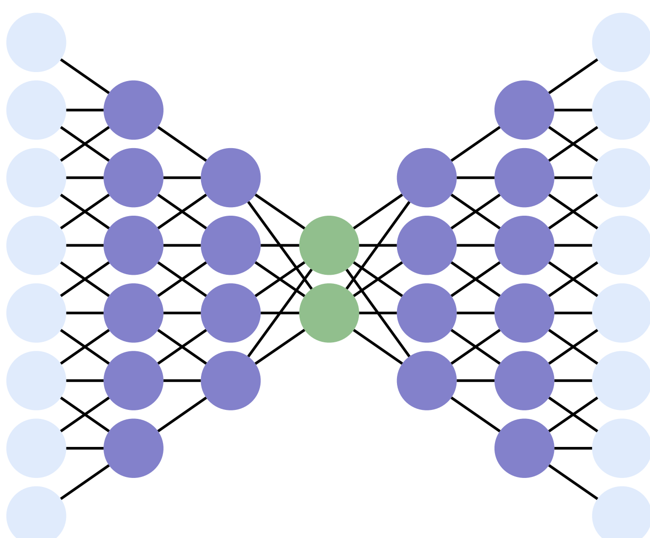
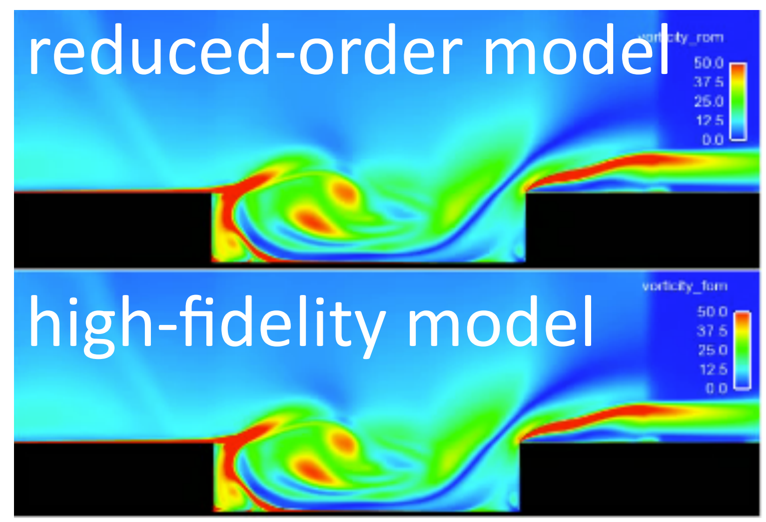
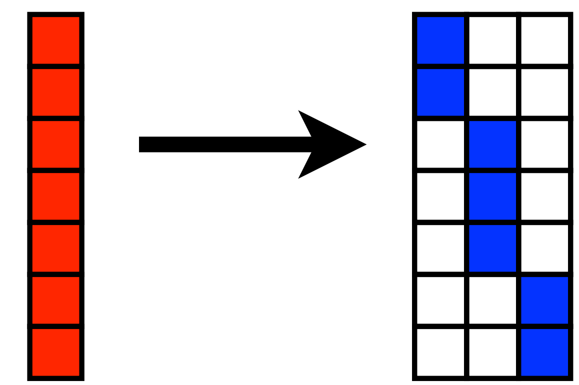
Nonlinear model reduction

Using machine learning to enable extreme-scale simulation for time-critical aerospace applications







Kevin Carlberg

Sandia National Laboratories

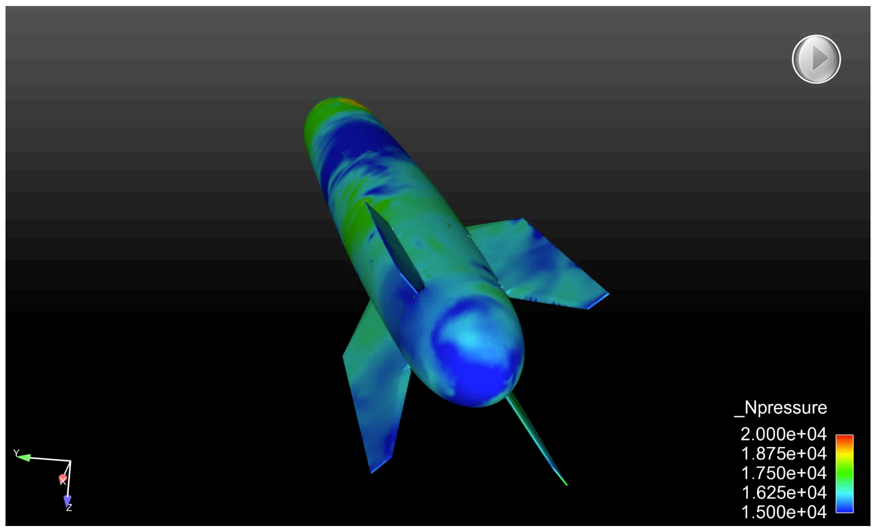
MIT

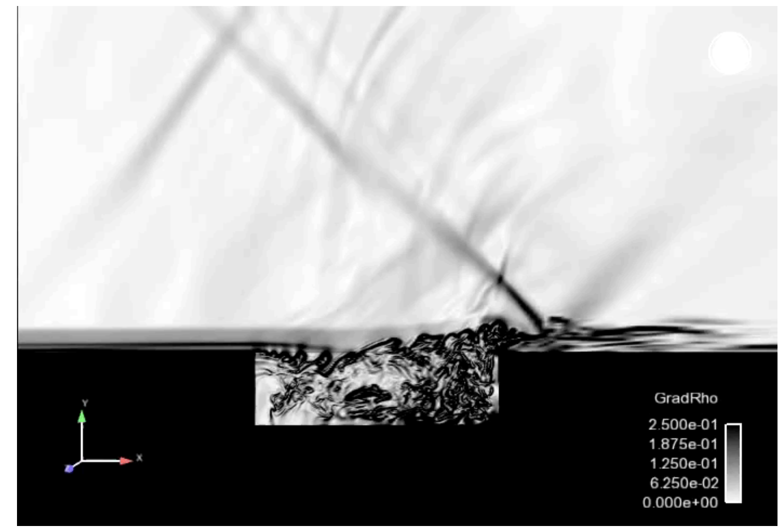
Cambridge, Massachusetts

February 22, 2019

High-fidelity simulation

+ Indispensable in aerospace applications
- Extreme-scale models required for high fidelity





+ *Validated and predictive*: matches wind-tunnel experiments to within 5%
- *Extreme scale*: 100 million cells, 200,000 time steps
- *High simulation costs*: 6 weeks, 5000 cores

computational barrier

Time-critical applications

- rapid design
- uncertainty quantification
- structural health monitoring
- model predictive control

Computational barrier at NASA

The New York Times

Geniuses Wanted: NASA Challenges Coders to Speed Up Its Supercomputer

"Despite tremendous progress made in the past few decades, CFD tools are too slow for simulation of complex geometry flows… [taking] from thousands to millions of computational core-hours."

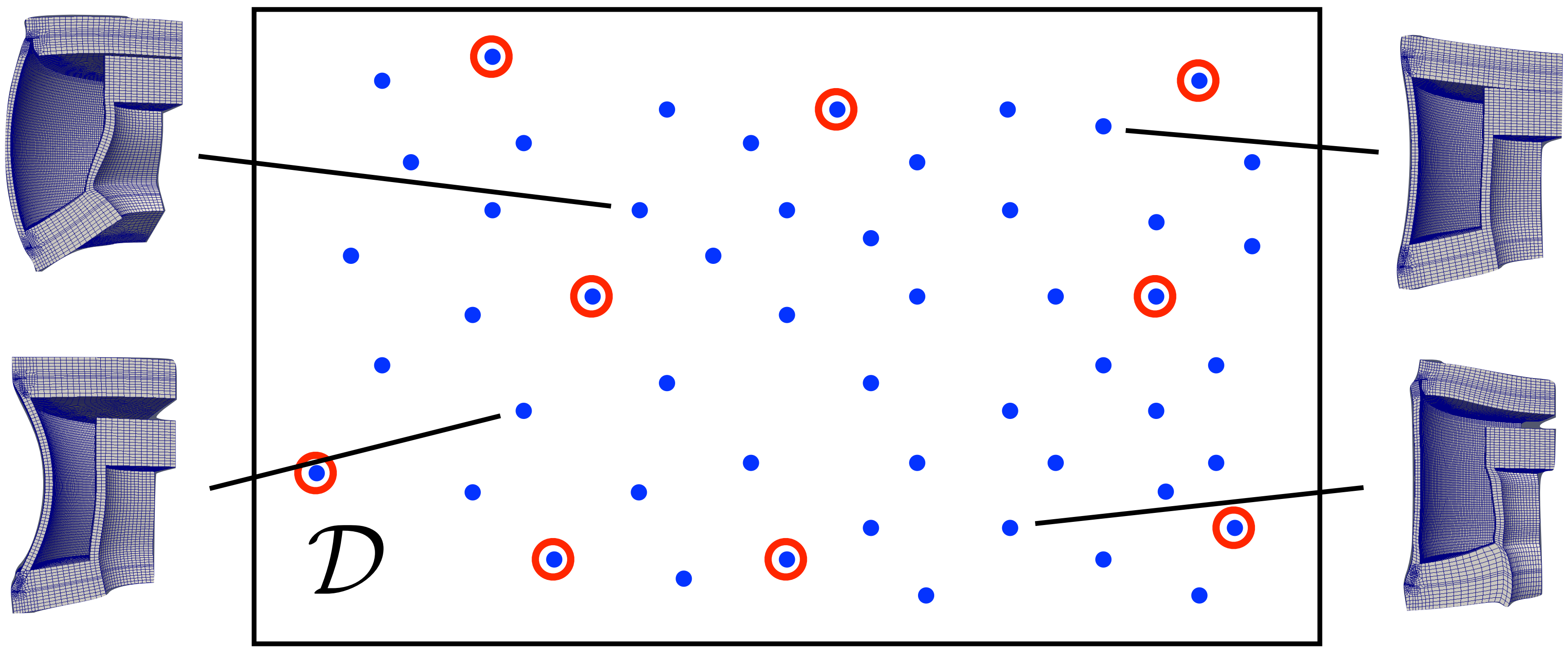
"To enable high-fidelity CFD for multi-disciplinary analysis and design, the speed of computation must be increased by orders of magnitude."

*"The desired outcome is any approach that can **accelerate calculations by a factor of 10x to 1000x**."*

Approach: exploit simulation data

$$\text{ODE:} \quad \frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}; t, \boldsymbol{\mu}), \quad \mathbf{x}(0, \boldsymbol{\mu}) = \mathbf{x}_0(\boldsymbol{\mu}), \quad t \in [0, T_{\text{final}}], \quad \boldsymbol{\mu} \in \mathcal{D}$$

***Time-critical problem**: rapidly solve ODE for* $\boldsymbol{\mu} \in \mathcal{D}_{\text{query}}$



***Idea**: exploit simulation data collected at a few points*

1. *Training:* Solve ODE for $\boldsymbol{\mu} \in \mathcal{D}_{\text{training}}$ and collect simulation data
2. *Machine learning:* Identify structure in data
3. *Reduction:* Reduce cost of ODE solve for $\boldsymbol{\mu} \in \mathcal{D}_{\text{query}} \setminus \mathcal{D}_{\text{training}}$

Model reduction criteria

1. ***Accuracy:*** achieves less than 1% error
2. ***Low cost:*** achieves at least 100x computational-cost savings
3. ***Structure preservation:*** preserves intrinsic physical properties
4. ***Robustness:*** guaranteed satisfaction of any accuracy requirement
5. ***Certification:*** accurately quantify the ROM error

Model reduction: existing approaches

Linear time-invariant systems: mature [Antoulas, 2005]

- Balanced truncation [Moore, 1981; Willcox and Peraire, 2002; Rowley, 2005]
- Transfer-function interpolation [Bai, 2002; Freund, 2003; Gallivan et al, 2004; Baur et al., 2001]

+ *Accurate*, *reliable*, *certified*: sharp *a priori* error bounds
+ *Inexpensive*: pre-assemble operators
+ *Structure preservation*: guaranteed stability

Elliptic/parabolic PDEs: mature [Prud'Homme et al., 2001; Barrault et al., 2004; Rozza et al., 2008]

- Reduced-basis method

+ *Accurate*, *reliable*, *certified*: sharp *a priori* error bounds, convergence
+ *Inexpensive*: pre-assemble operators
+ *Structure preservation*: preserve operator properties

Nonlinear dynamical systems: ineffective

- Proper orthogonal decomposition (POD)–Galerkin [Sirovich, 1987]

– *Inaccurate, unreliable*: often unstable
– *Not certified*: error bounds grow exponentially in time
– *Expensive*: projection insufficient for speedup
– *Structure not preserved*: physical properties ignored

Our research

Accurate, low-cost, structure-preserving, reliable, certified nonlinear model reduction

- *accuracy*: LSPG projection [C., Bou-Mosleh, Farhat, 2011; C., Barone, Antil, 2017]
- *low cost*: sample mesh [C., Farhat, Cortial, Amsallem, 2013]
- *low cost*: reduce temporal complexity
  [C., Ray, van Bloemen Waanders, 2015; C., Brencher, Haasdonk, Barth, 2017; Choi and C., 2019]
- *structure preservation* [C., Tuminaro, Boggs, 2015; Peng and C., 2017; C., Choi, Sargsyan, 2018]
- *robustness*: projection onto nonlinear manifolds [Lee, C., 2018]
- *robustness*: h-adaptivity [C., 2015]
- *certification*: machine learning error models
  [Drohmann and C., 2015; Trehan, C., Durlofsky, 2017; Freno and C., 2019; Pagani, Manzoni, C., 2019]

Our research

Accurate, low-cost, structure-preserving, reliable, certified nonlinear model reduction

- *accuracy*: LSPG projection [C., Bou-Mosleh, Farhat, 2011*; C., Barone, Antil, 2017]
- *low cost*: sample mesh [C., Farhat, Cortial, Amsallem, 2013]
- *low cost*: reduce temporal complexity
  [C., Ray, van Bloemen Waanders, 2015; C., Brencher, Haasdonk, Barth, 2017; Choi and C., 2019]
- *structure preservation* [C., Tuminaro, Boggs, 2015; Peng and C., 2017; C., Choi, Sargsyan, 2018]
- *robustness*: projection onto nonlinear manifolds [Lee, C., 2018]
- *robustness*: *h*-adaptivity [C., 2015]
- *certification*: machine learning error models
  [Drohmann and C., 2015; Trehan, C., Durlofsky, 2017; Freno and C., 2019; Pagani, Manzoni, C., 2019]

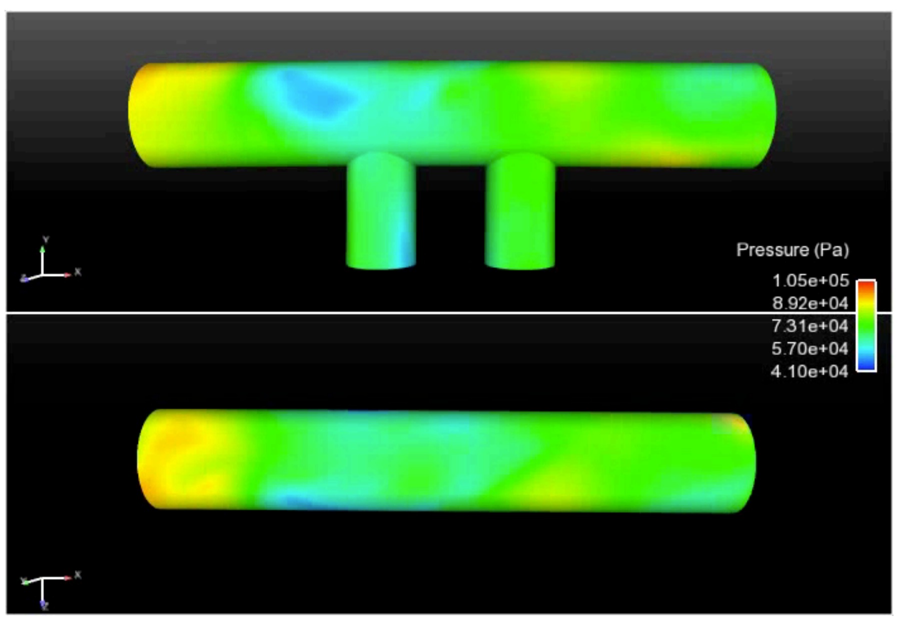
Collaborators: *Matthew Barone (Sandia), Harbir Antil (GMU)*

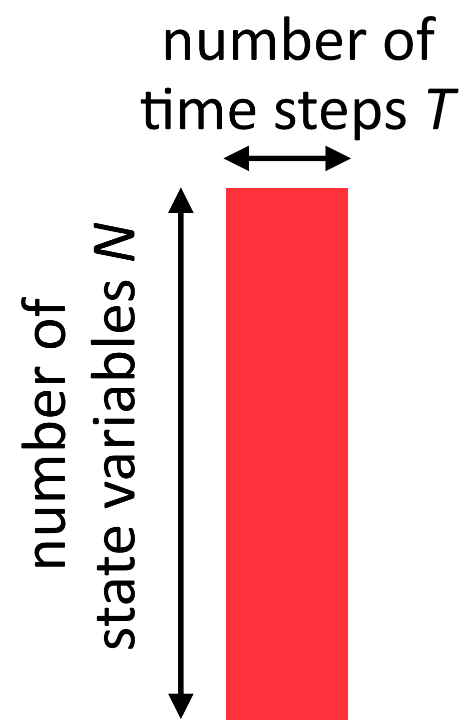
* #2 most-cited paper, Int J Numer Meth Eng, 2011

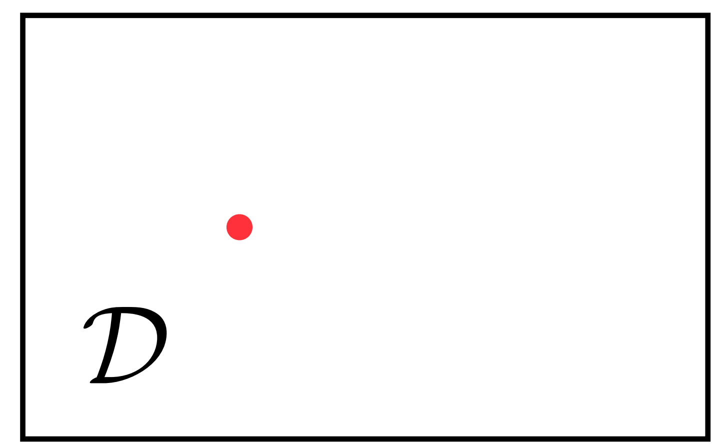
Training simulations: state tensor

$$\text{ODE:} \quad \frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}; t, \boldsymbol{\mu})$$

1. *Training:* Solve ODE for $\boldsymbol{\mu} \in \mathcal{D}_{\text{training}}$ and collect simulation data
2. *Machine learning:* Identify structure in data
3. *Reduction:* Reduce the cost of solving ODE for $\boldsymbol{\mu} \in \mathcal{D}_{\text{query}} \setminus \mathcal{D}_{\text{training}}$



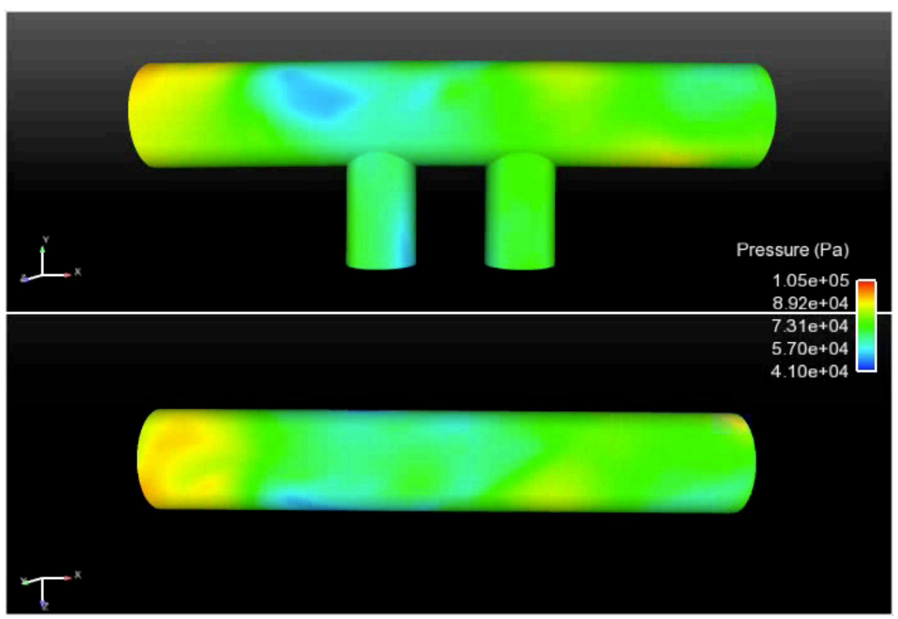


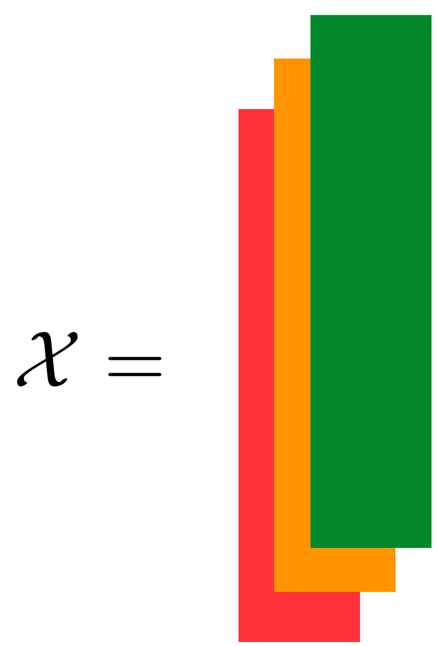


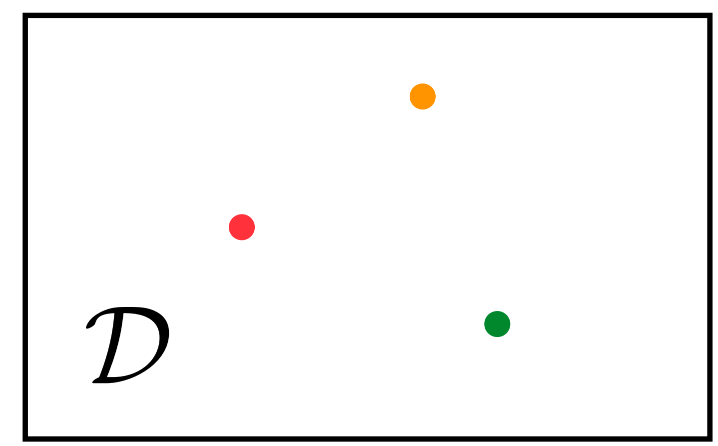
Training simulations: state tensor

$$\text{ODE:} \quad \frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}; t, \boldsymbol{\mu})$$

1. *Training:* Solve ODE for $\boldsymbol{\mu} \in \mathcal{D}_{\text{training}}$ and collect simulation data
2. *Machine learning:* Identify structure in data
3. *Reduction:* Reduce the cost of solving ODE for $\boldsymbol{\mu} \in \mathcal{D}_{\text{query}} \setminus \mathcal{D}_{\text{training}}$





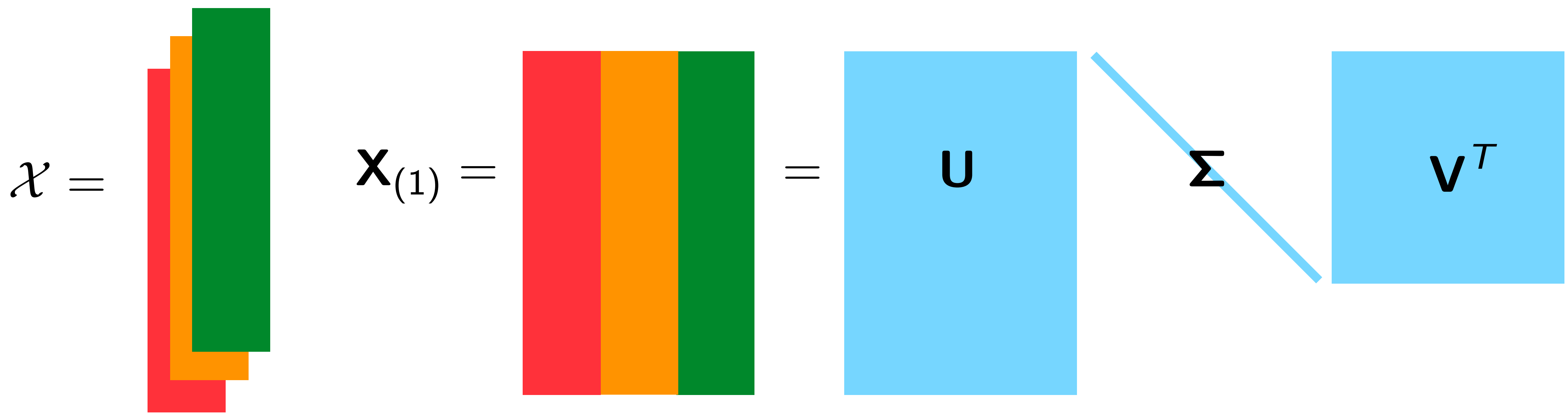


Tensor decomposition

$$\text{ODE:} \quad \frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}; t, \boldsymbol{\mu})$$

1. *Training:* Solve ODE for $\boldsymbol{\mu} \in \mathcal{D}_{\text{training}}$ and collect simulation data
2. *Machine learning:* Identify structure in data
3. *Reduction:* Reduce the cost of solving ODE for $\boldsymbol{\mu} \in \mathcal{D}_{\text{query}} \setminus \mathcal{D}_{\text{training}}$

Compute dominant left singular vectors of mode-1 unfolding

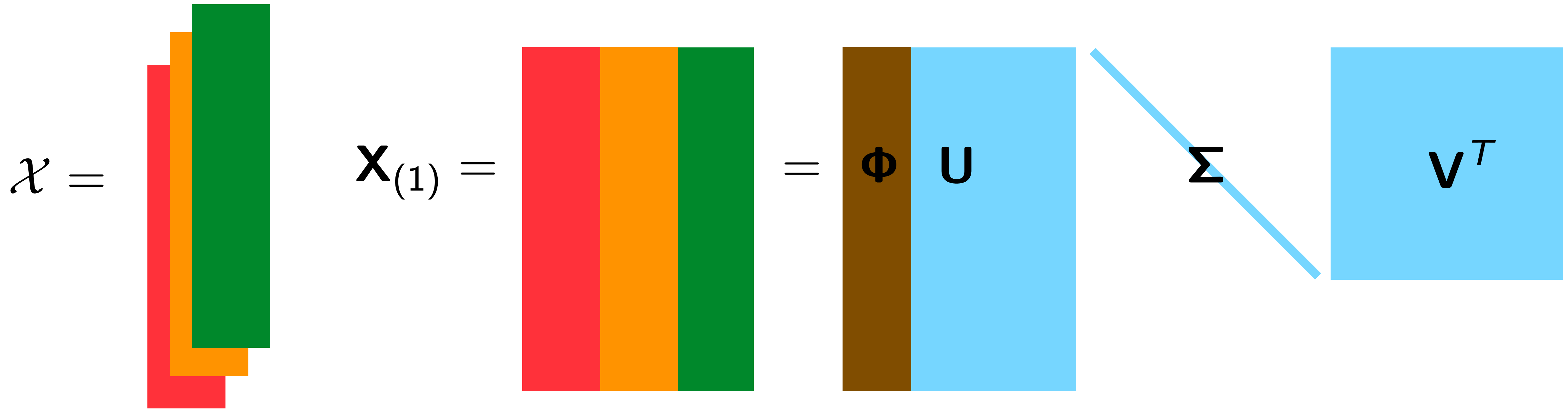


Tensor decomposition

$$\text{ODE:} \quad \frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}; t, \boldsymbol{\mu})$$

1. *Training:* Solve ODE for $\boldsymbol{\mu} \in \mathcal{D}_{\text{training}}$ and collect simulation data
2. *Machine learning:* Identify structure in data
3. *Reduction:* Reduce the cost of solving ODE for $\boldsymbol{\mu} \in \mathcal{D}_{\text{query}} \setminus \mathcal{D}_{\text{training}}$

Compute dominant left singular vectors of mode-1 unfolding

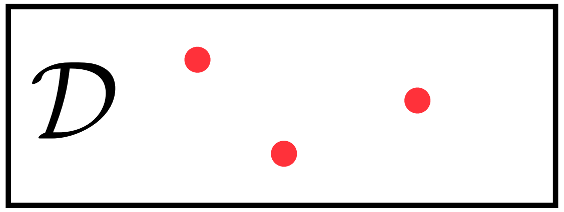


$\boldsymbol{\Phi}$ *columns are principal components of the spatial simulation data*

How to integrate these data with the computational model?

Previous state of the art: POD–Galerkin

ODE: $$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}; t, \boldsymbol{\mu})$$



1. *Training:* Solve ODE for $\mu \in \mathcal{D}_{\text{training}}$ and collect simulation data
2. *Machine learning:* Identify structure in data
3. *Reduction:* Reduce the cost of solving ODE for $\boldsymbol{\mu} \in \mathcal{D}_{\text{query}} \setminus \mathcal{D}_{\text{training}}$

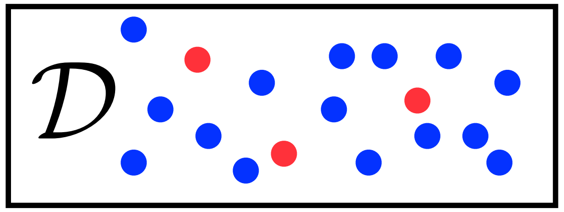
1. Reduce the number of unknowns

$$\mathbf{x}(t) \approx \tilde{\mathbf{x}}(t) = \boldsymbol{\Phi}\,\hat{\mathbf{x}}(t)$$

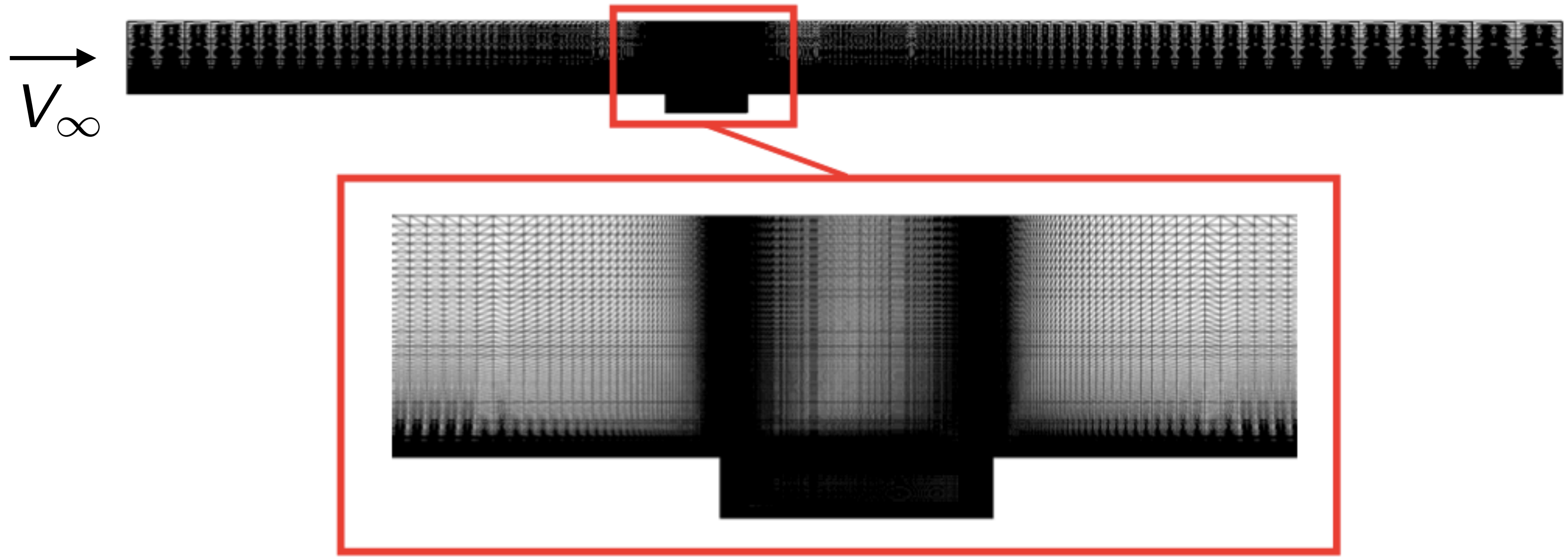
2. Reduce the number of equations

$$\boldsymbol{\Phi}^T\left(\mathbf{f}(\boldsymbol{\Phi}\hat{\mathbf{x}}; t, \boldsymbol{\mu}) - \boldsymbol{\Phi}\frac{d\hat{\mathbf{x}}}{dt}\right) = 0$$

Galerkin ODE: $$\frac{d\hat{\mathbf{x}}}{dt} = \boldsymbol{\Phi}^T\mathbf{f}(\boldsymbol{\Phi}\hat{\mathbf{x}}; t, \boldsymbol{\mu})$$



Captive carry



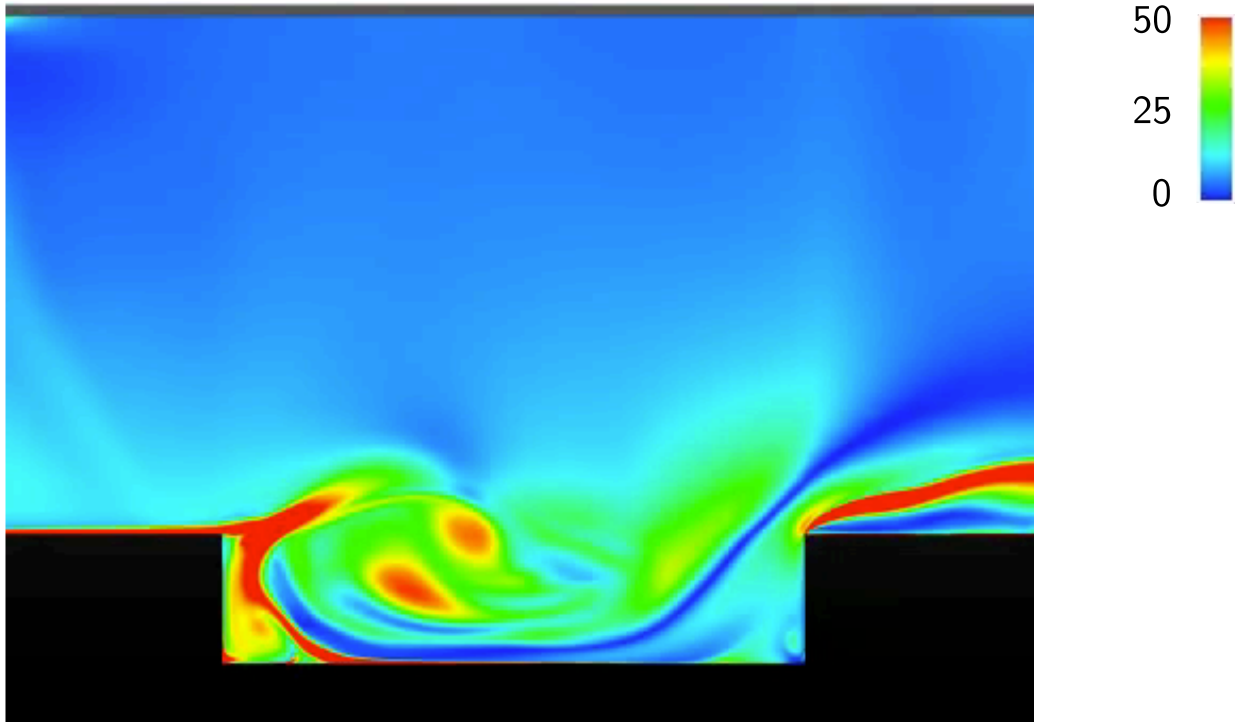
- Unsteady Navier–Stokes
- Re = 6.3 x 10^6
- $M_\infty = 0.6$

Spatial discretization
- 2nd-order finite volume
- DES turbulence model
- 1.2×10^6 degrees of freedom

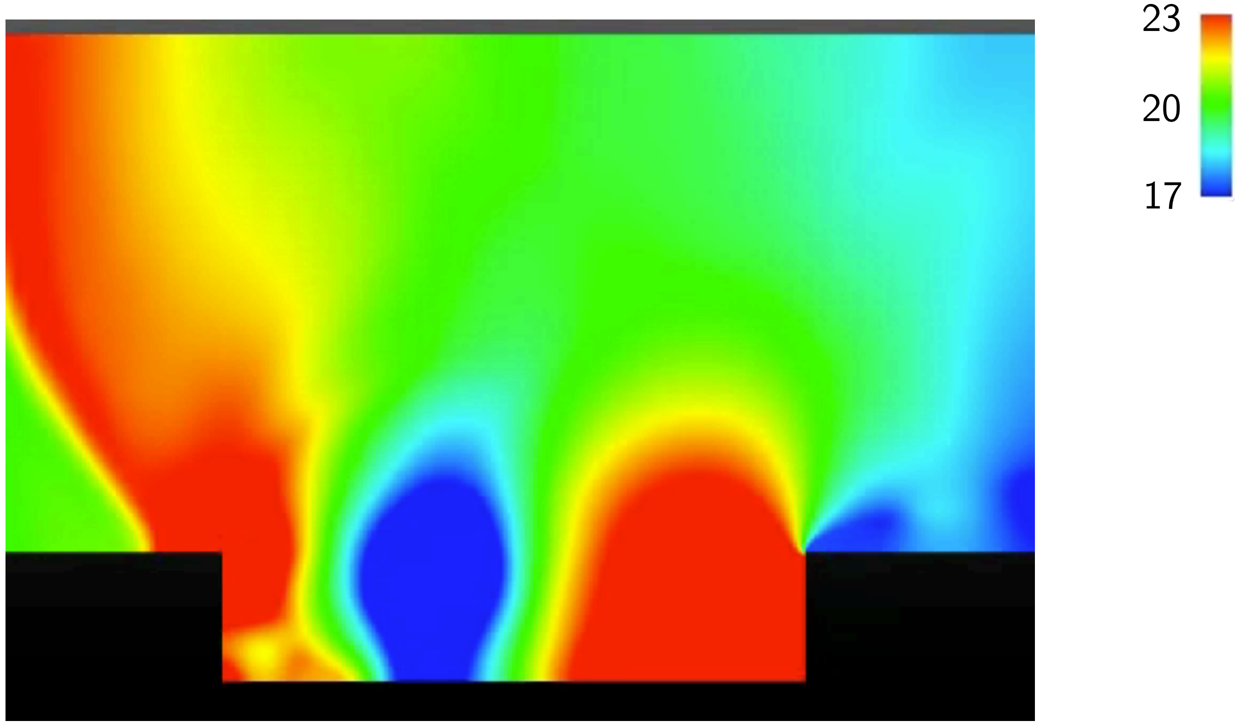
Temporal discretization
- 2nd-order BDF
- Verified time step $\Delta t = 1.5 \times 10^{-3}$
- 8.3×10^3 time instances

High-fidelity model solution

vorticity field

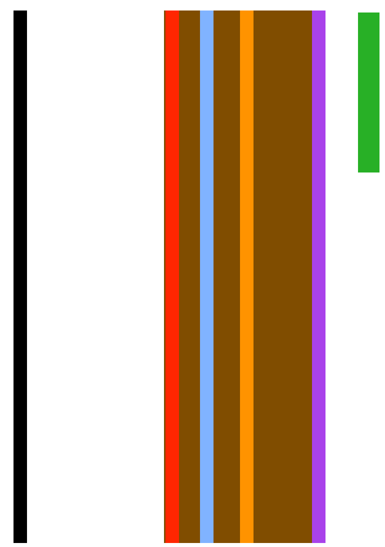


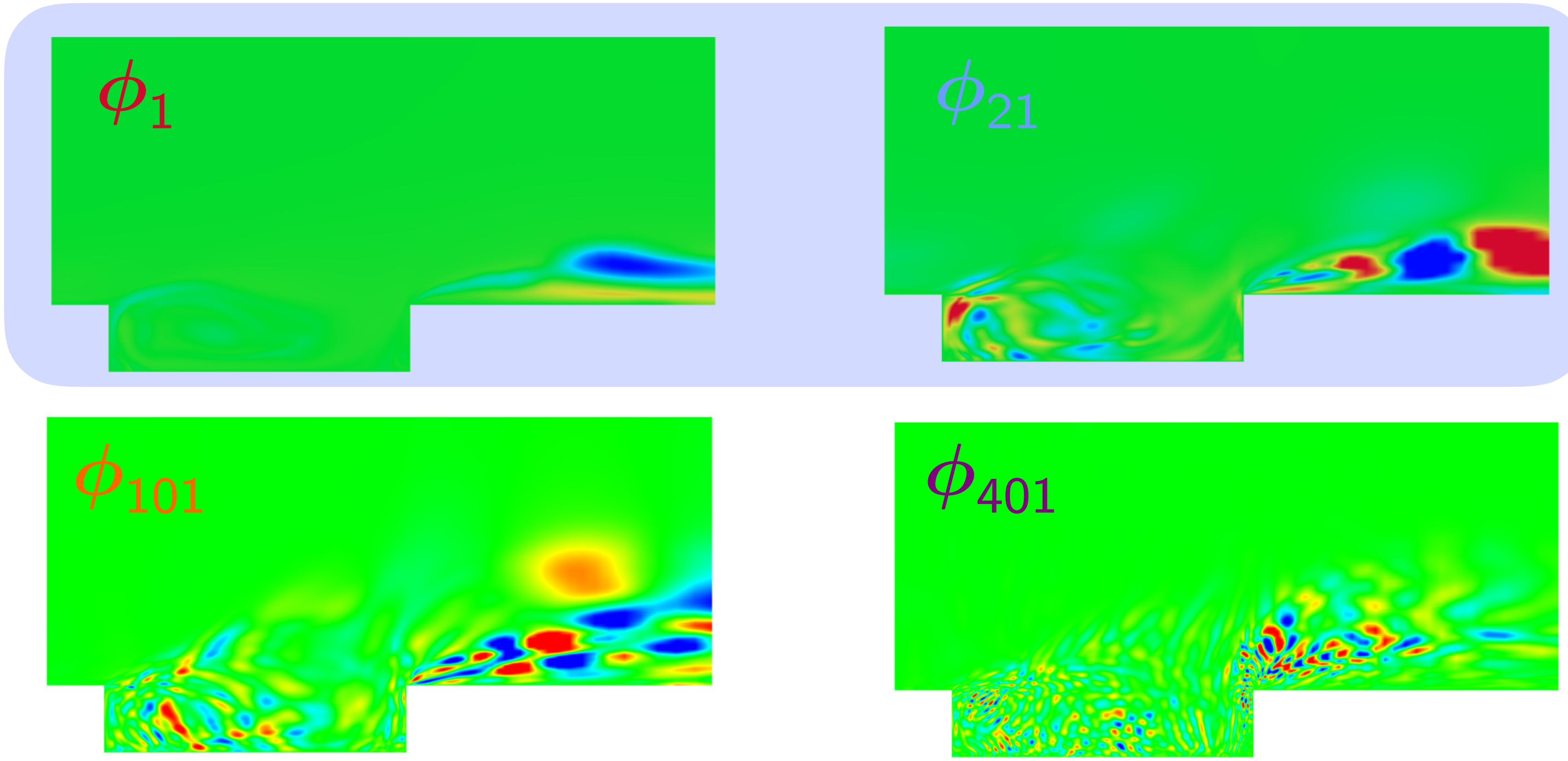
pressure field

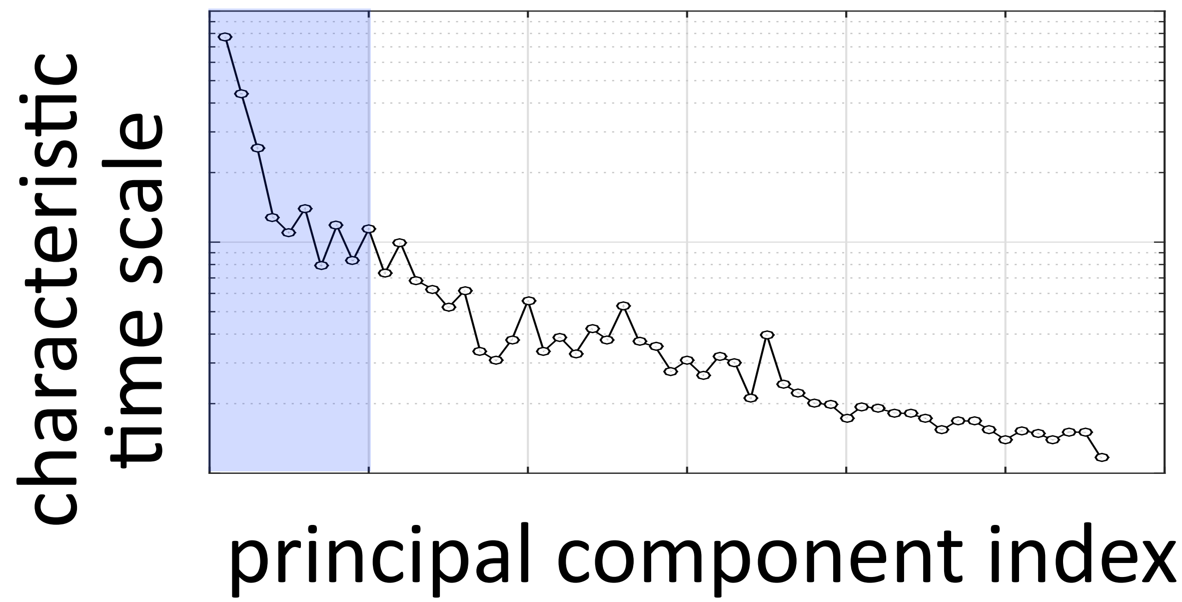


Principal components

$$\mathbf{x}(t) \approx \boldsymbol{\Phi}\ \hat{\mathbf{x}}(t)$$

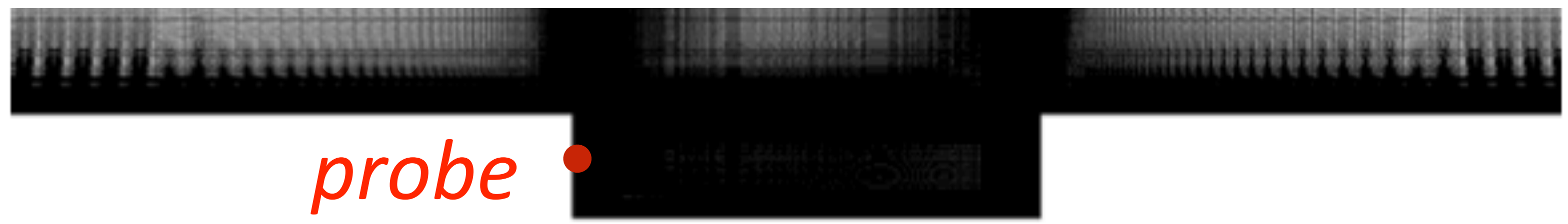


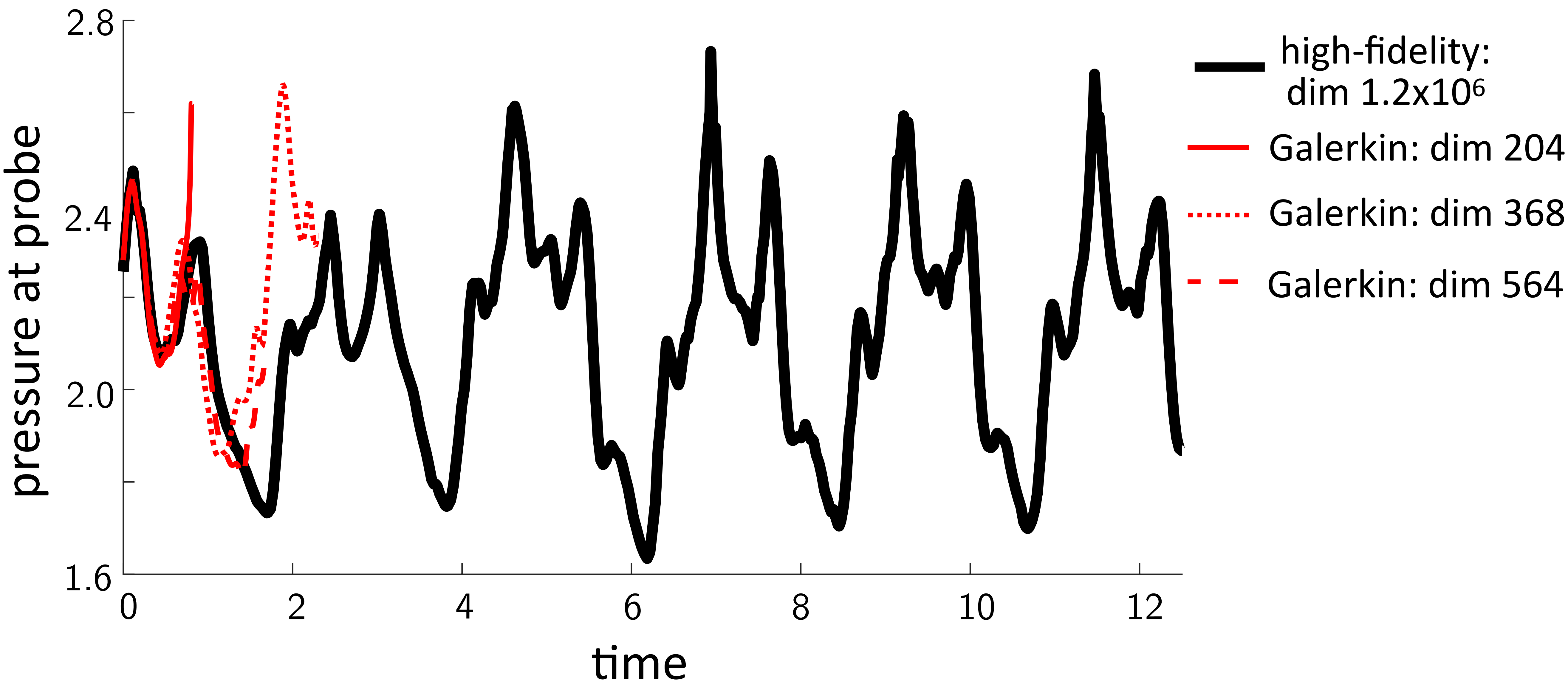




- Truncation preserves coarse spatiotemporal solution components

Galerkin performance





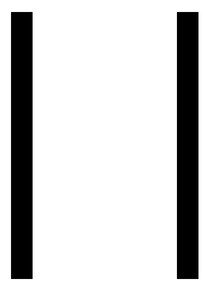
- *Galerkin projection fails regardless of basis dimension*

Can we construct a better projection?

Galerkin: time-continuous optimality

ODE

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}; t)$$



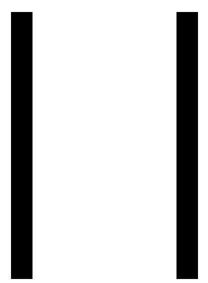
Galerkin ODE

$$\frac{d\hat{\mathbf{x}}}{dt} = \mathbf{\Phi}^T \mathbf{f}(\mathbf{\Phi}\hat{\mathbf{x}}; t)$$

Galerkin: time-continuous optimality

ODE

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}; t)$$



Galerkin ODE

$$\boldsymbol{\Phi} \frac{d\hat{\mathbf{x}}}{dt} = \boldsymbol{\Phi} \; \boldsymbol{\Phi}^T \mathbf{f}(\boldsymbol{\Phi}\hat{\mathbf{x}}; t)$$

+ *Galerkin ODE solution*: optimal in the minimum-residual sense:

$$\boldsymbol{\Phi}\frac{d\hat{\mathbf{x}}}{dt}(\mathbf{x}, t) = \underset{\mathbf{v}\in\text{range}(\boldsymbol{\Phi})}{\text{argmin}} \; \|\mathbf{r}(\mathbf{v}, \mathbf{x}; t)\|_2$$

$$\mathbf{r}(\mathbf{v}, \mathbf{x}; t) := \mathbf{v} - \mathbf{f}(\mathbf{x}; t)$$

OΔE

$$\mathbf{r}^n(\mathbf{x}^n) = 0, \;\; n = 1, \ldots, T$$

Galerkin OΔE

$$\boldsymbol{\Phi}^T\mathbf{r}^n(\boldsymbol{\Phi}\hat{\mathbf{x}}^n) = 0, \;\; n = 1, \ldots, T$$

$$\mathbf{r}^n(\mathbf{x}) := \alpha_0\mathbf{x} - \Delta t\beta_0\mathbf{f}(\mathbf{x}; t^n) + \sum_{j=1}^{k}\alpha_j\mathbf{x}^{n-j} - \Delta t\sum_{j=1}^{k}\beta_j\mathbf{f}(\mathbf{x}^{n-j}; t^{n-j})$$

- *Galerkin OΔE solution*: not generally optimal in any sense

Residual minimization and time discretization

ODE

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}; t)$$

residual minimization →

Galerkin ODE

$$\boldsymbol{\Phi}\frac{d\hat{\mathbf{x}}}{dt}(\mathbf{x}, t) = \underset{\mathbf{v}\in \text{range}(\boldsymbol{\Phi})}{\text{argmin}} \|\mathbf{r}(\mathbf{v}, \mathbf{x}; t)\|_2$$

time discretization ↓ (from ODE)

OΔE

$$\mathbf{r}^n(\mathbf{x}^n) = 0$$

$$n = 1, \ldots, T$$

← *residual minimization*

LSPG OΔE

$$\boldsymbol{\Phi}\hat{\mathbf{x}}^n = \underset{\mathbf{v}\in \text{range}(\boldsymbol{\Phi})}{\text{argmin}} \|\mathbf{r}^n(\mathbf{v})\|_2$$

$$n = 1, \ldots, T$$

time discretization ↓ (from Galerkin ODE)

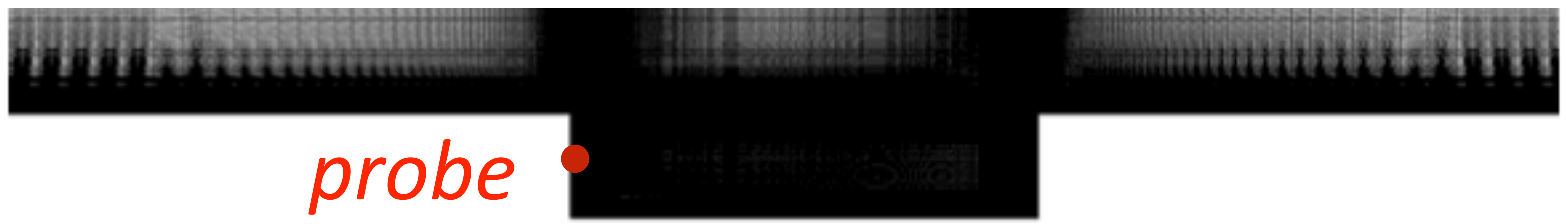
Galerkin OΔE

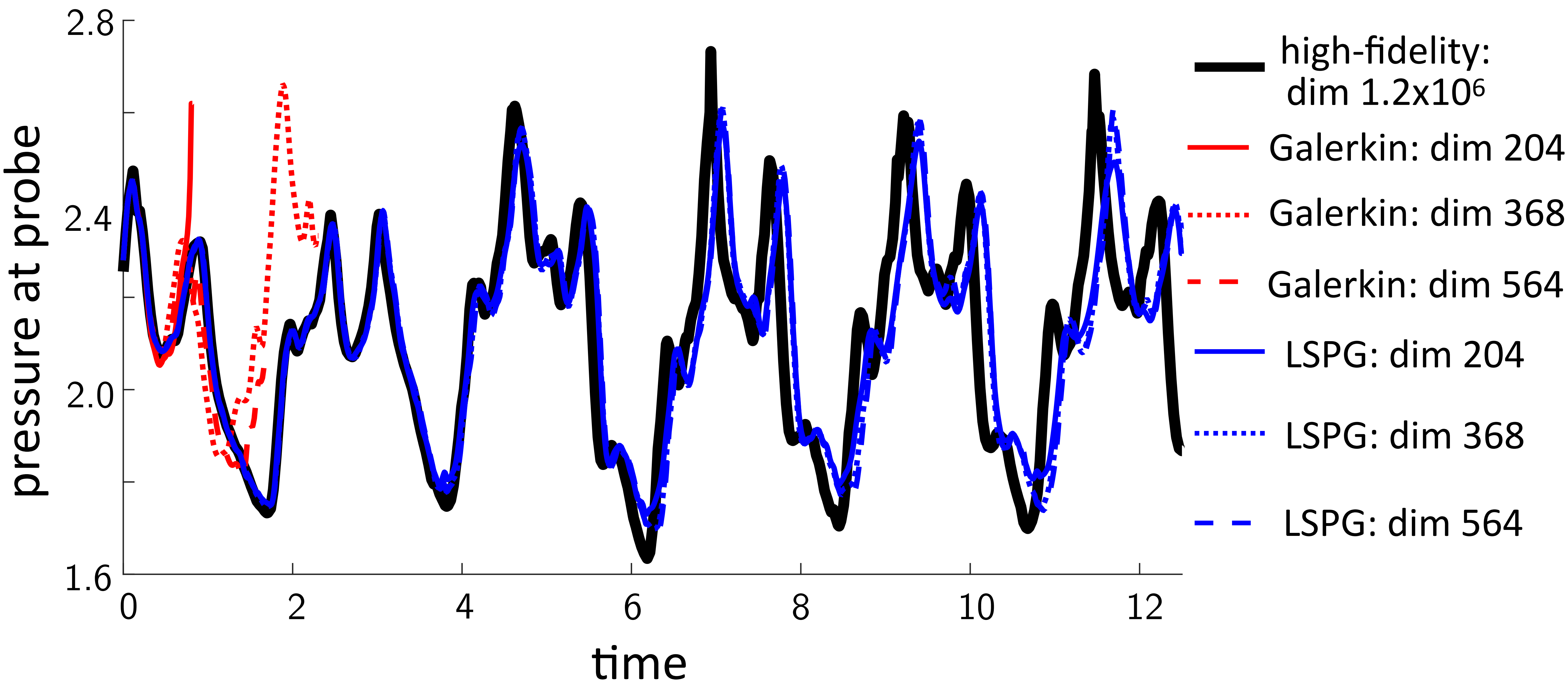
$$\boldsymbol{\Phi}^T\mathbf{r}^n(\boldsymbol{\Phi}\hat{\mathbf{x}}^n) = 0$$

$$n = 1, \ldots, T$$

Least-squares Petrov–Galerkin (LSPG) projection [C., Bou-Mosleh, Farhat, 2011]

LSPG performance





+ LSPG is far more accurate than Galerkin

Why?

Error bound

Theorem: error bound for BDF integrators [C., Barone, Antil, 2017]

If the following conditions hold:

1. $\mathbf{f}(\cdot; t)$ is Lipschitz continuous with Lipschitz constant κ
2. Δt is small enough such that $0 < h := |\alpha_0| - |\beta_0|\kappa\Delta t$, then

$$\|\mathbf{x}^n - \boldsymbol{\Phi}\hat{\mathbf{x}}_{\mathrm{G}}^n\|_2 \leq \frac{1}{h}\|\mathbf{r}_{\mathrm{G}}^n(\boldsymbol{\Phi}\hat{\mathbf{x}}_{\mathrm{G}}^n)\|_2 + \frac{1}{h}\sum_{\ell=1}^{k}|\alpha_\ell|\|\mathbf{x}^{n-\ell} - \boldsymbol{\Phi}\hat{\mathbf{x}}_{\mathrm{G}}^{n-\ell}\|_2$$

$$\|\mathbf{x}^n - \boldsymbol{\Phi}\hat{\mathbf{x}}_{\mathrm{LSPG}}^n\|_2 \leq \frac{1}{h}\min_{\hat{\mathbf{v}}}\|\mathbf{r}_{\mathrm{LSPG}}^n(\boldsymbol{\Phi}\hat{\mathbf{v}})\|_2 + \frac{1}{h}\sum_{\ell=1}^{k}|\alpha_\ell|\|\mathbf{x}^{n-\ell} - \boldsymbol{\Phi}\hat{\mathbf{x}}_{\mathrm{LSPG}}^{n-\ell}\|_2$$

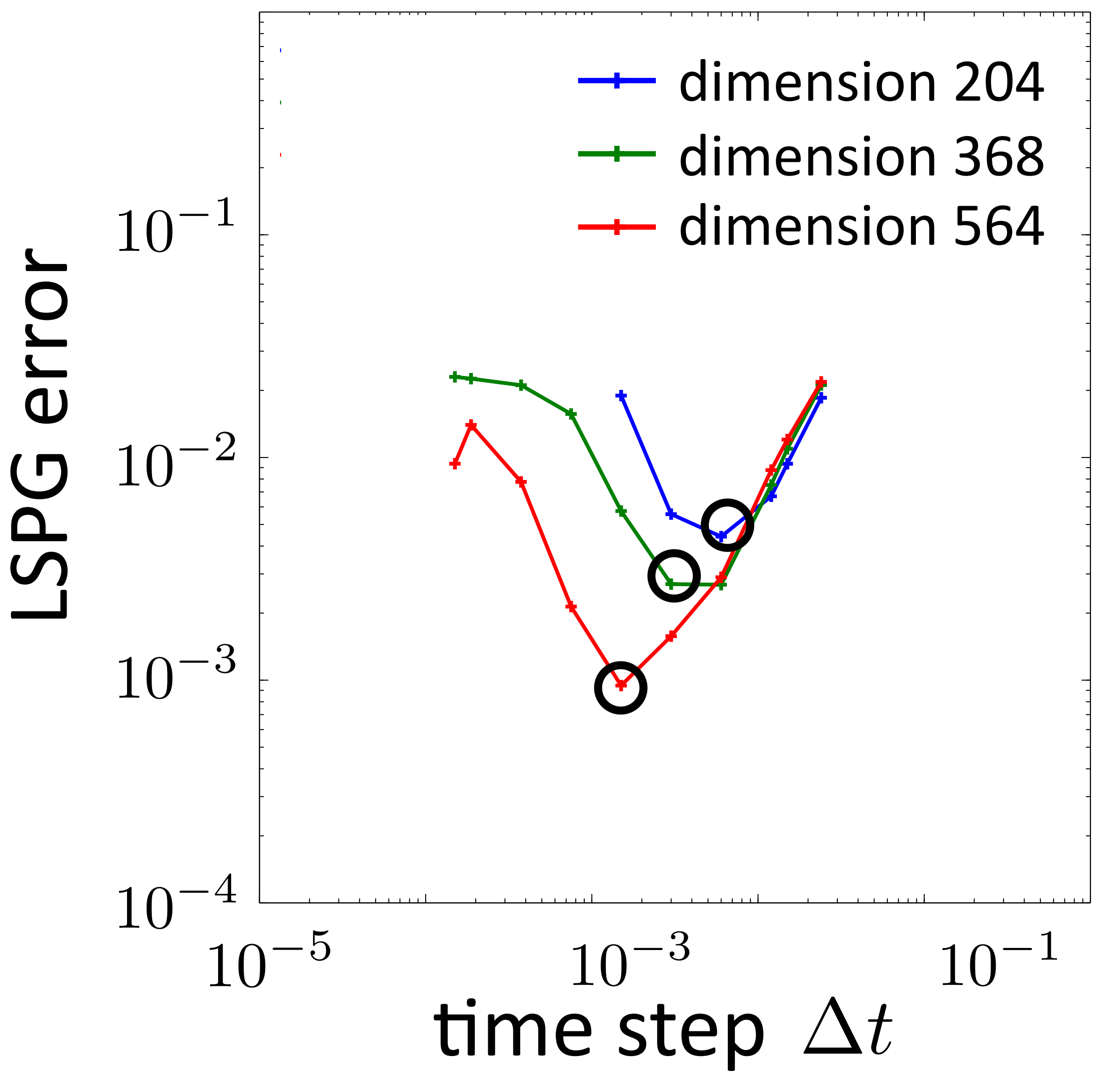
+ LSPG sequentially minimizes the error bound

$$\|\mathbf{r}_{\mathrm{LSPG}}^n(\boldsymbol{\Phi}\hat{\mathbf{v}})\|_2 = |\alpha_0|\|\underbrace{\boldsymbol{\Phi}(\hat{\mathbf{v}} - \hat{\mathbf{x}}_{\mathrm{LSPG}}^{n-1})}_{\text{approx increment}} - \underbrace{(\bar{\mathbf{x}}^n - \boldsymbol{\Phi}\hat{\mathbf{x}}_{\mathrm{LSPG}}^{n-1})}_{\text{increment}} - \frac{\Delta t\beta_0}{\alpha_0}(\mathbf{f}(\boldsymbol{\Phi}\hat{\mathbf{v}}; t^n) - \mathbf{f}(\bar{\mathbf{x}}^n; t^n))\|_2$$

Ensuring $\boldsymbol{\Phi}$ captures solution increments over Δt reduces LSPG error bound

LSPG dependence on time step

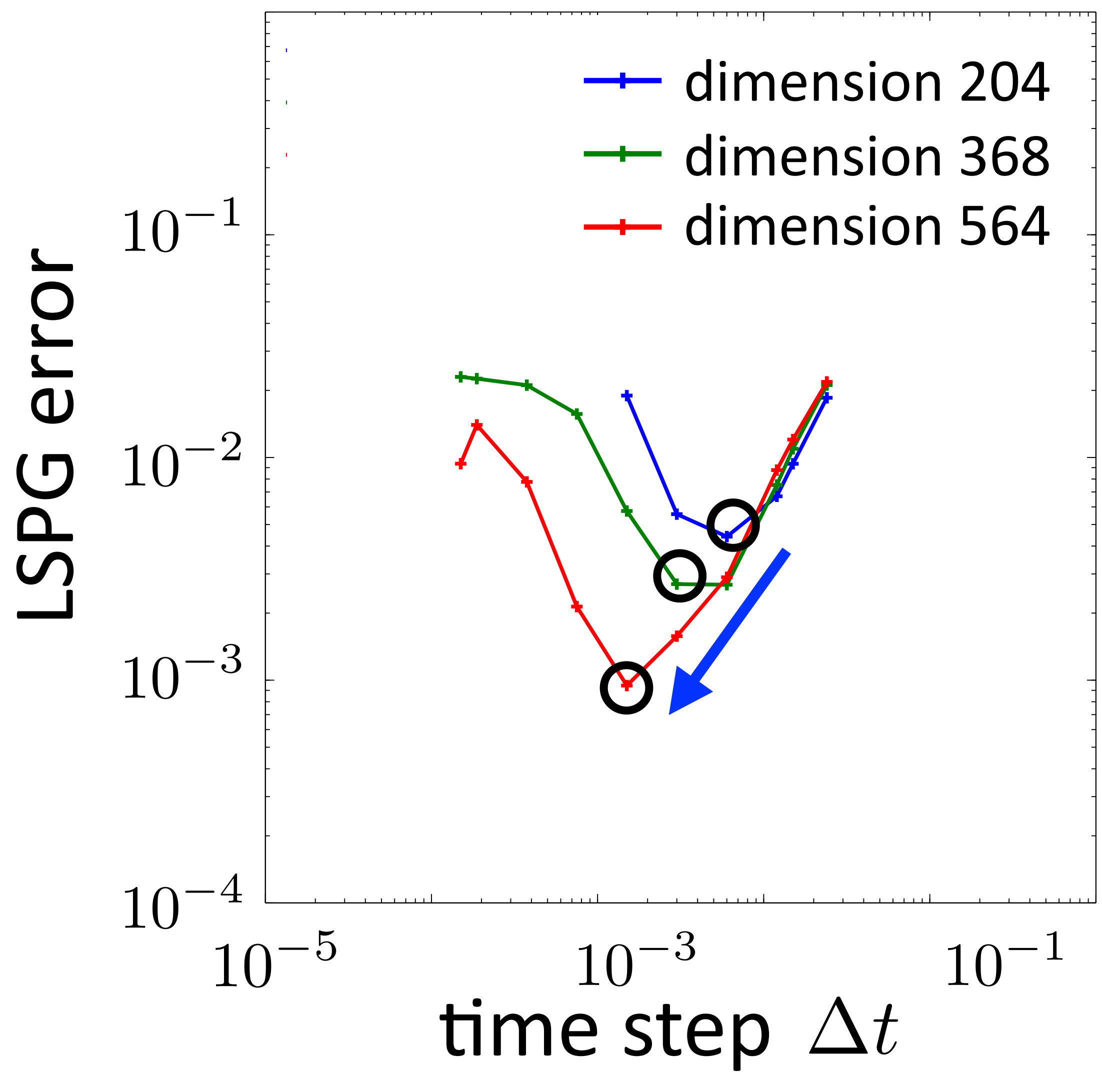
- Shrinking Δt has two competing effects:
  + *time-discretization error*: smaller
  - *error bound:* more difficult for $\mathbf{\Phi}$ to resolve solution increments



- *Best LSPG accuracy*: intermediate Δt balances these two effects

LSPG dependence on time step

- Shrinking Δt has two competing effects:
  + *time-discretization error*: smaller
  - *error bound:* more difficult for $\mathbf{\Phi}$ to resolve solution increments

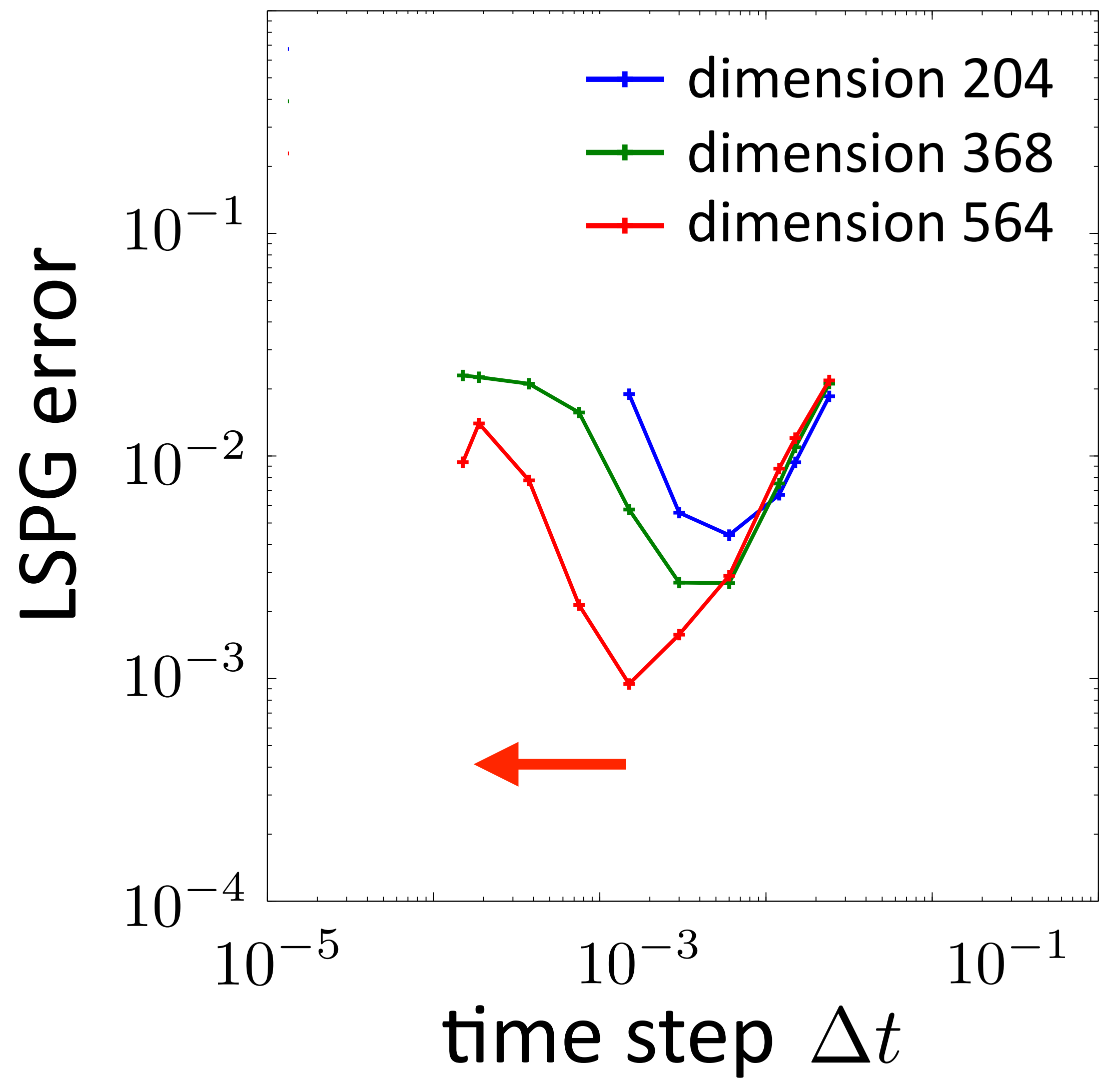


- *Best LSPG accuracy*: intermediate Δt balances these two effects
- *Higher-dimension* $\mathbf{\Phi}$: can capture solution increments over smaller Δt

Limiting equivalence

Theorem: Equivalence [C., Barone, Antil, 2017]

Galerkin and LSPG projection are equivalent in the limit $\Delta t \to 0$.



Explains poor Galerkin accuracy: *equivalent to LSPG as* $\Delta t \to 0$

Our research

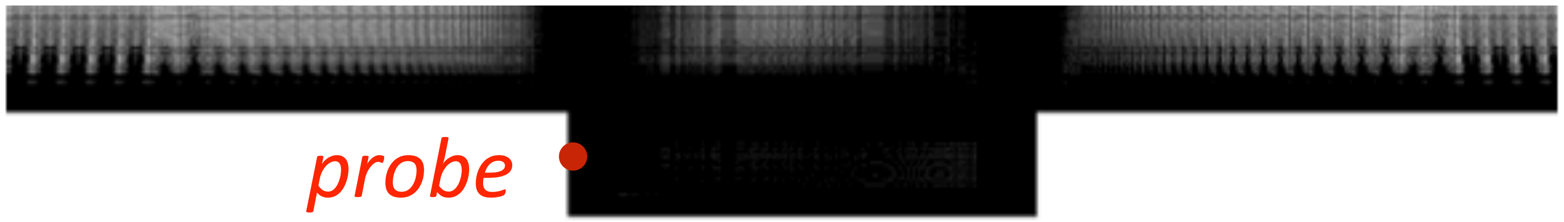
Accurate, low-cost, structure-preserving, reliable, certified nonlinear model reduction

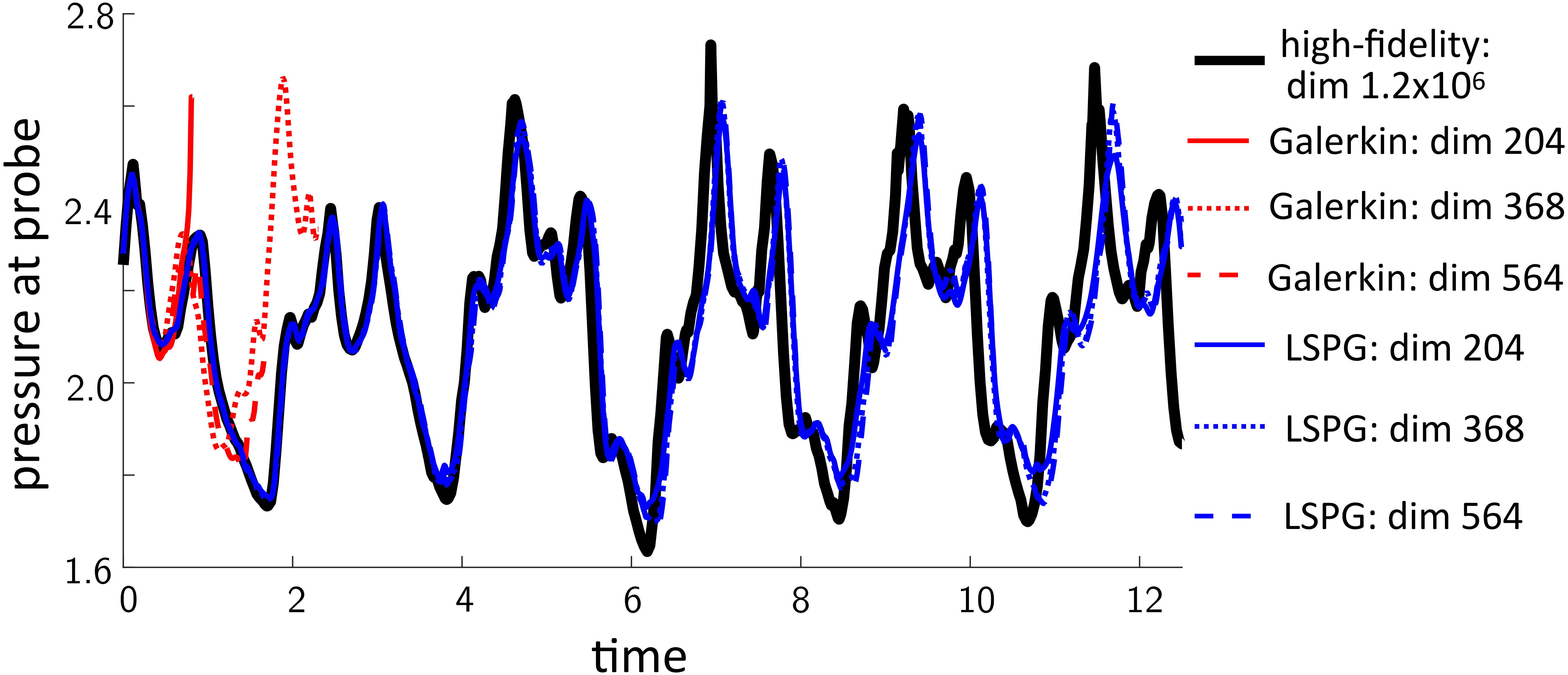
- *accuracy*: LSPG projection [C., Bou-Mosleh, Farhat, 2011; C., Barone, Antil, 2017]
- *low cost*: sample mesh [C., Farhat, Cortial, Amsallem, 2013*]
- *low cost*: reduce temporal complexity [C., Ray, van Bloemen Waanders, 2015; C., Brencher, Haasdonk, Barth, 2017; Choi and C., 2019]
- *structure preservation* [C., Tuminaro, Boggs, 2015; Peng and C., 2017; C., Choi, Sargsyan, 2017]
- *robustness*: projection onto nonlinear manifolds [Lee, C., 2018]
- *robustness*: *h*-adaptivity [C., 2015]
- *certification*: machine learning error models [Drohmann and C., 2015; Trehan, C., Durlofsky, 2017; Freno and C., 2019; Pagani, Manzoni, C., 2019]

Collaborators: *Julien Cortial (Stanford), Charbel Farhat (Stanford)*

* #2 most-cited paper, J Comp Phys, 2013

Wall-time problem





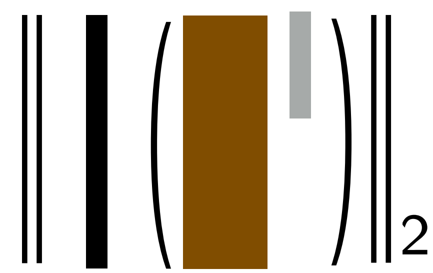
- *High-fidelity simulation*: 1 hour, 48 cores
- *Fastest LSPG simulation*: 1.3 hours, 48 cores

Why does this occur?

Can we fix it?

Cost reduction by gappy PCA [Everson and Sirovich, 1995]

$$\underset{\hat{\mathbf{v}}}{\text{minimize}} \left\| \mathbf{r}^n(\boldsymbol{\Phi}\hat{\mathbf{v}}) \right\|_2$$

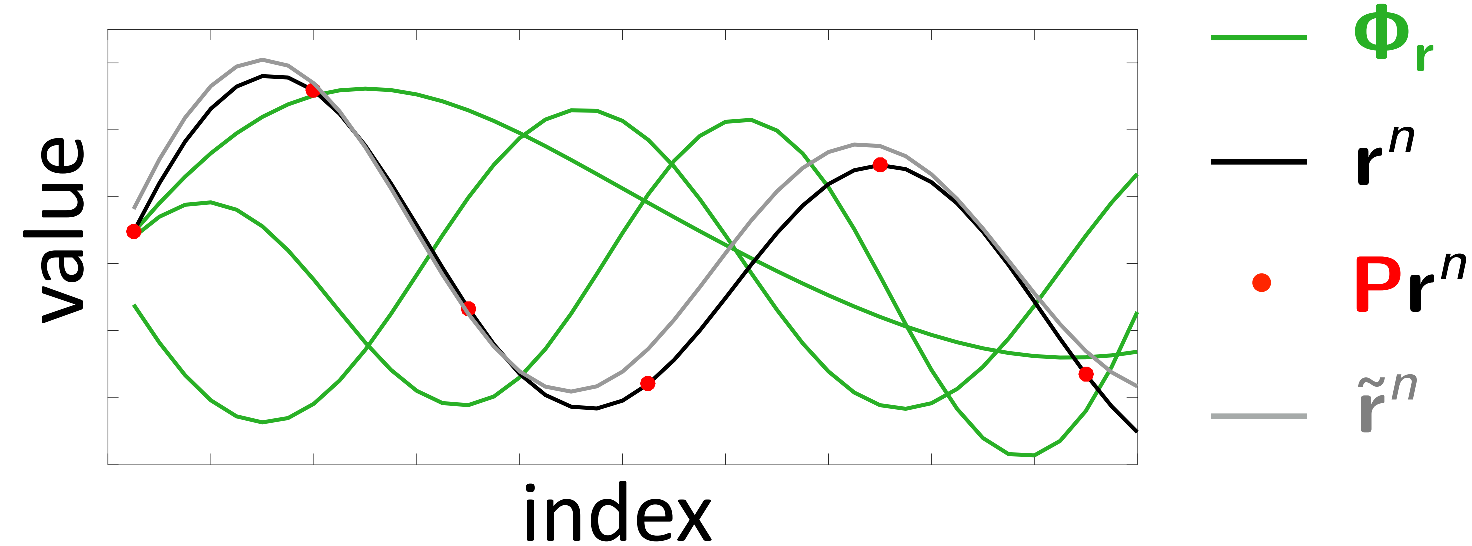


Cost reduction by gappy PCA [Everson and Sirovich, 1995]

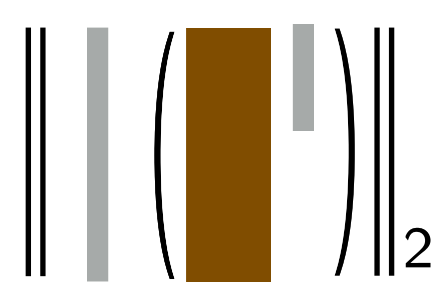
$$\underset{\hat{\mathbf{v}}}{\text{minimize}} \| \mathbf{A} \, \mathbf{r}^n(\boldsymbol{\Phi}\hat{\mathbf{v}}) \|_2$$

Can we introduce a weighting matrix $\mathbf{A}$ *to make this less expensive?*

- **Training:** collect residual tensor $\mathcal{R}^{ijk}$ while solving ODE for $\boldsymbol{\mu} \in \mathcal{D}_{\text{training}}$
- **Machine learning:** compute residual PCA $\boldsymbol{\Phi}_\mathbf{r}$ and sampling matrix $\mathbf{P}$
- **Reduction**: compute regression approximation $\mathbf{r}^n \approx \tilde{\mathbf{r}}^n = \boldsymbol{\Phi}_\mathbf{r}(\mathbf{P}\boldsymbol{\Phi}_\mathbf{r})^+\mathbf{P}\mathbf{r}^n$



$$\underset{\hat{\mathbf{v}}}{\text{minimize}} \| \tilde{\mathbf{r}}^n(\boldsymbol{\Phi}\hat{\mathbf{v}}) \|_2$$

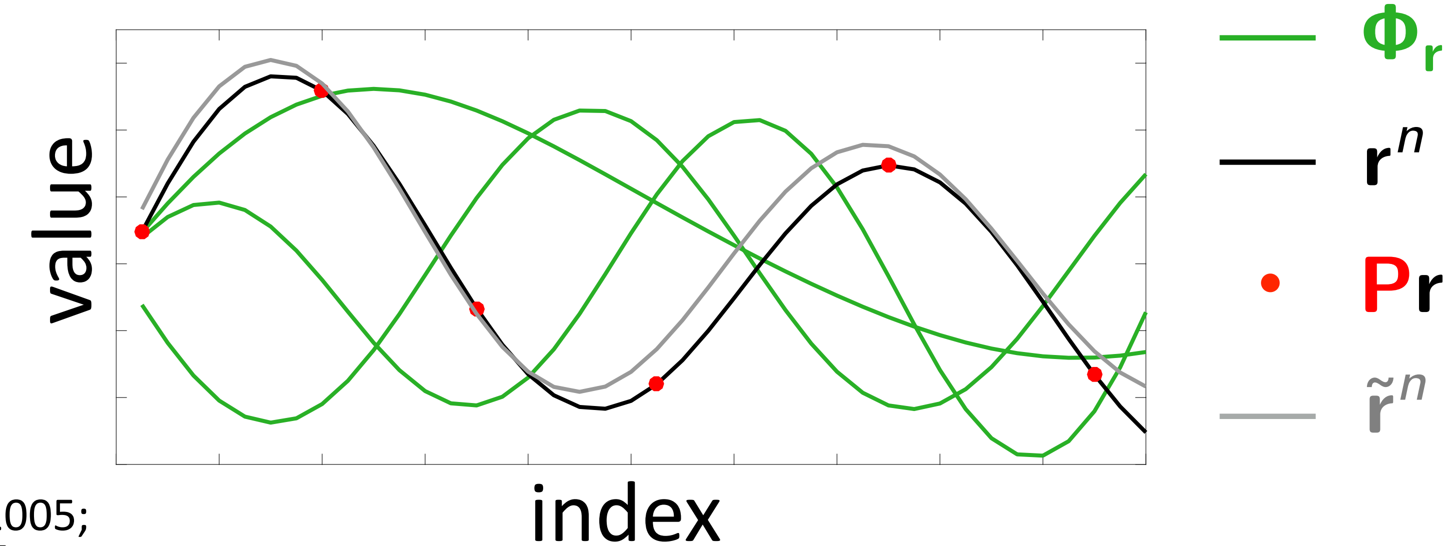


Cost reduction by gappy PCA [Everson and Sirovich, 1995]

$$\underset{\hat{\mathbf{v}}}{\text{minimize}} \| \mathbf{A} \, \mathbf{r}^n(\boldsymbol{\phi} \, \hat{\mathbf{v}}) \|_2$$

Can we introduce a weighting matrix $\mathbf{A}$ *to make this less expensive?*

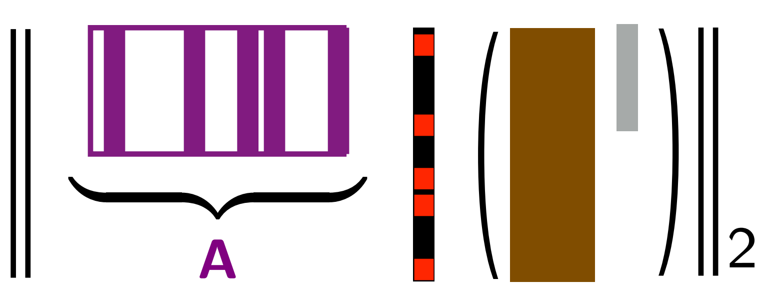
- **Training:** collect residual tensor $\mathcal{R}^{ijk}$ while solving ODE for $\boldsymbol{\mu} \in \mathcal{D}_{\text{training}}$
- **Machine learning:** compute residual PCA $\boldsymbol{\Phi}_{\mathrm{r}}$ and sampling matrix $\mathbf{P}$
- **Reduction**: compute regression approximation $\mathbf{r}^n \approx \tilde{\mathbf{r}}^n = \boldsymbol{\Phi}_{\mathrm{r}}(\mathbf{P}\boldsymbol{\Phi}_{\mathrm{r}})^+\mathbf{P}\mathbf{r}^n$



Related:

- collocation [Ryckelynck, 2005; Legresley, 2006; Astrid et al., 2008]
- empirical interpolation [Barrault et al., 2004; Nguyen, Peraire, 2008; Chaturantabut and Sorensen, 2010]
- FE subassembly [An et al., 2008; Farhat et al., 2014]

$$\underset{\hat{\mathbf{v}}}{\text{minimize}} \| (\mathbf{P}\boldsymbol{\Phi}_{\mathrm{r}})^+\mathbf{P} \, \mathbf{r}^n(\boldsymbol{\phi} \, \hat{\mathbf{v}}) \|_2$$

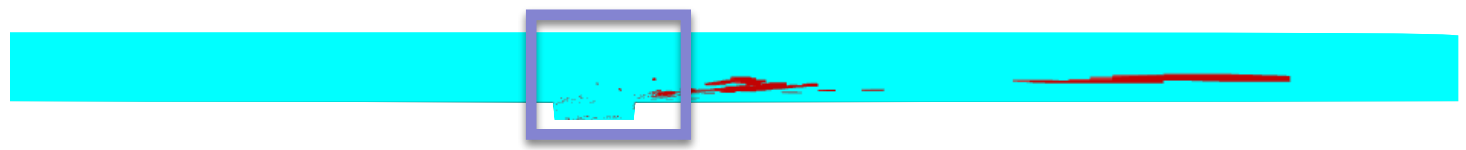


+ *Only a few elements of* $\mathbf{r}^n$ *must be computed*

Sample mesh [C., Farhat, Cortial, Amsallem, 2013]

$$\underset{\hat{\mathbf{v}}}{\text{minimize}} \|(\mathbf{P}\boldsymbol{\Phi}_{\mathbf{r}})^{+}\mathbf{P}\mathbf{r}^{n}(\boldsymbol{\Phi}\hat{\mathbf{v}})\|_2$$

sample mesh



+ *HPC on a laptop*

vorticity field *pressure field*

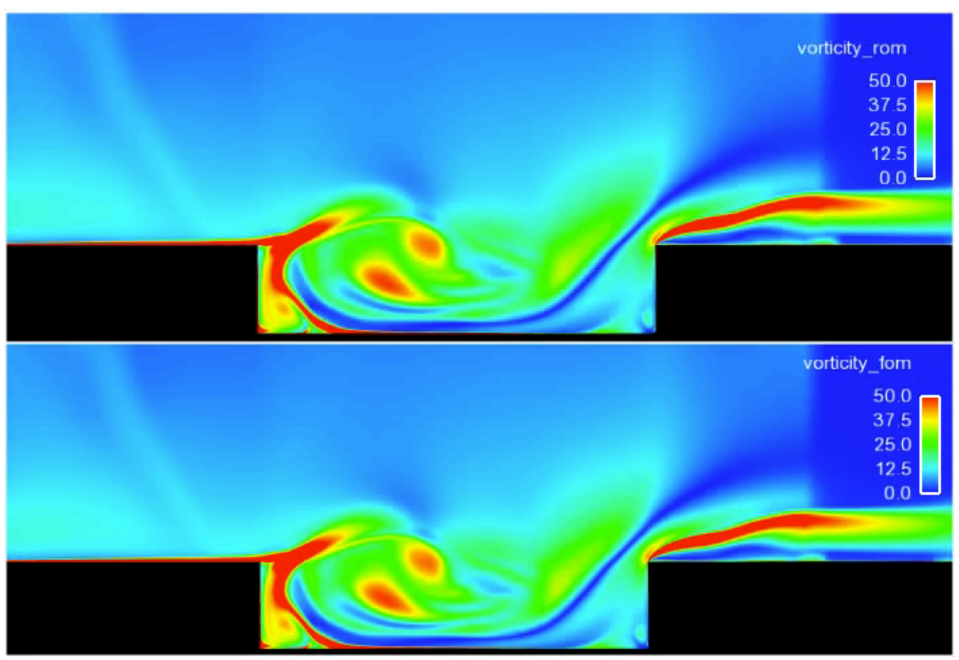
LSPG ROM with

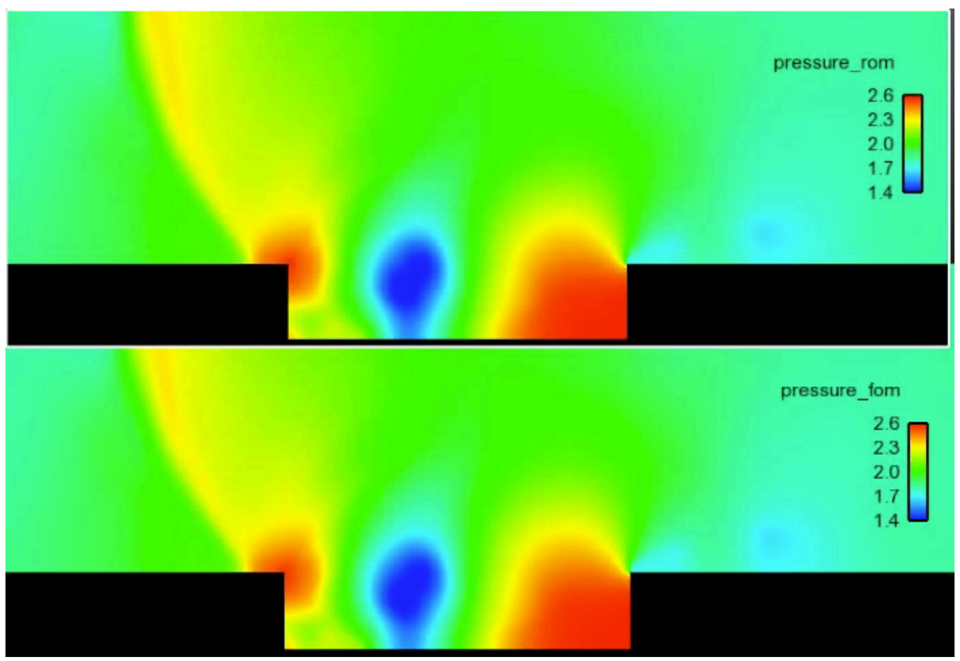
$\mathbf{A} = (\mathbf{P}\boldsymbol{\Phi}_{\mathbf{r}})^{+}\mathbf{P}$

32 min, 2 cores

high-fidelity

5 hours, 48 cores

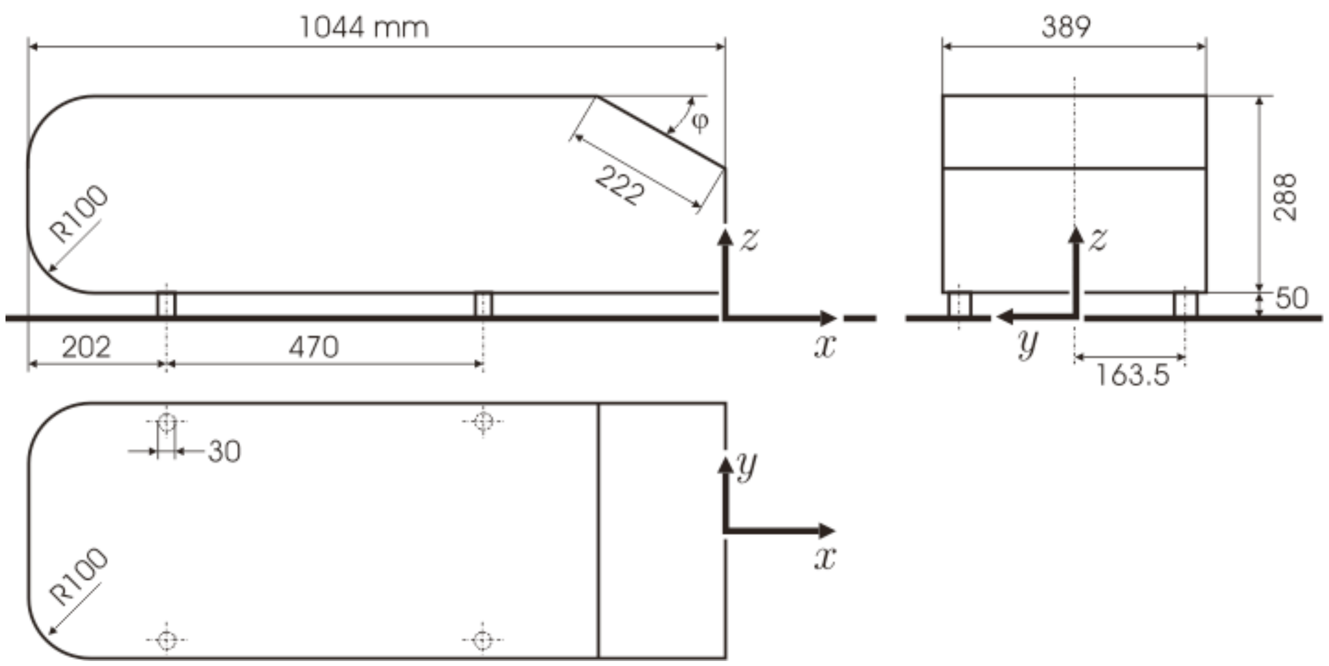


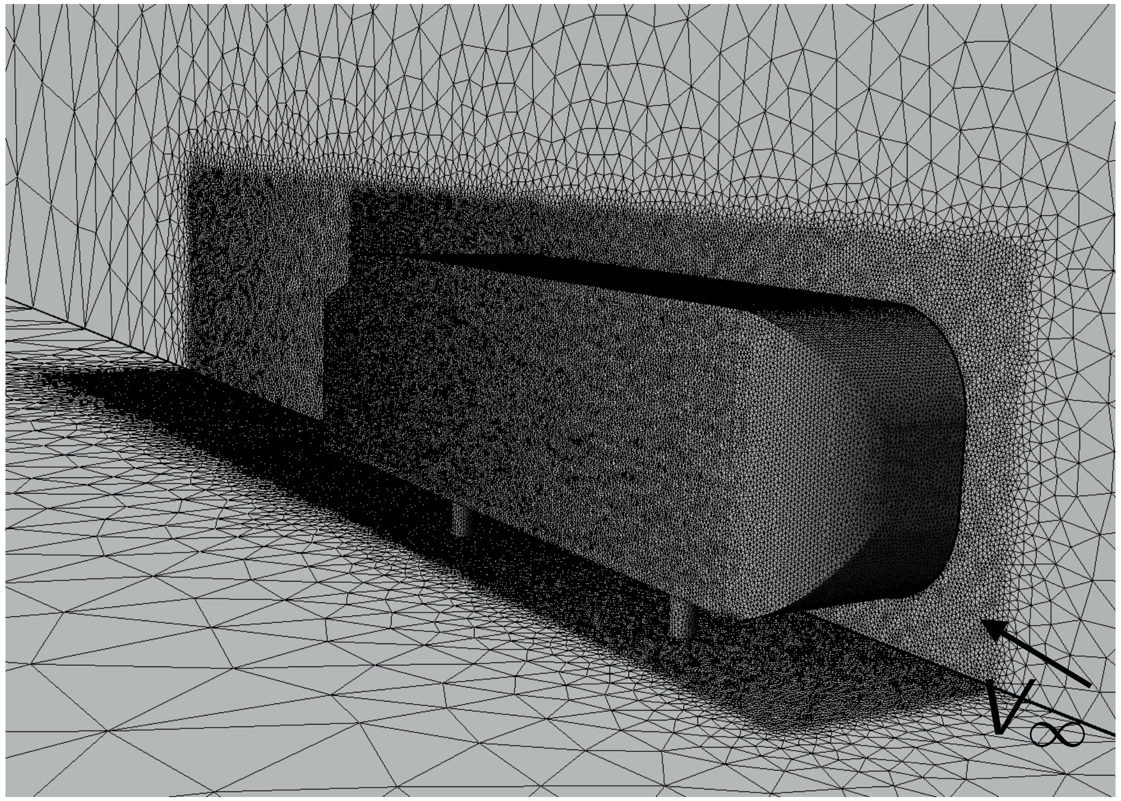


+ *229x savings in core–hours*

+ *< 1% error in time-averaged drag*

Ahmed body [Ahmed, Ramm, Faitin, 1984]





- Unsteady Navier–Stokes
- Re = 4.3 x 10^6
- M_∞ = 0.175

Spatial discretization

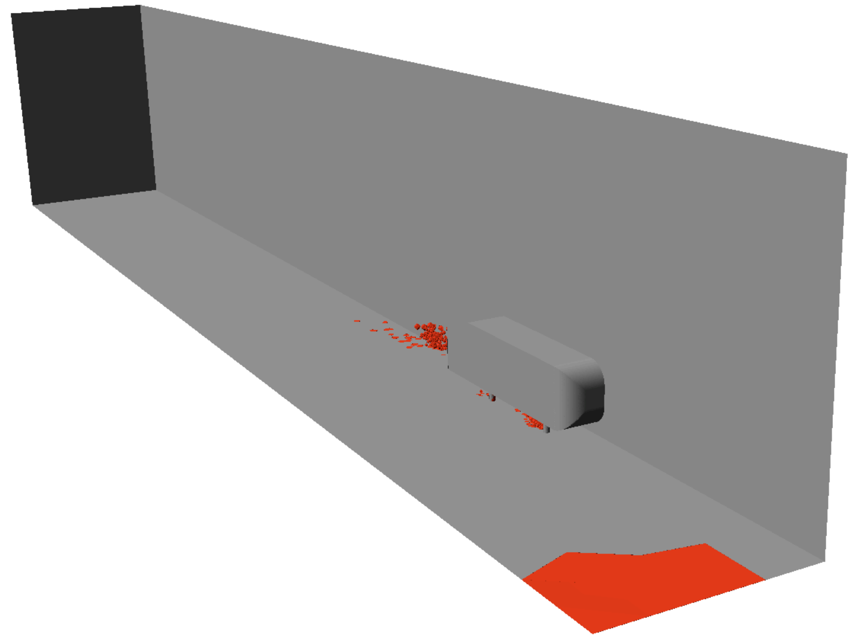
- 2nd-order finite volume
- DES turbulence model
- 1.7×10^7 degrees of freedom

Temporal discretization

- 2nd-order BDF
- Time step $\Delta t = 8 \times 10^{-5}\mathrm{s}$
- 1.3×10^3 time instances

Ahmed body results [C., Farhat, Cortial, Amsallem, 2013]

sample mesh



+ *HPC on a laptop*

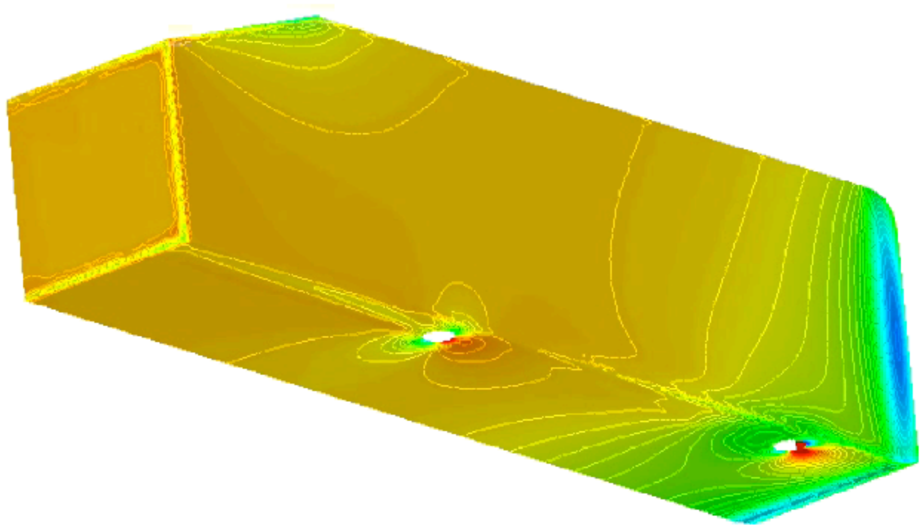
LSPG ROM with $\mathbf{A} = (\mathbf{P}\boldsymbol{\Phi}_{\mathbf{r}})^{+}\mathbf{P}$

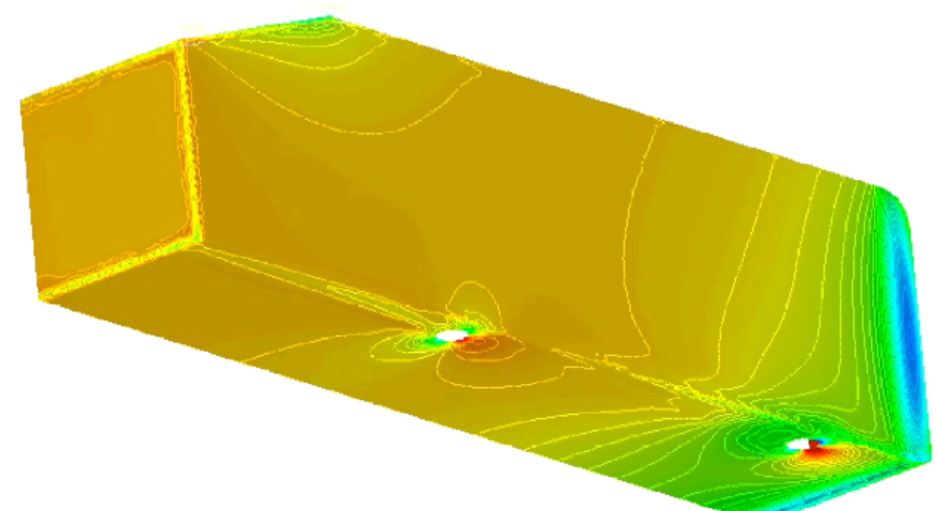
4 hours, 4 cores

high-fidelity model

13 hours, 512 cores

pressure field





+ *438x savings in core–hours*

+ *Largest nonlinear dynamical system on which ROM has ever had success*

Our research

Accurate, low-cost, structure-preserving, reliable, certified nonlinear model reduction

- *accuracy*: LSPG projection [C., Bou-Mosleh, Farhat, 2011; C., Barone, Antil, 2017]
- *low cost*: sample mesh [C., Farhat, Cortial, Amsallem, 2013]
- *low cost*: reduce temporal complexity
  [C., Ray, van Bloemen Waanders, 2015; C., Brencher, Haasdonk, Barth, 2017; Choi and C., 2019]
- *structure preservation* [C., Tuminaro, Boggs, 2015; Peng and C., 2017; C., Choi, Sargsyan, 2018]
- *robustness*: projection onto nonlinear manifolds [Lee, C., 2018]
- *robustness*: *h*-adaptivity [C., 2015]
- *certification*: machine learning error models
  - [Drohmann and C., 2015; Trehan, C., Durlofsky, 2017; Freno and C., 2019; Pagani, Manzoni, C., 2019]

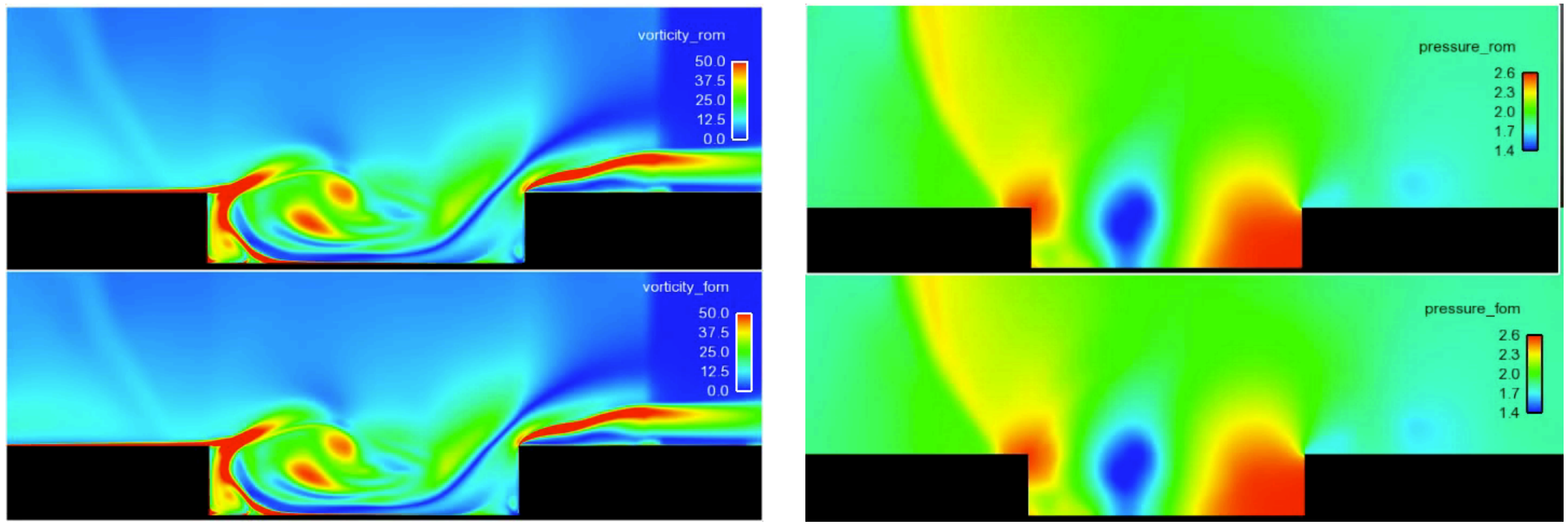
Collaborator: *Youngsoo Choi (Sandia)*

Captive-carry results [C., Barone, Antil, 2017]

vorticity field *pressure field*

GNAT ROM
32 min, 2 cores
spatial dim: 179
temporal dim: 458

high-fidelity
5 hours, 48 cores
spatial dim: 1.2M
temporal dim: 3,700



\+ *229X computational-cost reduction*

\+ *6,500X spatial-dimension reduction*

\- *8X temporal-dimension reduction*

How can we significantly reduce the temporal dimensionality?

Reducing temporal complexity:

Larger time steps with ROM

[Krysl et al., 2001; Lucia et al., 2004; Taylor et al., 2010; C. et al., 2017]

- Developed for explicit and implicit integrators

– Limited reduction of time dimension: <10X reductions typical

Space–time ROMs

- Reduced basis [Urban, Patera, 2012; Yano, 2013; Urban, Patera, 2014; Yano, Patera, Urban, 2014]
- POD–Galerkin [Volkwein, Weiland, 2006; Baumann, Benner, Heiland, 2016]
- ODE-residual minimization [Constantine, Wang, 2012]

\+ Reduction of time dimension

\+ Linear time-growth of error bounds ˆ

– Requires space–time finite element discretization ˆ

– No hyper-reduction

– Only one space–time basis vector per training simulation

ˆ Only reduced-basis methods

Goals

Preserve attractive properties of existing space–time ROMs

+ Reduce both space and time dimensions
+ Slow time-growth of error bound

Overcome shortcomings of existing space–time ROMs

+ Applicability to general nonlinear dynamical systems
+ Hyper-reduction
+ Extract multiple space–time basis vectors from each training simulation

Space–time least-squares Petrov–Galerkin (ST-LSPG) projection [Choi and C., 2019]

Spatial v. spatiotemporal trial

Full-order-model trial subspace

$$\begin{bmatrix}\mathbf{x}^1 & \cdots & \mathbf{x}^T\end{bmatrix} \in \mathbb{R}^N \otimes \mathbb{R}^T$$

Spatial trial subspace

$$\begin{bmatrix}\tilde{\mathbf{x}}^1 & \cdots & \tilde{\mathbf{x}}^T\end{bmatrix} = \boldsymbol{\Phi}\begin{bmatrix}\hat{\mathbf{x}}^1 & \cdots & \hat{\mathbf{x}}^T\end{bmatrix} \in \mathcal{S} \otimes \mathbb{R}^T \subseteq \mathbb{R}^N \otimes \mathbb{R}^T$$

\+ Spatial dimension reduced
\- Temporal dimension large

Space–time trial subspace

$$\begin{bmatrix}\tilde{\mathbf{x}}^1 & \cdots & \tilde{\mathbf{x}}^T\end{bmatrix} = \sum_{i=1}^{n_{st}} \boldsymbol{\pi}_i \hat{x}_i(\boldsymbol{\mu}) \in \mathcal{S}\mathcal{T} \subseteq \mathbb{R}^N \otimes \mathbb{R}^T$$

\+ Spatial dimension reduced
\+ Temporal dimension reduced
\- Additional approximation

Space–time LSPG projection

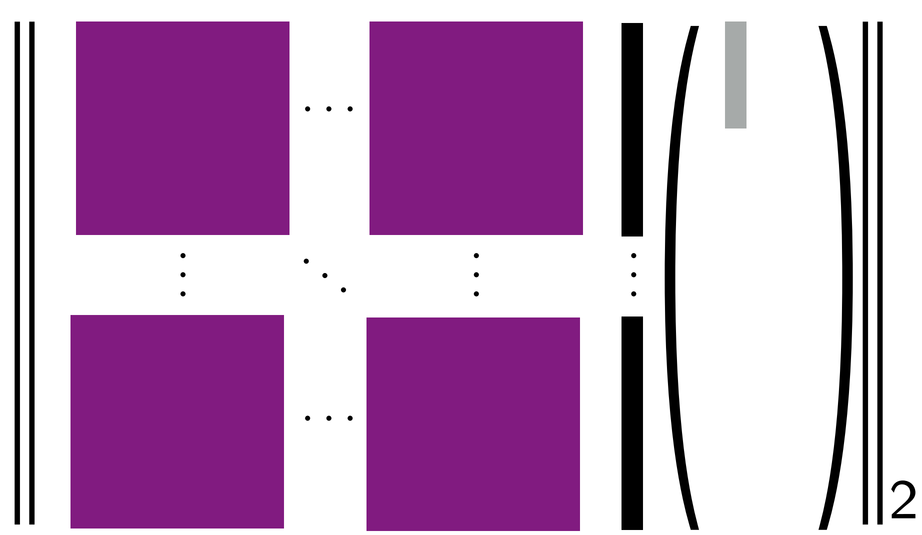
LSPG

$$\underset{\hat{\mathbf{v}}}{\text{minimize}} \left\| \mathbf{A} \, \mathbf{r}^n(\boldsymbol{\phi}\hat{\mathbf{v}}, \tilde{\mathbf{x}}^{n-1}, \ldots, \tilde{\mathbf{x}}^{n-k}; \boldsymbol{\mu}) \right\|_2, \quad n = 1, \ldots, T$$

ST-LSPG

$$\bar{\mathbf{r}}(\hat{\mathbf{v}}; \boldsymbol{\mu}) := \begin{bmatrix} \mathbf{r}^1 \left(\sum_{i=1}^{n_{st}} \boldsymbol{\pi}_i(t^1)\hat{v}_i, \sum_{i=1}^{n_{st}} \boldsymbol{\pi}_i(t^0)\hat{v}_i; \boldsymbol{\mu}\right) \\ \vdots \\ \mathbf{r}^T \left(\sum_{i=1}^{n_{st}} \boldsymbol{\pi}_i(t^T)\hat{v}_i, \sum_{i=1}^{n_{st}} \boldsymbol{\pi}_i(t^{T-1})\hat{v}_i, \ldots, \sum_{i=1}^{n_{st}} \boldsymbol{\pi}_i(t^{T-k})\hat{v}_i; \boldsymbol{\mu}\right) \end{bmatrix}$$

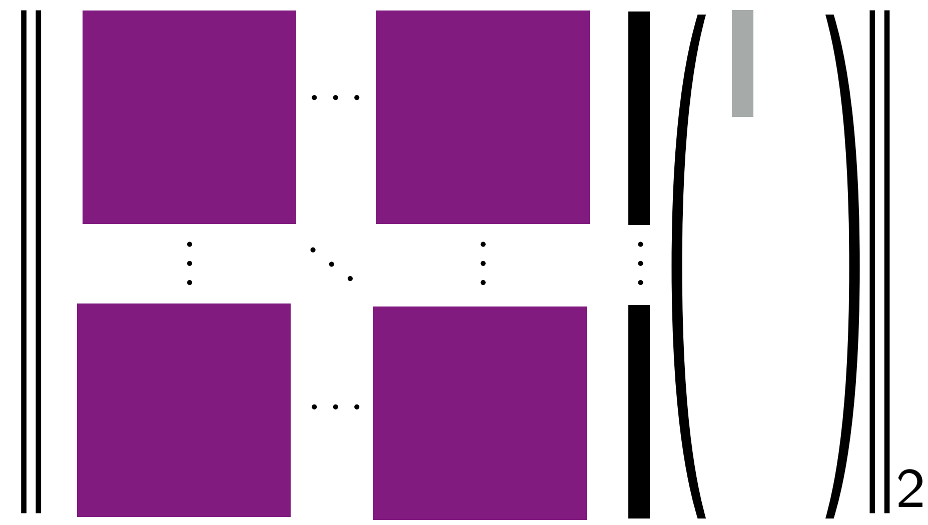
$$\underset{\hat{\mathbf{v}}}{\text{minimize}} \left\| \bar{\mathbf{A}} \, \bar{\mathbf{r}}(\hat{\mathbf{v}}; \boldsymbol{\mu}) \right\|_2$$



+ applicable to general nonlinear dynamical systems
- prohibitive cost: minimizing residual over all space and time

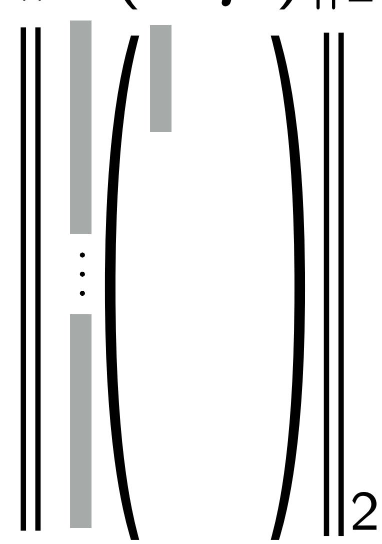
ST-LSPG hyper-reduction

$$\underset{\hat{\mathbf{v}}}{\text{minimize}} \; \| \bar{\mathbf{A}} \, \bar{\mathbf{r}}(\hat{\mathbf{v}}; \boldsymbol{\mu}) \|_2$$

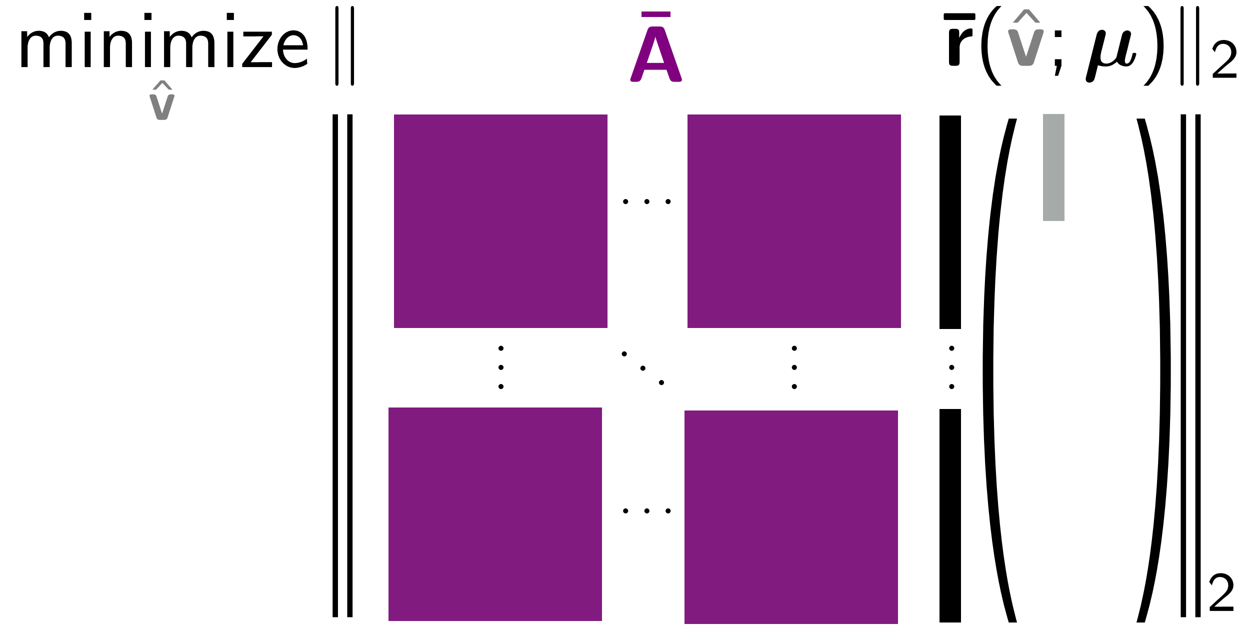


$$\bar{\mathbf{r}} \approx \tilde{\mathbf{r}} = \bar{\boldsymbol{\Phi}}_{\mathbf{r}} (\bar{\mathbf{P}} \bar{\boldsymbol{\Phi}}_{\mathbf{r}})^{+} \bar{\mathbf{P}} \bar{\mathbf{r}}$$

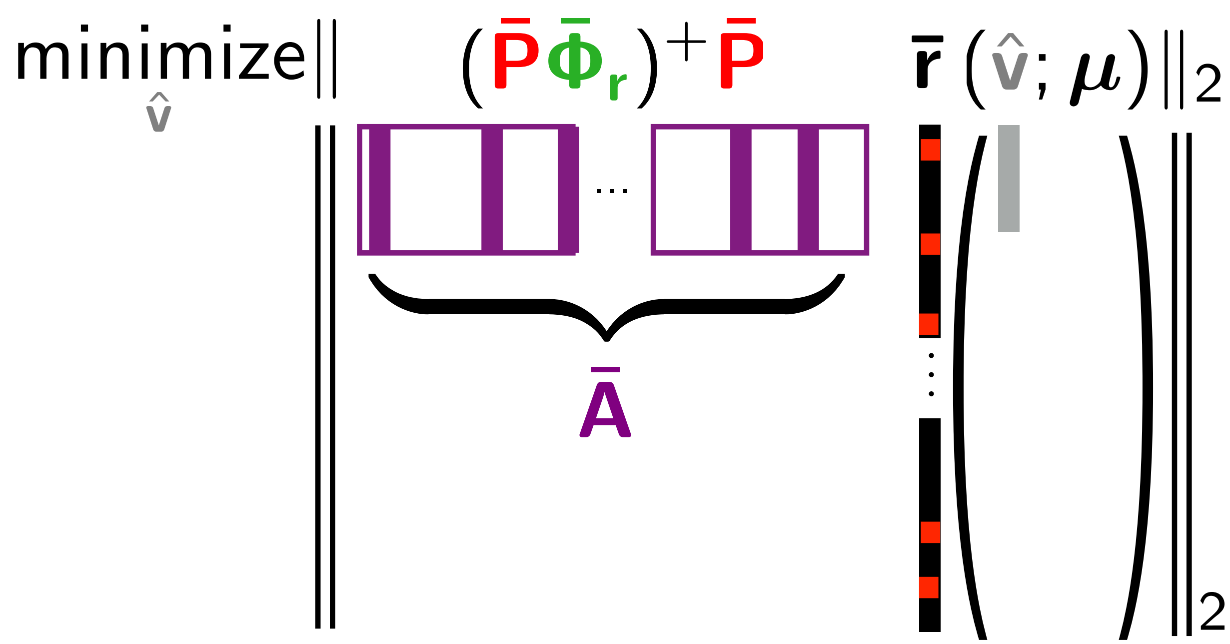
$$\underset{\hat{\mathbf{v}}}{\text{minimize}} \; \| \tilde{\mathbf{r}}(\hat{\mathbf{v}}; \boldsymbol{\mu}) \|_2$$



ST-LSPG hyper-reduction



$$\bar{\mathbf{r}} \approx \tilde{\mathbf{r}} = \bar{\boldsymbol{\Phi}}_{\mathbf{r}}(\bar{\mathbf{P}}\bar{\boldsymbol{\Phi}}_{\mathbf{r}})^{+}\bar{\mathbf{P}}\bar{\mathbf{r}}$$

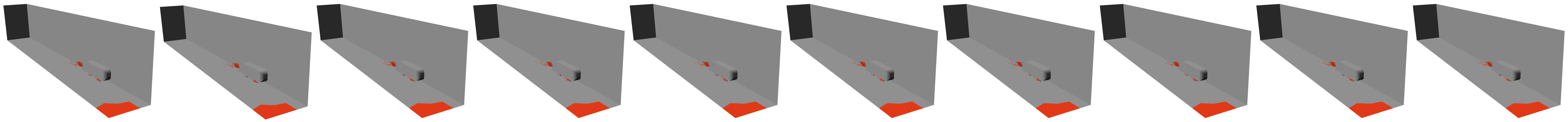


+ Residual computed at a few space–time degrees of freedom

Sample mesh

LSPG

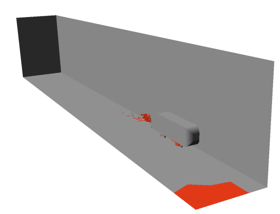
t^1 t^2 t^3 t^4 t^5 t^6 t^7 t^8 t^9 t^{10}



- Residual computed at a few spatial degrees of freedom, all time instances

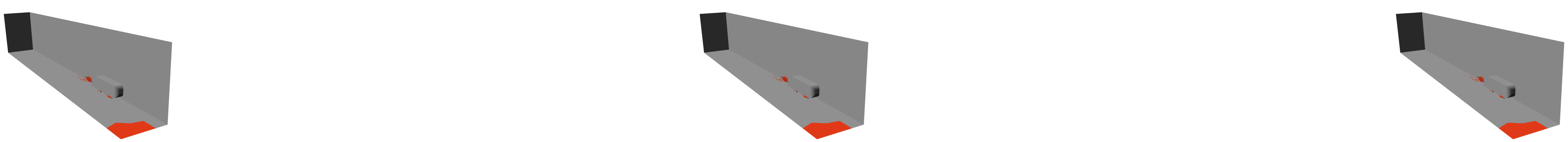
ST-LSPG

- $\bar{\mathbf{P}}$: Kronecker product of space sampling and time sampling



t^1, t^5, t^9

t^1 t^5 t^9



+ Residual computed at a few space–time degrees of freedom

Error bound

LSPG

- *Sequential solves*: sequential accumulation of time-local errors

$$\|\mathbf{x}^n - \mathbf{\Phi}\hat{\mathbf{x}}^n_{\text{LSPG}}\|_2 \leq \frac{\gamma_1(\gamma_2)^n \exp(\gamma_3 t^n)}{\gamma_4 + \gamma_5 \Delta t} \underbrace{\max_{j\in\{1,\ldots,n\}} \min_{\hat{\mathbf{v}}} \|\mathbf{r}^j_{\text{LSPG}}(\mathbf{\Phi}\hat{\mathbf{v}})\|_2}_{\text{worst best time-local approximation residual}}$$

- *Stability constant*: exponential time growth
- bounded by the worst (over time) best residual

ST-LSPG

+ *Single solve*: no sequential error accumulation

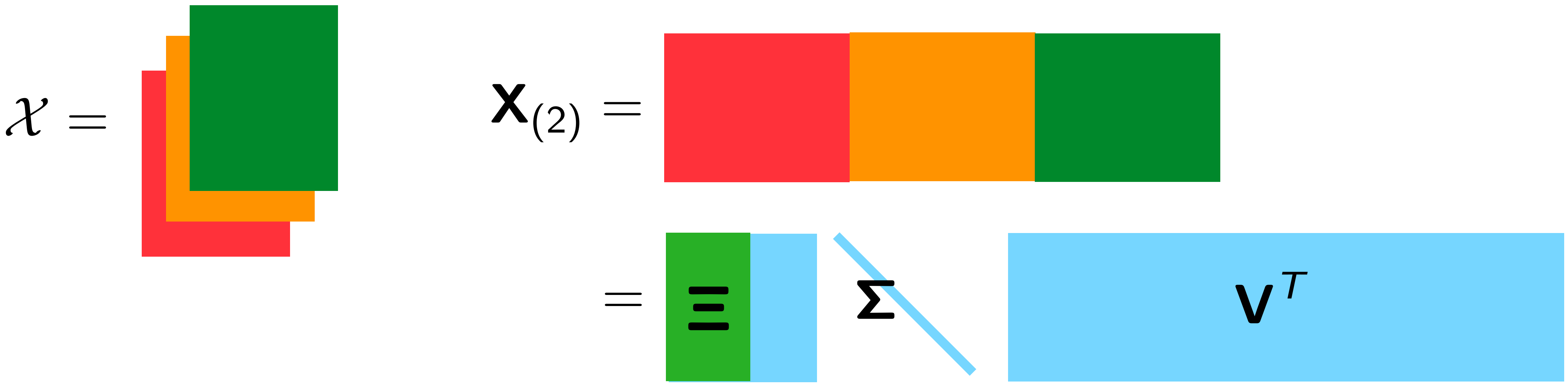
$$\|\mathbf{x}^n - \mathbf{\Phi}\hat{\mathbf{x}}^n_{\text{ST-LSPG}}\|_2 \leq \sqrt{T}(1+\Lambda) \underbrace{\min_{\mathbf{w}\in\mathcal{ST}} \max_{j\in\{1,\ldots,T\}} \|\mathbf{x}^n - \mathbf{w}^n\|_2}_{\text{best space-time approximation error}}$$

+ *Stability constant*: polynomial growth in time with degree 3/2
+ bounded by best space–time approximation error

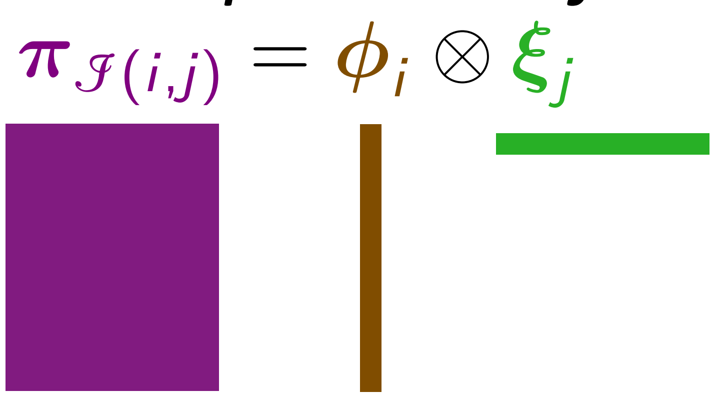
How to construct space–time trial basis $\{\boldsymbol{\pi}_i\}_{i=1}^{n_{\text{st}}}$ from snapshot data?

Algorithm

1. *Training:* Solve ODE for $\boldsymbol{\mu} \in \mathcal{D}_{\text{training}}$ and collect simulation data
2. *Machine learning:* Compute truncated high-order SVD (T-HOSVD)
3. *Reduction:* Solve space–time LSPG ROM for $\boldsymbol{\mu} \in \mathcal{D}_{\text{query}} \setminus \mathcal{D}_{\text{training}}$



$\boldsymbol{\Xi}$ *columns are principal components of the* ***temporal*** *simulation data*



+ extracts multiple space–time basis vectors from each training simulation
- **Experiments**: for fixed error, ST-LSPG almost 100X faster than LSPG

Our research

Accurate, low-cost, structure-preserving, reliable, certified nonlinear model reduction

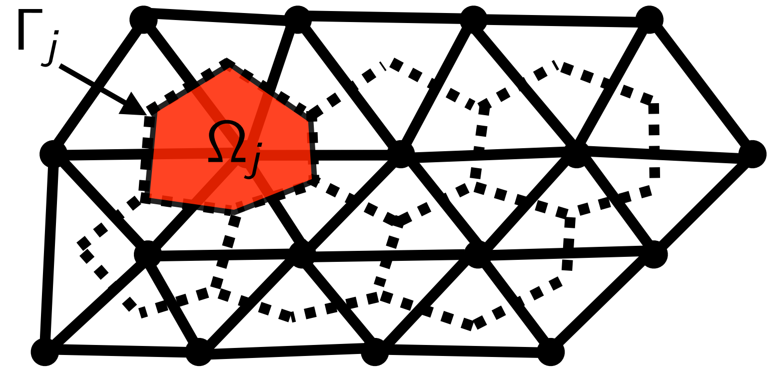
- *accuracy*: LSPG projection [C., Bou-Mosleh, Farhat, 2011; C., Barone, Antil, 2017]
- *low cost*: sample mesh [C., Farhat, Cortial, Amsallem, 2013]
- *low cost*: reduce temporal complexity
  [C., Ray, van Bloemen Waanders, 2015; C., Brencher, Haasdonk, Barth, 2017; Choi and C., 2019]
- *structure preservation* [C., Tuminaro, Boggs, 2015*; Peng and C., 2017; C., Choi, Sargsyan, 2018]
- *robustness*: projection onto nonlinear manifolds [Lee, C., 2018]
- *robustness*: *h*-adaptivity [C., 2015]
- *certification*: machine learning error models
  [Drohmann and C., 2015; Trehan, C., Durlofsky, 2017; Freno and C., 2019; Pagani, Manzoni, C., 2019]

Collaborators: *Youngsoo Choi (Sandia), Syuzanna Sargsyan (UW)*

* Featured Article, SIAM J Sci Comp, 2015

Finite-volume method

$$\text{ODE:} \quad \frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}; t)$$



$$x_{\mathcal{I}(i,j)}(t) = \frac{1}{|\Omega_j|} \int_{\Omega_j} u_i(\vec{x}, t)\, d\vec{x}$$

- average value of conserved variable i over control volume j

$$f_{\mathcal{I}(i,j)}(\mathbf{x}, t) = -\frac{1}{|\Omega_j|} \int_{\Gamma_j} \underbrace{\mathbf{g}_i(\mathbf{x}; \vec{x}, t)}_{\text{flux}} \cdot \mathbf{n}_j(\vec{x})\, d\vec{s}(\vec{x}) + \frac{1}{|\Omega_j|} \int_{\Omega_j} \underbrace{s_i(\mathbf{x}; \vec{x}, t)}_{\text{source}}\, d\vec{x}$$

- flux and source of conserved variable i within control volume j

$$r_{\mathcal{I}(i,j)} = \frac{dx_{\mathcal{I}(i,j)}}{dt}(t) - f_{\mathcal{I}(i,j)}(\mathbf{x}, t)$$

- **rate of conservation violation** of variable i in control volume j

$$\text{O}\Delta\text{E:} \quad \mathbf{r}^n(\mathbf{x}^n) = 0, \quad n = 1, \ldots, N$$

$$r^n_{\mathcal{I}(i,j)} = x_{\mathcal{I}(i,j)}(t^{n+1}) - x_{\mathcal{I}(i,j)}(t^n) + \int_{t^n}^{t^{n+1}} f_{\mathcal{I}(i,j)}(\mathbf{x}, t)\, dt$$

- **conservation violation** of variable i in control volume j over time step n

Conservation is the intrinsic structure enforced by finite-volume methods

Galerkin and LSPG violate conservation

Galerkin

$$\boldsymbol{\Phi}\frac{d\hat{\mathbf{x}}}{dt}(\boldsymbol{\Phi}\hat{\mathbf{x}}, t) = \underset{\mathbf{v}\in\text{range}(\boldsymbol{\Phi})}{\arg\min} \|\mathbf{r}(\mathbf{v}, \boldsymbol{\Phi}\hat{\mathbf{x}}, t)\|_2$$

- Minimize sum of squared **conservation-violation rates**

LSPG

$$\boldsymbol{\Phi}\hat{\mathbf{x}}^n = \underset{\mathbf{v}\in\text{range}(\boldsymbol{\Phi})}{\arg\min} \|\mathbf{r}^n(\mathbf{v})\|_2$$

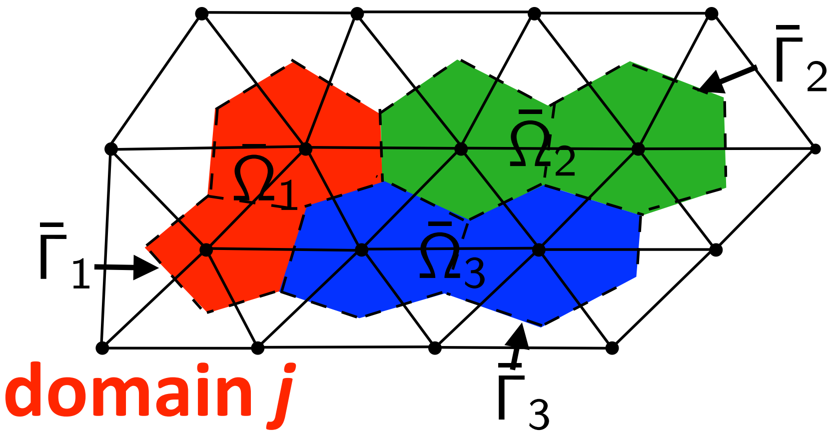
- Minimize sum of squared **conservation violations over time step *n***

- Neither ensures conservation!

- **Goal**: devise projections that enforce conservation over subdomains

Conservative model reduction for finite-volume models [C., Choi, Sargsyan, 2018]

Finite-volume method **over subdomains**



$$\text{ODE: } \bar{\mathbf{C}}\frac{d\mathbf{x}}{dt} = \bar{\mathbf{C}}\mathbf{f}(\mathbf{x}, t)$$

$$\bar{c}_{\bar{\mathcal{I}}(i,j),\mathcal{I}(\ell,k)} = |\Omega_k|/|\bar{\Omega}_j|\delta_{i\ell}I(\Omega_k \subseteq \bar{\Omega}_j)$$

- performs summation over control volumes within **subdomain *j***

$$[\bar{\mathbf{C}}\mathbf{x}(t)]_{\bar{\mathcal{I}}(i,j)}(\mathbf{x}, t; \boldsymbol{\mu}) = \frac{1}{|\bar{\Omega}_j|}\int_{\bar{\Omega}_j} u_i(\vec{x}, t; \boldsymbol{\mu})\, \mathrm{d}\vec{x}$$

- average value of conserved variable *i* over **subdomain *j***

$$[\bar{\mathbf{C}}\mathbf{f}(\mathbf{x}, t)]_{\bar{\mathcal{I}}(i,j)} = -\frac{1}{|\bar{\Omega}_j|}\int_{\bar{\Gamma}_j} \underbrace{\mathbf{g}_i(\mathbf{x}; \vec{x}, t)}_{\text{flux}} \cdot \bar{\mathbf{n}}_j(\vec{x})\, \mathrm{d}\vec{s}(\vec{x}) + \frac{1}{|\bar{\Omega}_j|}\int_{\bar{\Omega}_j} \underbrace{s_i(\mathbf{x}; \vec{x}, t)}_{\text{source}}\, \mathrm{d}\vec{x}$$

- flux and source of conserved variable *i* within **subdomain *j***

$$[\bar{\mathbf{C}}\mathbf{r}]_{\bar{\mathcal{I}}(i,j)} = d[\bar{\mathbf{C}}\mathbf{x}(t)]_{\bar{\mathcal{I}}(i,j)}/dt - [\bar{\mathbf{C}}\mathbf{f}(\mathbf{x}, t)]_{\bar{\mathcal{I}}(i,j)}$$

- **rate of conservation violation** of conserved variable *i* in **subdomain *j***

$$\text{O}\Delta\text{E: } \bar{\mathbf{C}}\mathbf{r}^n(\mathbf{x}^n) = \mathbf{0}, \quad n = 1, \ldots, T$$

$$[\bar{\mathbf{C}}\mathbf{r}^n]_{\bar{\mathcal{I}}(i,j)} = [\bar{\mathbf{C}}\mathbf{x}(t^{n+1})]_{\bar{\mathcal{I}}(i,j)} - [\bar{\mathbf{C}}\mathbf{x}(t^n)]_{\bar{\mathcal{I}}(i,j)} + \int_{t^n}^{t^{n+1}} [\bar{\mathbf{C}}\mathbf{f}(\mathbf{x}, t)]_{\bar{\mathcal{I}}(i,j)}\, dt$$

- **conservation violation** of conserved variable *i* in **subdomain *j*** over time step *n*

Conservative model reduction

Conservative Galerkin

$$\underset{\hat{\mathbf{v}}\in\mathbb{R}^p}{\text{minimize}} \ \|\mathbf{r}(\mathbf{\Phi}\hat{\mathbf{v}}, \mathbf{\Phi}\hat{\mathbf{x}}, t)\|_2$$

$$\text{subject to } \bar{\mathbf{C}}\mathbf{r}(\mathbf{\Phi}\hat{\mathbf{v}}, \mathbf{\Phi}\hat{\mathbf{x}}, t) = \mathbf{0}$$

- Minimize sum of squared **conservation-violation rates** subject to zero conservation-violation rates **over subdomains**

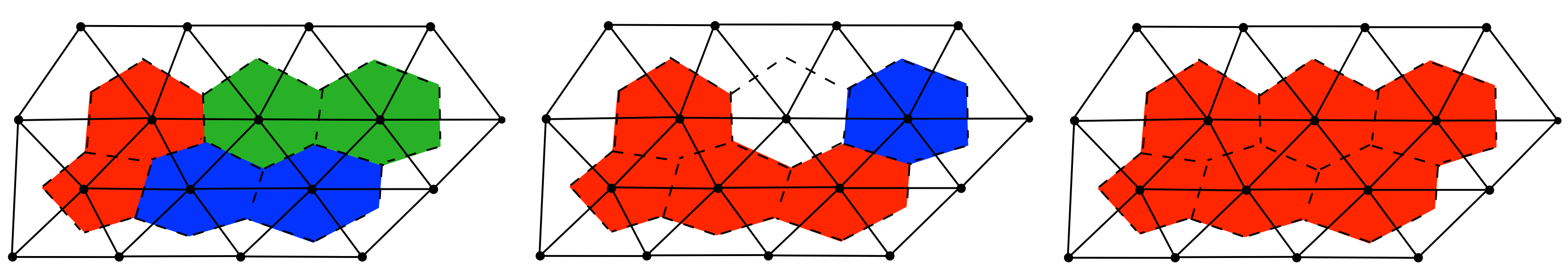
Conservative LSPG

$$\underset{\hat{\mathbf{v}}\in\mathbb{R}^p}{\text{minimize}} \ \|\mathbf{r}^n(\mathbf{\Phi}\hat{\mathbf{v}})\|_2$$

$$\text{subject to } \bar{\mathbf{C}}\mathbf{r}^n(\mathbf{\Phi}\hat{\mathbf{v}}) = \mathbf{0}$$

- Minimize sum of squared **conservation violations over time step *n*** subject to zero conservation violations over time step *n* **over subdomains**

+ Conservation enforced over prescribed subdomains



Conservative model reduction

Conservative Galerkin

$$\underset{\hat{\mathbf{v}}\in\mathbb{R}^p}{\text{minimize}} \ \|\mathbf{r}(\boldsymbol{\Phi}\hat{\mathbf{v}}, \boldsymbol{\Phi}\hat{\mathbf{x}}, t)\|_2$$

$$\text{subject to } \bar{\mathbf{C}}\mathbf{r}(\boldsymbol{\Phi}\hat{\mathbf{v}}, \boldsymbol{\Phi}\hat{\mathbf{x}}, t) = \mathbf{0}$$

- Minimize sum of squared **conservation-violation rates** subject to zero conservation-violation rates **over subdomains**

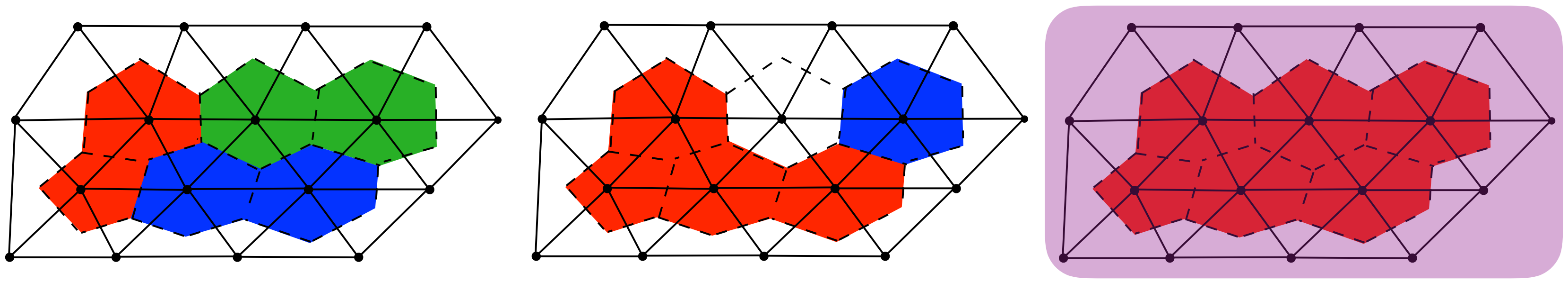
Conservative LSPG

$$\underset{\hat{\mathbf{v}}\in\mathbb{R}^p}{\text{minimize}} \ \|\mathbf{r}^n(\boldsymbol{\Phi}\hat{\mathbf{v}})\|_2$$

$$\text{subject to } \bar{\mathbf{C}}\mathbf{r}^n(\boldsymbol{\Phi}\hat{\mathbf{v}}) = \mathbf{0}$$

- Minimize sum of squared **conservation violations over time step *n*** subject to zero conservation violations over time step *n* **over subdomains**

+ Conservation enforced over prescribed subdomains



- **Experiments**: enforcing global conservation can reduce error by 10X

Our research

Accurate, low-cost, structure-preserving, reliable, certified nonlinear model reduction

- *accuracy*: LSPG projection [C., Bou-Mosleh, Farhat, 2011; C., Barone, Antil, 2017]
- *low cost*: sample mesh [C., Farhat, Cortial, Amsallem, 2013]
- *low cost*: reduce temporal complexity [C., Ray, van Bloemen Waanders, 2015; C., Brencher, Haasdonk, Barth, 2017; Choi and C., 2019]
- *structure preservation* [C., Tuminaro, Boggs, 2015; Peng and C., 2017; C., Choi, Sargsyan, 2018]
- *robustness*: projection onto nonlinear manifolds [Lee, C., 2018]
- *robustness*: *h*-adaptivity [C., 2015]
- *certification*: machine learning error models [Drohmann and C., 2015; Trehan, C., Durlofsky, 2017; Freno and C., 2019; Pagani, Manzoni, C., 2019]

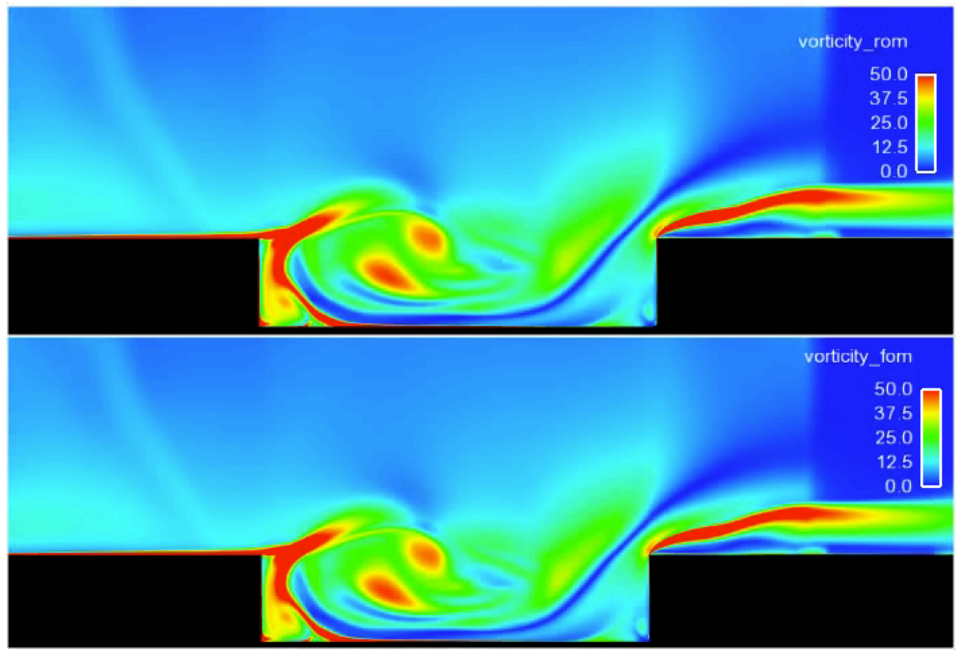
Collaborator: *Kookjin Lee (Sandia)*

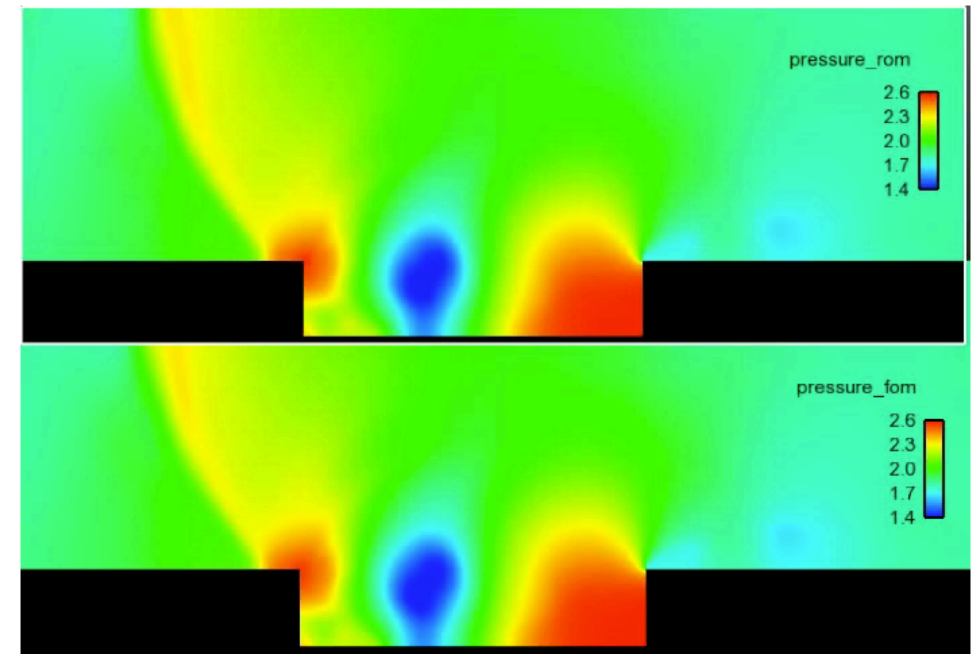
Model reduction can work well…

vorticity field *pressure field*

LSPG ROM with
$\mathbf{A} = (\mathbf{P}\boldsymbol{\Phi}_{\mathbf{r}})^{+}\mathbf{P}$
32 min, 2 cores

high-fidelity
5 hours, 48 cores

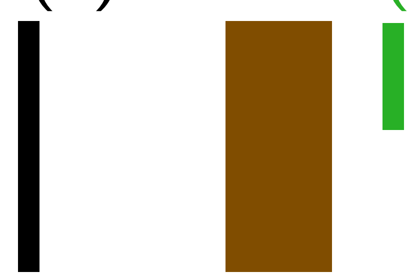




+ *229x savings* *in core–hours*
+ *< 1% error* *in time-averaged drag*

… however, this is not guaranteed

$$\mathbf{x}(t) \approx \boldsymbol{\Phi}\,\hat{\mathbf{x}}(t)$$



1) ***Linear-subspace assumption is strong***
2) ***Accuracy limited by content of*** $\boldsymbol{\Phi}$

Kolmogorov-width limitation of linear subspaces

$$d_p(\mathcal{M}) := \inf_{\mathcal{S}_p} P_\infty(\mathcal{M}, \mathcal{S}_p) \qquad P_\infty(\mathcal{M}, \mathcal{S}_p) := \sup_{\mathbf{x}\in\mathcal{M}} \inf_{\mathbf{y}\in\mathcal{S}_p} \|\mathbf{x} - \mathbf{y}\|$$

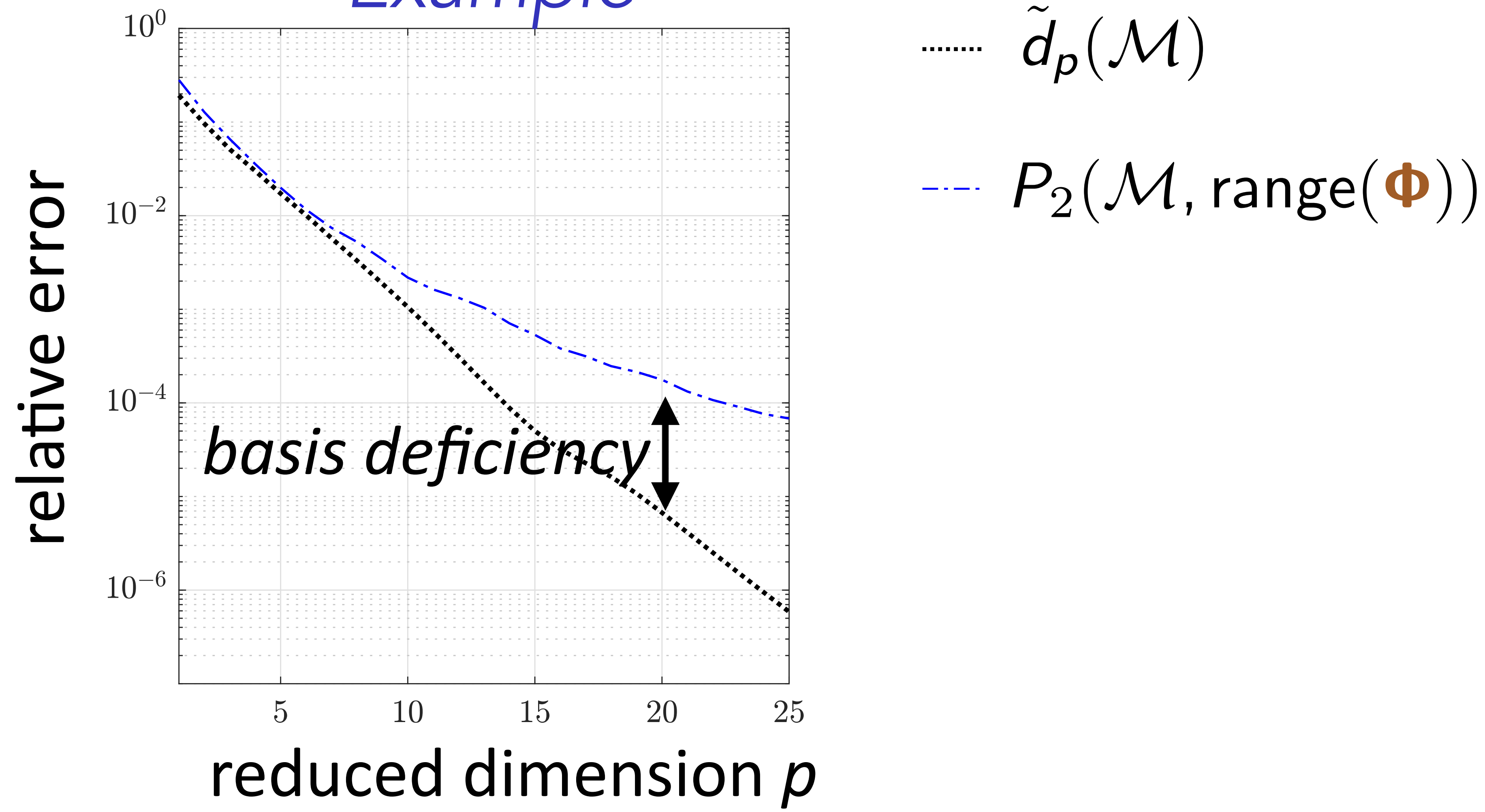
- $\mathcal{M} := \{\mathbf{x}(t, \boldsymbol{\mu}) \mid t \in [0, T_{\text{final}}],\ \boldsymbol{\mu} \in \mathcal{D}\}$: solution manifold
- $\mathcal{S}_p$: set of all *p*-dimensional linear subspaces

Kolmogorov-width limitation of linear subspaces

$$\tilde{d}_p(\mathcal{M}) := \inf_{\mathcal{S}_p} P_2(\mathcal{M}, \mathcal{S}_p) \qquad P_2(\mathcal{M}, \mathcal{S}_p) := \sqrt{\sum_{\mathbf{x}\in\mathcal{M}} \inf_{\mathbf{y}\in\mathcal{S}_p} \|\mathbf{x}-\mathbf{y}\|^2} \Big/ \sqrt{\sum_{\mathbf{x}\in\mathcal{M}} \|\mathbf{x}\|^2}$$

- $\mathcal{M} := \{\mathbf{x}(t, \boldsymbol{\mu}) \mid t \in [0, T_{\text{final}}], \boldsymbol{\mu} \in \mathcal{D}\}$: solution manifold
- $\mathcal{S}_p$: set of all *p*-dimensional linear subspaces

Example

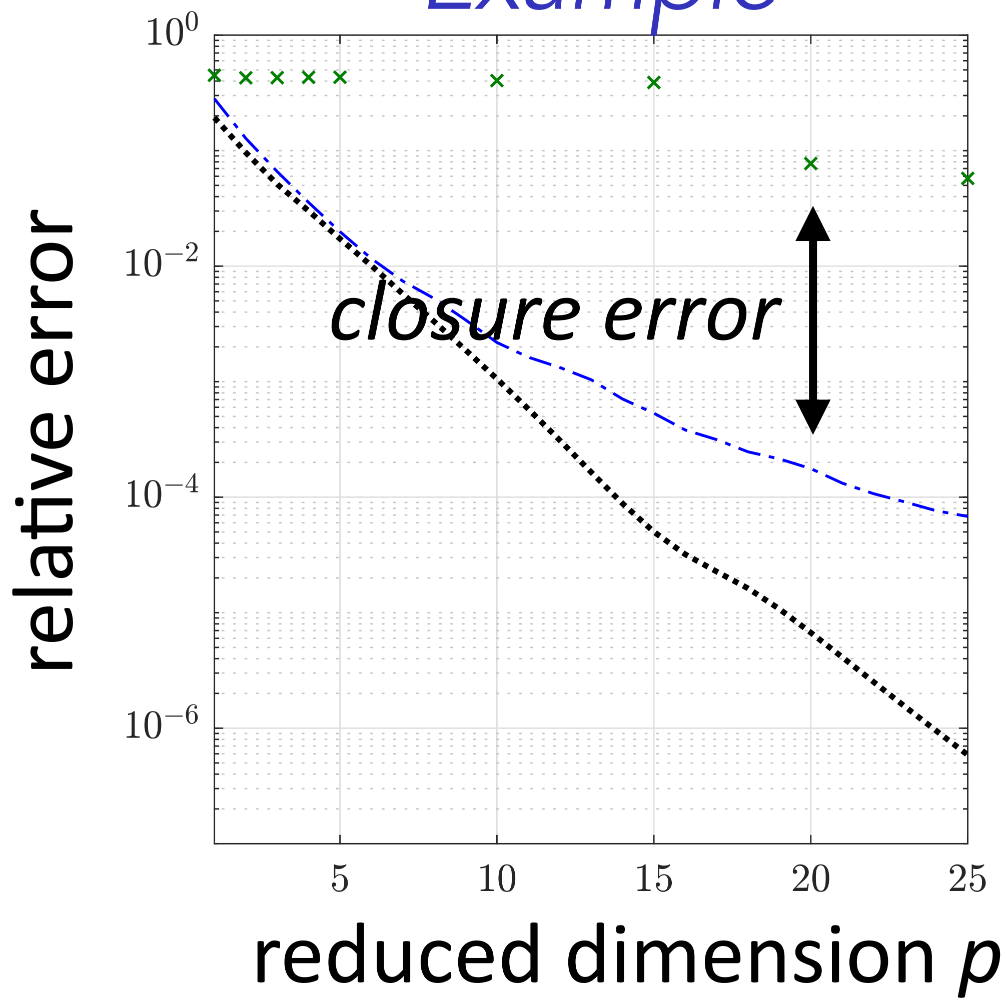


Kolmogorov-width limitation of linear subspaces

$$\tilde{d}_p(\mathcal{M}) := \inf_{\mathcal{S}_p} P_2(\mathcal{M}, \mathcal{S}_p) \qquad P_2(\mathcal{M}, \mathcal{S}_p) := \sqrt{\sum_{\mathbf{x}\in\mathcal{M}} \inf_{\mathbf{y}\in\mathcal{S}_p} \|\mathbf{x}-\mathbf{y}\|^2} \Big/ \sqrt{\sum_{\mathbf{x}\in\mathcal{M}} \|\mathbf{x}\|^2}$$

- $\mathcal{M} := \{\mathbf{x}(t, \boldsymbol{\mu}) \mid t \in [0, T_{\text{final}}],\ \boldsymbol{\mu} \in \mathcal{D}\}$: solution manifold
- $\mathcal{S}_p$: set of all p-dimensional linear subspaces

Example



$\tilde{d}_p(\mathcal{M})$

$P_2(\mathcal{M}, \text{range}(\boldsymbol{\Phi}))$

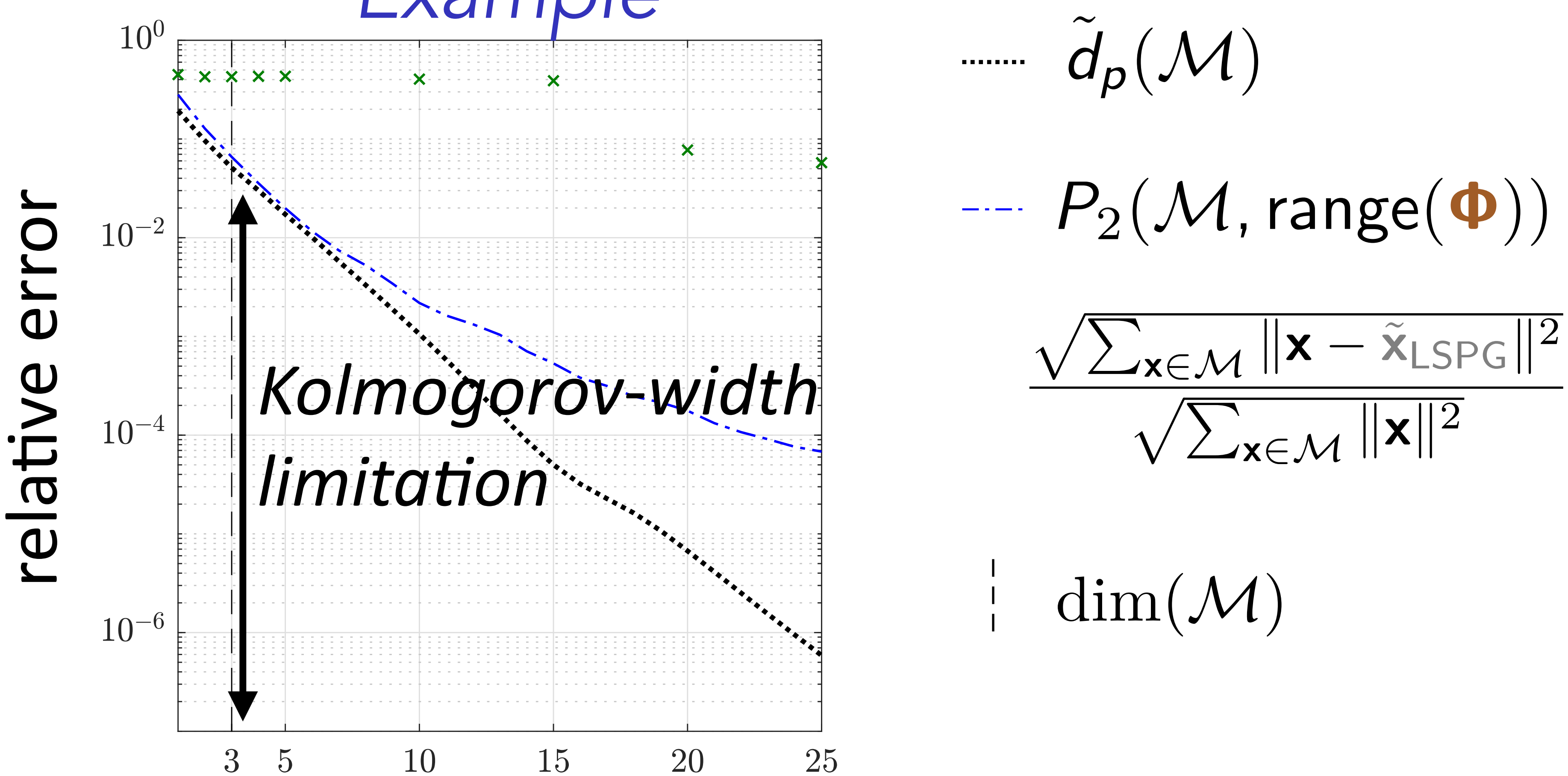
$$\frac{\sqrt{\sum_{\mathbf{x}\in\mathcal{M}} \|\mathbf{x}-\tilde{\mathbf{x}}_{\text{LSPG}}\|^2}}{\sqrt{\sum_{\mathbf{x}\in\mathcal{M}} \|\mathbf{x}\|^2}}$$

Kolmogorov-width limitation of linear subspaces

$$\tilde{d}_p(\mathcal{M}) := \inf_{\mathcal{S}_p} P_2(\mathcal{M}, \mathcal{S}_p) \qquad P_2(\mathcal{M}, \mathcal{S}_p) := \sqrt{\sum_{\mathbf{x}\in\mathcal{M}} \inf_{\mathbf{y}\in\mathcal{S}_p} \|\mathbf{x}-\mathbf{y}\|^2} \Big/ \sqrt{\sum_{\mathbf{x}\in\mathcal{M}} \|\mathbf{x}\|^2}$$

- $\mathcal{M} := \{\mathbf{x}(t, \boldsymbol{\mu}) \,|\, t \in [0, T_{\text{final}}],\ \boldsymbol{\mu} \in \mathcal{D}\}$: solution manifold
- $\mathcal{S}_p$: set of all p-dimensional linear subspaces

Example



- Kolmogorov-width limitation: significant error for $p = \dim(\mathcal{M})$
- **Goal**: overcome limitation via projection onto a nonlinear manifold

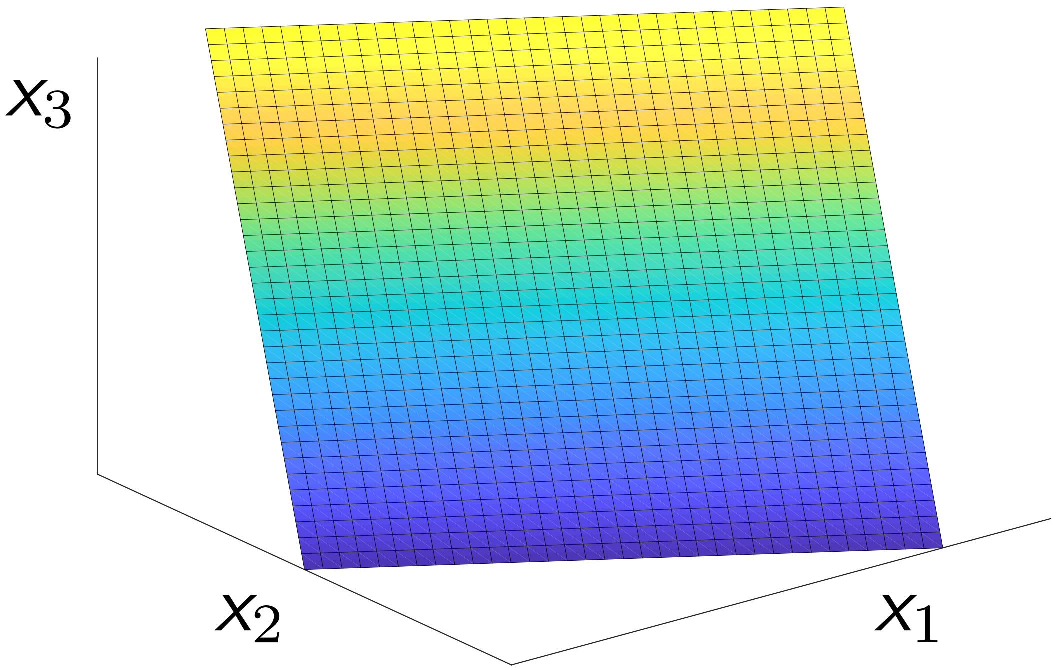
Nonlinear trial manifold

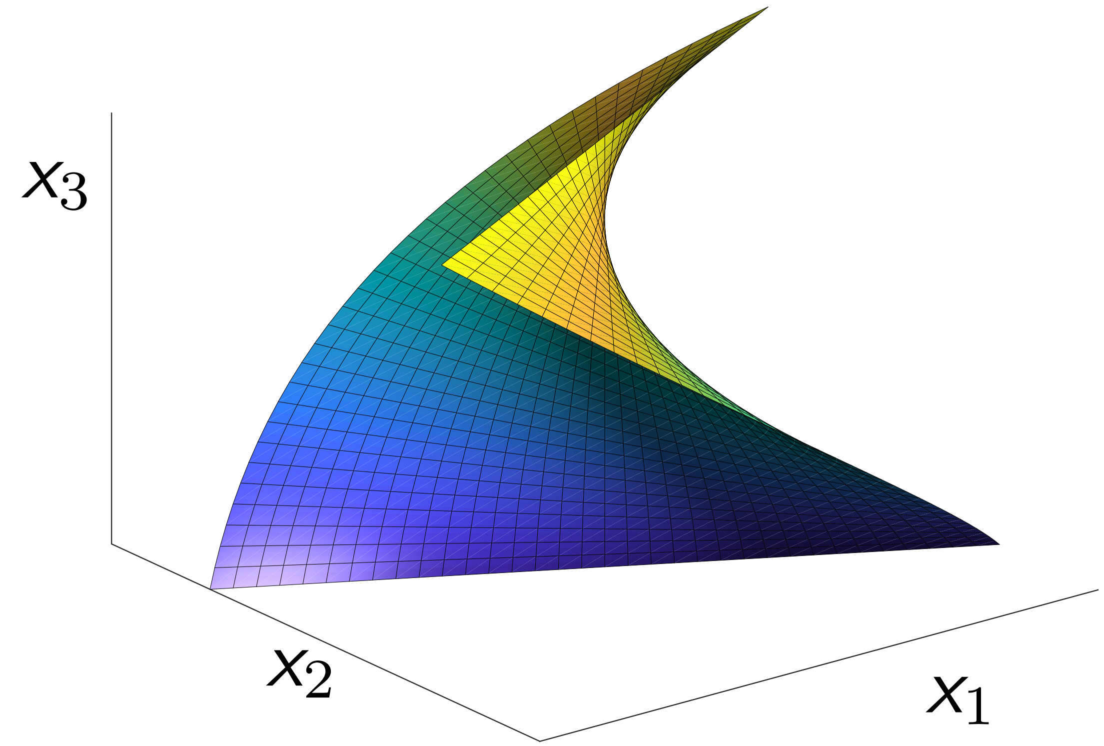
Linear trial subspace | **Nonlinear trial manifold**

$\text{range}(\boldsymbol{\Phi}) := \{\boldsymbol{\Phi}\hat{\mathbf{x}} \mid \hat{\mathbf{x}} \in \mathbb{R}^p\}$

$\mathcal{S} := \{\mathbf{g}(\hat{\mathbf{x}}) \mid \hat{\mathbf{x}} \in \mathbb{R}^p\}$

example
N=3
p=2

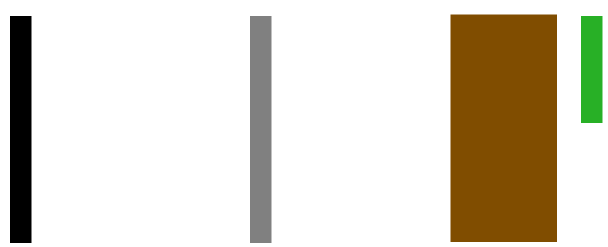




state

$\mathbf{x}(t) \approx \tilde{\mathbf{x}}(t) = \boldsymbol{\Phi}\,\hat{\mathbf{x}}(t) \in \text{range}(\boldsymbol{\Phi})$

$\mathbf{x}(t) \approx \tilde{\mathbf{x}}(t) = \mathbf{g}(\hat{\mathbf{x}}(t)) \in \mathcal{S}$



\+ manifold has general structure

velocity

$$\frac{d\mathbf{x}}{dt} \approx \frac{d\tilde{\mathbf{x}}}{dt} = \boldsymbol{\Phi}\frac{d\hat{\mathbf{x}}}{dt} \in \text{range}(\boldsymbol{\Phi})$$

$$\frac{d\mathbf{x}}{dt} \approx \frac{d\tilde{\mathbf{x}}}{dt} = \nabla\mathbf{g}(\hat{\mathbf{x}})\frac{d\hat{\mathbf{x}}}{dt} \in T_{\hat{\mathbf{x}}}\mathcal{S}$$

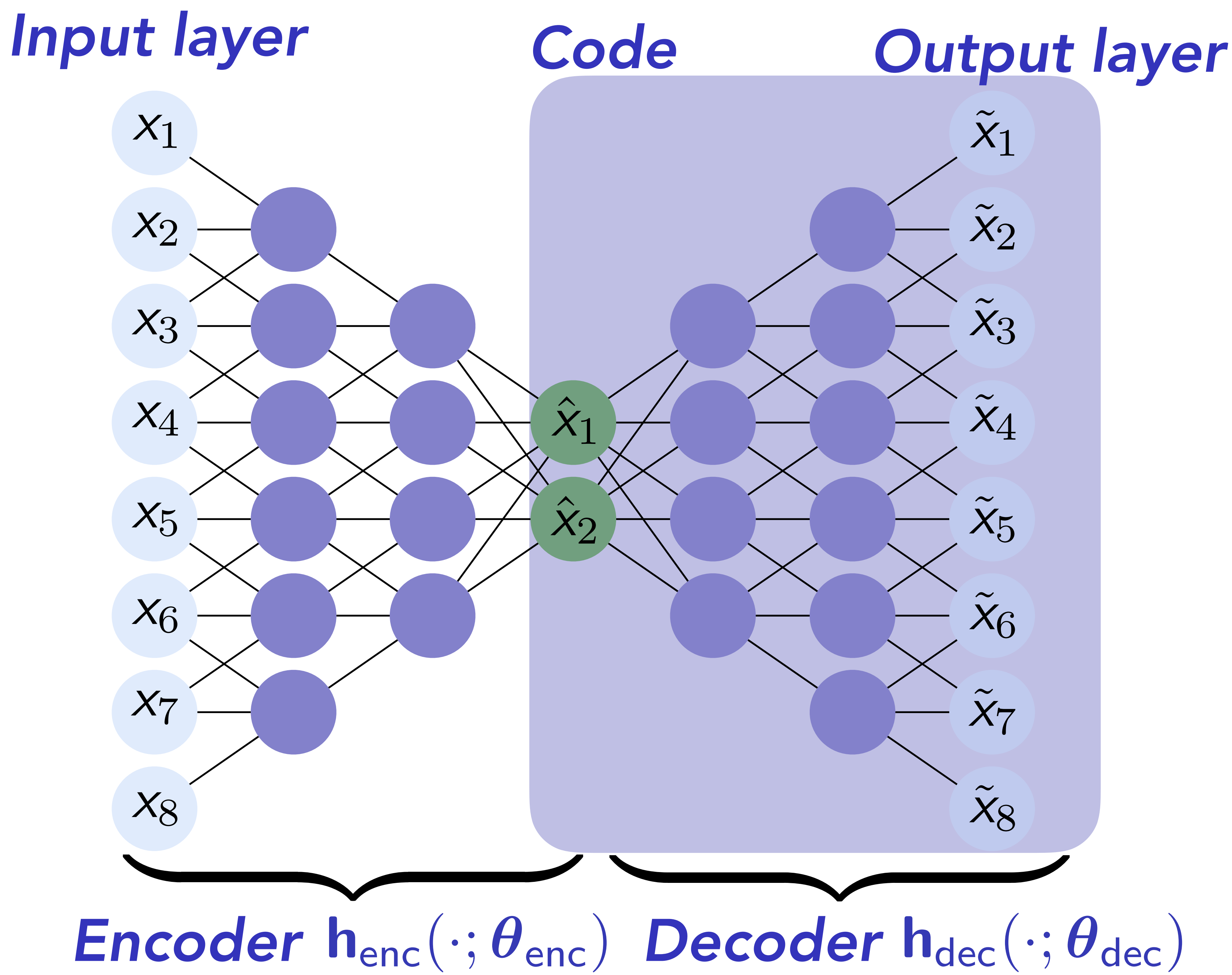
Manifold Galerkin and LSPG projection

	Linear-subspace ROM	***Nonlinear-manifold ROM***
Galerkin	$\frac{d\hat{\mathbf{x}}}{dt} = \underset{\hat{\mathbf{v}}\in\mathbb{R}^n}{\text{argmin}}\|\mathbf{r}(\boldsymbol{\Phi}\hat{\mathbf{v}},\boldsymbol{\Phi}\hat{\mathbf{x}};t)\|_2$	$\frac{d\hat{\mathbf{x}}}{dt} = \underset{\hat{\mathbf{v}}\in\mathbb{R}^n}{\text{argmin}}\|\mathbf{r}(\nabla\mathbf{g}(\hat{\mathbf{x}})\hat{\mathbf{v}},\mathbf{g}(\hat{\mathbf{x}});t)\|_2$
	$\Updownarrow$	$\Updownarrow$
	$\boldsymbol{\Phi}\frac{d\hat{\mathbf{x}}}{dt} = \underset{\hat{\mathbf{v}}\in\text{range}(\boldsymbol{\Phi})}{\text{argmin}}\|\hat{\mathbf{v}}-\mathbf{f}(\boldsymbol{\Phi}\hat{\mathbf{x}};t)\|_2$	$\nabla\mathbf{g}(\hat{\mathbf{x}})\frac{d\hat{\mathbf{x}}}{dt} = \underset{\hat{\mathbf{v}}\in T_{\hat{\mathbf{x}}}\mathcal{S}}{\text{argmin}}\|\hat{\mathbf{v}}-\mathbf{f}(\mathbf{g}(\hat{\mathbf{x}});t)\|_2$
	$\Updownarrow$	$\Updownarrow$
	$\frac{d\hat{\mathbf{x}}}{dt} = \boldsymbol{\Phi}^T\mathbf{f}(\boldsymbol{\Phi}\hat{\mathbf{x}};t)$	$\frac{d\hat{\mathbf{x}}}{dt} = \nabla\mathbf{g}(\hat{\mathbf{x}})^+\mathbf{f}(\mathbf{g}(\hat{\mathbf{x}});t)$
LSPG	$\hat{\mathbf{x}}^n = \underset{\hat{\mathbf{v}}\in\mathbb{R}^p}{\text{argmin}}\|\mathbf{r}^n(\boldsymbol{\Phi}\hat{\mathbf{v}})\|_2$	$\hat{\mathbf{x}}^n = \underset{\hat{\mathbf{v}}\in\mathbb{R}^p}{\text{argmin}}\|\mathbf{r}^n(\mathbf{g}(\hat{\mathbf{v}}))\|_2$

\+ *Satisfy residual-minimization properties*

How to construct manifold $\mathcal{S} := \{\mathbf{g}(\hat{\mathbf{x}}) \mid \hat{\mathbf{x}}\in\mathbb{R}^p\}$ ***from snapshot data?***

Deep autoencoders

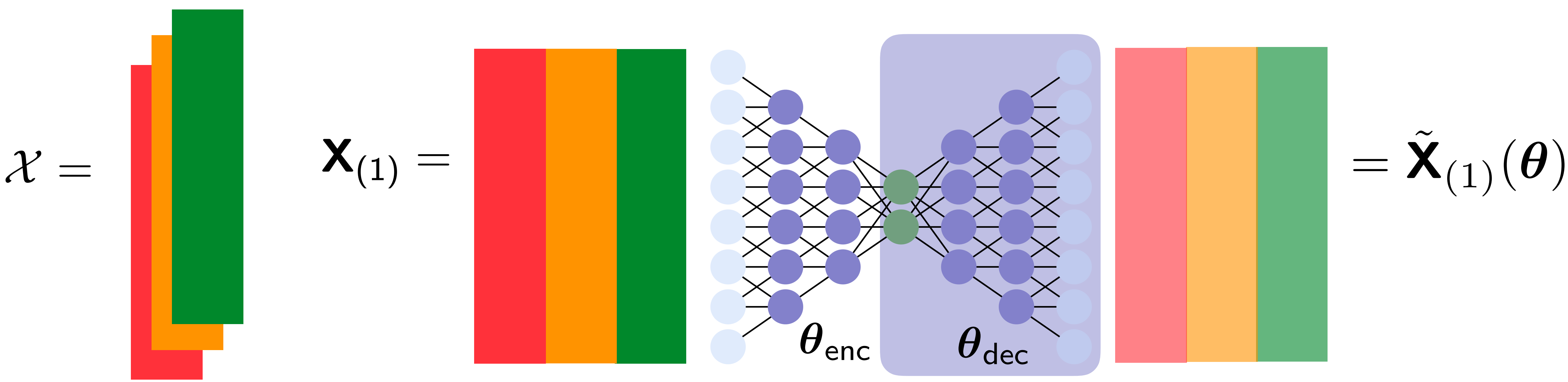


$$\tilde{\mathbf{x}} = \mathbf{h}_{\text{dec}}(\cdot;\boldsymbol{\theta}_{\text{dec}}) \circ \mathbf{h}_{\text{enc}}(\mathbf{x};\boldsymbol{\theta}_{\text{enc}})$$

+ If $\tilde{\mathbf{x}} \approx \mathbf{x}$ for parameters $\boldsymbol{\theta}^{\star}_{\text{dec}}$, $\mathbf{g} = \mathbf{h}_{\text{dec}}(\cdot;\boldsymbol{\theta}^{\star}_{\text{dec}})$ produces an accurate manifold

Algorithm

1. *Training:* Solve ODE for $\mu \in \mathcal{D}_{\text{training}}$ and collect simulation data
2. *Machine learning:* Train deep convolutional autoencoder
3. *Reduction:* Solve manifold Galerkin or LSPG for $\mu \in \mathcal{D}_{\text{query}} \setminus \mathcal{D}_{\text{training}}$



- Compute $\boldsymbol{\theta}^\star$ by approximately solving $\underset{\boldsymbol{\theta}}{\text{minimize}} \|\mathbf{X}_{(1)} - \tilde{\mathbf{X}}_{(1)}(\boldsymbol{\theta})\|_F$
- Define nonlinear trial manifold by setting $\mathbf{g} = \mathbf{h}_{\text{dec}}(\cdot;\boldsymbol{\theta}^\star_{\text{dec}})$

\+ Same snapshot data

Numerical results

1D Burgers' equation

$$\frac{\partial w(x,t;\boldsymbol{\mu})}{\partial t} + \frac{\partial f(w(x,t;\boldsymbol{\mu}))}{\partial x} = 0.02e^{\alpha x}$$

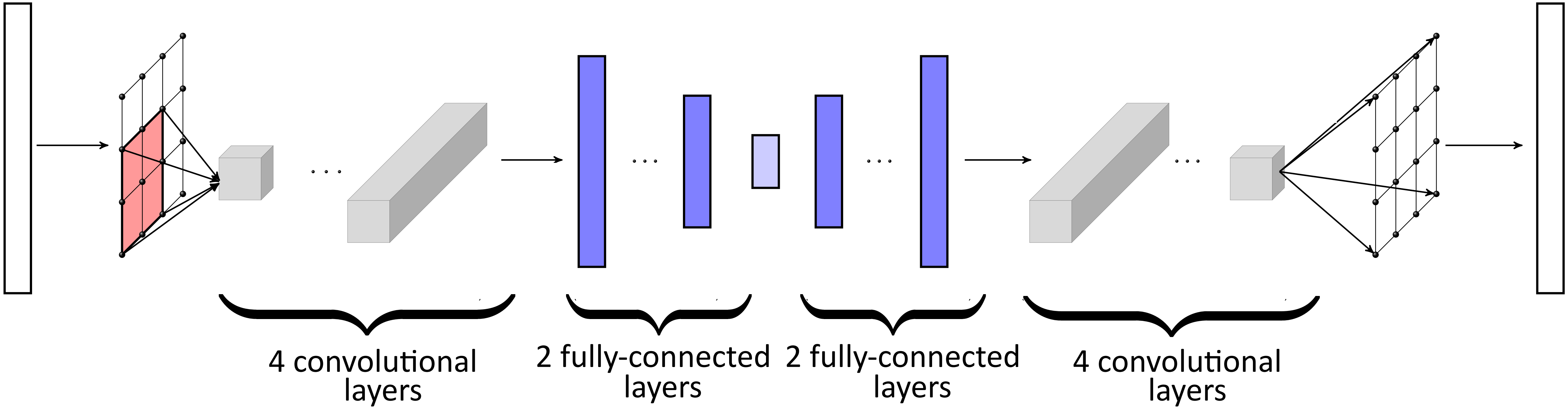
- $\boldsymbol{\mu}$: α, inlet boundary condition
- *Spatial discretization*: finite volume
- *Time integrator:* backward Euler

2D reacting flow

$$\frac{\partial \mathbf{w}(\vec{x},t;\boldsymbol{\mu})}{\partial t} = \nabla \cdot (\kappa \nabla \mathbf{w}(\vec{x},t;\boldsymbol{\mu})) - \mathbf{v} \cdot \nabla \mathbf{w}(\vec{x},t;\boldsymbol{\mu}) + \mathbf{q}(\mathbf{w}(\vec{x},t;\boldsymbol{\mu});\boldsymbol{\mu})$$

- $\boldsymbol{\mu}$: two terms in reaction
- *Spatial discretization*: finite difference
- *Time integrator*: BDF2

Autoencoder architecture



Manifold LSPG outperforms optimal linear subspace

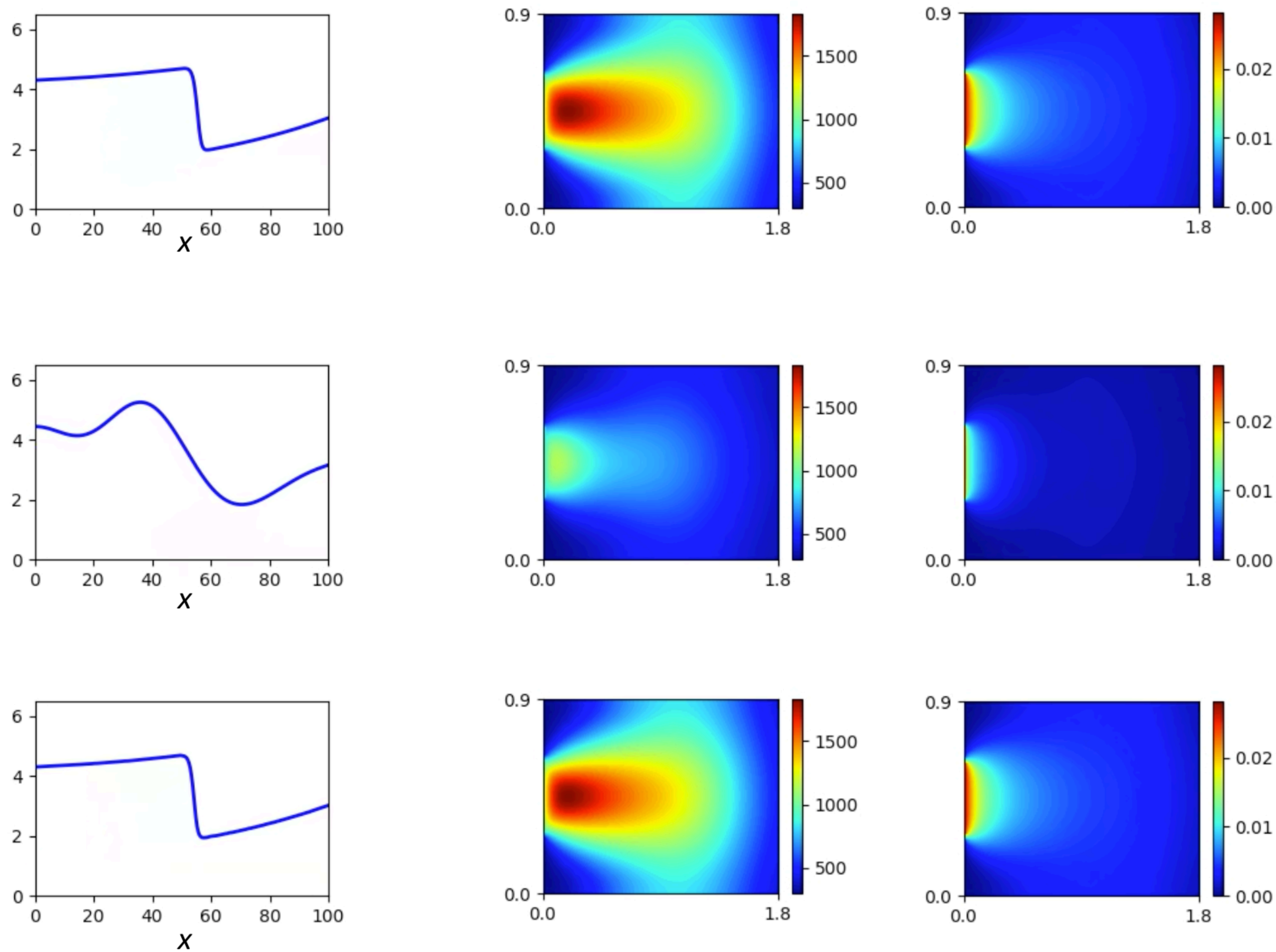
1D Burgers' equation | *2D reacting flow*

conserved variable | temperature | H_2 fraction

high-fidelity model

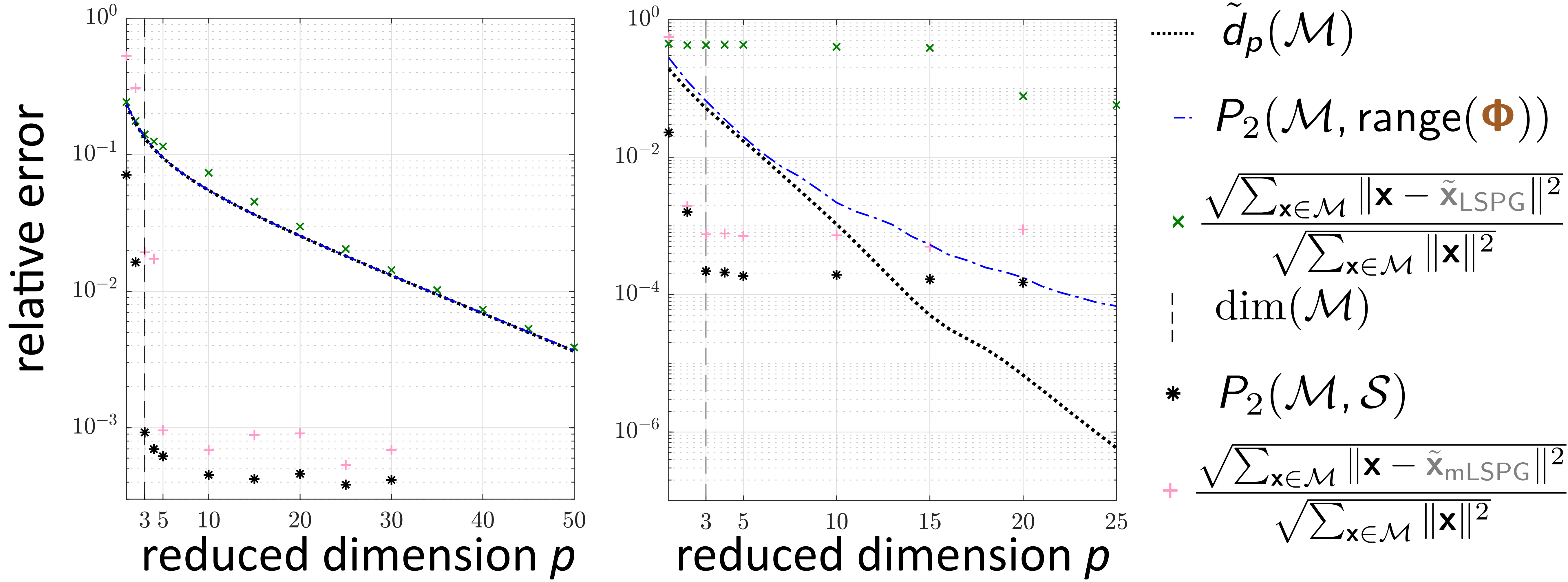
POD-LSPG p=5

Manifold LSPG p=5



Method overcomes Kolmogorov-width limitation

1D Burgers' equation *2D reacting flow*



- Autoencoder manifold significantly better than optimal linear subspace
- Manifold LSPG orders-of-magnitude more accurate than subspace LSPG
- Method overcomes Kolmogorov-width limitation

Our research

Accurate, low-cost, structure-preserving, reliable, certified nonlinear model reduction

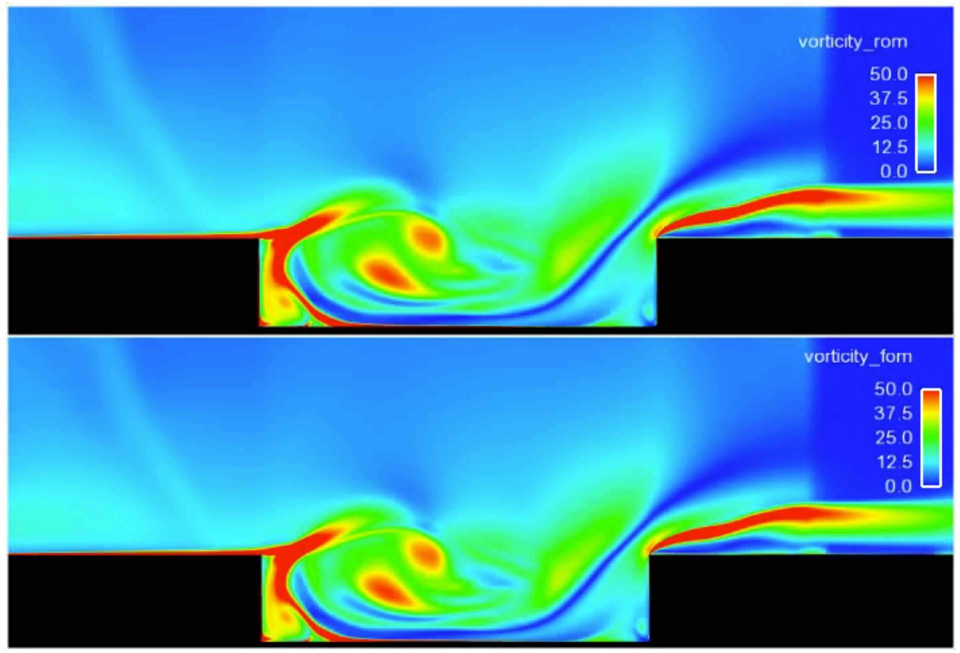
- *accuracy*: LSPG projection [C., Bou-Mosleh, Farhat, 2011; C., Barone, Antil, 2017]
- *low cost*: sample mesh [C., Farhat, Cortial, Amsallem, 2013]
- *low cost*: reduce temporal complexity
  [C., Ray, van Bloemen Waanders, 2015; C., Brencher, Haasdonk, Barth, 2017; Choi and C., 2019]
- *structure preservation* [C., Tuminaro, Boggs, 2015; Peng and C., 2017; C., Choi, Sargsyan, 2017]
- *robustness*: projection onto nonlinear manifolds [Lee, C., 2018]
- *robustness*: *h*-adaptivity [C., 2015]
- *certification*: machine learning error models
  [Drohmann and C., 2015; Trehan, C., Durlofsky, 2017; Freno and C., 2019; Pagani, Manzoni, C., 2019]

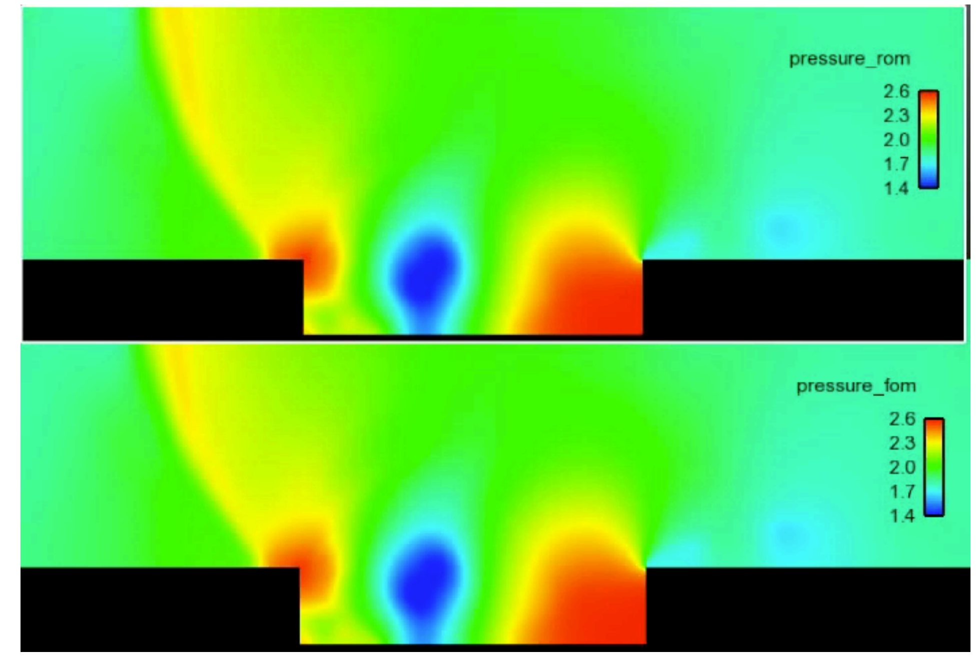
Model reduction can work well…

vorticity field | *pressure field*

LSPG ROM with
$\mathbf{A} = (\mathbf{P}\boldsymbol{\Phi}_{\mathbf{r}})^{+}\mathbf{P}$
32 min, 2 cores

high-fidelity
5 hours, 48 cores

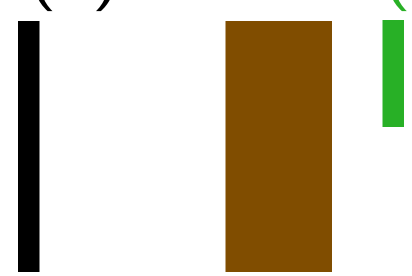




+ *229x savings in core–hours*
+ *< 1% error in time-averaged drag*

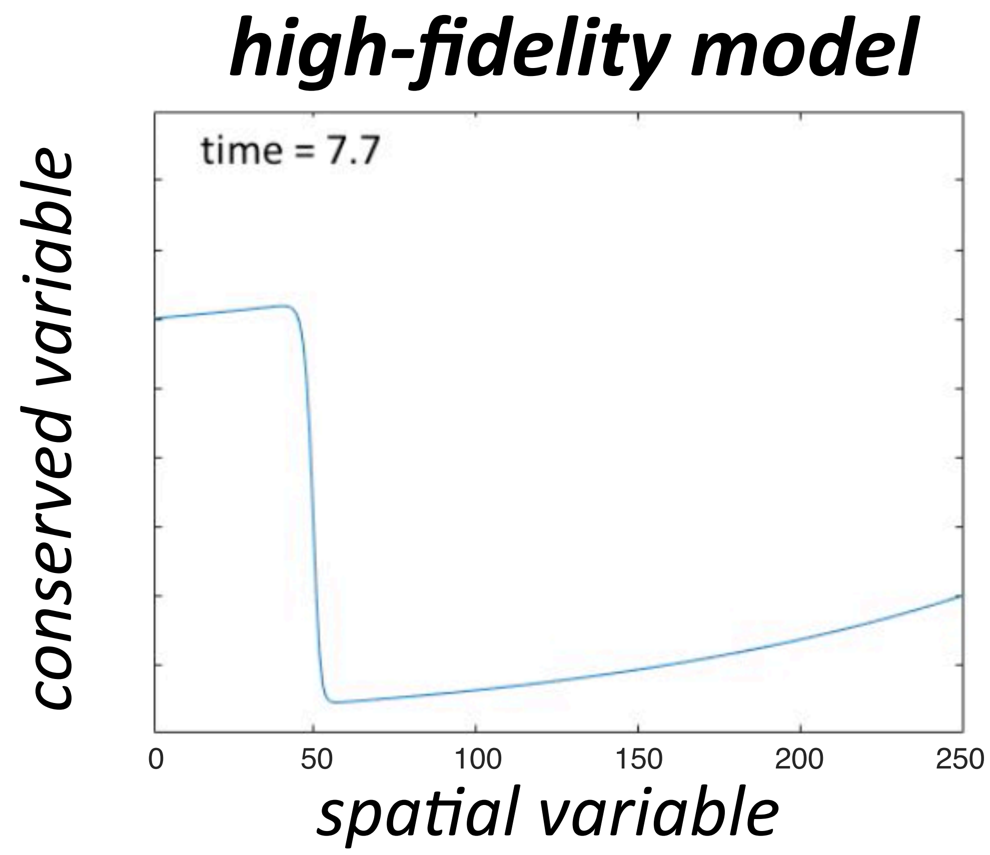
… however, this is not guaranteed

$$\mathbf{x}(t) \approx \boldsymbol{\Phi}\,\hat{\mathbf{x}}(t)$$



1) ***Linear-subspace assumption is strong***
2) ***Accuracy limited by content of $\boldsymbol{\Phi}$***

Illustration: inviscid 1D Burgers' equation



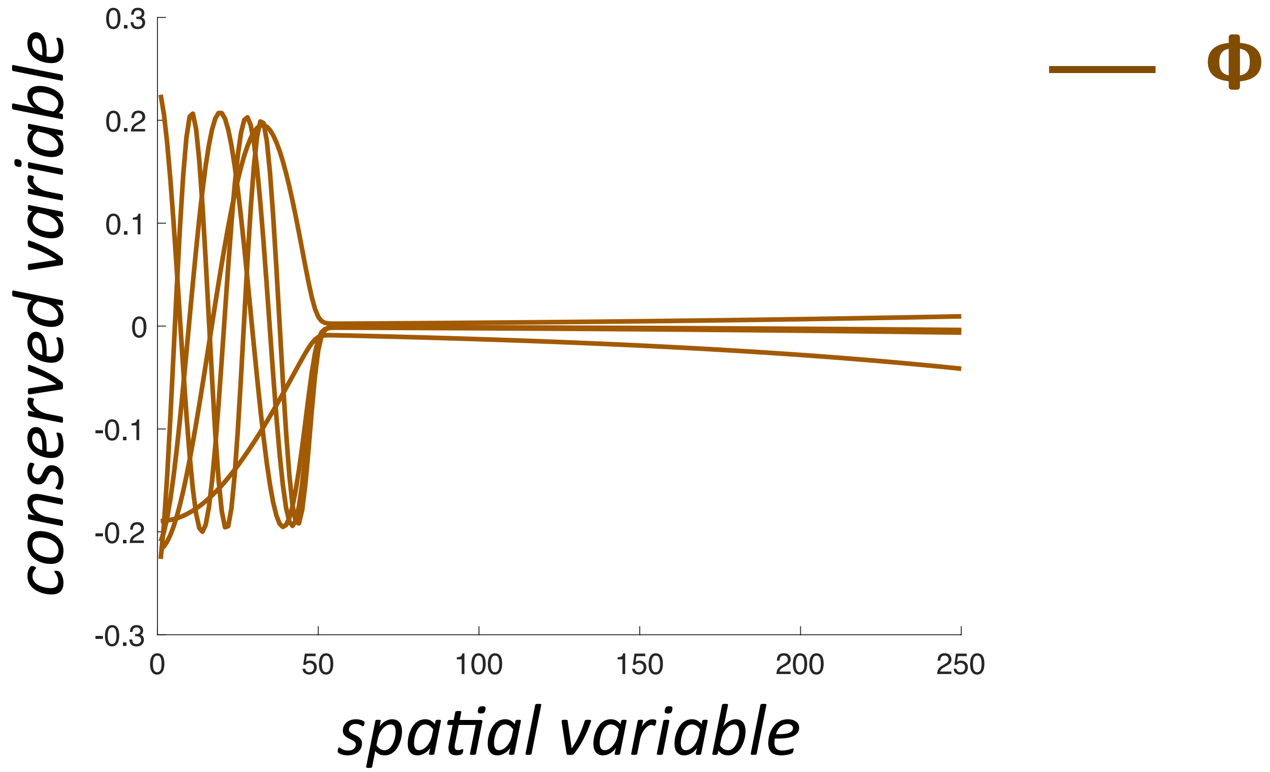
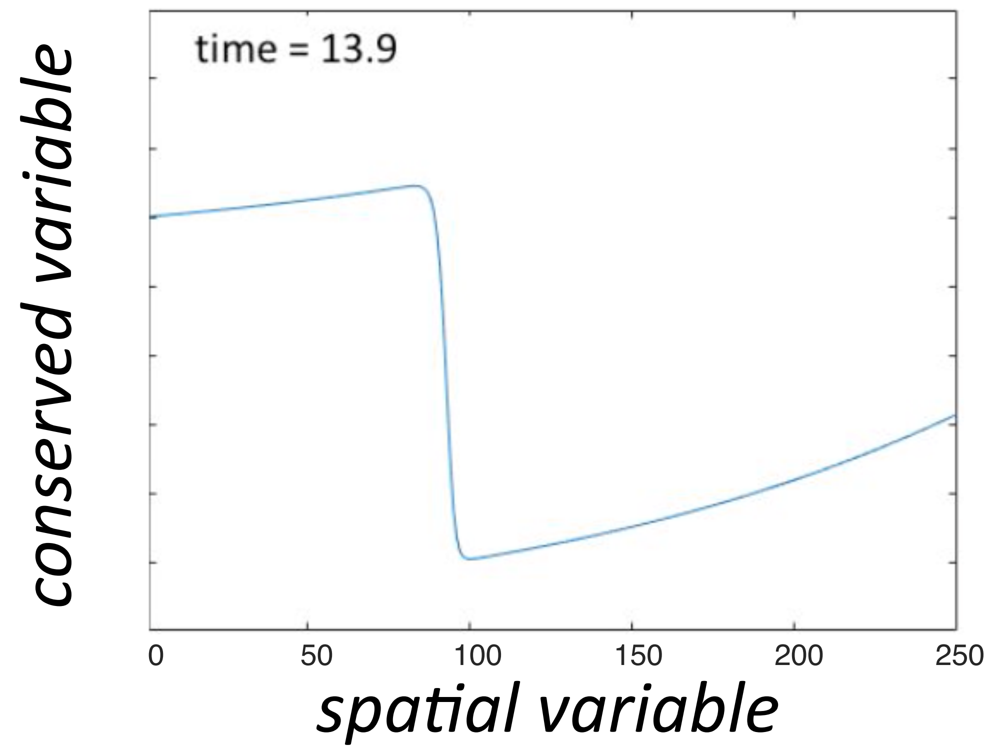
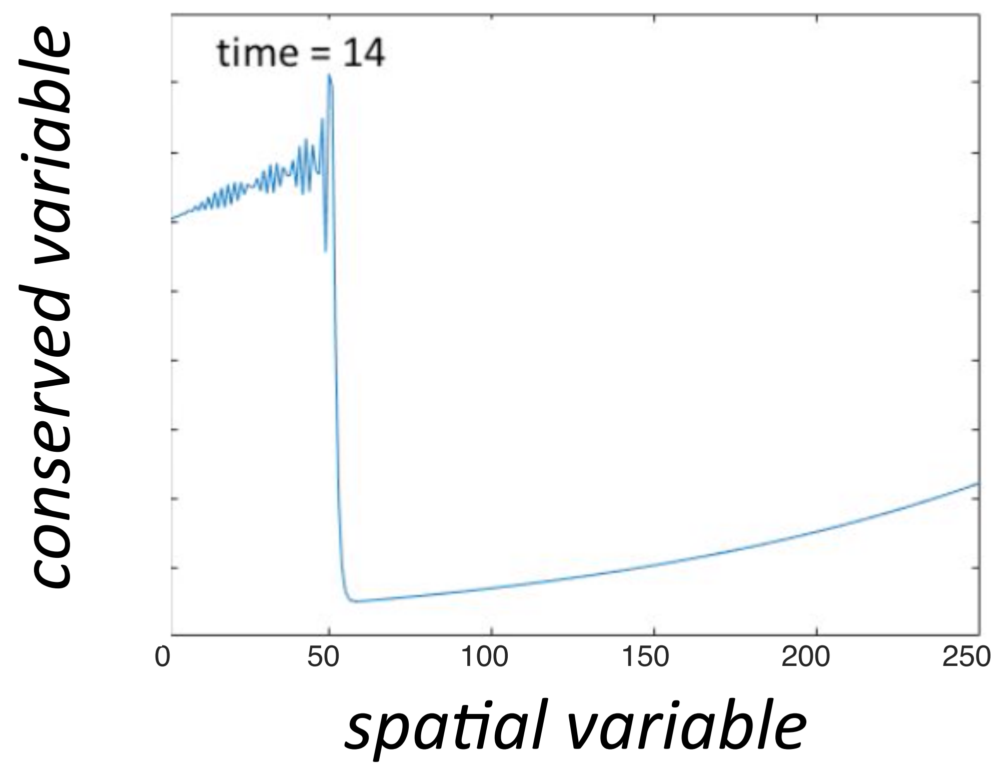


Illustration: inviscid 1D Burgers' equation

high-fidelity model



reduced-order model

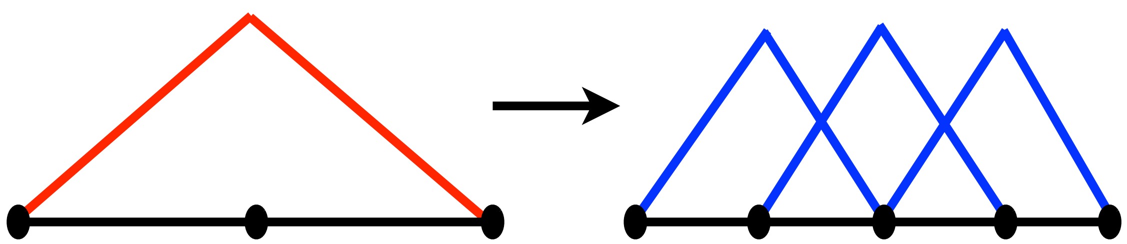


reduced-order model inaccurate when Φ insufficient

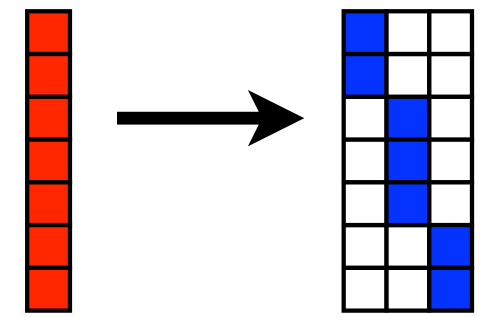
Main idea [C., 2015]

Model-reduction analogue to mesh-adaptive h-refinement

- 'Split' basis vectors

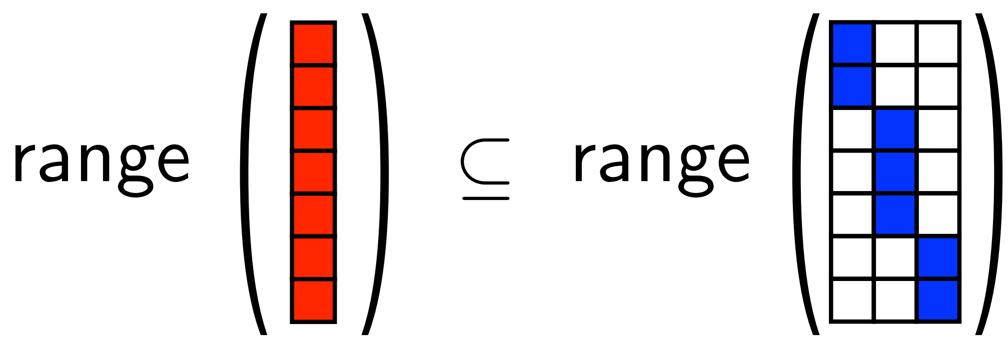


finite-element h-refinement

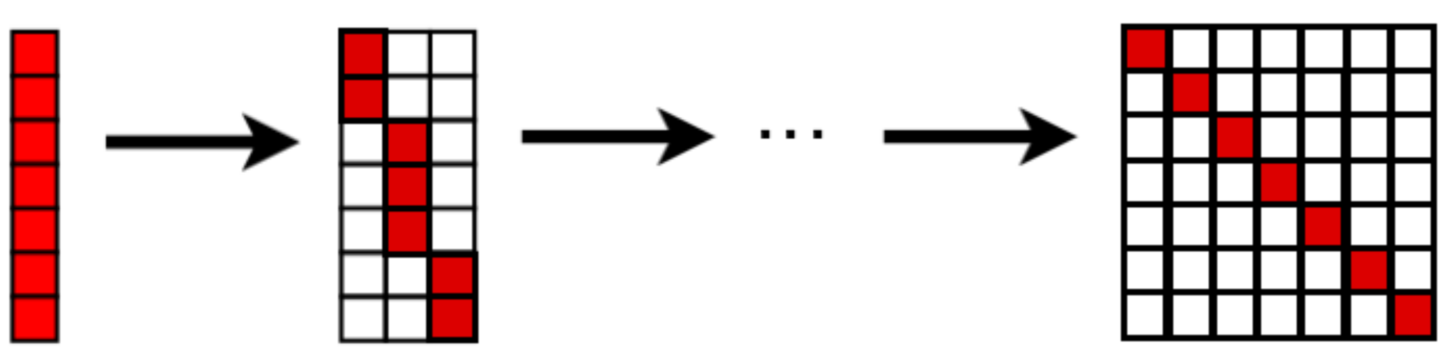


reduced-order-model h-refinement

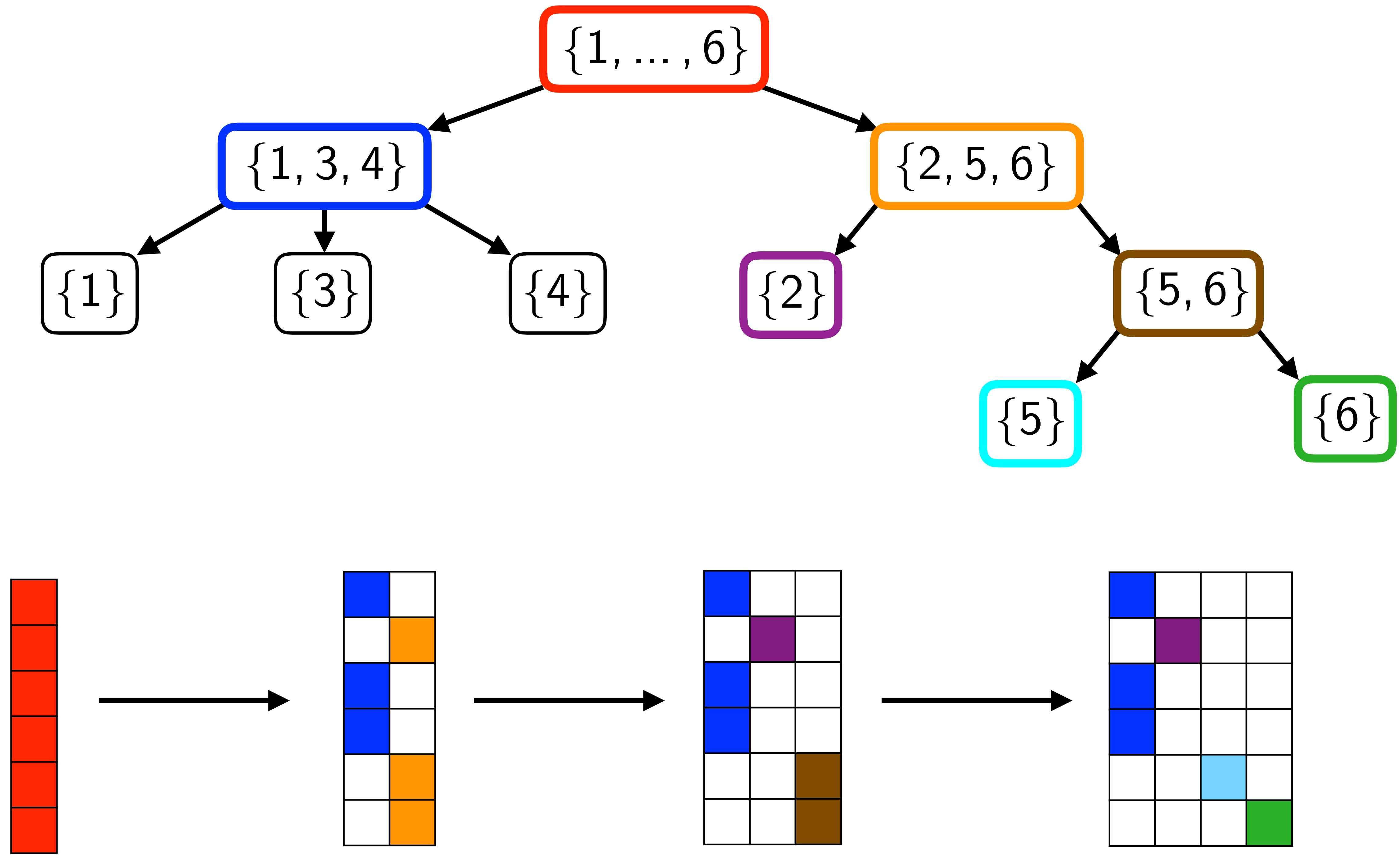
- Generate hierarchical subspaces



- Converges to the high-fidelity model



Refinement tree encodes splitting

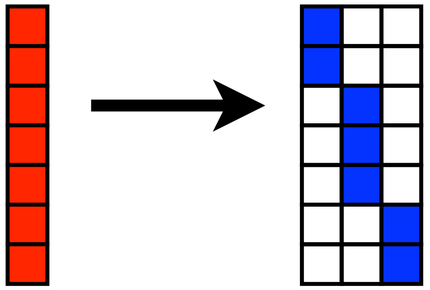


Refinement tree requirements

Theorem [C., 2015]

h-adaptivity generates a **hierarchy of subspaces** if:

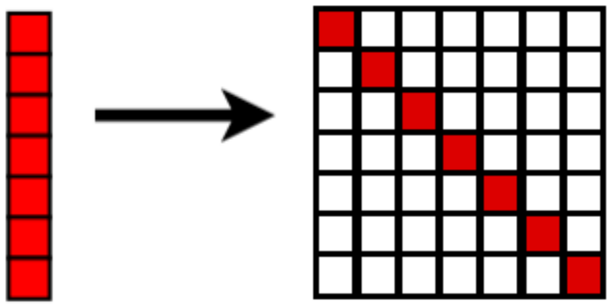
1. children have disjoint support, and
2. the union of the children elements is equal to the parent elements



Theorem [C., 2015]

h-adaptivity **converges to the high-fidelity model** if:

1. every element has a nonzero entry in >1 basis vector,
2. the root node includes all elements, and
3. each element has a leaf node.

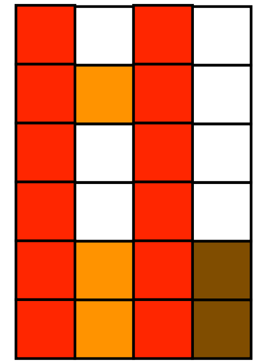


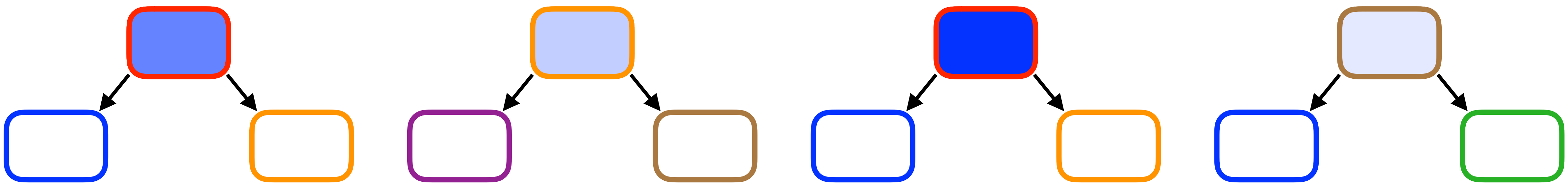
Tree-construction algorithm

- Identifies hierarchy of correlated states via *k*-means clustering
- \+ Ensures theorem conditions are satisfied

Which vectors to split?

$$\boldsymbol{\Phi}_H =$$



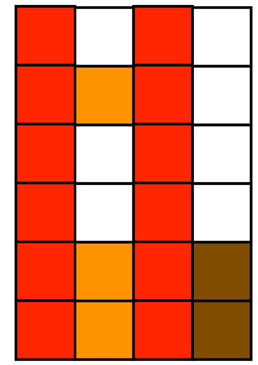


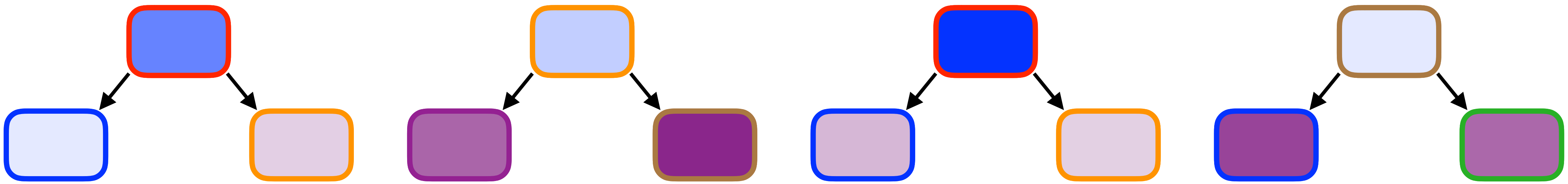
while $|\hat{\delta}^n| > \epsilon$

1. **Solve:** dual solve with coarse basis

$$\mathbf{y}_H^n = \underset{\hat{\mathbf{v}}}{\text{argmin}} \| \frac{\partial \mathbf{r}^n}{\partial \mathbf{x}} (\boldsymbol{\Phi}_H \hat{\mathbf{x}}_H^n)^T \boldsymbol{\Phi}_H \hat{\mathbf{v}} + \frac{\partial q}{\partial \mathbf{x}} (\boldsymbol{\Phi}_H \hat{\mathbf{x}}_H^n)^T \|_2$$

Which vectors to split?

$\boldsymbol{\Phi}_H =$ 



while $|\hat{\delta}^n| > \epsilon$

1. **Solve:** dual solve with coarse basis

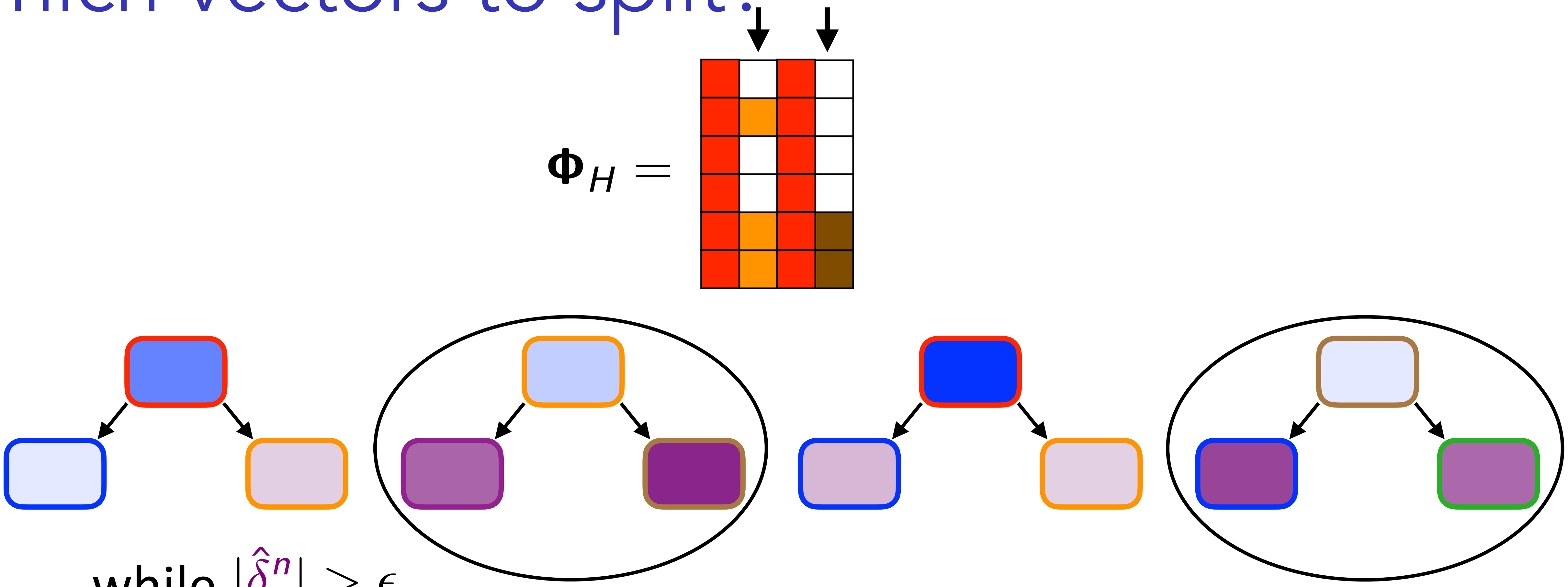
$$\mathbf{y}_H^n = \underset{\hat{\mathbf{v}}}{\text{argmin}} \| \frac{\partial \mathbf{r}^n}{\partial \mathbf{x}} (\boldsymbol{\Phi}_H \hat{\mathbf{x}}_H^n)^T \boldsymbol{\Phi}_H \hat{\mathbf{v}} + \frac{\partial q}{\partial \mathbf{x}} (\boldsymbol{\Phi}_H \hat{\mathbf{x}}_H^n)^T \|_2$$

2. **Estimate:** prolongate and compute fine error indicators

$$\Delta_i^n = |(\mathbf{I}_H^h \mathbf{y}_H^n)_i{}^T [\boldsymbol{\Phi}_h]_i{}^T \mathbf{r}^n(\boldsymbol{\Phi}_H \hat{\mathbf{x}}_H^n)|$$

Which vectors to split?

$$\boldsymbol{\Phi}_H =$$



while $|\hat{\delta}^n| > \epsilon$

1. **Solve:** dual solve with coarse basis

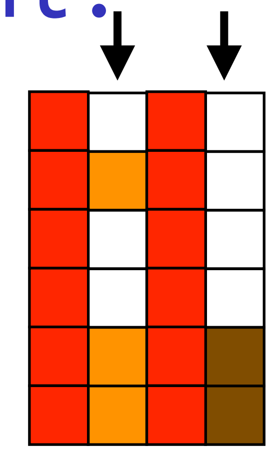
$$\mathbf{y}_H^n = \underset{\hat{\mathbf{v}}}{\text{argmin}} \| \frac{\partial \mathbf{r}^n}{\partial \mathbf{x}} (\boldsymbol{\Phi}_H \hat{\mathbf{x}}_H^n)^T \boldsymbol{\Phi}_H \hat{\mathbf{v}} + \frac{\partial q}{\partial \mathbf{x}} (\boldsymbol{\Phi}_H \hat{\mathbf{x}}_H^n)^T \|_2$$

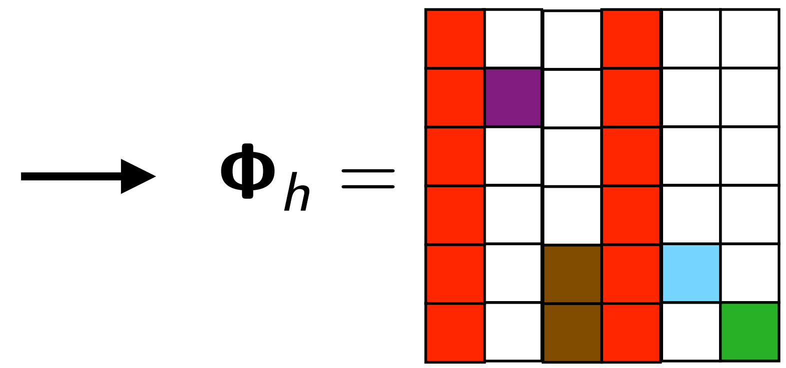
2. **Estimate:** prolongate and compute fine error indicators

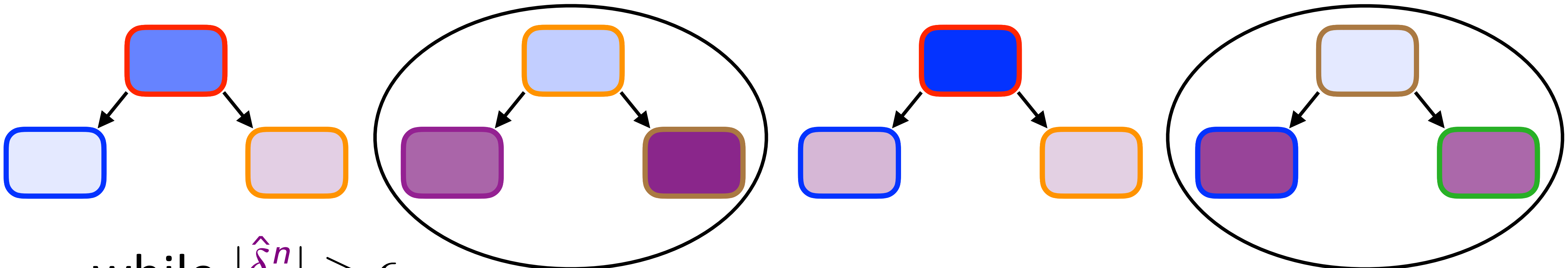
$$\Delta_i^n = |(\mathbf{I}_H^h \mathbf{y}_H^n)_i{}^T [\boldsymbol{\Phi}_h]_i{}^T \mathbf{r}^n (\boldsymbol{\Phi}_H \hat{\mathbf{x}}_H^n)|$$

3. **Mark**: identify basis vectors to refine $\{j \mid \sum_{i \in C(j)} \Delta_i^n > \tau\}$

Which vectors to split?

$\mathbf{\Phi}_H =$ 

$\mathbf{\Phi}_h =$ 



while $|\hat{\delta}^n| > \epsilon$

1. **Solve:** dual solve with coarse basis

$$\mathbf{y}_H^n = \underset{\hat{\mathbf{v}}}{\text{argmin}} \| \frac{\partial \mathbf{r}^n}{\partial \mathbf{x}}(\mathbf{\Phi}_H \hat{\mathbf{x}}_H^n)^T \mathbf{\Phi}_H \hat{\mathbf{v}} + \frac{\partial q}{\partial \mathbf{x}}(\mathbf{\Phi}_H \hat{\mathbf{x}}_H^n)^T \|_2$$

2. **Estimate:** prolongate and compute fine error indicators

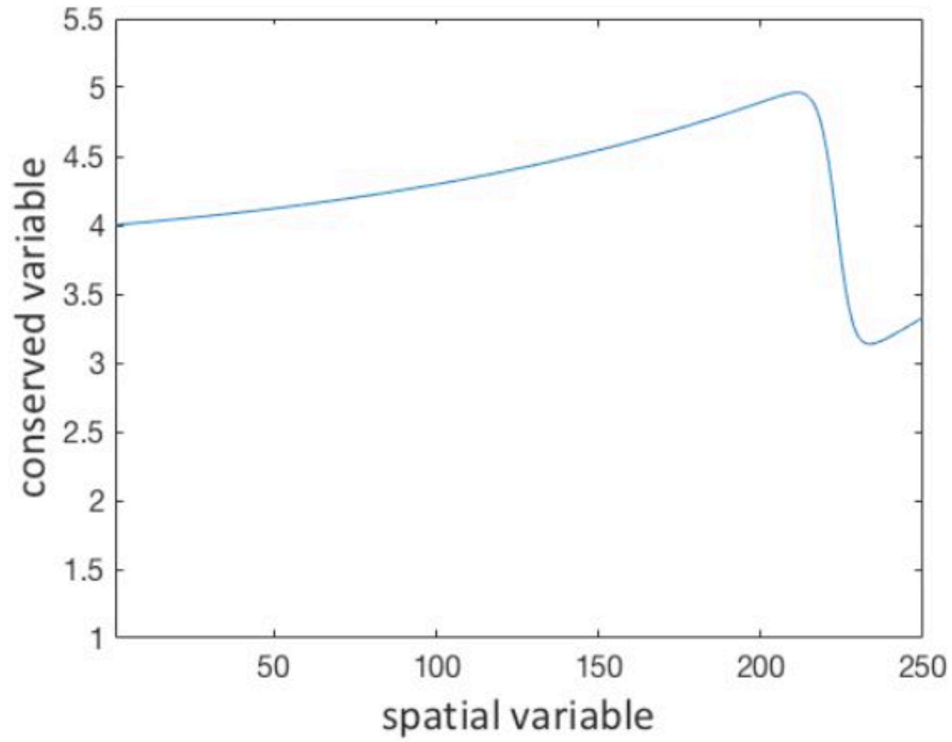
$$\Delta_i^n = |(\mathbf{I}_H^h \mathbf{y}_H^n)_i{}^T [\mathbf{\Phi}_h]_i{}^T \mathbf{r}^n(\mathbf{\Phi}_H \hat{\mathbf{x}}_H^n)|$$

3. **Mark**: identify basis vectors to refine $\{j \mid \sum_{i \in C(j)} \Delta_i^n > \tau\}$
4. **Refine**: split identified basis vectors
5. Compute solution with refined basis $\mathbf{x}_h^n = \underset{\mathbf{v} \in \text{range}(\mathbf{\Phi}_h)}{\text{argmin}} \|\mathbf{r}^n(\mathbf{v})\|_2$

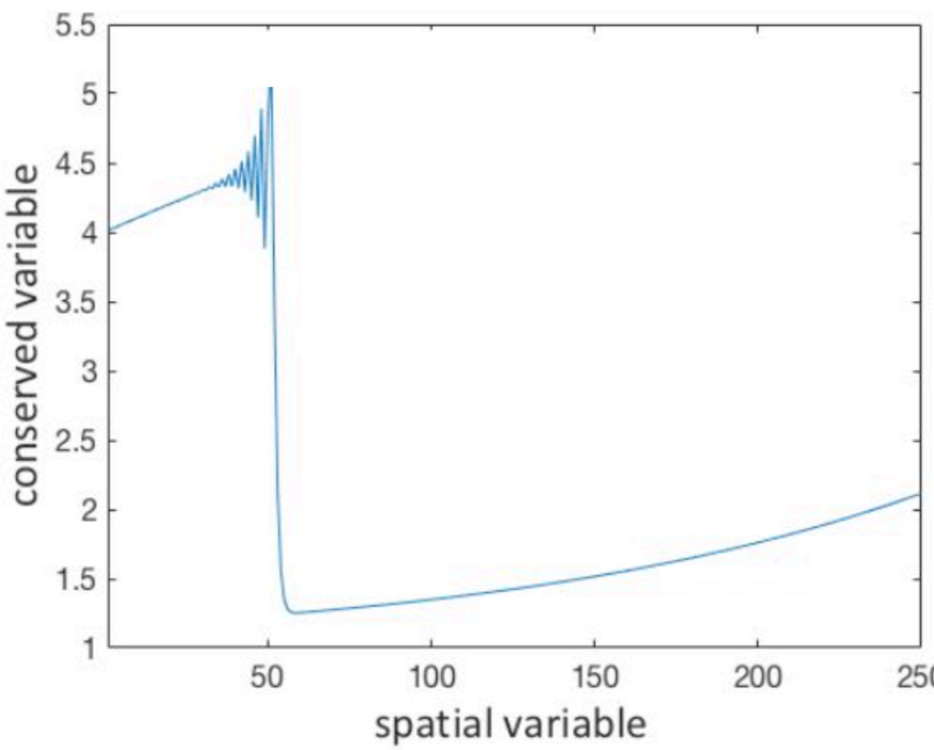
end

Illustration: inviscid 1D Burgers' equation

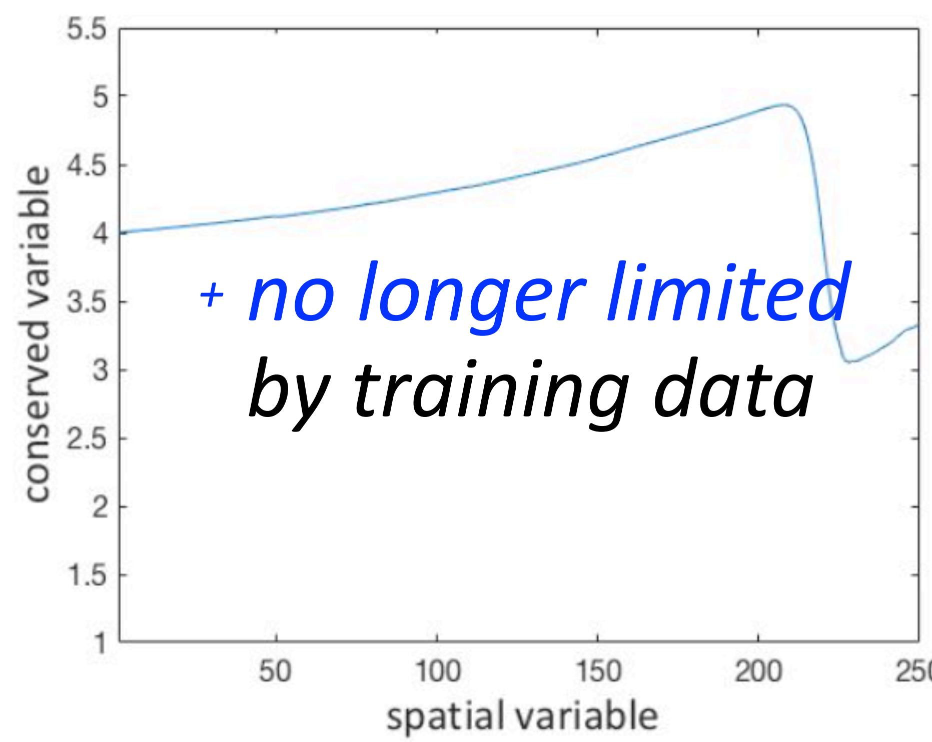
high-fidelity model



reduced-order model (dim 50)

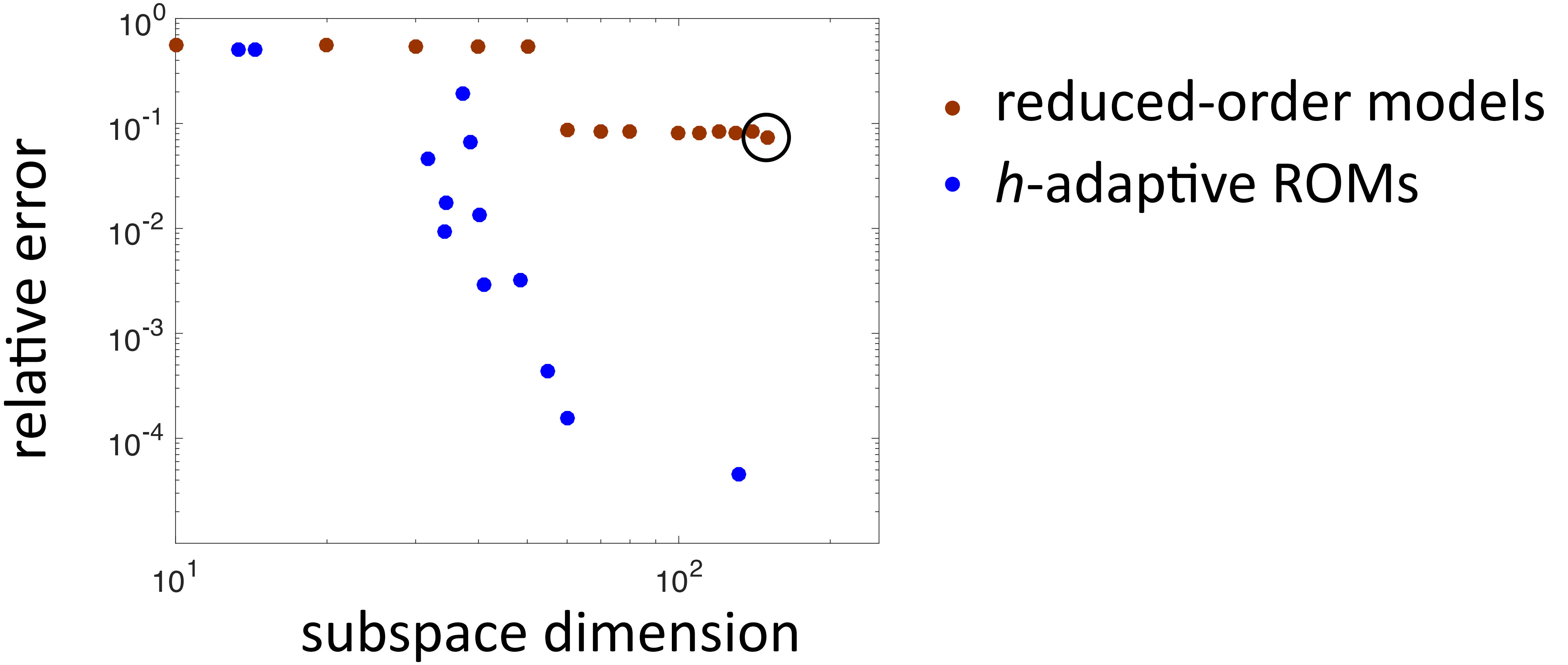


h-adaptive ROM (mean dim 48.5)



+ *no longer limited by training data*

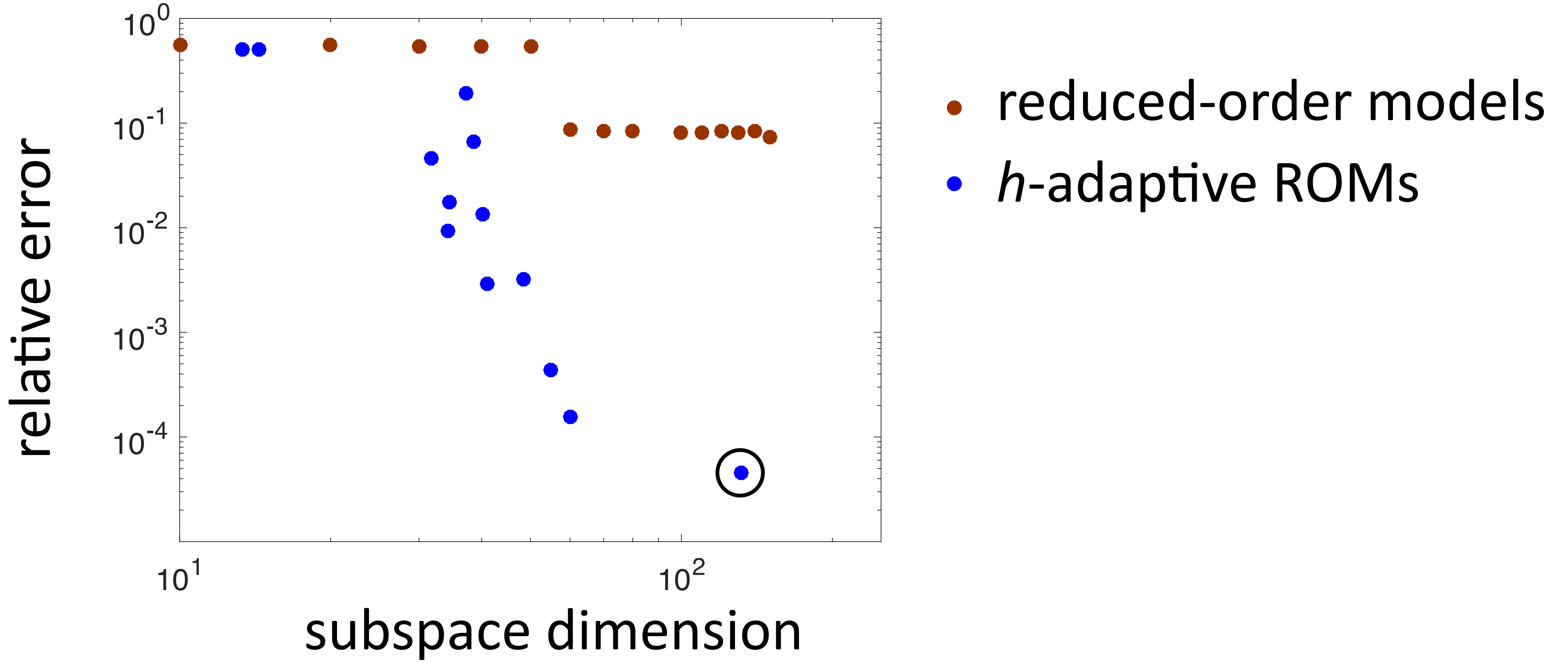
h-adaptivity provides an accurate, low-dim subspace



Reduced-order models

- minimum error 7.5%
- cannot overcome insufficient training data

h-adaptivity provides an accurate, low-dim subspace



Reduced-order models

- minimum error 7.5%
- cannot overcome insufficient training data

h-adaptive ROMs

+ minimum error <0.01% with lower subspace dimension
+ can overcome insufficient training data without collecting more data
+ can satisfy any prescribed error tolerance

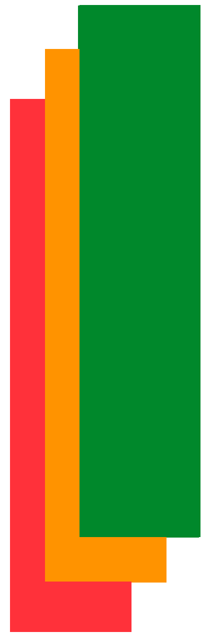
Our research

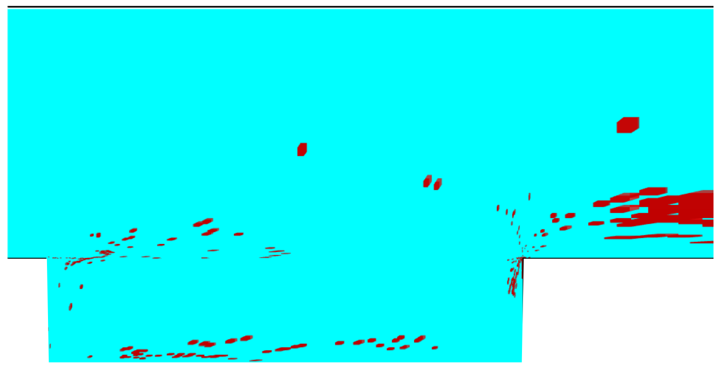
Accurate, low-cost, structure-preserving, reliable, certified nonlinear model reduction

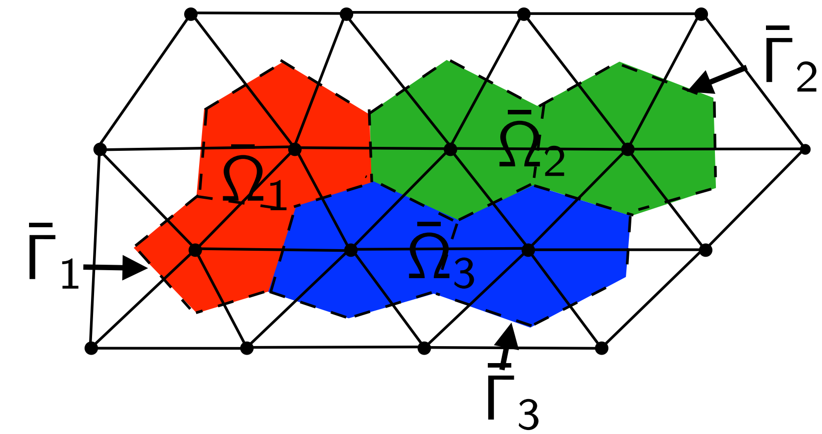
- *accuracy*: LSPG projection [C., Bou-Mosleh, Farhat, 2011; C., Barone, Antil, 2017]
- *low cost*: sample mesh [C., Farhat, Cortial, Amsallem, 2013]
- *low cost*: reduce temporal complexity
  [C., Ray, van Bloemen Waanders, 2015; C., Brencher, Haasdonk, Barth, 2017; Choi and C., 2019]
- *structure preservation* [C., Tuminaro, Boggs, 2015; Peng and C., 2017; C., Choi, Sargsyan, 2017]
- *robustness*: projection onto nonlinear manifolds [Lee, C., 2018]
- *robustness*: *h*-adaptivity [C., 2015]
- *certification*: machine learning error models
  [Drohmann and C., 2015; Trehan, C., Durlofsky, 2017; Freno and C., 2019; Pagani, Manzoni, C., 2019]

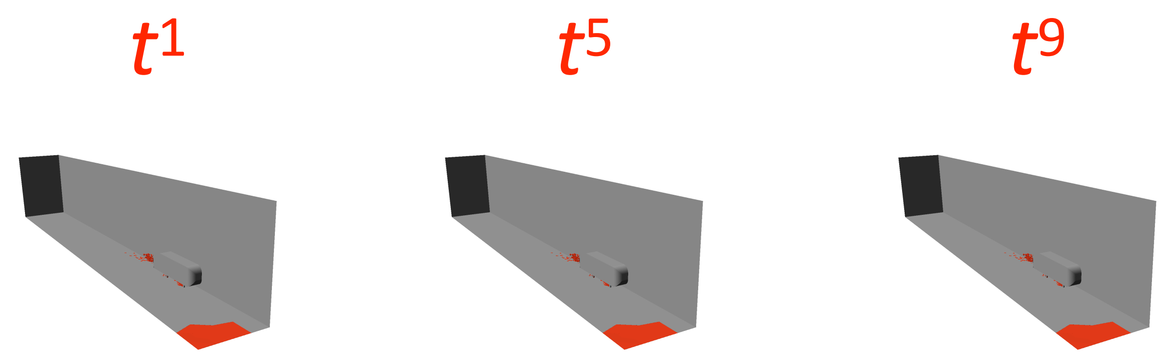
Questions?

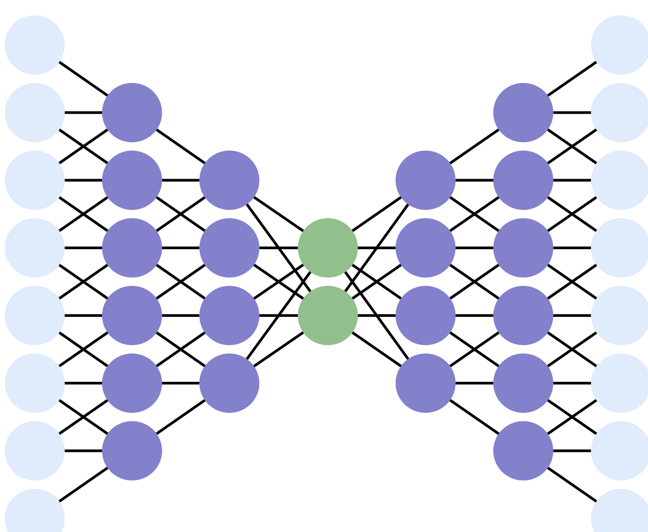
$\mathcal{X} =$

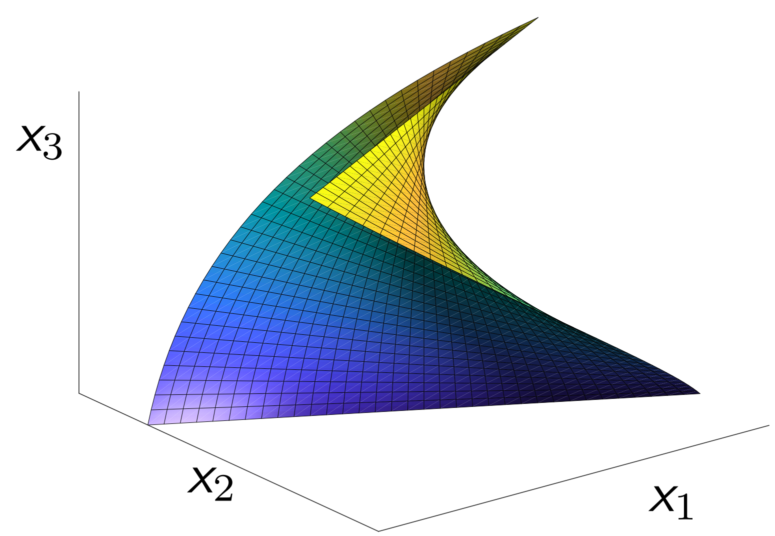


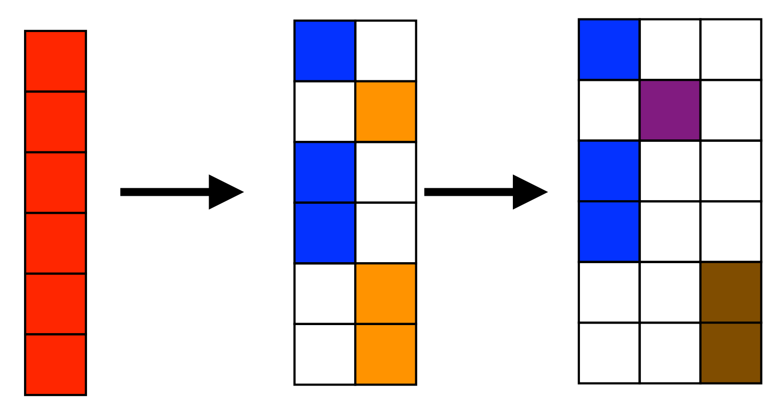












Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525