MAX PLANCK INSTITUTE
FOR DYNAMICS OF COMPLEX
TECHNICAL SYSTEMS
MAGDEBURG

CSC

COMPUTATIONAL METHODS IN
SYSTEMS AND CONTROL THEORY

# LOW-RANK TENSOR METHODS for SIMULATION, OPTIMIZATION and UNCERTAINTY QUANTIFICATION of PARAMETRIC PDEs

**Peter Benner**

Joint work with Sergey Dolgov (U Bath), Martin Stoll (TU Chemnitz) and Akwum Onwunta (U Maryland)

Workshop “HPC and Data Science for Scientific Discovery”

IPAM, UCLA, October 15–19, 2018

# Overview

## A (big) data problem in flow simulation

- Assume you want to design an airfoil, wing, car, etc.

## A (big) data problem in flow simulation

- Assume you want to design an airfoil, wing, car, etc.
- This requires 3D flow simulations for varying configurations, and you want to save the data for visualization and further studies/comparisons.

## A (big) data problem in flow simulation

- Assume you want to design an airfoil, wing, car, etc.
- This requires 3D flow simulations for varying configurations, and you want to save the data for visualization and further studies/comparisons.
- **Gedankenexperiment:**

## A (big) data problem in flow simulation

- Assume you want to design an airfoil, wing, car, etc.
- This requires 3D flow simulations for varying configurations, and you want to save the data for visualization and further studies/comparisons.
- **Gedankenexperiment:**
  - we use a grid with $10^6$ nodes, for each node we store 4 real numbers: the velocity in $x$-, $y$-, and $z$-direction, plus the pressure;

## A (big) data problem in flow simulation

- Assume you want to design an airfoil, wing, car, etc.
- This requires 3D flow simulations for varying configurations, and you want to save the data for visualization and further studies/comparisons.
- **Gedankenexperiment:**
  - we use a grid with $10^6$ nodes, for each node we store 4 real numbers: the velocity in $x$-, $y$-, and $z$-direction, plus the pressure;
  - we need $1,000$ time steps to reach steady-state;

## A (big) data problem in flow simulation

- Assume you want to design an airfoil, wing, car, etc.
- This requires 3D flow simulations for varying configurations, and you want to save the data for visualization and further studies/comparisons.
- **Gedankenexperiment:**
  - we use a grid with $10^6$ nodes, for each node we store 4 real numbers: the velocity in $x$-, $y$-, and $z$-direction, plus the pressure;
  - we need $1,000$ time steps to reach steady-state;
  - we have 10 different design parameters, and want to test 10 values for each of them $\rightsquigarrow 10^{10}$ configurations, i.e., $10^{10}$ transient simulations.

## A (big) data problem in flow simulation

- Assume you want to design an airfoil, wing, car, etc.
- This requires 3D flow simulations for varying configurations, and you want to save the data for visualization and further studies/comparisons.
- **Gedankenexperiment:**
  - we use a grid with $10^6$ nodes, for each node we store 4 real numbers: the velocity in $x$-, $y$-, and $z$-direction, plus the pressure;
  - we need $1,000$ time steps to reach steady-state;
  - we have 10 different design parameters, and want to test 10 values for each of them $\rightsquigarrow 10^{10}$ configurations, i.e., $10^{10}$ transient simulations.
- Assuming you have a HPC cluster allowing you to do so in reasonable time, just storing all trajectories requires

$$10^{10} \cdot 10^3 \cdot 10^6 \cdot 4 \cdot 8 = 3.2 \cdot 10^{20} \text{bytes} \approx 320 \text{ exabytes of memory!}$$

Data can be compressed, but first it needs to be generated and stored...

## A (big) data problem in flow simulation

- Assume you want to design an airfoil, wing, car, etc.
- This requires 3D flow simulations for varying configurations, and you want to save the data for visualization and further studies/comparisons.
- **Gedankenexperiment:**
  - we use a grid with $10^6$ nodes, for each node we store 4 real numbers: the velocity in $x$-, $y$-, and $z$-direction, plus the pressure;
  - we need $1,000$ time steps to reach steady-state;
  - we have 10 different design parameters, and want to test 10 values for each of them $\rightsquigarrow 10^{10}$ configurations, i.e., $10^{10}$ transient simulations.
- Assuming you have a HPC cluster allowing you to do so in reasonable time, just storing all trajectories requires

$$10^{10} \cdot 10^3 \cdot 10^6 \cdot 4 \cdot 8 = 3.2 \cdot 10^{20} \text{bytes} \approx 320 \text{ exabytes of memory!}$$

  Data can be compressed, but first it needs to be generated and stored...
- Why not starting with a compressed data format from the very beginning, and computing all trajectories all-at-once?

# Introduction

## A (big) data problem in flow simulation

- Assume you want to design an airfoil, wing, car, etc.
- This requires 3D flow simulations for varying configurations, and you want to save the data for visualization and further studies/comparisons.
- **Gedankenexperiment:**
  - we use a grid with $10^6$ nodes, for each node we store 4 real numbers: the velocity in $x$-, $y$-, and $z$-direction, plus the pressure;
  - we need $1,000$ time steps to reach steady-state;
  - we have 10 different design parameters, and want to test 10 values for each of them $\rightsquigarrow 10^{10}$ configurations, i.e., $10^{10}$ transient simulations.
- Assuming you have a HPC cluster allowing you to do so in reasonable time, just storing all trajectories requires

  $$10^{10} \cdot 10^3 \cdot 10^6 \cdot 4 \cdot 8 = 3.2 \cdot 10^{20} \text{bytes} \approx 320 \text{ exabytes of memory!}$$

  Data can be compressed, but first it needs to be generated and stored...
- Why not starting with a compressed data format from the very beginning, and computing all trajectories all-at-once?
- This is the idea we pursue in this talk, where we store the data in a compressed (low-rank) $1 + 10 + 1(3)$–way tensor! (In above example, $\sim$ terabyte-range.)

## PDEs with stochastic coefficients for UQ

- Physical, biological, chemical, etc. processes involve uncertainties.

## PDEs with stochastic coefficients for UQ

- Physical, biological, chemical, etc. processes involve uncertainties.
- Models of these processes should account for these uncertainties.

## PDEs with stochastic coefficients for UQ

- Physical, biological, chemical, etc. processes involve uncertainties.
- Models of these processes should account for these uncertainties.
- PDEs governing the processes can involve uncertain coefficients, or uncertain sources, or uncertain geometry.

# Introduction

## PDEs with stochastic coefficients for UQ

- Physical, biological, chemical, etc. processes involve uncertainties.
- Models of these processes should account for these uncertainties.
- PDEs governing the processes can involve uncertain coefficients, or uncertain sources, or uncertain geometry.
- Uncertain parameters modeled as random variables $\rightsquigarrow$ random PDEs, potentially also containing uncertain inputs (controls) $\rightsquigarrow$ (generalized) polynomial chaos approach $\rightsquigarrow$ high-dimensional PDE!

# Introduction

## PDEs with stochastic coefficients for UQ

- Physical, biological, chemical, etc. processes involve uncertainties.
- Models of these processes should account for these uncertainties.
- PDEs governing the processes can involve uncertain coefficients, or uncertain sources, or uncertain geometry.
- Uncertain parameters modeled as random variables $\rightsquigarrow$ random PDEs, potentially also containing uncertain inputs (controls) $\rightsquigarrow$ (generalized) polynomial chaos approach $\rightsquigarrow$ high-dimensional PDE!
- Here: no stochastic PDEs in the sense of dynamics driven by Wiener or Lévy or ... processes!

# Introduction

## PDEs with stochastic coefficients for UQ

- Physical, biological, chemical, etc. processes involve uncertainties.
- Models of these processes should account for these uncertainties.
- PDEs governing the processes can involve uncertain coefficients, or uncertain sources, or uncertain geometry.
- Uncertain parameters modeled as random variables $\rightsquigarrow$ random PDEs, potentially also containing uncertain inputs (controls) $\rightsquigarrow$ (generalized) polynomial chaos approach $\rightsquigarrow$ high-dimensional PDE!
- Here: no stochastic PDEs in the sense of dynamics driven by Wiener or Lévy or ... processes!

## Uncertainty arises because

- available data are incomplete;

# Introduction

## PDEs with stochastic coefficients for UQ

- Physical, biological, chemical, etc. processes involve uncertainties.
- Models of these processes should account for these uncertainties.
- PDEs governing the processes can involve uncertain coefficients, or uncertain sources, or uncertain geometry.
- Uncertain parameters modeled as random variables $\rightsquigarrow$ random PDEs, potentially also containing uncertain inputs (controls) $\rightsquigarrow$ (generalized) polynomial chaos approach $\rightsquigarrow$ high-dimensional PDE!
- Here: no stochastic PDEs in the sense of dynamics driven by Wiener or Lévy or ... processes!

## Uncertainty arises because

- available data are incomplete;
- data are predictable, but difficult to measure, e.g., porosity above oil reservoirs;

# Introduction

## PDEs with stochastic coefficients for UQ

- Physical, biological, chemical, etc. processes involve uncertainties.
- Models of these processes should account for these uncertainties.
- PDEs governing the processes can involve uncertain coefficients, or uncertain sources, or uncertain geometry.
- Uncertain parameters modeled as random variables $\rightsquigarrow$ random PDEs, potentially also containing uncertain inputs (controls) $\rightsquigarrow$ (generalized) polynomial chaos approach $\rightsquigarrow$ high-dimensional PDE!
- Here: no stochastic PDEs in the sense of dynamics driven by Wiener or Lévy or ... processes!

## Uncertainty arises because

- available data are incomplete;
- data are predictable, but difficult to measure, e.g., porosity above oil reservoirs;
- data are unpredictable, e.g, wind shear.

# Motivation I: Low-Rank Solvers

**Curse of Dimensionality** [Bellman '57]

Increase matrix size of discretized differential operator for $h \to \frac{h}{2}$ by factor $2^d$.

$\rightsquigarrow$ **Rapid Increase of Dimensionality**, called **Curse of Dimensionality** ($d > 3$).

# Motivation I: Low-Rank Solvers

## Curse of Dimensionality [Bellman '57]

Increase matrix size of discretized differential operator for $h \to \frac{h}{2}$ by factor $2^d$.

$\rightsquigarrow$ **Rapid Increase of Dimensionality**, called **Curse of Dimensionality** $(d > 3)$.

Consider $-\Delta u = f$ in $[0, 1] \times [0, 1] \subset \mathbb{R}^2$, uniformly discretized as

$$(I \otimes A + A \otimes I)\, x =: \mathcal{A}x = b \qquad \Longleftrightarrow \qquad AX + XA^T = B$$

with $x = \mathrm{vec}(X)$ and $b = \mathrm{vec}(B)$ with low-rank right hand side $B \approx b_1 b_2^T$.

**Curse of Dimensionality** [Bellman '57]

Increase matrix size of discretized differential operator for $h \to \frac{h}{2}$ by factor $2^d$.

$\rightsquigarrow$ **Rapid Increase of Dimensionality**, called **Curse of Dimensionality** ($d > 3$).

Consider $-\Delta u = f$ in $[0\,,\,1] \times [0\,,\,1] \subset \mathbb{R}^2$, uniformly discretized as

$$(I \otimes A + A \otimes I)\,x =: \mathcal{A}x = b \qquad \Longleftrightarrow \qquad AX + XA^T = B$$

with $x = \mathrm{vec}\,(X)$ and $b = \mathrm{vec}\,(B)$ with low-rank right hand side $B \approx b_1 b_2^T$.

- Low-rankness of $\tilde{X} := VW^T \approx X$ follows from properties of $A$ and $B$, e.g., [Penzl '00, Grasedyck '04].

# Motivation I: Low-Rank Solvers

**Curse of Dimensionality** [Bellman '57]

Increase matrix size of discretized differential operator for $h \to \frac{h}{2}$ by factor $2^d$.
$\rightsquigarrow$ **Rapid Increase of Dimensionality**, called **Curse of Dimensionality** $(d > 3)$.

Consider $-\Delta u = f$ in $[0, 1] \times [0, 1] \subset \mathbb{R}^2$, uniformly discretized as

$$(I \otimes A + A \otimes I)\, x =: \mathcal{A}x = b \qquad \Longleftrightarrow \qquad AX + XA^T = B$$

with $x = \mathrm{vec}\,(X)$ and $b = \mathrm{vec}\,(B)$ with low-rank right hand side $B \approx b_1 b_2^T$.

- Low-rankness of $\tilde{X} := VW^T \approx X$ follows from properties of $A$ and $B$, e.g., [Penzl '00, Grasedyck '04].
- We solve this using low-rank Krylov subspace solvers. These essentially require matrix-vector multiplication and vector computations.

# Motivation I: Low-Rank Solvers

**Curse of Dimensionality** [Bellman '57]

Increase matrix size of discretized differential operator for $h \to \frac{h}{2}$ by factor $2^d$.
$\rightsquigarrow$ **Rapid Increase of Dimensionality**, called **Curse of Dimensionality** $(d > 3)$.

Consider $-\Delta u = f$ in $[0\,,\,1] \times [0\,,\,1] \subset \mathbb{R}^2$, uniformly discretized as

$$(I \otimes A + A \otimes I)\,x =: \mathcal{A}x = b \qquad \Longleftrightarrow \qquad AX + XA^T = B$$

with $x = \mathrm{vec}\,(X)$ and $b = \mathrm{vec}\,(B)$ with low-rank right hand side $B \approx b_1 b_2^T$.

- Low-rankness of $\tilde{X} := VW^T \approx X$ follows from properties of $A$ and $B$, e.g., [Penzl '00, Grasedyck '04].
- We solve this using low-rank Krylov subspace solvers. These essentially require matrix-vector multiplication and vector computations.
- Hence, $\mathcal{A}\mathrm{vec}\,(X_k) = \mathcal{A}\mathrm{vec}\,\left(V_k W_k^T\right) = \mathrm{vec}\left(\left[\,AV_k,\ V_k\,\right]\left[\,W_k,\ AW_k\,\right]^T\right)$

# Motivation I: Low-Rank Solvers

## Curse of Dimensionality [Bellman '57]

Increase matrix size of discretized differential operator for $h \to \frac{h}{2}$ by factor $2^d$.
$\rightsquigarrow$ **Rapid Increase of Dimensionality**, called **Curse of Dimensionality** ($d > 3$).

Consider $-\Delta u = f$ in $[0\,,\,1] \times [0\,,\,1] \subset \mathbb{R}^2$, uniformly discretized as

$$(I \otimes A + A \otimes I)\, x =: \mathcal{A}x = b \qquad \Longleftrightarrow \qquad AX + XA^T = B$$

with $x = \mathrm{vec}\,(X)$ and $b = \mathrm{vec}\,(B)$ with low-rank right hand side $B \approx b_1 b_2^T$.

- Low-rankness of $\tilde{X} := VW^T \approx X$ follows from properties of $A$ and $B$, e.g., [Penzl '00, Grasedyck '04].
- We solve this using low-rank Krylov subspace solvers. These essentially require matrix-vector multiplication and vector computations.
- Hence, $\mathcal{A}\mathrm{vec}\,(X_k) = \mathcal{A}\mathrm{vec}\,(V_k W_k^T) = \mathrm{vec}\left([\,AV_k,\ V_k\,]\,[\,W_k,\ AW_k\,]^T\right)$
- The rank of $[AV_k \quad V_k] \in \mathbb{R}^{n,2r}$, $[W_k \quad AW_k] \in \mathbb{R}^{n_t,2r}$ increases but can be controlled using truncation. $\rightsquigarrow$ Low-rank Krylov subspace solvers. [Kressner/Tobler, B/Breiten, Savostyanov/Dolgov, ...].

# Motivation II: Optimization under Uncertainty

We consider the problem:

$$\min_{y \in \mathcal{Y}, u \in \mathcal{U}} \mathcal{J}(y, u) \quad \text{subject to} \quad c(y, u) = 0,$$

where

- $c(y, u) = 0$ represents a (linear or nonlinear) PDE (system) with uncertain coefficient(s).
- The state $y$ and control $u$ are random fields.
- The cost functional $\mathcal{J}$ is a real-valued Fréchet-differentiable functional on $\mathcal{Y} \times \mathcal{U}$.

# This Talk

## Curse of Dimensionality [Bellman '57]

Increase matrix size of discretized differential operator for $h \to \frac{h}{2}$ by factor $2^d$.

$\rightsquigarrow$ **Rapid Increase of Dimensionality**, called **Curse of Dimensionality** ($d > 3$).

## Goal of this talk

Apply low-rank iterative solvers to discrete optimality systems resulting from

**PDE-constrained optimization problems under uncertainty,**

and go one step further applying low-rank tensor (instead of matrix) techniques.

# This Talk

## Curse of Dimensionality [Bellman '57]

Increase matrix size of discretized differential operator for $h \to \frac{h}{2}$ by factor $2^d$.

$\rightsquigarrow$ **Rapid Increase of Dimensionality**, called **Curse of Dimensionality** ($d > 3$).

## Goal of this talk

Apply low-rank iterative solvers to discrete optimality systems resulting from

**PDE-constrained optimization problems under uncertainty,**

and go one step further applying low-rank tensor (instead of matrix) techniques.

## Take home message

Biggest problem solved so far has $n = 1.29 \cdot 10^{15}$ unknowns (KKT system for unsteady incompressible Navier-Stokes control problem with uncertain viscosity).

# This Talk

## Curse of Dimensionality [Bellman '57]

Increase matrix size of discretized differential operator for $h \rightarrow \frac{h}{2}$ by factor $2^d$.

$\rightsquigarrow$ **Rapid Increase of Dimensionality**, called **Curse of Dimensionality** ($d > 3$).

## Goal of this talk

Apply low-rank iterative solvers to discrete optimality systems resulting from

**PDE-constrained optimization problems under uncertainty,**

and go one step further applying low-rank tensor (instead of matrix) techniques.

## Take home message

Biggest problem solved so far has $n = 1.29 \cdot 10^{15}$ unknowns (KKT system for unsteady incompressible Navier-Stokes control problem with uncertain viscosity).

Would require $\approx 10$ **petabytes (PB)** $= 10,000$ **TB** to store the solution vector!

# This Talk

## Curse of Dimensionality [Bellman '57]

Increase matrix size of discretized differential operator for $h \to \frac{h}{2}$ by factor $2^d$.

$\rightsquigarrow$ **Rapid Increase of Dimensionality**, called **Curse of Dimensionality** ($d > 3$).

## Goal of this talk

Apply low-rank iterative solvers to discrete optimality systems resulting from

**PDE-constrained optimization problems under uncertainty,**

and go one step further applying low-rank tensor (instead of matrix) techniques.

## Take home message

Biggest problem solved so far has $n = 1.29 \cdot 10^{15}$ unknowns (KKT system for unsteady incompressible Navier-Stokes control problem with uncertain viscosity).

Would require $\approx 10$ **petabytes (PB)** $= 10,000$ **TB** to store the solution vector!

Using low-rank tensor techniques, we need $\approx 7 \cdot 10^7$ **bytes** $= 70$ **GB** to solve the KKT system in MATLAB in less than one hour!

# Unsteady Heat Equation

Consider the optimization problem

$$\mathcal{J}(t,y,u) = \frac{1}{2}||y-\bar{y}||^2_{L^2(0,T;\mathcal{D})\otimes L^2(\Omega)} + \frac{\alpha}{2}||\mathrm{std}(y)||^2_{L^2(0,T;\mathcal{D})} + \frac{\beta}{2}||u||^2_{L^2(0,T;\mathcal{D})\otimes L^2(\Omega)}$$

subject, $\mathbb{P}$-almost surely, to

$$\begin{cases} \dfrac{\partial y(t,\mathbf{x},\omega)}{\partial t} - \nabla\cdot(a(\mathbf{x},\omega)\nabla y(t,\mathbf{x},\omega)) = u(t,\mathbf{x},\omega), & \text{in } (0,T]\times\mathcal{D}\times\Omega, \\ y(t,\mathbf{x},\omega) = 0, & \text{on } (0,T]\times\partial\mathcal{D}\times\Omega, \\ y(0,\mathbf{x},\omega) = y_0, & \text{in } \mathcal{D}\times\Omega, \end{cases}$$

where

- for any $z:\mathcal{D}\times\Omega\to\mathbb{R}$, $z(\mathbf{x},\cdot)$ is a random variable defined on the complete probability space $(\Omega,\mathcal{F},\mathbb{P})$ for each $\mathbf{x}\in\mathcal{D}$,
- $\exists\ 0 < a_{\min} < a_{\max} < \infty$ s.t. $\mathbb{P}(\omega\in\Omega : a(x,\omega)\in[a_{\min},a_{\max}]\ \forall x\in\mathcal{D}) = 1.$

# Discretization

We discretize and then optimize the stochastic control problem.

- Under finite noise assumption we can use $N$-term (truncated) **Karhunen-Loève expansion (KLE)**

$$a \equiv a(\mathbf{x}, \omega) \approx a_N(\mathbf{x}, \xi(\omega)) \equiv a_N(\mathbf{x}, \xi_1(\omega), \xi_2(\omega), \ldots, \xi_N(\omega)).$$

- Assuming a known continuous covariance $C_a(\mathbf{x}, \mathbf{y})$, we get the KLE

$$a_N(\mathbf{x}, \xi(\omega)) = \mathbb{E}[a](\mathbf{x}) + \sigma_a \sum_{i=1}^{N} \sqrt{\lambda_i} \varphi_i(\mathbf{x}) \xi_i(\omega),$$

  where $(\lambda_i, \varphi_i)$ are the dominant eigenpairs of $C_a$.

- Doob-Dynkin Lemma allows same parametrization for solution $y$.
- Use linear finite elements for the spatial discretization and implicit Euler in time.

This is used within a **stochastic Galerkin FEM (SGFEM)** approach.

## Weak formulation of the random PDE

Seek $y \in H^1\left(0, T; H_0^1(\mathcal{D}) \otimes L^2(\Omega)\right)$ such that, $\mathbb{P}$-almost surely,

$$\langle y_t, v\rangle + \mathcal{B}(y, v) = \ell(u, v) \quad \forall\, v \in H_0^1(\mathcal{D}) \otimes L^2(\Omega),$$

with the coercive[1] bilinear form

$$\mathcal{B}(y, v) := \int_\Omega \int_\mathcal{D} a(\mathbf{x}, \omega) \nabla y(\mathbf{x}, \omega) \cdot \nabla v(\mathbf{x}, \omega) d\mathbf{x}\, d\mathbb{P}(\omega), \quad v, y \in H_0^1(\mathcal{D}) \otimes L^2(\Omega),$$

and

$$\begin{aligned} \ell(u, v) &= \langle u(\mathbf{x}, \omega), v(\mathbf{x}, \omega)\rangle \\ &=: \int_\Omega \int_\mathcal{D} u(\mathbf{x}, \omega) v(\mathbf{x}, \omega) d\mathbf{x}\, d\mathbb{P}(\omega), \quad u, v \in H_0^1(\mathcal{D}) \otimes L^2(\Omega). \end{aligned}$$

Coercivity and boundedness of $\mathcal{B}$ + Lax-Milgram $\Longrightarrow$ unique solution exists.

[1] due to the positivity assumption on $a(\mathbf{x}, \omega)$

## Weak formulation of the optimality system

**Theorem** [Chen/Quarteroni '14, B./Onwunta/Stoll '18]

Under appropriate regularity assumptions, there exists a unique adjoint state $p$ and optimal solution $(y, u, p)$ to the optimal control problem for the random unsteady heat equation, satisfying the stochastic optimality conditions (KKT system) for $t \in (0, T]$ $\mathbb{P}$-almost surely

$$
\begin{aligned}
\langle y_t, v\rangle + \mathcal{B}(y, v) &= \ell(u, v), && \forall\, v \in H_0^1(\mathcal{D}) \otimes L^2(\Omega),\\
\langle p_t, w\rangle - \mathcal{B}^*(p, w) &= \ell\left((y - \bar{y}) + \frac{\alpha}{2}\mathcal{S}(y), w\right), && \forall\, w \in H_0^1(\mathcal{D}) \otimes L^2(\Omega),\\
\ell(\beta u - p, \tilde{w}) &= 0, && \forall\, \tilde{w} \in L^2(\mathcal{D}) \otimes L^2(\Omega),
\end{aligned}
$$

where

- $\mathcal{S}(y)$ is the Fréchet derivative of $\|\mathrm{std}(y)\|^2_{L^2(0,T;\mathcal{D})}$;
- $\mathcal{B}^*$ is the adjoint operator of $\mathcal{B}$.

## Discretization of the random PDE

- $y, p, u$ are approximated using standard Galerkin ansatz, yielding approximations of the form

$$z(t,\mathbf{x},\omega) = \sum_{k=0}^{P-1}\sum_{j=1}^{J} z_{jk}(t)\phi_j(\mathbf{x})\psi_k(\xi) = \sum_{k=0}^{P-1} z_k(t,\mathbf{x})\psi_k(\xi).$$

- Here,
  - $\{\phi_j\}_{j=1}^{J}$ are linear finite elements;
  - $\{\psi_k\}_{k=0}^{P-1}$ are the $P = \frac{(N+n)!}{N!n!}$ multivariate Legendre polynomials of degree $\leq n$.
  - Implicit Euler/dG(0) used for temporal discretization with constant time step $\tau$.

# The Fully Discretized Optimal Control Problem

## Discrete first order optimality conditions/KKT system

$$\begin{bmatrix} \tau\mathcal{M}_1 & 0 & -\mathcal{K}_t^T \\ 0 & \beta\tau\mathcal{M}_2 & \tau\mathcal{N}^T \\ -\mathcal{K}_t & \tau\mathcal{N} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \tau\mathcal{M}_\alpha\bar{\mathbf{y}} \\ \mathbf{0} \\ \mathbf{d} \end{bmatrix},$$

where

- $\mathcal{M}_1 = D \otimes G_\alpha \otimes M =: D \otimes \mathcal{M}_\alpha, \quad \mathcal{M}_2 = D \otimes G_0 \otimes M,$
- $\mathcal{K}_t = I_{n_t} \otimes \left[\sum_{i=0}^{N} G_i \otimes \widehat{K}_i\right] + (C \otimes G_0 \otimes M),$
- $\mathcal{N} = I_{n_t} \otimes G_0 \otimes M,$

and

- $G_0 = \mathrm{diag}\left(\langle\psi_0^2\rangle, \langle\psi_1^2\rangle, \ldots, \langle\psi_{P-1}^2\rangle\right), \quad G_i(j,k) = \langle\xi_i\psi_j\psi_k\rangle, \quad i = 1, \ldots, N,$
- $G_\alpha = G_0 + \alpha\,\mathrm{diag}\left(0, \langle\psi_1^2\rangle, \ldots, \langle\psi_{P-1}^2\rangle\right)$ (with first moments $\langle\,.\,\rangle$ w.r.t. $\mathbb{P}$),
- $\hat{K}_0 = M + \tau K_0, \quad \hat{K}_i = \tau K_i, \quad i = 1, \ldots, N,$
- $M, K_i \in \mathbb{R}^{J\times J}$ are the mass and stiffness matrices w.r.t. the spatial discretization, where $K_i$ corresponds to the contributions of the $i$th KLE term to the stiffness,
- $C = -\mathrm{diag}(\mathtt{ones}, -1), \quad D = \mathrm{diag}\left(\frac{1}{2}, 1, \ldots, 1, \frac{1}{2}\right) \in \mathbb{R}^{n_t\times n_t}.$

## Discrete first order optimality conditions/KKT system

$$\begin{bmatrix} \tau\mathcal{M}_1 & 0 & -\mathcal{K}_t^T \\ 0 & \beta\tau\mathcal{M}_2 & \tau\mathcal{N}^T \\ -\mathcal{K}_t & \tau\mathcal{N} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \tau\mathcal{M}_\alpha\bar{\mathbf{y}} \\ \mathbf{0} \\ \mathbf{d} \end{bmatrix},$$

**Linear system with $3JPn_t$ unknowns!**

# Solving the KKT System

Optimality system leads to saddle point problem

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix}.$$

- Very large scale setting, (block-)structured sparsity $\rightsquigarrow$ iterative solution.

# Solving the KKT System

Optimality system leads to saddle point problem

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix}.$$

- Very large scale setting, (block-)structured sparsity $\rightsquigarrow$ iterative solution.
- Krylov subspace methods for indefinite symmetric systems: MINRES, . . . .

# Solving the KKT System

Optimality system leads to saddle point problem

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix}.$$

- Very large scale setting, (block-)structured sparsity $\rightsquigarrow$ iterative solution.
- Krylov subspace methods for indefinite symmetric systems: MINRES, . . . .
- Requires good preconditioner.

# Solving the KKT System

Optimality system leads to saddle point problem

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix}.$$

- Very large scale setting, (block-)structured sparsity $\rightsquigarrow$ iterative solution.
- Krylov subspace methods for indefinite symmetric systems: MINRES, ....
- Requires good preconditioner.
- Famous three-iterations-convergence result [Murphy/Golub/Wathen '00]: using ideal preconditioner

$$\mathcal{P} := \begin{bmatrix} A & 0 \\ 0 & -S \end{bmatrix} \qquad \text{with the Schur complement} \quad S := -BA^{-1}B^T,$$

MINRES finds the exact solution in at most three steps.

# Solving the KKT System

Optimality system leads to saddle point problem

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix}.$$

- Very large scale setting, (block-)structured sparsity $\rightsquigarrow$ iterative solution.
- Krylov subspace methods for indefinite symmetric systems: MINRES, ....
- Requires good preconditioner.
- Famous three-iterations-convergence result [Murphy/Golub/Wathen '00]: using ideal preconditioner

  $$\mathcal{P} := \begin{bmatrix} A & 0 \\ 0 & -S \end{bmatrix} \qquad \text{with the Schur complement} \quad S := -BA^{-1}B^T,$$

  MINRES finds the exact solution in at most three steps.
- Motivates to use approximate Schur complement preconditioner $\begin{bmatrix} \hat{A} & 0 \\ 0 & \hat{S} \end{bmatrix}$.

Optimality system leads to saddle point problem

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix}.$$

- Very large scale setting, (block-)structured sparsity $\rightsquigarrow$ iterative solution.
- Krylov subspace methods for indefinite symmetric systems: MINRES, ....
- Requires good preconditioner.
- Famous three-iterations-convergence result [Murphy/Golub/Wathen '00]: using ideal preconditioner

$$\mathcal{P} := \begin{bmatrix} A & 0 \\ 0 & -S \end{bmatrix} \qquad \text{with the Schur complement} \quad S := -BA^{-1}B^T,$$

  MINRES finds the exact solution in at most three steps.

- Motivates to use approximate Schur complement preconditioner $\begin{bmatrix} \hat{A} & 0 \\ 0 & \hat{S} \end{bmatrix}$.
- Here, $A \sim$ mass matrices $\rightsquigarrow$ application of $A^{-1}$ is approximated using a small number of Chebyshev semi-iterations.

# Solving the KKT System

Optimality system leads to saddle point problem

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \quad \text{with approximate Schur complement preconditioner} \quad \begin{bmatrix} \hat{A} & 0 \\ 0 & \hat{S} \end{bmatrix}.$$

- How to approximate the application of the inverse Schur complement efficiently?

# Solving the KKT System

Optimality system leads to saddle point problem

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \quad \text{with approximate Schur complement preconditioner} \quad \begin{bmatrix} \hat{A} & 0 \\ 0 & \hat{S} \end{bmatrix}.$$

- How to approximate the application of the inverse Schur complement efficiently?
- Pioneering work by Elman, Ernst, Ullmann, Powell, Silvester, ...

# Solving the KKT System

Optimality system leads to saddle point problem

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \quad \text{with approximate Schur complement preconditioner} \quad \begin{bmatrix} \hat{A} & 0 \\ 0 & \hat{S} \end{bmatrix}.$$

- How to approximate the application of the inverse Schur complement efficiently?
- Pioneering work by Elman, Ernst, Ullmann, Powell, Silvester, ...

**Theorem** [B./Onwunta/Stoll '16]

Let $\alpha \in [0, +\infty)$ and

$$\tilde{S} = \frac{1}{\tau} (\mathcal{K} + \tau\gamma\mathcal{N}) \mathcal{M}_1^{-1} (\mathcal{K} + \tau\gamma\mathcal{N})^T,$$

where $\gamma = \sqrt{(1+\alpha)/\beta}$ and $\mathcal{K} = \sum_{i=0}^{N} G_i \otimes K_i$.
Then the eigenvalues of $\tilde{S}^{-1}S$ satisfy

$$\lambda(\tilde{S}^{-1}S) \subset \left[\frac{1}{2(1+\alpha)}, 1\right), \quad \forall \alpha < \left(\frac{\sqrt{\kappa(\mathcal{K})}+1}{\sqrt{\kappa(\mathcal{K})}-1}\right)^2 - 1.$$

Optimality system leads to saddle point problem

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \quad \text{with approximate Schur complement preconditioner} \quad \begin{bmatrix} \hat{A} & 0 \\ 0 & \hat{S} \end{bmatrix}.$$

- How to approximate the application of the inverse Schur complement efficiently?
- Pioneering work by Elman, Ernst, Ullmann, Powell, Silvester, ...

**Corollary** [B./Onwunta/Stoll '16]

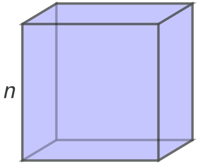
Let $\mathcal{A}$ be the KKT matrix from the stochastic Galerkin approach, and $\mathcal{P}$ the preconditioner using the Schur complement approximation $\tilde{S}$ (and exact $A$). Then

$$\lambda(\mathcal{P}^{-1}\mathcal{A}) \subset \{1\} \cup \mathcal{I}^+ \cup \mathcal{I}^-,$$

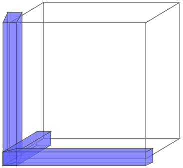
where

$$\mathcal{I}^\pm = \frac{1}{2}\left(1 \pm \left[\sqrt{1+\frac{2}{1+\alpha}}, \sqrt{5}\right]\right).$$

## Separation of variables and low-rank approximation




$\rightsquigarrow$



- Approximate: $\underbrace{\mathbf{x}(i_1,\ldots,i_d)}_{\text{tensor}} \approx \underbrace{\sum_{\alpha} \mathbf{x}_{\alpha}^{(1)}(i_1)\mathbf{x}_{\alpha}^{(2)}(i_2)\cdots\mathbf{x}_{\alpha}^{(d)}(i_d)}_{\text{tensor product decomposition}}$.

Goals:

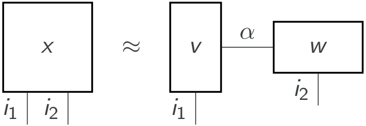
- Store and manipulate $x$ — $\mathcal{O}(dn)$ cost *instead of* $\mathcal{O}(n^d)$.
- Solve equations $Ax = b$ — $\mathcal{O}(dn^2)$ cost *instead of* $\mathcal{O}(n^{2d})$.

## Data Compression in 2D: Low-Rank Matrices

- Discrete *separation of variables*:

$$\begin{bmatrix} x_{1,1} & \cdots & x_{1,n} \\ \vdots & & \vdots \\ x_{n,1} & \cdots & x_{n,n} \end{bmatrix} = \sum_{\alpha=1}^{r} \begin{bmatrix} v_{1,\alpha} \\ \vdots \\ v_{n,\alpha} \end{bmatrix} \begin{bmatrix} w_{\alpha,1} & \cdots & w_{\alpha,n} \end{bmatrix} + \mathcal{O}(\varepsilon).$$

- Diagrams:




- *Rank* $r \ll n$.
- $\mathtt{mem}(v) + \mathtt{mem}(w) = 2nr \ll n^2 = \mathtt{mem}(x)$.
- *Singular Value Decomposition (SVD)*

  $\Longrightarrow \quad \varepsilon(r)$ optimal w.r.t. spectral/Frobenius norm.

## Tensor Trains/Matrix Product States

[Wilson '75, White '93, Verstraete '04, Oseledets '09/'11]

For indices

$$\overline{i_p \dots i_q} = (i_p - 1)n_{p+1} \cdots n_q + (i_{p+1} - 1)n_{p+2} \cdots n_q + \cdots + (i_{q-1} - 1)n_q + i_q,$$

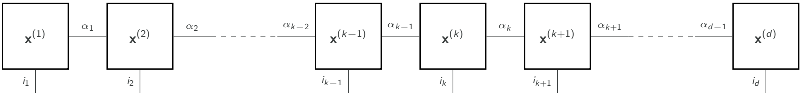
the TT format can be expressed as

$$x(\overline{i_1 \dots i_d}) = \sum_{\boldsymbol{\alpha}=\mathbf{1}}^{\mathbf{r}} \mathbf{x}^{(1)}_{\alpha_1}(i_1) \cdot \mathbf{x}^{(2)}_{\alpha_1,\alpha_2}(i_2) \cdot \mathbf{x}^{(3)}_{\alpha_2,\alpha_3}(i_3) \cdots \mathbf{x}^{(d)}_{\alpha_{d-1},\alpha_d}(i_d)$$

or

$$x(\overline{i_1 \dots i_d}) = \mathbf{x}^{(1)}(i_1) \cdots \mathbf{x}^{(d)}(i_d), \quad \mathbf{x}^{(k)}(i_k) \in \mathbb{R}^{r_{k-1} \times r_k} \text{ w/ } r_0, r_d = 1,$$

or




**Storage:** $\mathcal{O}(dnr^2)$ instead of $\mathcal{O}(n^d)$.

# Overloading Tensor Operations

*Always* work with *factors* $\mathbf{x}^{(k)} \in \mathbb{R}^{r_{k-1} \times n_k \times r_k}$ instead of full tensors.

- Sum $z = x + y \quad \rightsquigarrow \quad$ increase of tensor rank $\quad r_z = r_x + r_y$.
- TT format for a high-dimensional operator

$$A(\overline{i_1 \dots i_d}, \overline{j_1 \dots j_d}) = \mathbf{A}^{(1)}(i_1, j_1) \cdots \mathbf{A}^{(d)}(i_d, j_d)$$

- *Matrix-vector* multiplication $y = Ax$; $\quad \rightsquigarrow$ tensor rank $r_y = r_A \cdot r_x$.
- Additions and multiplications *increase* TT ranks.
- *Decrease* ranks quasi-optimally via QR and SVD.

# Solving KKT System using TT Format

The dimensionality of the saddle point system is vast $\Rightarrow$ use tensor structure and low tensor ranks.

# Solving KKT System using TT Format

The dimensionality of the saddle point system is vast $\Rightarrow$ use tensor structure and low tensor ranks.

Use tensor train format to approximate the solution as

$$\mathbf{y}(i_1,\ldots,i_d) \approx \sum_{\alpha_1\ldots\alpha_{d-1}=1}^{r_1\ldots r_{d-1}} \mathbf{y}^{(1)}_{\alpha_1}(i_1)\mathbf{y}^{(2)}_{\alpha_1,\alpha_2}(i_2)\cdots\mathbf{y}^{(d-1)}_{\alpha_{d-2},\alpha_{d-1}}(i_{d-1})\mathbf{y}^{(d)}_{\alpha_{d-1}}(i_d),$$

and represent the coefficient matrix as

$$\mathcal{A}(i_1\cdots i_d, j_1\cdots j_d) \approx \sum_{\beta_1\ldots\beta_{d-1}=1}^{R_1\ldots R_{d-1}} \mathbf{A}^{(1)}_{\beta_1}(i_1,j_1)\mathbf{A}^{(2)}_{\beta_1,\beta_2}(i_2,j_2)\cdots\mathbf{A}^{(d)}_{\beta_{d-1}}(i_d,j_d),$$

where the multi-index $\mathbf{i}=(i_1,\ldots,i_d)$ is implied by the parametrization of the approximate solutions of the form

$$\mathbf{z}(t,\xi_1,\ldots,\xi_N,\mathbf{x}), \quad \mathbf{z}=\mathbf{y},\mathbf{u},\mathbf{p},$$

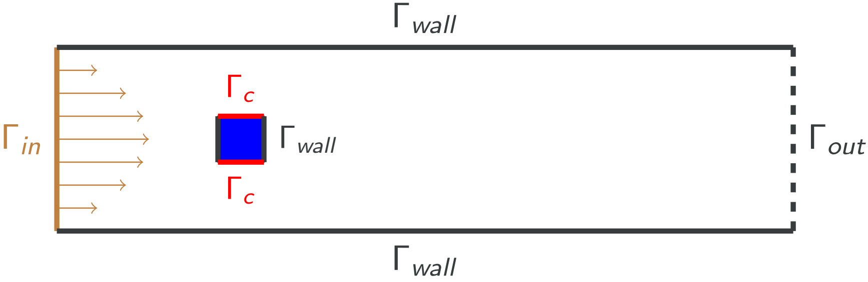
i.e., solution vectors are represented by $d$-way tensor with $d=N+2$.

### Mean-Based Preconditioned TT-MinRes

| TT-MINRES | # iter (t) | # iter (t) | # iter (t) |
|---|---|---|---|
| $n_t$ | $2^5$ | $2^6$ | $2^8$ |
| $\dim(\mathcal{A}) = 3JPn_t$ | 10, 671, 360 | 21, 342, 720 | 85, 370, 880 |
| $\alpha = 1,\ \text{tol} = 10^{-3}$ | | | |
| $\beta = 10^{-5}$ | 6 (285.5) | 6 (300.0) | 8 (372.2) |
| $\beta = 10^{-6}$ | 4 (77.6) | 4 (130.9) | 4 (126.7) |
| $\beta = 10^{-8}$ | 4 (56.7) | 4 (59.4) | 4 (64.9) |
| $\alpha = 0,\ \text{tol} = 10^{-3}$ | | | |
| $\beta = 10^{-5}$ | 4 (207.3) | 6 (366.5) | 6 (229.5) |
| $\beta = 10^{-6}$ | 4 (153.9) | 4 (158.3) | 4 (172.0) |
| $\beta = 10^{-8}$ | 2 (35.2) | 2 (37.8) | 2 (40.0) |

## Model Problem: 'Uncertain' flow past a rectangular obstacle domain

▸ to Numerical Experiments




- We model this as a boundary control problem.
- Our constraint $c(y, u) = 0$ is given by the unsteady incompressible Navier-Stokes equations with uncertain viscosity $\nu := \nu(\omega)$.

# Model Problem

Minimize:

$$\mathcal{J}(v,u) = \frac{1}{2}\|\mathrm{curl} v\|^2_{L^2(0,T;\mathcal{D})\otimes L^2(\Omega)} + \frac{\beta}{2}\|u\|^2_{L^2(0,T;\mathcal{D})\otimes L^2(\Omega)} \tag{1}$$

subject to

$$\begin{aligned}
\frac{\partial v}{\partial t} - \nu\Delta v + (v\cdot\nabla)v + \nabla p = 0, &\quad \text{in} \quad \mathcal{D},\\
-\nabla\cdot v = 0, &\quad \text{in} \quad \mathcal{D},\\
v = \theta, &\quad \text{on} \quad \Gamma_{in},\\
v = 0, &\quad \text{on} \quad \Gamma_{wall},\\
\frac{\partial v}{\partial n} = u, &\quad \text{on} \quad \Gamma_{c},\\
\frac{\partial v}{\partial n} = 0, &\quad \text{on} \quad \Gamma_{out},\\
v(\cdot,0,\cdot) = v_0, &\quad \text{in} \quad \mathcal{D}.
\end{aligned} \tag{2}$$

## Model Problem Set-up (cf. [Powell/Silvester '12])

We assume

- $\nu(\omega) = \nu_0 + \nu_1 \xi(\omega), \;\; \nu_0, \nu_1 \in \mathbb{R}^+, \;\; \xi \sim \mathcal{U}(-1, 1).$
- $\mathbb{P}\left(\omega \in \Omega : \nu(\omega) \in [\nu_{\min}, \nu_{\max}]\right) = 1$, for some $0 < \nu_{\min} < \nu_{\max} < +\infty$.
- $\Rightarrow$ velocity $v$, control $u$ and pressure $p$ are random fields on $L^2(\Omega)$.
- $L^2(\Omega) := L^2(\Omega, \mathcal{F}, \mathbb{P})$ is a complete probability space.
- $L^2(0, T; \mathcal{D}) := L^2(\mathcal{D}) \times L^2(\mathcal{T})$.

# Model Problem Set-up (cf. [Powell/Silvester '12])

We assume

- $\nu(\omega) = \nu_0 + \nu_1 \xi(\omega), \;\; \nu_0, \nu_1 \in \mathbb{R}^+, \;\; \xi \sim \mathcal{U}(-1, 1).$
- $\mathbb{P}\left(\omega \in \Omega : \nu(\omega) \in [\nu_{\min}, \nu_{\max}]\right) = 1$, for some $0 < \nu_{\min} < \nu_{\max} < +\infty$.
- $\Rightarrow$ velocity $v$, control $u$ and pressure $p$ are random fields on $L^2(\Omega)$.
- $L^2(\Omega) := L^2(\Omega, \mathcal{F}, \mathbb{P})$ is a complete probability space.
- $L^2(0, T; \mathcal{D}) := L^2(\mathcal{D}) \times L^2(\mathcal{T})$.

**Computational challenges**

- Nonlinearity (due to the nonlinear convection term $(v \cdot \nabla) v$) .
- Uncertainty (due to random $\nu(\omega)$).
- High dimensionality (of the resulting linear/optimality systems).

## OTD Strategy and Picard (Oseen) Iteration $\rightsquigarrow$

**state equation**

$$v_t - \nu \Delta v + (\bar{v} \cdot \nabla) v + \nabla p = 0$$

$$\nabla \cdot v = 0 + \text{ boundary conditions}$$

**adjoint equation**

$$-\chi_t - \Delta \chi - (\bar{v} \cdot \nabla) \chi + (\nabla \bar{v})^T \chi + \nabla \mu = -\mathrm{curl}^2 v$$

$$\nabla \cdot \chi = 0$$

$$\text{on} \quad \Gamma_{wall} \cup \Gamma_{in} : \qquad \chi = 0$$

$$\text{on} \quad \Gamma_{out} \cup \Gamma_c : \qquad \frac{\partial \chi}{\partial n} = 0$$

$$\chi(\cdot, T, \cdot) = 0$$

**gradient equation**

$$\beta u + \chi|_{\Gamma_c} = 0.$$

## OTD Strategy and Picard (Oseen) Iteration $\rightsquigarrow$

state equation

$$v_t - \nu \Delta v + (\bar{v} \cdot \nabla) v + \nabla p = 0$$
$$\nabla \cdot v = 0 + \text{ boundary conditions}$$

adjoint equation

$$-\chi_t - \Delta \chi - (\bar{v} \cdot \nabla) \chi + (\nabla \bar{v})^T \chi + \nabla \mu = -\mathrm{curl}^2 v$$
$$\nabla \cdot \chi = 0$$
$$\text{on} \quad \Gamma_{wall} \cup \Gamma_{in} : \qquad \chi = 0$$
$$\text{on} \quad \Gamma_{out} \cup \Gamma_c : \qquad \frac{\partial \chi}{\partial n} = 0$$
$$\chi(\cdot, T, \cdot) = 0$$

gradient equation

$$\beta u + \chi|_{\Gamma_c} = 0.$$

- $\bar{v}$ denotes the velocity from the previous Oseen iteration.
- Having solved this system, we update $\bar{v} = v$ until convergence.

- Velocity $v$ and control $u$ are of the form

$$z(t,x,\omega) = \sum_{k=0}^{P-1} \sum_{j=1}^{J_v} z_{jk}(t)\phi_j(x)\psi_k(\xi) = \sum_{k=0}^{P-1} z_k(t,x)\psi_k(\xi).$$

- Pressure $p$ is of the form

$$p(t,x,\omega) = \sum_{k=0}^{P-1} \sum_{j=1}^{J_p} p_{jk}(t)\tilde{\phi}_j(x)\psi_k(\xi) = \sum_{k=0}^{P-1} p_k(t,x)\psi_k(\xi).$$

- Here,
  - $\{\phi_j\}_{j=1}^{J_v}$ and $\{\tilde{\phi}_j\}_{j=1}^{J_p}$ are Q2–Q1 finite elements;
  - $\{\psi_k\}_{k=0}^{P-1}$ are Legendre polynomials.
- Implicit Euler/dG(0) used for temporal discretization.

# Discrete Stochastic Control Problem

Linearization and SGFEM discretization yields the following saddle point system

$$\underbrace{\begin{bmatrix} M_y & 0 & L^* \\ 0 & M_u & N^\top \\ L & N & 0 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} y \\ u \\ \lambda \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} f \\ 0 \\ g \end{bmatrix}}_{b}.$$

Each of the block matrices in $A$ is of the form

$$\sum_{\alpha=1}^{R} X_\alpha \otimes Y_\alpha \otimes Z_\alpha,$$

corresponding to temporal, stochastic, and spatial discretizations.

# Discrete Stochastic Control Problem

Linearization and SGFEM discretization yields the following saddle point system

$$\underbrace{\begin{bmatrix} M_y & 0 & L^* \\ 0 & M_u & N^\top \\ L & N & 0 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} y \\ u \\ \lambda \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} f \\ 0 \\ g \end{bmatrix}}_{b}.$$

Each of the block matrices in $A$ is of the form

$$\sum_{\alpha=1}^{R} X_\alpha \otimes Y_\alpha \otimes Z_\alpha,$$

corresponding to temporal, stochastic, and spatial discretizations.

Size: $\sim 3n_t P(J_v + J_p)$, e.g., for $P = 10$, $n_t = 2^{10}$, $J \approx 10^5$ $\rightsquigarrow \approx 10^9$ unknowns!

# Linear Systems in TT Format

## Central Question

How to solve $Ax = b$ if Krylov solvers become too expensive?

# Linear Systems in TT Format

## Central Question

How to solve $Ax = b$ if Krylov solvers become too expensive?

Data are given in TT format:

- $A(i,j) = \mathbf{A}^{(1)}(i_1, j_1) \cdots \mathbf{A}^{(d)}(i_d, j_d)$.
- $b(i) = \mathbf{b}^{(1)}(i_1) \cdots \mathbf{b}^{(d)}(i_d)$.

Seek the solution in the same format:

- $x(i) = \mathbf{x}^{(1)}(i_1) \cdots \mathbf{x}^{(d)}(i_d)$.

# Linear Systems in TT Format

**Central Question**

How to solve $Ax = b$ if Krylov solvers become too expensive?

Data are given in TT format:

- $A(i,j) = \mathbf{A}^{(1)}(i_1, j_1) \cdots \mathbf{A}^{(d)}(i_d, j_d)$.
- $b(i) = \mathbf{b}^{(1)}(i_1) \cdots \mathbf{b}^{(d)}(i_d)$.

Seek the solution in the same format:

- $x(i) = \mathbf{x}^{(1)}(i_1) \cdots \mathbf{x}^{(d)}(i_d)$.

Use a new block-variant of *Alternating Least Squares* in a new block TT format to overcome difficulties with indefiniteness of KKT system matrix.

- If $A = A^\top > 0$: minimize $J(x) = x^\top A x - 2x^\top b$.

*Alternating Least Squares (ALS):*

- replace $\min_x J(x)$ by iteration — size $n^d$
- `for` $k = 1, \ldots, d$, — size $r^2 n$

  solve $\min_{\mathbf{x}^{(k)}} J\left(\mathbf{x}^{(1)}(i_1) \cdots \mathbf{x}^{(k)}(i_k) \cdots \mathbf{x}^{(d)}(i_d)\right)$.

  (all other blocks are fixed)

1. $\hat{\mathbf{x}}^{(1)} = \arg\min_{\mathbf{x}^{(1)}} J\left(\mathbf{x}^{(1)}(i_1)\mathbf{x}^{(2)}(i_2)\mathbf{x}^{(3)}(i_3)\right)$

1. $\hat{\mathbf{x}}^{(1)} = \arg\min_{\mathbf{x}^{(1)}} J\left(\mathbf{x}^{(1)}(i_1)\mathbf{x}^{(2)}(i_2)\mathbf{x}^{(3)}(i_3)\right)$
2. $\hat{\mathbf{x}}^{(2)} = \arg\min_{\mathbf{x}^{(2)}} J\left(\hat{\mathbf{x}}^{(1)}(i_1)\mathbf{x}^{(2)}(i_2)\mathbf{x}^{(3)}(i_3)\right)$

# ALS for $d = 3$

1. $\hat{\mathbf{x}}^{(1)} = \arg\min_{\mathbf{x}^{(1)}} J\left(\mathbf{x}^{(1)}(i_1)\mathbf{x}^{(2)}(i_2)\mathbf{x}^{(3)}(i_3)\right)$
2. $\hat{\mathbf{x}}^{(2)} = \arg\min_{\mathbf{x}^{(2)}} J\left(\hat{\mathbf{x}}^{(1)}(i_1)\mathbf{x}^{(2)}(i_2)\mathbf{x}^{(3)}(i_3)\right)$
3. $\hat{\mathbf{x}}^{(3)} = \arg\min_{\mathbf{x}^{(3)}} J\left(\hat{\mathbf{x}}^{(1)}(i_1)\hat{\mathbf{x}}^{(2)}(i_2)\mathbf{x}^{(3)}(i_3)\right)$

# ALS for $d = 3$

1. $\hat{\mathbf{x}}^{(1)} = \arg\min_{\mathbf{x}^{(1)}} J\left(\mathbf{x}^{(1)}(i_1)\mathbf{x}^{(2)}(i_2)\mathbf{x}^{(3)}(i_3)\right)$
2. $\hat{\mathbf{x}}^{(2)} = \arg\min_{\mathbf{x}^{(2)}} J\left(\hat{\mathbf{x}}^{(1)}(i_1)\mathbf{x}^{(2)}(i_2)\mathbf{x}^{(3)}(i_3)\right)$
3. $\hat{\mathbf{x}}^{(3)} = \arg\min_{\mathbf{x}^{(3)}} J\left(\hat{\mathbf{x}}^{(1)}(i_1)\hat{\mathbf{x}}^{(2)}(i_2)\mathbf{x}^{(3)}(i_3)\right)$
4. $\mathbf{x}^{(2)} = \arg\min_{\mathbf{x}^{(2)}} J\left(\hat{\mathbf{x}}^{(1)}(i_1)\mathbf{x}^{(2)}(i_2)\hat{\mathbf{x}}^{(3)}(i_3)\right)$

# ALS for $d = 3$

1. $\hat{\mathbf{x}}^{(1)} = \arg\min_{\mathbf{x}^{(1)}} J\left(\mathbf{x}^{(1)}(i_1)\mathbf{x}^{(2)}(i_2)\mathbf{x}^{(3)}(i_3)\right)$
2. $\hat{\mathbf{x}}^{(2)} = \arg\min_{\mathbf{x}^{(2)}} J\left(\hat{\mathbf{x}}^{(1)}(i_1)\mathbf{x}^{(2)}(i_2)\mathbf{x}^{(3)}(i_3)\right)$
3. $\hat{\mathbf{x}}^{(3)} = \arg\min_{\mathbf{x}^{(3)}} J\left(\hat{\mathbf{x}}^{(1)}(i_1)\hat{\mathbf{x}}^{(2)}(i_2)\mathbf{x}^{(3)}(i_3)\right)$
4. $\mathbf{x}^{(2)} = \arg\min_{\mathbf{x}^{(2)}} J\left(\hat{\mathbf{x}}^{(1)}(i_1)\mathbf{x}^{(2)}(i_2)\hat{\mathbf{x}}^{(3)}(i_3)\right)$
5. repeat 1.–4. until convergence

# ALS = Projection method

If we differentiate $J$ w.r.t. TT blocks, we see that...

- ...each step means solving a *Galerkin linear system*

$$\left(X_{\neq k}^{\top} A X_{\neq k}\right) \hat{x}^{(k)} = \left(X_{\neq k}^{\top} b\right) \in \mathbb{R}^{nr^2}.$$

- $X_{\neq k} = \underbrace{\mathrm{TT}\left(\hat{\mathbf{x}}^{(1)} \dots \hat{\mathbf{x}}^{(k-1)}\right)}_{n^{k-1} \times r_{k-1}} \otimes \underbrace{I}_{n \times n} \otimes \underbrace{\mathrm{TT}\left(\mathbf{x}^{(k+1)} \dots \mathbf{x}^{(d)}\right)}_{n^{d-k} \times r_k}.$

# ALS = Projection method

If we differentiate $J$ w.r.t. TT blocks, we see that...

- ...each step means solving a *Galerkin linear system*

$$\left(X_{\neq k}^{\top} A X_{\neq k}\right) \hat{x}^{(k)} = \left(X_{\neq k}^{\top} b\right) \in \mathbb{R}^{nr^2}.$$

- $X_{\neq k} = \underbrace{\mathrm{TT}\left(\hat{\mathbf{x}}^{(1)} \dots \hat{\mathbf{x}}^{(k-1)}\right)}_{n^{k-1} \times r_{k-1}} \otimes \underbrace{I}_{n \times n} \otimes \underbrace{\mathrm{TT}\left(\mathbf{x}^{(k+1)} \dots \mathbf{x}^{(d)}\right)}_{n^{d-k} \times r_k}.$

## Properties of ALS include:

+ Effectively 1D complexity in a prescribed format.
− Tensor format (ranks) is fixed and cannot be adapted.
− Convergence may be very slow, stagnation is likely.

- Density Matrix Renormalization Group (DMRG) [White '92]
  – updates *two* blocks $\mathbf{x}^{(k)}\mathbf{x}^{(k+1)}$ *simultaneously*.
- Alternating Minimal Energy (AMEn) [Dolgov/Savostyanov '13]
  – *augments* $\mathbf{x}^{(k)}$ by a TT block of the *residual* $\mathbf{z}^{(k)}$.

# ALS: Getting rid of "$-$"

- Density Matrix Renormalization Group (DMRG) [White '92]
  – updates *two* blocks $\mathbf{x}^{(k)}\mathbf{x}^{(k+1)}$ *simultaneously*.
- Alternating Minimal Energy (AMEn) [Dolgov/Savostyanov '13]
  – *augments* $\mathbf{x}^{(k)}$ by a TT block of the *residual* $\mathbf{z}^{(k)}$.

**But. . . , what about saddle point systems $A$?**

- Recall our KKT system:

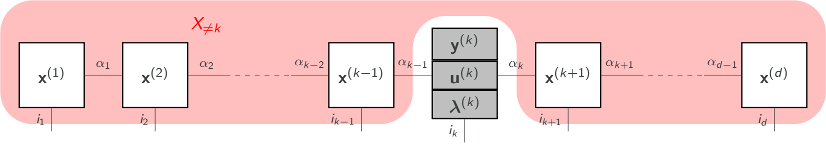
$$\underbrace{\begin{bmatrix} M_y & 0 & L^* \\ 0 & M_u & N^\top \\ L & N & 0 \end{bmatrix}}_{A} \begin{bmatrix} y \\ u \\ \lambda \end{bmatrix} = \begin{bmatrix} f \\ 0 \\ g \end{bmatrix}.$$

- The whole matrix is indefinite $\Rightarrow X_{\neq k}^\top A X_{\neq k}$ can be degenerate.

# Block ALS [B./Dolgov/Onwunta/Stoll 2016, 2017]

- Work-around: Block TT representation

$$\begin{bmatrix} y \\ u \\ \lambda \end{bmatrix} = \mathbf{x}^{(1)}_{\alpha_1} \otimes \cdots \otimes \begin{bmatrix} \mathbf{y}^{(k)}_{\alpha_{k-1},\alpha_k} \\ \mathbf{u}^{(k)}_{\alpha_{k-1},\alpha_k} \\ \boldsymbol{\lambda}^{(k)}_{\alpha_{k-1},\alpha_k} \end{bmatrix} \otimes \cdots \otimes \mathbf{x}^{(d)}_{\alpha_{d-1}}.$$

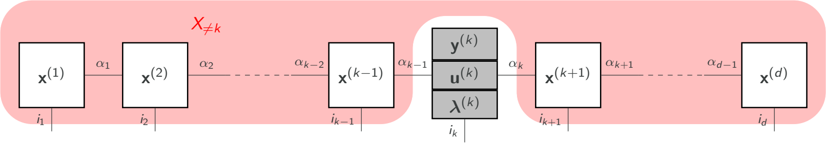



- $X_{\neq k}$ is the *same* for $y, u, \lambda$.

# Block ALS [B./Dolgov/Onwunta/Stoll 2016, 2017]

- Work-around: Block TT representation

$$\begin{bmatrix} y \\ u \\ \lambda \end{bmatrix} = \mathbf{x}^{(1)}_{\alpha_1} \otimes \cdots \otimes \begin{bmatrix} \mathbf{y}^{(k)}_{\alpha_{k-1},\alpha_k} \\ \mathbf{u}^{(k)}_{\alpha_{k-1},\alpha_k} \\ \boldsymbol{\lambda}^{(k)}_{\alpha_{k-1},\alpha_k} \end{bmatrix} \otimes \cdots \otimes \mathbf{x}^{(d)}_{\alpha_{d-1}}.$$




- $X_{\neq k}$ is the *same* for $y, u, \lambda$.
- Project each *submatrix*:

$$\begin{bmatrix} \hat{M}_y & 0 & \hat{L}^* \\ 0 & \hat{M}_u & \hat{N}^\top \\ \hat{L} & \hat{N} & 0 \end{bmatrix} \begin{bmatrix} y^{(k)} \\ u^{(k)} \\ \lambda^{(k)} \end{bmatrix} = \begin{bmatrix} \hat{f} \\ 0 \\ \hat{g} \end{bmatrix}, \qquad \widehat{(\cdot)} = X_{\neq k}^\top (\cdot) X_{\neq k}$$

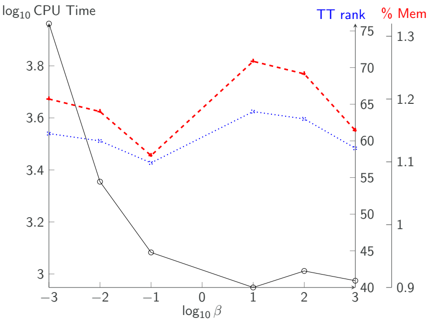
Kármán vortex street

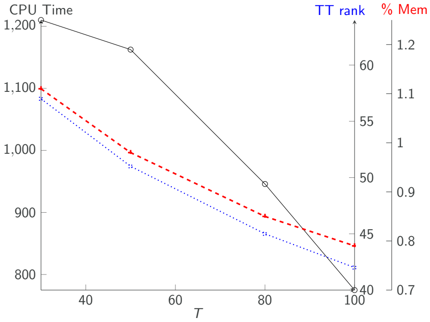
## Vary one of the default parameters:

- TT truncation tolerance $\varepsilon = 10^{-4}$,
- mean viscosity $\nu_0 = 1/20$,
- uncertainty $\nu_1 = 1/80$,
- regularization/penalty parameter $\beta = 10^{-1}$,
- number of time steps: $n_t = 2^{10}$,
- time horizon $T = 30$,
- spatial grid size $h = 1/4 \rightsquigarrow J = 2488$,
- max. degree of Legendre polynomials: $P = 8$.

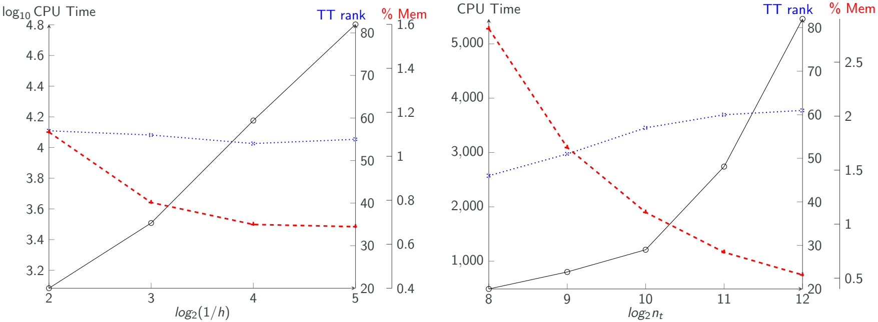
Solve projected linear systems using block-preconditioned GMRES using efficient approximation of Schur complement [B/Onwunta/Stoll 2016].

# Varying regularization $\beta$ (left) and time $T$ (right)



# Varying spatial $h$ (left) / temporal $n_t$ (right) mesh



# Varying different viscosity parameters

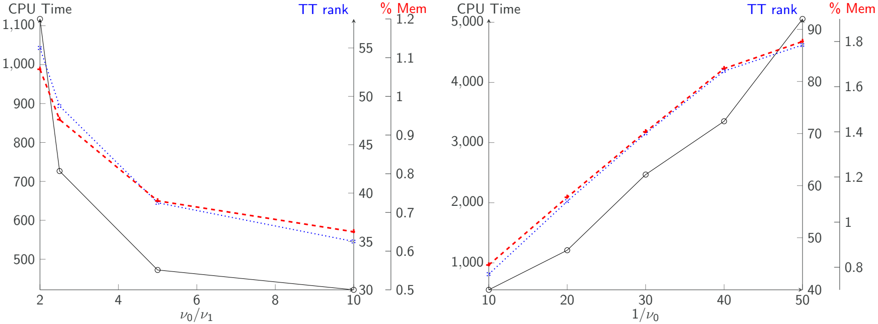
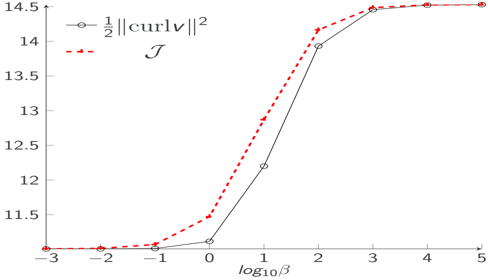
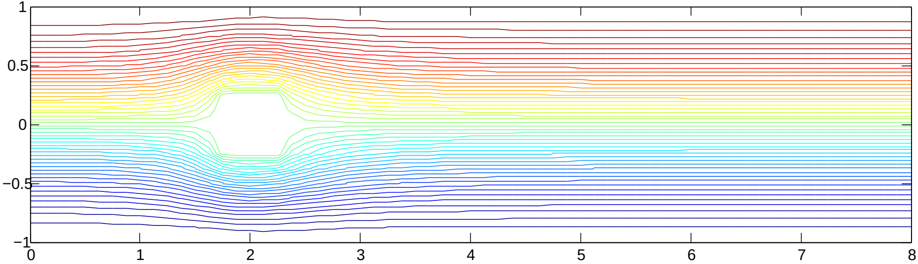



Figure: Left: $\nu_0 = 1/10$, $\nu_1$ is varied. Right: $\nu_1$ and $\nu_0$ are varied together as $\nu_1 = 0.25\nu_0$


14.5
14
13.5
13
12.5
12
11.5
$\frac{1}{2}\|\mathrm{curl}\,\boldsymbol{v}\|^2$
$\mathcal{J}$
−3 −2 −1 0 1 2 3 4 5
$log_{10}\beta$


1
0.5
0
−0.5
−1
0 1 2 3 4 5 6 7 8

# Conclusions & Outlook

- Low-rank tensor solver for unsteady heat (and Navier-Stokes) equations with uncertain viscosity.
- Similar techniques already used for ▸ 3D Stokes(-Brinkman) optimal control problems.
- With 1 stochastic parameter, the scheme reduces complexity by up to 2–3 orders of magnitude.
- To consider next:

# Conclusions & Outlook

- Low-rank tensor solver for unsteady heat (and Navier-Stokes) equations with uncertain viscosity.
- Similar techniques already used for 3D Stokes(-Brinkman) optimal control problems.
- With 1 stochastic parameter, the scheme reduces complexity by up to 2–3 orders of magnitude.
- To consider next:
  - many parameters coming from uncertain geometry or Karhunen-Loève expansion of random fields;
    Basic observation: the more parameters, the more significant is the complexity reduction w.r.t. memory — up to a factor of $10^9$ for the control problem for a backward facing step set-up.
  - HPC implementation of AMEn-like solver to deal with even larger problems.

# References


P. Benner, S. Dolgov, A. Onwunta, and M. Stoll.
Low-rank solvers for unsteady Stokes-Brinkman optimal control problem with random data.
Computer Methods in Applied Mechanics and Engineering, 304:26–54, 2016.

P. Benner, A. Onwunta, and M. Stoll.
Low rank solution of unsteady diffusion equations with stochastic coefficients.
SIAM/ASA Journal on Uncertainty Quantification, 3(1):622–649, 2015.

P. Benner, A. Onwunta, and M. Stoll.
Block-diagonal preconditioning for optimal control problems constrained by PDEs with uncertain inputs.
SIAM Journal on Matrix Analysis and Applications, 37(2):491–518, 2016.

P. Benner, S. Dolgov, A. Onwunta, and M. Stoll.
Solving optimal control problems governed by random Navier-Stokes equations using low-rank methods. arXiv:1703.06097, March 2017.

P. Benner, A. Onwunta, and M. Stoll.
On the existence and uniqueness of the solution of parabolic optimal control problems with uncertain inputs. Submitted, September 2018.

# Low-rank Techniques in Other Areas

## Low-rank techniques in stochastic eigenvalue problems

Consider eigenproblem depending on uncertain parameter(s), e.g., uncertainty in Young's modulus in elasticity problems,

$$\begin{aligned} \mathcal{A}(\omega)\varphi(\omega) &= \lambda(\omega)\varphi(\omega), \quad \omega \in \Omega, \\ \varphi(\omega)^T\varphi(\omega) &= 1. \end{aligned}$$

## Low-rank techniques in stochastic eigenvalue problems

Consider eigenproblem depending on uncertain parameter(s), e.g., uncertainty in Young's modulus in elasticity problems,

$$\begin{aligned} \mathcal{A}(\omega)\varphi(\omega) &= \lambda(\omega)\varphi(\omega), \quad \omega \in \Omega, \\ \varphi(\omega)^T\varphi(\omega) &= 1. \end{aligned}$$

Parametrization of uncertain parameter, (generalized) polynomial chaos expansion and stochastic Galerkin ansatz $\rightsquigarrow$ multi-parametric eigenvalue problem

$$\left[G_0 \otimes A_0 + \sum_{k=1}^{m} G_k \otimes A_k\right]\Phi = \left[\sum_{k=0}^{N_\xi - 1} \lambda_k (H_k \otimes I_{N_x})\right]\Phi,$$

where $(\lambda_0, \ldots, \lambda_{N_\xi - 1})$ represents the parametrization of a single stochastic eigenvalue $\lambda(\omega)$, and $\Phi = (\Phi) = \mathrm{vec}\left(\left[\varphi_0, \varphi_1, \ldots, \varphi_{N_\xi - 1}\right]\right) \in \mathbb{R}^{N_x N_\xi}$ parameterizes the corresponding stochastic eigenvector $\varphi(\omega)$.

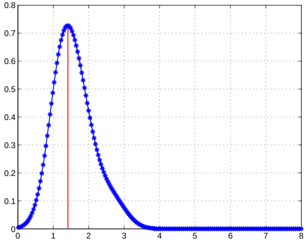
## Low-rank techniques in stochastic eigenvalue problems

Consider eigenproblem depending on uncertain parameter(s), e.g., uncertainty in Young's modulus in elasticity problems,

$$\begin{aligned} \mathcal{A}(\omega)\varphi(\omega) &= \lambda(\omega)\varphi(\omega), \quad \omega \in \Omega, \\ \varphi(\omega)^T\varphi(\omega) &= 1. \end{aligned}$$

Developed an inexact Krylov-Newton solver for the multi-parametric eigenvalue problem, using again low-rank techniques.

**Example:** Probability density of $\lambda_2(\omega)$, $\langle\lambda_2\rangle = 1.412$, for uncertain linear elasticity problem, $\sigma_a = 0.1$.



## Low-rank techniques in stochastic eigenvalue problems

Consider eigenproblem depending on uncertain parameter(s), e.g., uncertainty in Young's modulus in elasticity problems,

$$\begin{aligned} \mathcal{A}(\omega)\varphi(\omega) &= \lambda(\omega)\varphi(\omega), \quad \omega \in \Omega, \\ \varphi(\omega)^T\varphi(\omega) &= 1. \end{aligned}$$

P. Benner, A. Onwunta, and M. Stoll.
A low-rank inexact Newton-Krylov method for stochastic eigenvalue problems.
Computational Methods in Applied Mathematics (CMAM), 2018 (online).

# Low-rank Techniques in Other Areas

## Low-rank techniques in Bayesian inverse problems

Bayesian inference problem: determine unknown parameters $U$ from observations $y_{\text{obs}}$, assuming additive noise:

$$Y = f(U) + E.$$

# Low-rank Techniques in Other Areas

## Low-rank techniques in Bayesian inverse problems

Bayesian inference problem: determine unknown parameters $U$ from observations $y_{\text{obs}}$, assuming additive noise:

$$Y = f(U) + E.$$

Want posterior probability density function $\pi_{\text{post}}$, apply Bayes' theorem ... $\rightsquigarrow$

$$\pi_{\text{post}} \propto \pi_{\text{prior}}(u)\pi_{\text{noise}}(y_{\text{obs}} - f(u))$$

Want (diagonal of) posterior covariance matrix $\Gamma_{\text{post}}$:

$$\Gamma_{\text{post}} = \left( A^T \Gamma_{\text{noise}}^{-1} A + \Gamma_{\text{prior}}^{-1} \right)^{-1}.$$

For a discretized partial differential operator $A$, this formula is numerically infeasible.

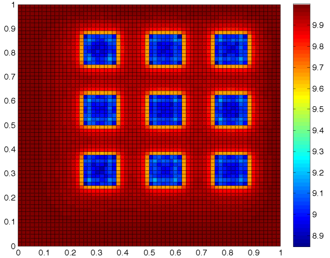
## Low-rank techniques in Bayesian inverse problems

Bayesian inference problem: determine unknown parameters $U$ from observations $y_{\text{obs}}$, assuming additive noise:

$$Y = f(U) + E.$$

Developed a low-rank (TT format) Lanczos technique to compute dominant eigenvectors, and to obtain low-rank approximation to $\Gamma_{\text{post}}$.

**Example:** determine initial condition for heat flow problem, 9 sensors distributed in the domain, $64 \times 64$ mesh.



# Low-rank Techniques in Other Areas

## Low-rank techniques in Bayesian inverse problems

Bayesian inference problem: determine unknown parameters $U$ from observations $y_{\text{obs}}$, assuming additive noise:

$$Y = f(U) + E.$$

P. Benner, Y. Qiu, and M. Stoll.

Low-rank eigenvector compression of posterior covariance matrices for linear Gaussian inverse problems.

SIAM/ASA Journal on Uncertainty Quantification, 6(2):965–989, 2018.

## Low-rank techniques in quantum chemistry

- Using low-rank techniques in numerical algorithms for grid-based solution of Bethe-Salpeter equation for excitation energies,

  PB, V. Khoromskaia, B.N. Khoromskij, Mol. Phys. 114(7–8):1148–1161, 2016,

  additional acceleration using TT format for eigenvctors,

  PB, S. Dolgov, V. Khoromskaia, B.N. Khoromskij, JCP 334:221–239, 2017,

  application to computation of density-of-states and absorption spectra,

  PB, V. Khoromskaia, B.N. Khoromskij, C. Yang, arXiv:1801.03852, January 2018.

## Low-rank techniques in quantum chemistry

- Using low-rank techniques in numerical algorithms for grid-based solution of Bethe-Salpeter equation for excitation energies,

  PB, V. Khoromskaia, B.N. Khoromskij, Mol. Phys. 114(7–8):1148–1161, 2016,

  additional acceleration using TT format for eigenvctors,

  PB, S. Dolgov, V. Khoromskaia, B.N. Khoromskij, JCP 334:221–239, 2017,

  application to computation of density-of-states and absorption spectra,

  PB, V. Khoromskaia, B.N. Khoromskij, C. Yang, arXiv:1801.03852, January 2018.

- Introduced the range-separated tensor format to efficiently deal with short and long range potentials independently in many-particle modeling,

  PB, V. Khoromskaia, B.N. Khoromskij, SISC 40(2):A1034–A1062, 2017,

  and applied it to decomposing the solution of the (non)linear Poisson-Boltzmann equation for calculating the electrostatic potential and energies of ionic solvated biomolecules,

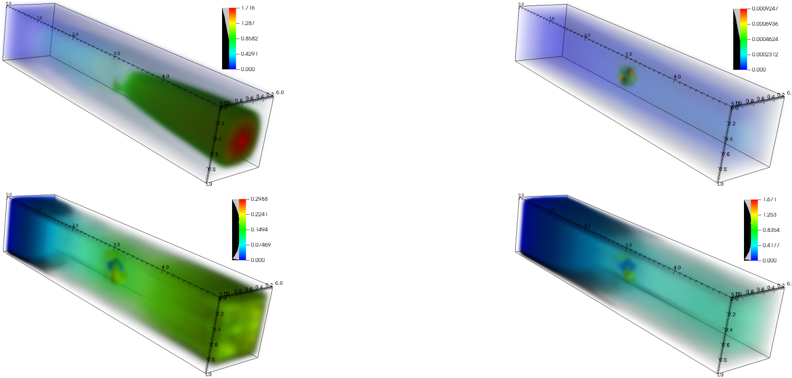
  PB, V. Khoromskaia, B.N. Khoromskij, C. Kweyu, M. Stein, work in progress.

# 3D Stokes-Brinkman control problem

Mean | Standard deviation

State

Control




- Full size: $n_{\mathbf{x}} n_{\xi} n_t \approx 3 \cdot 10^9$. Reduction: $\dfrac{\texttt{mem}(TT)}{\texttt{mem}(\mathit{full})} = 0.002$.

return