Low-rank Solution Algorithms for Stochastic Partial Differential Equations

Kookjin Lee

Department of Computer Science
University of Maryland

klee@cs.umd.edu

March 9, 2016

Overview

1. Introduction
   - Problem definition
2. Model problem
   - KL expansion
   - Stochastic Galerkin method
   - Tensor format
3. Preliminary work
   - Low-rank projection method in tensor format
   - Truncation methods
   - Numerical experiments
4. Proposed work
   - Nonlinear random fields
   - Truncation based on randomized tensor decomposition
   - Active subspace methods

1. Introduction
   - Problem definition
2. Model problem
   - KL expansion
   - Stochastic Galerkin method
   - Tensor format
3. Preliminary work
   - Low-rank projection method in tensor format
   - Truncation methods
   - Numerical experiments
4. Proposed work
   - Nonlinear random fields
   - Truncation based on randomized tensor decomposition
   - Active subspace methods

Partial Differential Equations with Stochastic Coefficients

Examples:

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

Convection-diffusion equations: $\nu\nabla \cdot (a(\boldsymbol{x}, \xi)\nabla u) + \vec{w} \cdot \nabla u = f$

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

Partial Differential Equations with Stochastic Coefficients

Examples:

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

Convection-diffusion equations: $\nu\nabla \cdot (a(\boldsymbol{x}, \xi)\nabla u) + \vec{w} \cdot \nabla u = f$

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

Randomness:

Partial Differential Equations with Stochastic Coefficients

Examples:

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

Convection-diffusion equations: $\nu\nabla \cdot (a(\mathbf{x}, \xi)\nabla u) + \vec{w} \cdot \nabla u = f$

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

Randomness:

- An inherent irregularity in the phenomenon being observed e.g., the kinetic theory of gas

Partial Differential Equations with Stochastic Coefficients

Examples:

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

Convection-diffusion equations: $\nu\nabla \cdot (a(\mathbf{x}, \xi)\nabla u) + \vec{w} \cdot \nabla u = f$

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

Randomness:

- An inherent irregularity in the phenomenon being observed e.g., the kinetic theory of gas
- The impossibility of an exhaustive deterministic description e.g., groundwater flow through a heterogeneous porous media

Partial Differential Equations with Stochastic Coefficients

Examples:

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

Convection-diffusion equations: $\nu\nabla \cdot (a(\mathbf{x}, \xi)\nabla u) + \vec{w} \cdot \nabla u = f$

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

Randomness:

- An inherent irregularity in the phenomenon being observed e.g., the kinetic theory of gas
- The impossibility of an exhaustive deterministic description e.g., groundwater flow through a heterogeneous porous media
- $a(\mathbf{x}, \xi)$ is a random process/ random field parameterized by a set of random variables $\xi = [\xi_1, \ldots, \xi_M]^T$
- The numerical solution $u(\mathbf{x}, \xi)$ can be described by ξ

Partial Differential Equations with Stochastic Coefficients

Examples:

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

Convection-diffusion equations: $\nu\nabla \cdot (a(\boldsymbol{x}, \xi)\nabla u) + \vec{w} \cdot \nabla u = f$

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

Randomness:

- An inherent irregularity in the phenomenon being observed e.g., the kinetic theory of gas
- The impossibility of an exhaustive deterministic description e.g., groundwater flow through a heterogeneous porous media
- $a(\boldsymbol{x}, \xi)$ is a random process/ random field parameterized by a set of random variables $\xi = [\xi_1, \ldots, \xi_M]^T$
- The numerical solution $u(\boldsymbol{x}, \xi)$ can be described by ξ

Goal:

Efficient computation of the numerical solution $u(\boldsymbol{x}, \xi)$ and solution statistics (e.g., $\mathbb{E}[u]$, $\mathrm{Var}(u)$) using linear algebraic algorithms

Linear systems:

Large linear systems arise from discretizations of stochastic PDEs:

$$Au = f$$

Linear systems:

Large linear systems arise from discretizations of stochastic PDEs:

$$Au = f$$

- High-dimensional problem: a large M in $\xi = [\xi_1, \ldots, \xi_M]^T$

Linear systems:

Large linear systems arise from discretizations of stochastic PDEs:

$$Au = f$$

- High-dimensional problem: a large M in $\xi = [\xi_1, \ldots, \xi_M]^T$
- Linear systems with a special structure,

$$A = \sum_{i=1} G_i \otimes K_i$$

where $\otimes$ is the Kronecker-product:

$$G \otimes K = \begin{bmatrix} g_{11}K & \cdots & g_{1n_\xi}K \\ \vdots & \ddots & \vdots \\ g_{n_\xi 1}K & \cdots & g_{n_\xi n_\xi}K \end{bmatrix} \in \mathbb{R}^{n_\xi n_x \times n_\xi n_x},$$

$$G \in \mathbb{R}^{n_\xi \times n_\xi} \quad \text{and} \quad K \in \mathbb{R}^{n_x \times n_x}$$

Low-rank solution of linear systems:

Solutions in the Kronecker-product structure:

$$u = \sum_{k=1}^{\kappa_u} y_k \otimes z_k, \qquad y_k \in \mathbb{R}^{n_\xi} \text{ and } z_k \in \mathbb{R}^{n_x}$$

where κ_u is the rank of u

Low-rank solution of linear systems:

Solutions in the Kronecker-product structure:

$$u = \sum_{k=1}^{\kappa_u} y_k \otimes z_k, \qquad y_k \in \mathbb{R}^{n_\xi} \text{ and } z_k \in \mathbb{R}^{n_x}$$

where κ_u is the rank of u

Low-rank approximations to solutions:

$$u \approx \tilde{u} = \sum_{k=1}^{\kappa_{\tilde{u}}} \tilde{y}_k \otimes \tilde{z}_k, \qquad \tilde{y}_k \in \mathbb{R}^{n_\xi} \text{ and } \tilde{z}_k \in \mathbb{R}^{n_x}$$

where $\kappa_{\tilde{u}} \ll \kappa_u$ s.t. $\|A\tilde{u} - f\|_2 / \|f\|_2 < \epsilon$

1 Introduction
- Problem definition

2 Model problem
- KL expansion
- Stochastic Galerkin method
- Tensor format

3 Preliminary work
- Low-rank projection method in tensor format
- Truncation methods
- Numerical experiments

4 Proposed work
- Nonlinear random fields
- Truncation based on randomized tensor decomposition
- Active subspace methods

Karhunen-Loéve expansion

Random field $a(\boldsymbol{x}, \xi)$**:** $-\nabla \cdot (a(\boldsymbol{x}, \xi)\nabla u(\boldsymbol{x}, \xi)) = f$

Karhunen-Loéve expansion

Random field $a(\boldsymbol{x}, \xi)$**:** $-\nabla \cdot (a(\boldsymbol{x}, \xi) \nabla u(\boldsymbol{x}, \xi)) = f$

$a(\boldsymbol{x}, \xi)$ has affine dependence on $\{\xi_i\}_{i=1}^{M}$,

$$a(\boldsymbol{x}, \xi) \approx a^{(M)}(\boldsymbol{x}, \xi) = a_0(\boldsymbol{x}) + \sum_{i=1}^{M} a_i(\boldsymbol{x}) \xi_i$$

Karhunen-Loéve expansion

Random field $a(\boldsymbol{x}, \xi)$: $-\nabla \cdot (a(\boldsymbol{x}, \xi) \nabla u(\boldsymbol{x}, \xi)) = f$

$a(\boldsymbol{x}, \xi)$ has affine dependence on $\{\xi_i\}_{i=1}^M$,

$$a(\boldsymbol{x}, \xi) \approx a^{(M)}(\boldsymbol{x}, \xi) = a_0(\boldsymbol{x}) + \sum_{i=1}^{M} a_i(\boldsymbol{x}) \xi_i$$

In this study, a truncated Karhunen-Loéve expansion is considered,

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

- (μ, σ^2) are mean and variance of $a(\boldsymbol{x}, \xi)$
- $\{(\lambda_i, a_i)\}_{i=1}^M$ are eigenvalue and eigenfunction pairs of an integral operator of covariance function, $C(\boldsymbol{x}, \boldsymbol{y})$, $\boldsymbol{x}, \boldsymbol{y} \in D$, of $a(\boldsymbol{x}, \xi)$
- $\{\xi_i\}_{i=1}^M$ are uncorrelated random variables (additional assumption: i.i.d.)

Eigenpairs $\{(\lambda_i, a_i)\}_{i=1}^M$ can be obtained by solving:

$$\int_D C(\boldsymbol{x}, \boldsymbol{y}) a_i(\boldsymbol{y}) d\boldsymbol{y} = \lambda_i a_i(\boldsymbol{x}), \quad i = 1 \ldots, M$$

The series converges in L^2 sense:

$$\lim_{M \to \infty} \left\langle \left(a(\boldsymbol{x}, \xi) - a^{(M)}(\boldsymbol{x}, \xi) \right)^2 \right\rangle_\rho = 0$$

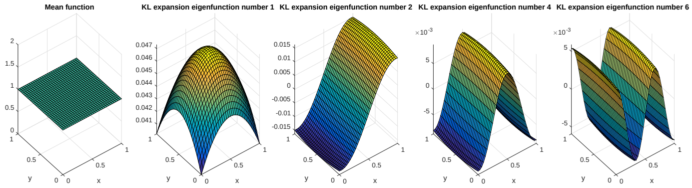


Figure: Mean function and example eigenfunctions

1. Introduction
   - Problem definition
2. Model problem
   - KL expansion
   - Stochastic Galerkin method
   - Tensor format
3. Preliminary work
   - Low-rank projection method in tensor format
   - Truncation methods
   - Numerical experiments
4. Proposed work
   - Nonlinear random fields
   - Truncation based on randomized tensor decomposition
   - Active subspace methods

Solution $u(\boldsymbol{x}, \xi)$: $-\nabla \cdot (a(\boldsymbol{x}, \xi) \nabla u(\boldsymbol{x}, \xi)) = f$

Generalized polynomial chaos expansion:

$$u(\boldsymbol{x}, \xi) \approx \sum_{s=1}^{n_\xi} u_s(\boldsymbol{x}) \psi_s(\xi)$$

Solution $u(\boldsymbol{x}, \xi)$**:** $-\nabla \cdot (a(\boldsymbol{x}, \xi) \nabla u(\boldsymbol{x}, \xi)) = f$

Generalized polynomial chaos expansion:

$$u(\boldsymbol{x}, \xi) \approx \sum_{s=1}^{n_\xi} u_s(\boldsymbol{x}) \psi_s(\xi)$$

- Orthogonality basis: $\int_\Gamma \psi_i(\xi)\psi_j(\xi)\rho(\xi)d\xi = \delta_{ij}$

Solution $u(\boldsymbol{x}, \xi)$**:** $-\nabla \cdot (a(\boldsymbol{x}, \xi) \nabla u(\boldsymbol{x}, \xi)) = f$

Generalized polynomial chaos expansion:

$$u(\boldsymbol{x}, \xi) \approx \sum_{s=1}^{n_\xi} u_s(\boldsymbol{x}) \psi_s(\xi)$$

- Orthogonality basis: $\int_\Gamma \psi_i(\xi)\psi_j(\xi)\rho(\xi)d\xi = \delta_{ij}$
- Product form: $\psi_s(\xi) = \psi_{\alpha(s)}(\xi) = \prod_{i=1}^{M} \pi_{\alpha_i(s)}(\xi_i)$, $\alpha(s) = (\alpha_1(s), \ldots, \alpha_M(s))$

Solution $u(\boldsymbol{x}, \xi)$**:** $-\nabla \cdot (a(\boldsymbol{x}, \xi)\nabla u(\boldsymbol{x}, \xi)) = f$

Generalized polynomial chaos expansion:

$$u(\boldsymbol{x}, \xi) \approx \sum_{s=1}^{n_\xi} u_s(\boldsymbol{x})\psi_s(\xi)$$

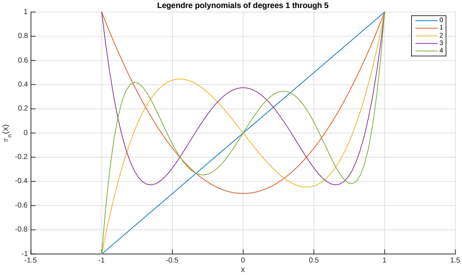
- Orthogonality basis: $\int_\Gamma \psi_i(\xi)\psi_j(\xi)\rho(\xi)d\xi = \delta_{ij}$
- Product form: $\psi_s(\xi) = \psi_{\alpha(s)}(\xi) = \prod_{i=1}^{M} \pi_{\alpha_i(s)}(\xi_i)$, $\alpha(s) = (\alpha_1(s), \ldots, \alpha_M(s))$
- Total degree space:

$$\Lambda^{M,p} = \{\alpha(s) \in \mathbb{N}_0^M : \|\alpha(s)\|_1 \leq p\},$$

where $\|\alpha(s)\|_1 = \sum_{k=1}^{M} \alpha_k(s)$ and $n_\xi = \dim(\Lambda^{M,p}) = \frac{(M+p)!}{M!p!}$ (DoFs of stochastic domain)

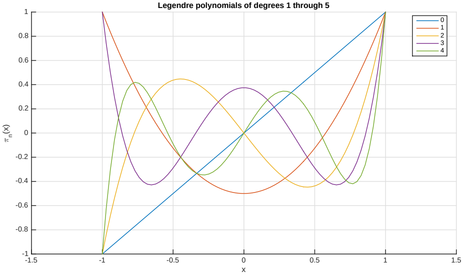
If $\{\xi_i\}$ are uniform random variables
$\{\pi_i\}$ are Legendre polynomials.

$$\pi_0(\xi_i) = 1,$$
$$\pi_1(\xi_i) = \xi_i,$$
$$\pi_2(\xi_i) = \frac{1}{2}(3\xi_i^2 - 1),$$
$$\pi_3(\xi_i) = \frac{1}{2}(5\xi_i^3 - 3\xi_i)$$
$$\pi_4(\xi_i) = \frac{1}{8}(35\xi_i^4 - 30\xi_i^2 + 3)$$



If $\{\xi_i\}$ are uniform random variables
$\{\pi_i\}$ are Legendre polynomials.

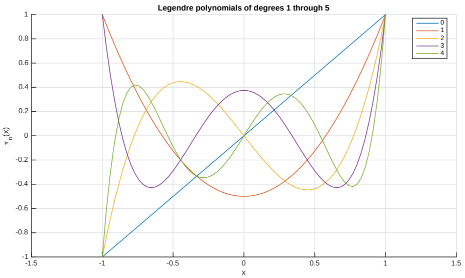
$$\pi_0(\xi_i) = 1,$$
$$\pi_1(\xi_i) = \xi_i,$$
$$\pi_2(\xi_i) = \frac{1}{2}(3\xi_i^2 - 1),$$
$$\pi_3(\xi_i) = \frac{1}{2}(5\xi_i^3 - 3\xi_i)$$
$$\pi_4(\xi_i) = \frac{1}{8}(35\xi_i^4 - 30\xi_i^2 + 3)$$



$$\Lambda^{2,3} = \{(\alpha_1(s),\, \alpha_2(s)\}_{s=1}^{10}$$
$$= \{(0,0),(1,0),(2,0),(3,0),(0,1),(1,1),(2,1),(0,2),(1,2),(0,3)\}$$

If $\{\xi_i\}$ are uniform random variables $\{\pi_i\}$ are Legendre polynomials.

$$\pi_0(\xi_i) = 1,$$
$$\pi_1(\xi_i) = \xi_i,$$
$$\pi_2(\xi_i) = \frac{1}{2}(3\xi_i^2 - 1),$$
$$\pi_3(\xi_i) = \frac{1}{2}(5\xi_i^3 - 3\xi_i)$$
$$\pi_4(\xi_i) = \frac{1}{8}(35\xi_i^4 - 30\xi_i^2 + 3)$$



$$\Lambda^{2,3} = \{(\alpha_1(s), \alpha_2(s)\}_{s=1}^{10}$$
$$= \{(0,0),(1,0),(2,0),(3,0),(0,1),(1,1),(2,1),(0,2),(1,2),(0,3)\}$$
$$\psi_1 = \pi_0(\xi_1)\pi_0(\xi_2),\ \psi_2 = \pi_1(\xi_1)\pi_0(\xi_2),\ \psi_3 = \pi_2(\xi_1)\pi_0(\xi_2),\ \ldots$$

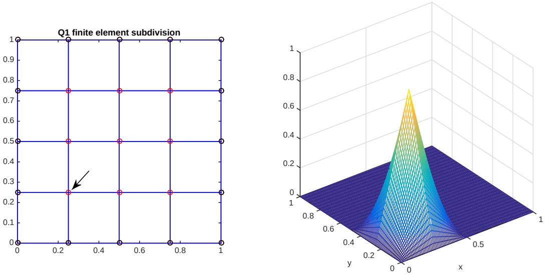
and $n_\xi = 10$

Solution: $u(\boldsymbol{x}, \xi) \approx \sum_{s=1}^{n_\xi} u_s(\boldsymbol{x})\psi_s(\xi)$

Finite Element Methods:

$$u_s(\boldsymbol{x}) \approx \sum_{r=1}^{n_x} u_{rs}\phi_r(\boldsymbol{x})$$

using piecewise linear "hat functions"



# of nodes $= n_x$ (DoFs of deterministic domain)

The Stochastic Galerkin Method

Discrete solution:

- Discretization in physical space (Finite Element Methods): basis $\{\phi_r\}_{r=1}^{n_x}$, piecewise linear "hat functions"
- Discretization in stochastic space (Polynomial Chaos Expansion): basis $\{\psi_s\}_{s=1}^{n_\xi}$, M-variate polynomials in ξ of total degree p

The Stochastic Galerkin Method

Discrete solution:

- Discretization in physical space (Finite Element Methods): basis $\{\phi_r\}_{r=1}^{n_x}$, piecewise linear "hat functions"
- Discretization in stochastic space (Polynomial Chaos Expansion): basis $\{\psi_s\}_{s=1}^{n_\xi}$, M-variate polynomials in ξ of total degree p

$$u^{(sg)}(\mathbf{x}, \xi) = \sum_{s=1}^{n_\xi} \sum_{r=1}^{n_x} u_{rs} \phi_r(\mathbf{x}) \psi_s(\xi)$$

The weak formulation leads to a large coupled system order of $n_x n_\xi$

$$Au = f$$

where $u = [u_{11} \; \dots \; u_{n_x 1} \; u_{12} \; \dots \; u_{n_x 2} \quad \dots \quad u_{1 n_\xi} \; \dots \; u_{n_x n_\xi}]^T$

The Stochastic Galerkin Method

Strong formulation: $-\nabla \cdot (a(\boldsymbol{x}, \xi)\nabla u(\boldsymbol{x}, \xi)) = f$

(Deterministic) weak formulation: find $u \in H^1_E(D)$ s.t.

$$\int_D a\nabla u \nabla v \, d\boldsymbol{x} = \int_D fv \, d\boldsymbol{x}, \quad \forall v \in H^1_0(D)$$

The Stochastic Galerkin Method

Strong formulation: $-\nabla \cdot (a(\boldsymbol{x}, \xi)\nabla u(\boldsymbol{x}, \xi)) = f$

(Deterministic) weak formulation: find $u \in H^1_E(D)$ s.t.

$$\int_D a\nabla u \nabla v \, d\boldsymbol{x} = \int_D fv \, d\boldsymbol{x}, \quad \forall v \in H^1_0(D)$$

Stochastic weak formulation: find $u \in H^1_E(D) \otimes L_2(\Gamma)$ s.t.

$$\int_\Gamma \int_D a\nabla u \nabla v \, d\boldsymbol{x}\rho(\xi)d\xi = \int_\Gamma \int_D fv \, d\boldsymbol{x}\rho(\xi)d\xi, \quad \forall v \in H^1_0(D) \otimes L_2(\Gamma)$$

where $\Gamma = \prod \Gamma_i$ and $\Gamma_i = \xi_i(\Omega)$

LHS:
Substituting the truncated KL expansion $a(\boldsymbol{x}, \xi) = a_0(\boldsymbol{x}) + \sum_{i=1}^{M} a_i(\boldsymbol{x})\xi_i$:

$$\int_\Gamma \int_D \left(a_0(\boldsymbol{x}) + \sum_{i=1}^{M} a_i(\boldsymbol{x})\xi_i \right) \nabla u^{(sg)}(\boldsymbol{x}, \xi) \nabla v \, d\boldsymbol{x} \rho(\xi) d\xi$$

Substituting $u^{(sg)}$ and v:

$$\int_\Gamma \int_D \left(a_0(\boldsymbol{x}) + \sum_{i=1}^{M} a_i(\boldsymbol{x})\xi_i \right) \nabla \left(\sum_{s=1}^{n_\xi} \sum_{r=1}^{n_x} u_{rs} \phi_r(\boldsymbol{x}) \psi_s(\xi) \right) \nabla \phi_i(\boldsymbol{x}) \psi_j(\xi) \, d\boldsymbol{x} \rho(\xi) d\xi, \quad i = 1, \ldots, n_x, \; j = 1, \ldots, n_\xi$$

RHS:
Substituting v:

$$\int_\Gamma \int_D f \phi_i(\boldsymbol{x}) \psi_j(\xi) \, d\boldsymbol{x} \rho(\xi) d\xi, \quad i = 1, \ldots, n_x, \; j = 1, \ldots, n_\xi$$

Stochastic Galerkin systems in the Kronecker-product structure:

$$Au = f$$

$$A = G_0 \otimes K_0 + \sum_{l=1}^{M} G_l \otimes K_l, \quad f = g_0 \otimes f_0$$

"Stochastic" matrices:

$$[G_0]_{ij} = \langle \psi_i(\xi)\psi_j(\xi) \rangle_\rho, \; [G_l]_{ij} = \langle \xi_l \, \psi_i(\xi)\psi_j(\xi) \rangle_\rho, \; l = 1, \dots, M$$

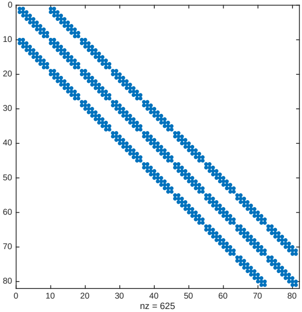
Weighted stiffness matrices:

$$[K_0]_{ij} = \int_D a_0 \nabla\phi_i(\mathbf{x})\nabla\phi_j(\mathbf{x}) d\mathbf{x},$$
$$[K_l]_{ij} = \int_D a_l(\mathbf{x}) \nabla\phi_i(\mathbf{x})\nabla\phi_j(\mathbf{x}) d\mathbf{x}, \quad l = 1, \dots, M$$

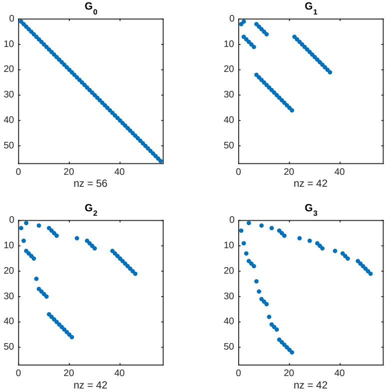
Forcing term:

$$[f_0]_i = \int_D f \phi_i(\mathbf{x}) d\mathbf{x}, [g_0]_i = \langle \psi_i(\xi) \rangle_\rho$$

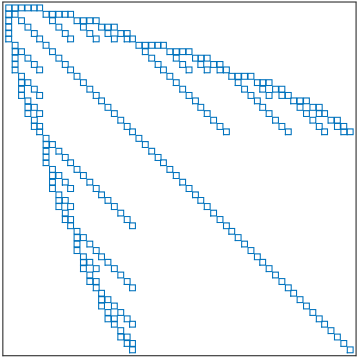
Nonzero structures of matrices:



Stiffness matrices $\{K_i\}_{i=0}^{M}$



Stochastic matrice $\{G_i\}_{i=1}^{3}$



Galerkin matrix A (each block has dimension $n_x \times n_x$)

Solutions in tensor format:

$$Au = f,$$

$$u = \sum_{k=1}^{\kappa_u} z_k \otimes y_k, \quad z_k \in \mathbb{R}^{n_\xi} \text{ and } y_k \in \mathbb{R}^{n_x},$$

or, equivalently,

$$U = \sum_{k=1}^{\kappa_u} y_k z_k^T = Y_{\kappa_u} Z_{\kappa_u}^T$$

where

$Y_{\kappa_u} = [y_1, \ldots, y_{\kappa_u}] \in \mathbb{R}^{n_x \times \kappa_u}, \quad Z_{\kappa_u} = [z_1, \ldots, z_{\kappa_u}] \in \mathbb{R}^{n_\xi \times \kappa_u}$

Isomorphism between $\mathbb{R}^{n_x \times n_\xi}$ and $\mathbb{R}^{n_x n_\xi}$ defined by two operators: $u = \text{vec}(U)$ and $U = \text{mat}(u)$

$$\begin{bmatrix} u_{11} \\ \vdots \\ u_{n_x 1} \\ u_{12} \\ \vdots \\ u_{n_x 2} \\ \vdots \\ u_{1 n_\xi} \\ \vdots \\ u_{n_x n_\xi} \end{bmatrix} \in \mathbb{R}^{n_x n_\xi} \quad \Leftrightarrow \quad \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1 n_\xi} \\ \vdots & \vdots & & \vdots \\ u_{n_x 1} & u_{n_x 2} & \cdots & u_{n_x n_\xi} \end{bmatrix} \in \mathbb{R}^{n_x \times n_\xi}$$

$$= \begin{bmatrix} | & | & & | \\ y_1 & y_2 & \cdots & y_{\kappa_u} \\ | & | & & | \end{bmatrix} \begin{bmatrix} - z_1^T - \\ - z_2^T - \\ \vdots \\ - z_{\kappa_u}^T - \end{bmatrix}$$

Linear systems in tensor format: $Au = f$

$$\left(\sum_{l=0}^{M} G_l \otimes K_l \right) \left(\sum_{k=1}^{\kappa_u} z_k \otimes y_k \right) = g_0 \otimes f_0$$

Linear systems in tensor format: $Au = f$

$$\left(\sum_{l=0}^{M} G_l \otimes K_l\right)\left(\sum_{k=1}^{\kappa_u} z_k \otimes y_k\right) = g_0 \otimes f_0$$

$$\sum_{l=0}^{M}\sum_{k=1}^{\kappa_u}(G_l z_k) \otimes (K_l y_k) = g_0 \otimes f_0$$

Linear systems in tensor format: $Au = f$

$$\left(\sum_{l=0}^{M} G_l \otimes K_l\right)\left(\sum_{k=1}^{\kappa_u} z_k \otimes y_k\right) = g_0 \otimes f_0$$

$$\sum_{l=0}^{M}\sum_{k=1}^{\kappa_u} (G_l z_k) \otimes (K_l y_k) = g_0 \otimes f_0$$

Matricizing

$$\sum_{l=0}^{M}\sum_{k=1}^{\kappa_u} (K_l y_k)(G_l z_k)^T = \sum_{l=0}^{M}\sum_{k=1}^{\kappa_u} K_l y_k z_k^T G_l^T = f_0 g_0^T$$

$$\sum_{l=0}^{M} K_l U G_l^T = \sum_{l=0}^{M} (K_l Y_{\kappa_u})(G_l Z_{\kappa_u})^T = f_0 g_0^T$$

Stochastic Galerkin system:

Exploiting the properties of the Kronecker product structure:

$$Au = \left(\sum_{l=0}^{M} G_l \otimes K_l\right)\left(\sum_{k=1}^{\kappa_u} z_k \otimes y_k\right) = \sum_{l=0}^{M}\sum_{k=1}^{\kappa_u} (G_l z_k) \otimes (K_l y_k)$$

- Operation counts of matrix operations are additive $O(nnz(G) + nnz(K))$ instead of $O(nnz(G) \times nnz(K))$
- This motivates the use of Krylov subspace methods in tensor format

Basic operations in tensor notation

Two essential operations for Krylov subspace methods: matrix-vector product and linear combination

- MVP: $\kappa_u \mapsto (M+1)\kappa_u$

$$Au = \sum_{l=0}^{M} \sum_{k=1}^{\kappa_u} (G_l z_k) \otimes (K_l y_k) = \sum_{i=1}^{(M+1)\kappa_u} \tilde{z}_i \otimes \tilde{y}_i$$

- Addition/subtraction: $(\kappa_u, \kappa_v) \mapsto (\kappa_u + \kappa_v)$

$$u + v = \sum_{i=1}^{\kappa_u} z_i \otimes y_i + \sum_{j=1}^{\kappa_v} \hat{z}_j \otimes \hat{y}_j = \sum_{i=1}^{\kappa_u+\kappa_v} z_i \otimes y_i$$

where $y_{i+\kappa_u} = \hat{y}_i$ and $z_{i+\kappa_u} = \hat{z}_i$, $i = 1, \ldots, \kappa_v$

Two of the fundamental operations used in Krylov subspace methods tend to increase the rank of the quantities produced

$$u \Rightarrow \tilde{u}$$

$$\text{Rank} : \kappa_u \Rightarrow \kappa_{\tilde{u}}$$

$$Y : \begin{bmatrix} | & | & & | \\ y_1 & y_2 & \cdots & y_{\kappa_u} \\ | & | & & | \end{bmatrix} \Rightarrow \begin{bmatrix} | & | & & | & & | \\ \tilde{y}_1 & \tilde{y}_2 & \cdots & \tilde{y}_{\kappa_u} & \cdots & \tilde{y}_{\kappa_{\tilde{u}}} \\ | & | & & | & & | \end{bmatrix}$$

$$Z : \begin{bmatrix} | & | & & | \\ z_1 & z_2 & \cdots & z_{\kappa_u} \\ | & | & & | \end{bmatrix} \Rightarrow \begin{bmatrix} | & | & & | & & | \\ \tilde{z}_1 & \tilde{z}_2 & \cdots & \tilde{z}_{\kappa_u} & \cdots & \tilde{z}_{\kappa_{\tilde{u}}} \\ | & | & & | & & | \end{bmatrix}$$

1. Introduction
   - Problem definition
2. Model problem
   - KL expansion
   - Stochastic Galerkin method
   - Tensor format
3. Preliminary work
   - Low-rank projection method in tensor format
   - Truncation methods
   - Numerical experiments
4. Proposed work
   - Nonlinear random fields
   - Truncation based on randomized tensor decomposition
   - Active subspace methods

The GMRES method

The generalized minimum residual method

Compute an approximate solution $u_m \in u_0 + \mathcal{K}_m(A, v_1)$ on mth *Krylov subspace*, $\mathcal{K}_m = \text{span}\{v_1, Av_1, \ldots, A^{m-1}v_1\}$
u_0 is an initial vector with residual $r_0 = f - Au_0$, $v_1 = r_0/\|r_0\|_2$

Algorithm 1 GMRES method without restarting

set the initial solution u_0
$r_0 := f - Au_0$
$\tilde{v}_1 := r_0$
$v_1 := \tilde{v}_1/\|\tilde{v}_1\|$
for $j = 1, \ldots, m$ **do**
  $w_j := Av_j$
  solve $(V_j^T V_j)\alpha = V_j^T w_j$
  $\tilde{v}_{j+1} := w_j - \sum_{i=1}^{j} \alpha_i v_i$
  $v_{j+1} := \tilde{v}_{j+1}/\|\tilde{v}_{j+1}\|$
end for
solve $(W_m^T A V_m)y = W_m^T r_0 \qquad (W_m = AV_m)$
$u_1 := u_0 + V_m y$

Low-rank projection method

Goal: compute a low-rank solution of rank κ satisfying $\|f - A\tilde{u}\|_2/\|f\|_2 < \epsilon$

Low-rank projection method

Goal: compute a low-rank solution of rank κ satisfying $\|f - A\tilde{u}\|_2 / \|f\|_2 < \epsilon$ (and maintain rank of all vectors to be κ)

Low-rank projection method

Goal: compute a low-rank solution of rank κ satisfying $\|f - A\tilde{u}\|_2/\|f\|_2 < \epsilon$ (and maintain rank of all vectors to be κ)

1. Construct a new basis vector $w_j = Av_j$

Low-rank projection method

Goal: compute a low-rank solution of rank κ satisfying $\|f - A\tilde{u}\|_2/\|f\|_2 < \epsilon$ (and maintain rank of all vectors to be κ)

1. Construct a new basis vector $w_j = Av_j$
2. Orthogonalize w_j with respect to the previously generated basis vectors $\{v_i\}_{i=1}^{j}$ (i.e., $\hat{w}_j = w_j - \sum_{i=1}^{j} \alpha_i v_i$ where $V_j^T V_j \alpha = V_j^T w_j$)

Low-rank projection method

Goal: compute a low-rank solution of rank κ satisfying $\|f - A\tilde{u}\|_2/\|f\|_2 < \epsilon$ (and maintain rank of all vectors to be κ)

1. Construct a new basis vector $w_j = Av_j$
2. Orthogonalize w_j with respect to the previously generated basis vectors $\{v_i\}_{i=1}^{j}$ (i.e., $\hat{w}_j = w_j - \sum_{i=1}^{j} \alpha_i v_i$ where $V_j^T V_j \alpha = V_j^T w_j$)
3. Truncate the new vector $\tilde{v}_{j+1} = \mathcal{T}_\kappa(\hat{w}_j)$ and orthonormalize $v_{j+1} = \tilde{v}_{j+1}/\|\tilde{v}_{j+1}\|_2$

Low-rank projection method

Goal: compute a low-rank solution of rank κ satisfying $\|f - A\tilde{u}\|_2/\|f\|_2 < \epsilon$ (and maintain rank of all vectors to be κ)

1. Construct a new basis vector $w_j = Av_j$
2. Orthogonalize w_j with respect to the previously generated basis vectors $\{v_i\}_{i=1}^{j}$ (i.e., $\hat{w}_j = w_j - \sum_{i=1}^{j} \alpha_i v_i$ where $V_j^T V_j \alpha = V_j^T w_j$)
3. Truncate the new vector $\tilde{v}_{j+1} = \mathcal{T}_\kappa(\hat{w}_j)$ and orthonormalize $v_{j+1} = \tilde{v}_{j+1}/\|\tilde{v}_{j+1}\|_2$
4. Compute the iterate ($\tilde{u}_1 = \tilde{u}_0 + V_m\beta$) by projecting the residual $r_0 = b - Au_0$ onto the subspace $\mathcal{W}_m = \text{span}\{w_1, \ldots, w_m\}$ $(W_m^T A V_m)\beta = W_m^T r_0$

Low-rank projection method

Goal: compute a low-rank solution of rank κ satisfying $\|f - A\tilde{u}\|_2/\|f\|_2 < \epsilon$ (and maintain rank of all vectors to be κ)

1. Construct a new basis vector $w_j = Av_j$
2. Orthogonalize w_j with respect to the previously generated basis vectors $\{v_i\}_{i=1}^{j}$ (i.e., $\hat{w}_j = w_j - \sum_{i=1}^{j} \alpha_i v_i$ where $V_j^T V_j \alpha = V_j^T w_j$)
3. Truncate the new vector $\tilde{v}_{j+1} = \mathcal{T}_\kappa(\hat{w}_j)$ and orthonormalize $v_{j+1} = \tilde{v}_{j+1}/\|\tilde{v}_{j+1}\|_2$
4. Compute the iterate ($\tilde{u}_1 = \tilde{u}_0 + V_m\beta$) by projecting the residual $r_0 = b - Au_0$ onto the subspace $\mathcal{W}_m = \text{span}\{w_1, \ldots, w_m\}$ $(W_m^T A V_m)\beta = W_m^T r_0$

- Truncation operator $\mathcal{T}_\kappa$ compresses a tensor of higher rank into one of a desired rank κ
- Due to truncation, $\mathcal{V}_m = \text{span}\{v_1, \ldots, v_m\}$ is not a Krylov subspace
- If κ is the full rank, the algorithm is the restarted GMRES method

Algorithm 2 Restarted low-rank projection method in tensor format

```
set the initial solution ũ_0
for k = 0, 1, ... do
    r_k := f − A ũ_k
    if ‖r_k‖/‖f‖ < ϵ then
        return ũ_k
    end if
    ṽ_1 := T_κ(r_k)
    v_1 := ṽ_1/‖ṽ_1‖
    for j = 1, ..., m do
        w_j := A v_j
        solve (V_j^T V_j) α = V_j^T w_j
        ṽ_{j+1} := T_κ( w_j − Σ_{i=1}^{j} α_i v_i )
        v_{j+1} := ṽ_{j+1}/‖ṽ_{j+1}‖
    end for
    solve (W_m^T A V_m) β = W_m^T r_k
    ũ_{k+1} := T_κ(ũ_k + V_m β)
end for
```

1. Introduction
    - Problem definition
2. Model problem
    - KL expansion
    - Stochastic Galerkin method
    - Tensor format
3. Preliminary work
    - Low-rank projection method in tensor format
    - Truncation methods
    - Numerical experiments
4. Proposed work
    - Nonlinear random fields
    - Truncation based on randomized tensor decomposition
    - Active subspace methods

Truncation of a tensor

Truncation operator:

$$\mathcal{T}_\kappa : \kappa' \mapsto \kappa$$

where $\kappa \ll \kappa'$

Truncation of a tensor

Truncation operator:

$$\mathcal{T}_\kappa : \kappa' \mapsto \kappa$$

where $\kappa \ll \kappa'$

$$\begin{bmatrix} | & & | \\ y_1 & \cdots & y_{\kappa'} \\ | & & | \end{bmatrix} \begin{bmatrix} - z_1^T - \\ \vdots \\ - z_{\kappa'}^T - \end{bmatrix} \approx \begin{bmatrix} | & & | \\ y_1 & \cdots & y_\kappa \\ | & & | \end{bmatrix} \begin{bmatrix} - z_1^T - \\ \vdots \\ - z_\kappa^T - \end{bmatrix}$$

Truncation based on singular values:
Given $U = Y_{\kappa'} Z_{\kappa'}^T$ of rank κ' where $Y_{\kappa'} \in \mathbb{R}^{n_x \times \kappa'}$ and $Z_{\kappa'} \in \mathbb{R}^{n_\xi \times \kappa'}$, compute the singular value decomposition (SVD) of U.

Truncation based on singular values:

Given $U = Y_{\kappa'} Z_{\kappa'}^T$ of rank κ' where $Y_{\kappa'} \in \mathbb{R}^{n_x \times \kappa'}$ and $Z_{\kappa'} \in \mathbb{R}^{n_\xi \times \kappa'}$, compute the singular value decomposition (SVD) of U.

An efficient way to compute the SVD of $U = Y_{\kappa'} Z_{\kappa'}^T$,

1. Compute QR factorizations of $Y_{\kappa'}$ and $Z_{\kappa'}$:
   $Y_{\kappa'} = Q_Y R_Y \in \mathbb{R}^{n_x \times \kappa'}$, $Z_{\kappa'} = Q_Z R_Z \in \mathbb{R}^{n_\xi \times \kappa'}$,
2. Compute the SVD of $R_Y R_Z^T$:
   $R_Y R_Z^T = \hat{U}_{\kappa'} \hat{\Sigma}_{\kappa'} \hat{V}_{\kappa'}^T = \sum_{k=1}^{\kappa'} \hat{\sigma}_k \hat{u}_k \hat{v}_k^T$,
3. Truncate the sum with the κ terms to produce $\tilde{Y}_\kappa$ and $\tilde{Z}_\kappa$,
   $\tilde{Y}_\kappa = Q_Y \hat{U}_\kappa \hat{\Sigma}_\kappa \in \mathbb{R}^{n_x \times \kappa}$, $\tilde{Z}_\kappa = Q_Z \hat{V}_\kappa \in \mathbb{R}^{n_\xi \times \kappa}$.

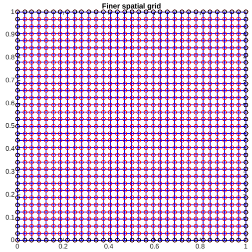
Computationally expensive and an appropriate κ is unknown

Truncation based on multilevel rank-reduction

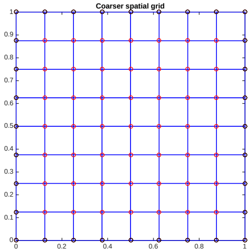
A coarse-spatial grid solution $u^c(\boldsymbol{x},\xi)$:

$$u^c(\boldsymbol{x},\xi) = \left(\Phi^c(\boldsymbol{x})\right)^T\ U^c\,\Psi(\xi) = \left((Y^c)^T\Phi^c(\boldsymbol{x})\right)^T\left((Z^c)^T\Psi(\xi)\right)$$

Recall that $u^{sg}(\boldsymbol{x},\xi) = \left(Y_{\kappa_u}^T\Phi(\boldsymbol{x})\right)^T\left(Z_{\kappa_u}^T\Psi(\xi)\right)$



Finer spatial grid



Coarser spatial grid

Truncation based on multilevel rank-reduction

Mutlilevel rank-reduction strategy

- Define a truncation operator based on the information obtained from a coarse spatial grid computation: Given $U = Y_{\kappa'} Z_{\kappa'}^T$ of rank κ',

$$\mathcal{T}_\kappa(U) \equiv \left(Y_{\kappa'} Z_{\kappa'}^T Z_\kappa^c\right)\left(Z_\kappa^c\right)^T = \tilde{U}$$

where $\tilde{U} = \tilde{Y}_\kappa \tilde{Z}_\kappa^T$, $\tilde{Y}_\kappa = Y_{\kappa'} Z_{\kappa'}^T Z_\kappa^c \in \mathbb{R}^{n_x \times \kappa}$ and $\tilde{Z}_\kappa = Z_\kappa^c \in \mathbb{R}^{n_\xi \times \kappa}$

Truncation based on multilevel rank-reduction

Mutlilevel rank-reduction strategy

- Define a truncation operator based on the information obtained from a coarse spatial grid computation: Given $U = Y_{\kappa'} Z_{\kappa'}^T$ of rank κ',

$$\mathcal{T}_\kappa(U) \equiv \left(Y_{\kappa'} Z_{\kappa'}^T Z_\kappa^c\right) \left(Z_\kappa^c\right)^T = \tilde{U}$$

  where $\tilde{U} = \tilde{Y}_\kappa \tilde{Z}_\kappa^T$, $\tilde{Y}_\kappa = Y_{\kappa'} Z_{\kappa'}^T Z_\kappa^c \in \mathbb{R}^{n_x \times \kappa}$ and $\tilde{Z}_\kappa = Z_\kappa^c \in \mathbb{R}^{n_\xi \times \kappa}$

- Identify a desired rank κ s.t.

$$\|f^c - A^c u^{c,\kappa}\|_2 / \|f^c\|_2 \leq \epsilon$$

  where $u^{c,\kappa}$ is a κ-term approximation to u^c
- The κ-term approximation on a coarse spatial grid can be computed efficiently using the Proper Generalized Decomposition method

Preconditioning

Preconditioned system:

$$AM^{-1}\hat{u} = f, \quad \hat{u} = M\tilde{u}$$

- Mean-based preconditioner: $M = G_0 \otimes K_0$
- Preconditioned system in tensor notation:

$$AM^{-1}\hat{u} = \sum_{l=0}^{M} \sum_{k=1}^{\kappa_{\hat{u}}} G_l \hat{z}_k \otimes K_l K_0^{-1} \hat{y}_k$$

- Practical application of the preconditioner: the action of K_0^{-1} is replaced by an application of a single V-cycle of an algebraic multigrid method

With right preconditioning and the AMG preconditioner, the strategy for handling tensor rank is largely unaffected by preconditioning

Algorithm 3 Preconditioned low-rank projection method with the multi-level rank-reduction

1: Compute $u^{c,\kappa}$ which satisfies $\frac{\|f^c - A^c u^{c,\kappa}\|_2}{\|f^c\|_2} < \epsilon$ using the PGD method
2: Compute Z^c_κ such that $U^{c,\kappa} = Y^c_\kappa (Z^c_\kappa)^T$ and define $\mathcal{T}_\kappa(U) \equiv (UZ^c_\kappa)(Z^c_\kappa)^T$
3: Run Algorithm 2 with $\mathcal{L} = AM^{-1}$, f, and $\mathcal{T}_\kappa$

1. Introduction
   - Problem definition
2. Model problem
   - KL expansion
   - Stochastic Galerkin method
   - Tensor format
3. Preliminary work
   - Low-rank projection method in tensor format
   - Truncation methods
   - Numerical experiments
4. Proposed work
   - Nonlinear random fields
   - Truncation based on randomized tensor decomposition
   - Active subspace methods

Stochastic diffusion problems

Steady-state diffusion problems with homogeneous Dirichlet boundary condition:

$$\begin{cases} -\nabla \cdot (a(\boldsymbol{x}, \xi)\nabla u(\boldsymbol{x}, \xi)) & = f(\boldsymbol{x}, \xi) \quad \text{in} \quad D \times \xi, \\ u(\boldsymbol{x}, \xi) & = 0 \quad \text{on} \quad \partial D \times \Gamma, \end{cases}$$

with $f(\boldsymbol{x}, \xi) = 1$.

Covariance function of $a(\boldsymbol{x}, \xi)$:

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

The truncated KL-expansion:

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

$\mu = 1$, $\sigma = 0.05$, and M is chosen to capture 95% of the total variance of the random field (i.e., $\sum_{i=1}^{M} \lambda_i / \sum_{i=1}^{n_x} \lambda_i > 95\%$)

Coarse spatial grid computation:

Table: Rank (κ) of coarse-grid solutions satisfying a specified tolerance ϵ for the PGD computation, and for varying c and M

	$\epsilon = 10^{-5}$				$\epsilon = 10^{-6}$			
c	4	3	2.5	2	4	3	2.5	2
M, n_ξ	5, 56	7, 120	10, 286	15, 816	5, 56	7, 120	10, 286	15, 816
n_x^c	15^2	15^2	31^2	31^2	15^2	15^2	31^2	31^2
Rank(κ)	25	40	65	115	35	65	100	210

n_ξ: DoFs of stochastic domain
n_x^c: DoFs of coarse spatial domain

Fine spatial grid computation:

Table: CPU time to compute approximate solutions satisfying $\epsilon = 10^{-5}$, 10^{-6} using the preconditioned low-rank projection method with the multilevel rank-reduction. Here, t_f is the time to compute the fine-grid solution, t_f, and, t is the total time, $t = t_f + t_c$

n_x		$\epsilon = 10^{-5}$				$\epsilon = 10^{-6}$			
	M	5	7	10	15	5	7	10	15
129^2	t_f	5.87	8.96	20.53	87.07	7.21	14.28	36.85	235.34
	t	8.35	12.43	28.88	132.15	10.14	19.32	51.69	398.06
257^2	t_f	22.69	34.90	84.85	340.51	27.61	56.36	148.07	1014.97
	t	25.17	38.37	93.20	385.59	30.55	61.41	162.90	1177.68
513^2	t_f	144.69	194.41	445.36	2809.54	163.31	310.14	1318.79	OoM
	t	147.17	197.87	453.71	2854.62	166.24	315.18	1333.63	OoM

n_x: DoFs of fine spatial domain

Comparison to a truncation operator based on singular values:

Table: CPU time to compute approximate solutions satisfying $\epsilon = 10^{-5}, 10^{-6}$ using the preconditioned low-rank projection (LRP) methods with the multilevel rank-reduction and the singular value based truncation on the level 8 spatial grid ($n_x = 257^2$)

	Solver	M	5	7	10	15	20
$\epsilon = 10^{-5}$	LRP-SVD	t	55.04	108.11	284.27	1280.65	5691.19
	LRP-Multilevel	t	25.17	38.37	93.20	385.59	1943.49
$\epsilon = 10^{-6}$	LRP-SVD	t	76.03	198.20	564.12	5131.32	OoM
	LRP-Multilevel	t	30.55	61.41	162.90	1177.68	OoM

PGD as a solver on a finer spatial grid:

Table: Computation time to obtain approximate solutions satisfying $\epsilon = 10^{-5}$ using the PGD method and the preconditioned low-rank projection method on the level 8 spatial grid ($n_x = 257^2$)

	Solver	M	5	7	10	15	20
$\epsilon = 10^{-5}$	PGD	κ	25	45	65	125	195
		t	43.78	109.72	228.73	940.69	3066.87
	LRP-Multilevel	κ	25	40	65	115	180
		t	25.17	38.37	93.20	385.59	1943.49
$\epsilon = 10^{-6}$	PGD	κ	40	70	110	225	OoM
		t	74.43	214.82	533.10	2713.70	OoM
	LRP-Multilevel	κ	35	65	100	210	OoM
		t	30.55	61.41	162.90	1177.68	OoM

Stochastic convection-diffusion problems

Steady-state convection-diffusion problems with non-homogeneous boundary condition:

$$\begin{cases} \nu\nabla\cdot(a(\mathbf{x},\xi)\nabla u(\mathbf{x},\xi)) + \vec{w}\cdot\nabla u(\mathbf{x},\xi) &= f(\mathbf{x},\xi) \quad \text{in } D\times\Gamma, \\ u(\mathbf{x},\xi) &= g_D(\mathbf{x}) \quad \text{on } \partial D\times\Gamma, \end{cases}$$

where $g_D(\mathbf{x})$ is determined by

$$g_D(\mathbf{x}) = \begin{cases} g_D(x,-1) = x, & g_D(x,1) = 0, \\ g_D(-1,y) \approx -1, & g_D(1,y) \approx 1, \end{cases}$$

where the latter two approximations hold except near $y = 1$, and ν is the viscosity parameter.

The solution has *exponential boundary layer* near $y = 1$

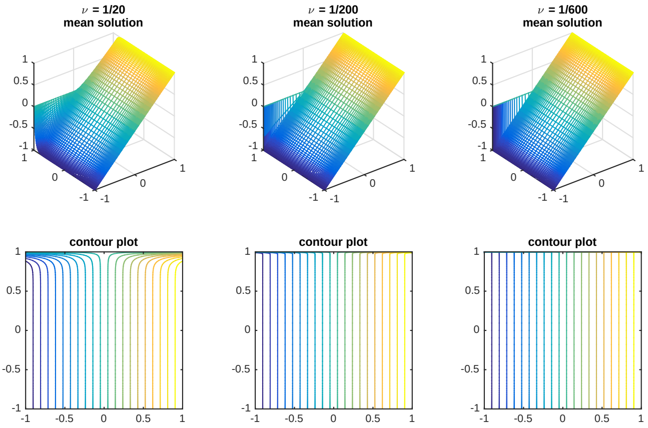


Figure: Mean solutions (top) and their contour plots (bottom) for varying ν

Stochastic Galerkin system:

$$\left(G_0 \otimes \nu K_0 + \sum_{l=1}^{M} G_l \otimes \nu K_l + G_0 \otimes N + G_0 \otimes S \right) u = g_0 \otimes f_0$$

- the convection term N: $[N]_{ij} = \int_D \vec{w} \cdot \nabla \phi_i(\boldsymbol{x}) \phi_j(\boldsymbol{x}) d\boldsymbol{x}$
- the streamline-diffusion term S: $[S]_{ij} = \sum_{l=1}^{n_e} \delta_l \int_D (\vec{w} \cdot \nabla \phi_i)(\vec{w} \cdot \nabla \phi_j) d\boldsymbol{x}$

  where n_e: the number of element in the finite element discretization and $\delta_k = \frac{h_k}{2\|\vec{w}\|_2}\left(1 - \frac{1}{\mathcal{P}_k}\right)$ if $\mathcal{P}_k > 1$

Preconditioned system:

Mean-based preconditioner: $M = G_0 \otimes (K_0 + N + S)$

the action of $(K_0 + N + S)$ is replaced by an application of a single V-cycle of an algebraic multigrid method

Comparison to a truncation operator based on singular values:

Table: CPU time to compute approximate solutions satisfying $\epsilon = 10^{-5}, 10^{-6}$ using the preconditioned low-rank projection (LRP) methods with the multilevel rank-reduction and the singular value based truncation on the level 8 spatial grid ($n_x = 257^2$)

			$\nu = 1/600$				$\nu = 1/20$			
ϵ	Solver	M	5	7	10	15	5	7	10	15
10^{-5}	LRP-SVD	t	90.33	103.44	218.35	484.08	68.45	100.83	201.34	448.25
	LRP-Multilevel	t	65.48	73.28	142.46	321.99	51.50	67.24	128.45	291.46
10^{-6}	LRP-SVD	t	122.44	231.07	421.76	1208.88	132.08	234.15	570.56	2055.44
	LRP-Multilevel	t	81.93	107.84	186.56	530.89	83.43	136.69	341.32	1266.53

Table: Computation time and the number of cycles (k) to compute approximate solutions with $\epsilon = 10^{-5}$ and 10^{-6} using the preconditioned low-rank projection methods with the multilevel rank-reduction method for varying ν

		$\epsilon = 10^{-5}$			$\epsilon = 10^{-6}$		
ν	n_x^c	$M = 5$	$M = 7$	$M = 10$	$M = 5$	$M = 7$	$M = 10$
$\frac{1}{20}$	17^2	25	35	55	35	50	75
$\frac{1}{100}$	17^2	20	25	45	30	40	65
$\frac{1}{200}$	33^2	20	25	45	25	40	60
$\frac{1}{400}$	33^2	20	20	35	25	35	55
$\frac{1}{600}$	65^2	20	20	35	30	35	45

n_x^c: DoFs of coarse spatial grid

1 Introduction
- Problem definition

2 Model problem
- KL expansion
- Stochastic Galerkin method
- Tensor format

3 Preliminary work
- Low-rank projection method in tensor format
- Truncation methods
- Numerical experiments

4 Proposed work
- Nonlinear random fields
- Truncation based on randomized tensor decomposition
- Active subspace methods

Nonlinear random fields

Log-normal random field:

$$a(\boldsymbol{x}, \xi) = e^{g(\boldsymbol{x}, \xi)}, \quad g(\boldsymbol{x}, \xi) = g_0(\boldsymbol{x}, \xi) + \sum_{i=1}^{M} g_i(\boldsymbol{x})\xi_i$$

where $g(\boldsymbol{x}, \xi)$ is a truncated KL expansion and $\{\xi_i\}_{i=1}^{M}$ are independent normal random variables

Polynomial approximation of a random field:

$$a(\boldsymbol{x}, \xi) = \sum_{\alpha \in \Lambda} a_\alpha(\boldsymbol{x})\psi_\alpha(\xi)$$

where Λ is a multi-index set

Stochastic Galerkin system

Stochastic Galerkin solution: $u^{(sg)}(\boldsymbol{x}, \xi) = \sum_{s=1}^{n_\xi} \sum_{r=1}^{n_x} u_{rs}\phi_r(\boldsymbol{x})\psi_s(\xi)$

Stochastic weak formulation:

$$\int_\Gamma \int_D \left(\sum_{\alpha \in \Lambda} a_\alpha(\boldsymbol{x})\psi_\alpha(\xi) \right) \nabla \left(\sum_{s=1}^{n_\xi} \sum_{r=1}^{n_x} u_{rs}\phi_r(\boldsymbol{x})\psi_s(\xi) \right) \nabla\phi_i(\boldsymbol{x})\psi_j(\xi)\, d\boldsymbol{x}\rho(\xi)d\xi,$$

$$i = 1, \ldots, n_x,\ j = 1, \ldots, n_\xi$$

Stiffness matrices and "stochastic matrices":

$$[K_l]_{ij} = \int_D a_l\phi_i\phi_j d\boldsymbol{x}, \qquad [G_l]_{ij} = \langle \psi_l\psi_i\psi_j \rangle_\rho.$$

Total degree space for approximating $u^{(sg)}(\boldsymbol{x}, \xi)$:

$$\Lambda^{M,p} = \{\alpha(s) \in \mathbb{N}_0^M : \|\alpha(s)\|_0 \leq M, \ \|\alpha(s)\|_1 \leq p\}$$

Polynomial expansion is implicitly truncated with polynomials of total degree $\leq 2p$:

$$a(\boldsymbol{x}, \xi) \approx \sum_{\alpha \in \Lambda^{M,2p}} a_\alpha(\boldsymbol{x}) \psi_\alpha(\xi)$$

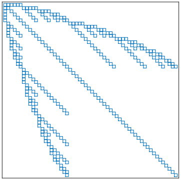
because

$$\langle \psi_l \psi_i \psi_j \rangle_\rho = 0 \quad \forall\, i, j \ \text{ s.t. } \ \alpha(i), \alpha(j) \in \Lambda^{M,p} \quad \text{if} \quad \sum_k \alpha_k(l) > 2p,$$

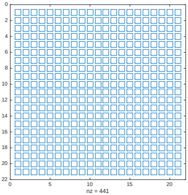
- $n_a = \dim(\Lambda^{M,2p}) = \frac{(M+2p)!}{M!(2p)!}$
- Special case: $n_a = M + 1$

$$a(\boldsymbol{x}, \xi) \approx \sum_{k=0}^{M} a_k(\boldsymbol{x}) \xi_k$$

Block sparse linear system, A:

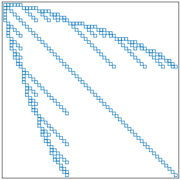


linear RF

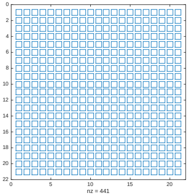


nonlinear RF

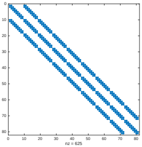
Block sparse linear system, A:



linear RF



nonlinear RF



Each block is sparse

Two potential directions of the study:

- Use a randomized tensor Interpolative Decomposition as a truncation operator
  - to replace costly SVD
  - when efficient coarse grid computation is impossible
  - ID provides means to approximate a matrix/ tensor in efficient way
- Reduce dimensions of problem using active subspace methods
  - the number of terms in the polynomial expansion of $a(\boldsymbol{x}, \xi)$ is large
  - Active subspace methods represents a dimension-reduction method that can be used to reduce the number of terms in the expansion

Matrix Interpolative Decomposition

Matrix approximation by a column subset:

$$A \approx A_{CS}[I|T]P^T$$

- G matrices are rank-deficient
- $w = Av$ where $\mathrm{mat}(v) = Y_v Z_v^T$, $\mathrm{mat}(w) = Y_w Z_w^T$

Matrix Interpolative Decomposition

Matrix approximation by a column subset:

$$A \approx A_{CS}[I|T]P^T$$

G matrices are rank-deficient

$w = Av$ where $\text{mat}(v) = Y_v Z_v^T$, $\text{mat}(w) = Y_w Z_w^T$

$Y_v \in \mathbb{R}^{n_x \times \kappa}$, $Z_v \in \mathbb{R}^{n_\xi \times \kappa}$, $Y_w \in \mathbb{R}^{n_x \times \kappa'}$, and $Z_w \in \mathbb{R}^{n_\xi \times \kappa'}$

- $W = \text{mat}(w)$ is rank deficient
- $Z_w = \left[G_0 Z_v \mid G_1 Z_v \mid \cdots \mid G_{n_a} Z_v\right]$, each block is rank deficient
  In tensor format, $y \otimes x + z \otimes x = (y + z) \otimes x$

Interpolative decomposition:

$$W \approx W_{CS}[I|T]P^T$$

$$(G_i Z_v) \approx G_{CS}[I|T]P^T$$

Randomization makes the computation efficient without losing too much accuracy

Active subpace methods

Applying active subspace methods on a nonlinear random field:

$$a(\boldsymbol{x},\,\xi) = \exp\left(g_0(\boldsymbol{x},\,\xi) + \sum_{k=1}^{M} g_k(\boldsymbol{x})\xi_k\right) \approx \sum_{\alpha\in\Lambda^{M,\,2p}} a_\alpha(\boldsymbol{x})\psi_\alpha(\xi)$$

The gradient of $a(\boldsymbol{x},\,\xi)$:

$$\nabla_\xi a(\boldsymbol{x},\xi) = \left[\frac{\partial a}{\partial \xi_1},\,\ldots,\,\frac{\partial a}{\partial \xi_M}\right]^T$$

where

$$\frac{\partial a}{\partial \xi_i} = \exp\left(g_0(\boldsymbol{x},\,\xi) + \sum_{k=1}^{M} g_k(\boldsymbol{x})\xi_k\right) g_i(\boldsymbol{x}).$$

The covariance matrix $C(\boldsymbol{x})$:

$$[C(\boldsymbol{x})]_{ij} = \mathbb{E}\left[\frac{\partial a}{\partial \xi_i}\frac{\partial a}{\partial \xi_j}\right] = \exp\left(2g_0(\boldsymbol{x},\,\xi) + \sum_{k=1}^{M} 2g_k^2(\boldsymbol{x})\right) g_i(\boldsymbol{x})g_j(\boldsymbol{x}).$$

An eigendecomposition of $C_i = C(x_i) \in \mathbb{R}^{M \times M}$:

$$C_i = W_i \Lambda_i W_i^T$$

A new set of random variable $\{\eta_j^{(i)}\}_{j=1}^{M}$:

$$\eta^{(i)} = W_i^T \xi \Leftrightarrow W_i \eta^{(i)} = W_i W_i^T \xi = \xi,$$

and $\{\eta^{(i)}\}$ are also independent normal random variables:

$$\mathbb{E}\left[\eta_j^{(i)}\right] = \mathbb{E}\left[\boldsymbol{w}_j^T \xi\right] = 0,$$

$$\mathbb{E}\left[(\eta_j^{(i)})^2\right] = \mathbb{E}\left[(\boldsymbol{w}_j^T \xi)^2\right] = \|\boldsymbol{w}_j\|_2^2 = 1,$$

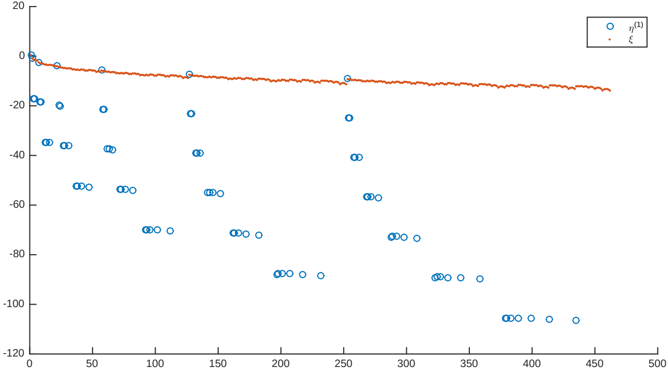
$$\mathbb{E}\left[\left(\eta_j^{(i)}\right)\left(\eta_j^{(i)}\right)^T\right] = \mathbb{E}\left[W_i^T \xi \xi^T W_i\right] = W_i^T \mathbb{E}\left[\xi \xi^T\right] W_i = I.$$

Change of variable:

$$
\begin{aligned}
a(x_i, \xi) &= \exp\left(g_0(x_i) + \sum_{k=1}^{M} g_k(x_i)\xi_k \right) \\
&= \exp\left(g_0(x_i) + \sum_{k=1}^{M} g_k(x_i) \left(\sum_{l=1}^{M} w_{kl}\eta_l^{(i)} \right) \right) \\
&= \exp\left(g_0(x_i) + \sum_{k=1}^{M} \tilde{g}_k(x_i)\eta_k^{(i)} \right), \quad \left(\tilde{g}_k(x_i) = \sum_{l=1}^{M} g_l(x_i) w_{ik} \right), \\
&= a(x_i, \eta^{(i)})
\end{aligned}
$$

A new linear expansion for $a(\boldsymbol{x}, \xi)$:

$$
a(x_i, \xi) = \hat{a}(x_i, \eta^{(i)}) = \sum_{\alpha \in \Lambda^{M,\,2p}} \hat{a}_\alpha(x_i)\psi_\alpha(\eta^{(i)})
$$



At spatial point x_1, $\log_{10}(|a_\alpha(x_1)|)$ v.s. $\log_{10}(|\hat{a}_\alpha(x_1)|)$

Thank you!

Coarse Spatial Grid Computation

The Proper Generalized Decomposition method:

- Computes the separated representation of a coarse-grid solution: $u^{c,\kappa}(\boldsymbol{x}, \xi) = \sum_{i=1}^{\kappa} \tilde{y}_i(\boldsymbol{x})\tilde{z}_i(\xi)$
  - Discretization in physical space: $\tilde{y}_i(\boldsymbol{x}) = \sum_{k=1}^{n_x} \tilde{y}_k^{(i)} \phi_k^c(\boldsymbol{x})$
  - Discretization in stochastic space: $\tilde{z}_i(\xi) = \sum_{l=1}^{n_\xi} \tilde{z}_l^{(i)} \psi_l(\xi)$

Coarse Spatial Grid Computation

The Proper Generalized Decomposition method:

- Computes the separated representation of a coarse-grid solution: $u^{c,\kappa}(\boldsymbol{x},\xi) = \sum_{i=1}^{\kappa} \tilde{y}_i(\boldsymbol{x})\tilde{z}_i(\xi)$
  - Discretization in physical space: $\tilde{y}_i(\boldsymbol{x}) = \sum_{k=1}^{n_x} \tilde{y}_k^{(i)} \phi_k^c(\boldsymbol{x})$
  - Discretization in stochastic space: $\tilde{z}_i(\xi) = \sum_{l=1}^{n_\xi} \tilde{z}_l^{(i)} \psi_l(\xi)$
- Identifies the function pairs $(\tilde{y}_i(\boldsymbol{x}), \tilde{z}_i(\xi))$ incrementally until the relative residual of the computed solution satisfies a given tolerance, $\|f^c - A^c u^{c,\kappa}\|_2 / \|f^c\|_2 < \epsilon$

Coarse Spatial Grid Computation

The Proper Generalized Decomposition method:

- Computes the separated representation of a coarse-grid solution: $u^{c,\kappa}(\boldsymbol{x},\xi) = \sum_{i=1}^{\kappa} \tilde{y}_i(\boldsymbol{x})\tilde{z}_i(\xi)$
  - Discretization in physical space: $\tilde{y}_i(\boldsymbol{x}) = \sum_{k=1}^{n_x} \tilde{y}_k^{(i)} \phi_k^c(\boldsymbol{x})$
  - Discretization in stochastic space: $\tilde{z}_i(\xi) = \sum_{l=1}^{n_\xi} \tilde{z}_l^{(i)} \psi_l(\xi)$
- Identifies the function pairs $(\tilde{y}_i(\boldsymbol{x}), \tilde{z}_i(\xi))$ incrementally until the relative residual of the computed solution satisfies a given tolerance, $\|f^c - A^c u^{c,\kappa}\|_2 / \|f^c\|_2 < \epsilon$
- Once i such pairs have been computed, $(\tilde{y}_{i+1}, \tilde{z}_{i+1})$ is sought in $X_h \times S_M$ by imposing Galerkin orthogonality with respect to the tangent manifold of the set of rank-one elements at $\tilde{y}_{i+1}\tilde{z}_{i+1}$, which is $\{\tilde{y}_{i+1}\zeta + \upsilon\tilde{z}_{i+1};\ \upsilon \in X_h,\ \zeta \in S_M\}$: find $\tilde{y}_{i+1}\tilde{z}_{i+1}$ s.t.

$$\left\langle \int_D a(\boldsymbol{x},\xi)\nabla(u^{c,i} + \tilde{y}_{i+1}\tilde{z}_{i+1}) \cdot \nabla(\tilde{y}_{i+1}\zeta + \upsilon\tilde{z}_{i+1}) \right\rangle = \left\langle \int_D f(\tilde{y}_{i+1}\zeta + \upsilon\tilde{z}_{i+1}) \right\rangle, \quad \forall(\upsilon,\zeta) \in X_h \times S_M$$

- Two coupled problems: a deterministic problem and a stochastic problem
  - **Deterministic problem:** given $\tilde{z}_{i+1}$, find $\tilde{y}_{i+1} \in X_h$ s.t.

$$\left\langle \int_D a(\mathbf{x}, \xi) \nabla (u^{c,i} + \tilde{y}_{i+1}\tilde{z}_{i+1}) \cdot \nabla (\phi_j^c \tilde{z}_{i+1}) \right\rangle = \left\langle \int_D f \phi_j^c \tilde{z}_{i+1} \right\rangle, \, j = 1, \ldots, n_x^c$$

  - **Stochastic problem:** given $\tilde{y}_{i+1}$, finds $\tilde{z}_{i+1} \in S_M$ s.t.

$$\left\langle \int_D a(\mathbf{x}, \xi) \nabla (u^{c,i} + \tilde{y}_{i+1}\tilde{z}_{i+1}) \cdot \nabla (\tilde{y}_{i+1} \psi_j) \right\rangle = \left\langle \int_D f \tilde{y}_{i+1} \psi_j \right\rangle, \, j = 1, \ldots, n_\xi$$

- Enhances accuracy of the κ-term approximation by solving a set of κ coupled equations
  - **Update problem:** given $\{\tilde{y}_i\}_{i=1}^{\kappa}$, find $\{\tilde{z}_i\}_{i=1}^{\kappa}$ s.t.

$$\left\langle \int_D a(\mathbf{x}, \xi) \nabla (u^{(\kappa)}) \cdot \nabla (\tilde{y}_i \psi_j) \right\rangle = \left\langle \int_D f \tilde{y}_i \psi_j \right\rangle, \, i = 1, \ldots, \kappa, \, j = 1, \ldots, n_\xi.$$

Matrix Interpolative Decomposition

QR factorization:

$$
\begin{aligned}
AP = QR &= \left[\begin{array}{c|c} Q_{11} & Q_{12} \\ \hline Q_{21} & Q_{22} \end{array}\right] \left[\begin{array}{c|c} R_{11} & R_{12} \\ \hline 0 & R_{22} \end{array}\right] \\
&= \left[\begin{array}{c} Q_{11} \\ \hline Q_{21} \end{array}\right] [R_{11} | R_{12}] + \left[\begin{array}{c} Q_{12} \\ \hline Q_{22} \end{array}\right] [0 | R_{22}] \\
&= \left[\begin{array}{c} Q_{11} R_{11} \\ \hline Q_{21} R_{11} \end{array}\right] [I | R_{11}^{-1} R_{12}] + \left[\begin{array}{c|c} 0 & Q_{12} R_{22} \\ \hline 0 & Q_{22} R_{22} \end{array}\right] \\
&= A_{CS} [I | T] + XP
\end{aligned}
$$

where

$$
A_{CS} = \left[\begin{array}{c} Q_{11} R_{11} \\ \hline Q_{21} R_{11} \end{array}\right], \quad T = R_{11}^{-1} R_{12}, \quad X = \left[\begin{array}{c|c} 0 & Q_{12} R_{22} \\ \hline 0 & Q_{22} R_{22} \end{array}\right] P^T
$$

$$A \approx \hat{A} = A_{\mathrm{CS}}[I \,|\, T]P^T,$$
$$\|A - \hat{A}\|_2 = \|X\|_2 \leq \sigma_{k+1}(A)\sqrt{1 + k(n-k)}.$$

$$R = \left[\begin{array}{c|c} R_{11} & R_{12} \\ \hline 0 & R_{22} \end{array}\right]$$

$$\sigma_1(R_{22}) \leq \sigma_{k+1}(A)\sqrt{1 + k(n-k)}.$$

Randomized Matrix Interpolative Decomposition

Randomized scheme for approximating the range $A \in \mathbb{R}^{m \times n}$:

- Draw an $n \times l$ Gaussian random matrix Ω
- Form the matrix product $Y = A\Omega$
- Construct a matrix Q whose columns form an orthonormal basis for the range of Y

Intuition:

$$y^{(i)} = A\omega^{(i)}, \quad i = 1, 2, \dots, k$$

Consider $A = B + E$, where B is a rank-k matrix and E is a small perturbation, then

$$y^{(i)} = (B + E)\omega^{(i)} = B\omega^{(i)} + E\omega^{(i)}, \quad i = 1, 2, \dots, k + p$$

where p is a small number

Randomized Matrix Interpolative Decomposition

Algorithm 4 Randomized matrix interpolative decomposition

Input: An $m \times n$ matrix A and integer $l > k$
Output: Indices set $\mathcal{L}_k$ of the k columns, the permutation matrix P, and the column subset matrix A_{CS}

1: Draw an $n \times l$ Gaussian random matrix, Ω.
2: Form $m \times l$ matrix $Y = A\Omega$.
3: Construct an $m \times k$ orthonormal matrix Q for approximate the range of Y via the QR factorization, $Y = QR$.
4: Construct $\mathcal{L}_k$, P, and A_{CS} from the QR of Y.

Randomized Tensor Interpolative Decomposition

Tensor notation:

$$\mathcal{U} = \sum_{l=1}^{\kappa_{\mathcal{U}}} \bigotimes_{j=1}^{d} \boldsymbol{u}_j^{(l)} = \sum_{l=1}^{\kappa_{\mathcal{U}}} \mathcal{U}^{(l)}, \quad \mathcal{U}^{(l)} = \bigotimes_{j=1}^{d} \boldsymbol{u}_j^{(l)}.$$

where $\boldsymbol{u}^j \in \mathbb{R}^{M_j}$ for $j = 1, \ldots, d$ and $\kappa_{\mathcal{U}}$ is the rank

A matricized tensor U:

$$U = \begin{bmatrix} | & & | \\ \mathcal{U}^{(1)} & \ldots & \mathcal{U}^{(\kappa_{\mathcal{U}})} \\ | & & | \end{bmatrix}$$

The inner product between two tensors $\mathcal{U}$ and $\mathcal{V}$:

$$\langle \mathcal{U}, \mathcal{V} \rangle = \sum_{l=1}^{\kappa_{\mathcal{U}}} \sum_{m=1}^{\kappa_{\mathcal{V}}} \prod_{j=1}^{d} \langle \boldsymbol{u}_j^{(l)}, \boldsymbol{v}_j^{(m)} \rangle$$

Randomized Tensor Interpolative Decomposition

Algorithm 5 Randomized tensor interpolative decomposition

Input: A rank κ' tensor $\mathcal{U}$ and integer $l > \kappa$
Output: A rank κ tensor $\tilde{\mathcal{U}}$

1: Draw a random tensor $\mathcal{R}$ of rank l.
2: Form $l \times \kappa'$ matrix $Y = \mathcal{R}^T \mathcal{U}$.
3: Compute an ID of Y and, as a result, a $\kappa \times \kappa'$ permutation matrix P, and a column index set $\mathcal{L}_k$.
4: Compute $\tilde{\mathcal{U}}^{(l)} = \sum_{m=1}^{\kappa'} P_{ml} \mathcal{U}^{(l_m)}$ where $l_m \in \mathcal{L}_k$

Active subspace methods

A general multivariate function $f : \mathbb{R}^m \mapsto \mathbb{R}$

$$f = f(\xi), \quad \xi \in \mathbb{R}^m$$

Active subspace method

1. Compute the gradient of f:

$$\nabla_\xi f(\xi) = \left[\frac{\partial f}{\partial \xi_1}, \ldots, \frac{\partial f}{\partial \xi_m}\right]^T$$

2. Construct a covariance matrix C:

$$C = \mathbb{E}\left[(\nabla_\xi f(\xi))(\nabla_\xi f(\xi))^T\right]$$

3. Compute an eigendecomposition of C:

$$C = W \Lambda W^T, \quad \Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_m),$$

   where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$

4. Partition W and Λ:

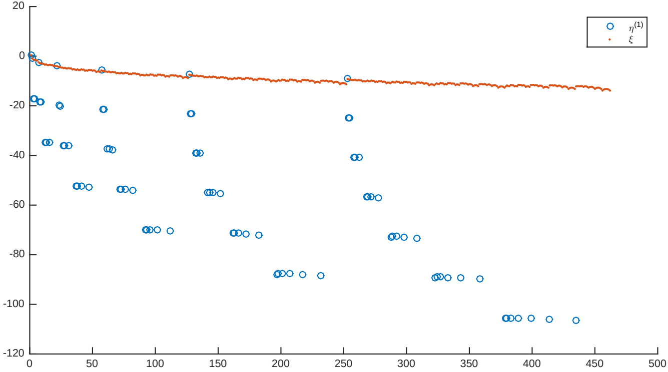
$$\Lambda = \begin{bmatrix} \Lambda_1 & \\ & \Lambda_2 \end{bmatrix}, \quad W = [W_1, \, W_2]$$

   where $\Lambda_1 = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$ with $n < m$ and $W_1 \in \mathbb{R}^{m \times n}$

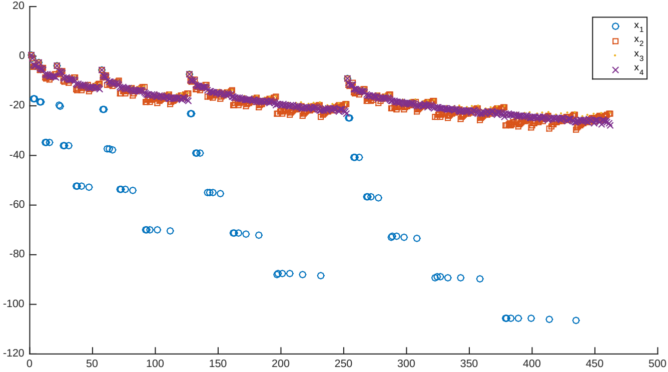
5. Rotate ξ:

$$y = W_1^T \xi, \quad z = W_2^T \xi$$

   where $y \in \mathbb{R}^n$ and $z \in \mathbb{R}^{m-n}$



At spatial point x_1, $\log_{10}(|a_\alpha(x_1)|)$ v.s. $\log_{10}(|\hat{a}_\alpha(x_1)|)$



At spatial points x_i, $i = 1, 2, 3, 4$, $\log_{10}(|\hat{a}_\alpha(x_i, \eta^{(1)})|)$

Group n_{set} sets of spatial points

Construct the Jacobian of $a(\boldsymbol{x}, \xi)$ at a set of spatial points $\boldsymbol{x}^{(k)} = \{x_i^{(k)}\}_{i=1}^{n'_x}$, $n'_x = n_x / n_{\text{set}}$ and $k = \{1, \ldots, n_{\text{set}}\}$:

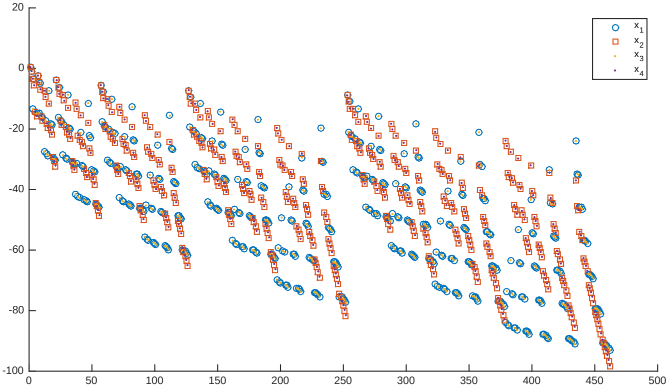
$$J(\boldsymbol{x}^{(k)}) = \begin{bmatrix} \frac{\partial a(x_1^{(k)}, \xi)}{\partial \xi_1} & \ldots & \frac{\partial a(x_{n'_x}^{(k)}, \xi)}{\partial \xi_1} \\ \vdots & & \vdots \\ \frac{\partial a(x_1^{(k)}, \xi)}{\partial \xi_M} & \ldots & \frac{\partial a(x_{n'_x}^{(k)}, \xi)}{\partial \xi_M} \end{bmatrix}$$

Compute the covariance matrix:

$$C(\boldsymbol{x}^{(k)}) = \mathbb{E}\left[J(\boldsymbol{x}^{(k)}) J(\boldsymbol{x}^{(k)})^T\right].$$

Rotate ξ:

$$\eta^{\boldsymbol{x}^{(k)}} = W_{\boldsymbol{x}^{(k)}}^T \xi \in \mathbb{R}^{M \times 1}$$



At spatial points x_i, $i = 1, 2, 3, 4$, $\log_{10}(|\hat{a}_\alpha(x_i, \eta^{\mathbf{x}^{(1)}})|)$