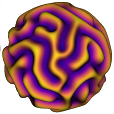
A class of high-order non-polynomial finite volume methods

February 5, 2021

Ian May, Dongwook Lee

Department of Applied Mathematics
University of California Santa Cruz

Santa Cruz, CA



Goal

Solve systems of hyperbolic conservation laws

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}) = 0$$

with an *accurate* and *robust* finite volume method

$$\frac{\partial \langle \mathbf{U} \rangle_\Omega}{\partial t} = \frac{1}{|\Omega|} \int_{\partial\Omega} \hat{\mathbf{F}} \left(\mathbf{U}^-(\mathbf{x}), \mathbf{U}^+(\mathbf{x}) \right) \cdot \mathbf{n} dx$$

in *multiple* dimensions.

For today, consider

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial}{\partial x}\mathbf{F}(\mathbf{U}) + \frac{\partial}{\partial y}\mathbf{G}(\mathbf{U}) = 0$$

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{pmatrix} \quad \mathbf{F}(\mathbf{U}) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ \rho u w \\ u(E+p) \end{pmatrix} \quad \mathbf{G}(\mathbf{U}) = \begin{pmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ \rho v w \\ v(E+p) \end{pmatrix},$$

for a calorically ideal gas,

$$p = (\gamma - 1)\rho\epsilon, \quad \epsilon = \frac{E}{\rho} - \frac{\mathbf{v}\cdot\mathbf{v}}{2}.$$

Quick overview of FVM

Abstract formulation

Partition full domain Ω into *finite volumes* Ω_i such that $\Omega = \bigcup_i \Omega_i$, and $\Omega_i \cap \Omega_j = \varnothing,\ i \neq j$. Denote

$$\langle \cdot \rangle_i = \frac{1}{||\Omega_i||} \int_{\Omega_i} \cdot d\mathbf{x},$$

then for (systems of) hyperbolic conservation laws

$$\frac{\partial}{\partial t} \langle \mathbf{U} \rangle_i = -\frac{1}{||\Omega_i||} \oint_{\partial \Omega_i} \hat{\mathbf{F}} \left(\mathbf{U}^-, \mathbf{U}^+ \right) \cdot \mathbf{n} ds$$

for numeric flux $\hat{\mathbf{F}}$, and states $\mathbf{U}^-$ and $\mathbf{U}^+$ inside and outside Ω_i.

Quick overview of FVM

Uniform 2D Cartesian grids

Let $\Omega_{i,j} = \left[x_i - \frac{\Delta x}{2}, x_i - \frac{\Delta x}{2}\right] \times \left[y_i - \frac{\Delta y}{2}, y_j - \frac{\Delta y}{2}\right]$, then

$$\frac{\partial}{\partial t}\langle \mathbf{U} \rangle_{i,j} = -\frac{1}{||\Omega_{i,j}||} \oint\limits_{\partial \Omega_{i,j}} \hat{\mathbf{F}}\left(\mathbf{U}^-, \mathbf{U}^+\right) \cdot \mathbf{n} ds$$

$$= -\frac{1}{\Delta x}\left(\langle \hat{\mathbf{F}} \rangle_{i+\frac{1}{2},j} - \langle \hat{\mathbf{F}} \rangle_{i-\frac{1}{2},j}\right) - \frac{1}{\Delta y}\left(\langle \hat{\mathbf{G}} \rangle_{i,j+\frac{1}{2}} - \langle \hat{\mathbf{G}} \rangle_{i,j-\frac{1}{2}}\right)$$

where half-indices indicate integration over faces.

Two barriers to high order in multiple dimensions

- Face integral must be done accurately
- Numerical flux is defined *pointwise*, thus need accurate *pointwise* values of $\mathbf{U}^{\pm}_{i\pm 1/2}$

Issues with polynomials

- Matching stencils to multivariate polynomial spaces is hard
- Forming valid substencils for WENO is even harder
- Dimension-by-dimension approaches do work, but get messy

[1] Omitting many technical details

Issues with polynomials

- Matching stencils to multivariate polynomial spaces is hard
- Forming valid substencils for WENO is even harder
- Dimension-by-dimension approaches do work, but get messy

Kernel based interpolation/recovery

Each SPD *kernel* $K : \Omega \times \Omega \to \mathbb{R}$, induces a *reproducing kernel Hilbert space*[1], $\mathcal{H}$, consisting of

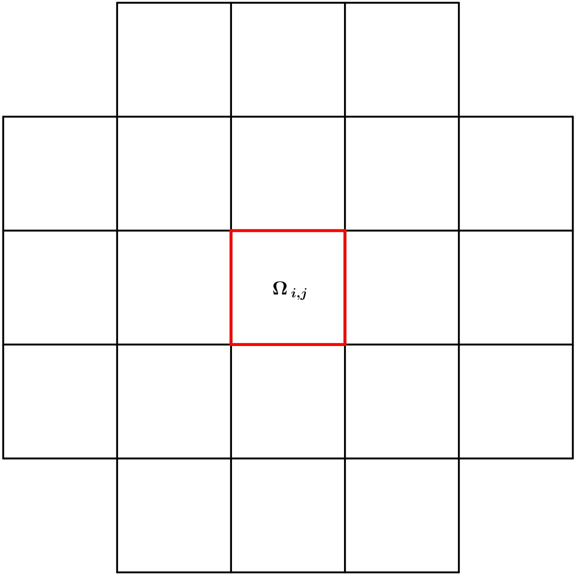
$$f(x) = \sum_i a_i K(x, x_i)$$

$$\sum_i \sum_j a_i a_j K(x_i, x_j) < \infty$$

For this talk: $K(x, y) = e^{-\frac{||x-y||^2}{2\ell^2}}$.

[1] Omitting many technical details

An exemplary stencil: $R = 2$



Let $f \in \mathcal{H}$ with known values $y_i = f(x_i)$ for $x_i \in \Omega,\ i = 1, \ldots, N$.
Seek an interpolant of the form:

$$\widetilde{f}(x) = \sum_{j=1}^{N} \alpha_j K(x, x_j)$$

then enforcing that $\widetilde{f}(x_i) = y_i$ gives that the coefficients satisfy

$$[K(x_i, x_j)]\, \boldsymbol{\alpha} = \mathbf{y}.$$

Let $f \in \mathcal{H}$ with known values $y_i = f(x_i)$ for $x_i \in \Omega,\ i = 1, \ldots, N$.
Seek an interpolant of the form:

$$\widetilde{f}(x) = \sum_{j=1}^{N} \alpha_j K(x, x_j)$$

then enforcing that $\widetilde{f}(x_i) = y_i$ gives that the coefficients satisfy

$$[K(x_i, x_j)]\,\boldsymbol{\alpha} = \mathbf{y}.$$

Properties and interpretation of $\widetilde{f}$

Let $\mathcal{H}_0 = \mathrm{span}\{K(\cdot, x_i)\} \subset \mathcal{H}$.

- $(f - \widetilde{f}) \perp \mathcal{H}_0$
- $\widetilde{f}$ is the *optimal* approximant in $\mathcal{H}_0$
- For noise-free y_i, $\widetilde{f}$ is also the *best linear unbiased estimate* of f
- $\widetilde{f}$ is the posterior mean function of $\mathcal{GP}(0, K)$ conditioned on $\mathbf{y}$

What can we do when we do not know point values of f?

What can we do when we do not know point values of f?
Let $\{\lambda_i\} \subset \mathcal{H}'$ be linearly independent, and $y_i = \lambda_i f$ known.
Seek an interpolant of the form:

$$\widetilde{f}(x) = \sum_{j=1}^{N} \alpha_j \lambda_j^{(y)} K(x, y)$$

then enforcing that $\lambda_i^{(x)} \widetilde{f}(x) = y_i$, requires that $\boldsymbol{\alpha}$ satisfy

$$\left[\lambda_i^{(x)} \lambda_j^{(y)} K(x_i, x_j)\right] \boldsymbol{\alpha} = \mathbf{y}.$$

What can we do when we do not know point values of f?
Let $\{\lambda_i\} \subset \mathcal{H}'$ be linearly independent, and $y_i = \lambda_i f$ known.
Seek an interpolant of the form:

$$\widetilde{f}(x) = \sum_{j=1}^{N} \alpha_j \lambda_j^{(y)} K(x, y)$$

then enforcing that $\lambda_i^{(x)} \widetilde{f}(x) = y_i$, requires that $\boldsymbol{\alpha}$ satisfy

$$\left[\lambda_i^{(x)} \lambda_j^{(y)} K(x_i, x_j)\right] \boldsymbol{\alpha} = \mathbf{y}.$$

Relationship to regular interpolation

- $\lambda_j^{(y)} K(x, y) \in \mathcal{H}$, hence $\widetilde{f} \in \mathcal{H}$
- $(f - \widetilde{f}) \perp \mathcal{H}_0$, but $\mathcal{H}_0$ is different
- Using point evaluation functionals, $\lambda_j = \delta_{x_j}$, recovers former result

For FVMs the relevant linear functionals are given by cell-averages. Thus we need to solve

$$\left[\frac{1}{||\Omega_i||}\frac{1}{||\Omega_j||}\int_{\Omega_i}\int_{\Omega_j} K(x,y)dxdy\right]\boldsymbol{\alpha} = \mathbf{y},$$

and evaluating the interpolant at x^* gives

$$\widetilde{f}(x) = \sum_{j=1}^{N}\alpha_j \int_{\Omega_j} K(x^*,y)dy = \mathbf{z}^T\mathbf{y}$$

where the *prediction vector* is given by:

$$\mathbf{z}^T = \left[\int_{\Omega_j} K(x^*,y)dy\right]^T \left[\frac{1}{||\Omega_i||}\frac{1}{||\Omega_j||}\int_{\Omega_i}\int_{\Omega_j} K(x,y)dxdy\right]^{-1} \mathbf{y}$$

Stabilizing large ℓ

We need to compute

$$\mathbf{z}^T = \mathbf{w}^T \mathbf{C}^{(-1)},$$

where $\mathbf{C}$ and $\mathbf{w}$ both depend on ℓ.

- Large values of ℓ tend to give more accurate interpolants
- Large values of ℓ give horribly conditioned linear systems

Stabilizing large ℓ

We need to compute

$$\mathbf{z}^T = \mathbf{w}^T \mathbf{C}^{(-1)},$$

where $\mathbf{C}$ and $\mathbf{w}$ both depend on ℓ.

- Large values of ℓ tend to give more accurate interpolants
- Large values of ℓ give horribly conditioned linear systems

Stable evaluation of prediction vectors

Consider $\epsilon = \ell^{-1}$, and allow complex ϵ. Then

- $z_i(\ell^{-1}) = \mathbf{w}^T \mathbf{C}^{(-1)} \mathbf{e}_i$ is holomorphic apart from isolated poles
- Evaluate $z_i(\ell^{-1})$ on a circle in $\mathbb{C}$ where computation is stable
- Back out an approximate Laurent expansion of $z_i(\ell^{-1})$
- Evaluate that Laurent expansion at the real $\epsilon = \ell^{-1}$ of interest

- We can now obtain accurate point estimates of the solution
- Call an (approximate) Riemann solver to find pointwise fluxes
- But *where* should we do this?

- We can now obtain accurate point estimates of the solution
- Call an (approximate) Riemann solver to find pointwise fluxes
- But *where* should we do this?

Buchmuller-Helzel correction

Generate pointwise fluxes at the center of each face, fit a polynomial in the transverse direction, integrate that polynomial exactly.

Accurate flux integrals
Transverse corrections

- We can now obtain accurate point estimates of the solution
- Call an (approximate) Riemann solver to find pointwise fluxes
- But *where* should we do this?

Buchmuller-Helzel correction

Generate pointwise fluxes at the center of each face, fit a polynomial in the transverse direction, integrate that polynomial exactly.

Fit another Gaussian process

Use a similar stencil as Buchmuller-Helzel, but fit a GP through the fluxes and integrate it exactly.

Accurate flux integrals
Transverse corrections

- We can now obtain accurate point estimates of the solution
- Call an (approximate) Riemann solver to find pointwise fluxes
- But *where* should we do this?

Buchmuller-Helzel correction

Generate pointwise fluxes at the center of each face, fit a polynomial in the transverse direction, integrate that polynomial exactly.

Fit another Gaussian process

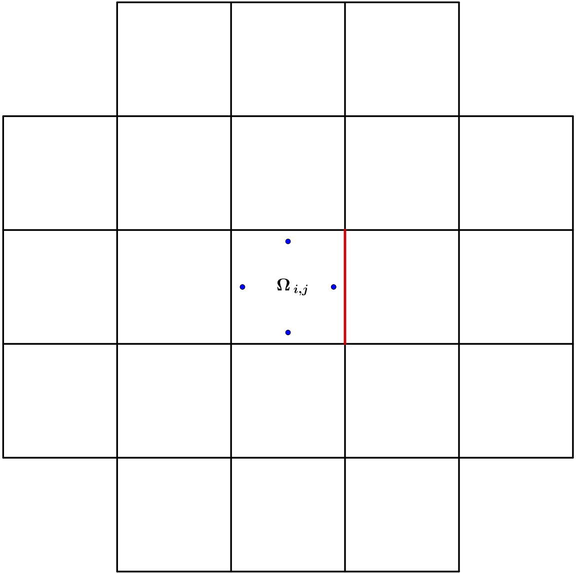
Use a similar stencil as Buchmuller-Helzel, but fit a GP through the fluxes and integrate it exactly.

Gaussian quadrature

Solve multiple Riemann problems on each face, and approximate flux integral with a Gaussian quadrature rule.

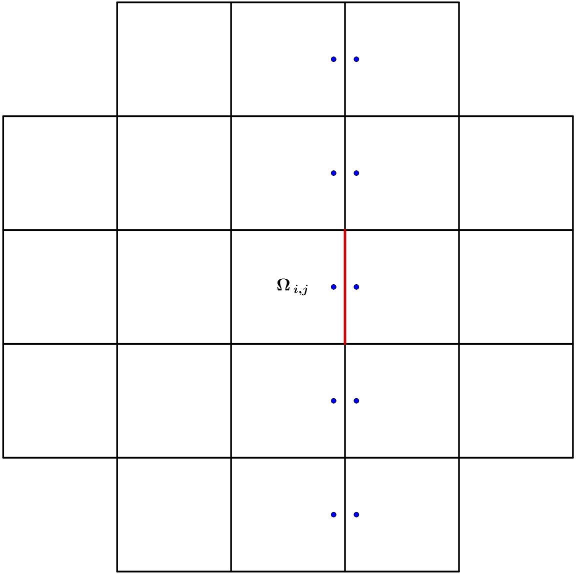
Graphical summary of the method

Find Riemann states at each face of $\Omega_{i,j}$

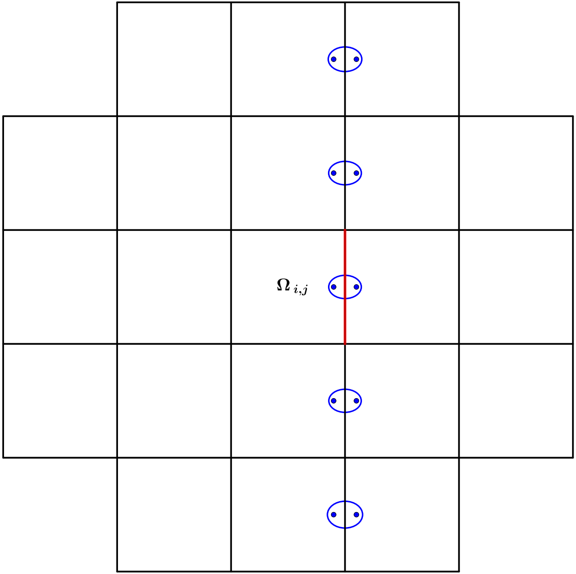


Graphical summary of the method

Find Riemann states for all other $\Omega_{i,j}$



Graphical summary of the method
Call Riemann solver, and perform transverse integration



A truly nonlinear benchmark problem

The Euler equations on $[-L, L]^2$ with periodic boundaries and initial condition

$$\begin{pmatrix} \rho \\ u \\ v \\ p \end{pmatrix} = \begin{pmatrix} T^{1/(\gamma-1)} \\ 1 - y\omega \\ 1 + x\omega \\ T^{\gamma/(\gamma-1)} \end{pmatrix}$$

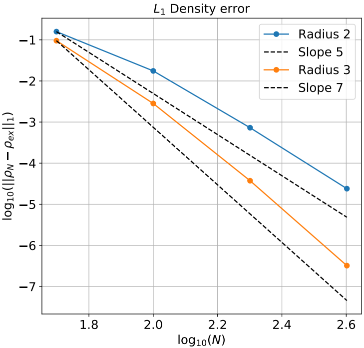
$$T = 1 - \frac{\gamma - 1}{8\gamma\pi^2} e^{1-x^2-y^2}$$

$$\omega = \frac{1}{2\pi} e^{(1-x^2-y^2)/2}$$

recover the initial condition at time $T_f = 2L$

The isentropic vortex problem

$\Omega = [-10, 10]^2$, $\ell = 4$, Linear scheme



Grid	L_1 Error	L_1 Order	L_∞ Error	L_∞ Order
		$R = 2$		
50^2	$1.58e-1$	–	$2.45e-2$	–
100^2	$1.75e-2$	**3.17**	$4.99e-3$	**2.30**
200^2	$7.28e-4$	**4.59**	$1.15e-4$	**5.44**
400^2	$2.40e-5$	**4.92**	$3.94e-6$	**4.87**
		$R = 3$		
50^2	$9.54e-2$	–	$2.04e-2$	–
100^2	$2.83e-3$	**5.08**	$3.77e-4$	**5.75**
200^2	$3.74e-5$	**6.24**	$9.99e-6$	**5.24**
400^2	$3.22e-7$	**6.86**	$9.11e-8$	**6.78**

Dealing with shocks: WENO methods

Nonlinear GP reconstruction

The reconstruction presented is linear, i.e.

$$\mathbf{U}_{i+\frac{1}{2},j} = \widetilde{\mathbf{U}}(x^*) = \mathbf{z}^T \left[\langle \mathbf{U} \rangle\right]_{S(i,j)}$$

which is hopeless near discontinuities (Godunov)

Nonlinear GP reconstruction

The reconstruction presented is linear, i.e.

$$\mathbf{U}_{i+\frac{1}{2},j} = \widetilde{\mathbf{U}}(x^*) = \mathbf{z}^T \left[\langle \mathbf{U} \rangle\right]_{S(i,j)}$$

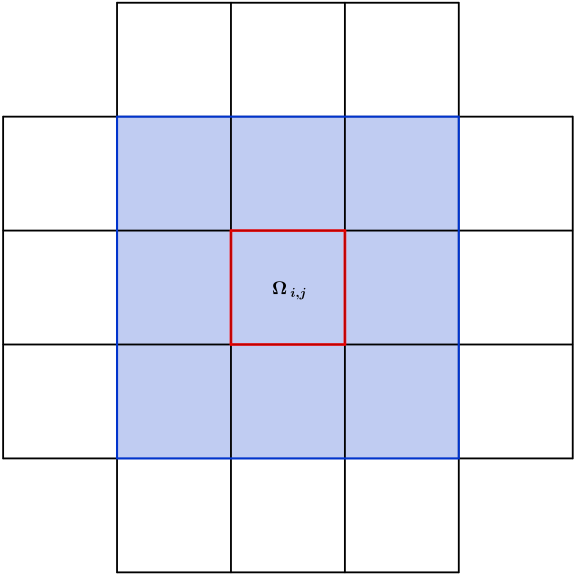
which is hopeless near discontinuities (Godunov)

WENO (weighted essentially non-oscillatory) methods

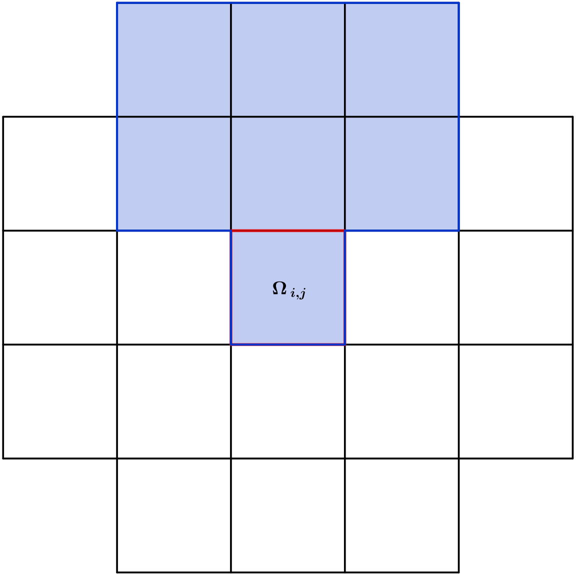
Break full stencil into substencils, use weighted combination of individual reconstructions

$$\mathbf{U}_{i+\frac{1}{2},j} = \sum_{S_k \in \mathcal{S}_{i,j}} \omega_k \widetilde{\mathbf{U}}_k(x^*)$$

where $\mathcal{S}_{i,j}$ is set of substencils, and ω_k depends on the data in S_k.



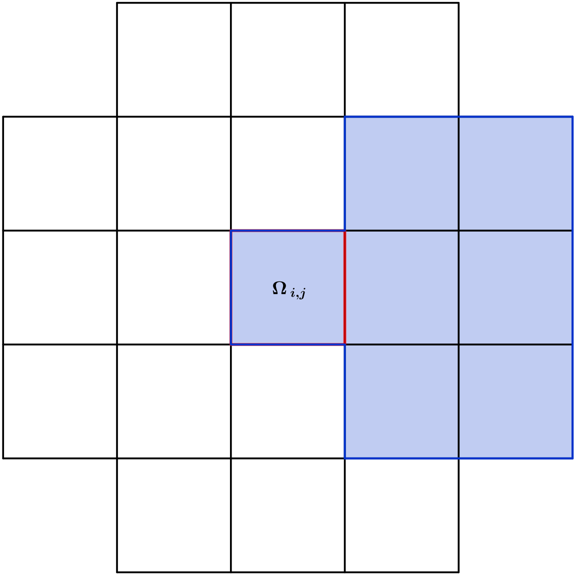
$\Omega_{i,j}$

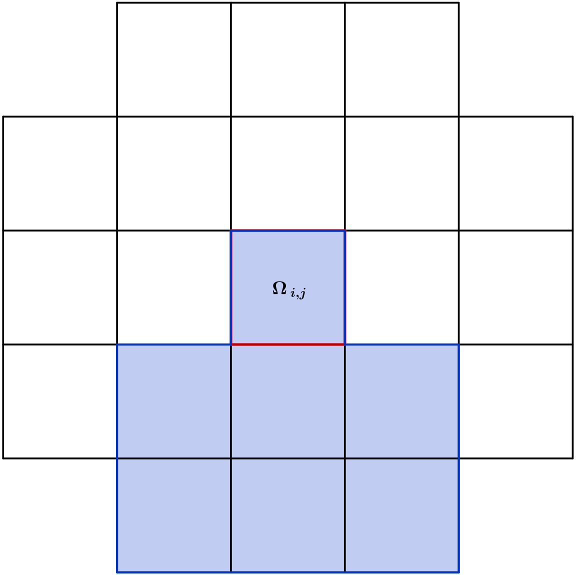


$\Omega_{i,j}$

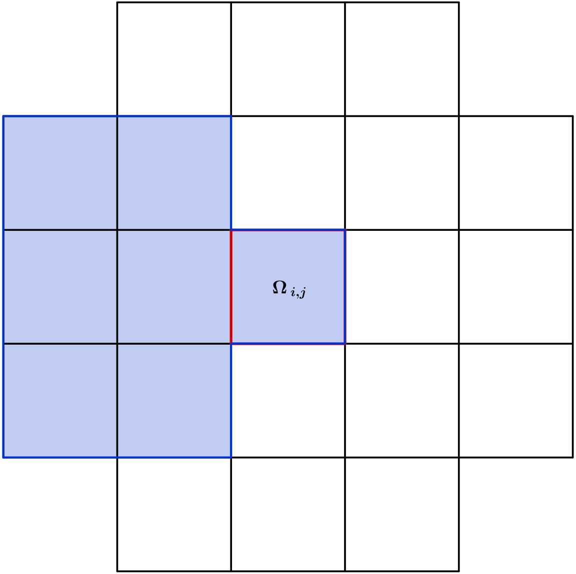
S_3: East substencil

Substencils in the spirit of standard WENO





$\Omega_{i,j}$



$\mathbf{\Omega}_{i,j}$

Optimal weights and standard WENO

The optimal linear weights γ_k minimize discrepancy in

$$\widetilde{\mathbf{U}}(x^*) \approx \sum_{k=1}^{5} \gamma_k \widetilde{\mathbf{U}}_k(x^*)$$

independent of the data.

Optimal weights and standard WENO

The optimal linear weights γ_k minimize discrepancy in

$$\widetilde{\mathbf{U}}(x^*) \approx \sum_{k=1}^{5} \gamma_k \widetilde{\mathbf{U}}_k(x^*)$$

independent of the data.

Special cases: Polynomial reconstruction

For *some* polynomial degrees on *some* (sub)stencil choices, equality can be obtained (e.g. classical WENO5).

Optimal weights and standard WENO

The optimal linear weights γ_k minimize discrepancy in

$$\widetilde{\mathbf{U}}(x^*) \approx \sum_{k=1}^{5} \gamma_k \widetilde{\mathbf{U}}_k(x^*)$$

independent of the data.

Special cases: Polynomial reconstruction

For *some* polynomial degrees on *some* (sub)stencil choices, equality can be obtained (e.g. classical WENO5).

Desired behavior of ω_k

- For smooth data $\omega_k \approx \gamma_k$ on all substencils
- For rough data $\omega_k \approx 0$ on rough substencils

This is obtained by use of *smoothness indicators*.

WENO-AO

Generally, no linear weights, γ_k, exist that can reproduce the accuracy of the full stencil.

Adaptive order WENO

Let S_0 correspond to the full stencil, and include it in the nonlinear reconstruction:

$$\mathbf{U}_{i+\frac{1}{2},j} = \frac{\omega_0}{\gamma_0}\widetilde{\mathbf{U}}_0(x^*) + \sum_{k=1}^{5}\left(\omega_k - \omega_0\frac{\gamma_k}{\gamma_0}\right)\widetilde{\mathbf{U}}_k(x^*)$$

WENO-AO

Generally, no linear weights, γ_k, exist that can reproduce the accuracy of the full stencil.

Adaptive order WENO

Let S_0 correspond to the full stencil, and include it in the nonlinear reconstruction:

$$\mathbf{U}_{i+\frac{1}{2},j} = \frac{\omega_0}{\gamma_0}\widetilde{\mathbf{U}}_0(x^*) + \sum_{k=1}^{5}\left(\omega_k - \omega_0\frac{\gamma_k}{\gamma_0}\right)\widetilde{\mathbf{U}}_k(x^*)$$

Now we can choose γ_k to ensure stability

$$\begin{aligned}
\gamma_0 &= C_h, \\
\gamma_1 &= (1 - C_h)C_l, \\
\gamma_2 = \gamma_3 = \gamma_4 = \gamma_5 &= \frac{(1 - C_h) * (1 - C_l)}{4},
\end{aligned}$$

where $0 < C_h, C_l < 1$, e.g. $C_h = C_l = 0.8$.

The smoothness of the solution on each substencil can be measured by

$$\beta_k = \sum_{r=1}^{2} \sum_{|\alpha|=r} \left(\left. \frac{\partial^{|\alpha|} \widetilde{U}_k}{\partial x^\alpha} \right|_{(x_i, y_j)} \right)^2,$$

Then nonlinear weights are formed using a modified WENO-Z scheme

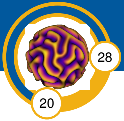
$$\tau = \frac{1}{5} \sum_{k=1}^{5} |\beta_0 - \beta_k|$$

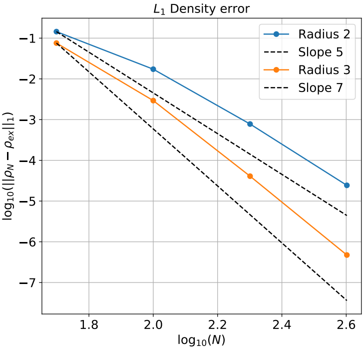
$$\widetilde{\omega}_k = \gamma_k \left(1 + \left(\frac{\tau}{\beta_k + \epsilon} \right)^p \right)$$

$$\omega_k = \frac{\widetilde{\omega}_k}{\sum \widetilde{\omega}_k}$$

The isentropic vortex problem

$\Omega = [-10, 10]^2$, $\ell = 4$, WENO





Grid	L_1 Error	L_1 Order	L_∞ Error	L_∞ Order
		$R = 2$		
50^2	$1.46e-1$	–	$2.41e-2$	–
100^2	$1.73e-2$	**3.06**	$5.00e-3$	**2.27**
200^2	$7.78e-4$	**4.47**	$1.15e-4$	**5.44**
400^2	$2.43e-5$	**5.00**	$3.94e-6$	**4.87**
		$R = 3$		
50^2	$7.57e-2$	–	$2.09e-2$	–
100^2	$2.93e-3$	**4.69**	$3.82e-4$	**5.77**
200^2	$4.08e-5$	**6.17**	$1.00e-6$	**5.25**
400^2	$4.73e-7$	**6.43**	$9.13e-8$	**6.78**

2D Riemann problem configuration 3

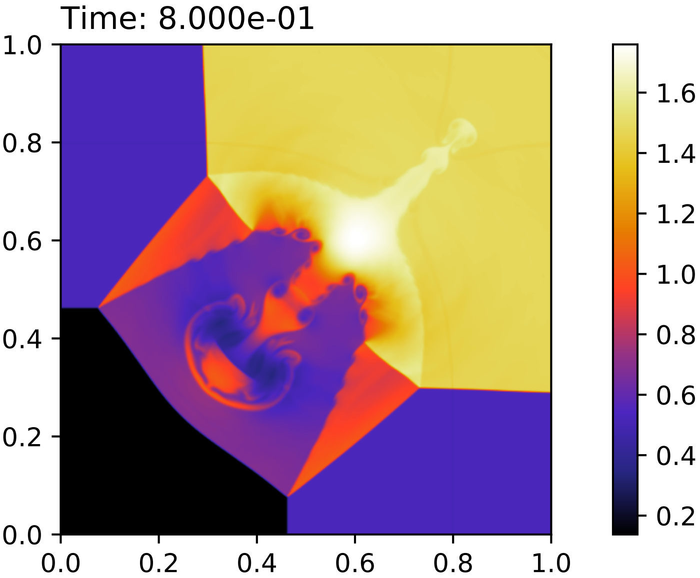
Euler equations on $[0,1]^2$ with outflow boundaries and initial condition

$$\begin{pmatrix}\rho_1\\u_1\\v_1\\p_1\end{pmatrix}=\begin{pmatrix}0.5323\\1.206\\0\\0.3\end{pmatrix} \qquad \begin{pmatrix}\rho_2\\u_2\\v_2\\p_2\end{pmatrix}=\begin{pmatrix}1.5\\0\\0\\1.5\end{pmatrix}$$

$$\begin{pmatrix}\rho_3\\u_3\\v_3\\p_3\end{pmatrix}=\begin{pmatrix}0.138\\1.206\\1.206\\0.029\end{pmatrix} \qquad \begin{pmatrix}\rho_4\\u_4\\v_4\\p_4\end{pmatrix}=\begin{pmatrix}0.5323\\0\\1.206\\0.3\end{pmatrix}$$

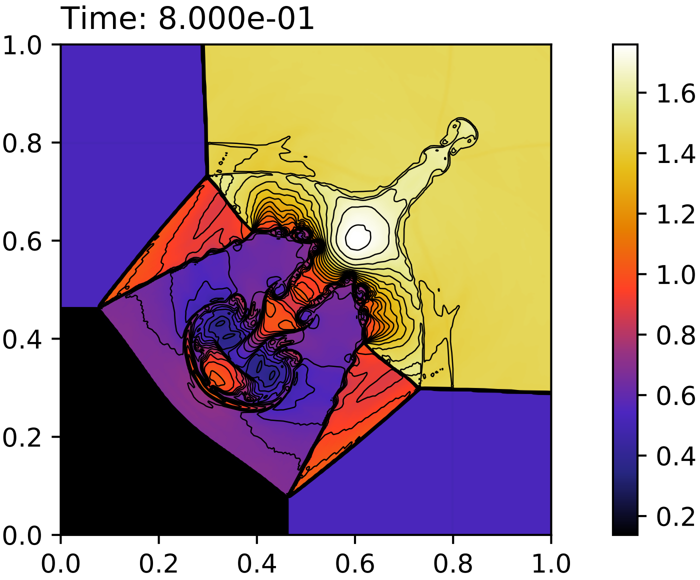
2D Riemann problem configuration 3

400×400, Radius 2, $\ell = 12\Delta$, HLLC



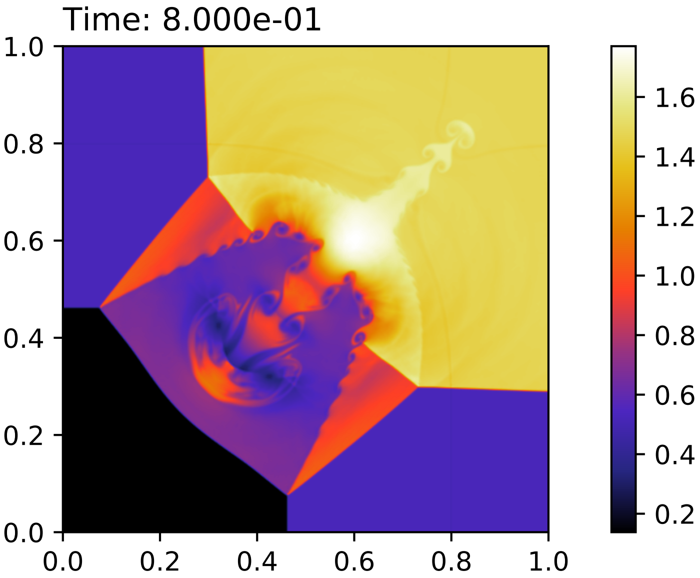
2D Riemann problem configuration 3

400×400, Radius 2, $\ell = 12\Delta$, HLLC



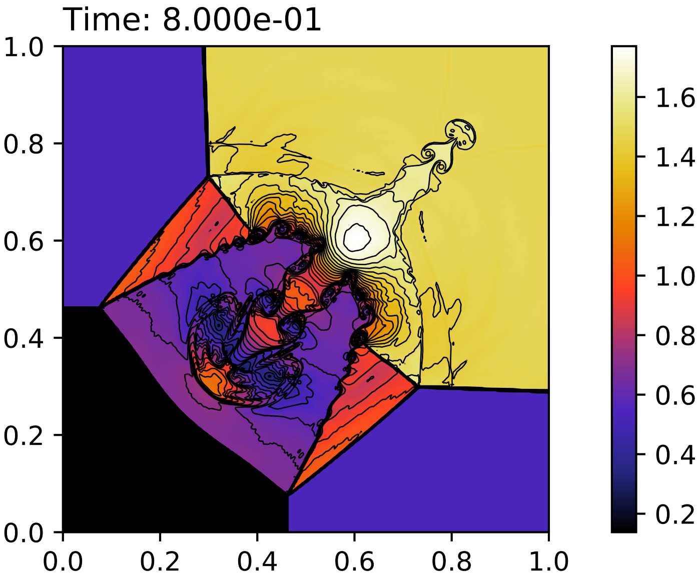
2D Riemann problem configuration 3

400×400, Radius 3, $\ell = 12\Delta$, HLLC



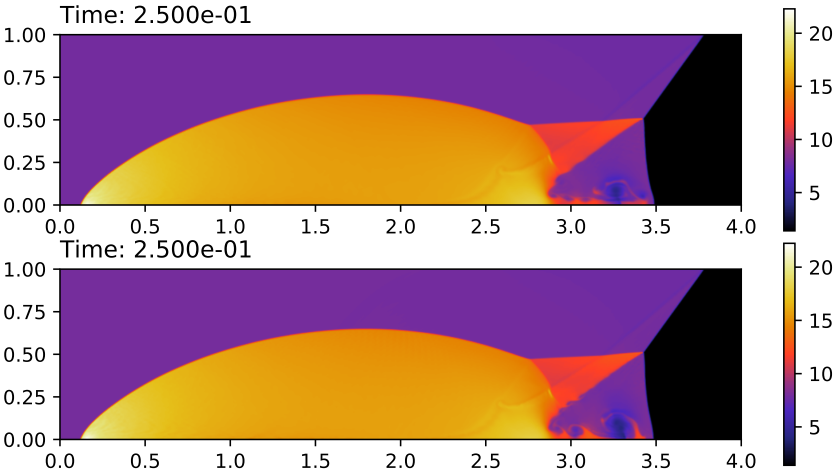
2D Riemann problem configuration 3

400×400, Radius 3, $\ell = 12\Delta$, HLLC



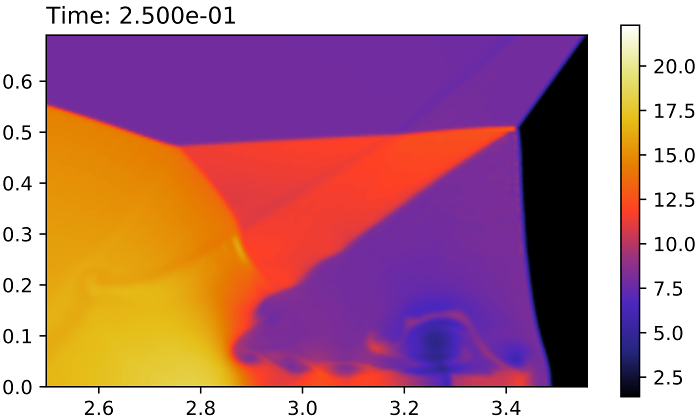
Double mach reflection problem

800×200, Radii 2 and 3, $\ell = 12\Delta$, HLLC



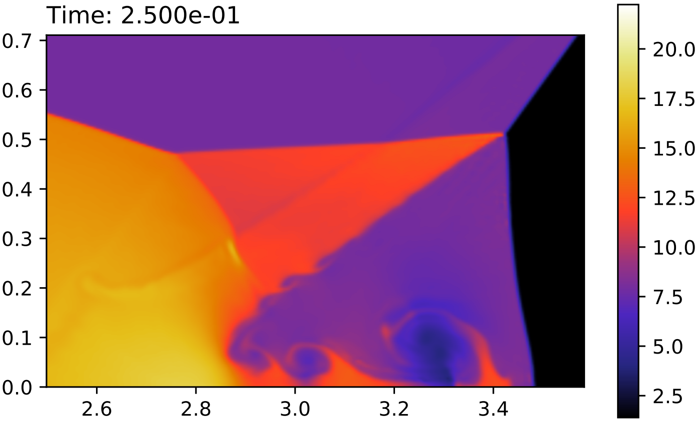
Double mach reflection problem

800×200, Radius 2, $\ell = 12\Delta$, HLLC



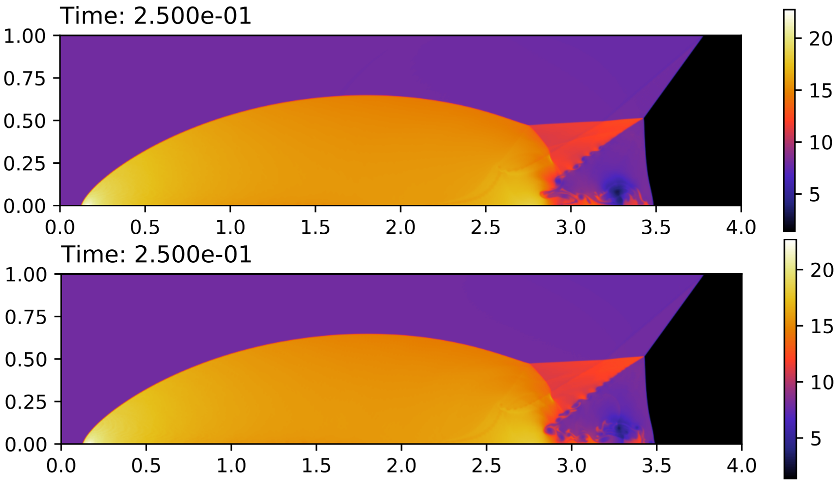
Double mach reflection problem

800×200, Radius 3, $\ell = 12\Delta$, HLLC



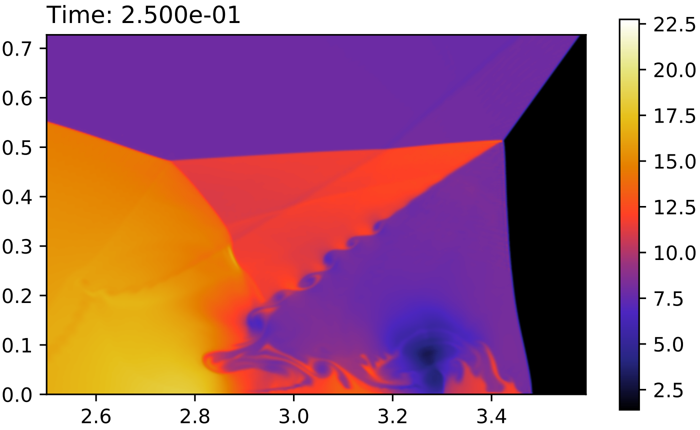
Double mach reflection problem

1600×400, Radii 2 and 3, $\ell = 12\Delta$, HLLC



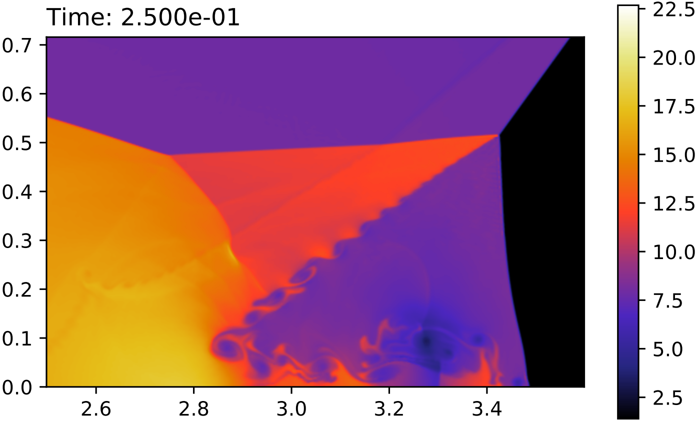
Double mach reflection problem

1600×400, Radius 2, $\ell = 12\Delta$, HLLC



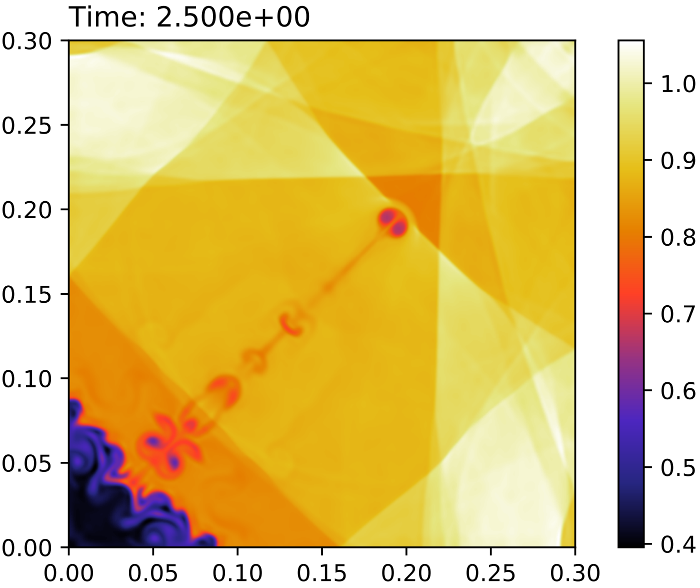
Double mach reflection problem

1600×400, Radius 3, $\ell = 12\Delta$, HLLC



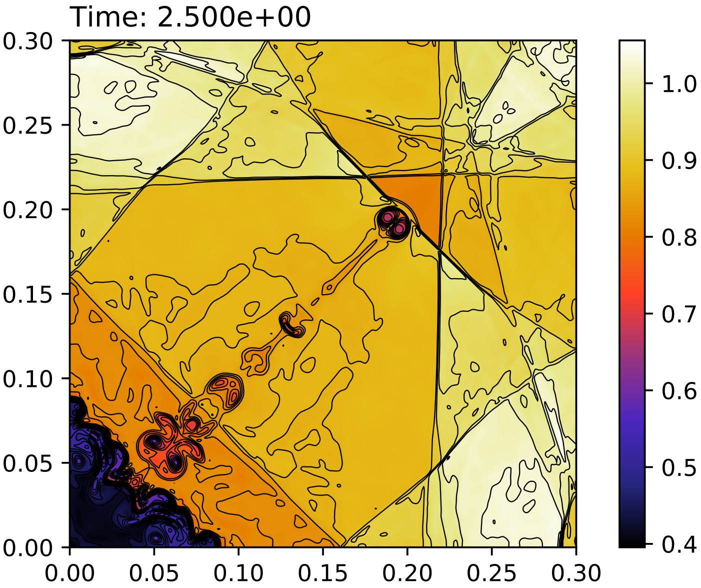
Liska-Wendroff implosion problem

400×400, Radius 2, $\ell = 12\Delta$, HLL



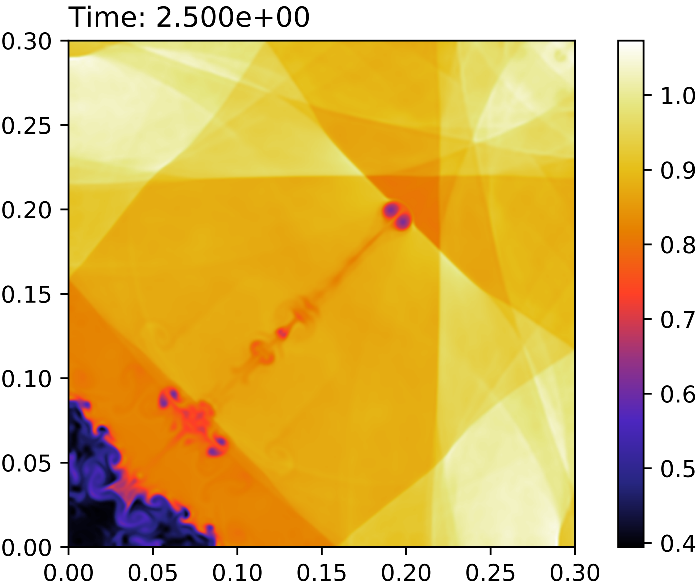
Liska-Wendroff implosion problem

400×400, Radius 2, $\ell = 12\Delta$, HLL



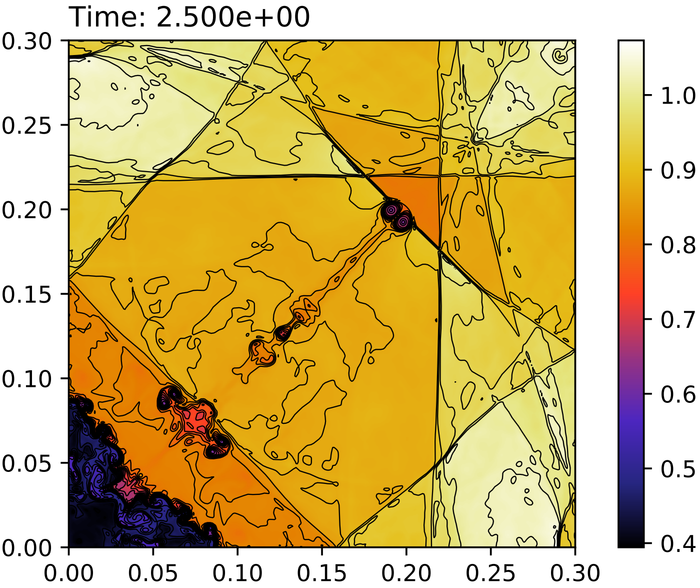
Liska-Wendroff implosion problem

400×400, Radius 3, $\ell = 12\Delta$, HLL



Liska-Wendroff implosion problem

400×400, Radius 3, $\ell = 12\Delta$, HLL



Final thoughts

Conclusion

- High-order multidimensional FVMs require careful implementation
- Kernel based reconstruction is very flexible
  - We can use the flexibility to simplify the implementation
  - The length scale is an interesting knob to have available

Next steps

- Investigate HWENO methods
- Extend to MHD
- Extend to 3D/AMR
- Incorporate viscous terms – implicit time stepping
- Time stepping without RK