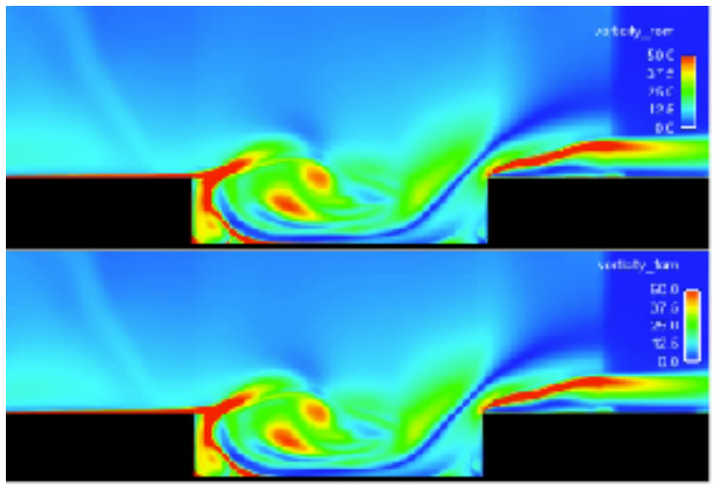
SAND2017-10710PE

Breaking computational barriers

Using data to enable extreme-scale simulations for UQ and design



Kevin Carlberg

Sandia National Laboratories

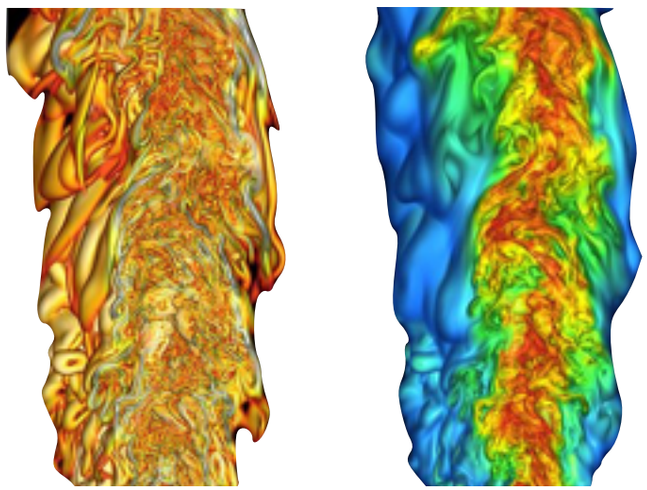
SILO Seminar Series

University of Wisconsin, Madison

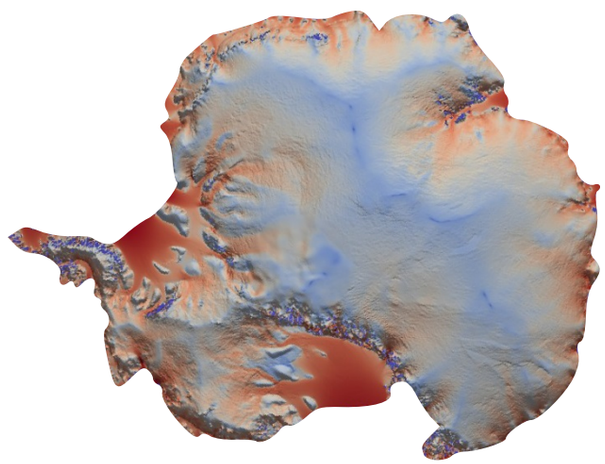
October 11, 2017

High-fidelity simulation

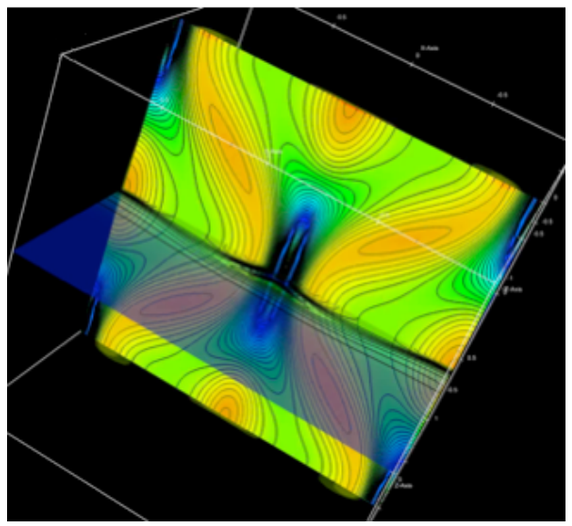
+ Indispensable across science and engineering
- *High fidelity*: extreme-scale nonlinear dynamical system models



Turbulent reacting flows
courtesy J. Chen, Sandia



Antarctic ice sheet modeling
courtesy R. Tuminaro, Sandia



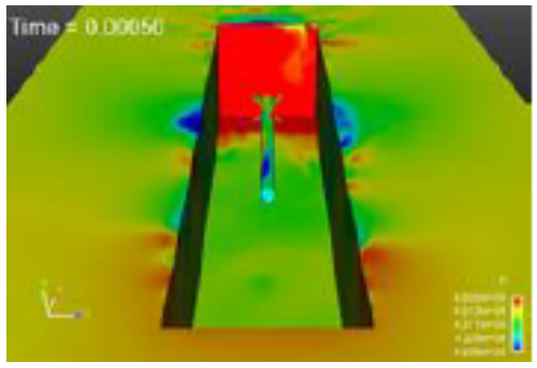
Magnetohydrodynamics
courtesy J. Shadid, Sandia

computational barrier

Many-query problems

- uncertainty propagation
- Bayesian inference
- multi-objective optimization
- stochastic optimization

High-fidelity simulation: B61 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 problems

- explore flight envelope
- quantify effects of uncertainties on store load
- robust design of store and cavity

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."

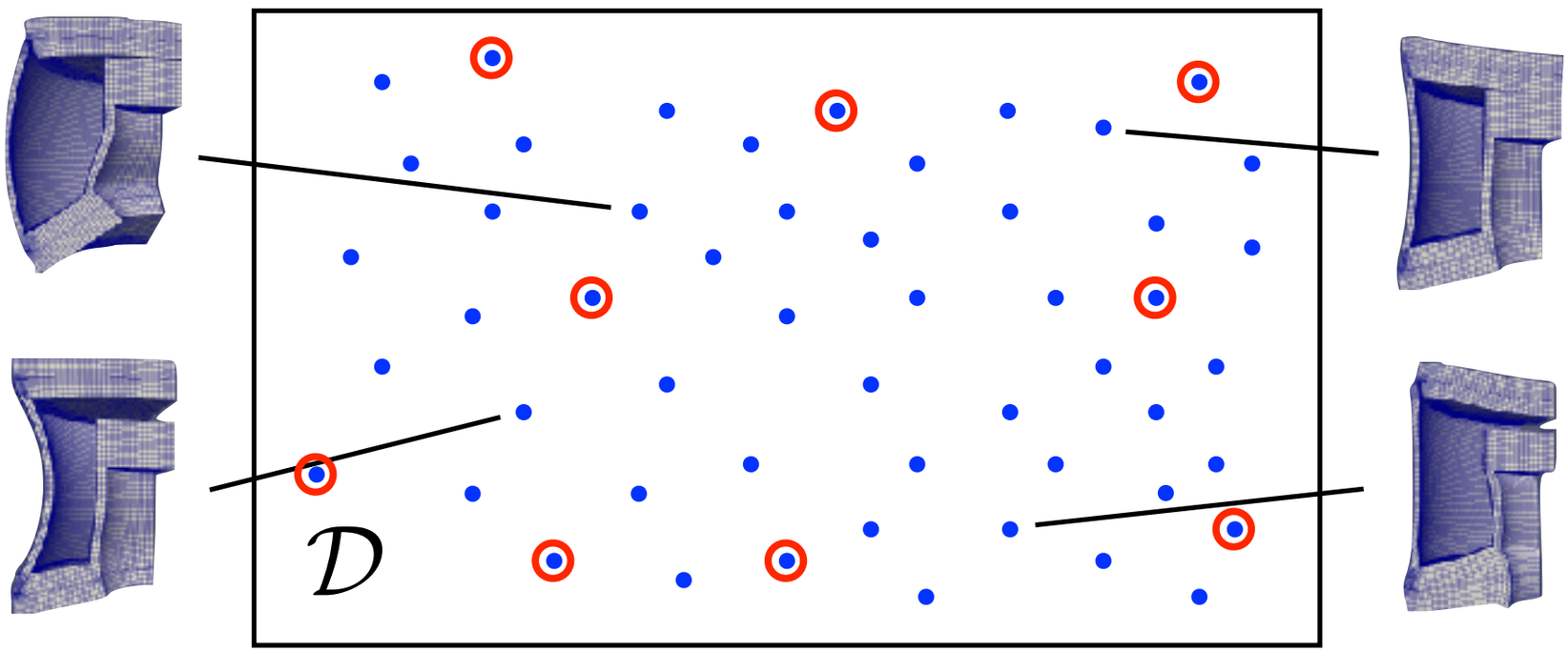
"To enable high-fidelity CFD for multi-disciplinary analysis and design, the speed of computation must be increased by orders of magnitude."

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

Approach: exploit simulation data

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

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



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

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

My research

Enable extreme-scale simulations for many-query problems

1. **Nonlinear model reduction**
   - \+ reduces model dimensionality and complexity
   - ‣ LSPG projection [Carlberg, Bou-Mosleh, Farhat, 2011; Carlberg, Antil, Barone, 2017]
   - ‣ sample mesh [Carlberg, Farhat, Cortial, Amsallem, 2013]
   - ‣ structure preservation [Carlberg, Tuminaro, Boggs, 2015; Peng and Carlberg, 2017; Carlberg and Choi, 2017]

2. **Data-driven error modeling**
   - \+ rigorously accounts for model-reduction error
   - ‣ regression error modeling
     [Drohmann and Carlberg, 2015; Trehan, Carlberg, Durlofsky, 2017; Freno and Carlberg, 2017]

3. **Data-driven numerical solvers**
   - \+ improves performance of numerical solvers
   - ‣ adaptive subspaces [Carlberg, 2015; Carlberg, Forstall, Tuminaro, 2016]
   - ‣ reduce temporal complexity
     [Carlberg, Ray, van Bloemen Waanders, 2015; Carlberg, Brencher, Haasdonk, Barth, 2017; Choi and Carlberg, 2017]

My research

Enable extreme-scale simulations for many-query problems

1. **Nonlinear model reduction**
   - \+ reduces model dimensionality and complexity
   - ‣ LSPG projection [Carlberg, Bou-Mosleh, Farhat, 2011; Carlberg, Antil, Barone, 2017]
   - ‣ sample mesh [Carlberg, Farhat, Cortial, Amsallem, 2013]
   - ‣ structure preservation [Carlberg, Tuminaro, Boggs, 2015; Peng and Carlberg, 2017; Carlberg and Choi, 2017]

2. **Data-driven error modeling**
   - \+ rigorously accounts for model-reduction error
   - ‣ regression error modeling
     [Drohmann and Carlberg, 2015; Trehan, Carlberg, Durlofsky, 2017; Freno and Carlberg, 2017]

3. **Data-driven numerical solvers**
   - \+ improves performance of numerical solvers
   - ‣ adaptive subspaces [Carlberg, 2015; Carlberg, Forstall, Tuminaro, 2016]
   - ‣ reduce temporal complexity
     [Carlberg, Ray, van Bloemen Waanders, 2015; Carlberg, Brencher, Haasdonk, Barth, 2017; Choi and Carlberg, 2017]

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*: *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*: *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*: often unstable

\- *Expensive*: projection insufficient for speedup

\- *Structure not preserved*: dynamical-system properties ignored

***Goal**: accurate, inexpensive, structure-preserving nonlinear model reduction*

My research

Enable extreme-scale simulations for many-query problems

1. **Nonlinear model reduction**
   - \+ reduces model dimensionality and complexity
   - *accuracy*: LSPG projection [Carlberg, Bou-Mosleh, Farhat, 2011; Carlberg, Antil, Barone, 2017]
   - *low cost*: sample mesh [Carlberg, Farhat, Cortial, Amsallem, 2013]
   - *structure preservation* [Carlberg, Tuminaro, Boggs, 2015; Peng and Carlberg, 2017; Carlberg and Choi, 2017]

2. **Data-driven error modeling**
   - \+ rigorously accounts for model-reduction error
   - regression error modeling
     [Drohmann and Carlberg, 2015; Trehan, Carlberg, Durlofsky, 2017; Freno and Carlberg, 2017]

3. **Data-driven numerical solvers**
   - \+ improves performance of numerical solvers
   - adaptive subspaces [Carlberg, 2015; Carlberg, Forstall, Tuminaro, 2016]
   - reduce temporal complexity
     [Carlberg, Ray, van Bloemen Waanders, 2015; Carlberg, Brencher, Haasdonk, Barth, 2017; Choi and Carlberg, 2017]

My research

Enable extreme-scale simulations for many-query problems

1. **Nonlinear model reduction**
   - \+ reduces model dimensionality and complexity
   - *accuracy*: LSPG projection [Carlberg, Bou-Mosleh, Farhat, 2011*; Carlberg, Antil, Barone, 2017]
   - *low cost*: sample mesh [Carlberg, Farhat, Cortial, Amsallem, 2013]
   - *structure preservation* [Carlberg, Tuminaro, Boggs, 2015; Peng and Carlberg, 2017; Carlberg and Choi, 2017]

2. **Data-driven error modeling**
   - \+ rigorously accounts for model-reduction error
   - regression error modeling
     [Drohmann and Carlberg, 2015; Trehan, Carlberg, Durlofsky, 2017; Freno and Carlberg, 2017]

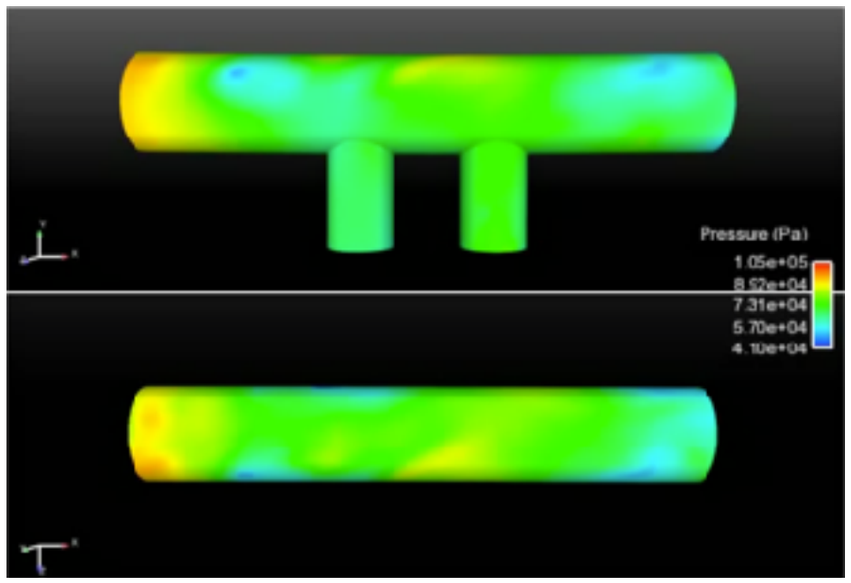
3. **Data-driven numerical solvers**
   - \+ improves performance of numerical solvers
   - adaptive subspaces [Carlberg, 2015; Carlberg, Forstall, Tuminaro, 2016]
   - reduce temporal complexity
     [Carlberg, Ray, van Bloemen Waanders, 2015; Carlberg, Brencher, Haasdonk, Barth, 2017; Choi and Carlberg, 2017]

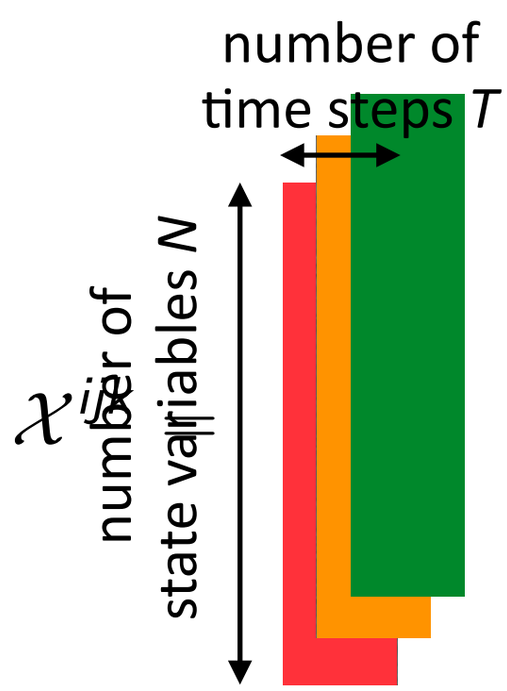
* #2 most-cited paper, Int J Numer Meth Eng, 2011

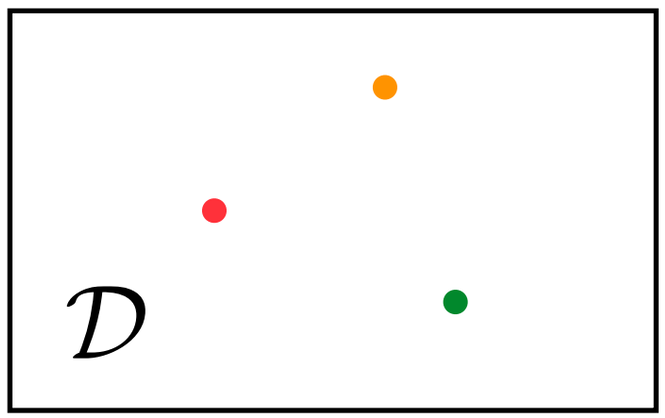
Training simulations: state tensor

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

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





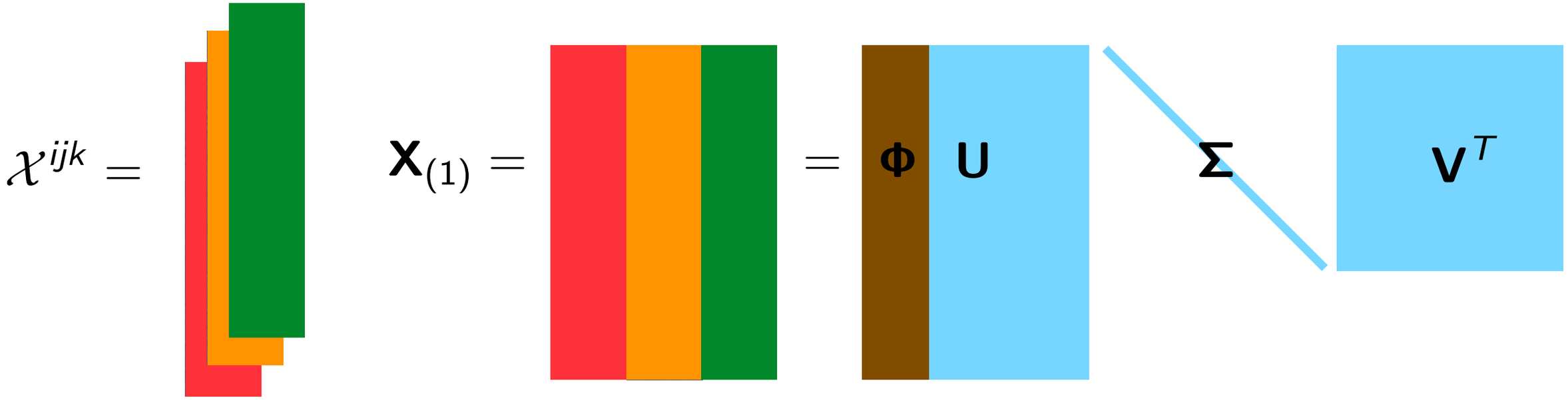


Tensor decomposition

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

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 values of mode-1 unfolding

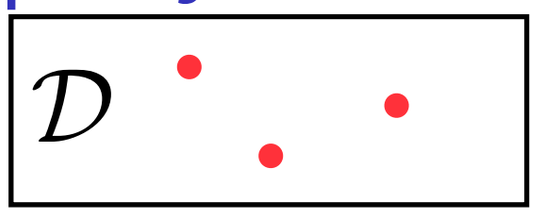


Φ *columns are principal components of the spatial simulation data*

How to integrate these data with the computational model?

Previous state of the art: Galerkin projection

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



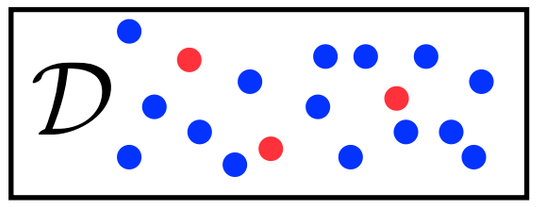
1. *Training:* Solve ODE for and collect simulation data
2. *Machine learning:* Identify structure in data
3. *Reduction:* Reduce the cost of solving ODE for $\boldsymbol{\mu} \in \mathcal{D}_{\text{query}} \setminus \mathcal{D}_{\text{training}}$

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

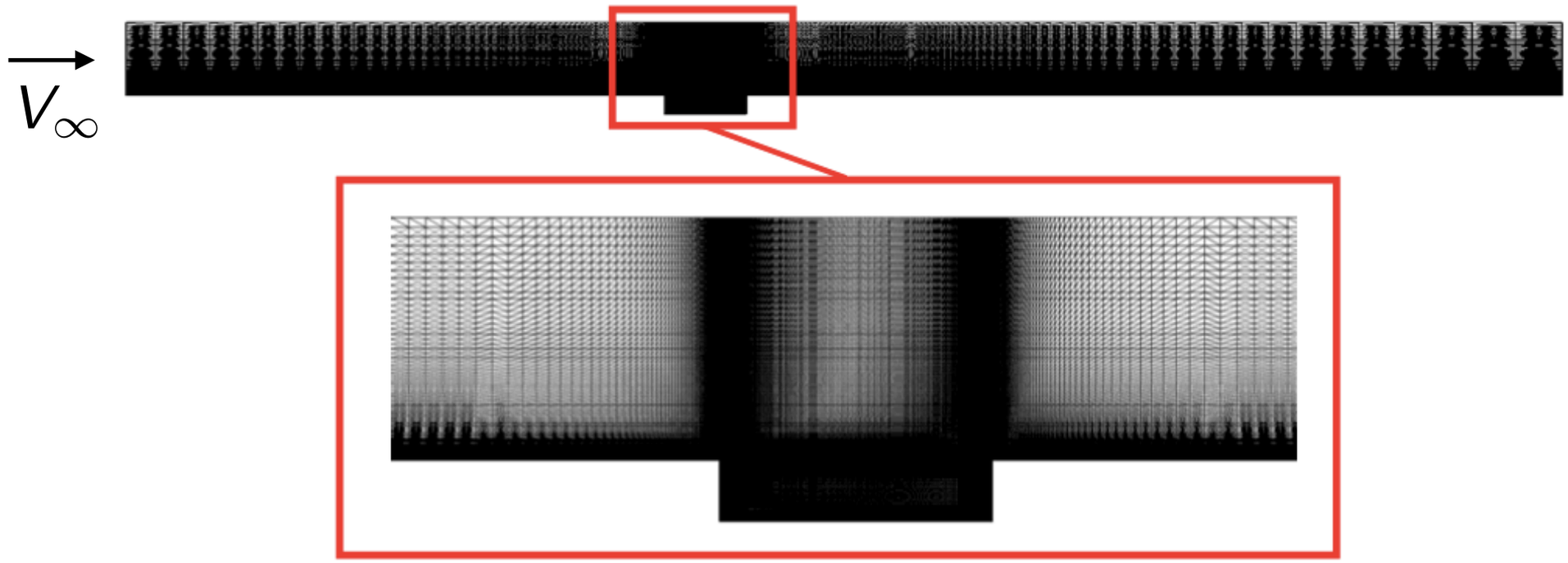
$$\mathbf{x}(t) \approx \tilde{\mathbf{x}}(t) = \boldsymbol{\Phi}\, \hat{\mathbf{x}}(t)$$

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

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
- $M_\infty = 0.6$

Spatial discretization

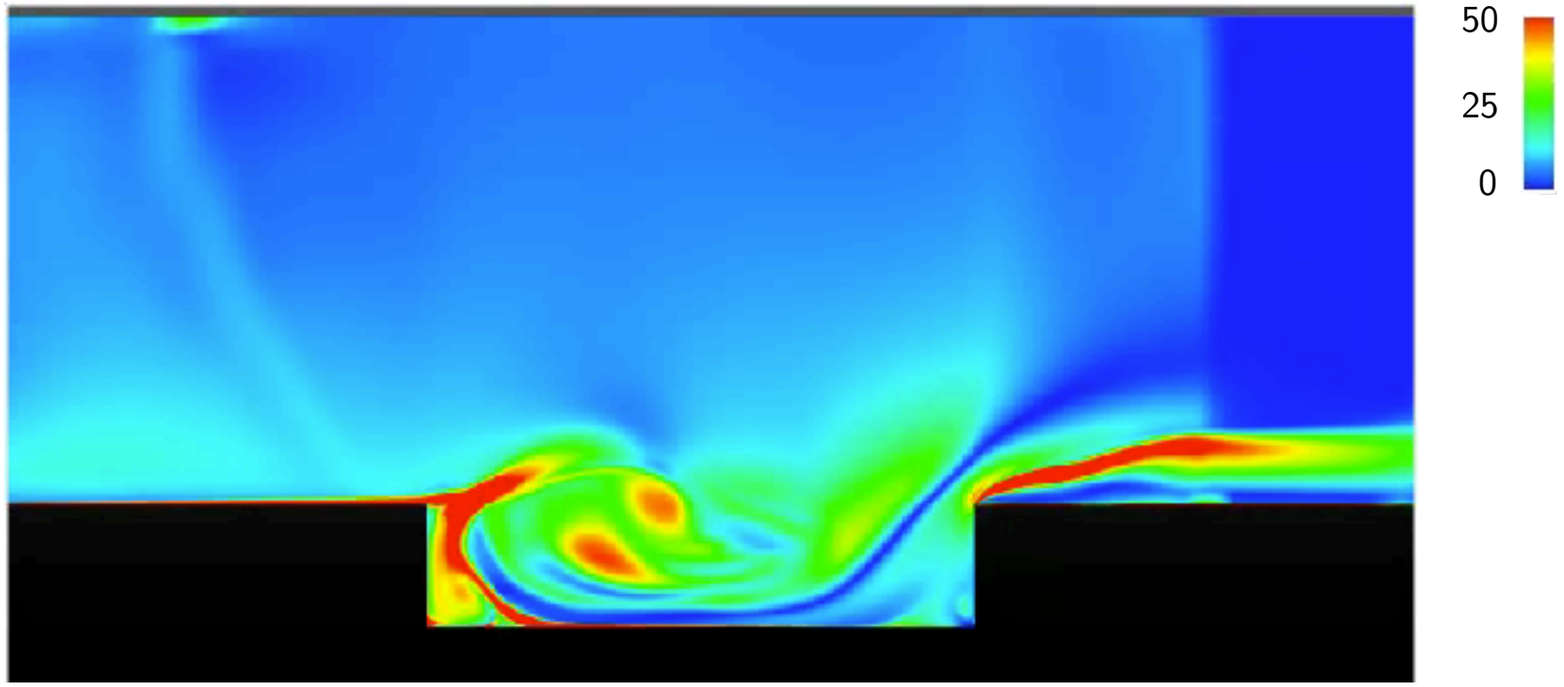
- 2nd-order finite volume
- DES turbulence model
- 1.2×10^6 degrees of freedom

Temporal discretization

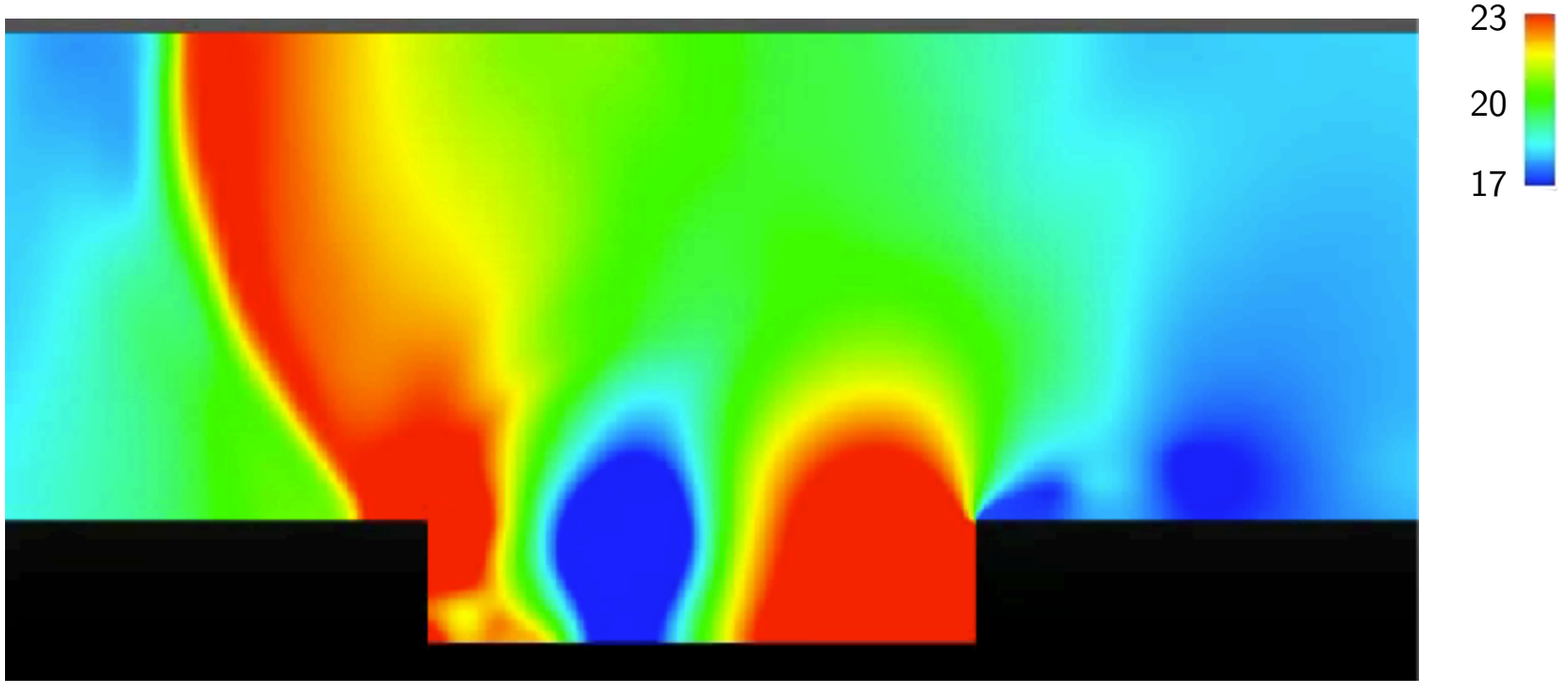
- 2nd-order BDF
- Verified time step $\Delta t = 1.5 \times 10^{-3}$
- 8.3×10^3 time instances

High-fidelity model solution

vorticity field

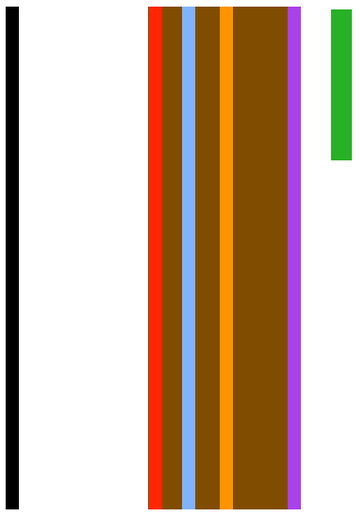


pressure field

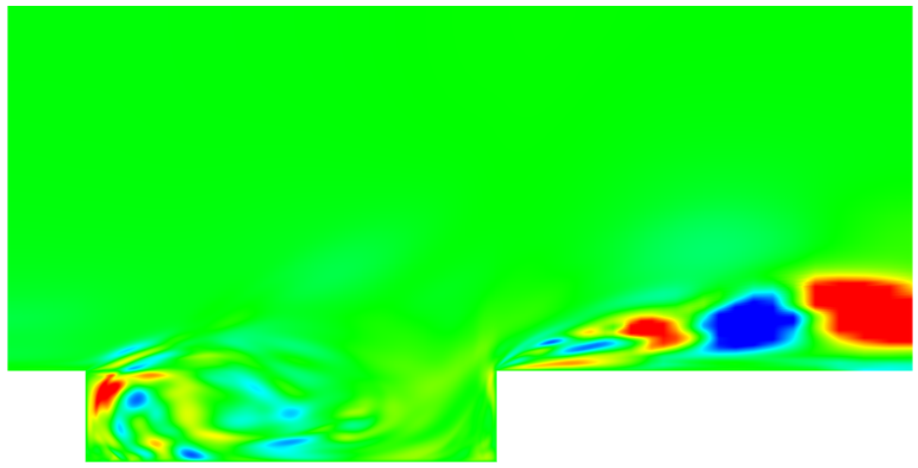


Principal components

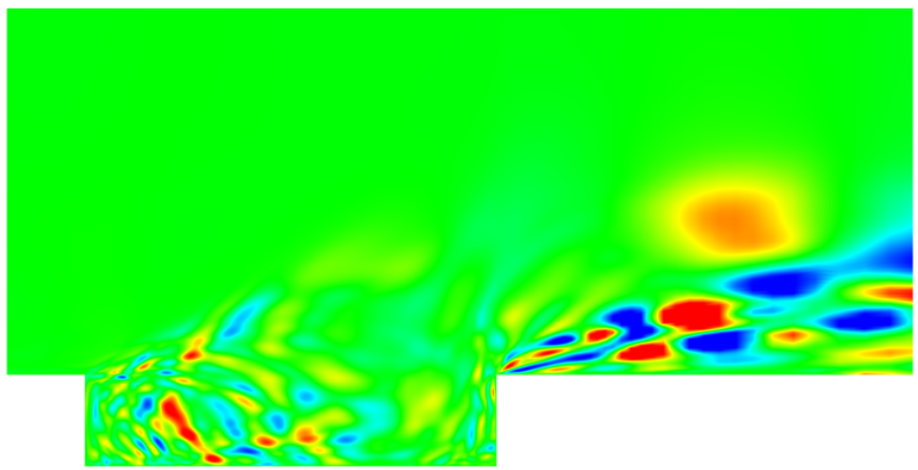
$$\mathbf{x}(t) \approx \mathbf{\Phi}\ \hat{\mathbf{x}}(t)$$



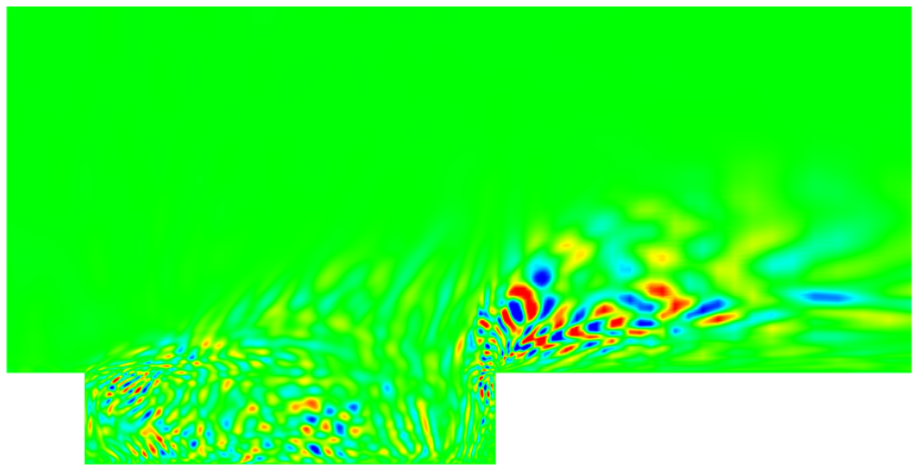
ϕ_1



ϕ_{21}

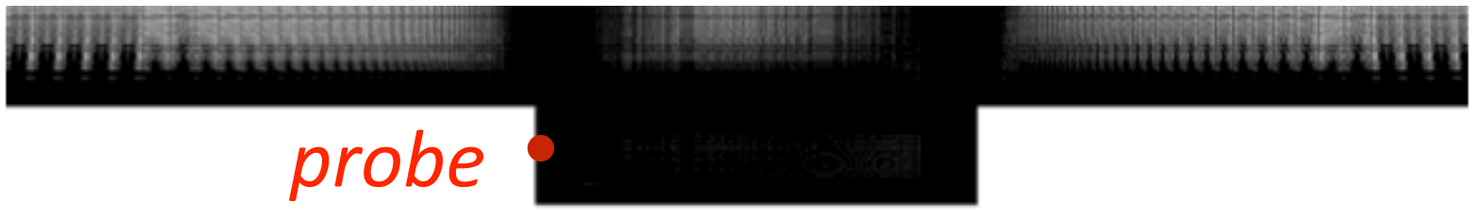


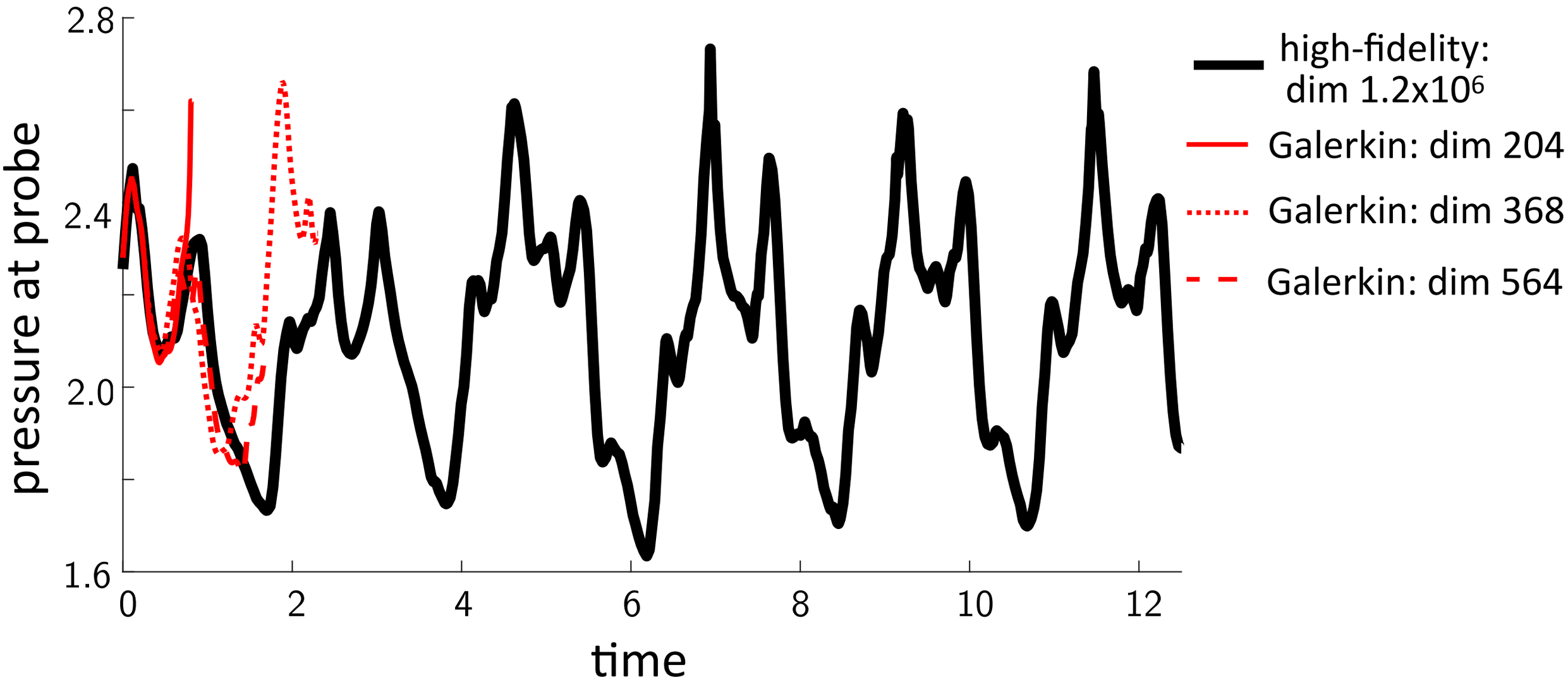
ϕ_{101}



ϕ_{401}

Galerkin performance





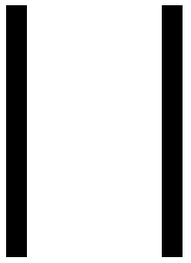
- Galerkin projection fails regardless of basis dimension

Can we construct a better projection?

Galerkin: time-continuous optimality

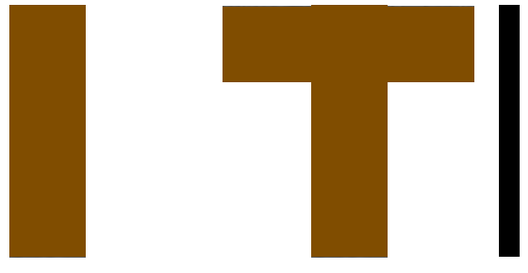
ODE

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



Galerkin ODE

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



\+ *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

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

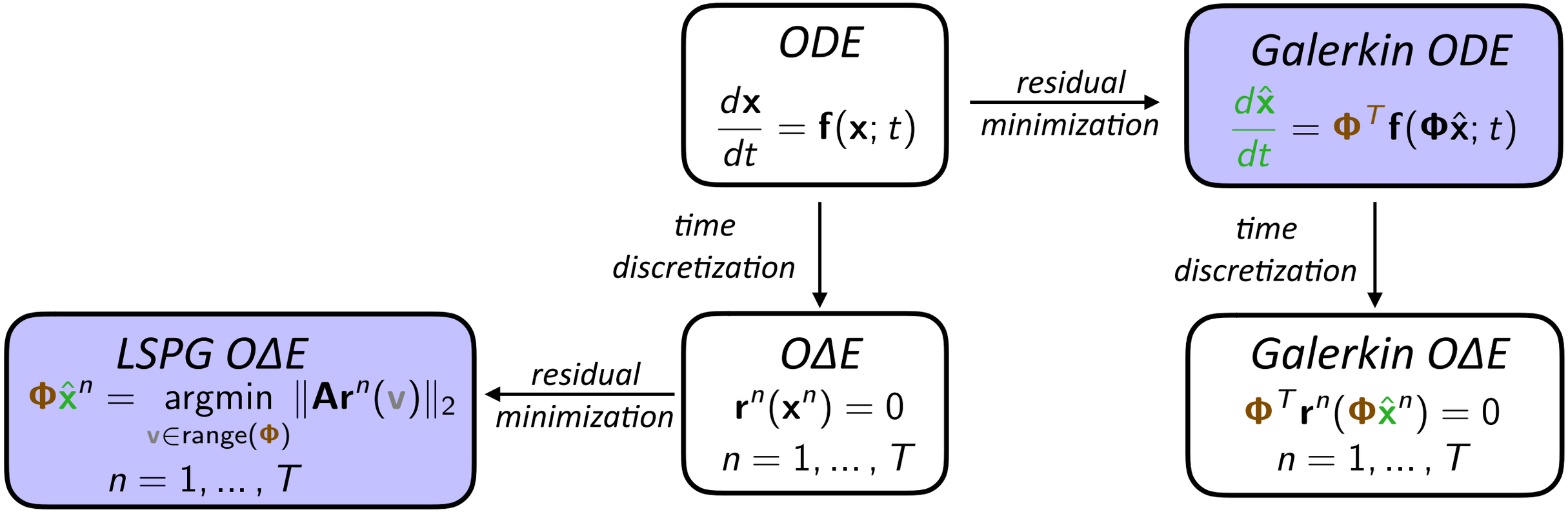
$$\mathbf{r}^n(\mathbf{x}) := \alpha_0\mathbf{x} - \Delta t\beta_0\mathbf{f}(\mathbf{x}; t^n) + \sum_{j=1}^{k} \alpha_j\mathbf{x}^{n-j} - \Delta t\sum_{j=1}^{k} \beta_j\mathbf{f}(\mathbf{x}^{n-j}; t^{n-j})$$

Galerkin OΔE

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

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

Residual minimization and time discretization



[Carlberg, Bou-Mosleh, Farhat, 2011]

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

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

*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

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

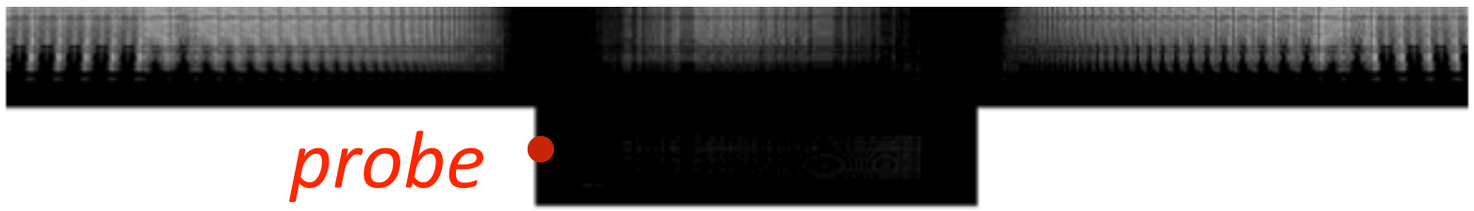
$$\|\mathbf{x}^n - \boldsymbol{\Phi}\hat{\mathbf{x}}^n_{\mathrm{LSPG}}\|_2 \leq \frac{1}{h}\min_{\hat{\mathbf{v}}}\|\mathbf{r}^n_{\mathrm{LSPG}}(\boldsymbol{\Phi}\hat{\mathbf{v}})\|_2 + \frac{1}{h}\sum_{\ell=1}^{k}|\alpha_\ell|\|\mathbf{x}^{n-\ell} - \boldsymbol{\Phi}\hat{\mathbf{x}}^{n-\ell}_{\mathrm{LSPG}}\|_2$$

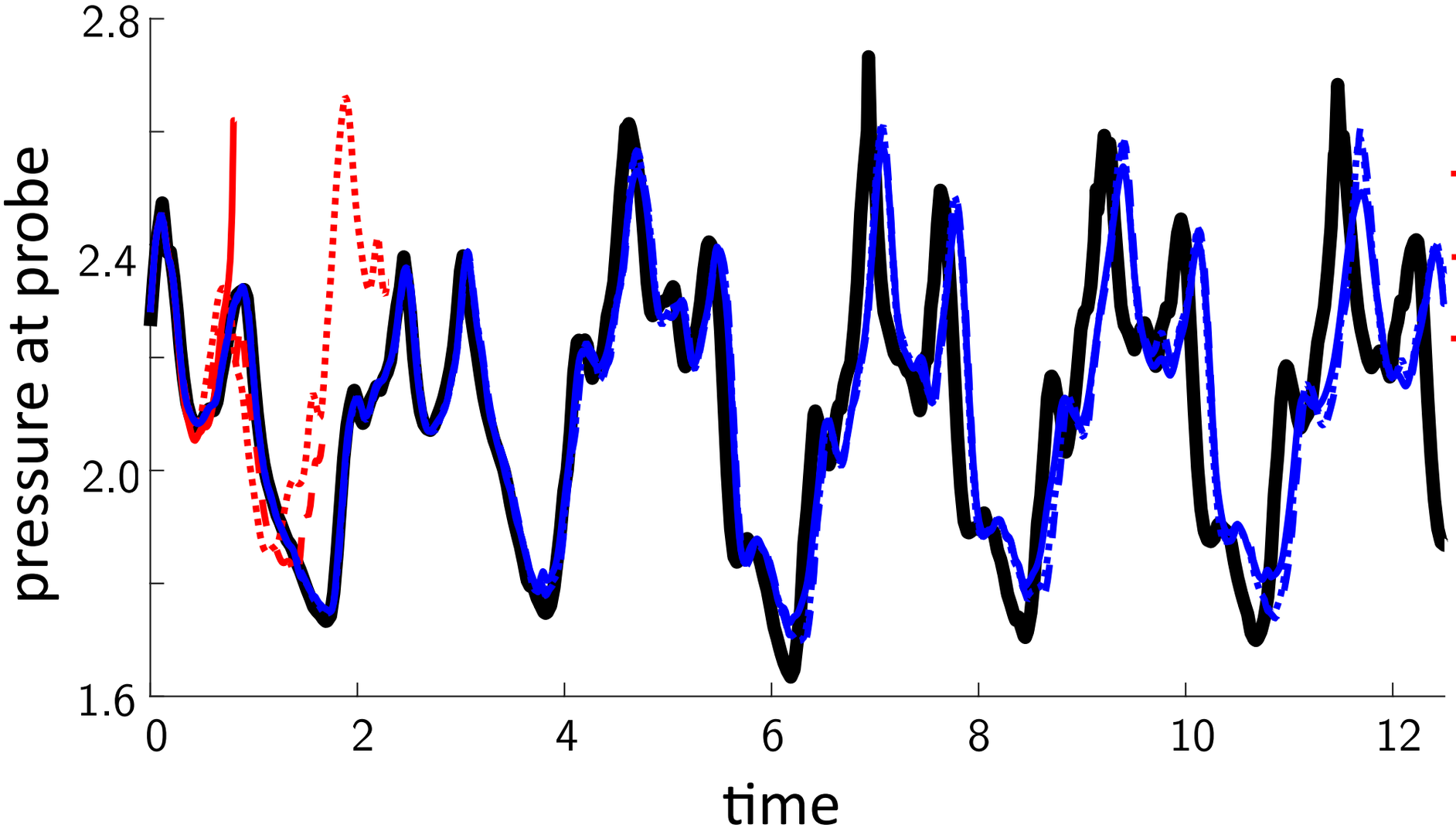
+ LSPG sequentially minimizes the error bound

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

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

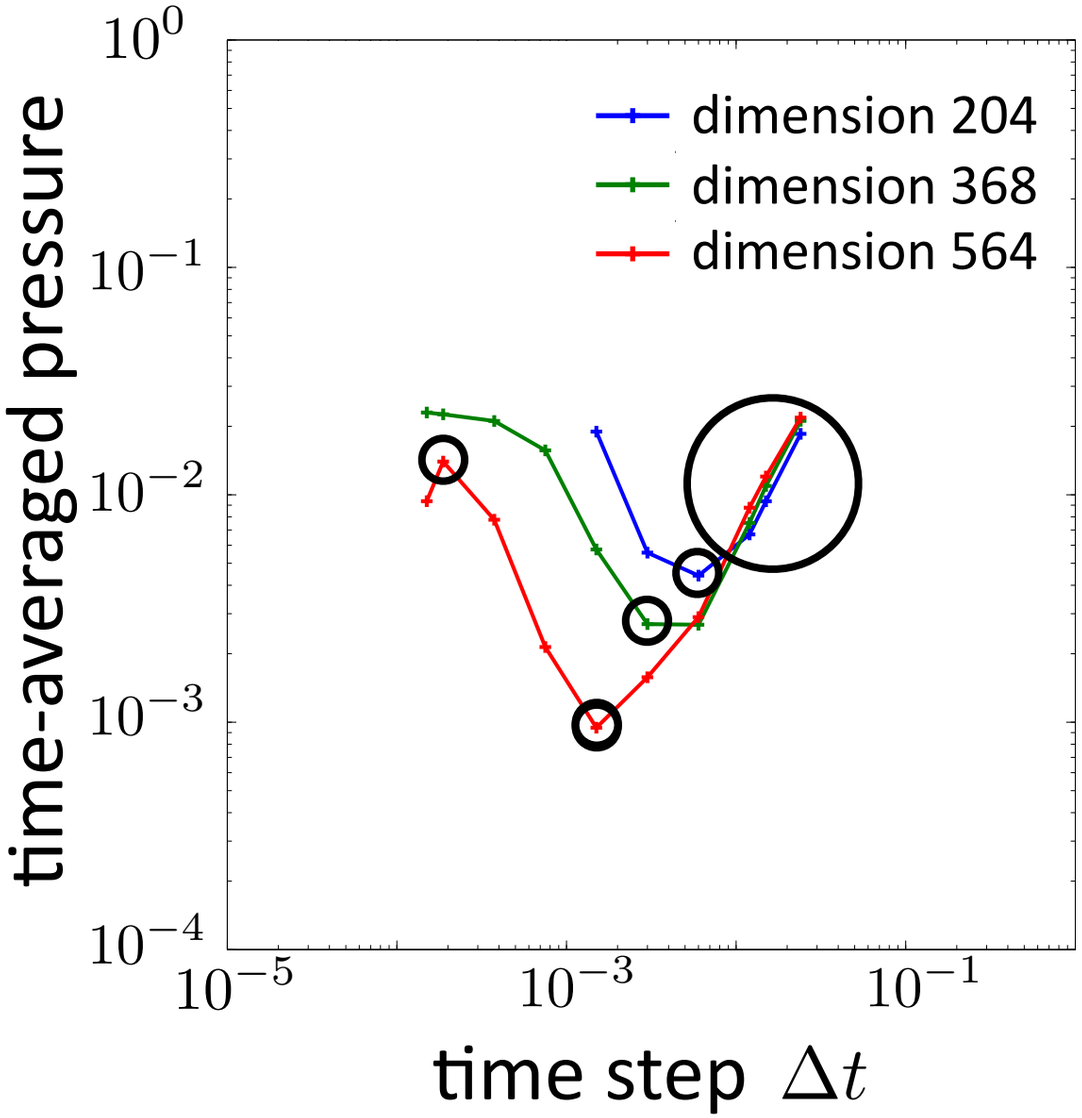
LSPG performance

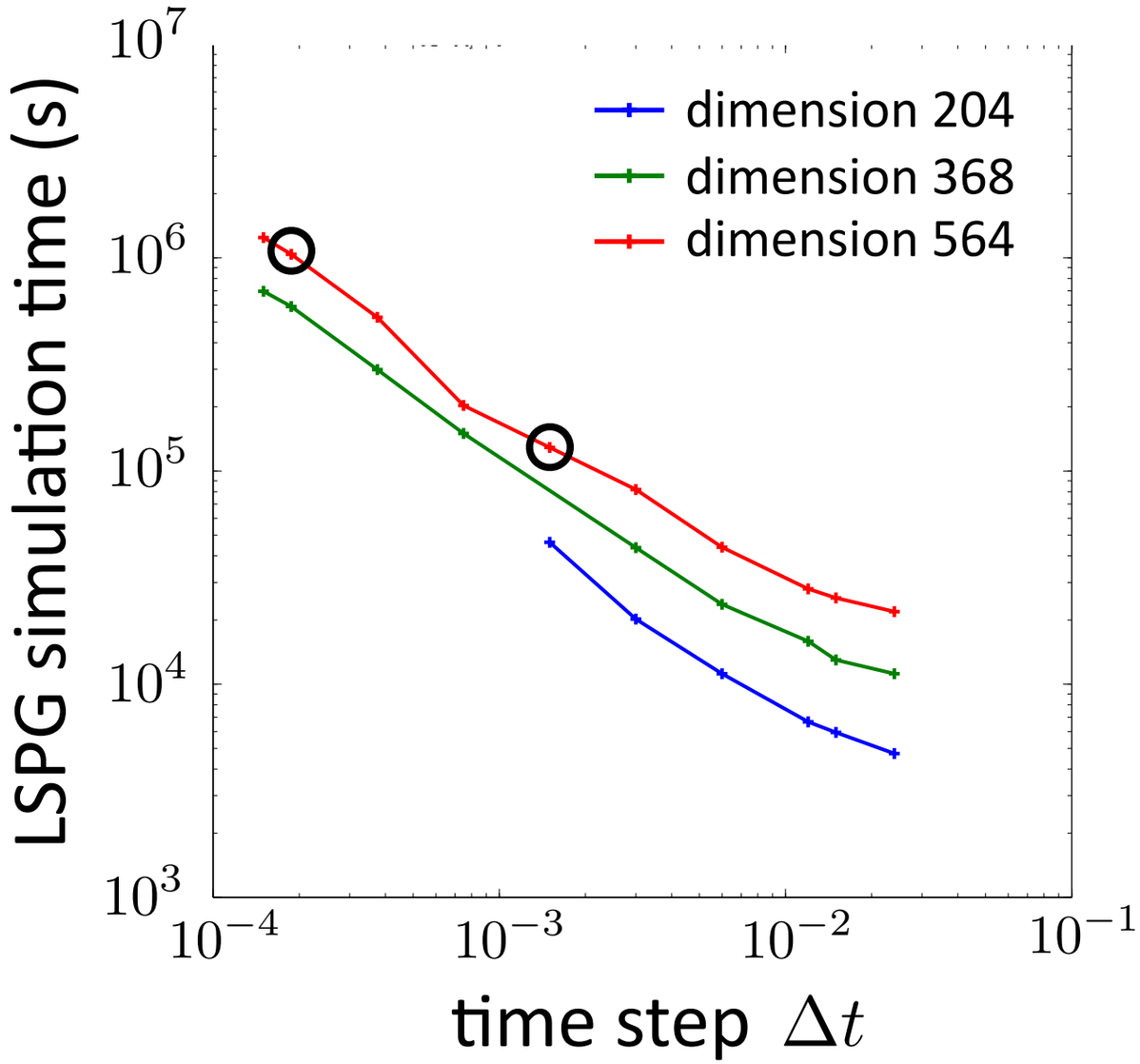




+ LSPG is far more accurate than Galerkin

LSPG dependence on time step





- **Surprising result:** intermediate Δt can reduce error and time by 10x
- Higher-dimension $\boldsymbol{\Phi}$:
  + can capture solution fluctuations over smaller Δt
  - provides no benefit for large Δt

My research

Enable extreme-scale simulations for many-query problems

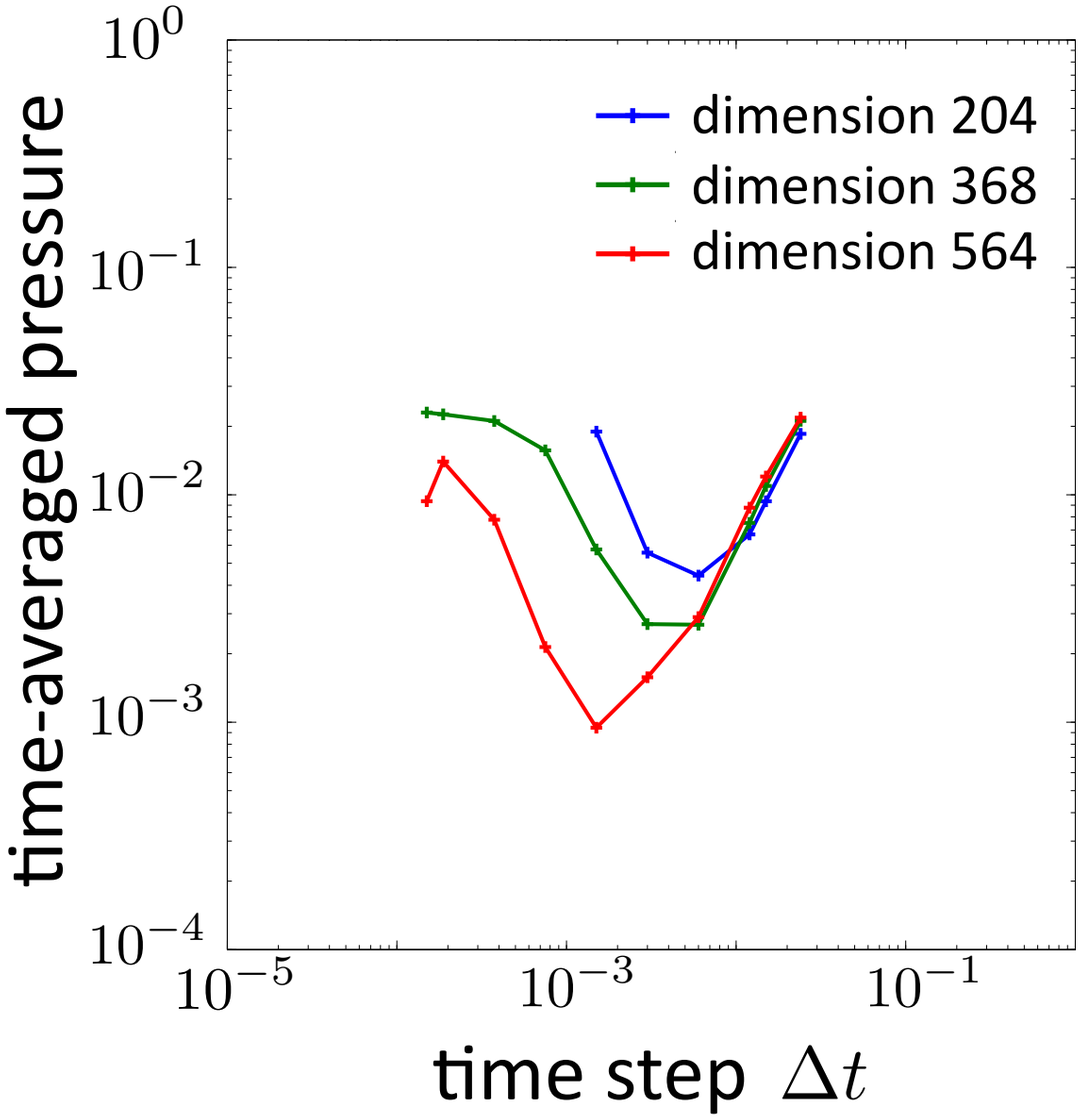
1. **Nonlinear model reduction**
   - \+ reduces model dimensionality and complexity
   - *accuracy*: LSPG projection [Carlberg, Bou-Mosleh, Farhat, 2011; Carlberg, Antil, Barone, 2017]
   - *low cost*: sample mesh [Carlberg, Farhat, Cortial, Amsallem, 2013]*
   - *structure preservation* [Carlberg, Tuminaro, Boggs, 2015; Peng and Carlberg, 2017; Carlberg and Choi, 2017]

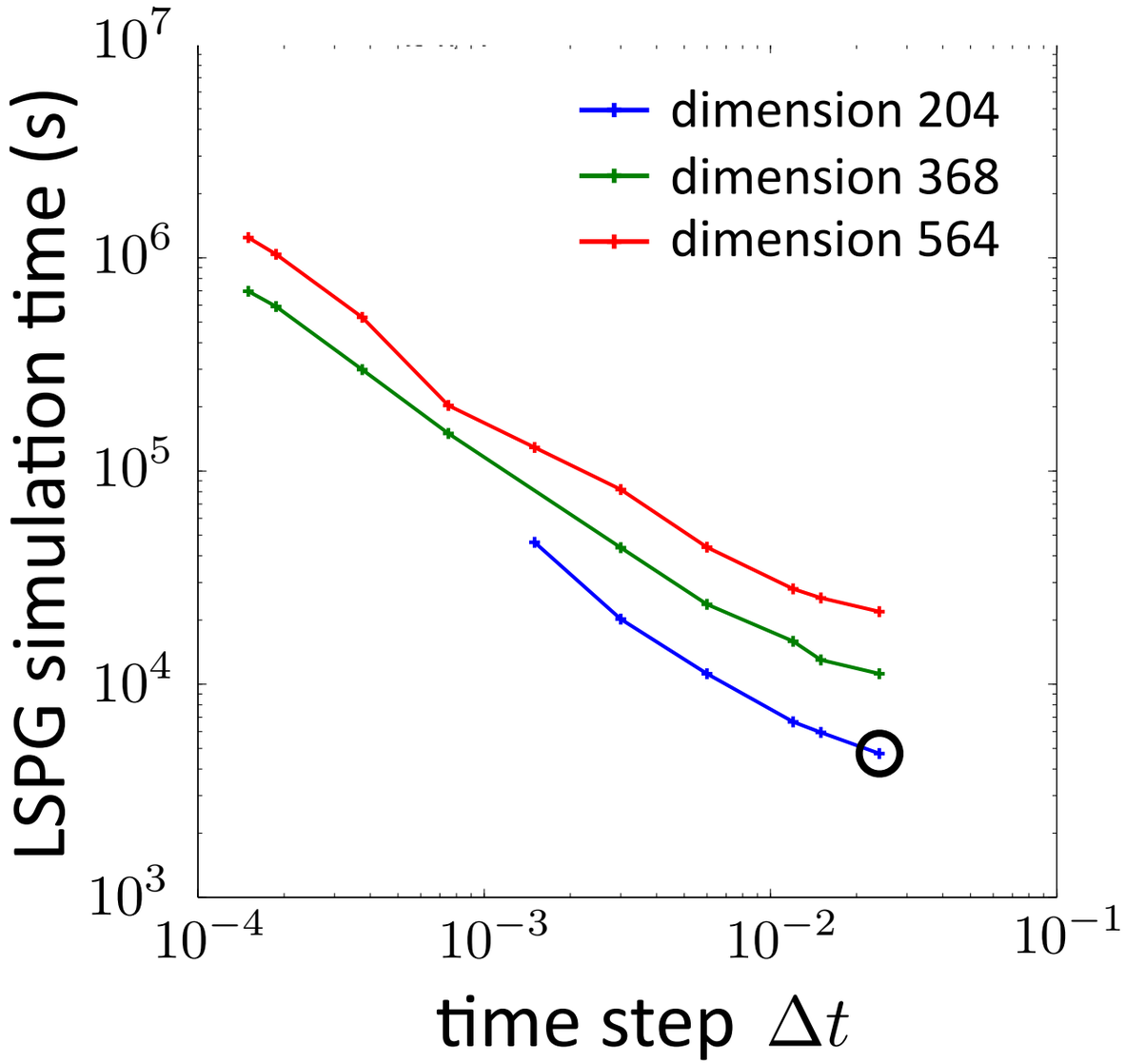
2. **Data-driven error modeling**
   - \+ rigorously accounts for model-reduction error
   - regression error modeling
     [Drohmann and Carlberg, 2015; Trehan, Carlberg, Durlofsky, 2017; Freno and Carlberg, 2017]

3. **Data-driven numerical solvers**
   - \+ improves performance of numerical solvers
   - adaptive subspaces [Carlberg, 2015; Carlberg, Forstall, Tuminaro, 2016]
   - reduce temporal complexity
     [Carlberg, Ray, van Bloemen Waanders, 2015; Carlberg, Brencher, Haasdonk, Barth, 2017; Choi 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

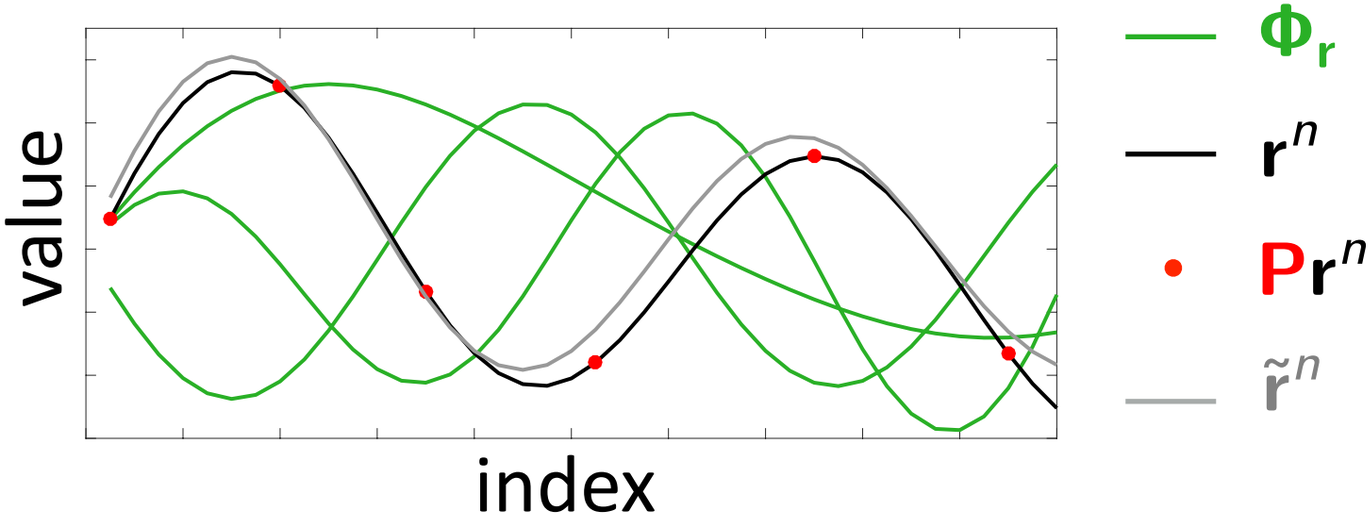
Why does this occur? Can we fix it?

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

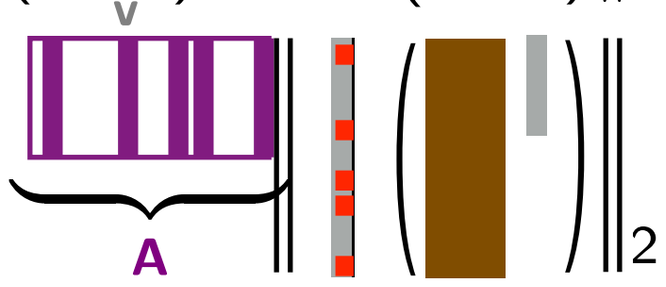
$$\underset{\hat{\mathbf{v}}}{\text{minimize}} \left\| \mathbf{A}\, \mathbf{r}^n(\boldsymbol{\Phi}\,\hat{\mathbf{v}}) \right\|_2$$

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

- **ML:** compute residual principal components $\boldsymbol{\Phi}_r$ and sampling $\mathbf{P}$
- **Reduction**: compute regression approximation $\mathbf{r}^n \approx \tilde{\mathbf{r}}^n = \boldsymbol{\Phi}_r(\mathbf{P}\boldsymbol{\Phi}_r)^{+}\mathbf{P}\mathbf{r}^n$



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

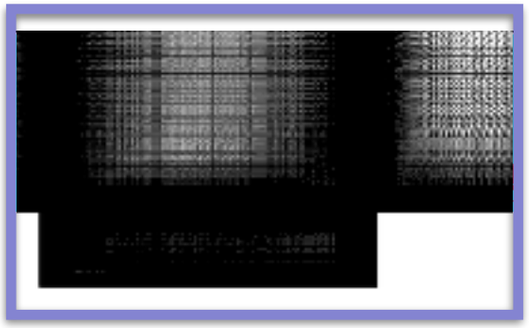


+ *Only a few elements of* $\mathbf{r}^n$ *must be computed*

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

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

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

My research

Enable extreme-scale simulations for many-query problems

1. **Nonlinear model reduction**
   - \+ reduces model dimensionality and complexity
   - *accuracy*: LSPG projection [Carlberg, Bou-Mosleh, Farhat, 2011; Carlberg, Antil, Barone, 2017]
   - *low cost*: sample mesh [Carlberg, Farhat, Cortial, Amsallem, 2013]*
   - *structure preservation* [Carlberg, Tuminaro, Boggs, 2015*; Peng and Carlberg, 2017; Carlberg and Choi, 2017]

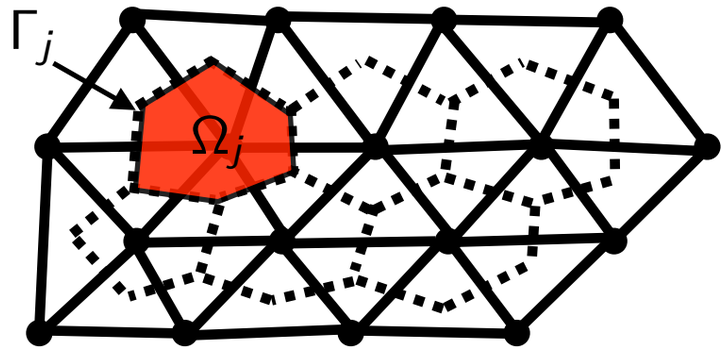
2. **Data-driven error modeling**
   - \+ rigorously accounts for model-reduction error
   - regression error modeling
     [Drohmann and Carlberg, 2015; Trehan, Carlberg, Durlofsky, 2017; Freno and Carlberg, 2017]

3. **Data-driven numerical solvers**
   - \+ improves performance of numerical solvers
   - adaptive subspaces [Carlberg, 2015; Carlberg, Forstall, Tuminaro, 2016]
   - reduce temporal complexity
     [Carlberg, Ray, van Bloemen Waanders, 2015; Carlberg, Brencher, Haasdonk, Barth, 2017; Choi and Carlberg, 2017]

* Featured Article, SIAM J Sci Comp, 2015

LSPG for finite-volume models

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



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

- average value of conserved variable *i* over control volume *j*

$$f_{\mathcal{I}(i,j)} = -\frac{1}{|\Omega_j|}\int_{\Omega_j} u_i(\vec{x}, t)v_\ell(\vec{x}, t)n_\ell(\vec{x})d\vec{x}$$

- flux of conserved variable *i* entering control volume *j*

OΔE: $\mathbf{r}^n(\mathbf{x}^n) = 0,\ \ n = 1, \ldots, N$

$$r^n_{\mathcal{I}(i,j)} = \frac{1}{|\Omega_j|}\int_{\Omega_j} u_i(\vec{x}, t^{n+1}) - u_i(\vec{x}, t^n)d\vec{x} + \frac{1}{|\Omega_j|}\int_{t^n}^{t^{n+1}}\int_{\Gamma_j} u_i(\vec{x}, t)v_\ell(\vec{x}, t)n_\ell(\vec{x})d\vec{x}dt$$

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

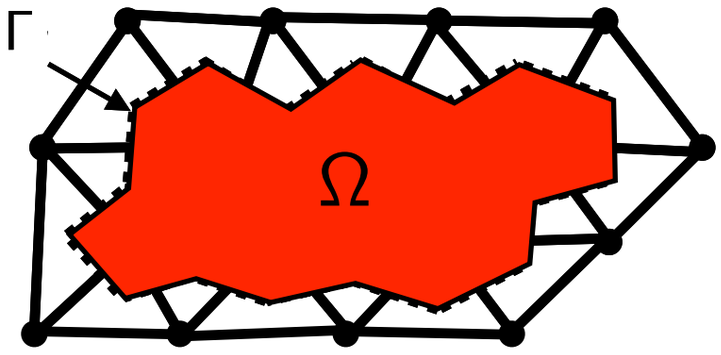
LSPG OΔE: $\underset{\hat{\mathbf{v}}}{\text{minimize}}\ \left\|\mathbf{A}\mathbf{r}^n(\boldsymbol{\Phi}\hat{\mathbf{v}})\right\|_2$

- minimize weighted sum of squared conservation-law violations over time step *n*
  - \+ Lower error bounds, better observed performance than Galerkin
  - \- Does not ensure conservation anywhere

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

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

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

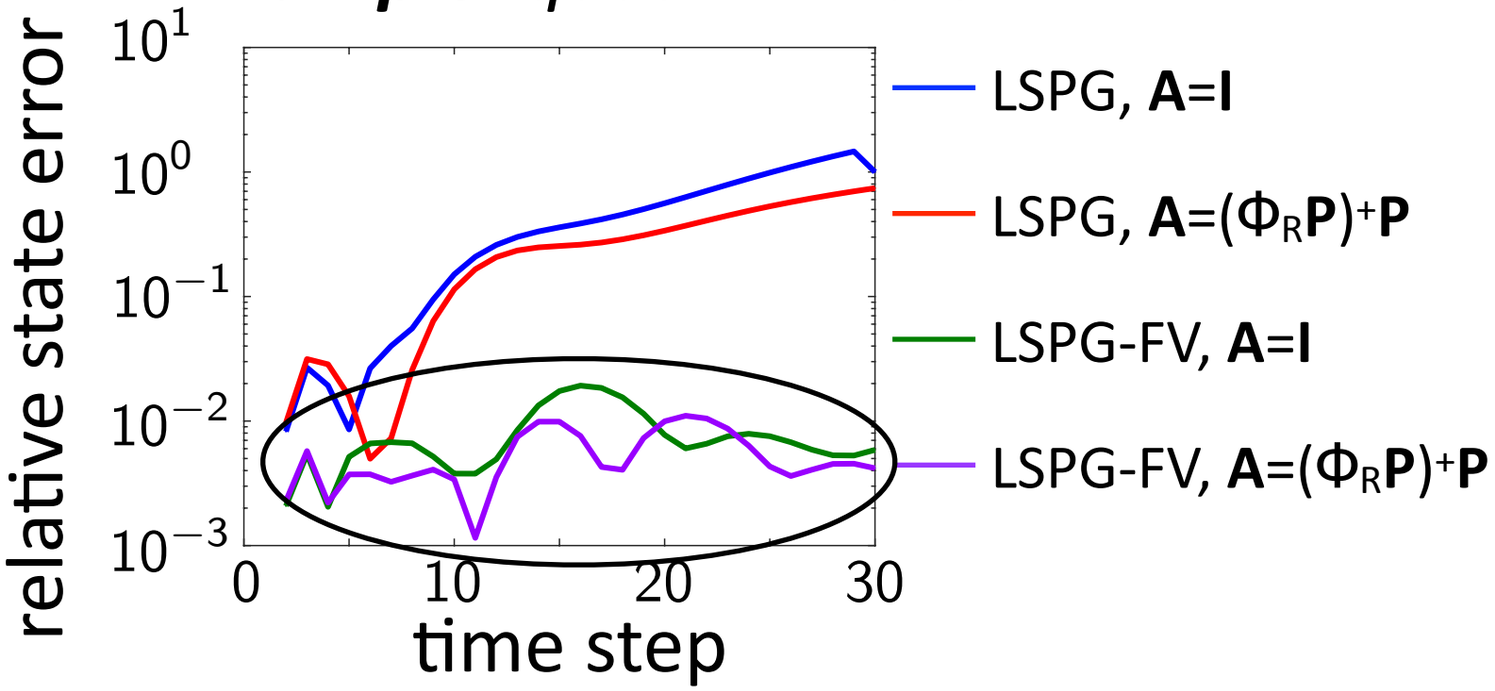


- minimize weighted sum of squared conservation-law violations over time step *n* subject to global conservation

$$\bar{r}_i^n = \frac{1}{|\Omega|}\int_\Omega u_i(\vec{x}, t^{n+1}) - u_i(\vec{x}, t^n)d\vec{x} + \frac{1}{|\Omega|}\int_{t^n}^{t^{n+1}}\int_\Gamma u_i(\vec{x}, t)v_\ell(\vec{x}, t)n_\ell(\vec{x})d\vec{x}dt$$

- violation of conservation of variable *i* in entire domain over time step *n*

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



speedup

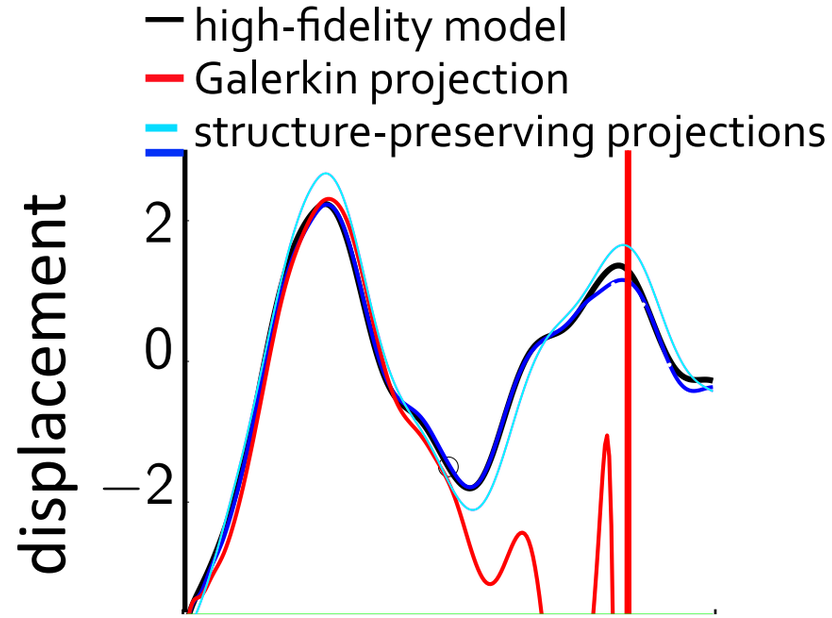
	LSPG	LSPG-FV
A=**I**	0.57	0.44
A=$(\Phi_R\mathbf{P})^+\mathbf{P}$	4.4	5.3

+ structure preservation improves accuracy
+ sample mesh improves wall time

Structure preservation

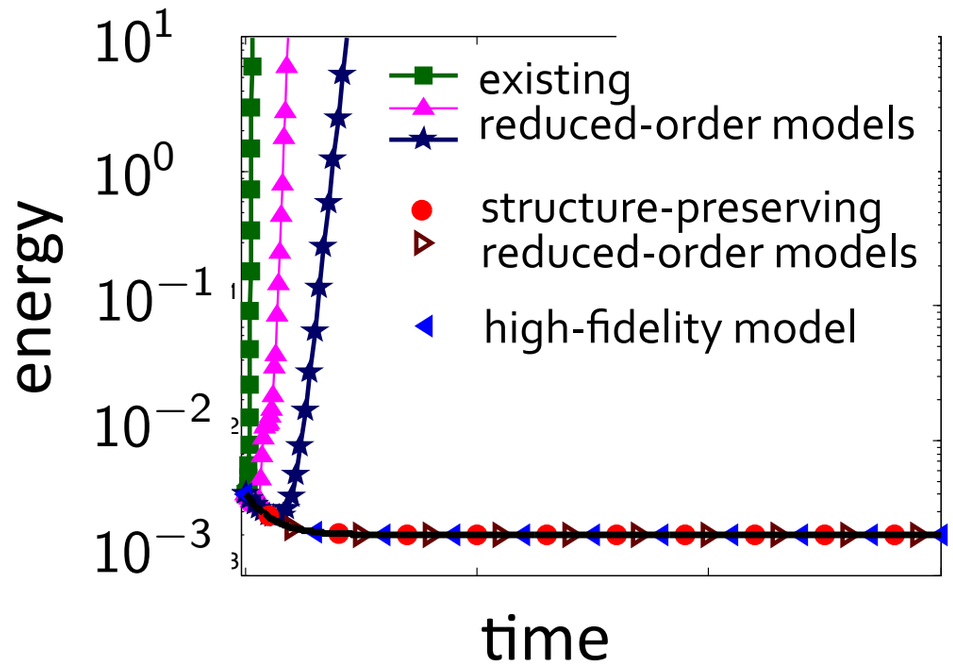
Nonlinear Lagrangian dynamical systems
[Carlberg, Tuminaro, Boggs, 2015]

- 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

Enable extreme-scale simulations for many-query problems

1. **Nonlinear model reduction**
   - \+ reduces model dimensionality and complexity
   - *accuracy*: LSPG projection [Carlberg, Bou-Mosleh, Farhat, 2011; Carlberg, Antil, Barone, 2017]
   - *low cost*: sample mesh [Carlberg, Farhat, Cortial, Amsallem, 2013]
   - *structure preservation* [Carlberg, Tuminaro, Boggs, 2015; Peng and Carlberg, 2017; Carlberg and Choi, 2017]

2. **Data-driven error modeling**
   - \+ rigorously accounts for model-reduction error
   - regression error modeling
     [Drohmann and Carlberg, 2015; Trehan, Carlberg, Durlofsky, 2017; Freno and Carlberg, 2017]

3. **Data-driven numerical solvers**
   - \+ improves performance of numerical solvers
   - adaptive subspaces [Carlberg, 2015; Carlberg, Forstall, Tuminaro, 2016]
   - reduce temporal complexity
     [Carlberg, Ray, van Bloemen Waanders, 2015; Carlberg, Brencher, Haasdonk, Barth, 2017; Choi and Carlberg, 2017]

My research

Enable extreme-scale simulations for many-query problems

1. **Nonlinear model reduction**
   - \+ reduces model dimensionality and complexity
   - *accuracy*: LSPG projection [Carlberg, Bou-Mosleh, Farhat, 2011; Carlberg, Antil, Barone, 2017]
   - *low cost*: sample mesh [Carlberg, Farhat, Cortial, Amsallem, 2013]
   - *structure preservation* [Carlberg, Tuminaro, Boggs, 2015; Peng and Carlberg, 2017; Carlberg and Choi, 2017]

2. **Data-driven error modeling**
   - \+ rigorously accounts for model-reduction error
   - regression error modeling
     [Drohmann and Carlberg, 2015; Trehan, Carlberg, Durlofsky, 2017; Freno and Carlberg, 2017]

3. **Data-driven numerical solvers**
   - \+ improves performance of numerical solvers
   - adaptive subspaces [Carlberg, 2015; Carlberg, Forstall, Tuminaro, 2016]
   - reduce temporal complexity
     [Carlberg, Ray, van Bloemen Waanders, 2015; Carlberg, Brencher, Haasdonk, Barth, 2017; Choi and Carlberg, 2017]

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

$$\|\mathbf{x}^n - \boldsymbol{\Phi}\hat{\mathbf{x}}^n_{\mathrm{G}}\|_2 \leq \frac{1}{h}\frac{\gamma_1(\gamma_2)^n\exp(\gamma_3 t^n)}{\gamma_4+\gamma_5\Delta t}\|\mathbf{r}_{\mathrm{G}}(\boldsymbol{\Phi}\hat{\mathbf{x}}^0_{\mathrm{G}})\|_2 + \frac{1}{h}\sum_{\ell=1}^{k}|\alpha_\ell|\max_{j\in\{1,\ldots,N\}}\|\mathbf{x}^{n-\ell} - \boldsymbol{\Phi}\hat{\mathbf{x}}^j_{\mathrm{G}}\|_2$$

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

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

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*

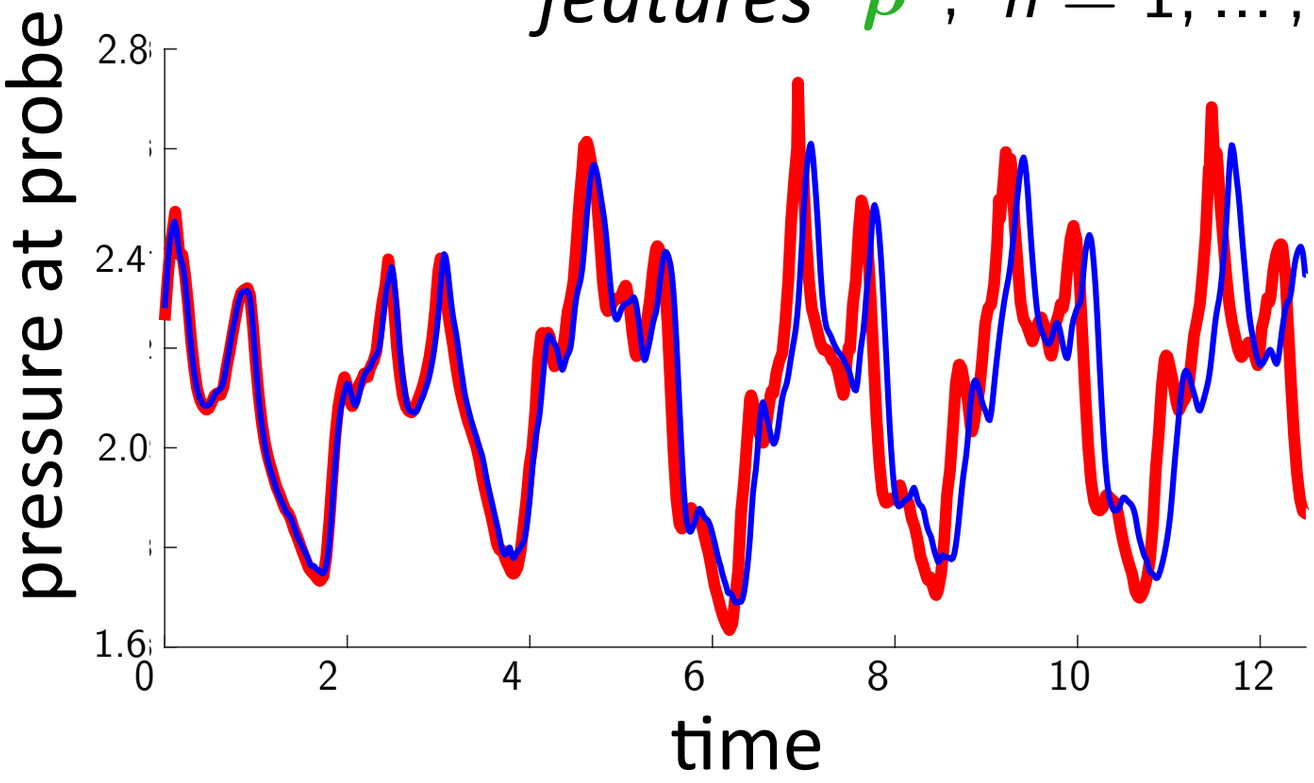
Key insight

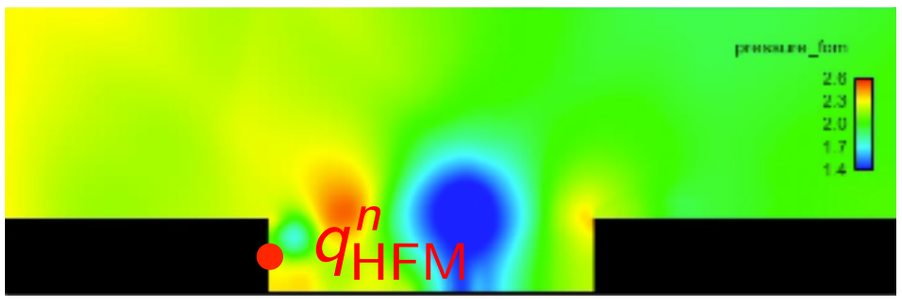
inputs μ → *high-fidelity model* → *outputs* $q^n_{\text{HFM}},\ n = 1, \ldots, T$

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

↓

features $\rho^n,\ n = 1, \ldots, T$



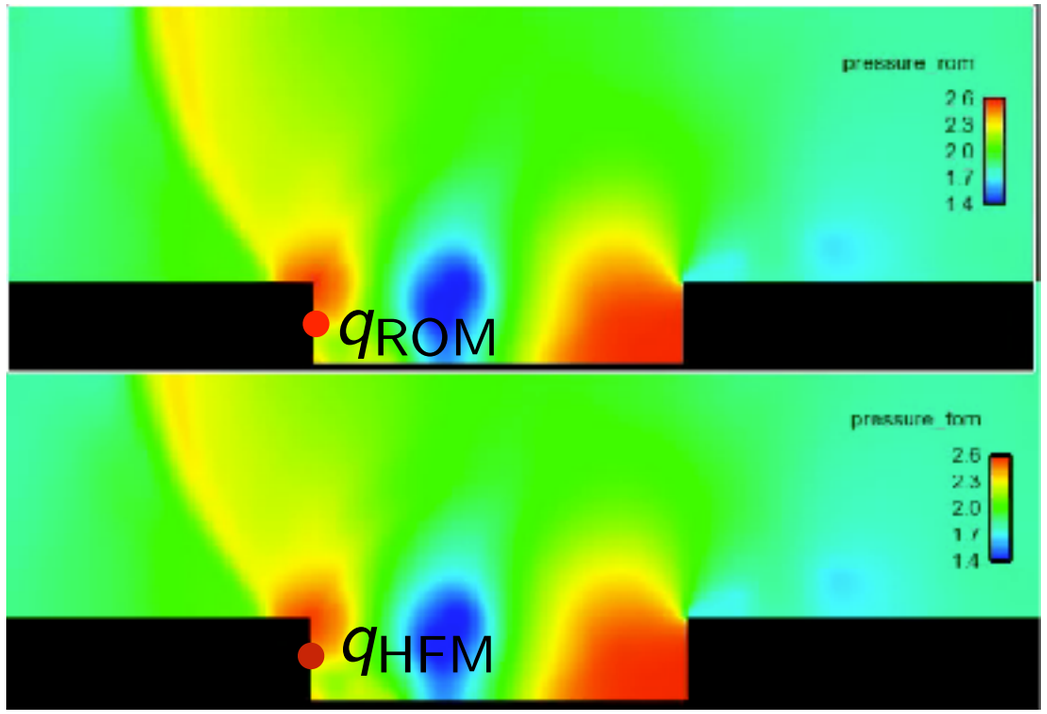


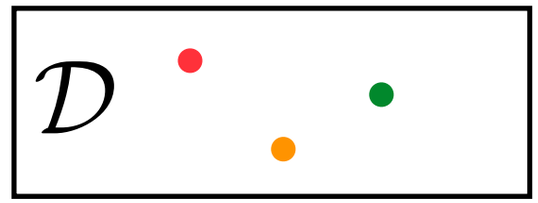
Reduced-order models generate features ρ^n ***that may inform its error***

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
2. *Machine learning:* Construct regression model
3. *Reduction:* predict reduced-order-model error for



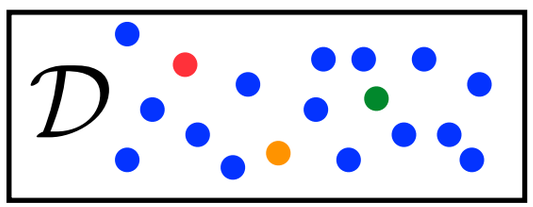


features ρ^n → *regression model* → *error prediction*

- *Regression methods*: Gaussian process, LASSO, random forest, SVM

Regression model for the error

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



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

↓

features $\boldsymbol{\rho}^n$, $n = 1, \ldots, N$

↓

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

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

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

Which features $\boldsymbol{\rho}^n$ ***should be included in regression?***

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

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

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

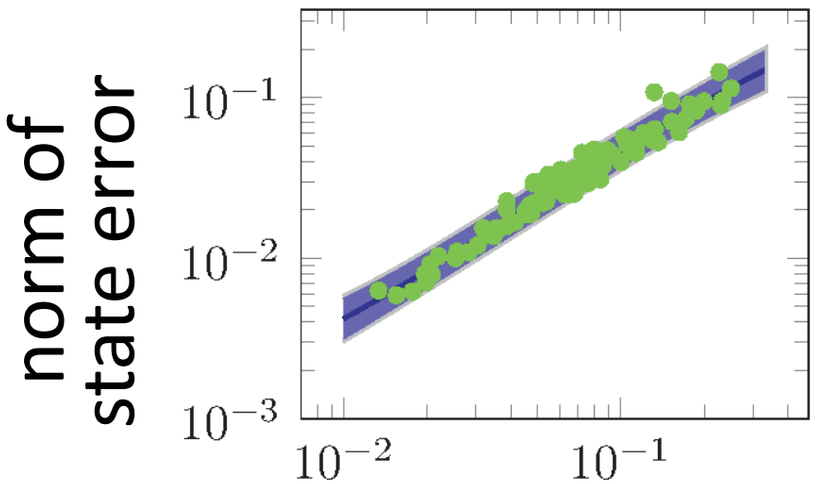
+ *Inexpensive to solve*

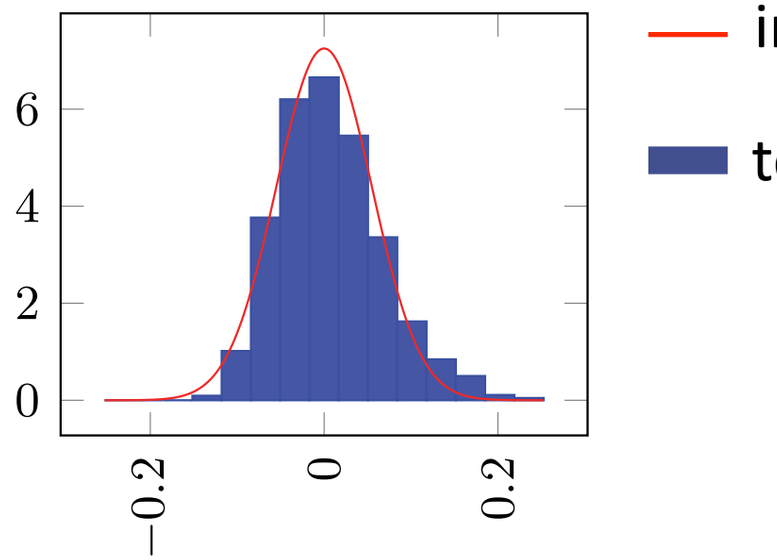
Costly to solve

- 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





inferred Gaussian

test data

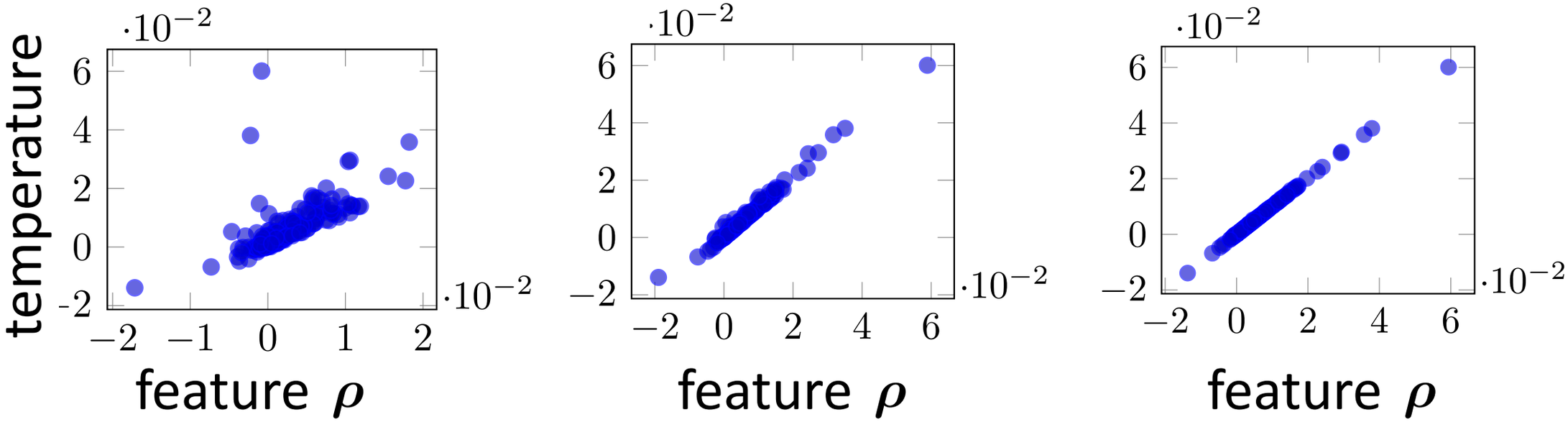
\+ low-variance model of the error

\+ 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 $T(\mathbf{x}) - T(\boldsymbol{\Phi}\hat{\mathbf{x}})$
- *1 feature ρ:* dual-weighted residual $(\tilde{\mathbf{y}}^n)^T \mathbf{r}^n(\boldsymbol{\Phi}\hat{\mathbf{x}}^n)$
- *regression:* Gaussian process



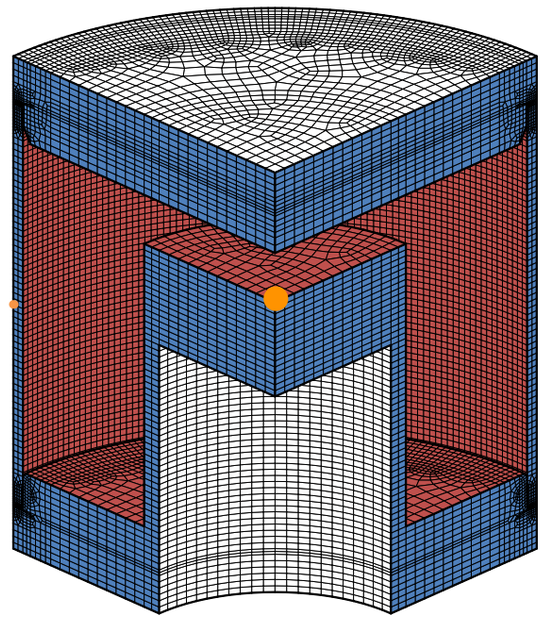
+ *uncertainty control:* variance reduced as feature improved

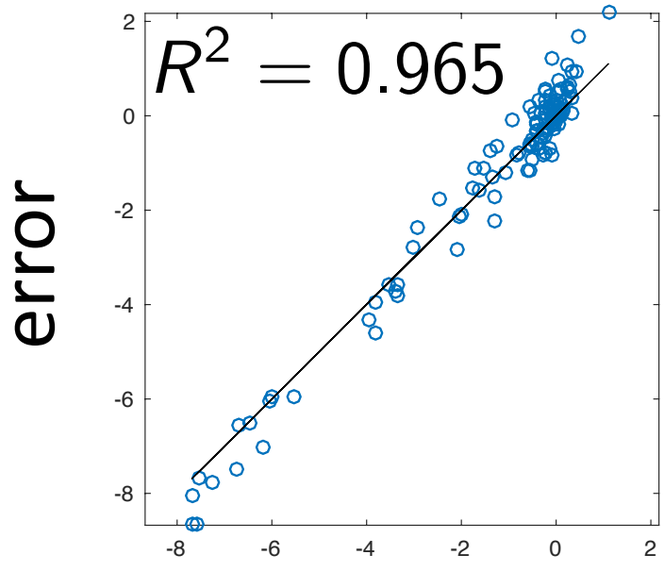
- *Other application*: rigorous integration with Bayesian inference
  [Carlberg, Uy, Lu, Morzfeld, 2017]

Application 2: nonlinear static mechanical response

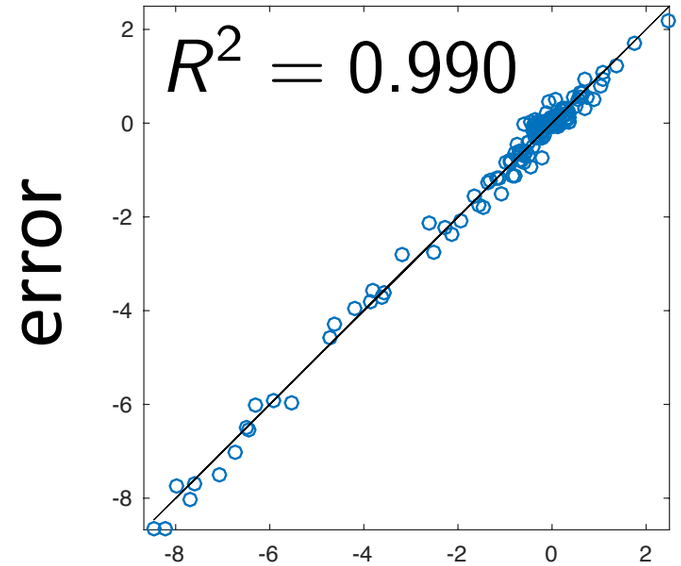
[Freno, Carlberg, 2017]

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

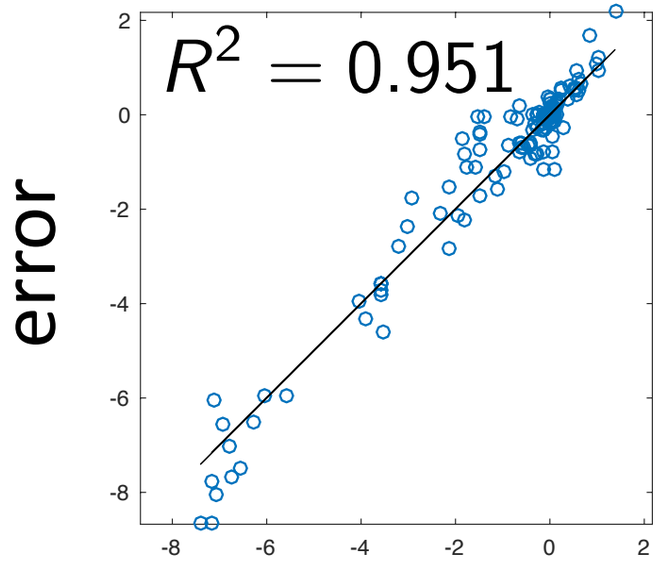




random forest
error prediction



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]

My research

Enable extreme-scale simulations for many-query problems

1. **Nonlinear model reduction**
   - \+ reduces model dimensionality and complexity
   - *accuracy*: LSPG projection [Carlberg, Bou-Mosleh, Farhat, 2011; Carlberg, Antil, Barone, 2017]
   - *low cost*: sample mesh [Carlberg, Farhat, Cortial, Amsallem, 2013]
   - *structure preservation* [Carlberg, Tuminaro, Boggs, 2015; Peng and Carlberg, 2017; Carlberg and Choi, 2017]

2. **Data-driven error modeling**
   - \+ rigorously accounts for model-reduction error
   - regression error modeling

     [Drohmann and Carlberg, 2015; Trehan, Carlberg, Durlofsky, 2017; Freno and Carlberg, 2017]

3. **Data-driven numerical solvers**
   - \+ improves performance of numerical solvers
   - adaptive subspaces [Carlberg, 2015; Carlberg, Forstall, Tuminaro, 2016]
   - reduce temporal complexity

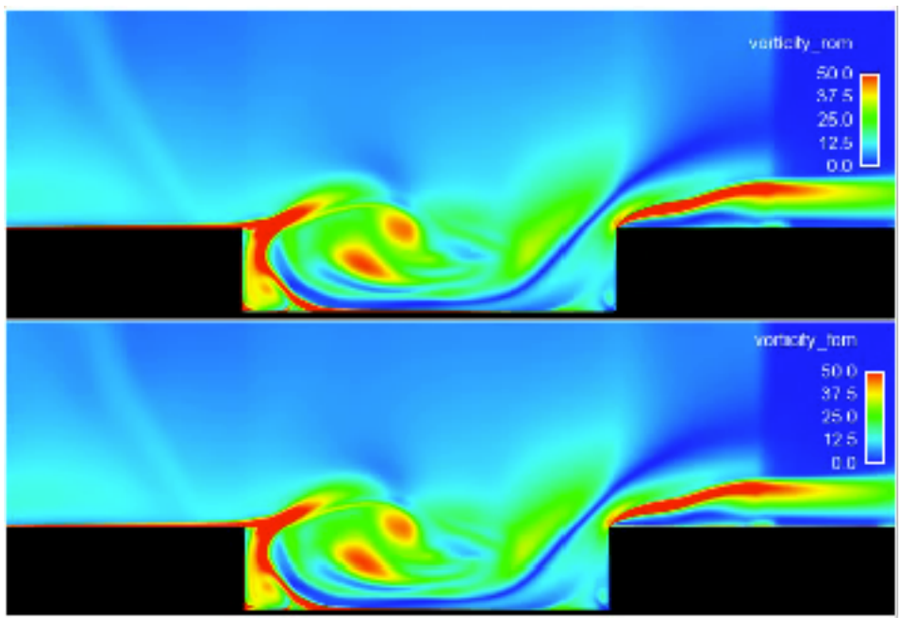
     [Carlberg, Ray, van Bloemen Waanders, 2015; Carlberg, Brencher, Haasdonk, Barth, 2017; Choi and Carlberg, 2017]

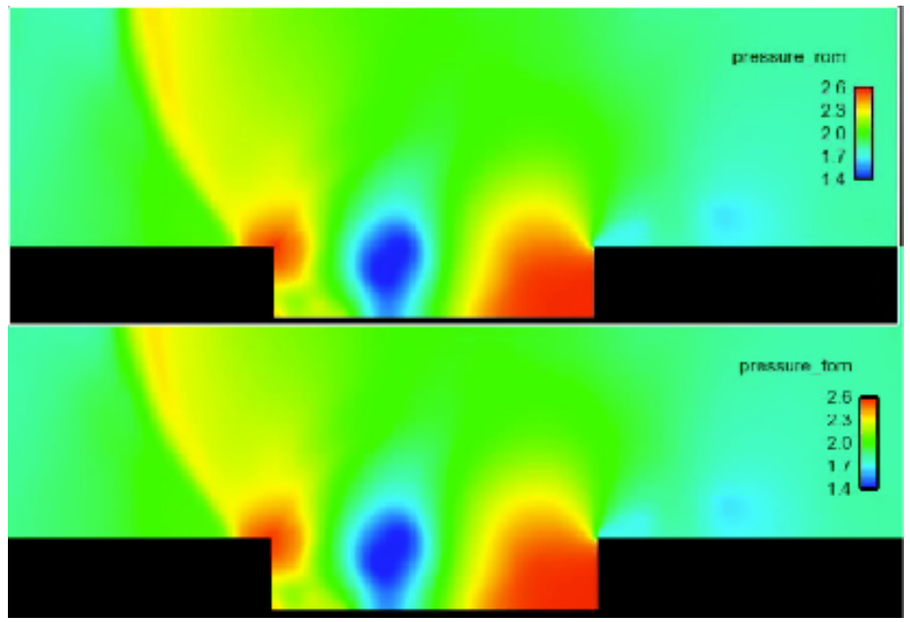
Model reduction can work well…

vorticity field | *pressure field*

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

high-fidelity
5 hours, 48 cores





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

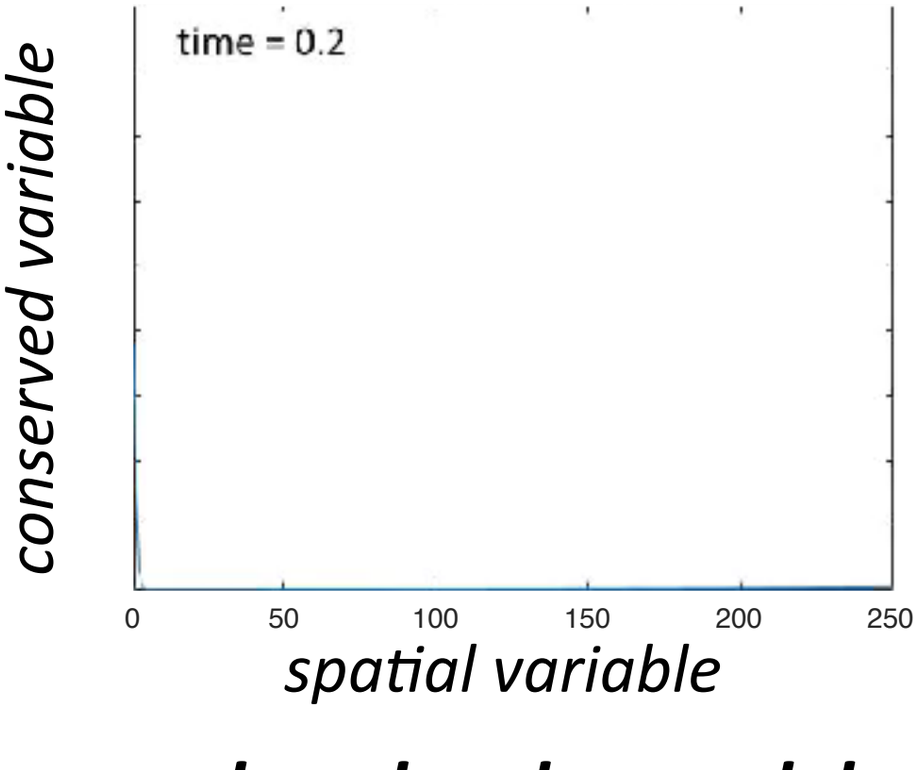
… however, this is not guaranteed

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

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

Illustration: viscous 1D Burgers' equation

high-fidelity model



reduced-order model

reduced-order model
inaccurate when Φ
insufficient

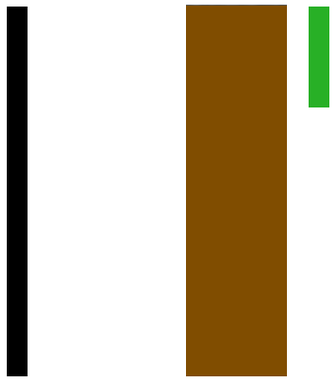
Key insight

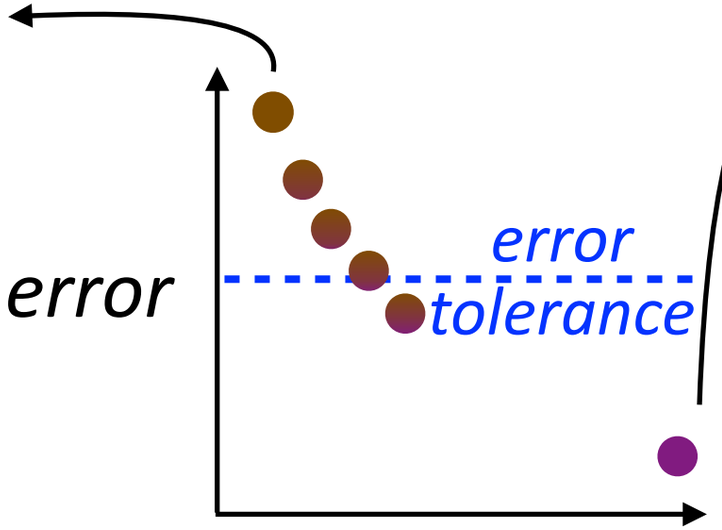
$$\mathbf{x}^n = \underset{\mathbf{v} \in \mathcal{S}}{\operatorname{argmin}} \|\mathbf{A}\mathbf{r}^n(\mathbf{v})\|_2^2$$

reduced-order model

$\mathcal{S} = \text{range}(\mathbf{\Phi}) \subset \mathbb{R}^N$

$\mathbf{x}^n = \mathbf{\Phi}\,\hat{\mathbf{x}}^n$

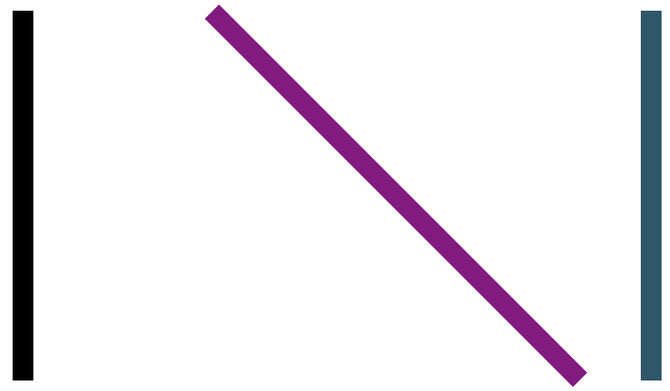




high-fidelity model

$\mathcal{S} = \text{range}(\mathbf{I}) = \mathbb{R}^N$

$\mathbf{x}^n = \mathbf{I}\,\hat{\mathbf{x}}^n$



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]

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

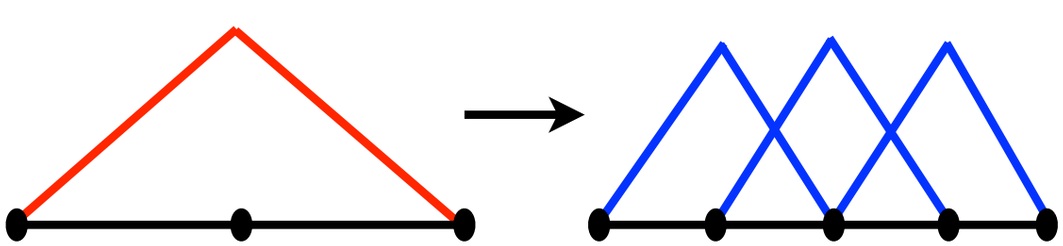
2. Augmented Krylov method [Carlberg, Forstall, Tuminaro, 2016]

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

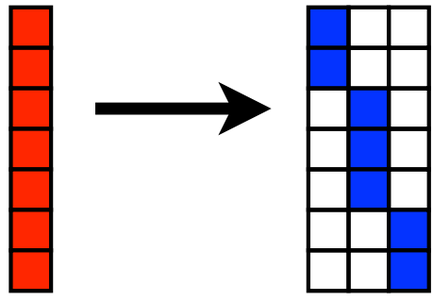
Main idea [Carlberg, 2015]

Model-reduction analogue to mesh-adaptive h-refinement

- 'Split' basis vectors



finite-element h-refinement

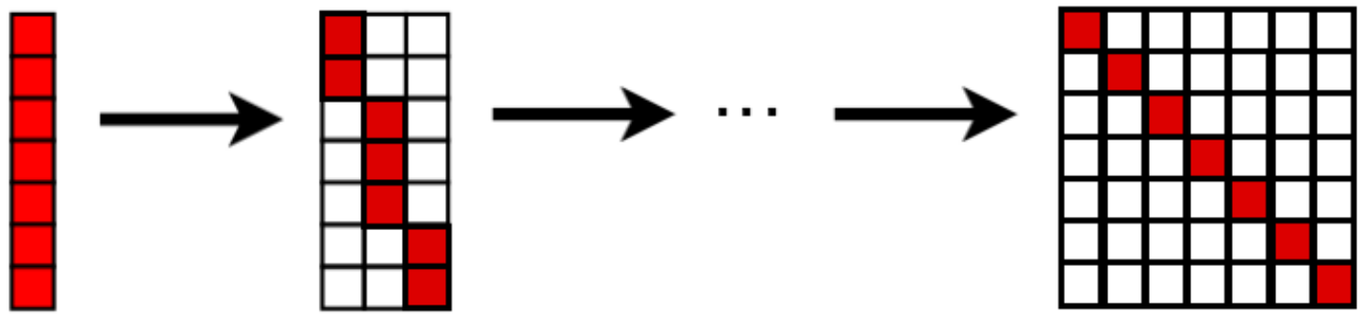


reduced-order-model h-refinement

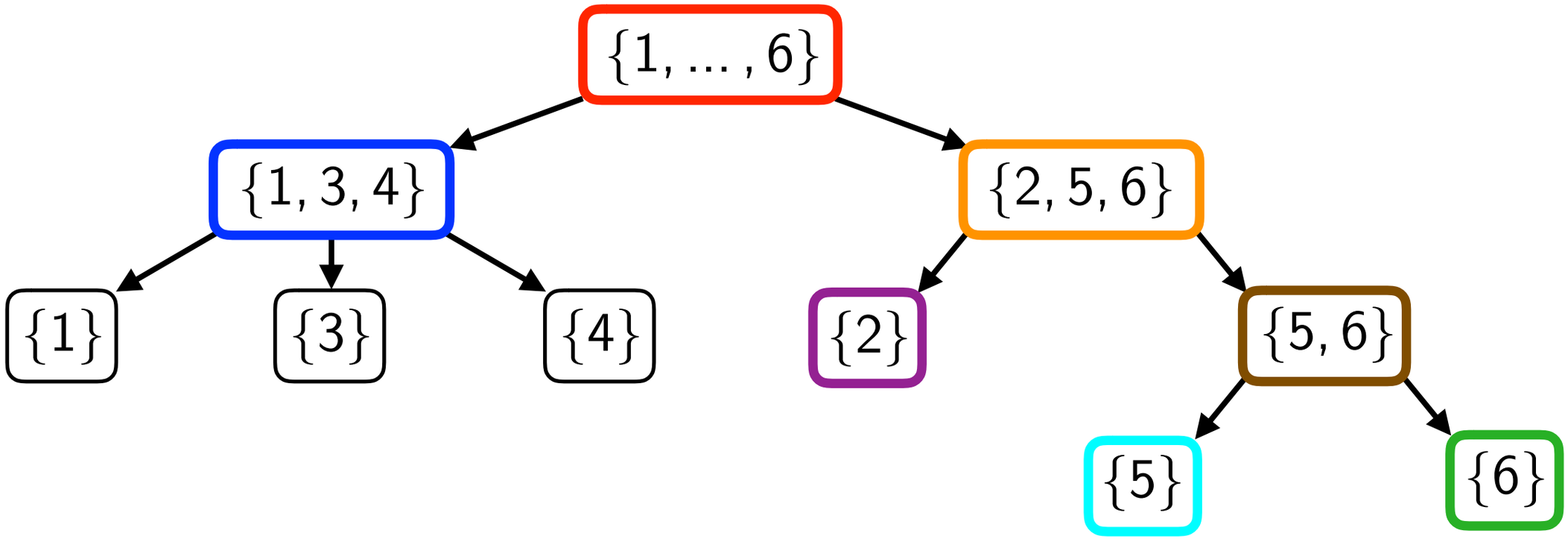
- Generate hierarchical subspaces

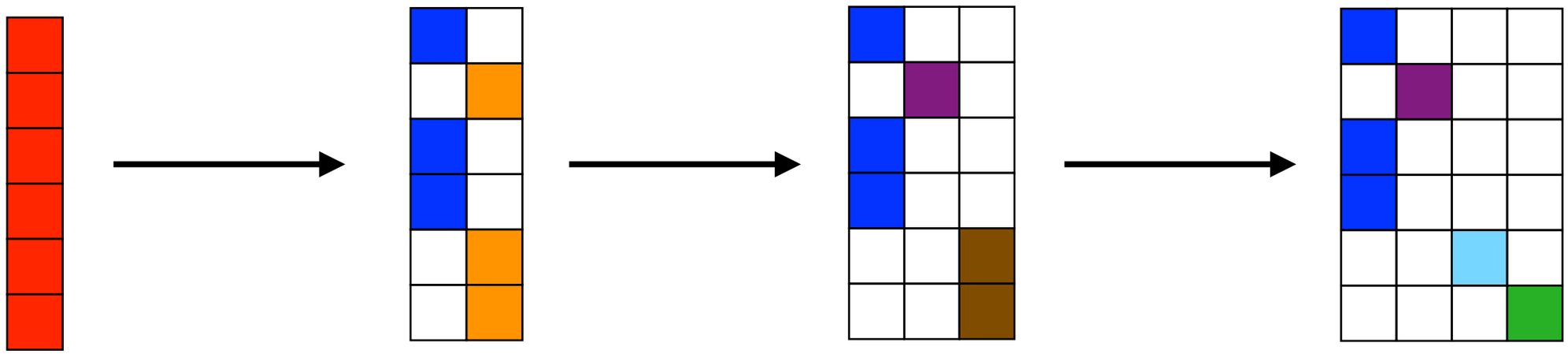
$$\text{range}\left(\begin{array}{c}\blacksquare\\ \blacksquare\\ \blacksquare\\ \blacksquare\\ \blacksquare\\ \blacksquare\end{array}\right) \subseteq \text{range}\left(\begin{array}{ccc}\blacksquare & & \\ \blacksquare & & \\ & \blacksquare & \\ & \blacksquare & \\ & \blacksquare & \\ & & \blacksquare\end{array}\right)$$

- Converges to the high-fidelity model



Tree encodes splitting



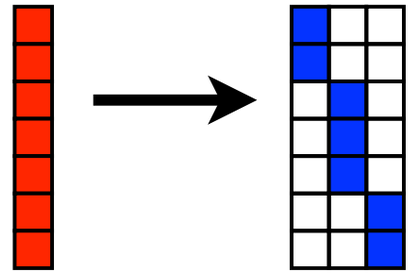


Tree requirements

Theorem [Carlberg, 2015]

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

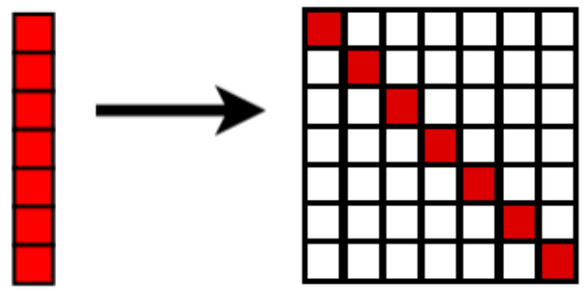
1. children have disjoint support, and
2. the union of the children is equal to the parent elements



Theorem [Carlberg, 2015]

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

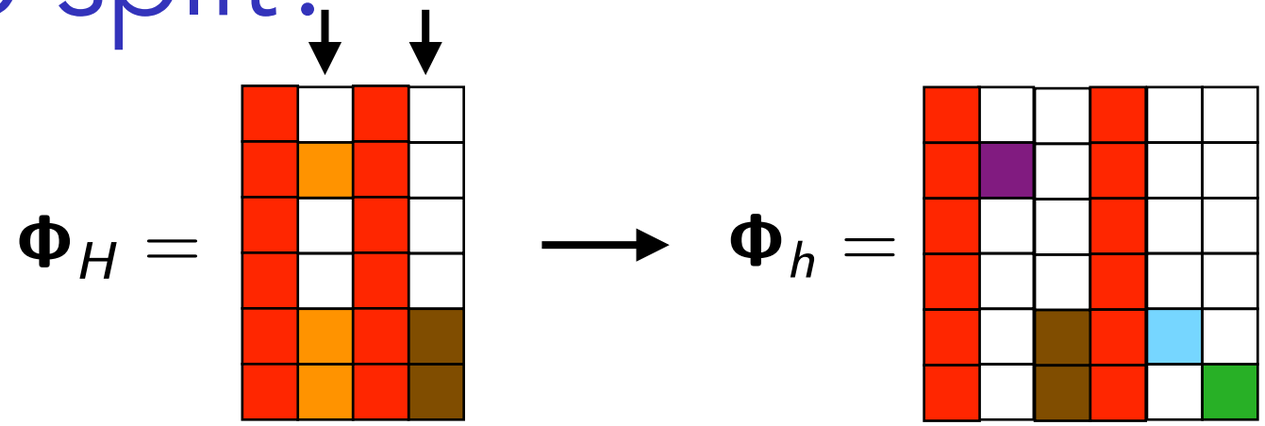
1. every element has a nonzero entry in >1 basis vector,
2. the root node includes all elements, and
3. each element has a leaf node.

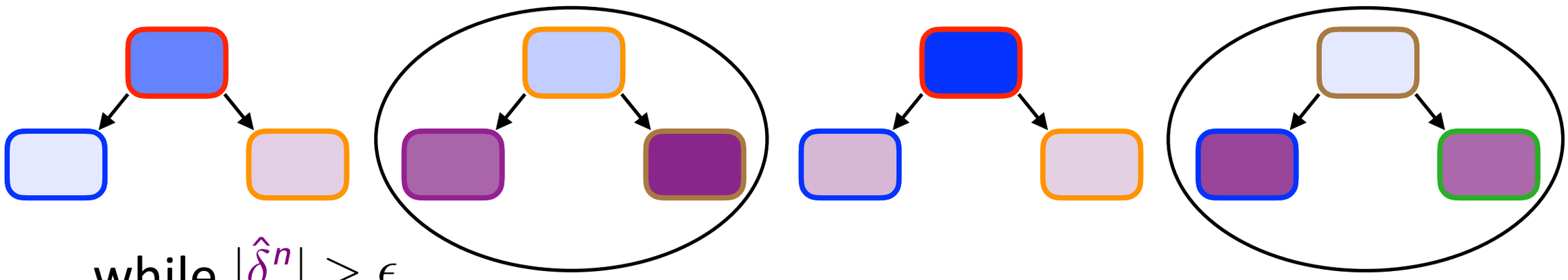


Tree-construction algorithm

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

Which vectors to split?





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

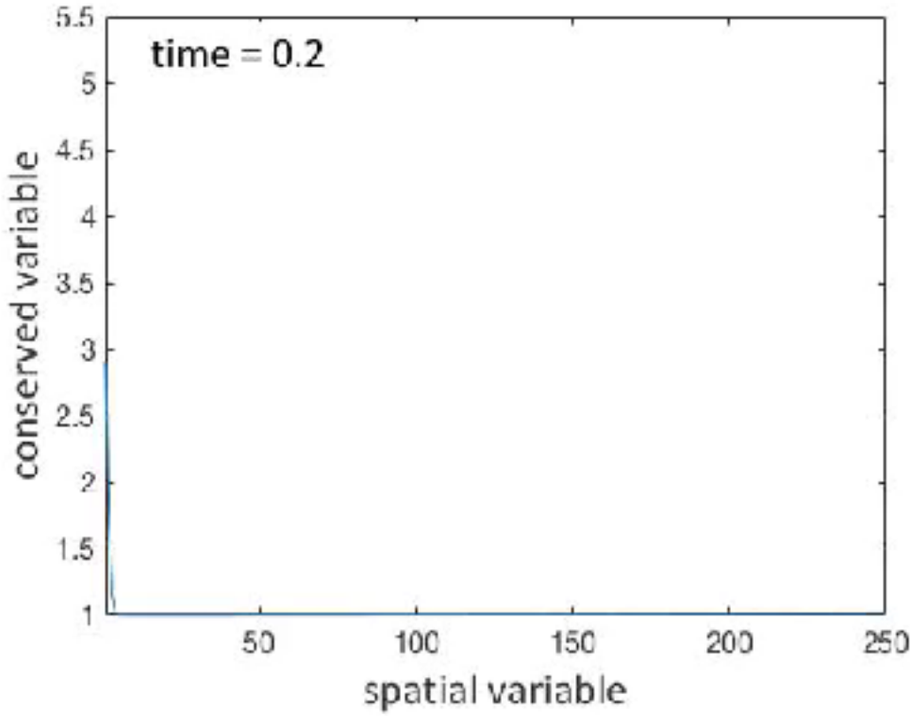
$$\Delta_i^n = |(\mathbf{I}_H^h \mathbf{y}_H^n)_i{}^T [\mathbf{\Phi}_h]_i{}^T \mathbf{r}^n(\mathbf{\Phi}_H \hat{\mathbf{x}}_H^n)|$$

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

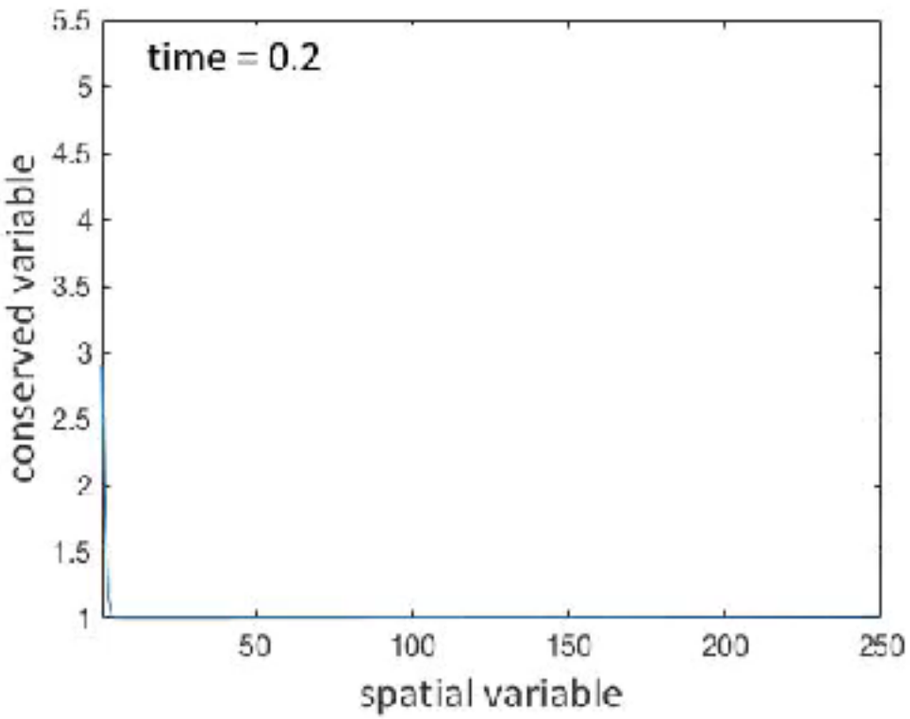
end

Illustration: inviscid 1D Burgers' equation

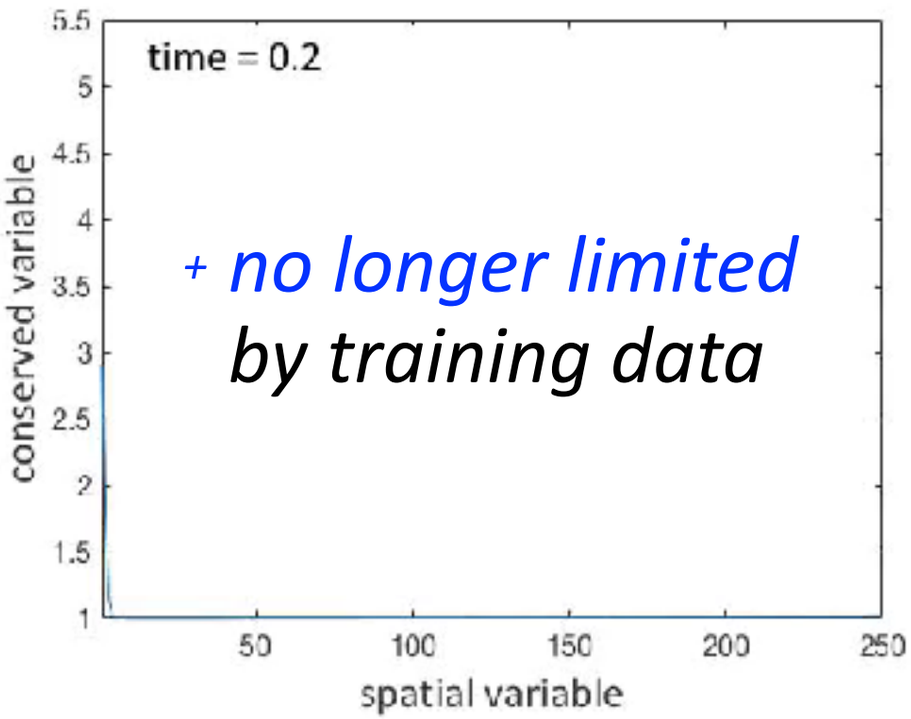
high-fidelity model



reduced-order model (dim 50)



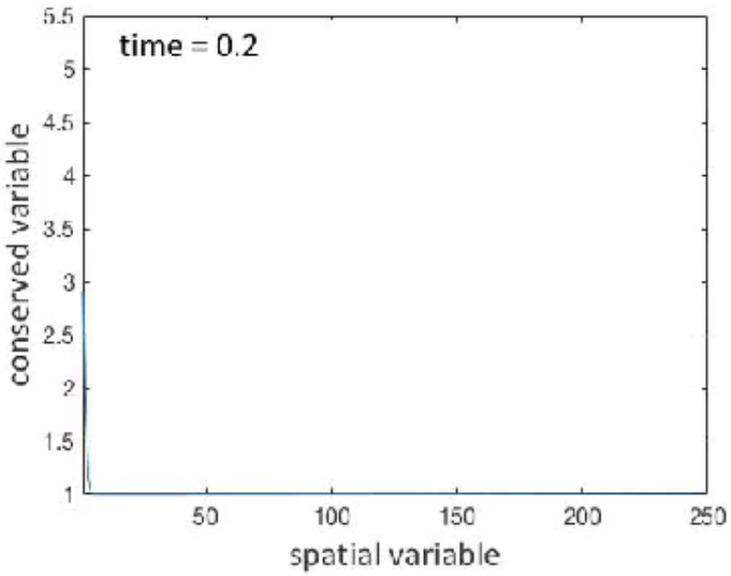
h-adaptive subspace (mean dim 48.5)



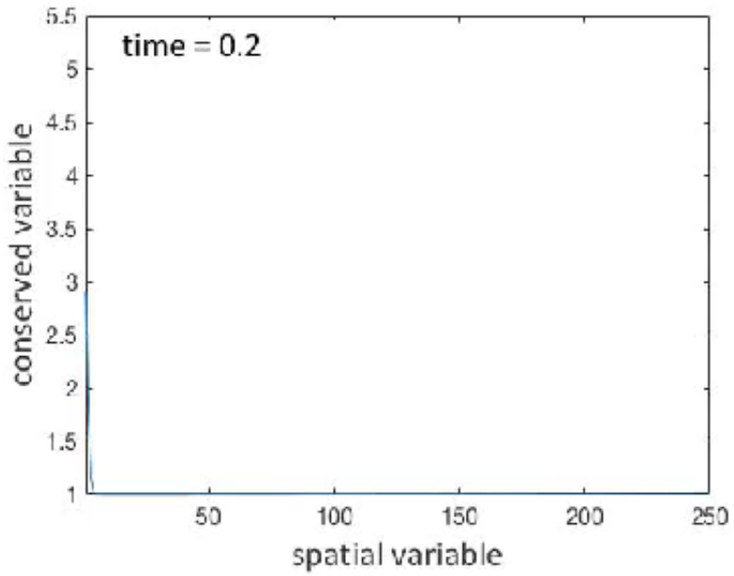
\+ *no longer limited by training data*

h-adaptivity enables error control

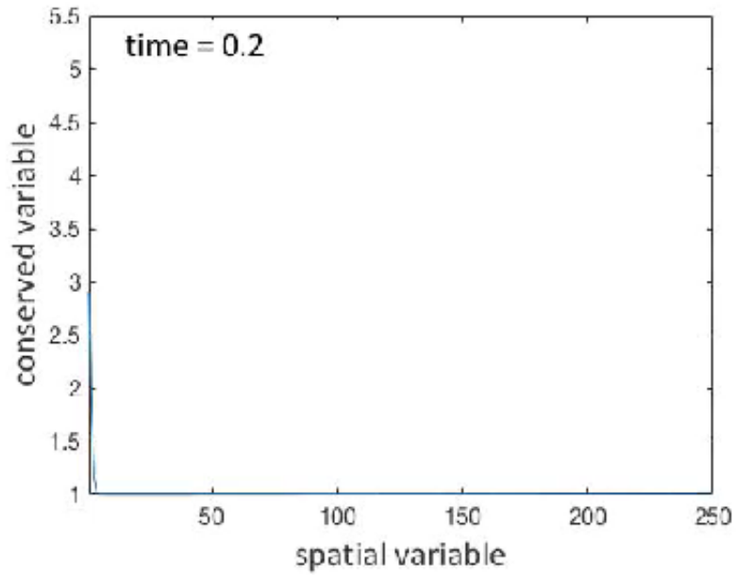
high-fidelity model



reduced-order models

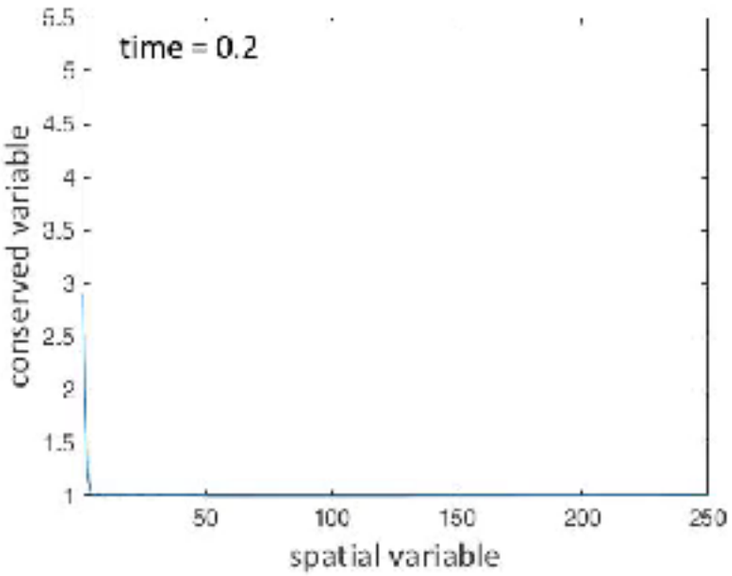


dimension 50

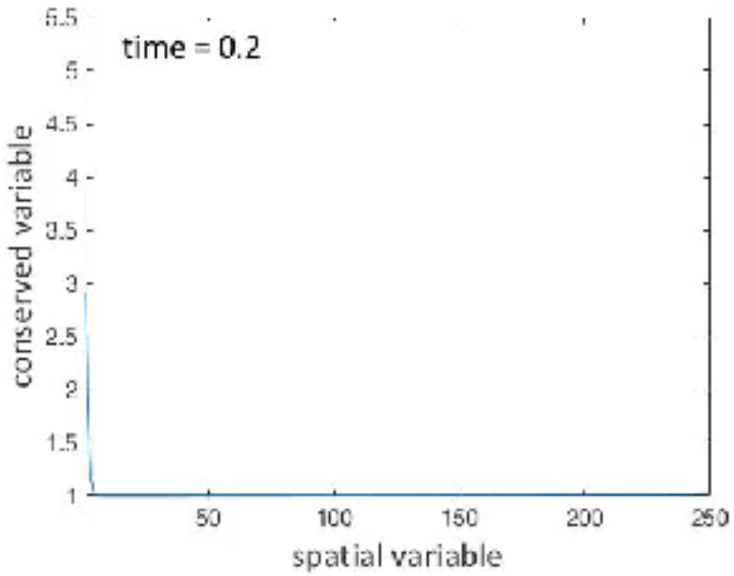


(maximum) dimension 150

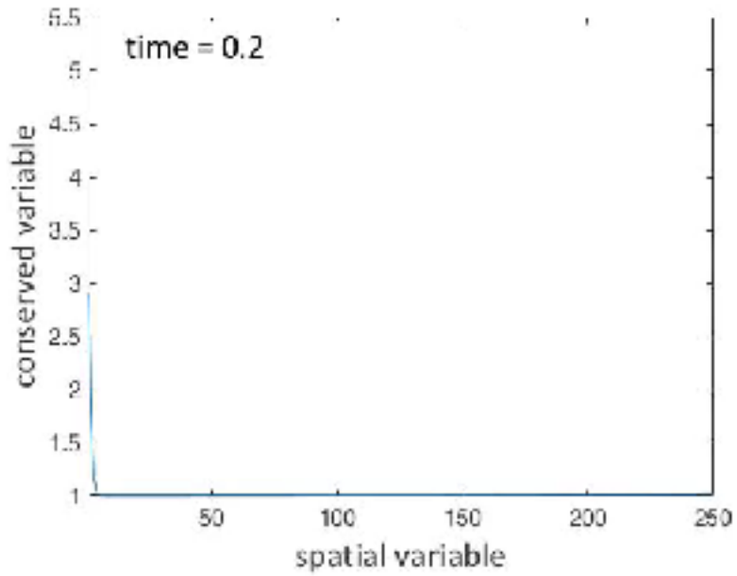
***h*-adaptive subspaces**



tolerance 0.3
mean dimension 31.7

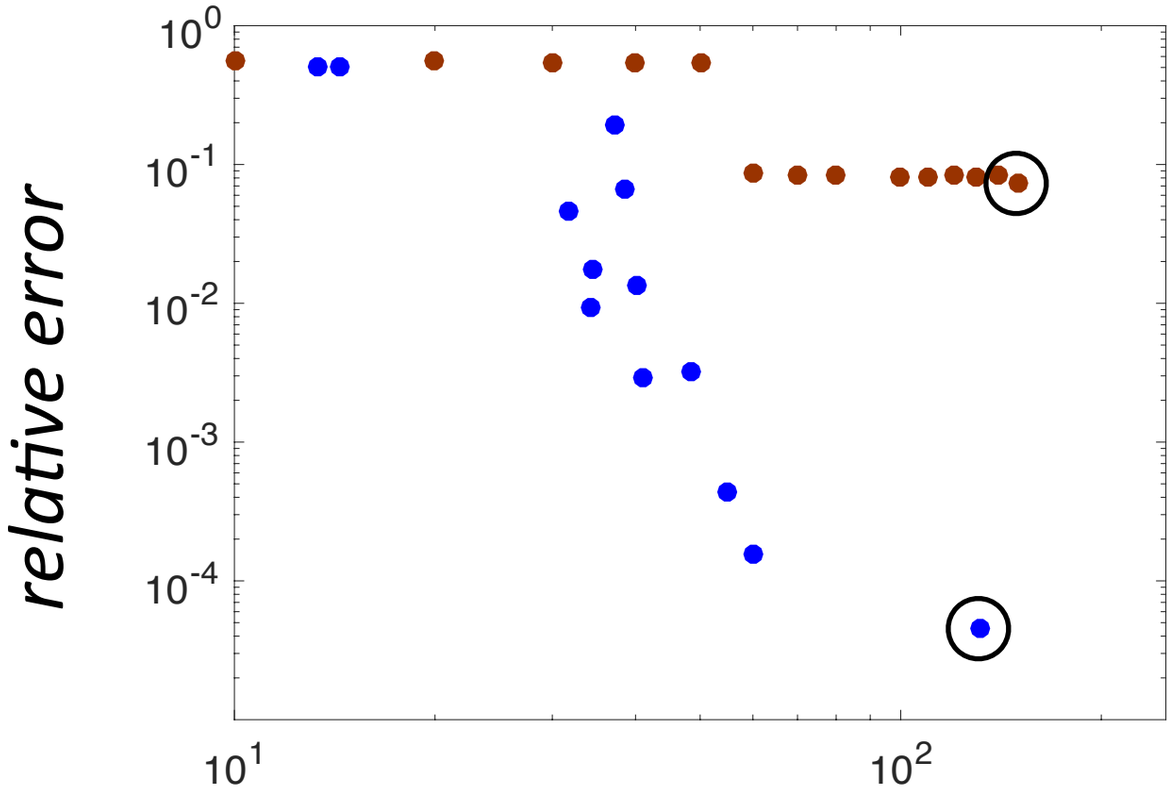


tolerance 0.1
mean dimension 41.0



tolerance 0.01
mean dimension 55.0

h-adaptivity provides an accurate, low-dim subspace



- reduced-order models
- *h*-adaptive subspaces

Reduced-order models

- *minimum error 7.5%*
- *cannot overcome insufficient training data*

h-adaptive subspaces

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

My research

Enable extreme-scale simulations for many-query problems

1. **Nonlinear model reduction**
   - \+ reduces model dimensionality and complexity
   - *accuracy*: LSPG projection [Carlberg, Bou-Mosleh, Farhat, 2011; Carlberg, Antil, Barone, 2017]
   - *low cost*: sample mesh [Carlberg, Farhat, Cortial, Amsallem, 2013]
   - *structure preservation* [Carlberg, Tuminaro, Boggs, 2015; Peng and Carlberg, 2017; Carlberg and Choi, 2017]

2. **Data-driven error modeling**
   - \+ rigorously accounts for model-reduction error
   - regression error modeling

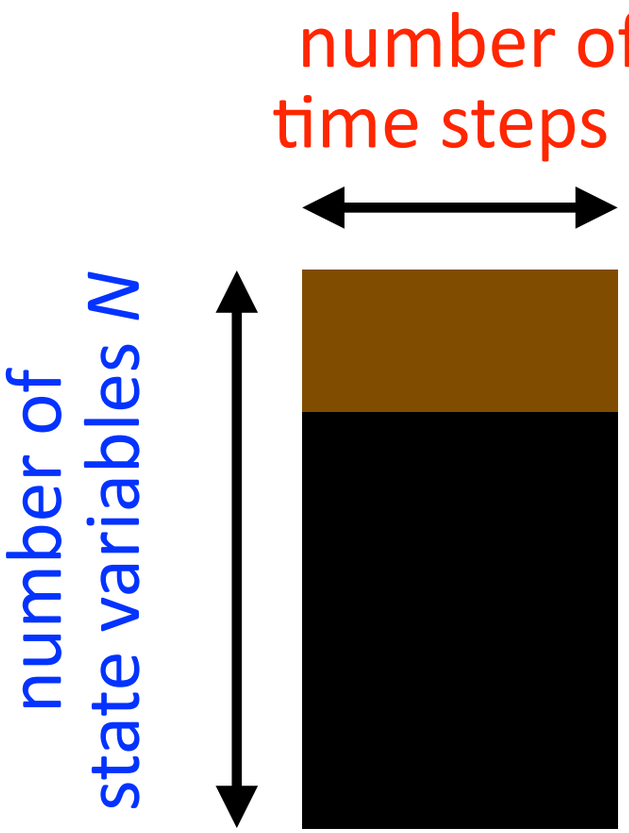
     [Drohmann and Carlberg, 2015; Trehan, Carlberg, Durlofsky, 2017; Freno and Carlberg, 2017]

3. **Data-driven numerical solvers**
   - \+ improves performance of numerical solvers
   - adaptive subspaces [Carlberg, 2015; Carlberg, Forstall, Tuminaro, 2016]
   - reduce temporal complexity

     [Carlberg, Ray, van Bloemen Waanders, 2015; Carlberg, Brencher, Haasdonk, Barth, 2017; Choi and Carlberg, 2017]

Temporal complexity

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



So far, we have focused on reducing the spatial complexity

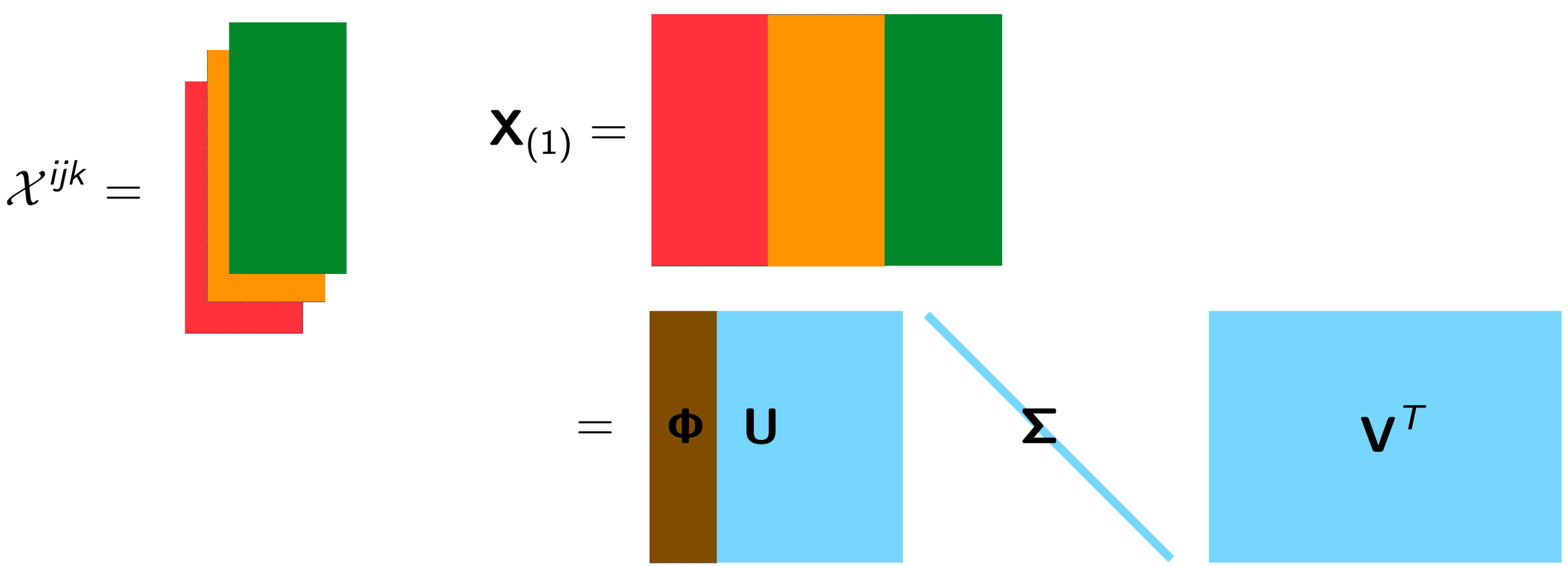
What about the temporal complexity?

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 values of **mode-1** unfolding*



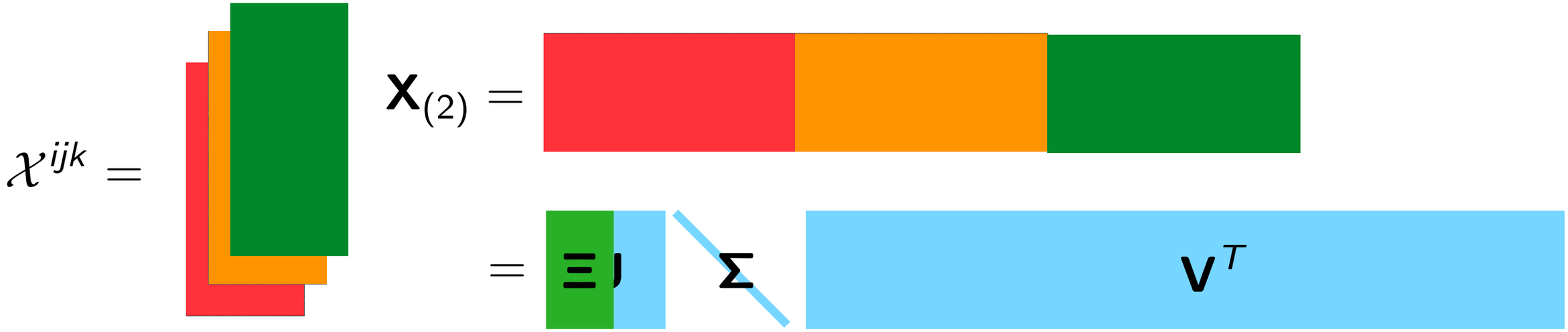
Φ *columns are principal components of the **spatial** simulation data*

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 values of **mode-2** unfolding*



*Ξ columns are principal components of the **temporal** simulation data*

How to integrate these data with the computational model?

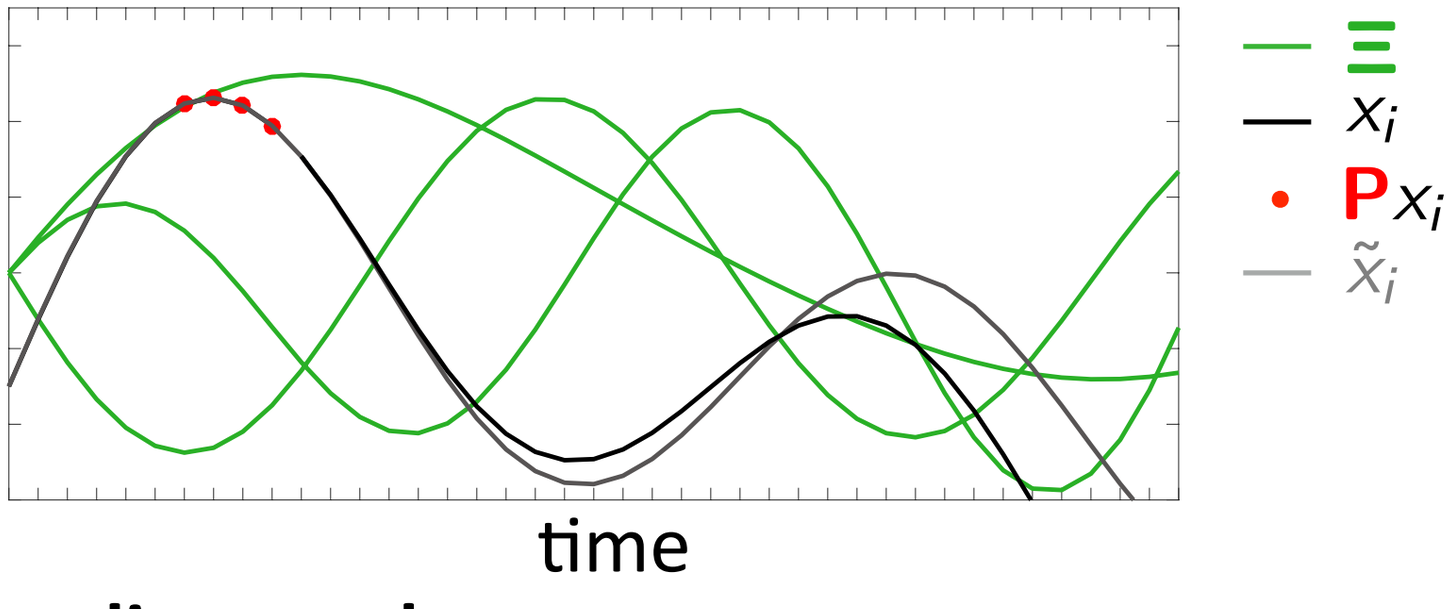
1. Space–time LSPG projection [Choi and Carlberg, 2017]
2. Data-driven solvers [Carlberg, Ray, van Bloemen Waanders, 2015; Carlberg, Brencher, Haasdonk, Barth, 2017]

Idea: forecasting via Gappy PCA in time

[Carlberg, Ray, van Bloemen Waanders, 2015]

$$x_i(t) \approx \tilde{x}_i(t) = \boldsymbol{\Xi}(t)(\mathbf{P}\boldsymbol{\Xi}(t))^{+}\mathbf{P}x_i(t)$$

state variable x_i



Data-driven nonlinear solver

[Carlberg, Ray, van Bloemen Waanders, 2015]

- use forecast $\tilde{x}_i$ as accurate initial guess for the Newton solver

\+ 50% speedup improvement observed; no accuracy loss

Data-driven time-parallel solver

[Carlberg, Brencher, Haasdonk, Barth, 2016]

- use forecast $\tilde{x}_i$ as accurate coarse propagator
- provably stable; ideal speedups possible; no accuracy loss
- 10x speedup improvements observed

Summary

Enable extreme-scale simulations for many-query problems

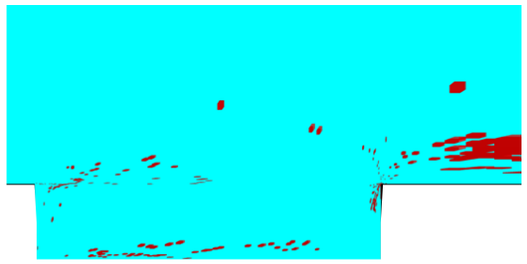
1. **Nonlinear model reduction**
   - \+ reduces model dimensionality and complexity
   - ‣ LSPG projection [Carlberg, Bou-Mosleh, Farhat, 2011; Carlberg, Antil, Barone, 2017]
   - ‣ sample mesh [Carlberg, Farhat, Cortial, Amsallem, 2013]
   - ‣ structure preservation [Carlberg, Tuminaro, Boggs, 2015; Peng and Carlberg, 2017; Carlberg and Choi, 2017]

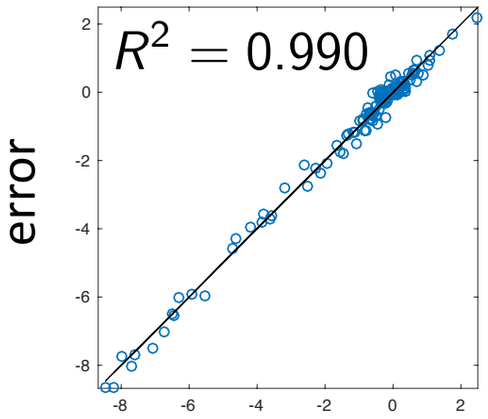
2. **Data-driven error modeling**
   - \+ rigorously accounts for model-reduction error
   - ‣ regression error modeling
     [Drohmann and Carlberg, 2015; Trehan, Carlberg, Durlofsky, 2017; Freno and Carlberg, 2017]

3. **Data-driven numerical solvers**
   - \+ improves performance of numerical solvers
   - ‣ adaptive subspaces [Carlberg, 2015; Carlberg, Forstall, Tuminaro, 2016]
   - ‣ reduce temporal complexity
     [Carlberg, Ray, van Bloemen Waanders, 2015; Carlberg, Brencher, Haasdonk, Barth, 2017; Choi and Carlberg, 2017]

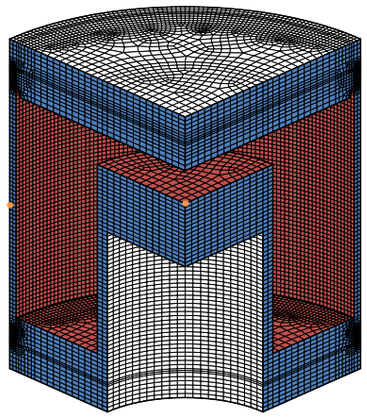
Questions?

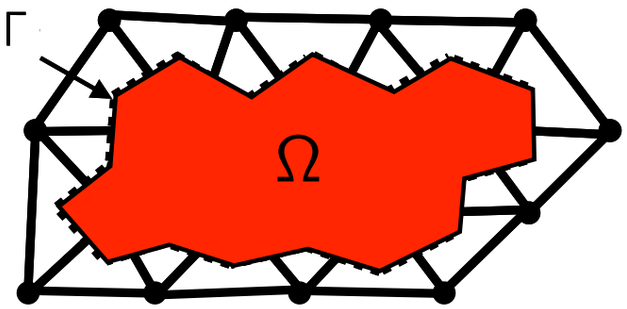
$\mathcal{X}^{ijk} =$

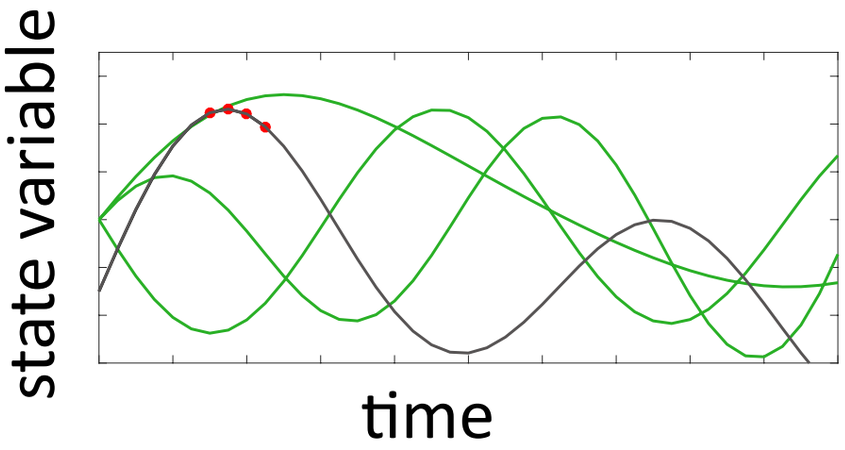




support vector machine
error prediction







$\mu \rightarrow$ *reduced-order model* $\rightarrow q^n_{\text{ROM}}$

$\downarrow \rho^n$

regression model $\rightarrow \hat{\delta}^n$

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