)$.

- The evolution of a single Fourier mode $u_k(x,t) = \hat{u}_k(t)\exp(ikx)$ is given by

$$\frac{\mathrm{d}\hat{u}_k}{\mathrm{d}t} e^{ikx} = -ik\,a\,\hat{u}_k(t)\;\; e^{ikx}, \qquad u(x, t=0) = u_0(x).$$

- Then the exact solution can be written as

$$u(x,t) = \sum_k \hat{u}_k(0) e^{ik(x-at)}, \qquad \hat{u}_k(0) = \frac{1}{2\pi}\int_0^{2\pi} u_0(x) e^{-ikx}.$$

Von Neumann analysis

Solving the same equation with a linear finite difference scheme on the stencil $\{x_{\ell h}\}, \ell = -r \dots r$, for a single Fourier mode $u_k(x,t) = \hat{u}_k(t) e^{ikx}$ yields

$$\frac{\mathrm{d}\hat{u}_k}{\mathrm{d}t} e^{ikx} = -a\, \hat{u}_k(t) D_x(e^{ikx}),$$

and the discrete derivative D_x is given by

$$D_x(e^{ikx}) = \left(\sum_{\ell=-r}^{r} c_\ell e^{ikh\ell} \right) e^{ikx} = (ik + \tilde{\omega}_k)\, e^{ikx}.$$

- So e^{ikx} is an eigenfunction also for the discrete derivative D_x, except that the amplitude of a single Fourier mode is modified to

$$u_k(x,t) = \hat{u}_k(0) e^{ik(x-at)} e^{-a\tilde{\omega}_k t}.$$

Thus the quantity $\tilde{\omega}_k$ measures the spurious effects due to the discrete approximation, with $\tilde{\omega}_k \approx O(h^p)$.

Von Neumann analysis

Solving the same equation with a linear finite difference scheme on the stencil $\{x_{\ell h}\}, \ell = -r \dots r$, for a single Fourier mode $u_k(x,t) = \hat{u}_k(t) e^{ikx}$ yields

$$\frac{\mathrm{d}\hat{u}_k}{\mathrm{d}t} e^{ikx} = -a\, \hat{u}_k(t) D_x(e^{ikx}),$$

and the discrete derivative D_x is given by

$$D_x(e^{ikx}) = \left(\sum_{\ell=-r}^{r} c_\ell e^{ikh\ell} \right) e^{ikx} = (ik + \tilde{\omega}_k)\, e^{ikx}.$$

- So e^{ikx} is an eigenfunction also for the discrete derivative D_x, except that the amplitude of a single Fourier mode is modified to

$$u_k(x,t) = \hat{u}_k(0) e^{ik(x-at)} e^{-a\tilde{\omega}_k t}.$$

Thus the quantity $\tilde{\omega}_k$ measures the spurious effects due to the discrete approximation, with $\tilde{\omega}_k \approx O(h^p)$.

Artificial diffusion

The real part of $\tilde{\omega}_k$ induces a spurious damping of the amplitude of $u_k(x,t)$, which is faster for high frequency modes ($k >> 1$).
This is called numerical diffusion: the small scale modes tend to disappear.

.

For first order Upwind

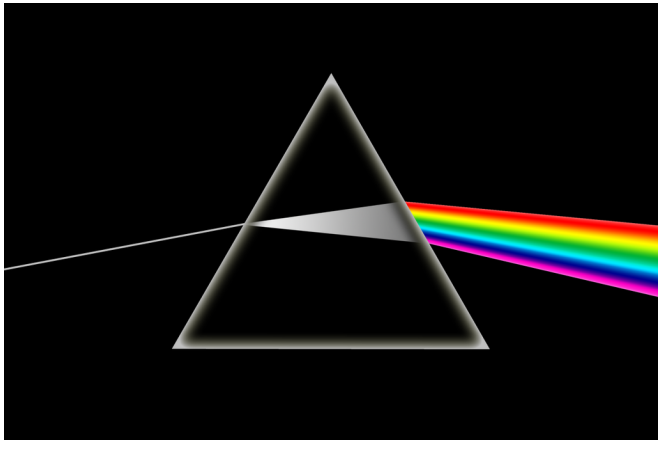
$$\tilde{\omega}_k = -\tfrac{1}{2}k^2 h + O(h^2)$$

and

$$u_k(x,t) \approx \hat{u}_k(0) e^{ik(x-at)} e^{-\frac{1}{2}ak^2 ht}$$

Artificial dispersion

The imaginary part of $\tilde{\omega}_k$ induces a spurious propagation speed. Each mode $u_k(x,t)$ moves with speed $\tilde{a} = a + \frac{a}{k}\text{Im}(\tilde{\omega}_k)$. Again, this effect is stronger for high frequency modes ($k >> 1$).
This is called numerical dispersion: the small scale modes tend to move with high relative speed with respect to the initial wave packet. Thus the Fourier modes separate, and the solution becomes oscillatory.



For a second order scheme

$$\tilde{\omega}_k = -\tfrac{1}{6} i k^3 h^2 + O(h^3)$$

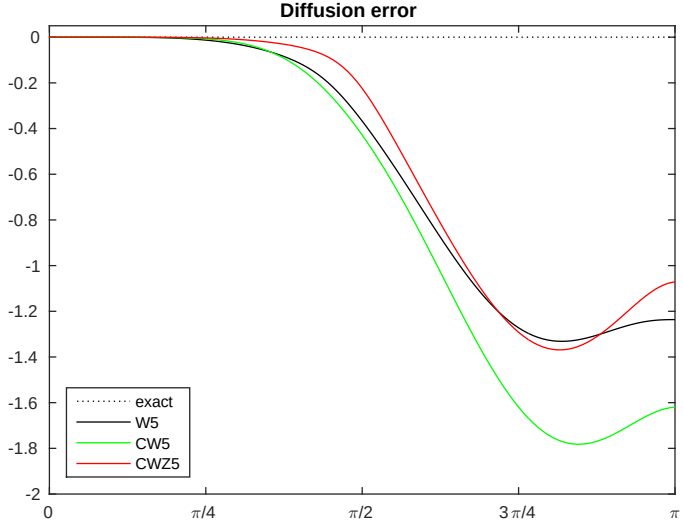
and

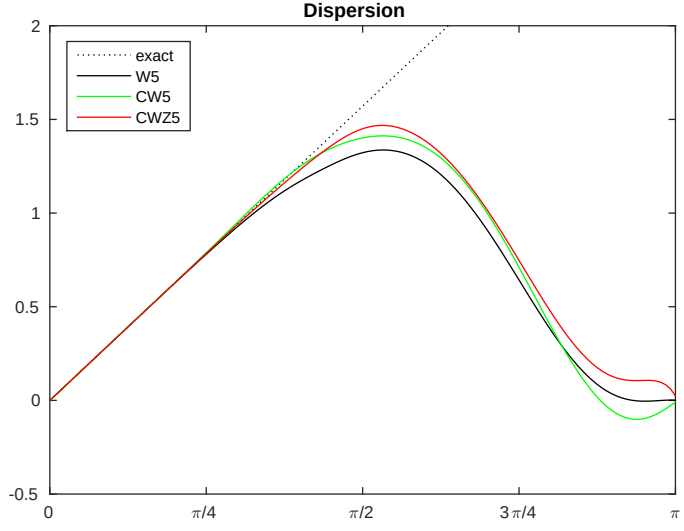
$$u_k(x,t) \approx \hat{u}_k(0) e^{ik(x - a(1 - \frac{1}{6}h^2k^2)t)}.$$

Pirozzoli
JCP (2006)

Diffusion and dispersion for WENO and CWENO

Re($\widetilde{\omega_k}$) and Im($\widetilde{\omega_k}$), as a function of $\ell = \pi k/N$ for WENO (black), CWENO (green) and the modified version CWENOZ (red). Order 5.





- Clearly, for $\ell > \pi/2$, no scheme can resolve the waves correctly: one has less than 2 grid points per wave number.
- All schemes are comparable, but with a definite edge for CWENOZ.

Diffusion and dispersion in the non linear case

In the non linear case, Fourier modes are coupled. But still one can study the effect of the numerical derivative on each mode $D_x e^{ikx}$. Since we are working on real functions, let

$$D_x \begin{bmatrix} \sin(kx) \\ \cos(kx) \end{bmatrix} = \sum_{\ell=1}^{N} \begin{bmatrix} \omega_{2\ell,2k} & \omega_{2\ell,2k+1} \\ \omega_{2\ell+1,2k} & \omega_{2\ell+1,2k+1} \end{bmatrix} \begin{bmatrix} \sin(\ell x) \\ \cos(\ell x) \end{bmatrix},$$

This defines a matrix Ω. The exact derivative is

$$\mathbb{D} = \mathsf{diag}\left(k \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \right), \qquad k = 1, \ldots, N.$$

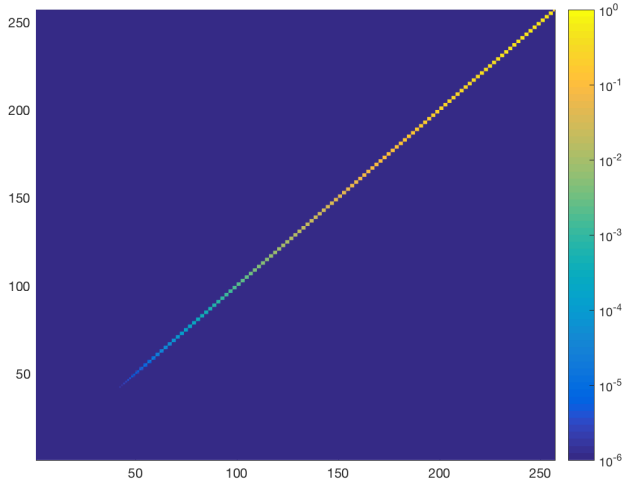
Thus $\mathbb{E} = \Omega - \mathbb{D}$ defines the error matrix.

Cravero, P., Semplice, Visconti
Comp. Fluids (2018)

Diffusion, dispersion and distortion

With the introduction of the error matrix $\mathbb{E}$, we extend the previous analysis for linear schemes to non linear schemes.

If the scheme is linear, the matrix $\mathbb{E}$ is block-diagonal with 2×2 blocks along the diagonal. These blocks contain the artificial diffusion and dispersion information of the scheme.

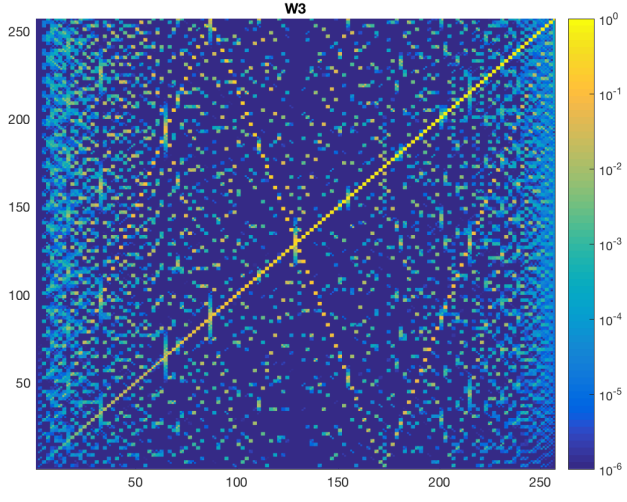


Diffusion, dispersion and distortion

With the introduction of the error matrix $\mathbb{E}$, we extend the previous analysis for linear schemes to non linear schemes.

WENO3

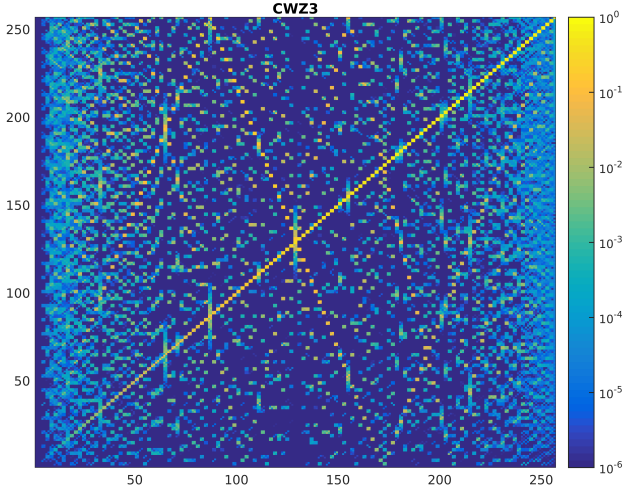
If the scheme is non linear, still the 2×2 blocks along the diagonal give information on how the k-th mode is transformed. But now there are non-zero terms also away from the main diagonals: the size of these terms measures distorsive effects



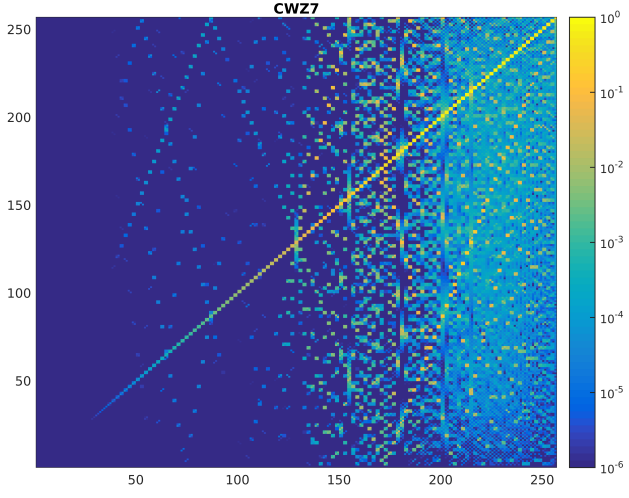
Distortion for CWENO schemes

The amplitude of the coefficients of the error matrix $\mathbb{E}$ shows that as the order is increased, distorsive effects decrease. .

CWENO3



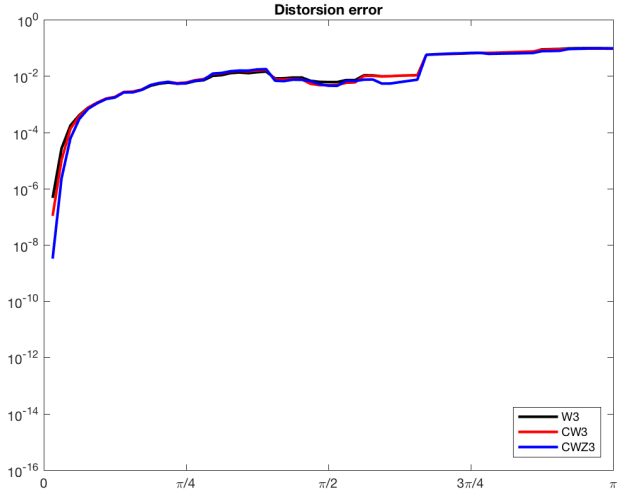
CWENO7



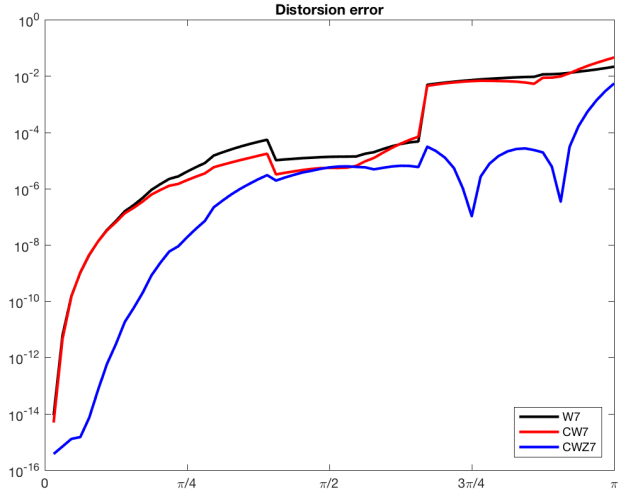
Comparing different high order schemes

We study the distortion errors (i.e. the norm 1 of off diagonal terms in $\mathbb{E}$) of CWENO and WENO schemes for different orders of accuracy.

3rd order



7th order



WENO (black), CWENO (red), CWENOZ (blue).

Temperature

The size of the spurious modes determines the distortion of a scheme, but another interesting parameter is also how far, in frequency space, are the spurious modes from the exact mode.
We quantify this idea with the notion of Temperature on the k-th mode

$$T_k = \frac{1}{N^3} \sum_{\ell=1}^{N} (\Omega_{\mathbb{C}})_{\ell k} \left(\frac{k-\ell}{\pi} \right)^2 .$$

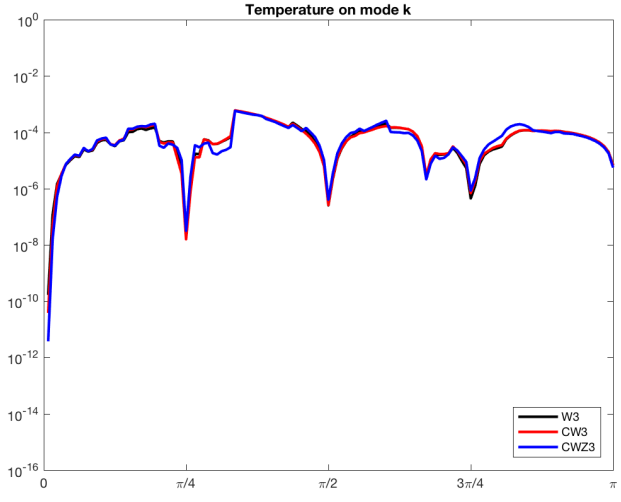
- CWENOZ are the coolest schemes retaining non oscillatory properties.

Temperature

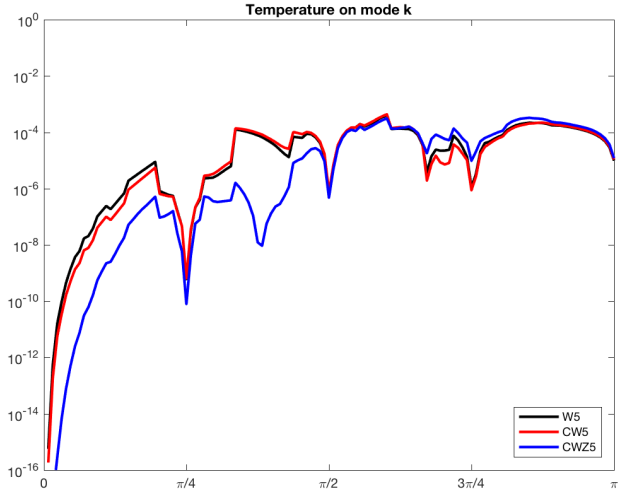
The size of the spurious modes determines the distortion of a scheme, but another interesting parameter is also how far, in frequency space, are the spurious modes from the exact mode.
We quantify this idea with the notion of Temperature on the k-th mode

$$T_k = \frac{1}{N^3} \sum_{\ell=1}^{N} (\Omega_{\mathbb{C}})_{\ell k} \left(\frac{k-\ell}{\pi} \right)^2 .$$

Order 3



Order 5



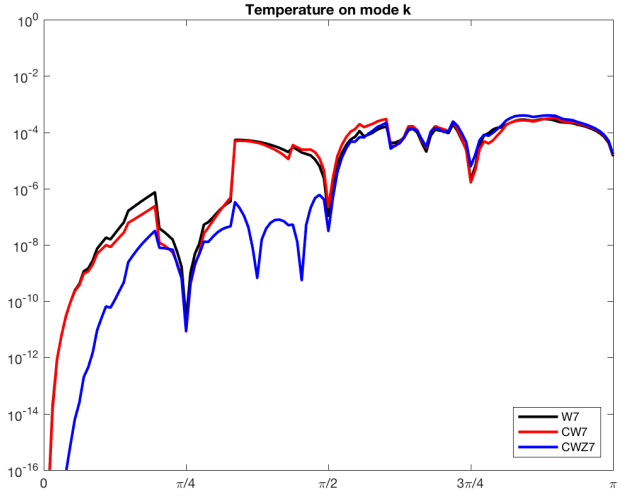
- CWENOZ are the coolest schemes retaining non oscillatory properties.

Temperature

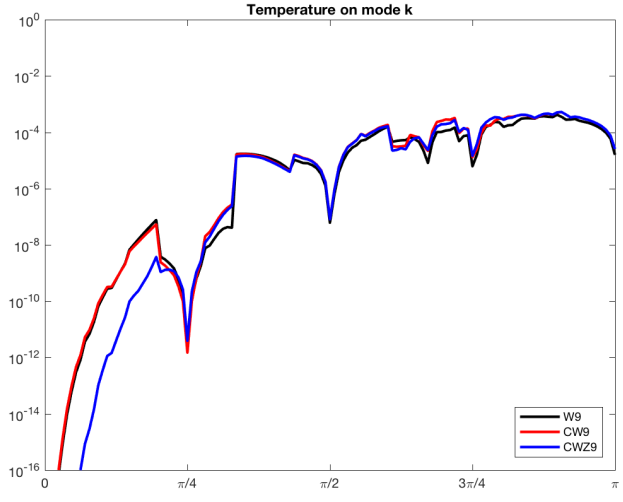
The size of the spurious modes determines the distortion of a scheme, but another interesting parameter is also how far, in frequency space, are the spurious modes from the exact mode.
We quantify this idea with the notion of Temperature on the k-th mode

$$T_k = \frac{1}{N^3} \sum_{\ell=1}^{N} (\Omega_{\mathbb{C}})_{\ell k} \left(\frac{k-\ell}{\pi} \right)^2 .$$

Order 7



Order 9



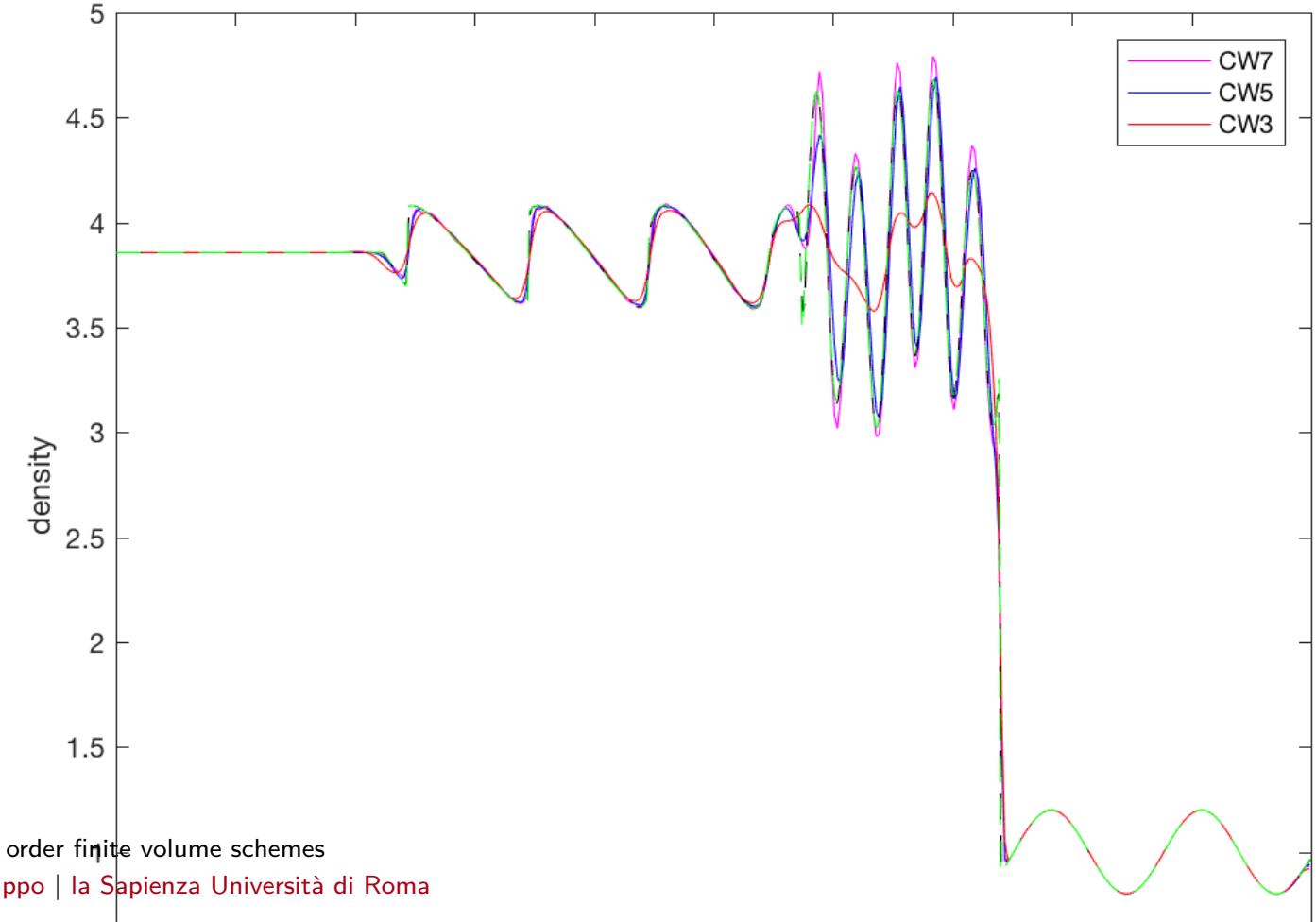
- CWENOZ are the coolest schemes retaining non oscillatory properties.

Cool WENO schemes

A scheme for conservation laws cannot be cold (I mean, with zero temperature), because it would be oscillatory. Some distortion is necessary to prevent spurious oscillations. In this sense, CWENO schemes are cool.

Cool WENO schemes

A scheme for conservation laws cannot be cold (I mean, with zero temperature), because it would be oscillatory. Some distortion is necessary to prevent spurious oscillations. In this sense, CWENO schemes are cool.



Background


LeVeque
Lectures in Math., Birkhäuser, (1992).


Trefethen
SIAM Review (1982)


Pirozzoli
JCP (2006)


Castro, Costa, Don,
JCP. (2011)


Gao, Don,
J. Sci. Comp. (2015)


Cravero, P., Semplice, Visconti
Comp. Fluids (2017)

SAPIENZA
UNIVERSITÀ DI ROMA

Semi-conservative schemes

Let's go back to finite volume schemes...

Consider a hyperbolic system of equations in 1D

$$\partial_t \boldsymbol{u} + \partial_x \boldsymbol{f}(\boldsymbol{u}) = 0.$$

Let's go back to finite volume schemes...

Consider a hyperbolic system of equations in 1D

$$\partial_t \boldsymbol{u} + \partial_x \boldsymbol{f}(\boldsymbol{u}) = 0.$$

The evolution equation for the cell averages in 1D is

$$\frac{\mathrm{d}\boldsymbol{u}_j}{\mathrm{d}t} = -\frac{1}{h}\left(F_{j+1/2}(t) - F_{j-1/2}(t)\right),$$

where $F_{j+1/2}(t) = F(u^-_{j+1/2}, u^+_{j+1/2})$ is the numerical flux.

Let's go back to finite volume schemes...

The evolution equation for the cell averages in 1D is

$$\frac{\mathrm{d}\boldsymbol{u}_j}{\mathrm{d}t} = -\frac{1}{h}\left(F_{j+1/2}(t) - F_{j-1/2}(t)\right),$$

where $F_{j+1/2}(t) = F(u^-_{j+1/2}, u^+_{j+1/2})$ is the numerical flux. One can integrate with Runge Kutta schemes

- $\boldsymbol{u}_j^{n+1} = \boldsymbol{u}_j^n - \lambda \sum_{k=1}^{\nu} b_k \left(F^{(k)}_{j+1/2} - F^{(k)}_{j-1/2}\right)$,
- here the numerical flux is evaluated at reconstructed stage values, $(u^{(k)}_{j+1/2})^{\pm}$,
- and the stage values are again computed evolving the solution.

The important point is that there is no need that the stage values and the final update are computed from the same equation.

Semi-Conservative schemes

The possibility of using different equations for the stage values, while enforcing the conservative equation only in the final step, is a tool that we applied initially to central schemes based on staggered grids.

- The computation of the correct shock speeds is assured by the Lax Wendroff theorem, which uses only the consistency of the numerical fluxes
- Accuracy instead can be obtained only on the smooth pieces of the solution. And where the solution is smooth, several formulations of the PDE can be equivalent, thus yielding the same solution (up to $O(h)^p$ terms).

Pareschi, P., Russo

SISC (2005)

Example

Consider the following two scalar conservation laws, which have the same characteristic form, but with different conservative formulations. Let u_L, u_R denote the left and right state of a discontinuity, with shock speed s',

$$u_t + \left(\tfrac{1}{2}u^2\right)_x = 0, \qquad \Longrightarrow s' = \tfrac{1}{2}(u_L + u_R)$$

$$\left(\tfrac{1}{2}u^2\right)_t + \left(\tfrac{1}{3}u^3\right)_x = 0, \qquad \Longrightarrow s' = \tfrac{2}{3}\frac{u_L^2 + u_L u_R + u_R^2}{u_L + u_R}$$

- If the initial data is $u_0(x) > 0$, the characteristic form is the same, $u_t + uu_x = 0$, but the shock speeds are different.

Example

Consider the following two scalar conservation laws, which have the same characteristic form, but with different conservative formulations. Let u_L, u_R denote the left and right state of a discontinuity, with shock speed s',

$$u_t + \left(\tfrac{1}{2}u^2\right)_x = 0, \qquad \Longrightarrow s' = \tfrac{1}{2}(u_L + u_R)$$

$$\left(\tfrac{1}{2}u^2\right)_t + \left(\tfrac{1}{3}u^3\right)_x = 0, \qquad \Longrightarrow s' = \tfrac{2}{3}\frac{u_L^2 + u_L u_R + u_R^2}{u_L + u_R}$$

- If the initial data is $u_0(x) > 0$, the characteristic form is the same, $u_t + uu_x = 0$, but the shock speeds are different.

First equation (Burgers), $f(u) = \frac{1}{2}u^2$

Suppose you are given the cell averages $\overline{U}^n$

- Reconstruct the point values U_j^n
- Compute the stage values, using the characteristic form and a reconstruction D_x of the space derivative

$$U_j^{(k)} = U_j^n - \Delta t \sum_{i=1}^{k-1} U_j^{(i)} D_x(U^{(i)})(x_j)$$

- Reconstruct the boundary extrapolated data, using the point values of the stages,

$$\left(U_{j+1/2}^{(k)}\right)^{\pm}$$

- Apply the conservative corrector step, evaluating the numerical flux F, consistent with $f = \frac{1}{2}u^2$, obtaining the new cell averages

$$\overline{U}_j^{n+1} = \overline{U}_j^n - \lambda \sum_{k=1}^{\nu} b_k \left(F_{j+1/2}^{(k)} - F_{j-1/2}^{(k)}\right)$$

First equation (Burgers), $f(u) = \frac{1}{2}u^2$

Suppose you are given the cell averages $\overline{U}^n$

- Reconstruct the point values U_j^n
- Compute the stage values, using the characteristic form and a reconstruction D_x of the space derivative

$$U_j^{(k)} = U_j^n - \Delta t \sum_{i=1}^{k-1} U_j^{(i)} D_x(U^{(i)})(x_j)$$

- Reconstruct the boundary extrapolated data, using the point values of the stages,

$$\left(U_{j+1/2}^{(k)}\right)^{\pm}$$

- Apply the conservative corrector step, evaluating the numerical flux F, consistent with $f = \frac{1}{2}u^2$, obtaining the new cell averages

$$\overline{U}_j^{n+1} = \overline{U}_j^n - \lambda \sum_{k=1}^{\nu} b_k \left(F_{j+1/2}^{(k)} - F_{j-1/2}^{(k)}\right)$$

First equation (Burgers), $f(u) = \frac{1}{2}u^2$

Suppose you are given the cell averages $\overline{U}^n$

- Reconstruct the point values U_j^n
- Compute the stage values, using the characteristic form and a reconstruction D_x of the space derivative

$$U_j^{(k)} = U_j^n - \Delta t \sum_{i=1}^{k-1} U_j^{(i)} D_x(U^{(i)})(x_j)$$

- Reconstruct the boundary extrapolated data, using the point values of the stages,

$$\left(U_{j+1/2}^{(k)}\right)^{\pm}$$

- Apply the conservative corrector step, evaluating the numerical flux F, consistent with $f = \frac{1}{2}u^2$, obtaining the new cell averages

$$\overline{U}_j^{n+1} = \overline{U}_j^n - \lambda \sum_{k=1}^{\nu} b_k \left(F_{j+1/2}^{(k)} - F_{j-1/2}^{(k)}\right)$$

First equation (Burgers), $f(u) = \frac{1}{2}u^2$

Suppose you are given the cell averages $\overline{U}^n$

- Reconstruct the point values U_j^n
- Compute the stage values, using the characteristic form and a reconstruction D_x of the space derivative

$$U_j^{(k)} = U_j^n - \Delta t \sum_{i=1}^{k-1} U_j^{(i)} D_x(U^{(i)})(x_j)$$

- Reconstruct the boundary extrapolated data, using the point values of the stages,

$$\left(U_{j+1/2}^{(k)}\right)^{\pm}$$

- Apply the conservative corrector step, evaluating the numerical flux F, consistent with $f = \frac{1}{2}u^2$, obtaining the new cell averages

$$\overline{U}_j^{n+1} = \overline{U}_j^n - \lambda \sum_{k=1}^{\nu} b_k \left(F_{j+1/2}^{(k)} - F_{j-1/2}^{(k)}\right)$$

First equation (Burgers), $f(u) = \frac{1}{2}u^2$

Suppose you are given the cell averages $\overline{U}^n$

- Reconstruct the point values U_j^n
- Compute the stage values, using the characteristic form and a reconstruction D_x of the space derivative

$$U_j^{(k)} = U_j^n - \Delta t \sum_{i=1}^{k-1} U_j^{(i)} D_x(U^{(i)})(x_j)$$

- Reconstruct the boundary extrapolated data, using the point values of the stages,

$$\left(U_{j+1/2}^{(k)}\right)^{\pm}$$

- Apply the conservative corrector step, evaluating the numerical flux F, consistent with $f = \frac{1}{2}u^2$, obtaining the new cell averages

$$\overline{U}_j^{n+1} = \overline{U}_j^n - \lambda \sum_{k=1}^{\nu} b_k \left(F_{j+1/2}^{(k)} - F_{j-1/2}^{(k)}\right)$$

Second equation (no name?),

$v = \frac{1}{2}u^2, v_t + (\frac{1}{3}\sqrt{(2v)^3})_x = 0$

Suppose you are given the cell averages $\overline{V}^n$

- Reconstruct the point values V_j^n and get $U_j^n = \sqrt{2V_j^n}$.
- Compute the stage values, using the characteristic form $u_t + uu_x = 0$

$$U_j^{(k)} = U_j^n - \Delta t \sum_{i=1}^{k-1} U_j^{(i)} D_x(U^{(i)})(x_j)$$

- Find the stage values in v: $V_j^{(k)} = \frac{1}{2}(U_j^{(k)})^2$. Reconstruct the boundary extrapolated data,

$$\left(V_{j+1/2}^{(k)}\right)^{\pm}$$

- Apply the conservative corrector step, evaluating the numerical flux F, consistent with $f = \frac{1}{3}\sqrt{(2v)^3}$, obtaining the new cell averages

$$\overline{V}_j^{n+1} = \overline{V}_j^n - \lambda \sum_{k=1}^{\nu} b_k \left(F_{j+1/2}^{(k)} - F_{j-1/2}^{(k)}\right)$$

Second equation (no name?),

$v = \frac{1}{2}u^2, v_t + (\frac{1}{3}\sqrt{(2v)^3})_x = 0$

Suppose you are given the cell averages $\overline{V}^n$

- Reconstruct the point values V_j^n and get $U_j^n = \sqrt{2V_j^n}$.
- Compute the stage values, using the characteristic form $u_t + uu_x = 0$

$$U_j^{(k)} = U_j^n - \Delta t \sum_{i=1}^{k-1} U_j^{(i)} D_x(U^{(i)})(x_j)$$

- Find the stage values in v: $V_j^{(k)} = \frac{1}{2}(U_j^{(k)})^2$. Reconstruct the boundary extrapolated data,

$$\left(V_{j+1/2}^{(k)}\right)^{\pm}$$

- Apply the conservative corrector step, evaluating the numerical flux F, consistent with $f = \frac{1}{3}\sqrt{(2v)^3}$, obtaining the new cell averages

$$\overline{V}_j^{n+1} = \overline{V}_j^n - \lambda \sum_{k=1}^{\nu} b_k \left(F_{j+1/2}^{(k)} - F_{j-1/2}^{(k)}\right)$$

Second equation (no name?),

$v = \frac{1}{2}u^2, v_t + (\frac{1}{3}\sqrt{(2v)^3})_x = 0$

Suppose you are given the cell averages $\overline{V}^n$

- Reconstruct the point values V_j^n and get $U_j^n = \sqrt{2V_j^n}$.
- Compute the stage values, using the characteristic form $u_t + uu_x = 0$

$$U_j^{(k)} = U_j^n - \Delta t \sum_{i=1}^{k-1} U_j^{(i)} D_x(U^{(i)})(x_j)$$

- Find the stage values in v: $V_j^{(k)} = \frac{1}{2}(U_j^{(k)})^2$. Reconstruct the boundary extrapolated data,

$$\left(V_{j+1/2}^{(k)}\right)^{\pm}$$

- Apply the conservative corrector step, evaluating the numerical flux F, consistent with $f = \frac{1}{3}\sqrt{(2v)^3}$, obtaining the new cell averages

$$\overline{V}_j^{n+1} = \overline{V}_j^n - \lambda \sum_{k=1}^{\nu} b_k \left(F_{j+1/2}^{(k)} - F_{j-1/2}^{(k)}\right)$$

Second equation (no name?),

$v = \frac{1}{2}u^2, v_t + (\frac{1}{3}\sqrt{(2v)^3})_x = 0$

Suppose you are given the cell averages $\overline{V}^n$

- Reconstruct the point values V_j^n and get $U_j^n = \sqrt{2V_j^n}$.
- Compute the stage values, using the characteristic form $u_t + uu_x = 0$

$$U_j^{(k)} = U_j^n - \Delta t \sum_{i=1}^{k-1} U_j^{(i)} D_x(U^{(i)})(x_j)$$

- Find the stage values in v: $V_j^{(k)} = \frac{1}{2}(U_j^{(k)})^2$. Reconstruct the boundary extrapolated data,

$$\left(V_{j+1/2}^{(k)}\right)^{\pm}$$

- Apply the conservative corrector step, evaluating the numerical flux F, consistent with $f = \frac{1}{3}\sqrt{(2v)^3}$, obtaining the new cell averages

$$\overline{V}_j^{n+1} = \overline{V}_j^n - \lambda \sum_{k=1}^{\nu} b_k \left(F_{j+1/2}^{(k)} - F_{j-1/2}^{(k)}\right)$$

Second equation (no name?),

$v = \frac{1}{2}u^2, v_t + (\frac{1}{3}\sqrt{(2v)^3})_x = 0$

Suppose you are given the cell averages $\overline{V}^n$

- Reconstruct the point values V_j^n and get $U_j^n = \sqrt{2V_j^n}$.
- Compute the stage values, using the characteristic form $u_t + uu_x = 0$

$$U_j^{(k)} = U_j^n - \Delta t \sum_{i=1}^{k-1} U_j^{(i)} D_x(U^{(i)})(x_j)$$

- Find the stage values in v: $V_j^{(k)} = \frac{1}{2}(U_j^{(k)})^2$. Reconstruct the boundary extrapolated data,

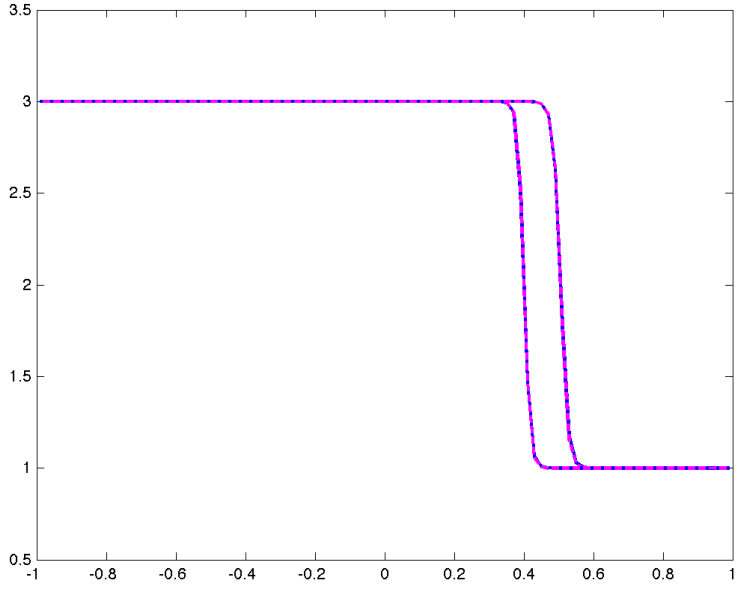
$$\left(V_{j+1/2}^{(k)}\right)^{\pm}$$

- Apply the conservative corrector step, evaluating the numerical flux F, consistent with $f = \frac{1}{3}\sqrt{(2v)^3}$, obtaining the new cell averages

$$\overline{V}_j^{n+1} = \overline{V}_j^n - \lambda \sum_{k=1}^{\nu} b_k \left(F_{j+1/2}^{(k)} - F_{j-1/2}^{(k)}\right)$$

Travelling discontinuity

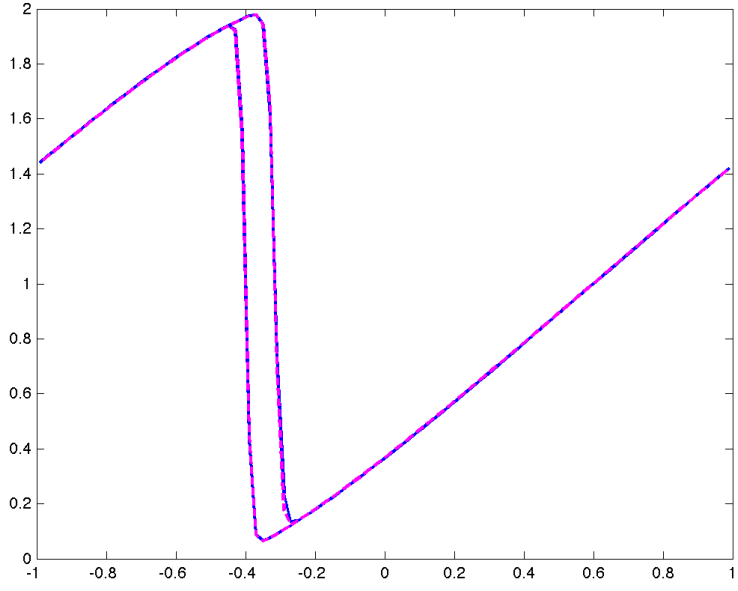
We start with an initial step, and we apply a standard FV scheme, and the new SC (Semi Conservative) scheme, using the two different equations. The correct shock locations are $x = 0.4$ and $x = 0.5$, respectively.



The different shock speeds are correctly caught by both schemes.

Shock formation

Starting with a smooth solution, we apply again a standard FV scheme, and the new SC scheme, to the two different equations,



we again find that the SC scheme gives the correct shock speeds in both cases, and the correct shock formation time.

Why does it work?

Conservative schemes

Let us start from the definition of conservative scheme, for $u_t + f_x(u) = 0$. A scheme is conservative if the numerical solution U can be written as

$$\overline{U}_j^{n+1} = \overline{U}_j^n - \frac{k}{h}\left(F_{j+1/2} - F_{j-1/2}\right)$$

where $F_{j+1/2} = F(\overline{U}_{j-p}^n, \ldots, \overline{U}_{j+k}^n)$, with $k, p > 0$ strictly positive integers, is the numerical flux function,

such that:

- $F(U, \ldots, U) = f(U)$, (consistency)
- $F(U, \ldots, U)$ is at least Lipschitz continuous in all of its arguments.

Conservative schemes

Let us start from the definition of conservative scheme, for $u_t + f_x(u) = 0$. A scheme is conservative if the numerical solution U can be written as

$$\overline{U}_j^{n+1} = \overline{U}_j^n - \frac{k}{h}\left(F_{j+1/2} - F_{j-1/2}\right)$$

where $F_{j+1/2} = F(\overline{U}_{j-p}^n, \ldots, \overline{U}_{j+k}^n)$, with $k, p > 0$ strictly positive integers, is the numerical flux function,
such that:

- $F(U, \ldots, U) = f(U)$, (consistency)
- $F(U, \ldots, U)$ is at least Lipschitz continuous in all of its arguments.

The Lax Wendroff theorem

Converging numerical solutions obtained with conservative schemes converge to correct weak solutions, because of Lax Wendroff theorem.

Lax Wendroff theorem

Let $U_h(x,t)$ be a numerical solution obtained on a grid of width h. Suppose that

- U_h has bounded variation;
- $U_h \to U$ as h goes to zero;
- U_h was obtained with a conservative scheme

Then the limit solution for $h \to 0$ is a weak solution of the conservation law.

Conservative schemes

In the proof:

- multiply the conservative form of the scheme by $\phi_j^n = \frac{1}{h\Delta t}\int_{V_j^n}\phi$, where ϕ is a smooth test function, sum over all grid points in space-time;

$$\sum_{j,n}\left[(U_j^{n+1}-U_j^n)\phi_j^n - \lambda(F_{j+1/2}^n - F_{j-1/2}^n)\phi_j^n\right] = 0$$

- sum by parts, discharging the differences from U and $F_{j+1/2}$ on ϕ;
- transform the sums in integrals, exploiting the definition of ϕ_j^n, and the fact that the numerical quantities are just numbers.
- pass to the limit for $h, \Delta t \to 0$, and hopefully get the weak form of the Conservation law.

Conservative schemes

In the proof:

- multiply the conservative form of the scheme by $\phi_j^n = \frac{1}{h\Delta t}\int_{V_j^n}\phi$, where ϕ is a smooth test function, sum over all grid points in space-time;
- sum by parts, discharging the differences from U and $F_{j+1/2}$ on ϕ;

$$\sum_{j,n}\left[(\phi_j^{n+1}-\phi_j^n)U_j^n - \lambda F_{j+1/2}^n(\phi_{j+1}^n-\phi_j^n)\right] = 0$$

- transform the sums in integrals, exploiting the definition of ϕ_j^n, and the fact that the numerical quantities are just numbers.
- pass to the limit for $h, \Delta t \to 0$, and hopefully get the weak form of the Conservation law.

Conservative schemes

In the proof:

- multiply the conservative form of the scheme by $\phi_j^n = \frac{1}{h\Delta t}\int_{V_j^n}\phi$, where ϕ is a smooth test function, sum over all grid points in space-time;
- sum by parts, discharging the differences from U and $F_{j+1/2}$ on ϕ;
- transform the sums in integrals, exploiting the definition of ϕ_j^n, and the fact that the numerical quantities are just numbers.

$$\sum_{j,n}\int_{V_j^n}\left[\frac{\phi(x,t+\Delta t)-\phi(x,t)}{\Delta t}U_j^n - F_{j+1/2}^n\frac{\phi(x+h,t)-\phi(x,t)}{h}\right]=0$$

- pass to the limit for $h, \Delta t \to 0$, and hopefully get the weak form of the Conservation law.

Conservative schemes

In the proof:

- multiply the conservative form of the scheme by $\phi_j^n = \frac{1}{h\Delta t}\int_{V_j^n}\phi$, where ϕ is a smooth test function, sum over all grid points in space-time;
- sum by parts, discharging the differences from U and $F_{j+1/2}$ on ϕ;
- transform the sums in integrals, exploiting the definition of ϕ_j^n, and the fact that the numerical quantities are just numbers.
- pass to the limit for $h, \Delta t \to 0$, and hopefully get the weak form of the Conservation law.

Conservative schemes

To transform this argument in a proof, you need to bound terms of the form

$$\left| \int_{V_j^n} \left[F(\overline{U}^n_{j-p}, \ldots, \overline{U}^n_{j+k}) - f(U(x,t)) \right] (\phi(x+h,t) - \phi(x,t)) \right|$$

$$\leq \int_{V_j^n} K \max_{-p \leq l \leq k} |U(x_j + lh, t) - U(x,t)| \; |\phi(x+h,t) - \phi(x,t)|$$

where we used the consistency and Lipshitz regularity (with constant K) of the numerical flux.

Conservative schemes

To transform this argument in a proof, you need to bound terms of the form

$$\left| \int_{V_j^n} \left[F(\overline{U}^n_{j-p}, \ldots, \overline{U}^n_{j+k}) - f(U(x,t)) \right] (\phi(x+h,t) - \phi(x,t)) \right|$$

$$\leq \int_{V_j^n} K \max_{-p \leq l \leq k} |U(x_j + lh, t) - U(x,t)| \; |\phi(x+h,t) - \phi(x,t)|$$

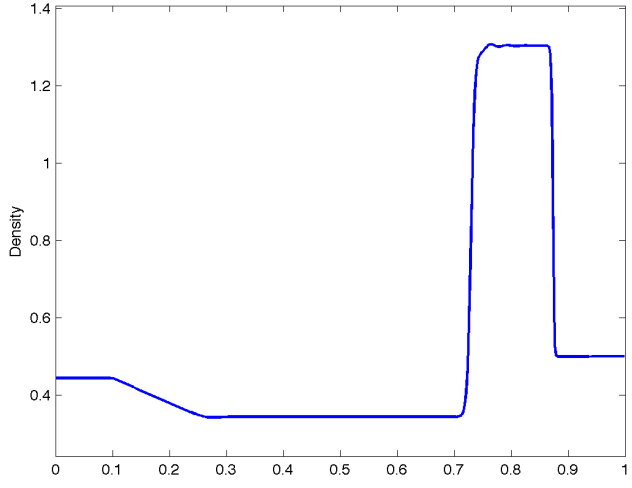
where we used the consistency and Lipshitz regularity (with constant K) of the numerical flux.

- Thus, the final conservative step is enough to ensure that the requirements of the Lax Wendroff theorem are satisfied.

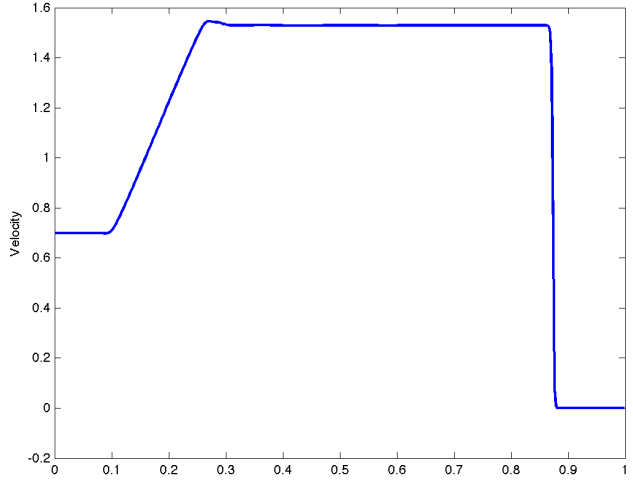
Lax' shock tube

Riemann Problem in gas dynamics. This is a standard test due to Lax.

Density



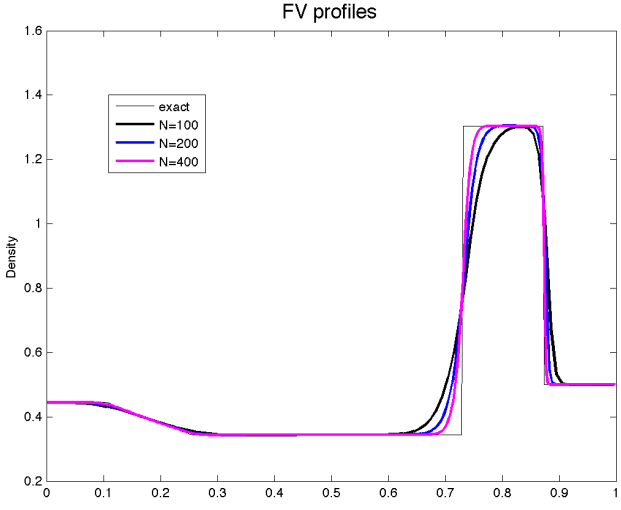
Velocity



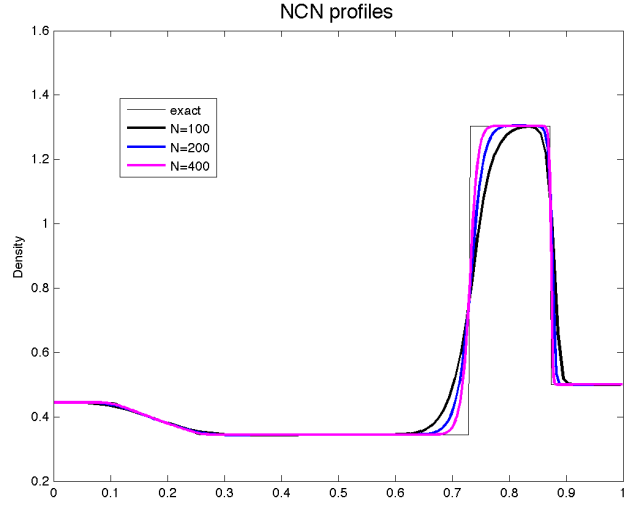
Lax' shock tube, Second order

Density profiles with second order schemes

Standard FV



SC scheme

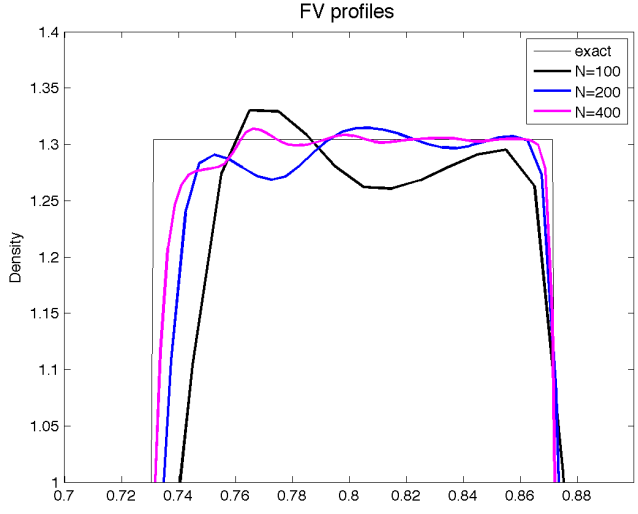


The solutions of the two schemes are almost identical.

Lax' shock tube, 4th order

Standard 4th order Finite Volume. Zoom on the density peak

Standard FV

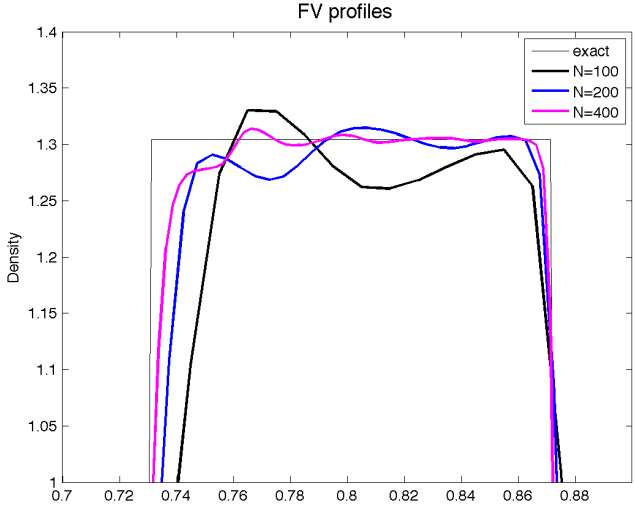


High order schemes produce small oscillations, whose amplitude decreases under grid refinement

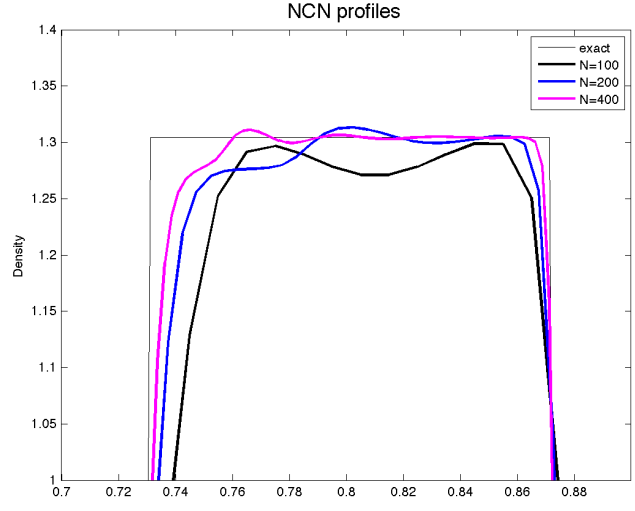
Lax' shock tube, 4th order

Standard 4th order Finite Volume, and new SC

Standard FV



SC scheme



This spurious effect is less pronounced on the SC profile.

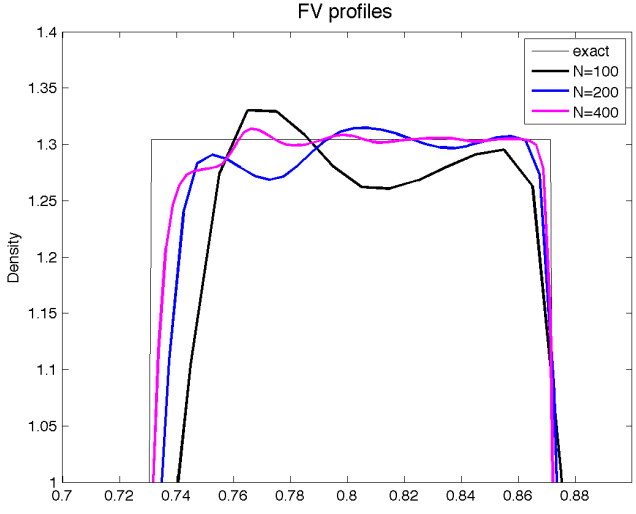
Lax' shock tube, 4th order

To diminish spurious oscillations, reconstruct projecting on characteristic directions

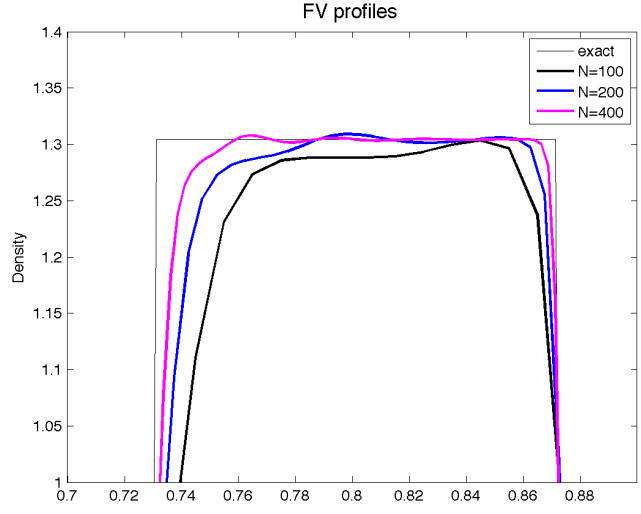
Lax' shock tube, 4th order

To diminish spurious oscillations, reconstruct projecting on characteristic directions

Standard FV without CP



Standard FV with CP

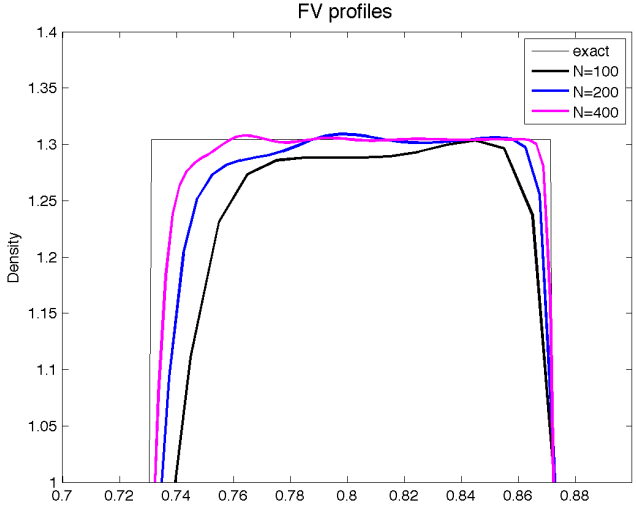


The oscillations have considerably decreased, with characteristic projection, and again the solution improves under grid refinement.

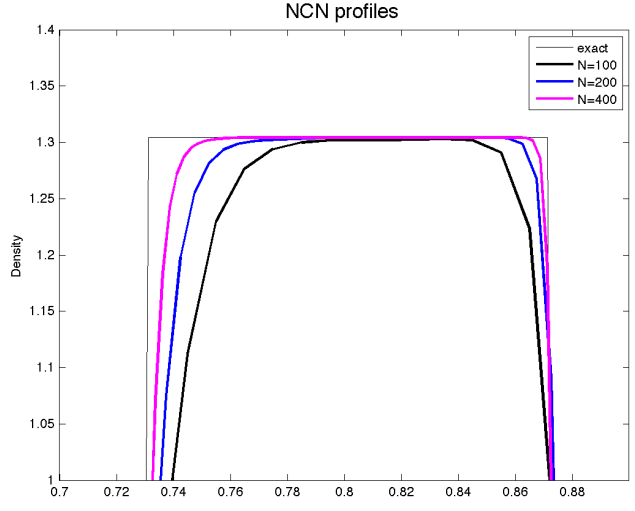
Lax' shock tube, 4th order

Standard 4th order Finite Volume, and new SC scheme

Standard FV with CP



SC scheme with CP



The spurious oscillations are flattened out on the SC profile.

A new family of schemes

We have shown the performance of a new class of schemes, which work under somewhat non-standard conditions.

- We understand why the SC schemes work. It is less apparent why they seem to be slightly less oscillatory than standard schemes.

A new family of schemes

We have shown the performance of a new class of schemes, which work under somewhat non-standard conditions.

- We understand why the SC schemes work. It is less apparent why they seem to be slightly less oscillatory than standard schemes.

Is that all?

OK, a nice toy. But does it have any real advantage over standard schemes?

- In some cases, SC schemes are much faster than standard FV schemes. To see why, we need to go

Is that all?

OK, a nice toy. But does it have any real advantage over standard schemes?

- In some cases, SC schemes are much faster than standard FV schemes. To see why, we need to go

Relativistic gas dynamics

RGD, Relativistic Gas Dynamics

RGD, Model equations

The system for RGD consists of conservation of mass D, momentum S and energy τ in the laboratory frame of reference. The equations are given by

$$\partial_t u + \nabla \cdot f(u) = 0$$

In one space dimension, the system is

$$u = \begin{pmatrix} D \\ S \\ \tau \end{pmatrix} \qquad f(u) = \begin{pmatrix} Dv \\ Sv + p \\ S - Dv \end{pmatrix} = 0$$

where v is the particle speed and p is the pressure.

Martì Müller

JCP (1996)

RGD, computing the flux

The problem is that, once D, S, τ are known, to compute the flux, one needs to find v, p, and this requires to solve the system

$$\begin{aligned} D &= \rho W \\ S &= \rho h W^2 v \qquad W = \frac{1}{\sqrt{1-v^2}} \\ \tau &= \rho h W^2 - p - D \end{aligned}$$

W is the Lorentz correction (we are considering the speed of light $c = 1$), ρ is mass at rest, and h is the enthalpy. We still need

$$\begin{aligned} p &= (\gamma - 1)\rho e \\ h &= 1 + e + p/\rho \end{aligned}$$

As $v \to 0$, classical mechanics holds, and one recovers standard compressible gas dynamics.

RGD, application of standard FV schemes

It is not too difficult to see that to compute the flux once D, S, τ are known requires to solve a non linear system of equations. More precisely, one needs to solve a non-linear equation for the pressure, $\mathcal{P}(p(D, S, \tau)) = 0$. Fortunately, $\mathcal{P}(p)$ is a monotone function, and it has a single zero, for admissible (D, S, τ). So we have

- For standard FV scheme, given the cell averages $\overline{D}^n, \overline{S}^n, \overline{\tau}^n$, one needs to compute the ν stage values, and each stage value requires the solution of $\mathcal{P}(p(D^{(i)}, S^{(i)}, \tau^{(i)})) = 0$

RGD, application of SC schemes

Again, we are given the cell averages $\overline{D}^n, \overline{S}^n, \overline{\tau}^n$. First, compute the point values D^n, S^n, τ^n. It is then necessary to invert again $\mathcal{P}(p(D,S,\tau)) = 0$, but this is done only once per time step.

- The stages in fact are computed updating the non conservative system for the primitive variables ρ, v, p

$$\begin{pmatrix} \rho \\ v \\ p \end{pmatrix}_t + \begin{pmatrix} v & \rho\mathcal{V} & -\frac{v}{hW^2}\mathcal{V} \\ 0 & v\mathcal{C}\mathcal{V} & \frac{\mathcal{V}}{\rho h W^4} \\ 0 & \rho h c^2 \mathcal{V} & v\mathcal{C}\mathcal{V} \end{pmatrix} \begin{pmatrix} \rho \\ v \\ p \end{pmatrix}_x = 0$$

where $c^2 = \gamma p/(\rho h)$, $\mathcal{V} = 1/(1-c^2v^2)$ and $\mathcal{C} = 1 - c^2$ Once the stage values $\rho^{(i)}, v^{(i)}, p^{(i)})$ are known, the stages for the conservative variables are easily found, $D^{(i)}, S^{(i)}, \tau^{(i)}$.

RGD, summary

We have two sets of variables: conservative variables D, S, τ and primitive variables ρ, v, p, linked by the diffeomorphism $(D, S, \tau) = \mathcal{M}(\rho, v, p)$.

- The direct map $\mathcal{M}$ is easy to compute. The inverse map $\mathcal{M}^{-1}$ is computationally expensive.
- Standard FV with Runge Kutta time integration requires to evaluate $\mathcal{M}^{-1}$ at each stage.
- For SC schemes with Runge Kutta time integration one needs to evaluate $\mathcal{M}^{-1}$ at the beginning of each time step, and then $\mathcal{M}$ at each stage. Much faster.

RGD, summary

We have two sets of variables: conservative variables D, S, τ and primitive variables ρ, v, p, linked by the diffeomorphism $(D, S, \tau) = \mathcal{M}(\rho, v, p)$.

- The direct map $\mathcal{M}$ is easy to compute. The inverse map $\mathcal{M}^{-1}$ is computationally expensive.
- Standard FV with Runge Kutta time integration requires to evaluate $\mathcal{M}^{-1}$ at each stage.
- For SC schemes with Runge Kutta time integration one needs to evaluate $\mathcal{M}^{-1}$ at the beginning of each time step, and then $\mathcal{M}$ at each stage. Much faster.

RGD, summary

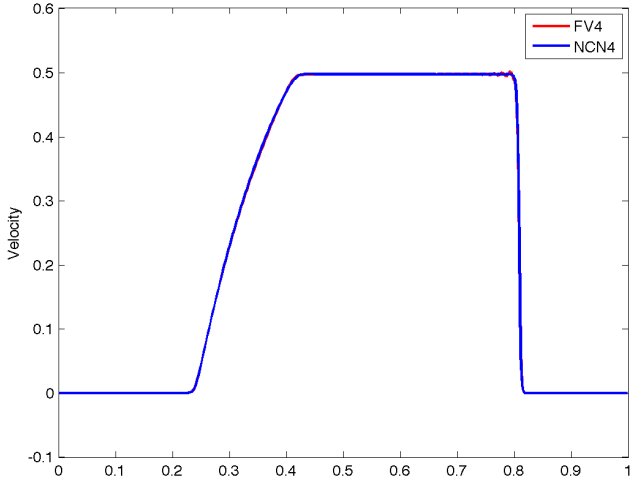
We have two sets of variables: conservative variables D, S, τ and primitive variables ρ, v, p, linked by the diffeomorphism $(D, S, \tau) = \mathcal{M}(\rho, v, p)$.

- The direct map $\mathcal{M}$ is easy to compute. The inverse map $\mathcal{M}^{-1}$ is computationally expensive.
- Standard FV with Runge Kutta time integration requires to evaluate $\mathcal{M}^{-1}$ at each stage.
- For SC schemes with Runge Kutta time integration one needs to evaluate $\mathcal{M}^{-1}$ at the beginning of each time step, and then $\mathcal{M}$ at each stage. Much faster.

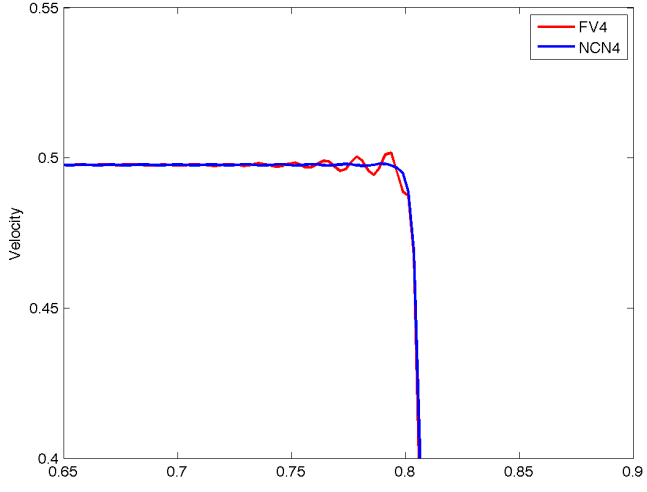
Relativistic shock tube, 4th order

Relativistic shock tube problem, $V_L = [10, 0, 13.3]$ while $V_R = [1, 0, 1]$. Recall that $V = [\rho, v, p]$.

Velocity



Zoom on the shock

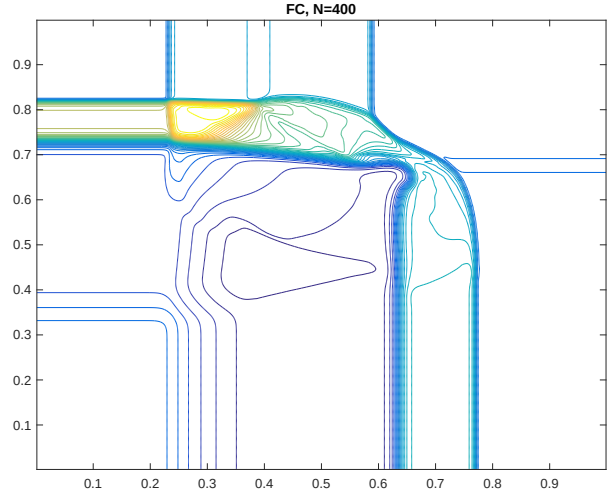


Again the SC solution is less oscillatory

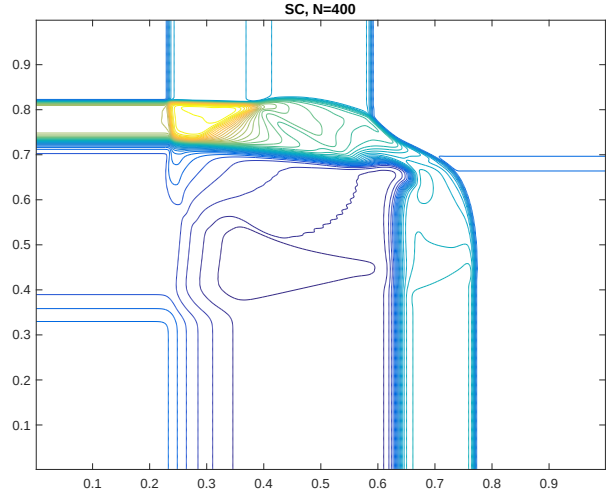
Relativistic shock tube, 2nd order

Relativistic 2D shock tube problem, second order schemes, $N = 400$.

Density, FV2



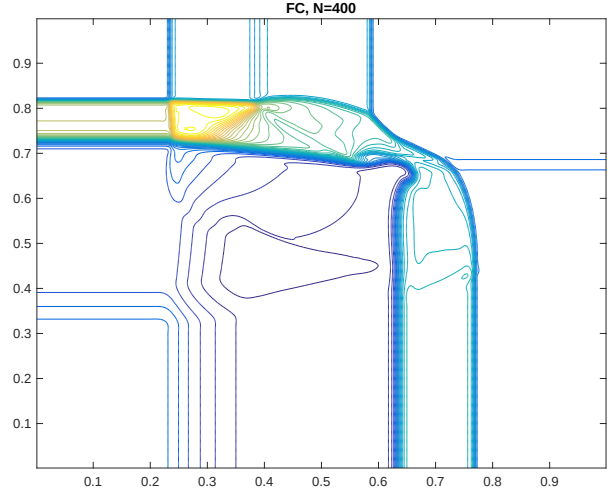
Density, SC2



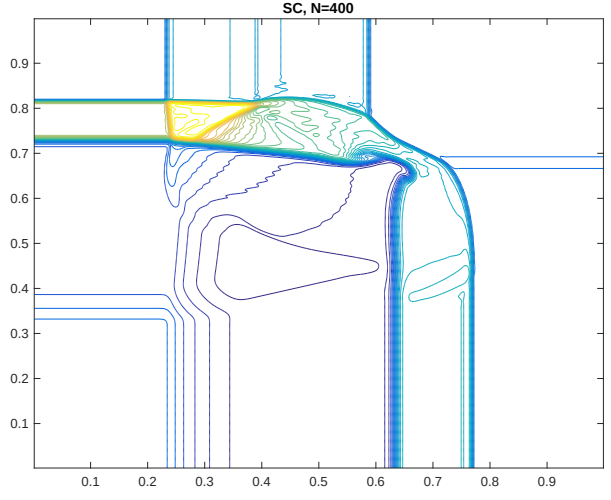
Relativistic shock tube, 3rd order

Relativistic 2D shock tube problem, third order schemes, $N = 400$.

Density, FV3



Density, SC3



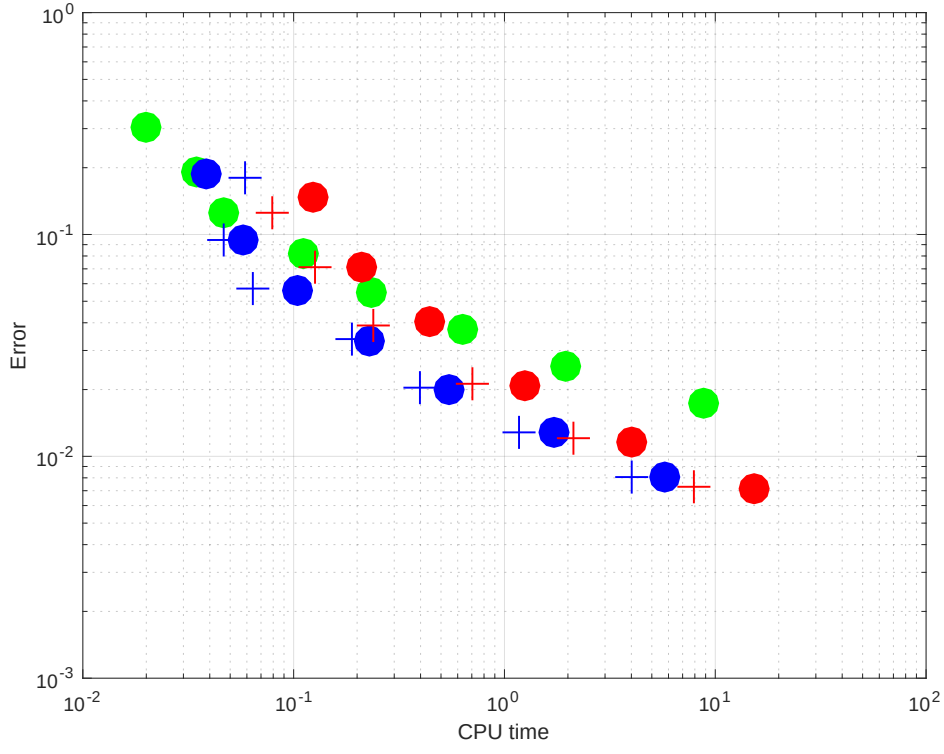
The third order reconstruction here is CWENO

Ok, it works, but...

Does it pay?

Ok, it works, but...

Does it pay?
Error vs Computational cost (seconds of CPU)



Green, red, blue: 1, 2 and 3rd order; Circle: FV, Plus: SC

We are almost there

We have presented an idea which introduces a large flexibility in schemes based on the method of lines approach for hyperbolic problems.

- We can optimize. I mean, primitive variables are not the only possible choice to build SC schemes. One could use, for instance, characteristic variables, possibly decreasing oscillations even further.
- We plan to apply these ideas to implicit schemes, for instance in the low Mach regime.
- These computations prompt, once more, the need for really non oscillatory high order reconstructions.

We are almost there

We have presented an idea which introduces a large flexibility in schemes based on the method of lines approach for hyperbolic problems.

- We can optimize. I mean, primitive variables are not the only possible choice to build SC schemes. One could use, for instance, characteristic variables, possibly decreasing oscillations even further.
- We plan to apply these ideas to implicit schemes, for instance in the low Mach regime.
- These computations prompt, once more, the need for really non oscillatory high order reconstructions.

We are almost there

We have presented an idea which introduces a large flexibility in schemes based on the method of lines approach for hyperbolic problems.

- We can optimize. I mean, primitive variables are not the only possible choice to build SC schemes. One could use, for instance, characteristic variables, possibly decreasing oscillations even further.
- We plan to apply these ideas to implicit schemes, for instance in the low Mach regime.
- These computations prompt, once more, the need for really non oscillatory high order reconstructions.

Background


Pidatella, P. Russo, Santagati, Semi-Conservative Finite Volume Schemes for Conservation Laws,
SISC (2019).


Martì, Müller, Numerical Hydrodynamics in Special Relativity,
Living Reviews in relativity, (2003)


J. Zhao, H. Tang, Central Runge-Kutta Discontinuous Galerkin Methods for the Special Relativistic Hydrodynamics,
CiCp (2017)


Zanotti, Dumbser, Efficient conservative ADER schemes based on WENO reconstruction and space-time predictor in primitive variables.
Computational Astrophysics and Cosmology, (2016)


Fambri, Dumbser, Köppel, Rezzolla, Zanotti, ADER discontinuous Galerkin schemes for general-relativistic ideal magnetohydrodynamics
Monthly Notices of the Royal Astronomical Society, (2018)

Thank you!

Gabriella Puppo* – gabriella.puppo@uniroma1.it

Dipartimento di Matematica
La Sapienza, Università di Roma
Piazzale Aldo Moro 5, 00185 Roma

www.gabriellapuppo.it

---

*With: **Isabella Cravero** (Università di Torino, Torino, Italy) **Matteo Semplice** (Università dell'Insubria, Como, Italy), **Giuseppe Visconti** (La Sapienza Università di Roma, Italy), Rosamaria Pidatella e Giovanni Russo (Università di Catania, Catania, Italy)