Time-parallel reduced-order models via forecasting

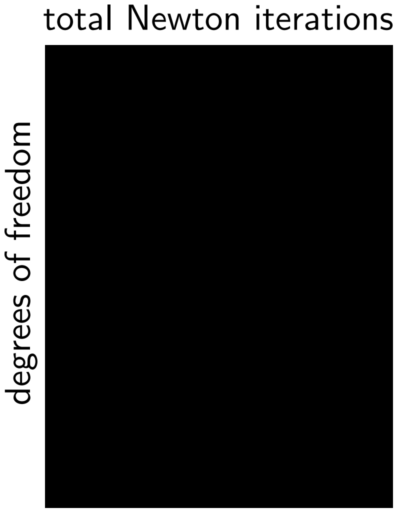
Kevin Carlberg[1], Lukas Brencher[2], Bernard Haasdonk[2]

Sandia National Laboratories[1]

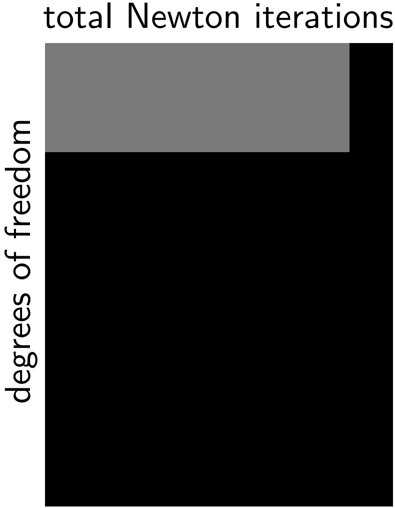
University of Stuttgart[2]

MoRePaS III
October 14, 2015

Nonlinear ODE, implicit time integration



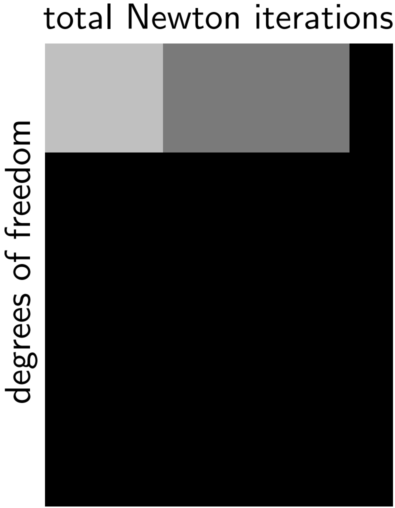
Reduced-order model (ROM): computed unknowns



Exploit *spatial-behavior* data to decrease # unknowns.

Can we do more?

Goal



Exploit *temporal-behavior* data to decrease total Newton iterations.

Main idea

- full-order model
  - 1st- or 2nd-order nonlinear ODE
  - implicit time integrator
- computational complexity
  - each time step, solve a large-scale system of nonlinear equations with a Newton-like method
  - **spatial complexity**: cost of each Newton iteration (i.e., linear-system solve)
  - **temporal complexity**: number of Newton iterations
- ROM: use *spatial-behavior* data to decrease *spatial complexity*
- goals
  1. exploit *temporal-behavior* data to decrease *temporal complexity*
  2. introduce *no additonal error* to ROM solution

Outline

Table of Contents

Parameterized, nonlinear ODE

- for simplicity, consider first-order ODEs

$$\dot{x} = f(x; t, \mu)$$
$$x(0; \mu) = x^0(\mu)$$

- state: $x \equiv x(t; \mu) \in \mathbb{R}^N$
- f nonlinear in x
- inputs: $\mu \in \mathcal{D}$
- initial condition: $x^0(\mu) \in \mathbb{R}^N$

Implicit time integration

- for simplicity, consider only single-stage methods
- system of nonlinear equations solved at each time step:

$$R^n\left(w^n;\mu\right) = 0, \quad n = 1, \ldots, M$$

- unknowns w^n: state or velocity at $t \in \left[t^{n-1}, t^n\right]$
- after computing w^n, explicitly update the state:

$$x^n = \gamma x^{n-1} + \beta w^n$$

Full-order model: computational burden

Solve

$$R^n\left(w^n;\mu\right)=0,\quad n=1,\dots,M$$

with a Newton-like method

- solve one N-dimensional linear system per Newton iteration
- **spatial complexity**: cost of each linear-system solve
  - direct solver: $\mathcal{O}\left(\omega^2 N\right)$ flops[1]
  - iterative solver: $\mathcal{O}\left(L\omega N\right)$ flops[2]
  - N large $\rightarrow$ spatial complexity large
- **temporal complexity**: total number of Newton iterations
  - N large $\rightarrow$ M large $\rightarrow$ temporal complexity large

---

[1] ω: average number of nonzeros per row of $\frac{\partial R^n}{\partial w}$

[2] L: average number of linear-solver iterations per Newton iteration

Projection-based model reduction

- **Offline**: exploit knowledge of *spatial behavior* to compute basis $\Phi \in \mathbb{R}^{N\times \hat{N}}$ with $\hat{N} \ll N$ (e.g., POD)
- **Online**: approximate state by $\tilde{x}$ in low-dim trial subspace:

$$\tilde{x}(t;\mu) = x^0(\mu) + \Phi\hat{x}(t;\mu) \quad (1)$$

$$\dot{\tilde{x}}(t;\mu) = \Phi\dot{\hat{x}}(t;\mu) \quad (2)$$

- substituting (1)–(2) into ODE (with $x = \tilde{x}$) yields

$$\Phi\dot{\hat{x}} = f\left(x^0(\mu) + \Phi\hat{x}; t, \mu\right). \quad (3)$$

- ODE (3) may not be solvable, because $\text{image}(f) \not\subset \text{range}(\Phi)$

Project, then discretize in time

Goal: compute solution to overdetermined ODE

- enforce orthogonality of ODE residual to range of $\Psi \in \mathbb{R}^{N\times\hat{N}}$

$$\Psi^T \Phi \dot{\hat{x}} = \Psi^T f\left(x^0(\mu) + \Phi\hat{x}; t, \mu\right)$$

$$\dot{\hat{x}} = \left(\Psi^T \Phi\right)^{-1} \Psi^T f\left(x^0(\mu) + \Phi\hat{x}; t, \mu\right) \tag{4}$$

- solve (4) with the same implicit numerical integrator

$$\left(\Psi^T \Phi\right)^{-1} \Psi^T R^n\left(w^0(\mu) + \Phi\hat{w}^n; \mu\right) = 0, \quad n = 1, \ldots, M$$

- $\hat{w}^n \in \mathbb{R}^{\hat{N}}$: generalized unknowns at time step n

Discretize in time, then project

Goal: compute solution to overdetermined ODE

1. apply time integrator to overdetermined ODE
2. minimize discrete residual over the trial subspace [LeGresley, 2006, Carlberg et al., 2011]

$$\hat{w}^n = \arg\min_{\mathrm{y}\in\mathbb{R}^{\hat{N}}} \left\| R^n \left(w^0 \left(\mu \right) + \Phi \mathrm{y}; \mu \right) \right\|^2$$

- solve with nonlinear least-squares method, e.g., Gauss–Newton

Problem: spatial complexity still scales with N

Hyperreduction

Goal: reduce spatial complexity by approximating nonlinear terms

- collocation [Astrid et al., 2008, Ryckelynck, 2005, LeGresley, 2006]

$$\tilde{R}^n = Z^T Z R^n$$

Z (sampling matrix): selected rows of $I_{N \times N}$

- empirical interpolation/gappy POD [Astrid et al., 2008, Bos et al., 2004, Chaturantabut and Sorensen, 2010, Galbally et al., 2009, Drohmann et al., 2012, Carlberg et al., 2011]

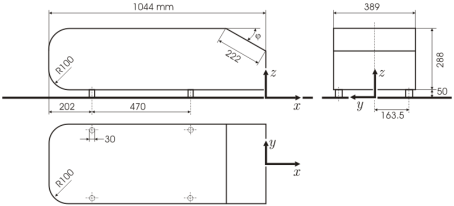
$$\tilde{f} = \Phi_f \left(Z \Phi_f \right)^+ Z f$$

or

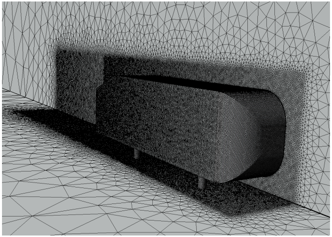
$$\tilde{R}^n = \Phi_R \left(Z \Phi_R \right)^+ Z R^n$$

Φ_R, Φ_f: bases that exploit observed spatial behavior

Example: Ahmed body



(a) Ahmed body [Hinterberger et al., 2004]



(b) mesh

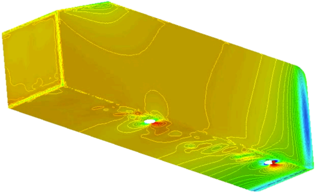
compressible Navier–Stokes (finite volume, AERO-F)

- DES turbulence model
- Re $= 4.48 \times 10^6$
- $M_\infty = 0.175$
- 3-point BDF integrator (implicit)
- FOM: $N = 1.73 \times 10^7$

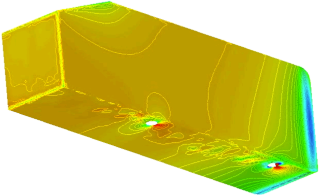
GNAT nonlinear ROM [Carlberg et al., 2011]

- discretize in time, then project (minimize discrete residual)
- hyperreduction: gappy POD applied to residual

Example: GNAT nonlinear model reduction [Carlberg et al., 2012]



(c) full-order (1.73×10^7 dofs)



(d) GNAT (283 dofs)

surface pressure at $t = 0.1$ seconds

model	*error in drag*	*cost, core-hours*	*Newton iterations per time step*
full-order model		6810	4.0
reduced-order model	0.68%	16	2.75

Complexity reduction

- **spatial complexity**: decreased by factor of **637**
- **temporal complexity**: decreased by factor of **1.5**

Can we do more?

Table of Contents

Overview

Goal: exploit temporal-behavior data to reduce temporal complexity

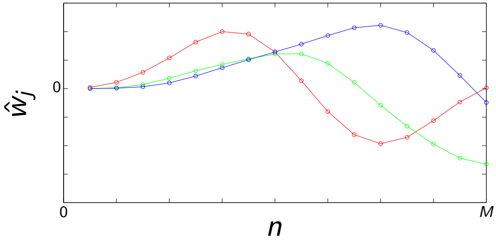
1. during ROM simulation, apply gappy POD *in the time domain* to generate a forecast for the generalized unknowns
2. use the forecast as an *accurate initial guess* for the Newton-like solver

+ good guess $\rightarrow$ few Newton its $\rightarrow$ low temporal complexity
+ introduces *no additional error*

Offline: compute time-evolution POD bases Ψ_j

1. collect snapshots of the *temporal behavior* of the jth generalized unknown:

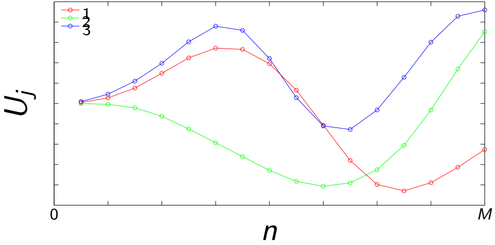
$$\hat{w}_j^n(\mu), \quad n = 1, \ldots, M, \ \mu \in \{\bar{\mu}_i\}_{i=1}^{n_{\text{train}}}$$



example with 3 training configurations ($n_{\text{train}} = 3$)

Offline: compute time-evolution POD bases Ψ_j

2 compute SVD of temporal-behavior snapshots



$$\begin{bmatrix} \hat{w}_j^1(\bar{\mu}_1) & \cdots & \hat{w}_j^1(\bar{\mu}_{n_{\text{train}}}) \\ \vdots & \ddots & \vdots \\ \hat{w}_j^M(\bar{\mu}_1) & \cdots & \hat{w}_j^M(\bar{\mu}_{n_{\text{train}}}) \end{bmatrix} = U_j \Sigma_j V_j^T$$

3 truncate: keep only $a_j \leq n_{\text{train}}$ vectors: $\Psi_j = U_j(:, 1 : a_j)$

Time-evolution bases: example

- implicit linear multi-step scheme: $w^n = x^n$
- one training configuration ($n_{\text{train}} = 1$)
- POD model reduction

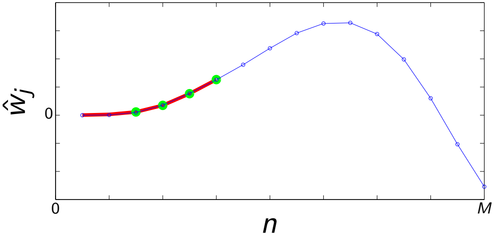
Here, the time-evolution bases Ψ_j are the *right singular vectors* generated when computing Φ:

$$\left[x^1(\bar{\mu}_1) \ \cdots \ x^M(\bar{\mu}_1)\right] = U\Sigma V^T$$

- $\Phi = U$
- $\Psi_j = V(:, j)$ for $j = 1, \ldots, M$

Online: compute forecast, use as initial guess

1. compute forecast by gappy POD in time domain: match generalized unknowns at previous α time steps



$\hat{w}_j$ so far; memory $\alpha = 4$; forecast

$$z_j = \arg\min_{z \in \mathbb{R}^{a_j}} \left\| \begin{bmatrix} \Psi_j(n-\alpha, 1) & \cdots & \Psi_j(n-\alpha, a_j) \\ \vdots & \ddots & \vdots \\ \Psi_j(n-1, 1) & \cdots & \Psi_j(n-1, a_j) \end{bmatrix} z - \begin{bmatrix} \hat{w}_j^{n-\alpha} \\ \vdots \\ \hat{w}_j^{n-1} \end{bmatrix} \right\|_2^2$$

2. use forecast $\Psi_j z_j$ as an *accurate initial guess* for Newton solver

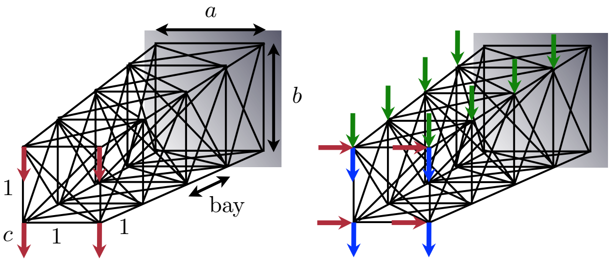
Online algorithm sketch

for $n = 1, \dots, M$ **do**
  if forecast is available **then**
    use forecast as initial guess for generalized unknowns
  end if
  solve reduced-order equations with a Newton-like method
  if # Newton iterations $> \tau$ **then** {recompute forecast}
    compute forecast using generalized unknowns at previous α time steps
  end if
end for

- many Newton iterations: heuristic for poor forecast

Table of Contents

Clamped-free truss structure, geometric nonlinearity

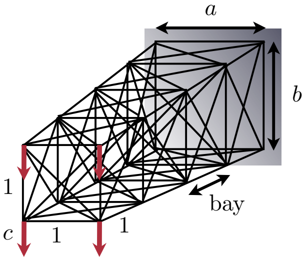


(a) initial displacement (red) (b) external forces

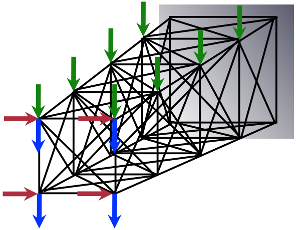
$$M(\mu)\ddot{x} + C(\mu)\dot{x} + f^{\text{int}}(x;\mu) = f^{\text{ext}}(t;\mu)$$

- M: mass matrix
- $C = \alpha M + \beta \nabla_x f^{\text{int}}(x^0)$: Rayleigh damping matrix
- f^{int}: internal force, *nonlinear in x*
- f^{ext}: sum of three sinusoidal forces, activated at $n = M/2$
- $N = 9000$ degrees of freedom in full-order model
- implicit midpoint rule: $w^n = \ddot{x}(t^{n-1} + 1/2\Delta t)$

Clamped-free truss structure, geometric nonlinearity



(c) initial displacement (red)



(d) external forces

9 inputs:

- 3 material properties: density, bar cross-sectional area, modulus of elasticity
- 2 geometrical parameters: base width *a*, base height *b*
- 1 initial-displacement magnitude
- 3 external-force magnitudes

Three reduced-order models compared

1. Galerkin projection

$$\Phi^T M(\mu)\ddot{\hat{x}} + \Phi C(\mu)\Phi\dot{\hat{x}} + \Phi^T f^{\text{int}}\left(x^0(\mu) + \Phi\hat{x};\mu\right) = \Phi^T f^{\text{ext}}(t;\mu)$$

2. Galerkin projection + collocation

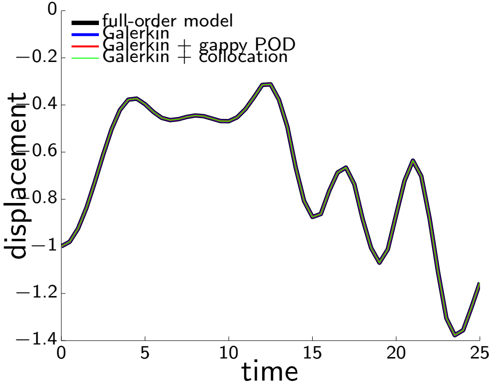
$$\Phi^T Z^T Z\left(M(\mu)\ddot{\hat{x}} + C(\mu)\Phi\dot{\hat{x}} + f^{\text{int}}\left(x^0(\mu) + \Phi\hat{x};\mu\right) - f^{\text{ext}}(t;\mu)\right) = 0$$

3. Galerkin projection + gappy POD approximation of residual

$$\Phi^T \Phi_R (Z\Phi_R)^+ Z\left(M(\mu)\ddot{\hat{x}} + C(\mu)\Phi\dot{\hat{x}} + f^{\text{int}}\left(x^0(\mu) + \Phi\hat{x};\mu\right)\right) = \Phi^T \Phi_R (Z\Phi_R)^+ Z f^{\text{ext}}(t;\mu))$$

- forecasting: memory $\alpha = 12$, Newton threshold $\tau = 0$

Case 1 (ideal): fixed inputs, no truncation of bases



- all responses nearly exact (as expected)

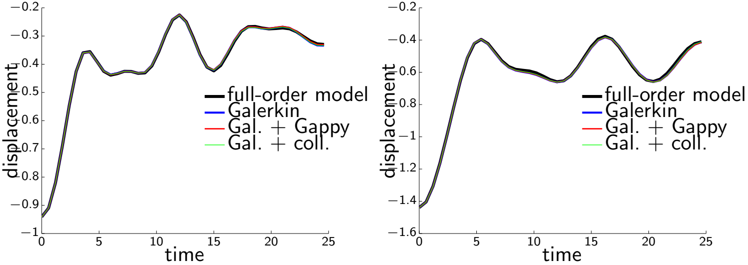
Case 1: forecasting drastically improves performance

ROM method	relative error	total Newton its		wall-time speedup	
		no forecast	forecast	no forecast	forecast
Galerkin	8.64×10^{-12}	99	2	1.01	1.84
Gal + Gappy	8.64×10^{-12}	99	2	36.4	69.3
Gal + coll	2.12×10^{-5}	100	16	36.5	61.9

+ forecast 'perfect': computation *only at first time step* for ROMs 1 and 2

Case 2: unforced dynamics, varying structure

- six varied inputs: material properties, geometry, initial cond
- six randomly chosen training configurations
- two randomly chosen online configurations
- 99.99% energy criterion for POD



(e) online configuration 1 (f) online configuration 2

+ all relative errors less than 1%

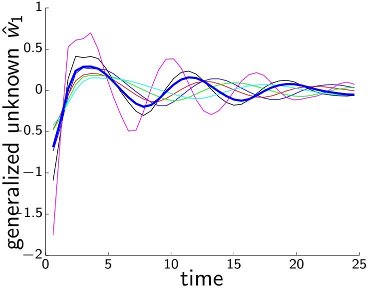
Case 2: forecasting improves performance by $\sim$ 60%

online config	ROM method	Newton its		wall-time speedup	
		no forecast	forecast	no forecast	forecast
1	Galerkin	82	49	0.998	1.71
	Gal + Gappy	82	48	58.2	82.4
	Gal + coll	82	48	59.4	80.8
2	Galerkin	82	48	1.03	1.55
	Gal + Gappy	82	48	54.0	86.1
	Gal + coll	82	48	55.5	88.7

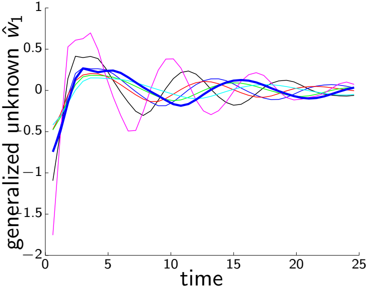
+ forecasting cuts Newton steps nearly in half
+ wall-time speedup increases by roughly 60%
- performance less impressive than ideal case

Case 2: temporal behavior similar across input variation

- temporal behavior of first generalized unknown $\hat{w}_1$ (bold=online; thin=training):



(g) online configuration 1



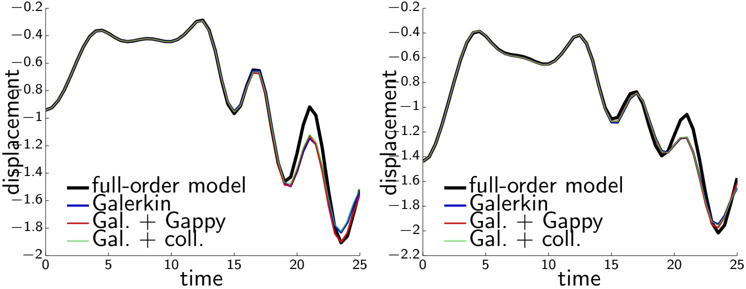
(h) online configuration 2

- temporal behavior similar across input variation: this explains the method's effectiveness

+ method seems to handle frequency shifts

Case 3: forced dynamics, fixed structure

- four varied inputs: external-force magnitudes, initial condition



(i) online configuration 1 (j) online configuration 2

+ relative errors roughly 1.5%

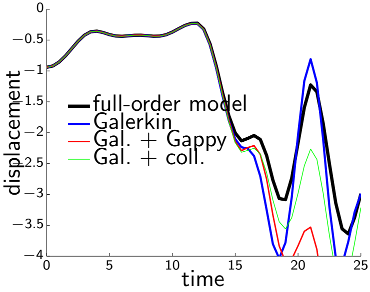
Case 3: forecasting improves performance by $\sim$ 50%

online config	ROM method	Newton its		wall-time speedup	
		no forecast	forecast	no forecast	forecast
1	Galerkin	100	60	1.02	1.47
	Gal + Gappy	100	61	52.7	75.2
	Gal + coll	100	71	49.5	70.9
2	Galerkin	100	60	1.02	1.52
	Gal + Gappy	100	61	52.1	73.2
	Gal + coll	100	68	54.3	71.8

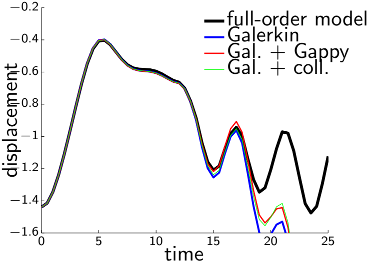
+ forecasting cuts Newton steps by 40%
+ wall-time speedup increases by roughly 50%
- performance again slightly worse (richer dynamics)

Case 4: forced dynamics, varying structure

- nine varied inputs: external-force magnitudes, initial condition



(k) online configuration 1



(l) online configuration 2

- ROMs increasingly inaccurate after forces activated ($t = 12.5$)

- relative errors between 5% and 17%

Case 4: forecasting improves performance by $\sim 35\%$

online config	ROM method	Newton its		wall-time speedup	
		no forecast	forecast	no forecast	forecast
1	Galerkin	104	109	1.21	1.38
	Gal + Gappy	124	94	8.0	9.48
	Gal + coll	120	90	8.12	11
2	Galerkin	95	62	1.04	1.47
	Gal + Gappy	95	64	7.47	10.4
	Gal + coll	100	73	7.36	9.98

- forecasting method does not always help: number of Newton steps increases in one case

+ forecasting cuts Newton steps by 25% in most cases

+ wall-time speedup increases by roughly 35%

Conclusions

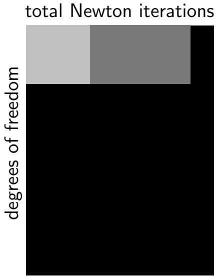
use temporal-behavior data to reduce ROM simulation time

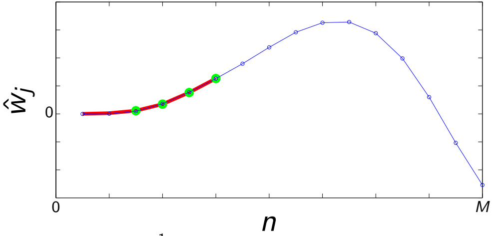
- offline: compute time-evolution bases
- online:
  1. use gappy POD to forecast generalized unknowns
  2. use forecast as initial guess in ROM Newton solver

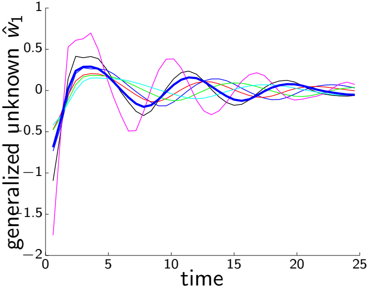
+ observed decrease in temporal complexity
+ observed decrease in ROM simulation wall time
+ no additional error introduced

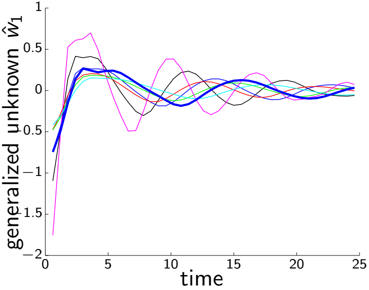
- best performance occurs in the case of:
  1. smooth dynamics (low frequency)
  2. temporal behavior similar across input variation
  3. accurate ROM
- **Reference**: K. Carlberg, J. Ray, and B. van Bloemen Waanders. 'Decreasing the temporal complexity for nonlinear, implicit reduced-order models by forecasting,' arXiv e-Print 1209.5455 (2012). (submitted to CMAME)

Questions?


total Newton iterations
degrees of freedom


$\hat{w}_j$
0
0
M
n


generalized unknown $\hat{w}_1$
time
1
0.5
0
−0.5
−1
−1.5
−2
0
5
10
15
20
25


generalized unknown $\hat{w}_1$
time
1
0.5
0
−0.5
−1
−1.5
−2
0
5
10
15
20
25

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.

Bibliography I

Astrid, P., Weiland, S., Willcox, K., and Backx, T. (2008).
Missing point estimation in models described by proper orthogonal decomposition.
IEEE Transactions on Automatic Control, 53(10):2237–2251.

Bos, R., Bombois, X., and Van den Hof, P. (2004).
Accelerating large-scale non-linear models for monitoring and control using spatial and temporal correlations.
In *Proceedings of the American Control Conference*, volume 4, pages 3705–3710.

Bibliography II

Carlberg, 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.

Carlberg, K., Farhat, C., Cortial, J., and Amsallem, D. (2012).
The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows.
arXiv e-print, (1207.1349).

Chaturantabut, S. and Sorensen, D. C. (2010).
Nonlinear model reduction via discrete empirical interpolation.
SIAM Journal on Scientific Computing, 32(5):2737–2764.

Drohmann, M., Haasdonk, B., and Ohlberger, M. (2012).
Reduced basis approximation for nonlinear parameterized evolution equations based on empirical operator interpolation.
SIAM Journal on Scientific Computing.

Galbally, D., Fidkowski, K., Willcox, K., and Ghattas, O. (2009).
Non-linear model reduction for uncertainty quantification in large-scale inverse problems.
International Journal for Numerical Methods in Engineering.

Hinterberger, C., Garcıa-Villalba, M., and Rodi, W. (2004).
Large eddy simulation of flow around the Ahmed body.
In R. McCallen, F. Browand, J. R., editor, *The Aerodynamics of Heavy Vehicles: Trucks, Buses, and Trains, Lecture Notes in Applied and Computational Mechanics*, volume 19. Springer.

Bibliography IV

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

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