

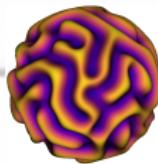
Gaussian processes for high order finite volume methods

February 20, 2020

Ian May, Dongwook Lee

Department of Applied Mathematics
University of California Santa Cruz

Santa Cruz, CA





Goal

Solve the compressible Euler equations (2D)

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

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

accurately and robustly



Numerical flux

$$\begin{aligned}\langle \mathbf{F} \rangle_{i \pm 1/2, j} &= \frac{1}{\Delta y} \int_{y_{j-1/2}}^{y_{j+1/2}} \mathbf{F}(\mathbf{U}(x_{i \pm 1/2}, y)) dy \\ &\approx \frac{1}{\Delta y} \int_{y_{j-1/2}}^{y_{j+1/2}} \hat{\mathbf{F}} \left(\mathbf{U}_{i \pm 1/2}^-(y), \mathbf{U}_{i \pm 1/2}^+(y) \right) dy\end{aligned}$$

Two barriers to high order in multiple dimensions

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

Naive dimension-by-dimension approach



Polynomial reconstruction

Given the stencil $\{\langle \mathbf{U} \rangle_{i-r,j}, \dots, \langle \mathbf{U} \rangle_{i,j}, \dots, \langle \mathbf{U} \rangle_{i+r,j}\}$, there is a unique polynomial $\mathbf{Q}(x)$ of degree $p = 2r$ satisfying:

$$\frac{1}{\Delta x} \int_{x_{i+s-1/2}}^{x_{i+s+1/2}} \mathbf{Q}(x) dx = \langle \mathbf{U} \rangle_{i+s,j}, \quad s = -r, \dots, r$$

which yields the approximation

$$\langle \mathbf{U} \rangle_{i \pm 1/2, j} = \mathbf{Q}(x_{i \pm 1/2}) + \mathcal{O}(\Delta x^{p+1})$$

Just plug it in, what could go wrong?

First note

$$\mathbf{U}(x_{i \pm 1/2}, y_j) = \langle \mathbf{U} \rangle_{i \pm 1/2, j} + \mathcal{O}(\Delta x^{p+1}) + \mathcal{O}(\Delta y^2),$$

and

Special cases



Linear flux functions

If $\mathbf{F}(\mathbf{U}) = \mathbf{AU}$ then,

$$\begin{aligned}\langle \mathbf{F} \rangle_{i \pm 1/2, j} &= \frac{1}{\Delta y} \int_{y_{j-1/2}}^{y_{j+1/2}} \mathbf{AU}(x_{i \pm 1/2}, y) dy \\ &= \mathbf{A} \langle \mathbf{U} \rangle_{i \pm 1/2, j} \\ &= \mathbf{AQ}(x_{i \pm 1/2}) + \mathcal{O}(\Delta x^{p+1}).\end{aligned}$$

Must use nonlinear benchmark problems



Expensive, but intuitive method

- Use multidimensional reconstruction to get point values on faces directly
- Approximate flux integral with a Gauss rule
- ⇒ Need multiple point values on each face, multiple calls to Riemann solver



Expensive, but intuitive method

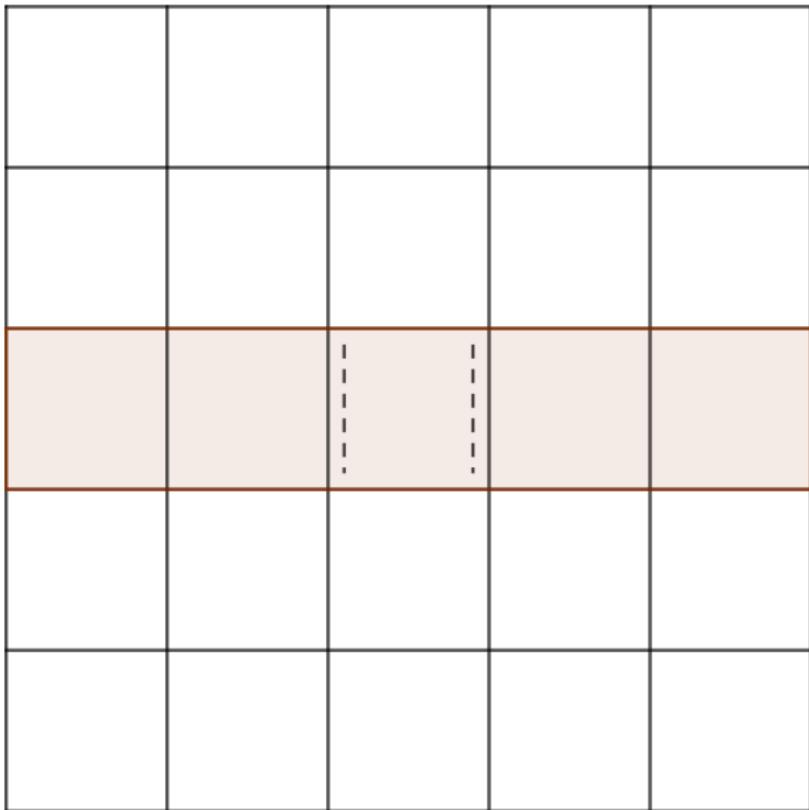
- Use multidimensional reconstruction to get point values on faces directly
- Approximate flux integral with a Gauss rule
- ⇒ Need multiple point values on each face, multiple calls to Riemann solver

Modified dimension-by-dimension

- Use 1D stencils to get accurate face-averaged states
- Reconstruct along faces to get accurate face-centered states
- Call Riemann solver once per interface
- Reconstruct face-average fluxes from face-centered fluxes

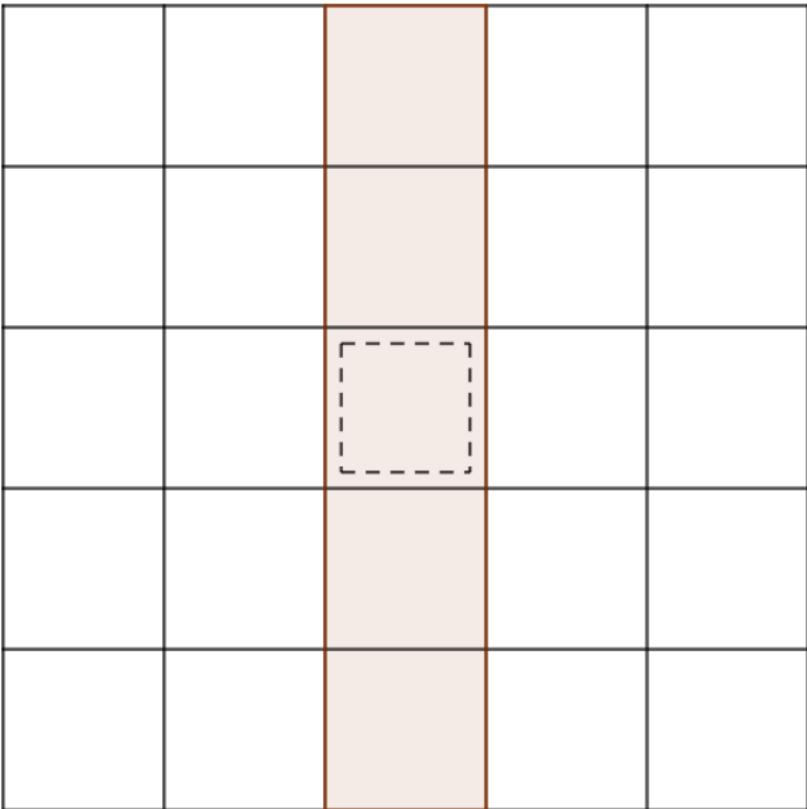
Modified dimension-by-dimension

Diagrammatically



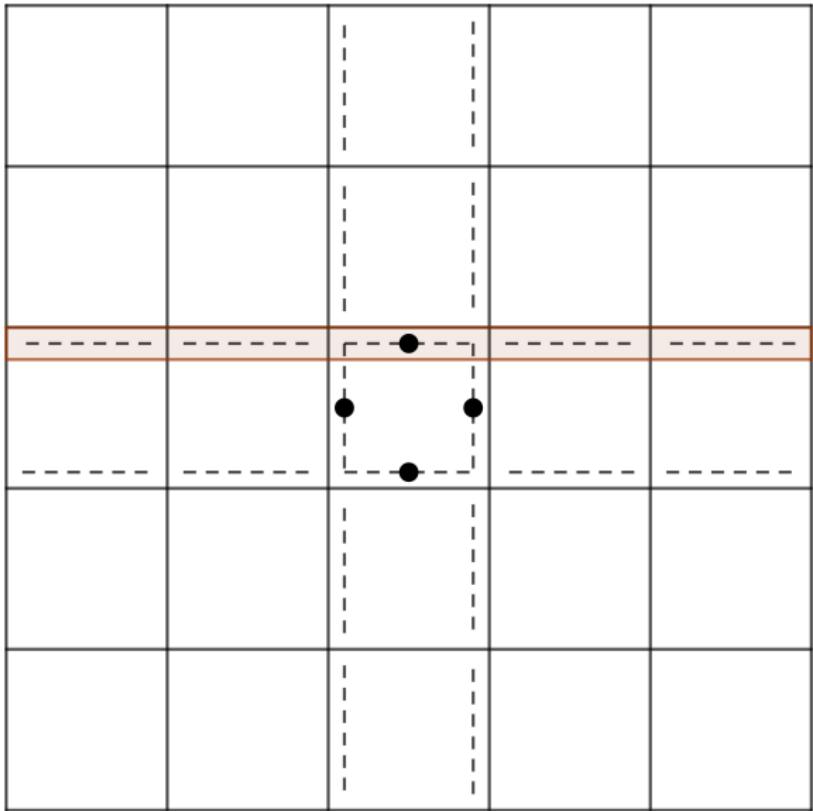
Modified dimension-by-dimension

Diagrammatically



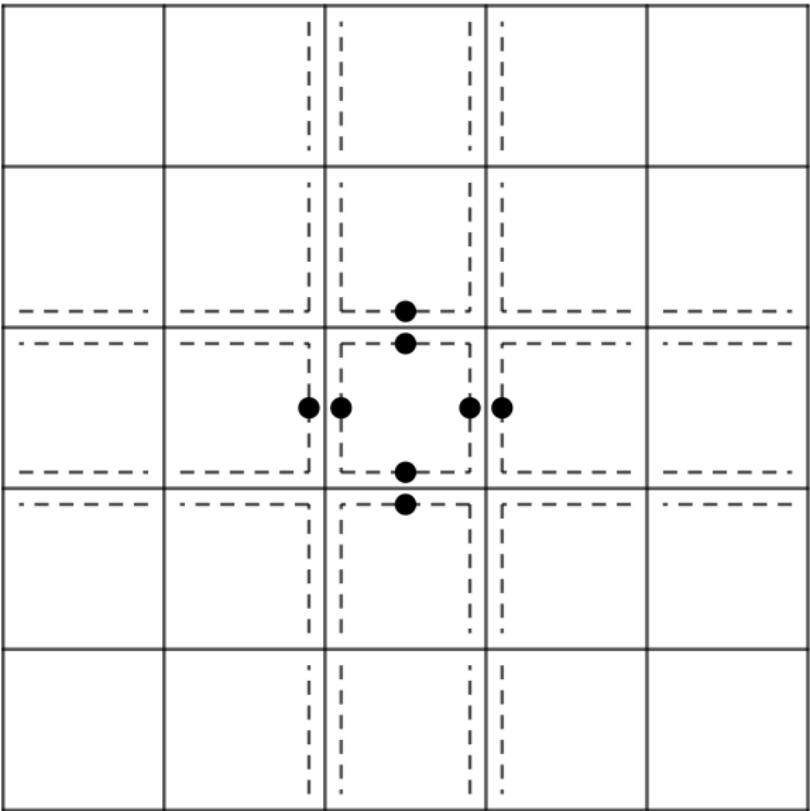
Modified dimension-by-dimension

Diagrammatically



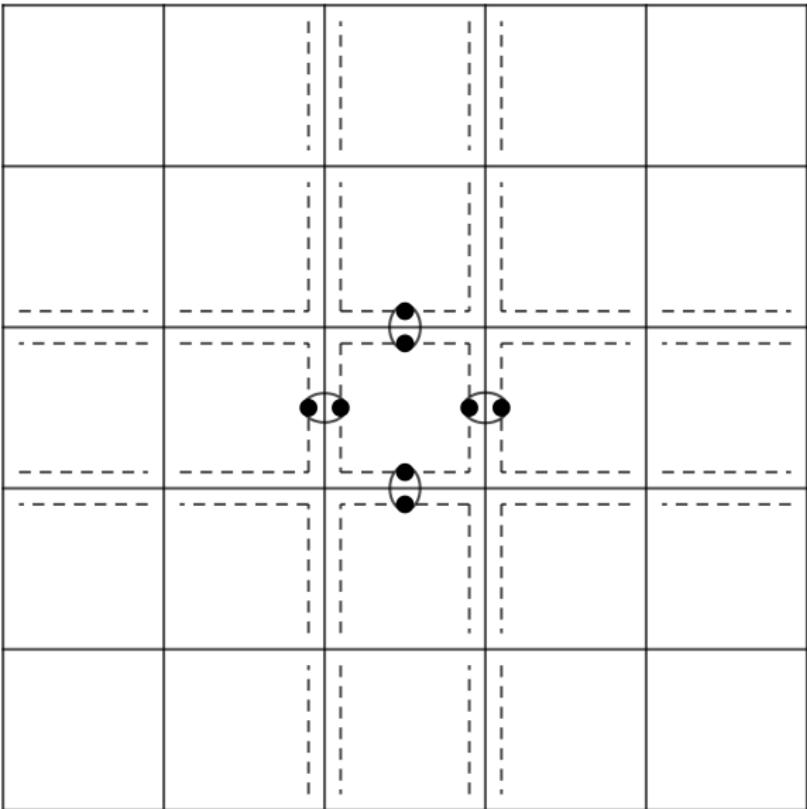
Modified dimension-by-dimension

Diagrammatically



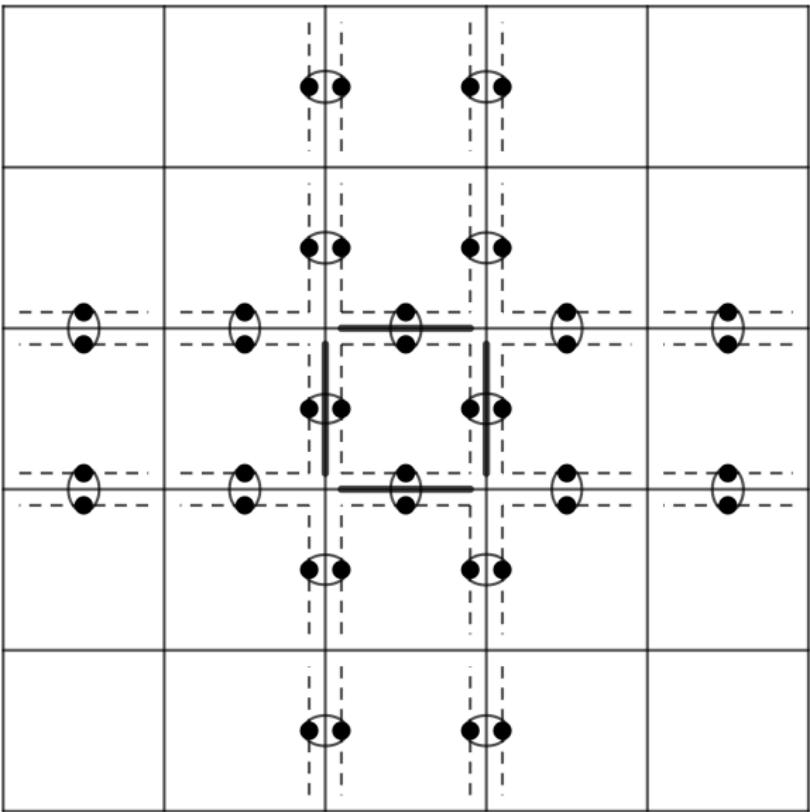
Modified dimension-by-dimension

Diagrammatically



Modified dimension-by-dimension

Diagrammatically



Why Gaussian processes?



They generalize very well

- Dimension agnostic
- Order agnostic
- (Un)Structured grid agnostic
- Flexible stencil choices
- Directly incorporate problem physics

Why Gaussian processes?



They generalize very well

- Dimension agnostic
- Order agnostic
- (Un)Structured grid agnostic
- Flexible stencil choices
- Directly incorporate problem physics

Downsides of Gaussian processes

- Ill-conditioning problems
- Less intuitive
- Almost too flexible, lot's of choices to investigate



Definition

For a domain D , a Gaussian process is given by a distribution over a function space:

$$f(\mathbf{x}) \sim \mathcal{N}(\mathbf{0}, K(\mathbf{x}, \mathbf{x}'; \ell)),$$

such that for $\mathbf{y} \in D$, $f(\mathbf{y})$ belongs to a multivariate normal distribution:

$$f(\mathbf{y}) \sim \mathcal{N}(\mathbf{0}, \mathbf{K})$$

$$\mathbf{K}_{ij} = K(y_i, y_j; \ell),$$

for some covariance (kernel) function K . Defined here as:

$$K(\mathbf{x}, \mathbf{y}; \ell) = e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\ell^2}}$$

Gaussian process interpolation



Posterior distribution

The Gaussian process conditioned on given data, $f(\mathbf{y}_k) = \mathbf{q}$ at locations $\mathbf{y}_k \in D$, goes as:

$$(f(\mathbf{x}) | f(\mathbf{y}) = \mathbf{q}) \sim \mathcal{N}(\mu_{\mathbf{y}}, K_{\mathbf{y}})$$

$$\mu_{\mathbf{y}} = K(\mathbf{x}, \mathbf{y}; \ell) \mathbf{K}^{-1} \mathbf{q}$$

$$K_y = K(\mathbf{x}, \mathbf{x}'; \ell) - K(\mathbf{x}, \mathbf{y}; \ell) \mathbf{K}^{-1} K(\mathbf{y}, \mathbf{x}'; \ell).$$

All functions described by this process interpolate the data.

Gaussian process interpolation



Posterior distribution

The Gaussian process conditioned on given data, $f(\mathbf{y}_k) = \mathbf{q}$ at locations $\mathbf{y}_k \in D$, goes as:

$$(f(\mathbf{x}) | f(\mathbf{y}) = \mathbf{q}) \sim \mathcal{N}(\mu_{\mathbf{y}}, K_{\mathbf{y}})$$

$$\mu_{\mathbf{y}} = K(\mathbf{x}, \mathbf{y}; \ell) \mathbf{K}^{-1} \mathbf{q}$$

$$K_y = K(\mathbf{x}, \mathbf{x}'; \ell) - K(\mathbf{x}, \mathbf{y}; \ell) \mathbf{K}^{-1} K(\mathbf{y}, \mathbf{x}'; \ell).$$

All functions described by this process interpolate the data.

Mean is the most likely interpolant

To predict $f(\mathbf{x}^*)$ for some $\mathbf{x}^* \in D$ evaluate the mean: $f(\mathbf{x}^*) \approx \mu_{\mathbf{y}}(\mathbf{x}^*)$. Compactly:

$$\begin{aligned} f(\mathbf{x}^*) &\approx K(\mathbf{x}^*, \mathbf{y}; \ell) \mathbf{K}^{-1} \mathbf{q} \\ &\approx \mathbf{w}_*^T \mathbf{q} \end{aligned}$$

Gaussian process reconstruction – 1D

Average values → point values



Dealing with cell/face averaged values

We want to use GP to convert between data types. Define correlation matrix to match input data,

$$\mathbf{C}_{ij} = \frac{1}{\|D_i\| \|D_j\|} \int_{D_i} \int_{D_j} K(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y}$$

and sample respecting the correlation between data types,

$$\mathbf{T}_i = \frac{1}{\|D_i\|} \int_{D_i} K(\mathbf{x}, \mathbf{x}^*) d\mathbf{x}$$

to find

$$f(\mathbf{x}^*) \approx \mathbf{T}^T \mathbf{C}^{-1} \mathbf{g}$$

for known cell/face averages, \mathbf{g}

Gaussian process reconstruction – 1D

Point values → average values



Converting point values back to average values

Very similar to interpolation, but with appropriate sample vector.

Defining:

$$\mathbf{K}_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$$
$$\mathbf{T}_i = \frac{1}{||D_*||} \int_{D_*} K(\mathbf{x}, \mathbf{x}_i) d\mathbf{x}$$

we find

$$\langle f(\mathbf{x}) \rangle_* \approx \mathbf{T}^T \mathbf{K}^{-1} \mathbf{q}$$

from known point values \mathbf{q}

The isentropic vortex problem



A truly nonlinear benchmark problem

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

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

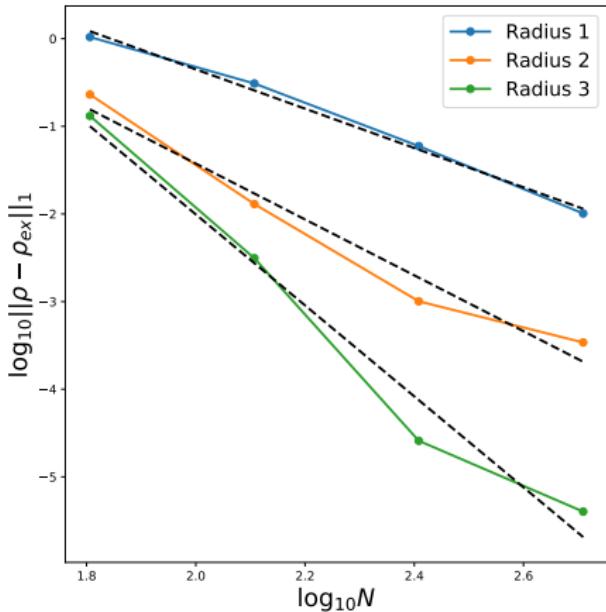
$$T = 1 - \frac{\gamma - 1}{8\gamma\pi^2} e^{1-x^2-y^2}$$

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

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

The isentropic vortex problem

$$\ell = 12\Delta, \sigma = 12\Delta$$



Radius	No WENO		WENO	
	L_1	L_∞	L_1	L_∞
1	2.06	1.89	2.24	2.03
2	3.84	3.69	3.19	4.41
3	5.65	5.50	5.19	5.08

The isentropic vortex problem

$$\ell = 12\Delta, \sigma = 12\Delta$$



Experimental convergence rates

Radius	No WENO		WENO	
	L_1	L_∞	L_1	L_∞
1	2.06	1.89	2.24	2.03
2	3.84	3.69	3.19	4.41
3	5.65	5.50	5.19	5.08



Nonlinear GP reconstruction

The reconstruction presented is linear, i.e.

$$\langle \mathbf{U} \rangle_{i \pm 1/2, j} = \sum_{s=-r}^r w_k^\pm \langle \mathbf{U} \rangle_{i+s, j}$$

which is hopeless near discontinuities (Godunov)

WENO (weighted essentially non-oscillatory) methods

Idea: Break full stencil into substencils, reconstruct on each separately, use a weighted combination of these reconstructions

$$\langle \mathbf{U} \rangle_{i \pm 1/2, j; k} = \sum_{s=k-r-1}^{k-1} \langle \mathbf{U} \rangle_{i+s, j}$$

$$\langle \mathbf{U} \rangle_{i \pm 1/2, j; k} = \sum \omega_k^\pm \langle \mathbf{U} \rangle_{i \pm 1/2, j; k}$$

Optimal weights



16

28

For *smooth* data, ω_k^\pm should reduce to some optimal weights such that

$$\sum_{s=-r}^r w_s^\pm \langle \mathbf{U} \rangle_{i+s,j} \approx \sum \gamma_k^\pm \langle \mathbf{U} \rangle_{i\pm 1/2,j;k}$$

which can be found by solving the least squares problem (e.g. $r = 2$)

$$\begin{pmatrix} w_{1,1} & 0 & 0 \\ w_{2,1} & w_{1,2} & 0 \\ w_{3,1} & w_{2,2} & w_{1,3} \\ 0 & w_{3,2} & w_{2,3} \\ 0 & 0 & w_{3,3} \end{pmatrix} \begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{pmatrix} = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \end{pmatrix}$$

Smoothness indicators



Following Jiang and Shu, we can define

$$\omega_k = \frac{\tilde{\omega}_k}{\sum \tilde{\omega}_s} \quad \tilde{\omega}_k = \frac{\gamma_k}{(\epsilon + \beta_k)^p}$$

where β_k measures the smoothness of the data on the k^{th} sub-stencil

Likelihood measures smoothness

A GP with a SE kernel is good at representing smooth functions, thus the log-likelihood

$$\log L_k = -\frac{1}{2} (\log |\mathbf{K}_k| + \mathbf{q}^T \mathbf{K}_k^{-1} \mathbf{q} + 2 \log(2\pi))$$

indicates how smooth the k^{th} sub-stencil is. Choosing

$$\beta_k = \mathbf{q}^T \mathbf{K}_k^{-1} \mathbf{q}$$

works well for equispaced grids



Normal prediction

- Use characteristic variables in direction of stencil
- Evaluate CV transformation only from center cell

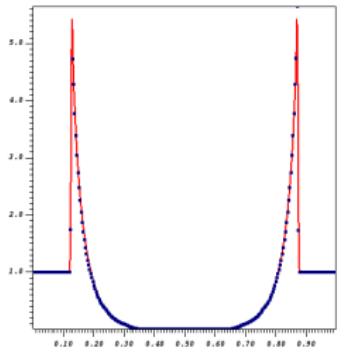
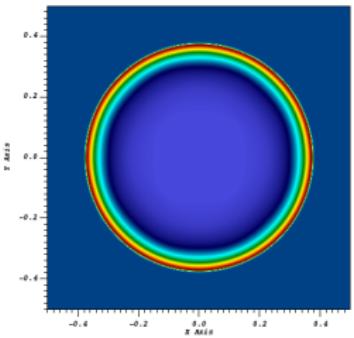
Tangential state correction

- Use characteristic variables in direction of stencil
- Evaluate CV transformation only from center face

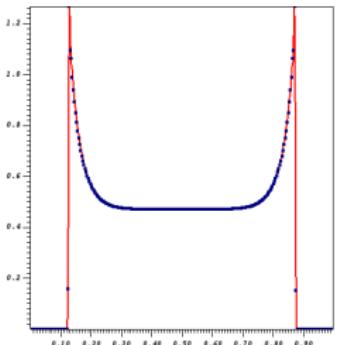
Tangential flux correction

- No variable transformation, yet
- Tends to work even without WENO

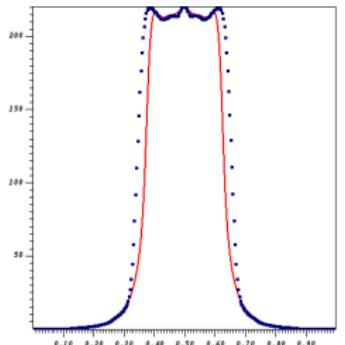
Sedov blast problem



Density



Pressure



Internal Energy

2D Riemann problem configuration 3



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

$$\begin{pmatrix} \rho_1 \\ u_1 \\ v_1 \\ p_1 \end{pmatrix} = \begin{pmatrix} 0.5323 \\ 1.206 \\ 0 \\ 0.3 \end{pmatrix}$$

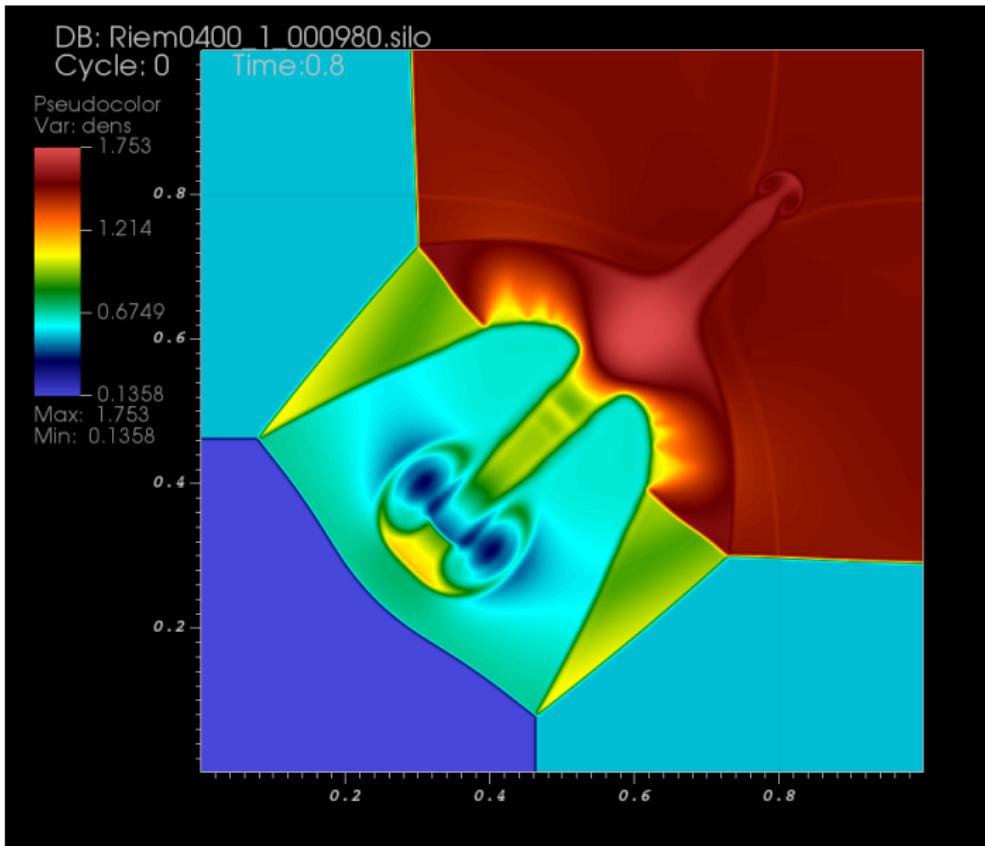
$$\begin{pmatrix} \rho_2 \\ u_2 \\ v_2 \\ p_2 \end{pmatrix} = \begin{pmatrix} 1.5 \\ 0 \\ 0 \\ 1.5 \end{pmatrix}$$

$$\begin{pmatrix} \rho_3 \\ u_3 \\ v_3 \\ p_3 \end{pmatrix} = \begin{pmatrix} 0.138 \\ 1.206 \\ 1.206 \\ 0.029 \end{pmatrix}$$

$$\begin{pmatrix} \rho_4 \\ u_4 \\ v_4 \\ p_4 \end{pmatrix} = \begin{pmatrix} 0.5323 \\ 0 \\ 1.206 \\ 0.3 \end{pmatrix}$$

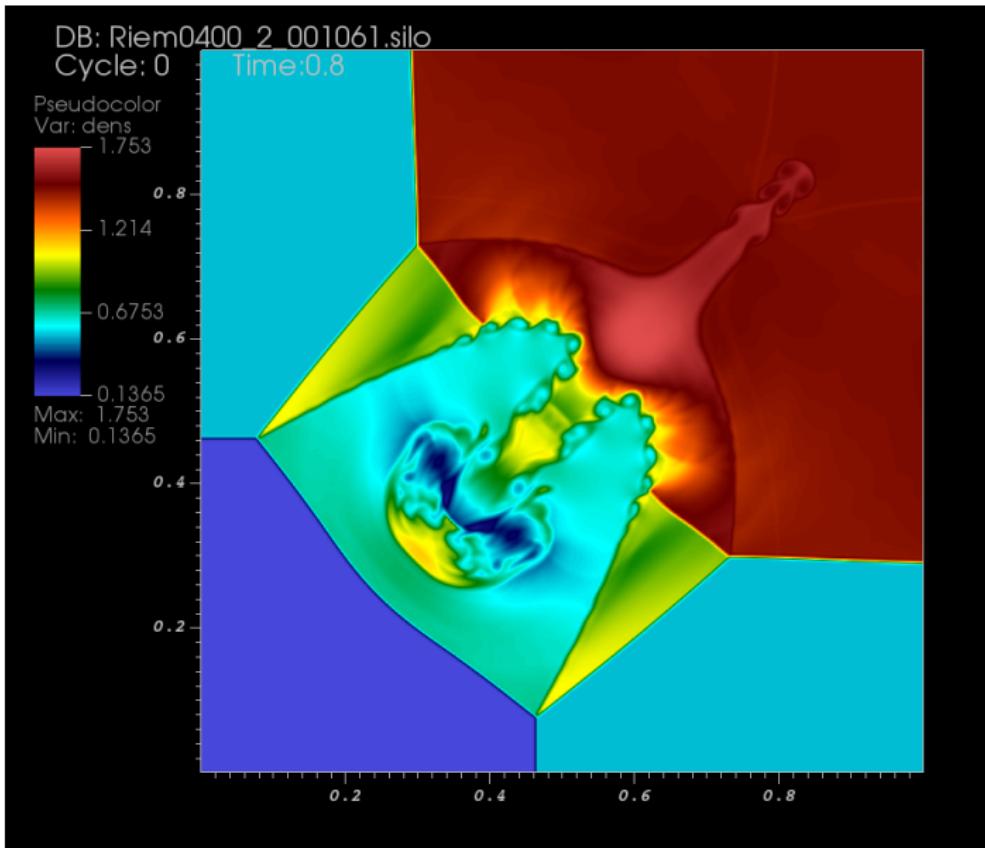
2D Riemann problem configuration 3

400×400 , Radius 1, $\ell = 12\Delta$, $\sigma = 3\Delta$



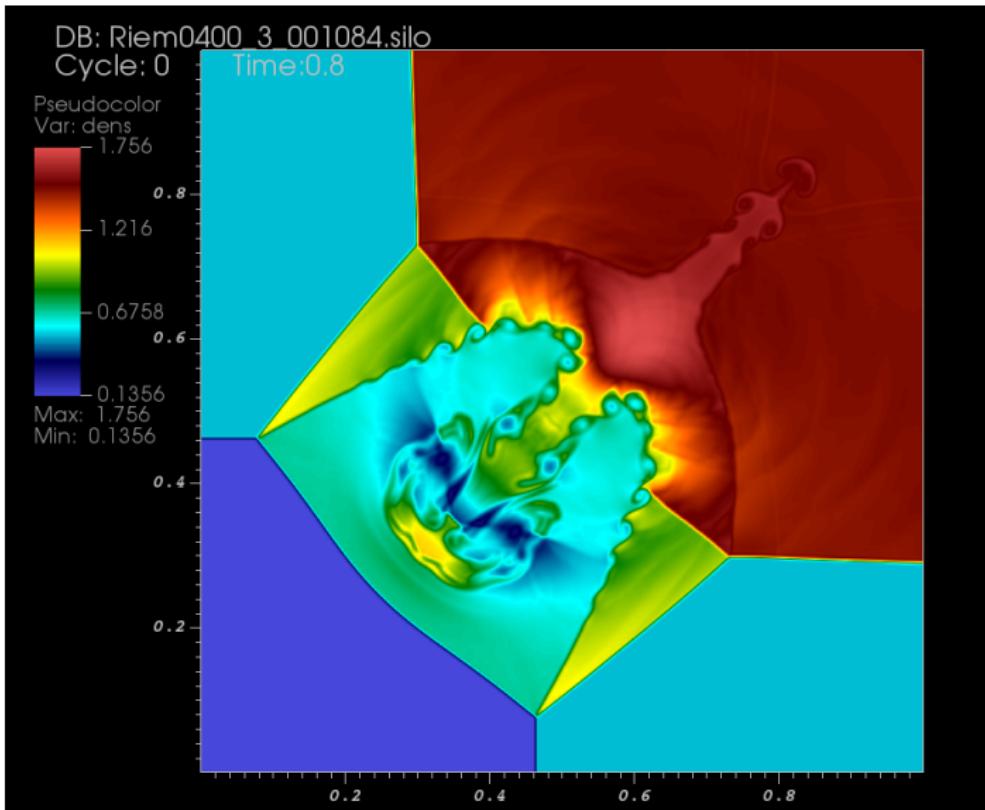
2D Riemann problem configuration 3

400×400 , Radius 2, $\ell = 12\Delta$, $\sigma = 3\Delta$



2D Riemann problem configuration 3

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



Double mach reflection

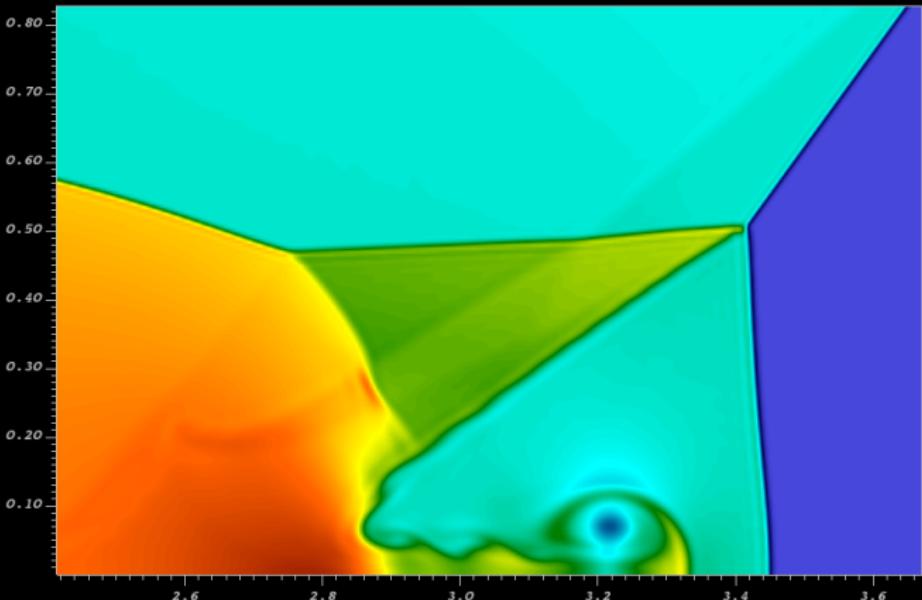
800 × 200, Radius 2, $\ell = 12\Delta$, $\sigma = 3\Delta$



DB: DoubleMach0800_2_001334.silo
Cycle: 0 Time: 0.25

Pseudocolor
Var: dens

22.68
17.36
12.04
6.719
1.400
Max: 22.68
Min: 1.400



Double mach reflection

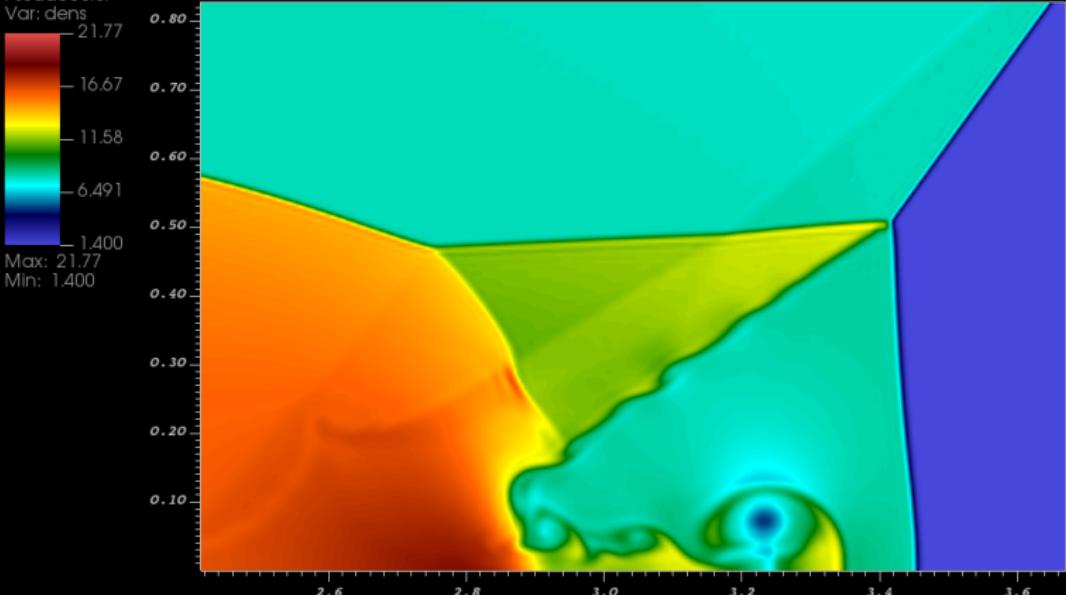
800×200 , Radius 3, $\ell = 12\Delta$, $\sigma = 3\Delta$



DB: DoubleMach0800_3_001350.silo
Cycle: 0 Time: 0.25

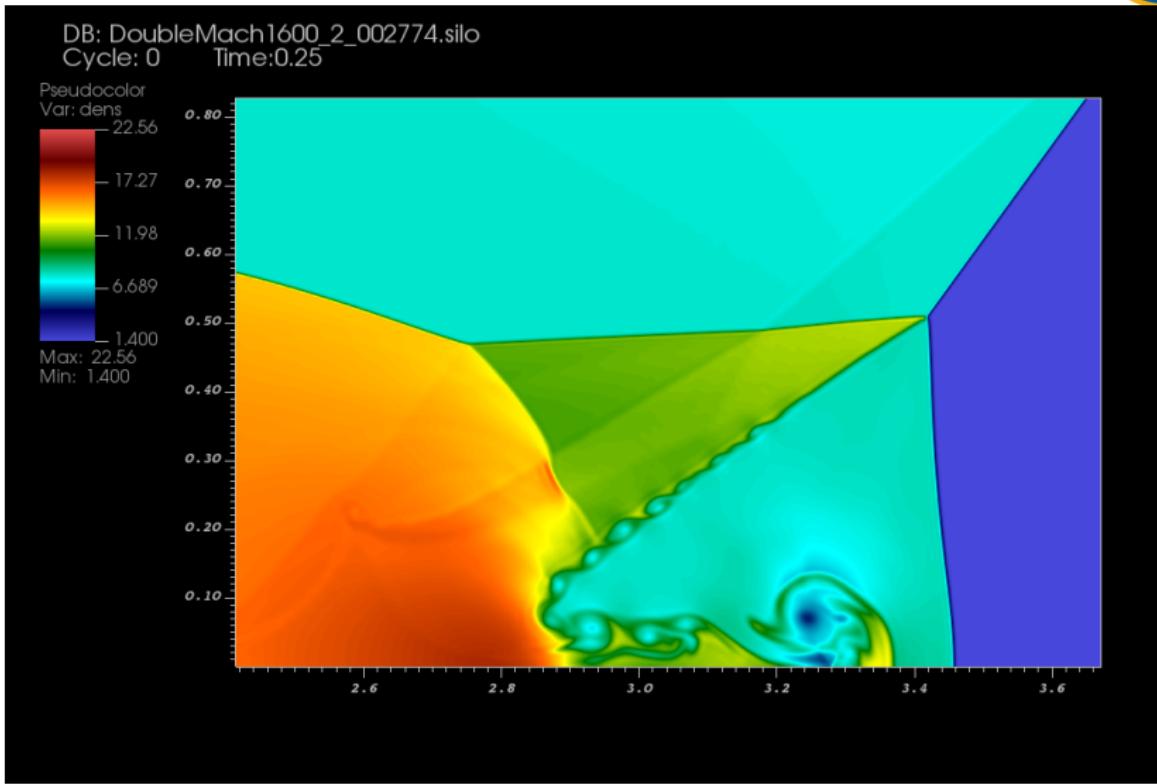
Pseudocolor
Var: dens

21.77
16.67
11.58
6.491
1.400
Max: 21.77
Min: 1.400



Double mach reflection

1600 × 400, Radius 2, $\ell = 12\Delta$, $\sigma = 3\Delta$

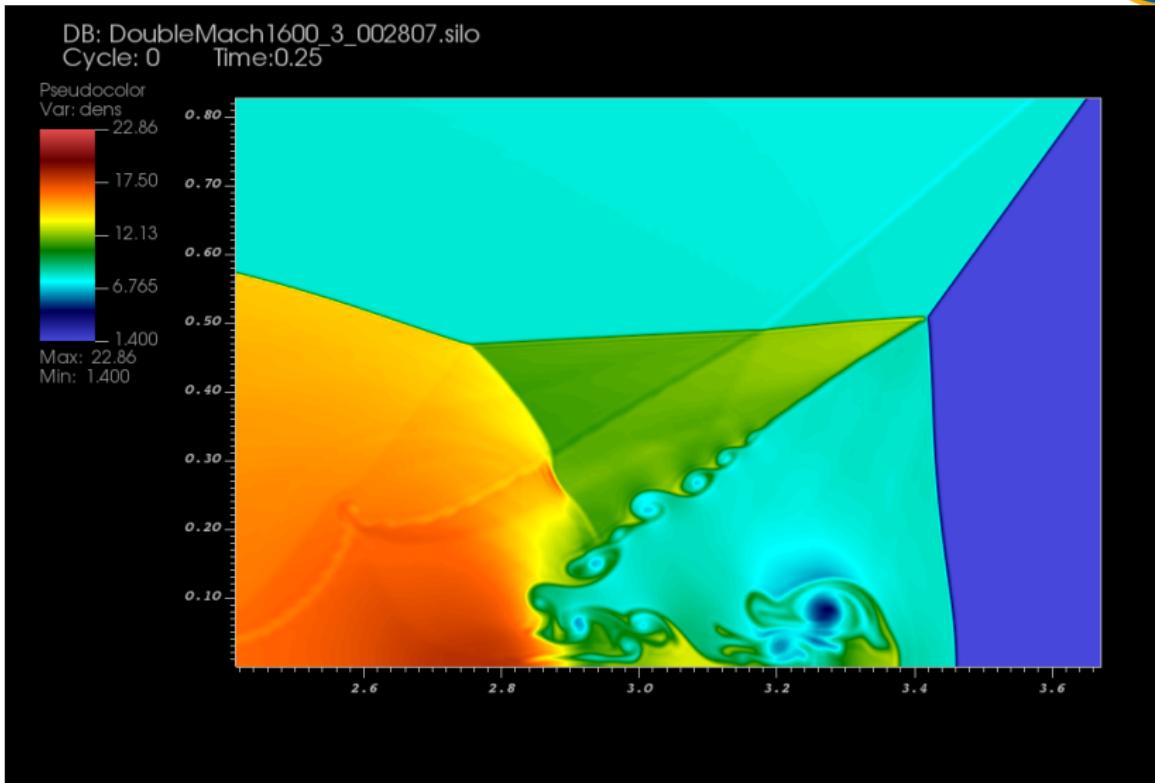


Double mach reflection

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



27





Conclusion

- Naive use of 1D stencils in 2D yields 2nd order accuracy
- Fairly cheap modification to the reconstruction recovers accuracy
 - GP version of Buchmuller and Helzel result
 - Allows re-use of established 1D GP-WENO method
- Gaussian process reconstruction is super flexible
 - Spatial order of accuracy is a runtime parameter

Next steps

- Time stepping without RK
- Ideal MHD, Divergence free GP – Preliminary results later today
- Extension to 3D and AMR

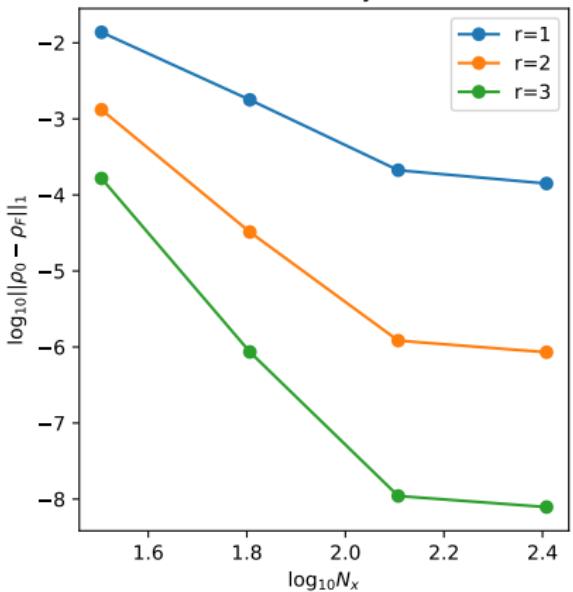
Linear benchmark

Convergence study



28

Naive D-by-D



Corrected D-by-D

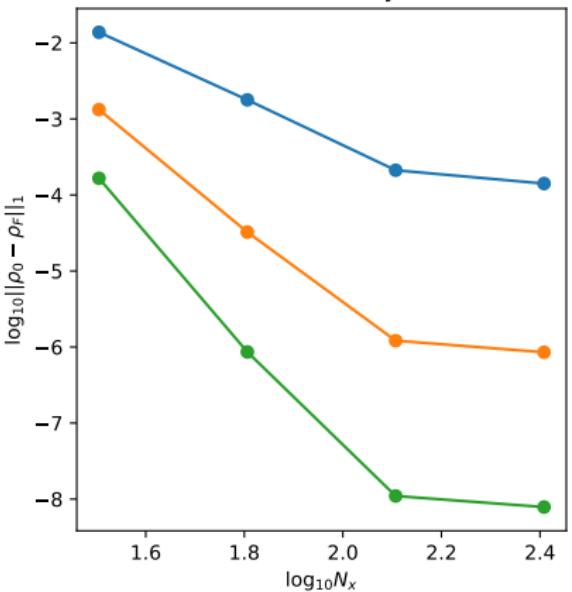


Figure: Modifications to retain non-linear accuracy don't matter for this benchmark.



Experimental convergence rates

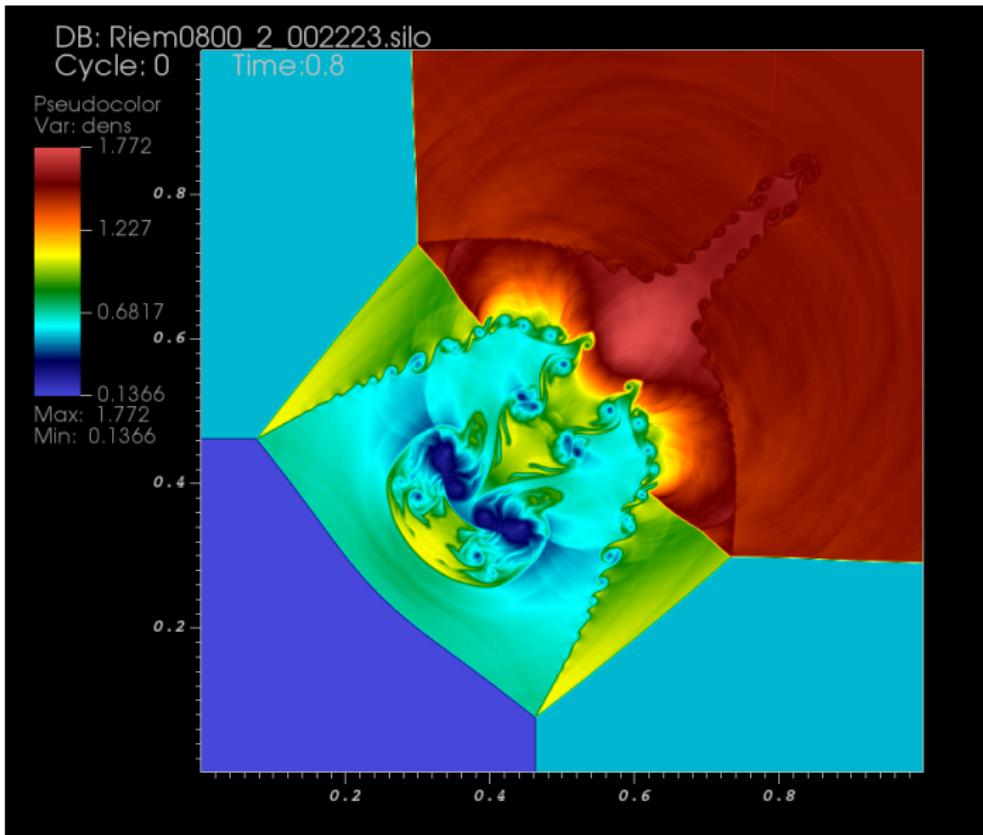
	L_1	L_∞	L_1	L_∞
$r = 1$	2.29	2.32	2.29	2.32
$r = 2$	3.65	3.75	3.65	3.75
$r = 3$	4.94	5.08	4.94	5.08

2D Riemann problem configuration 3

800 × 800, Radius 2



28

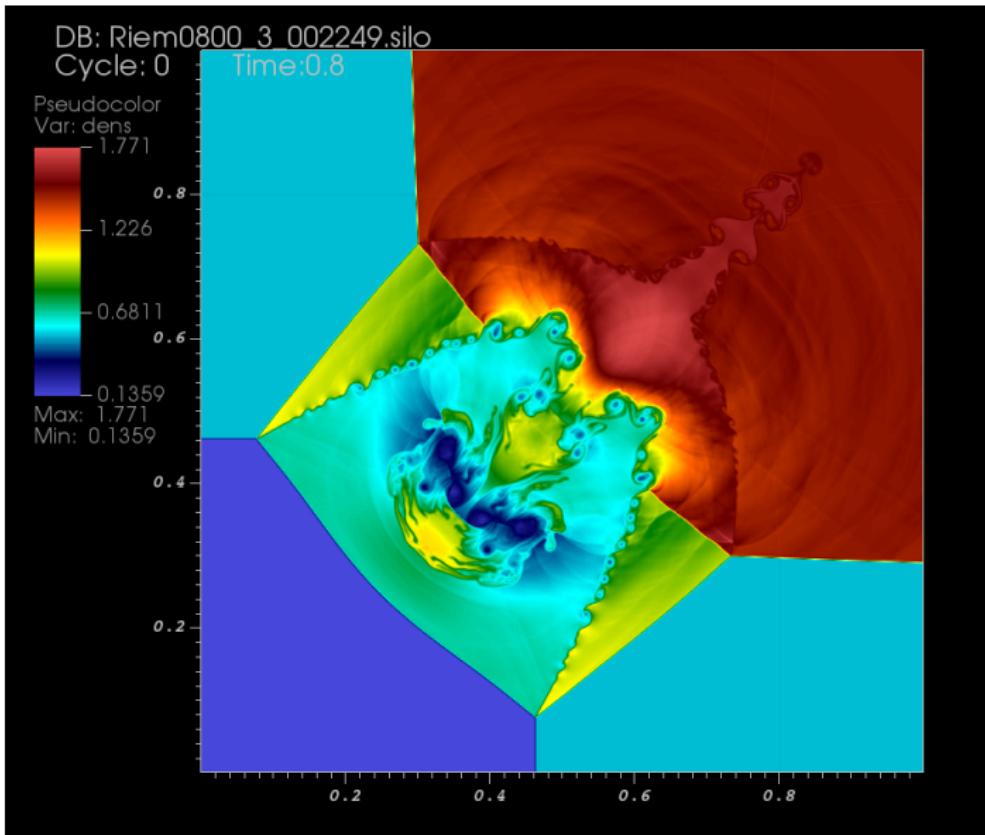


2D Riemann problem configuration 3

800 × 800, Radius 3

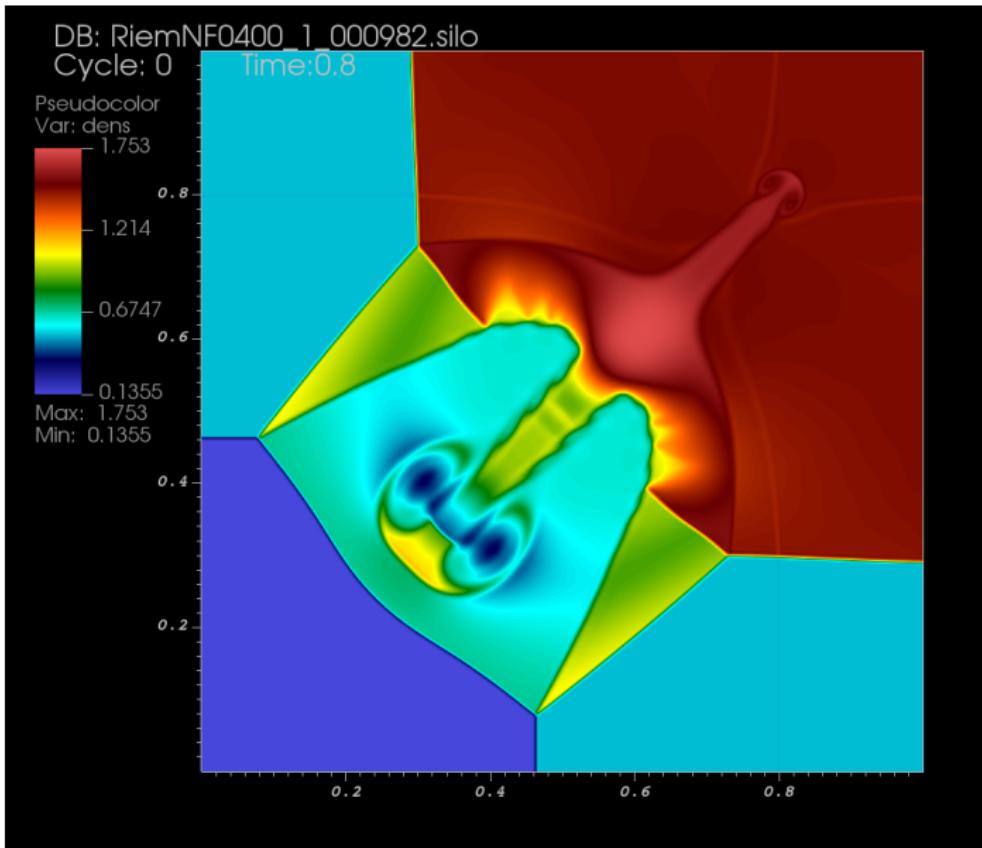


28



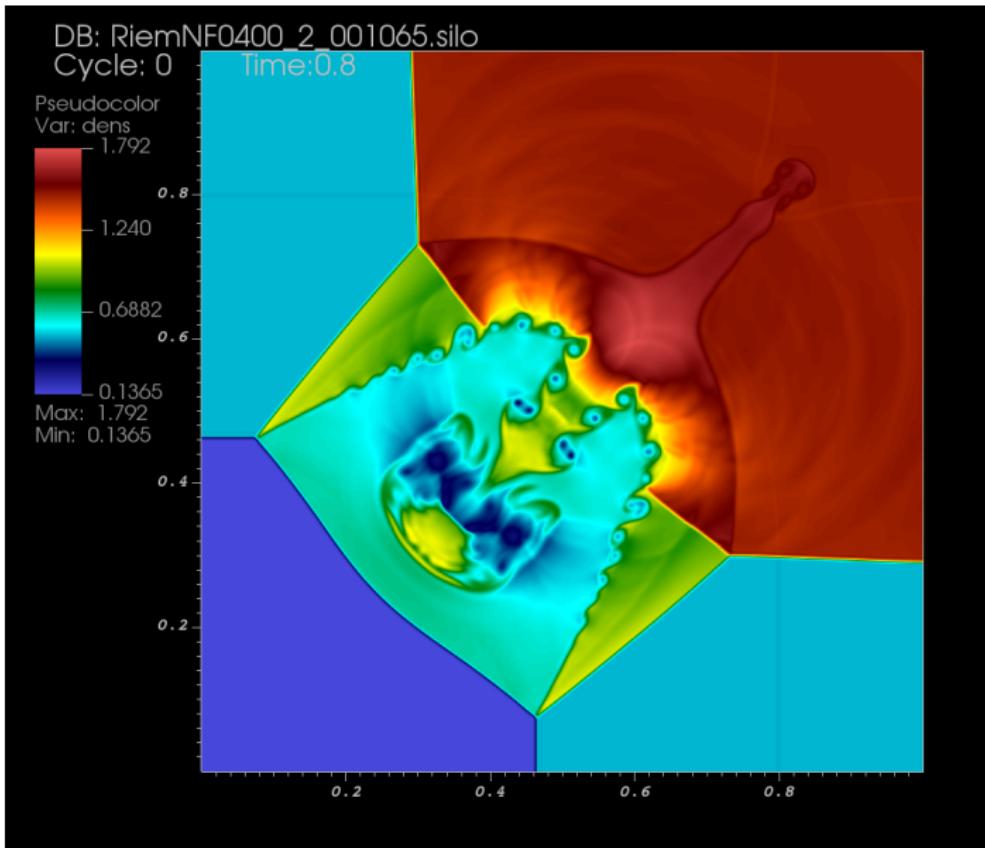
2D Riemann problem, naive

400×400 , Radius 1, $\ell = 12\Delta$, $\sigma = 3\Delta$



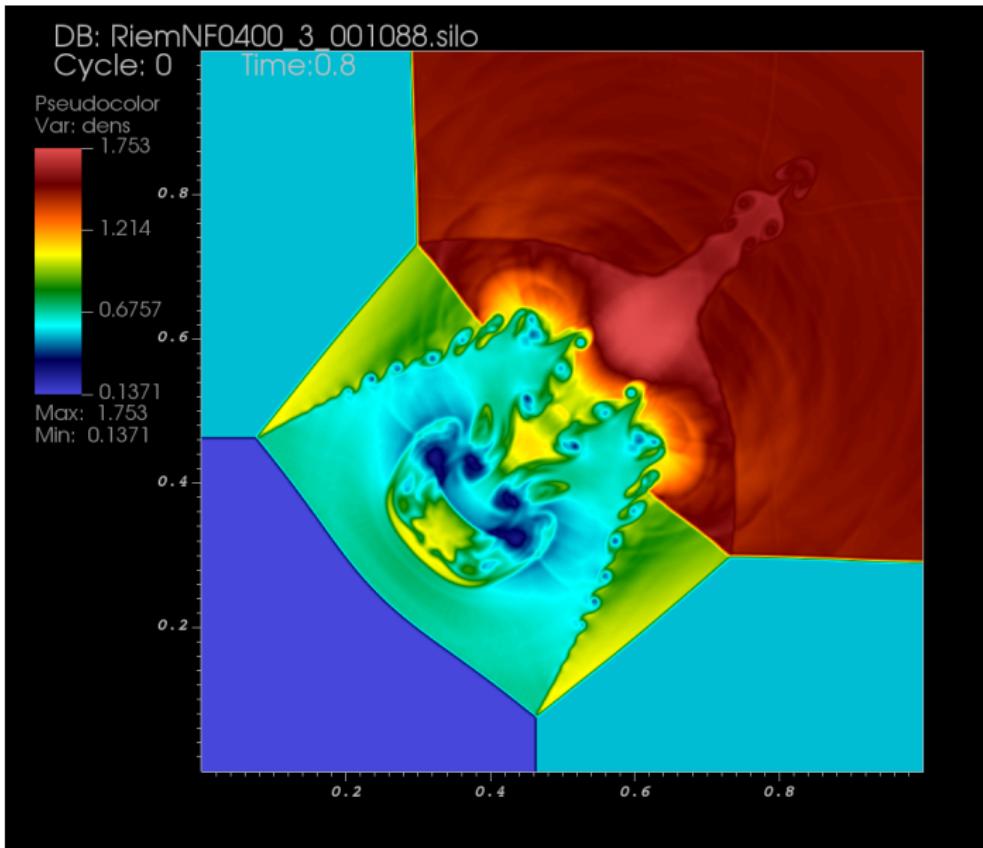
2D Riemann problem, naive

400×400 , Radius 2, $\ell = 12\Delta$, $\sigma = 3\Delta$



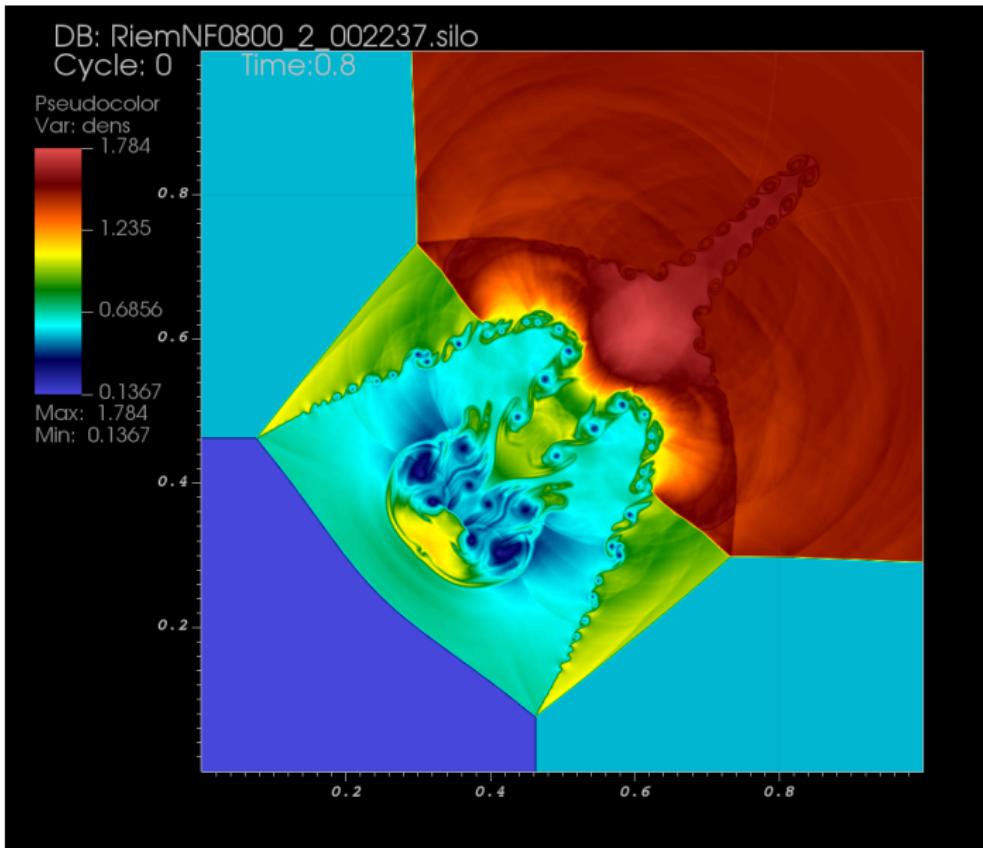
2D Riemann problem, naive

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



2D Riemann problem, naive

800×800 , Radius 2, $\ell = 12\Delta$, $\sigma = 3\Delta$



2D Riemann problem, naive

800×800 , Radius 3, $\ell = 12\Delta$, $\sigma = 3\Delta$

