SAND2016-10469PE

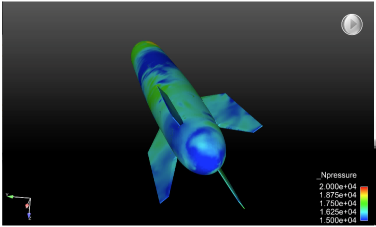
Model reduction for nonlinear dynamical systems: discrete optimality and time parallelism

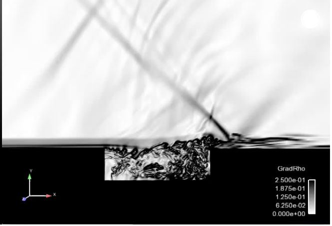
Kevin Carlberg

Sandia National Laboratories
Livermore, California

University of South Carolina
October 17, 2016

Computational barrier at Sandia



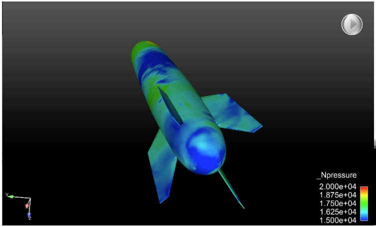


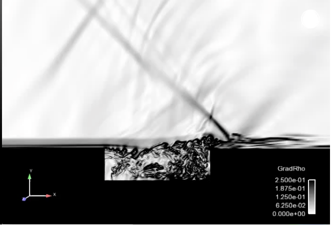
- CFD model
  - 100 million cells
  - 200,000 time steps
- High simulation costs
  - 6 weeks, 5000 cores
  - 6 runs maxes out Cielo

Barrier

- Fast-turnaround design
- Uncertainty quantification

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, Or and Speyer, 2010, Ma et al., 2011]
  - Moment matching [Bai, 2002, Freund, 2003, Gallivan et al., 2004, Gugercin et al., 2008, Baur et al., 2011]
  - Loewner framework [Lefteriu and Antoulas, 2010, Ionita and Antoulas, 2014]
  - \+ *Reliable*: guaranteed stability, moment matching, optimality
  - \+ *Certified*: *a priori* error bounds

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, Or and Speyer, 2010, Ma et al., 2011]
  - Moment matching [Bai, 2002, Freund, 2003, Gallivan et al., 2004, Gugercin et al., 2008, Baur et al., 2011]
  - Loewner framework [Lefteriu and Antoulas, 2010, Ionita and Antoulas, 2014]

  \+ *Reliable*: guaranteed stability, moment matching, optimality

  \+ *Certified*: *a priori* 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* convergence

  \+ *Certified*: sharp, computable *a posteriori* error bounds

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, Or and Speyer, 2010, Ma et al., 2011]
  - Moment matching [Bai, 2002, Freund, 2003, Gallivan et al., 2004, Gugercin et al., 2008, Baur et al., 2011]
  - Loewner framework [Lefteriu and Antoulas, 2010, Ionita and Antoulas, 2014]
  - \+ *Reliable*: guaranteed stability, moment matching, optimality
  - \+ *Certified*: *a priori* 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* convergence
  - \+ *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 [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]

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 [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]

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

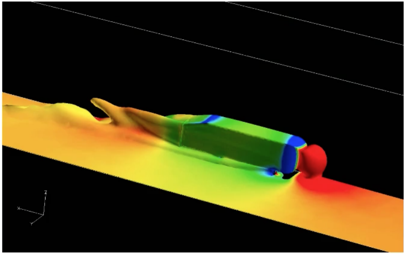
POD–Galerkin: offline data collection

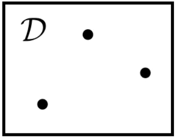
$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{f}(\boldsymbol{x}; t, \boldsymbol{\mu}); \quad \boldsymbol{x}(0, \boldsymbol{\mu}) = \boldsymbol{x}^0(\boldsymbol{\mu}), \quad t \in [0, T], \quad \boldsymbol{\mu} \in \mathcal{D}$$

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

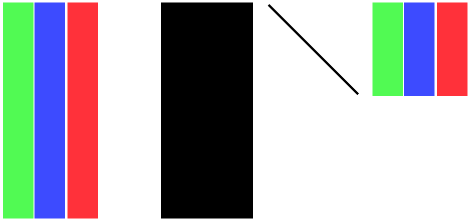




POD–Galerkin: offline data collection

2 Data compression

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

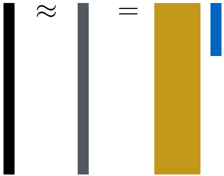


- Truncate: $\mathbf{\Phi} = [\boldsymbol{u}_1 \ \cdots \ \boldsymbol{u}_p]$

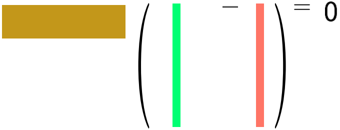
POD–Galerkin: online projection

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



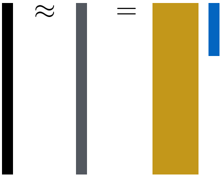
2 $\boldsymbol{\Phi}^T\left(\boldsymbol{f}(\tilde{\boldsymbol{x}}; t, \boldsymbol{\mu}) - \frac{d\tilde{\boldsymbol{x}}}{dt}\right) = 0$



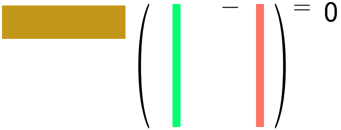
POD–Galerkin: online projection

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

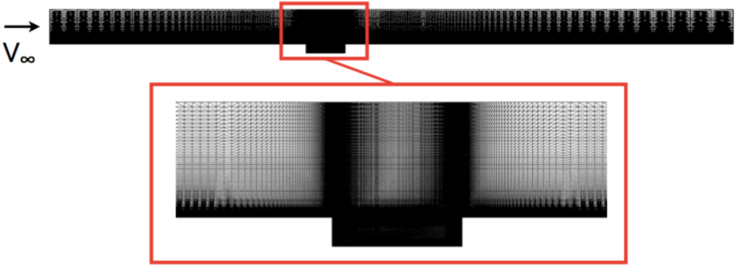


2 $\boldsymbol{\Phi}^T\left(\boldsymbol{f}(\tilde{\boldsymbol{x}}; t, \boldsymbol{\mu}) - \frac{d\tilde{\boldsymbol{x}}}{dt}\right) = 0$



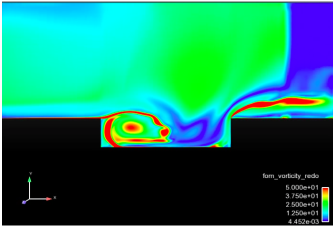
Galerkin ROM: $$\boxed{\frac{d\hat{\boldsymbol{x}}}{dt} = \boldsymbol{\Phi}^T\boldsymbol{f}(\boldsymbol{\Phi}\hat{\boldsymbol{x}}; t, \boldsymbol{\mu}), \quad \hat{\boldsymbol{x}}(0, \boldsymbol{\mu}) = \boldsymbol{\Phi}^T\boldsymbol{x}^0(\boldsymbol{\mu})}$$

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

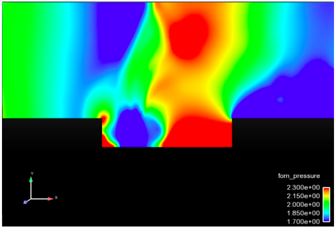


- Unsteady Navier–Stokes
- DES turbulence model
- 1.2 million degrees of freedom
- $\text{Re} = 6.3 \times 10^6$
- $M_\infty = 0.6$
- CFD code: AERO-F [Farhat et al., 2003]

Full-order model responses

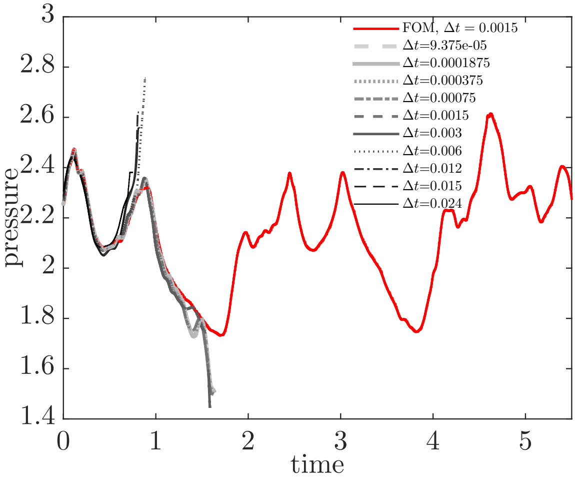


vorticity field



pressure field

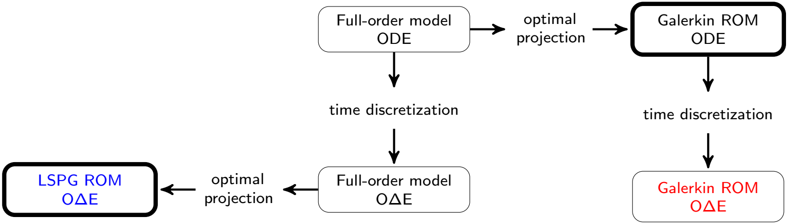
POD–Galerkin failure (basis dimension 204)



- Galerkin ROMs unstable

How to construct a ROM for nonlinear dynamical systems?

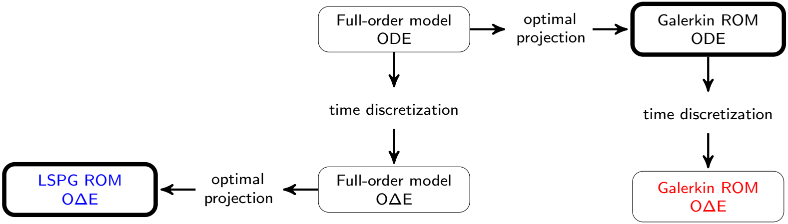
- Optimize then discretize? (Galerkin)
- Discretize then optimize? (Least-squares Petrov–Galerkin)



- Related investigation (LTI) [Lattimer, 2016]

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. Are the two techniques ever equivalent?
  3. Discrete-time error bounds?
  4. Time step selection?
- Related investigation (LTI) [Lattimer, 2016]

Full-order model
ODE

↓

time discretization

↓

Full-order model
OΔE

Full-order model (FOM)

- ODE: time continuous

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

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

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

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

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

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

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

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

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

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

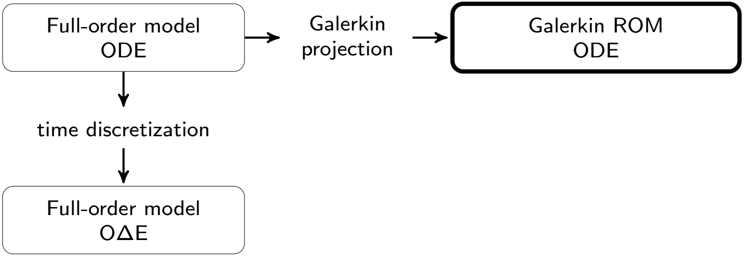
- OΔE, Runge–Kutta: $\boxed{\boldsymbol{r}_i^n(\boldsymbol{x}_1^n, \ldots, \boldsymbol{x}_s^n) = 0}$, $i = 1, \ldots, s$

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

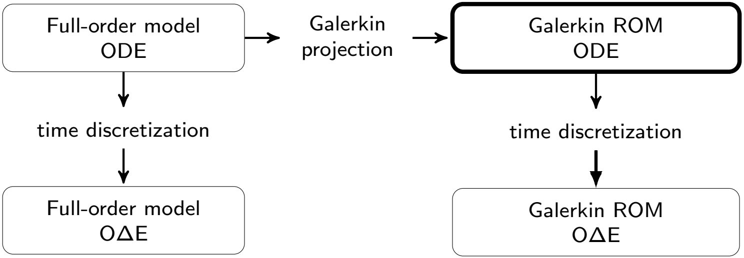
$$\boldsymbol{x}^n = \boldsymbol{x}^{n-1} + \Delta t \sum_{i=1}^{s} b_i \boldsymbol{x}_i^n \text{ (explicit state update)}$$

This talk focuses on linear multistep schemes.

Galerkin ROM: first optimize



Galerkin: first optimize, then discretize



Galerkin ROM

- ODE

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

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:

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

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

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

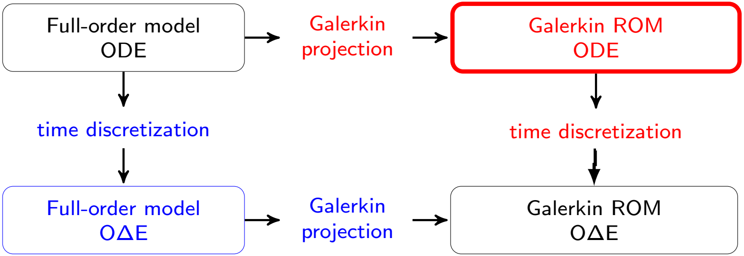
Can we fix this? Will doing so help?

Galerkin ROM: Commutativity

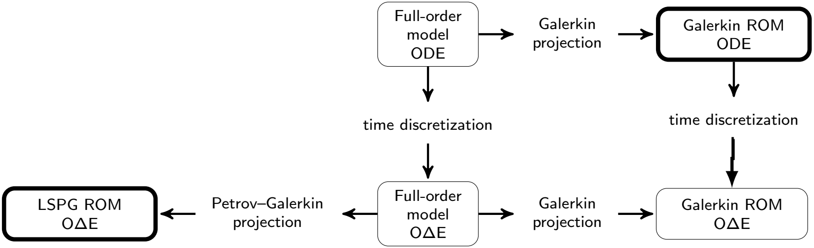
Theorem

Projection and time discretization are commutative for Galerkin ROMs:

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



LSPG ROM: first discretize, then optimize



LSPG ROM

- FOM OΔE

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

LSPG ROM

- FOM OΔE

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

- 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]

LSPG ROM

- FOM OΔE

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

- LSPG ROM OΔE:

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

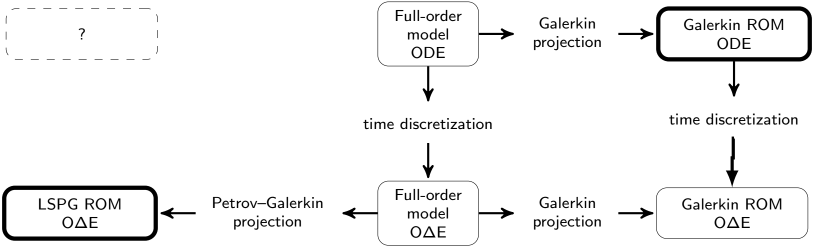
$$\Updownarrow$$

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

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

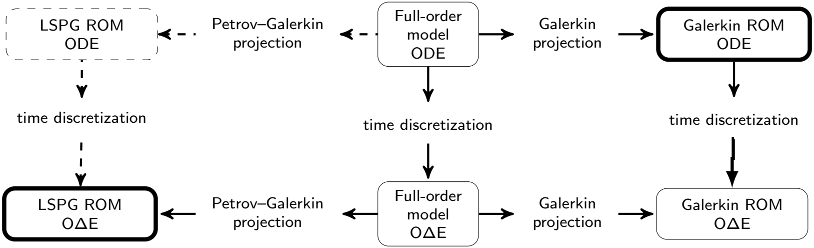
\+ Discrete solution is optimal

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

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

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).*

LSPG ROM: continuous representation

Theorem

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

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

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?

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?

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.

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

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

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

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

with

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

LSPG ROM yields a smaller error bound

Theorem (Backward Euler)

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

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

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

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

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

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

LSPG ROM has an interesting time-step dependence

Corollary (Backward Euler)

Define

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

Then, the LSPG error can also be bounded as

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

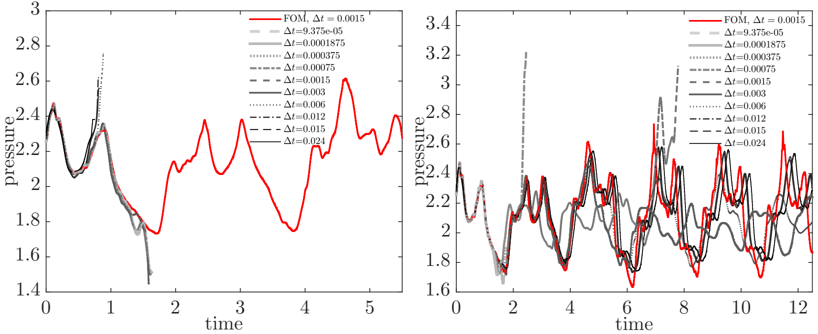
with $\mu^j := \|\boldsymbol{\Phi}\Delta\hat{\boldsymbol{x}}_L^j - \Delta\bar{\boldsymbol{x}}^j\|/\|\Delta\bar{\boldsymbol{x}}^j\|$*.*

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 LSPG responses for basis dimension $p = 204$

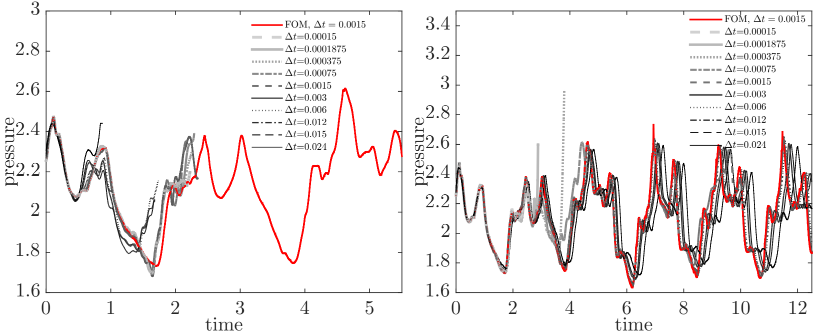


(a) Galerkin (b) LSPG

- Galerkin ROMs unstable for long time intervals

\+ LSPG ROMs accurate and stable (most time steps)

Galerkin and LSPG responses for basis dimension $p = 564$

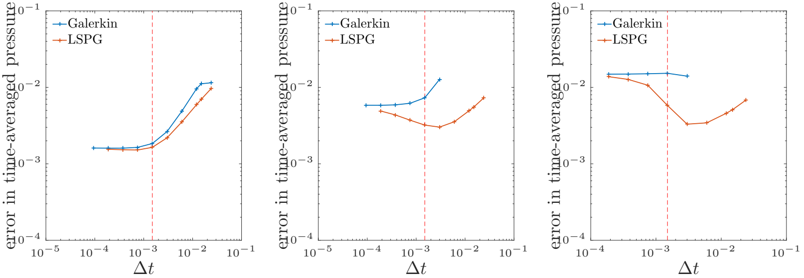


(c) Galerkin (d) LSPG

- Galerkin ROMs remain unstable

+ LSPG ROMs improve

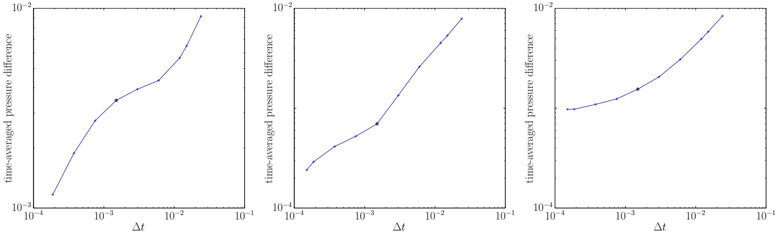
LSPG ROM: superior performance ($p = 204$)



(e) $0 \leq t \leq 0.55$ (f) $0 \leq t \leq 1.1$ (g) $0 \leq t \leq 1.54$

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

Limiting equivalence

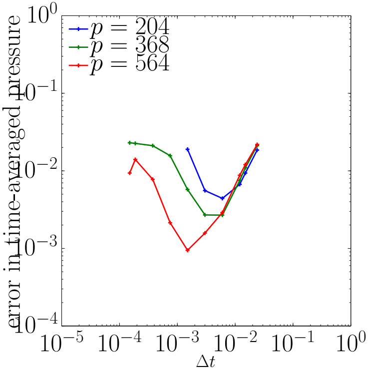


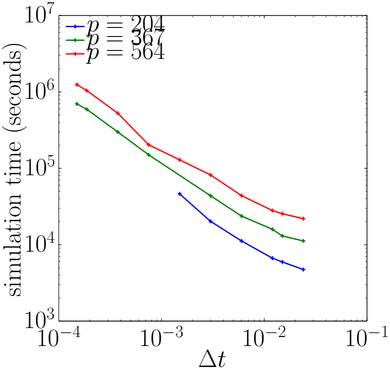
(h) $p = 204$ (i) $p = 368$ (j) $p = 564$

Galerkin/LSPG difference in the stable Galerkin interval $0 \leq t \leq 1.1$.

✓ The LSPG ROM converges to Galerkin as $\Delta t \rightarrow 0$.

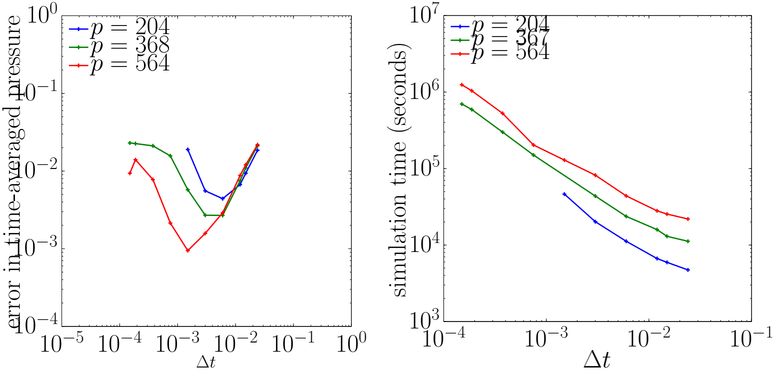
LSPG performance ($t \leq 12.5$ sec)





✓ An intermediate Δt produces the lowest error and better speedup.

LSPG 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

- *Galerkin*: projection and time-discretization are commutative
- *LSPG*: a continuous representation sometimes exists
- 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 Δt
- Numerical experiments
  - LSPG ROM yields a smaller error than Galerkin
  - Equivalent as $\Delta t \to 0$
  - Error minimized for intermediate Δt
- **Reference**: C., Barone, and Antil. Galerkin v. least-squares Petrov–Galerkin projection in nonlinear model reduction. *arXiv e-print*, (1504.03749), 2015.

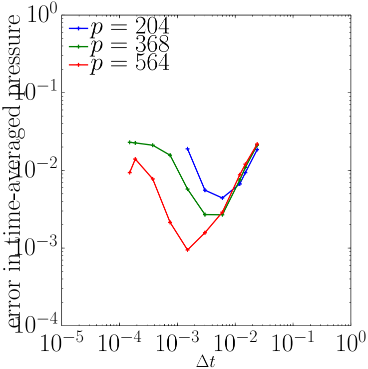
Our research goal

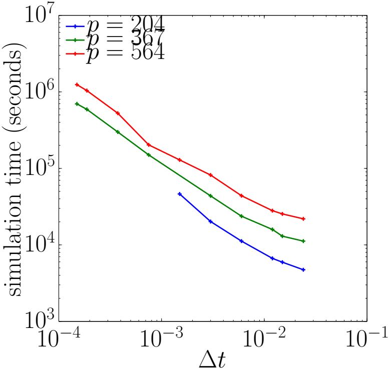
Nonlinear model-reduction methods that are
accurate, low cost, certified, and reliable.

- + Accuracy
  - Improve projection technique [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]

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]

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

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

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

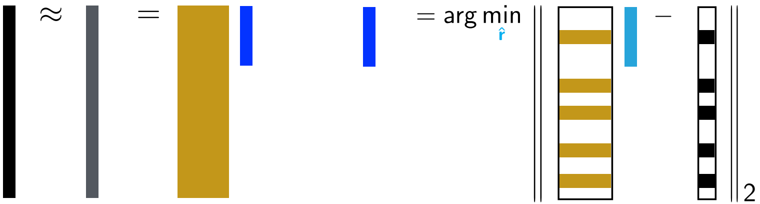
Can we select $\boldsymbol{A}$ *to make this inexpensive?*

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

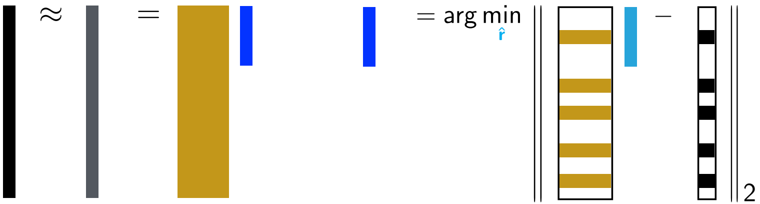


Hyper-reduction via Gappy POD [Everson and Sirovich, 1995]

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

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. $\hat{\boldsymbol{r}}^n(\boldsymbol{x}) = \arg\min_{\hat{\boldsymbol{r}}} \left\|\boldsymbol{P}\boldsymbol{\Phi}_R\hat{\boldsymbol{r}} - \boldsymbol{P}\boldsymbol{r}^n(\boldsymbol{x})\right\|_2$



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

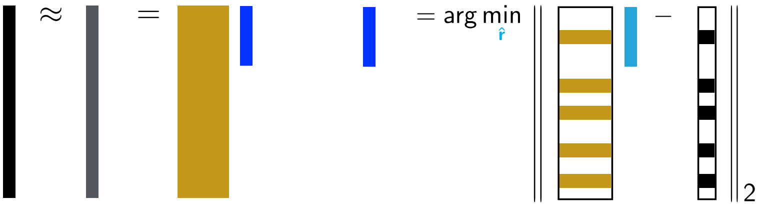
$$= \arg\min_{\hat{\boldsymbol{z}}\in\mathbb{R}^p} \left\|\underbrace{\left(\boldsymbol{P}\boldsymbol{\Phi}_R\right)^+\boldsymbol{P}}_{\boldsymbol{A}}\boldsymbol{r}^n\left(\boldsymbol{\Phi}\hat{\boldsymbol{z}}\right)\right\|_2^2.$$

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

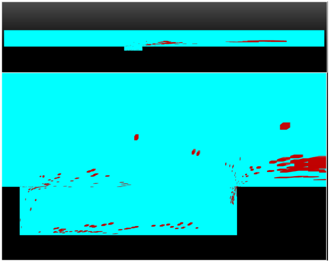
$$\hat{\boldsymbol{x}}^n = \arg\min_{\hat{\boldsymbol{z}}\in\mathbb{R}^p} \| (\boldsymbol{P}\boldsymbol{\Phi}_R)^+ \boldsymbol{P}\boldsymbol{r}^n (\boldsymbol{\Phi}\hat{\boldsymbol{z}}) \|_2^2$$

- *Key*: GNAT samples only a few entries of the residual $\boldsymbol{P}\boldsymbol{r}^n$
- *Idea*: Extract minimal subset of the mesh

Sample mesh: HPC implementation

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

- *Key*: GNAT samples only a few entries of the residual $\boldsymbol{P}\boldsymbol{r}^n$
- *Idea*: Extract minimal subset of the mesh



- Sample mesh: 4.1% nodes, 3.0% cells

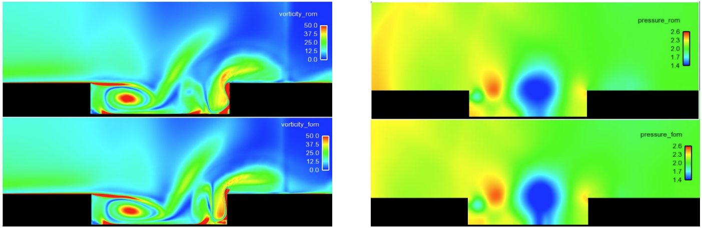
\+ Small problem size: can run on many fewer cores

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 [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]

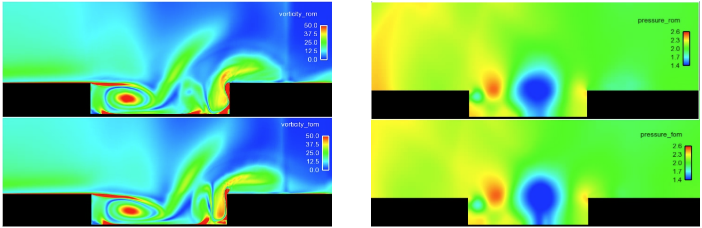
Collaborators: L. Brencher, B. Haasdonk, A. Barth (U Stuttgart)

GNAT performance

vorticity field | pressure field

GNAT ROM

FOM



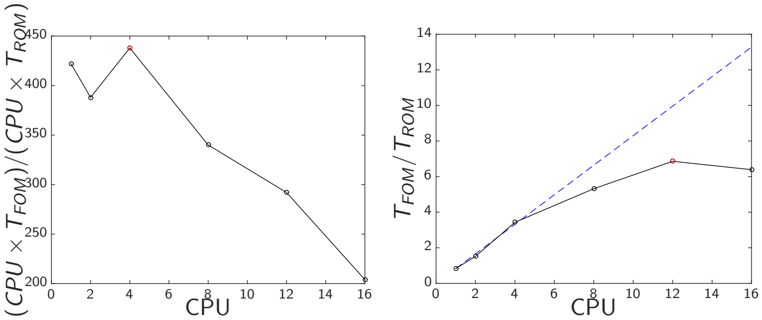
- FOM: 5 hour x 48 CPU
- GNAT ROM: 32 min x 2 CPU.

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

- 9.4x walltime savings. Bad for real time.

Why?

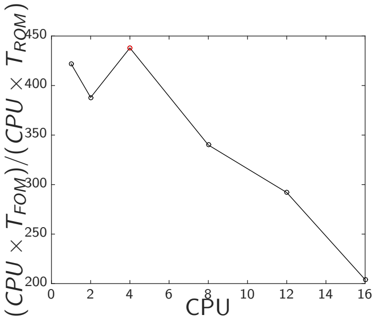
GNAT: strong scaling (Ahmed body) [C., 2011]



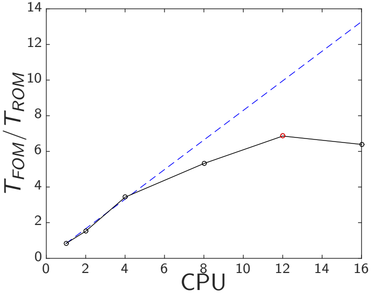
(g) CPU-hour savings (h) Walltime savings

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

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



(i) CPU-hour savings



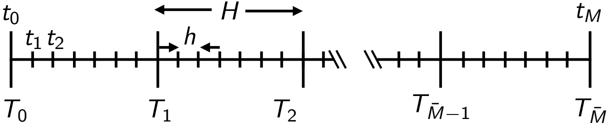
(j) Walltime savings

+ 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

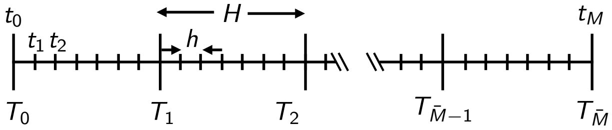
$$\mathcal{G}(\boldsymbol{x}; \tau_1, \tau_2)$$

- Parareal iteration k (sequential and parallel steps):

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

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

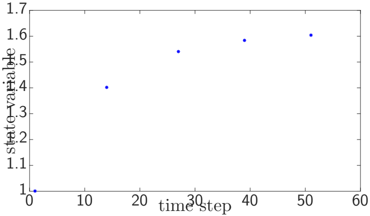
$$\mathcal{G}(\boldsymbol{x}; \tau_1, \tau_2)$$

- Parareal iteration k (sequential and parallel steps):

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

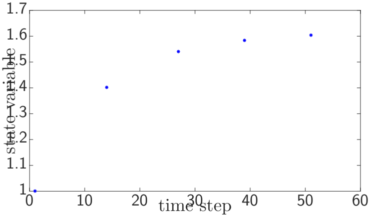
- Interpretations [Gander and Vandewalle, 2007, Falgout et al., 2014]:
  - Deferred/residual-correction scheme $\mathcal{B}(\boldsymbol{x}_{k+1}) = \mathcal{B}(\boldsymbol{x}_k) - \mathcal{A}(\boldsymbol{x}_k)$
  - Multiple shooting method with FD Jacobian approximation
  - Two-level multigrid

Parareal: sequential and parallel steps [Lions et al., 2001a]

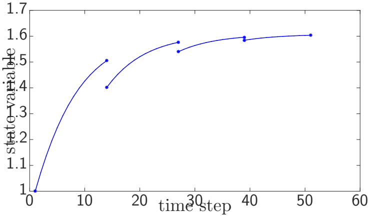


$$\boldsymbol{x}_0^{m+1} = \mathcal{G}(\boldsymbol{x}_0^m; T_m, T_{m+1})$$

Parareal: sequential and parallel steps [Lions et al., 2001a]

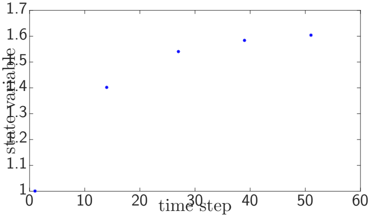


$$\mathbf{x}_0^{m+1} = \mathcal{G}(\mathbf{x}_0^m; T_m, T_{m+1})$$

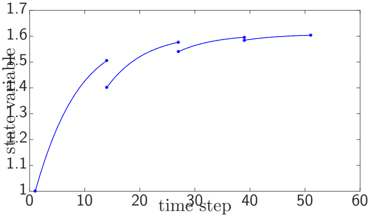


$$\mathcal{F}(\mathbf{x}_0^m; T_m, T_{m+1})$$

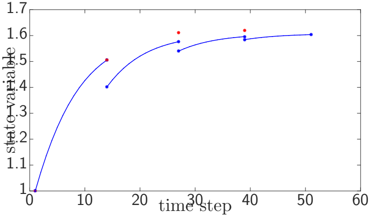
Parareal: sequential and parallel steps [Lions et al., 2001a]



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

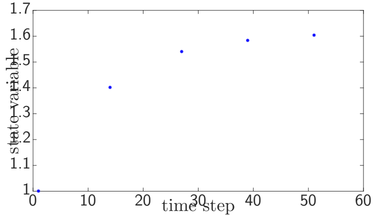


$$\mathcal{F}(\boldsymbol{x}_0^m; T_m, T_{m+1})$$

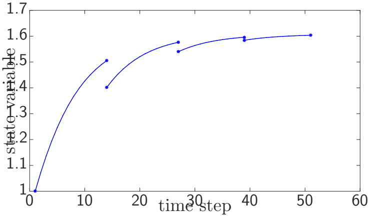


$$\boldsymbol{x}_1^{m+1} = \mathcal{F}(\boldsymbol{x}_0^m; T_m, T_{m+1}) + \mathcal{G}(\boldsymbol{x}_1^m; T_m, T_{m+1}) - \mathcal{G}(\boldsymbol{x}_0^m; T_m, T_{m+1})$$

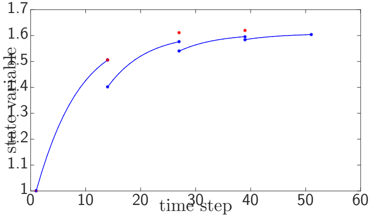
Parareal: sequential and parallel steps [Lions et al., 2001a]



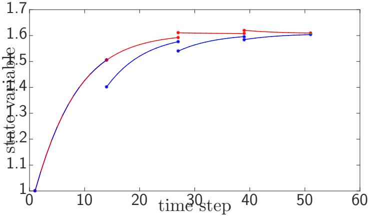
$$\mathbf{x}_0^{m+1} = \mathcal{G}(\mathbf{x}_0^m; T_m, T_{m+1})$$



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



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



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

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 physics 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]

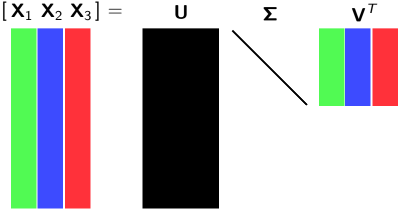
Coarse propagator

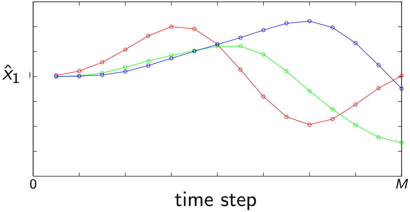
Critical: coarse propagator should be fast, accurate, stable

- 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 physics 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]

ROM context: can we leverage offline data to improve the coarse propagator?

Revisit the SVD

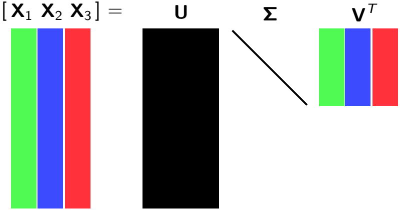


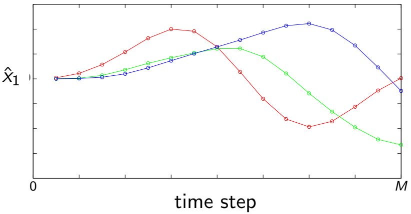


First row of $\boldsymbol{V}^T$

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

Revisit the SVD





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

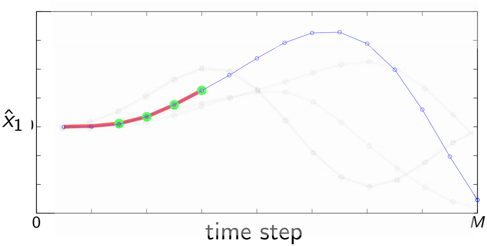
jth row of $\boldsymbol{V}^T$ contains a basis for time evolution of $\hat{x}_j$

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

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

First attempt [C. et al., 2015b]

1. compute **global forecast** by gappy POD in time domain:



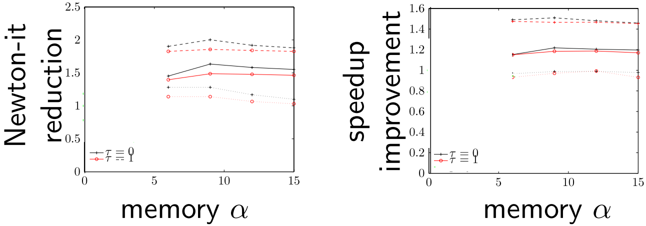
$\hat{\boldsymbol{x}}_1$ so far; memory $\alpha = 4$; forecast; temporal basis

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

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

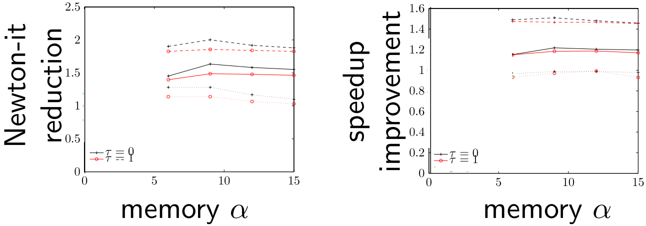
2. use $\boldsymbol{e}_m^T \boldsymbol{\Xi}_j \boldsymbol{z}_j$ as initial guess for $\hat{x}_j(t_m)$ in Newton solver

First attempt: 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

First attempt: structural dynamics [C. et al., 2015b]

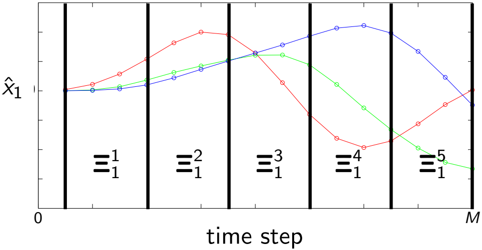


- $+$ Newton iterations reduced by up to ∼2x
- $+$ Speedup improved by up to ∼1.5x
- $+$ No accuracy loss
- $+$ Applicable to any nonlinear ROM
- \- Insufficient for real-time computation

Can we apply the same idea for the coarse propagator?

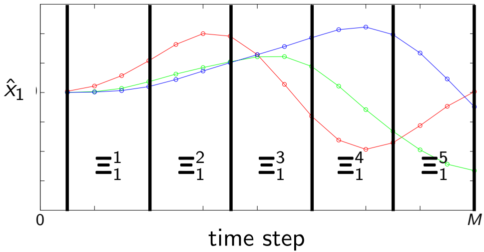
Coarse propagator via local forecasting

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



Coarse propagator via local forecasting

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

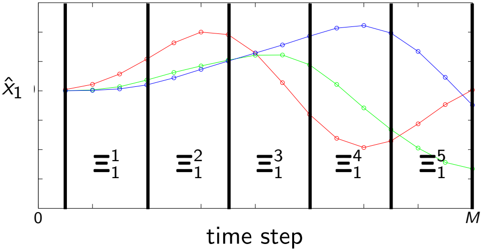


- **Online**: Coarse propagator $\mathcal{G}_j^m$ defined via forecasting:
  1. Compute α time steps with fine propagator

$$\begin{bmatrix} \mathcal{F}(\hat{\mathbf{x}}_j; T_m, T_m + \Delta t) \\ \vdots \\ \mathcal{F}(\hat{\mathbf{x}}_j; T_m, T_m + \Delta t\alpha) \end{bmatrix}$$

Coarse propagator via local forecasting

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

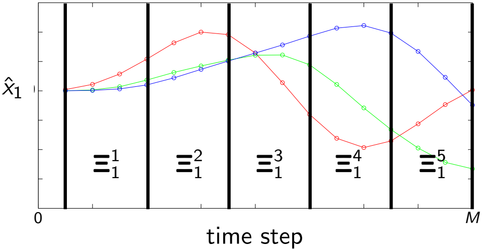


- **Online**: Coarse propagator $\mathcal{G}_j^m$ defined via forecasting:
  1. Compute α time steps with fine propagator
  2. Compute **local forecast** via gappy POD

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

Coarse propagator via local forecasting

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



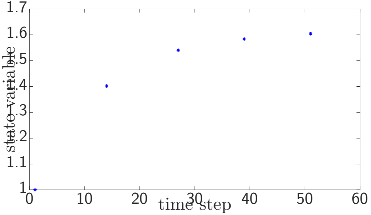
- **Online**: Coarse propagator $\mathcal{G}_j^m$ defined via forecasting:
  1. Compute α time steps with fine propagator
  2. Compute **local forecast** via gappy POD
  3. Select last timestep of **local forecast**

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

Initial seed

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

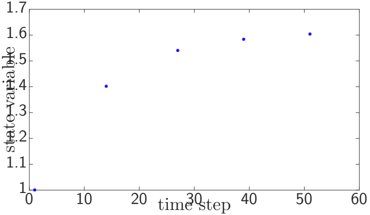
How to compute initial seed $\mathbf{x}_0^m$, $m = 0, \dots, \bar{M}$?



Initial seed

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

How to compute initial seed $\mathbf{x}_0^m$, $m = 0, \dots, \bar{M}$?



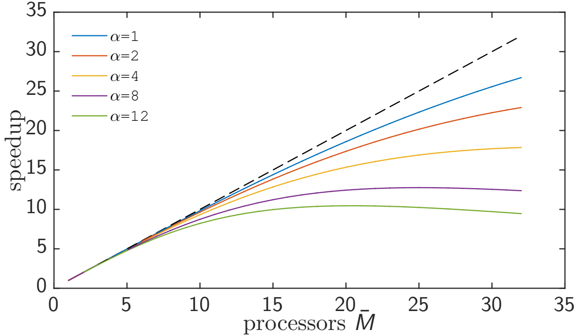
1. Typical time integrator
2. Local forecast
3. Global forecast

Ideal-conditions speedup

Theorem

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

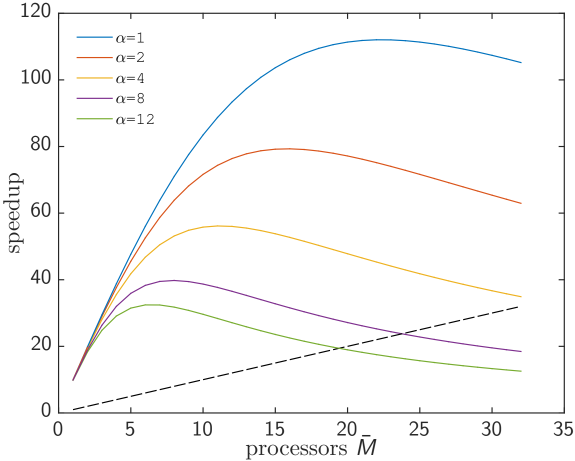
The method then realizes a theoretical speedup of

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

relative to the sequential algorithm without forecasting. Here,

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

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.,

$$\|\mathcal{F}(\boldsymbol{x};\tau,\tau+\Delta T)\| \leq (1+C_{\mathcal{F}}\Delta T)\|\boldsymbol{x}\|, \quad \forall 0\leq\tau\leq\tau+\Delta T$$

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

$$\|\hat{\boldsymbol{x}}_{k+1}^{m}\| \leq C_m \exp(C_{\mathcal{F}} m\Delta T)\|\hat{\boldsymbol{x}}^{0}\|.$$

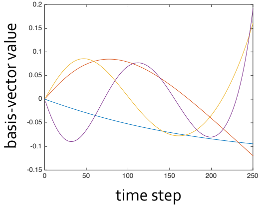
- $C_m := \sum_{k=1}^{m}\binom{k}{m}\beta_k\gamma^m\alpha^k(\Delta T/\Delta t)^{m-k}$
- $\beta_k := \exp(-C_{\mathcal{F}}k(\Delta T-\Delta t\alpha)) \leq 1$
- $\gamma := \max(\max_{m,j} 1/\|\boldsymbol{Z}(\alpha+1,\alpha)\boldsymbol{\Xi}_j^m\|, 1/\sigma_{\min}(\boldsymbol{Z}(\alpha+1,\alpha)\boldsymbol{\Xi}_j^m))$

Example: inviscid Burgers equation [Rewienski, 2003]

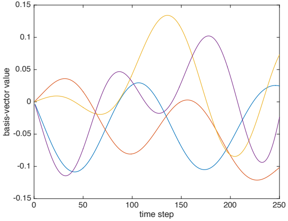
$$\frac{\partial u(x,\tau)}{\partial \tau} + \frac{1}{2}\frac{\partial\left(u^2\left(x,\tau\right)\right)}{\partial x} = 0.02e^{\mu_2 x}$$
$$u(0,\tau) = \mu_1,\ \forall \tau \in [0, 25]$$
$$u(x,0) = 1,\ \forall x \in [0, 100]\,,$$

- Discretization: Godunov's scheme
- $(\mu_1, \mu_2) \in [2.5, 3.5] \times [0.02, 0.03]$
- $\Delta t = 0.1$, $M = 250$ fine time steps
- FOM: $N = 500$ degrees of freedom
- ROM: LSPG [C. et al., 2011a], POD basis dimension $p = 100$
- $n_{\text{train}} = 4$ training points (LHS sampling); random online point
- **2 coarse propagators**: Backward Euler and local forecast
- **3 initial seeds**: Backward Euler, local forecast, global forecast

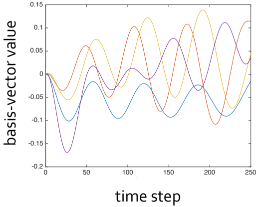
Global temporal bases



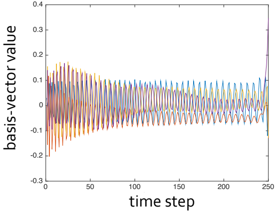
(a) coordinate 1



(b) coordinate 5



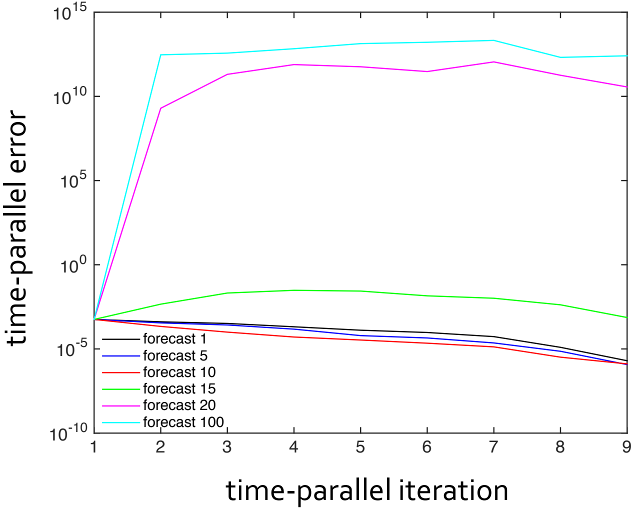
(c) coordinate 10



(d) coordinate 100

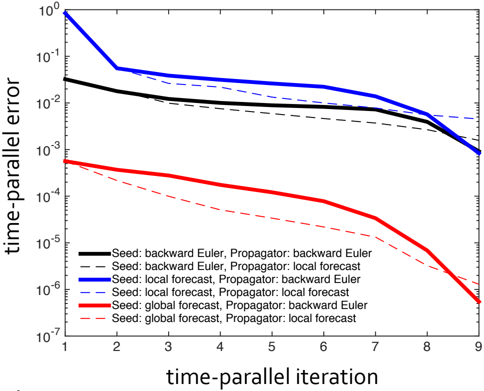
Higher-index generalized coordinates not 'forecastable'

Forecasting ‘high-frequency’ coordinates is dangerous



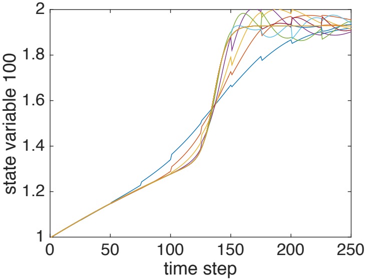
Proceed by forecasting the first 10 coordinates

Comparison: Initial seed and coarse propagator

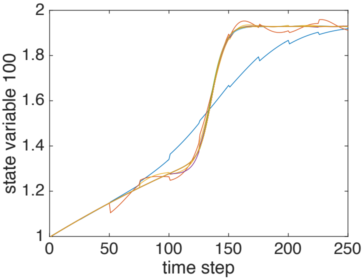


- Initial seed:
  + best performance: global forecast
  - worst performance: local forecast (error accumulation)
- Coarse propagator:
  + local forecast outperforms backward Euler

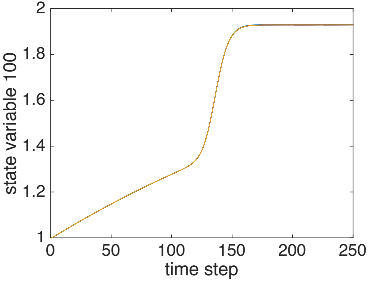
Forecasting improves improves initial seed and coarse propagator!



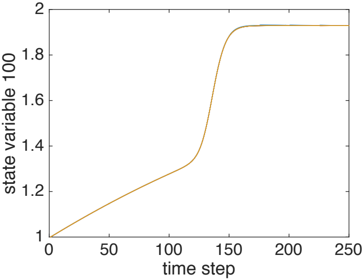
(e) Seed: Euler, Prop: Euler



(f) Seed: Euler, Prop: local forecast

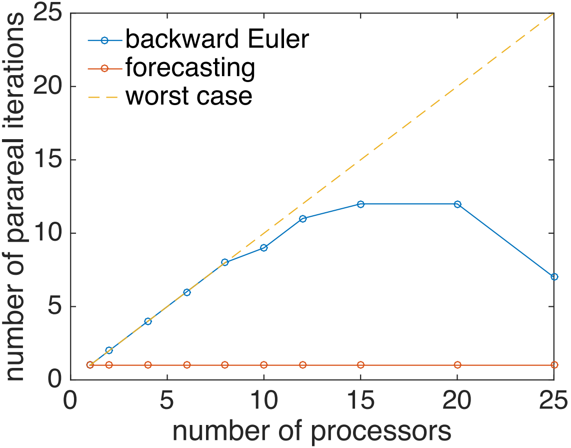


(g) Seed: glob forecast, Prop: Euler



(h) Seed: glob forecast, Prop: loc fore-

Parareal performance



+ *Forecasting*: minimum possible iterations
- *Backward Euler*: often close to worst-case performance

Conclusions

Use temporal data to reduce ROM simulation time

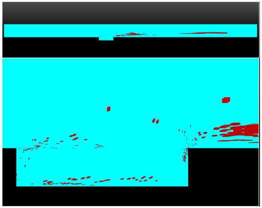
- **offline**: time-evolution bases from right singular vectors
- **online**:
  1. global forecast as initial seed
  2. local forecast as coarse propagator

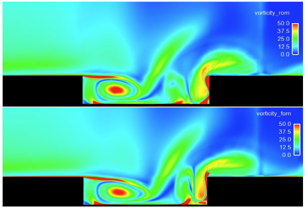
+ theory: excellent speedup and stability
+ ideal parareal performance observed
+ significant improvement over Backward Euler
+ no additional error introduced

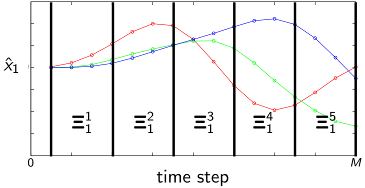
- **References**:

  - K. Carlberg, L. Brencher, B. Haasdonk, and A. Barth. "Data-driven time parallelism with application to reduced-order models," *in preparation*.
  - K. Carlberg, J. Ray, and B. van Bloemen Waanders. "Decreasing the temporal complexity for nonlinear, implicit reduced-order models by forecasting," CMAME, Vol. 289, p. 79–103 (2015).

Questions?




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


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

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.

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.

Falgout, R. D., Freidhoff, S., Kolev, T. V., MacLachlan, S. P., and Schroder, J. B. (2014).
Parallel time integration with multigrid.
SIAM J. Sci. Comput., 36(6):C635–C661.

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.

Gander, M. and Vandewalle, S. (2007).
Analysis of the parareal time-parallel time-integration method.
SIAM Journal on Scientific Computing, 29(2):556–578.

Gugercin, S., Antoulas, A. C., and Beattie, C. (2008).
h_2 model reduction for large-scale linear dynamical systems.
SIAM J. Matrix Anal. Appl., 30(2):609–638.

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.

Lattimer, A. (2016).
Model reduction of nonlinear fire dynamics models.
PhD thesis, Virginia Tech.

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.

Ma, Z., Ahuja, S., and Rowley, C. W. (2011).
Reduced-order models for control of fluids using the eigensystem realization algorithm.
Theoretical and Computational Fluid Dynamics, 25(1-4):233–247.

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.

Or, A. and Speyer, J. (2010).
Empirical pseudo-balanced model reduction and feedback control of weakly nonlinear convection patterns.
Journal of Fluid Mechanics, 662:36.

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 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.

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.