SAND2016-2345C

Nonlinear model reduction: discrete optimality, *h*-adaptivity, and error surrogates

Kevin Carlberg

Sandia National Laboratories

Workshop on Data-Driven Model Order Reduction and Machine Learning
April 1, 2016

Collaborators

2014–2015 M. Barone
Sandia National Laboratories

2014–2015 H. Antil
George Mason University

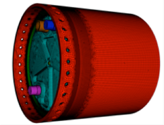
2015 L. Brencher, B. Haasdonk
University of Stuttgart

2010–2011 C. Farhat, C. Bou-Mosleh, J. Cortial, D. Amsallem
Stanford University

Computational barrier

High-fidelity simulation

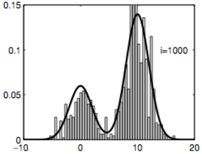
- + An indispensible tool
- - Very high computational cost



Barrier

Time-critical applications

Many query



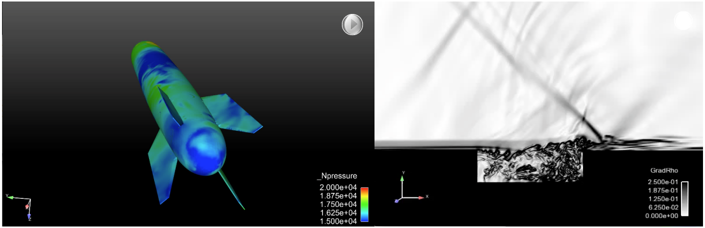
[Andrieu et al., 2003]

Real time

[Nat'l Power Grid Sim. Cap.]

Objective: break barrier

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

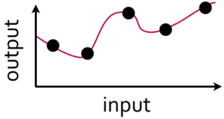
- Fast-turnaround design
- Uncertainty quantification (UQ)

Surrogate modeling

inputs $\boldsymbol{\mu}$ → full-order model → outputs $\mathbf{y}$

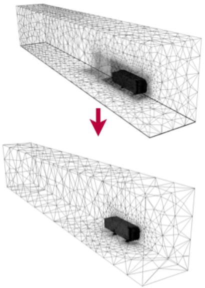
inputs $\boldsymbol{\mu}$ → surrogate model → outputs $\mathbf{y}$

1) Data fits



- Not physics based
+ High speedups

2) Coarsened physics



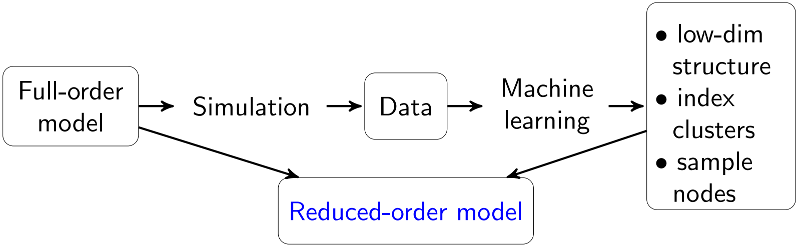
+ Physics based
- Low speedups

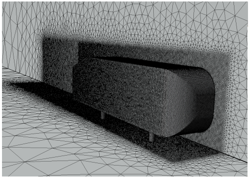
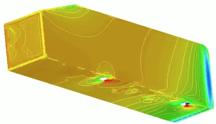
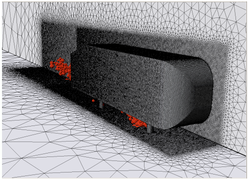
3) Reduced-order models (ROMs)

+ Physics based
+ High speedups
+ Preserve structure
+ Rigorous error analysis
- **Unproven for nonlinear dynamical systems**

ROM = data science + modeling and simulation

Goal: exploit simulation data to drastically reduce simulation costs



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

My 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 [C. et al., 2012, C. et al., 2015c]
- Low cost
  - Sample-mesh approach [C. et al., 2011b, C. et al., 2013]
  - Leverage time-domain data [C. et al., 2015b]
- Certification
  - Error bounds [C. et al., 2015a]
  - Statistical error modeling [Drohmann and C., 2015]
- Reliability
  - *A posteriori* h-refinement [C., 2015]

My research goal

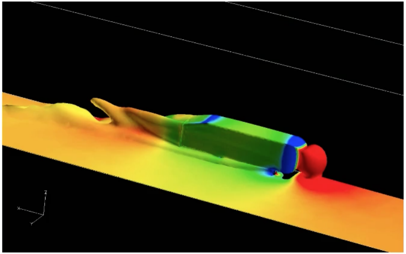
Nonlinear model-reduction methods that are
accurate, low cost, certified, and reliable.

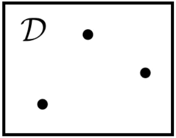
- Accuracy
  - Improve projection technique [C. et al., 2011a, C. et al., 2015a]
  - Preserve problem structure [C. et al., 2012, C. et al., 2015c]
- Low cost
  - Sample-mesh approach [C. et al., 2011b, C. et al., 2013]
  - Leverage time-domain data [C. et al., 2015b]
- Certification
  - Error bounds [C. et al., 2015a]
  - Statistical error modeling [Drohmann and C., 2015]
- Reliability
  - *A posteriori* h-refinement [C., 2015]

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

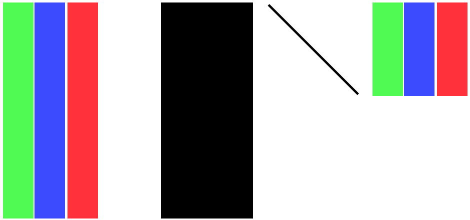




POD–Galerkin: offline data collection

2 Data compression

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

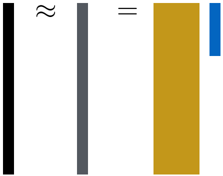


- Truncate: $\boldsymbol{\Phi} = [\boldsymbol{u}_1 \;\; \cdots \;\; \boldsymbol{u}_p]$

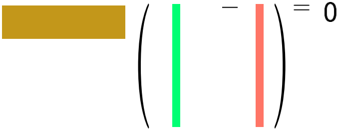
POD–Galerkin: online projection

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

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

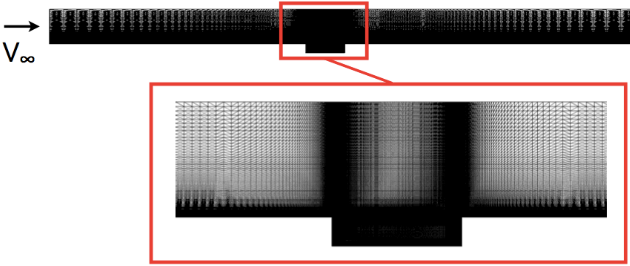


2 $\boldsymbol{\Phi}^T\left(\boldsymbol{f}(\tilde{\boldsymbol{x}}; t, \boldsymbol{\mu}) - \frac{d\tilde{\boldsymbol{x}}}{dt}\right) = 0$



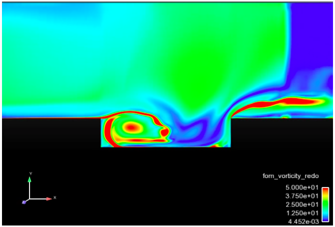
Galerkin ROM: $$\frac{d\hat{\boldsymbol{x}}}{dt} = \boldsymbol{\Phi}^T\boldsymbol{f}(\boldsymbol{\Phi}\hat{\boldsymbol{x}}; t, \boldsymbol{\mu}), \quad \hat{\boldsymbol{x}}(0, \boldsymbol{\mu}) = \boldsymbol{\Phi}^T\boldsymbol{x}^0(\boldsymbol{\mu})$$

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

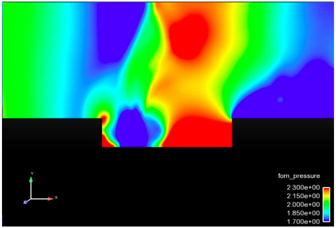


- Unsteady Navier–Stokes
- DES turbulence model
- 1.2 million degrees of freedom
- Re $= 6.3 \times 10^6$
- $M_\infty = 0.6$
- CFD code: AERO-F [Farhat et al., 2003]

Full-order model responses

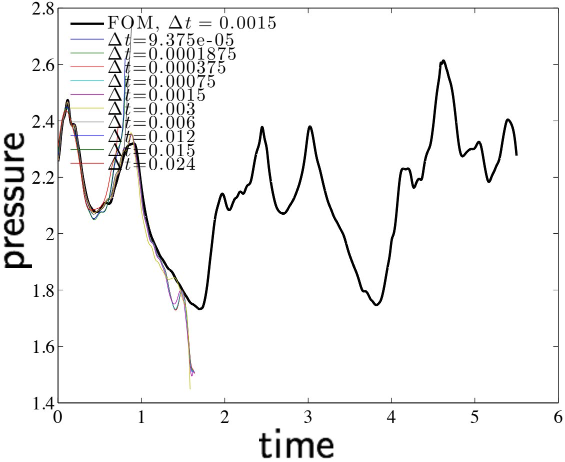


vorticity field



pressure field

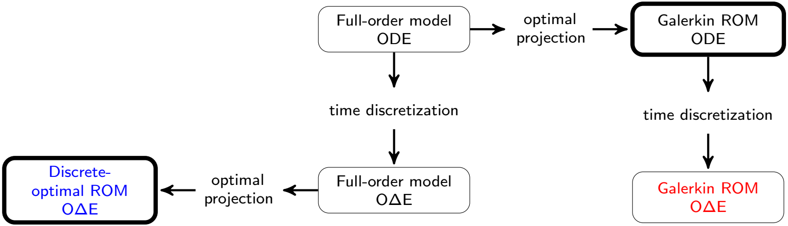
POD–Galerkin failure



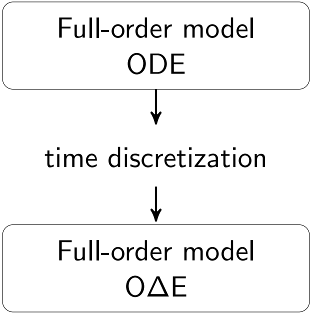
- Galerkin ROMs unstable

How to construct a ROM for nonlinear dynamical systems?

- Optimize then discretize? (Galerkin)
- Discretize then optimize? (discrete optimal)



- Outstanding questions:
  1. Which notion of optimality is better in practice?
  2. Are the two techniques ever equivalent?
  3. Discrete-time error bounds?


Full-order model
ODE
time discretization
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: $\boxed{\boldsymbol{r}^n(\boldsymbol{x}^n) = 0}$, $n = 1, \ldots, N$

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

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

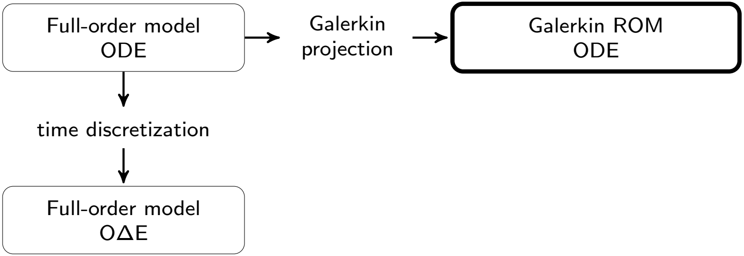
- OΔE, Runge–Kutta: $\boxed{\boldsymbol{r}_i^n(\boldsymbol{x}_1^n, \ldots, \boldsymbol{x}_s^n) = 0}$, $i = 1, \ldots, s$

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

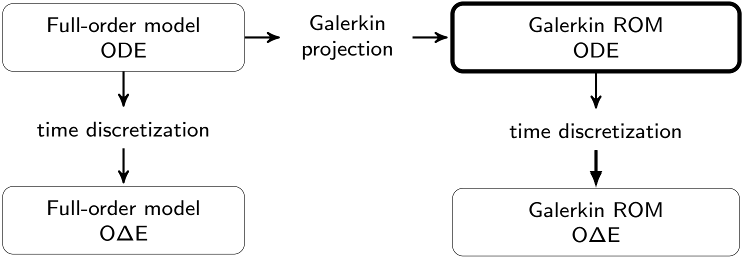
$$\boldsymbol{x}^n = \boldsymbol{x}^{n-1} + \Delta t \sum_{i=1}^{s} b_i \boldsymbol{x}_i^n \text{ (explicit state update)}$$

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

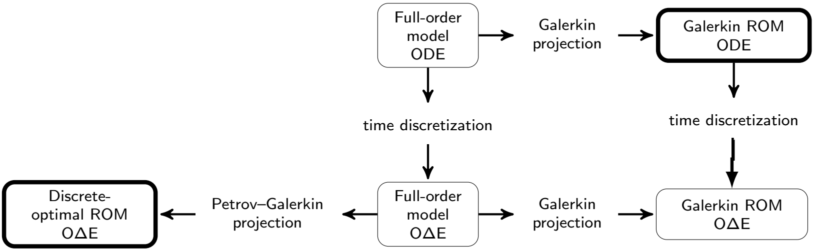
$$\hat{\boldsymbol{r}}^n(\hat{\boldsymbol{x}}^n) = 0, \quad n = 1, \ldots, N$$

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

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

Can we fix this? Will doing so help?

Discrete-optimal ROM: first discretize, then optimize



Discrete-optimal ROM

- FOM OΔE

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

- Discrete-optimal ROM OΔE:

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

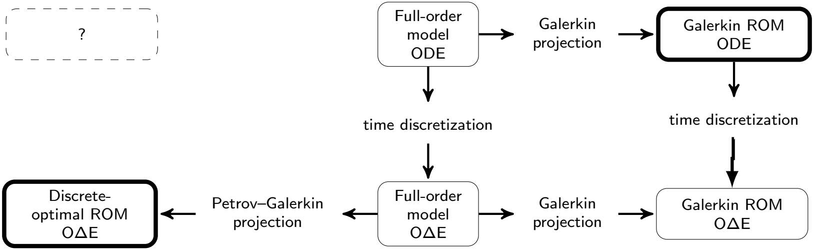
$$\Updownarrow$$

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

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

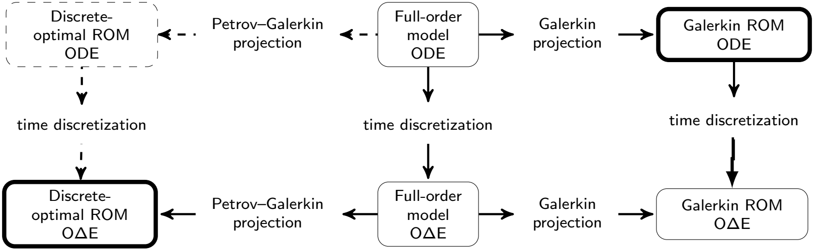
+ Discrete solution is optimal

Does the discrete-optimal ROM have a time-continuous representation?



Does the discrete-optimal ROM have a time-continuous representation?

Sometimes.



Discrete-optimal ROM: continuous representation

Theorem

The discrete-optimal ROM is equivalent to applying a Petrov–Galerkin projection to the FOM ODE with test basis

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

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

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 $\Delta t \to 0$ 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 $[\frac{\partial \boldsymbol{r}^n}{\partial \boldsymbol{x}}]^{-1} = \boldsymbol{A}^T\boldsymbol{A}$.*

Discrete-time error bound

Theorem

If the following conditions hold:

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

then

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

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

with

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

Discrete-optimal ROM yields a smaller error bound

Theorem (Backward Euler)

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

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

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

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

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

Corollary (Discrete-optimal smaller error bound)

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

Discrete-optimal ROM has an interesting time-step dependence

Corollary (Backward Euler)

Define

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

Then, the discrete-optimal error can also be bounded as

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

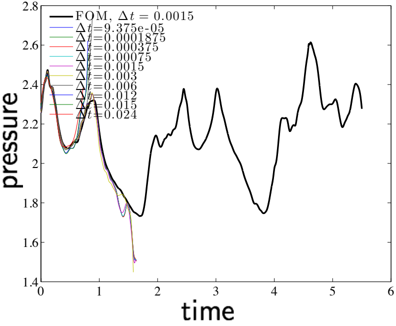
with $\mu^j := \|\boldsymbol{\Phi}\Delta\hat{\boldsymbol{x}}_D^j - \Delta\bar{\boldsymbol{x}}^j\| / \|\Delta\bar{\boldsymbol{x}}^j\|$.

Effect of decreasing Δt:

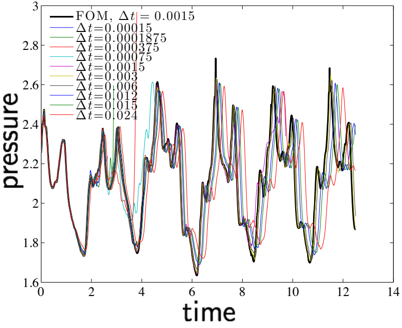
+ The terms $\Delta t(1+\kappa\Delta t)$ and $1/(h)^{j+1}$ decrease
- The number of total time instances n increases

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

Galerkin and discrete-optimal responses for basis dimension $p = 204$



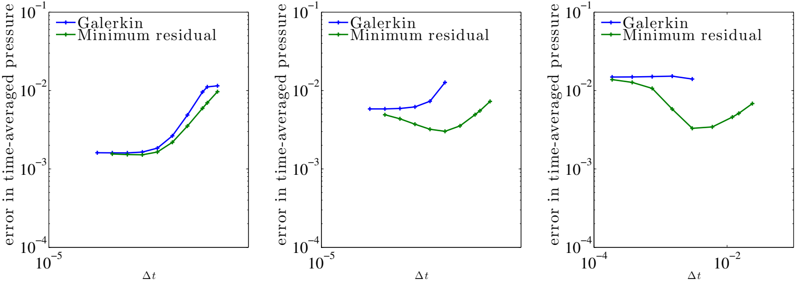
(a) Galerkin



(b) Discrete optimal

- Galerkin ROMs unstable for long time intervals (consistent with previous results [C. et al., 2013, C. et al., 2011a, C., 2011])
+ Discrete-optimal ROMs accurate and stable (most time steps)

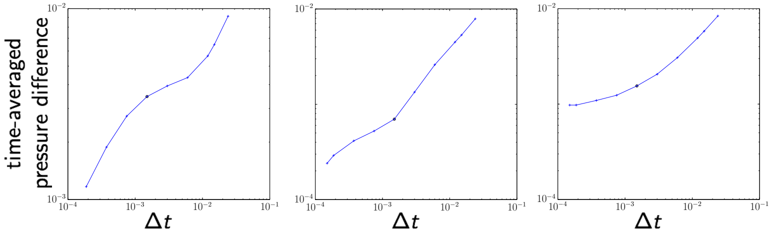
Discrete-optimal ROM: superior performance



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

✓ Discrete-optimal ROM yields a smaller error for all time intervals and time steps.

Limiting equivalence

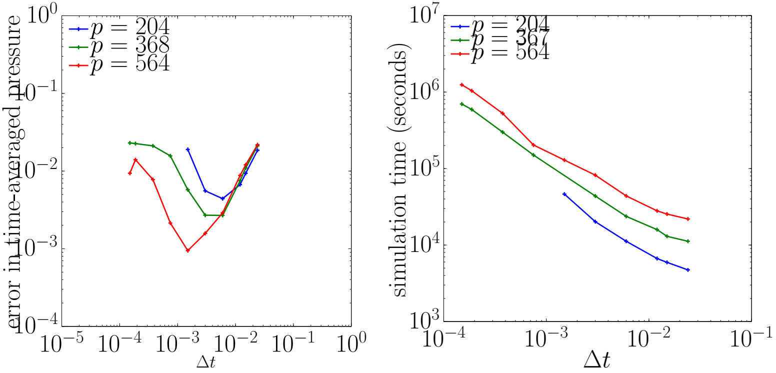


(f) $p = 204$ (g) $p = 368$ (h) $p = 564$

Galerkin/discrete-optimal difference in the stable Galerkin interval
$0 \leq t \leq 1.1$.

✓ The discrete-optimal ROM converges to Galerkin as $\Delta t \to 0$.

Discrete-optimal performance ($t \leq 12.5$ sec)



✓ An intermediate Δt 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

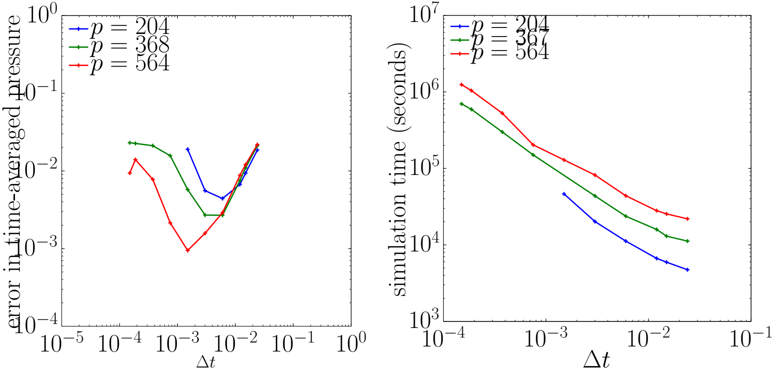
- Discrete optimality outperforms continuous optimality (Galerkin) in practice
- Equivalence conditions
  1. Limit of $\Delta t \rightarrow 0$
  2. Explicit schemes
  3. Positive definite residual Jacobians
- Discrete-time error bounds
  - Discrete-optimal ROM yields smaller error bound than Galerkin
  - Ambiguous role of time step Δt
- Numerical experiments
  - Discrete-optimal ROM yields a smaller error than Galerkin
  - Equivalent as $\Delta t \rightarrow 0$
  - Error minimized for intermediate Δt

My 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 [C. et al., 2012, C. et al., 2015c]
+ Low cost
  - Sample-mesh approach [C. et al., 2011b, C. et al., 2013]
  - Leverage time-domain data [C. et al., 2015b]
+ Certification
  - Error bounds [C. et al., 2015a]
  - Statistical error modeling [Drohmann and C., 2015]
+ Reliability
  - *A posteriori h*-refinement [C., 2015]

Discrete-optimal performance ($t \leq 2.5$ sec)



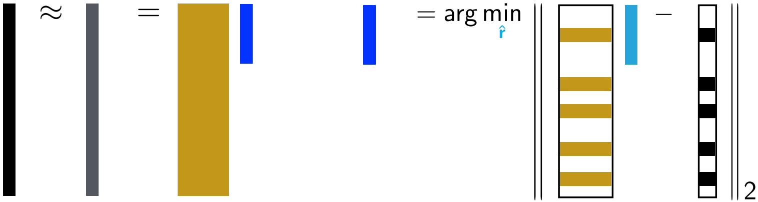
+ Always sub-3% errors
- More expensive than the FOM
  - FOM simulation: 1 hour, 48 CPU
  - Discrete-optimal ROM simulation (fastest): 1.3 hours, 48 CPU

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



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

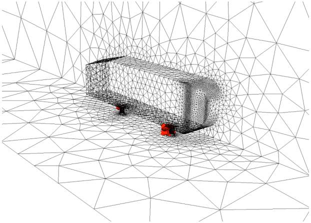
$$= \arg\min_{\hat{\boldsymbol{z}} \in \mathbb{R}^p} \|\underbrace{(\boldsymbol{P}\Phi_R)^+\boldsymbol{P}}_{\boldsymbol{A}}\boldsymbol{r}^n(\boldsymbol{\Phi}\hat{\boldsymbol{z}})\|_2^2.$$

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

Sample mesh: HPC implementation

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

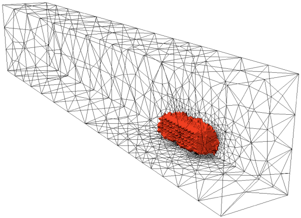
- *Goals*:
  - + Reuse existing computational-mechanics codes
  - + Minimize number of required computing cores
  - + Scalability
- *Key*: GNAT samples only a few entries of the residual $\boldsymbol{P}\boldsymbol{r}^n$
- *Idea*: Extract minimal subset of the mesh



Postprocessing mesh: HPC implementation

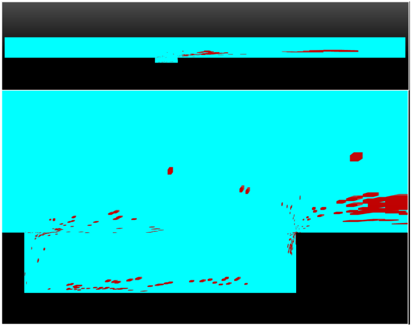
$$\text{inputs } \boldsymbol{\mu} \rightarrow \boxed{\text{reduced-order model}} \rightarrow \text{outputs } \mathbf{y}$$

- *Observations*:
  + Outputs $\mathbf{y}$ are often defined locally in space (e.g., lift)
  - Outputs may not be computable on sample mesh
- *Output computation*:
  1. Read reduced state $\hat{\boldsymbol{x}}^n$, $n = 1, \ldots, M$ computed by GNAT
  2. Assemble solution on minimal output-computation mesh
  3. Compute outputs



Cavity-flow problem: GNAT

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



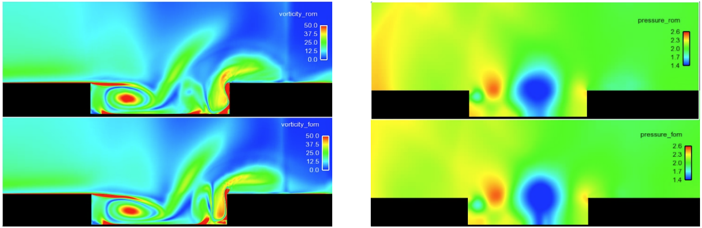
- Sample mesh: 4.1% nodes, 3.0% cells
- \+ Small problem size: can run on many fewer cores

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

My research goal

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

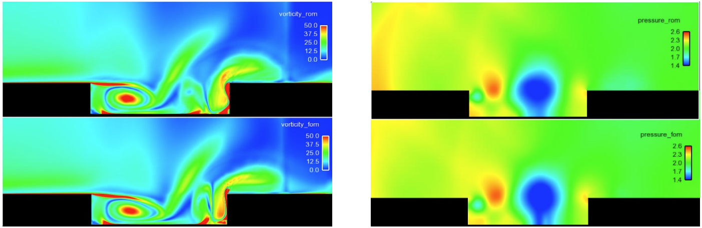
+ Accuracy
  - Improve projection technique [C. et al., 2011a, C. et al., 2015a]
  - Preserve problem structure [C. et al., 2012, C. et al., 2015c]
+ Low cost
  - Sample-mesh approach [C. et al., 2011b, C. et al., 2013]
  - Leverage time-domain data [C. et al., 2015b]
+ Certification
  - Error bounds [C. et al., 2015a]
  - Statistical error modeling [Drohmann and C., 2015]
+ Reliability
  - *A posteriori h*-refinement [C., 2015]

GNAT performance

vorticity field | pressure field

GNAT ROM

FOM



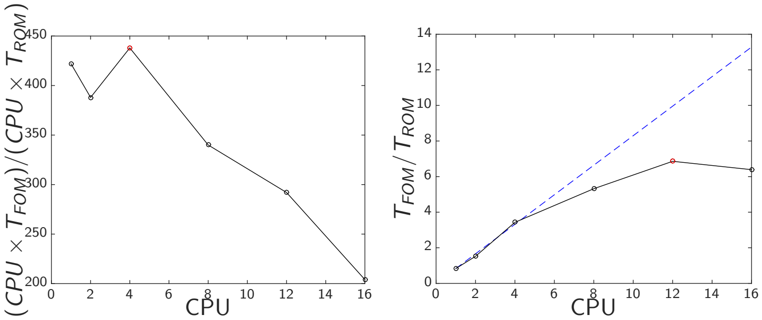
- FOM: 5 hour x 48 CPU
- GNAT ROM: 32 min x 2 CPU

\+ 229x CPU-hour savings. Good for many query.

\- 9.4x walltime savings. Bad for real time.

Why?

GNAT: strong scaling (Ahmed body) [C., 2011]



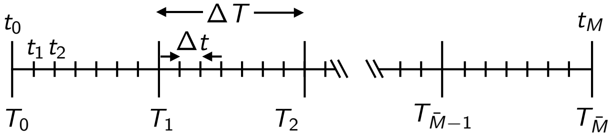
(e) CPU-hour savings (f) Walltime savings

+ Significant CPU-hour savings (max: 438 for 4 CPU)
- Modest walltime savings (max: 7 for 12 CPU)

Spatial parallelism is quickly saturated!

Time-parallel algorithms [Lions et al., 2001a, Farhat and Chandesris, 2003]

Goal: expose more parallelism to reduce walltime



- Fine propagator: time step Δt

$$\mathcal{F}(\boldsymbol{x}; \tau_1, \tau_2)$$

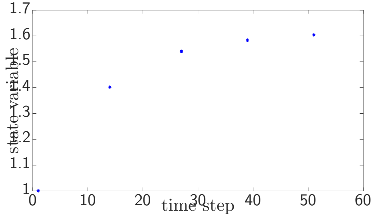
- Coarse propagator: time step ΔT

$$\mathcal{G}(\boldsymbol{x}; \tau_1, \tau_2)$$

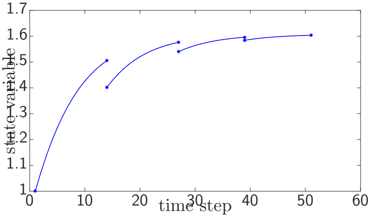
- Parareal iteration k (sequential and parallel steps):

$$\boldsymbol{x}_{k+1}^{m+1} = \mathcal{G}(\boldsymbol{x}_{k+1}^{m}; T_m, T_{m+1}) + \mathcal{F}(\boldsymbol{x}_{k}^{m}; T_m, T_{m+1}) - \mathcal{G}(\boldsymbol{x}_{k}^{m}; T_m, T_{m+1})$$

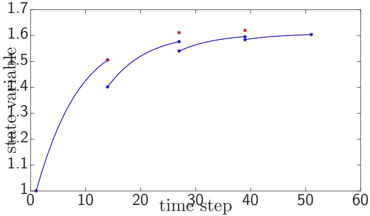
Illustration: sequential and parallel steps



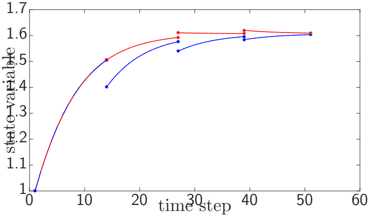
$\boldsymbol{x}_0^{m+1} = \mathcal{G}(\boldsymbol{x}_0^m; T_m, T_{m+1})$



$\mathcal{F}(\boldsymbol{x}_0^m; T_m, T_{m+1})$



$\boldsymbol{x}_1^{m+1} = \mathcal{F}(\boldsymbol{x}_0^m; T_m, T_{m+1}) + \mathcal{G}(\boldsymbol{x}_1^m; T_m, T_{m+1}) - \mathcal{G}(\boldsymbol{x}_0^m; T_m, T_{m+1})$



$\mathcal{F}(\boldsymbol{x}_1^m; T_m, T_{m+1})$

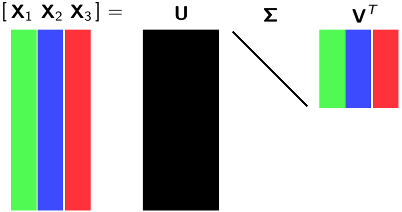
Coarse propagator

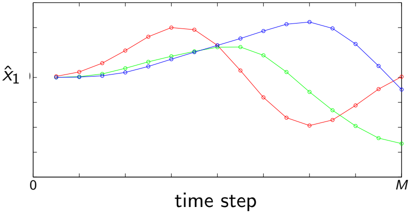
Critical: coarse propagator should be fast, accurate, stable

- Existing coarse propagators
  - Same integrator [Lions et al., 2001b, Bal and Maday, 2002]
  - Coarse spatial discretization
    [Fischer et al., 2005, Farhat et al., 2006, Cortial and Farhat, 2009]
  - Simplified phyics model [Baffico et al., 2002, Maday and Turinici, 2003, Blouza et al., 2011, Engblom, 2009, Maday, 2007]
  - Relaxed solver tolerance [Guibert and Tromeur-Dervout, 2007]
  - Reduced-order model (on the fly) [Farhat et al., 2006, Cortial and Farhat, 2009, Ruprecht and Krause, 2012, Chen et al., 2014]

Can we leverage offline data to improve the coarse propagator?

Revisit the SVD





First row of $\boldsymbol{V}^T$

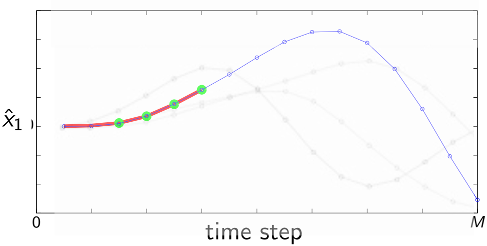
jth row of $\boldsymbol{V}^T$ contains a basis for time evolution of $\hat{x}_j$

- Construct $\boldsymbol{\Xi}_j$: basis for time evolution of $\hat{x}_j$

$$\boldsymbol{\Xi}_j := \left[\xi_j^1 \ \cdots \ \xi_j^{n_{\text{train}}}\right], \quad \xi_j^i := \left[v_{M(i-1)+1,j} \ \cdots \ v_{Mi,j}\right]^T$$

Previous method [C. et al., 2015b]

1. compute forecast by gappy POD in time domain:



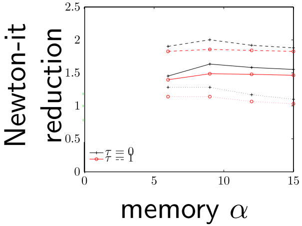
$\hat{\boldsymbol{x}}_1$ so far; memory $\alpha = 4$; forecast; temporal basis

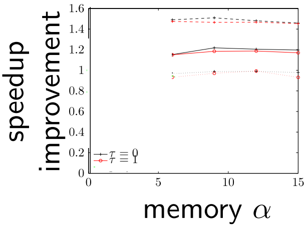
$$\boldsymbol{z}_j = \arg\min_{\boldsymbol{z}\in\mathbb{R}^{a_j}} \|\boldsymbol{Z}(m-1,\alpha)\boldsymbol{\Xi}_j\boldsymbol{z} - \boldsymbol{Z}(m-1,\alpha)g(\hat{x}_j)\|_2$$

- Time sampling: $\boldsymbol{Z}(k,\beta) := \begin{bmatrix}\boldsymbol{e}_{k-\beta} & \cdots & \boldsymbol{e}_k\end{bmatrix}^T$
- Time unrolling: $g(\hat{x}_j) : \hat{x}_j \mapsto [\hat{x}_j(t_0) \ \cdots \ \hat{x}_j(t_M)]^T$

2. use $\boldsymbol{e}_m^T\boldsymbol{\Xi}_j\boldsymbol{z}_j$ as *initial guess* for $\hat{x}_j(t_m)$ in Newton solver

Previous results: structural dynamics [C. et al., 2015b]



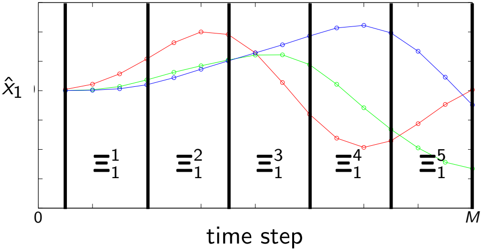


- + Newton iterations reduced by up to ∼2x
- + Speedup improved by up to ∼1.5x
- + No accuracy loss
- + Applicable to any nonlinear ROM
- - Insufficient for real-time computation

Can we apply the same idea for the coarse propagator?

Coarse propagator for coordinate j and time interval m

- **Offline**: Construct time-evolution basis $\boldsymbol{\Xi}_j^m$



- **Online**: Coarse propagator $\mathcal{G}_j^m$ defined via forecasting:
  1. Compute α time steps with fine propagator
  2. Compute forecast via gappy POD
  3. Select last timestep of forecast

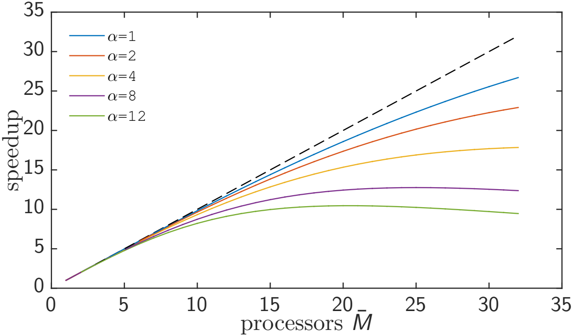
$$\mathcal{G}_j^m : (\hat{\mathbf{x}}_j; T_m, T_{m+1}) \mapsto \mathbf{e}_{\Delta T/\Delta t}^T \boldsymbol{\Xi}_j^m \left[\mathbf{Z}(\alpha+1, \alpha)\boldsymbol{\Xi}_j^m\right]^+ \begin{bmatrix} \mathcal{F}(\hat{\mathbf{x}}_j; T_m, T_m + \Delta t) \\ \vdots \\ \mathcal{F}(\hat{\mathbf{x}}_j; T_m, T_m + \Delta t\alpha) \end{bmatrix}$$

Ideal-conditions speedup

Theorem

If $g(\hat{x}_j) \in \text{range}(\boldsymbol{\Xi}_j), \quad j = 1, \ldots, p$, then the proposed method converges in one parareal iteration and realizes a theoretical speedup of

$$\frac{\bar{M}}{\bar{M}(\bar{M}-1)\alpha/M + 1}.$$



Ideal-conditions speedup for $M = 5000$

Ideal-conditions speedup with initial guesses

Corollary

If $\boldsymbol{f}$ is nonlinear, $g(\hat{x}_j) \in range(\boldsymbol{\Xi}_j)$, $\quad j = 1, \ldots, p$, and the forecasting method also provides Newton-solver initial guesses, then

1. *the method converges in* **one parareal iteration**, *and*
2. *only α nonlinear systems of algebraic equations are solved in each time interval.*

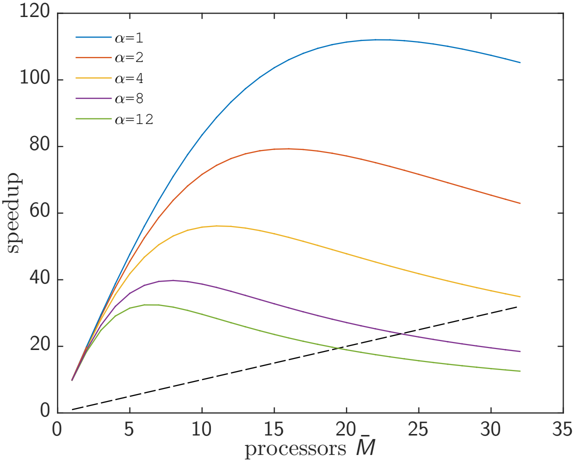
The method then realizes a theoretical speedup of

$$\frac{M}{(\bar{M}\alpha) + (M/\bar{M} - \alpha)\tau_r}$$

relative to the sequential algorithm without forecasting. Here,

$$\tau_r = \frac{\textit{residual computation time}}{\textit{nonlinear-system solution time}}.$$

Ideal-conditions speedup with initial-guesses



Ideal-condition speedup for $M = 5000$, $\tau_r = 1/10$

Significant speedups possible by leveraging time-domain data!

Stability

Theorem

If the fine propagator is stable, i.e.,

$$\|\mathcal{F}(\boldsymbol{x};\tau_1,\tau_2)\| \leq (1 + C_{\mathcal{F}}\Delta T)\|\boldsymbol{x}\|,$$

then the proposed method is also stable, i.e.,

$$\|\hat{\boldsymbol{x}}^m_{k+1}\| \leq C_m \exp(C_F m\Delta T)\|\hat{\boldsymbol{x}}^0\|.$$

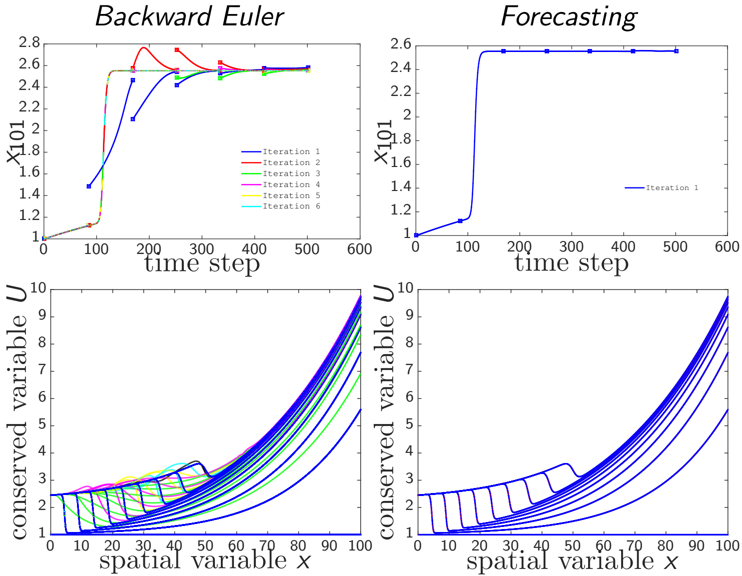
- $C_m := \sum_{k=1}^{m} \binom{k}{m}\beta_k\gamma^m\alpha^k(\Delta T/\Delta t)^{m-k}$
- $\beta_k := \exp(-C_{\mathcal{F}}k(\Delta T - \Delta t\alpha)) \leq 1$
- $\gamma := \max(\max_{m,j} 1/\|\boldsymbol{Z}(\alpha+1,\alpha)\boldsymbol{\Xi}_j^m\|, 1/\sigma_{\min}(\boldsymbol{Z}(\alpha+1,\alpha)\boldsymbol{\Xi}_j^m))$

Example: inviscid Burgers equation [Rewienski, 2003]

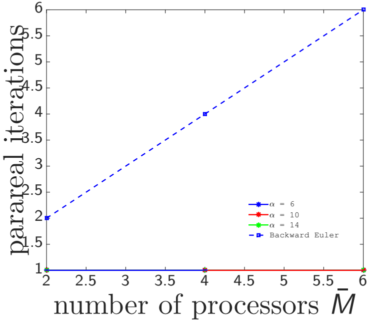
$$\frac{\partial u(x,\tau)}{\partial \tau} + \frac{1}{2}\frac{\partial\left(u^2(x,\tau)\right)}{\partial x} = 0.02e^{\mu_2 x}$$
$$u(0,\tau) = \mu_1,\ \forall \tau \in [0,25]$$
$$u(x,0) = 1,\ \forall x \in [0,100],$$

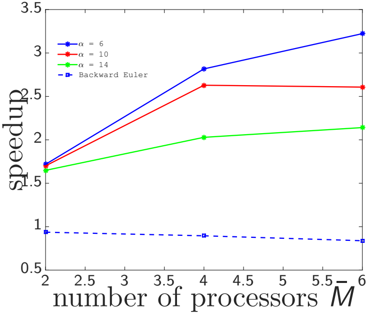
- Discretization: Godunov's scheme
- $(\mu_1, \mu_2) \in [2.5, 3.5] \times [0.02, 0.075]$
- $\Delta t = 0.1$, $M = 250$ fine time steps
- FOM: $N = 500$ degrees of freedom
- ROM: LSPG [C. et al., 2011a] with POD basis dimension $p = 100$
- $n_{\text{train}} = 4$ training points (LHS sampling); random online point
- Two coarse propagators: Backward Euler and forecasting

Forecasting outperforms backward Euler



Parareal performance





+ *Forecasting*: minimum possible iterations
- *Backward Euler*: maximum possible iterations

More parallelism successfully exposed!

Summary: Leverage time-domain data

Use temporal data to reduce ROM simulation time

- **offline**: time-evolution bases from right singular vectors
- **online**: use as coarse propagator
  1. compute α time steps with fine propagator
  2. use gappy POD to forecast

+ theory: excellent speedup and stabilty
+ ideal parareal performance observed
+ significant improvement over Backward Euler
+ no additional error introduced
+ generally applicable

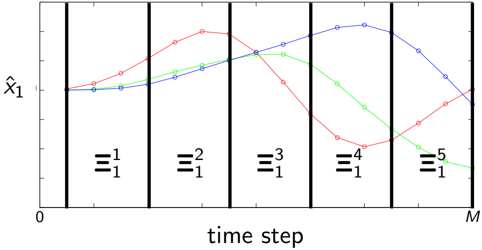
Opportunities at Sandia National Laboratories, Livermore

We are hiring summer interns, postdocs, and staff

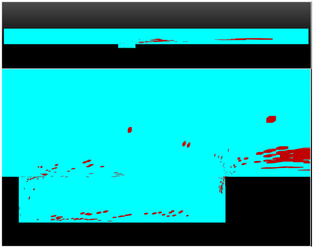
- Model reduction
- Uncertainty quantification
- Machine learning
- High-performance computing
- Cybersecurity
- Data analytics

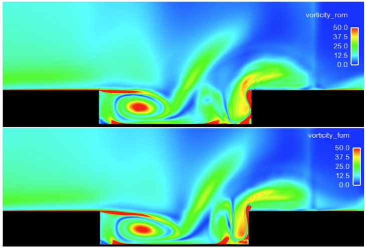
Email me if interested: `ktcarlb@sandia.gov`

Questions?



$\hat{x}_1$
Ξ_1^1
Ξ_1^2
Ξ_1^3
Ξ_1^4
Ξ_1^5
0
M
time step





vorticity_rom
50.0
37.5
25.0
12.5
0.0
vorticity_fom
50.0
37.5
25.0
12.5
0.0

My 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 [C. et al., 2012, C. et al., 2015c]
+ Low cost
  - Sample-mesh approach [C. et al., 2011b, C. et al., 2013]
  - Leverage time-domain data [C. et al., 2015b]
+ Certification
  - Error bounds [C. et al., 2015a]
  - Statistical error modeling [Drohmann and C., 2015]
+ Reliability
  - *A posteriori* h-refinement [C., 2015]

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.

Andrieu, C., De Freitas, N., Doucet, A., and Jordan, M. I. (2003).
An introduction to mcmc for machine learning.
Machine learning, 50(1-2):5–43.

Antoulas, A. C. (2005).
Approximation of Large-Scale Dynamical Systems.
Society for Industrial and Applied Mathematics, Philadelphia, PA.

Baffico, L., Bernard, S., Maday, Y., Turinici, G., and Zérah, G. (2002).
Parallel-in-time molecular-dynamics simulations.
Physical Review E, 66(5):057701.

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

Bal, G. and Maday, Y. (2002).

A "parareal" time discretization for non-linear pdes with application to the pricing of an american put.
In *Recent developments in domain decomposition methods*, pages 189–202. Springer Berlin Heidelberg.

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.

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.

Blouza, A., Boudin, L., and Kaber, S. M. (2011).
Parallel in time algorithms with reduction methods for solving chemical kinetics.
Communications in Applied Mathematics and Computational Science, 5(2):241–263.

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. (2011).
Model Reduction of Nonlinear Mechanical Systems via Optimal Projection and Tensor Approximation.
PhD thesis, Stanford University.

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.

C., K., Tuminaro, R., and Boggs, P. (2012).
Efficient structure-preserving model reduction for nonlinear mechanical systems with application to structural dynamics.
In *AIAA Paper 2012-1969, 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Honolulu, Hawaii.*

C., K., Tuminaro, R., and Boggs, P. (2015c).
Preserving Lagrangian structure in nonlinear model reduction with application to structural dynamics.
SIAM J. Sci. Comput., 37(2):B153—B184.

Chen, F., Hesthaven, J. S., and Zhu, X. (2014).
On the use of reduced basis methods to accelerate and stabilize the parareal method.
In *Reduced Order Methods for Modeling and Computational Reduction*, pages 187–214. Springer.

Cortial, J. and Farhat, C. (2009).
A time-parallel implicit method for accelerating the solution of non-linear structural dynamics problems.
International Journal for Numerical Methods in Engineering, 77(4):451.

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.

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.

Engblom, S. (2009).
Parallel in time simulation of multiscale stochastic chemical kinetics.
Multiscale Modeling & Simulation, 8(1):46–68.

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. and Chandesris, M. (2003).

Time-decomposed parallel time-integrators: theory and feasibility studies for fluid, structure, and fluid-structure applications.
International Journal for Numerical Methods in Engineering, 58(9):1397–1434.

Farhat, C., Cortial, J., Dastillung, C., and Bavestrello, H. (2006).
Time-parallel implicit integrators for the near-real-time prediction of linear structural dynamic responses.
International Journal for Numerical Methods in Engineering, 67:697–724.

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.

Fischer, P. F., Hecht, F., and Maday, Y. (2005).

A parareal in time semi-implicit approximation of the navier-stokes equations.
In *Domain decomposition methods in science and engineering*, pages 433–440. Springer.

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.

Guibert, D. and Tromeur-Dervout, D. (2007).
Adaptive parareal for systems of odes.
In *Domain decomposition methods in science and engineering XVI*, pages 587–594. Springer.

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.

Lions, J., Maday, Y., and Turinici, G. (2001a).
A "parareal" in time discretization of pdes.
Comptes Rendus de l'Academie des Sciences Series I Mathematics, 332(7):661–668.

Lions, J.-L., Maday, Y., and Turinici, G. (2001b).

Résolution d'edp par un schéma en temps "parareal".
Comptes Rendus de l'Académie des Sciences-Series I-Mathematics, 332(7):661–668.

Maday, Y. (2007).
Parareal in time algorithm for kinetic systems based on model reduction.
High-dimensional partial differential equations in science and engineering, 41:183–194.

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

Maday, Y. and Turinici, G. (2003).
Parallel in time algorithms for quantum control: Parareal time discretization scheme.
International journal of quantum chemistry, 93(3):223–228.

Moore, B. (1981).

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

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.

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.

Rewienski, M. J. (2003).
A Trajectory Piecewise-Linear Approach to Model Order Reduction of Nonlinear Dynamical Systems.

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.

Ruprecht, D. and Krause, R. (2012).
Explicit parallel-in-time integration of a linear acoustic-advection system.
Computers & Fluids, 59:72–83.

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.

Willcox, K. and Peraire, J. (2002).
Balanced model reduction via the proper orthogonal decomposition.
AIAA Journal, 40(11):2323–2330.