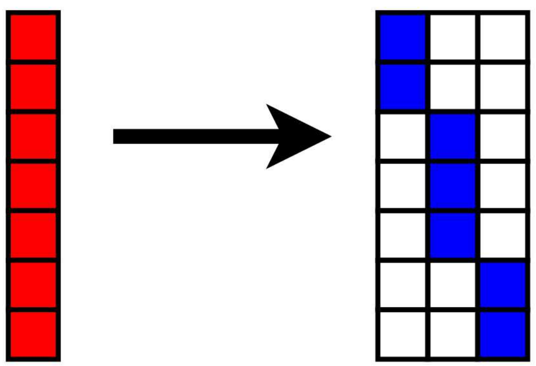
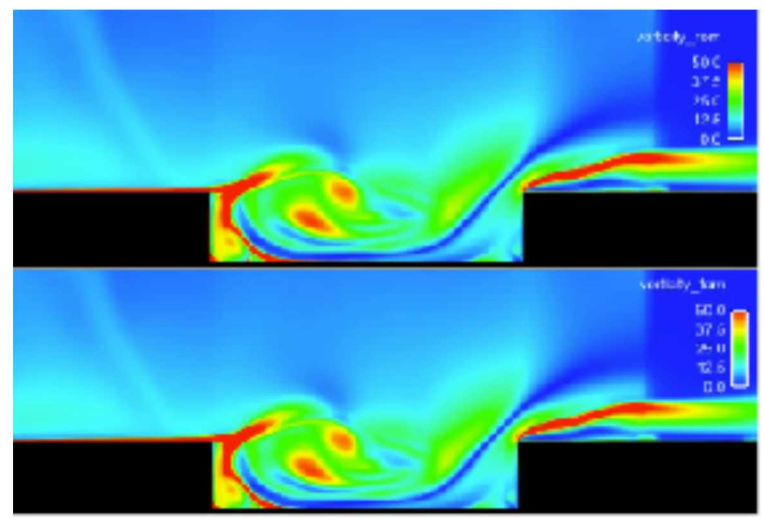
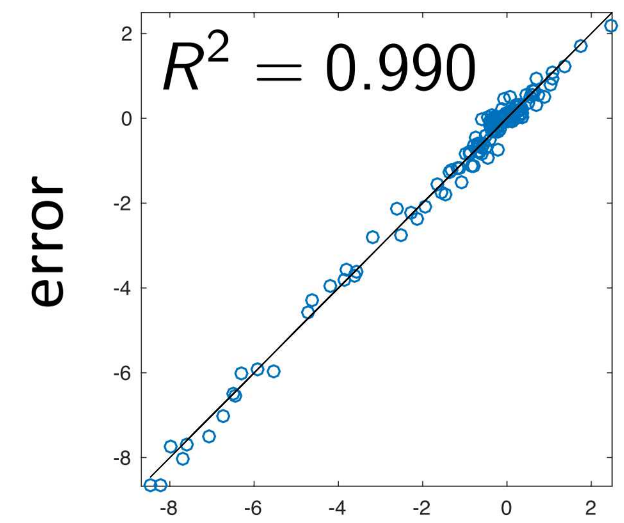
SAND2018-4639PE

Nonlinear reduced-order modeling

Enabling large-scale physics-based simulations for real-time and many-query problems







support vector machine
error prediction

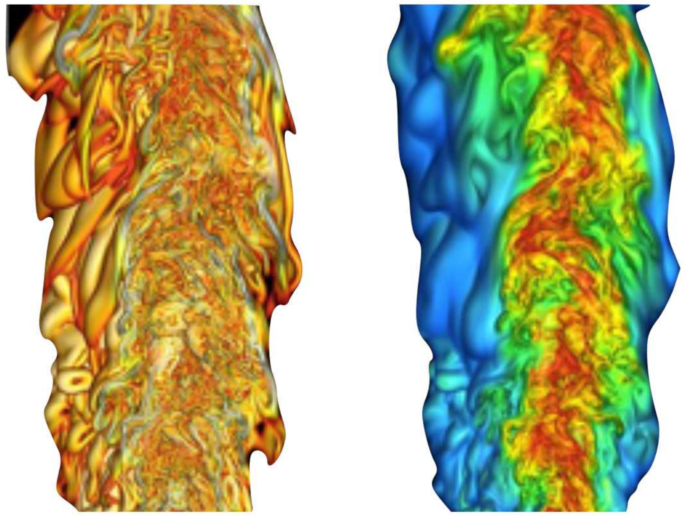
Kevin Carlberg

Sandia National Laboratories

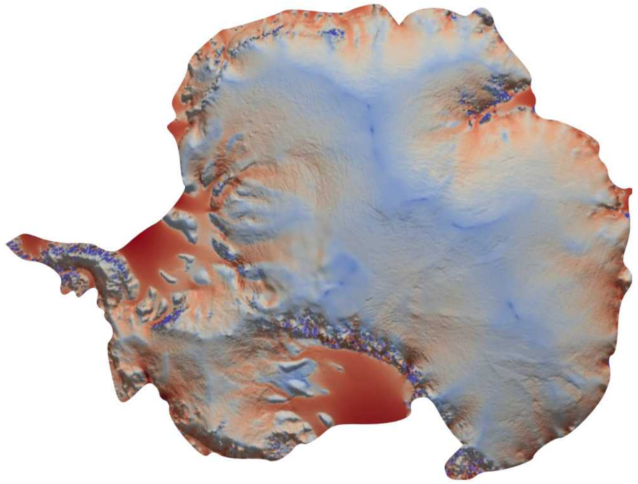
Pixar Research Group Seminar
Emeryville, California
April 24, 2018

High-fidelity simulation

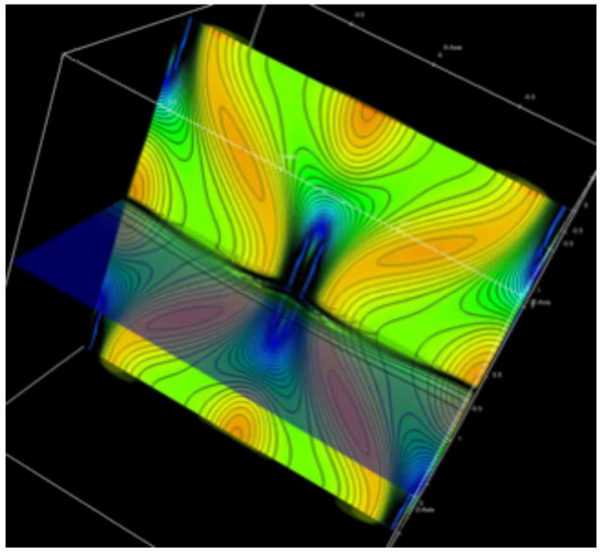
+ Indispensable across science and engineering
- *High fidelity*: extreme-scale nonlinear dynamical system models



Turbulent reacting flows
courtesy J. Chen, Sandia



Antarctic ice sheet modeling
courtesy R. Tuminaro, Sandia



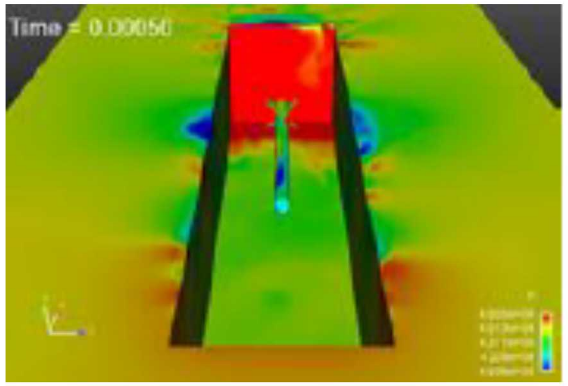
Magnetohydrodynamics
courtesy J. Shadid, Sandia

computational barrier

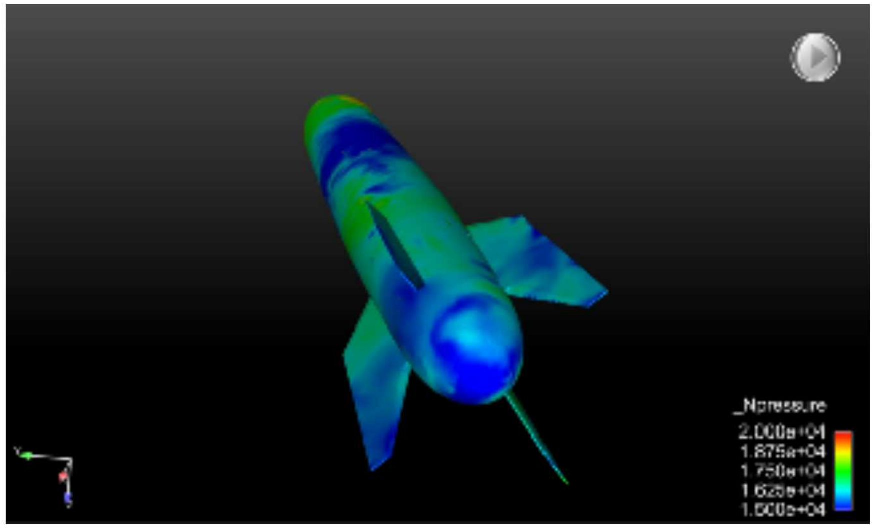
Many-query and real-time problems

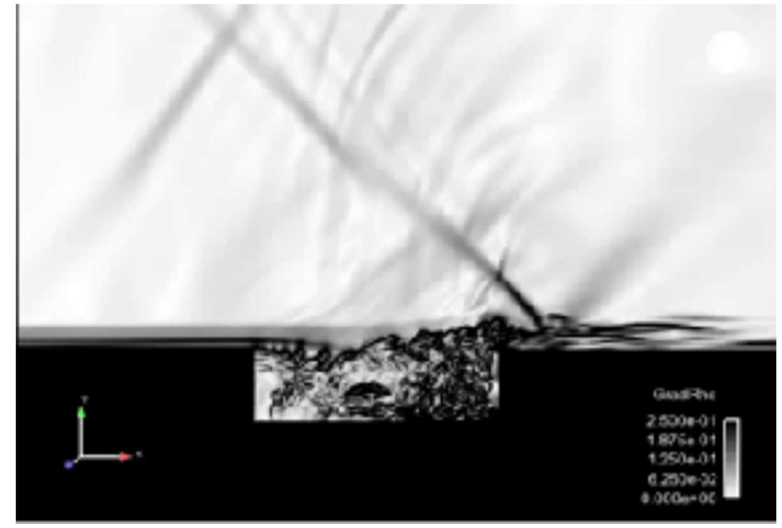
- uncertainty propagation
- Bayesian inference
- fast-turnaround simulation
- in-the-field analysis

High-fidelity simulation: B61 LEP captive carry



High-fidelity simulation: B61 LEP captive carry





+ *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

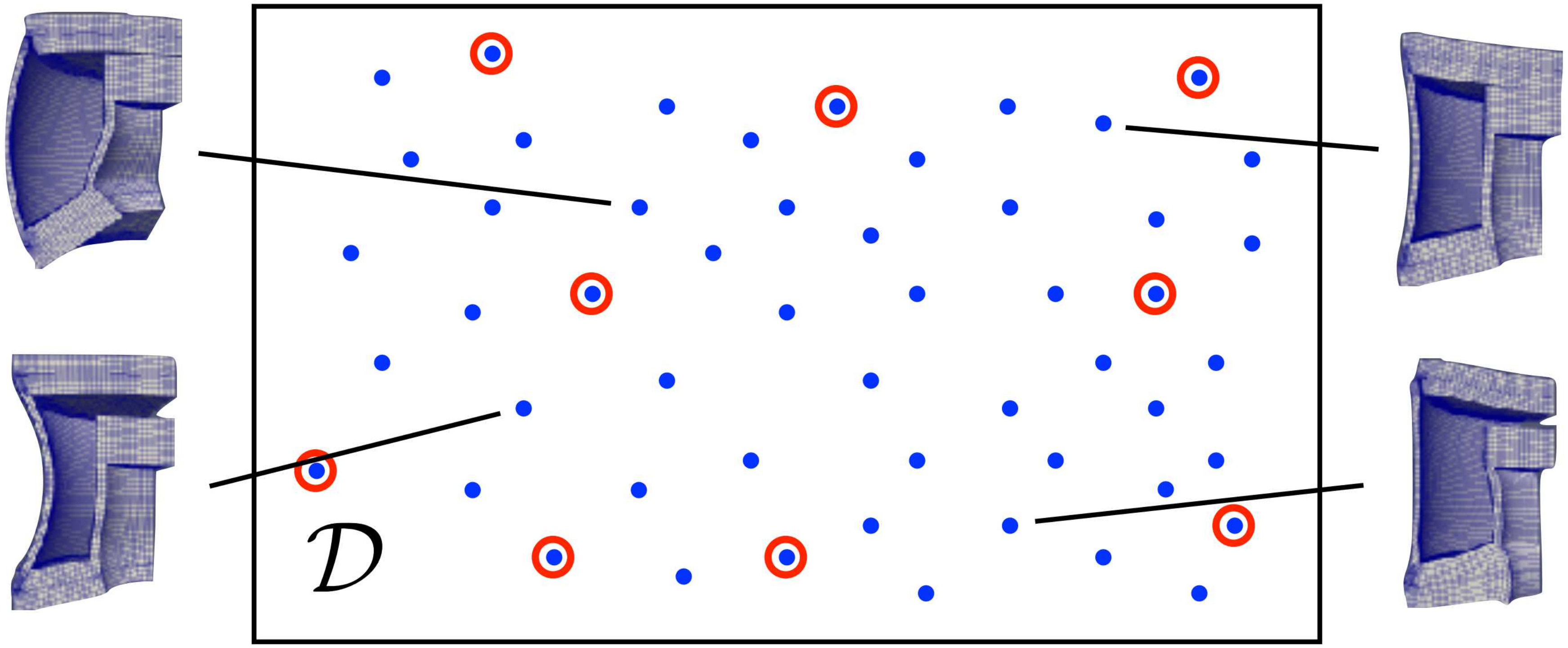
Many-query and real-time problems

- explore flight envelope
- quantify effects of uncertainties on store load
- model predictive control

Approach: exploit simulation data

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

Many-query/real-time 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 savings
3. ***Structure preservation:*** preserves important physical properties
4. ***Reliability:*** guaranteed satisfaction of any error tolerance (fail safe)
5. ***Certification:*** quantifies ROM-induced epistemic uncertainty

Model reduction: previous state of the art

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

Model reduction: previous state of the art

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

Model reduction: previous state of the art

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*: dynamical-system properties ignored

My research

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

- *accuracy*: LSPG projection [Carlberg, Bou-Mosleh, Farhat, 2011; Carlberg, Antil, Barone, 2017]
- *low cost*: sample mesh [Carlberg, Farhat, Cortial, Amsallem, 2013]
- *low cost*: reduce temporal complexity
  [Carlberg, Ray, van Bloemen Waanders, 2015; Carlberg, Brencher, Haasdonk, Barth, 2017; Choi and Carlberg, 2017]
- *structure preservation* [Carlberg, Tuminaro, Boggs, 2015; Peng and Carlberg, 2017; Carlberg and Choi, 2017]
- *reliability*: adaptivity [Carlberg, 2015]
- *certification*: machine learning error models
  [Drohmann and Carlberg, 2015; Trehan, Carlberg, Durlofsky, 2017; Freno and Carlberg, 2017]

My research

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

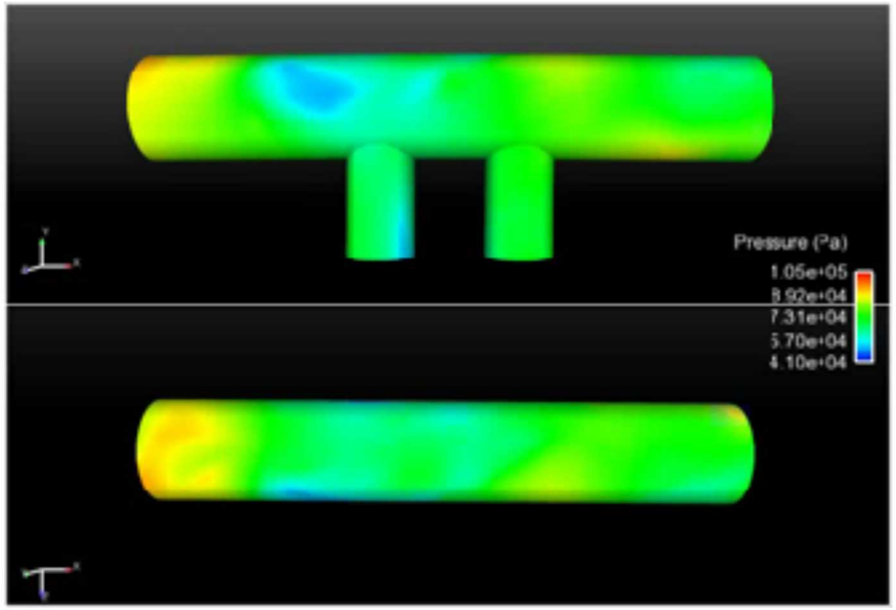
- *accuracy*: LSPG projection [Carlberg, Bou-Mosleh, Farhat, 2011*; Carlberg, Antil, Barone, 2017]
- *low cost*: sample mesh [Carlberg, Farhat, Cortial, Amsallem, 2013]
- *low cost*: reduce temporal complexity
  [Carlberg, Ray, van Bloemen Waanders, 2015; Carlberg, Brencher, Haasdonk, Barth, 2017; Choi and Carlberg, 2017]
- *structure preservation* [Carlberg, Tuminaro, Boggs, 2015; Peng and Carlberg, 2017; Carlberg and Choi, 2017]
- *reliability*: adaptivity [Carlberg, 2015]
- *certification*: machine learning error models
  [Drohmann and Carlberg, 2015; Trehan, Carlberg, Durlofsky, 2017; Freno and Carlberg, 2017]

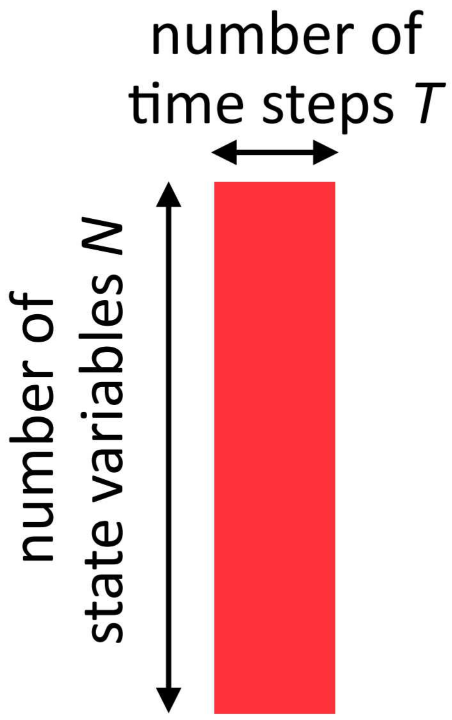
* #2 most-cited paper, Int J Numer Meth Eng, 2011

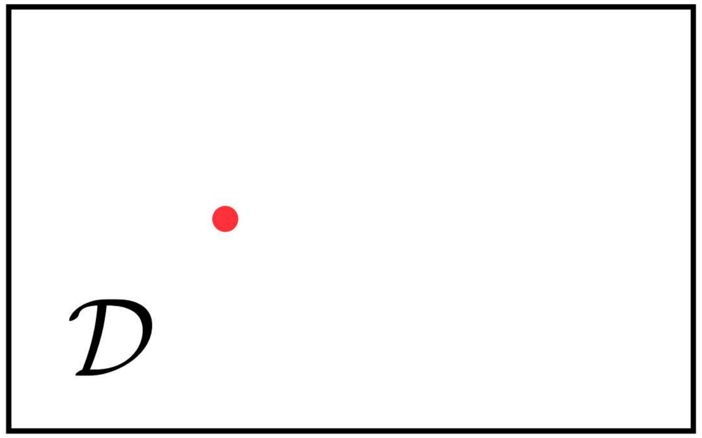
Training simulations: state tensor

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

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 $\boldsymbol{\mu} \in \mathcal{D}_{\text{query}} \setminus \mathcal{D}_{\text{training}}$



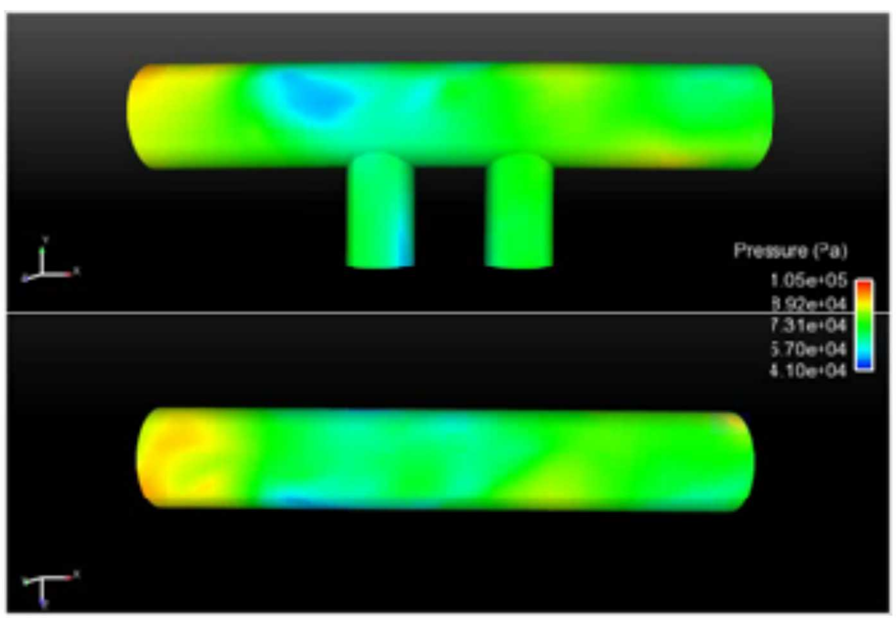




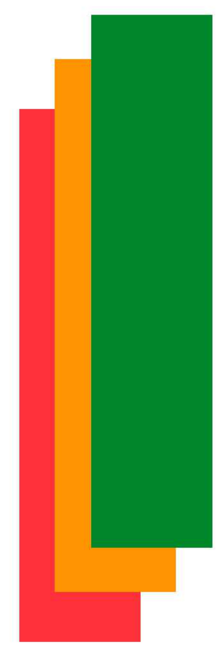
Training simulations: state tensor

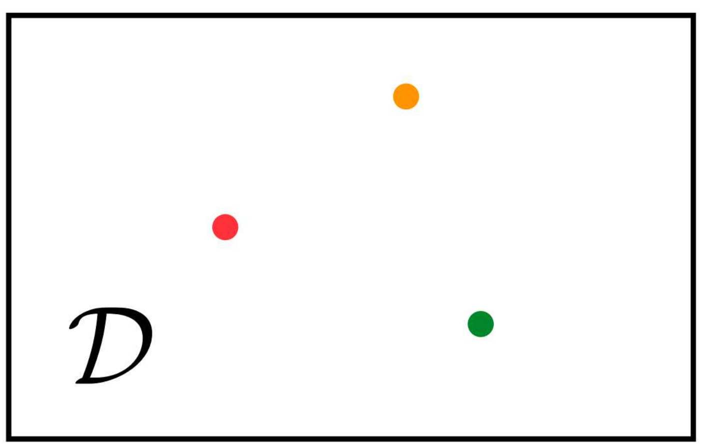
$$\text{ODE:} \qquad \frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}; t, \boldsymbol{\mu})$$

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 $\boldsymbol{\mu} \in \mathcal{D}_{\text{query}} \setminus \mathcal{D}_{\text{training}}$



$\mathcal{X} =$



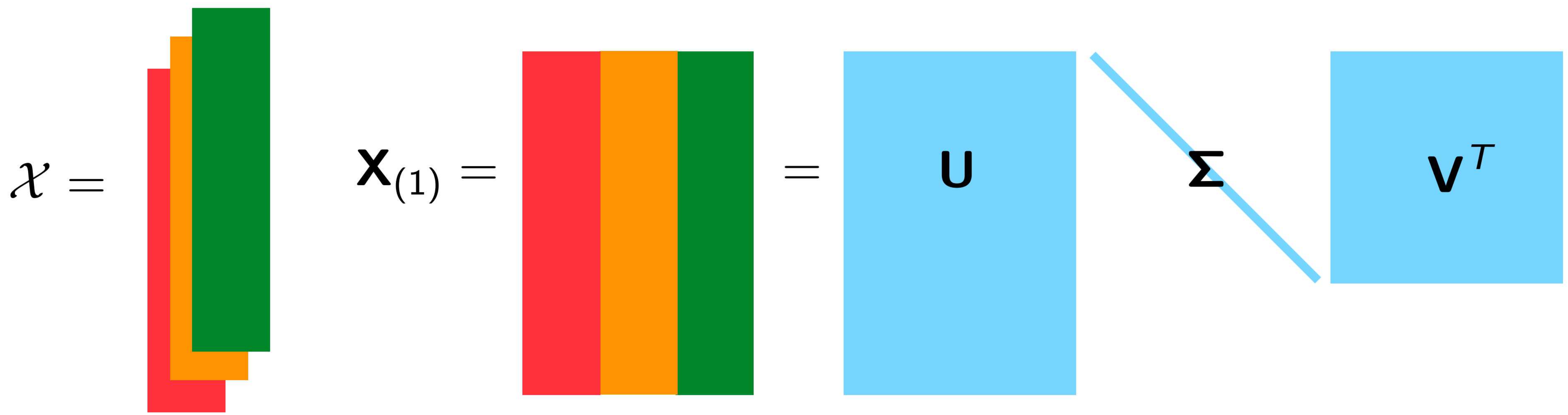


Tensor decomposition

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

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

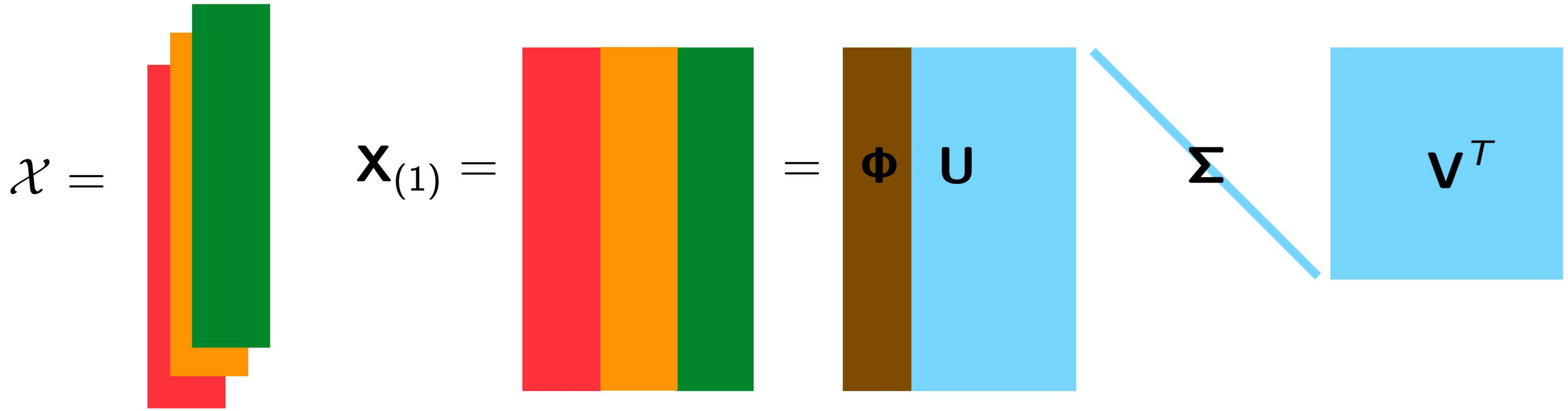


Tensor decomposition

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

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

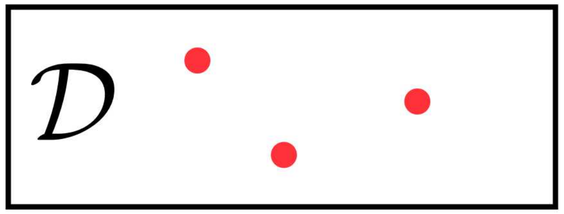


$\boldsymbol{\Phi}$ *columns are principal components of the spatial simulation data*

How to integrate these data with the computational model?

Previous state of the art: POD–Galerkin

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



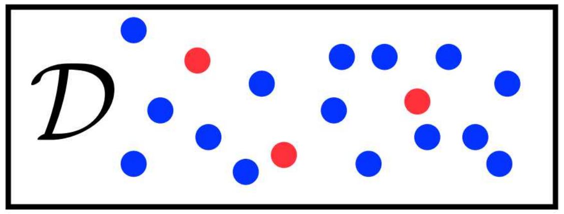
1. *Training:* Solve ODE for $\mu \in \mathcal{D}_{\text{training}}$ and collect simulation data
2. *Machine learning:* Identify structure in data
3. *Reduction:* Reduce the cost of solving ODE for $\boldsymbol{\mu} \in \mathcal{D}_{\text{query}} \setminus \mathcal{D}_{\text{training}}$

1. Reduce the number of unknowns
2. Reduce the number of equations

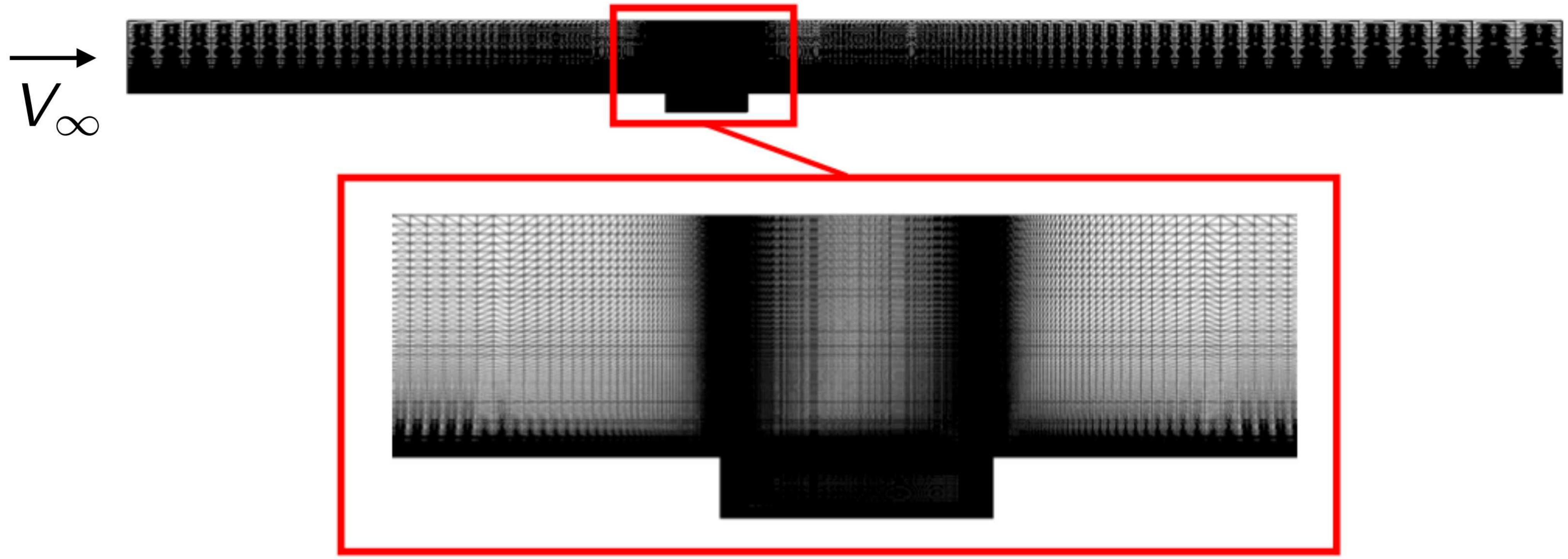
$$\mathbf{x}(t) \approx \tilde{\mathbf{x}}(t) = \boldsymbol{\Phi}\,\hat{\mathbf{x}}(t)$$

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

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



B61 captive carry



- Unsteady Navier–Stokes
- Re = 6.3 x 10^6
- M_∞ = 0.6

Spatial discretization

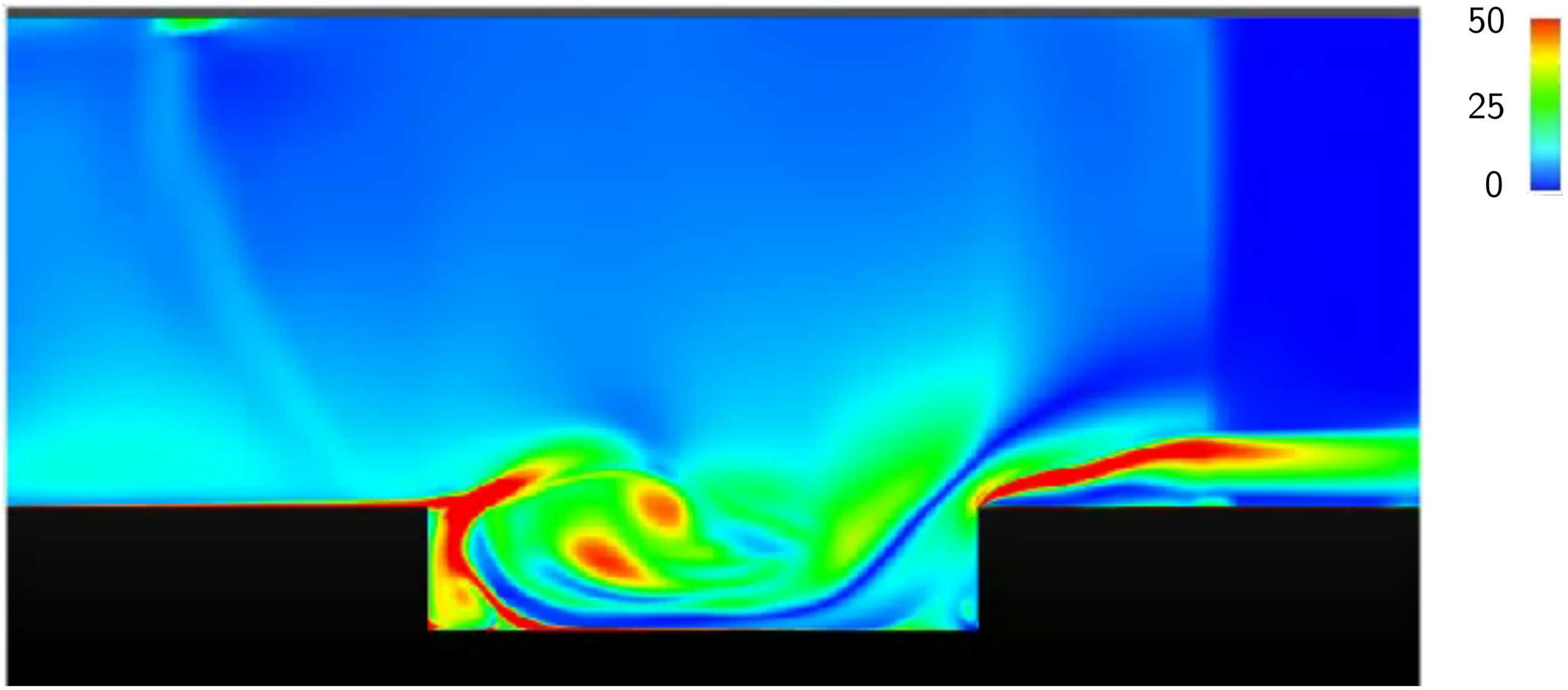
- 2nd-order finite volume
- DES turbulence model
- 1.2×10^6 degrees of freedom

Temporal discretization

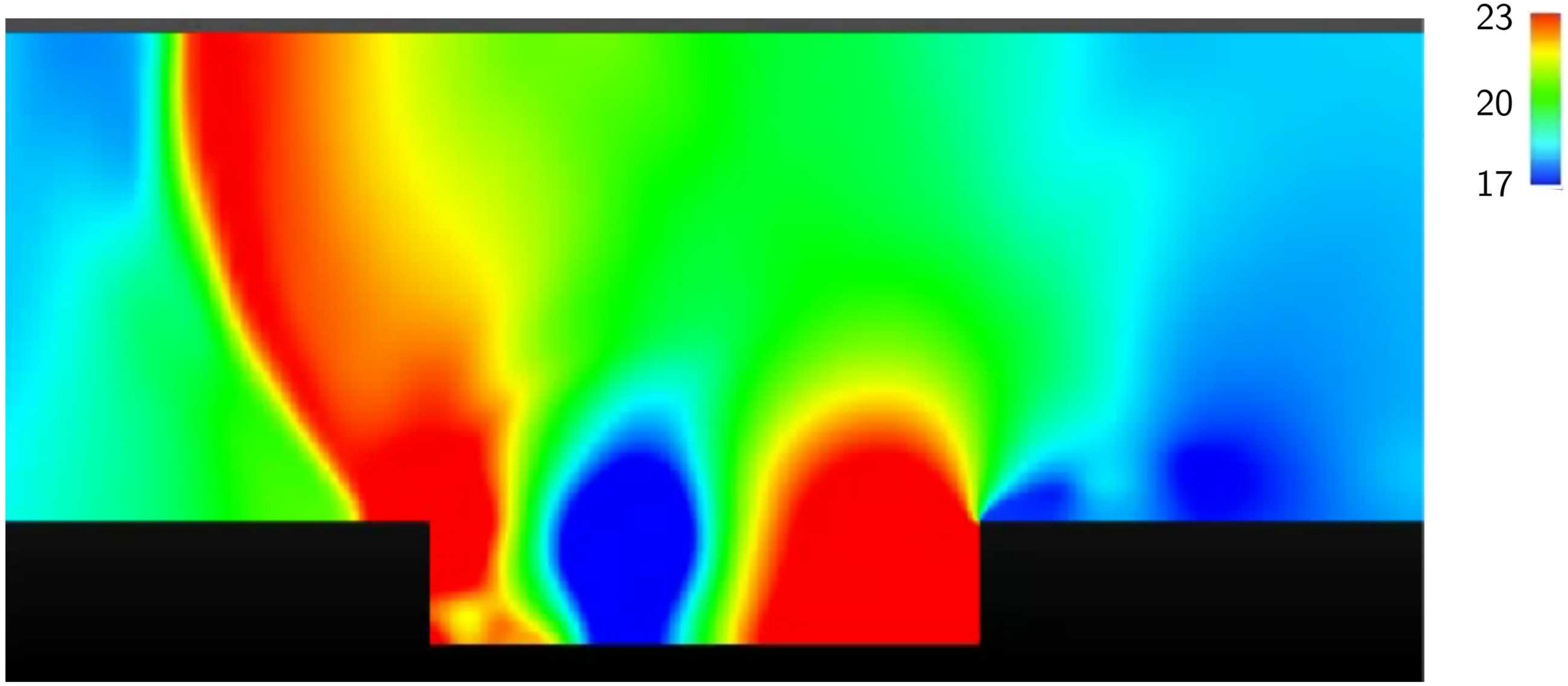
- 2nd-order BDF
- Verified time step $\Delta t = 1.5 \times 10^{-3}$
- 8.3×10^3 time instances

High-fidelity model solution

vorticity field

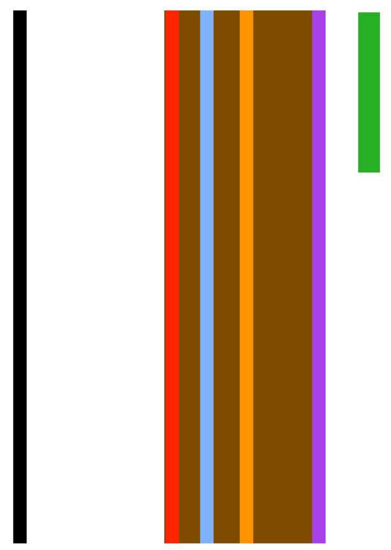


pressure field

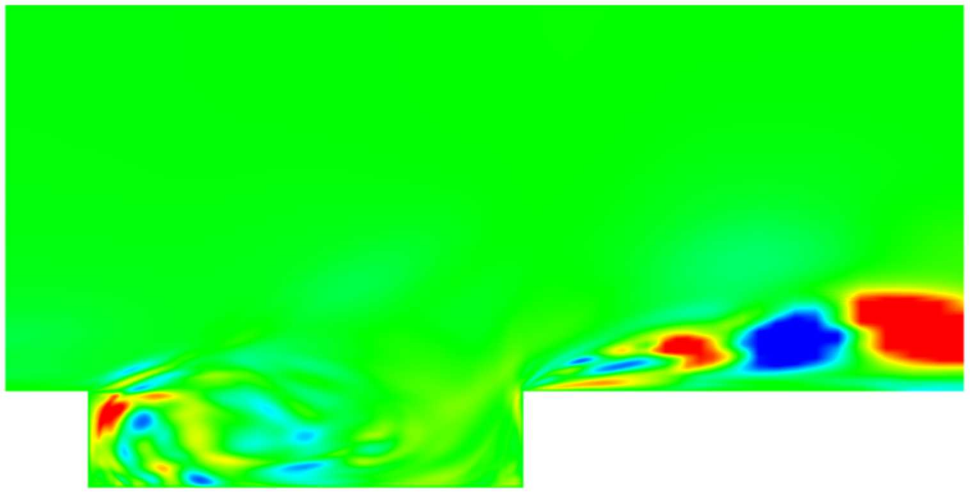


Principal components

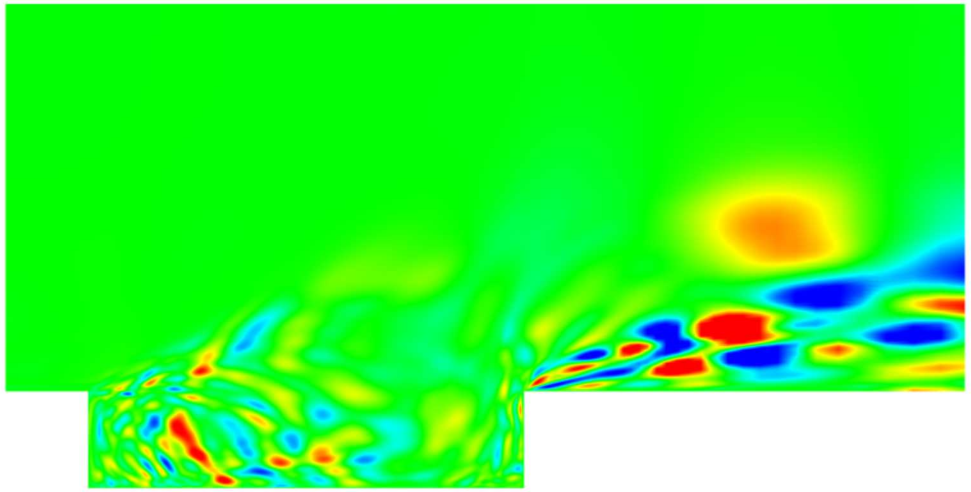
$\mathbf{x}(t) \approx \boldsymbol{\Phi}\ \hat{\mathbf{x}}(t)$



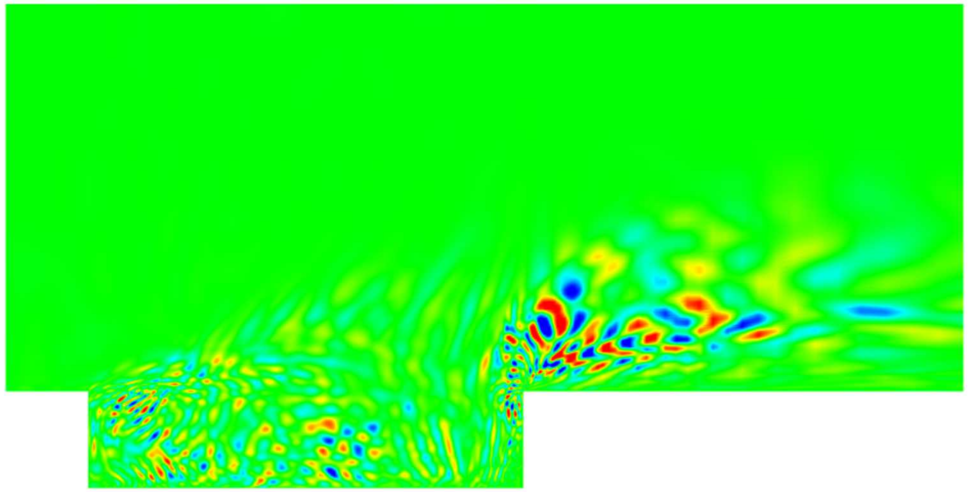
ϕ_1



ϕ_{21}

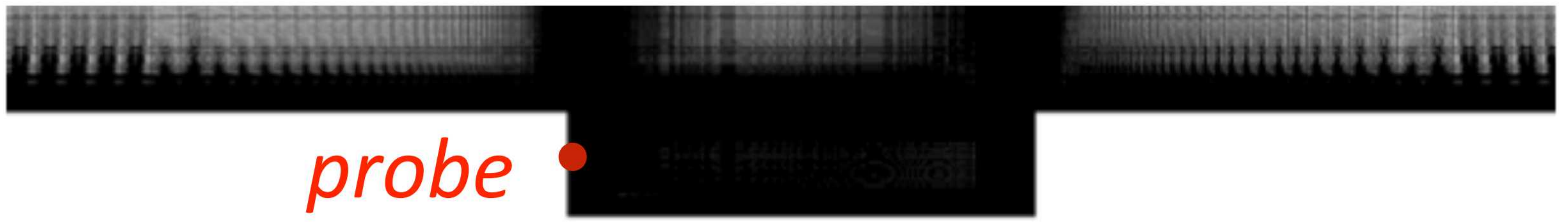


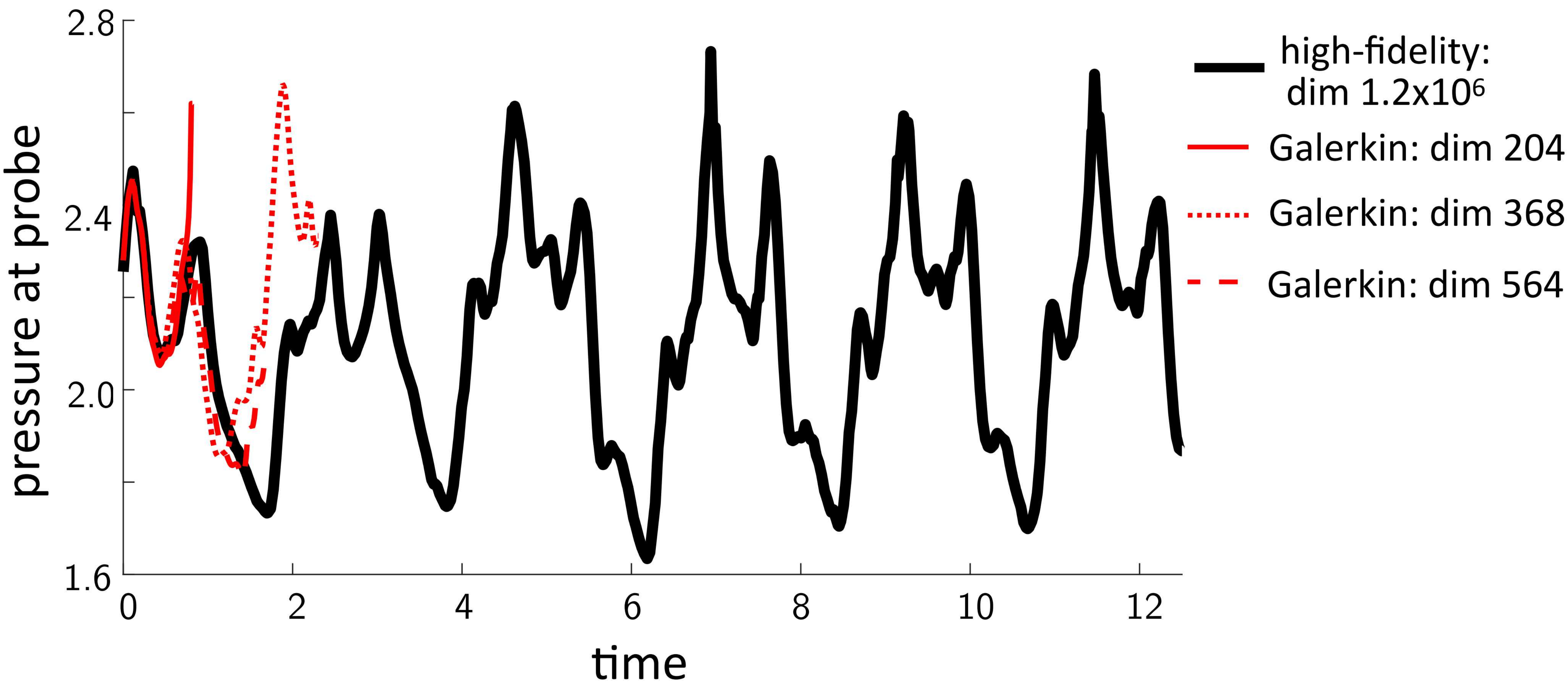
ϕ_{101}



ϕ_{401}

Galerkin performance





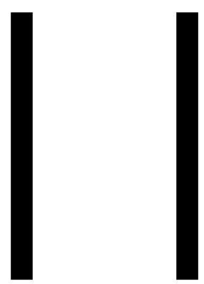
- *Galerkin projection fails regardless of basis dimension*

Can we construct a better projection?

Galerkin: time-continuous optimality

ODE

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



Galerkin ODE

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

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

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

OΔE

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

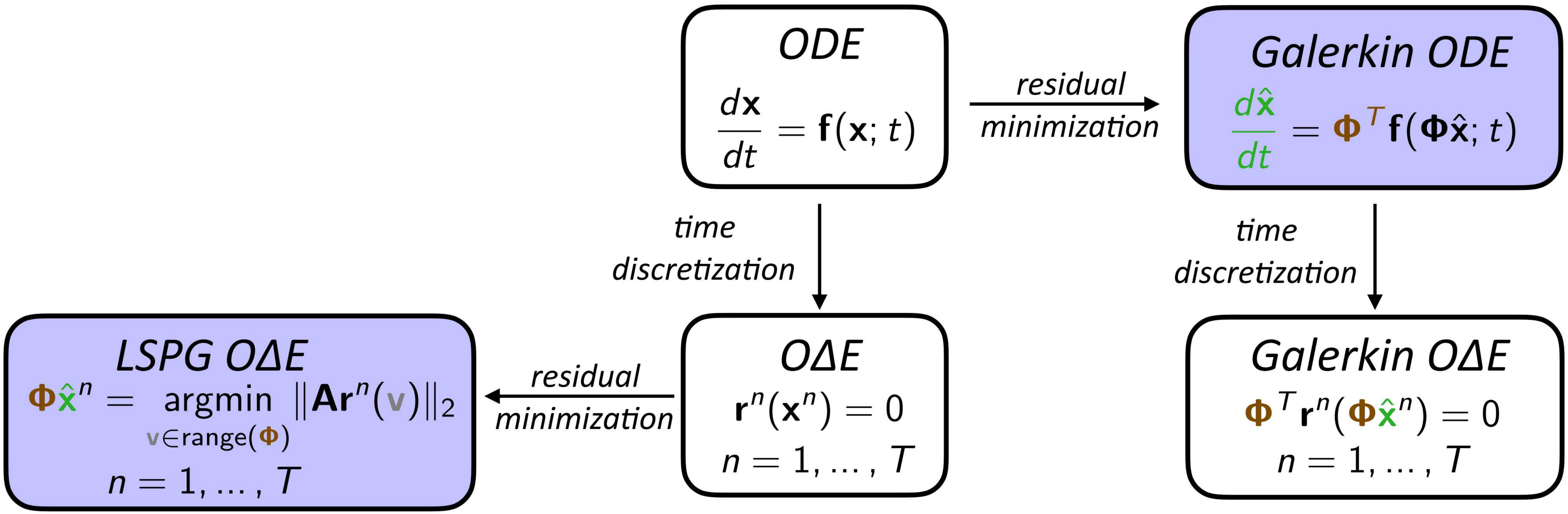
Galerkin OΔE

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

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

- *Time-discrete Galerkin solution*: not generally optimal in any sense

Residual minimization and time discretization



[Carlberg, Bou-Mosleh, Farhat, 2011]

$$\mathbf{\Phi}\hat{\mathbf{x}}^n = \underset{\mathbf{v}\in \mathrm{range}(\mathbf{\Phi})}{\mathrm{argmin}} \|\mathbf{A}\mathbf{r}^n(\mathbf{v})\|_2 \quad \Leftrightarrow \quad \mathbf{\Psi}^n(\hat{\mathbf{x}}^n)^T\mathbf{r}^n(\mathbf{\Phi}\hat{\mathbf{x}}^n) = 0$$

$$\mathbf{\Psi}^n(\hat{\mathbf{x}}^n) := \mathbf{A}^T\mathbf{A}\left(\alpha_0\mathbf{I} - \Delta t\beta_0\frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{\Phi}\hat{\mathbf{x}}^n; t)\right)\mathbf{\Phi}$$

*Least-squares Petrov–Galerkin (**LSPG**) projection*

Discrete-time error bound

Theorem [Carlberg, Antil, Barone, 2017]

If the following conditions hold:

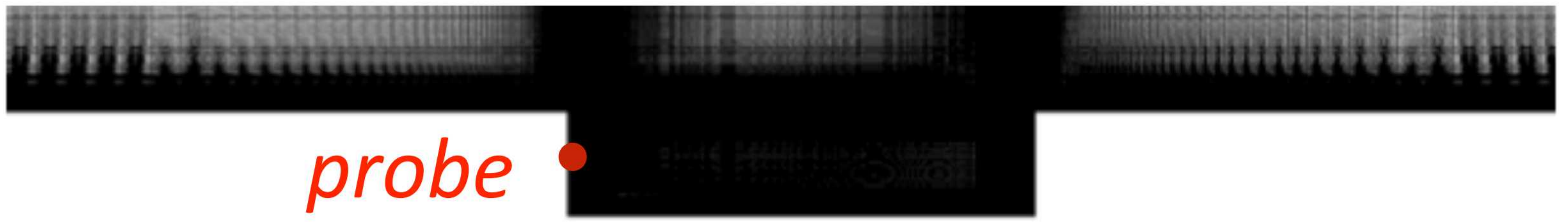
1. $\mathbf{f}(\cdot; t)$ is Lipschitz continuous with Lipschitz constant κ
2. The time step Δt is small enough such that $0 < h := |\alpha_0| - |\beta_0|\kappa\Delta t$,
3. A backward differentiation formula (BDF) time integrator is used,
4. LSPG employs $\mathbf{A} = \mathbf{I}$, then

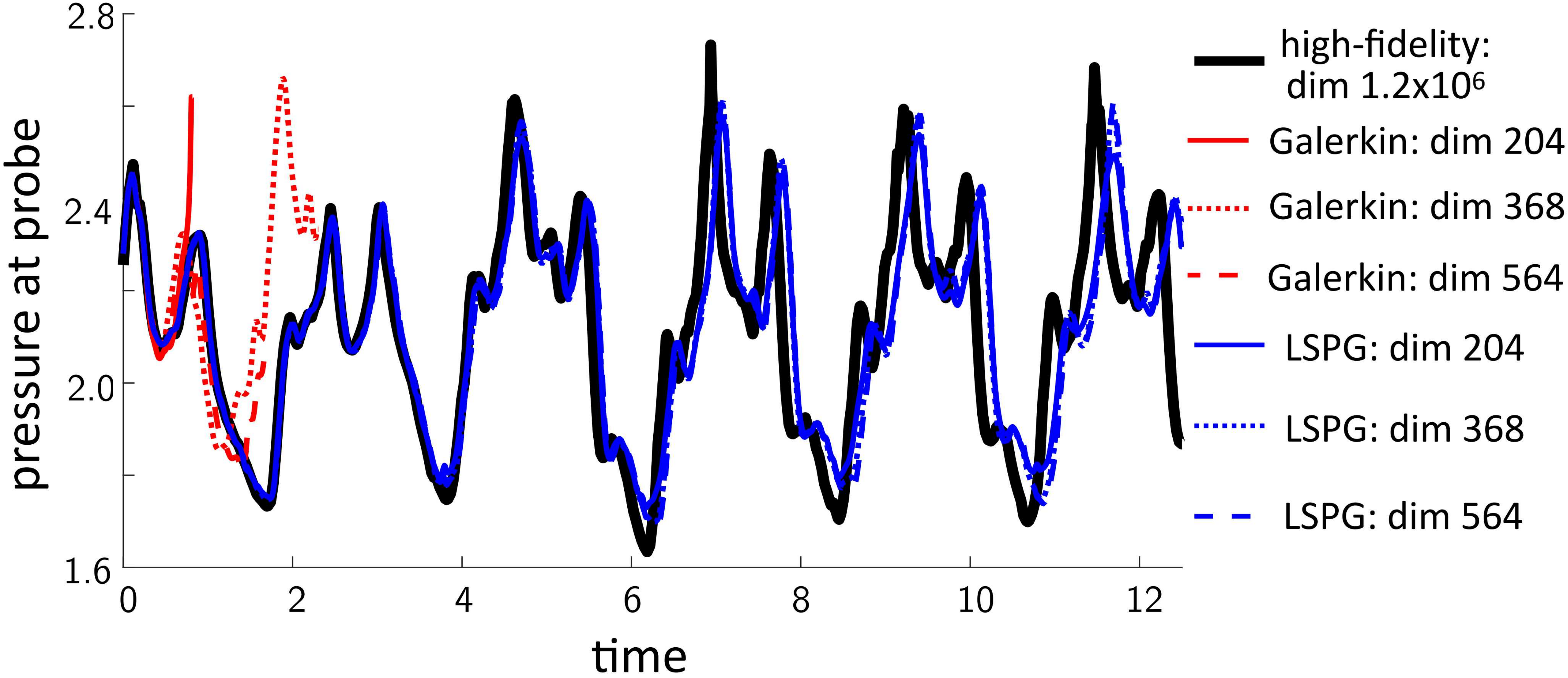
$$\|\mathbf{x}^n - \mathbf{\Phi}\hat{\mathbf{x}}_{\mathrm{G}}^n\|_2 \leq \frac{1}{h}\|\mathbf{r}_{\mathrm{G}}^n(\mathbf{\Phi}\hat{\mathbf{x}}_{\mathrm{G}}^n)\|_2 + \frac{1}{h}\sum_{\ell=1}^{k}|\alpha_\ell|\|\mathbf{x}^{n-\ell} - \mathbf{\Phi}\hat{\mathbf{x}}_{\mathrm{G}}^{n-\ell}\|_2$$

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

+ LSPG sequentially minimizes the error bound

LSPG performance





+ LSPG is far more accurate than Galerkin

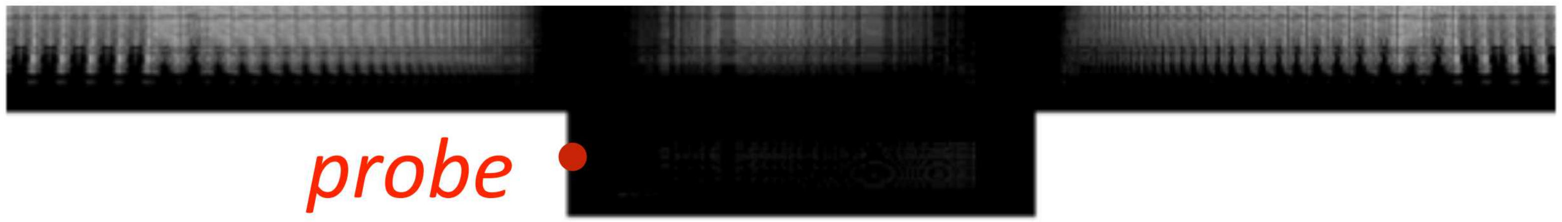
My research

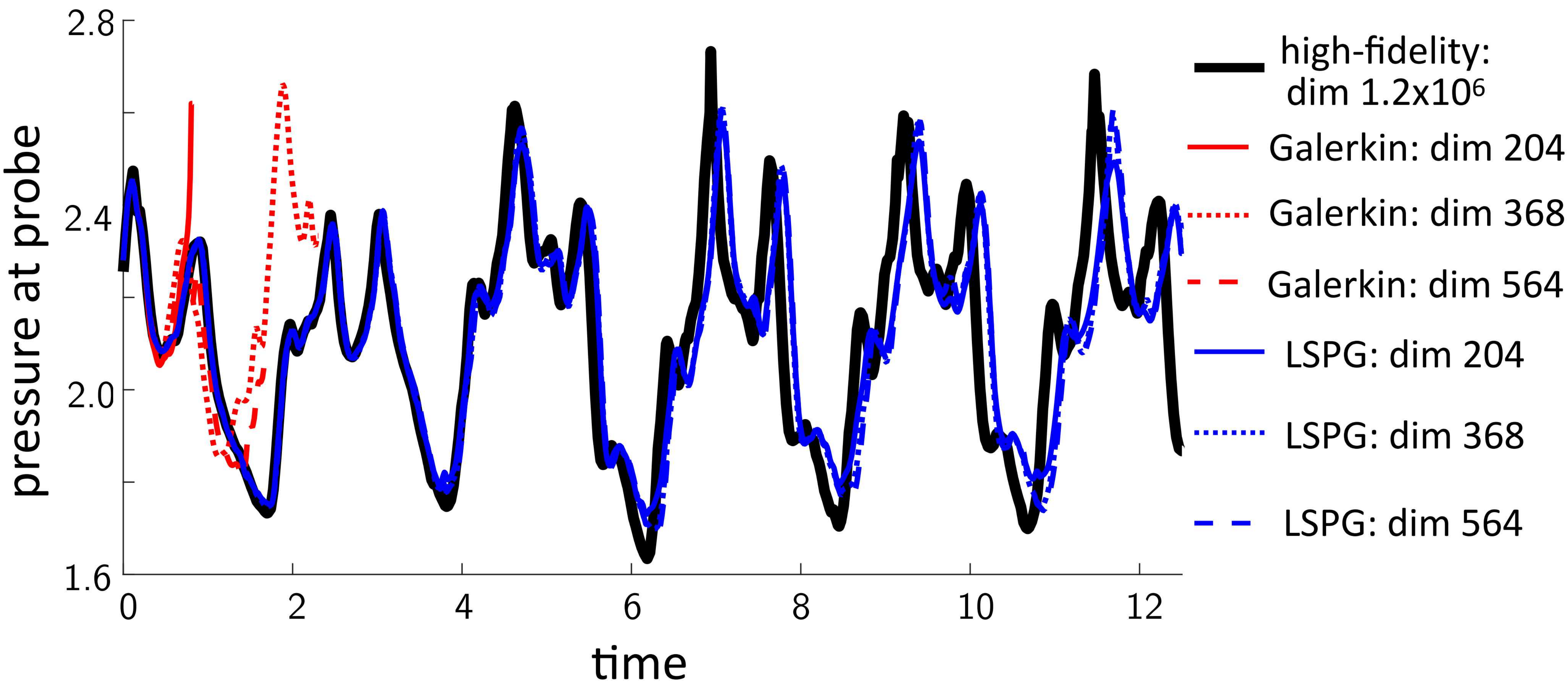
Accurate, low-cost, structure-preserving, reliable, certified nonlinear model reduction

- *accuracy*: LSPG projection [Carlberg, Bou-Mosleh, Farhat, 2011; Carlberg, Antil, Barone, 2017]
- *low cost*: sample mesh [Carlberg, Farhat, Cortial, Amsallem, 2013*]
- *low cost*: reduce temporal complexity [Carlberg, Ray, van Bloemen Waanders, 2015; Carlberg, Brencher, Haasdonk, Barth, 2017; Choi and Carlberg, 2017]
- *structure preservation* [Carlberg, Tuminaro, Boggs, 2015; Peng and Carlberg, 2017; Carlberg and Choi, 2017]
- *reliability*: adaptivity [Carlberg, 2015]
- *certification*: machine learning error models [Drohmann and Carlberg, 2015; Trehan, Carlberg, Durlofsky, 2017; Freno and Carlberg, 2017]

* #1 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

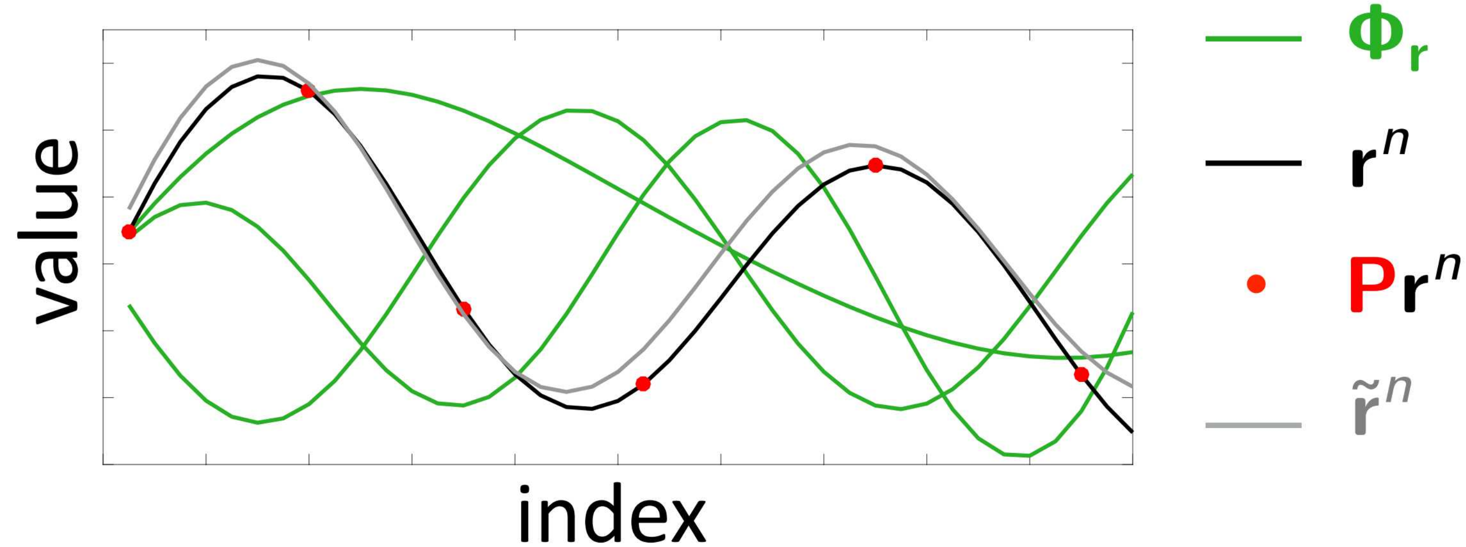
Why does this occur? Can we fix it?

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

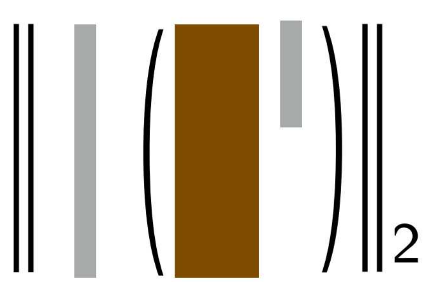
$$\underset{\hat{\mathbf{v}}}{\text{minimize}} \| \mathbf{A} \, \mathbf{r}^n(\boldsymbol{\Phi}\hat{\mathbf{v}}) \|_2$$

Can we select **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 $\mathbf{r}^n \approx \tilde{\mathbf{r}}^n = \boldsymbol{\Phi}_{\mathbf{r}}(\mathbf{P}\boldsymbol{\Phi}_{\mathbf{r}})^+\mathbf{P}\mathbf{r}^n$



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

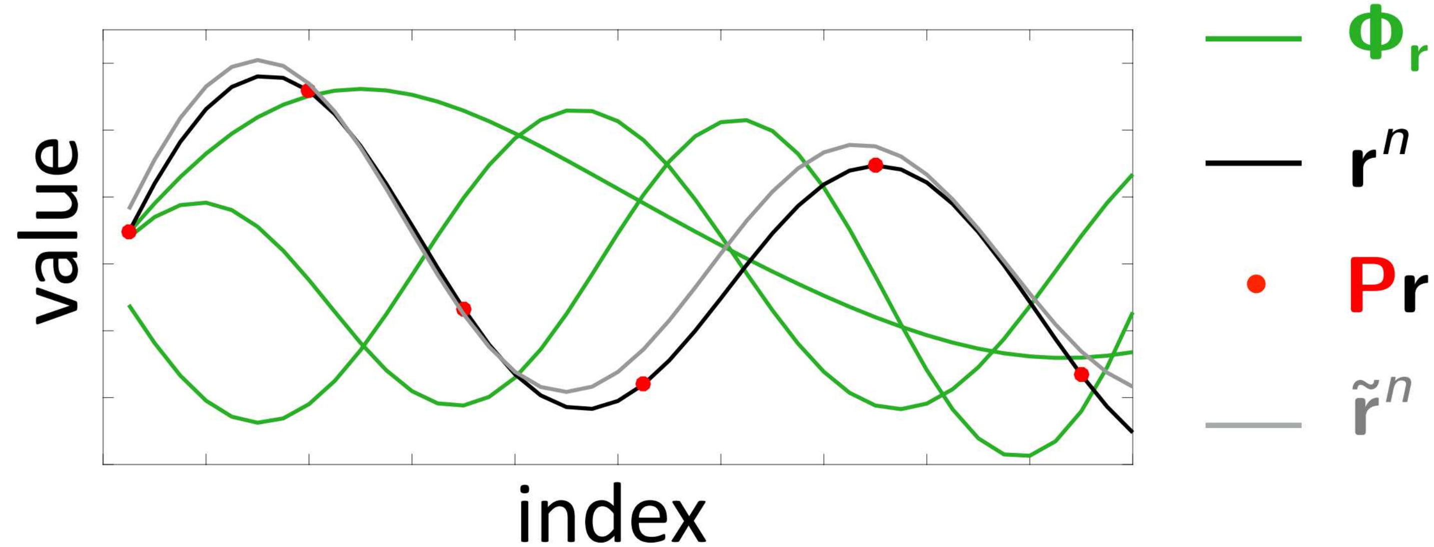


Cost reduction by gappy PCA [Everson and Sirovich, 1995]

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

Can we select **A** *to make this less expensive?*

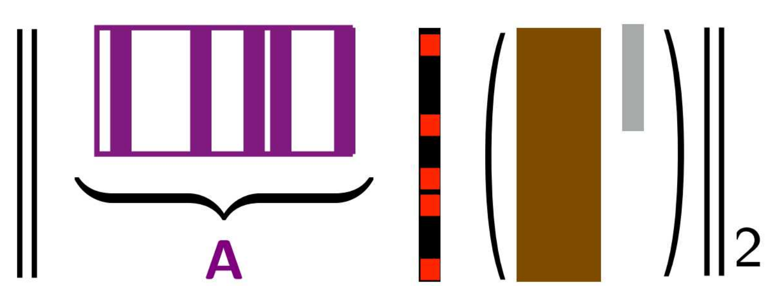
- **Training:** collect residual tensor $\mathcal{R}^{ijk}$ while solving ODE for $\boldsymbol{\mu} \in \mathcal{D}_{\text{training}}$
- **Machine learning:** compute residual PCA $\boldsymbol{\Phi}_{\mathbf{r}}$ and sampling matrix $\mathbf{P}$
- **Reduction**: compute regression approximation $\mathbf{r}^n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



Related:

- collocation [Ryckelynck, 2005; Legresley, 2006; Astrid et al., 2008]
- (D)EIM [Barrault et al., 2004; Chaturantabut and Sorensen, 2010]
- FE subassembly [**An et al., 2008**; Farhat et al., 2014]

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

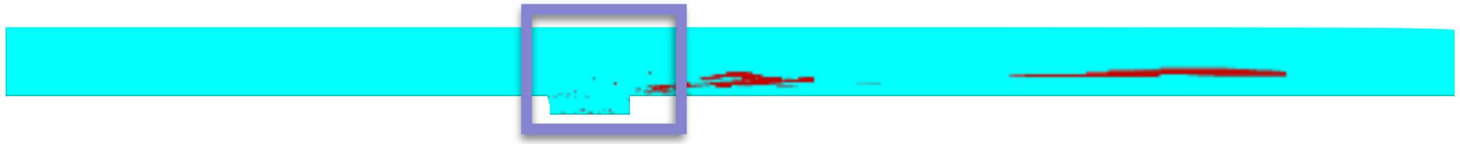


\+ *Only a few elements of* $\mathbf{r}^n$ *must be computed*

Sample mesh [Carlberg, Farhat, Cortial, Amsallem, 2013]

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

sample mesh



+ *HPC on a laptop*

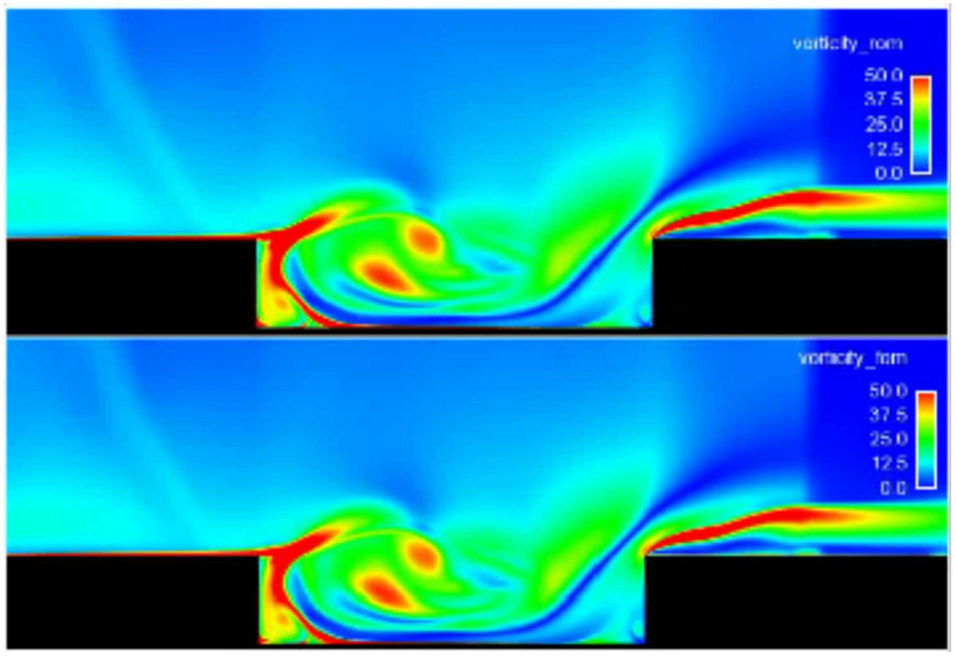
vorticity field *pressure field*

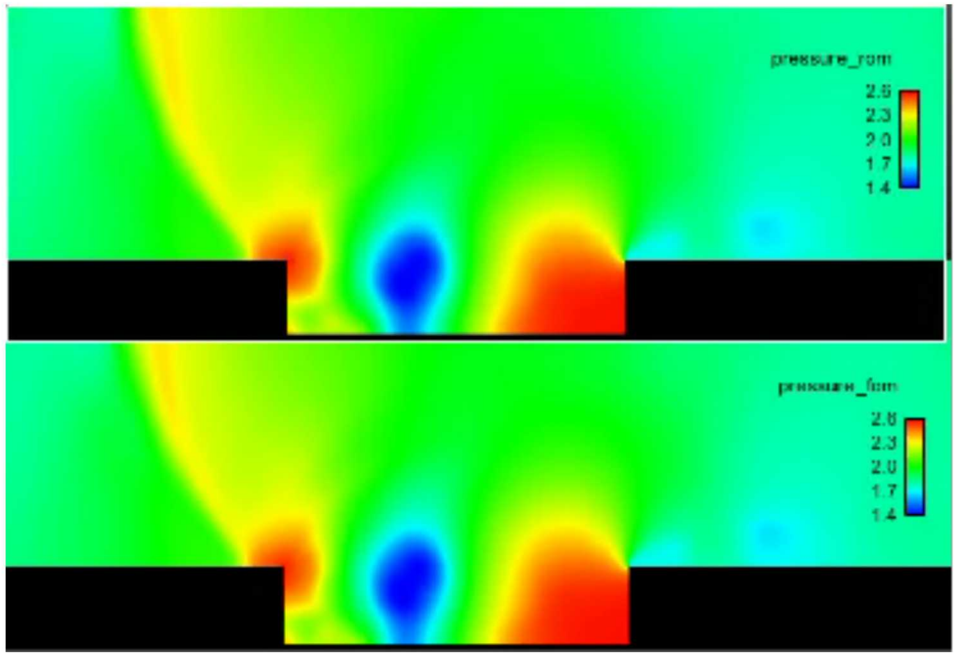
LSPG ROM with $\mathbf{A} = (\mathbf{P}\boldsymbol{\Phi}_{\mathbf{r}})^{+}\mathbf{P}$

32 min, 2 cores

high-fidelity

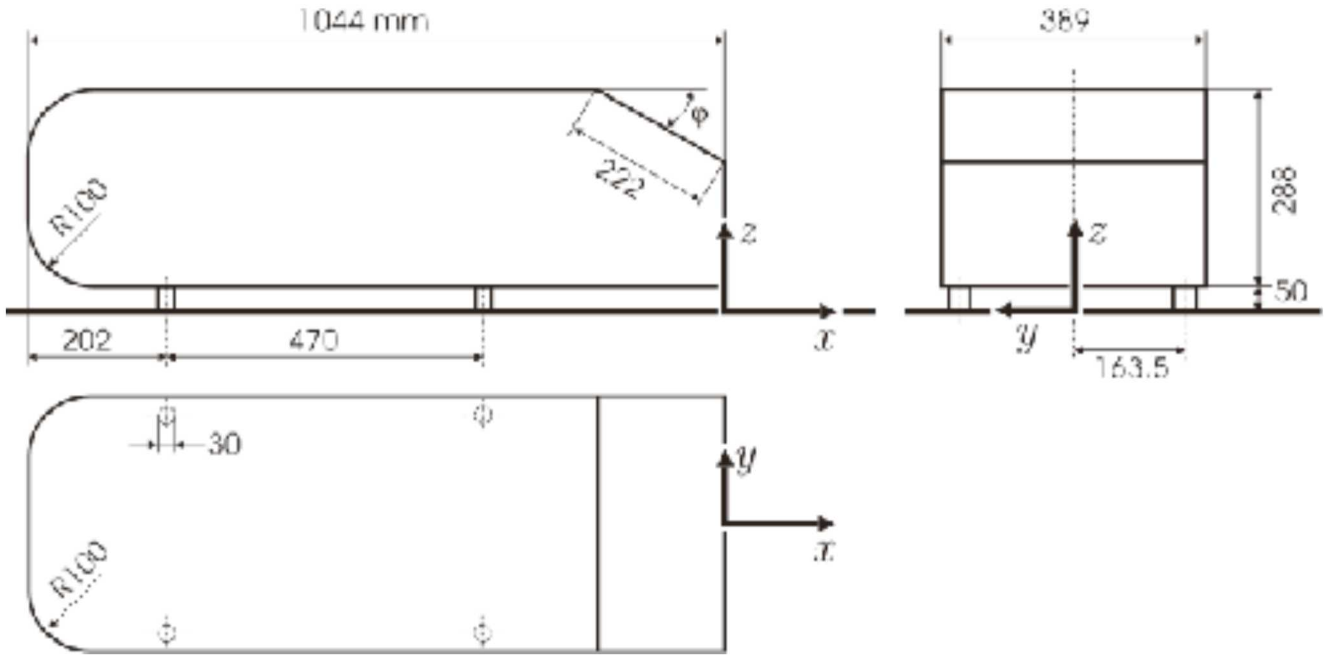
5 hours, 48 cores

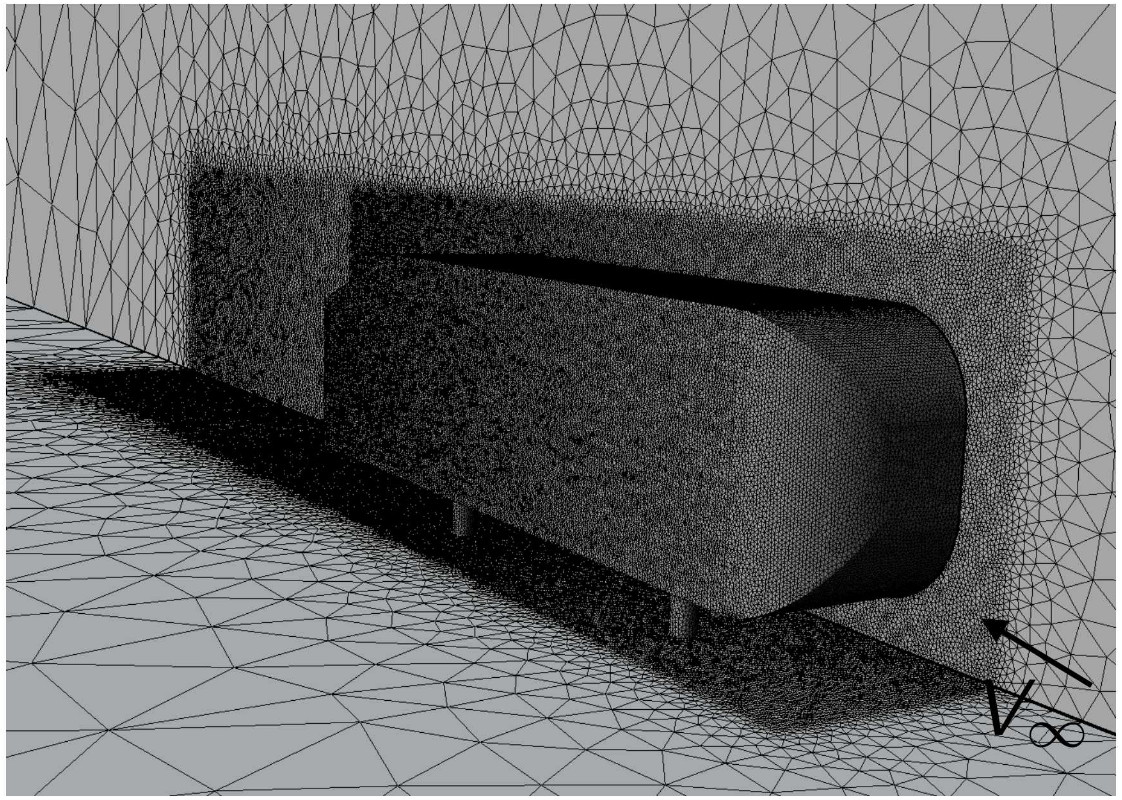




+ *229x savings in core–hours*
+ *< 1% error in time-averaged drag*
- implemented in three computational-mechanics codes

Ahmed body [Ahmed, Ramm, Faitin, 1984]





- Unsteady Navier–Stokes
- Re = 4.3 x 10^6
- $M_\infty = 0.175$

Spatial discretization

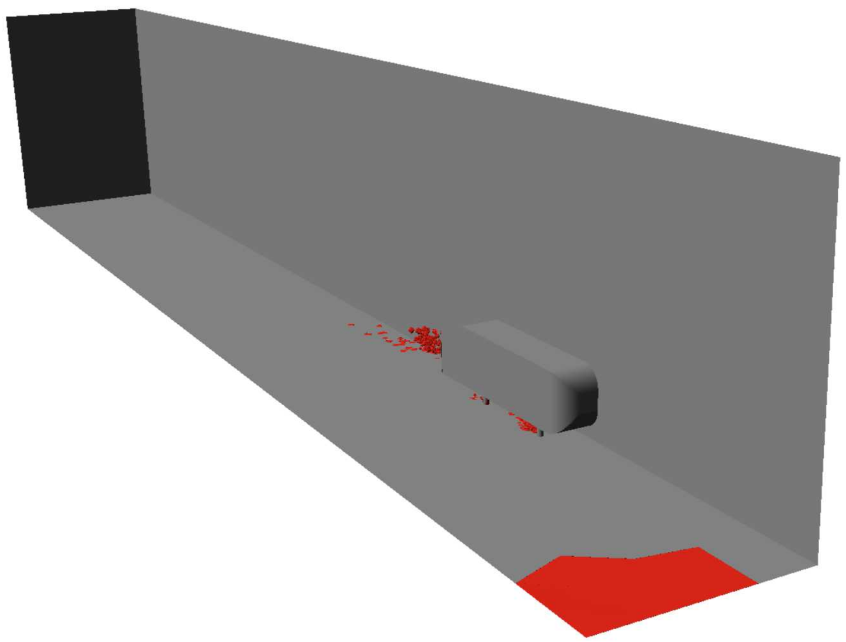
- 2nd-order finite volume
- DES turbulence model
- 1.7×10^7 degrees of freedom

Temporal discretization

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

Ahmed body results [Carlberg, Farhat, Cortial, Amsallem, 2013]

sample mesh



+ *HPC on a laptop*

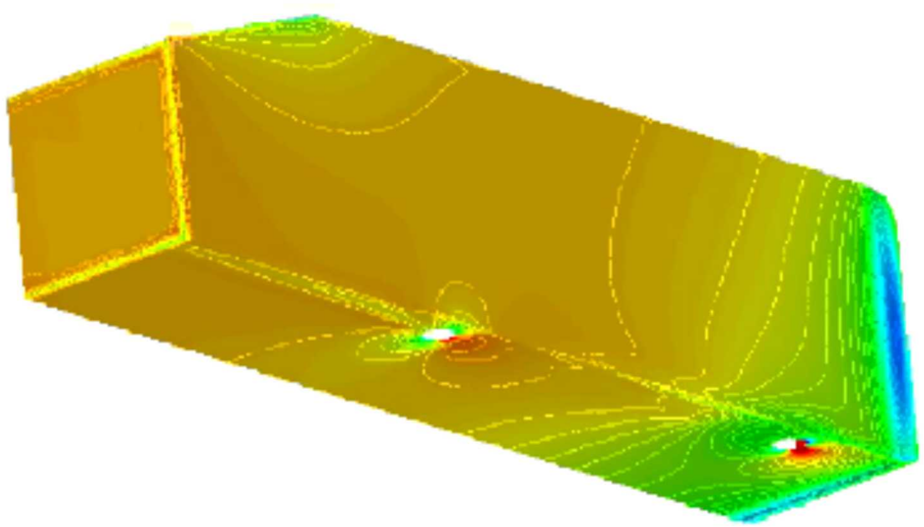
LSPG ROM with $\mathbf{A} = (\mathbf{P}\boldsymbol{\Phi}_{\mathbf{r}})^{+}\mathbf{P}$

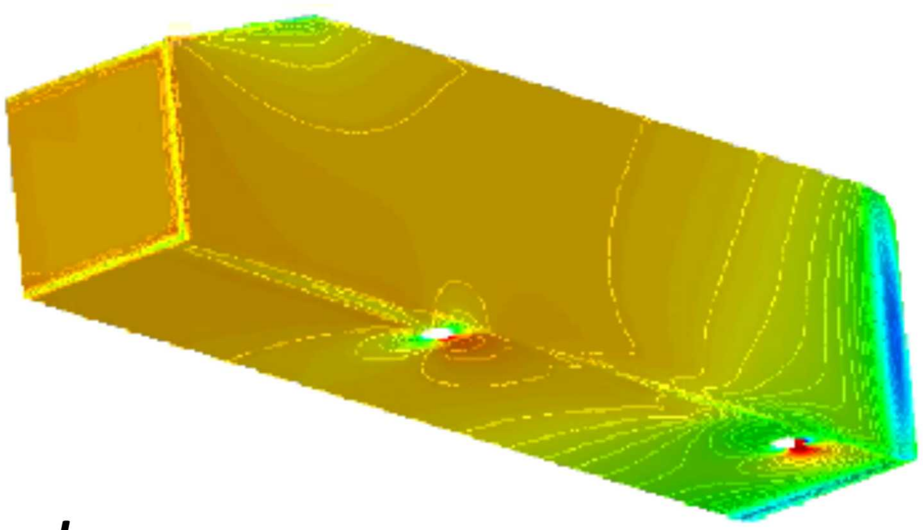
4 hours, 4 cores

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*

My research

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

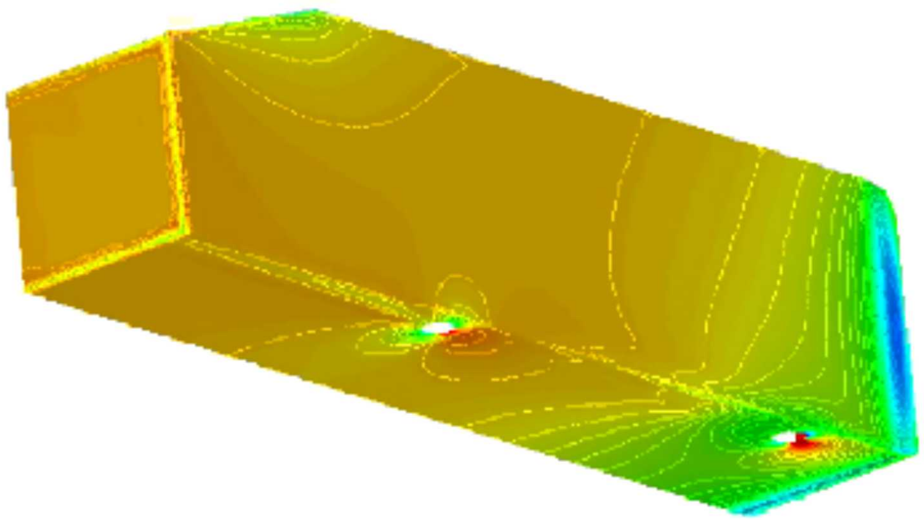
- *accuracy*: LSPG projection [Carlberg, Bou-Mosleh, Farhat, 2011; Carlberg, Antil, Barone, 2017]
- *low cost*: sample mesh [Carlberg, Farhat, Cortial, Amsallem, 2013*]
- *low cost*: reduce temporal complexity
  [Carlberg, Ray, van Bloemen Waanders, 2015; Carlberg, Brencher, Haasdonk, Barth, 2017; Choi and Carlberg, 2017]
- *structure preservation* [Carlberg, Tuminaro, Boggs, 2015; Peng and Carlberg, 2017; Carlberg and Choi, 2017]
- *reliability*: adaptivity [Carlberg, 2015]
- *certification*: machine learning error models
  [Drohmann and Carlberg, 2015; Trehan, Carlberg, Durlofsky, 2017; Freno and Carlberg, 2017]

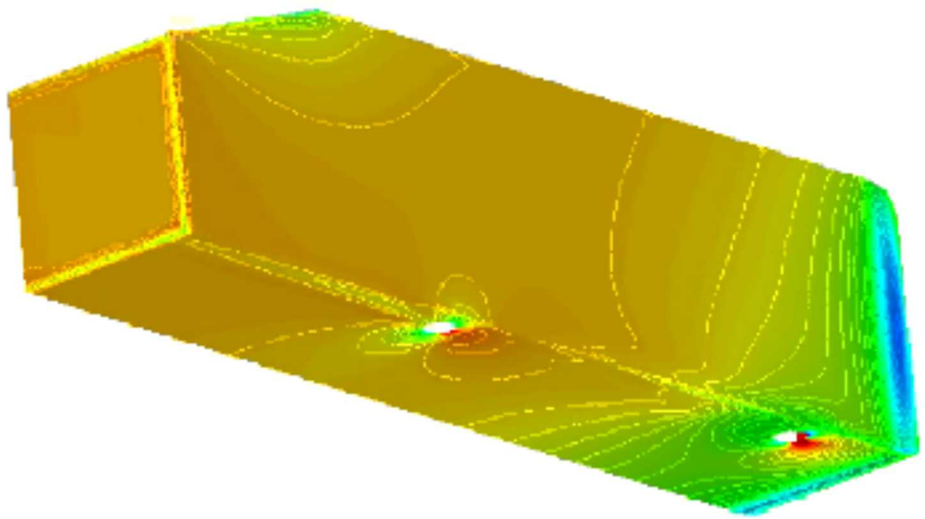
Ahmed body results [C., Farhat, Cortial, Amsallem, 2013]

GNAT ROM ($\mathbf{A} = (\mathbf{P\Phi}_r)^+\mathbf{P}$)
4 hours, 4 cores

high-fidelity model
13 hours, 512 cores

pressure field





spatial dim: 283
temporal dim: 1.3 x 10^3

spatial dim: 1.7 x 10^7
temporal dim: 1.3 x 10^3

+ *438X computational-cost reduction*
+ *60,500X spatial-dimension reduction*
- *Zero 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

Forecasting using gappy POD in time

- ‣ Accurate Newton-solver initial guess [C., Ray, van Bloemen Waanders, 2015]
- ‣ Coarse propagator in time-parallel setting [C., Brencher, Haasdonk, Barth, 2016]
- \+ No error incurred and wall-time improvements observed
- – No reduction of time dimension

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

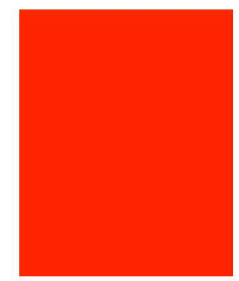
+ Applicability to general nonlinear dynamical systems
+ Hyper-reduction to reduce complexity of nonlinearities
+ Extract multiple space–time basis vectors from each training simulation

Space–time least-squares Petrov–Galerkin (ST-LSPG) projection [Choi and C., 2017]

Spatial v. spatiotemporal trial

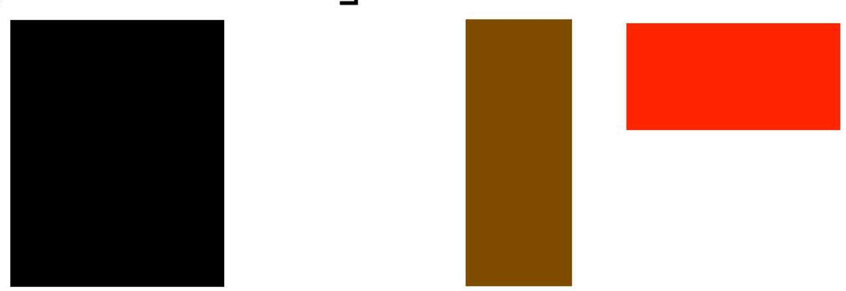
Full-order-model trial subspace

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



Spatial trial subspace

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

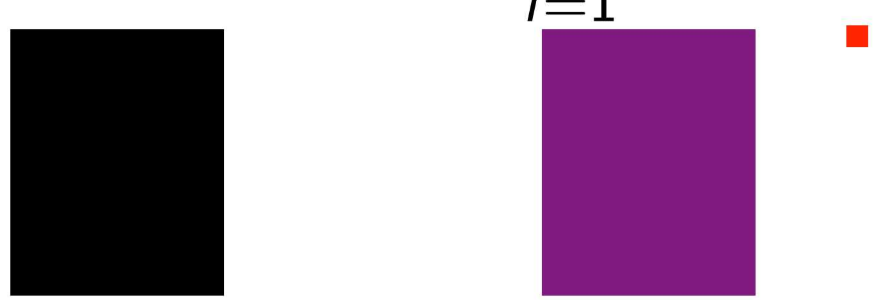


+ Spatial dimension reduced
- Temporal dimension large

Space–time trial subspace

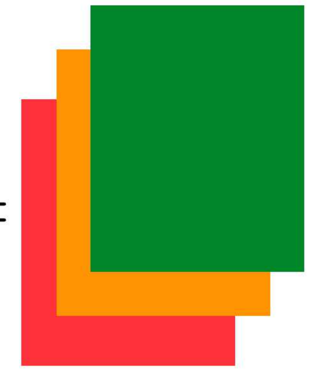
$$\begin{bmatrix}\tilde{\mathbf{x}}^1 & \cdots & \tilde{\mathbf{x}}^T\end{bmatrix} = \sum_{i=1}^{n_{st}} \boldsymbol{\pi}_i \hat{x}_i(\boldsymbol{\mu}) \in \mathcal{S}\mathcal{T} \subseteq \mathbb{R}^N \otimes \mathbb{R}^T$$

How to compute space–time bases $\boldsymbol{\pi}_i$?



+ Spatial dimension reduced
+ Temporal dimension reduced
- Additional approximation

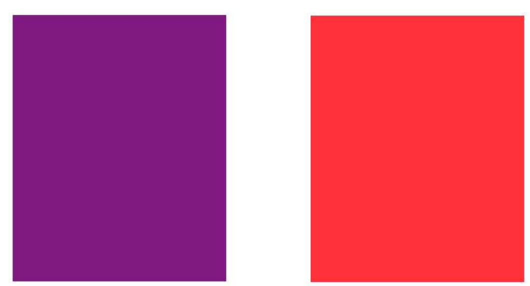
Space–time basis computation

$\mathcal{X} =$ 

Tensor slices

[Urban, Patera, 2012; Yano, 2013; Urban, Patera, 2014; Yano, Patera, Urban, 2014; Volkwein, Weiland, 2006; Constantine, Wang, 2012]

$$\boldsymbol{\pi}_i = \mathbf{X}_i$$

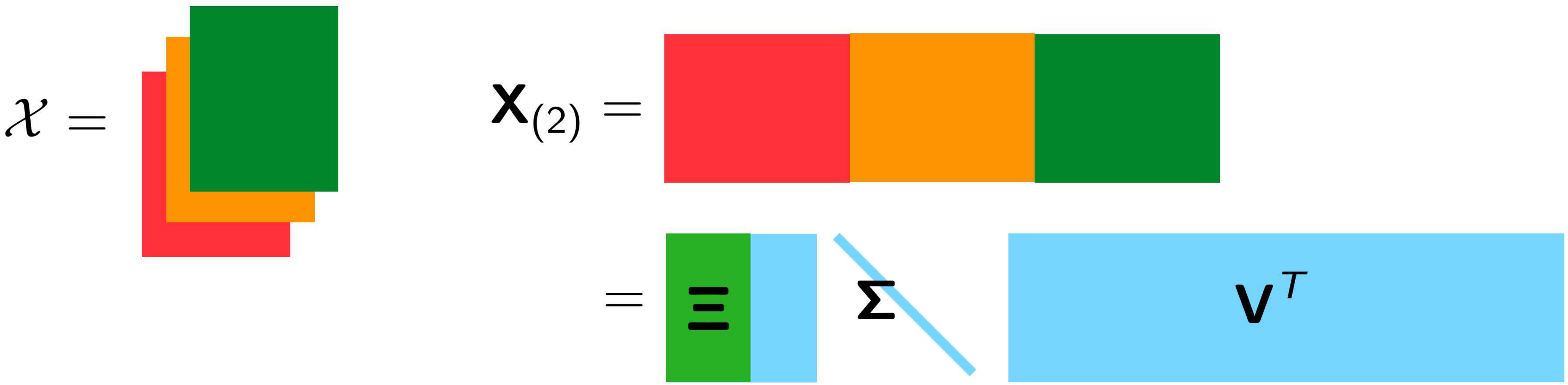


+ General space–time structure
- Only one basis vector per training simulation
- NT storage per basis vector

Space–time basis computation

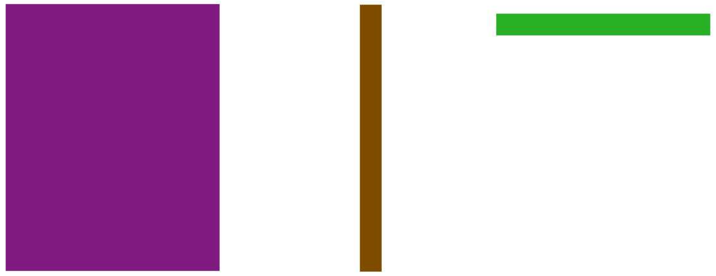
Truncated high-order SVD (T-HOSVD) [Baumann, Benner, Heiland, 2016]

- Compute dominant left singular vectors of **mode-2** unfolding



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

$$\boldsymbol{\pi}_{\mathcal{J}(i,j)} = \boldsymbol{\phi}_i \otimes \boldsymbol{\xi}_j$$



\+ Multiple basis vectors per training simulation

\+ N+T storage per basis vector

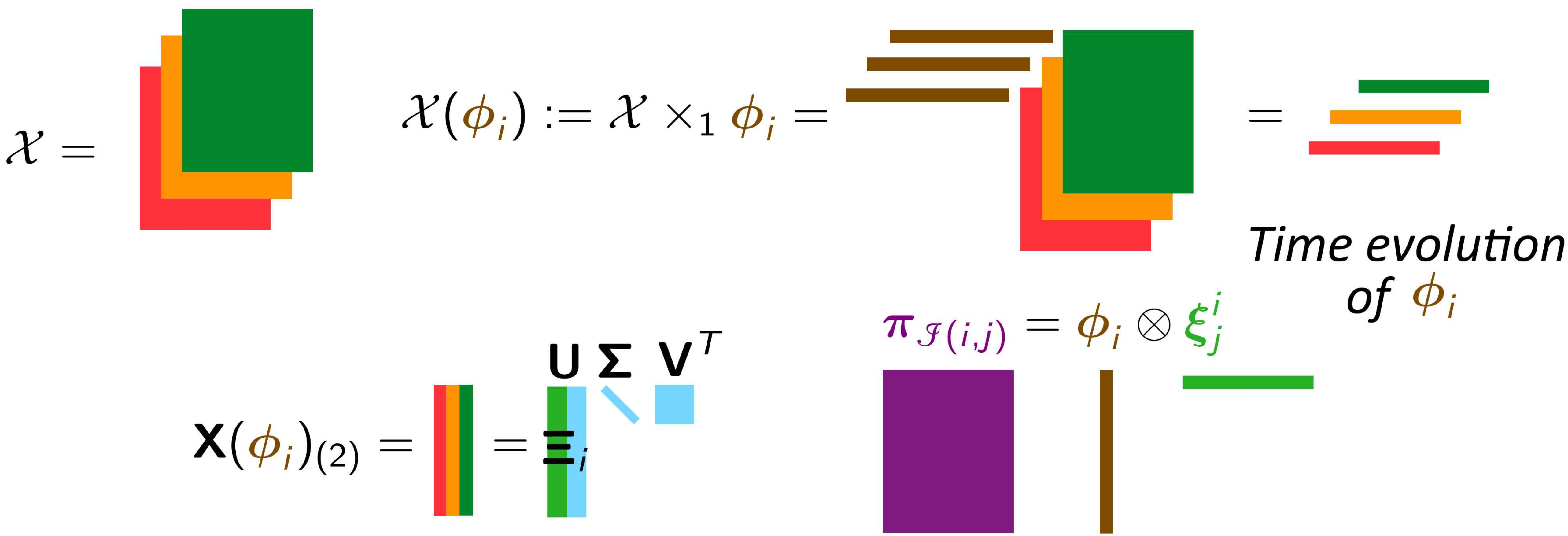
\- Enforces Kronecker–product structure

\- Same temporal modes for each spatial mode

Space–time basis computation

Sequentially truncated high-order SVD (ST-HOSVD)

[C., Ray, van Bloemen Waanders, 2015; C., Brencher, Haasdonk, Barth, 2016]



$\mathbf{\Xi}_i$ *columns are principal components of the* ***temporal*** *simulation data of* ϕ_i

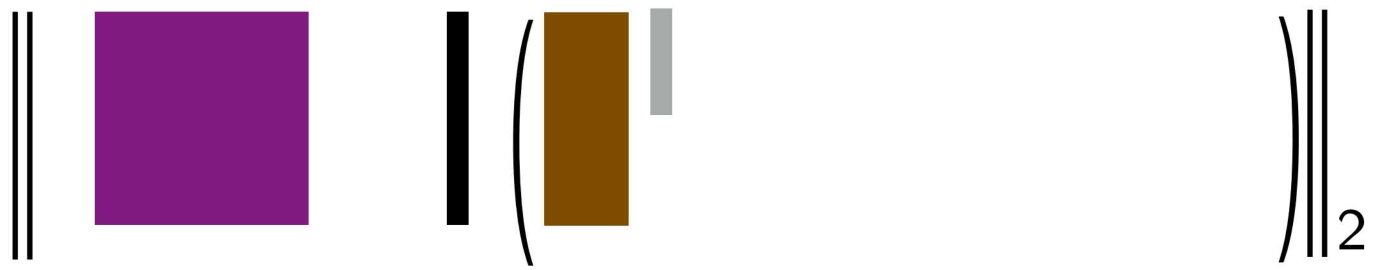
- \+ Multiple basis vectors per training simulation
- \+ N+T storage per basis vector
- \+ Tailored temporal modes for each spatial mode
- \- Enforces Kronecker-product structure

How to project governing equations?

Space–time LSPG projection

LSPG

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

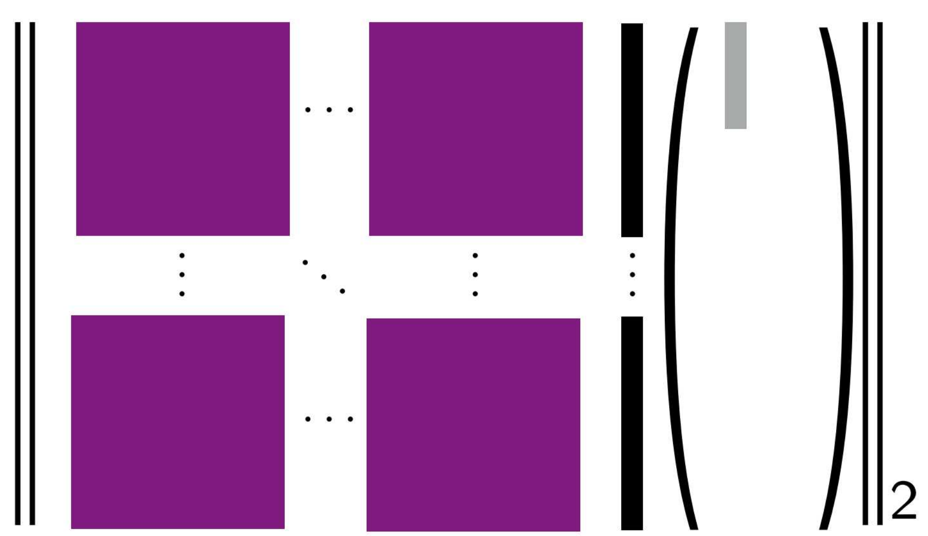


\+ efficient: time-sequential solve

ST-LSPG

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

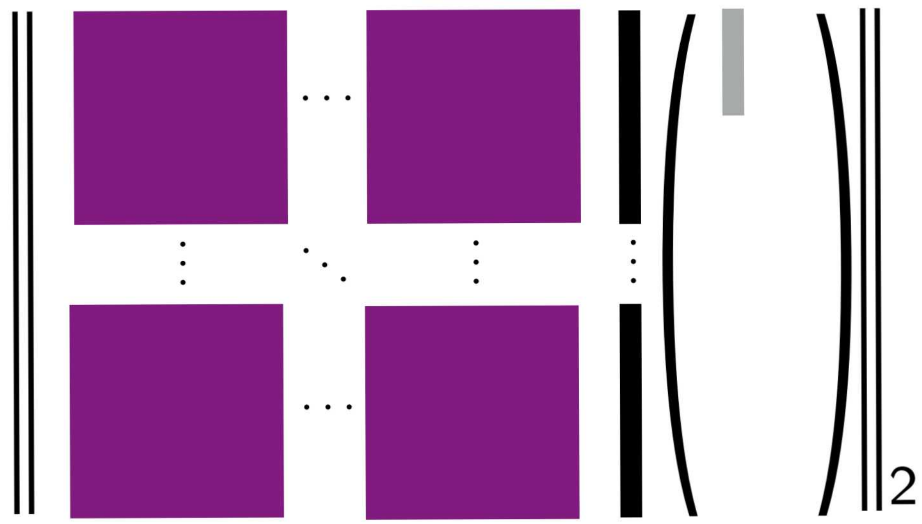
$$\underset{\hat{\mathbf{v}}}{\text{minimize}} \left\| \bar{\mathbf{A}} \, \bar{\mathbf{r}}(\hat{\mathbf{v}}; \boldsymbol{\mu}) \right\|_2$$



\- costly: minimizing residual simultaneously over space and time

ST-LSPG hyper-reduction

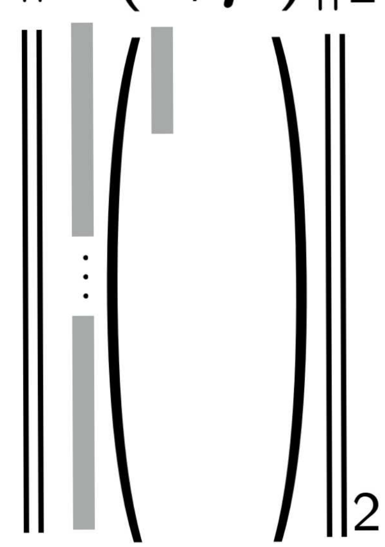
$$\underset{\hat{\mathbf{v}}}{\text{minimize}} \left\| \bar{\mathbf{A}} \, \bar{\mathbf{r}}(\hat{\mathbf{v}};\boldsymbol{\mu}) \right\|_2$$



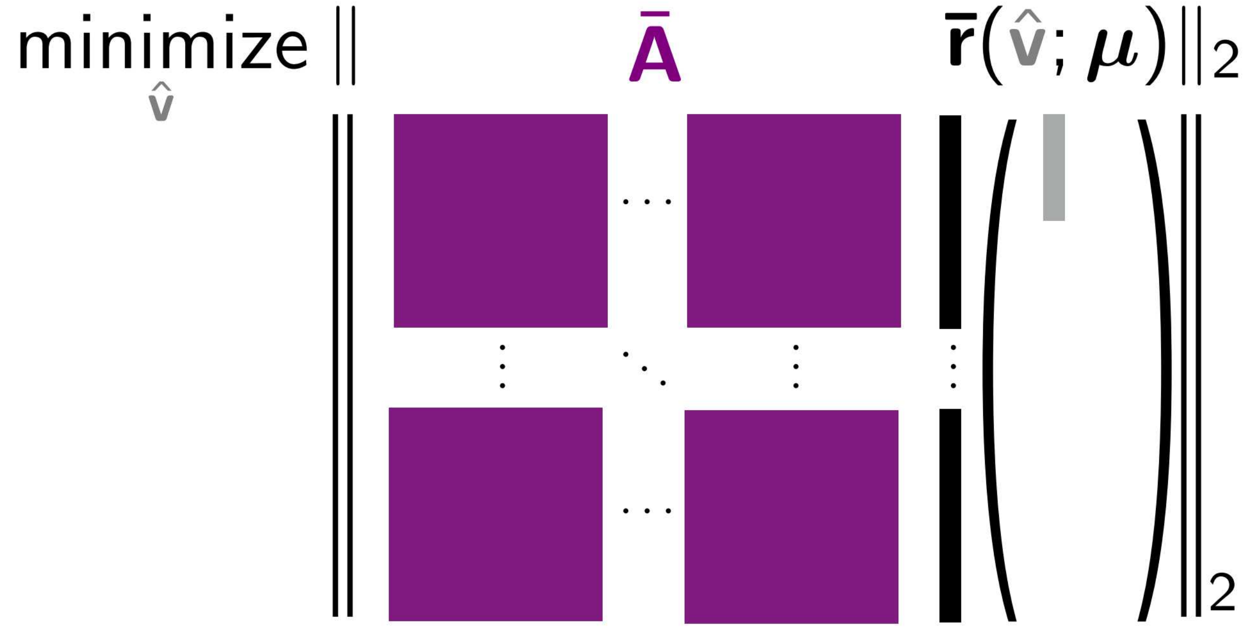
$$\bar{\mathbf{r}} \approx \tilde{\mathbf{r}} = \bar{\boldsymbol{\Phi}}_{\mathbf{r}}(\bar{\mathbf{P}}\bar{\boldsymbol{\Phi}}_{\mathbf{r}})^{+}\bar{\mathbf{P}}\bar{\mathbf{r}}$$

- space–time residual basis $\bar{\boldsymbol{\Phi}}_{\mathbf{r}}$ via tensor decomposition
- space–time sampling $\bar{\mathbf{P}}$ via sequential greedy

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

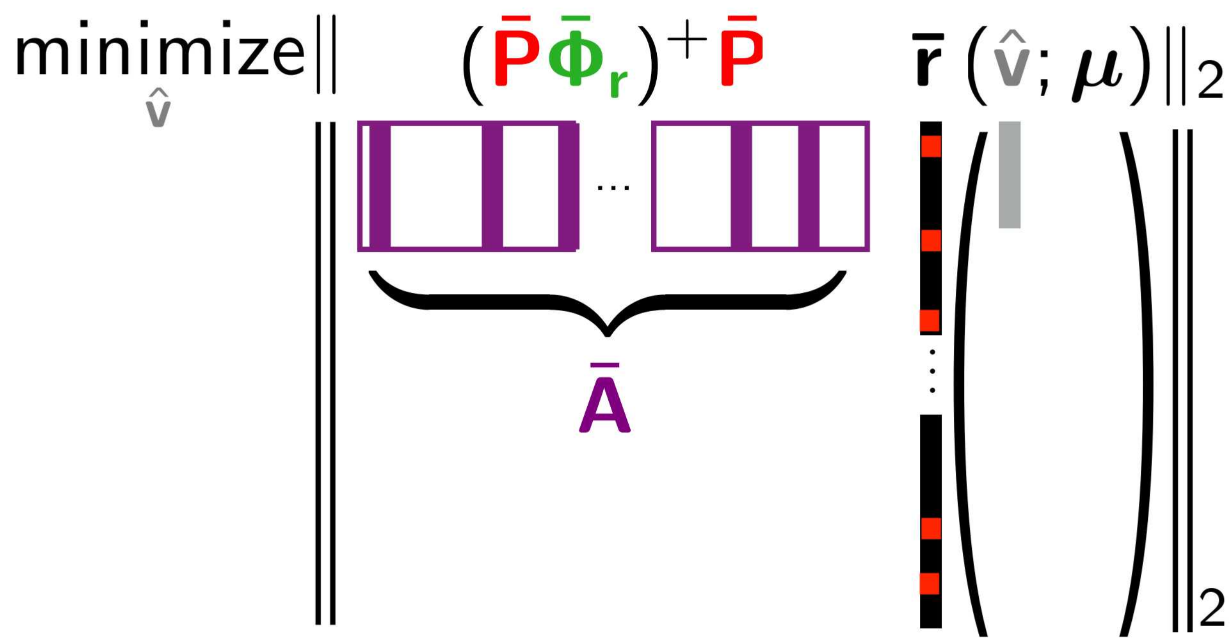


ST-LSPG hyper-reduction



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

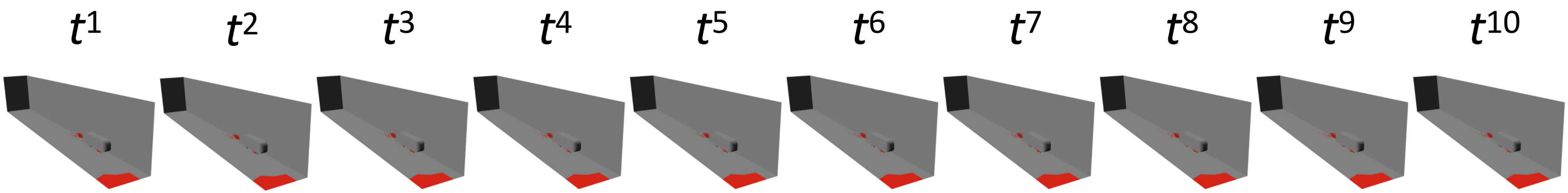
- space–time residual basis $\bar{\boldsymbol{\Phi}}_\mathbf{r}$ via tensor decomposition
- space–time sampling $\bar{\mathbf{P}}$ via sequential greedy



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

Sample mesh

LSPG

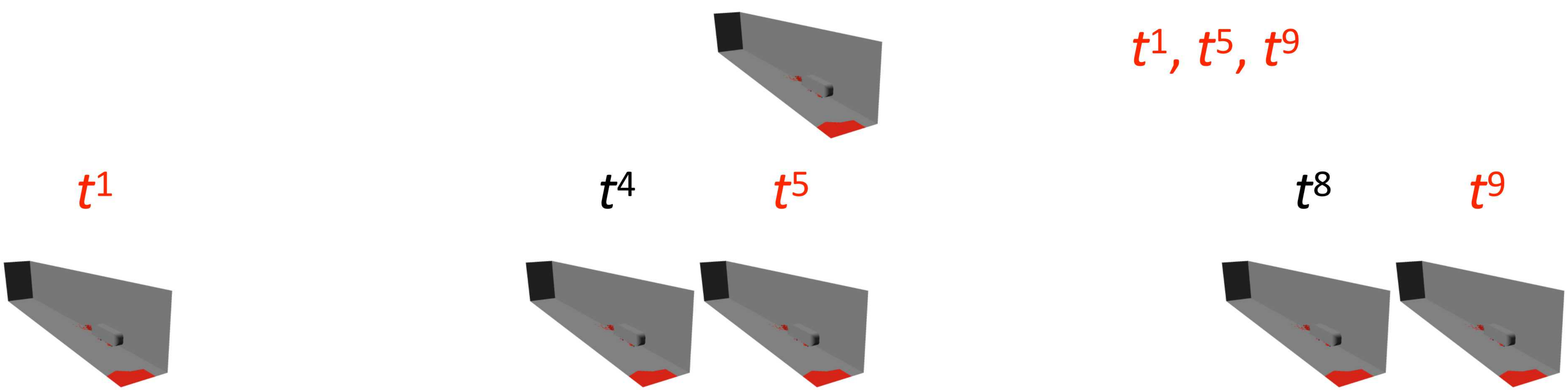


\+ Residual computed at a few spatial degrees of freedom

\- Residual computed at all time instances

ST-LSPG

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



\+ Residual computed at a few space—time degrees of freedom

Error bound

LSPG

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

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

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

ST-LSPG

+ *Single solve*: no sequential error accumulation

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

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

- **Experiments**: for fixed error, ST-LSPG almost 100X faster than LSPG

My research

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

- *accuracy*: LSPG projection [Carlberg, Bou-Mosleh, Farhat, 2011; Carlberg, Antil, Barone, 2017]
- *low cost*: sample mesh [Carlberg, Farhat, Cortial, Amsallem, 2013]
- *low cost*: reduce temporal complexity
  [Carlberg, Ray, van Bloemen Waanders, 2015; Carlberg, Brencher, Haasdonk, Barth, 2017; Choi and Carlberg, 2017]
- *structure preservation* [Carlberg, Tuminaro, Boggs, 2015*; Peng and Carlberg, 2017; Carlberg and Choi, 2017]
- *reliability*: adaptivity [Carlberg, 2015]
- *certification*: machine learning error models
  [Drohmann and Carlberg, 2015; Trehan, Carlberg, Durlofsky, 2017; Freno and Carlberg, 2017]

* Featured Article, SIAM J Sci Comp, 2015

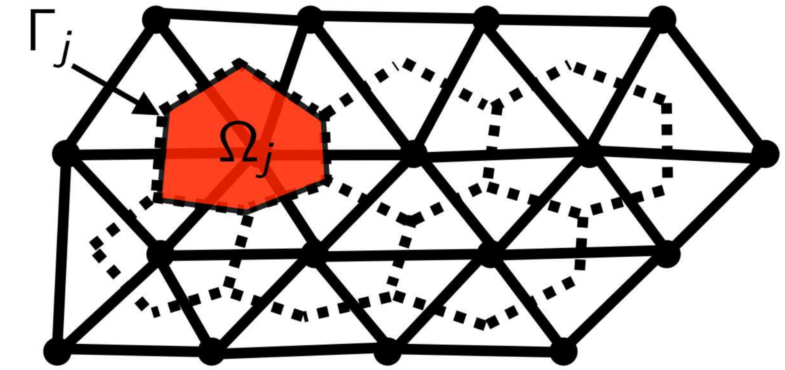
Structure preservation: previous state of the art

- Stability [Moore, 1981; Bond and Daniel, 20018; Amsallem and Farhat, 2012; Kalashnikova et al., 2014]
- Second-order structure [Freund 2005; Salimbahrami, 2005; Chahlaoui, 2015]
- Delay [Beattie and Gugercin, 2008; Michiels et al., 2011; Schulze and Unger, 2015]
- Bilinear [Zhang and Lam, 2002; Benner and Damm, 2011; Benner and Breiten, 2012; Flagg and Gugercin, 2015]
- Inf–sup stability [Rozza and Veroy, 2007; Gerner and Veroy, 2012; Rozza et al., 2013; Ballarin et al., 2014]
- Passivity [Phillips et al., 2003; Sorensen 2005; Wolf et al., 2010]
- Energy conservation [**An et al., 2008**; Farhat et al., 2014; Farhat et al., 2015]
- (Port-)Hamiltonian [Polyuga and van der Schaft, 2008; Beattie and Gugercin, 2011; Arkham and Hesthaven, 2016; Chaturantabut et al., 2016; Peng and Mohseni, 2016]

What structure should we preserve in finite-volume models?

LSPG for finite-volume models

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



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

- average value of conserved variable i over control volume j

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

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

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

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

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

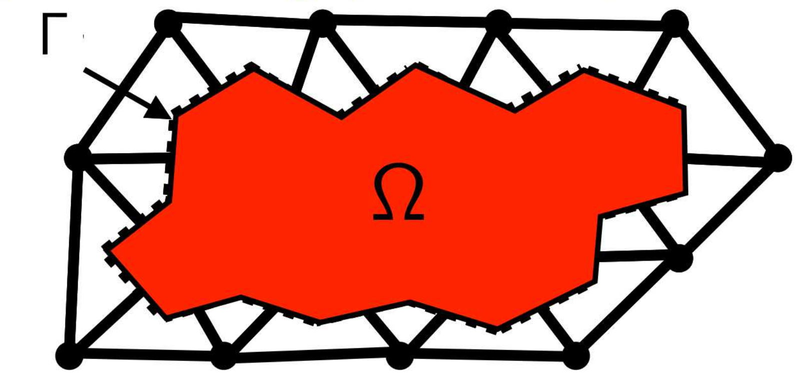
$$\text{LSPG O}\Delta\text{E:} \quad \underset{\hat{\mathbf{v}}}{\text{minimize}} \ \|\mathbf{A}\mathbf{r}^n(\boldsymbol{\Phi}\hat{\mathbf{v}})\|_2$$

- minimize weighted sum of squared conservation violations over time step n
  - Does not guarantee conservation anywhere

Enforce global conservation [Carlberg, Choi, Sargsyan, 2017]

LSPG-FV: $\underset{\hat{\mathbf{v}}}{\text{minimize}} \; \|\mathbf{A}\mathbf{r}^n(\boldsymbol{\Phi}\hat{\mathbf{v}})\|_2$

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

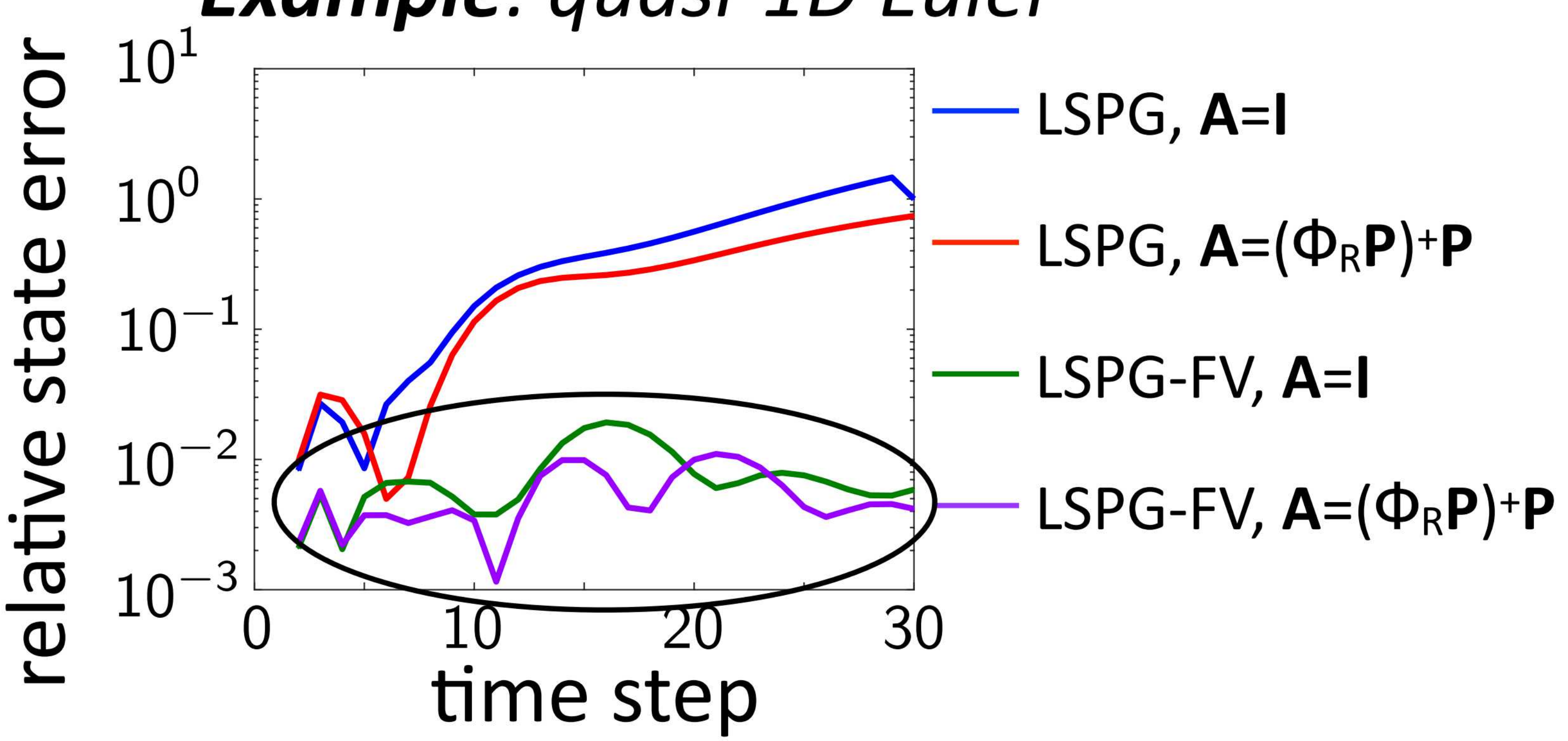


- minimize weighted sum of squared conservation-law violations over time step n subject to global conservation

$$\bar{x}_i(t) = \frac{1}{|\Omega|}\int_\Omega u_i(\vec{x}, t)\, d\vec{x}$$

$$\bar{f}_i(\mathbf{x}, t) = -\frac{1}{|\Omega|}\int_\Gamma \mathbf{g}_i(\mathbf{x}; \vec{x}, t)\cdot \mathbf{n}_j(\vec{x})\, d\vec{s}(\vec{x}) + \frac{1}{|\Omega|}\int_\Omega s_i(\mathbf{x}; \vec{x}, t)\, d\vec{x}$$

***Example**: quasi-1D Euler*



speedup

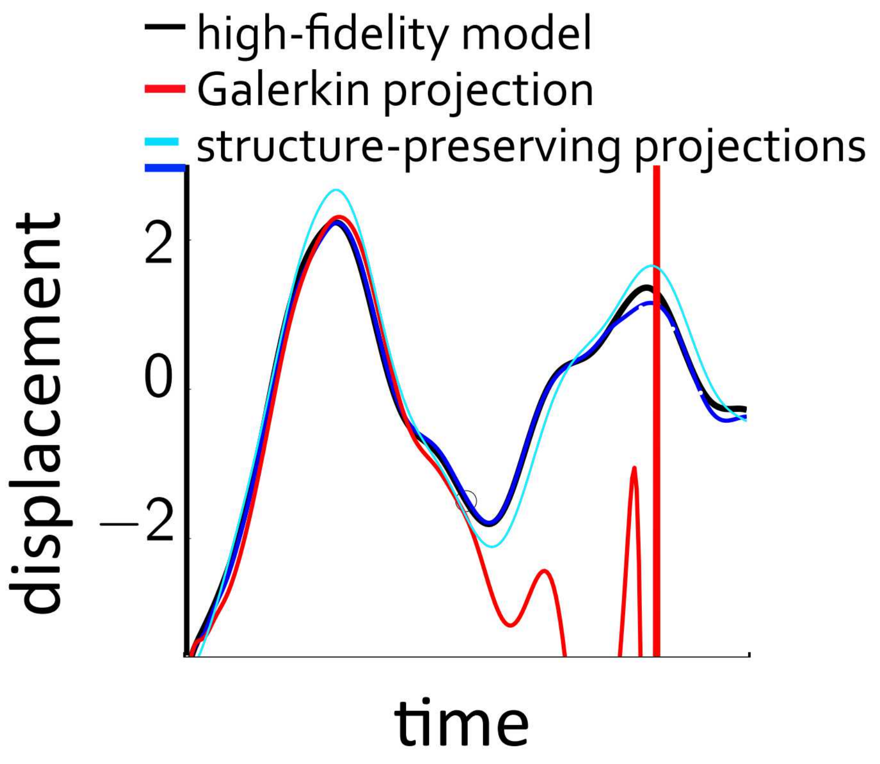
	LSPG	LSPG-FV
A=I	0.57	0.44
A=(Φ_RP)$^+$P	4.4	5.3

+ structure preservation improves accuracy
+ sample mesh improves wall time

Structure preservation

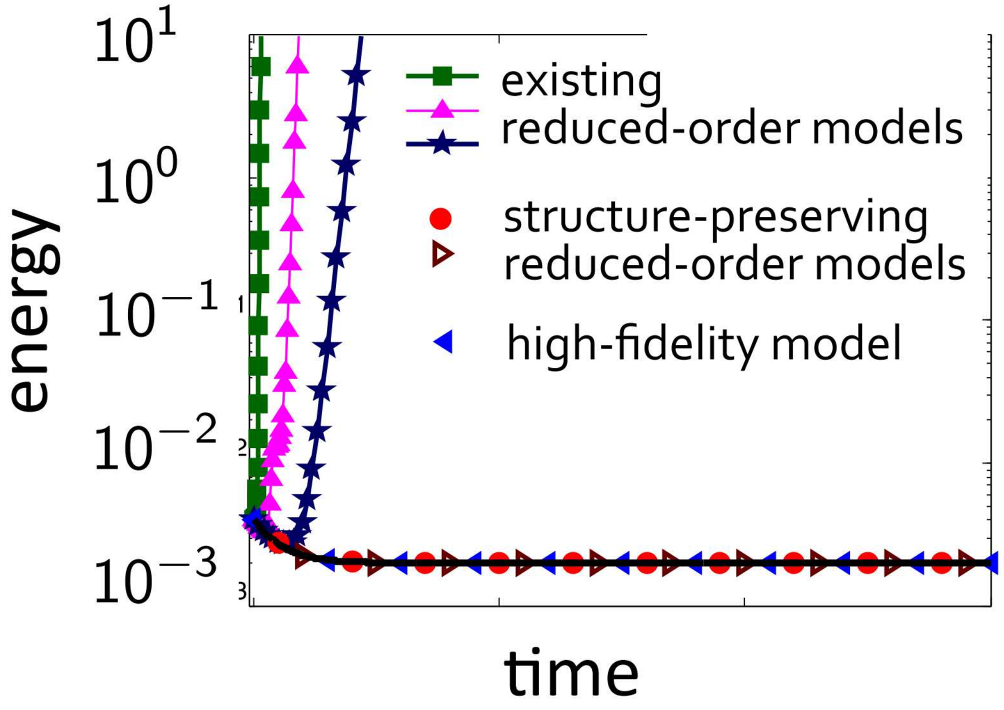
Nonlinear Lagrangian dynamical systems
[Carlberg, Tuminaro, Boggs, 2015]

- approximates Lagrangian ingredients, then derives equations of motion
- + ensures symplectic time evolution
- + conserves total energy



Preserving marginal stability (LTI systems)
[Peng and Carlberg, 2017]

- applies symplectic projection to ensure ROM has purely imaginary poles
- + guarantees finite infinite-time energy
- + enables extension of balanced truncation



My research

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

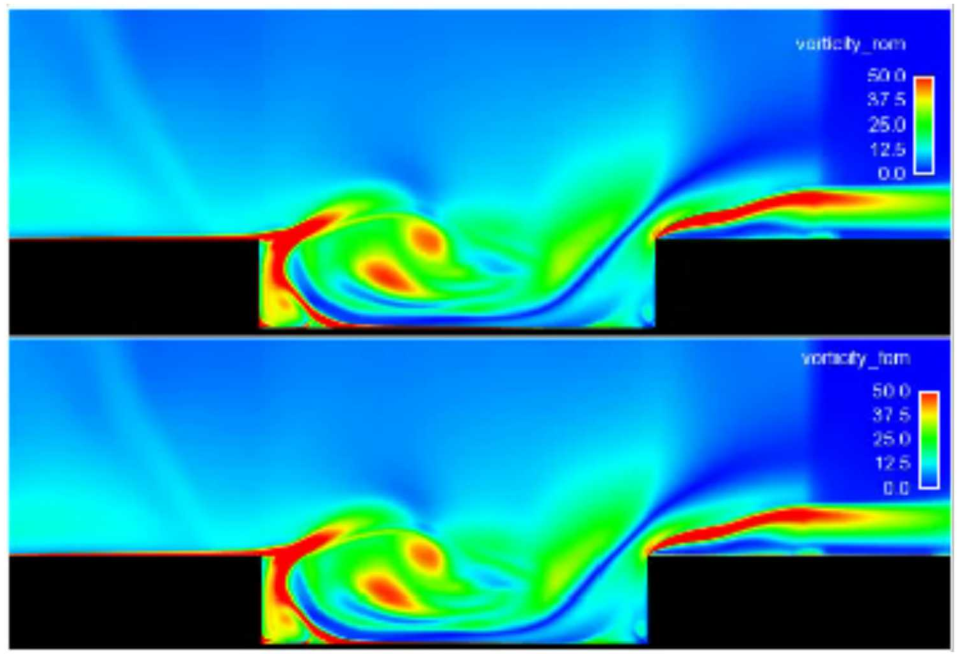
- *accuracy*: LSPG projection [Carlberg, Bou-Mosleh, Farhat, 2011; Carlberg, Antil, Barone, 2017]
- *low cost*: sample mesh [Carlberg, Farhat, Cortial, Amsallem, 2013]
- *low cost*: reduce temporal complexity [Carlberg, Ray, van Bloemen Waanders, 2015; Carlberg, Brencher, Haasdonk, Barth, 2017; Choi and Carlberg, 2017]
- *structure preservation* [Carlberg, Tuminaro, Boggs, 2015; Peng and Carlberg, 2017; Carlberg and Choi, 2017]
- *reliability*: adaptivity [Carlberg, 2015]
- *certification*: machine learning error models [Drohmann and Carlberg, 2015; Trehan, Carlberg, Durlofsky, 2017; Freno and Carlberg, 2017]

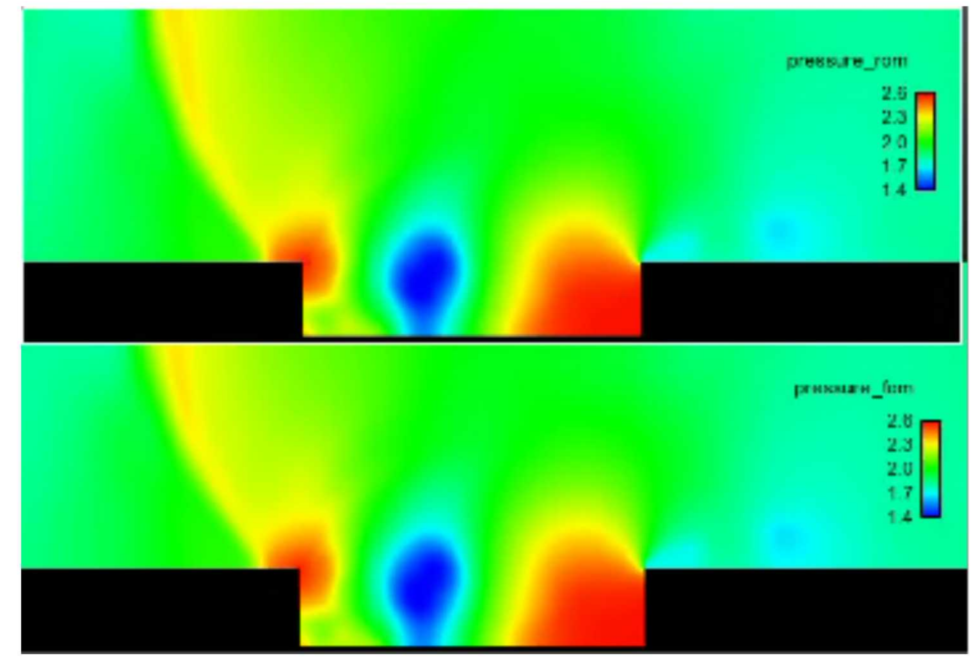
Model reduction can work well…

vorticity field *pressure field*

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

high-fidelity
5 hours, 48 cores

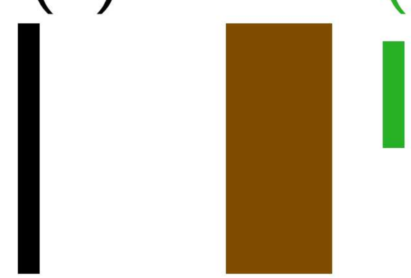




+ *229x savings in core–hours*
+ *< 1% error in time-averaged drag*

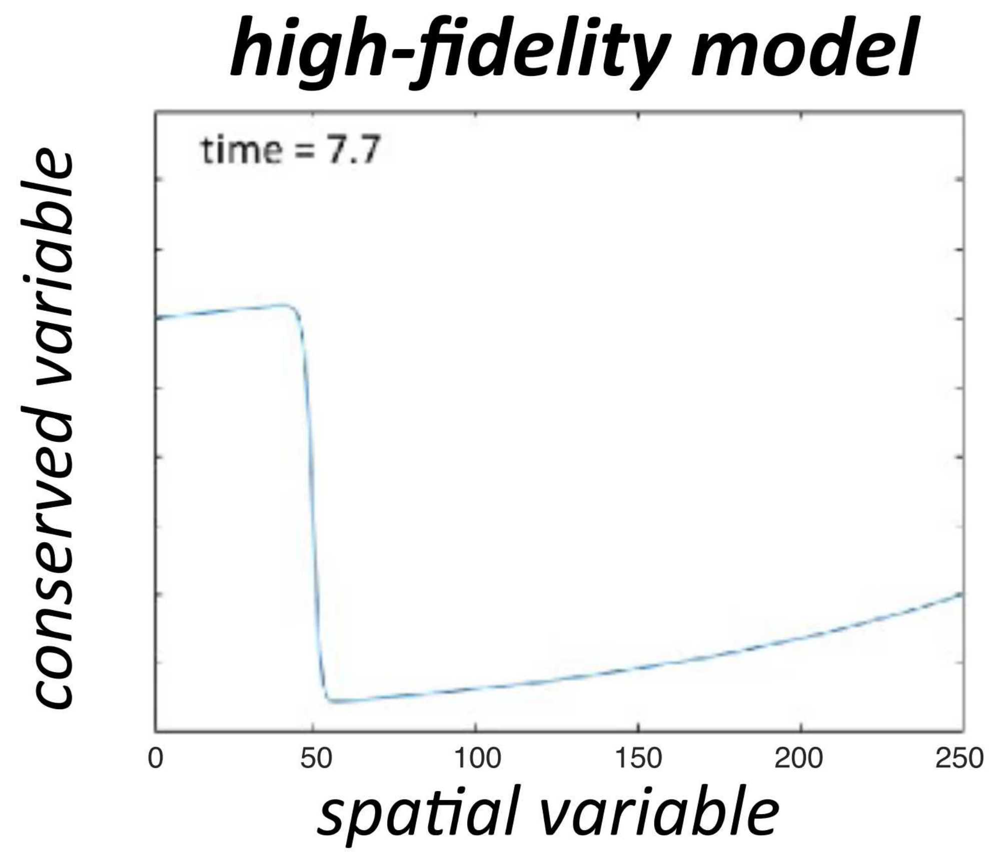
… however, this is not guaranteed

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



Accuracy limited by information in $\mathbf{\Phi}$

Illustration: inviscid 1D Burgers' equation



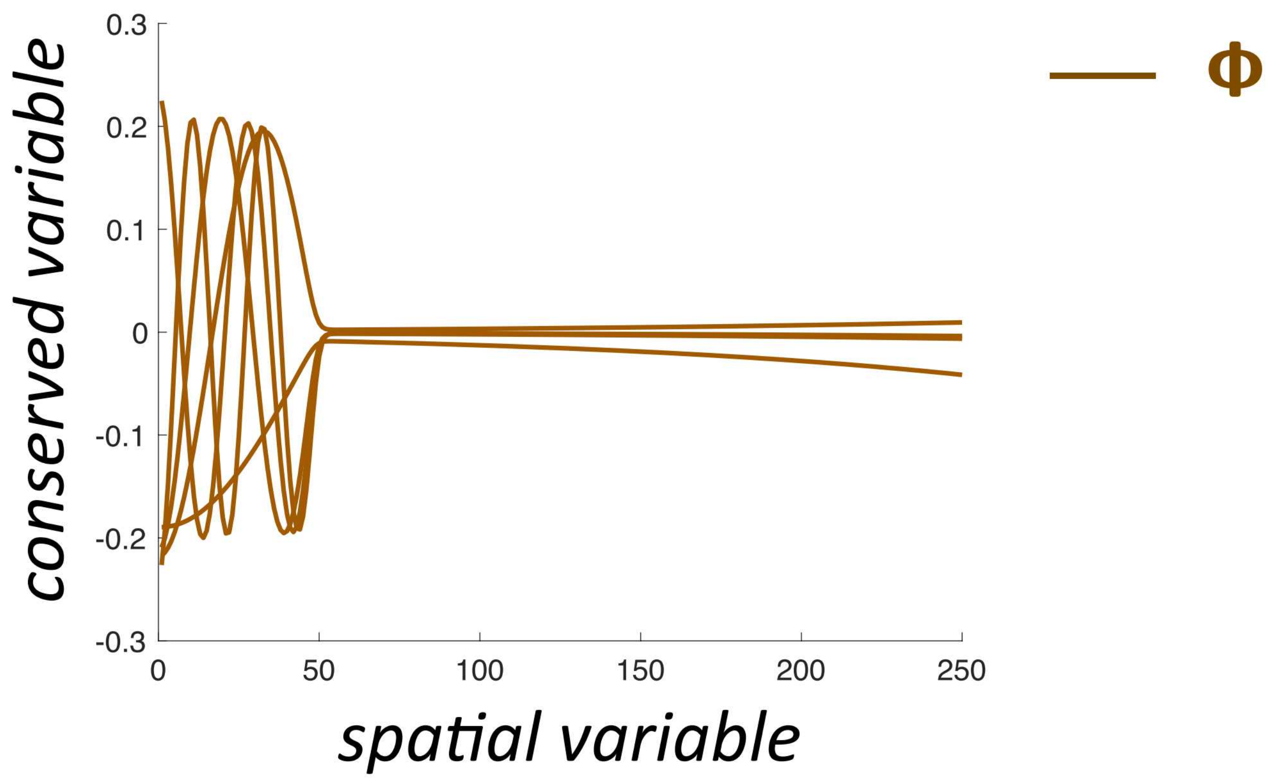
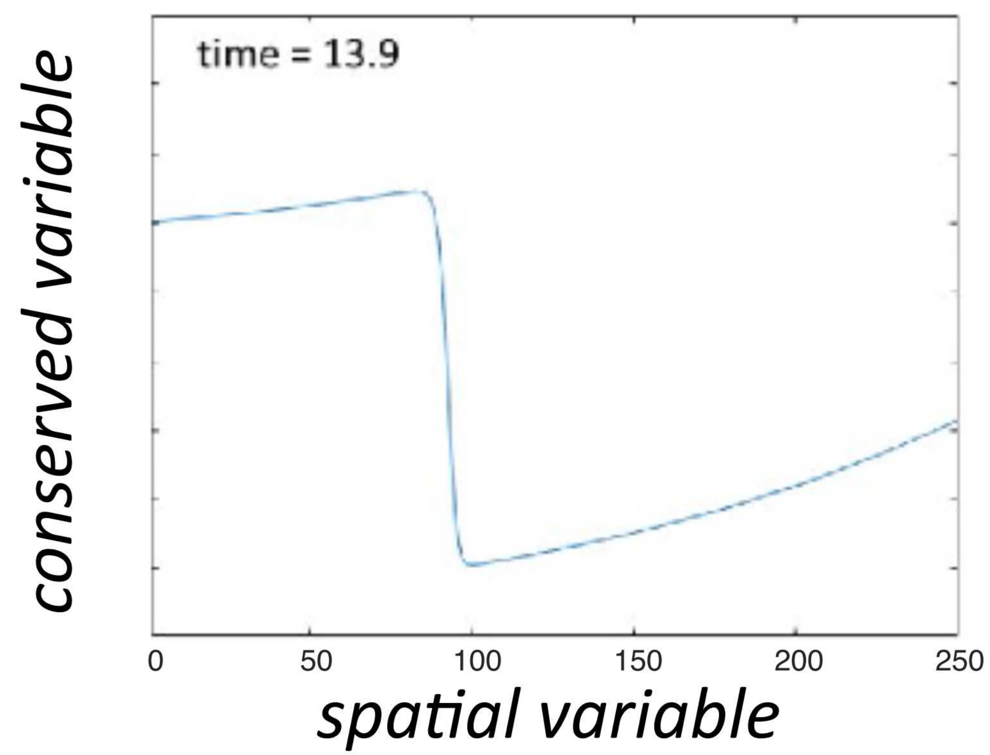
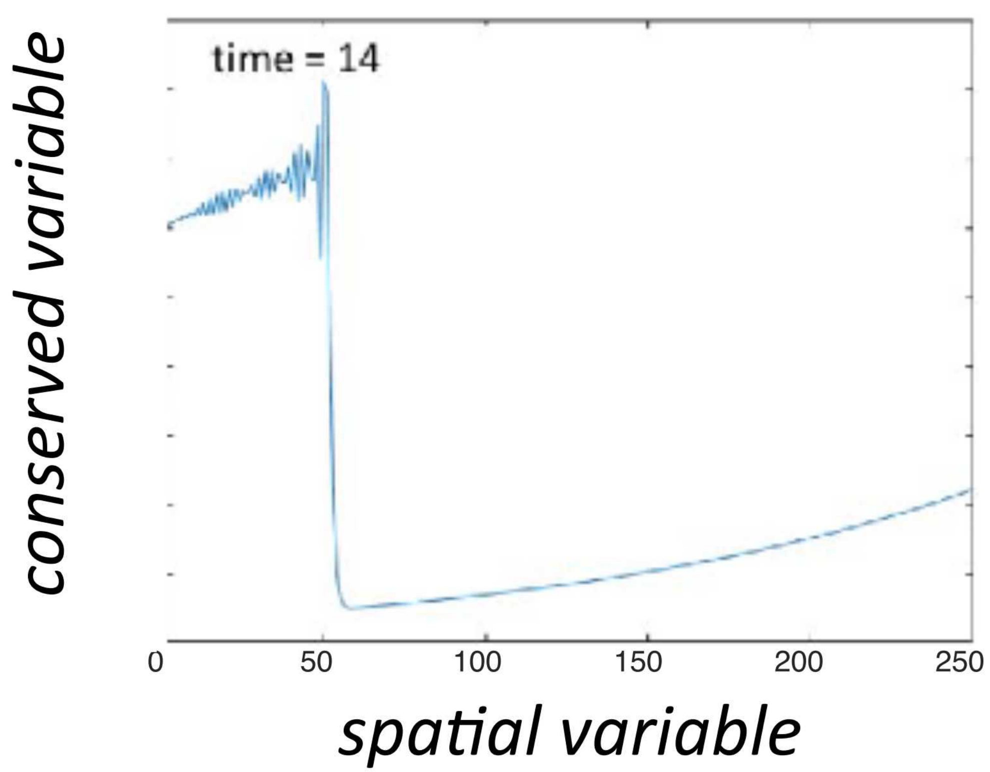


Illustration: inviscid 1D Burgers' equation

high-fidelity model



reduced-order model



reduced-order model inaccurate when Φ insufficient

ROM adaptation: previous state of the art

A priori **adaptation: unique ROM for separate regions of the**

- input space [Amsallem and Farhat, 2008; Amsallem et al., 2009; Eftang et al., 2010]
- time domain [Drohmann et al., 2011; Dihlmann et al., 2011]
- state space [Amsallem et al., 2012; Washabaugh et al., 2012; Peherstorfer et al., 2013; **Hahn et al., 2014**]

\+ Reduces the ROM dimension

\- Does not improve the ROM *a posteriori*

ROM adaptation: previous state of the art

A priori **adaptation: unique ROM for separate regions of the**

- input space [Amsallem and Farhat, 2008; Amsallem et al., 2009; Eftang et al., 2010]
- time domain [Drohmann et al., 2011; Dihlmann et al., 2011]
- state space [Amsallem et al., 2012; Washabaugh et al., 2012; Peherstorfer et al., 2013; **Hahn et al., 2014**]

\+ Reduces the ROM dimension

\- Does not improve the ROM *a posteriori*

A posteriori **adaptation**

- Revert to the FOM, solve it, and add solution to the basis
  [Arian et al., 2000; Ryckelynck, 2005; Eldred et al., 2009; **Kim and James, 2009**]
- Revert to the FOM in subdomains only
  [**Teng et al., 2015**]

\+ Improves the ROM *a posteriori*

\- Incurs large-scale operations

\- Requires *a priori* identification of subdomains

ROM adaptation: previous state of the art

***A priori* adaptation: unique ROM for separate regions of the**

- input space [Amsallem and Farhat, 2008; Amsallem et al., 2009; Eftang et al., 2010]
- time domain [Drohmann et al., 2011; Dihlmann et al., 2011]
- state space [Amsallem et al., 2012; Washabaugh et al., 2012; Peherstorfer et al., 2013; **Hahn et al., 2014**]

\+ Reduces the ROM dimension

− Does not improve the ROM *a posteriori*

***A posteriori* adaptation**

- Revert to the FOM, solve it, and add solution to the basis [Arian et al., 2000; Ryckelynck, 2005; Eldred et al., 2009; **Kim and James, 2009**]
- Revert to the FOM in subdomains only [**Teng et al., 2015**]

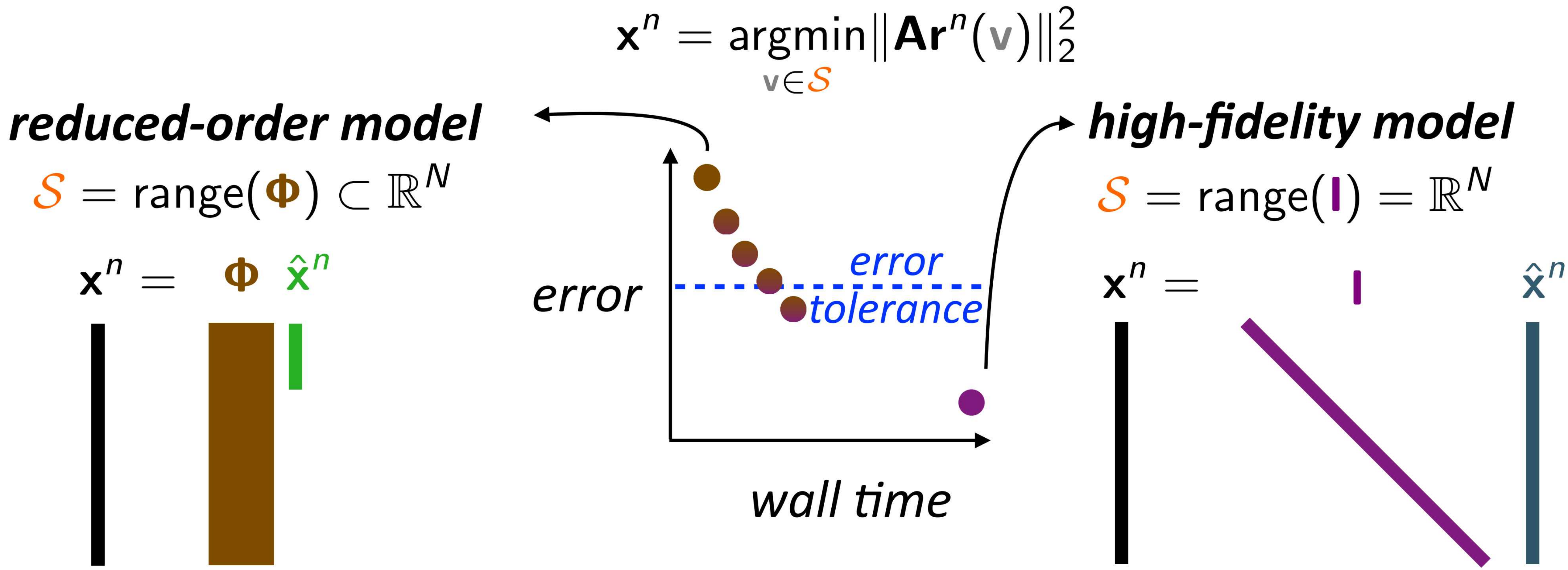
\+ Improves the ROM *a posteriori*

− Incurs large-scale operations

− Requires *a priori* identification of subdomains

Goal: *Cheap, a posteriori, general improvement of the ROM*

Key insight



Idea: *the data provide an initial, low-dim subspace that* ***can be refined*** *to satisfy any error tolerance*

1. Generalization of mesh-adaptive *h*-refinement [Carlberg, 2015]

$$\mathcal{S} = \operatorname{range}(\mathbf{\Phi}_{h\text{-refine}}) \supset \operatorname{range}(\mathbf{\Phi})$$

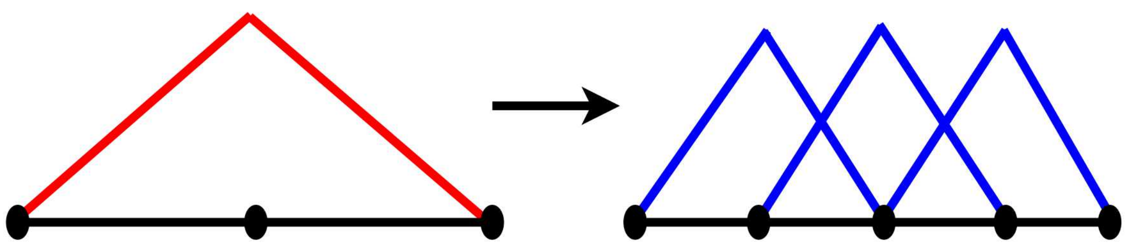
2. Augmented Krylov method [Carlberg, Forstall, Tuminaro, 2016]

$$\mathcal{S} = \operatorname{range}(\mathbf{\Phi}) + \mathcal{K}(A, b)$$

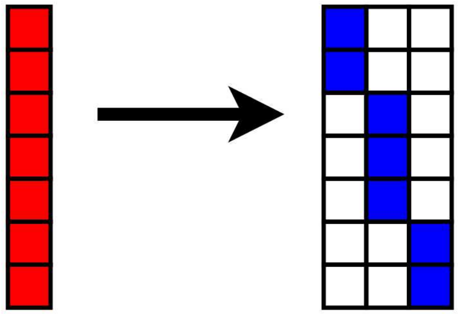
Main idea [Carlberg, 2015]

Model-reduction analogue to mesh-adaptive h-refinement

- ‘Split’ basis vectors

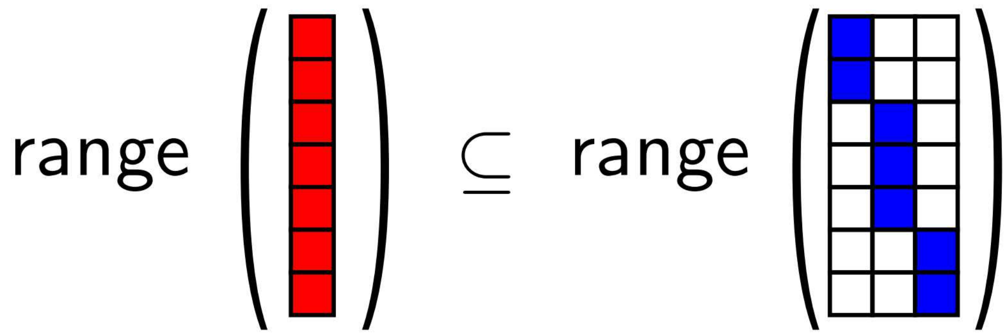


finite-element h-refinement

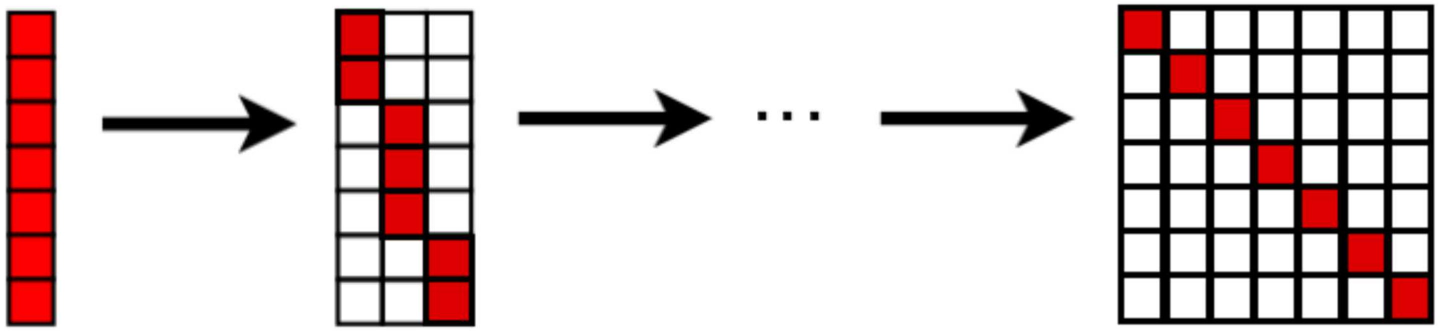


reduced-order-model h-refinement

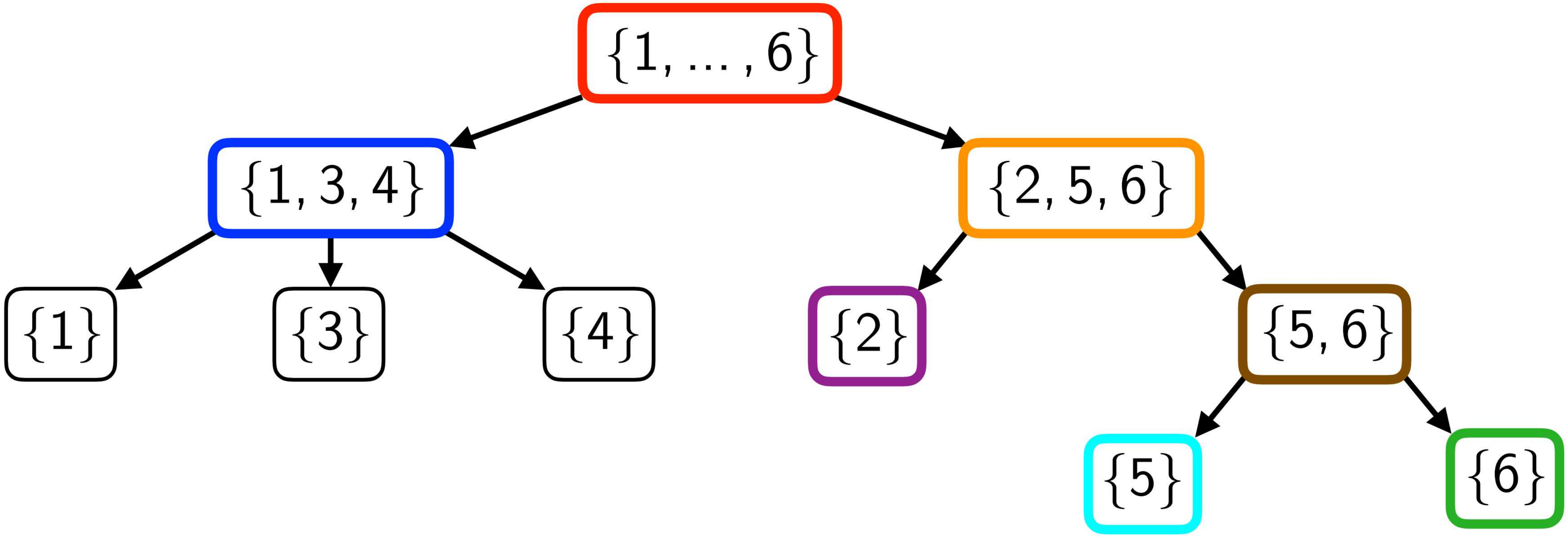
- Generate hierarchical subspaces

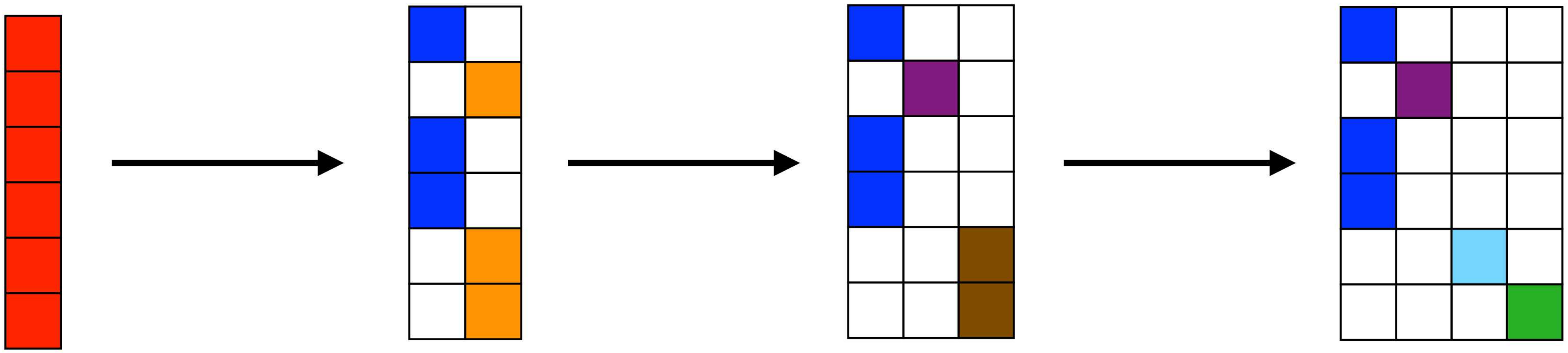


- Converges to the high-fidelity model



Tree encodes splitting

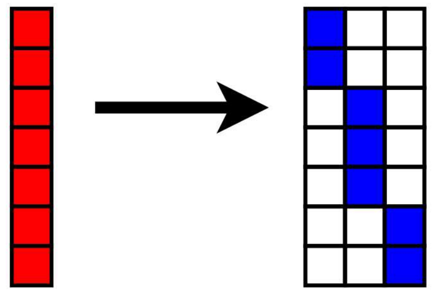




Tree requirements

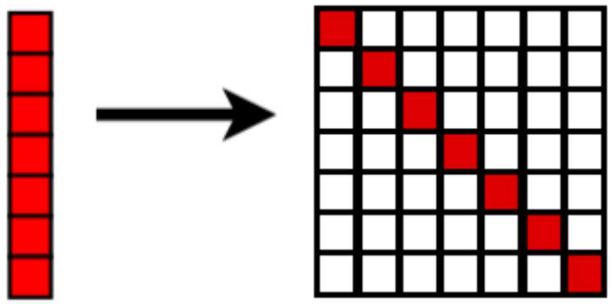
Theorem [Carlberg, 2015]

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 [Carlberg, 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.

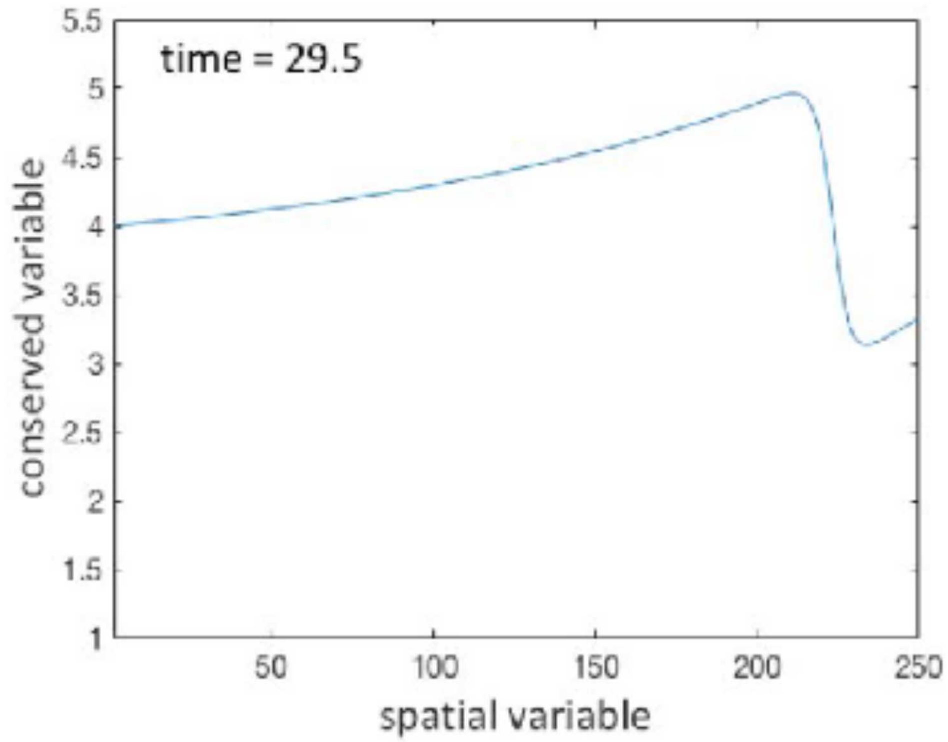


Tree-construction algorithm
- Identifies hierarchy of correlated states via *k*-means clustering
- \+ Ensures theorem conditions are satisfied

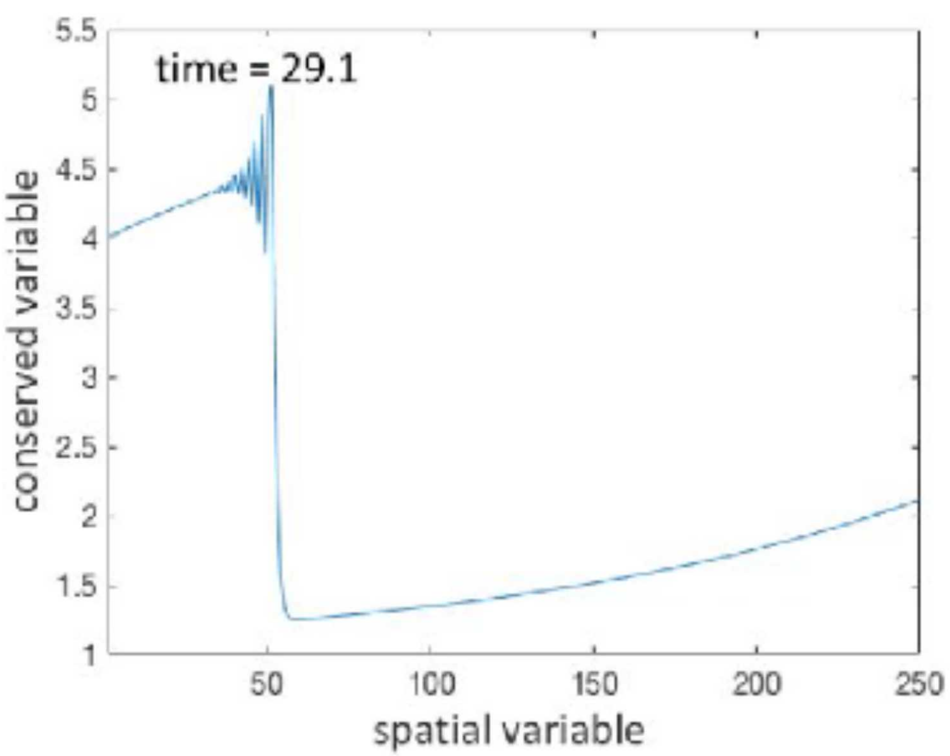
Which vectors to split?
- Dual-weighted-residual error estimation

Illustration: inviscid 1D Burgers' equation

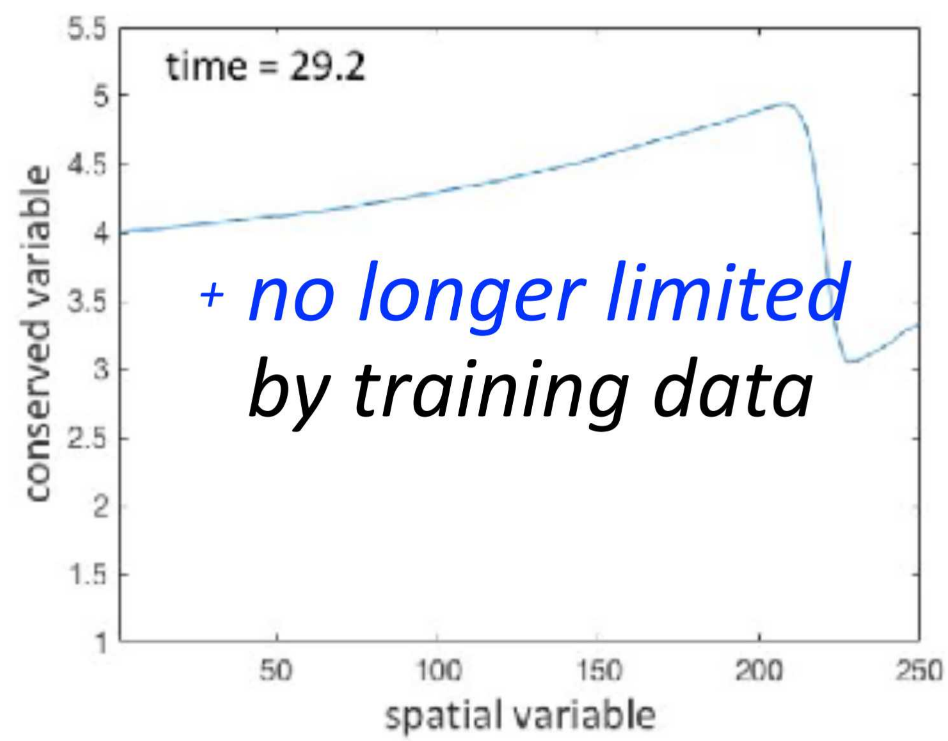
high-fidelity model



reduced-order model (dim 50)



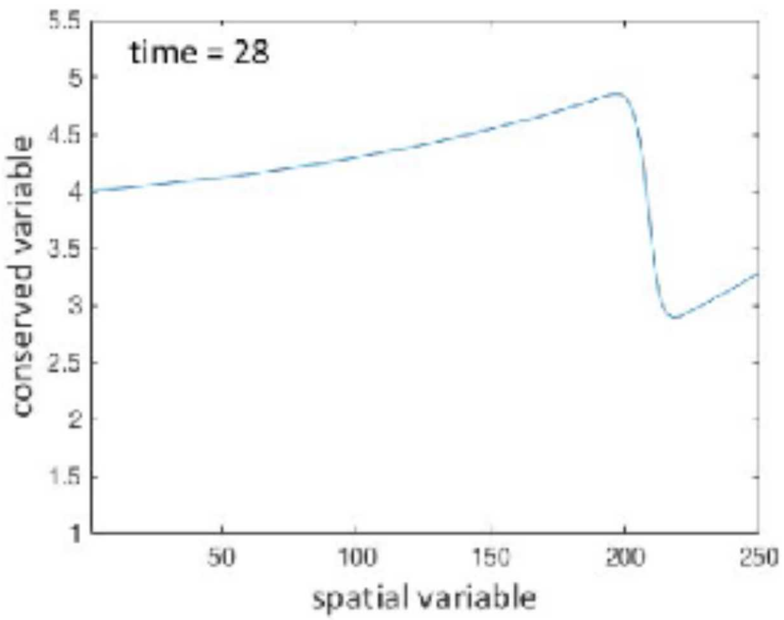
h-adaptive ROM (mean dim 48.5)



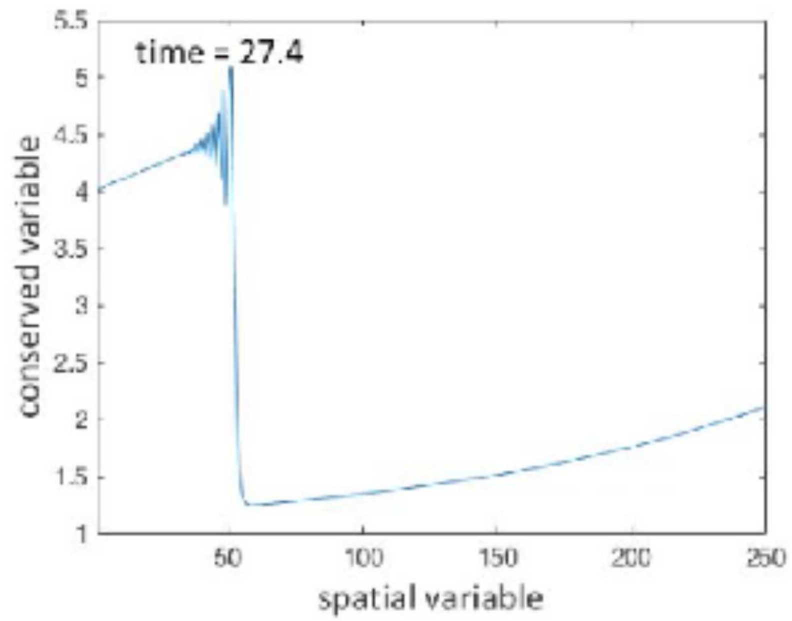
+ *no longer limited by training data*

h-adaptivity enables error control

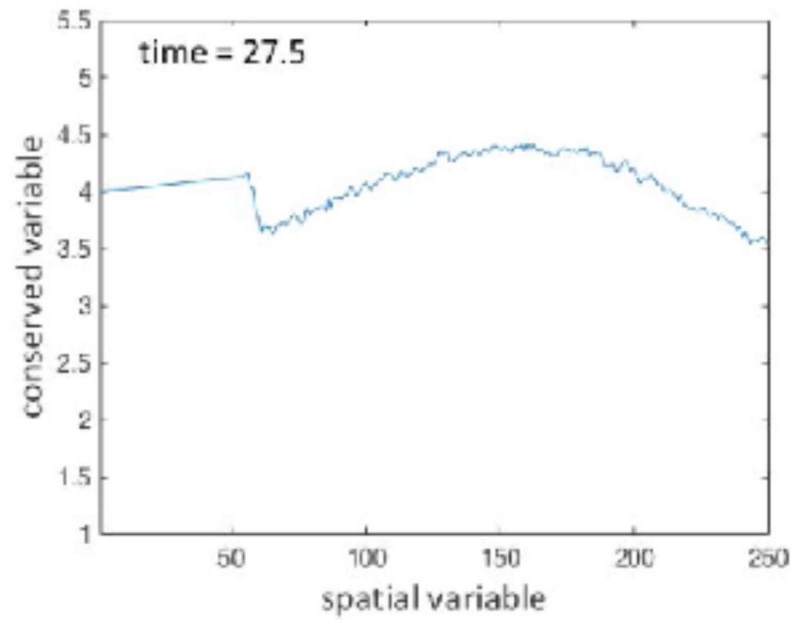
high-fidelity model



reduced-order models

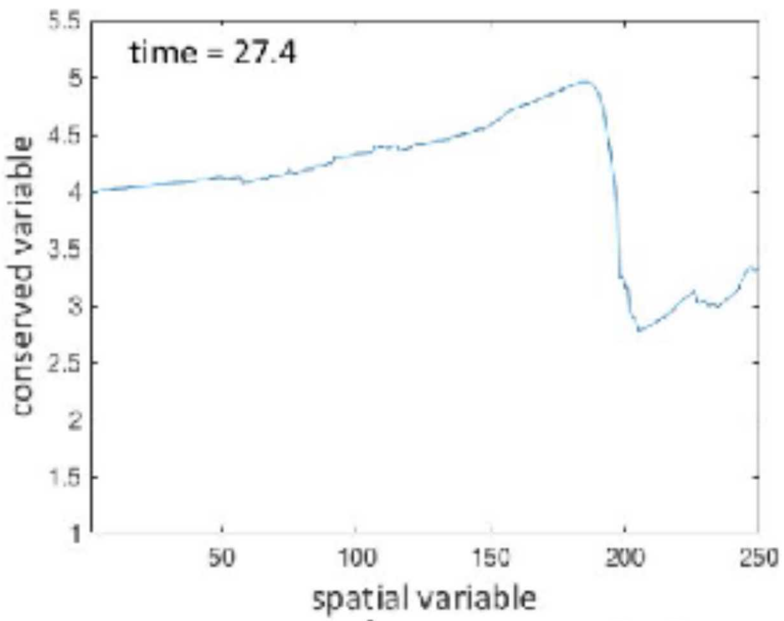


dimension 50

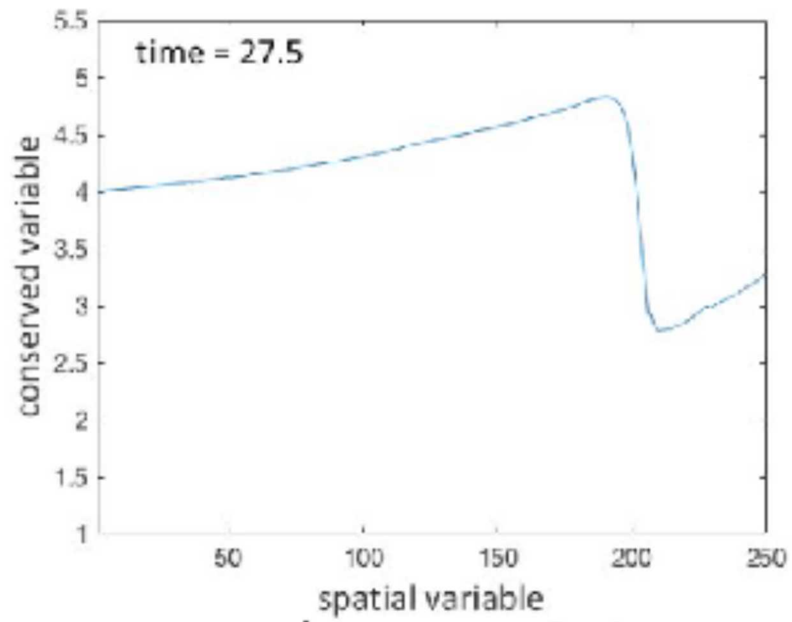


(maximum) dimension 150

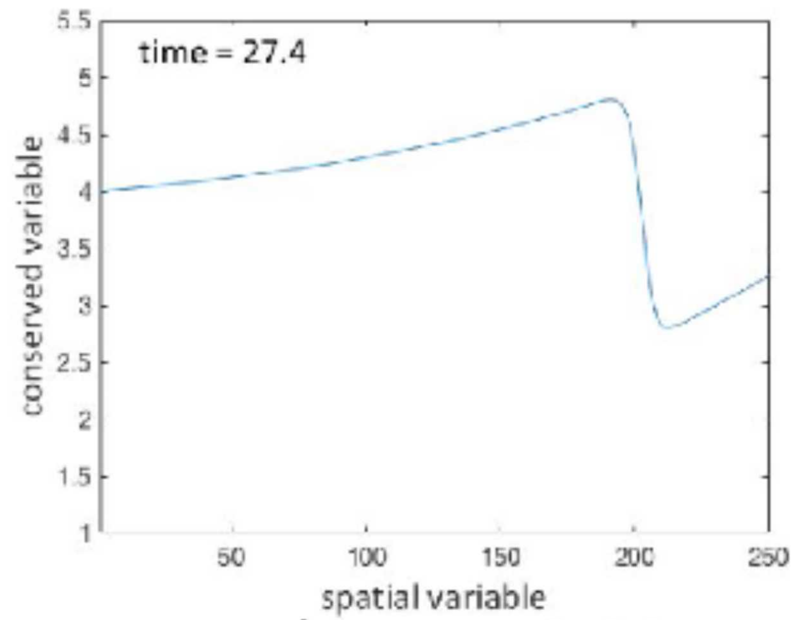
h-adaptive reduced-order models



tolerance 0.3
mean dimension 31.7

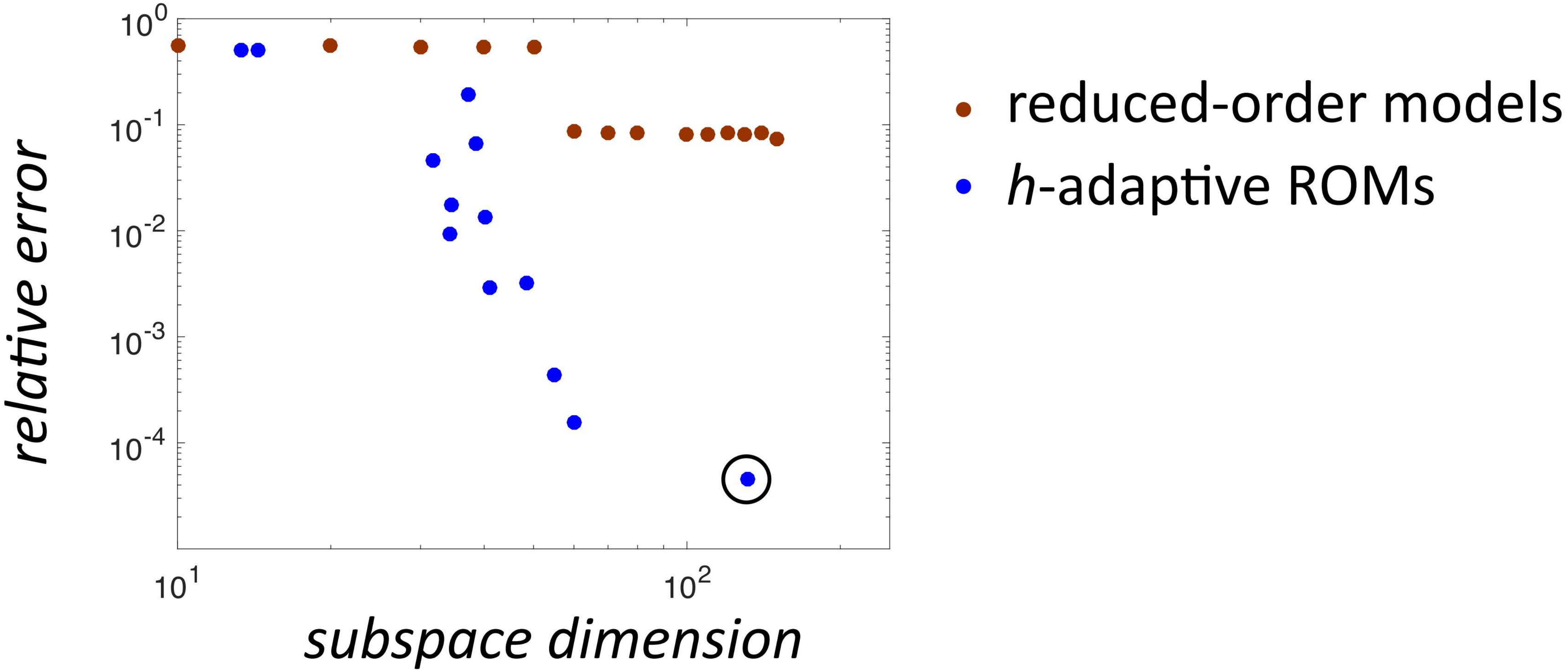


tolerance 0.1
mean dimension 41.0



tolerance 0.01
mean dimension 55.0

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*

My research

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

- *accuracy*: LSPG projection [Carlberg, Bou-Mosleh, Farhat, 2011; Carlberg, Antil, Barone, 2017]
- *low cost*: sample mesh [Carlberg, Farhat, Cortial, Amsallem, 2013]
- *low cost*: reduce temporal complexity
  [Carlberg, Ray, van Bloemen Waanders, 2015; Carlberg, Brencher, Haasdonk, Barth, 2017; Choi and Carlberg, 2017]
- *structure preservation* [Carlberg, Tuminaro, Boggs, 2015; Peng and Carlberg, 2017; Carlberg and Choi, 2017]
- *reliability*: adaptivity [Carlberg, 2015]
- *certification*: machine learning error models
  [Drohmann and Carlberg, 2015*; Trehan, Carlberg, Durlofsky, 2017; Freno and Carlberg, 2018]

* Top 5 most cited papers, SIAM/ASA JUQ, 2015

Discrete-time error bound

Theorem [Carlberg, Antil, Barone, 2017]

If the following conditions hold:

1. $\mathbf{f}(\cdot;t)$ is Lipschitz continuous with Lipschitz constant κ
2. The time step Δt is small enough such that $0 < h := |\alpha_0| - |\beta_0|\kappa\Delta t$,
3. A backward differentiation formula (BDF) time integrator is used,
4. LSPG employs $\mathbf{A} = \mathbf{I}$, then

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

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

Can we use these error bounds for error estimation?

Discrete-time error bound

Theorem [Carlberg, Antil, Barone, 2017]

If the following conditions hold:

1. $\mathbf{f}(\cdot; t)$ is Lipschitz continuous with Lipschitz constant κ
2. The time step Δt is small enough such that $0 < h := |\alpha_0| - |\beta_0|\kappa\Delta t$,
3. A backward differentiation formula (BDF) time integrator is used,
4. LSPG employs $\mathbf{A} = \mathbf{I}$, then

$$\|\mathbf{x}^n - \mathbf{\Phi}\hat{\mathbf{x}}_{\mathrm{G}}^n\|_2 \leq \frac{\gamma_1(\gamma_2)^n \exp(\gamma_3 t^n)}{\gamma_4 + \gamma_5 \Delta t} \max_{j\in\{1,\ldots,N\}} \|\mathbf{r}_{\mathrm{G}}^j(\mathbf{\Phi}\hat{\mathbf{x}}_{\mathrm{G}}^j)\|_2$$

$$\|\mathbf{x}^n - \mathbf{\Phi}\hat{\mathbf{x}}_{\mathrm{LSPG}}^n\|_2 \leq \frac{\gamma_1(\gamma_2)^n \exp(\gamma_3 t^n)}{\gamma_4 + \gamma_5 \Delta t} \max_{j\in\{1,\ldots,N\}} \min_{\hat{\mathbf{v}}} \|\mathbf{r}_{\mathrm{LSPG}}^j(\mathbf{\Phi}\hat{\mathbf{v}})\|_2$$

Can we use these error bounds for error estimation?

- grow exponentially in time
- deterministic: not amenable to uncertainty quantification

Error quantification: previous state of the art

Rigorous error bounds
[Rathinam and Petzold, 2003; Grepl and Patera, 2005; Antoulas, 2005; Hinze and Volkwein, 2005; Carlberg et al., 2017]

- Developed for reduced-order models
  - *Overestimation*: exponential growth for time-dependent problems
  - *Deterministic*: not amenable to uncertainty quantification

Error quantification: previous state of the art

Rigorous error bounds

[Rathinam and Petzold, 2003; Grepl and Patera, 2005; Antoulas, 2005; Hinze and Volkwein, 2005; Carlberg et al., 2017]

- Developed for reduced-order models

– *Overestimation*: exponential growth for time-dependent problems

– *Deterministic*: not amenable to uncertainty quantification

Dual-weighted-residual error estimation

[Babuska and Miller, 1984; Becker and Rannacher, 1996; Rannacher, 1999; Venditti and Darmofal, 2000; Fidkowski, 2007]

- Developed for finite-element, finite-volume, DG discretizations

+ *Accurate*: first-order, coarse-model approximation of error

– *Deterministic*: not amenable to uncertainty quantification

Error quantification: previous state of the art

Rigorous error bounds

[Rathinam and Petzold, 2003; Grepl and Patera, 2005; Antoulas, 2005; Hinze and Volkwein, 2005; Carlberg et al., 2017]

- Developed for reduced-order models
- *Overestimation*: exponential growth for time-dependent problems
- *Deterministic*: not amenable to uncertainty quantification

Dual-weighted-residual error estimation

[Babuska and Miller, 1984; Becker and Rannacher, 1996; Rannacher, 1999; Venditti and Darmofal, 2000; Fidkowski, 2007]

- Developed for finite-element, finite-volume, DG discretizations
+ *Accurate*: first-order, coarse-model approximation of error
- *Deterministic*: not amenable to uncertainty quantification

Data fit: map inputs μ to error

- Developed for general surrogate models
+ *Statistical*: amenable to uncertainty quantification
- *Limited*: does not leverage data informing the error

Error quantification: previous state of the art

Rigorous error bounds

[Rathinam and Petzold, 2003; Grepl and Patera, 2005; Antoulas, 2005; Hinze and Volkwein, 2005; Carlberg et al., 2017]

- Developed for reduced-order models
- *Overestimation*: exponential growth for time-dependent problems
- *Deterministic*: not amenable to uncertainty quantification

Dual-weighted-residual error estimation

[Babuska and Miller, 1984; Becker and Rannacher, 1996; Rannacher, 1999; Venditti and Darmofal, 2000; Fidkowski, 2007]

- Developed for finite-element, finite-volume, DG discretizations
- \+ *Accurate*: first-order, coarse-model approximation of error
- \- *Deterministic*: not amenable to uncertainty quantification

Data fit: map inputs μ to error

- Developed for general surrogate models
- \+ *Statistical*: amenable to uncertainty quantification
- \- *Limited*: does not leverage data informing the error

Goal: *Statistical model that leverages data informing the error*

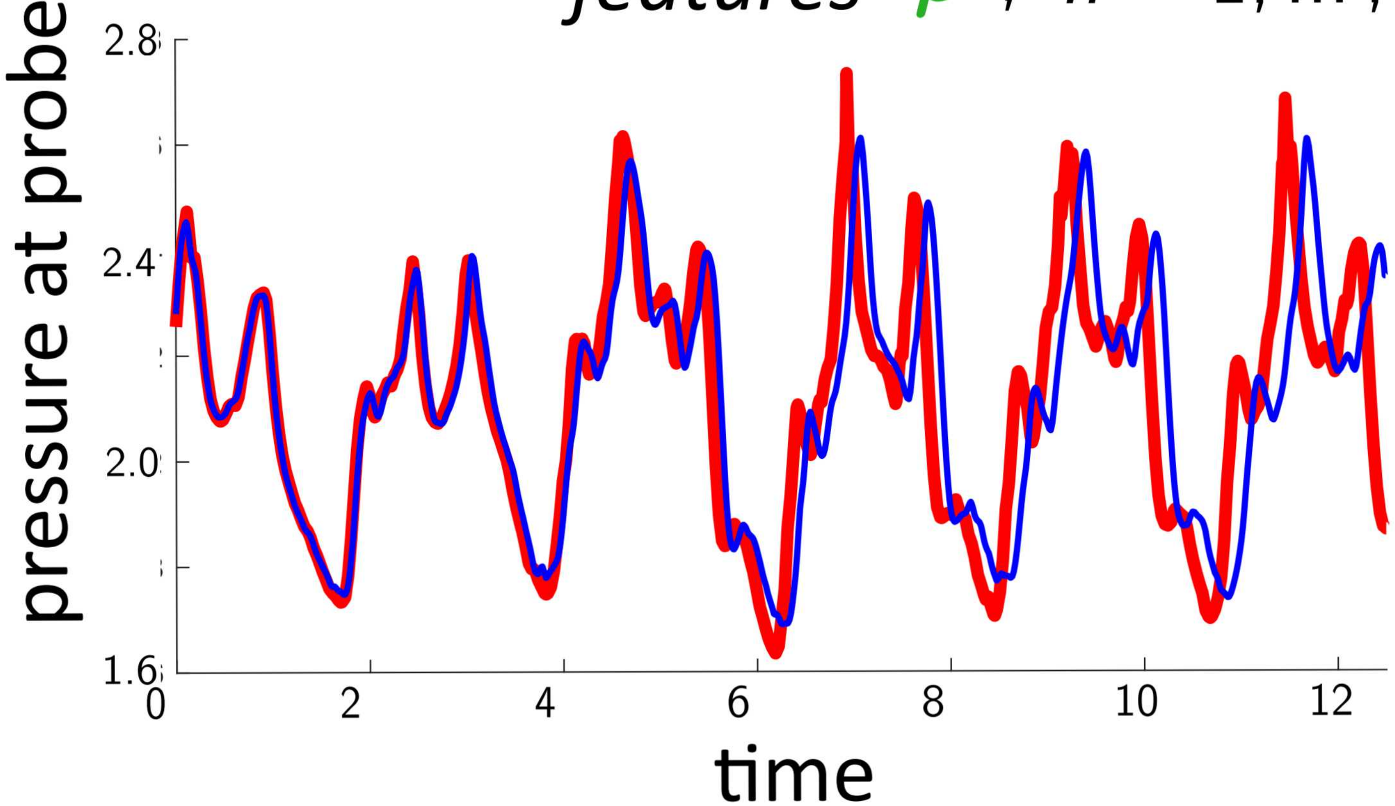
Key insight

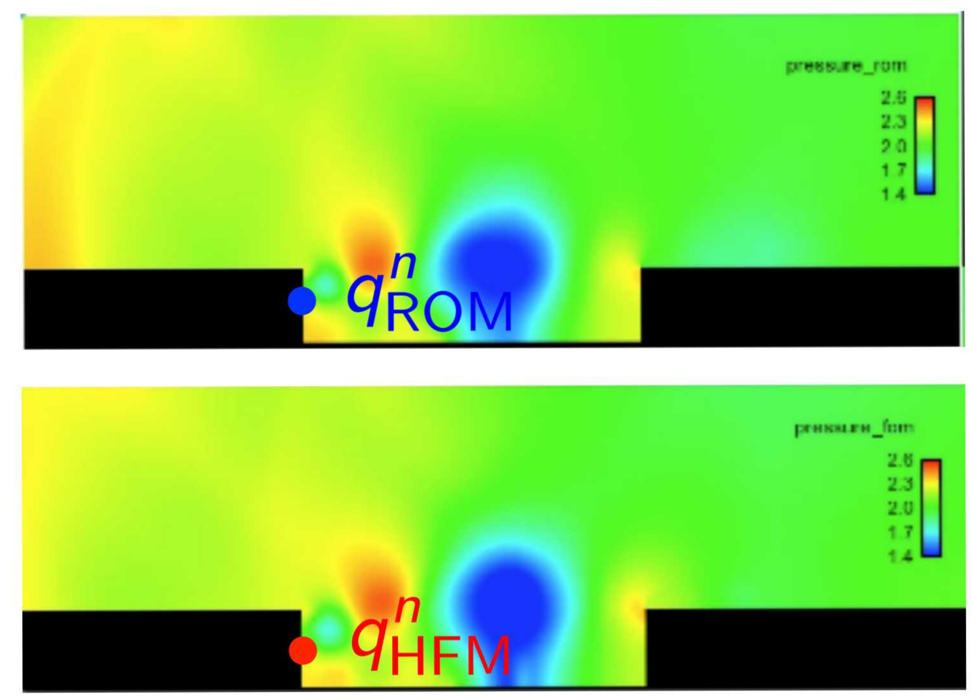
inputs μ → *high-fidelity model* → *outputs* q^n_{HFM}, $n = 1, \ldots, T$

inputs μ → *reduced-order model* → *outputs* q^n_{ROM}, $n = 1, \ldots, T$

↓

features ρ^n, $n = 1, \ldots, T$



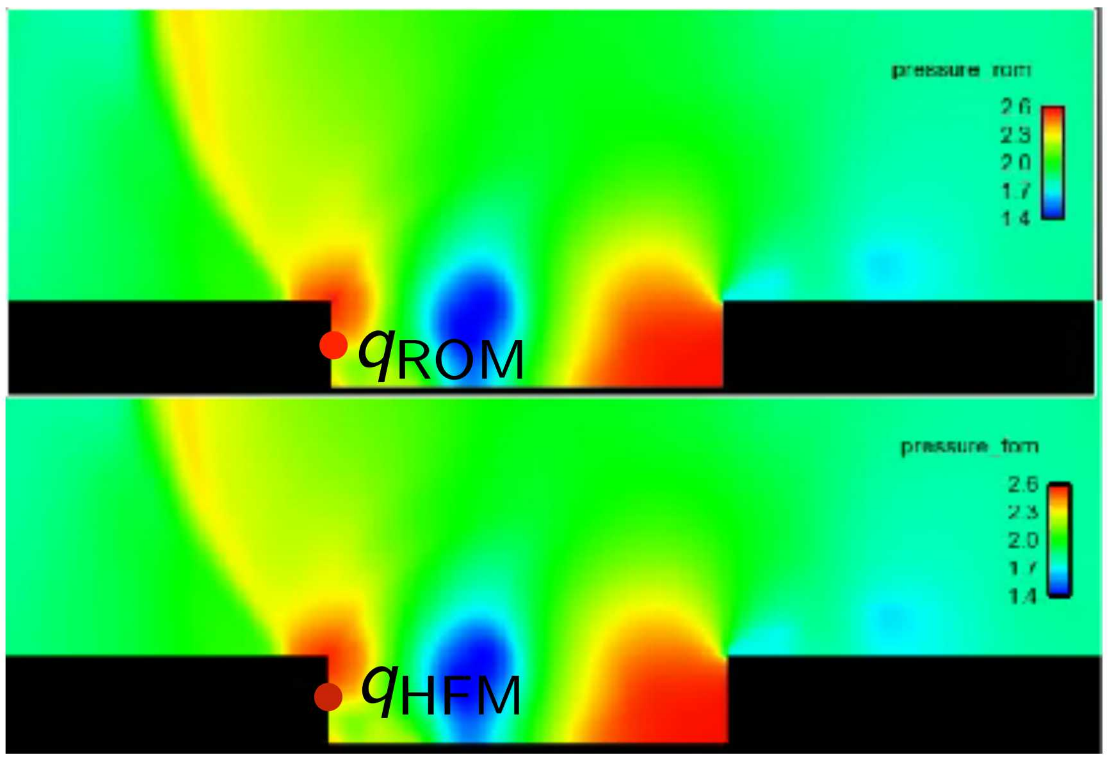


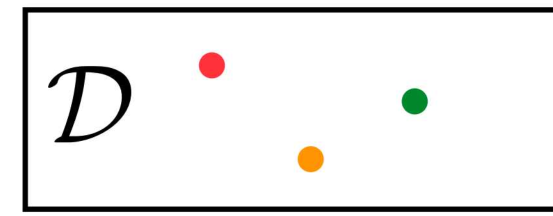
Reduced-order models generate features ρ^n that may inform its error

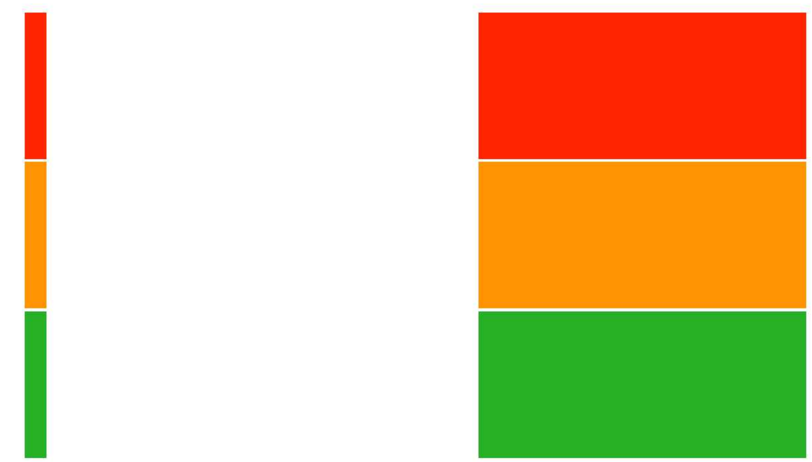
Idea: *regression model that predicts error* $q^n_{\text{HFM}} - q^n_{\text{ROM}}$ *from features* ρ^n

Training and machine learning: error modeling

1. *Training:* Solve high-fidelity and reduced-order models for $\mu \in \mathcal{D}_{\text{training}}$
2. *Machine learning:* Construct regression model
3. *Reduction:* predict reduced-order-model error for $\mu \in \mathcal{D}_{\text{query}} \setminus \mathcal{D}_{\text{training}}$





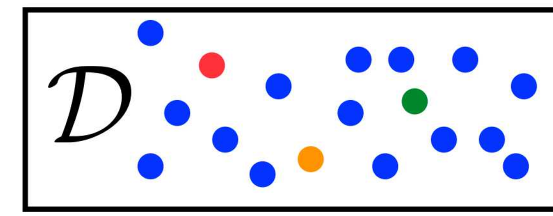


features $\boldsymbol{\rho}^n$ $\longrightarrow$ *regression model* $\longrightarrow$ *error prediction* $\hat{\delta}^n \approx q^n_{\text{HFM}} - q^n_{\text{ROM}}$

- *Regression methods*: Gaussian process, random forest, SVM, neural nets

Regression model for the error

1. *Training:* Solve high-fidelity and reduced-order models for $\mu \in \mathcal{D}_{\text{training}}$
2. *Machine learning:* Construct regression model
3. *Reduction:* predict reduced-order-model error for $\mu \in \mathcal{D}_{\text{query}} \setminus \mathcal{D}_{\text{training}}$



inputs μ → *reduced-order model* → *outputs* q^n_{ROM}, $n = 1, \dots, T$

↓

features ρ^n, $n = 1, \dots, N$

↓

regression model → *statistical error prediction* $\hat{\delta}^n \approx q^n_{\text{HFM}} - q^n_{\text{ROM}}$, $n = 1, \dots, T$

$$q^n_{\text{HFM}} \approx q^n_{\text{ROM}} + \hat{\delta}^n$$

+ *Statistical model of high-fidelity-model output*

Physics-based feature engineering to determine ρ^n

Physics-based feature engineering

- error bound

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

- terms in the error bound

$$\boldsymbol{\rho}^n = [\tau,\ h,\ \|\mathbf{r}^n(\boldsymbol{\Phi}\hat{\mathbf{x}}^n)\|_2]$$

- dual-weighted residual

1st-order output approx: $q(\mathbf{x}^n) \approx q(\boldsymbol{\Phi}\hat{\mathbf{x}}^n) + \frac{\partial q}{\partial \mathbf{x}}(\boldsymbol{\Phi}\hat{\mathbf{x}}^n)(\mathbf{x}^n - \boldsymbol{\Phi}\hat{\mathbf{x}}^n)$

1st-order residual approx: $\mathbf{r}^n(\mathbf{x}^n) = \mathbf{0} \approx \mathbf{r}^n(\boldsymbol{\Phi}\hat{\mathbf{x}}^n) + \frac{\partial \mathbf{r}^n}{\partial \mathbf{x}}(\boldsymbol{\Phi}\hat{\mathbf{x}}^n)(\mathbf{x}^n - \boldsymbol{\Phi}\hat{\mathbf{x}}^n)$

$$q(\mathbf{x}^n) - q(\boldsymbol{\Phi}\hat{\mathbf{x}}^n) \approx (\mathbf{y}^n)^T\mathbf{r}^n(\boldsymbol{\Phi}\hat{\mathbf{x}}^n) = \boldsymbol{\rho}^n$$

$$\frac{\partial \mathbf{r}^n}{\partial \mathbf{x}}(\boldsymbol{\Phi}\hat{\mathbf{x}}^n)^T\mathbf{y}^n = -\frac{\partial q}{\partial \mathbf{x}}(\boldsymbol{\Phi}\hat{\mathbf{x}}^n)^T$$

- *Costly solve*

Physics-based feature engineering

- error bound

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

- terms in the error bound

$$\boldsymbol{\rho}^n = [\tau,\ h,\ \|\mathbf{r}^n(\mathbf{\Phi}\hat{\mathbf{x}}^n)\|_2]$$

- dual-weighted residual

1st-order output approx: $q(\mathbf{x}^n) \approx q(\mathbf{\Phi}\hat{\mathbf{x}}^n) + \frac{\partial q}{\partial \mathbf{x}}(\mathbf{\Phi}\hat{\mathbf{x}}^n)(\mathbf{x}^n - \mathbf{\Phi}\hat{\mathbf{x}}^n)$

1st-order residual approx: $\mathbf{r}^n(\mathbf{x}^n) = \mathbf{0} \approx \mathbf{r}^n(\mathbf{\Phi}\hat{\mathbf{x}}^n) + \frac{\partial \mathbf{r}^n}{\partial \mathbf{x}}(\mathbf{\Phi}\hat{\mathbf{x}}^n)(\mathbf{x}^n - \mathbf{\Phi}\hat{\mathbf{x}}^n)$

$$q(\mathbf{x}^n) - q(\mathbf{\Phi}\hat{\mathbf{x}}^n) \approx (\tilde{\mathbf{y}}^n)^T\mathbf{r}^n(\mathbf{\Phi}\hat{\mathbf{x}}^n) = \boldsymbol{\rho}^n$$

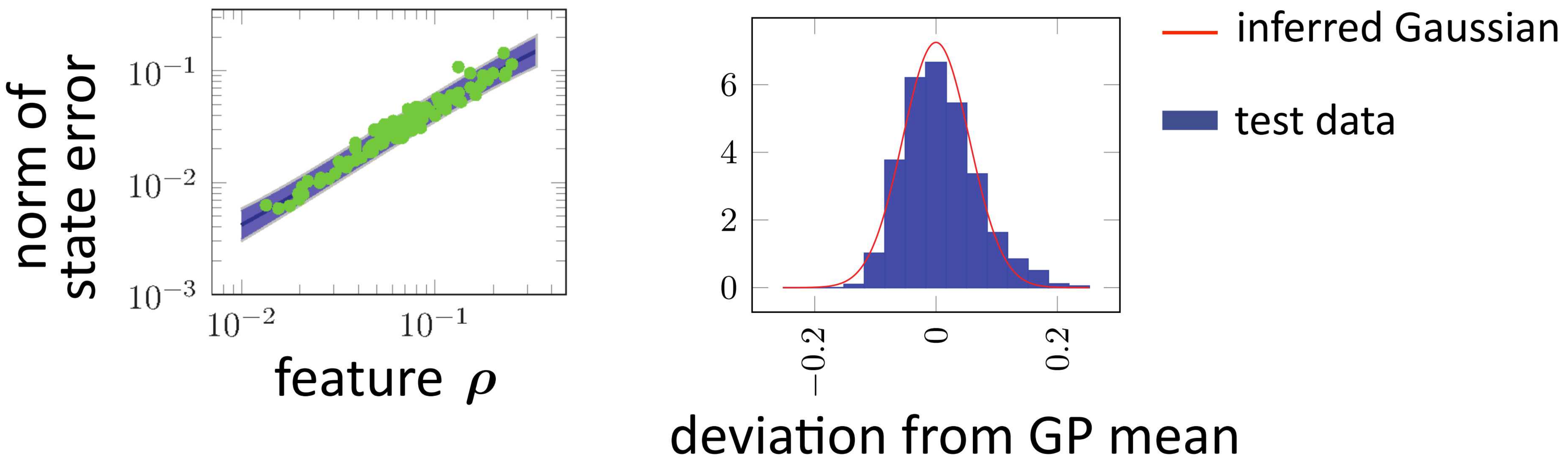
$$\tilde{\mathbf{y}}^n = \underset{\hat{\mathbf{v}}}{\operatorname{argmin}}\left\|\frac{\partial \mathbf{r}^n}{\partial \mathbf{x}}(\mathbf{\Phi}\hat{\mathbf{x}}^n)^T\mathbf{\Phi}\hat{\mathbf{v}} + \frac{\partial q}{\partial \mathbf{x}}(\mathbf{\Phi}\hat{\mathbf{x}}^n)^T\right\|_2$$

\+ *Inexpensive solve*

- application-specific quantities

Application 1: Poisson equation [Drohmann, Carlberg, 2015]

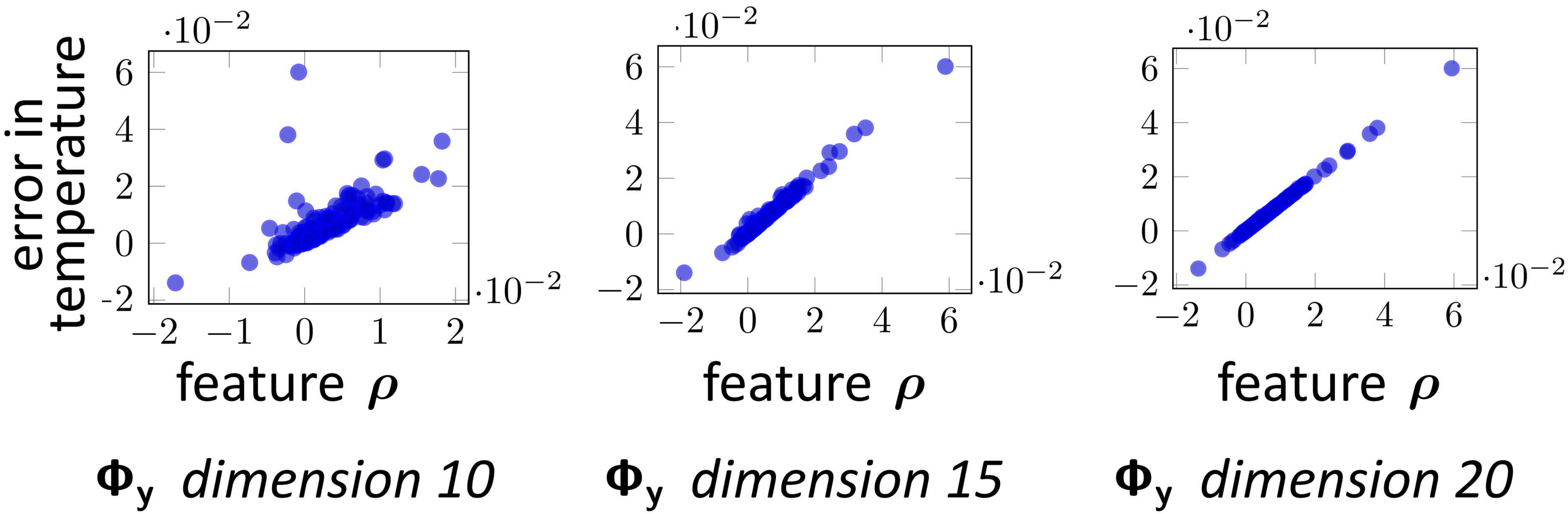
- *error:* norm of state error $\|\mathbf{x} - \boldsymbol{\Phi}\hat{\mathbf{x}}\|$
- *1 feature ρ :* residual norm $\|\mathbf{r}(\boldsymbol{\Phi}\hat{\mathbf{x}})\|_2$
- *regression:* Gaussian process



+ low-variance model of the error
+ numerically validated on test set
- error bound overproduction as high as 8.0

Application 1: Poisson equation [Drohmann, Carlberg, 2015]

- *error:* error in temperature at a point $T(\mathbf{x}) - T(\mathbf{\Phi}\hat{\mathbf{x}})$
- *1 feature ρ:* dual-weighted residual $(\tilde{\mathbf{y}}^n)^T \mathbf{r}^n(\mathbf{\Phi}\hat{\mathbf{x}}^n)$
- *regression:* Gaussian process



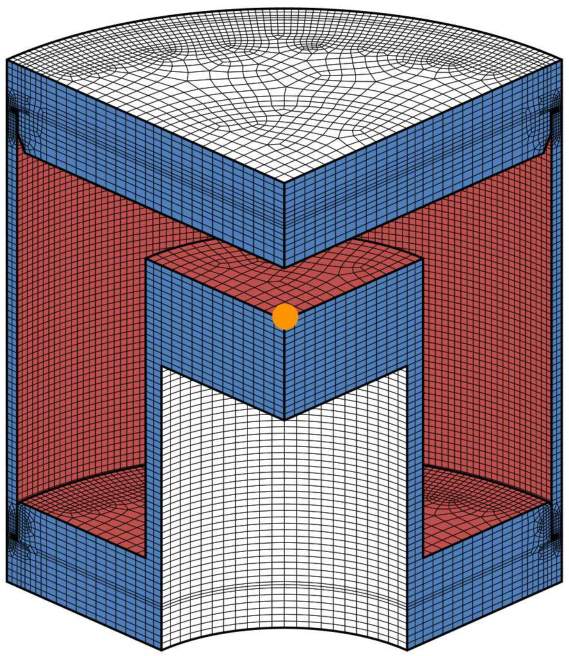
+ *uncertainty control:* variance reduced as feature improves

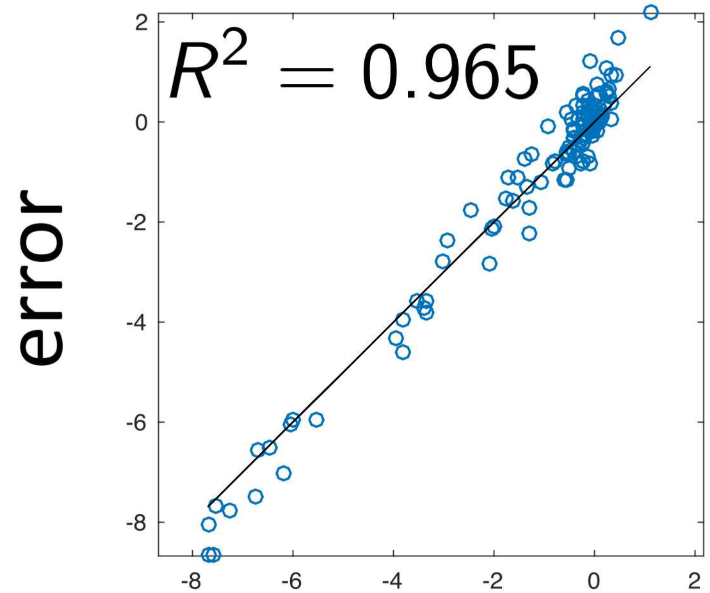
- *Other application*: rigorous integration with Bayesian inference
[Carlberg, Uy, Lu, Morzfeld, 2017]

Application 2: nonlinear static mechanical response

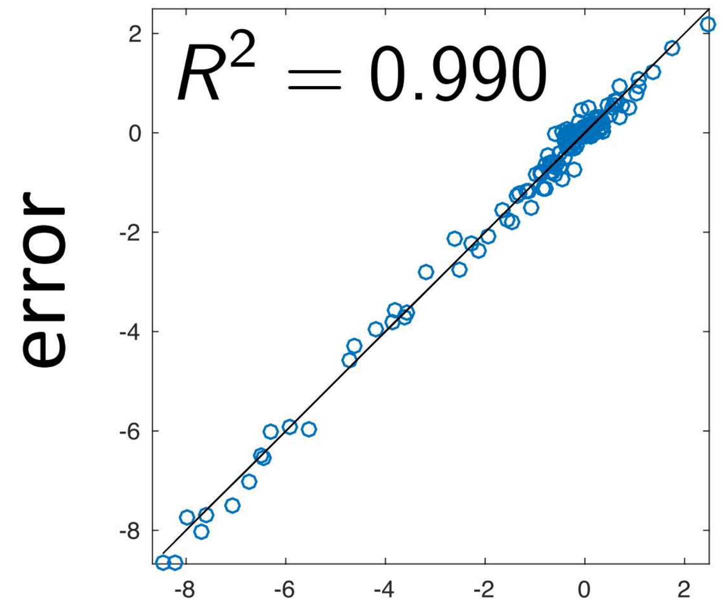
[Freno, Carlberg, 2017]

- *high-fidelity model dimension:* 2.8×10^5
- *reduced-order model dimension:* 6
- *inputs* $\boldsymbol{\mu}$*:* elastic modulus, Poisson ratio
- *error:* error in *y*-displacement at point
- *50 features* $\boldsymbol{\rho}$*:* residual approx $(\mathbf{P}\boldsymbol{\Phi}_{\mathbf{r}})^{+}\mathbf{P}\mathbf{r}^{n}$, inputs $\boldsymbol{\mu}$
- *regression:* random forest, SVM, *k*-NN

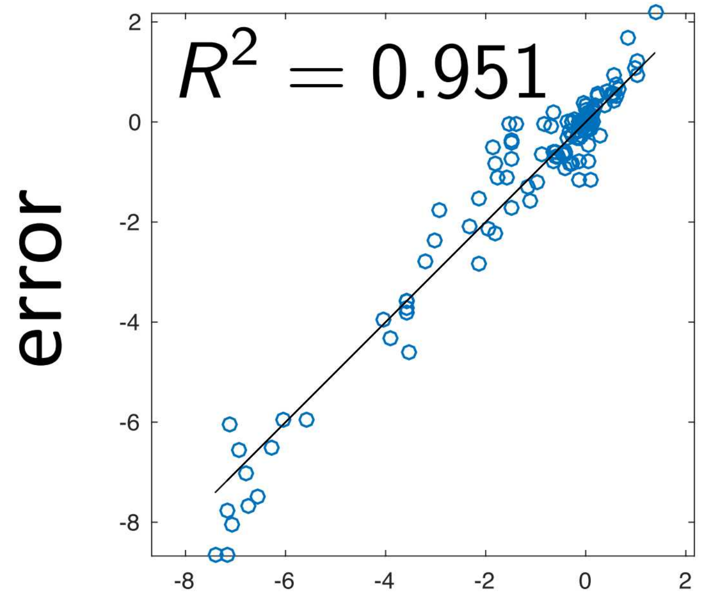




random forest error prediction



support vector machine error prediction



k-NN error prediction

+ *ML methods yield low-variance error predictions*

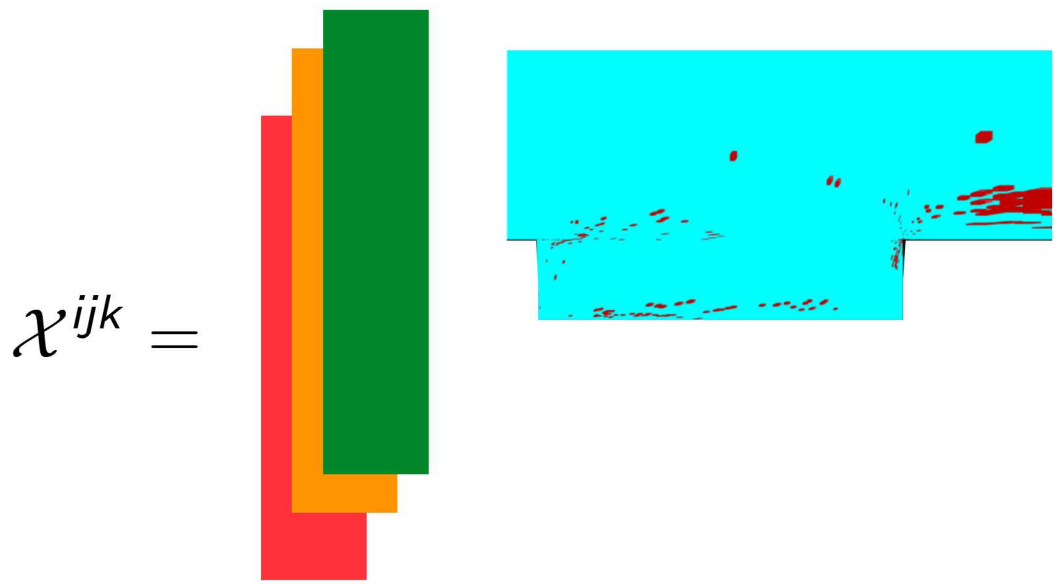
- *Other application*: nonlinear oil–water flow [Trehan, Carlberg, Durlofsky, 2017]

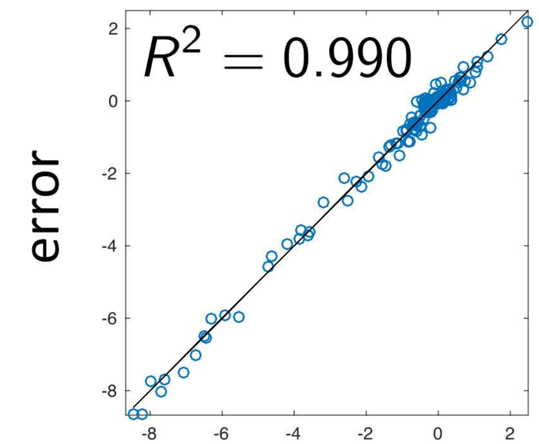
Summary

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

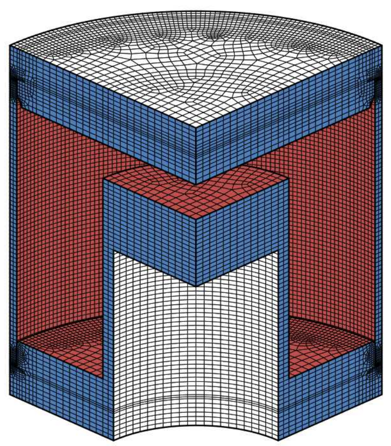
- *accuracy*: LSPG projection [Carlberg, Bou-Mosleh, Farhat, 2011; Carlberg, Antil, Barone, 2017]
- *low cost*: sample mesh [Carlberg, Farhat, Cortial, Amsallem, 2013]
- *low cost*: reduce temporal complexity
  [Carlberg, Ray, van Bloemen Waanders, 2015; Carlberg, Brencher, Haasdonk, Barth, 2017; Choi and Carlberg, 2017]
- *structure preservation* [Carlberg, Tuminaro, Boggs, 2015; Peng and Carlberg, 2017; Carlberg and Choi, 2017]
- *reliability*: adaptivity [Carlberg, 2015]
- *certification*: machine learning error models
  [Drohmann and Carlberg, 2015; Trehan, Carlberg, Durlofsky, 2017; Freno and Carlberg, 2017]

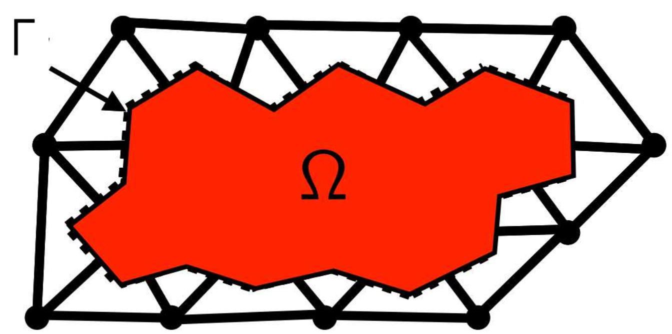
Questions?

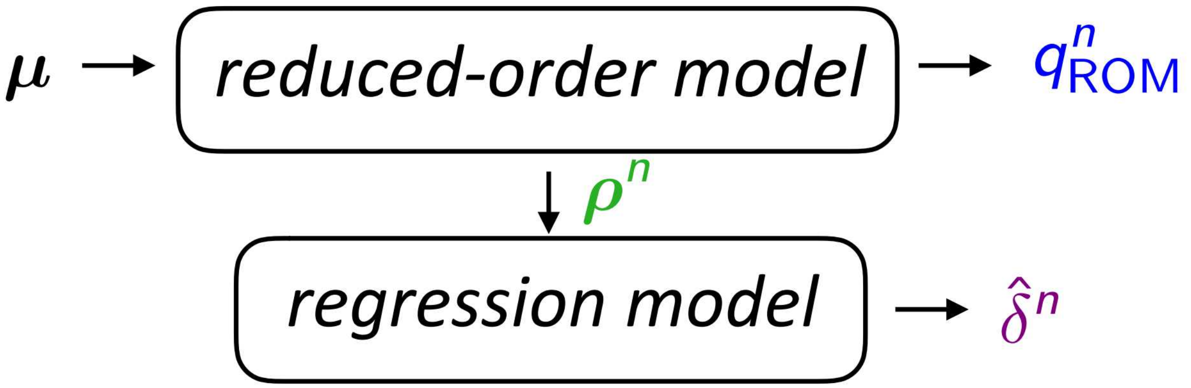


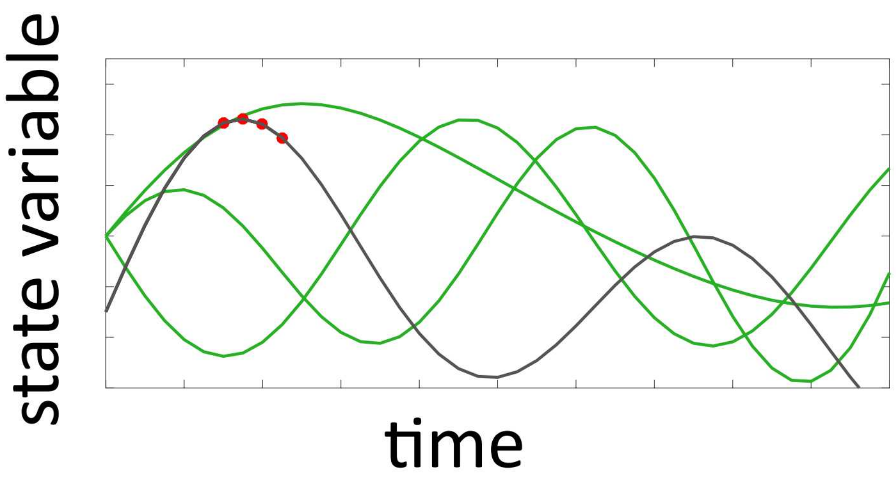


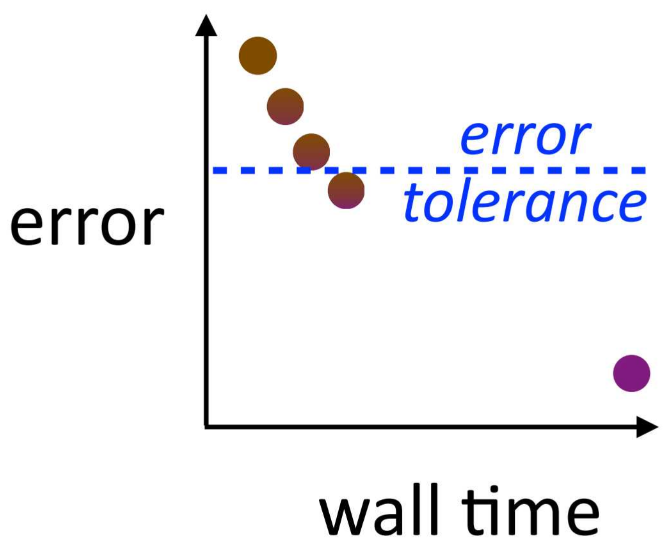
support vector machine
error prediction

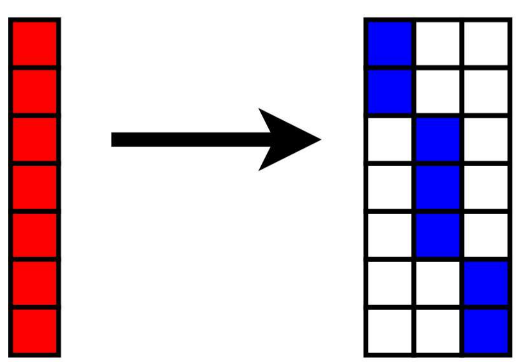












Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525