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
Introduction

Goal
Solve systems of hyperbolic conservation laws

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

with an accurate and robust finite volume method

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

in multiple dimensions.
Compressible Euler equations, 2D
System of interest

For today, consider

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

\[
U = \begin{pmatrix}
\rho \\
\rho u \\
\rho v \\
\rho w \\
E
\end{pmatrix}
\]

\[
F(U) = \begin{pmatrix}
\rho u \\
\rho u^2 + p \\
\rho uv \\
\rho uw \\
u(E + p)
\end{pmatrix}
\]

\[
G(U) = \begin{pmatrix}
\rho v \\
\rho uv \\
\rho v^2 + p \\
\rho vw \\
v(E + p)
\end{pmatrix}
\]

for a calorically ideal gas,

\[
p = (\gamma - 1) \rho \epsilon, \quad \epsilon = \frac{E}{\rho} - \frac{v \cdot v}{2}.
\]
Quick overview of FVM

Abstract formulation
Partition full domain $\Omega$ into finite volumes $\Omega_i$ such that $\Omega = \bigcup_i \Omega_i$, and $\Omega_i \cap \Omega_j = \emptyset$, $i \neq j$. Denote

$$\langle \cdot \rangle_i = \frac{1}{||\Omega_i||} \int_{\Omega_i} \cdot \, dx,$$

then for (systems of) hyperbolic conservation laws

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

for numeric flux $\hat{F}$, and states $U^-$ and $U^+$ inside and outside $\Omega_i$. 
Quick overview of FVM

Uniform 2D Cartesian grids
Let \( \Omega_{i,j} = [x_i - \frac{\Delta x}{2}, x_i - \frac{\Delta x}{2}] \times [y_i - \frac{\Delta y}{2}, y_j - \frac{\Delta y}{2}] \), then

\[
\frac{\partial}{\partial t} \langle \mathbf{U} \rangle_{i,j} = -\frac{1}{||\Omega_{i,j}||} \oint_{\partial \Omega_{i,j}} \mathbf{F}(\mathbf{U}^-, \mathbf{U}^+) \cdot \mathbf{n} \, ds
= -\frac{1}{\Delta x} \left( \langle \hat{\mathbf{F}} \rangle_{i+\frac{1}{2},j} - \langle \hat{\mathbf{F}} \rangle_{i-\frac{1}{2},j} \right) - \frac{1}{\Delta y} \left( \langle \hat{\mathbf{G}} \rangle_{i,j+\frac{1}{2}} - \langle \hat{\mathbf{G}} \rangle_{i,j-\frac{1}{2}} \right)
\]

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 \textit{pointwise}, thus need accurate \textit{pointwise} values of \( \mathbf{U}_{i\pm1/2}^\pm \)
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

\[ K(x,y) = e^{-||x-y||^2}_\ell^2. \]

\(^1\) Omitting many technical details
Accurate construction of Riemann states
Multidimensional concerns

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 $\mathcal{H}$, 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:

\[
\tilde{f}(x) = \sum_{j=1}^{N} \alpha_j K(x, x_j)
\]

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

\[
[K(x_i, x_j)] \alpha = y.
\]
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:

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

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

$$[K(x_i, x_j)] \alpha = y.$$

Properties and interpretation of $\tilde{f}$

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

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

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

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

$$\begin{bmatrix} \lambda_i^{(x)} \lambda_j(y) K(x_i, x_j) \end{bmatrix} \alpha = y.$$
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:

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

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

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

Relationship to regular interpolation

- $\lambda_j^{(y)} K(x, y) \in \mathcal{H}$, hence $\tilde{f} \in \mathcal{H}$
- $(f - \tilde{f}) \perp \mathcal{H}_0$, 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

\[
\begin{bmatrix}
\frac{1}{||\Omega_i||} & \frac{1}{||\Omega_j||}
\end{bmatrix}
\begin{bmatrix}
\int \int_{\Omega_i \Omega_j} K(x, y) dx dy
\end{bmatrix}
\alpha = y,
\]

and evaluating the interpolant at \(x^*\) gives

\[
\tilde{f}(x) = \sum_{j=1}^{N} \alpha_j \int_{\Omega_j} K(x^*, y) dy = z^T y
\]

where the prediction vector is given by:

\[
z^T = \left[ \int_{\Omega_j} K(x^*, y) dy \right]^T \left[ \begin{bmatrix}
\frac{1}{||\Omega_i||} & \frac{1}{||\Omega_j||}
\end{bmatrix}
\begin{bmatrix}
\int \int_{\Omega_i \Omega_j} K(x, y) dx dy
\end{bmatrix}
\right]^{-1} y
\]
We need to compute

\[ z^T = w^T C^{-1}, \]

where \( C \) and \( w \) both depend on \( \ell \).

- Large values of \( \ell \) tend to give more accurate interpolants
- Large values of \( \ell \) give horribly conditioned linear systems
Stabilizing large $\ell$

We need to compute

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

where $C$ and $w$ both depend on $\ell$.

- Large values of $\ell$ tend to give more accurate interpolants
- Large values of $\ell$ give horribly conditioned linear systems

**Stable evaluation of prediction vectors**

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

- $z_i(\ell^{-1}) = w^T C^{(-1)} e_i$ 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?
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.
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

\[ \Omega_{i,j} \]
The isentropic vortex problem

A truly nonlinear benchmark problem

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

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

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

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

recover the initial condition at time $T_f = 2L$
The isentropic vortex problem
\[ \Omega = [-10, 10]^2, \ell = 4, \text{Linear scheme} \]

\[ \log_{10}(||\rho_N||_1) \]

\[ L_1 \text{ Density error} \]

\[ \text{Grid} \quad L_1 \text{ Error} \quad L_1 \text{ Order} \quad L_\infty \text{ Error} \quad L_\infty \text{ Order} \]

\begin{align*}
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
\end{align*}
Nonlinear GP reconstruction
The reconstruction presented is linear, i.e.

\[ U_{i+\frac{1}{2},j} = \tilde{U}(x^*) = z^T [\langle U \rangle]_{S(i,j)} \]

which is hopeless near discontinuities (Godunov)
Dealing with shocks: WENO methods

Nonlinear GP reconstruction
The reconstruction presented is linear, i.e.

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

which is hopeless near discontinuities (Godunov)

WENO (weighted essentially non-oscillatory) methods
Break full stencil into substencils, use weighted combination of individual reconstructions

$$U_{i + \frac{1}{2}, j} = \sum_{S_k \in S_{i,j}} \omega_k \tilde{U}_k(x^*)$$

where $S_{i,j}$ is set of substencils, and $\omega_k$ depends on the data in $S_k$. 
$S_1$: Central substencil
Substencils in the spirit of standard WENO
$S_2$: North substencil

Substencils in the spirit of standard WENO
$S_3$: East stencil

Substencils in the spirit of standard WENO
$S_4$: South substencil
Substencils in the spirit of standard WENO
Substencils in the spirit of standard WENO

$\Omega_{i,j}$
The optimal linear weights $\gamma_k$ minimize discrepancy in

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

*independent* of the data.
The optimal linear weights $\gamma_k$ minimize discrepancy in

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

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 $\gamma_k$ minimize discrepancy in

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

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 $\omega_k$

- 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.
Generally, no linear weights, $\gamma_k$, 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:

$$U_{i+\frac{1}{2},j} = \frac{\omega_0}{\gamma_0} \tilde{U}_0(x^*) + \sum_{k=1}^{5} \left( \omega_k - \omega_0 \frac{\gamma_k}{\gamma_0} \right) \tilde{U}_k(x^*)$$
Generally, no linear weights, $\gamma_k$, 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:

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

Now we can choose $\gamma_k$ to ensure stability

$$
\gamma_0 = C_h, \\
\gamma_1 = (1 - C_h)C_l, \\
\gamma_2 = \gamma_3 = \gamma_4 = \gamma_5 = \frac{(1 - C_h) \ast (1 - C_l)}{4},
$$

where $0 < C_h, C_l < 1$, e.g. $C_h = C_l = 0.8$. 
Smoothness indicators
The last numerical ingredient

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

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

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

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

\[ \tilde{\omega}_k = \gamma_k \left( 1 + \left( \frac{\tau}{\beta_k + \epsilon} \right)^p \right) \]

\[ \omega_k = \frac{\tilde{\omega}_k}{\sum \tilde{\omega}_k} \]
The isentropic vortex problem

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

### Grid $L_1$ Error $L_1$ Order $L_\infty$ Error $L_\infty$ Order

$R = 2$

<table>
<thead>
<tr>
<th>Grid</th>
<th>$L_1$ Error</th>
<th>$L_1$ Order</th>
<th>$L_\infty$ Error</th>
<th>$L_\infty$ Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>$50^2$</td>
<td>$1.46e-1$</td>
<td>$-$</td>
<td>$2.41e-2$</td>
<td>$-$</td>
</tr>
<tr>
<td>$100^2$</td>
<td>$1.73e-2$</td>
<td>$3.06$</td>
<td>$5.00e-3$</td>
<td>$2.27$</td>
</tr>
<tr>
<td>$200^2$</td>
<td>$7.78e-4$</td>
<td>$4.47$</td>
<td>$1.15e-4$</td>
<td>$5.44$</td>
</tr>
<tr>
<td>$400^2$</td>
<td>$2.43e-5$</td>
<td>$5.00$</td>
<td>$3.94e-6$</td>
<td>$4.87$</td>
</tr>
</tbody>
</table>

$R = 3$

<table>
<thead>
<tr>
<th>Grid</th>
<th>$L_1$ Error</th>
<th>$L_1$ Order</th>
<th>$L_\infty$ Error</th>
<th>$L_\infty$ Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>$50^2$</td>
<td>$7.57e-2$</td>
<td>$-$</td>
<td>$2.09e-2$</td>
<td>$-$</td>
</tr>
<tr>
<td>$100^2$</td>
<td>$2.93e-3$</td>
<td>$4.69$</td>
<td>$3.82e-4$</td>
<td>$5.77$</td>
</tr>
<tr>
<td>$200^2$</td>
<td>$4.08e-5$</td>
<td>$6.17$</td>
<td>$1.00e-6$</td>
<td>$5.25$</td>
</tr>
<tr>
<td>$400^2$</td>
<td>$4.73e-7$</td>
<td>$6.43$</td>
<td>$9.13e-8$</td>
<td>$6.78$</td>
</tr>
</tbody>
</table>
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},
\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},
\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, ℓ = 12Δ, HLLC

Time: 8.000e-01
2D Riemann problem configuration 3

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

Time: $8.000e-01$
Double mach reflection problem
800 × 200, Radii 2 and 3, ℓ = 12Δ, HLLC

Time: 2.500e-01
Double mach reflection problem

800 × 200, Radius 2, ℓ = 12Δ, HLLC

Time: 2.500e-01
Double mach reflection problem

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

Ian May
KMLS-FVM
UC Santa Cruz
Double mach reflection problem
1600 × 400, Radius 2, ℓ = 12Δ, HLLC

Time: 2.500e-01
Double mach reflection problem

1600 × 400, Radius 3, \( \ell = 12\Delta \), HLLC
Liska-Wendroff implosion problem
400 × 400, Radius 2, ℓ = 12Δ, HLL

Time: 2.500e+00
Liska-Wendroff implosion problem

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

Time: $2.500e+00$
Liska-Wendroff implosion problem
$400 \times 400$, Radius 3, $\ell = 12\Delta$, HLL

Time: 2.500e+00
Liska-Wendroff implosion problem

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

Time: $2.500\times10^0$
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