SAND2016-2344C

Data-driven time parallelism and model reduction

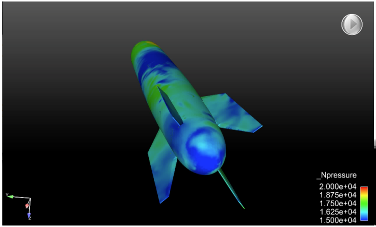
Kevin Carlberg[1], Lukas Brencher[2],
Bernard Haasdonk[2]

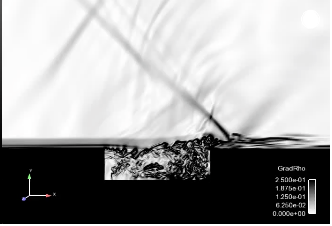
Sandia National Laboratories[1]

University of Stuttgart[2]

SIAM Conference on UQ
April 7, 2016

Model reduction at Sandia



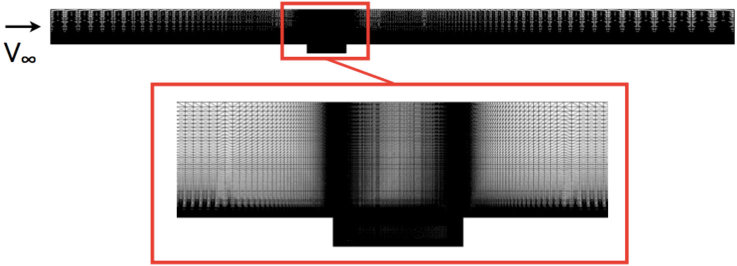


- CFD model
  - 100 million cells
  - 200,000 time steps
- High simulation costs
  - 6 weeks, 5000 cores
  - 6 runs maxes out Cielo

Barrier

- Real time (rapid design)
- Many query (UQ)

Cavity-flow problem

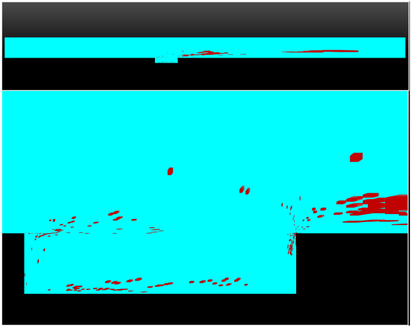


- Unsteady Navier–Stokes
- DES turbulence model
- 1.2 million degrees of freedom

- $Re = 6.3 \times 10^6$
- $M_\infty = 0.6$
- CFD code: AERO-F [Farhat et al., 2003]

GNAT model [C. et al., 2011, C. et al., 2013]

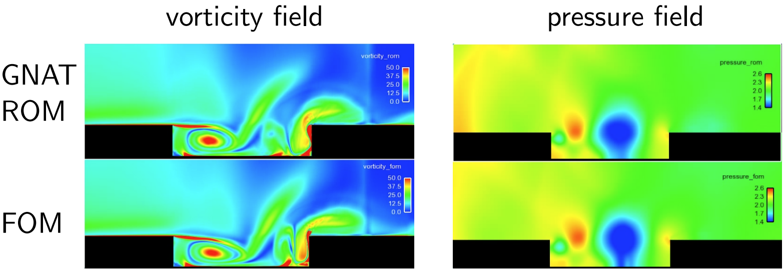
$$\hat{\boldsymbol{x}}^n = \arg\min_{\hat{\boldsymbol{z}}\in\mathbb{R}^{\hat{N}}} \| (\boldsymbol{P}\Phi_R)^+ \boldsymbol{P}\boldsymbol{r}^n(\Phi\hat{\boldsymbol{z}}) \|_2^2$$



- Sample mesh: 4.1% nodes, 3.0% cells

+ Small problem size: can run on many fewer cores

GNAT performance



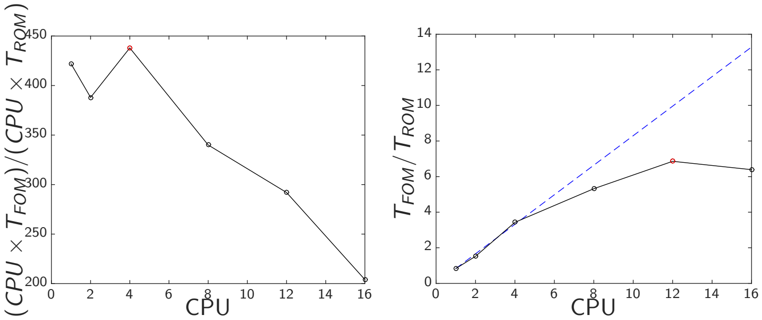
- FOM: 5 hour x 48 CPU
- GNAT ROM: 32 min x 2 CPU.

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

- 9.4x walltime savings. Bad for real time.

Why?

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



(a) CPU-hour savings

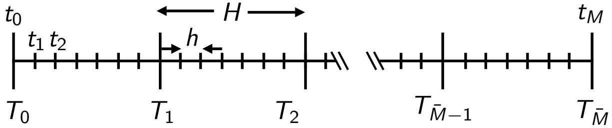
(b) Walltime savings

+ 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 h

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

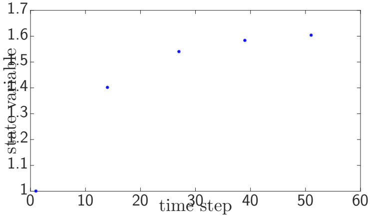
- Coarse propagator: time step H

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

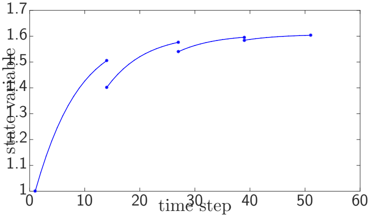
- Parareal iteration k (sequential and parallel steps):

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

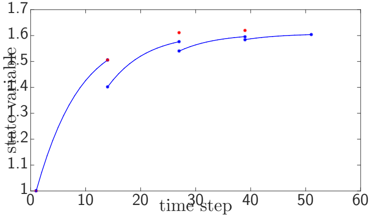
Illustration: sequential and parallel steps



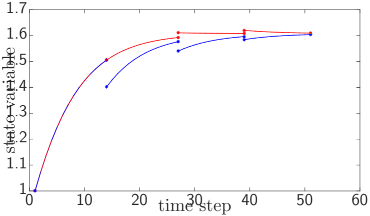
$$\mathbf{x}_0^{m+1} = \mathcal{G}(\mathbf{x}_0^m; T_m, T_{m+1})$$



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



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



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

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]

Can we leverage offline data to improve the coarse propagator?

Model reduction

- full-order model (FOM)

$$\boxed{\dot{\boldsymbol{x}}(t,\boldsymbol{p}) = \boldsymbol{f}(\boldsymbol{x};t,\boldsymbol{p}), \quad \boldsymbol{x}(0,\boldsymbol{p}) = \boldsymbol{x}^0(\boldsymbol{p})}$$

- **Offline**: snapshot collection

$$\boldsymbol{X}_i := [\boldsymbol{x}(0,\boldsymbol{p}_i) \ \cdots \ \boldsymbol{x}(t_M,\boldsymbol{p}_i)] \in \mathbb{R}^{N\times M}$$

$$\left[\boldsymbol{X}_1 \ \cdots \ \boldsymbol{X}_{n_{\text{train}}}\right] = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$$

- **Online**: projection
  - trial subspace $\boldsymbol{\Phi} = \left[\boldsymbol{u}_1 \ \cdots \ \boldsymbol{u}_{\hat{N}}\right] \in \mathbb{R}^{N\times\hat{N}}$

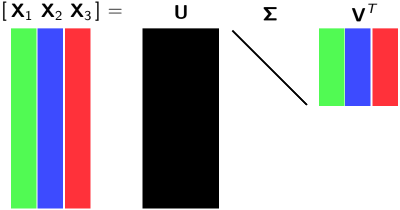
$$\boldsymbol{x} \approx \tilde{\boldsymbol{x}}(t,\boldsymbol{p}) = \boldsymbol{\Phi}\hat{\boldsymbol{x}}(t,\boldsymbol{p})$$

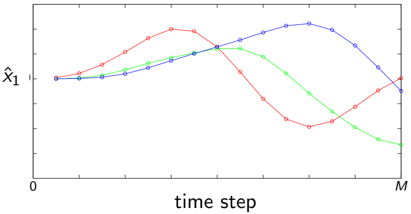
  - test subspace $\boldsymbol{\Psi} \in \mathbb{R}^{N\times\hat{N}}$
    - $\boldsymbol{\Psi} = \boldsymbol{\Phi}$: Galerkin
    - $\boldsymbol{\Psi} = (\alpha_o \boldsymbol{I} - \delta t \beta_0 \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}})\boldsymbol{\Phi}$: LSPG

[C. et al., 2015a]

$$\boxed{\dot{\hat{\boldsymbol{x}}}(t,\boldsymbol{p}) = (\boldsymbol{\Psi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Psi}^T\boldsymbol{f}(\boldsymbol{\Phi}\hat{\boldsymbol{x}};t,\boldsymbol{p}), \quad \hat{\boldsymbol{x}}(0,\boldsymbol{p}) = \boldsymbol{\Phi}^T\boldsymbol{x}^0(\boldsymbol{p})}$$

Revisit the SVD





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

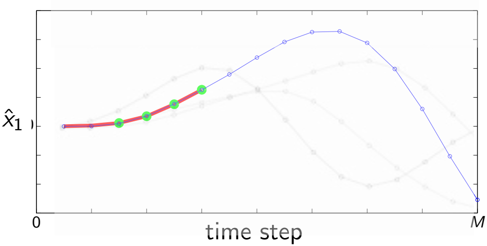
jth row of $\boldsymbol{V}^T$ contains a basis for time evolution of $\hat{x}_j$

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

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

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

1. compute forecast by gappy POD in time domain:



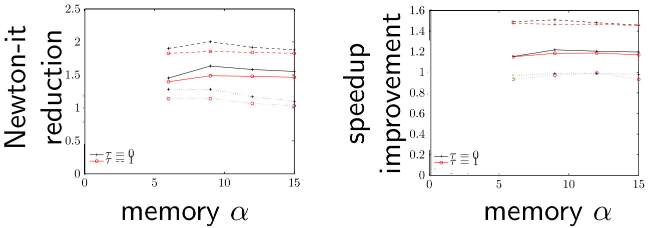
$\hat{\boldsymbol{x}}_1$ so far; memory $\alpha = 4$; forecast; temporal basis

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

- 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

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

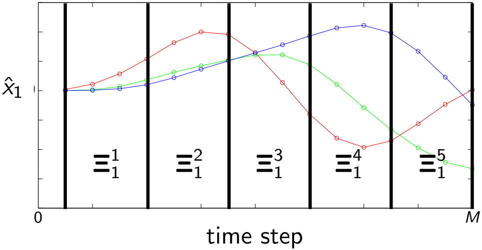


+ Newton iterations reduced by up to ∼2x
+ Speedup improved by up to ∼1.5x
+ No accuracy loss
+ Applicable to any nonlinear ROM
- Insufficient for real-time computation

Can we apply the same idea for the coarse propagator?

Coarse propagator 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

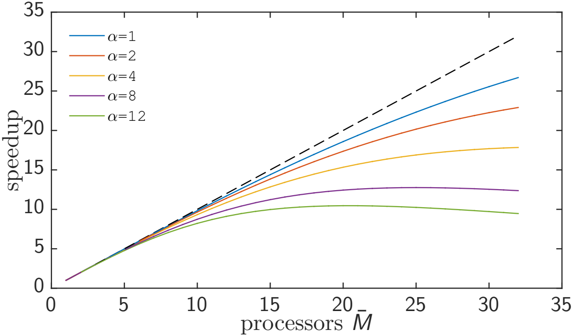
$$\mathcal{G}_j^m : (\hat{\boldsymbol{x}}_j; T_m, T_{m+1}) \mapsto \boldsymbol{e}_{H/h}^T \boldsymbol{\Xi}_j^m \left[\boldsymbol{Z}(\alpha+1,\alpha)\boldsymbol{\Xi}_j^m\right]^+ \begin{bmatrix} \mathcal{F}(\hat{\boldsymbol{x}}_j; T_m, T_m + h) \\ \vdots \\ \mathcal{F}(\hat{\boldsymbol{x}}_j; T_m, T_m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

Ideal-conditions speedup

Theorem

If $g(\hat{x}_j) \in \text{range}(\boldsymbol{\Xi}_j), \quad j = 1, \ldots, \hat{N}$, 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,* $g(\hat{x}_j) \in range(\boldsymbol{\Xi}_j), \quad j = 1, \ldots, \hat{N}$*, 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.*

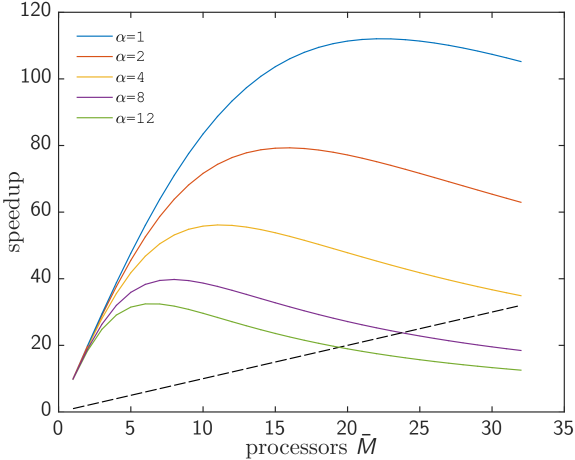
The method then realizes a theoretical speedup of

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

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.,

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

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

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

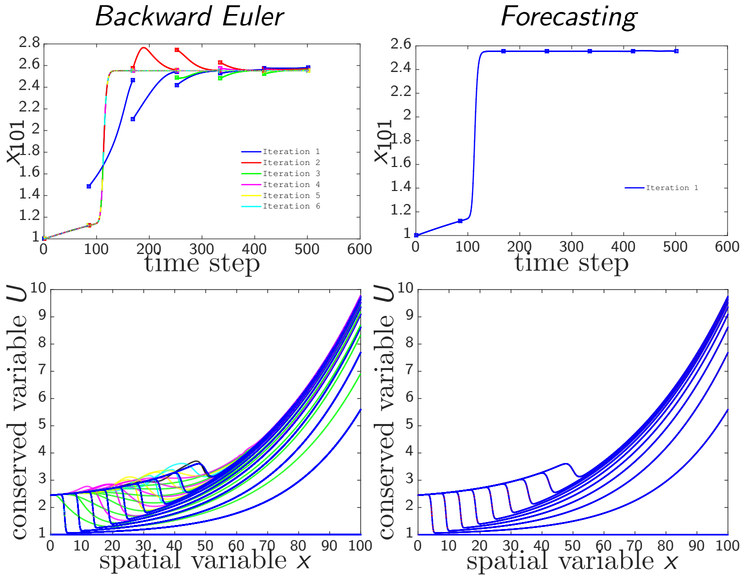
- $C_m := \sum_{k=1}^{m} \binom{k}{m} \beta_k \gamma^m \alpha^k (H/h)^{m-k}$
- $\beta_k := \exp(-C_{\mathcal{F}} k(H - h\alpha)) \leq 1$
- $\gamma := \max(\max_{m,j} 1/\|\boldsymbol{Z}(\alpha+1,\alpha)\boldsymbol{\Xi}_j^m\|, 1/\sigma_{\min}(\boldsymbol{Z}(\alpha+1,\alpha)\boldsymbol{\Xi}_j^m))$

Example: inviscid Burgers equation [Rewienski, 2003]

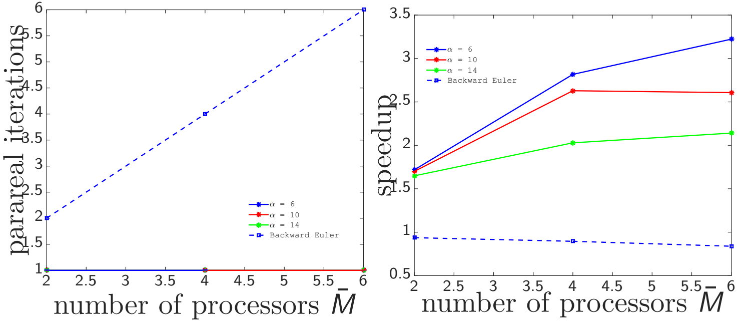
$$\frac{\partial u(x,\tau)}{\partial \tau} + \frac{1}{2}\frac{\partial \left(u^2(x,\tau)\right)}{\partial x} = 0.02e^{p_2 x}$$
$$u(0,\tau) = p_1,\ \forall \tau \in [0, 25]$$
$$u(x,0) = 1,\ \forall x \in [0, 100],$$

- Discretization: Godunov's scheme
- $(p_1, p_2) \in [2.5, 3.5] \times [0.02, 0.075]$
- $h = 0.1$, $M = 250$ fine time steps
- FOM: $N = 500$ degrees of freedom
- ROM: LSPG [C. et al., 2011] with POD basis dimension $\hat{N} = 100$
- $n_{\text{train}} = 4$ 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!

Conclusions

Use temporal data to reduce ROM simulation time

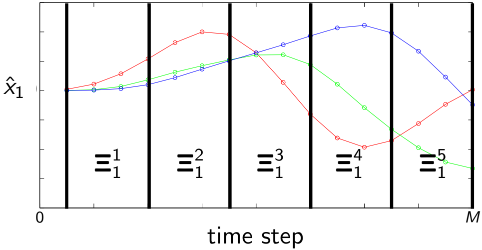
- **offline**: time-evolution bases from right singular vectors
- **online**: use as coarse propagator
  1. compute α time steps with fine propagator
  2. use gappy POD to forecast

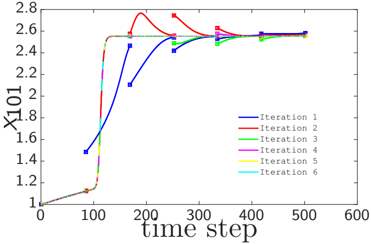
+ theory: excellent speedup and stability
+ ideal parareal performance observed
+ significant improvement over Backward Euler
+ no additional error introduced
+ generally applicable

- **References**:

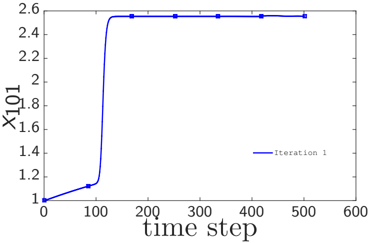
  - K. Carlberg, L. Brencher, B. Haasdonk, and A. Barth. "Time-parallel reduced-order models via forecasting," *in preparation*.
  - K. Carlberg, J. Ray, and B. van Bloemen Waanders. "Decreasing the temporal complexity for nonlinear, implicit reduced-order models by forecasting," CMAME, Vol. 289, p. 79–103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

Questions?





Backward Euler



Forecasting

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.

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

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.

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.

C., K. (2011).
Model Reduction of Nonlinear Mechanical Systems via Optimal Projection and Tensor Approximation.

PhD thesis, Stanford University.

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. (2011).
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., 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.

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.

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

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.

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.

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

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

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