MAX PLANCK INSTITUTE
FOR DYNAMICS OF COMPLEX
TECHNICAL SYSTEMS
MAGDEBURG

CSC
COMPUTATIONAL METHODS IN
SYSTEMS AND CONTROL THEORY

PDE-CONSTRAINED OPTIMIZATION UNDER UNCERTAINTY USING LOW-RANK METHODS

Peter Benner

Joint work with Sergey Dolgov (U Bath), Akwum Onwunta and Martin Stoll (both MPI DCTS).

August 26, 2017

Topical Lecture
S 15: Uncertainty Quantification
GAMM Jahrestagung Ilmenau@Weimar
March 06–10, 2017

Overview

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 uncertainties.

PDEs with stochastic coefficients for UQ

- Physical, biological, chemical, etc. processes involve uncertainties.
- Models of these processes should account for 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 uncertainties.
- PDEs governing the processes can involve uncertain coefficients, or uncertain sources, or uncertain geometry.
- Uncertain parameters modeled as stochastic variables $\rightsquigarrow$ random PDEs with uncertain inputs.

Introduction

PDEs with stochastic coefficients for UQ

- Physical, biological, chemical, etc. processes involve uncertainties.
- Models of these processes should account for uncertainties.
- PDEs governing the processes can involve uncertain coefficients, or uncertain sources, or uncertain geometry.
- Uncertain parameters modeled as stochastic variables $\rightsquigarrow$ random PDEs with uncertain inputs.

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 uncertainties.
- PDEs governing the processes can involve uncertain coefficients, or uncertain sources, or uncertain geometry.
- Uncertain parameters modeled as stochastic variables $\rightsquigarrow$ random PDEs with uncertain inputs.

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 uncertainties.
- PDEs governing the processes can involve uncertain coefficients, or uncertain sources, or uncertain geometry.
- Uncertain parameters modeled as stochastic variables $\rightsquigarrow$ random PDEs with uncertain inputs.

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 $B \approx b_1 b_2^T$.

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 $x = \operatorname{vec}(X)$ and $b = \operatorname{vec}(B)$ 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 $h \rightarrow \frac{h}{2}$ 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[1]. These essentially require matrix-vector multiplication and vector computations.

[1] Recent work by H. Elman analyzes multigrid solver in this context.

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[1]. These essentially require matrix-vector multiplication and vector computations.
- Hence, $\mathcal{A}\mathrm{vec}(X_k) = \mathcal{A}\mathrm{vec}\left(V_k W_k^T\right) = \mathrm{vec}\left(\left[\, AV_k,\ V_k \,\right]\left[\, W_k,\ AW_k \,\right]^T\right)$

[1] Recent work by H. Elman analyzes multigrid solver in this context.

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

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

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[1]. 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, ...].

[1] Recent work by H. Elman analyzes multigrid solver in this context.

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 nonlinear PDE with uncertain coefficient(s).
- The state y and control u are random fields.
- The cost functional $\mathcal{J}$ is a real-valued 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 (Krylov) 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 $h \rightarrow \frac{h}{2}$ by factor 2^d.

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

Goal of this talk

Apply low-rank (Krylov) 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 (Krylov) 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 ≈ 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 (Krylov) 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 ≈ 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

- 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 $\mathbf{x}\in\mathcal{D}$,
- $a(\mathbf{x},\omega)$ is assumed to be uniformly positive in $\mathcal{D}\times\Omega$.

Discretization

We discretize and then optimize the stochastic control problem.

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

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

- 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 (λ_i, φ_i) are the dominant eigenpairs of C_a.

- Doob-Dynkin Lemma admits 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.

Overview of UQ techniques

- **Monte Carlo Sampling**
  Given a sample $\{\omega_i\}_{i=1}^{M} \in \Omega$, we estimate desired statistical quantities using the law of large numbers.
  - Pros: Simple, code reusability, etc.
  - Cons: Slow convergence $\mathcal{O}(1/\sqrt{M})$.

Overview of UQ techniques

- **Monte Carlo Sampling**
  Given a sample $\{\omega_i\}_{i=1}^{M} \in \Omega$, we estimate desired statistical quantities using the law of large numbers.
  - Pros: Simple, code reusability, etc.
  - Cons: Slow convergence $\mathcal{O}(1/\sqrt{M})$.
- **Parametric**
  Expand $y(\mathbf{x}, \xi) = \sum_i y_i(\mathbf{x}) H_i(\xi)$.

Overview of UQ techniques

- **Monte Carlo Sampling**
  Given a sample $\{\omega_i\}_{i=1}^{M} \in \Omega$, we estimate desired statistical quantities using the law of large numbers.
  - Pros: Simple, code reusability, etc.
  - Cons: Slow convergence $\mathcal{O}(1/\sqrt{M})$.
- **Parametric**
  Expand $y(\mathbf{x}, \xi) = \sum_i y_i(\mathbf{x}) H_i(\xi)$.
  - Stochastic collocation.
    Compute y_i for a set of interpolation points ξ_i, then connect the realizations with Lagrangian basis functions $H_i := L_i$.

Overview of UQ techniques

- **Monte Carlo Sampling**
  Given a sample $\{\omega_i\}_{i=1}^{M} \in \Omega$, we estimate desired statistical quantities using the law of large numbers.
  - Pros: Simple, code reusability, etc.
  - Cons: Slow convergence $\mathcal{O}(1/\sqrt{M})$.
- **Parametric**
  Expand $y(\mathbf{x}, \xi) = \sum_i y_i(\mathbf{x}) H_i(\xi)$.
  - Stochastic collocation.
    Compute y_i for a set of interpolation points ξ_i, then connect the realizations with Lagrangian basis functions $H_i := L_i$.
  - Stochastic Galerkin (Generalized Polynomial Chaos).
    Compute y_i projecting the equation onto a subspace spanned by orthogonal polynomials $H_i := \psi_i$.
    - ξ are uniform random variables $\rightarrow$ ψ_i Legendre polynomials.
    - ξ are Gaussian random variables $\rightarrow$ ψ_i Hermite polynomials.

The fully discretized problem

First order conditions of the discrete heat control problem are given by the **KKT system**

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

$$\mathcal{M}_1 = D \otimes G_\alpha \otimes M = D \otimes \mathcal{M}_\alpha,$$

$$\mathcal{K}_t = (I_{n_t} \otimes \mathcal{L}) + (C \otimes \mathcal{M}) = I_{n_t} \otimes \left[\sum_{i=0}^{N} G_i \otimes \widehat{K}_i\right] + (C \otimes G_0 \otimes M),$$

$$\mathcal{N} = I_{n_t} \otimes G_0 \otimes M, \quad \mathcal{M}_2 = D \otimes G_0 \otimes M$$

and

$$\begin{cases} G_0 = \operatorname{diag}\left(\langle\psi_0^2\rangle, \langle\psi_1^2\rangle, \ldots, \langle\psi_{P-1}^2\rangle\right), \\ G_i(j,k) = \langle\xi_i\psi_j\psi_k\rangle, \quad i = 1, \ldots, N, \end{cases}$$

with ψ the orthogonal (Legendre, Hermite, ...) polynomials and K_i are stiffness matrices involving terms from the KLE.

Solving the KKT System

This system is a saddle point system

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

Lots of pioneering work by ELMAN, ERNST, ULLMANN, POWELL, SILVESTER, ...

Solving the KKT System

This system is a saddle point system

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

Lots of pioneering work by Elman, Ernst, Ullmann, Powell, Silvester, ...

Theorem ([B./Onwunta/Stoll '16])

Let $\alpha \in [0, +\infty)$. Then, the eigenvalues of $S_2^{-1}S$ satisfy

$$\lambda(S_2^{-1}S) \subset \left[\frac{1}{2(1+\alpha)}, 1\right), \quad \forall \alpha < \left(\frac{\sqrt{\kappa(\mathcal{K})}+1}{\sqrt{\kappa(\mathcal{K})}-1}\right)^2 - 1,$$

where $\mathcal{K} = \sum_{i=0}^{N} G_i \otimes K_i$ and

$$S_2 = \frac{1}{\tau}\left(\mathcal{K}_t + \tau\sqrt{\frac{1+\alpha}{\beta}}\mathcal{N}\right)\mathcal{M}_1^{-1}\left(\mathcal{K}_t + \tau\sqrt{\frac{1+\alpha}{\beta}}\mathcal{N}\right)^T.$$

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

Use tensor train format and represent the tensor objects as

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

and

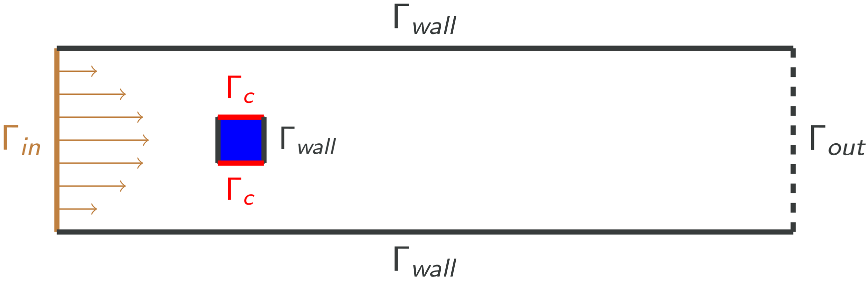
$$A(i_1\cdots i_d, j_1\cdots j_d) \approx \sum_{\beta_1\ldots\beta_{d-1}=1}^{R_1\ldots R_{d-1}} \mathbf{A}^{(1)}_{\beta_1}(i_1,j_1)\mathbf{A}^{(2)}_{\beta_1,\beta_2}(i_2,j_2)\cdots\mathbf{A}^{(d)}_{\beta_{d-1}}(i_d,j_d).$$

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)

Unsteady Navier-Stokes Equations

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



- 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 $\nu := \nu(\omega)$.

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

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

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: Setting (cf. [Powell/Silvester '12])

We assume

- $\nu(\omega) = \nu_0 + \nu_1 \xi(\omega), \;\; \nu_0, \nu_1 \in \mathbb{R}^+, \;\; \xi \sim \mathcal{U}(-1, 1).$
- $\mathbb{P}\left(\omega \in \Omega : \nu(\omega) \in [\nu_{\text{min}}, \nu_{\text{max}}]\right) = 1$, for some $0 < \nu_{\text{min}} < \nu_{\text{max}} < +\infty.$
- $\Rightarrow$ 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

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

adjoint equation

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

gradient equation

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

OTD Strategy and Picard (Oseen) Iteration ⇝

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

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

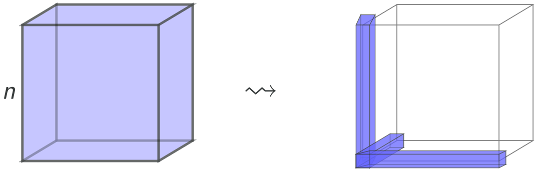
Each of the block matrices in A is of the form

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

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!

Separation of variables and low-rank approximation



- 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:

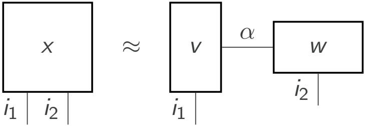
- Store and manipulate x — $\mathcal{O}(dn)$ cost *instead of* $\mathcal{O}(n^d)$.
- Solve equations $Ax = b$ — $\mathcal{O}(dn^2)$ cost *instead of* $\mathcal{O}(n^{2d})$.

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.

Data Compression in Higher Dimensions:
Tensor Trains

- Matrix Product States/Tensor Train (TT) format
  [Wilson '75, White '93, Verstraete '04, Oseledets '09/'11]:
  For indices

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

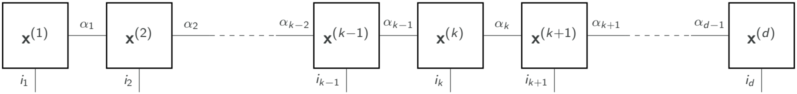
  the TT format can be expressed as

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

  or

  $$x(\overline{i_1 \dots i_d}) = \mathbf{x}^{(1)}(i_1) \cdots \mathbf{x}^{(d)}(i_d), \qquad \mathbf{x}^{(k)}(i_k) \in \mathbb{R}^{r_{k-1} \times r_k}.$$

  or



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

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

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

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

Central Question

How to solve $Ax = b$?

Central Question

How to solve $Ax = b$?

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 $Ax = b$?

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.

Linear Systems in TT Format

Central Question

How to solve $Ax = b$?

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.

Alternating Least Squares Method
[Kroonenberg '80, De Lathauwer '00, Schneider '12]

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

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

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

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*

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

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

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.

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* $\mathbf{z}^{(k)}$.

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?

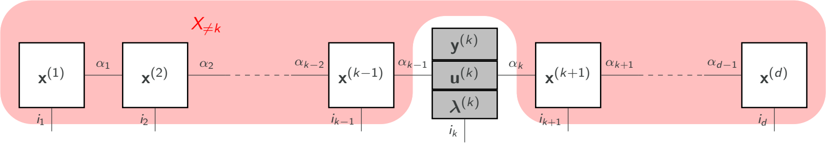
- Recall our KKT system:

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

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

- Work-around: Block TT representation

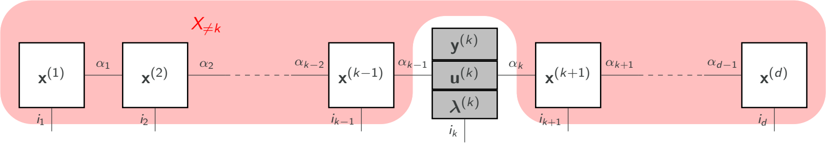
$$\begin{bmatrix} y \\ u \\ \lambda \end{bmatrix} = \mathbf{x}^{(1)}_{\alpha_1} \otimes \cdots \otimes \begin{bmatrix} \mathbf{y}^{(k)}_{\alpha_{k-1},\alpha_k} \\ \mathbf{u}^{(k)}_{\alpha_{k-1},\alpha_k} \\ \boldsymbol{\lambda}^{(k)}_{\alpha_{k-1},\alpha_k} \end{bmatrix} \otimes \cdots \otimes \mathbf{x}^{(d)}_{\alpha_{d-1}}.$$



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

- 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, λ.
- Project each *submatrix*:

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

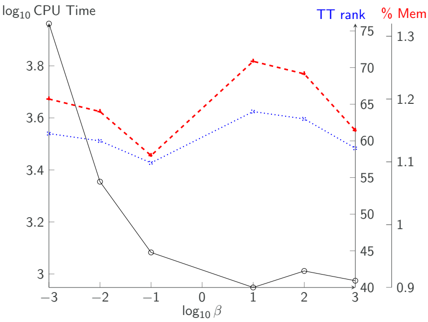
Numerical experiments

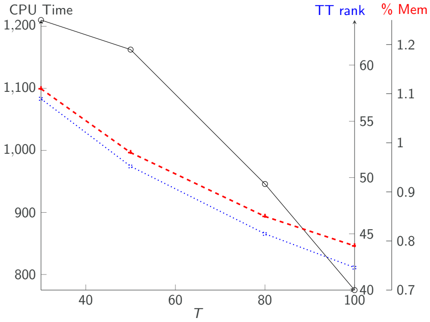
Vary one of the default parameters:

- 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: $P = 8$.

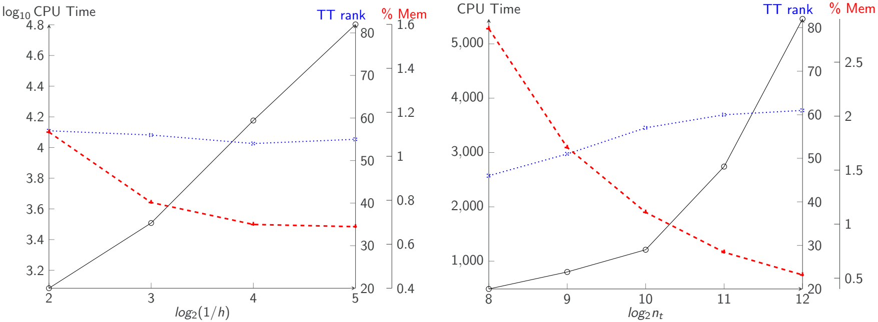
Solve projected linear systems using block-preconditioned GMRES using efficient approximation of Schur complement [B/Dolgov/Onwunta/Stoll '16a].

Varying regularization β (left) and time T (right)

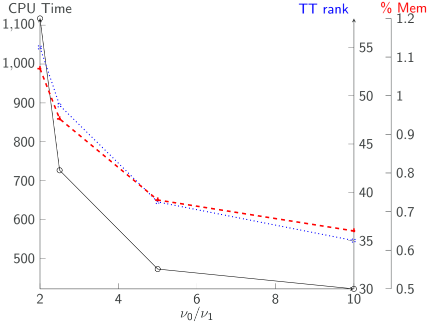




Varying spatial h (left) / temporal n_t (right) mesh



Varying different viscosity parameters



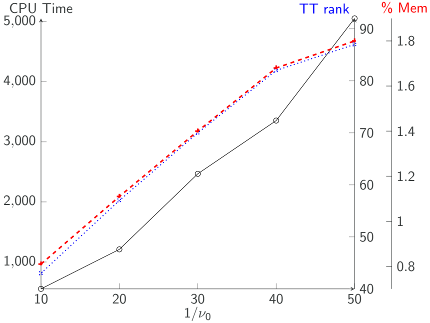
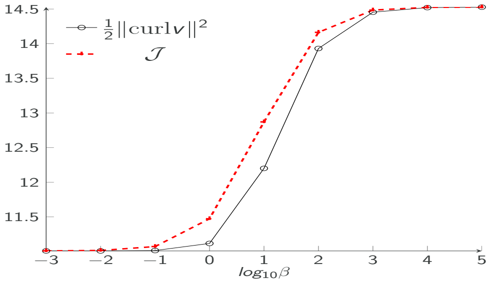
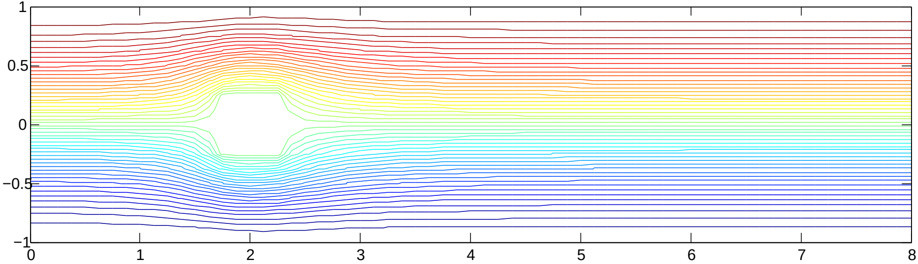


Figure: Left: $\nu_0 = 1/10$, ν_1 is varied. Right: ν_1 and ν_0 are varied together as $\nu_1 = 0.25\nu_0$

Cost functional, squared vorticity (top) and streamlines (bottom)





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.
- Adapted AMEn (TT) solver to saddle point systems.
- 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.
- Adapted AMEn (TT) solver to saddle point systems.
- 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;
    Initial results: 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.
  - exploit multicore technology (need efficient parallelization of AMEn).

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.

C.E. Powell and D.J. Silvester.
Preconditioning steady-state Navier–Stokes equations with random data.
SIAM Journal on Scientific Computing, 34(5):A2482–A2506, 2012.

E. Rosseel and G.N. Wells.
Optimal control with stochastic PDE constraints and uncertain controls.
Computer Methods in Applied Mechanics and Engineering, 213–216:152–167, 2012.

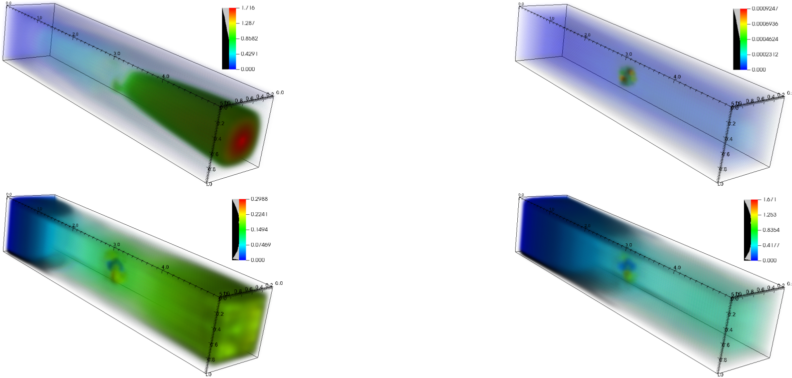
M. Stoll and T. Breiten.
A low-rank in time approach to PDE-constrained optimization.
SIAM Journal on Scientific Computing, 37(1):B1–B29, 2015.

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: $\frac{\mathtt{mem}(TT)}{\mathtt{mem}(full)} = 0.002$.

return