Efficient Solution of Parameterized Partial Differential Equations using Reduced-Order Models

Howard C. Elman
University of Maryland at College Park

Collaborators: **Qifeng Liao**, MIT
Virginia Forstall, Maryland

1. Preliminary: Spectral Methods for PDEs with Uncertain Coefficients
   - Problem Definition
   - Solution Methods
2. Reduced Basis Methods
   - Offline Computations
   - Reduced Problem
   - Reduced Problem: Costs
   - Reduced Problem: Capturing Features of Model
3. Reduced Basis + Sparse Grid Collocation
   - Introduction
   - Performance for Diffusion Equation
   - Application to the Navier-Stokes Equations
4. Iterative Solution of Reduced Problem
   - Introduction
   - Implementation
   - Performance
   - Trends and Analysis
5. Concluding Remarks

Partial Differential Equations with Uncertain Coefficients

Examples:

Diffusion equation: $-\nabla \cdot (a(\mathbf{x},\xi)\nabla u) = f$

Navier-Stokes equations: $-\nabla \cdot (a\,(\mathbf{x},\xi)\,\nabla \vec{u}) + (\vec{u}\cdot\nabla)\vec{u} + \nabla p = \vec{f}$
$\nabla \cdot \vec{u} = 0$

Posed on $\mathcal{D} \subset \mathbb{R}^d$ with suitable boundary conditions

Sources: models of diffusion in media with uncertain permeabilities
multiphase flows

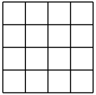
Uncertainty / randomness:

$a = a(\mathbf{x},\xi)$ is a *random field*: for each fixed $x \in \mathcal{D}$, $a(\mathbf{x},\xi)$ is a random variable depending on m random parameters $\xi_1,\ldots,\xi_m$

In this study: $a(\mathbf{x},\xi) = a_0(\mathbf{x}) + \sum_{r=1}^m a_r(\mathbf{x})\,\xi_r$

Possible sources:

Karhunen-Loève expansion or Piecewise constant coefficients on $\mathcal{D}$



The Stochastic Galerkin Method

Standard weak diffusion problem: find $u \in H_E^1(\mathcal{D})$ s.t.

$$a(u,v) = \int_{\mathcal{D}} a\nabla u \cdot \nabla v dx = \int_{\mathcal{D}} f\, v\, dx \quad \forall\, v \in H_0^1(\mathcal{D})$$

Extended (*stochastic*) weak formulation: find $u \in H_E^1(\mathcal{D}) \otimes L_2(\Omega)$ s.t.

$$\underbrace{\int_\Omega \int_{\mathcal{D}} a\nabla u \cdot \nabla v\, dx\, dP(\Omega)}_{\int_\Gamma \int_{\mathcal{D}} a(\mathbf{x},\boldsymbol{\xi})\, \nabla u \cdot \nabla v\, d\mathbf{x}\, \rho(\boldsymbol{\xi})\, d\boldsymbol{\xi}} = \underbrace{\int_\Omega \int_{\mathcal{D}} f\, v\, dx\, dP(\Omega)}_{\int_\Gamma \int_{\mathcal{D}} f\, v\, dx\, \rho(\boldsymbol{\xi})\, d\boldsymbol{\xi}} \quad \forall\, v \in H_0^1(\mathcal{D}) \otimes L_2(\Omega)$$

$$(\Gamma = \boldsymbol{\xi}(\Omega))$$

- **Discretization** in physical space: $\mathcal{S}_E^{(h)} \subset H_E^1(\mathcal{D})$, basis $\{\phi_j\}_{j=1}^N$
  Example: piecewise linear "hat functions"
- **Discretization** in space of random variables: $\mathcal{T}^{(p)} \subset L^2(\Gamma)$, basis $\{\psi_\ell\}_{\ell=1}^M$
  Example: m-variate polynomials in $\boldsymbol{\xi}$ of total degree p

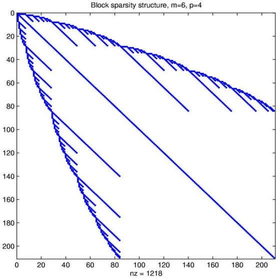
Discrete solution:

$u_{hp}(\mathbf{x},\boldsymbol{\xi}) = \sum_{j=1}^{N}\sum_{\ell=1}^{M} u_{j\ell}\phi_j(\mathbf{x})\psi_\ell(\boldsymbol{\xi})$

Requires solution of large coupled system

Matrix (right): $G_0 \otimes A_0 + \sum_{r=1}^{m} G_r \otimes A_r$

Stochastic dimension: $M = \binom{m+p}{p}$



(Ghanem, Spanos, Babuška, Deb, Oden, Matthies, Keese, Karniadakis, Xue, Schwab, Todor)

The Stochastic Collocation Method

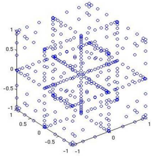
Monte-Carlo (sampling) method: find $u \in H^1_E(\mathcal{D})$ s.t.

$$\int_{\mathcal{D}} a(\mathbf{x}, \boldsymbol{\xi}^{(k)}) \nabla u \cdot \nabla v dx \quad \text{for all } v \in H^1_{E_0}(\mathcal{D})$$

for a collection of samples $\{\boldsymbol{\xi}^{(k)}\} \in L^2(\Gamma)$

Collocation (Xiu, Hesthaven, Babuška, Nobile, Tempone, Webster)

Choose $\{\boldsymbol{\xi}^{(k)}\}$ in a special way (sparse grids), then construct discrete solution $u_{hp}(\mathbf{x}, \boldsymbol{\xi})$ to interpolate $\{u_h(\mathbf{x}, \boldsymbol{\xi}^{(k)})\}$



Structure of collocation solution:

$$u_{hp}(\mathbf{x}, \boldsymbol{\xi}^{(k)}) := \sum_{\boldsymbol{\xi}^{(k)} \in \Theta_p} u_c(\mathbf{x}, \boldsymbol{\xi}^{(k)}) L_{\boldsymbol{\xi}^{(k)}}(\boldsymbol{\xi})$$

Advantages (vs. stochastic Galerkin):

- decouples algebraic system (like MC)
- applies in a straightforward way to nonlinear random terms

Properties of These Methods

For both Galerkin and collocation

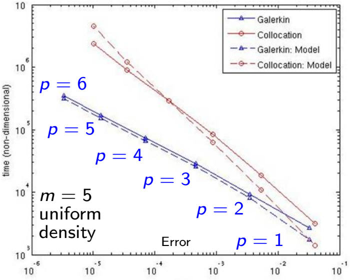
- Each computes a discrete function u_{hp}
- Moments of u estimated using moments of u_{hp} (cheap)
- Convergence: $\|E(u) - E(u_{hp})\|_{H_1(\mathcal{D})} \le c_1 h + c_2 r^p$, $r < 1$
  Exponential in polynomial degree
- Contrast with Monte Carlo:
  Perform N_{MC} (discrete) PDE solves to obtain samples $\{u_h^{(s)}\}_{s=1}^{N_{MC}}$
  Moments from averaging, e.g., $\hat{E}(u_h) = \frac{1}{N_{MC}} \sum_{s=1}^{N_{MC}} u_h^{(s)}$
  Error $\sim 1/\sqrt{N_{MC}}$

One other thing: "p" has different meaning for Galerkin and collocation

- **Disadvantage of collocation:** For comparable accuracy
  # stochastic dof (collocation) $\approx 2^p$ (# stochastic dof (Galerkin))

Representative Comparison for Diffusion Equation

Representative comparative performance (E., Miller, Phipps, Tuminaro)



Using mean-based preconditioner for Galerkin system
Kruger, Pellisetti, Ghanem
Le Maître, et al., E. & Powell

Question: Can we reduce costs of collocation?

1 Preliminary: Spectral Methods for PDEs with Uncertain Coefficients

2 Reduced Basis Methods
- Offline Computations
- Reduced Problem
- Reduced Problem: Costs
- Reduced Problem: Capturing Features of Model

3 Reduced Basis + Sparse Grid Collocation

4 Iterative Solution of Reduced Problem

5 Concluding Remarks

Reduced Basis Methods

Starting point: Parameter-dependent PDE $\mathcal{L}_{\boldsymbol{\xi}} u = f$

In examples given: $\mathcal{L}_{\boldsymbol{\xi}} = -\nabla \cdot (a_0 + \sigma \sum_{r=1}^{m} \sqrt{\lambda_r} a_r(\mathbf{x})\xi_r)\nabla$

Discretize: Discrete system $\mathcal{L}_{h,\boldsymbol{\xi}}(u_h) = f$

Algebraic system $\mathcal{F}_{\boldsymbol{\xi}}(\mathbf{u}_h) = 0$ ($A_{\boldsymbol{\xi}}\mathbf{u}_h = \mathbf{f}$) of order N

Complication:

Expensive if many realizations (samples of $\boldsymbol{\xi}$) are required

Idea (Patera, Boyaval, Bris, Lelièvre, Maday, Nguyen, ...):

Solve the problem on a *reduced space*

That is: by some means, choose $\boldsymbol{\xi}^{(1)}, \boldsymbol{\xi}^{(2)}, \ldots, \boldsymbol{\xi}^{(n)}$, $n \ll N$

Solve $\mathcal{F}_{\boldsymbol{\xi}^{(i)}}(u_h^{(i)}) = 0$, $u_h^{(i)} = u_h(\cdot, \boldsymbol{\xi}^{(i)})$, $i = 1, \ldots, n$

For other $\boldsymbol{\xi}$, approximate $u_h(\cdot, \boldsymbol{\xi})$ by $\tilde{u}_h(\cdot, \boldsymbol{\xi}) \in span\{u_h^{(1)}, \ldots, u_h^{(n)}\}$

Terminology: $\{u_h^{(1)}, \ldots, u_h^{(n)}\}$ called **snapshots**

Offline Computations

Strategy for generating a basis / choosing snapshots (Patera, et al.):

For $\tilde{u}_h(\cdot,\boldsymbol{\xi}) \approx u_h(\cdot,\boldsymbol{\xi})$ (equivalently, $\tilde{\mathbf{u}}_{\boldsymbol{\xi}} \approx \mathbf{u}_{\boldsymbol{\xi}}$), use an **error indicator** $\eta(\tilde{u}_h) \approx \|e_h\|$, $e_h = u_h - \tilde{u}_h$

Given: a set of candidate parameters $\mathcal{X} = \{\boldsymbol{\xi}\}$,
an initial choice $\boldsymbol{\xi}^{(1)} \in \mathcal{X}$, and $u^{(1)} = u(\cdot,\boldsymbol{\xi}^{(1)})$

Set $Q = \mathbf{u}^{(1)}$

`while` $max_{\boldsymbol{\xi}\in\mathcal{X}} \left(\eta(\tilde{u}_h(\cdot,\boldsymbol{\xi}))\right) > \tau$

compute $\tilde{u}_h(\cdot,\boldsymbol{\xi})$, $\eta(\tilde{u}_h(\cdot,\boldsymbol{\xi}))$, $\forall\,\boldsymbol{\xi}\in\mathcal{X}$ % use current reduced

let $\boldsymbol{\xi}^* = argmax_{\boldsymbol{\xi}\in\mathcal{X}} \left(\eta(\tilde{u}_h(\cdot,\boldsymbol{\xi})\right)$ % basis

`if` $\eta(\tilde{u}_h(\cdot,\boldsymbol{\xi}^*)) > \tau$ `then`

augment basis with $u_h(\cdot,\boldsymbol{\xi}^*)$, update Q with $\mathbf{u}_{\boldsymbol{\xi}^*}$

`endif`

`end`

Potentially expensive, but viewed as "offline" *preprocessing*
"*Online*" simulation done using reduced basis

For set of candidate parameters $\mathcal{X} = \{\boldsymbol{\xi}\}$:

- Greedy search (Patera, et al.):
  Search over large set of parameters $\{\boldsymbol{\xi}\}$
  May be randomly or systematically chosen
- Optimization methods (Bui-thanh, Willcox, Ghattas):
  Find $\boldsymbol{\xi}$ that minimizes error estimator
  May need derivative information
- Not a concern in today's setting – we will use sparse grids

Reduced Problem

For linear problems, matrix form:

Coefficient matrix $A_{\boldsymbol{\xi}}$, nodal coefficients $\mathbf{u}_h$, $\tilde{\mathbf{u}}_h$, $\mathbf{u}^{(1)}, \ldots \mathbf{u}^{(n)}$

$Q =$ orthogonal matrix whose columns span space spanned by $\{\mathbf{u}^{(i)}\}$

Galerkin condition: make residual orthogonal to spanning space

$r = f - A_{\boldsymbol{\xi}}\tilde{\mathbf{u}}_{\boldsymbol{\xi}} = f - A_{\boldsymbol{\xi}} Q \mathbf{y}_{\boldsymbol{\xi}}$ orthogonal to Q

Result is **reduced problem**: Galerkin system of order $n \ll N$:

$$[Q^T A Q]\mathbf{y}_{\boldsymbol{\xi}} = Q^T f, \quad \tilde{\mathbf{u}}_{\boldsymbol{\xi}} = Q\mathbf{y}_{\boldsymbol{\xi}}$$

Goals: Reduced solution should

- be available at significantly lower cost
- capture features of the model

How are costs reduced?

- Matrix A of order N
- Reduced matrix Q^TAQ of order $n \ll N$
- Solving reduced problem is cheap for small n
- Note: making assumption that $\mathcal{L}_{\boldsymbol{\xi}}$ is affinely dependent on $\boldsymbol{\xi}$

$$\mathcal{L}_{\boldsymbol{\xi}} = \mathcal{L}_0 + \sum_{i=1}^k \phi_i(\boldsymbol{\xi})\mathcal{L}_i$$
$$\Rightarrow A_{\boldsymbol{\xi}} = A_0 + \sum_{i=1}^k \phi_i(\boldsymbol{\xi})A_i$$
$$\Rightarrow Q^TA_{\boldsymbol{\xi}}Q = \underbrace{Q^TA_0Q} + \sum_{i=1}^k \phi_i(\boldsymbol{\xi})\underbrace{[Q^TA_iQ]}$$

part of offline computation

True for example seen so far, $\phi_i(\boldsymbol{\xi}) = \xi_i$

- Means: constructing reduced matrix for new $\boldsymbol{\xi}$ is cheap
- Analogue for nonlinear problems is more complex

N.B. One other important issue:

Error indicator must be inexpensive to compute

In present study: use residual indicator

$$\eta_Q(\boldsymbol{\xi}) \equiv \frac{\|A_{\boldsymbol{\xi}}\tilde{\mathbf{u}}_{\boldsymbol{\xi}} - \mathbf{f}\|_2}{\|\mathbf{f}\|_2} = \frac{\|A_{\boldsymbol{\xi}} Q\mathbf{y}_{\boldsymbol{\xi}} - \mathbf{f}\|_2}{\|\mathbf{f}\|_2}$$

Using affine structure $A_{\boldsymbol{\xi}} = \sum_{i=1}^{k} \phi_i(\boldsymbol{\xi}) A_i$, efficiency derives from

$$\|A_{\boldsymbol{\xi}} Q\mathbf{y}_{\boldsymbol{\xi}} - \mathbf{f}\|_2^2 = \mathbf{y}_{\boldsymbol{\xi}}^T \left(\sum_{i=1}^{K} \sum_{j=1}^{K} \phi_i \phi_j \underbrace{Q^T A_i^T A_j Q}_{\text{Offline}} \right) \mathbf{y}_{\boldsymbol{\xi}}$$

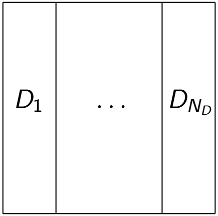
$$- 2\mathbf{y}_{\boldsymbol{\xi}}^T \sum_{i=1}^{K} \left(\phi_i \underbrace{Q^T A_i^T \mathbf{f}}_{\text{Offline}} \right) + \underbrace{\mathbf{f}^T \mathbf{f}}_{\text{Offline}}$$

Reduced Problem: Capturing Features of Model

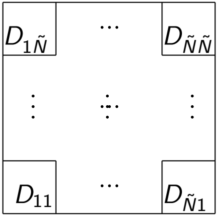
Consider benchmark problems:

Diffusion equation $-\nabla \cdot (a(\mathbf{x}, \boldsymbol{\xi})\nabla u) = f$ in $\mathbb{R}^2$

Piecewise constant diffusion coefficient parameterized as a random variable $\boldsymbol{\xi} = [\xi_1, \cdots, \xi_{N_D}]^T$ independently and uniformly distributed in $\Gamma = [0.01, 1]^{N_D}$



(a) Case 1: N_D subdomains



(b) Case 2: $N_D = \tilde{N} \times \tilde{N}$ subdomains

Does reduced basis capture features of model?

To assess this: consider

Full snapshot set, set of snapshots for all possible parameter values:

$$\mathcal{S}_\Gamma := \{u_h(\cdot, \boldsymbol{\xi}), \boldsymbol{\xi} \in \Gamma\}$$

Finite snapshot set, for finite $\Theta \subset \Gamma$:

$$\mathcal{S}_\Theta := \{u_h(\cdot, \boldsymbol{\xi}), \boldsymbol{\xi} \in \Theta\}$$

Question:

How many samples $\{\boldsymbol{\xi}\}$ / $\{u_h(\cdot, \boldsymbol{\xi})\}$ are needed to accurately represent the features of $\mathcal{S}_\Gamma$?

Experiment: to gain insight into this, estimate "rank" of $\mathcal{S}_\Gamma$

Generate a large set Θ of samples of $\boldsymbol{\xi}$

Generate the finite snapshot set S_Θ associated with Θ

Construct the matrix S_Θ of coefficient vectors $\mathbf{u}_{\boldsymbol{\xi}}$ from $\mathcal{S}_\Theta$

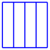
Compute the rank of S_Θ

Results follow. Used 3000 samples

Experiment was repeated ten times with similar results

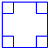
Estimated sizes of reduced basis for two benchmark problems

Case 1



Grid \ N_D	2	3	4	5	6	7	8	9	10
$33^2 = 1089$	3	12	18	30	40	53	55	76	84
$65^2 = 4225$	3	12	18	30	40	48	55	70	87
$129^2 = 16641$	3	12	18	28	39	48	55	72	81

Case 2



Grid \ N_D	4	9	16	25	36	49	64
$33^2 = 1089$	27	121	193	257	321	385	449
$65^2 = 4225$	28	148	290	465	621	769	897
$129^2 = 16641$	28	153	311	497	746	1016	1298

From: E & Liao, SIAM/ASA J. on Uncertainty Quantification 1, 2013

1 Preliminary: Spectral Methods for PDEs with Uncertain Coefficients

2 Reduced Basis Methods

3 Reduced Basis + Sparse Grid Collocation
- Introduction
- Performance for Diffusion Equation
- Application to the Navier-Stokes Equations

4 Iterative Solution of Reduced Problem

5 Concluding Remarks

Combine Reduced Basis with Sparse Grid Collocation

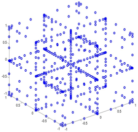
Recall collocation solution

$$u_{hp}(x, \boldsymbol{\xi}^{(k)}) = \sum_{\boldsymbol{\xi}^{(k)} \in \Theta_q} u_c(x, \boldsymbol{\xi}^{(k)}) L_{\boldsymbol{\xi}^{(k)}}(\boldsymbol{\xi}) \qquad (1)$$

Goal: Reduce cost of collocation via

1. Use sparse grid collocation points as candidate set $\mathcal{X}$
2. Use reduced solution as coefficient $u_c(\cdot, \boldsymbol{\xi}^{(k)})$ whenever possible

Algorithm

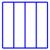


`for` each sparse grid level p
  `for` each point $\boldsymbol{\xi}^{(k)}$ at level p
    compute reduced solution $u_R(\cdot, \boldsymbol{\xi}^{(k)})$
    `if` $\eta(u_R(\cdot, \boldsymbol{\xi}^{(k)})) \leq \tau$, `then`
      use $u_R(\cdot, \boldsymbol{\xi}^{(k)})$ as coefficient $u_c(\cdot, \boldsymbol{\xi}^{(k)})$ in (1)
    `else`
      compute snapshot $u_h(\cdot, \boldsymbol{\xi}^{(k)})$, use it as $u_c(\cdot, \boldsymbol{\xi}^{(k)})$ in (1)
      augment reduced basis with $u_h(\cdot, \boldsymbol{\xi}^{(k)})$, update Q with $\mathbf{u}_{\boldsymbol{\xi}^{(k)}}$
    `endif`
  `end`
`end`

Number of Full System Solves, Diffusion Equation

Does this work? Look at diffusion problem
Various sparse grid levels p ($q = p + M$)

Case 1



Case 1, 5 × 1 subdomains, 65 × 65 grid, rank=30

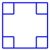
q	6	7	8	9	10	11	12	13	16
tol \ $\lvert\Theta_q\rvert$	11	61	241	801	2433	7K	19K	52K	870K
10^{-3}	10	9	0	0	0	0	0	0	0
10^{-4}	10	11	1	0	0	0	0	0	0
10^{-5}	10	13	0	0	0	0	0	0	0

Case 1, 9 × 1 subdomains, 65 × 65 grid, rank=70, $tol = 10^{-4}$

q	10	11	12	13	14	15	16	17
$\lvert\Theta_p\rvert$	19	181	1177	6001	26017	100897	361249	1218049
$N_{full\ solve}$	18	34	2	1	1	0	0	0

Number of Full System Solves, Diffusion Equation

Case 2



Case 2, 2×2 subdomains, 65×65 grid, rank=28

q	5	6	7	8	9	10	11	12	15
tol \ $\lvert\Theta_q\rvert$	9	41	137	401	1105	2.9K	7.5K	18.9K	272K
10^{-3}	7	11	3	0	0	0	0	0	0
10^{-4}	7	12	3	0	0	0	0	0	0
10^{-5}	7	13	2	3	0	0	0	0	0

Case 2, 4×4 subdomains, 65×65 grid, rank=290, $tol = 10^{-4}$

q	17	18	19	20	21
$\lvert\Theta_q\rvert$	33	545	6049	51137	353729
$N_{full\ solve}$	32	168	27	3	4

Refined Assessment of Accuracy

Examine error (vs. reference solution) in estimates of

Expected values:

Full collocation $\epsilon_h \equiv \left\| \tilde{\mathbb{E}}\left(u_q^{hsc}\right) - \tilde{\mathbb{E}}\left(u_r^{hsc}\right) \right\|_0 \Big/ \left\| \tilde{\mathbb{E}}\left(u_r^{hsc}\right) \right\|_0$

Reduced collocation $\epsilon_R \equiv \left\| \tilde{\mathbb{E}}\left(u_q^{rsc}\right) - \tilde{\mathbb{E}}\left(u_r^{hsc}\right) \right\|_0 \Big/ \left\| \tilde{\mathbb{E}}\left(u_r^{hsc}\right) \right\|_0$

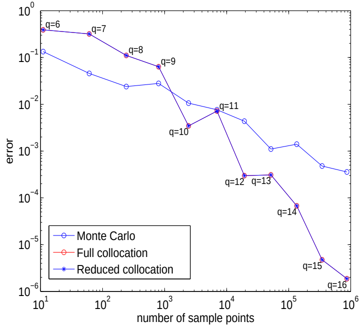
Variances:

Full collocation $\zeta_h \equiv \left\| \tilde{\mathbb{V}}\left(u_q^{hsc}\right) - \tilde{\mathbb{V}}\left(u_r^{hsc}\right) \right\|_0 \Big/ \left\| \tilde{\mathbb{V}}\left(u_r^{hsc}\right) \right\|_0$

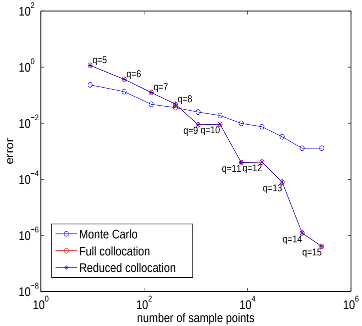
Reduced collocation $\zeta_R \equiv \left\| \tilde{\mathbb{V}}\left(u_q^{rsc}\right) - \tilde{\mathbb{V}}\left(u_r^{hsc}\right) \right\|_0 \Big/ \left\| \tilde{\mathbb{V}}\left(u_r^{hsc}\right) \right\|_0$

Errors in Expected Value

Case 1: 5×1 vertical subdomains



Case 2: 2×2 square subdomains



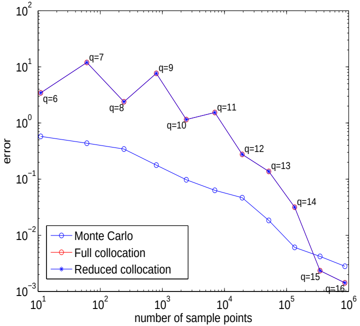
Comments:

Results for reduced/full systems are identical

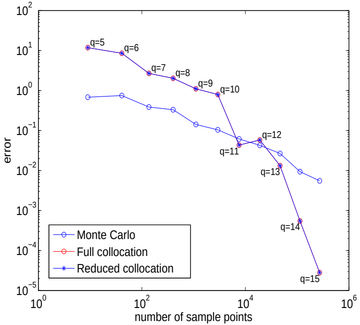
Results also compare favorably with Monte Carlo

Errors in Variance

Case 1: 5×1 vertical subdomains



Case 2: 2×2 square subdomains



Comments:

Trends for reduced/full systems are similar

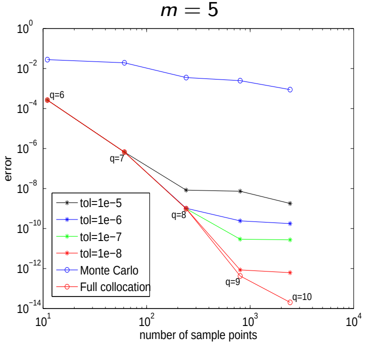
Noteworthy because error indicator is not effective as a fem error estimator

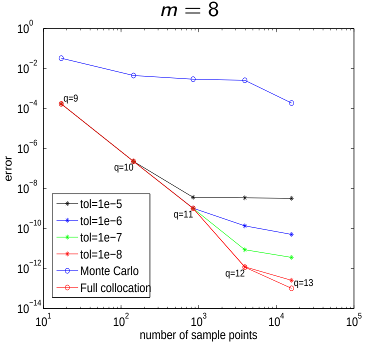
Diffusion problem with truncated Karhunen-Loève expansion

Diffusion coefficient $a_0 + \sigma \sum_{r=1}^{m} \sqrt{\lambda_r} a_r(\mathbf{x}) \xi_r$

From covariance function $c(\mathbf{x}, \mathbf{y}) = \sigma \exp\left(-\frac{|x_1 - y_1|}{c} - \frac{|x_2 - y_2|}{c}\right)$

Smaller correlation length $c \sim$ more terms m
Examine $c = 4,\ m = 4$ and $c = 2.5,\ m = 8$.





Comments on Costs

One difference from "pure" reduced basis method:

"Offline" and "Online" steps are not as clearly separated

Statement of costs of collocation:

Full: (# of collocation points) × (cost of full system solve)

Reduced: (# of collocation points where error tolerance is met)
× (cost of reduced system solve) +
(# of collocation points where error tolerance is not met)
× (cost of augmenting reduced basis and updating offline quantities).

For Reduced Collocation:

Red costs depend on N, large-scale parameter

Favors reduced if many collocation points use reduced model

Application to the Navier-Stokes Equations

$$
\begin{aligned}
-\nu(\cdot,\xi)\,\nabla^2\vec{u}(\cdot,\xi)+\vec{u}(\cdot,\xi)\cdot\nabla\vec{u}(\cdot,\xi)+\nabla p(\cdot,\xi) = 0 &\quad \text{in} \quad D\times\Gamma\\
\nabla\cdot\vec{u}(\cdot,\xi) = 0 &\quad \text{in} \quad D\times\Gamma\\
\vec{u}(\cdot,\xi) = \vec{g}(\cdot,\xi) &\quad \text{on} \quad \partial D\times\Gamma
\end{aligned}
$$

Possible sources of uncertainty:

- viscosity $\nu(x,\xi)$ (in multiphase flow)
- boundary conditions $g(x,\xi)$

Picard iteration (in weak form), for any realization of parameter ξ:

$$
\begin{aligned}
(\nu\nabla\delta\vec{u},\nabla\vec{v}) &+ (\vec{u}^{\ell}\cdot\nabla\delta\vec{u},\vec{v}) - (\delta p,\nabla\vec{v})\\
&= -(\nu\nabla\vec{u}^{\ell},\nabla\vec{v}) - (\vec{u}^{\ell}\cdot\nabla\vec{u}^{\ell},\vec{v}) + (p^{\ell},\nabla\vec{v}) \quad \forall\vec{v}\in X_0^h\\
(\nabla\cdot\delta\vec{u},q) &= -(\nabla\cdot\vec{u}^{\ell},q) \quad \forall q\in M^h
\end{aligned}
$$

$$
\vec{u}^{\ell+1} = \vec{u}^{\ell}+\delta\vec{u}, \qquad p^{\ell+1} = p^{\ell}+\delta p.
$$

Result: Matrix equation

$$\begin{pmatrix} A_{\boldsymbol{\xi}} + N_{\mathbf{u}^\ell,\boldsymbol{\xi}} & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \delta\mathbf{u} \\ \delta\mathbf{p} \end{pmatrix} = \begin{pmatrix} \mathbf{f}^r_{\mathbf{u}^\ell,\mathbf{p}^\ell,\boldsymbol{\xi}} \\ \mathbf{g}^r_{\mathbf{u}^\ell,\mathbf{p}^\ell,\boldsymbol{\xi}} \end{pmatrix}$$

Using div-stable Q_2-P_{-1} element

Reduced Problem: Given (matrix) representations Q_u, Q_p of velocity/pressure bases:

$$\begin{pmatrix} Q_u^T(A_{\boldsymbol{\xi}} + N_{\mathbf{u}^\ell,\boldsymbol{\xi}})Q_u & Q_u^T B^T Q_p \\ Q_p^T B Q_u & 0 \end{pmatrix} \begin{pmatrix} \delta\mathbf{w} \\ \delta\mathbf{y} \end{pmatrix} = \begin{pmatrix} Q_u^T \mathbf{f}^r_{\mathbf{u}^\ell,\mathbf{p}^\ell,\boldsymbol{\xi}} \\ Q_p^T \mathbf{g}^r_{\mathbf{u}^\ell,\mathbf{p}^\ell,\boldsymbol{\xi}} \end{pmatrix}$$

$$\delta\mathbf{u} \approx Q_u\,\delta\mathbf{w}, \quad \delta\mathbf{p} \approx Q_p\,\delta\mathbf{y}$$

Additional Requirements

Stability requirements As above, generate snapshots

$$\left\{ \begin{pmatrix} \vec{u}\left(\cdot,\xi^{(1)}\right) \\ p\left(\cdot,\xi^{(1)}\right) \end{pmatrix}, \ldots, \begin{pmatrix} \vec{u}\left(\cdot,\xi^{(n)}\right) \\ p\left(\cdot,\xi^{(n)}\right) \end{pmatrix} \right\}$$

Complication: reduced solution does not automatically satisfy inf-sup condition

Fix: (Quarteroni & Rozza): Supplement velocity basis with *supremizers* $\vec{r}\left(\cdot,\xi^{(k)}\right)$ that satisfy

$$\vec{r}\left(\cdot,\xi^{(k)}\right) = \arg\sup_{\vec{v}\in X_0^h} \frac{\left(p\left(\cdot,\xi^{(k)}\right), \nabla\cdot\vec{v}\right)}{|\vec{v}|_1}.$$

Result: Dim(reduced velocity space) = 2×dim(reduced pressure space)

Treatment of nonlinearities

- Recall: affine structure of linear operators $A_{\boldsymbol{\xi}} = \sum_{i=1}^{k} \phi_i(\boldsymbol{\xi}) A_i$

  $\rightarrow$ offline construction $Q^T A_{\boldsymbol{\xi}} Q = \sum_{i=1}^{k} \phi_i(\boldsymbol{\xi}) \, [Q^T A_i Q]$

- At step ℓ of reduced Picard iteration, reduced velocity iterate is $\mathbf{u}^\ell = Q_u \mathbf{w}^\ell$

  Convection operator has the form

$$\vec{u}^\ell \cdot \nabla = \sum_{i=1}^{n} w_i^\ell \, (\vec{q}^{(i)} \cdot \nabla)$$

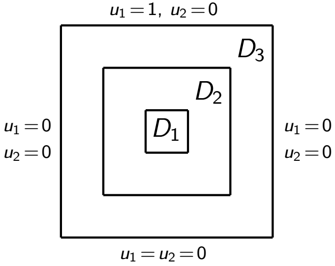
  Equivalently, convection matrix is $N = \sum_{i=1}^{n} N_i y_i$

$$\Rightarrow Q_u^T N Q_u = \sum_{i=1}^{n} \underbrace{[Q_u^T N_i Q_u]} \, w_i^\ell$$

Offline computation
cost $O(n^2 N) \times n$

Navier-Stokes with Uncertain Viscosity

$$
\begin{aligned}
-\nu(\cdot,\boldsymbol{\xi})\nabla^2\vec{u}(\cdot,\boldsymbol{\xi}) + \vec{u}(\cdot,\boldsymbol{\xi})\cdot\nabla\vec{u}(\cdot,\boldsymbol{\xi}) + \nabla p(\cdot,\boldsymbol{\xi}) = 0 \quad &\text{in} \quad D\times\Gamma \\
\nabla\cdot\vec{u}(\cdot,\boldsymbol{\xi}) = 0 \quad &\text{in} \quad D\times\Gamma \\
\vec{u}(\cdot,\boldsymbol{\xi}) = \vec{g}(\cdot,\boldsymbol{\xi}) \quad &\text{on} \quad \partial D\times\Gamma
\end{aligned}
$$



Driven cavity problem with

- variable random viscosity $\nu = [\nu_1, \nu_2, \nu_3]^T$
- piecewise constant on subdomains
- independently and uniformly distributed
    - in $[0.01, 1]^3$

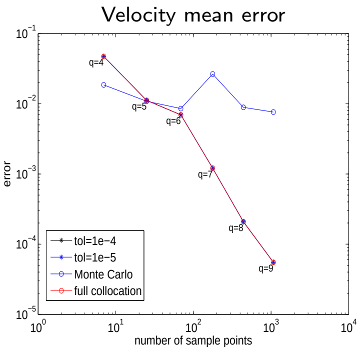
Number of full system solves

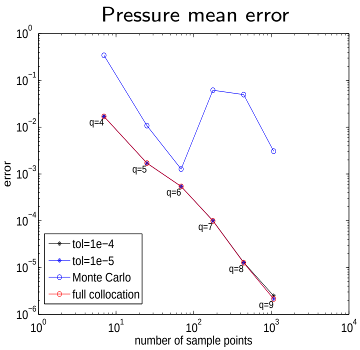
q		3	4	5	6	7	8	9	
tol	Grids \ $\lvert\Theta_q\rvert$	1	7	25	69	177	441	1073	Total
10^{-4}	33×33	1	6	17	23	26	26	25	124
10^{-4}	65×65	1	6	16	20	21	21	18	103
10^{-5}	33×33	1	6	18	29	40	44	41	179
10^{-5}	65×65	1	6	18	27	32	40	32	156

Inf-sup constants γ_R^2 for reduced problem ($\gamma_h^2 = .2137$)

N_u	2	4	20	50	100	200
γ_R^2	0.2431	0.2430	0.2374	0.2359	0.2327	0.2292

Assessment of errors





1. Preliminary: Spectral Methods for PDEs with Uncertain Coefficients

2. Reduced Basis Methods

3. Reduced Basis + Sparse Grid Collocation

4. Iterative Solution of Reduced Problem
   - Introduction
   - Implementation
   - Performance
   - Trends and Analysis

5. Concluding Remarks

Reduced Problem: Look Closely at Costs

Key requirement: Solution of reduced problem is cheap

- **Reduced problem and solution:**
  $[Q^T A_\xi Q]\mathbf{y}_\xi = Q^T\mathbf{f}, \quad \tilde{\mathbf{u}}_\xi = Q\mathbf{y}_\xi$
  Dense system of order $k \ll N$
  Cost of solution: $O(k^3)$
- **Full problem:**
  $A_\xi \mathbf{u}_\xi = \mathbf{f}$
  Sparse discrete PDE of order N
  Cost of solution by multigrid: $O(N)$
- **Conventional wisdom:**
  k is small, N is large,
  In this scenario: reduced model is the clear winner
- **Not so clear when:**
  $k \ll N$ but $k^3 \not\ll N$

Iterative Solution of Reduced Problem

- Reduced problem: $[Q^T A_\xi Q]\mathbf{y}_\xi = Q^T \mathbf{f}$

  Solve by iterative method (e.g., conjugate gradient)

  Seek **preconditioner** $P \approx Q^T A_\xi Q$

- Reformulate reduced problem as a saddle-point problem:

$$\begin{bmatrix} A_\xi^{-1} & Q \\ Q^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \mathbf{y}_\xi \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ Q^T \mathbf{f} \end{bmatrix}$$

  Reduced matrix = **Schur complement operator** S

- Approximate Schur complement:

$$\hat{P}_S := (Q^T Q)(Q^T A_\xi^{-1} Q)^{-1}(Q^T Q) = (Q^T A_\xi^{-1} Q)^{-1}$$

- Approximate A_ξ^{-1} using multigrid: $P_{A_\xi}^{-1} \longrightarrow P_S = (Q^T P_{A_\xi}^{-1} Q)^{-1}$

- For preconditioning: require action of $\boldsymbol{P_S^{-1} = Q^T P_{A_\xi}^{-1} Q}$

Implementation

For parameter $\boldsymbol{\xi}$:

- Construct reduced matrix of order $k \ll N$

$$Q^T A_{\boldsymbol{\xi}} Q = Q^T A_0 Q + \sum_{i=1}^{m} \phi_i(\boldsymbol{\xi})[Q^T A_i Q]$$

- Explicitly construct preconditioning operator $P_S^{-1} = Q^T P_{A_{\boldsymbol{\xi}}}^{-1} Q$
  N.B. not practical, "**online**," costs $O(N)$
- Alternative: use a single $\boldsymbol{\xi}_0$, $P_{A_{\boldsymbol{\xi}_0}}$ for all $A_{\boldsymbol{\xi}}$
  Done once: Apply MG to each column of $Q \longrightarrow P_{A_{\boldsymbol{\xi}_0}}^{-1} Q$
  Premultiply result by Q^T
  Produces (dense) preconditioning operator of order k
- Variant: use a finite fixed set $\{\boldsymbol{\xi}_j\}$ to construct $\{P_{S,j}^{-1}\}$
  For $A_{\boldsymbol{\xi}}$, use $P_{S,j}$ for $\boldsymbol{\xi}_j$ closest to $\boldsymbol{\xi}$
- Cost per step of matrix operations $O(k^2)$, $k \ll N$

Experimental Performance

For all experiments:

- PDE posed on a square domain
- Spatial discretization: Bilinear fem
- Error indicator: Matrix residual norm

$$\frac{\|\mathbf{f} - A_{\boldsymbol{\xi}}\tilde{\mathbf{u}}\|_2}{\|\mathbf{f}\|_2} \leq \tau, \quad \tau = 10^{-8}$$

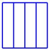
- Iteration stopping test:

$$\frac{\|Q^T\mathbf{f} - Q^T A_{\boldsymbol{\xi}} Q\mathbf{y}_i\|_2}{\|Q^T\mathbf{f}\|_2} \leq \frac{\tau}{10},$$

- MG preconditioner: PyAMG (Bell, Olson, Schroder)
- Test: Solve 100 randomly generated systems

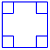
Estimated sizes of reduced basis for two benchmark problems

Case 1



Grid \ N_D	2	3	4	5	6	7	8	9	10
$33^2 = 1089$	3	12	18	30	40	53	55	76	84
$65^2 = 4225$	3	12	18	30	40	48	55	70	87
$129^2 = 16641$	3	12	18	28	39	48	55	72	81

Case 2



Grid \ N_D	4	9	16	25	36	49	64
$33^2 = 1089$	27	121	193	257	321	385	449
$65^2 = 4225$	28	148	290	465	621	769	897
$129^2 = 16641$	28	153	311	497	746	1016	1298

- Benchmark problem 1:

Diffusion equation $-\nabla \cdot (a(\mathbf{x}, \boldsymbol{\xi})\nabla u) = f$ on $[0,1] \times [0,1]$

$a(x, \boldsymbol{\xi}) = \mu(x) + \sum_{i=1}^{m} \sqrt{\lambda_i}\, a_i(x)\xi_i$

a derived from covariance function

$$C(x,y) = \sigma^2 exp\left(-\frac{|x_1 - y_1|}{c} - \frac{|x_2 - y_2|}{c}\right)$$

$\{\xi_r\}$ uniform on $[-1,1]$, $\sigma = .5$, $\mu \equiv 1$

Benchmark problem 1
(P)CG terations
$m = \#$ parameters
$k =$ size of reduced basis

N		3	1.5	0.75	0.375
	c	3	1.5	0.75	0.375
	m	7	17	65	325
33^2	k	97	254	607	982
	None	60.1	90.7	101.7	103.9
	Single	10.0	9.3	9.5	8.9
	Online	10.0	9.0	9.0	8.0
65^2	k	100	269	699	1679
	None	68.8	129.3	175.5	200.3
	Single	10.0	10.0	8.5	8.7
	Online	10.0	9.8	8.0	8.0
129^2	k	102	269	729	1808
	None	70.1	149.5	252.5	339.1
	Single	11.2	14.6	12.9	11.0
	Online	11.0	14.8	13.0	11.0
257^2	k	102	275	740	1846
	None	70.4	154.0	293.6	473.7
	Single	11.0	13.7	15.4	13.5
	Online	11.0	13.0	15.0	13.0

Benchmark Problem 1: CPU Times

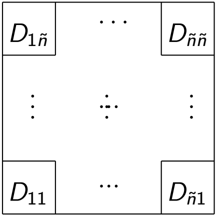
N	c m (# parameters)		3 7	1.5 17	0.75 65	0.375 325
33^2	k (basis size)		97	254	607	982
	Full	AMG	0.0202	0.0205	0.0214	0.0228
	Reduced	Direct	**0.0003**	0.0016	0.0181	0.0699
	Reduced	Iterative	0.0004	**0.0008**	**0.0036**	**0.0103**
65^2	k		100	269	699	1679
	Full	AMG	0.1768	0.1961	0.1947	0.1974
	Reduced	Direct	**0.0003**	0.0021	0.0262	0.3207
	Reduced	Iterative	0.0004	**0.0010**	**0.0044**	**0.0252**
129^2	k		102	269	729	1808
	Full	AMG	0.1195	0.1286	0.1347	0.1443
	Reduced	Direct	**0.0003**	0.0020	0.0287	0.4452
	Reduced	Iterative	0.0005	**0.0013**	**0.0070**	**0.0449**
257^2	k		102	275	740	1846
	Full	AMG	0.3163	0.2988	0.3030	0.3778
	Reduced	Direct	**0.0004**	0.0024	0.0302	0.4498
	Reduced	Iterative	0.0005	**0.0012**	**0.0088**	**0.0619**

- **Benchmark problem 2:**

Diffusion equation $-\nabla \cdot (a(\mathbf{x}, \boldsymbol{\xi}) \nabla u) = f$ on $[-1, 1] \times [-1, 1]$

$a(x, \boldsymbol{\xi}) = \sum_{r=1}^{m} a_r(x)\xi_r \,, \;\; m = \tilde{n}^2$

$\{\xi_r\}$ uniform on $[0.1, 1]$



Benchmark Problem 2: CG Iterations

	m	4	16	36	64	100
33^2	k	27	193	321	449	577
	None	31.9	113.9	126.4	127.9	128
	Single	17.2	32.3	44.0	52.0	59.5
	Online	11.4	13.0	13.6	12.7	12.0
65^2	k	29	309	625	897	1153
	None	42.3	234.1	254.9	258.3	256.4
	Single	20.1	38.2	47.0	54.8	64.2
	Online	14.3	17.0	18.2	18.9	18.9
129^2	k	33	359	862	1519	2219
	None	60.3	432.9	493.6	519.2	518.9
	Single	24.2	37.5	47.6	58.1	64.9
	Online	19.1	19.0	22.0	24.1	25.2
257^2	k	36	394	979	1789	2801
	None	82.0	808.9	976.8	1035.6	1037.3
	Single	30.4	44.0	50.9	62.2	71.1
	Online	25.1	25.8	25.7	27.8	29.7

Benchmark Problem 2: CPU Times

N	m		4	16	36	64	100
33^2	k		27	193	321	449	577
	Full	AMG	0.0218	0.0203	0.0215	0.0210	0.0208
	Reduced	Direct	**0.0001**	**0.0010**	**0.0032**	**0.0073**	**0.0152**
	Reduced	Iterative	0.0006	0.0019	0.0045	0.0090	0.0181
65^2	k		29	309	625	897	1153
	Full	AMG	0.1679	0.1601	0.1669	0.1811	0.1760
	Reduced	Direct	**0.0002**	**0.0026**	0.0187	0.0543	0.1088
	Reduced	Iterative	0.0007	0.0034	**0.0176**	**0.0458**	**0.0832**
129^2	k		33	359	862	1519	2219
	Full	AMG	0.1134	0.1202	0.1357	**0.1184**	**0.1194**
	Reduced	Direct	**0.0002**	**0.0038**	0.0461	0.2319	0.6659
	Reduced	Iterative	0.0009	0.0041	**0.0364**	0.1340	0.3060
257^2	k		36	394	979	1789	2801
	Full	AMG	0.3376	0.3519	0.3291	0.3365	**0.3568**
	Reduced	Direct	**0.0002**	**0.0051**	0.0670	0.3555	1.2972
	Reduced	Iterative	0.0010	0.0060	**0.0485**	**0.1928**	0.5365

Trends and Analysis

- Iteration counts (for solving $Q^T A_\xi Q\mathbf{y} = Q^T\mathbf{f}$) are
  - essentially independent of spatial mesh size
  - essentially independent of number of parameters / size of reduced basis
  - hence, iterative solution costs ck^2
  - smaller than direct costs ($O(k^3)$) for all large k
- In comparison to *full* $O(N)$ AMG solve:
  - for fixed number of parameters, reduced iterative solve cheaper for large N
  - for fixed N, full solve cheaper as number of parameters increases

For analysis: interested in bounds on Rayleigh quotient

$$\frac{(\mathbf{q}, Q^T A_{\boldsymbol{\xi}} Q\mathbf{q})}{(\mathbf{q}, (Q^T P_{A_{\boldsymbol{\xi}}}^{-1} Q)^{-1}\mathbf{q})} = \underbrace{\frac{(\mathbf{q}, Q^T A_{\boldsymbol{\xi}} Q\mathbf{q})}{(\mathbf{q}, (Q^T A_{\boldsymbol{\xi}}^{-1} Q)^{-1}\mathbf{q})}}_{(1)} \underbrace{\frac{(\mathbf{q}, (Q^T A_{\boldsymbol{\xi}}^{-1} Q)^{-1}\mathbf{q})}{(\mathbf{q}, (Q^T P_{A_{\boldsymbol{\xi}}}^{-1} Q)^{-1}\mathbf{q})}}_{(2)}$$

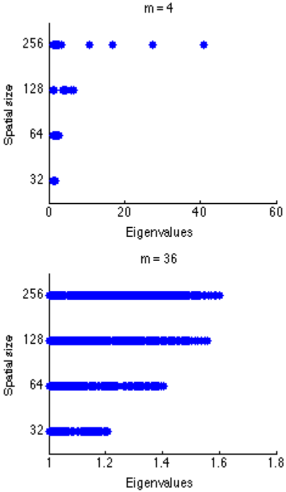
to establish bounds on iteration counts

(2): Bounded using spectral equivalence of $A_{\boldsymbol{\xi}}$ and MG approximation

$$\beta_1 \le \frac{(\mathbf{v}, A_{\boldsymbol{\xi}}\mathbf{v})}{(\mathbf{v}, P_{A_{\boldsymbol{\xi}}}\mathbf{v})} \le \beta_2$$

(1): Work in progress

Eigenvalue distributions of $(Q^T A_\xi^{-1} Q)(Q^T A_\xi Q)$



- **Benchmark problem 3:**

  Convection-diffusion-reaction equation:

  $$-\nu\nabla^2 u + (\vec{w}\cdot\nabla)u + r(\mathbf{x},\boldsymbol{\xi})u = f$$

  $$r(x,\boldsymbol{\xi}) = \mu(x) + \sum_{i=1}^{m}\sqrt{\lambda_i}\, r_i(x)\xi_i$$

  r derived from covariance function

  $$C(x,y) = \sigma^2 exp\left(-\frac{|x_1-y_1|}{c} - \frac{|x_2-y_2|}{c}\right)$$

  $\{\xi_r\}$ uniform on $[-1,1]$, $\sigma = .5$, $\mu \equiv 1$

Benchmark Problem 3: BiCGStab Iterations

N				
	c	2	1	0.5
	m	36	145	785
33^2	k	210	421	798
	None	49.5	45.9	41.7
	Single	8.2	7.0	6.1
	Online	8.3	7.0	6.0
65^2	k	178	372	952
	None	84.5	87.4	86.5
	Single	12.0	10.0	9.0
	Online	12.0	10.0	9.0
129^2	k	138	265	749
	None	122.8	153.0	176.2
	Single	12.9	13.1	13.0
	Online	12.7	13.5	13.0
257^2	k	99	197	686
	None	126.8	234.0	293.8
	Single	14.2	14.4	15.1
	Online	13.9	14.5	15.0

Benchmark Problem 3: CPU Time

N		c / m	2 / 36	1 / 145	0.5 / 785
33^2		k	210	421	798
	Full	AMG	0.0419	0.0428	0.0440
	Reduced	Direct	0.0011	0.0067	0.0400
	Reduced	Iterative	**0.0009**	**0.0019**	**0.0066**
65^2		k	178	372	952
	Full	AMG	0.2188	0.2258	0.2311
	Reduced	Direct	**0.0009**	0.0046	0.0679
	Reduced	Iterative	0.0013	**0.0022**	**0.0148**
129^2		k	138	265	749
	Full	AMG	0.3228	0.3284	0.3271
	Reduced	Direct	**0.0007**	**0.0020**	0.0323
	Reduced	Iterative	0.0012	**0.0020**	**0.0132**
257^2		k	99	197	686
	Full	AMG	1.5330	1.5468	1.5396
	Reduced	Direct	**0.0003**	**0.0010**	0.0234
	Reduced	Iterative	0.0010	0.0016	**0.0140**

Concluding Remarks

- Reduced basis methods offer significant promise for reducing the cost of collocation methods for uncertainty quantification
- Addresses issue of cost associated with collocation
- Amenable to mildly nonlinear problems
- General nonlinear problems: active area of research