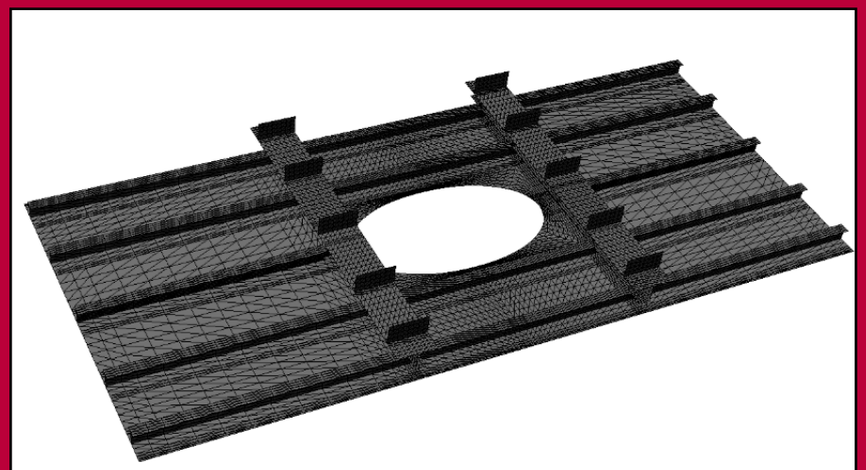
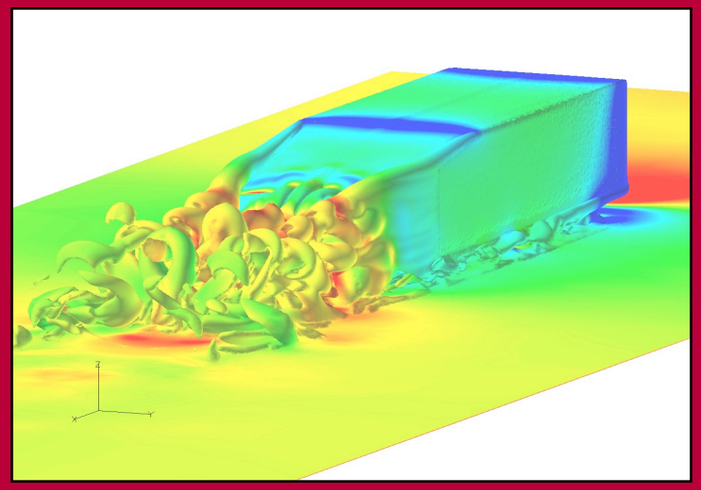
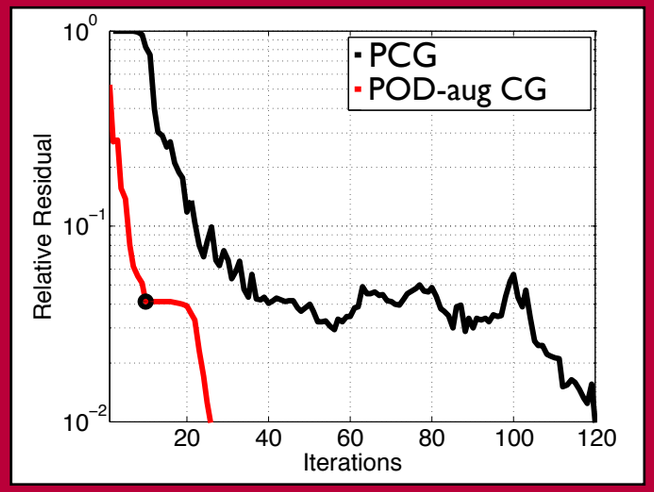
Model reduction-based iterative methods for real-time simulation and repeated analyses of mathematical models





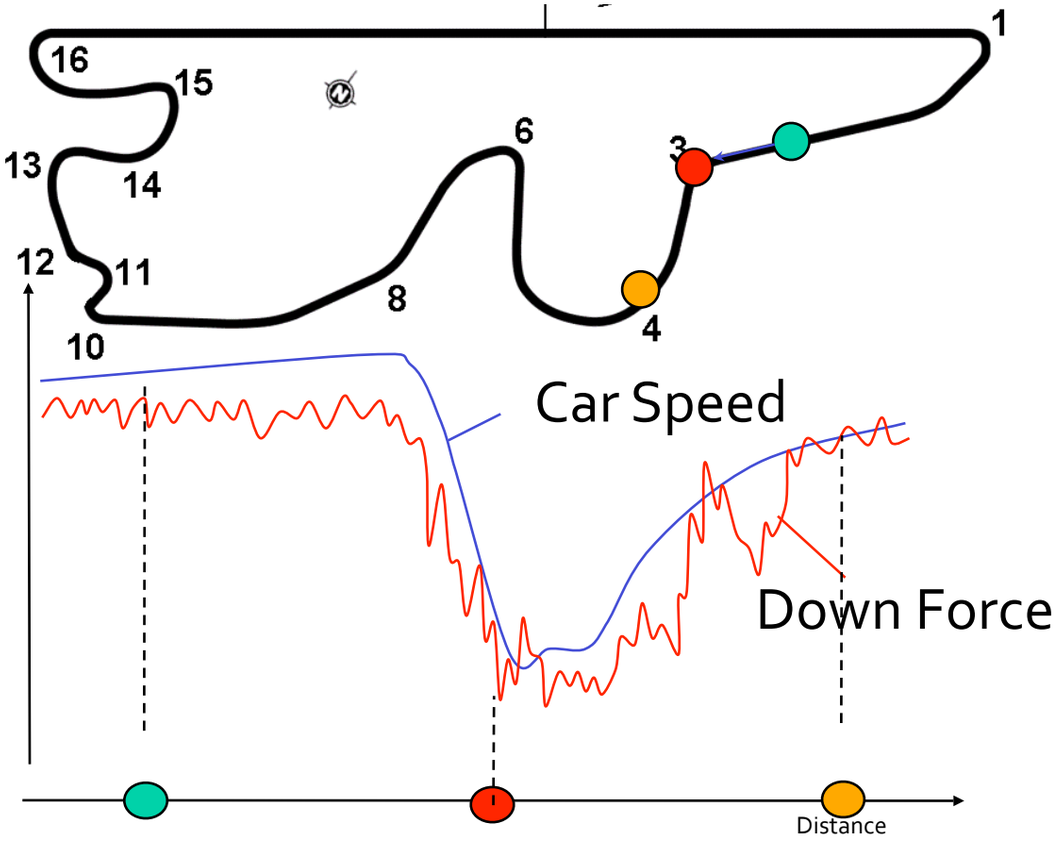


Kevin Carlberg and Charbel Farhat
Stanford University
Department of Aeronautics & Astronautics

Linear Algebra and Optimization Seminar
Stanford, CA
October 27, 2010

(Near) Real-time analysis

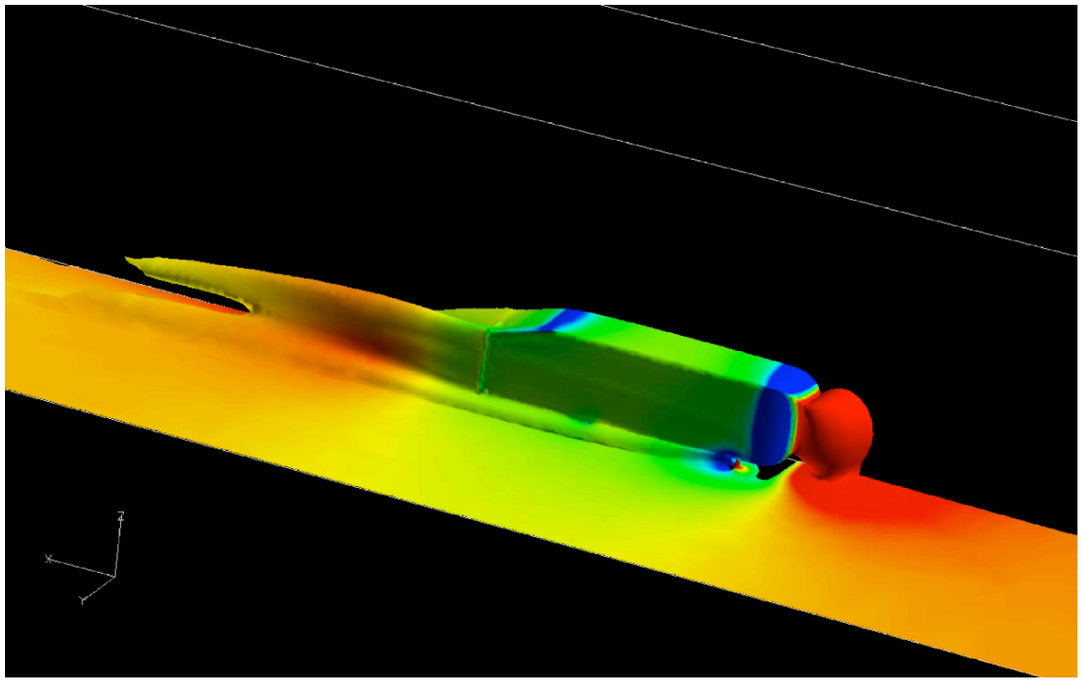
TOYOTA motorsports



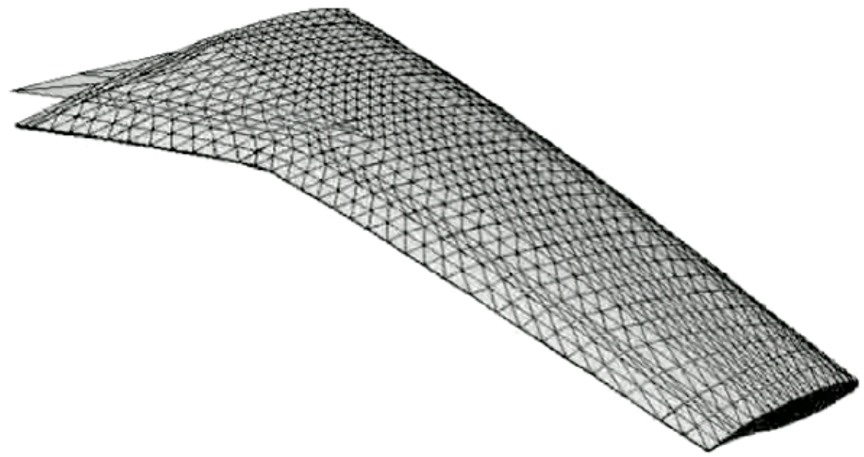
Requirements

- Fast analysis during the season ("online")
- Can sacrifice some accuracy
- Can run expensive analyses before the season ("offline")

Repeated analyses



Nonlinear analysis



Design optimization

Requirements

- Stringent accuracy requirement
- Want to minimize the overall computational cost of analyses

Model reduction problem types

Real-time analysis

- Fast-turnaround design
- "In-field" analysis
- Model predictive control

Repeated analyses

- Nonlinear analysis
- Design optimization
- Parameter space sampling

Multiple objectives: error and cost

Real-time analysis

Online cost more important:

$$\begin{aligned} &\text{minimize} && \text{error} \\ &\text{subject to} && \text{online cost} \leq \tau \end{aligned}$$

Repeated analyses

Error more important:

$$\begin{aligned} &\text{minimize} && \text{total cost} \\ &\text{subject to} && \text{error} \leq \epsilon \end{aligned}$$

Model reduction problem types

Real-time analysis

- Fast-turnaround design
- "In-field" analysis
- Model predictive control

Repeated analyses

- Nonlinear analysis
- Design optimization
- Parameter space sampling

Multiple objectives: error and cost

Real-time analysis

Online cost more important:

$$\begin{aligned} &\text{minimize} && \text{error} \\ &\text{subject to} && \text{online cost} \leq \tau \end{aligned}$$

Repeated analyses

Error more important:

$$\begin{aligned} &\text{minimize} && \text{total cost} \\ &\text{subject to} && \text{error} \leq \epsilon \end{aligned}$$

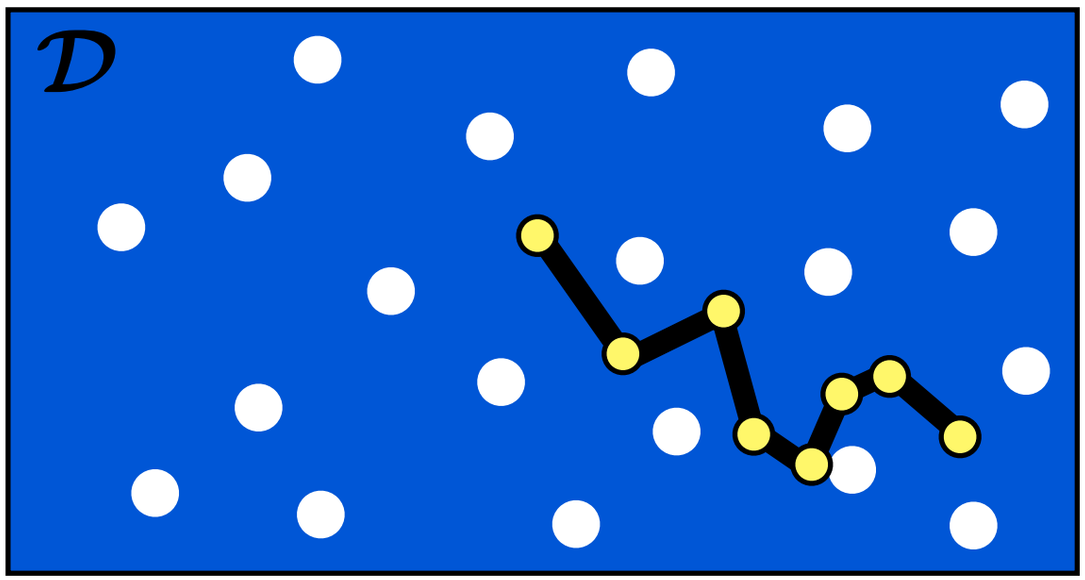
Offline/online strategy

1) Offline

- Sample input space
- Build surrogate model

2) Online

- Analysis with surrogate model



Real-time analysis

✓ High offline cost okay

✓ Very low online cost

$\mathcal{D}$ Input space

○ Full-order model evaluation

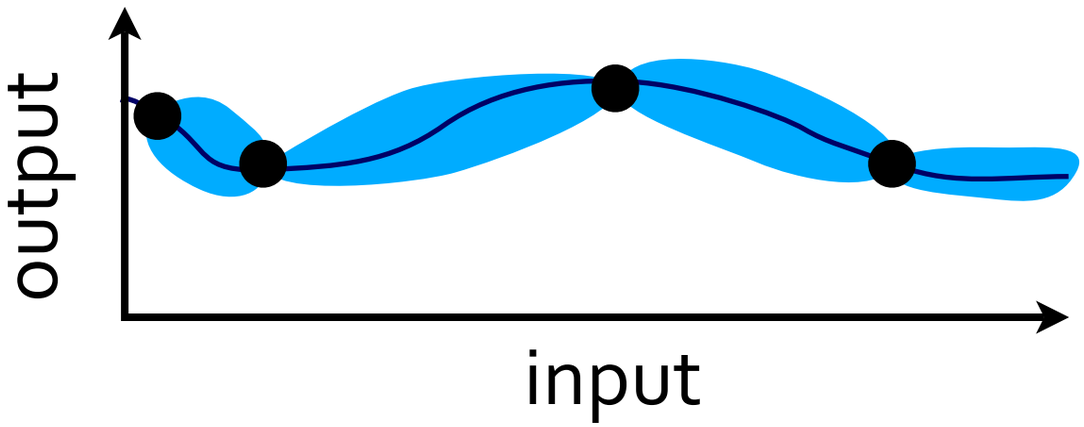
○ Surrogate model evaluation

▬ Online trajectory

Surrogate modeling strategies

1) Data fitting (response surfaces, Kriging)

- ◉ Directly model input-output map
- x "Blind" to problem physics
- x Impractical for large-dimensional input spaces



2) Reduced-order modeling

- ◉ Approximately solve state equations, then compute outputs
- ✓ Robust: queries the physics online
- ✓ Can apply arbitrary excitations/boundary conditions

Reduced-order modeling

- Linear dynamical systems: "mature"

$$x^{n+1} = Ax^n + Bu^n$$
$$y^n = Cx^n + Du^n$$

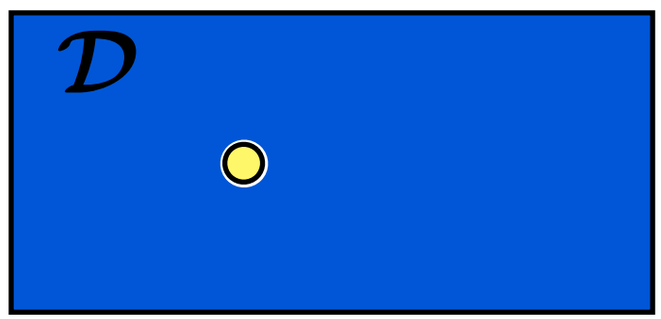
SVD-based

- Balanced truncation
- Hankel norm approximation
- Balanced POD

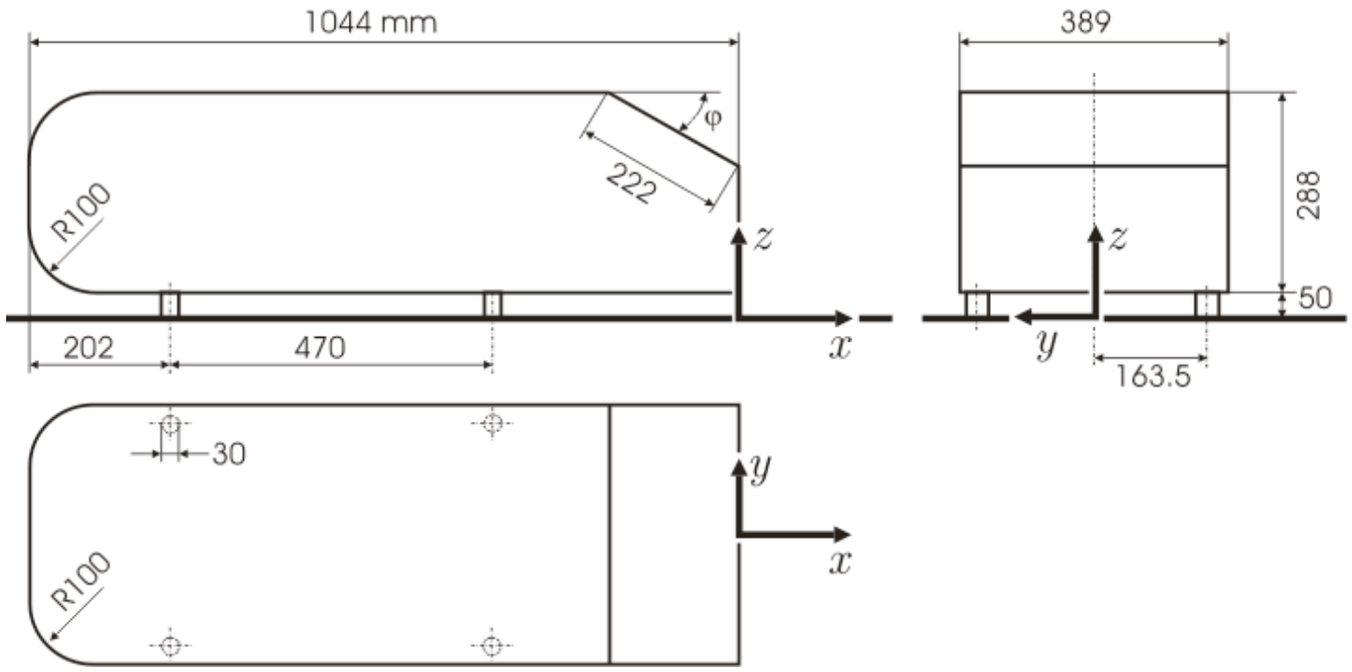
Krylov-based

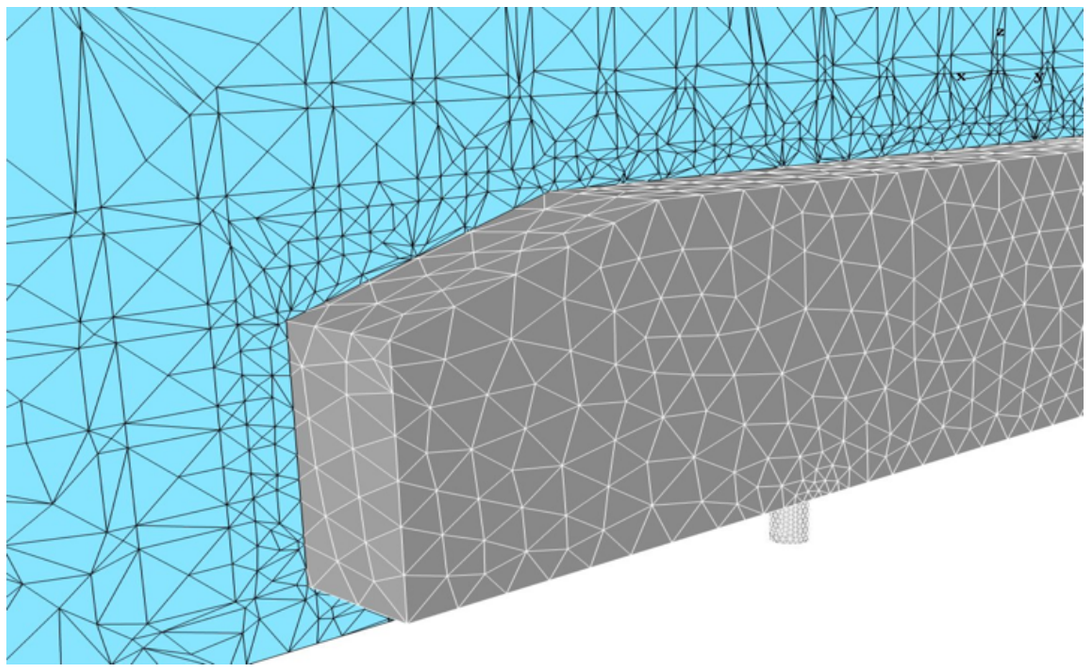
- Partial realization
- Padé approximation
- Rational interpolation

- Nonlinear systems: major problems remain
  1. Accuracy
  2. Computational cost



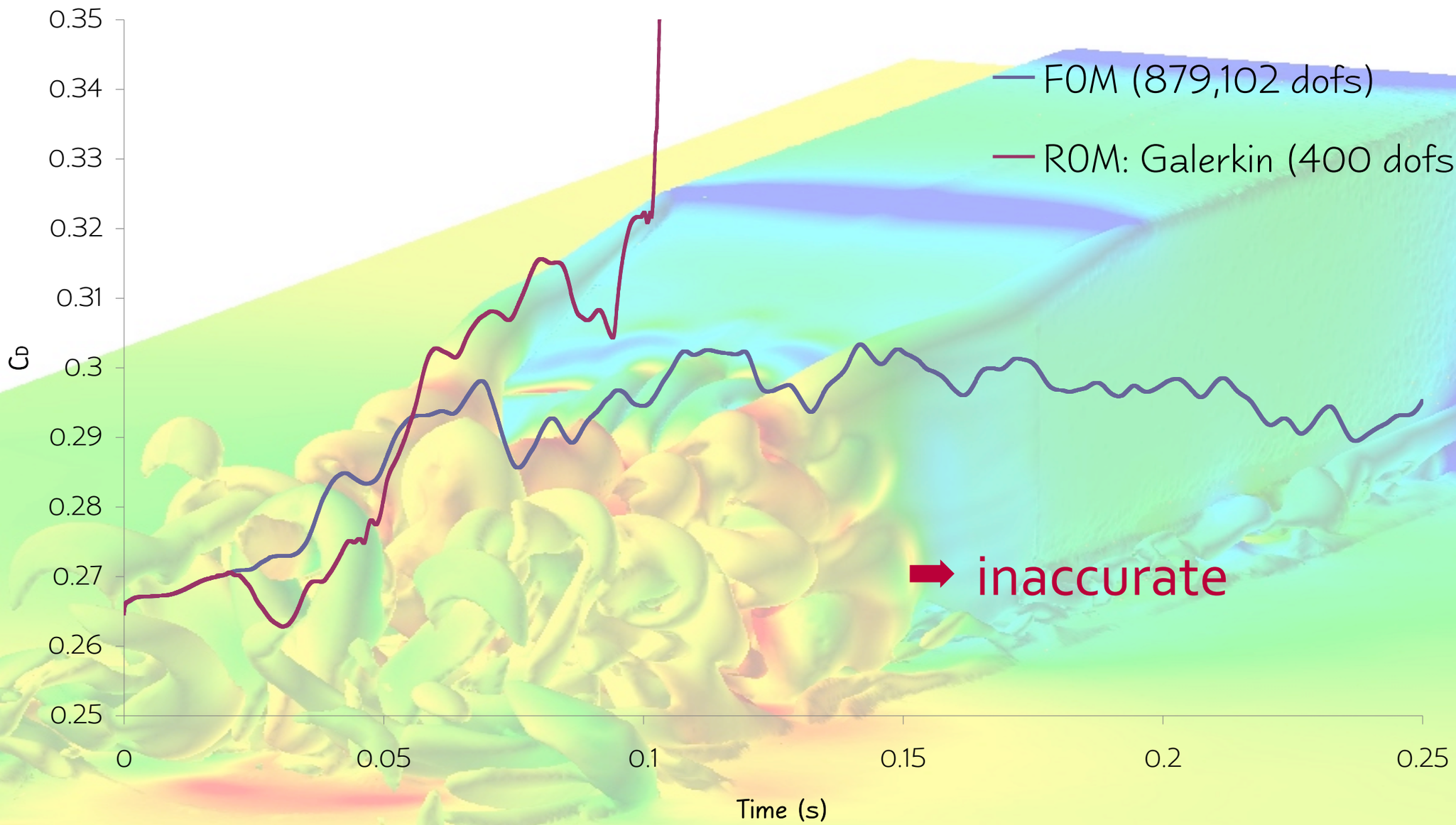
Nonlinear example: Ahmed body





- Navier-Stokes simulation
  - ϕ = 20 deg
  - V = 60 m/s
  - Re = 4.29×10^6
  - DES turbulence model
  - 146,517 nodes
  - 837,894 tetrahedra
- Full-order model (FOM): 879,102 degrees of freedom
- POD/Galerkin reduced-order model (ROM): 400 degrees of freedom

Ahmed body: POD/Galerkin results



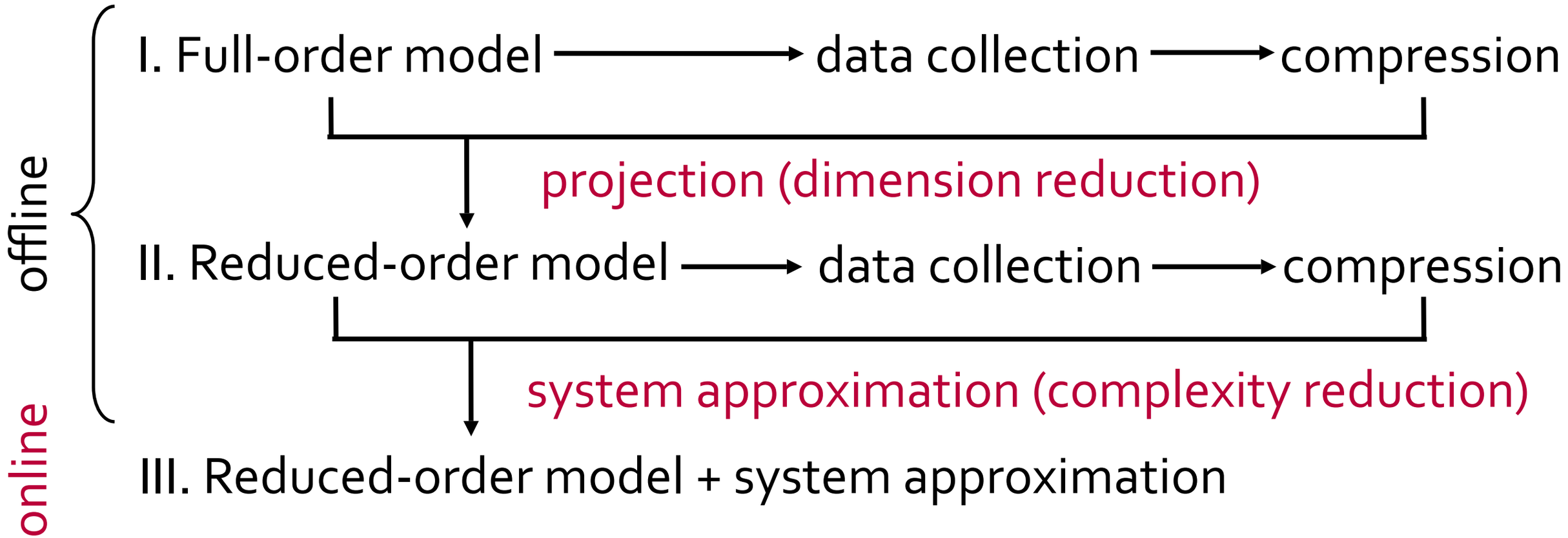
➡ inaccurate

- CPU time (first 0.1 s of simulated time)
  - FOM: 9,292 s
  - ROM POD/Galerkin: 24,984 s

➡ computationally expensive

Methodology

- Consecutively introduce approximations that satisfy 2 properties



1. Consistency: In the limit of no compression, reproduce exactly the solution of the previous model for sampled problems
2. Optimality: Minimize an error measure with respect to the previous model (*a priori* convergence)

Mathematical framework

- Nonlinear PDE

$$\mathcal{L}(u; x, t) = 0$$

- Semi-discretized PDE in space (ODE)

$$\mathcal{L}^d(u; t) = 0$$

- Fully-discretized PDE with implicit time integration

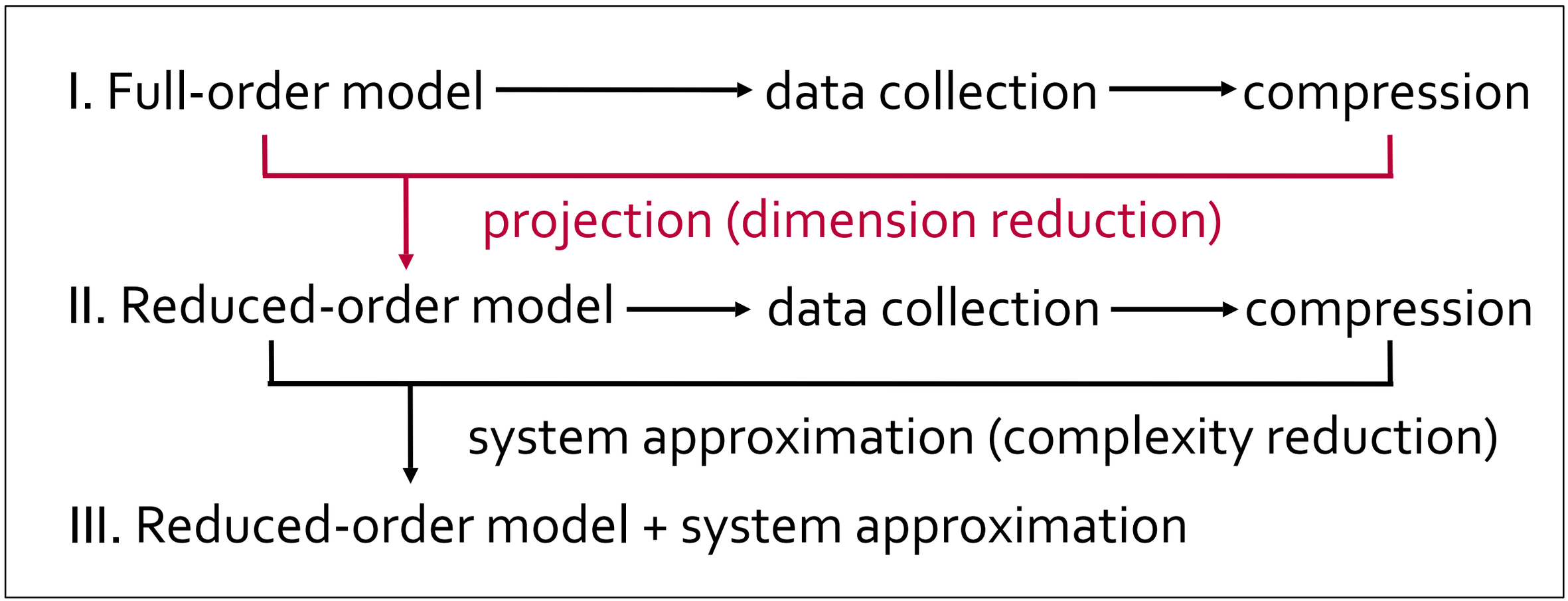
$$R(y^{n+1}; y^n, \dots, y^0; t^n) = 0$$

- FOM: sequence of nonlinear systems of equations of general form

$$R(y) = 0$$

  - $R : \mathbb{R}^N \to \mathbb{R}^N, \ (w) \mapsto R(w)$
  - dimension N "large"

Projection


I. Full-order model
data collection
compression
projection (dimension reduction)
II. Reduced-order model
data collection
compression
system approximation (complexity reduction)
III. Reduced-order model + system approximation

Petrov-Galerkin Projection

- To decrease the dimension, search for a solution in the affine subspace: $y^{(0)}$+range(Φ_y) of dimension $n_y \ll N$

$$y \approx y^{(0)} + \Phi_y y_r$$

- Enforce residual orthogonal to n_y-dimensional subspace range(Ψ)

$$\Psi^T R(y^{(0)} + \Phi_y y_r) = 0$$

- Solve reduced system by Newton's method for k=1,…,K (converged)

$$\Psi^T J^{(k)} \Phi_y p^{(k)} = -\Psi^T R^{(k)}$$

$$y_r^{(k+1)} = y_r^{(k)} + \alpha^{(k)} p^{(k)}$$

  - $R^{(k)} \equiv R(y^{(0)} + \Phi_y y_r^{(k)}),\ J^{(k)} \equiv \dfrac{dR}{dw}\left(y^{(0)} + \Phi_y y_r^{(k)}\right)$

Projection consistency

Proposition

The projection approximation is consistent if:

1. Φ_y is a POD basis computed with snapshots (y-$y^{(0)}$) collected during the evaluation of Model I
2. y is sufficiently close to $y^{(0)}$
3. The reduced Jacobian $\Psi^T \frac{dR}{dw}(\cdot)\Phi_y : \Omega \rightarrow \mathbb{R}^{n_y \times n_y}$ is Lipschitz continuous, with $\Omega \subset \mathbb{R}^N$ an open set containing $y - y^{(0)}$
4. The reduced Jacobian at the solution $\Psi^T \frac{dR}{dw}(y)\Phi_y$ is nonsingular

1: Defines data and procedure for computing Φ_y

2–4: Newton's method convergence conditions

Proper orthogonal decomposition

- Given n_x "snapshots" x^i, the first k POD vectors satisfy:

$$\text{span}\{\phi^i\}_{i=1}^{k} = \arg\min_{\mathcal{S}\in\mathcal{G}(k,N)} \sum_{j=1}^{n_x} \| (I - \Pi_{\mathcal{S}})\, x_j \|^2$$

  - $\Pi_{\mathcal{S}}$: orthogonal projection onto $\mathcal{S}$
  - $\mathcal{G}(k, N)$: set of n-dimensional subspaces of $\mathbb{R}^N$
  - $\Phi = \begin{bmatrix} \phi^1 & \cdots & \phi^k \end{bmatrix}$: POD basis in matrix form

- Properties
  - "No compression": $k = n_x$ and range(Φ) = range ($[x^1 \cdots x^{n_x}]$)
  - Efficient computation by singular value decomposition (SVD)

Projection optimality

- Want solution $p^{(k)}$ to satisfy for some norm Θ

$$p^{(k)} = \arg \min_{p \in \mathbb{R}^{n_y}} \| \Phi_y p - \left(J^{(k)} \right)^{-1} R^{(k)} \|_\Theta \qquad (1)$$

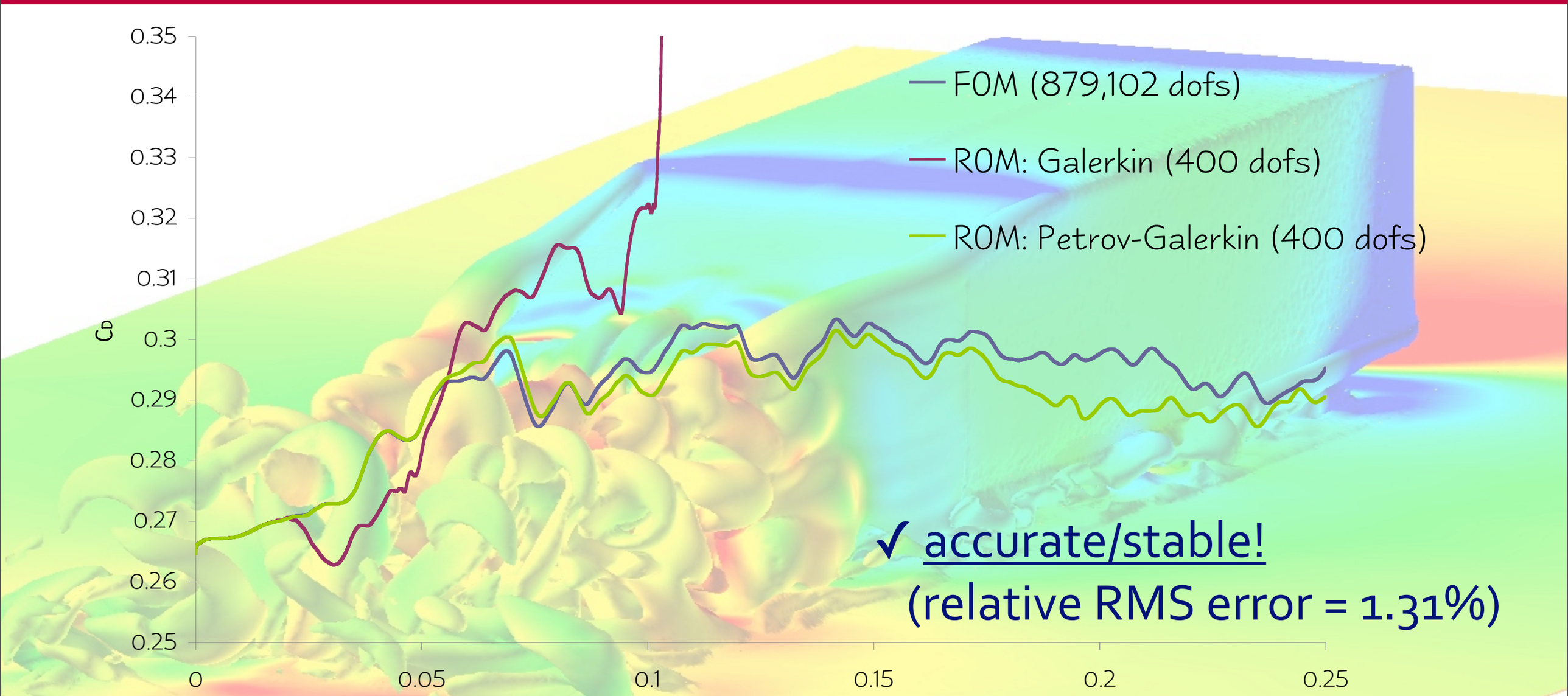
  - Galerkin: $\Psi = \Phi_y$ satisfies (1) with $\Theta = J^{(k)}$ only if $J^{(k)}$ SPD
  - Petrov-Galerkin: $\Psi = J^{(k)} \Phi_y$ satisfies (1) with $\Theta = J^{(k)T} J^{(k)}$ if $J^{(k)}$ nonsingular

- Least-squares Petrov-Galerkin projection
  - Equivalent to globally convergent Gauss-Newton method for

$$\min \| R(y + \Phi_y y_r) \|_2$$

  - Ψ is state-dependent, changes each Newton iteration
  - Can only be introduced on fully discrete system

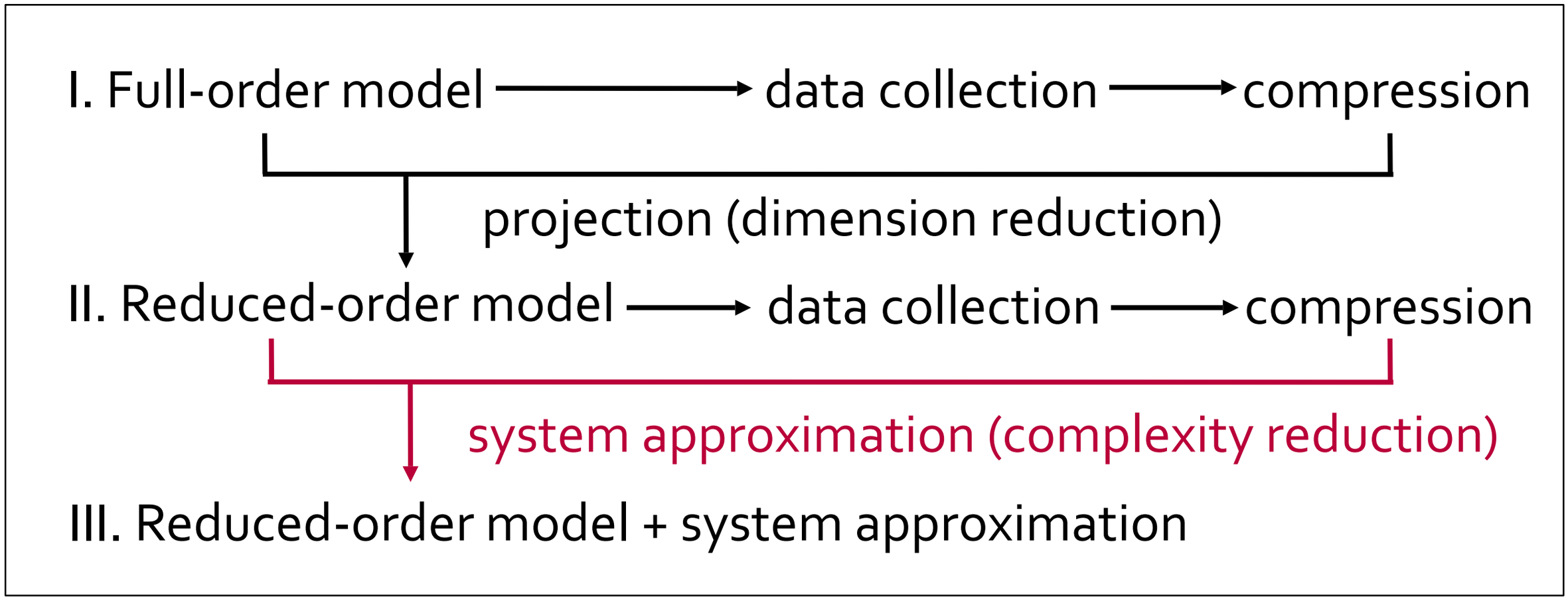
Ahmed body: ROM results



✓ accurate/stable!
(relative RMS error = 1.31%)

- CPU time (first 0.1 s)
  - FOM. 9,292 s
  - ROM: Galerkin. 24,973 s
  - ROM: Petrov-Galerkin. 24,984 s ➡ remains expensive

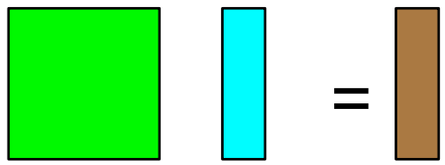
System approximation



Computational bottleneck

- Reduced-order Newton iterations are

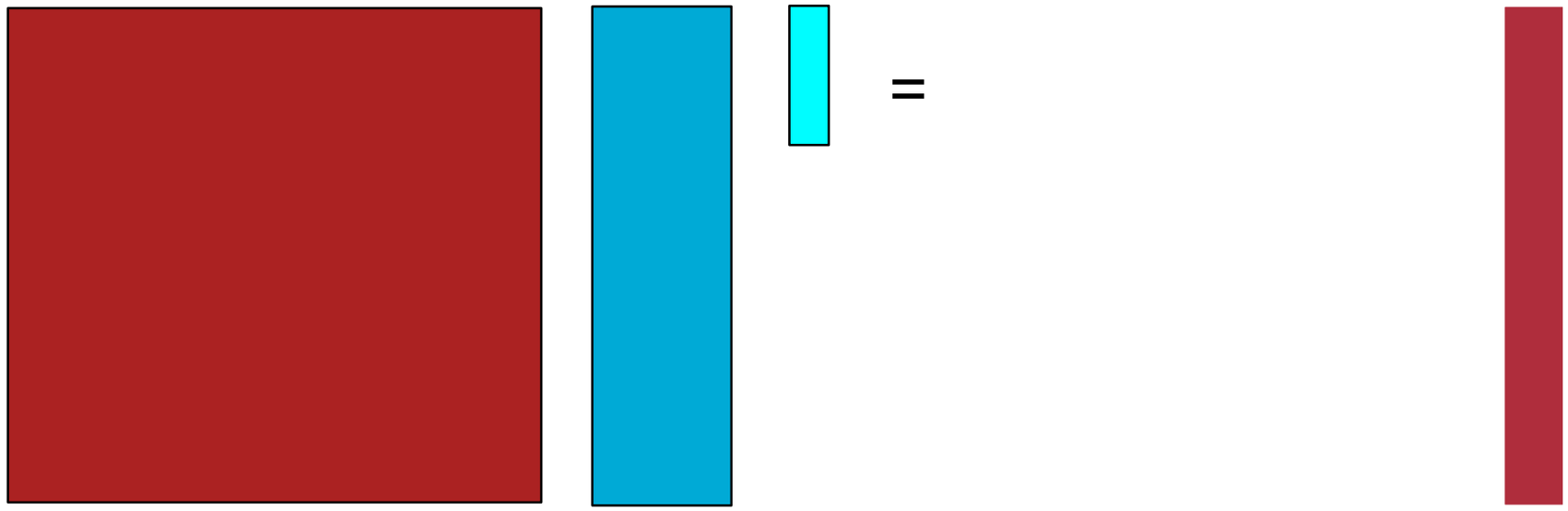
$$[J^{(k)}\Phi_y]^T[J^{(k)}\Phi_y]p^{(k)} = -[J^{(k)}\Phi_y]^T R^{(k)}$$



Computational bottleneck

- Reduced-order Newton iterations are

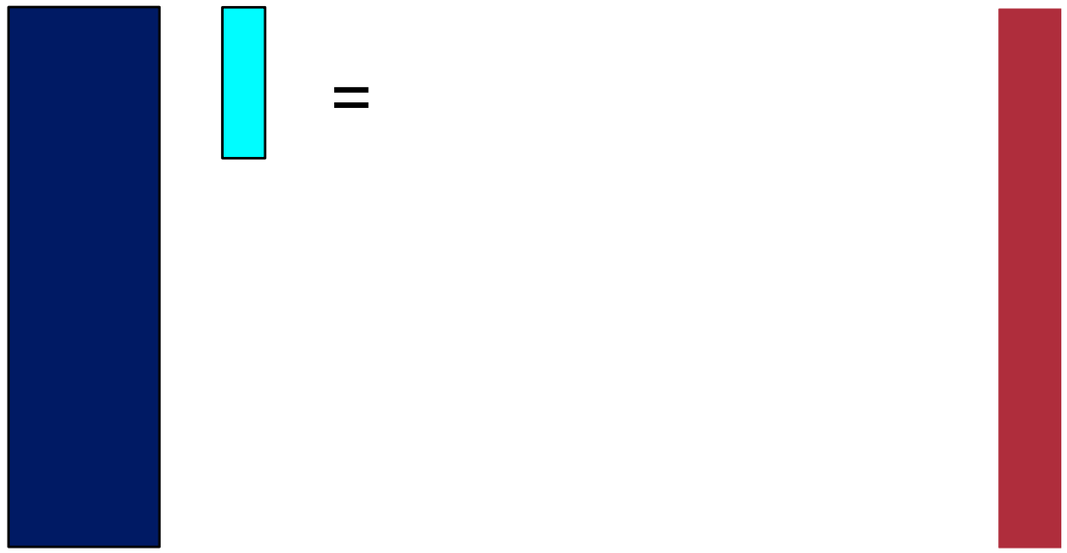
$$[J^{(k)}\Phi_y]^T[J^{(k)}\Phi_y]p^{(k)} = -[J^{(k)}\Phi_y]^T R^{(k)}$$



Computational bottleneck

- Reduced-order Newton iterations are

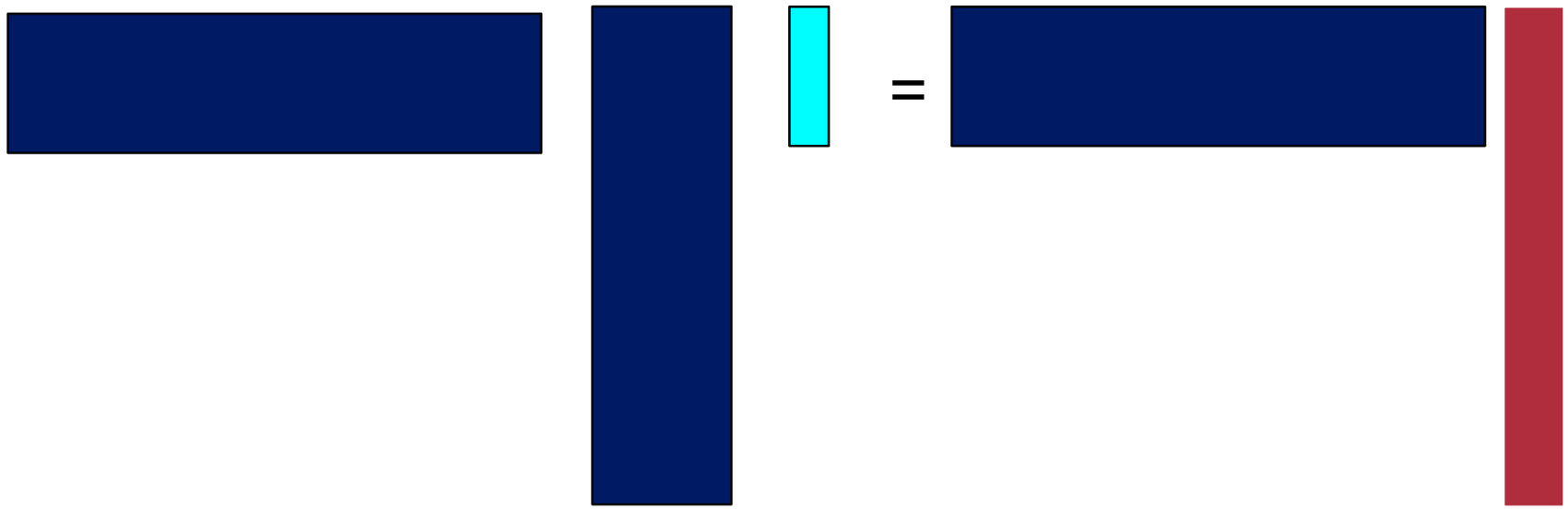
$$[J^{(k)}\Phi_y]^T[J^{(k)}\Phi_y]p^{(k)} = -[J^{(k)}\Phi_y]^T R^{(k)}$$



Computational bottleneck

- Reduced-order Newton iterations are

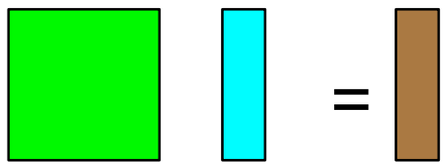
$$[J^{(k)}\Phi_y]^T[J^{(k)}\Phi_y]p^{(k)} = -[J^{(k)}\Phi_y]^T R^{(k)}$$



Computational bottleneck

- Reduced-order Newton iterations are

$$[J^{(k)}\Phi_y]^T[J^{(k)}\Phi_y]p^{(k)} = -[J^{(k)}\Phi_y]^T R^{(k)}$$



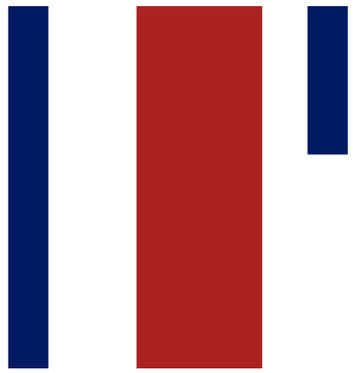
- Operation count at each iteration scales with large dimension N

➡ Expensive even though reduced equations are of small dimension

System approximation

- Tensor approximations: $R^{(k)} \in \text{range}(\Phi_R)$, $J^{(k)}\Phi_y \in \text{range}(\Phi_J)$

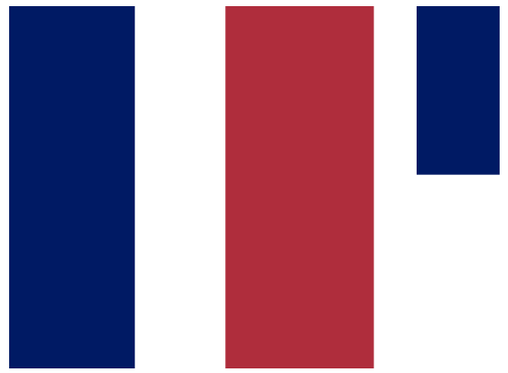
$$R^{(k)} \approx \Phi_R R_r^{(k)}$$



invariant

iteration-dependent

$$J^{(k)}\Phi_y \approx \Phi_J J_r^{(k)}$$



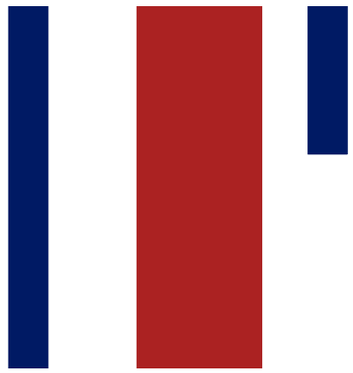
- Newton iterations become

$$[J^{(k)}\Phi_y]^T[J^{(k)}\Phi_y]p^{(k)} = -[J^{(k)}\Phi_y]^T R^{(k)}$$

System approximation

- Tensor approximations: $R^{(k)} \in \text{range}(\Phi_R)$, $J^{(k)}\,\Phi_y \in \text{range}(\Phi_J)$

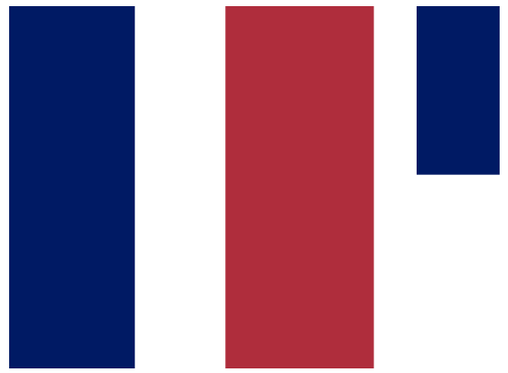
$$R^{(k)} \approx \Phi_R\, R_r^{(k)}$$



invariant

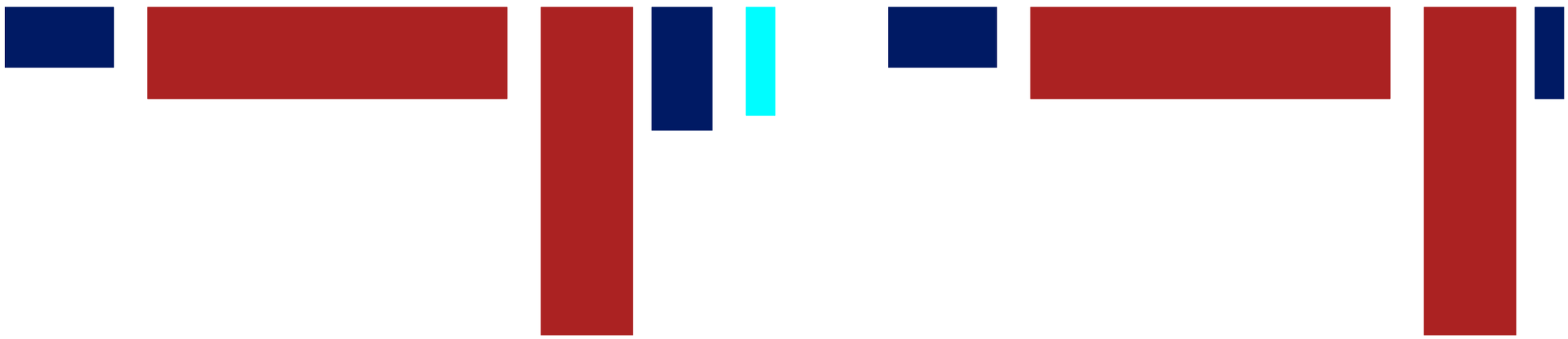
iteration-dependent

$$J^{(k)}\,\Phi_y \approx \Phi_J\, J_r^{(k)}$$



- Newton iterations become

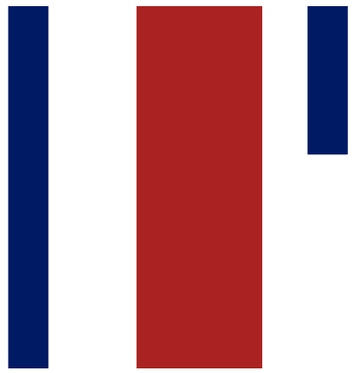
$$(J_r^{(k)})^T\,\Phi_J{}^T\Phi_J J_r^{(k)}\,p^{(k)} = -(J_r^{(k)})^T\Phi_J{}^T\Phi_R R_r^{(k)}$$



System approximation

- Tensor approximations: $R^{(k)} \in \text{range}(\Phi_R)$, $J^{(k)}\,\Phi_y \in \text{range}(\Phi_J)$

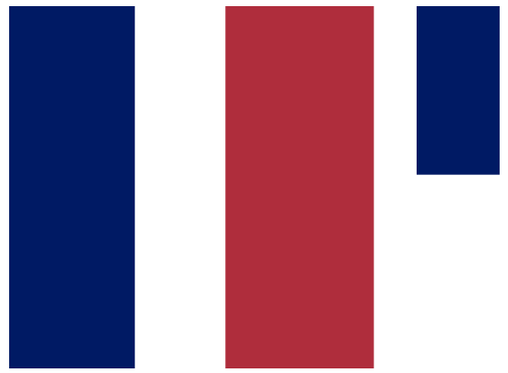
$$R^{(k)} \approx \Phi_R\, R_r^{(k)}$$



invariant

iteration-dependent

$$J^{(k)}\,\Phi_y \approx \Phi_J\, J_r^{(k)}$$



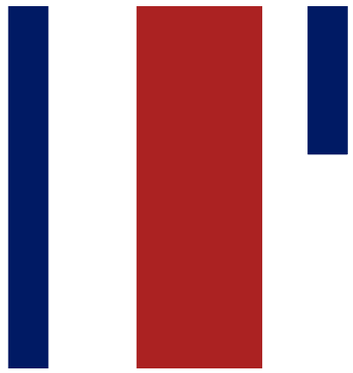
- Newton iterations become

$$(J_r^{(k)})^T\,\Phi_J{}^T\Phi_J J_r^{(k)}\,p^{(k)} = -(J_r^{(k)})^T\Phi_J{}^T\Phi_R R_r^{(k)}$$

System approximation

- Tensor approximations: $R^{(k)} \in \text{range}(\Phi_R)$, $J^{(k)}\,\Phi_y \in \text{range}(\Phi_J)$

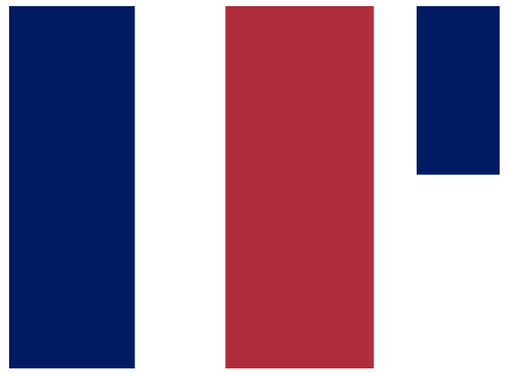
$$R^{(k)} \approx \Phi_R\, R_r^{(k)}$$



invariant

iteration-dependent

$$J^{(k)}\,\Phi_y \approx \Phi_J\, J_r^{(k)}$$



- Newton iterations become

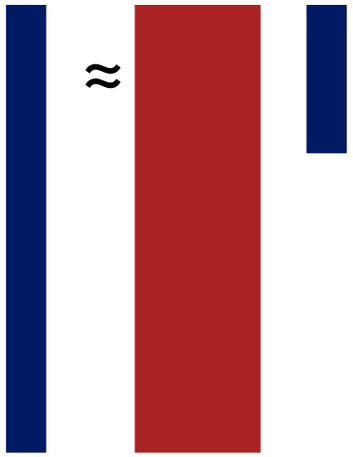
$$(J_r^{(k)})^T\,\Phi_J{}^T\Phi_J J_r^{(k)}\,p^{(k)} = -(J_r^{(k)})^T\Phi_J{}^T\Phi_R R_r^{(k)}$$

- Form $\Phi_J{}^T\Phi_J$, $\Phi_J{}^T\Phi_R$ offline: online operations independent of N!
- Similar: Barrault et al., 2004; Grepl et al., 2007; Nguyen & Peraire, 2008; Chatarantabut & Sorensen, 2009; Galbally et al., 2010

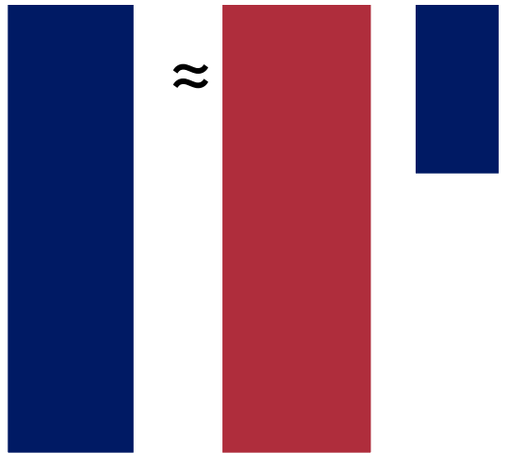
System approximation

- Tensor approximations

$$R^{(k)} \approx \Phi_R \, R_r^{(k)}$$



$$J^{(k)} \Phi_y \approx \Phi_J \, J_r^{(k)}$$



- Property 1: Optimality
  - $R_r^{(k)}$, $J_r^{(k)}$ computed online by gappy data reconstruction

- Property 2: Consistency
  - Φ_R, Φ_J computed offline by POD with specific snapshots

Gappy data reconstruction

- Goal: accurately reconstruct vector $F \in \mathbb{R}^N$
- Given: 1) a basis $Z = [z^1 \ \cdots z^p]$

  2) some sampled entries of the vector

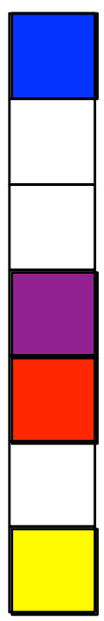
  $F_i, \ i \in I$ with $|I| = q \geq p$ and $q \ll N$
- Define restriction: $\underline{F} \equiv [F]_i, \ i \in I$

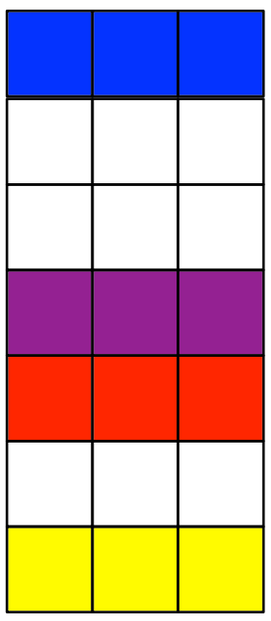
  $\underline{Z} \equiv [\underline{z}^1 \ \cdots \ \underline{z}^p]$

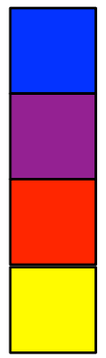
[Everson & Sirovich, 1995]

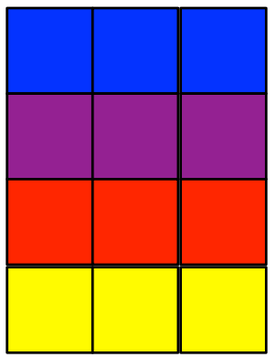
Example

N=7
p=3
q=4
I={1,4,5,7}

$F =$ 

$Z =$ 

$\underline{F} =$ 

$\underline{Z} =$ 

Gappy data reconstruction

- Least squares minimization on sampled entries

$$F_r = \arg\min_{x \in \mathbb{R}^p} \|\underline{Z}x - \underline{F}\|_2$$

- Reconstruct all entries of the vector

$$F \approx ZF_r$$

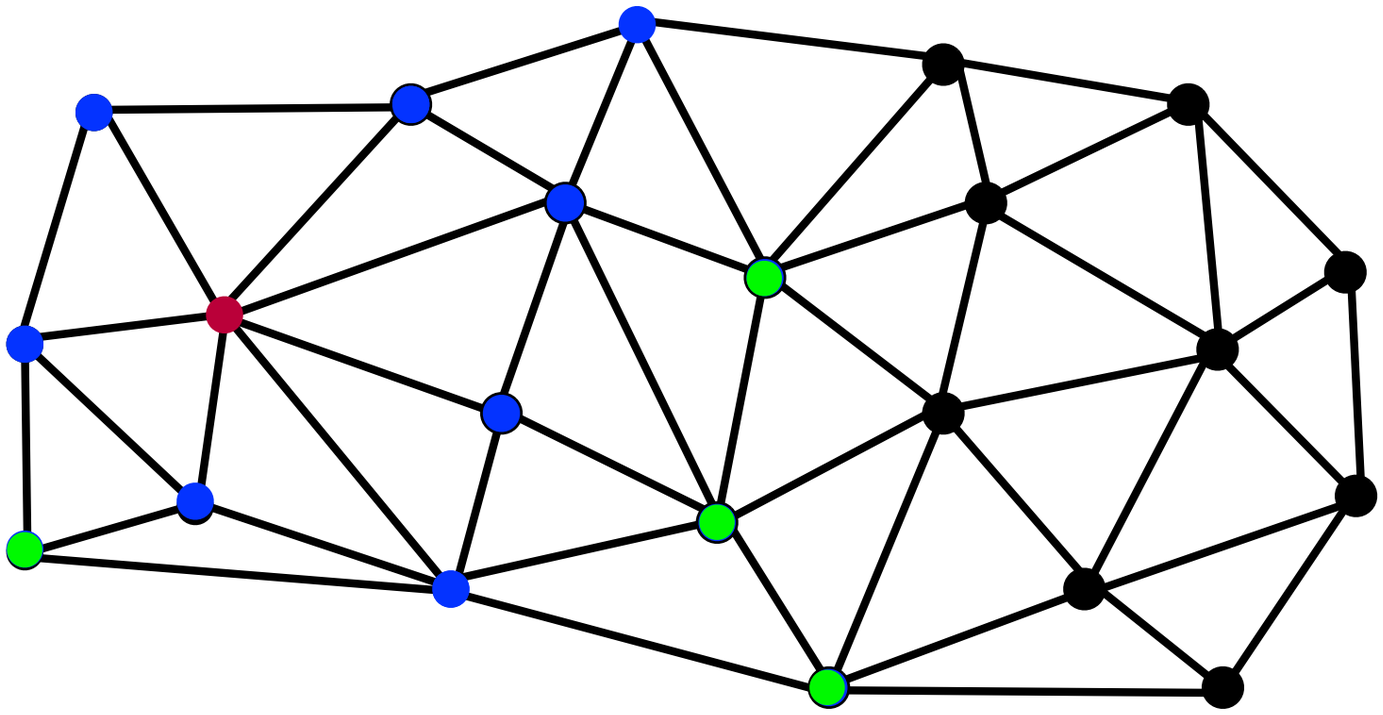
- Optimality
  - error $\|\underline{Z}F_r - \underline{F}\|_2$ monotonically decreases as p increases

- Apply at each Newton iteration to reconstruct

1. $R^{(k)}$, with $F_r = R_r^{(k)}$ and $Z = \Phi_R$

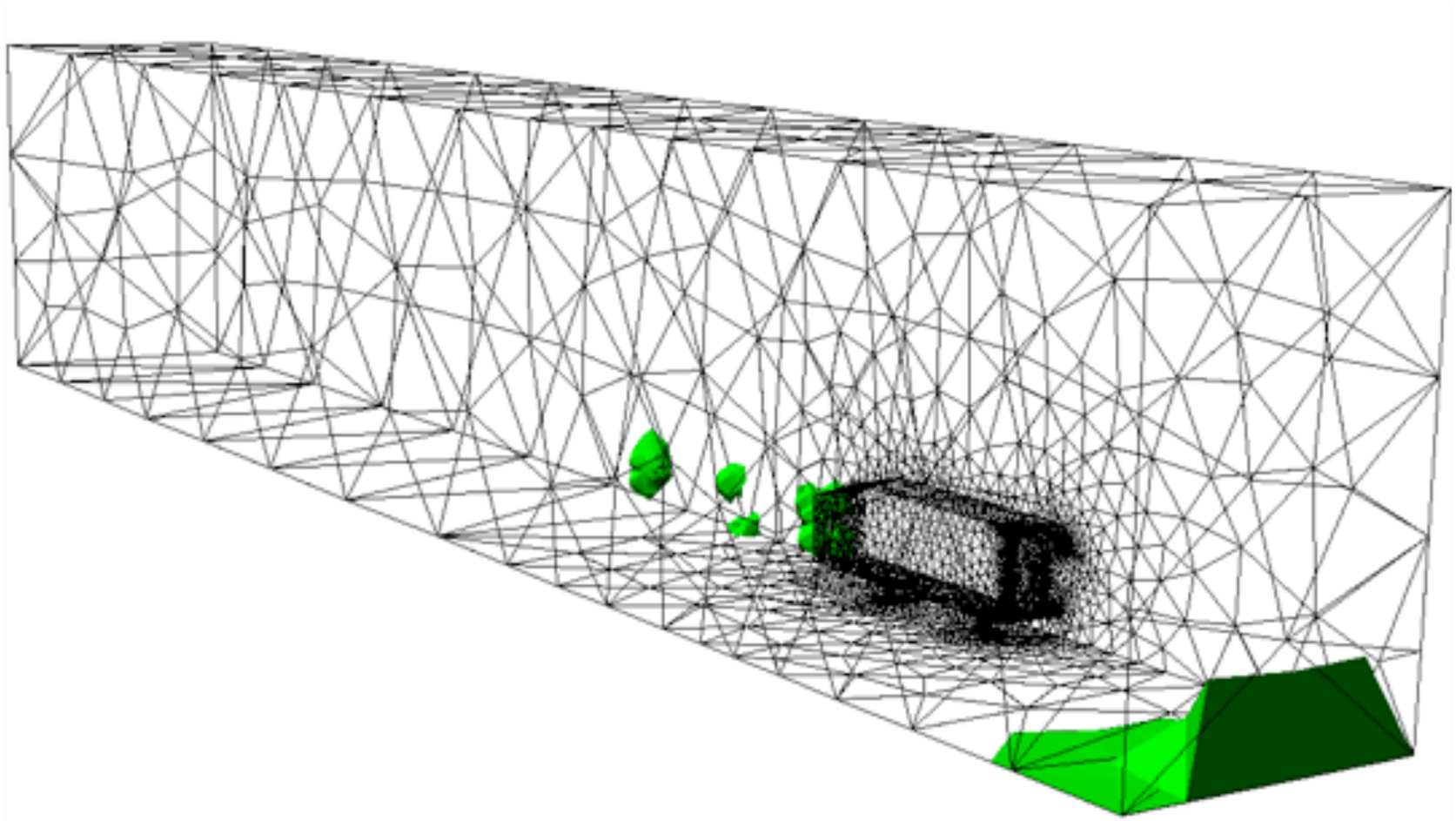
2. $J^{(k)}\,\Phi_y$, with $F_r = J_r^{(k)}$ and $Z = \Phi_J$

Reduced mesh: CFD, 2nd order flux



- $R^{(k)}$ and $J^{(k)}\,\Phi_y$ computed at sample nodes
- To do this, must compute the state at:
  1. Sample nodes
  2. Neighbors of sample nodes
  3. Neighbors of neighbors of sample nodes
- In general, this depends on the Jacobian's sparsity pattern

Reduced mesh: CFD



- Only compute the physics in isolated mesh clusters
- Gappy POD "fills in" the rest of the mesh

System approximation consistency

Proposition

The system approximation is consistent if Φ_R and Φ_J are POD bases computed with snapshots that satisfy:

1. $R^{(k)}$ from the Model II simulation is a snapshot for Φ_R
2. $J^{(k)}\,\Phi_y\,\mathrm{p}^{(k)}$ from the Model II simulation is a snapshot for Φ_J
3. Each column of $J^{(k)}\,\Phi_y$ from the Model II simulation is a snapshot for Φ_J

- Leads to hierarchy of snapshot collection procedures characterized by tradeoffs between consistency and offline cost/storage

Snapshot collection hierarchy

ID	0	1	2	3
Snapshots for y	y_I-$y_I^{(0)}$	y_I-$y_I^{(0)}$	y_I-$y_I^{(0)}$	y_I-$y_I^{(0)}$
Snapshots for $R^{(k)}$	$R_I^{(k)}$	$R_{II}^{(k)}$	$R_{II}^{(k)}$	$R_{II}^{(k)}$
Snapshots for $J^{(k)}\Phi_y$	$R_I^{(k)}$	$R_{II}^{(k)}$	$[J^{(k)}\Phi_y p^{(k)}]_{II}$	$[J^{(k)}\Phi_y]_{II}$
# simulations	1	2	2	2
# snapshots per Newton iteration	1	1	2	1+ n_y
consistency conditions satisfied	none	1	1, 2	1, 2, 3

- $()_I$, $()_{II}$: snapshot saved during Model I, II
- $()^{(k)}$: snapshot saved at each Newton iteration

Snapshot collection hierarchy

ID	0	1	2	3
Snapshots for y	y_I-$y_I^{(0)}$	y_I-$y_I^{(0)}$	y_I-$y_I^{(0)}$	y_I-$y_I^{(0)}$
Snapshots for $R^{(k)}$	$R_I^{(k)}$	$R_{II}^{(k)}$	$R_{II}^{(k)}$	$R_{II}^{(k)}$
Snapshots for $J^{(k)}\Phi_y$	$R_I^{(k)}$	$R_{II}^{(k)}$	$[J^{(k)}\Phi_y p^{(k)}]_{II}$	$[J^{(k)}\Phi_y]_{II}$
# simulations	1	2	2	2
# snapshots per Newton iteration	1	1	2	1+ n_y
consistency conditions satisfied	none	1	1, 2	1, 2, 3

x Procedure 0: satisfies no consistency conditions

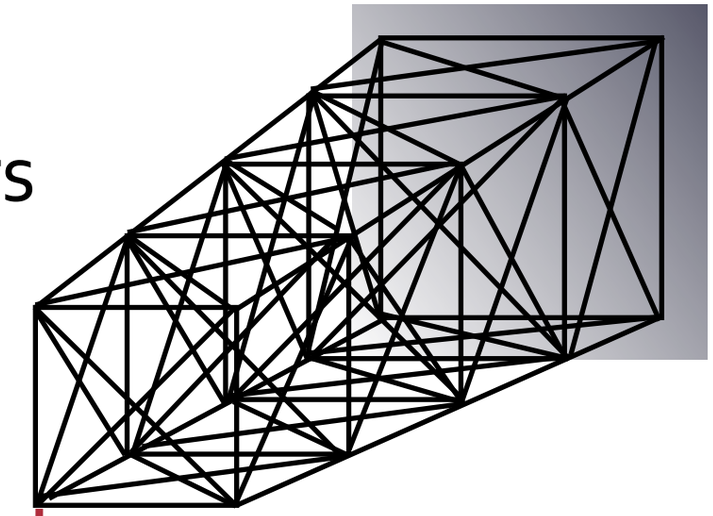
x Procedure 3: consistent, but prohibitive cost

✓ Procedure 2: similar cost as 1, more consistency conditions

Performance assessment

- Geometrically nonlinear structural dynamics

16 nonlinear 3D bars per bay

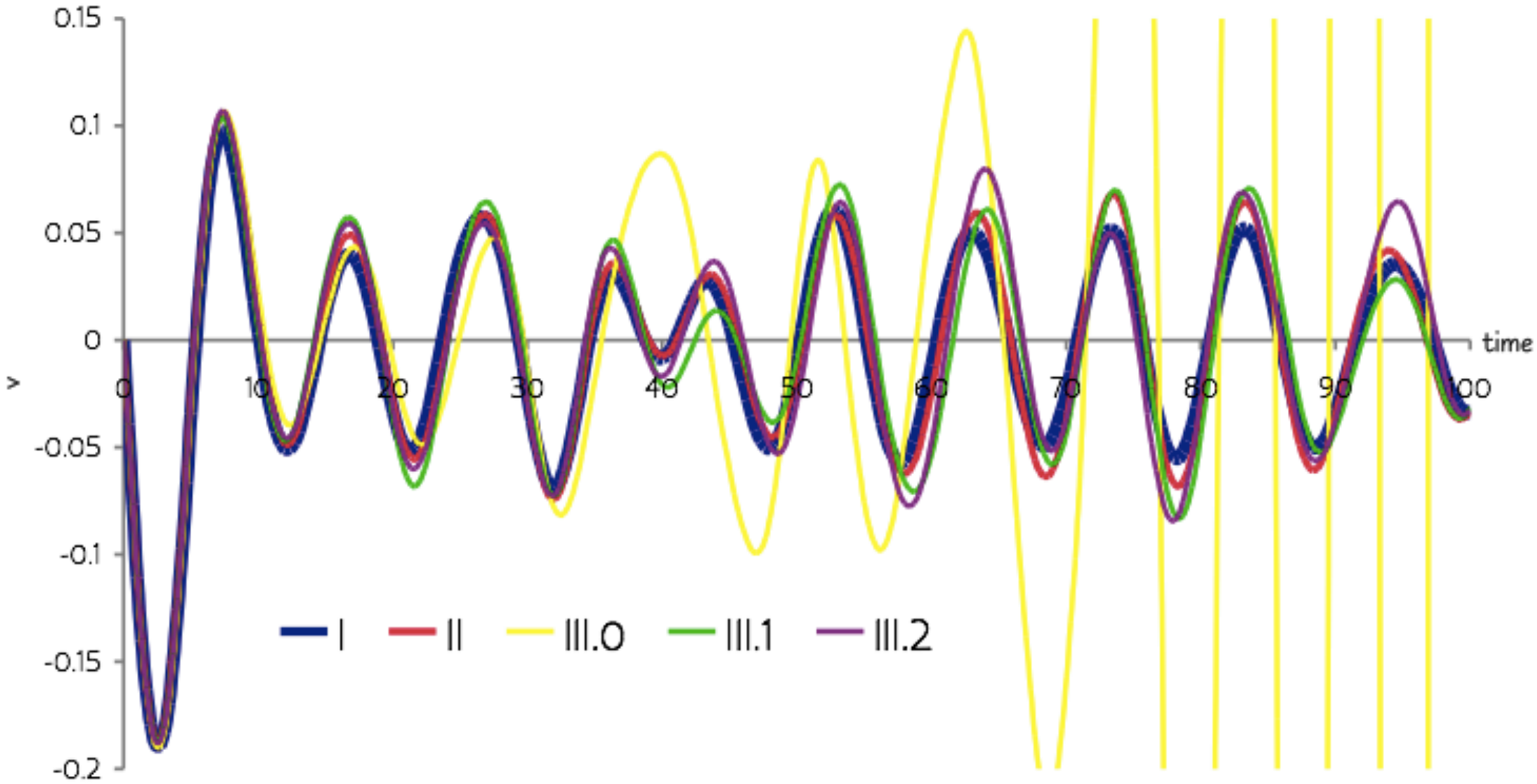


initial displacement

- Full-order model: 12,000 degrees of freedom (1000 bays)
- Output of interest: downward velocity at tip
- Truncation of POD bases
  - Φ_y: n_y=26 (99% of y_l-$y_l^{(0)}$ snapshot energy)
  - Φ_R, Φ_y: n_R=n_J=28 (99% of $R_l^{(k)}$ snapshot energy)

Consistency importance

- $|I|=28$ (interpolation)



- III.0, which satisfies no consistency conditions, is unstable

Consistency importance

consistency conditions satisfied

Model	e_l (%)	Total Newton it	Speed-up over I
II	3.44	601	1.03
III.0	unstable	–	–
III.1	6.31	603	158
III.2	6.01	891	122

- Model II: Good accuracy, but insufficient speed-up
- Model III.0: inaccurate, perhaps because not consistent
- Models III.1, III.2: accurate, promising speed-ups

➡ Meeting more consistency conditions improves accuracy/stability

Least-squares v. interpolation

- n_y=26, n_R=n_J=28

	$\|I\|$	Relative error e_I (%)			Speed-up over Model I		
		III.0	III.1	III.2	III.0	III.1	III.2
interpolation	28	unstable	6.31	6.01	120	158	122
least squares	42	4.95	3.77	3.63	137	152	119
	56	4.03	3.50	3.44	129	136	116

- Increasing $|I|$ improves accuracy, even stabilizing III.0

➡ Least-squares reconstruction performs better than interpolation

System-approximated Gauss-Newton

- Conclusions
  - Carefully build approximations to be consistent & optimal
  - Least-squares Petrov-Galerkin led to accuracy
  - System approximation led to low computational complexity
- Future work
  - Currently implementing in AERO-F flow solver

Repeated analyses

Real-time analysis

- Fast-turnaround design
- "In-field" analysis
- Model predictive control

Repeated analyses

- Nonlinear analysis
- Design optimization
- Parameter space sampling

Multiple objectives: error and cost

Real-time analysis

Online cost more important:

$$\begin{aligned} &\text{minimize} && \text{error} \\ &\text{subject to} && \text{online cost} \leq \tau \end{aligned}$$

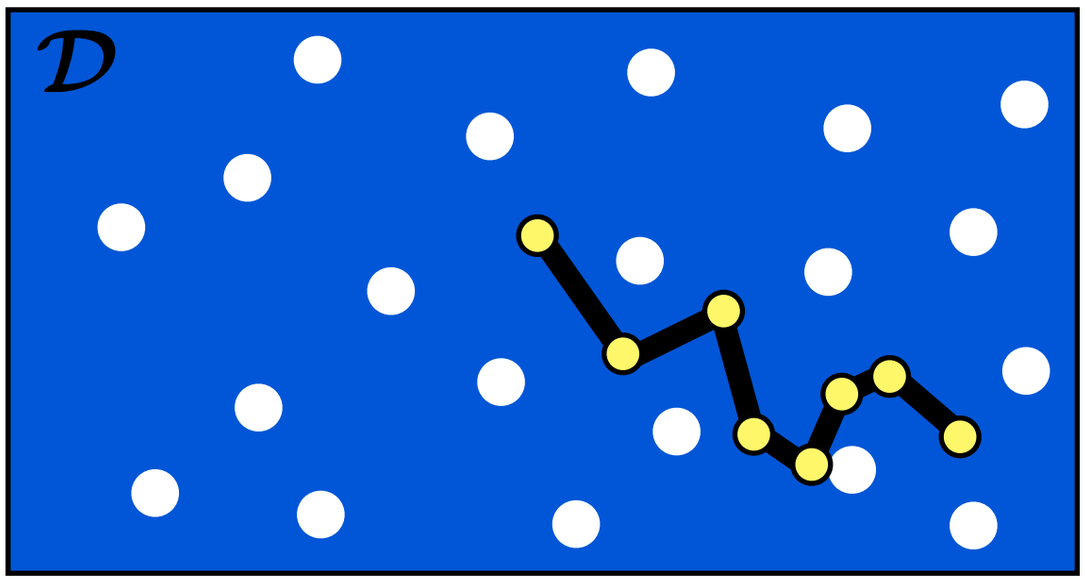
Repeated analyses

Error more important:

$$\begin{aligned} &\text{minimize} && \text{total cost} \\ &\text{subject to} && \text{error} \leq \epsilon \end{aligned}$$

Offline/online strategy

	Real-time analysis	Repeated analyses
1) Offline ◉ Sample input space ◉ Build surrogate model	✓ High offline cost okay	X May preclude total cost savings
2) Online ◉ Analysis with surrogate	✓ Very low online cost	X May not be accurate enough



$\mathcal{D}$ Input space

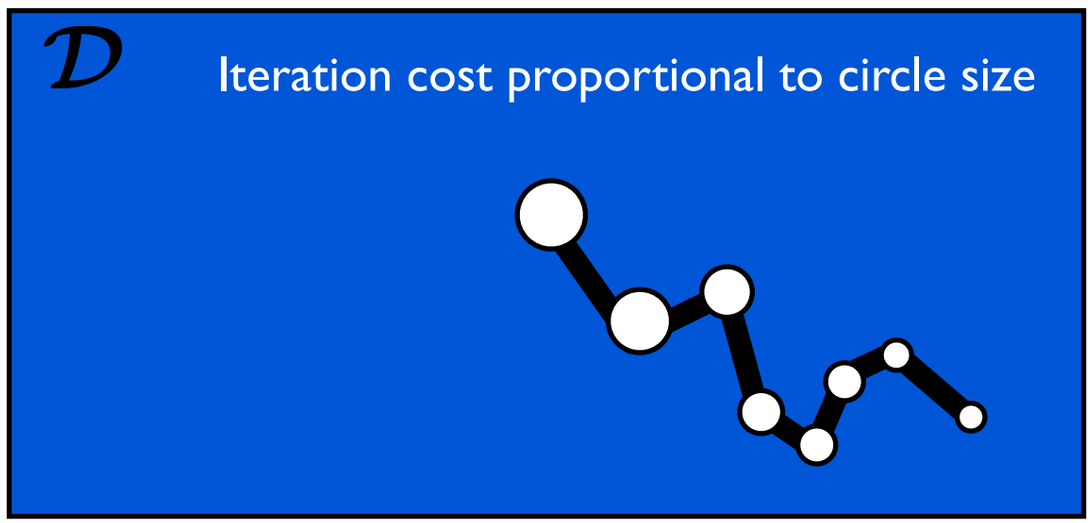
○ Full-order model evaluation

○ Surrogate model evaluation

▬ Online trajectory

Methodology

- Guiding philosophy
  1. Avoid extra computations
  2. Fully exploit data generated from previous analyses
  3. Use data to accelerate solver convergence to required tolerance
- Procedure
  1. Execute only required analyses (no offline sampling)
  2. Build a POD basis on-the-fly
  3. Use the POD basis within an iterative method to accelerate analyses



$\mathcal{D}$ Input space

○ Accelerated analyses

▬ Trajectory

Repeated analyses

- For $i=1,\ldots,n_{mat}$ and $j=1,\ldots,n_{RHS}$, solve

$$A^{(i)} x^{(i,j)} = b^{(i,j)}$$

  - $A^{(i)} \in \mathbb{R}^{N \times N}$ sparse, SPD, and "nearby"

- Solve each system by preconditioned conjugate gradient (PCG)

$$\tilde{x}^{(i,j)} = \arg\min_{v \in \mathcal{S}} \left\| x^{(i,j)} - v \right\|_{A^{(i)}}$$

  - $\mathcal{S} = \mathcal{K}_k$: k-dimensional Krylov subspace of $\mathbb{R}^N$
  - Current PCG acceleration approach: augmented CG methods

Augmented CG methods

$$\mathcal{S} = \mathcal{K}_k + \mathcal{Y}$$

- Choices of $\mathcal{Y}$

  1. All previous search directions

  - Multi-RHS [O'Leary, 1980; Saad, 1987; Farhat *et al.*, 1994; Erhel, *et al.*, 2000]
  - Multi-matrices [Rey, 1994; Roux, 1995; Farhat, 1995; Farhat *et al.*, 2000, Risler *et al.*, 2000]

  x Large cost/storage for many linear systems

  2. Approximated extreme eigenvectors (deflation)

  - Multi-RHS [Chapman *et al.*, 1997; Saad *et al.*, 2000]
  - Multi-matrices [Rey *et al.*, 1998; Parks *et al.*, 2007]

  x Only effective if a few eigenvalues hamper convergence

- Goals: 1) Feasibility

  2) Effective in the general case

Proper orthogonal decomposition

- Given 1) n_w snapshots w_j, 2) n_w weights γ_j, 3) a Θ-norm
- The first $n \leq n_w$ POD vectors satisfy:

$$\text{span}\{\phi_i\}_{i=1}^{n} = \arg \min_{\mathcal{S} \in \mathcal{G}(n,N)} \sum_{j=1}^{n_w} \| \left(I - \Pi_{\mathcal{S}}^{\Theta} \right) \gamma_j w_j \|_{\theta}^2$$

  - $\Pi_{\mathcal{S}}^{\Theta}$: Θ-orthogonal projection onto $\mathcal{S}$
  - $\mathcal{G}(n, N)$: set of n-dimensional subspaces of $\mathbb{R}^N$
  - $\Phi(n) \equiv [\phi_1 \ \cdots \ \phi_n]$: n-dimensional POD basis in matrix form
- Key properties

1. Optimal ordering
   - First n POD basis vectors span optimal n-dimensional space
2. Θ -orthonormality: $\Phi(n)^T \Theta \Phi(n) = I$

POD strategy

- Align POD with minimizing the error for a nearby "target system"

$$\bar{A}\bar{x}^{(j)} = \bar{b}^{(j)}$$

- To do this, choose three ingredients to be:

1. Snapshots w_j: all previous search directions
2. Weights γ_j: estimate the solution for the target problem

$$\bar{x}^{(j)} \approx \bar{x}^{(j)}_{\text{est}} = \sum_{j=1}^{n_w} \gamma_j w_j$$

3. Θ : matrix for the target problem $\bar{A}$

POD bases and key properties

- Compute one POD basis for each RHS

$$\Phi_j(n) \equiv [\phi_1^{(j)} \ \cdots \ \phi_n^{(j)}]$$

- Key properties

1. Optimal ordering

➡ POD vectors optimally ordered to minimize error at target

2. $\bar{A}$ -orthonormality

$$\Phi_j(n)^T \bar{A} \Phi_j(n) = I$$

➡ $\Phi_j(n)^T A^{(i)} \Phi_j(n) \approx I$ for $A^{(i)}$ near $\bar{A}$

POD-augmented CG algorithm

$$\mathcal{S} = \mathcal{K}_k + \text{range}(\Phi_j(n))$$

1. Directly solve n_1-dimensional reduced equations (n_1 small)

$$\Phi_j(n_1)^T A^{(i)} \Phi_j(n_1) \hat{x} = \Phi_j(n_1)^T b^{(i,j)}$$

$$\tilde{x}_1^{(i,j)} = \Phi_j(n_1)\hat{x}$$

- Accurate (Property 1) and low cost (n_1 small)

2. Iteratively solve n-dimensional reduced equations ($n > n_1$)

$$\Phi_j(n)^T A^{(i)} \Phi_j(n) \hat{x} = \Phi_j(n)^T \left(b^{(i,j)} - A^{(i)} \tilde{x}_1^{(i,j)} \right)$$

$$\tilde{x}_2^{(i,j)} = \tilde{x}_1^{(i,j)} + \Phi_j(n_1)\hat{x}$$

- Use augmented CG without forming reduced matrix
- More accurate (Property 1) and low cost (Property 2)

POD-augmented CG algorithm

3. Iteratively solve full state equations to specified tolerance

$$A^{(i)}\hat{x} = b^{(i,j)} - A^{(i)}\tilde{x}_2^{(i,j)}$$

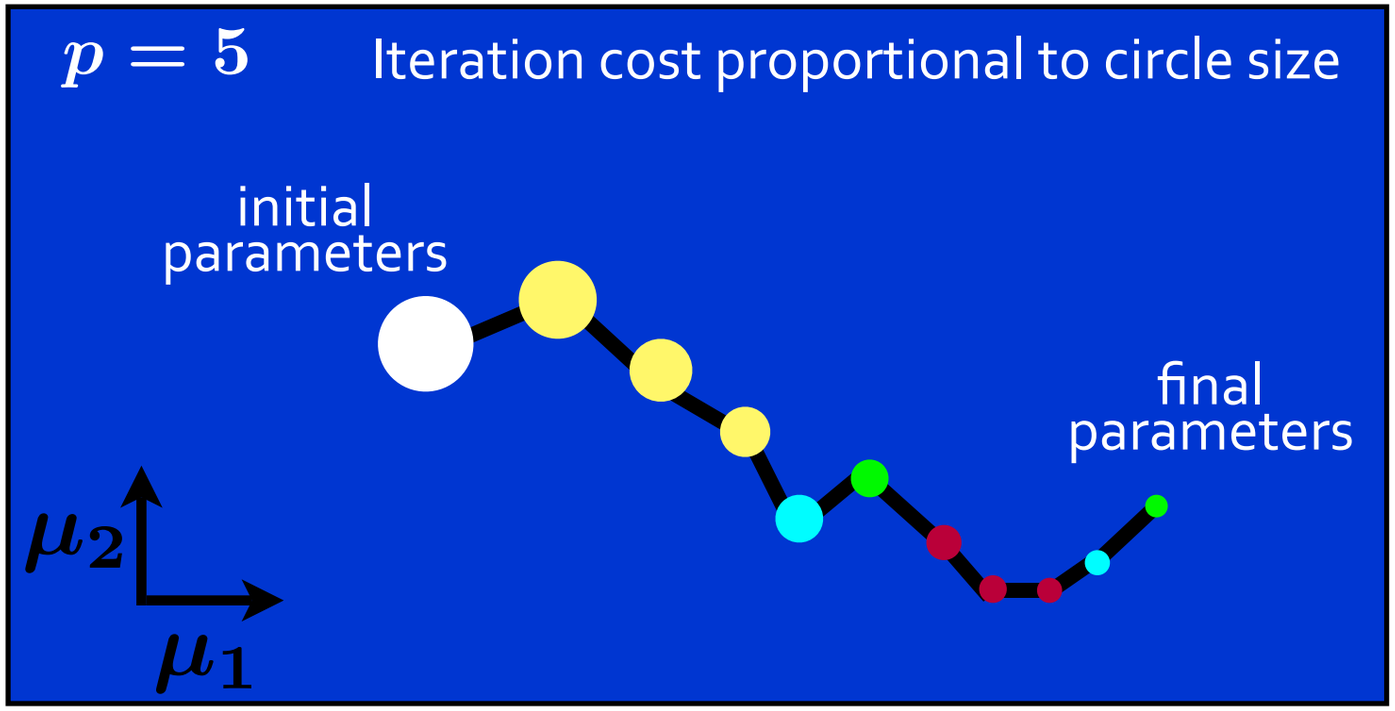
$$\tilde{x}^{(i,j)} = \tilde{x}_2^{(i,j)} + \hat{x}$$

- Maintain $A^{(i)}$-conjugacy to old search directions using augmented PCG [Farhat *et al.*, 1994]

- Multiple-RHS
  - Sequentially execute Stages 1-3 for j=1,…,n_{RHS}
  - Stage 1 approximation space includes search directions from all previous RHS

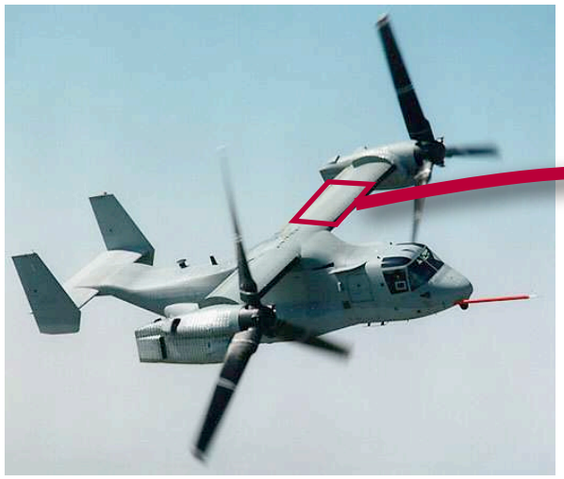
Implementation

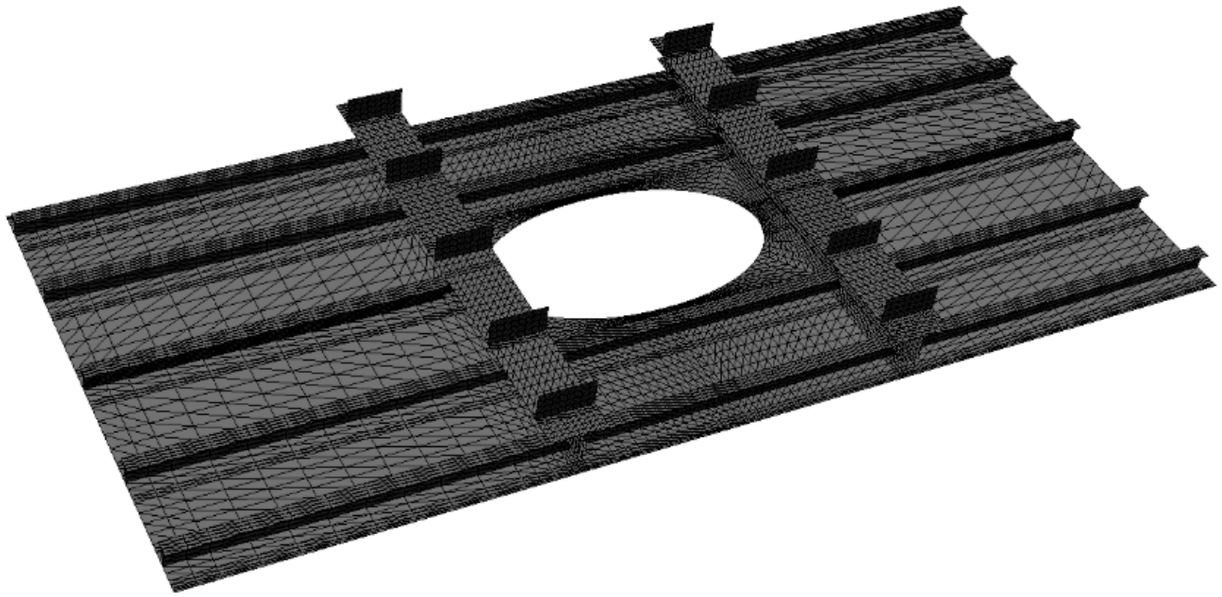
- p = # mat before recomputing POD basis (data compression)

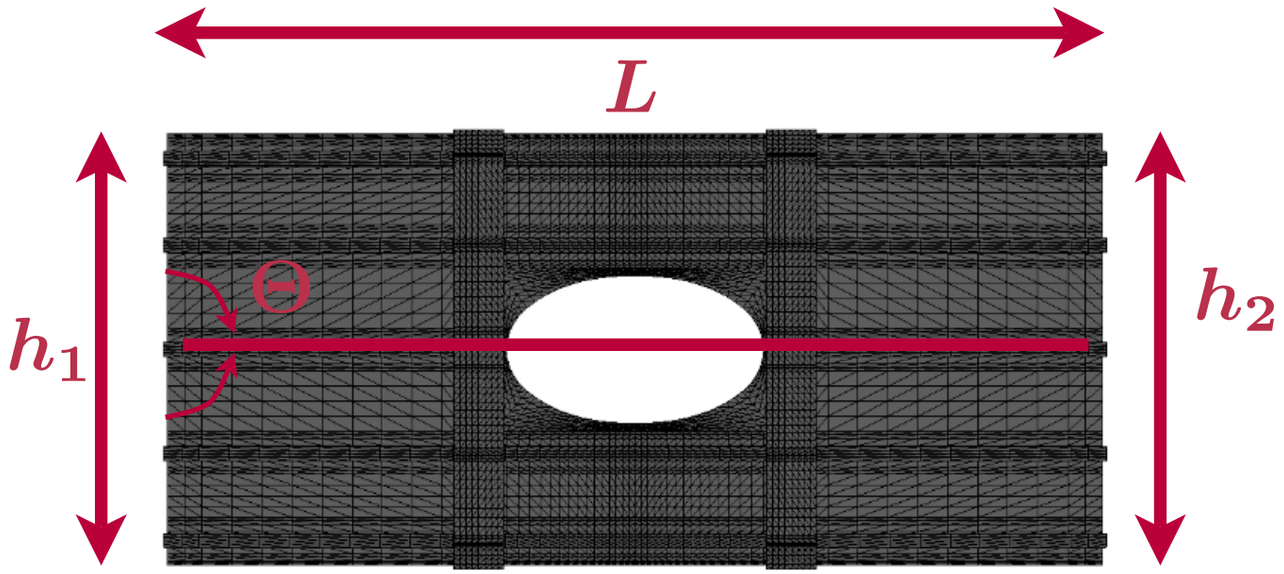


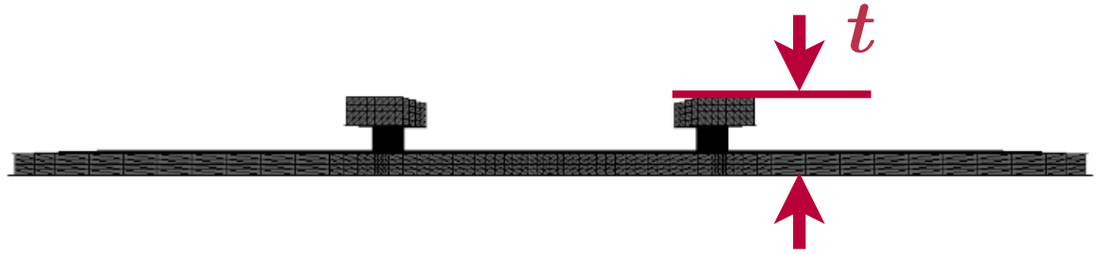
	i	Stage 1 basis	Stage 2 basis	Compute POD?
○	1			
●	$2 : p-1$	$\boldsymbol{W}$		
●	np	$\boldsymbol{W}$		✓
●	$np+1$	$\boldsymbol{\Phi}(n_1)$	$\boldsymbol{\Phi}(n)$	
●	$np+2 : (n+1)p-1$	$[\boldsymbol{W}, \boldsymbol{\Phi}(n_1)]$	$[\boldsymbol{W}, \boldsymbol{\Phi}(n)]$	

Example: V-22 Osprey wing panel





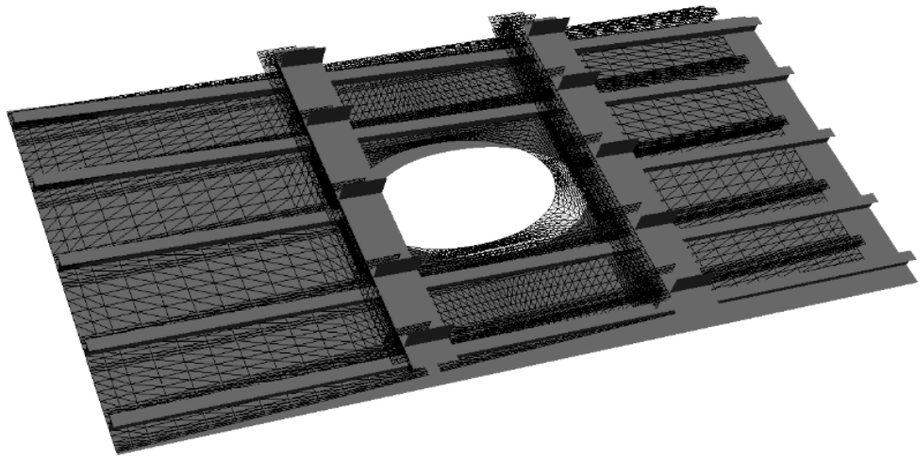




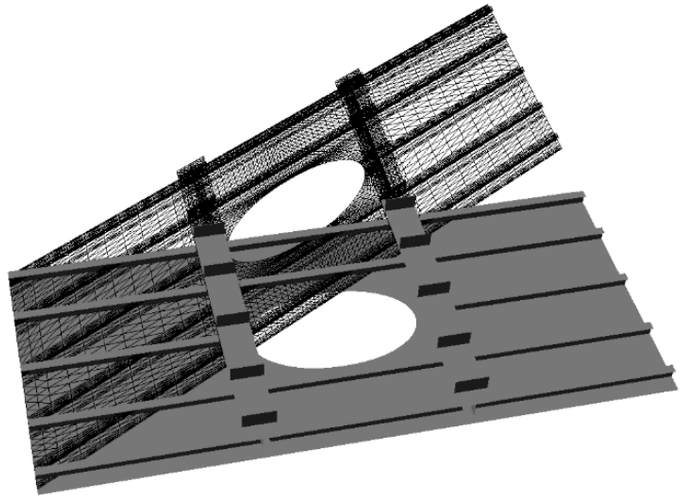
- Finite element model with 56,916 degrees of freedom
- 13 design variables (5 shape, 8 material)

Example: V-22 Osprey wing panel

- Problem Statement: n_{mat}=11, n_{RHS}=14
  - Given: 10 randomly-chosen (not necessarily nearby) previously-queried designs and designs for i=11:



Design *A*

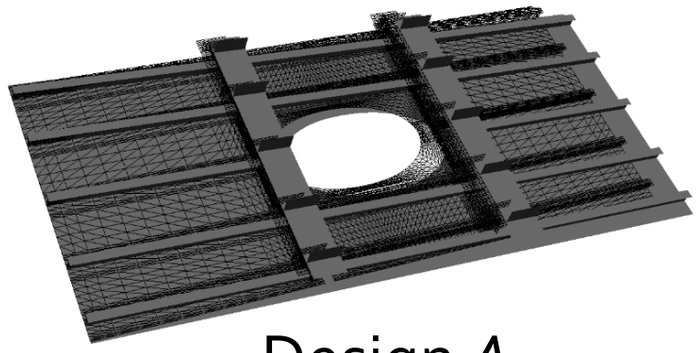


Design *B*

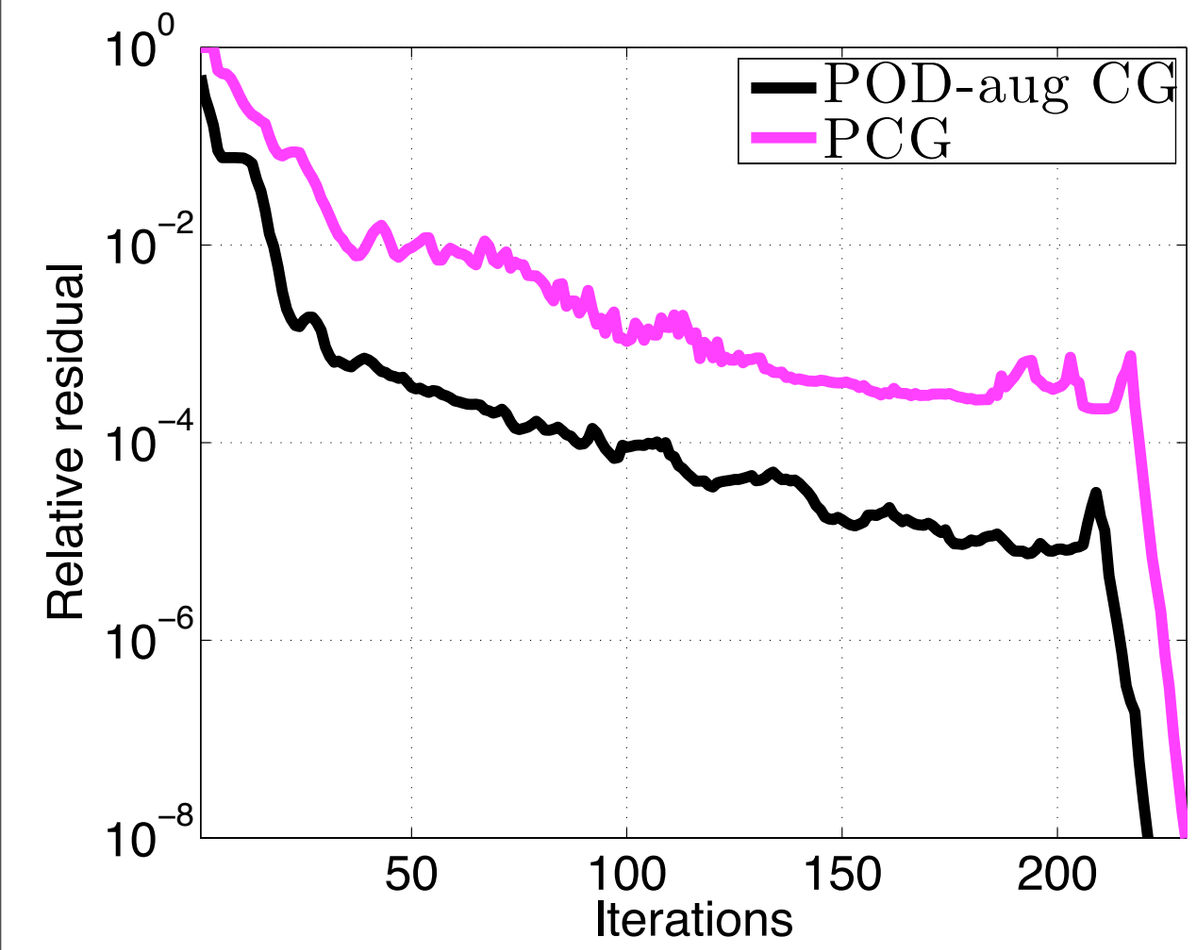
- Compute: $\tilde{x}^{(11,j)},\ j = 1, \ldots, n_{\text{RHS}}$ satisfying

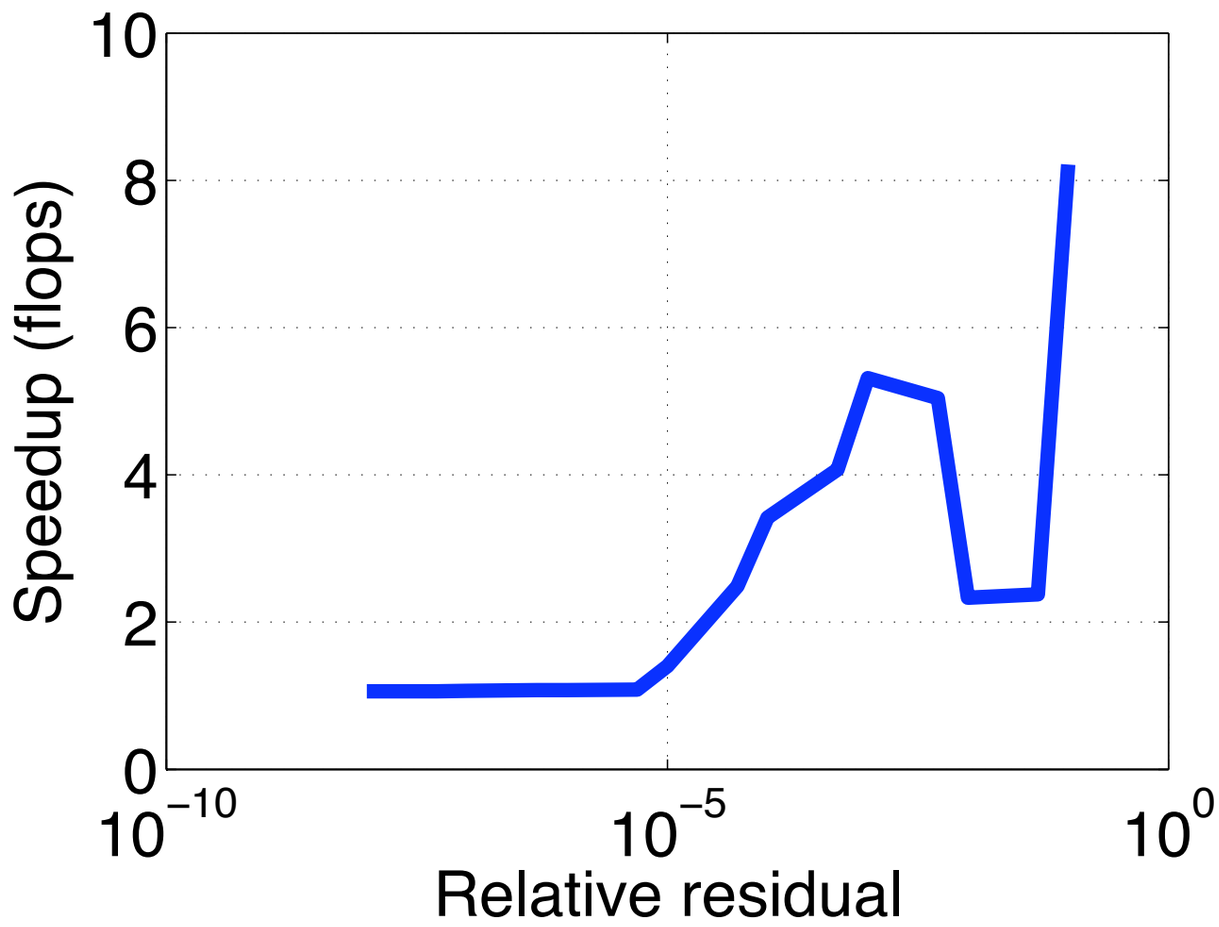
$$\frac{\| b^{(11,j)} - A^{(11)} \tilde{x}^{(11,j)} \|_2}{\| b^{(11,j)} \|_2} \leq \epsilon$$

Results: Design B, $n_{RHS} = 1$

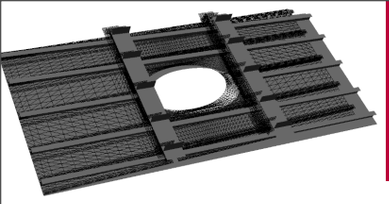


Design *A*



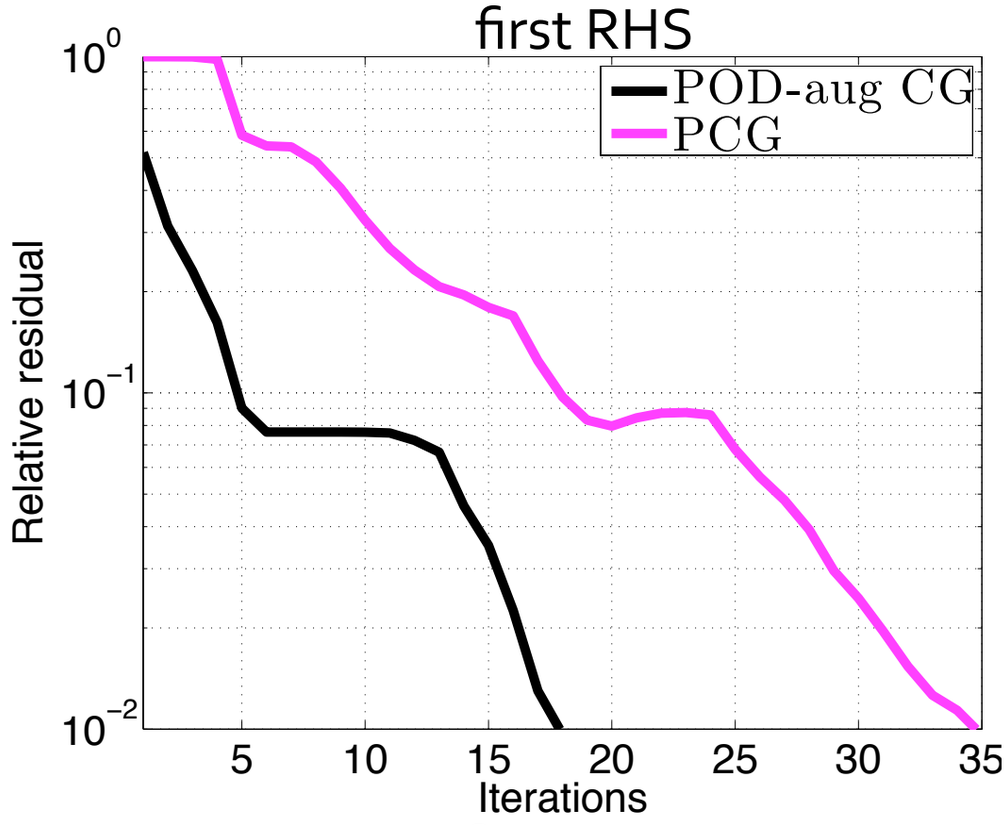


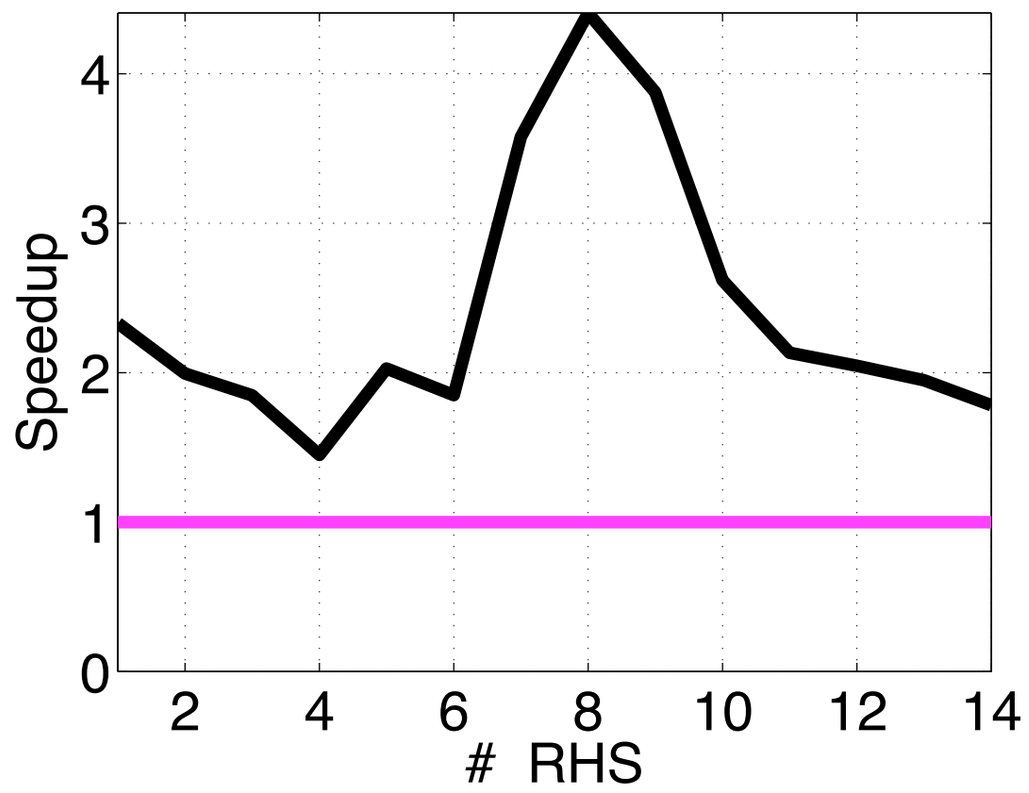
Results: Design A, $n_{RHS} = 14$



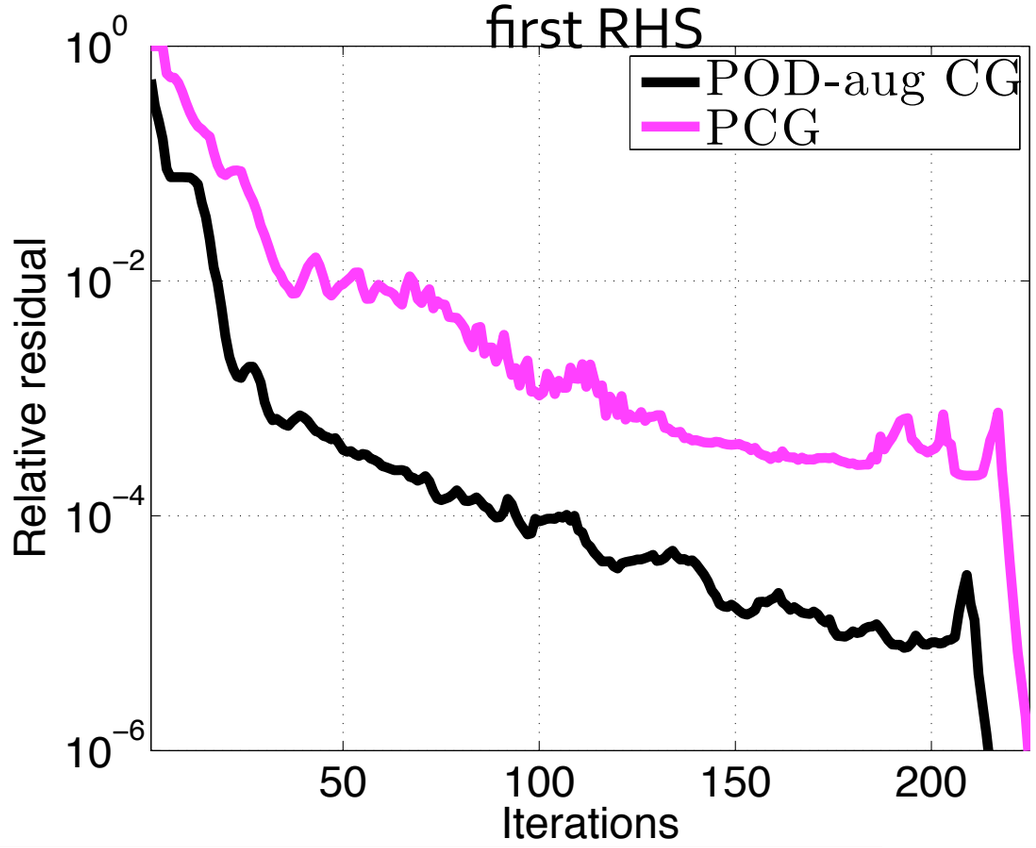
Design *A*

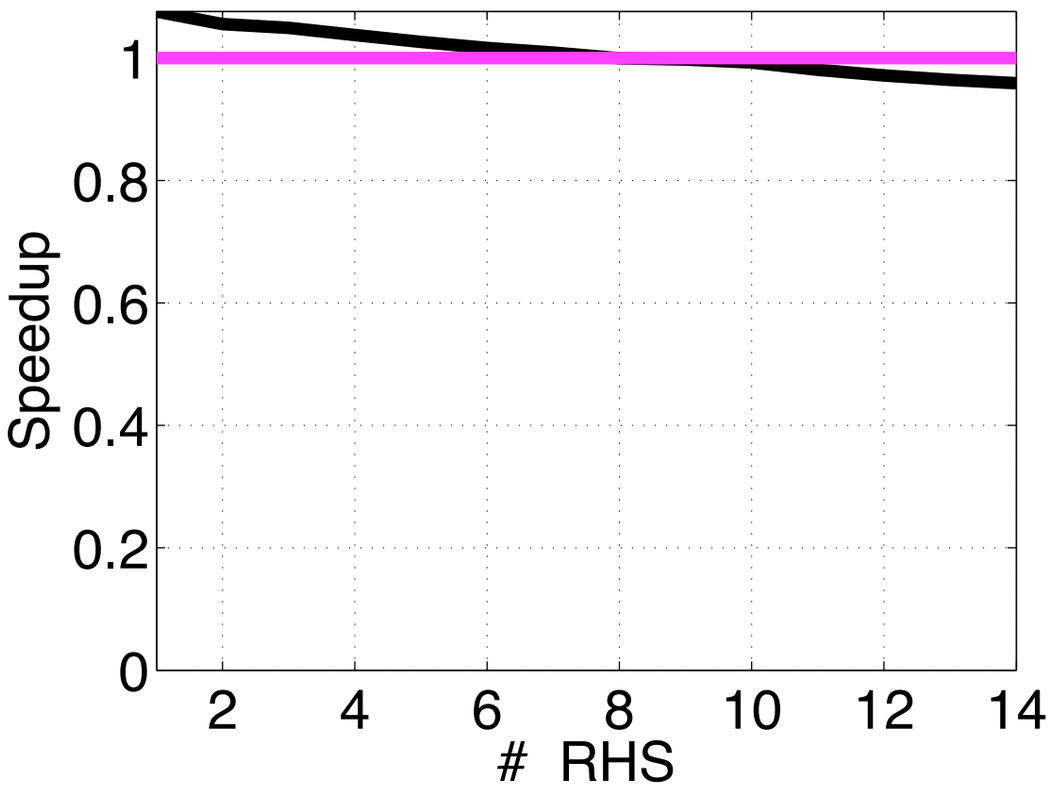
$\epsilon = 10^{-2}$



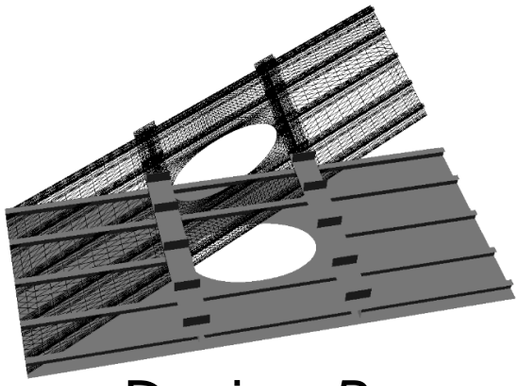


$\epsilon = 10^{-6}$

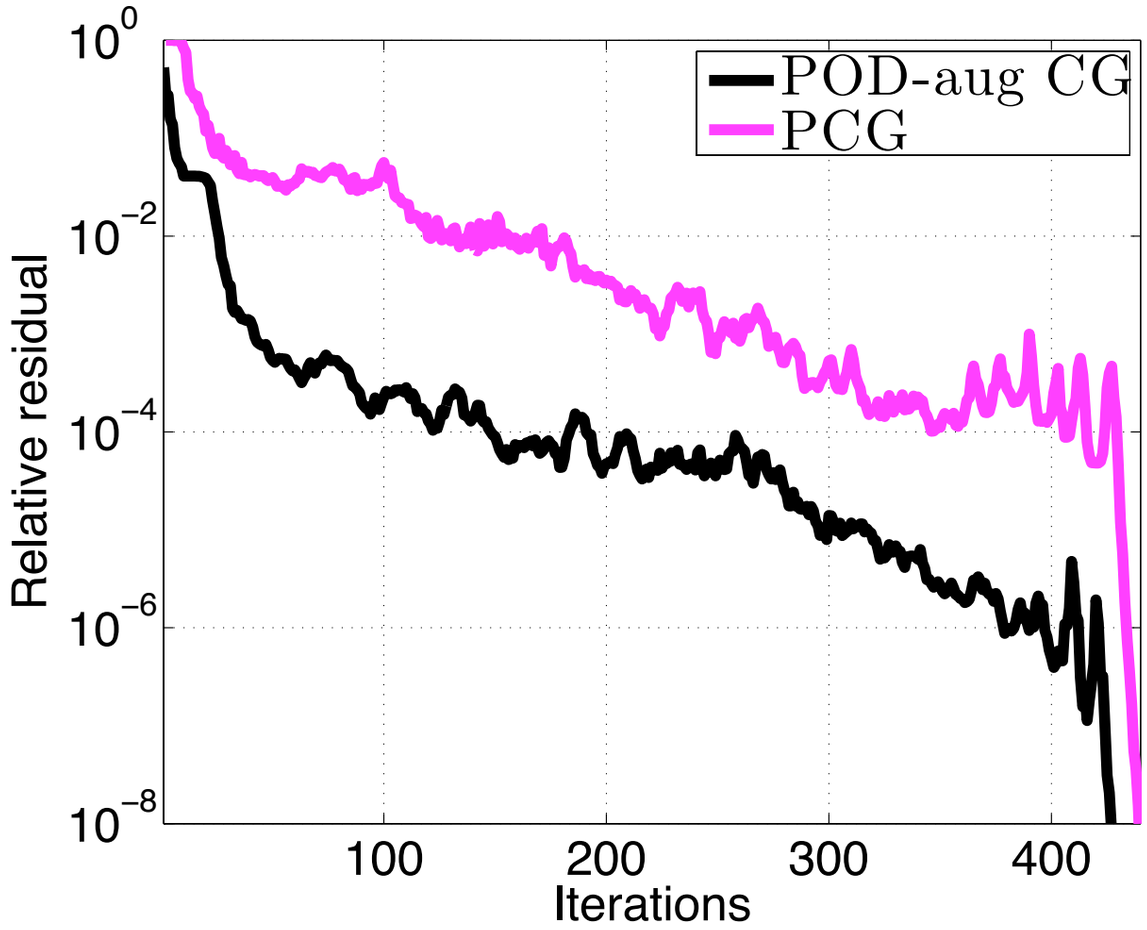


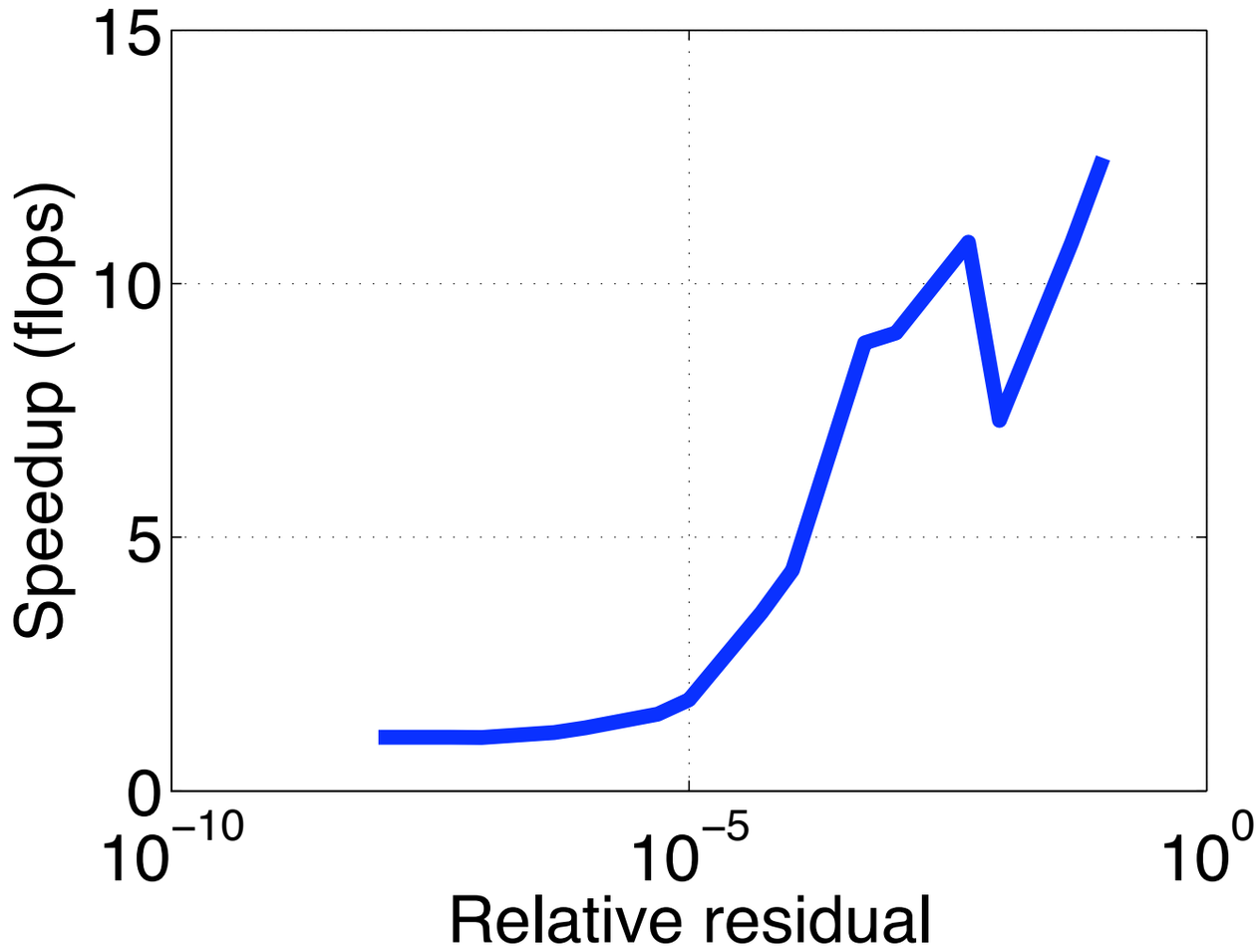


Results: Design B, $n_{RHS} = 1$

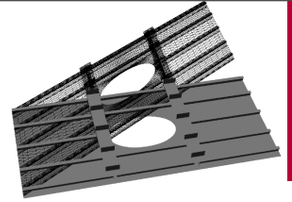


Design *B*



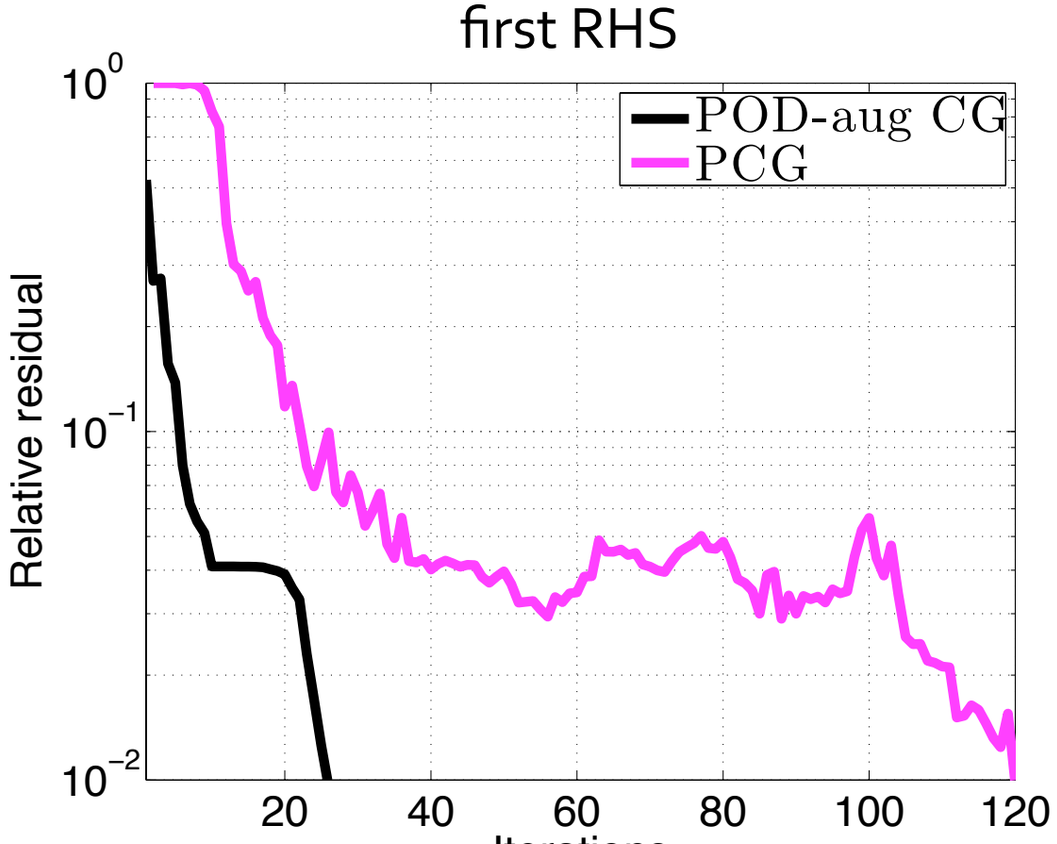


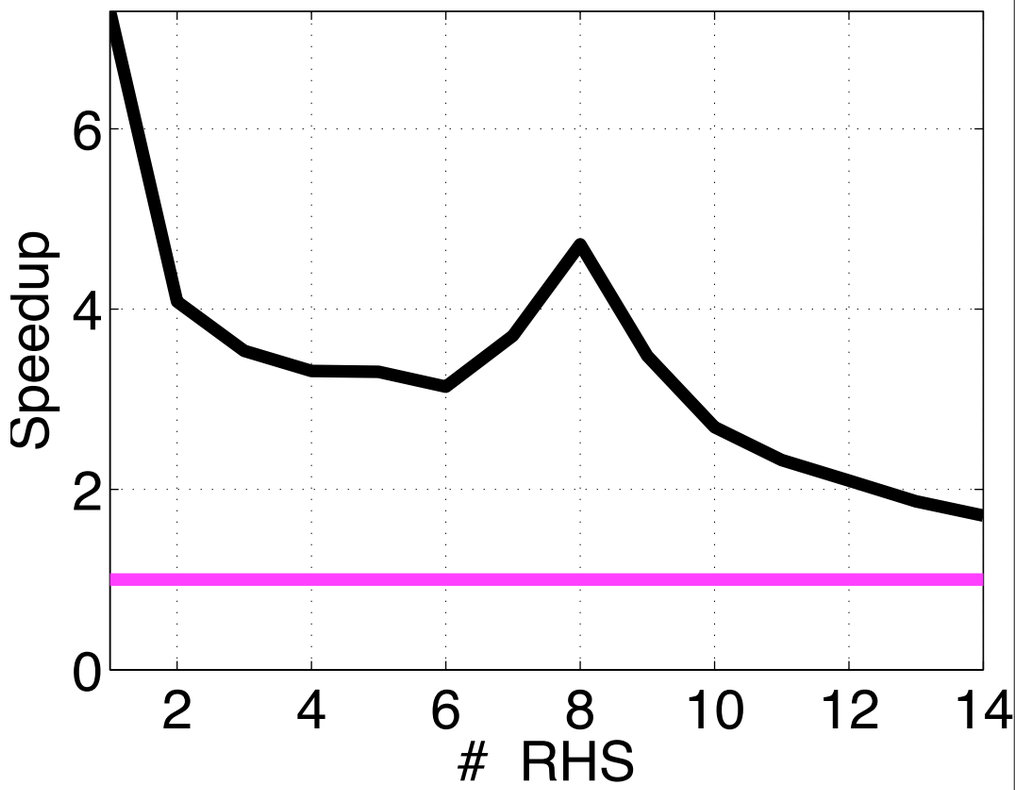
Results: Design B, $n_{RHS} = 14$



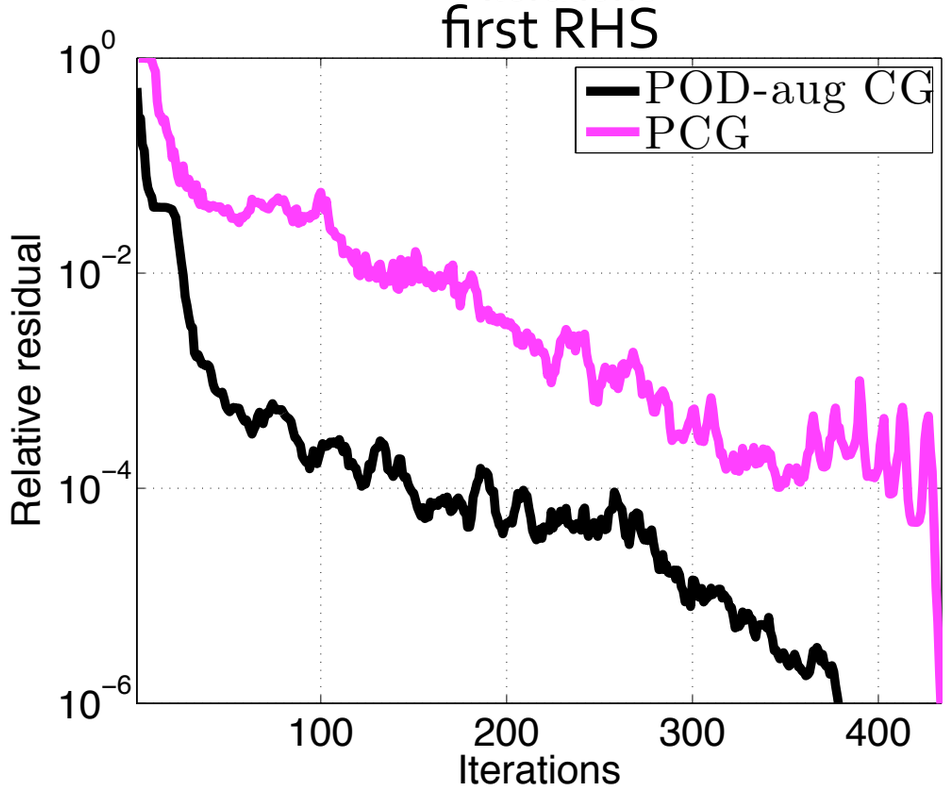
Design *B*

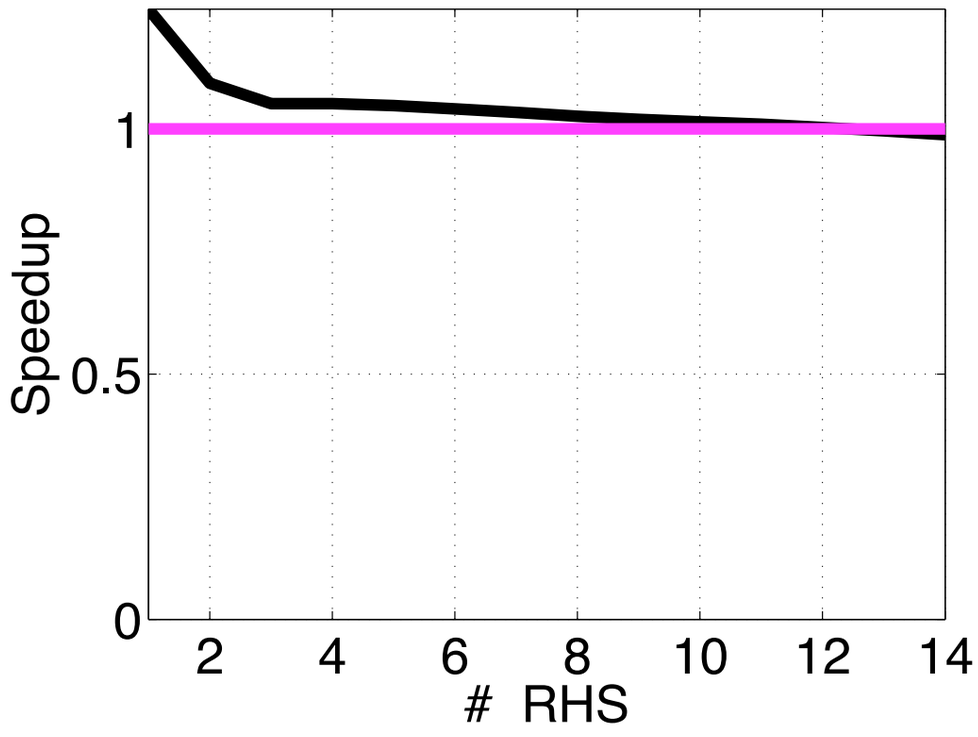
$\epsilon = 10^{-2}$





$\epsilon = 10^{-6}$





Conclusions: POD-augmented CG

- A POD-based augmented conjugate gradient method
  - Accelerates convergence to inexact tolerances
  - Efficiency due to choice of POD snapshots, weights, and norm
  - Highest speedups for modest tolerances, few right-hand sides

- Future work
  - Combine with other augmented CG approaches
    - Deflation: include approximated eigenvectors in stage 1
  - Extend to systems with non-SPD matrices

Questions?

- Thanks to Charbel Bou-Mosleh, David Amsallem, and Julien Cortial for their contributions to this research.

- References

  - K. Carlberg, C. Bou-Mosleh, and C. Farhat, "Efficient nonlinear model reduction via a least-squares Petrov-Galerkin projection and compressive tensor approximations," International Journal of Numerical Methods in Engineering, to appear, 2010.
  - K. Carlberg and C. Farhat, "An Adaptive POD-Krylov Reduced-Order Model for Structural Optimization," Proceedings of 8th World Congress on Structural and Multidisciplinary Optimization, Lisbon, Portugal, June 1–5, 2009.