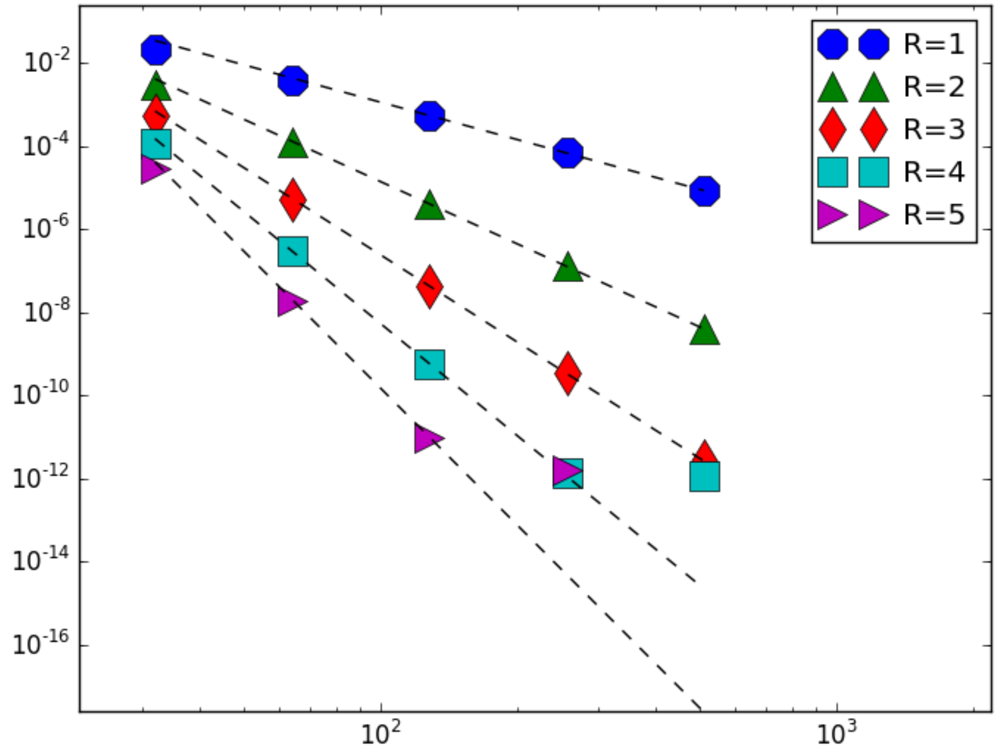
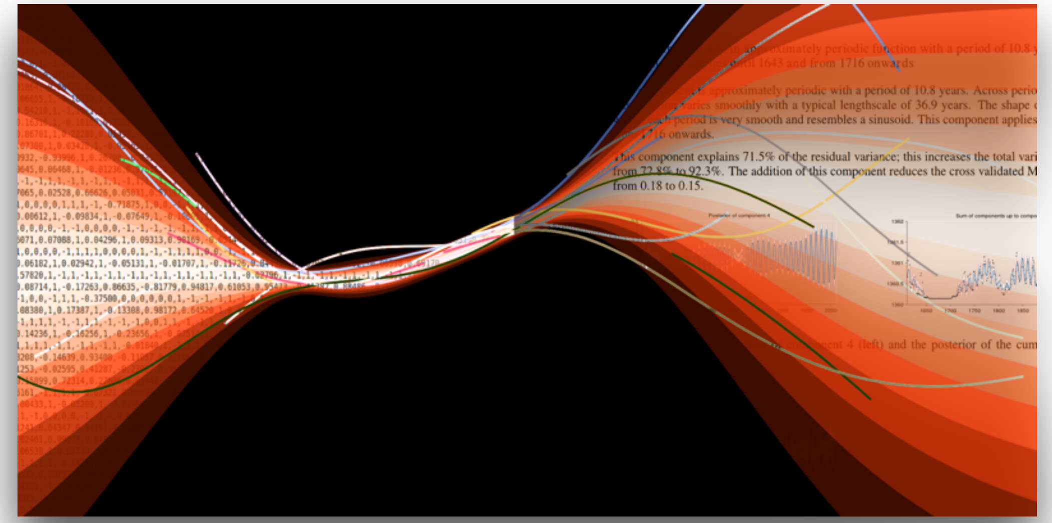
New High-Order Methods using Gaussian Processes

Dongwook Lee
Applied Mathematics & Statistics
University of California, Santa Cruz





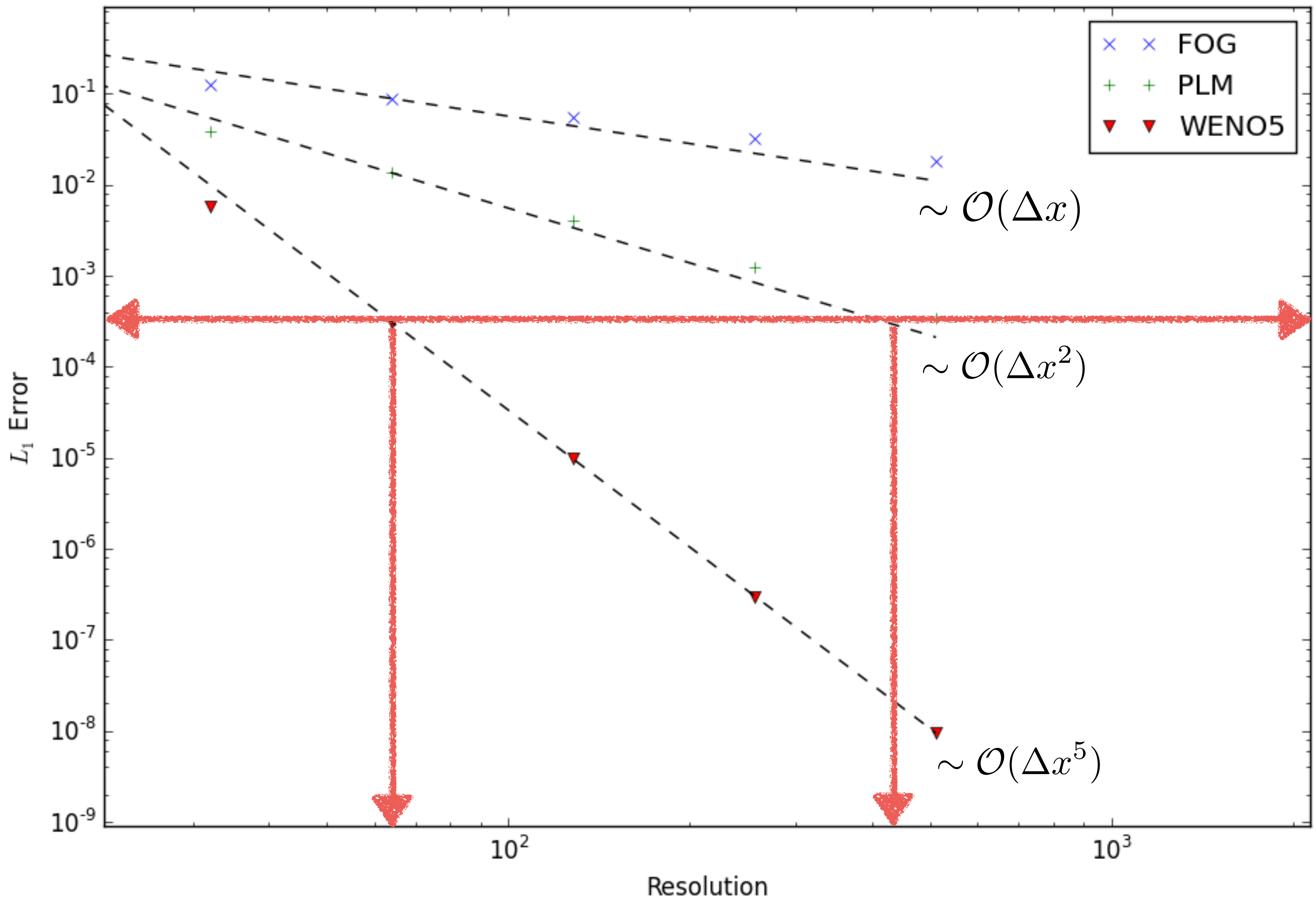
ICM, Theoretical Physics Institute, UMN
8/22-8/24, 2016

People

UC Santa Cruz
D. Lee
A. Reyes

U of Chicago
C. Graziani
P. Tzeferacos

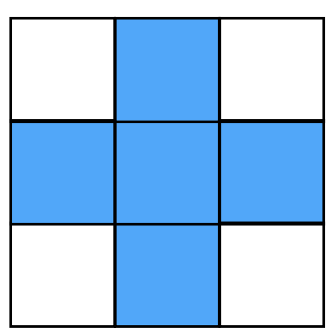
High-Order Numerical Algorithms

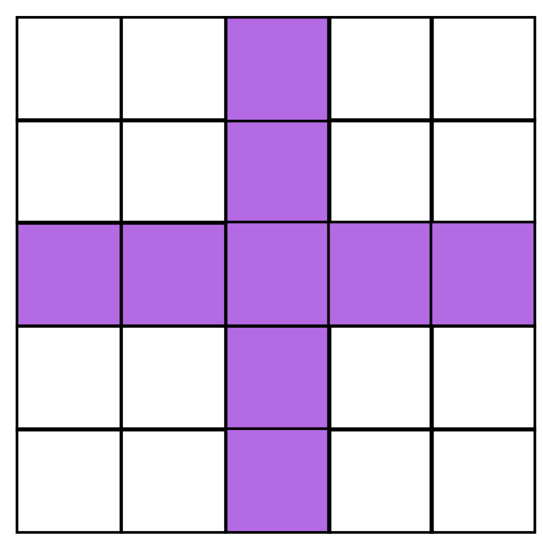


Traditional High-Order Schemes

- Traditional approaches to get Nth high-order schemes take (N-1)th degree polynomial for interpolation/reconstruction
  - only for normal direction (e.g., PLM, PPM, ENO, WENO, etc)
  - with some monotonicity controls (e.g., slope limiters, smoothness indicators, artificial viscosity, etc.)

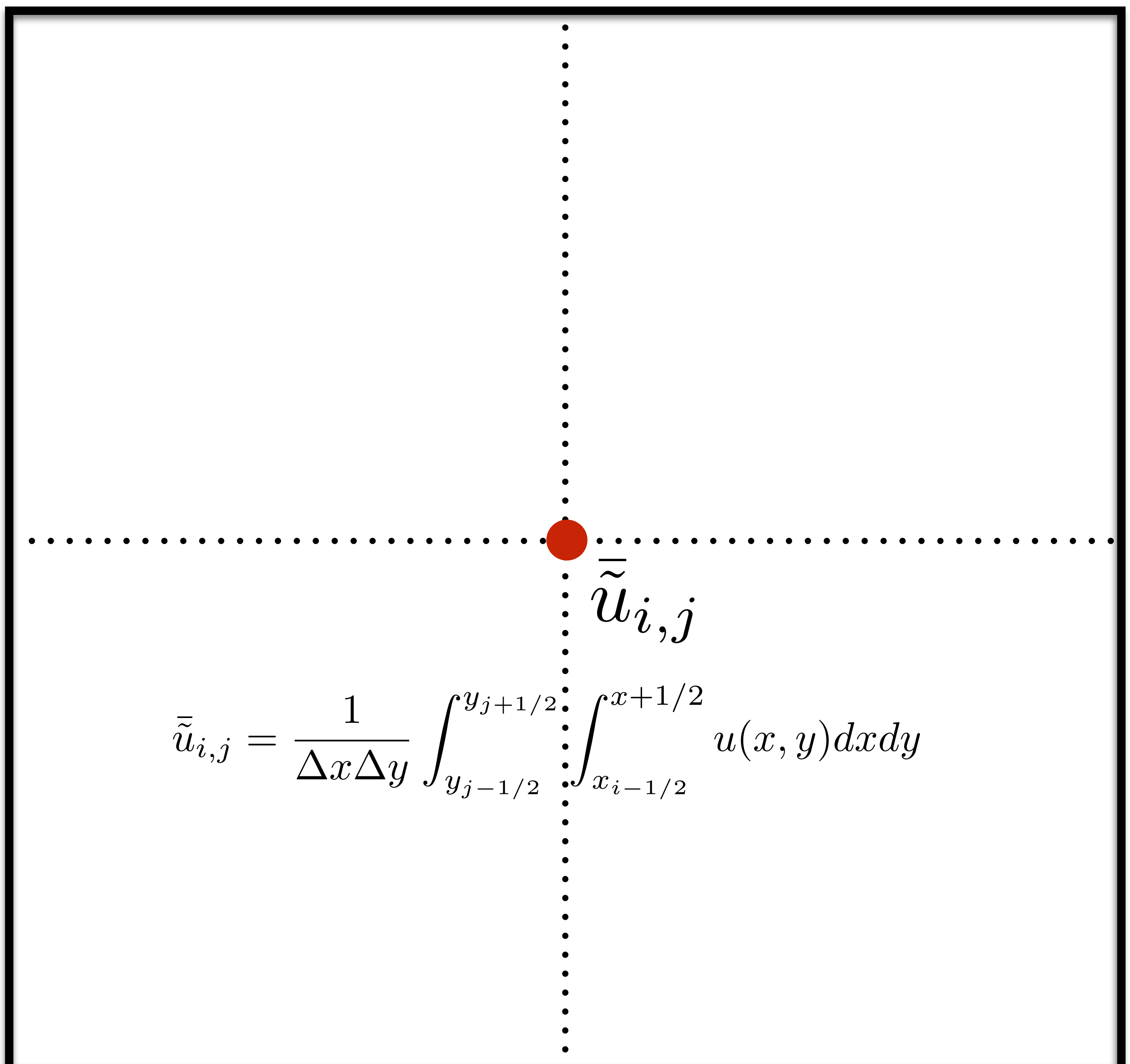
2D stencil for
2nd order PLM





2D stencil for
3rd order PPM;
5th order WENO

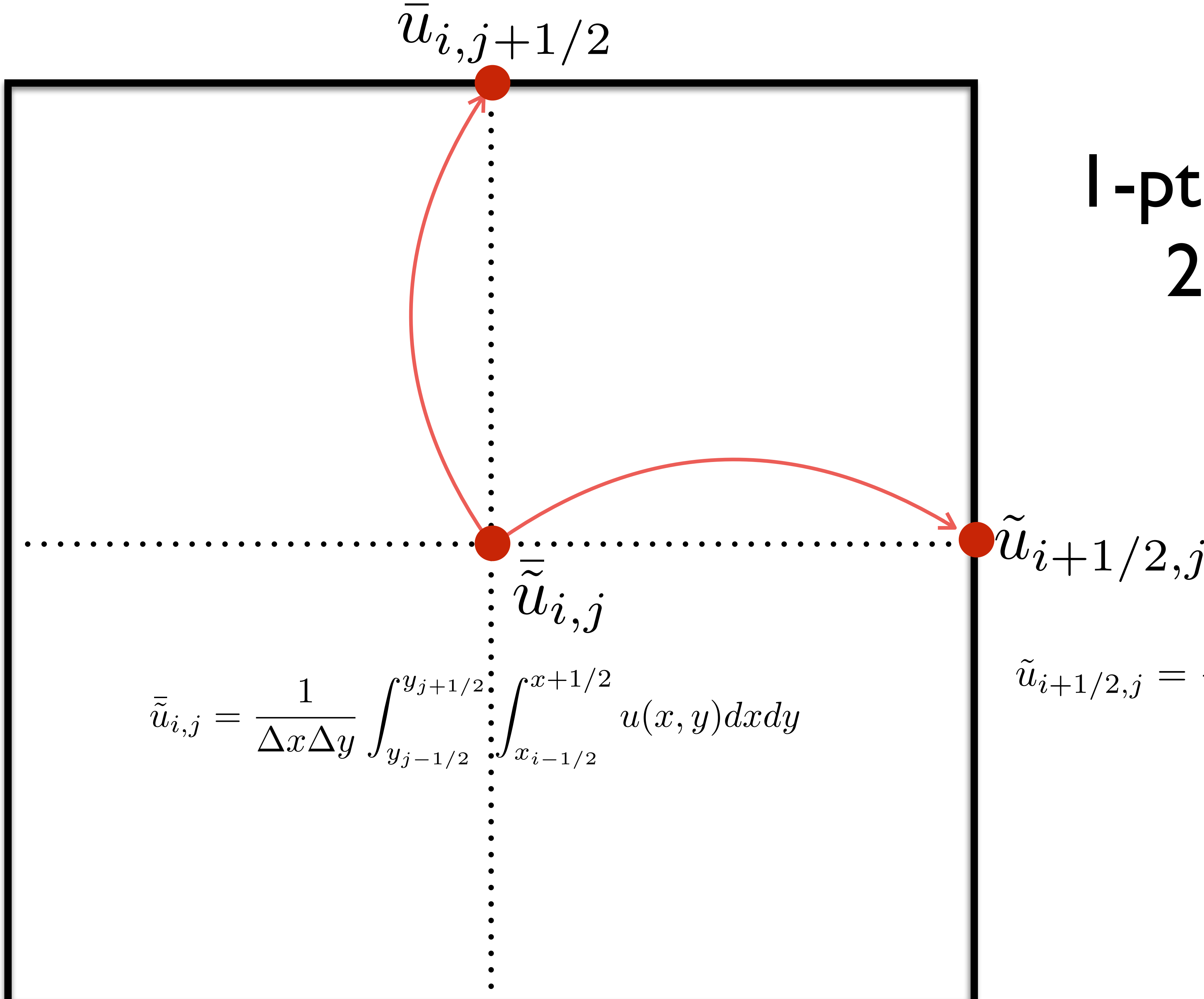
Complexity of High-Order with FVM



$$\bar{\tilde{u}}_{i,j} = \frac{1}{\Delta x \Delta y} \int_{y_{j-1/2}}^{y_{j+1/2}} \int_{x_{i-1/2}}^{x+1/2} u(x,y) dx dy$$

Complexity of High-Order with FVM

$$\bar{u}_{i,j+1/2} = \frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_{i+1/2}} u(x, y_{j+1/2})dx$$

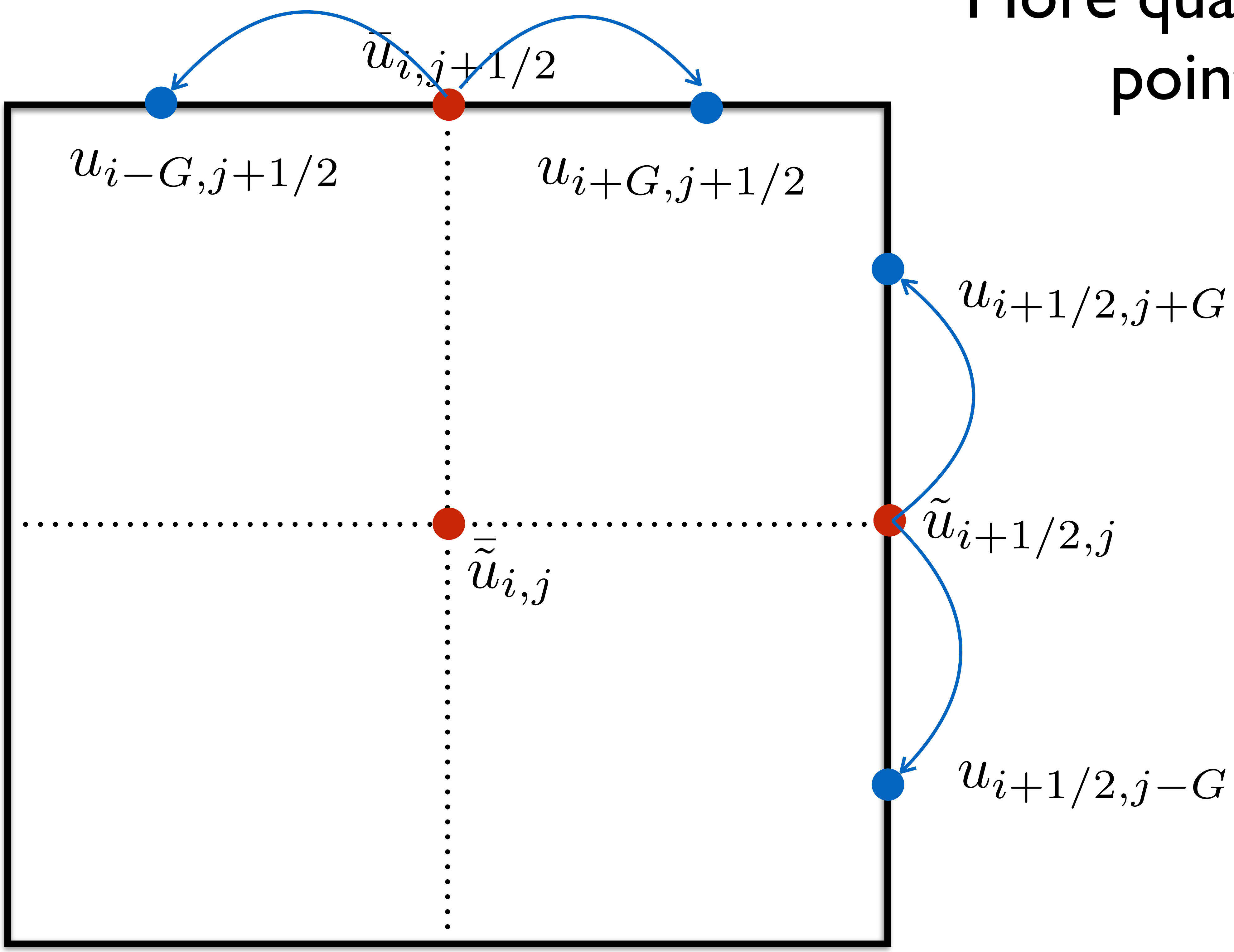


$$\bar{\tilde{u}}_{i,j} = \frac{1}{\Delta x \Delta y}\int_{y_{j-1/2}}^{y_{j+1/2}}\int_{x_{i-1/2}}^{x+1/2} u(x,y)dxdy$$

1-pt quadrature:
2nd-order

$$\tilde{u}_{i+1/2,j} = \frac{1}{\Delta y}\int_{y_{j-1/2}}^{y_{j+1/2}} u(x_{i+1/2}, y)dy$$

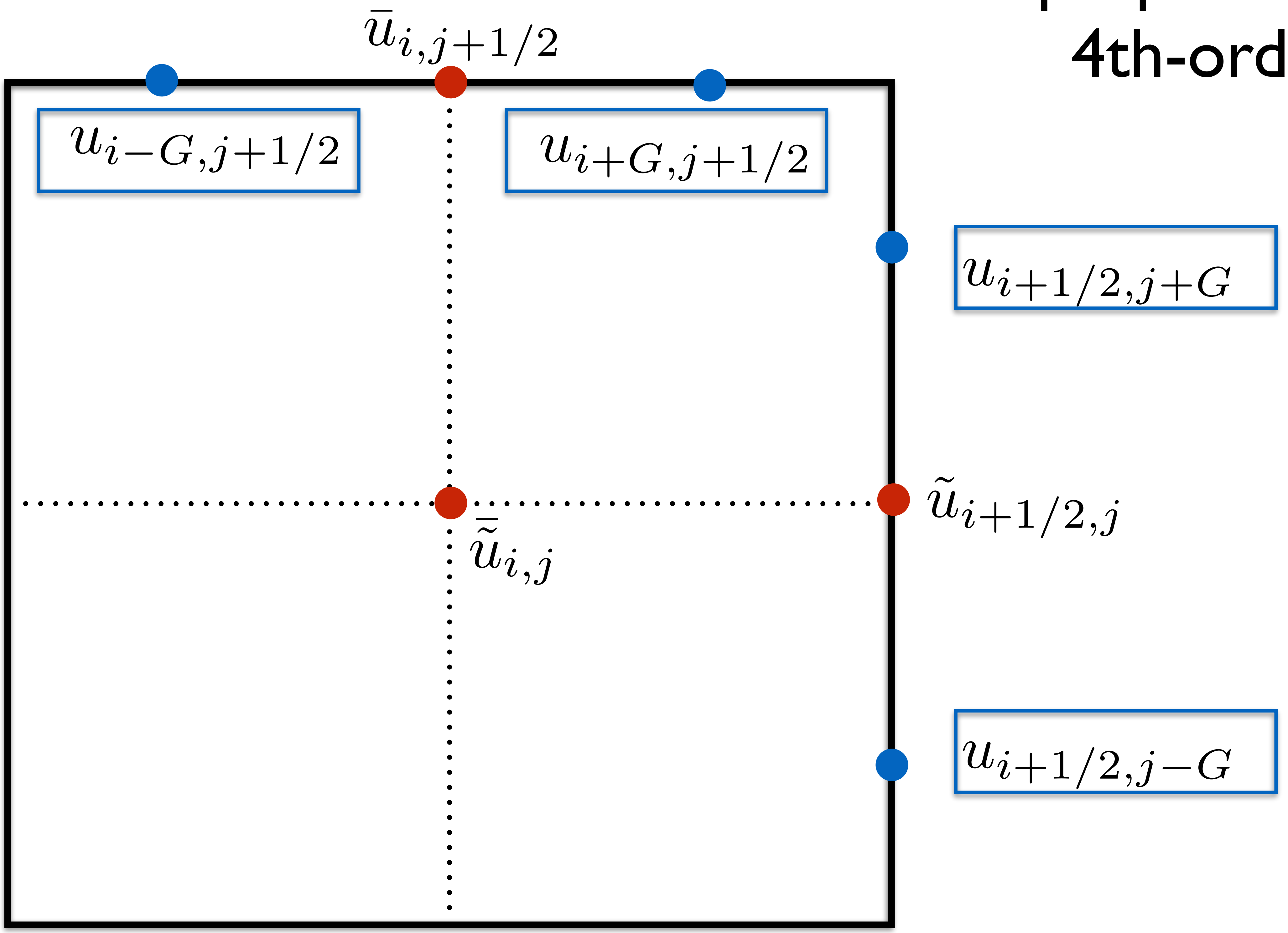
Complexity of High-Order with FVM



More quadrature points!

Complexity of High-Order with FVM

2-pt quadrature:
4th-order



Gaussian Process

- Multivariate Gaussian: f is normally distributed if

$$P(f) = \frac{1}{\sqrt{(2\pi)^n|\Sigma|}} \exp\Big(-\frac{1}{2}(f-\mu)^T\Sigma^{-1}(f-\mu)\Big)$$

$$f \in \mathcal{N}(\mu, \Sigma)$$

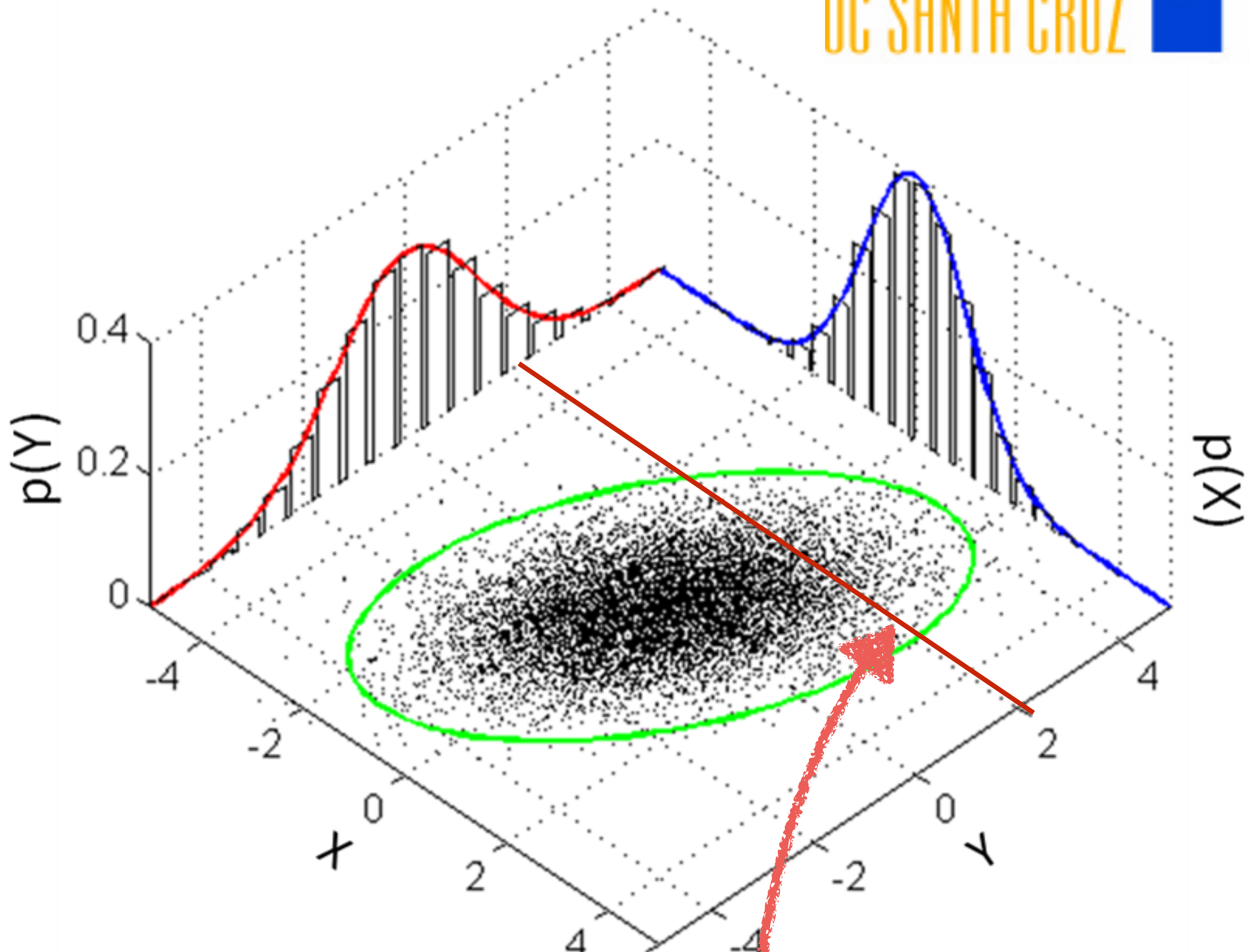
- Gaussian Process (GP) is a generalized version of a multivariate Gaussian (MG), extending the finiteness of MG to uncountably infinite (Rasmussen & Williams)

- GP is collection of random variables, any finite number of which have a joint multivariate Gaussian distribution

Joint (multivariate) Gaussian Distribution

- Joint normal distribution

$$\begin{bmatrix} x \\ y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}, \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{bmatrix} \right)$$



- Distribution of x conditional on measurement of y:

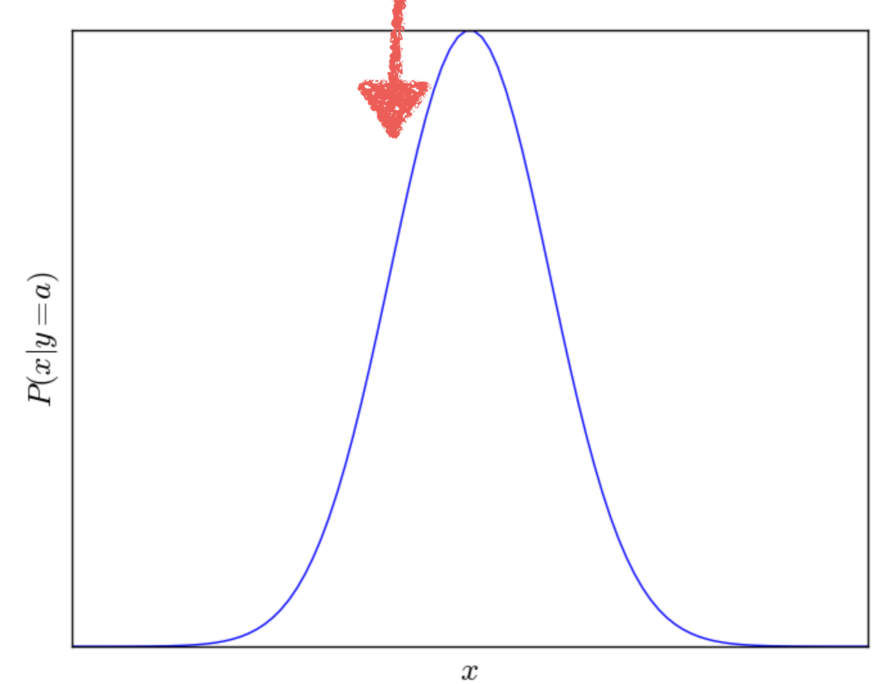
$$(x|y=a) \sim \mathcal{N}(\bar{\mu}, \bar{\Sigma})$$

- mean:

$$\bar{\mu} = \mu_x + \Sigma_{xy}\Sigma_{yy}^{-1}(a - \mu_y)$$

- variance:

$$\bar{\Sigma} = \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx}$$



Gaussian Process

- $f(x) \sim \mathcal{GP}(\mu(x), k(x, x'))$

- Completely specified by a mean and covariance function

$$\mu(x) = \mathbb{E}[f(x)]$$
$$k(x, x') = \mathbb{E}[(f(x) - \mu(x))(f(x') - \mu(x'))]$$

- Some popular choices:
  - ➡ constant mean function
  - ➡ Squared Exponential (SE) covariance

$$\mu(x) = C$$
$$k(x, x') = \exp\left(-\frac{(x - x')^2}{2l^2}\right)$$

GP Prediction via Conditioning

- We want to predict function value $f(x^*)$ at the point x^*, given samples:

$$\Big\{\Big(x_i, f(x_i)\Big)\Big| i = 1, \ldots, n\Big\}$$

$$\begin{bmatrix} \mathbf{f} \\ f^* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}(\mathbf{x}) \\ \mu(x^*) \end{bmatrix}, \begin{bmatrix} \mathbf{K} & \mathbf{K}_*^T \\ \mathbf{K}_* & k_{**} \end{bmatrix}\right)$$

?

$$\mathbf{K}_{ij} = k(x_i, x_j) \qquad \mathbf{K}_{*i} = k(x^*, x_i)$$

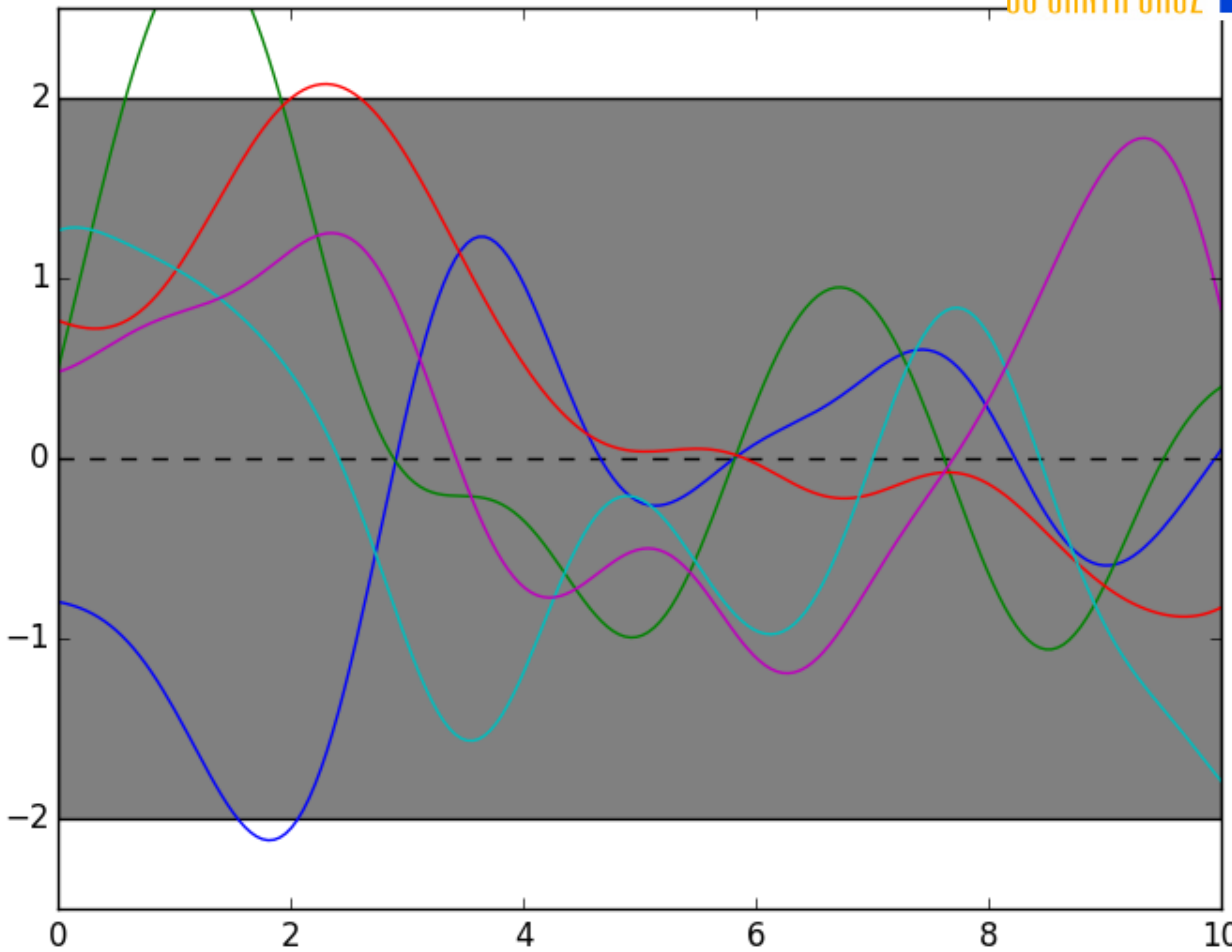
GP Prediction via Conditioning

$$f^*|x^*, \mathbf{x}, \mathbf{f} \sim \mathcal{N}(\bar{f}^*, \bar{\Sigma})$$

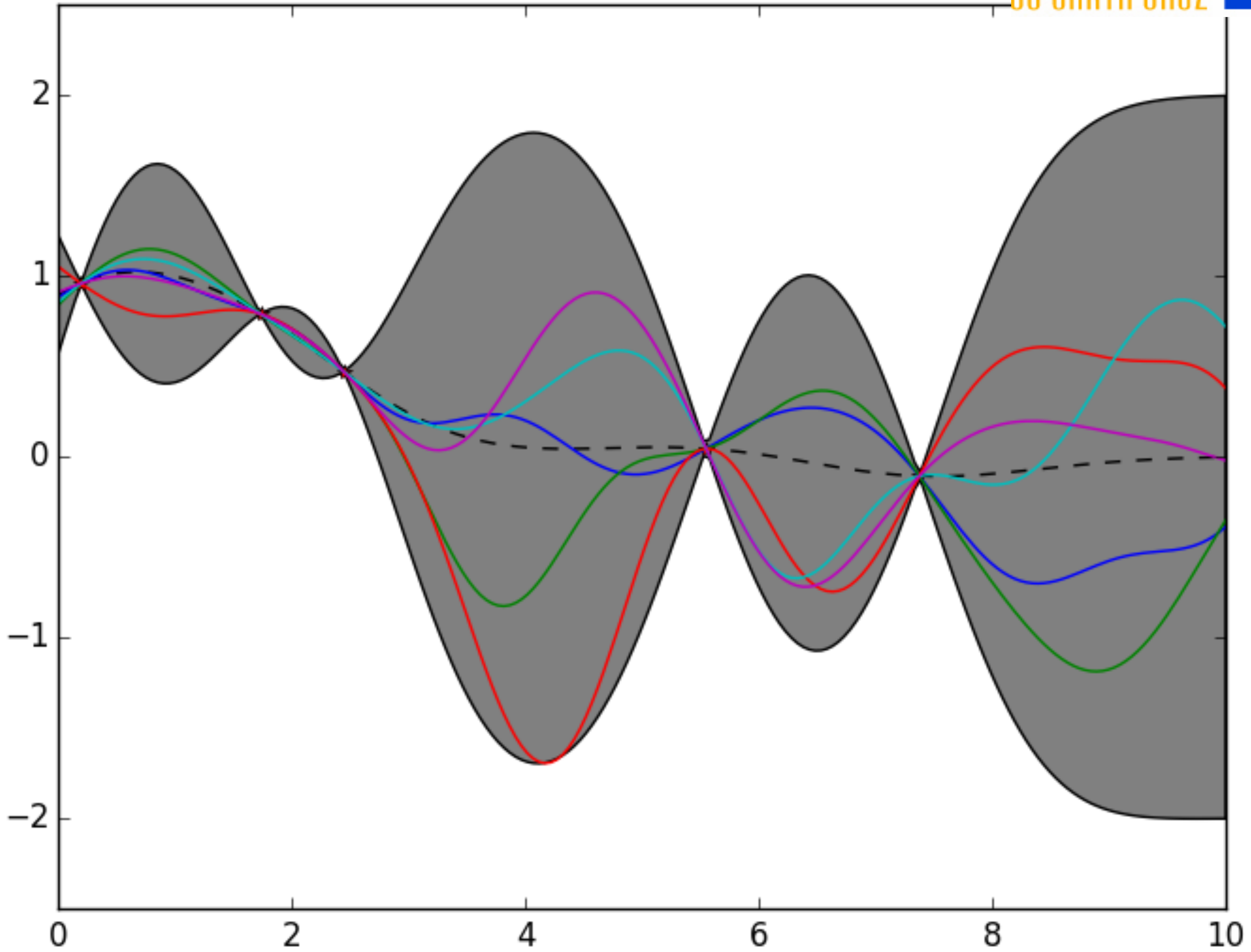
$$\boxed{\bar{f}^* = \mu(x^*) + \mathbf{K}_*^T \cdot \mathbf{K}^{-1}(\mathbf{f} - \boldsymbol{\mu}(\mathbf{x}))}$$

$$\bar{\Sigma} = k_{**} - \mathbf{K}_*^T \cdot \mathbf{K}^{-1} \cdot \mathbf{K}_*$$

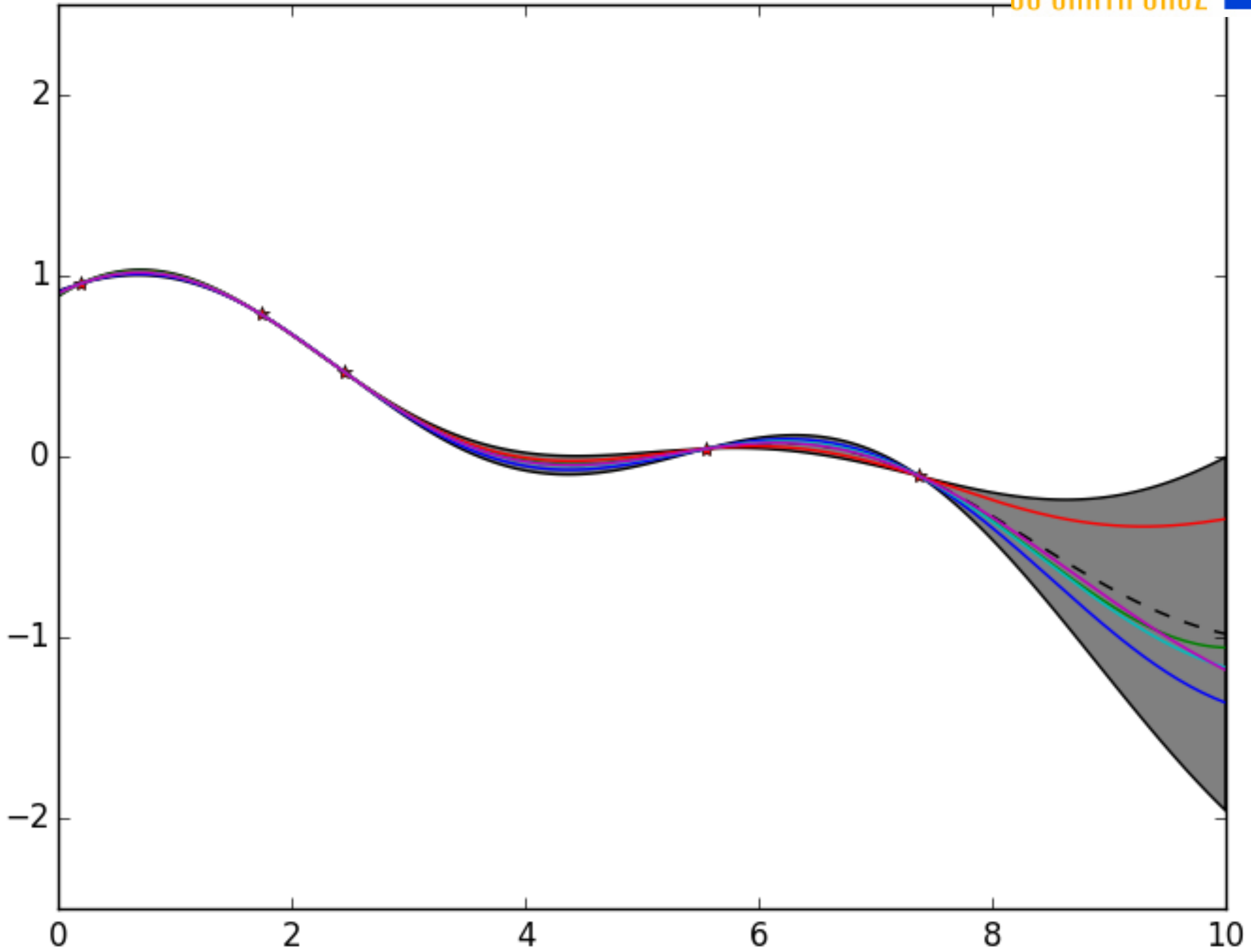
$$k_{SE} = \exp\left(-\frac{(x-x')^2}{2l^2}\right) \qquad l = 1$$



$$k_{SE} = \exp\left(-\frac{(x-x')^2}{2l^2}\right) \qquad l = 1$$



$$k_{SE} = \exp\left(-\frac{(x-x')^2}{2l^2}\right) \qquad l = 3$$



Grid Averaged GP for FVM

Given a pointwise: $\bar{f}^* = \mu(x^*) + \mathbf{K}_*^T \cdot \mathbf{K}^{-1}(\mathbf{f} - \boldsymbol{\mu}(\mathbf{x}))$

Integrate:

- $\mathbb{E}(G_\alpha)$:

$$\bar{G}_\alpha = \int dg_\alpha(x)\bar{f}(x)$$

- $\mathrm{cov}(f(x_\alpha), G_\beta)$:

$$T_{\alpha\beta} = \int dg_\beta(y)k(x_\alpha, y)$$

- $\mathrm{cov}(G_\alpha, G_\beta)$:

$$C_{\alpha\beta} = \int dg_\alpha(x)dg_\beta(y)k(x, y)$$

Pointwise vs. Averaged Values

$$u_i =< u_i > - \frac{\Delta x^2}{24} u''(x_i) - \frac{\Delta x^4}{1920} u^{(4)}(x_i) + \ldots$$

This conversion becomes important in high-order.

Otherwise, FVM become only 2nd order!

GP Reconstruction

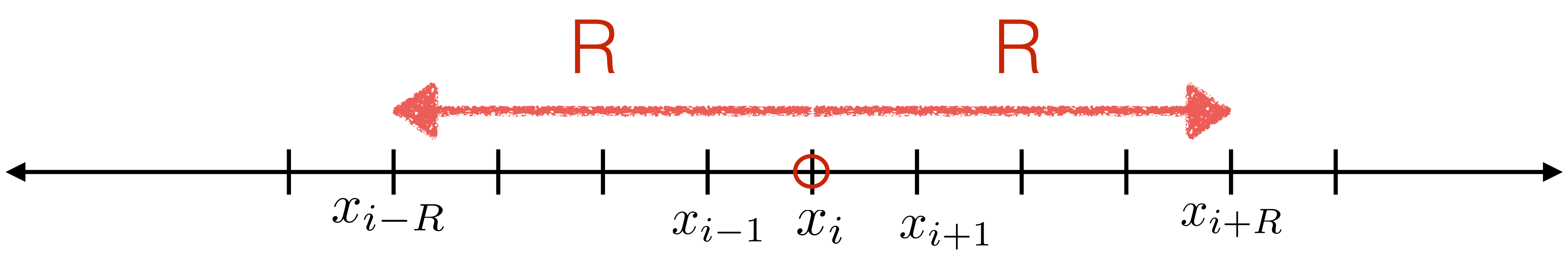
- Assume constant mean function

$$\mu(x) = f_0$$

- Prediction for left and right Riemann states

$$f(x^*) = f_0 + \mathbf{T}\mathbf{C}^{-1}(\mathbf{G} - f_0\mathbf{u}_N)$$

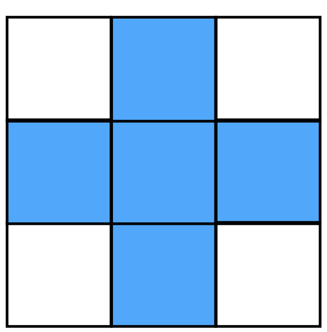
- Take (2R+1) samples ($\mathbf{G}$) around cell center
  - Compute $\mathbf{T}$ and $\mathbf{C}^{-1}$ using kernel function
    - $k(x, x')$ is PD: Cholesky decomposition for $\mathbf{C}^{-1}$



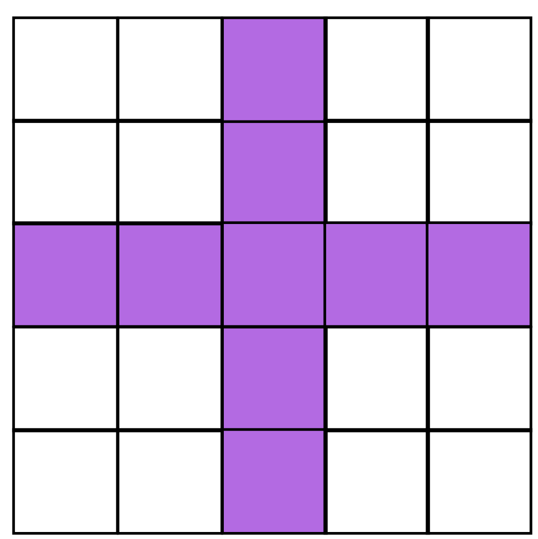
GP Reconstruction

- $\mathbf{T}$ & $\mathbf{C}^{-1}$ don't depend on the sample data
  - determined entirely by the grid geometry
  - identical for every cell in uniform grid

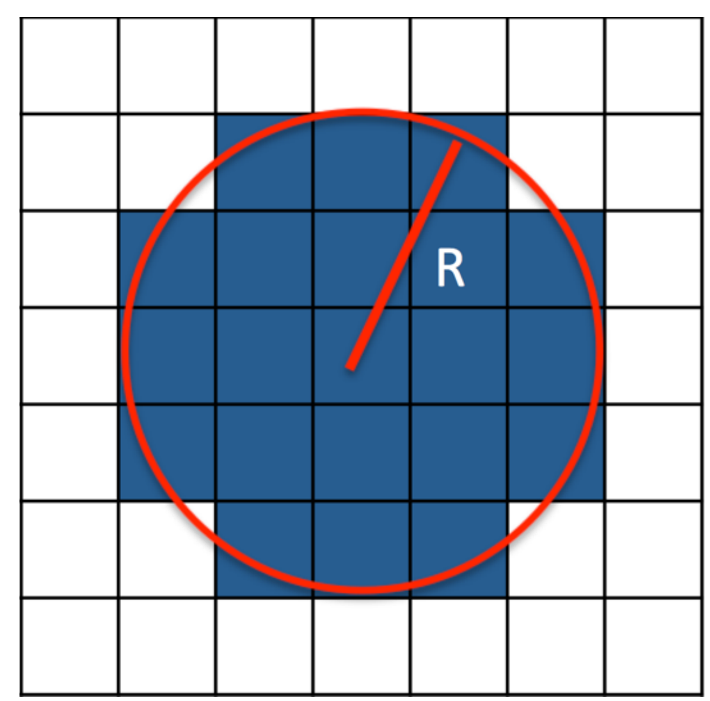
$$\mathbf{C}\mathbf{Z}^T = \mathbf{T}^T \qquad f(x^*) = f_0 + \mathbf{Z}(\mathbf{G} - f_0\mathbf{u}_N)$$



2D stencil for PLM



2D stencil for PPM, WENO



2D stencil for GP

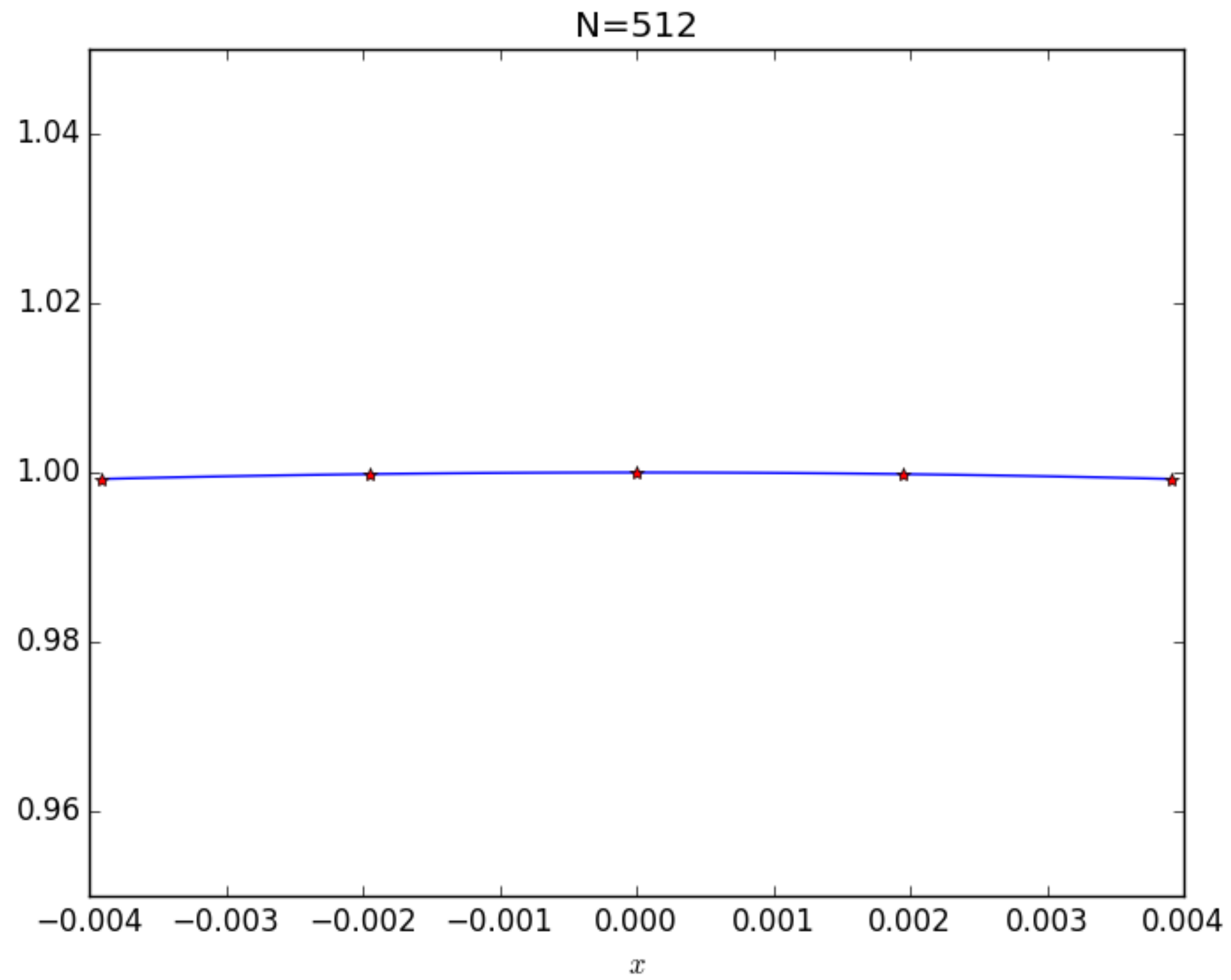
Singularity of C

$\mathbf{C}$ becomes singular!

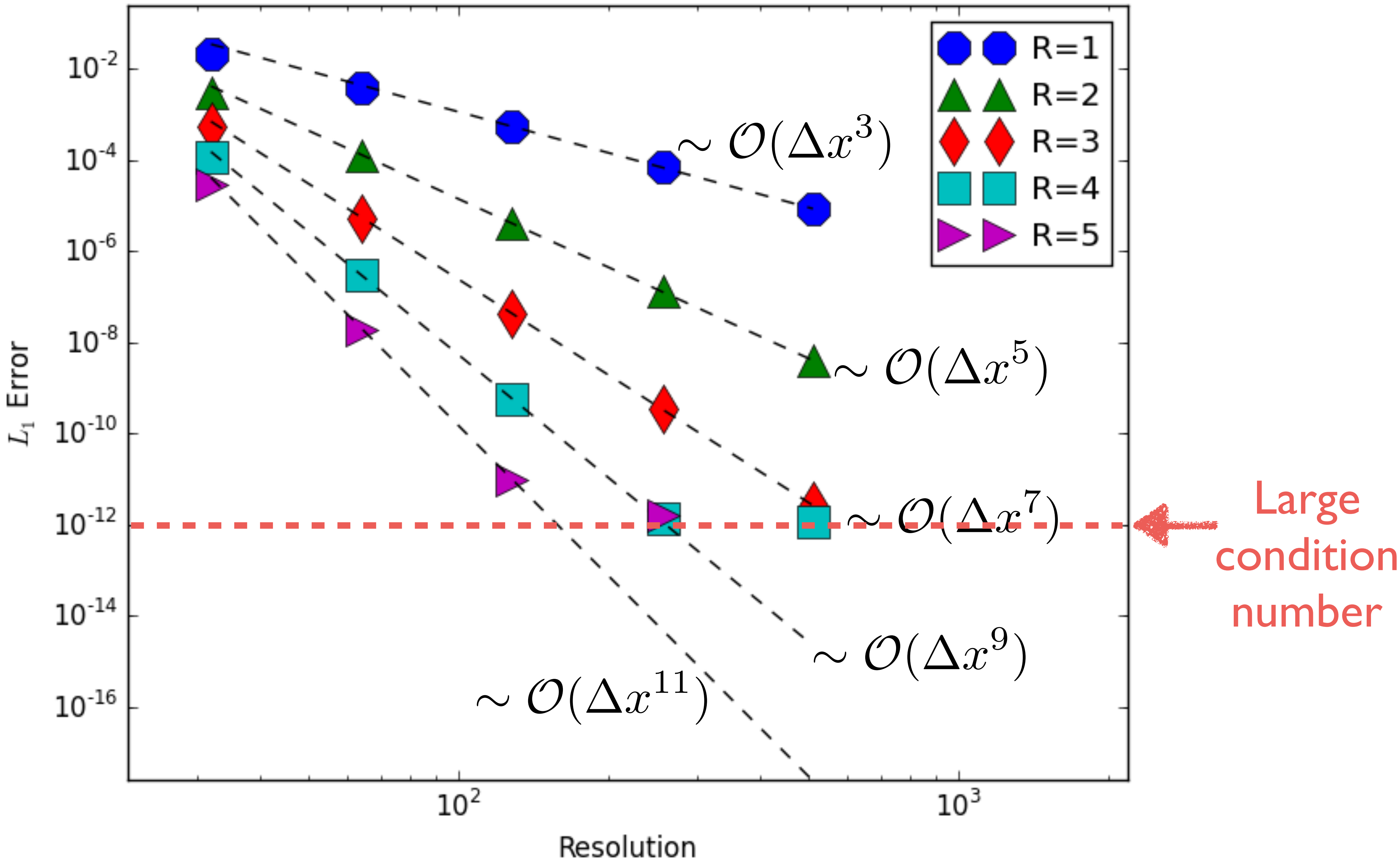
Condition Number:

$$\kappa \approx 10^{12}$$

$$k_{SE} = \exp\left(-\frac{(x - x')^2}{2l^2}\right)$$

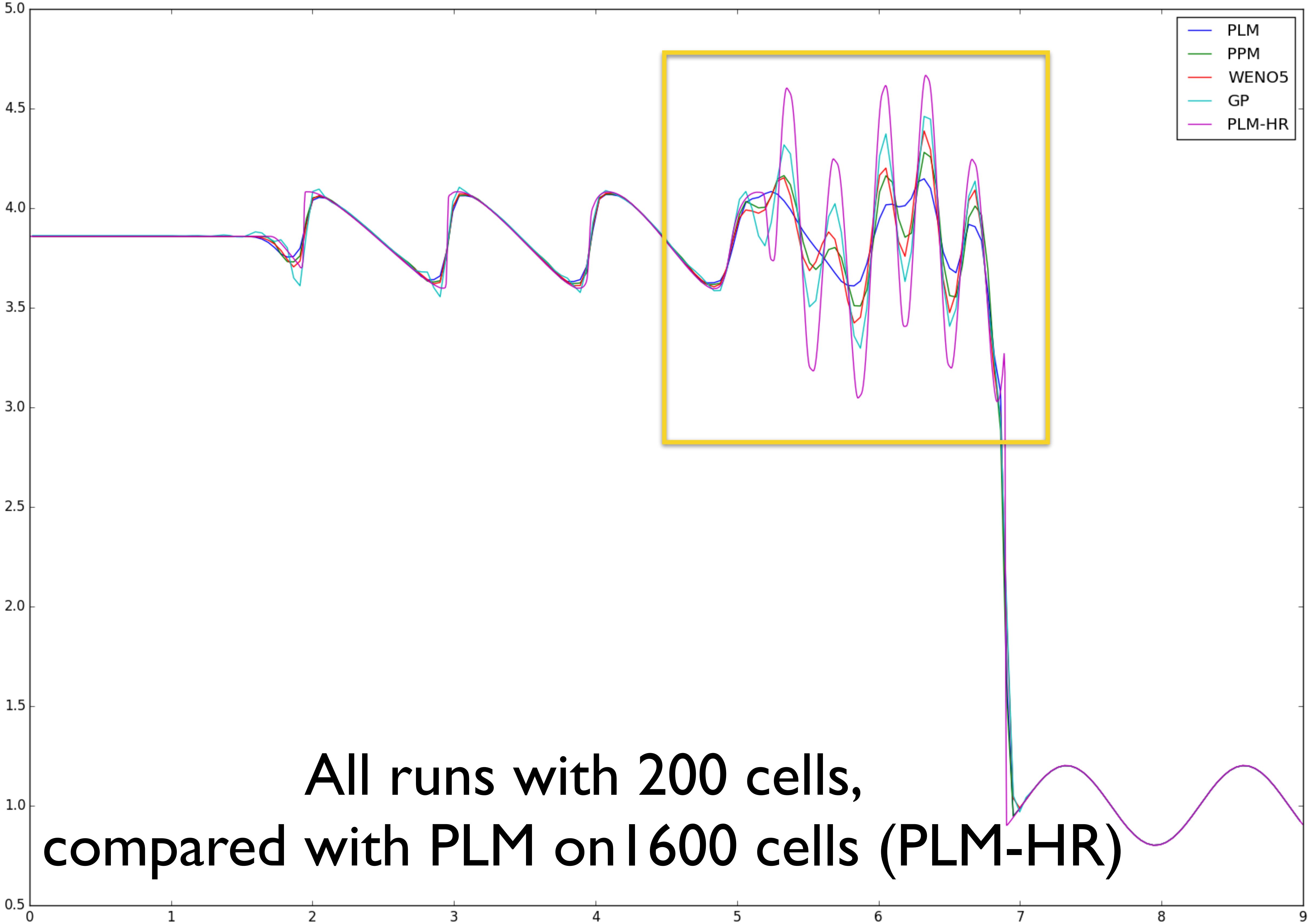


Convergence for 1D Smooth Advection

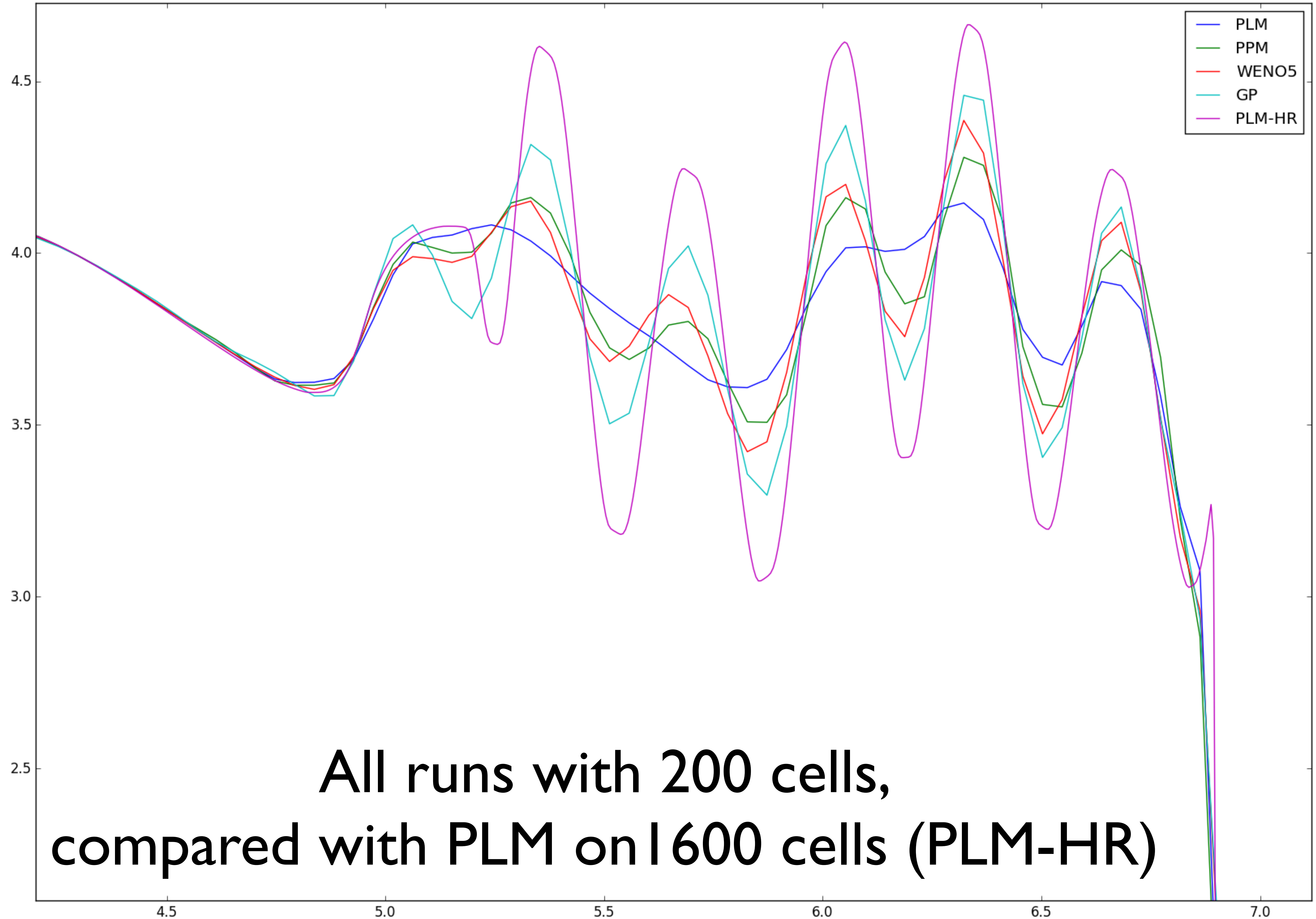


Convergence rate: 2R+1

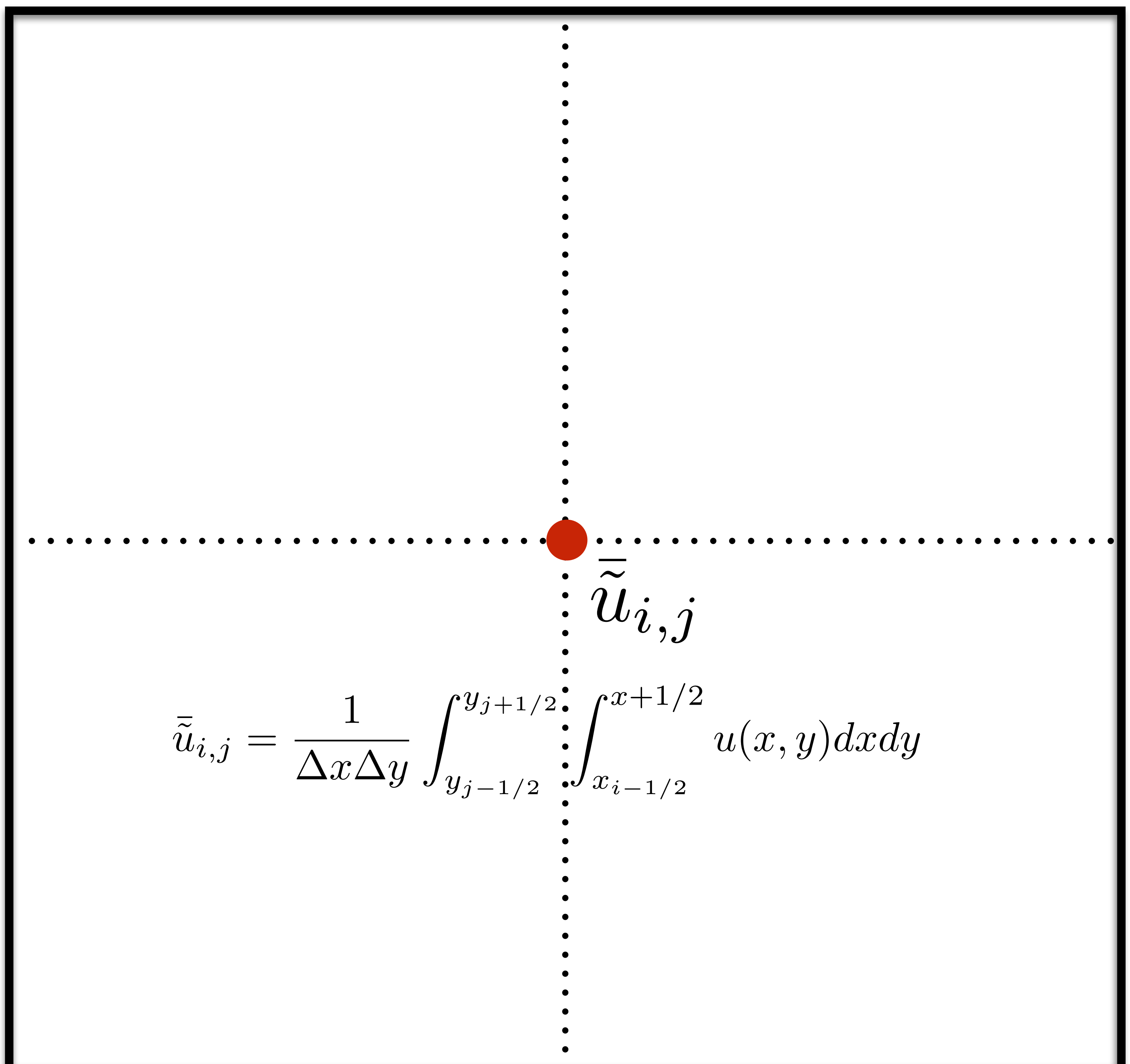
Shu-Osher 1D Mach 3 Shock



Shu-Osher 1D Mach 3 Shock



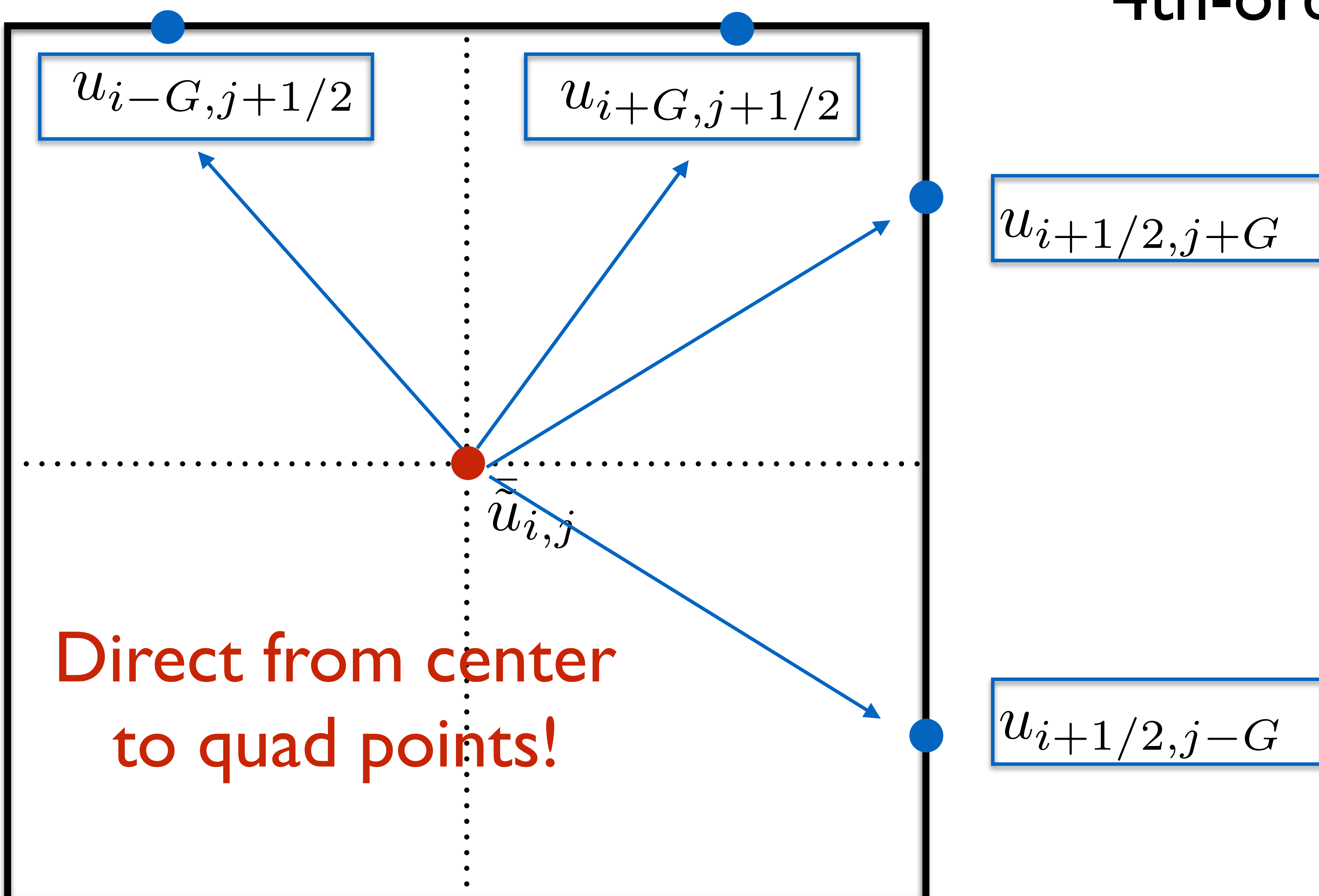
Complexity of High-Order with FVM



$$\bar{\bar{u}}_{i,j} = \frac{1}{\Delta x \Delta y} \int_{y_{j-1/2}}^{y_{j+1/2}} \int_{x_{i-1/2}}^{x+1/2} u(x,y) dx dy$$

Easiness of High-Order with GP

2-pt quadrature:
4th-order



Performance of GP

Methods	Relative Speedup
GP	1
FOG	0.5
PPM	2.8
WENO5	2.3

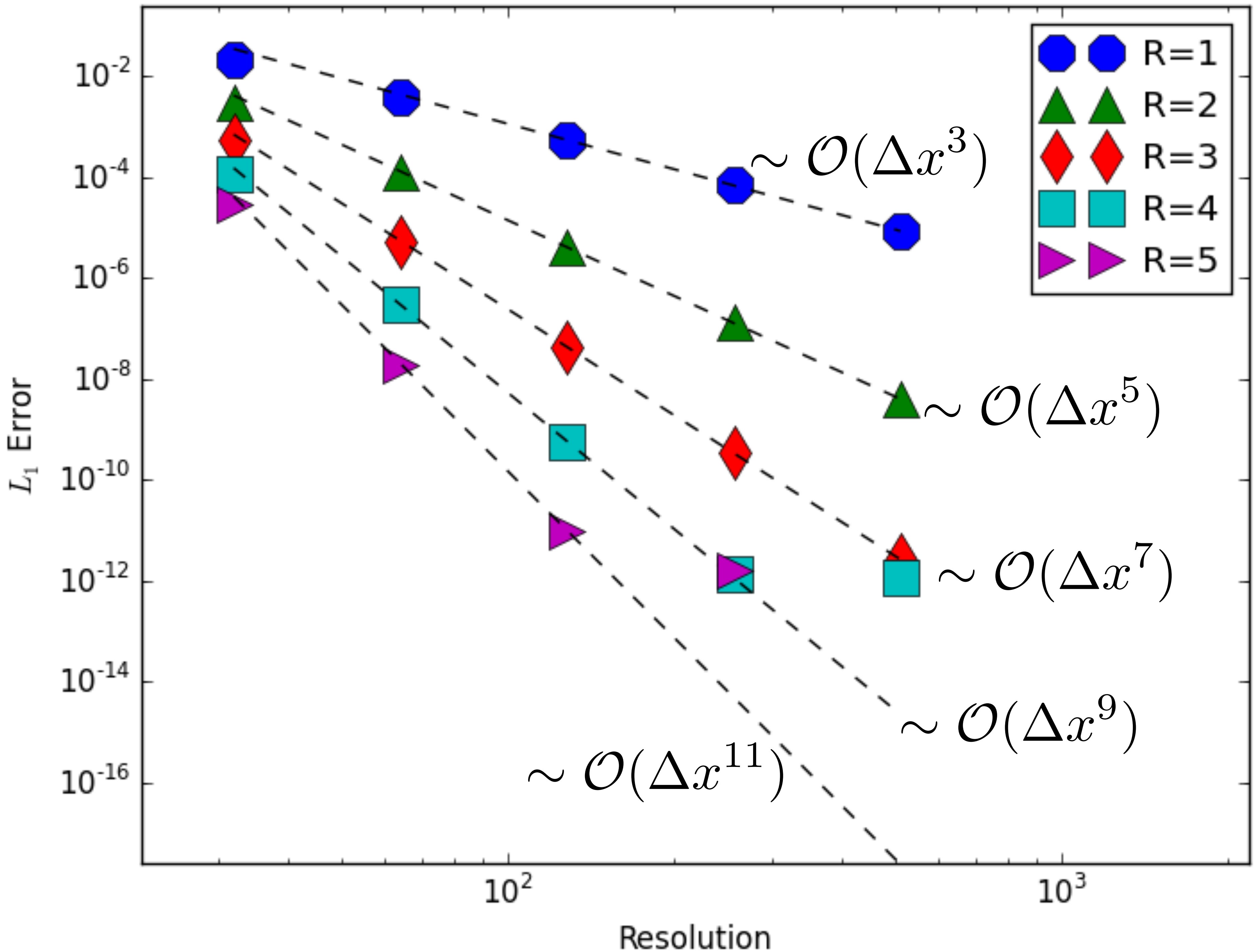
Conclusion

- Relatively painless extension to higher order in FVM
- Capable of better and faster errors than polynomial methods
- Lots of promise on:
  - AMR
  - Genuinely multidimensional for
    - interpolation and reconstruction
    - Riemann problems using multiD flux quadrature points

Thank You

Supplementary Slides

Convergence for 1D Smooth Advection

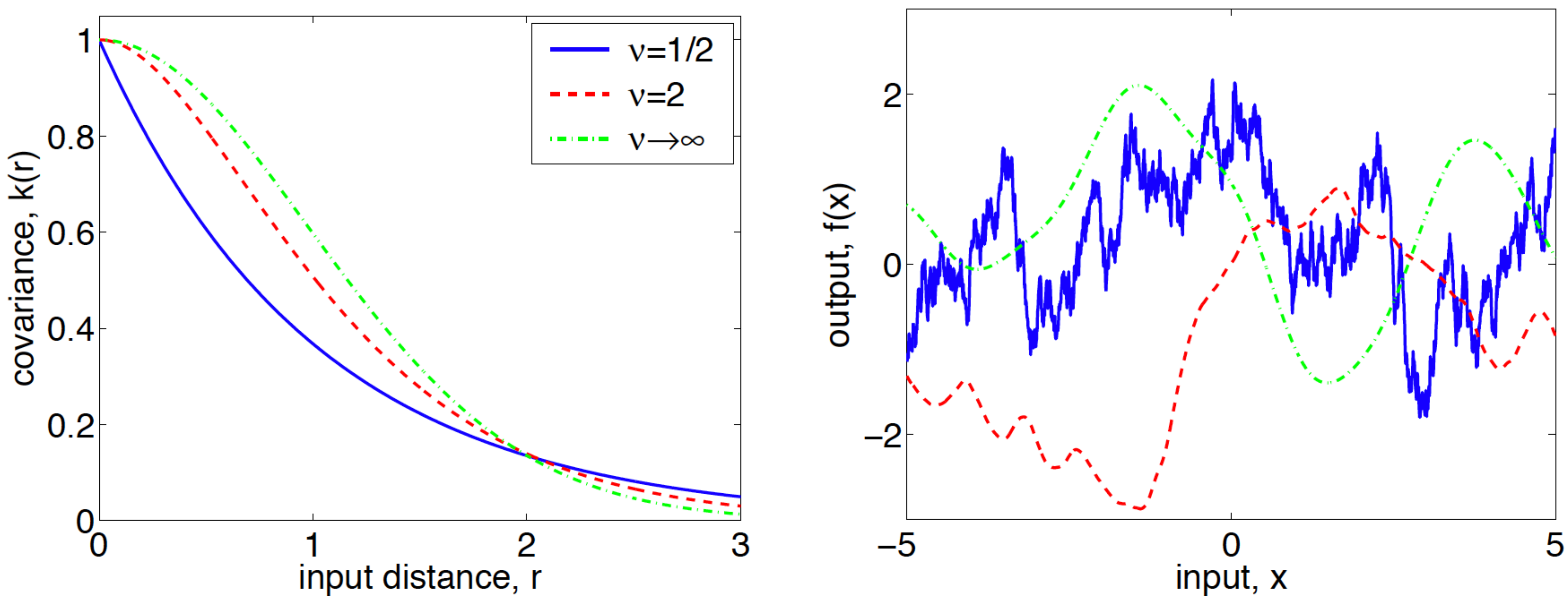


Convergence rate: 2R+1

Other Kernels

$$K_{SE}(r) = \exp(-\frac{r^2}{2l^2})$$

$$K_{Matern}(r) = \frac{2^{1-\nu}}{\Gamma(\nu)}\Big(\frac{\sqrt{2\nu r}}{l}\Big)^{\nu} K_{\nu}\Big(\frac{\sqrt{2\nu r}}{l}\Big)$$



(a) (b)

Model Selection with l

- $k_{SE} = \exp\left(-\frac{(x-x')^2}{2l^2}\right)$

- We want:
  - $l \geqslant R > \Delta$
  - $l \approx L$, where L is the length scale under consideration

- Currently we have:
  - $l = constant$

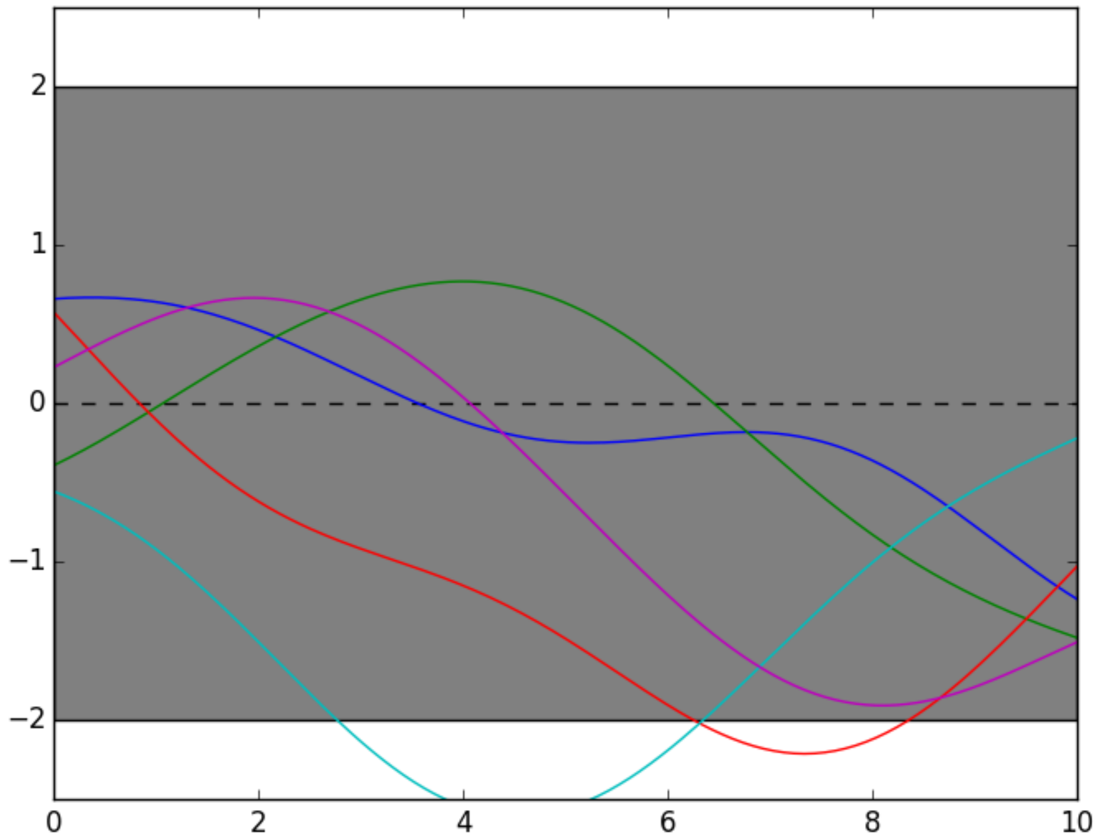
- Also can try:

  $l = l(x, y, z, t)$

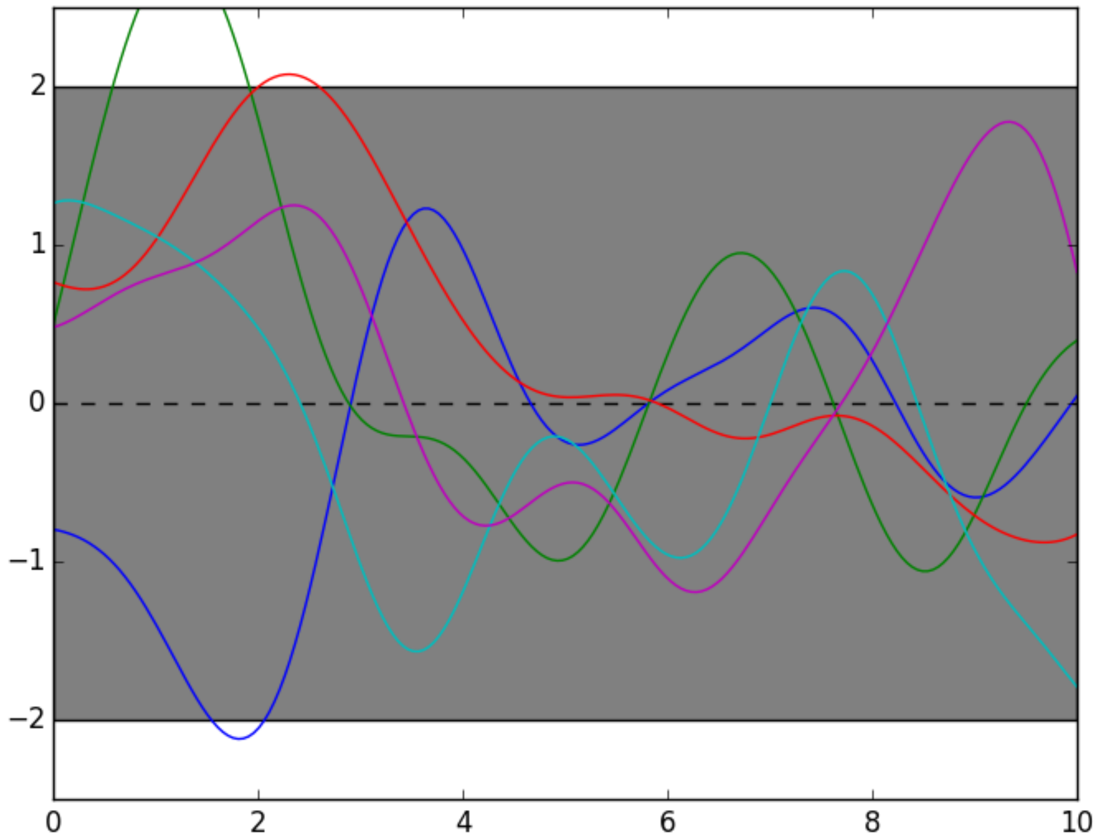
Model Selection with l

- Small l begins to fit high frequency behavior not present in underlying function

$l = 3$

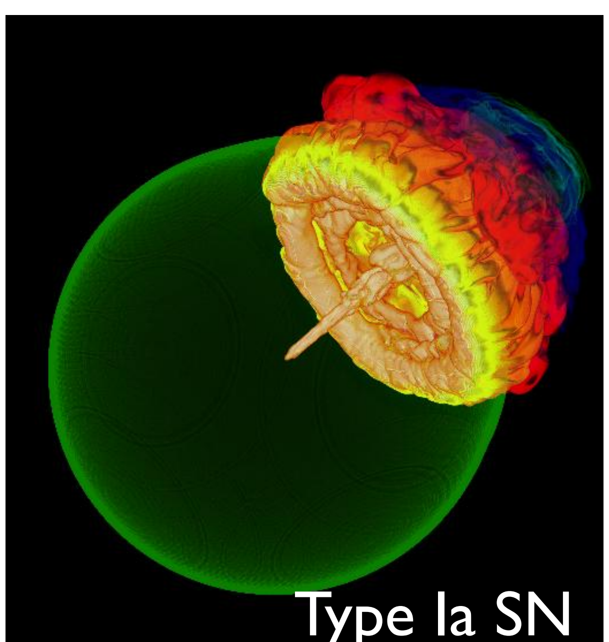


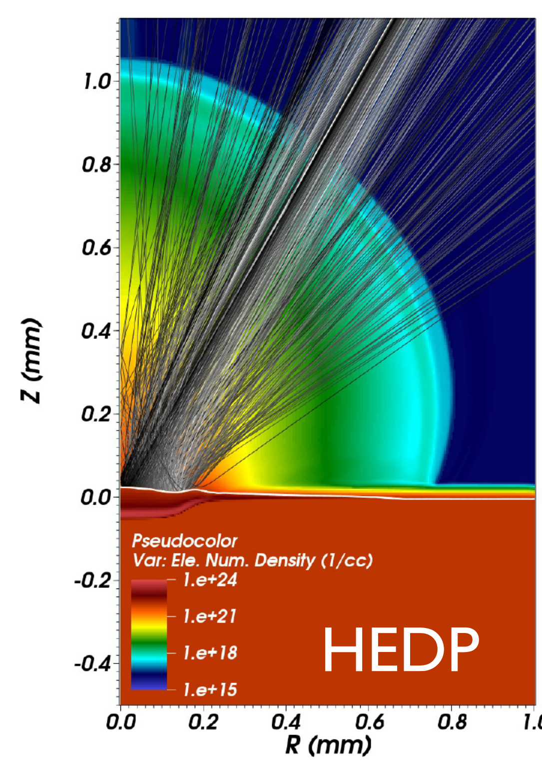
$l = 1$



Scientific Goal

To develop solution accurate, efficient, and stable numerical algorithms for a wide range of physical regimes using high-performance computer simulations

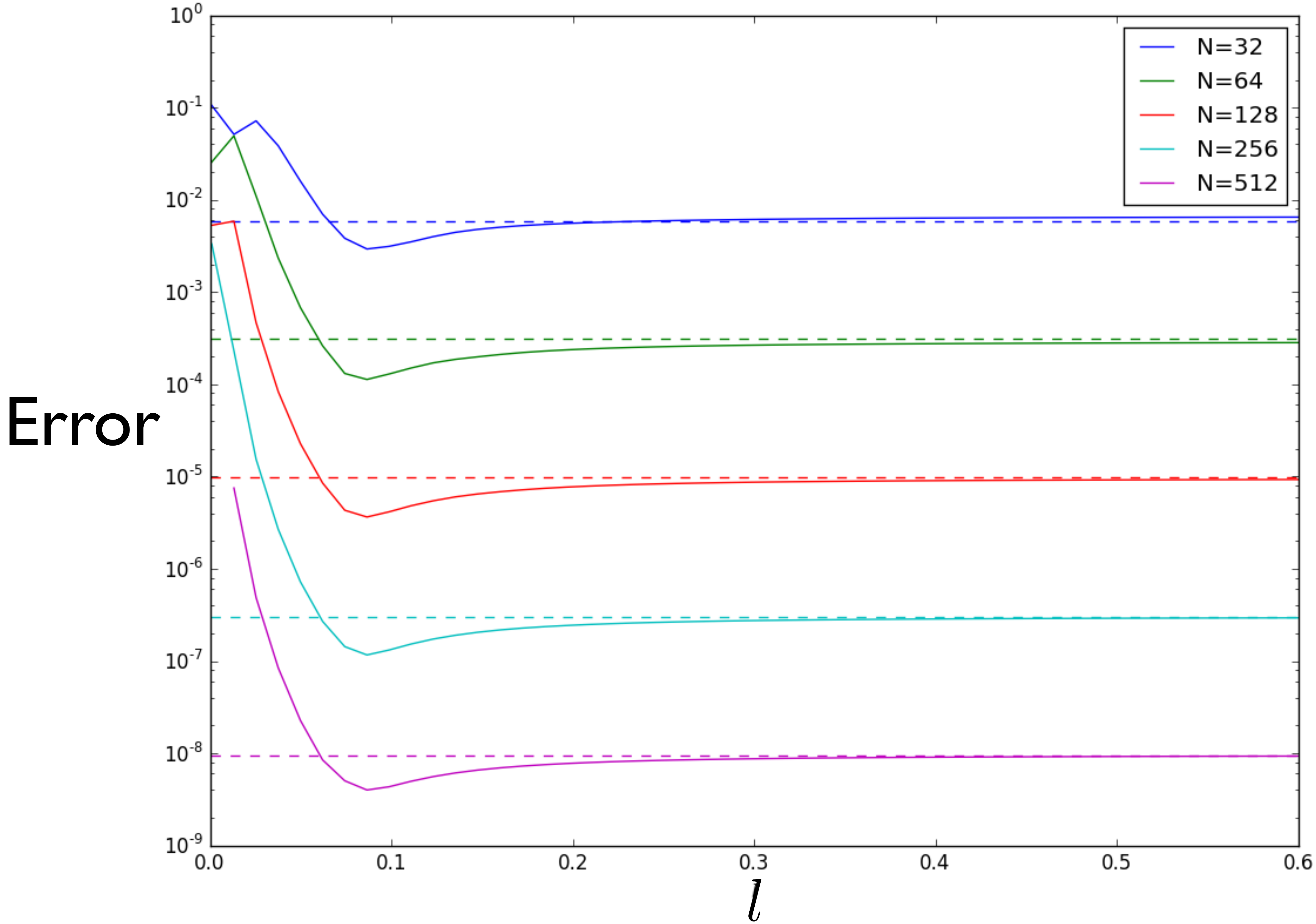




High-Order Numerical Algorithms

- A good solution to this is to use high-order algorithms
- They provide more accurate numerical solutions using
  - less grid points (=memory save)
  - higher-order mathematical approximations (promoting floating point operations, or computation)
  - faster convergence to solution

Model Selection with l



High Performance Computing

- To solve large problems in science, engineering, or business
- Modern HPC architectures have
  - increasing number of cores
  - declining memory/core
- This trend will continue for the foreseeable future

GP Prediction via conditioning

- Given n samples of

$$(x_i, f(x_i)), i = 1, \ldots, n$$

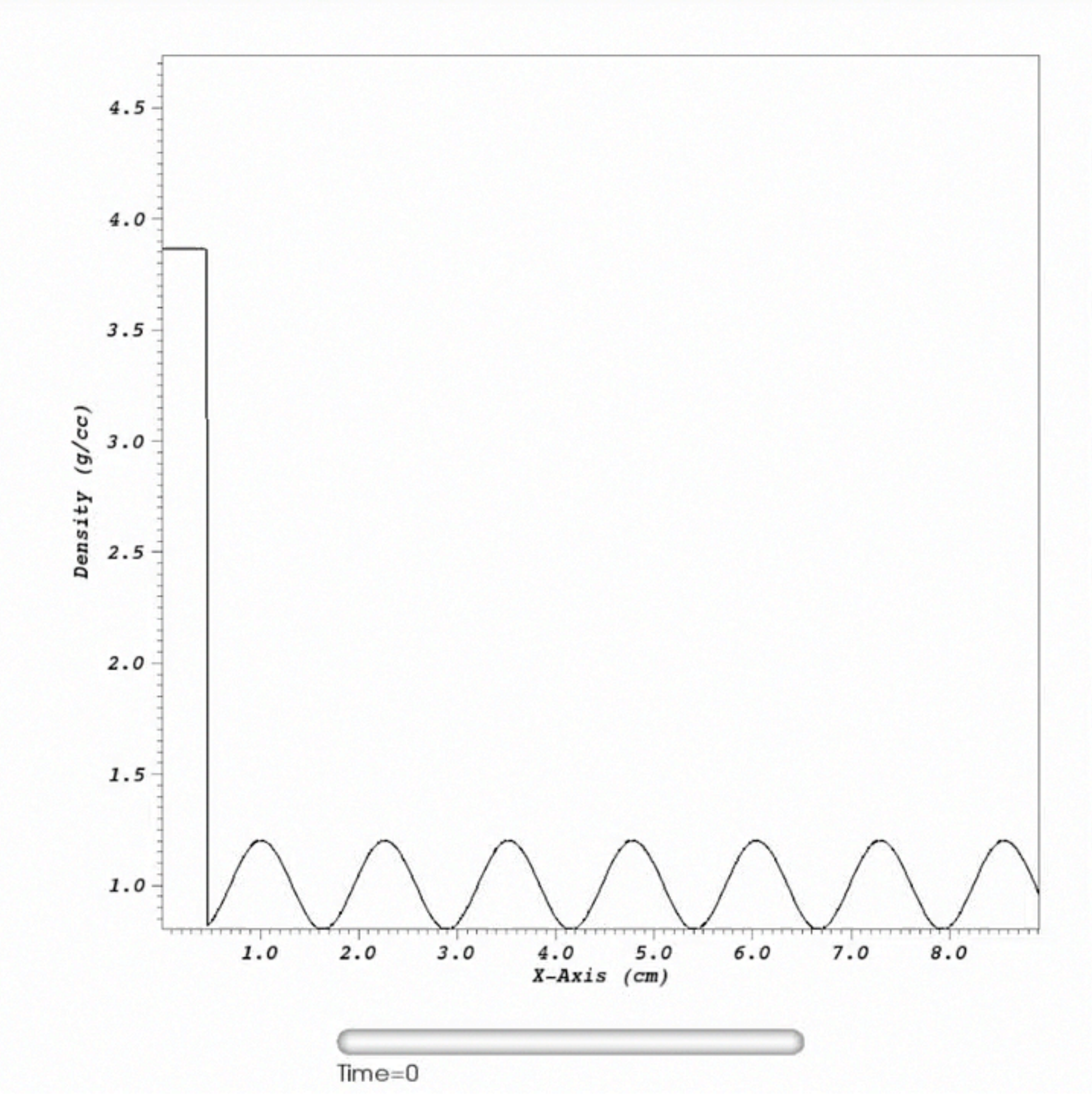
  - Predict the function value f(x*) at point x*

- Using an arbitrary mean function and a covariance function, GP defines a Gaussian distribution over space of functions

- Together with the sample points get a distribution for f(x*) conditional on the "training" points

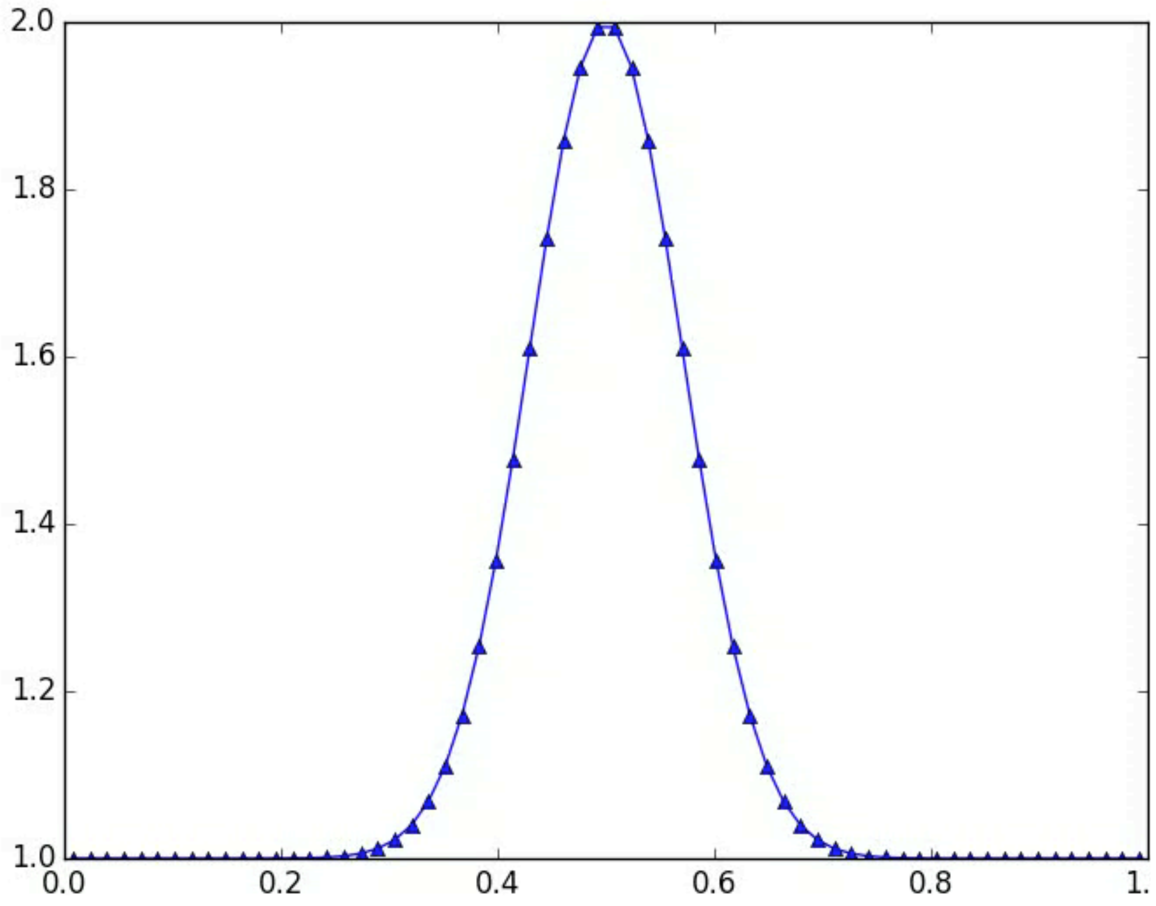
  - Gives an expected value of f(x*)

Shu-Osher 1D Mach 3 Shock

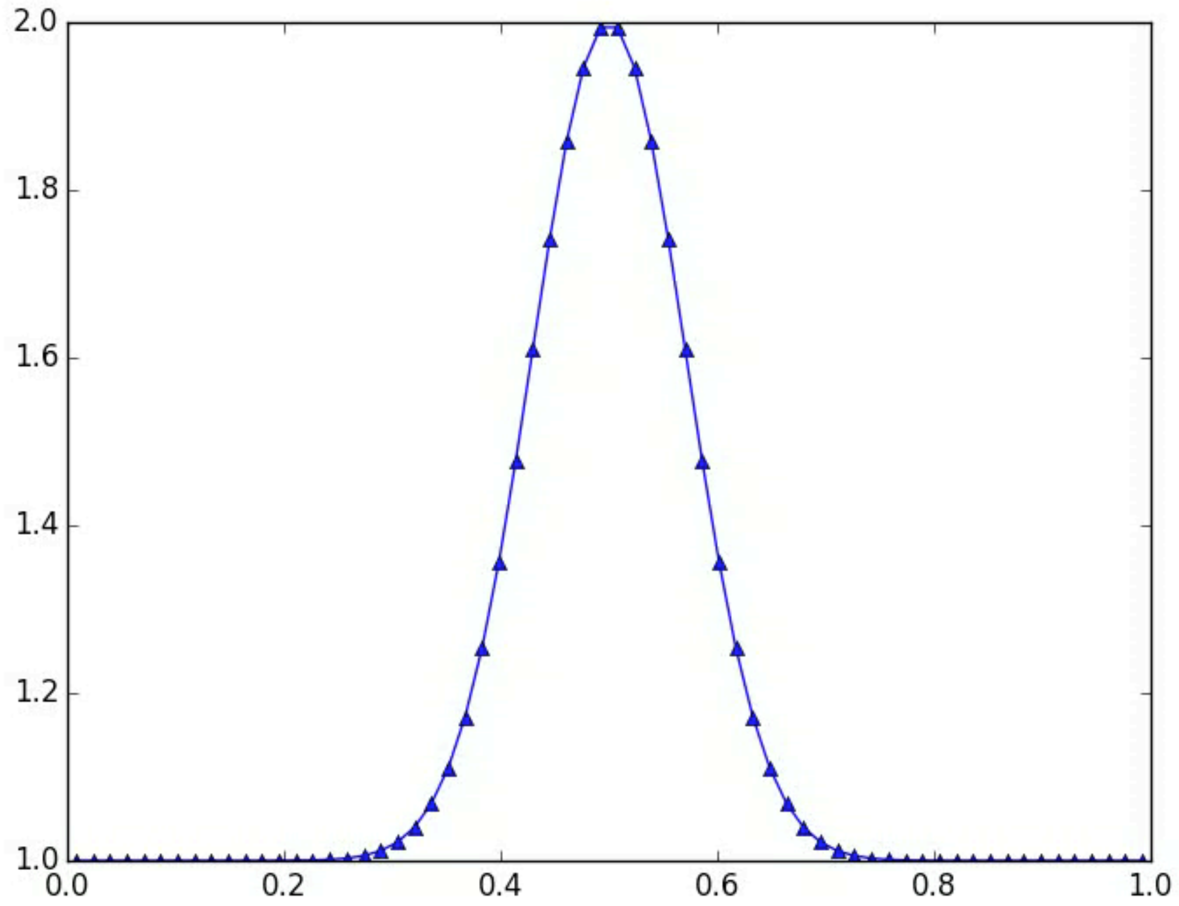


High-Order Numerical Algorithms

FOG

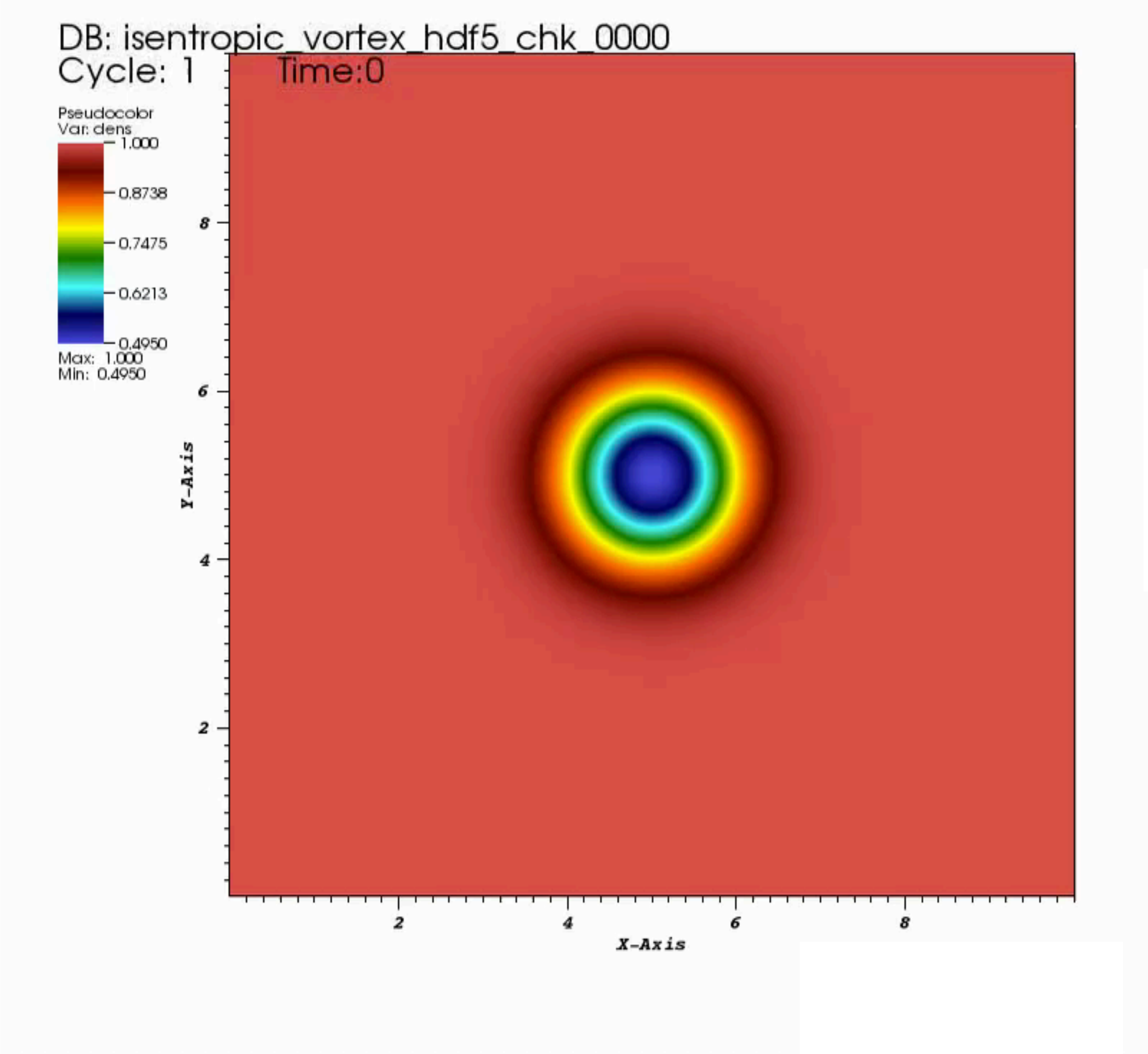


WENO5



2D Smooth Flows

- 2D advection of an isentropic vortex along the domain diagonal on a periodic box $(R = 2\Delta, \sigma = 6\Delta)$



Scheme	ρ	u	v	P
PLM	2.37E-2	8.63E-2	8.62E-2	3.08E-2
PPM	5.44E-4	1.88E-3	1.99E-3	6.93E-4
GP	1.33E-4	4.56E-4	3.75E-4	1.63E-4

Table 1: L1 error norm for the vortex problem.

Results on Smooth Flows II

- 1D advection of Gaussian profile ($R = 2\Delta, \sigma = 12\Delta$)

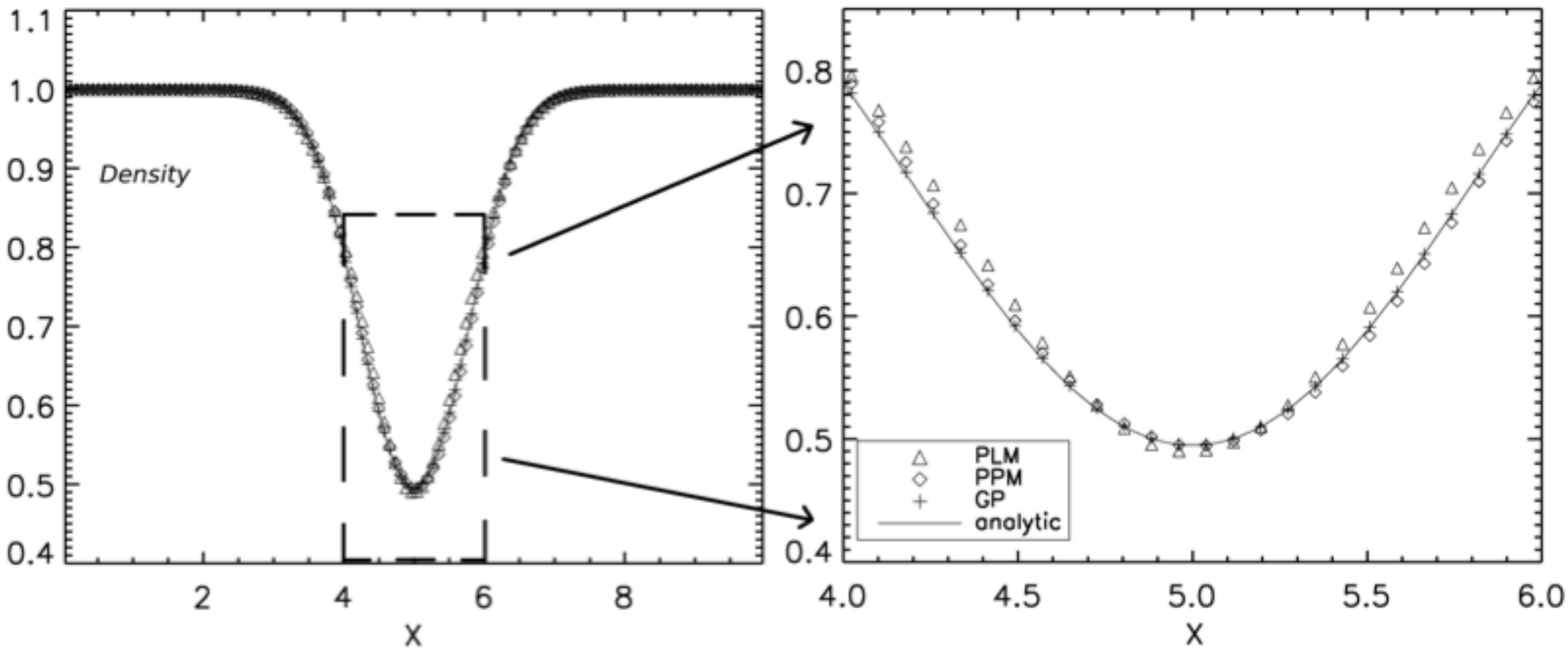
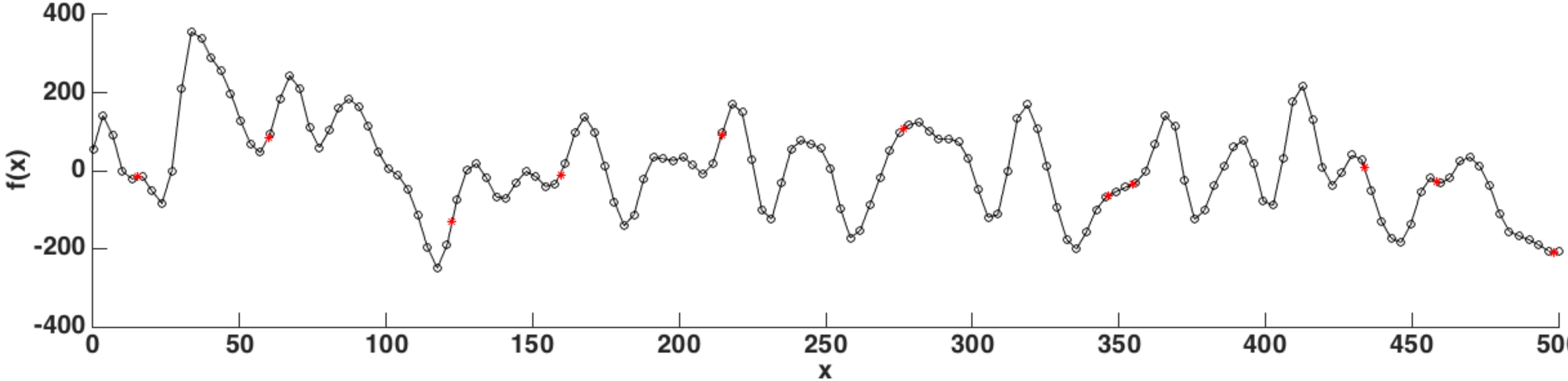


Figure 2: Left panel: One dimensional profile of density after one transition superposed on the initial condition (solid line). The symbols represent different interpolation schemes, namely cross for GP, diamond for PPM and triangle for PLM. Right panel: Close-up in the central region. The solution recovered with GP matches perfectly the reference solution, while the errors are smaller with respect to the other schemes.

Data Prediction by Gaussian Processes



- A pictorial illustration of GP prediction. A randomly drawn function f from a GP prior, $f \sim \mathcal{GP}(\mathbf{0}, K)$, is represented in a solid line by joining a set of 150 equally spaced data points (circles).

- The red symbols are the GP predicted values, whose locations are chosen arbitrarily at multiple locations.