SAND2016-3699PE

# Nonlinear model reduction:

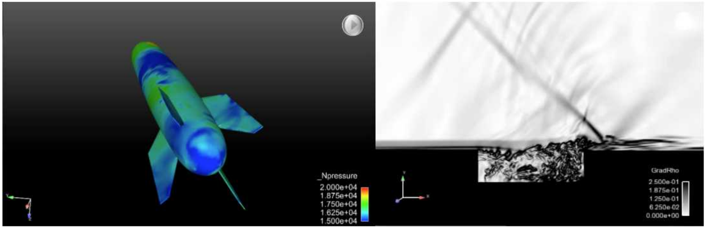
## discrete optimality, *h*-adaptivity, and time parallelism

Kevin Carlberg

Sandia National Laboratories

AMATH seminar
April 18, 2016

# Computational barrier at Sandia




- CFD model
  - 100 million cells
  - 200,000 time steps
- High simulation costs
  - 6 weeks, 5000 cores
  - 6 runs maxes out Cielo

**Barrier**

- Fast-turnaround design
- Uncertainty quantification

**Objective: break barrier via nonlinear model reduction**

# ROM: state of the art [Benner et al., 2015]

- Linear time-invariant systems: mature [Antoulas, 2005]
  - Balanced truncation [Moore, 1981]
  - Empirical balanced truncation [Willcox and Peraire, 2002, Rowley, 2005]
  - Moment matching

    [Bai, 2002, Freund, 2003, Gallivan et al., 2004, Baur et al., 2011]
  - Loewner framework [Lefteriu and Antoulas, 2010, Ionita and Antoulas, 2014]
  
  \+ *Reliable*: guaranteed stability, *a priori* error bounds
  
  \+ *Certified*: sharp, computable *a posteriori* error bounds
- Elliptic/parabolic PDEs (FEM): mature [Rozza et al., 2008]
  - Reduced-basis method

    [Prud'Homme et al., 2001, Veroy et al., 2003, Barrault et al., 2004]
  - Subsystem-based reduced-basis method

    [Maday and Rønquist, 2002, Phuong Huynh et al., 2013, Eftang and Patera, 2013]
  
  \+ *Reliable*: *a priori* error bounds
  
  \+ *Certified*: sharp, computable *a posteriori* error bounds
- Nonlinear dynamical systems: unproven
  - Proper orthogonal decomposition (POD)–Galerkin
  
  \- *Not reliable*: Stability and accuracy not guaranteed
  
  \- *Not certified*: error bounds not sharp

# Our research goal

Nonlinear model-reduction methods that are
accurate, low cost, certified, and reliable.

- Accuracy
  - Improve projection technique
  - Preserve problem structure
- Low cost
  - Sample-mesh approach
  - Leverage time-domain data
- Certification
  - Error bounds
  - Statistical error modeling
- Reliability
  - *A posteriori* $h$-refinement

# Our research goal

Nonlinear model-reduction methods that are
accurate, low cost, certified, and reliable.

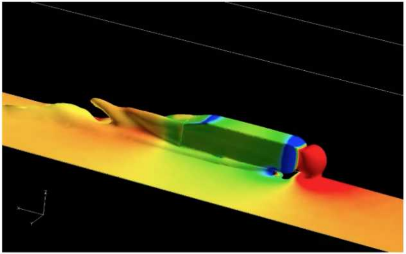
+ Accuracy
  - Improve projection technique [C. et al., 2011a, C. et al., 2015a]
  - Preserve problem structure
+ Low cost
  - Sample-mesh approach
  - Leverage time-domain data
+ Certification
  - Error bounds [C. et al., 2015a]
  - Statistical error modeling
+ Reliability
  - *A posteriori* *h*-refinement

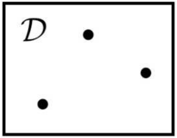
**Collaborators**: M. Barone (Sandia), H. Antil (GMU)

# POD–Galerkin: offline data collection

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{f}(\boldsymbol{x}; t, \boldsymbol{\mu}); \quad \boldsymbol{x}(0, \boldsymbol{\mu}) = \boldsymbol{x}^0(\boldsymbol{\mu}), \quad t \in [0, T], \quad \boldsymbol{\mu} \in \mathcal{D}$$

1 Collect ‘snapshots’ of the state

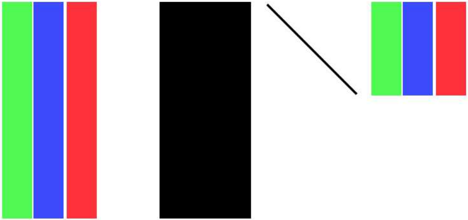


# POD–Galerkin: offline data collection

2 Data compression

- Compute SVD: $[\, \mathbf{X}_1 \ \mathbf{X}_2 \ \mathbf{X}_3 \,] = \mathbf{U} \ \boldsymbol{\Sigma} \ \mathbf{V}^T$

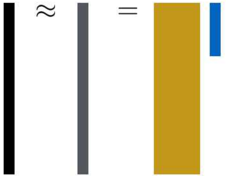


- Truncate: $\boldsymbol{\Phi} = [\boldsymbol{u}_1 \ \cdots \ \boldsymbol{u}_p]$

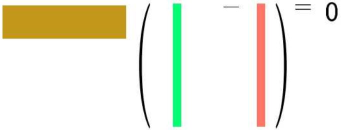
# POD–Galerkin: online projection

Full-order model: $$\boxed{\frac{d\boldsymbol{x}}{dt} = \boldsymbol{f}(\boldsymbol{x}; t, \boldsymbol{\mu}), \quad \boldsymbol{x}(0, \boldsymbol{\mu}) = \boldsymbol{x}^0(\boldsymbol{\mu})}$$

1 $\boldsymbol{x}(t) \approx \tilde{\boldsymbol{x}}(t) = \boldsymbol{\Phi}\hat{\boldsymbol{x}}(t)$

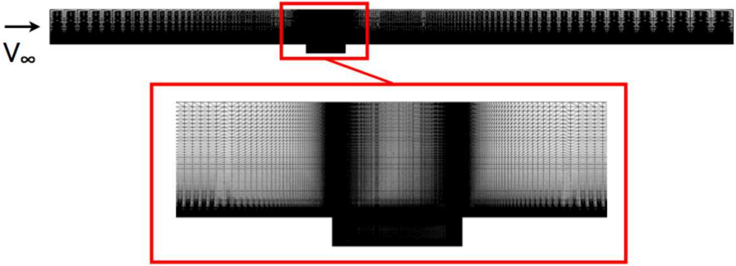


2 $\boldsymbol{\Phi}^T\left(\boldsymbol{f}(\tilde{\boldsymbol{x}}; t, \boldsymbol{\mu}) - \frac{d\tilde{\boldsymbol{x}}}{dt}\right) = 0$



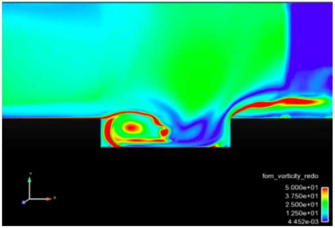

Galerkin ROM: $$\boxed{\frac{d\hat{\boldsymbol{x}}}{dt} = \boldsymbol{\Phi}^T\boldsymbol{f}(\boldsymbol{\Phi}\hat{\boldsymbol{x}}; t, \boldsymbol{\mu}), \quad \hat{\boldsymbol{x}}(0, \boldsymbol{\mu}) = \boldsymbol{\Phi}^T\boldsymbol{x}^0(\boldsymbol{\mu})}$$

# Cavity-flow problem. Collaborator: M. Barone (SNL)

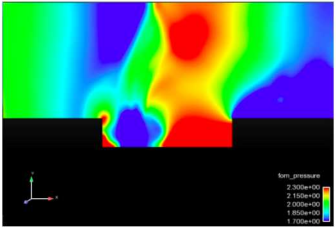



- Unsteady Navier–Stokes
- DES turbulence model
- 1.2 million degrees of freedom
- Re $= 6.3 \times 10^6$
- $M_\infty = 0.6$
- CFD code: AERO-F [Farhat et al., 2003]

# Full-order model responses

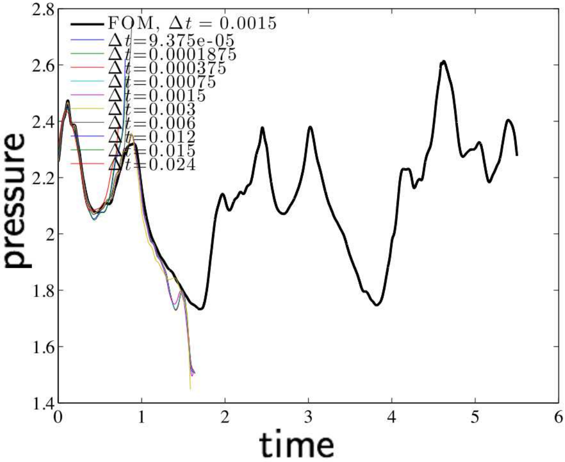



vorticity field




pressure field

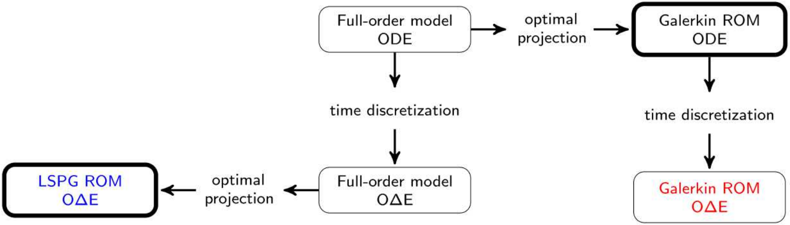
# POD–Galerkin failure




- Galerkin ROMs unstable

# How to construct a ROM for nonlinear dynamical systems?

- Optimize then discretize? (Galerkin)
- Discretize then optimize? (Least-squares Petrov–Galerkin)




- Outstanding questions:
  1. Which notion of optimality is better in practice?
  2. Discrete-time error bounds?
  3. Time step selection?

Full-order model
ODE

$\downarrow$

time discretization

$\downarrow$

Full-order model
OΔE

# Full-order model (FOM)

- ODE: time continuous

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{f}(\boldsymbol{x}, t), \quad \boldsymbol{x}(0) = \boldsymbol{x}^0, \quad t \in [0, T]$$

- OΔE, linear multistep schemes: $\boxed{\boldsymbol{r}^n(\boldsymbol{x}^n) = 0}$, $n = 1, \ldots, N$

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

$$\boldsymbol{x}^n = \boldsymbol{x}^n \text{ (explicit state update)}$$

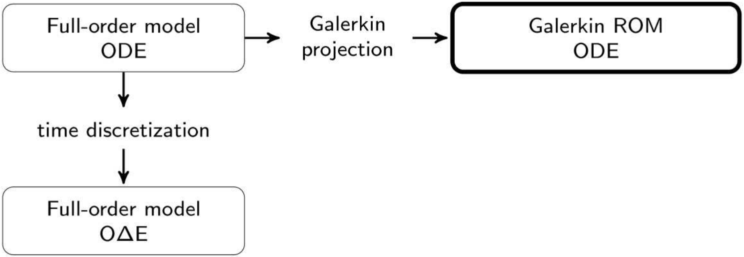
- OΔE, Runge–Kutta: $\boxed{\boldsymbol{r}_i^n(\boldsymbol{x}_1^n, \ldots, \boldsymbol{x}_s^n) = 0}$, $i = 1, \ldots, s$

$$\boldsymbol{r}_i^n(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_s) := \boldsymbol{x}_i - \boldsymbol{f}(\boldsymbol{x}^{n-1} + \Delta t \sum_{j=1}^{s} a_{ij} \boldsymbol{x}_j, t^{n-1} + c_i \Delta t)$$

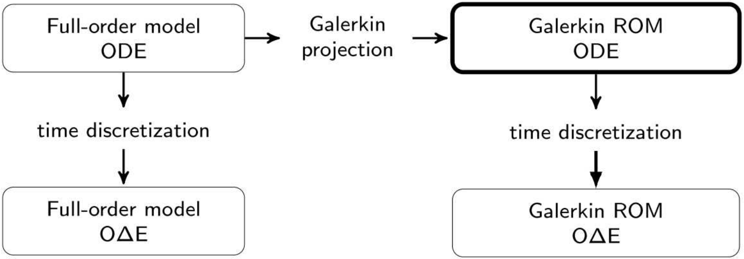
$$\boldsymbol{x}^n = \boldsymbol{x}^{n-1} + \Delta t \sum_{i=1}^{s} b_i \boldsymbol{x}_i^n \text{ (explicit state update)}$$

*This talk focuses on linear multistep schemes.*

# Galerkin ROM: first optimize



# Galerkin: first optimize, then discretize



# Galerkin ROM

- ODE

$$\frac{d\hat{\boldsymbol{x}}}{dt} = \boldsymbol{\Phi}^T \boldsymbol{f}(\boldsymbol{\Phi}\hat{\boldsymbol{x}}, t), \quad \hat{\boldsymbol{x}}(0) = \boldsymbol{\Phi}^T \boldsymbol{x}^0, \quad t \in [0, T]$$

+ Continuous velocity $\frac{d\hat{\boldsymbol{x}}}{dt}$ is optimal

**Theorem (Galerkin ROM: continuous optimality)**

*The Galerkin ROM velocity minimizes the time-continuous FOM residual:*

$$\frac{d\tilde{\boldsymbol{x}}}{dt}(\boldsymbol{x}, t) = \arg \min_{\boldsymbol{v} \in range(\boldsymbol{\Phi})} \|\boldsymbol{v} - \boldsymbol{f}(\boldsymbol{x}, t)\|_2^2$$

- OΔE

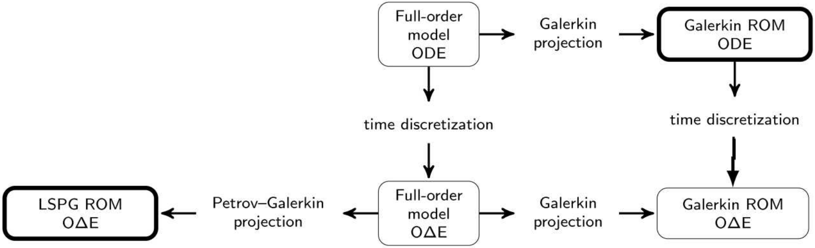
$$\hat{\boldsymbol{r}}^n(\hat{\boldsymbol{x}}^n) = 0, \quad n = 1, \dots, N$$

$$\hat{\boldsymbol{r}}^n(\hat{\boldsymbol{x}}) := \alpha_0 \hat{\boldsymbol{x}} - \Delta t \beta_0 \boldsymbol{\Phi}^T \boldsymbol{f}(\boldsymbol{\Phi}\hat{\boldsymbol{x}}, t^n) + \sum_{j=1}^{k} \alpha_j \hat{\boldsymbol{x}}^{n-j} - \Delta t \sum_{j=1}^{k} \beta_j \boldsymbol{\Phi}^T \boldsymbol{f}\left(\boldsymbol{\Phi}\hat{\boldsymbol{x}}^{n-j}, t^{n-j}\right)$$

- Discrete state $\hat{\boldsymbol{x}}^n$ is not generally optimal

*Can we fix this? Will doing so help?*

# LSPG ROM: first discretize, then optimize



# LSPG ROM

- FOM OΔE

$$\boldsymbol{r}^n(\boldsymbol{x}^n) = 0, \quad n = 1, \dots, N$$

- LSPG ROM OΔE:

$$\hat{\boldsymbol{x}}^n = \arg\min_{\hat{\boldsymbol{z}} \in \mathbb{R}^p} \|\boldsymbol{A}\boldsymbol{r}^n(\boldsymbol{\Phi}\hat{\boldsymbol{z}})\|_2^2.$$

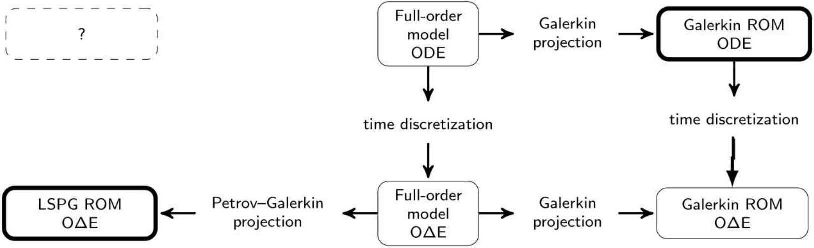
$$\Updownarrow$$

$$\boldsymbol{\Psi}^n(\hat{\boldsymbol{x}}^n)^T \boldsymbol{r}^n(\boldsymbol{\Phi}\hat{\boldsymbol{x}}^n) = 0, \quad \boldsymbol{\Psi}^n(\hat{\boldsymbol{x}}) := \boldsymbol{A}^T\boldsymbol{A}\frac{\partial \boldsymbol{r}^n}{\partial \boldsymbol{x}}(\boldsymbol{\Phi}\hat{\boldsymbol{x}})\boldsymbol{\Phi}$$

  - $\boldsymbol{A} = \boldsymbol{I}$: LSPG [LeGresley, 2006, Bui-Thanh et al., 2008, C. et al., 2011a]

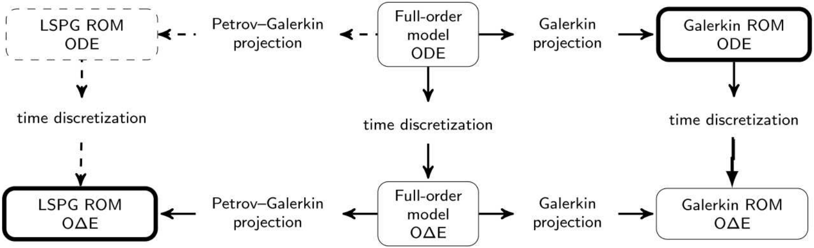
\+ Discrete solution is optimal

# Does the LSPG ROM have a time-continuous representation?



# Does the LSPG ROM have a time-continuous representation?

Sometimes.



# LSPG ROM: continuous representation

**Theorem**

*The LSPG ROM is equivalent to applying a Petrov–Galerkin projection to the FOM ODE with test basis*

$$\boldsymbol{\Psi}(\hat{\mathbf{x}}, t) = \boldsymbol{A}^T\boldsymbol{A}\left(\alpha_0\boldsymbol{I} - \Delta t\beta_0\frac{\partial \boldsymbol{f}}{\partial \mathbf{x}}(\mathbf{x}^0 + \boldsymbol{\Phi}\hat{\mathbf{x}}, t)\right)\boldsymbol{\Phi}$$

*if*

1. $\beta_j = 0$, $j \geq 1$ *(e.g., a single-step method),*
2. *the velocity* $\boldsymbol{f}$ *is linear in the state, or*
3. $\beta_0 = 0$ *(i.e., explicit schemes).*

*Time-continuous test basis depends on time-discretization parameters!*

# Are the two approaches ever equivalent?

- Galerkin: $\boldsymbol{\Phi}^T \boldsymbol{r}^n(\boldsymbol{\Phi}\hat{\boldsymbol{x}}^n) = 0$
- LSPG: $\boldsymbol{\Psi}^n(\hat{\boldsymbol{x}}^n)^T \boldsymbol{r}^n(\boldsymbol{\Phi}\hat{\boldsymbol{x}}^n) = 0$

Does $\boldsymbol{\Psi}^n(\hat{\boldsymbol{x}}^n) = \boldsymbol{\Phi}$ ever?

Yes.

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

**Theorem**

*The two approaches are equivalent ($\boldsymbol{\Psi}^n(\hat{\boldsymbol{x}}) = \boldsymbol{\Phi}$)*

1. *in the limit of $\Delta t \rightarrow 0$ with $\boldsymbol{A} = 1/\sqrt{\alpha_0}\boldsymbol{I}$,*
2. *if the scheme is explicit ($\beta_0 = 0$) with $\boldsymbol{A} = 1/\sqrt{\alpha_0}\boldsymbol{I}$, or*
3. *if $\frac{\partial \boldsymbol{r}^n}{\partial \boldsymbol{x}}$ is positive definite with $[\frac{\partial \boldsymbol{r}^n}{\partial \boldsymbol{x}}]^{-1} = \boldsymbol{A}^T\boldsymbol{A}$.*

- Runge–Kutta

$$\boldsymbol{\Psi}_{ij}^{n}(\hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_s) := \boldsymbol{A}_i^T \boldsymbol{A}_i \frac{\partial \mathbf{r}_i^n}{\partial \mathbf{x}_j}(\hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_s)$$

$$= \boldsymbol{A}_i^T \boldsymbol{A}_i \left( \boldsymbol{I}\delta_{ij} - \Delta t a_{ij} \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x}^{n-1} + \Delta t \boldsymbol{\Phi} \sum_{k=1}^{s} a_{ik} \hat{\mathbf{x}}_k, t^{n-1} + c_i \Delta t) \right) \boldsymbol{\Phi}$$

## Corollary

*Galerkin projection is discrete-optimal (i.e., $\boldsymbol{\Psi}^n(\hat{\mathbf{x}}) = \boldsymbol{\Phi}$) for Runge–Kutta schemes with $\boldsymbol{A}_i = \boldsymbol{I}$ in the limit of $\Delta t \to 0$.*

# Discrete-time error bound

**Theorem**

*If the following conditions hold:*

1. $\boldsymbol{f}(\cdot, t)$ *is Lipschitz continuous with Lipschitz constant* $\kappa$*, and*
2. $\Delta t$ *is such that* $0 < h := |\alpha_0| - |\beta_0|\kappa\Delta t$*,*

*then*

$$\|\delta \mathbf{x}_G^n\| \leq \frac{\Delta t}{h}\sum_{\ell=0}^{k}|\beta_\ell|\|\left(\boldsymbol{I} - \mathbb{V}\right)\boldsymbol{f}\left(\mathbf{x}_0 + \boldsymbol{\Phi}\hat{\mathbf{x}}_G^{n-\ell}\right)\| + \frac{1}{h}\sum_{\ell=1}^{k}\left(|\beta_\ell|\kappa\Delta t + |\alpha_\ell|\right)\|\delta \mathbf{x}_G^{n-\ell}\|$$

$$\|\delta \mathbf{x}_L^n\| \leq \frac{\Delta t}{h}\sum_{\ell=0}^{k}|\beta_\ell|\|\left(\boldsymbol{I} - \mathbb{P}^n\right)\boldsymbol{f}\left(\mathbf{x}_0 + \boldsymbol{\Phi}\hat{\mathbf{x}}_L^{n-\ell}\right)\| + \frac{1}{h}\sum_{\ell=1}^{k}\left(|\beta_\ell|\kappa\Delta t + |\alpha_\ell|\right)\|\delta \mathbf{x}_L^{n-\ell}\|,$$

*with*

- $\delta \mathbf{x}_G^n := \mathbf{x}_\star^n - \boldsymbol{\Phi}\hat{\mathbf{x}}_G^n$.
- $\delta \mathbf{x}_L^n := \mathbf{x}_\star^n - \boldsymbol{\Phi}\hat{\mathbf{x}}_L^n$
- $\mathbb{V} := \boldsymbol{\Phi}\boldsymbol{\Phi}^T$
- $\mathbb{P}^n := \boldsymbol{\Phi}\left(\left(\boldsymbol{\Psi}^n\right)^T\boldsymbol{\Phi}\right)^{-1}\left(\boldsymbol{\Psi}^n\right)^T$

# LSPG ROM yields a smaller error bound

**Theorem (Backward Euler)**

*If conditions (1) and (2) hold, then*

$$\|\delta \boldsymbol{x}_G^n\| \leq \Delta t \sum_{j=0}^{n-1} \frac{1}{(h)^{j+1}} \underbrace{\left\| \left(\boldsymbol{I} - \mathbb{V}\right) \boldsymbol{f}\left(\boldsymbol{x}_0 + \boldsymbol{\Phi}\hat{\boldsymbol{x}}_G^{n-j}\right) \right\|}_{\varepsilon_G^{n-j}}$$

$$\|\delta \boldsymbol{x}_L^n\| \leq \Delta t \sum_{j=0}^{n-1} \frac{1}{(h)^{j+1}} \underbrace{\left\| \left(\boldsymbol{I} - \mathbb{P}^{n-j}\right) \boldsymbol{f}\left(\boldsymbol{x}_0 + \boldsymbol{\Phi}\hat{\boldsymbol{x}}_L^{n-j}\right) \right\|}_{\varepsilon_L^{n-j}}$$

$$\varepsilon_G^k = \|\boldsymbol{\Phi}\hat{\boldsymbol{x}}_G^k - \Delta t \boldsymbol{f}\left(\boldsymbol{x}_0 + \boldsymbol{\Phi}\hat{\boldsymbol{x}}_G^k\right) - \boldsymbol{\Phi}\hat{\boldsymbol{x}}_G^{k-1}\|$$

$$\varepsilon_L^k = \|\boldsymbol{\Phi}\hat{\boldsymbol{x}}_L^k - \Delta t \boldsymbol{f}\left(\boldsymbol{x}_0 + \boldsymbol{\Phi}\hat{\boldsymbol{x}}_L^k\right) - \boldsymbol{\Phi}\hat{\boldsymbol{x}}_L^{k-1}\| = \min_{\boldsymbol{y}} \|\boldsymbol{\Phi}\boldsymbol{y} - \Delta t \boldsymbol{f}\left(\boldsymbol{x}_0 + \boldsymbol{\Phi}\boldsymbol{y}\right) - \boldsymbol{\Phi}\hat{\boldsymbol{x}}_L^{k-1}\|$$

**Corollary (LSPG smaller error bound)**

*If* $\hat{\boldsymbol{x}}_L^{k-1} = \hat{\boldsymbol{x}}_G^{k-1}$, *then* $\varepsilon_L^k \leq \varepsilon_G^k$.

# LSPG ROM has an interesting time-step dependence

**Corollary (Backward Euler)**

*Define*

- $\Delta \hat{\mathbf{x}}_L^j := \hat{\mathbf{x}}_L^j - \hat{\mathbf{x}}_L^{j-1}$ *and*
- $\Delta \bar{\mathbf{x}}^j$*: full-space solution increment from* $\hat{\mathbf{x}}_L^{j-1}$.

*Then, the LSPG error can also be bounded as*

$$\|\delta \mathbf{x}_L^n\| \leq \Delta t(1+\kappa\Delta t) \sum_{j=0}^{n-1} \frac{\mu^{n-j}}{(h)^{j+1}} \|\boldsymbol{f}(\hat{\mathbf{x}}_L^{j-1} + \Delta \bar{\mathbf{x}}^{n-j})\|$$

*with* $\mu^j := \|\boldsymbol{\Phi}\Delta \hat{\mathbf{x}}_L^j - \Delta \bar{\mathbf{x}}^j\| / \|\Delta \bar{\mathbf{x}}^j\|$.

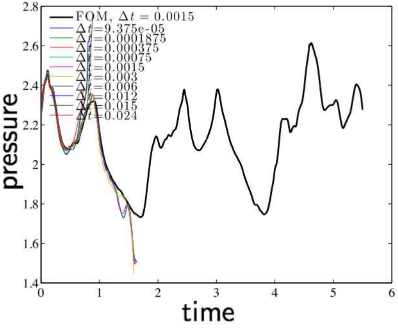
Effect of decreasing $\Delta t$:

\+ The terms $\Delta t(1+\kappa\Delta t)$ and $1/(h)^{j+1}$ decrease

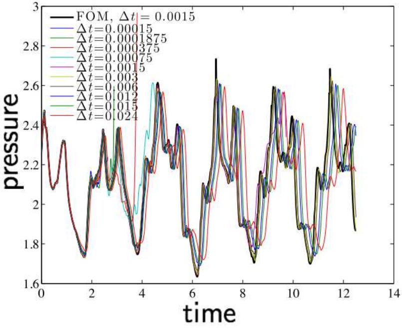
\- The number of total time instances $n$ increases

? The term $\mu^{n-j}$ may increase or decrease, depending on the spectral content of the basis $\boldsymbol{\Phi}$

# Galerkin and LSPG responses for basis dimension $p = 204$




(a) Galerkin

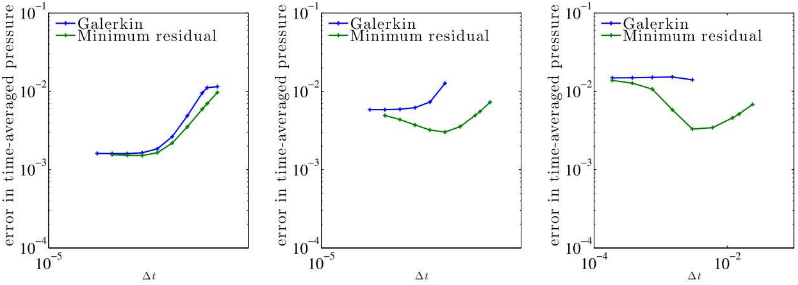



(b) LSPG

- Galerkin ROMs unstable for long time intervals

+ LSPG ROMs accurate and stable (most time steps)

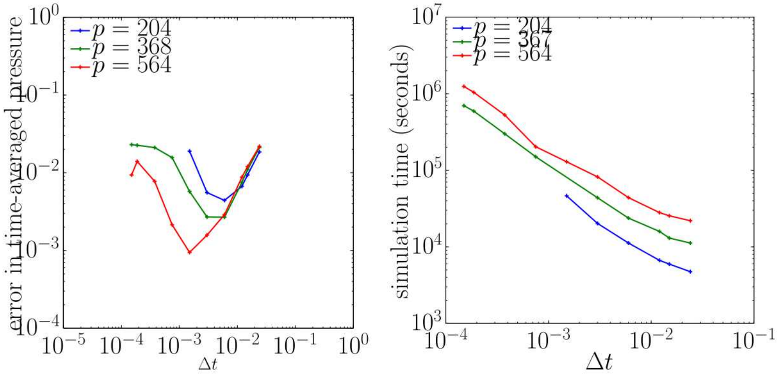
# LSPG ROM: superior performance




(c) $0 \leq t \leq 0.55$ (d) $0 \leq t \leq 1.1$ (e) $0 \leq t \leq 1.54$

✓ LSPG ROM yields a smaller error for all time intervals and time steps.

# LSPG performance ($t \leq 12.5$ sec)




✓ An intermediate $\Delta t$ produces the lowest error and better speedup.

$p = 564$ case:

- $\Delta t = 1.875 \times 10^{-4}$ sec: relative error $= 1.40\%$, time $= 289$ hrs
- $\Delta t = 1.5 \times 10^{-3}$ sec: relative error $= 0.095\%$, time $= 35.8$ hrs

# Summary: Improve projection technique

- Discrete optimality (LSPG) outperforms continuous optimality (Galerkin) in practice
- Equivalence conditions
  1. Limit of $\Delta t \to 0$
  2. Explicit schemes
  3. Positive definite residual Jacobians
- Discrete-time error bounds
  - LSPG ROM yields smaller error bound than Galerkin
  - Ambiguous role of time step $\Delta t$
- Numerical experiments
  - LSPG ROM yields a smaller error than Galerkin
  - Equivalent as $\Delta t \to 0$
  - Error minimized for intermediate $\Delta t$
- **Reference**: C., Barone, and Antil. Galerkin v. least-squares Petrov–Galerkin projection in nonlinear model reduction. *arXiv e-print*, (1504.03749), 2015.

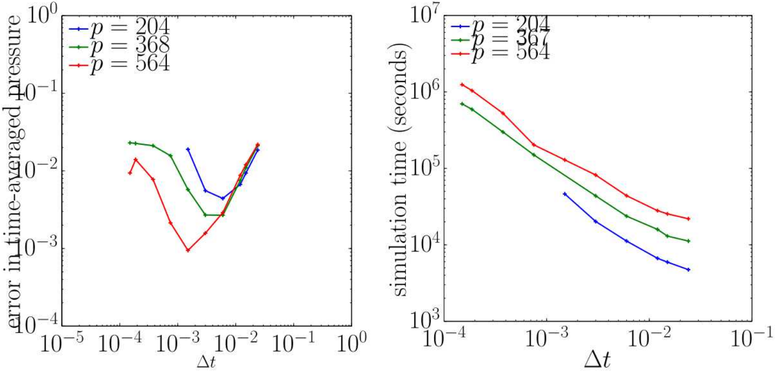
# Our research goal

Nonlinear model-reduction methods that are
accurate, low cost, certified, and reliable.

- Accuracy
  - Improve projection technique
  - Preserve problem structure
- Low cost
  - Sample-mesh approach [C. et al., 2011b, C. et al., 2013]
  - Leverage time-domain data
- Certification
  - Error bounds
  - Statistical error modeling
- Reliability
  - *A posteriori* $h$-refinement

**Collaborators**: C. Farhat, J. Cortial (Stanford)

# LSPG performance ($t \leq 2.5$ sec)



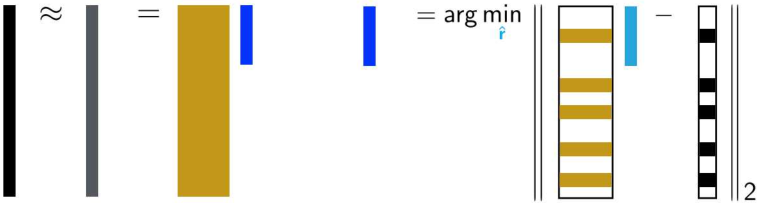

+ Always sub-3% errors
- More expensive than the FOM
  - FOM simulation: 1 hour, 48 CPU
  - LSPG ROM simulation (fastest): 1.3 hours, 48 CPU

# Hyper-reduction via Gappy POD [Everson and Sirovich, 1995]

$$\hat{\boldsymbol{x}}^n = \arg\min_{\hat{\boldsymbol{z}}\in\mathbb{R}^p} \|\boldsymbol{A}\boldsymbol{r}^n(\boldsymbol{\Phi}\hat{\boldsymbol{z}})\|_2^2.$$

*Can we select* $\boldsymbol{A}$ *to make this inexpensive?*

1. $\boldsymbol{r}^n(\boldsymbol{x}) \approx \tilde{\boldsymbol{r}}^n(\boldsymbol{x}) = \boldsymbol{\Phi}_R\hat{\boldsymbol{r}}^n(\boldsymbol{x})$
2. $\hat{\boldsymbol{r}}^n(\boldsymbol{x}) = \arg\min_{\hat{\boldsymbol{r}}} \|\boldsymbol{P}\boldsymbol{\Phi}_R\hat{\boldsymbol{r}} - \boldsymbol{P}\boldsymbol{r}^n(\boldsymbol{x})\|_2$




$$\hat{\boldsymbol{x}}^n = \arg\min_{\hat{\boldsymbol{z}}\in\mathbb{R}^p} \|\tilde{\boldsymbol{r}}^n(\boldsymbol{\Phi}\hat{\boldsymbol{z}})\|_2^2 = \arg\min_{\hat{\boldsymbol{z}}\in\mathbb{R}^p} \|\boldsymbol{\Phi}_R\hat{\boldsymbol{r}}^n(\boldsymbol{\Phi}\hat{\boldsymbol{z}})\|_2^2 = \arg\min_{\hat{\boldsymbol{z}}\in\mathbb{R}^p} \|\hat{\boldsymbol{r}}^n(\boldsymbol{\Phi}\hat{\boldsymbol{z}})\|_2^2$$

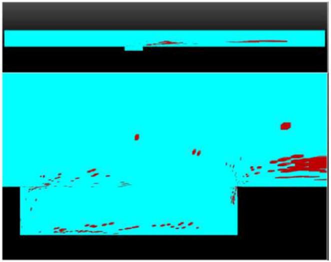
$$= \arg\min_{\hat{\boldsymbol{z}}\in\mathbb{R}^p} \|\underbrace{(\boldsymbol{P}\boldsymbol{\Phi}_R)^+\boldsymbol{P}}_{\boldsymbol{A}}\boldsymbol{r}^n(\boldsymbol{\Phi}\hat{\boldsymbol{z}})\|_2^2.$$

- \+ GNAT: $\boldsymbol{A} = (\boldsymbol{P}\boldsymbol{\Phi}_R)^+\boldsymbol{P}$ leads to low-cost
  - *Offline*: Construct $\boldsymbol{\Phi}_R$ (POD) and $\boldsymbol{P}$ (greedy method)

# Sample mesh: HPC implementation

$$\hat{\boldsymbol{x}}^n = \arg\min_{\hat{\boldsymbol{z}}\in\mathbb{R}^p} \| (\boldsymbol{P\Phi}_R)^+ \boldsymbol{P}\boldsymbol{r}^n(\boldsymbol{\Phi}\hat{\boldsymbol{z}}) \|_2^2$$

- *Key*: GNAT samples only a few entries of the residual $\boldsymbol{P}\boldsymbol{r}^n$
- *Idea*: Extract minimal subset of the mesh
- *Related*: subgrid [Haasdonk et al., 2008], reduced integration domain [Ryckelynck, 2005]



- Sample mesh: 4.1% nodes, 3.0% cells

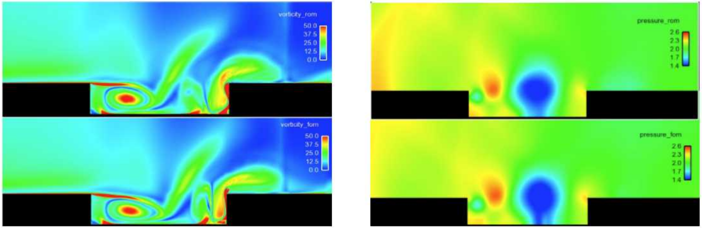
+ Small problem size: can run on many fewer cores

# GNAT performance ($t \leq 12.5$ sec)

vorticity field | pressure field

GNAT ROM

FOM




+ $< 1\%$ error in time-averaged drag
+ 229x CPU-hour savings
  - FOM: 5 hour x 48 CPU
  - GNAT ROM: 32 min x 2 CPU

# Our research goal

Nonlinear model-reduction methods that are
accurate, low cost, certified, and reliable.

+ Accuracy
  - Improve projection technique
  - Preserve problem structure
+ Low cost
  - Sample-mesh approach
  - Leverage time-domain data [C. et al., 2015b]
+ Certification
  - Error bounds
  - Statistical error modeling
+ Reliability
  - *A posteriori* $h$-refinement

**Collaborators**: L. Brencher, B. Haasdonk, A. Barth (U Stuttgart)

# Our research goal

Nonlinear model-reduction methods that are
accurate, low cost, certified, and reliable.

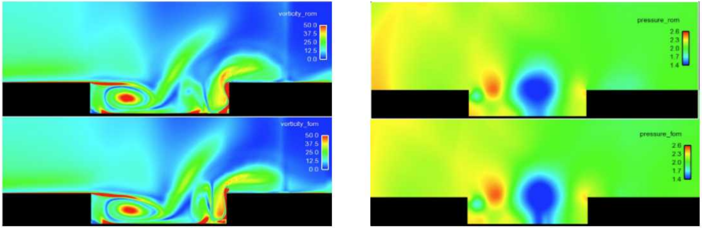
- Accuracy
  - Improve projection technique
  - Preserve problem structure
- Low cost
  - Sample-mesh approach
  - Leverage time-domain data
- Certification
  - Error bounds
  - Statistical error modeling
- Reliability
  - *A posteriori* $h$-refinement [C., 2015]

# GNAT performance ($t \leq 12.5$ sec)

vorticity field | pressure field

GNAT ROM

FOM




+ $< 1\%$ error in time-averaged drag
+ 229x CPU-hour savings
    - FOM: 5 hour x 48 CPU
    - GNAT ROM: 32 min x 2 CPU
- However, ROMs are not guaranteed to be accurate.

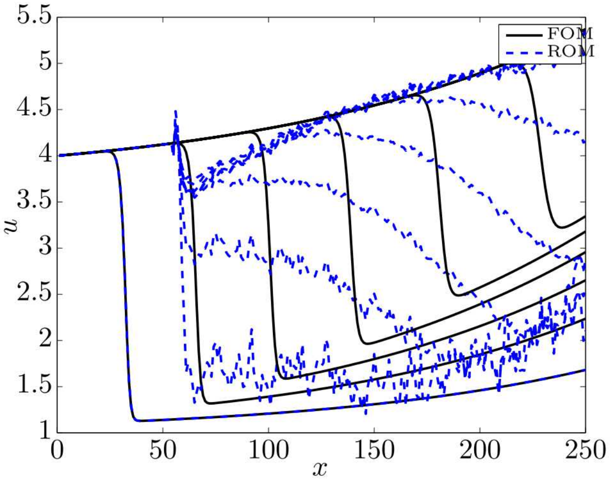
*ROM accuracy is limited by the information in* $\boldsymbol{\Phi}$.

# Example: inviscid Burgers equation [Rewienski, 2003]

$$\frac{\partial u(x,\tau)}{\partial \tau} + \frac{1}{2}\frac{\partial \left(u^2(x,\tau)\right)}{\partial x} = 0.02e^{0.02x}$$
$$u(0,\tau) = 3, \ \forall \tau > 0$$
$$u(x,0) = 1, \ \forall x \in [0, \ 100],$$

- Discretization: Godunov's scheme
- Simulate $\tau \in [0, 50]$
- FOM: 250 degrees of freedom
- ROM: 150 degrees of freedom
- $\mathbf{\Phi}$ constructed via POD using snapshots in $\tau_{\mathrm{train}} \in [0, 2.5]$

# ROM accuracy limited by relevance of training data




- ROM inaccurate when outside predictive domain of $\boldsymbol{\Phi}$

# Existing ROM adaptation methods

- *A priori* adaptation: unique ROM for separate regions of the
  - input space [Amsallem and Farhat, 2008, Amsallem et al., 2009, Eftang et al., 2010, Eftang et al., 2011, Haasdonk et al., 2011, Drohmann et al., 2011, Peherstorfer et al., 2014]
  - time domain [Drohmann et al., 2011, Dihlmann et al., 2011]
  - state space [Amsallem et al., 2012, Washabaugh et al., 2012, Peherstorfer et al., 2014].

  \+ Reduces the dimension of the ROM

  \- No mechanism to improve the ROM *a posteriori*
- *A posteriori* adaptation
  - Revert to the FOM, solve it, and add solution to the basis [Eldred et al., 2009, Arian et al., 2000, Ryckelynck, 2005]

  \+ Improves the ROM *a posteriori*

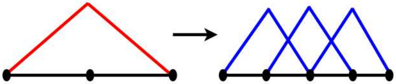
  \- Incurs large-scale operations

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

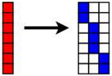
# Main idea

*ROM analog to mesh-adaptive h-refinement*

- 'Split' basis vectors

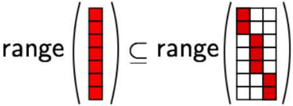


*finite element h-refinement*

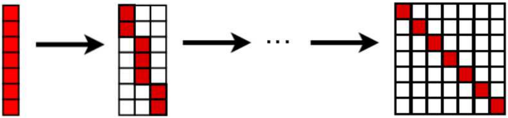


*ROM h-refinement*

- Generate hierarchical subspaces

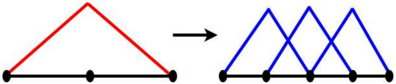



- ROM converges to the FOM

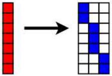


# *h*-refinement ingredients

1. *Adaptive algorithm*
2. *Refinement*

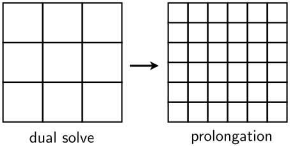


*finite element h-refinement*

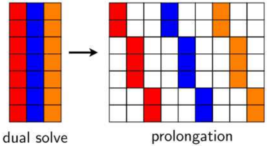


*ROM h-refinement*

3. *Error indicators*



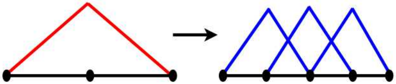

*finite element h-refinement*



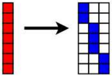

*ROM h-refinement*

# Ingredient 1: Adaptive algorithm

1. *Adaptive algorithm*
2. *Refinement*

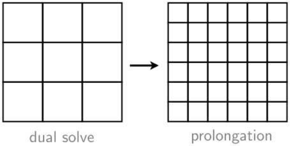


*finite element h-refinement*

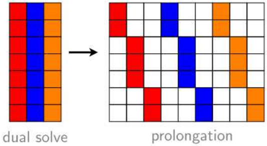


*ROM h-refinement*

3. *Error indicators*




*finite element h-refinement*




*ROM h-refinement*

# Adaptive algorithm

**Algorithm 1** Outer loop

**Input:** timestep $n$, current basis $\boldsymbol{\Phi}$
**Output:** updated basis $\boldsymbol{\Phi}$, generalized state $\hat{\boldsymbol{x}}^n$

1: Solve $\boldsymbol{\Phi}^T \boldsymbol{r}^n(\boldsymbol{\Phi}\hat{\boldsymbol{x}}^n; \boldsymbol{\mu}) = 0$ for current ROM solution $\hat{\boldsymbol{x}}^n$.
2: **if** estimate of output error $\delta_s$ is 'too large' **then**
3: Refine basis: $\boldsymbol{\Phi} \leftarrow \text{Refine}\left(\boldsymbol{\Phi}, \hat{\boldsymbol{x}}^n\right)$.
4: Return to Step 1.
5: **end if**
6: **if** $\mod\left(n, n_{\text{reset}}\right) = 0$ **then**
7: Reset basis: $\boldsymbol{\Phi} \leftarrow \boldsymbol{\Phi}^{(0)}$.
8: **end if**

# Adaptive algorithm

**Algorithm 2** Refine

**Input:** initial basis $\boldsymbol{\Phi}$, reduced solution $\hat{\boldsymbol{x}}$

**Output:** refined basis $\boldsymbol{\Phi}$

1: Compute prolongation operator $\boldsymbol{I}_H^h$ and fine basis $\boldsymbol{\Phi}^h$ (Ingredient 2)

2: Solve: Compute coarse adjoint solution (Ingredient 3)

3: Estimate: Compute fine error indicators (Ingredient 3)

4: Mark: Identify basis vectors to refine $I$

5: **for** $i \in I$ **do**

6:     Refine: Split $\phi_i$ into child vectors

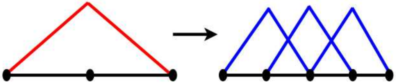
7: **end for**

8: Compute QR factorization with column pivoting $\boldsymbol{\Phi} = \boldsymbol{Q}\boldsymbol{R}$, $\boldsymbol{R}\bar{\boldsymbol{\Pi}} = \bar{\boldsymbol{Q}}\bar{\boldsymbol{R}}$

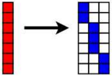
9: Ensure full-rank matrix $\boldsymbol{\Phi} \leftarrow \boldsymbol{\Phi}\left[\bar{\boldsymbol{\pi}}_1 \ \cdots \ \bar{\boldsymbol{\pi}}_r\right]$

# Ingredient 2: Refinement

1. *Adaptive algorithm*
2. *Refinement*

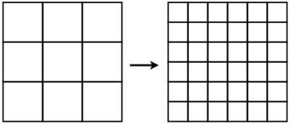


*finite element h-refinement*



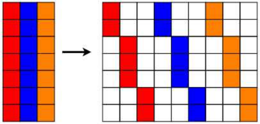
*ROM h-refinement*

3. *Error indicators*



dual solve prolongation

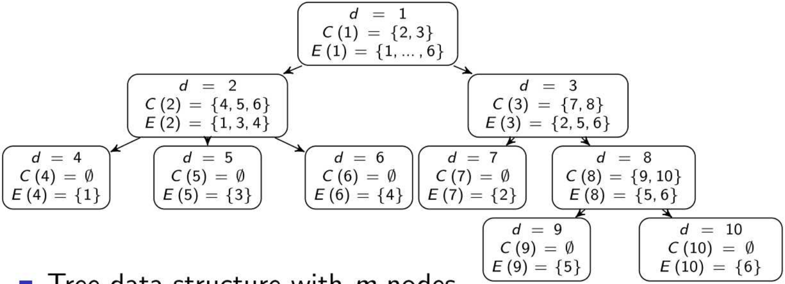
*finite element h-refinement*



dual solve prolongation

*ROM h-refinement*

# Tree data structure

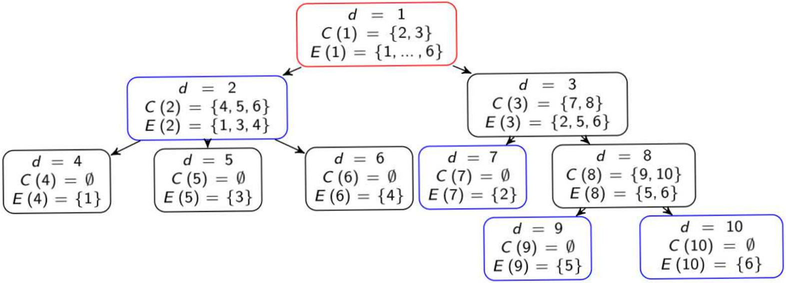



- Tree data structure with $m$ nodes
  - *child* function $C : \mathbb{N}(m) \to \mathcal{P}(\mathbb{N}(m))$
  - *element* function $E : \mathbb{N}(m) \to \mathcal{P}(\mathbb{N}(N))$
- Requirements
  1. Root node includes all elements $E(1)$
  2. Each element has a single leaf node
  3. Disjoint support of children $E(j) \cap E(k) = \emptyset, \ \forall j \neq k \in C(i)$
  4. $\cup_{j \in C(i)} E(j) = E(i)$

  \+ 1–2 ensure the ROM converges to the FOM

  \+ 4 ensures hierarchical refined subspaces

# Tree example: $N = 6$

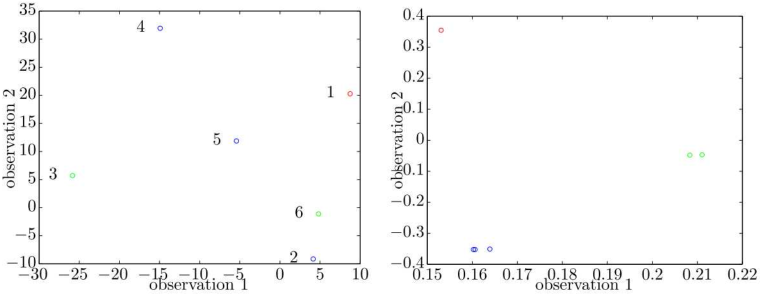



$$\boldsymbol{\Phi}^{(0)} = \begin{bmatrix} \phi_{11}^{(0)} \\ \phi_{21}^{(0)} \\ \phi_{31}^{(0)} \\ \phi_{41}^{(0)} \\ \phi_{51}^{(0)} \\ \phi_{61}^{(0)} \end{bmatrix} \rightarrow \quad \boldsymbol{\Phi} = \begin{bmatrix} \phi_{11}^{(0)} & 0 & 0 & 0 \\ 0 & \phi_{21}^{(0)} & 0 & 0 \\ \phi_{31}^{(0)} & 0 & 0 & 0 \\ \phi_{41}^{(0)} & 0 & 0 & 0 \\ 0 & 0 & \phi_{51}^{(0)} & 0 \\ 0 & 0 & 0 & \phi_{61}^{(0)} \end{bmatrix}.$$

# Tree construction

*State variables that are strongly* **correlated** *or* **anticorrelated** *should reside in the same tree node.*

1. Normalize state-variable observation history
2. If first observation is negative, flip over origin
3. Recursively apply **k-means clustering**

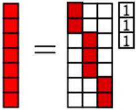



(j) before modification (k) after modification

State-variable observation history (variable index labeled).

# Refinement machinery

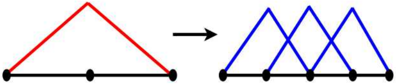
$$\mathbf{\Phi}^H = \mathbf{\Phi}^h \boldsymbol{I}_H^h$$



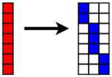

- coarse basis $\mathbf{\Phi}^H \in \mathbb{R}^{N \times p}$
- fine basis $\mathbf{\Phi}^h \in \mathbb{R}^{N \times q}$ with $q = \sum_{i=1}^{p} |C(d_i)|$
- prolongation operator $\boldsymbol{I}_H^h \in \{0,1\}^{q \times p}$
- prolongated generalized coordinates $\hat{\boldsymbol{x}}_H^h = \boldsymbol{I}_H^h \hat{\boldsymbol{x}}^H$
- restriction operator $\boldsymbol{I}_h^H = \left(\boldsymbol{I}_H^h\right)^+$

# Ingredient 3: Error indicators

1. *Adaptive algorithm*
2. *Refinement*

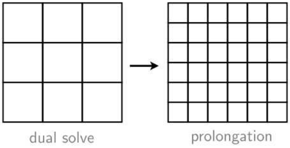


*finite element h-refinement*

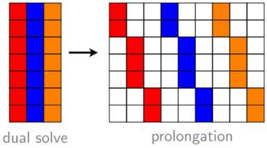


*ROM h-refinement*

3. *Error indicators*: a) dual solve (coarse), b) prolongation (fine)




*finite element h-refinement*




*ROM h-refinement*

# Dual-weighted residual error indicators

- **Goal-oriented**: reduce the error in output $g(\boldsymbol{x})$
- Analogous to duality-based error control for
  - differential equations [Estep, 1995, Pierce and Giles, 2000]
  - finite elements [Babuška and Miller, 1984, Becker and Rannacher, 1996, Rannacher, 1999, Bangerth and Rannacher, 1999, Becker and Rannacher, 2001, Bangerth and Rannacher, 2003],
  - finite volumes [Venditti and Darmofal, 2000, Venditti and Darmofal, 2002, Park, 2004, Nemec and Aftosmis, 2007]
  - discontinuous Galerkin methods [Lu, 2005, Fidkowski, 2007]

# Dual-weighted residual error indicators

- Approximate fine output:

$$g(\boldsymbol{\Phi}^h \hat{\boldsymbol{x}}^h) \approx g(\boldsymbol{\Phi}^H \hat{\boldsymbol{x}}^H) + \frac{\partial g}{\partial \boldsymbol{x}}(\boldsymbol{\Phi}^H \hat{\boldsymbol{x}}^H)\boldsymbol{\Phi}^h(\hat{\boldsymbol{x}}^h - \boldsymbol{I}_H^h \hat{\boldsymbol{x}}^H) \qquad (1)$$

- Approximate the fine residual:

$$\boldsymbol{0} = (\boldsymbol{\Phi}^h)^T \boldsymbol{r}(\boldsymbol{\Phi}^h \hat{\boldsymbol{x}}^h) \approx (\boldsymbol{\Phi}^h)^T \boldsymbol{r}(\boldsymbol{\Phi}^H \hat{\boldsymbol{x}}^H) + (\boldsymbol{\Phi}^h)^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{x}}(\boldsymbol{\Phi}^H \hat{\boldsymbol{x}}^H)\boldsymbol{\Phi}^h(\hat{\boldsymbol{x}}^h - \boldsymbol{I}_H^h \hat{\boldsymbol{x}}^H)$$

- Solve for the error:

$$(\hat{\boldsymbol{x}}^h - \boldsymbol{I}_H^h \hat{\boldsymbol{x}}^H) \approx -[(\boldsymbol{\Phi}^h)^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{x}}(\boldsymbol{\Phi}^H \hat{\boldsymbol{x}}^H)\boldsymbol{\Phi}^h]^{-1} (\boldsymbol{\Phi}^h)^T \boldsymbol{r}(\boldsymbol{\Phi}^H \hat{\boldsymbol{x}}^H) \qquad (2)$$

- Substitute (2) in (1):

$$\boxed{g(\boldsymbol{\Phi}^h \hat{\boldsymbol{x}}^h) - g(\boldsymbol{\Phi}^H \hat{\boldsymbol{x}}^H) \approx -(\hat{\boldsymbol{y}}^h)^T (\boldsymbol{\Phi}^h)^T \boldsymbol{r}(\boldsymbol{\Phi}^H \hat{\boldsymbol{x}}^H)},$$

with the fine adjoint solution $\hat{\boldsymbol{y}}^h \in \mathbb{R}^q$ satisfying

$$\boxed{(\boldsymbol{\Phi}^h)^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{x}}(\boldsymbol{\Phi}^H \hat{\boldsymbol{x}}^H)^T \boldsymbol{\Phi}^h \hat{\boldsymbol{y}}^h = (\boldsymbol{\Phi}^h)^T \frac{\partial g}{\partial \boldsymbol{x}}(\boldsymbol{\Phi}^H \hat{\boldsymbol{x}}^H)^T}.$$

# Dual-weighted residual error indicators

$$\boxed{g(\boldsymbol{\Phi}^h\hat{\mathbf{x}}^h) - g(\boldsymbol{\Phi}^H\hat{\mathbf{x}}^H) \approx -\left(\hat{\mathbf{y}}^h\right)^T (\boldsymbol{\Phi}^h)^T \mathbf{r}(\boldsymbol{\Phi}^H\hat{\mathbf{x}}^H)} \tag{3}$$

$$\boxed{(\boldsymbol{\Phi}^h)^T \frac{\partial \mathbf{r}}{\partial \mathbf{x}}(\boldsymbol{\Phi}^H\hat{\mathbf{x}}^H)^T \boldsymbol{\Phi}^h \hat{\mathbf{y}}^h = (\boldsymbol{\Phi}^h)^T \frac{\partial g}{\partial \mathbf{x}}(\boldsymbol{\Phi}^H\hat{\mathbf{x}}^H)^T} \tag{4}$$

- We want to avoid fine solve (4), so approximate $\hat{\mathbf{y}}^h$ as

$$\hat{\mathbf{y}}_H^h = \mathbf{I}_H^h \hat{\mathbf{y}}_H,$$

  where $\hat{\mathbf{y}}_H$ is the coarse adjoint solution to

$$(\boldsymbol{\Phi}^H)^T \frac{\partial \mathbf{r}}{\partial \mathbf{x}}(\boldsymbol{\Phi}^H\hat{\mathbf{x}}^H)^T \boldsymbol{\Phi}^H \hat{\mathbf{y}}_H = (\boldsymbol{\Phi}^H)^T \frac{\partial g}{\partial \mathbf{x}}(\boldsymbol{\Phi}^H\hat{\mathbf{x}}^H)^T.$$

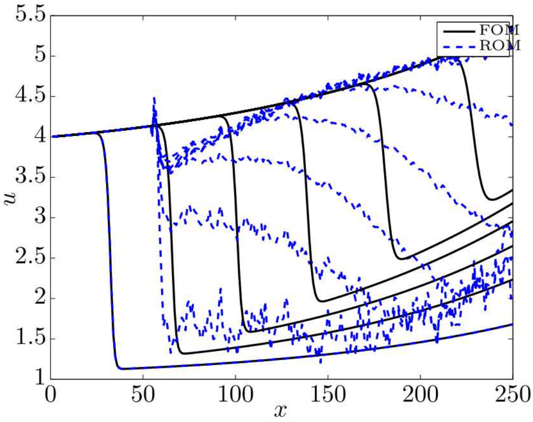
- Substituting $\hat{\mathbf{y}}_H^h$ for $\hat{\mathbf{y}}^h$ in (3) yields cheaply computable

$$g(\boldsymbol{\Phi}^h\hat{\mathbf{x}}^h) - g(\boldsymbol{\Phi}^H\hat{\mathbf{x}}^H) \approx -(\hat{\mathbf{y}}_H^h)^T (\boldsymbol{\Phi}^h)^T \mathbf{r}(\boldsymbol{\Phi}^H\hat{\mathbf{x}}^H).$$

- The RHS can be bounded by cheaply computable error indicators

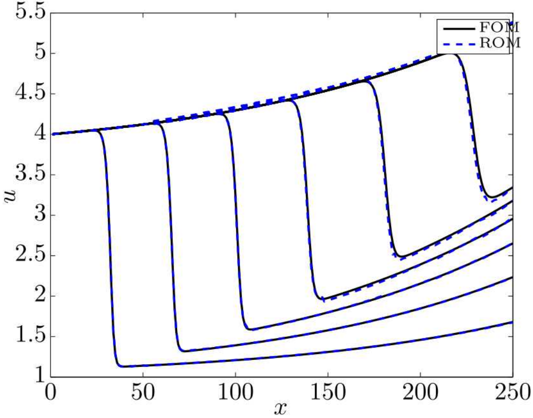
$$\left|\left(\hat{\mathbf{y}}_H^h\right)^T (\boldsymbol{\Phi}^h)^T \mathbf{r}(\boldsymbol{\Phi}^H\hat{\mathbf{x}}^H)\right| \leq \sum_{i=1}^{q} \delta_i^h, \qquad \boxed{\delta_i^h = \left|\left[\hat{y}_H^h\right]_i \left(\phi_i^h\right)^T \mathbf{r}\left(\boldsymbol{\Phi}^H\hat{\mathbf{x}}^H\right)\right|.}$$

# Previous example




- Generated by $\mathbf{\Phi} \in \mathbb{R}^{250\times150}$ using $\tau_{\text{train}} \in [0, 2.5]$
- Now try *h*-adaptivity with $\mathbf{\Phi}^{(0)} \in \mathbb{R}^{250\times10}$, $\tau_{\text{train}} \in [0, 2.5]$.

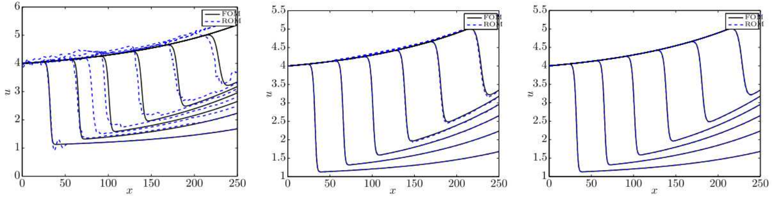
# Previous example with *h*-adaptivity




- $\dim \mathbf{\Phi}^{(0)} = 10$
- $\text{mean}_k(\dim \mathbf{\Phi}^{(k)}) = 44.3$

*h-adaptation allows the ROM to capture phenomena not present in the training data!*

# *h*-adaptivity enables error control




(a) tolerance $\epsilon = 0.35$ (b) tolerance $\epsilon = 0.05$ (c) tolerance $\epsilon = 0.01$

| | $\epsilon = 0.35$ | $\epsilon = 0.05$ | $\epsilon = 0.01$ |
|---|---|---|---|
| average basis dimension per Newton iteration | 33.6 | 44.2507 | 53.9 |
| relative error (%)<br>online time (seconds) | 12.2<br>4.61 | 0.51<br>4.63 | 0.078<br>7.64 |

# Summary: *a posteriori* h-refinement

- Adaptive $h$-refinement via splitting
  + Incrementally improves ROM
  + Does not require large-scale operations
  + Enables error control
  + Extends utility of ROMs to hyperbolic PDEs
- **Reference**: C., Adaptive $h$-refinement for reduced-order models. *International Journal for Numerical Methods in Engineering*, 102(5):1192–1210, 2015.

# Our research goal

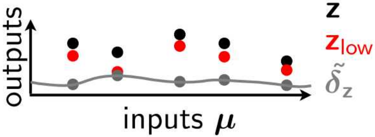
Nonlinear model-reduction methods that are
accurate, low cost, certified, and reliable.

+ Accuracy
  - Improve projection technique
  - Preserve problem structure
+ Low cost
  - Sample-mesh approach
  - Leverage time-domain data
+ Certification
  - Error bounds
  - Statistical error modeling [Drohmann and C., 2015]
+ Reliability
  - *A posteriori* *h*-refinement

**Collaborator**: M. Drohmann (Sandia)

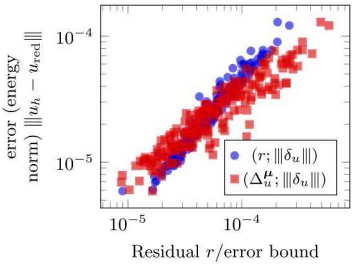
# Strategies for ROM error quantification

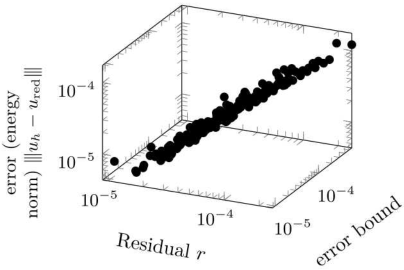
1. Rigorous error bounds
   - \+ independent of input-space dimension
   - \- high effectivity (overestimation), especially for nonlinear, time depedent problems
   - \- improving effectivity incurs high costs [Huynh et al., 2010, Wirtz et al., 2012] or intrusive reformulation of discretization [Yano et al., 2012]
   - \- not amenable to statistics
2. Multifidelity correction [Eldred et al., 2004]




   - model low-fidelity error as a function of inputs
   - 'correct' low-fidelity outputs with model
   - \+ amenable to statistics
   - \- curse of dimensionality
   - \- ROM errors highly oscillatory in the input space [Ng and Eldred, 2012]

# Data-driven observation

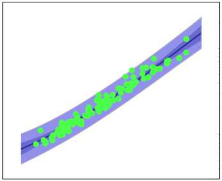



- ROMs generate error indicators that correlate with the error
- **Main idea**: map error indicators to a distribution over the true error using Gaussian process regression
  - \+ independent of input-space dimension

*Reduced-order model error surrogates*

# ROMES




- Approximate *deterministic* $\boldsymbol{\mu} \mapsto d(\delta)$ by *stochastic* $\boldsymbol{\rho}(\boldsymbol{\mu}) \mapsto \tilde{d}$
  - $d$: invertible *transformation function*
  - $\tilde{d}$: random variable for transformed error.
  - $\tilde{\delta} := d^{-1}(\tilde{d})$: random variable for the error
- ROMES ingredients:
  1. error indicators $\boldsymbol{\rho}$
  2. transformation function $d$
  3. statistical model: Gaussian process
- Desired conditions
  1. indicators $\boldsymbol{\rho}(\boldsymbol{\mu})$ are *low dimensional* and *cheaply computable*
  2. distribution of random variable $\tilde{\delta}$ has *low variance*
  3. statistical model is *validated*

# Normed errors: $\delta(\boldsymbol{\mu}) = \|\mathbf{x}(\boldsymbol{\mu}) - \boldsymbol{\Phi}\hat{\mathbf{x}}(\boldsymbol{\mu})\|$

- ROMs often equipped with bounds $\Delta(\boldsymbol{\mu}) \geq \delta(\boldsymbol{\mu}) \geq 0$
- Bound effectivity $\eta(\boldsymbol{\mu}) := \frac{\Delta(\boldsymbol{\mu})}{\delta(\boldsymbol{\mu})} \geq 1$ often lies in a small range:

$$\eta_1 \leq \eta(\boldsymbol{\mu}) \leq \eta_2, \quad \forall \boldsymbol{\mu}$$

$$\boxed{\log \Delta(\boldsymbol{\mu}) - \log \eta_1 \geq \log \delta(\boldsymbol{\mu}) \geq \log \Delta(\boldsymbol{\mu}) - \log \eta_2, \quad \forall \boldsymbol{\mu}}$$

- **Ingredient 1**: indicator $\boldsymbol{\rho} = \log \Delta$
  - Also consider cheaper $\boldsymbol{\rho} = \log \|\boldsymbol{r}(\boldsymbol{\Phi}\hat{\mathbf{x}}(\boldsymbol{\mu}); \boldsymbol{\mu})\|_2$ because

$$\Delta_{\mathbf{x}}^{\mu}(\boldsymbol{\mu}) := \frac{\|\boldsymbol{r}(\boldsymbol{\Phi}\hat{\mathbf{x}}(\boldsymbol{\mu}); \boldsymbol{\mu})\|_2}{\sqrt{\alpha_{\mathrm{LB}}(\boldsymbol{\mu})}} \geq \|\!|\mathbf{x}(\boldsymbol{\mu}) - \boldsymbol{\Phi}\hat{\mathbf{x}}(\boldsymbol{\mu})|\!\|$$

$$\Delta_{\mathbf{x}}(\boldsymbol{\mu}) := \frac{\|\boldsymbol{r}(\boldsymbol{\Phi}\hat{\mathbf{x}}(\boldsymbol{\mu}); \boldsymbol{\mu})\|_2}{\alpha_{\mathrm{LB}}(\boldsymbol{\mu})} \geq \|\mathbf{x}(\boldsymbol{\mu}) - \boldsymbol{\Phi}\hat{\mathbf{x}}(\boldsymbol{\mu})\|_{X_h}$$

and $\alpha_{\mathrm{LB}}(\boldsymbol{\mu})$ is costly to compute.

- **Ingredient 2**: transformation function $d = \log$

# General errors: $\delta(\mu) = g(\boldsymbol{x}(\mu)) - g(\boldsymbol{\Phi}\hat{\boldsymbol{x}}(\mu))$

- Recall from before that dual-weighted residuals lead to

$$g(\boldsymbol{x}) - g(\boldsymbol{\Phi}\hat{\boldsymbol{x}}) \approx -\boldsymbol{y}^T \boldsymbol{r}(\boldsymbol{\Phi}\hat{\boldsymbol{x}}).$$

with the adjoint solution satisfying

$$\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{x}}(\boldsymbol{\Phi}\hat{\boldsymbol{x}})^T \boldsymbol{y} = \frac{\partial g}{\partial \boldsymbol{x}}(\boldsymbol{\Phi}\hat{\boldsymbol{x}})$$

- To avoid this costly solve, we approximate it as $\boldsymbol{y} \approx \boldsymbol{Y}\hat{\boldsymbol{y}}$ with

$$\boldsymbol{Y}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{x}}(\boldsymbol{\Phi}\hat{\boldsymbol{x}})^T \boldsymbol{Y}\hat{\boldsymbol{y}} = \boldsymbol{Y}^T \frac{\partial g}{\partial \boldsymbol{x}}(\boldsymbol{\Phi}\hat{\boldsymbol{x}})$$

such that

$$\boxed{g(\boldsymbol{x}) - g(\boldsymbol{\Phi}\hat{\boldsymbol{x}}) \approx -\hat{\boldsymbol{y}}^T \boldsymbol{Y}^T \boldsymbol{r}(\boldsymbol{\Phi}\hat{\boldsymbol{x}})}.$$

- **Ingredient 1**: indicator $\rho = \hat{\boldsymbol{y}}^T \boldsymbol{Y}^T \boldsymbol{r}(\boldsymbol{\Phi}\hat{\boldsymbol{x}})$
  - $+$ Uncertainty control: can add columns to dual basis $\boldsymbol{Y}$
- **Ingredient 2**: transformation function $d = \mathrm{id}$

# **Ingredient 3**: Gaussian process [Rasmussen and Williams, 2006]

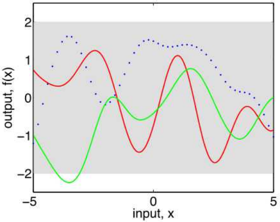
**Definition (Gaussian process)**

A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution.

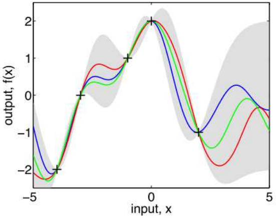
$$\tilde{d}(\rho) \sim \mathcal{GP}(m(\rho), k(\rho, \rho'))$$

- mean function $m(\rho)$; covariance function $k(\rho, \rho')$
- given a training set $\{(d(\delta_i), \rho_i)\}$, can infer $m(\rho)$ and $k(\rho, \rho')$
- Consider kernel regression [Rasmussen and Williams, 2006]

# Kernel regression [Rasmussen and Williams, 2006]




(d) prior




(e) posterior

- **prior**: $\tilde{d}(\underline{\rho}) \sim \mathcal{N}(0, K\left(\underline{\rho}, \underline{\rho}\right) + \sigma^2 \boldsymbol{I})$
  - $k(\rho_i, \rho_j) = \exp - \frac{\|\rho_i - \rho_j\|^2}{2r^2}$ is a positive definite kernel
  - $\underline{\rho} := \left[\underline{\rho}_{\text{train}} \ \underline{\rho}_{\text{predict}}\right]^T$
- **posterior**: $\tilde{d}(\rho_{\text{predict}}) \sim \mathcal{N}(m(\rho_{\text{predict}}), \mathrm{cov}(\rho_{\text{predict}}))$

# ROMES Algorithm

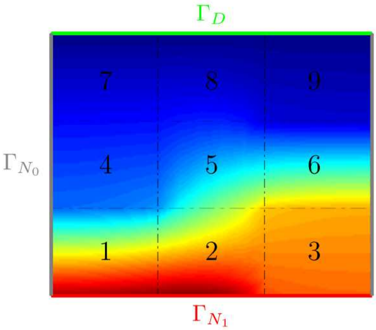
Offline

1. Populate ROMES database $\{(d(\delta(\boldsymbol{\mu})), \bar{\rho}(\boldsymbol{\mu})) \mid \boldsymbol{\mu} \in \mathcal{D}\}$, where $\bar{\rho}$ denotes candidate indicators.
2. Identify a few error indicators $\rho \subset \bar{\rho}$ that lead to a low-variance GP.
3. Construct the Gaussian process $\tilde{d}(\rho) \sim \mathcal{GP}(m(\rho), k(\rho, \rho'))$ by Bayesian inference.

Online (for any $\boldsymbol{\mu}^\star \in \mathcal{D}$)

1. compute the ROM solution
2. compute error indicators $\rho(\boldsymbol{\mu}^\star)$
3. obtain $\tilde{d}(\rho(\boldsymbol{\mu}^\star)) \sim \mathcal{N}(m(\rho(\boldsymbol{\mu}^\star))), k(\rho(\boldsymbol{\mu}^\star), \rho(\boldsymbol{\mu}^\star))$
4. obtain random variable for the error $\tilde{\delta}(\boldsymbol{\mu}^\star) = d^{-1}(\tilde{d}(\boldsymbol{\mu}^\star))$
5. correct the ROM solution

# Thermal block

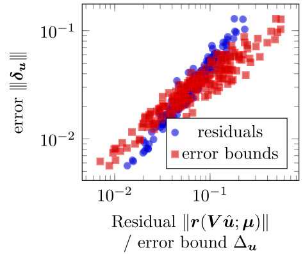


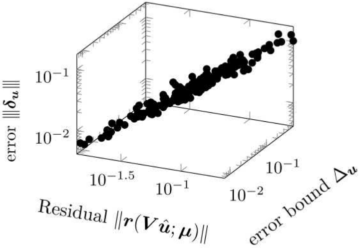

$$\triangle c(x;\boldsymbol{\mu})u(x;\boldsymbol{\mu}) = 0 \text{ in } \Omega \qquad \boldsymbol{x}(\boldsymbol{\mu}) = 0 \text{ on } \Gamma_D$$

$$\nabla c(\boldsymbol{\mu})\boldsymbol{x}(\boldsymbol{\mu}) \cdot n = 0 \text{ on } \Gamma_{N_0} \qquad \nabla c(\boldsymbol{\mu})\boldsymbol{x}(\boldsymbol{\mu}) \cdot n = 1 \text{ on } \Gamma_{N_1}$$

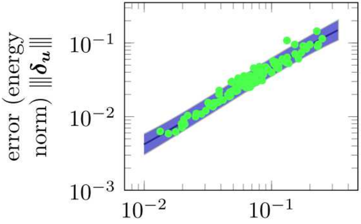
- Inputs $\boldsymbol{\mu} \in [0.1, 10]^9$ define diffusivity $c$ in subdomains
- ROM constructed via RB–Greedy [Patera and Rozza, 2006]

# Error #1: energy norm of error $\delta(\boldsymbol{\mu}) = |||\boldsymbol{x}(\boldsymbol{\mu}) - \hat{\boldsymbol{x}}(\boldsymbol{\mu})|||$



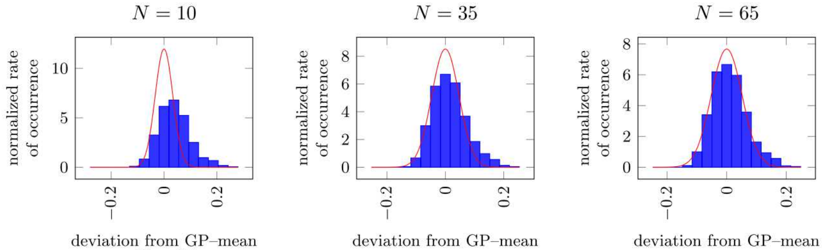



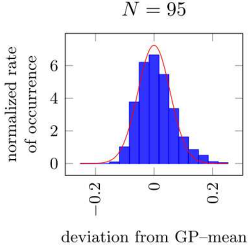

+ Residual norm and error bound correlate with error



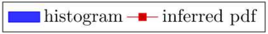

+ ROMES ($\rho = \log \|\boldsymbol{r}(\boldsymbol{\Phi}\hat{\boldsymbol{x}}(\boldsymbol{\mu});\boldsymbol{\mu})\|_2$, $d = \log$) promising

# Gaussian process validated

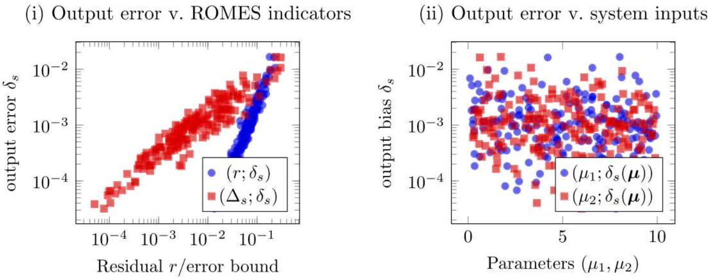



| | Validation frequency $\omega_{\text{validation}}(\omega)$ | | | |
|---|---|---|---|---|
| predicted $\omega$ | $N = 10$ | $N = 35$ | $N = 65$ | $N = 95$ |
| 0.80 | 0.49 | 0.71 | 0.76 | 0.78 |
| 0.90 | 0.59 | 0.82 | 0.87 | 0.88 |
| 0.95 | 0.68 | 0.89 | 0.92 | 0.93 |
| 0.98 | 0.76 | 0.93 | 0.95 | 0.96 |
| 0.99 | 0.80 | 0.94 | 0.96 | 0.97 |

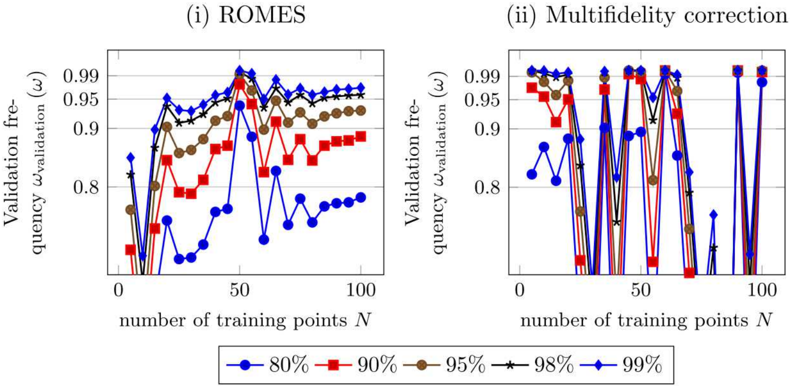


# Error #2: compliant-output error $\delta(\boldsymbol{\mu}) = |y(\boldsymbol{\mu}) - y_{\text{red}}(\boldsymbol{\mu})|$




+ ROMES: residual and error bound correlate with error
- Multifidelity correction: inputs are poor indicators

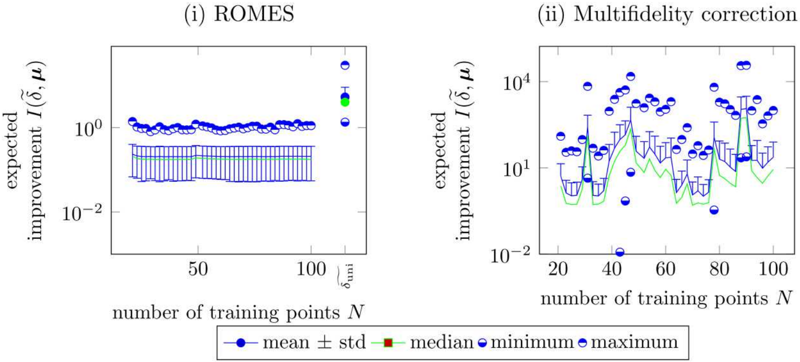
# Gaussian-process validation




+ ROMES: confidence intervals converge
- Multifidelity correction: confidence intervals do not converge

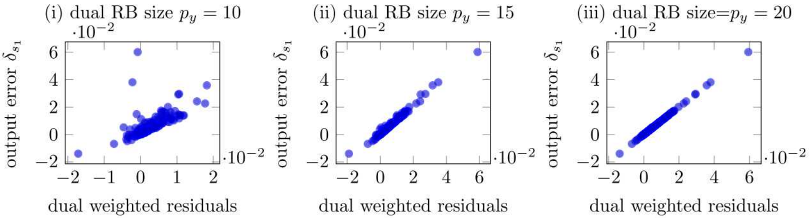
# Error reduction

$$\text{expected improvement} = \frac{|s(\boldsymbol{\mu}) - s_{\text{red}}(\boldsymbol{\mu}) - \text{mode}(\widetilde{\delta_s}(\boldsymbol{\mu}))|}{|s(\boldsymbol{\mu}) - s_{\text{red}}(\boldsymbol{\mu})|}$$

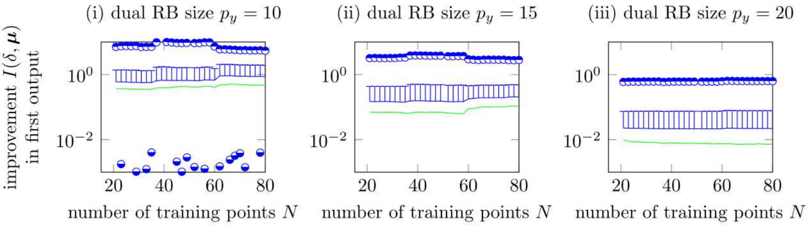



+ ROMES: reduces error by roughly an order of magnitude
- Multifidelity correction: often *increases* the error

# Error #3: error in general output




+ Dual-weighted residuals correlate with error
+ Uncertainty control: less variance as columns added to $\boldsymbol{Y}$

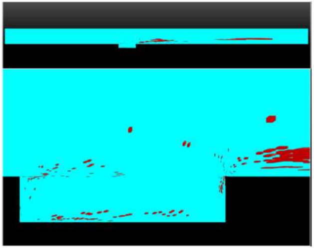


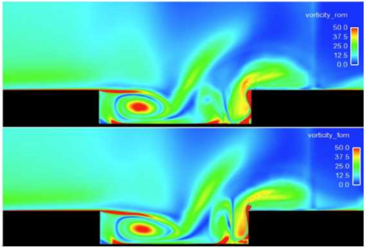

+ Uncertainty control: error reduces as columns added to $\boldsymbol{Y}$

# Summary: statistical error modeling

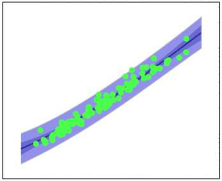
- ROMES
  - + Uses cheap error indicators to statistical quantify ROM error
  - + Outperforms multifidelity correction (inputs = poor indicators)
  - + Uncertainty control for general errors
- Related follow-on work
  - Reduction error models (REM) [Manzoni et al., 2016]
- Our current work
  - Apply to nonlinear, time-dependent problems
  - **Collaborators**: Trehan, Durlofsky (Stanford)
- **Reference**: Drohmann, C. The ROMES method for statistical modeling of reduced-order-model error. *SIAM/ASA Journal on Uncertainty Quantification*, 3(1):116–145, 2015.

# Questions?

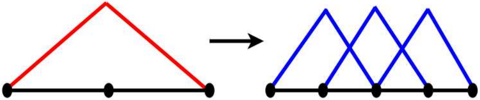


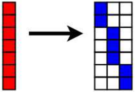



tranformed error
$d(\delta)$



error indicator $\boldsymbol{\rho}$





# Acknowledgments


- This research was supported in part by an appointment to the Sandia National Laboratories Truman Fellowship in National Security Science and Engineering, sponsored by Sandia Corporation (a wholly owned subsidiary of Lockheed Martin Corporation) as Operator of Sandia National Laboratories under its U.S. Department of Energy Contract No. DE-AC04-94AL85000.

Amsallem, D., Cortial, J., C., K., and Farhat, C. (2009).
A method for interpolating on manifolds structural dynamics reduced-order models.
International Journal for Numerical Methods in Engineering, 80(9):1241–1258.

Amsallem, D. and Farhat, C. (2008).
An interpolation method for adapting reduced-order models and application to aeroelasticity.
AIAA Journal, 46(7):1803–1813.

Amsallem, D., Zahr, M. J., and Farhat, C. (2012).
Nonlinear model order reduction based on local reduced-order bases.
International Journal for Numerical Methods in Engineering, 92(10):891–916.

Antoulas, A. C. (2005).
Approximation of Large-Scale Dynamical Systems.

Society for Industrial and Applied Mathematics, Philadelphia, PA.

Arian, E., Fahl, M., and Sachs, E. W. (2000).
Trust-region proper orthogonal decomposition for flow control.
Technical Report 25, ICASE.

Babuška, I. and Miller, A. (1984).
The post-processing approach in the finite element method—part 1: Calculation of displacements, stresses and other higher derivatives of the displacements.
International Journal for numerical methods in engineering, 20(6):1085–1109.

Bai, Z. (2002).
Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems.
Applied Numerical Mathematics, 43(1):9–44.

Bangerth, W. and Rannacher, R. (1999).

Finite element approximation of the acoustic wave equation: Error control and mesh adaptation.
East West Journal of Numerical Mathematics, 7(4):263–282.

Bangerth, W. and Rannacher, R. (2003).
Adaptive finite element methods for differential equations.
Springer.

Barrault, M., Maday, Y., Nguyen, N. C., and Patera, A. T. (2004).
An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations.
Comptes Rendus Mathématique Académie des Sciences, 339(9):667–672.

Baur, U., Beattie, C., Benner, P., and Gugercin, S. (2011).
Interpolatory projection methods for parameterized model reduction.
SIAM Journal on Scientific Computing, 33(5):2489–2518.

Becker, R. and Rannacher, R. (1996).

Weighted a posteriori error control in finite element methods, volume preprint no. 96-1.
Universitat Heidelberg.

Becker, R. and Rannacher, R. (2001).
An optimal control approach to a posteriori error estimation in finite element methods.
Acta Numerica 2001, 10:1–102.

Benner, P., Gugercin, S., and Willcox, K. (2015).
A survey of projection-based model reduction methods for parametric dynamical systems.
SIAM Review, 57(4):483–531.

Bui-Thanh, T., Willcox, K., and Ghattas, O. (2008).
Model reduction for large-scale systems with high-dimensional parametric input space.
SIAM Journal on Scientific Computing, 30(6):3270–3288.

C., K. (2015).
Adaptive *h*-refinement for reduced-order models.

International Journal for Numerical Methods in Engineering, 102(5):1192–1210.

C., K., Barone, M., and Antil, H. (2015a).
Galerkin v. discrete-optimal projection in nonlinear model reduction.
arXiv e-print, (1504.03749).

C., K., Bou-Mosleh, C., and Farhat, C. (2011a).
Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations.
International Journal for Numerical Methods in Engineering, 86(2):155–181.

C., K., Cortial, J., Amsallem, D., Zahr, M., and Farhat, C. (2011b).
The GNAT nonlinear model reduction method and its application to fluid dynamics problems.

AIAA Paper 2011-3112, 6th AIAA Theoretical Fluid Mechanics Conference, Honolulu, HI.

C., K., Farhat, C., Cortial, J., and Amsallem, D. (2013).
The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows.
Journal of Computational Physics, 242:623–647.

C., K., Ray, J., and van Bloemen Waanders, B. (2015b).
Decreasing the temporal complexity for nonlinear, implicit reduced-order models by forecasting.
Computer Methods in Applied Mechanics and Engineering, 289:79–103.

Dihlmann, M., Drohmann, M., and Haasdonk, B. (2011).
Model reduction of parametrized evolution problems using the reduced basis method with adaptive time partitioning.

Drohmann, M. and C., K. (2015).

The romes method for reduced-order-model uncertainty quantification.
SIAM/ASA Journal on Uncertainty Quantification, 3(1):116–145.

Drohmann, M., Haasdonk, B., and Ohlberger, M. (2011).
Adaptive reduced basis methods for nonlinear convection–diffusion equations.
In Finite Volumes for Complex Applications VI Problems & Perspectives, pages 369–377. Springer.

Eftang, J. L., Knezevic, D. J., and Patera, A. T. (2011).
An hp certified reduced basis method for parametrized parabolic partial differential equations.
Mathematical and Computer Modelling of Dynamical Systems, 17(4):395–422.

Eftang, J. L. and Patera, A. T. (2013).
Port reduction in parametrized component static condensation: approximation and a posteriori error estimation.

International Journal for Numerical Methods in Engineering, 96(5):269–302.

Eftang, J. L., Patera, A. T., and Rønquist, E. M. (2010).
An 'hp' certified reduced basis method for parametrized elliptic partial differential equations.
SIAM Journal on Scientific Computing, 32(6):3170–3200.

Eldred, M. S., Giunta, A. A., Collis, S. S., Alexandrov, N. A., and Lewis, R. M. (2004).
Second-order corrections for surrogate-based optimization with model hierarchies.
In Proceedings of the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, NY, number AIAA Paper 2004-4457.

Eldred, M. S., Weickum, G., and Maute, K. (2009).
A multi-point reduced-order modeling approach of transient structural dynamics with application to robust design optimization.

Structural and Multidisciplinary Optimization, 38(6):599–611.

Estep, D. (1995).
A posteriori error bounds and global error control for approximation of ordinary differential equations.
SIAM Journal on Numerical Analysis, 32(1):1–48.

Everson, R. and Sirovich, L. (1995).
Karhunen–Loève procedure for gappy data.
Journal of the Optical Society of America A, 12(8):1657–1664.

Farhat, C., Geuzaine, P., and Brown, G. (2003).
Application of a three-field nonlinear fluid-structure formulation to the prediction of the aeroelastic parameters of an F-16 fighter.
Computers & Fluids, 32(1):3–29.

Fidkowski, K. J. (2007).
A simplex cut-cell adaptive method for high-order discretizations of the compressible Navier-Stokes equations.
PhD thesis, Massachusetts Institute of Technology.

Freund, R. (2003).
Model reduction methods based on Krylov subspaces.
Acta Numerica, 12:267–319.

Gallivan, K., Vandendorpe, A., and Van Dooren, P. (2004).
Model reduction of mimo systems via tangential interpolation.
SIAM Journal on Matrix Analysis and Applications, 26(2):328–349.

Haasdonk, B., Dihlmann, M., and Ohlberger, M. (2011).
A training set and multiple bases generation approach for parameterized model reduction based on adaptive grids in parameter space.
Mathematical and Computer Modelling of Dynamical Systems, 17(4):423–442.

Haasdonk, B., Ohlberger, M., and Rozza, G. (2008).
A reduced basis method for evolution schemes with parameter-dependent explicit operators.
Electronic Transactions on Numerical Analysis, 32:145–161.

Huynh, D. B. P., Knezevic, D. J., Chen, Y., Hesthaven, J. S., and Patera, A. T. (2010).
A natural-norm successive constraint method for inf-sup lower bounds.
Comput. Methods Appl. Mech. Engrg., 199:1963–1975.

Ionita, A. and Antoulas, A. (2014).
Data-driven parametrized model reduction in the loewner framework.
SIAM Journal on Scientific Computing, 36(3):A984–A1007.

Lefteriu, S. and Antoulas, A. C. (2010).
A new approach to modeling multiport systems from frequency-domain data.
Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, 29(1):14–27.

LeGresley, P. A. (2006).
Application of Proper Orthogonal Decomposition (POD) to Design Decomposition Methods.

PhD thesis, Stanford University.

Lu, J. C.-C. (2005).
An a posteriori error control framework for adaptive precision optimization using discontinuous Galerkin finite element method.
PhD thesis, Massachusetts Institute of Technology.

Maday, Y. and Rønquist, E. M. (2002).
A reduced-basis element method.
J. Sci Comput, 17(1-4):447–459.

Manzoni, A., Pagani, S., and Lassila, T. (2016).
Accurate solution of bayesian inverse uncertainty quantification problems using model and error reduction methods.

Moore, B. (1981).
Principal component analysis in linear systems: Controllability, observability, and model reduction.
Automatic Control, IEEE Transactions on, 26(1):17–32.

Nemec, M. and Aftosmis, M. (2007).

Adjoint error estimation and adaptive refinement for embedded-boundary cartesian meshes.
In 18th AIAA CFD Conference, Miami. Paper AIAA-2007-4187.

Ng, L. and Eldred, M. S. (2012).
Multifidelity uncertainty quantification using non-intrusive polynomial chaos and stochastic collocation.
In AIAA 2012-1852.

Park, M. A. (2004).
Adjoint-based, three-dimensional error prediction and grid adaptation.
AIAA journal, 42(9):1854–1862.

Patera, A. T. and Rozza, G. (2006).
Reduced basis approximation and a posteriori error estimation for parametrized partial differential equations.
MIT.

Peherstorfer, B., Butnaru, D., Willcox, K., and Bungartz, H.-J. (2014).
Localized discrete empirical interpolation method.
SIAM Journal on Scientific Computing, 36(1):A168—A192.

Phuong Huynh, D. B., Knezevic, D. J., and Patera, A. T. (2013).
A static condensation reduced basis element method: approximation and a posteriori error estimation.
ESAIM: Mathematical Modelling and Numerical Analysis, 47(01):213–251.

Pierce, N. A. and Giles, M. B. (2000).
Adjoint recovery of superconvergent functionals from pde approximations.
SIAM review, 42(2):247–264.

Prud'Homme, C., Rovas, D. V., Veroy, K., Machiels, L., Maday, Y., Patera, A. T., Turinici, G., et al. (2001).

Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bound methods.
Journal of Fluids Engineering, 124(1):70–80.

Rannacher, R. (1999).
The dual-weighted-residual method for error control and mesh adaptation in finite element methods.
MAFELEAP, 99:97–115.

Rasmussen, C. and Williams, C. (2006).
Gaussian Processes for Machine Learning.
MIT Press.

Rewienski, M. J. (2003).
A Trajectory Piecewise-Linear Approach to Model Order Reduction o

PhD thesis, Massachusetts Institute of Technology.

Rowley, C. W. (2005).
Model reduction for fluids, using balanced proper orthogonal decomposition.

Int. J. on Bifurcation and Chaos, 15(3):997–1013.

Rozza, G., Huynh, D., and Patera, A. T. (2008).
Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations.
Archives of Computational Methods in Engineering, 15(3):229–275.

Ryckelynck, D. (2005).
A priori hyperreduction method: an adaptive approach.
Journal of Computational Physics, 202(1):346–366.

Venditti, D. and Darmofal, D. (2000).
Adjoint error estimation and grid adaptation for functional outputs: Application to quasi-one-dimensional flow.
Journal of Computational Physics, 164(1):204–227.

Venditti, D. A. and Darmofal, D. L. (2002).
Grid adaptation for functional outputs: application to two-dimensional inviscid flows.

Journal of Computational Physics, 176(1):40–69.

Veroy, K., Prud'homme, C., Rovas, D. V., and Patera, A. T. (2003).
A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations.
AIAA Paper 2003-3847, 16th AIAA Computational Fluid Dynamics Conference, Orlando, FL.

Washabaugh, K., Amsallem, D., Zahr, M., and Farhat, C. (2012).
Nonlinear model reduction for cfd problems using local reduced-order bases.
In 42nd AIAA Fluid Dynamics Conference and Exhibit, Fluid Dynamics and Co-located Conferences, AIAA Paper, volume 2686.

Willcox, K. and Peraire, J. (2002).
Balanced model reduction via the proper orthogonal decomposition.

AIAA Journal, 40(11):2323–2330.

Wirtz, D., Sorensen, D. C., and Haasdonk, B. (2012).
A-posteriori error estimation for DEIM reduced nonlinear dynamical systems.
Preprint Series, Stuttgart Research Centre for Simulation Technology.

Yano, M., Patera, A. T., and Urban, K. (2012).
A space-time certified reduced-basis method for Burgers' equation.
Math. Mod. Meth. Appl. S., submitted.