Linear Algebra Methods for Parameter-Dependent Partial Differential Equations

Howard C. Elman
Department of Computer Science
Institute for Advanced Computer Studies
University of Maryland at College Park

Collaborators:

Darran Furnival	UMD / Université de Bretagne
Catherine Powell	University of Manchester
Tengfei Su	University of Maryland
Virginia Forstall	UMD / Leidos
Qifeng Liao	Shanghai Tech University
Kookjin Lee	University of Maryland
Oliver Ernst	TU Chemnitz
Dianne O'Leary	University of Maryland
Michael Stewart	Georgia State University
Chris Miller	UMD / Mitre Corporation
Eric Phipps	Sandia National Laboratories
Ray Tuminaro	Sandia National Laboratories
Elisabeth Ullmann	TU Munich
David Silvester	University of Manchester

Work supported by: NSF, DOE

1. Problem Statement
2. Background: Some early results on iterative solution
   - Multigrid in spatial component
   - Multigrid for mean-based prconditioner
3. Low-rank / Reduced-order models
   - Introduction
   - Low-rank Krylov subspace methods
   - Low-rank multigrid methods
   - Reduced-order models
4. Nonlinear problems
   - Empirical Interpolation Methods
   - Application to Navier-Stokes Equations
   - Construction of bases
   - Experimental results
   - Preconditioning

Parameter-Dependent Partial Differential Equations

Parameter-dependent partial differential equations

$\mathcal{G}_{\boldsymbol{\xi}}(u) = 0$ on domain $\mathcal{D} \subset \mathbb{R}^2$ or $\mathbb{R}^3$

$\boldsymbol{\xi} = [\xi_1, \xi_2, \ldots, \xi_m]^T$ m-dimensional vector of parameters

Want solution $u = u(\cdot, \boldsymbol{\xi})$ for many values of $\boldsymbol{\xi}$

Examples:

- Diffusion equation: $-\nabla \cdot (a(\mathbf{x}, \boldsymbol{\xi}) \nabla u) = f$
- Convection-diffusion equation: $-\nabla \cdot (a(\mathbf{x}, \boldsymbol{\xi}) \nabla u) + \vec{w} \cdot \nabla u = f$
- Navier-Stokes equations: $-\nabla \cdot (a(\mathbf{x}, \boldsymbol{\xi}) \nabla \vec{u}) + (\vec{u} \cdot \nabla)\vec{u} + \nabla p = \vec{f}$
  $\nabla \cdot \vec{u} = 0$
- Helmholtz equation: $-\Delta u - k(\mathbf{x}, \boldsymbol{\xi})^2 u = f$

Posed on $\mathcal{D} \subset \mathbb{R}^d$ with suitable boundary conditions
Boundary conditions, forcing function could also depend on parameters

Discrete parameter-dependent partial differential equations

- For computation, seeking discrete solution $G_\xi(\mathbf{u}) = 0$, $\mathbf{u} = \mathbf{u}(\boldsymbol{\xi})$
  Simulation: requires solutions for many $\boldsymbol{\xi}$, too expensive
  Fix: construct surrogate discrete solution $\mathbf{u}^s(\boldsymbol{\xi})$, cheaper to evaluate
  Purpose: reduce cost of function evaluation
- Our point of view: $\mathbf{u}^s(\boldsymbol{\xi}) = \sum_{\ell=1}^{n_\xi} \mathbf{u}_\ell \psi_\ell(\boldsymbol{\xi})$
  $\{\mathbf{u}_\ell\}$ could be (nodal) finite element coefficients
  or from finite difference or finite volume discretization
  $\{\psi_\ell\}$ are basis functions in a parameter-space discretization
  such as *polynomial chaos* for stochastic Galerkin method,
  Lagrange interpolating functions for stochastic collocation
- Once we have $\{\mathbf{u}_\ell\}$, or equivalently $U = \left[\mathbf{u}_1, \cdots, \mathbf{u}_{n_\xi}\right]$, simulation is cheap
- In addition, achieve further savings by constructing low-rank approximations $U \approx VW^T$

Consider parameter-dependent coefficient with affine structure

$$a(\mathbf{x}, \boldsymbol{\xi}) \equiv a_0(x) + \sum_{\ell=1}^{m} a_\ell(\mathbf{x})\xi_\ell$$

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

For linear PDEs with stochastic Galerkin discretization

spatial discretization size n_x, parameter space discretization size n_ξ

$\longrightarrow$ linear system with Kronecker product structure

$$\mathcal{A}\mathbf{u} = \mathbf{f}, \quad \mathcal{A} = I \otimes A_0 + \sum_{\ell=1}^{m} G_\ell \otimes A_\ell, \quad \mathbf{f} = \mathbf{1} \otimes \mathbf{f}_0$$

$\mathbf{u}$ = "stacked" $[\mathbf{u}_1^T, \ldots, \mathbf{u}_{n_\xi}^T]^T$, each $\mathbf{u}_\ell$ of dimension n_x

Equivalent "matricized" form:

$$A_0 U(I) + \sum_{\ell=1}^{m} A_\ell U G_\ell^T = F, \quad U = [\mathbf{u}_1, \cdots, \mathbf{u}_{n_\xi}], \; F = \mathbf{f}_0 \mathbf{1}^T$$

Not restricted to stochastic Galerkin

Discrete parameterized system:

$$\left(A_0 + \sum_{\ell=0}^{m} A_\ell \xi_\ell \right) \mathbf{u}(\boldsymbol{\xi}) = \mathbf{f}_0$$

Monte Carlo or stochastic collocation with n_ξ samples $\longrightarrow$

$$\mathcal{A}\mathbf{u} = \mathbf{f}, \quad \mathcal{A} = I \otimes A_0 + \sum_{\ell=1}^{m} G_\ell \otimes A_\ell, \quad \mathbf{f} = \mathbf{1} \otimes \mathbf{f}_0$$

$$G_\ell \equiv diag\left(\xi_\ell^{(1)}, \ldots, \xi_\ell^{(n_{\boldsymbol{\xi}})} \right)$$

Diagonal with sample values of ℓth parameter

Result: matricized system of same form

$$A_0 U + \sum_{\ell=1}^{m} A_\ell U G_\ell^T = F$$

Potential for low-rank structure is the same

1 Problem Statement

2 Background: Some early results on iterative solution
- Multigrid in spatial component
- Multigrid for mean-based prconditioner

3 Low-rank / Reduced-order models

4 Nonlinear problems

Background: Multigrid for Kronecker systems

I. Apply multigrid across spatial component (E. & Furnival, 2007)

Fine grid operators: $\mathcal{A}^{(h)}$, $A_\ell^{(h)}$ spatial discretization parameter h
Coarse grid operators: $\mathcal{A}^{(2h)}$, $A_\ell^{(2h)}$ spatial discretization parameter $2h$

One multigrid (two-grid) step:

for $j = 1$ to ν
$\quad \mathbf{u}^{(h)} \leftarrow \mathbf{u}^{(h)} + Q^{-1}(\mathbf{f}^{(h)} - \mathcal{A}^{(h)}\mathbf{u}^{(h)})$ $\quad$ ν smoothing steps
end
$\mathbf{r}^{(2h)} = \mathcal{R}(\mathbf{f}^{(h)} - \mathcal{A}^{(h)}\mathbf{u}^{(h)})$ $\quad$ Restriction
Solve $\mathcal{A}^{(2h)}\mathbf{c}^{(2h)} = \mathbf{r}^{(2h)}$ $\quad$ Coarse grid correction $\mathcal{R} = I \otimes R$
$\mathbf{u}^{(h)} \leftarrow \mathbf{u}^{(h)} + \mathcal{P}\mathbf{c}^{(2h)}$ $\quad$ Prolongation $\mathcal{P} = I \otimes P$

Sketch of convergence analysis: use "classic" approach

$$\mathbf{e}^{(i+1)} = [(\mathcal{A}^{(h)})^{-1} - \mathcal{P}(\mathcal{A}^{(2h)})^{-1}\mathcal{R}]\,[\mathcal{A}^{(h)}(I - Q^{-1}\mathcal{A}^{(h)})^{\nu}]\,\mathbf{e}^{(i)}$$

Approximation property $\left\|[(\mathcal{A}^{(h)})^{-1} - \mathcal{P}(\mathcal{A}^{(2h)})^{-1}\mathcal{R}]\mathbf{y}\right\|_{\mathcal{A}^{(h)}} \leq \|\mathbf{y}\|_2$

Smoothing property $\left\|\mathcal{A}^{(h)}(I - Q^{-1}\mathcal{A}^{(h)})^k\mathbf{y}\right\|_2 \leq \|\mathbf{y}\|_{\mathcal{A}^{(h)}}$

For approximation property: Introduce *semi-discrete space* $H_0^1(\mathcal{D}) \otimes \mathcal{T}^{(p)}$

$\mathcal{T}^{(p)}$ = discrete stochastic space

Weak formulation: $a(u^{(p)}, v^{(p)}) = (f, v^{(p)})$ for all $v^{(p)} \in H_0^1(\mathcal{D}) \otimes \mathcal{T}^{(p)}$

Theorem

$$\begin{aligned}\left\|[(\mathcal{A}^{(h)})^{-1} - \mathcal{P}(\mathcal{A}^{(2h)})^{-1}\mathcal{R}]\mathbf{y}\right\|_{\mathcal{A}^{(h)}} &= \|u^{(hp)} - u^{(2h,p)}\|_a \\ &\leq \|u^{(hp)} - u^{(p)}\|_a + \|u^{(p)} - u^{(2h,p)}\|_a \\ &\leq c\|\mathbf{y}\|_{\mathcal{A}^{(h)}}\end{aligned}$$

Last step: from standard arguments based on approximability, regularity for *every realization* in the semi-discrete space

II. Use multigrid for mean-based preconditioner

Solving $\mathcal{A}\mathbf{u} = \mathbf{f}$

Preconditioner for use with CG (Kruger, Pellisetti, Ghanem):

Mean $Q = G_0 \otimes A_0$

$A_0 \sim \int_{\mathcal{D}} \bar{a}(x,\cdot)\nabla\phi_k(x)\cdot\nabla\phi_j(x)dx, \quad G_0 = I$

Further refinement (Le Maître, et al.)

Use multigrid to approximate action of Q^{-1}: $Q_{MG}^{-1} \equiv I \otimes A_{0,MG}^{-1}$

Convergence analysis (Powell & E., 2008):

Coefficient:	$a(\mathbf{x},\boldsymbol{\xi}) = a_0 + \sum_{\ell=1}^{m} a_\ell(\mathbf{x})\boldsymbol{\xi}_\ell$
Coefficient matrix:	$\mathcal{A} = G_0 \otimes A_0 + \sum_{\ell=1}^{m} G_\ell \otimes A_\ell$
Mean-based preconditioner:	$Q = G_0 \otimes A_0$
Multigrid preconditioner:	$Q_{MG} = G_0 \otimes A_{0,MG}$

Theorem

For $a_0 = \mu$ *constant,*

$$1 - \tau \leq \frac{(w,Aw)}{(w,Qw)} \leq 1 + \tau$$

where $$\tau = (1/\mu)\, c(p) \sum_{\ell=1}^{m} \|a_\ell\|_\infty .$$

If in addition the MG approximation satisfies $\beta_1 \leq \frac{(w,Qw)}{(w,Q_{MG}w)} \leq \beta_2$, *then*

$$\frac{(w,Aw)}{(w,Q_{MG}w)} = \frac{(w,Aw)}{(w,Qw)} \, \frac{(w,Qw)}{(w,Q_{MG}w)} \leq \left(\frac{1+\tau}{1-\tau}\right) \left(\frac{\beta_2}{\beta_1}\right)$$

Upshot: Two types of multigrid methods with textbook convergence behavior

1 Problem Statement

2 Background: Some early results on iterative solution

3 Low-rank / Reduced-order models
- Introduction
- Low-rank Krylov subspace methods
- Low-rank multigrid methods
- Reduced-order models

4 Nonlinear problems

Low-rank / Reduced-order models

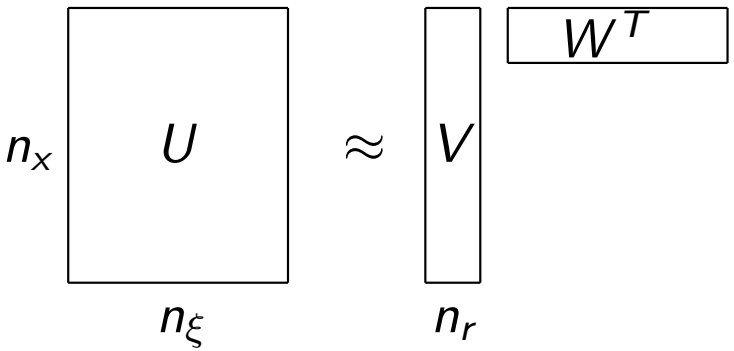
Return to matricized system $A_0 U + \sum_{\ell=1}^{m} A_\ell U G_\ell^T = F$

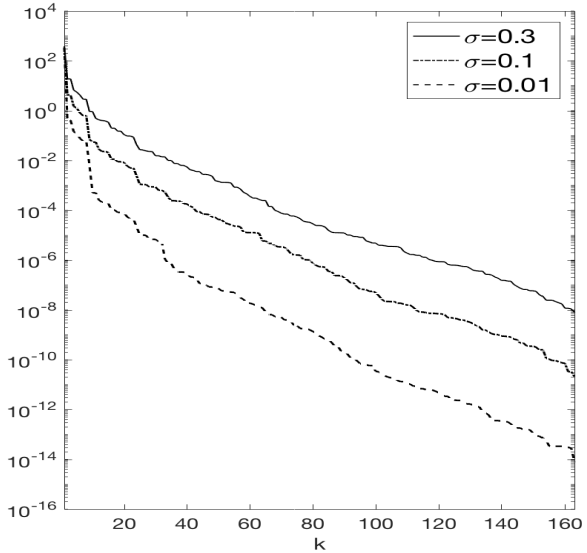
Solution matrix $U \in \mathbb{R}^{n_x \times n_\xi}$

Observation (Kressner & Tobler, Ballani & Grasedyck, Benner, Onwunta, & Stoll):

U is well approximated by low rank-matrix, $U \approx VW^T$

$rank(V) = rank(W) = n_r \ll min(n_x, n_\xi)$





Singular values of U, $n_x = 16,129$, $n_\xi = 165$

Idea: Iterative methods that take advantage of low-rank structure

Background: Krylov subspace methods with truncation
(Kressner & Tobler, Ballani & Grasedyck, Benner, Onwunta & Stoll)

- Apply CG / GMRES to Kronecker / matricized equations
- Construct iterates $\{U_j\}$ to be in low-rank format, $U_j = V_j W_j^T$
- Key points:

  (+) Individual products for matrix-vector products are cheaper

$$A_0 U_j + \sum_{\ell=1}^{m} \underbrace{A_\ell U_j G_\ell^T}_{O(n_x \times n_\xi)} = (A_0 V_j) W_j^T + \sum_{\ell=1}^{m} \underbrace{(A_\ell V_j)(G_\ell W_j)^T}_{O((n_x + n_\xi) \times n_r)}$$

  (–) Sums tend to increase the ranks

  (–) Similar difficulty for vector updates $U \leftarrow U + \alpha P$

Remedy, rank-compression:

Force intermediate quantities to be of low rank

Example: MVP $\mathcal{A}\mathbf{u} \sim \sum_{\ell=0}^{m} A_\ell U G_\ell^T$, $U = VW^T$, rank n_r

$$\widetilde{X} \equiv \underbrace{[A_0 V, A_1 V, \ldots, A_m V]}_{\widetilde{V}} \underbrace{[G_0 W, G_1 W, \ldots, G_m W]^T}_{\widetilde{W}^T} \quad (G_0 = I)$$

$\widetilde{V}$ ranks $\leq (m+1)n_r$ $\widetilde{W}^T$

Construct low-rank approximation to the product (Kressner & Tobler):

Compute QR factorizations $\widetilde{V} = Q_{\widetilde{V}} R_{\widetilde{V}}$, $\widetilde{W} = Q_{\widetilde{W}} R_{\widetilde{W}}$

Then $$\widetilde{X} = Q_{\widetilde{V}} (R_{\widetilde{V}} R_{\widetilde{W}}^T) Q_{\widetilde{W}}^T = Q_{\widetilde{V}} \widehat{V} \widehat{\Sigma} \widehat{W}^T Q_{\widetilde{W}}^T$$

truncate $$\approx \underbrace{\left[Q_{\widetilde{V}} \widehat{V}(:, 1:n_r)\right]}_{V} \underbrace{\widehat{\Sigma}(1:n_r) \left[\widehat{W}(:, 1:n_r)^T Q_{\widetilde{W}}^T\right]}_{W^T}$$

$= (\text{new}) \quad V$

N.B. ranks may change as iteration proceeds

Features and key points (Kressner + others)

- Computing SVD of small matrix $R_{\widetilde{V}} R_{\widetilde{W}}^T = \widehat{V}\widehat{\Sigma}\widehat{W}^T$
- Truncation is based on singular values in $\widehat{\Sigma}$
  Various strategies. Loosely: retain singular values $\geq$ tolerance τ
- No longer have true CG or GMRES (lose orthogonality)
  (+) Empirically: if done carefully, behaves like standard methods
  Reduces error to be like size of truncation tolerance
- **(–)** Difficult to analyze convergence properties
- **(–)** Shifts costs to truncation operation
  QR factorization is cheaper than SVD of $\widetilde{X}$ but is still costly

---

Next: explore multigrid as alternative

Standard V-cycle multigrid:

initialization $i = 0$, $R^{(0)} = F$, $r = r_0 = \|F\|_F$
while $r > tol * r_0$ & $i \leq maxit$ **do**
$C^{(i)} = \text{VCYCLE}(\mathcal{A}, 0, R^{(i)})$
$\tilde{U}^{(i+1)} = U^{(i)} + C^{(i)}$
$\tilde{R}^{(i+1)} = F - \mathcal{A}(U^{(i+1)})$
$r = \|R^{(i+1)}\|_F$, $i \leftarrow i + 1$
end

function $U^h = \text{VCYCLE}(\mathcal{A}^h, U_0^h, F^h)$
if $h == h_0$ **then**
solve $\mathcal{A}^h(U^h) = F^h$ directly
else
$U^h = \text{SMOOTH}(\mathcal{A}^h, U_0^h, F^h)$
$\tilde{R}^h = F^h - \mathcal{A}^h(U^h)$
$R^{2h} = \mathcal{R}(R^h)$
$C^{2h} = \text{VCYCLE}(\mathcal{A}^{2h}, 0, R^{2h})$
$U^h = U^h + \mathcal{P}(C^{2h})$
$U^h = \text{SMOOTH}(A^h, U^h, F^h)$
end

function $U = \text{SMOOTH}(\mathcal{A}, U, F)$
for ν steps **do**
$\tilde{U} = U + \mathcal{S}(F - \mathcal{A}(U))$

end

Low-rank variant of multigrid:

initialization $i = 0$, $R^{(0)} = F$ in low-rank format, $r = r_0 = \|F\|_F$
while $r > tol * r_0$ & $i \leq maxit$ **do**
$C^{(i)} = \text{VCYCLE}(\mathcal{A}, 0, R^{(i)})$
$\tilde{U}^{(i+1)} = U^{(i)} + C^{(i)}$, $\quad U^{(i+1)} = \mathcal{T}_{\text{abs}}(\tilde{U}^{(i+1)})$
$\tilde{R}^{(i+1)} = F - \mathcal{A}(U^{(i+1)})$, $\quad R^{(i+1)} = \mathcal{T}_{\text{abs}}(\tilde{R}^{(i+1)})$
$r = \|R^{(i+1)}\|_F$, $i \leftarrow i + 1$
end

function $U^h = \text{VCYCLE}(\mathcal{A}^h, U_0^h, F^h)$
if $h == h_0$ **then**
solve $\mathcal{A}^h(U^h) = F^h$ directly
else
$U^h = \text{SMOOTH}(\mathcal{A}^h, U_0^h, F^h)$
$\tilde{R}^h = F^h - \mathcal{A}^h(U^h)$, $\quad R^h = \mathcal{T}_{\text{rel}_2}(\tilde{R}^h)$
$R^{2h} = \mathcal{R}(R^h)$
$C^{2h} = \text{VCYCLE}(\mathcal{A}^{2h}, 0, R^{2h})$
$U^h = U^h + \mathcal{P}(C^{2h})$
$U^h = \text{SMOOTH}(A^h, U^h, F^h)$
end

function $U = \text{SMOOTH}(\mathcal{A}, U, F)$
for ν steps **do**
$\tilde{U} = U + \mathcal{S}(F - \mathcal{A}(U))$
$U = \mathcal{T}_{\text{rel}_1}(\tilde{U})$
end

Truncation strategies depend on tolerances ϵ_{abs}, ϵ_{rel}

Convergence analysis

Theorem

Let $\mathbf{e}^{(i)} = \mathbf{u} - \mathbf{u}^{(i)}$ *denote the error at the* i*th iteration of the low-rank multigrid algorithm. Then*

$$\|\mathbf{e}^{(i+1)}\|_{\mathcal{A}} \leq C_1(\nu)\|\mathbf{e}^{(i)}\|_{\mathcal{A}} + C_2(\nu)\sqrt{n_\xi}\,\epsilon_{abs}$$

where ϵ_{abs} *is the truncation tolerance for rank reduction,* $C_1(\nu) < 1$ *for small enough* ν *and* $C_2(\nu)$ *is bounded. Moreover,*

$$\|\mathbf{e}^{(i)}\|_{\mathcal{A}} \leq C_1^i(\nu)\|\mathbf{e}^{(0)}\|_{\mathcal{A}} + \left(\frac{1 - C_1(\nu)^i}{1 - C_1(\nu)}\right) C_2(\nu)\sqrt{n_\xi}\,\epsilon_{abs}.$$

Meaning: Error is reduced in a manner analogous to standard multigrid until it becomes similar in magnitude to the truncation tolerance.

Approach of proof

$\mathbf{e}^{(i)} =$ error at step i

$\mathbf{e}^{(i)}_{notrunc} =$ error that would be obtained if no truncation were done at step i

$$\mathbf{e}^{(i+1)} = \mathbf{e}^{(i+1)}_{notrunc} + \text{perturbation}$$

$\|\mathbf{e}^{(i+1)}_{notrunc}\| \leq C(\nu)\|\mathbf{e}^{(i)}\|$ usual multigrid bound,
$\nu = \#$ smoothing steps

$$\|\text{perturbation}\| \leq \hat{C}(\nu)\|\mathbf{e}^{(i)}\| + C_2(\nu)\sqrt{n_\xi}\,\epsilon$$

$$\|\mathbf{e}^{(i+1)}\| \leq C_1(\nu)\|\mathbf{e}^{(i)}\| + C_2(\nu)\sqrt{n_\xi}\epsilon, \qquad C_1(\nu) = C(\nu) + \hat{C}(\nu)$$

Significant feature enabling analysis:

MG is a linear operation

More amenable than (nonlinear) Krylov subspace methods

Experimental results, benchmark problem:

Diffusion equation $-\nabla \cdot (a(\mathbf{x}, \boldsymbol{\xi}) \nabla u) = f$ on $\mathcal{D} = [-1, 1] \times [-1, 1]$,
$f \equiv 1$, Dirichlet b.c.

Diffusion coefficient $a(\mathbf{x}, \boldsymbol{\xi}) = a_0(\mathbf{x}) + \sqrt{3} \sum_{\ell=1}^{m} \sqrt{\lambda_\ell}\, a_\ell(\mathbf{x})\, \xi_\ell$

$a_0 = 1$, $\{(\lambda_\ell, a_\ell)\}$ discrete eigenvalues/eigenvectors of
exponential covariance function

$$c(x, y) = \sigma^2 \exp\left(-\tfrac{1}{b}\|x - y\|_1\right)$$

$\{\xi_\ell\}$ uniformly distributed on $[-1, 1]$, assumed to be independent

N.B. Shorter correlation length $b \longrightarrow$ larger m, chosen s.t.

$$\left(\sum_{j=1}^{m} \lambda_j\right) \Big/ \left(\sum_{j=1}^{\infty} \lambda_j\right) \geq 95\%$$

Discretization: piecewise bilinear fem in physical space
stochastic Galerkin in parameter space,
polynomial chaos, total degree $p = 3$

Example of performance:

Fix the test problem with $\sigma = .01$ and $b = 4$ giving $m = 11$ parameters and $n_\xi = 364$. Vary the spatial discretization and the truncation tolerance.

		$\epsilon_{\text{abs}} = 10^{-6}$	$\epsilon_{\text{abs}} = 10^{-4}$	No truncation	
64×64 grid $h = 2^{-5}$ $N_x = 3969$	Rank Iterations Elapsed time Rel residual	51 5 6.26 1.51e-6	12 4 1.63 6.05e-5	 5 12.60 9.97e-7	 4 10.08 1.38e-5
128×128 grid $h = 2^{-6}$ $N_x = 16129$	Rank Iterations Elapsed time Rel residual	51 6 20.90 2.45e-6	12 4 5.17 9.85e-5	 5 54.59 1.23e-6	 3 32.92 2.20e-4
256×256 grid $h = 2^{-7}$ $N_x = 65025$	Rank Iterations Elapsed time Rel residual	49 5 76.56 4.47e-6	13 4 24.31 2.07e-4	 5 311.27 1.36e-6	 3 188.70 2.35e-04
512×512 grid $h = 2^{-8}$ $N_x = 261121$	Rank Iterations Elapsed time Rel residual	39 5 370.98 9.93e-6	16 3 86.30 4.33e-4	 4 2857.82 1.85e-5	 3 2099.06 2.43e-4

Example of performance:

Fix the spatial discretization ($n_x = 16129$) and size of the parameter space ($n_\xi = 364$), vary the magnitude of the parameterized perturbation from the mean by varying σ. Time spent on truncation in parens.

		$\epsilon_{\text{abs}} = 10^{-6}$	$\epsilon_{\text{abs}} = 10^{-4}$	No truncation	
$\sigma = 0.001$	Rank	13	12		
	Iterations	6	4	5	4
	Elapsed time	7.61 (4.77)	3.73 (2.29)	54.43	43.58
	Rel residual	1.09e-6	6.53e-5	1.22e-6	1.63e-5
$\sigma = 0.01$	Rank	51	12		
	Iterations	6	4	5	3
	Elapsed time	20.90 (15.05)	5.17 (3.16)	54.59	32.92
	Rel residual	2.45e-6	9.85e-5	1.23e-6	2.20e-4
$\sigma = 0.1$	Rank	136	54		
	Iterations	6	4	5	3
	Elapsed time	54.44 (33.91)	18.12 (12.70)	55.49	33.62
	Rel residual	3.28e-6	2.47e-4	1.88e-6	2.62e-4
$\sigma = 0.3$	Rank	234	128		
	Iterations	9	7	8	4
	Elapsed time	138.63 (77.54)	60.96 (38.66)	86.77	43.42
	Rel residual	6.03e-6	4.71e-4	2.99e-6	7.76e-4

Reduced-order models

Different approach for low-rank solutions

(Boyaval, Le Bris, Levire, Maday, Nguyen, Patera, many others; Quarteroni, Manzoni, Negri; Hesthaven, Rozza, Stamm)

Consider (again) parameter-dependent coefficient with affine structure

$$a(\mathbf{x},\boldsymbol{\xi}) \equiv a_0(x) + \sum_{\ell=1}^{m} a_\ell(\mathbf{x})\phi_\ell(\boldsymbol{\xi})$$

- Discrete system $A(\boldsymbol{\xi})\mathbf{u}(\boldsymbol{\xi}) = \mathbf{f}_0$, $A(\boldsymbol{\xi}) \equiv A_0 + \sum_{\ell=0}^{m} A_\ell\phi_\ell(\boldsymbol{\xi}))$
- Compute solutions (**snapshots**)
  $\{\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \ldots, \mathbf{u}^{(n_r)}\}$, $\mathbf{u}^{(j)} = \mathbf{u}(\boldsymbol{\xi}^{(\mathbf{j})})$, $n_r \ll N$
- **Reduced basis** in columns of matrix Q,
  $\text{range}(Q) = \text{span}\{\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \ldots, \mathbf{u}^{(n_r)}\}$
- For other $\boldsymbol{\xi}$, surrogate $\mathbf{u}^s(\boldsymbol{\xi}) \approx \mathbf{u}(\boldsymbol{\xi})$, $\mathbf{u}^s(\boldsymbol{\xi}) = Q\mathbf{y}(\boldsymbol{\xi}) \in \text{span}\{\mathbf{u}^{(\mathbf{j})}\}$

  Determined by Galerkin condition:
  $A(\boldsymbol{\xi})\mathbf{u}^s - \mathbf{f}_0 \perp \text{span}\{\mathbf{u}^{(j)}\}$, $\quad Q^T A(\boldsymbol{\xi})Q\mathbf{y}(\boldsymbol{\xi})) = Q^T\mathbf{f}_0$

Features and key points of this methodology:

- How to choose snapshots?
  Main approach: *greedy search* based on error indicator $\eta(\mathbf{u}^s)$
  Given a candidate basis, find $\boldsymbol{\xi}^* \equiv argmax_{\boldsymbol{\xi}\in\mathcal{S}}\ \eta(\mathbf{u}^s(\boldsymbol{\xi}))$
  if $\eta(\mathbf{u}^s(\boldsymbol{\xi}^*)) > tolerance$, add $\mathbf{u}(\boldsymbol{\xi}^*)$ to basis
- Once basis is obtained: Galerkin system is

$$\underbrace{\left[\sum_{\ell=1}^{m} \underbrace{\left(Q^T A_\ell Q\right)}\, \phi_\ell(\boldsymbol{\xi})\right]}_{\text{Matrix of order } n_r} \mathbf{y}(\boldsymbol{\xi}) = Q^T \mathbf{f} \qquad (1)$$

  Precompute
- Simulation: New $\boldsymbol{\xi} \longrightarrow$ new system (1)
  Construct, solve at cost depending on $n_r \ll n_x$
- Tasks divided between *offline* step (generate basis, may be expensive) and *online step*, intended to be cheap.

Comparison of approaches for computing low-rank surrogates:

- Low-rank methods
  - Truncation may limit accuracy
  - Offline computations less expensive
  - Online computations very cheap
- Reduced-basis methods:
  - "Certified" accuracy, guaranteed small error if training set is large enough
  - Offline step can be very expensive (because of search)
  - Online step requires system solution

1 Problem Statement

2 Background: Some early results on iterative solution

3 Low-rank / Reduced-order models

4 Nonlinear problems
- Empirical Interpolation Methods
- Application to Navier-Stokes Equations
- Construction of bases
- Experimental results
- Preconditioning

Nonlinear problems

For nonlinear system $G(\mathbf{u}(\xi)) = \mathbf{0}$, reduced operator $Q^T \underbrace{G(Q\mathbf{y}(\xi))}$

n_x (scalar) nonlinear function evaluations

Jacobian $J_G(Q\mathbf{y})$, cost of evaluation also depends on n_x

Advantages of reduced basis are gone

Empirical Interpolation Methods (EIM, DEIM)

(Barrault, Maday, Nguyen, & Patera, Chaturantabut & Sorensen)

For $G(\mathbf{u}) = A_\xi \mathbf{u} + F(\mathbf{u}) - \mathbf{b}$, reduced model has form

$$G^{(r)}(\hat{\mathbf{u}}) = Q^T A_\xi Q\, \mathbf{y} + Q^T F(Q\mathbf{y}) - Q^T \mathbf{b}$$

Empirical Interpolation

Strategy for approximating nonlinear term:

- Generate matrix of snapshots $S \equiv \left[F(\mathbf{u}^{(1)}),\ F(\mathbf{u}^{(2)}), \ldots, F(\mathbf{u}^{(\mathbf{M})})\right]$
- Generate low-rank Φ for which $range(S) \approx range(\Phi)$ (via SVD)
  $n_s \equiv rank(\Phi)$, analogous to n_r
- Identify "index choosing" matrix $P = [e_{i_1}, e_{i_2}, \ldots, e_{i_{n_s}}]$
- Replace $F(Q\mathbf{y})$ with approximation $\widehat{F}(Q\mathbf{y}) \equiv \Phi(P^T\Phi)^{-1}P^TF(Q\mathbf{y})$
  $\longrightarrow$ approximation $\widehat{G}(Q\mathbf{y}) = A_\xi Q\mathbf{y} + \widehat{F}(Q\mathbf{y}) - \mathbf{b}$
- Galerkin condition: $Q^T\widehat{G}(Q\mathbf{y}) = 0$

$$Q^TA_\xi Q\mathbf{y} + \underbrace{Q^T\Phi(P^T\Phi)^{-1}}_{\text{Precompute}} P^TF(Q\mathbf{y}) - Q^T\mathbf{b} = 0$$

- Approximation interpolates desired quantity at indices of P:

Navier-Stokes equations

- **Example:** Parameterized Navier-Stokes equations

$$-\nabla \cdot (a(\cdot, \boldsymbol{\xi}) \nabla \vec{u}) + (\vec{u} \cdot \nabla)\vec{u} + \nabla p = \vec{f}, \quad \nabla \cdot \vec{u} = 0$$

  Parameterized diffusion coefficient $a(\mathbf{x}, \boldsymbol{\xi})$

  First component (from momentum equation) of algebraic system contains function of velocity

$$G(\mathbf{u}(\boldsymbol{\xi})) = A_{\boldsymbol{\xi}}\mathbf{u}(\boldsymbol{\xi}) + F(\mathbf{u}(\boldsymbol{\xi})) - \mathbf{b}$$

  $A_{\boldsymbol{\xi}}$ = discrete parameter-dependent diffusion operator

  $F(\mathbf{u}) = N(\mathbf{u})\mathbf{u}$ = discrete version of $(\vec{u} \cdot \nabla)\vec{u}$

- **Notation:**

  Solutions: velocity $\mathbf{u}$, pressure $\mathbf{p}$

  Approximations from reduced model: $\tilde{\mathbf{u}}$, $\tilde{\mathbf{p}}$

  Analogous quantities in reduced spaces: $\mathbf{y}$, $\mathbf{z}$

Reduced Models for Navier-Stokes

For Stokes equations (Rozza & Veroy)

$$\begin{aligned} (A_\xi \mathbf{u} + B^T \mathbf{p} - \mathbf{f}, \mathbf{v}) &= 0 \quad \text{for all } \mathbf{v} \\ (B\mathbf{u} - \mathbf{g}, q) &= 0 \quad \text{for all } \mathbf{q} \end{aligned}$$

Seeking subspaces of velocity and pressure spaces. Try building using snapshot solutions $\begin{bmatrix} \mathbf{u}^{(1)} \\ \mathbf{p}^{(1)} \end{bmatrix}, \cdots, \begin{bmatrix} \mathbf{u}^{(n_r)} \\ \mathbf{p}^{(n_r)} \end{bmatrix}, \quad \begin{bmatrix} \mathbf{u}^{(j)} \\ \mathbf{p}^{(j)} \end{bmatrix}$ = Stokes solution with parameter $\boldsymbol{\xi}^{(j)}$

$\rightarrow$ velocity basis Q_u, range $= span\{\mathbf{u}^{(j)}\}$
req. augmentation for inf-sup stability (Quarteroni, Rozza, Veroy)
pressure basis Q_p, range $= span\{\mathbf{p}^{(j)}\}$

Leads to reduced operator with saddle-point structure:

$$\begin{bmatrix} Q_u^T & 0 \\ 0 & Q_p^T \end{bmatrix} \begin{bmatrix} A_\xi & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} Q_u & 0 \\ 0 & Q_p \end{bmatrix} = \begin{bmatrix} Q_u^T A_\xi Q_u & Q_u^T B^T Q_p \\ Q_p^T B Q_u & 0 \end{bmatrix}$$

Step n of Picard iteration for full system has form:

$$\left(\begin{bmatrix} A_\xi & B^T \\ B & 0 \end{bmatrix} + \begin{bmatrix} N(\mathbf{u}_n) & 0 \\ 0 & 0 \end{bmatrix} \right) \begin{bmatrix} \delta \mathbf{u}_n \\ \delta \mathbf{p}_n \end{bmatrix} = \begin{bmatrix} \mathbf{f}(\mathbf{u}_n, \mathbf{p}_n) \\ \mathbf{g}(\mathbf{u}_n, \mathbf{p}_n) \end{bmatrix}$$

For reduced system:

$$\left(Q^T \begin{bmatrix} A_\xi & B^T \\ B & 0 \end{bmatrix} Q + Q^T \begin{bmatrix} N(\tilde{\mathbf{u}}_n) & 0 \\ 0 & 0 \end{bmatrix} Q \right) \begin{bmatrix} \delta \mathbf{y}_n \\ \delta \mathbf{z}_n \end{bmatrix} = \begin{bmatrix} \mathbf{f}^{(r)}(\mathbf{y}_n, \mathbf{z}_n) \\ \mathbf{g}^{(r)}(\mathbf{y}_n, \mathbf{z}_n) \end{bmatrix}$$

For DEIM system:

$$\left(Q^T \begin{bmatrix} A_\xi & B^T \\ B & 0 \end{bmatrix} Q + \begin{bmatrix} L^T P^T N(\tilde{\mathbf{u}}_n) Q_u & 0 \\ 0 & 0 \end{bmatrix} \right) \begin{bmatrix} \delta \mathbf{y}_n \\ \delta \mathbf{z}_n \end{bmatrix} = \begin{bmatrix} \mathbf{f}^{(r)}(\mathbf{y}_n, \mathbf{z}_n) \\ \mathbf{g}^{(r)}(\mathbf{y}_n, \mathbf{z}_n) \end{bmatrix}$$

$$L^T \equiv Q_u^T \Phi (P^T \Phi)^{-1}$$

DEIM Algorithm for Navier-Stokes

Initialize: solve reduced Stokes problem

$$\left(Q^T \begin{bmatrix} A(\xi) & B^T \\ B & 0 \end{bmatrix} Q \right) \begin{bmatrix} \mathbf{y}_0 \\ \mathbf{z}_0 \end{bmatrix} = Q^T \mathbf{b}$$

for $n = 0, 1, \ldots$ (Picard iteration)

solve reduced problem

$$\left(Q^T \begin{bmatrix} A_\xi & B^T \\ B & 0 \end{bmatrix} Q + \begin{bmatrix} L^T (P^T N(\tilde{\mathbf{u}}_n) Q_u & 0 \\ 0 & 0 \end{bmatrix} \right) \begin{bmatrix} \delta \mathbf{y}_n \\ \delta \mathbf{z}_n \end{bmatrix} = - \begin{bmatrix} \mathbf{f}^{(r)}(\mathbf{y}_n, \mathbf{z}_n) \\ \mathbf{g}^{(r)}(\mathbf{y}_n, \mathbf{z}_n) \end{bmatrix}$$

update reduced iterate

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \delta \mathbf{y}_n, \quad \mathbf{z}_{n+1} = \mathbf{z}_n + \delta \mathbf{z}_n$$

compute $P^T N(\tilde{\mathbf{u}}_{n+1}) = P^T N(Q_u y_{n+1})$ **(only at required indices)**

exit if $$\left\| \begin{bmatrix} \mathbf{f}^{(r)}(\mathbf{y}_{n+1}, \mathbf{z}_{n+1}) \\ \mathbf{g}^{(r)}(\mathbf{y}_{n+1}, \mathbf{z}_{n+1}) \end{bmatrix} \right\|_2 < \tau \| Q^T \mathbf{b} \|_2$$

end

Construction of Reduced Basis

Offline step: Choose snapshots using random sampling

Our experience: as effective as greedy sampling, much faster

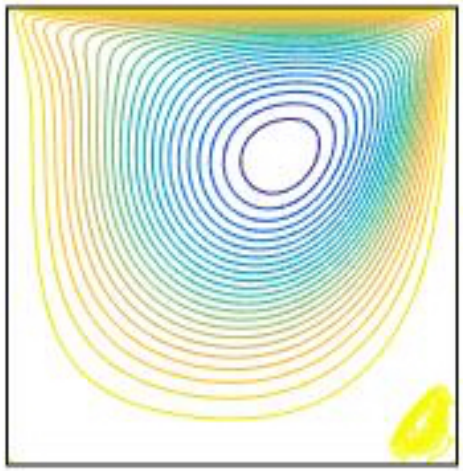
Initialize reduced basis with $\mathbf{u}(\boldsymbol{\xi}^{(0)})$, $\mathbf{p}(\boldsymbol{\xi}^{(0)})$, enriched velocity $\mathbf{w}(\boldsymbol{\xi}^{(0)})$
for $n_{trial} = 2000$ random samples of $\boldsymbol{\xi}$
  Compute reduced solution $\mathbf{u}(\boldsymbol{\xi})$, $\mathbf{p}(\boldsymbol{\xi})$ and error indicator
  if error indicator $>$ tolerance τ
    Compute full solution and enriched velocity, augment basis
  endif
end

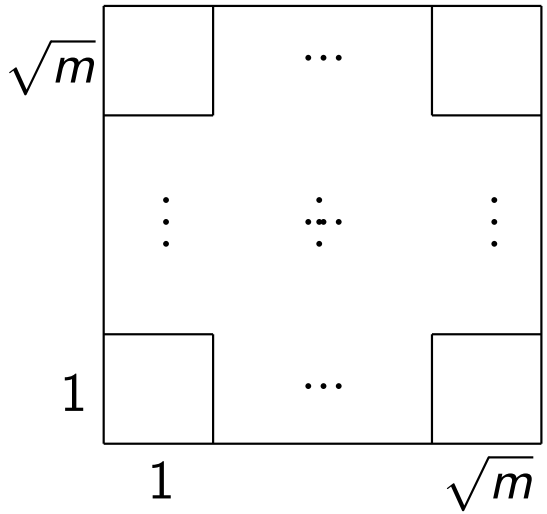
For nonlinear snapshot matrix $S = \left[F(\mathbf{u}^{(1)}),\ F(\mathbf{u}^{(2)}), \ldots, F(\mathbf{u}^{(\mathbf{M})})\right]$

Options: (i) use $M = n_{trial}$ ($\sim$ Chaturantabut & Sorensen)
(ii) only include $\mathbf{u}(\boldsymbol{\xi})$ in S when it's added to the reduced basis – much cheaper

Benchmark problem

Driven cavity flow, piecewise constant viscosity on $\sqrt{m} \times \sqrt{m}$ subdomains





Piecewise constant viscosity

$$\nu(\mathbf{x}, \boldsymbol{\xi}) = \sum_{r=1}^{m} a_r(\mathbf{x})\, \xi_r, \quad a_r = \chi_{D_r}$$

parameterized by random variables $\boldsymbol{\xi} = [\xi_1, \cdots, \xi_m]^T$ independently and uniformly distributed in $\Gamma = [0.01, 1]^m$

Experiment: Solve three versions of the discrete NS equations using Picard iteration:

1. the discrete full system (with sparse direct methods for solves)
2. the discrete reduced system w/o special treatment of nonlinear term
3. the discrete reduced system obtained from DEIM

Report: Average CPU times over 10 simulations
Relative residual norms $\eta \equiv \|F_{\boldsymbol{\xi}}\|_2/\|\mathbf{b}\|_2$
N.B. this error measure is not available at cost $<$ O(N)

Important note:

Stopping tolerance for reduced residual is $\delta = 10^{-8} \rightarrow \eta \sim 5 \cdot 10^{-5}$
For full problem, get same η with tolerance $\delta = 10^{-4}$

64×64 spatial grid

m	4		16		36		49	
k	273		1503		3339		4083	
n_{deim}	91		501		1113		1485	
	time	η	time	η	time	η	time	η
Full	11	1.E-8	10.2	1.E-8	10.1	1.E-8	10.3	1.E-8
	4.58	4.42E-5	5.61	1.07E-5	**5.70**	1.44E-5	**5.42**	3.97E-5
Reduced	0.36	4.94E-5	7.13	4.49E-5	37.2	5.89E-5	70.8	7.42E-5
DEIM	**0.07**	4.55E-5	**1.57**	4.57E-5	13.4	5.98E-5	29.6	7.51E-5

128×128 spatial grid

m	4		16		36		49	
k	237		1383		3039		4083	
n_{deim}	79		461		1013		1361	
	time	η	time	η	time	η	time	η
Full	135	1.E-8	147	1.E-8	132	1.E-8	148	1.E-8
	55.1	3.24.E-5	53.6	4.50E-5	66.8	3.35E-5	57.1	5.74E-5
Reduced	1.13	1.38E-5	22.2	2.76E-5	101	5.33E-5	196	7.05E-5
DEIM	**0.11**	1.41E-5	**1.76**	2.80E-5	**11.0**	5.49E-5	**24.8**	7.16E-5

Preconditioning for DEIM

During nonlinear iteration, have sequence of systems of order n_r

$$\left(Q^T \begin{bmatrix} A_\xi & B^T \\ B & 0 \end{bmatrix} Q + \begin{bmatrix} Q_u^T \widehat{F}(\tilde{\mathbf{u}}_n) Q_u & 0 \\ 0 & 0 \end{bmatrix} \right) \begin{bmatrix} \delta \mathbf{u}_j \\ \delta \mathbf{p}_j \end{bmatrix} = -r_n^{deim}$$

Want preconditioners whose construction depends on $n_r \ll N$

Changes the game. Choices:

- Stokes ("beginning") preconditioner: $M = Q^T \begin{bmatrix} A(\boldsymbol{\xi}_0) & B^T \\ B & 0 \end{bmatrix} Q$
- Navier-Stokes ("end") preconditioner, from last Picard step for $\boldsymbol{\xi}_0$:

$$M = Q^T \begin{bmatrix} A(\xi_0) & B^T \\ B & 0 \end{bmatrix} Q + \begin{bmatrix} Q_u^T \widehat{F}(\tilde{\mathbf{u}}_n(\boldsymbol{\xi}_0)) Q_u & 0 \\ 0 & 0 \end{bmatrix}$$

Entails computing/factoring M for fixed $\boldsymbol{\xi} = \boldsymbol{\xi}_0$ in "offline" stage

For comparison – **not practical**: "online" version $M(\boldsymbol{\xi})$

Average Bicgstab Iterations

n	m	4	16	36	49
32	k	306	1485		
	n_{deim}	102	495		
	Stokes	9.1	18.0		
	Navier-Stokes	9.1	18.1		
	Online NS	2.0	2.1		
64	k	273	1503	3339	4455
	n_{deim}	91	501	1113	1485
	Stokes	13.2	19.8	21.8	23.8
	Navier-Stokes	13.0	19.9	21.6	23.4
	Online NS	2.1	2.1	2.1	2.2
128	k	732	1383	3039	4083
	n_{deim}	79	461	1013	1361
	Stokes	10.3	16.7	20.6	23.2
	Navier-Stokes	10.4	16.1	20.0	22.8
	Online NS	1.9	1.9	2.2	2.1

Average CPU Times

n	m	4	16	36	49
32	k	306	1485		
	n_{deim}	102	495		
	Full	.44	**.66**		
	DEIM / Direct	**.06**	1.86		
	DEIM / Iterative-NS	**.07**	1.49		
64	k	273	1503	3339	4455
	n_{deim}	91	501	1113	1485
	Full	4.58	5.61	**5.70**	**5.42**
	DEIM / Direct	**0.07**	1.57	13.4	29.6
	DEIM / Iterative-NS	0.09	**1.38**	8.3	17.4
128	k	732	1383	3039	4083
	n_{deim}	79	461	1013	1361
	Full	55.1	53.6	66.8	57.1
	DEIM / Direct	0.11	1.76	11.0	24.8
	DEIM / Iterative-NS	**0.11**	**1.42**	**7.13**	**14.9**

Concluding remarks

- Many interesting issues arising from parameter-dependent PDEs
- Classic iterative methods (multigrid, Krylov subspace methods) can be put to good use
- Other classic methods (SVD, QR factorization) play important roles
- Nonlinear problems not thoroughly explored

References


I. Babuška, R. Tempone, and G. E. Zouraris, *Galerkin finite element approximations of stochastic elliptic partial differential equations*, SIAM J. Numer. Anal., 42 (2004), pp. 800–825.


J. Ballani and L. Grasedyck, *A projection method to solve linear systems in tensor format*, Numer. Linear Alg. with Applications, 20 (2013), pp. 27–43.


M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera, *An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations*, Comptes Rendus Mathematique, 339 (2004), pp. 667–672.


P. Benner, A. Onwunta, and M. Stoll, *Low-rank solution of unsteady diffusion equations with stochastic coefficients*, SIAM/ASA J. Uncertainty Quantification, 3 (2015), pp. 622–649.


S. Boyaval, C. L. Bris, T. Lelièvre, Y. Maday, N. C. Nguyen, and A. T. Patera, *Reduced basis techniques for stochastic problems*, Arch. Comput. Meth. Engrg., 17 (2010), pp. 1–20.


S. Chaturantabut and D. C. Sorensen, *A state space error estimate for POD-DEIM nonlinear model reduction*, SIAM Journal on Numerical Analysis, 50 (2012), pp. 46–63.


H. Elman and D. Furnival, *Solving the stochastic steady-state diffusion problem using multigrid*, IMA J. Numer. Anal., 27 (2007), pp. 675–688.


H. C. Elman and V. Forstall, *Numerical solution of the parameterized steady-state Navier-Stokes equations using empirical interplation methods*, Comput. Methods Appl. Mech. Engrg., 317 (2017), pp. 380–399.


H. C. Elman and T. Su, *Collocation Methods for Exploring Perturbations in Linear Stability Analysis*, Tech. Rep. 1612.05496v2, arXiv, 2017.
To appear in SIAM J. Matrix Anal. Appl.


J. S. Hesthaven, G. Rozza, and B. Stamm, *Certified reduced basis methods for parametrized partial differential equations*, SpringerBriefs in Mathematics. Springer, (2016).


D. Kressner and C. Tobler, *Low-rank tensor Krylov subspace methods for parametrized linear systems*, SIAM J. Matr. Anal. Appl., 32 (2011), pp. 1288–1316.


C. E. Powell and H. C. Elman, *Block-diagonal preconditioning for spectral stochastic finite-element systems*, IMA Journal of Numerical Analysis, 29 (2009), pp. 350–375.


A. Quarteroni and G. Rozza, *Numerical solution of parametrized Navier-Stokes equations by reduced basis methods*, Numerical Methods for Partial Differential Equations, 23 (2007), pp. 923–948.


G. Rozza and K. Veroy, *On the stability of the reduced basis method for Stokes equations in parametrized domains*, Computer Methods in Applied Mechanics and Engineering, 196 (2007), pp. 1244–1260.


D. Xiu, *Numerical Methods for Stochastic Computations: A Spectral Method Approach*, Princeton University Press, 2010.