

# Solving optimal control problems governed by random Navier-Stokes equations using low-rank methods

Peter Benner\*, Sergey Dolgov<sup>†</sup>, Akwum Onwunta\*, Martin Stoll\*

March 17, 2017

## Abstract

Many problems in computational science and engineering are simultaneously characterized by the following challenging issues: uncertainty, nonlinearity, nonstationarity and high dimensionality. Existing numerical techniques for such models would typically require considerable computational and storage resources. This is the case, for instance, for an optimization problem governed by time-dependent Navier-Stokes equations with uncertain inputs. In particular, the stochastic Galerkin finite element method often leads to a prohibitively high dimensional saddle-point system with tensor product structure. In this paper, we approximate the solution by the low-rank Tensor Train decomposition, and present a numerically efficient algorithm to solve the optimality equations directly in the low-rank representation. We show that the solution of the vorticity minimization problem with a distributed control admits a representation with ranks that depend modestly on model and discretization parameters even for high Reynolds numbers. For lower Reynolds numbers this is also the case for a boundary control. This opens the way for a reduced-order modeling of the stochastic optimal flow control with a moderate cost at all stages.

*Keywords:* Stochastic Galerkin system, iterative methods, PDE-constrained optimization, saddle-point system, tensor train format, low-rank solution, preconditioning, Schur complement.

## 1 Introduction

We consider the numerical simulation of optimization problems constrained by partial differential equations (PDEs). This class of problems can be computationally challenging. This is particularly so if the constraints are time-dependent PDEs since time-stepping methods

---

<sup>†</sup>University of Bath, Department of Mathematical Sciences, Claverton Down, BA2 7AY Bath, United Kingdom (S.Dolgov@bath.ac.uk). S. Dolgov gratefully acknowledges funding by the EPSRC Fellowship EP/M019004/1.

\*Computational Methods in Systems and Control Theory, Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, 39106 Magdeburg, Germany. *Email addresses:* {benner,onwunta,stollm}@mpi-magdeburg.mpg.de

quickly reach their limitations due to the enormous demand for storage [44]. The computational complexity associated with these problems is further increased when the constraints are nonlinear [13] and contain (up to countably many) parametric or uncertain inputs [5, 8]. This is the core problem of this paper. A viable solution approach to optimization problems with stochastic constraints employs the spectral stochastic Galerkin finite element method (SGFEM). However, this intrusive approach<sup>1</sup> leads to the so-called *curse of dimensionality*, in the sense that it results in prohibitively high dimensional linear systems with tensor product structure [5, 7, 40].

It is worth pursuing computationally efficient ways to simulate optimization problems with stochastic constraints using SGFEMs since the Galerkin approximation yields the best approximation with respect to the energy norm, as well as a favorable framework for error estimation [10]. In order to cope with the curse of dimensionality we exploit the underlying mathematical structure of the discretized optimality system. We develop a low-rank technique based on recent advances in numerical tensor methods [22, 30] for efficient solution of an optimization problem governed by nonlinear PDEs with random coefficients. More specifically, we numerically simulate an Optimization Problem constrained by time-dependent Navier-Stokes equations with random coefficients<sup>2</sup> (OPNS). Our aim in this paper is to lift the curse of dimensionality inherent in the OPNS and allow for efficient simulations of the model on not much more than an average desktop computer. Such simulations would enhance the understanding of the underlying physical model as the computed data can then be used for the quantification of the statistics of the system response.

Alternative approaches to tackle optimization problems with stochastic constraints include stochastic collocation schemes [11, 31, 46], as well as Monte Carlo methods [1]. These methods are essentially sampling-based and non-intrusive. However, for optimization problems, the SGFEM exhibits superior performance compared to the stochastic collocation method [40]. This is due to the fact that, unlike SGFEM, the non-intrusivity property of the stochastic collocation method is lost when moments of the state variable appear in the cost functional, or when the control is a deterministic function. On the other hand, Monte Carlo methods are relatively straightforward to implement. However, they generally converge rather slowly and do not exploit the regularity with respect to the parameters that the solution might have [46].

The rest of the paper is organised as follows. First, we present in Section 2 the PDE-constrained optimization that we would like to solve, as well as the necessary mathematical concepts and notation on which we shall rely in the rest of our discussion. Next, we proceed to Section 3 to discuss the SGFEM discretization of the problem. Section 4 presents our low-rank iterative solver and preconditioner which we use to tackle the high-dimensional saddle point systems arising from the SGFEM discretization of the optimization problem. Finally, in Section 5, we present numerical experiments to illustrate the performance of the low-rank approach.

---

<sup>1</sup>Generally speaking, SGFEM techniques are intrusive in the sense that the codes for the associated deterministic problems cannot be directly reused. These methods are mainly non-ensemble-based methods and require the solution of discrete systems that couple all spatial and probabilistic degrees of freedom.

<sup>2</sup>We remark here that our approach can be easily generalized to other nonlinear optimization problems with stochastic constraints as well.

## 2 Problem statement and mathematical description

Let  $(\Omega, \mathcal{F}, \mathbb{P})$  be a complete probability space, where  $\Omega$  is the set of outcomes,  $\mathcal{F} \subseteq 2^\Omega$  is the  $\sigma$ -algebra of events, and  $\mathbb{P} : \mathcal{F} \rightarrow [0, 1]$  is an appropriate probability measure. Let  $\mathcal{D} \subset \mathbb{R}^d$  with  $d \in \{1, 2, 3\}$ , be a bounded spatial domain with a piecewise Lipschitz boundary  $\Upsilon$ . Moreover, we consider a finite time interval  $[0, T]$ . A random field  $z : \Omega \times \mathcal{D} \rightarrow \mathbb{R}$ , means that  $z(\cdot, \mathbf{x})$  is a random variable defined on  $(\Omega, \mathcal{F}, \mathbb{P})$  for each  $\mathbf{x} \in \mathcal{D}$ . We assume that  $z$  belongs to the tensor product Hilbert space  $L^2(\Omega) \otimes L^2(\mathcal{D})$  which is endowed with the norm

$$\|z\|_{L^2(\Omega) \otimes L^2(\mathcal{D})} := \left( \int_{\Omega} \|z(\omega, \cdot)\|_{L^2(\mathcal{D})}^2 d\mathbb{P}(\omega) \right)^{\frac{1}{2}} < \infty,$$

where  $L^2(\Omega) := L^2(\Omega, \mathcal{F}, \mathbb{P})$ . For any random variable  $g$  defined on  $(\Omega, \mathcal{F}, \mathbb{P})$ , the mean  $\mathbb{E}(g)$  of  $g$  is given by

$$\langle g \rangle := \mathbb{E}(g) = \int_{\Omega} g d\mathbb{P}(\omega) < \infty. \quad (1)$$

For a Hilbert space  $H$  of functions on  $\mathcal{D}$  and a time interval  $[0, T]$ , we write  $L^2(0, T; H)$  for the tensor product space  $L^2([0, T]) \otimes H$ . However, in particular, we write  $L^2(0, T; \mathcal{D})$  for  $L^2(0, T; L^2(\mathcal{D}))$ ; we will be using these last two notations interchangeably.

We consider an optimization problem involving an incompressible Newtonian flow with an uncertain inflow in a backward step domain. More precisely, the OPNS we want to solve is given by the minimization of the total vorticity:

$$\mathcal{J} = \frac{1}{2} \|\nabla \times \mathbf{v}\|^2 + \frac{\beta}{2} \|\mathbf{u}\|^2, \quad (2)$$

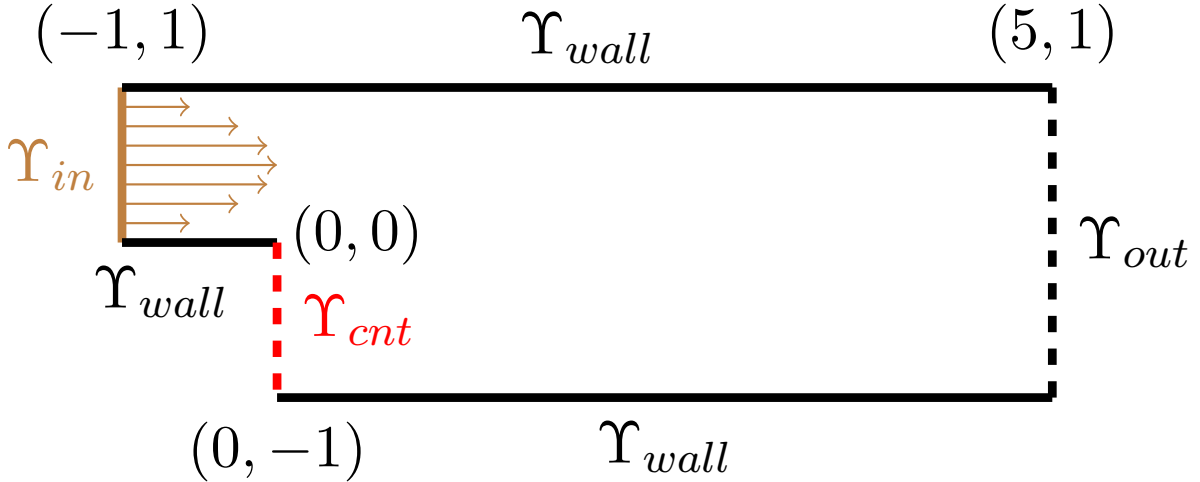
where  $\mathbf{v}, \mathbf{u} : [0, T] \times \Omega \times \mathcal{D} \rightarrow \mathbb{R}^d$  are random fields, representing the state (velocity) and the control functions. The regularization constant  $\beta$  in (2) balances between minimization of the vorticity and penalization of the control magnitude. The objective function  $\mathcal{J}(\mathbf{v}, \mathbf{u})$  is a deterministic quantity with uncertain components  $\mathbf{v}$  and  $\mathbf{u}$ .

The minimization of  $\mathcal{J}$  is considered subject,  $\mathbb{P}$ -almost surely, to the Navier-Stokes equations<sup>3</sup>

$$\left\{ \begin{array}{ll} \frac{\partial \mathbf{v}}{\partial t} - \nu \Delta \mathbf{v} + (\mathbf{v} \cdot \nabla) \mathbf{v} + \nabla p = F_d \mathbf{u}, & \text{in } (0, T] \times \Omega \times \mathcal{D}, \\ -\nabla \cdot \mathbf{v} = 0, & \text{in } (0, T] \times \Omega \times \mathcal{D}, \\ \mathbf{v} = \boldsymbol{\theta}, & \text{on } (0, T] \times \Omega \times \Upsilon_{in}, \\ \mathbf{v} = 0, & \text{on } (0, T] \times \Omega \times \Upsilon_{wall}, \\ \frac{\partial \mathbf{v}}{\partial \mathbf{n}} = 0, & \text{on } (0, T] \times \Omega \times \Upsilon_{out}, \\ \frac{\partial \mathbf{v}}{\partial \mathbf{n}} = F_b \mathbf{u}, & \text{on } (0, T] \times \Omega \times \Upsilon_{cnt}, \\ \mathbf{v}(0, \cdot, \cdot) = 0, & \text{in } \Omega \times \mathcal{D}. \end{array} \right. \quad (3)$$

<sup>3</sup>In this paper, we do not consider the case of state- or control- or mixed control-state-constrained problems [24, 36, 38]. These problems can be tackled via, for instance, semi-smooth Newton algorithms [23, 27, 29].

Figure 1: An uncertain inflow in a backward step domain



Here,  $\boldsymbol{\theta} : [0, T] \times \Omega \times \Upsilon_{in} \rightarrow \mathbb{R}^d$  is the inflow boundary condition,  $p : [0, T] \times \Omega \times \mathcal{D} \rightarrow \mathbb{R}$  is the pressure, and  $\mathbf{n}$  is the outward-pointing normal to the boundary. The linear operators  $F_d$  and  $F_b$  apply the control to the appropriate part of the system. We consider two cases.

1. *Distributed* (or full) control:  $F_d = I$ ,  $F_b = 0$ , and
2. *Boundary* control:  $F_d = 0$ , and  $F_b = I_{\Upsilon_{cnt}}$ , the identity on  $\Upsilon_{cnt}$  and zero otherwise.

The parameter  $\nu$  is the kinematic viscosity. There exist variations of this problem such as a deterministic control [40] or an uncertain domain [6].

In our experiments, we use a zero initial condition, applying instead an exponentially growing stochastic inflow

$$\boldsymbol{\theta} = \begin{bmatrix} \kappa(\mathbf{x}, \omega)(1 - e^{-t}) \\ 0 \end{bmatrix}$$

(see Fig. 1), where  $\kappa(\mathbf{x}, \omega)$  is a random field on  $\Upsilon_{in}$ . We assume that  $\kappa$  admits the Karhunen-Loeve expansion (KLE):

$$\kappa(\mathbf{x}, \omega) = 4x_2(1 - x_2) + \sum_{k=1}^{\infty} \frac{1}{2} \gamma^{-k} \cdot \sin(2\pi k x_2) \cdot \xi_k(\omega). \quad (4)$$

Here, the first term is the mean parabolic inflow, whereas  $\xi_k$ ,  $k = 1, 2, \dots$  are independent uniformly distributed random quantities,  $\xi_k \sim \mathcal{U}(-1, 1)$ . The parameter  $\gamma$  governs the KLE decay rate, and hence the smoothness of the field. In computational practice, we truncate (4) after  $m \in \mathbb{N}$  terms such that the error is sufficiently small:

$$\kappa(\mathbf{x}, \omega) \approx \kappa_m(\mathbf{x}, \xi(\omega)) = 4x_2(1 - x_2) + \sum_{k=1}^m \frac{1}{2} \gamma^{-k} \cdot \sin(2\pi k x_2) \cdot \xi_k(\omega), \quad (5)$$

where  $\xi := \{\xi_1, \xi_2, \dots, \xi_m\}$ . This coefficient mimics the (truncated) Fourier expansion of the Matern covariance function for  $\kappa$ . The assumption that  $\kappa(x, \omega)$  admits a KLE representation allows to transform the stochastic OPNS into a parametric deterministic problem, depending on  $\xi := \{\xi_1, \xi_2, \dots, \xi_m\}$ . For brevity, since we consider  $m$  as an a priory model parameter, we always denote  $\kappa_m$  by  $\kappa$  in the rest of the paper.

We also assume that each random variable is characterized by a probability density function  $\rho_i : \Gamma_i \rightarrow [0, 1]$ . If the distribution measure of the random vector  $\xi(\omega)$  is absolutely continuous with respect to the Lebesgue measure, then there exists a joint probability density function  $\rho : \Gamma \rightarrow \mathbb{R}^+$ , where  $\Gamma := \bigotimes_{k=1}^m \Gamma_k \subset \mathbb{R}^m$ ,  $\rho(\xi) = \prod_{k=1}^m \rho_k(\xi_k)$ , and  $\rho \in L^\infty(\Gamma)$ . In particular, given the parametric representation (5) of  $\kappa(\mathbf{x}, \omega)$ , the Doob-Dynkin Lemma [4], guarantees that the solution, for example,  $\mathbf{v}$ , corresponding to the the OPNS as given by (2) and (3) admits exactly the same parametrization; that is,

$$\mathbf{v}(t, \omega, \mathbf{x}) = \mathbf{v}(t, \xi_1(\omega), \xi_2(\omega), \dots, \xi_m(\omega), \mathbf{x}). \quad (6)$$

Furthermore, we can now replace the probability space  $(\Omega, \mathcal{F}, \mathbb{P})$  with  $(\Omega, \mathbb{B}(\Gamma), \rho(\xi)d\xi)$ , where  $\mathbb{B}(\Gamma)$  denotes the Borel  $\sigma$ -algebra on  $\Gamma$  and  $\rho(\xi)d\xi$  is the finite measure of the vector  $\xi$ . Besides, denoting the space of square-integrable random variables with respect to the density  $\rho$  by  $L^2_\rho(\Gamma)$ , we introduce the space  $L^2_\rho(\Gamma) \otimes L^2(\mathcal{D})$  equipped with the norm

$$\|z\|_{L^2_\rho(\Gamma) \otimes L^2(\mathcal{D})} := \left( \int_\Gamma \|z(\xi, \cdot)\|_{L^2(\mathcal{D})}^2 \rho(\xi) d\xi \right)^{\frac{1}{2}} < \infty. \quad (7)$$

Similarly, using equation (1), we have

$$\langle g \rangle := \mathbb{E}[g] = \int_\Gamma g(\xi) \rho(\xi) d\xi < \infty. \quad (8)$$

For the numerical simulation of the OPNS given by (2) and (3), we will adopt the so-called *optimize-then-discretize* (OTD) strategy in which case we first build an infinite dimensional Lagrangian and then consider its variation with respect to state, pressure, control, and two Lagrange multipliers that can be identified as the adjoint state  $\boldsymbol{\lambda}$  and adjoint pressure  $\mu$  [47, 25]. To this end, observe now that the optimization problem under consideration is nonlinear due to the nonlinearity of the convective term  $(\mathbf{v} \cdot \nabla)\mathbf{v}$ . Both Newton and Picard iterations have shown to be good iterative solvers to tackle these nonlinear equations [19]. Since the Picard iteration has a larger radius of convergence compared to the Newton iteration, our choice in this contribution is the Picard iteration. We apply the so-called Karush-Kuhn-Tucker

procedure [19, Chapter 8.2] to obtain the optimality system [14]:

$$\begin{aligned}
\partial_t \mathbf{v} - \nu \Delta \mathbf{v} + (\bar{\mathbf{v}} \cdot \nabla) \mathbf{v} + \nabla p &= F_d \mathbf{u} \\
\nabla \cdot \mathbf{v} &= 0 \\
\text{on } \Upsilon_{wall} : \quad \mathbf{v} &= 0 \\
\text{on } \Upsilon_{in} : \quad \mathbf{v} &= \boldsymbol{\theta} \\
\text{on } \Upsilon_{out} : \quad \frac{\partial \mathbf{v}}{\partial \mathbf{n}} &= 0, \\
\text{on } \Upsilon_{cnt} : \quad \frac{\partial \mathbf{v}}{\partial \mathbf{n}} &= F_b \mathbf{u}, \\
\mathbf{v}(0, \cdot, \cdot) &= 0,
\end{aligned} \tag{9}$$

$$\begin{aligned}
-\partial_t \boldsymbol{\lambda} - \nu \Delta \boldsymbol{\lambda} - (\bar{\mathbf{v}} \cdot \nabla) \boldsymbol{\lambda} + (\nabla \bar{\mathbf{v}})^T \boldsymbol{\lambda} + \nabla \mu &= -\nabla \times (\nabla \times \mathbf{v}) \\
\nabla \cdot \boldsymbol{\lambda} &= 0 \\
\text{on } \Upsilon_{wall} \cup \Upsilon_{in} : \quad \boldsymbol{\lambda} &= 0 \\
\text{on } \Upsilon_{out} \cup \Upsilon_{cnt} : \quad \frac{\partial \boldsymbol{\lambda}}{\partial \mathbf{n}} &= 0 \\
\boldsymbol{\lambda}(T, \cdot, \cdot) &= 0,
\end{aligned} \tag{10}$$

$$\beta \mathbf{u} + (F_d + F_b) \boldsymbol{\lambda} = 0, \tag{11}$$

where  $\bar{\mathbf{v}}$  denotes the velocity from the previous Picard iteration. Having solved this system, we update  $\bar{\mathbf{v}} = \mathbf{v}$  and so on until convergence.

### 3 The discrete problem

The stochasticity of the inflow  $\kappa(\mathbf{x}, \omega)$  is inherited by all solution components. For example, the velocity becomes a function of time,  $d$  spatial variables and  $m$  random parameters,  $\mathbf{v}(t, \xi, \mathbf{x})$ . This makes the problem far more challenging than its deterministic counterpart. In this paper, we employ simultaneous independent discretization of all variables, and the stochastic Galerkin method (SGFEM) w.r.t. the parameters  $\xi$ .

The SGFEM is an intrusive approach in which one seeks the solution in a finite-dimensional subspace  $\mathcal{Y}_{n_\xi} \otimes \mathcal{X}_h \subset L^2_\rho(\Gamma) \otimes L^2(0, T; L^2(\mathcal{D}))$  consisting of tensor products of deterministic functions defined on the spatial domain and stochastic functions defined on the probability space [21, 39]. Different classes of SGFEMs are distinguished by their choices for  $\mathcal{Y}_{n_\xi}$ . When all random variables  $\xi_k$  are independent and identically distributed Gaussian, the basis of multidimensional Hermite polynomials of the total degree  $n_\xi$  is called the *polynomial chaos*, a terminology originally introduced by Norbert Wiener [49] in the context of turbulence modeling. The use of Hermite polynomials ensures that the corresponding basis functions are orthogonal with respect to the Gaussian probability measure. This leads to sparse linear systems, a crucial property that must be exploited for fast solution schemes [39]. Relying on the fact that there exists a one-to-one correspondence between the probability density functions of alternative distributions and the weight functions of certain orthogonal polynomials,

the concept of Hermite polynomials chaos has been extended to *generalized* polynomial chaos [50]. For instance, if uniform random variables (having support on a bounded interval) are chosen, then Legendre polynomials are the correct choice. Similarly, Jacobi polynomials go with beta-distributed random variables. When random variables with bounded images are used, the convergence and approximation properties of the resulting SGFEM are discussed in [4]. Some SGFEM classes use tensor products of piecewise polynomials on the subdomains  $\Gamma_k \subset \Gamma$  [4, 17, 33], where the polynomial degree is fixed and approximation is improved by refining the partition of  $\Gamma$ .

Throughout this paper we use the classical, so-called *spectral* SGFEM (see e.g. [18, 20, 32, 39]), which employs global Legendre polynomials for uniformly distributed  $\xi_k$ . We discretize each  $\xi_k$  independently via polynomials of the maximal degree  $n_\xi - 1$ . That is, we use the space

$$\mathcal{Y}_{n_\xi} = \left\{ \text{span} \left( \psi_j(\xi) = \prod_{k=1}^m \psi_{j_k}(\xi_k) \right), \quad \forall j_k = 0, \dots, n_\xi - 1, \quad j = (j_1, \dots, j_m) \right\}.$$

This implies that the global discretization is done via multilinear polynomials of total degree  $m(n_\xi - 1)$ . The total number of basis functions in  $\mathcal{Y}_{n_\xi}$  is  $N_\xi = n_\xi^m$ , which can be prohibitively large for a straightforward storage. We circumvent this problem by using low-rank product decompositions on the discrete level in Section 4.

For the spatial discretization we use mixed finite element spaces with stable elements, i.e. elements that satisfy the *inf-sup* condition, e.g. the  $Q_2$ - $Q_1$  pair [43]. This gives us different spaces  $V_h = \text{span}\{\phi_1, \dots, \phi_{N_v}\} \subset L^2(0, T; H_0^1(\mathcal{D}))$  for the velocity and  $\tilde{V}_h = \text{span}\{\tilde{\varphi}_1, \dots, \tilde{\varphi}_{N_p}\} \subset L^2(0, T; L^2(\mathcal{D}))$  for the pressure.

The time is discretized on a uniform grid,  $t_n = \tau n$ ,  $n = 1, \dots, N_t$ , such that  $t_{N_t} = T$ . The fully discrete coefficients of the velocity will be denoted as  $\mathbf{v}_h(n, j, k)$ , and similarly the coefficients of  $p$ ,  $\mathbf{u}$ ,  $\boldsymbol{\lambda}$ , and  $\mu$  are denoted by  $p_h$ ,  $\mathbf{u}_h$ ,  $\boldsymbol{\lambda}_h$  and  $\mu_h$ , respectively. The solution functions are approximated as

$$\begin{aligned} \mathbf{v}(t_n, \xi, \mathbf{x}) &\approx \sum_{j=0}^{N_\xi-1} \sum_{k=1}^{N_v} \mathbf{v}_h(n, j, k) \psi_j(\xi) \phi_k(\mathbf{x}), \\ p(t_n, \xi, \mathbf{x}) &\approx \sum_{j=0}^{N_\xi-1} \sum_{k=1}^{N_p} p_h(n, j, k) \psi_j(\xi) \tilde{\varphi}_k(\mathbf{x}). \end{aligned} \tag{12}$$

Analogous expressions hold for  $\mathbf{u}$ ,  $\boldsymbol{\lambda}$  and  $\mu$ .

We plug these expressions into (9)–(11), and project the optimality system onto  $\text{span}(\psi_j \phi_k)$  for the velocity equations and  $\text{span}(\psi_j \tilde{\varphi}_k)$  for the pressure equations. We introduce the following matrices, composed from the corresponding bilinear forms:

- $L_0(k, k') = \int_{\mathcal{D}} \nabla \phi_k(\mathbf{x}) \cdot \nabla \phi_{k'}(\mathbf{x}) d\mathbf{x}$  is the Laplace matrix for a single velocity component, and  $L = \text{blkdiag}(L_0, L_0)$  is the Laplace matrix for both velocity components.
- $M_0(k, k') = \int_{\mathcal{D}} \phi_k(\mathbf{x}) \phi_{k'}(\mathbf{x}) d\mathbf{x}$  and  $M = \text{blkdiag}(M_0, M_0)$ , the mass matrix.
- $B(k, k') = \int_{\mathcal{D}} \tilde{\varphi}_k(\mathbf{x}) \nabla \phi_{k'}(\mathbf{x}) d\mathbf{x}$ , the mixed gradient matrix.

- $N_n = \text{blkdiag}(N_{0,n}, N_{0,n})$ , the convection matrix, where

$$N_{0,n}(jk, j'k') = \int_{\Gamma} \psi_j \sum_{j''=0}^{N_{\xi}-1} N_s[\bar{\mathbf{v}}_h(n, j'', :)](k, k') \psi_{j''} \psi_{j'} \rho d\xi, \quad (13)$$

$$N_s[\bar{\mathbf{v}}_h](k, k') = \int_{\mathcal{D}} \phi_k(\mathbf{x}) \sum_{k''=1}^{N_v} \phi_{k''}(\mathbf{x}) \bar{\mathbf{v}}_h(k'') \cdot \nabla \phi_{k'}(\mathbf{x}) d\mathbf{x}. \quad (14)$$

- $W_n(jk, j'k') = \int_{\Gamma} \psi_j(\xi) \sum_{j''=0}^{N_{\xi}-1} W_s[\bar{\mathbf{v}}_h(n, j'', :)](k, k') \psi_{j''}(\xi) \psi_{j'}(\xi) \rho(\xi) d\xi$ , the adjoint convection matrix, where

$$W_s[\bar{\mathbf{v}}_h](k, k') = \int_{\mathcal{D}} \phi_k(\mathbf{x}) \sum_{k''=1}^{N_v} \nabla \bar{\mathbf{v}}_h(k'') \phi_{k''}(\mathbf{x}) \phi_{k'}(\mathbf{x}) d\mathbf{x}. \quad (15)$$

Since the Legendre polynomials are orthogonal, a proper normalization can turn the stochastic mass matrix into an identity.

We use the implicit Euler approximation for the time derivative, although it can be replaced by higher-order schemes. Overall, this gives the following discrete optimality equations.

$$\begin{aligned} & \frac{\mathbf{v}_h(n, j, k) - \mathbf{v}_h(n-1, j, k)}{\tau} - \sum_{k'=1}^{N_v} \nu L(k, k') \mathbf{v}_h(n, j, k') \\ & + \sum_{j'=0}^{N_{\xi}-1} \sum_{k'=1}^{N_v} N_n(jk, j'k') \mathbf{v}_h(n, j', k') + \sum_{k'=1}^{N_p} B(k', k)^T p_h(n, j, k') = \sum_{k'=1}^{N_v} M_{cnt}(k, k') \mathbf{u}_h(n, j, k') \\ & \sum_{k'=1}^{N_v} B(k, k') \mathbf{v}_h(n, j, k') = 0 \quad (16) \\ & \text{for } k \in \Upsilon_{wall} : \quad \mathbf{v}_h(n, j, k) = 0 \\ & \text{for } k \in \Upsilon_{in} : \quad \mathbf{v}_h(n, j, k) = \boldsymbol{\theta}_h(n, j, k) \\ & \mathbf{v}_h(0, j, k) = 0, \end{aligned}$$

$$\begin{aligned} & \frac{\boldsymbol{\lambda}_h(n, j, k) - \boldsymbol{\lambda}_h(n+1, j, k)}{\tau} - \sum_{k'=1}^{N_v} \nu L(k, k') \boldsymbol{\lambda}_h(n, j, k') \\ & + \sum_{j'=0}^{N_{\xi}-1} \sum_{k'=1}^{N_v} (-N_n(jk, j'k') + W_n(jk, j'k')) \boldsymbol{\lambda}_h(n, j', k') \\ & + \sum_{k'=1}^{N_p} B(k', k)^T \mu_h(n, j, k') = \sum_{k'=1}^{N_v} L(k, k') \mathbf{v}_h(n, j, k') \\ & \sum_{k'=1}^{N_v} B(k, k') \boldsymbol{\lambda}_h(n, j, k') = 0 \quad (17) \end{aligned}$$



$$\text{for } k \in \Upsilon_{wall} \cup \Upsilon_{in} : \quad \begin{aligned} \boldsymbol{\lambda}_h(n, j, k) &= 0 \\ \boldsymbol{\lambda}_h(N_t, j, k) &= 0, \end{aligned}$$

$$\beta \sum_{k'=1}^{N_v} M(k, k') \mathbf{u}_h(n, j, k') + \sum_{k'=1}^{N_v} M_{cnt}(k, k') \boldsymbol{\lambda}_h(n, j, k') = 0, \quad (18)$$

where  $M_{cnt} = F_d + F_b$ . This system can be written in a compact matrix form

$$\underbrace{\begin{bmatrix} \mathbf{M}_1 & 0 & -\mathbf{K}^* \\ 0 & \beta \mathbf{M}_2 & \mathbf{M}_3^T \\ -\mathbf{K} & \mathbf{M}_3 & 0 \end{bmatrix}}_{:=\mathfrak{A}} \underbrace{\begin{bmatrix} \mathbf{v}_h \\ p_h \\ \mathbf{u}_h \\ \boldsymbol{\lambda}_h \\ \mu_h \end{bmatrix}}_{:=\mathbf{y}} = \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{g}_h \end{bmatrix}}_{:=\mathbf{b}}. \quad (19)$$

Since we use tensor product basis functions, each of the block matrices in  $\mathfrak{A}$  can be represented via Kronecker products. In particular, the forward and adjoint operators for the time-stochastic-space Navier-Stokes equations write as

$$\mathbf{K} = I_{N_t} \otimes I_{n_\xi}^{\otimes m} \otimes \mathcal{L} + C\tau^{-1} \otimes I_{n_\xi}^{\otimes m} \otimes \mathcal{M} + \mathbf{N}[\bar{\mathbf{v}}_h], \quad (20)$$

$$\mathbf{K}^* = I_{N_t} \otimes I_{n_\xi}^{\otimes m} \otimes \mathcal{L} + C^\top \tau^{-1} \otimes I_{n_\xi}^{\otimes m} \otimes \mathcal{M} + \mathbf{W}[\bar{\mathbf{v}}_h] - \mathbf{N}[\bar{\mathbf{v}}_h], \quad (21)$$

where [5]

- $I_{N_t}$  is the identity matrix and  $C = \text{tridiag}(-1, 1, 0)$  comes from the implicit Euler discretization;
- $I_{n_\xi}$  is the identity mass matrix for each variable  $\xi$ .
- $\mathcal{L} = \begin{bmatrix} \nu L & B^\top \\ B & 0 \end{bmatrix}$  is the discretization of the stationary Stokes operator, and  $\mathcal{M} = \text{blkdiag}(M, 0)$  for the velocity mass matrix, extended by zeros to the pressure space;
- $\mathbf{N}$  and  $\mathbf{W}$  correspond to the convection terms. They depend on the *low-rank* structure of the solution  $\bar{\mathbf{v}}_h$ , which will be introduced in the next section.

Furthermore, we introduce an extended velocity Laplace matrix  $\mathcal{L}_0 = \text{blkdiag}(L, 0)$ . Then we have

$$\mathbf{M}_1 = I_{N_t} \otimes I_{n_\xi}^{\otimes m} \otimes \mathcal{L}_0, \quad \mathbf{M}_2 = I_{N_t} \otimes I_{n_\xi}^{\otimes m} \otimes M, \quad \mathbf{M}_3 = I_{N_t} \otimes I_{n_\xi}^{\otimes m} \otimes \begin{bmatrix} M_{cnt} \\ 0 \end{bmatrix}, \quad (22)$$

where

1.  $M_{cnt} = M$  for the distributed control, and
2.  $M_{cnt}(k, k') = \begin{cases} M(k, k'), & \text{if } k, k' \in \Upsilon_{cnt}, \\ 0, & \text{otherwise,} \end{cases}$  for the boundary control.

The third vector  $\mathbf{g}_h$  in the right-hand side  $\mathbf{b}$  in (19) depends on the boundary conditions. The KLE yields the following tensor form for the inflow function,

$$\boldsymbol{\theta}_h = \mathbf{g}_t \otimes \sum_{k=0}^m \left[ \left( \bigotimes_{\ell=1}^{k-1} \mathbf{e}_1 \right) \otimes \mathbf{e}_2 \otimes \left( \bigotimes_{\ell=k+1}^m \mathbf{e}_1 \right) \otimes \boldsymbol{\theta}_{kh} \right], \quad (23)$$

where  $\mathbf{g}_t$  is the vector of the  $1 - \exp(-t)$  factor sampled at all time points,  $\mathbf{e}_1$  and  $\mathbf{e}_2$  are the first and second unit vectors, respectively, and  $\boldsymbol{\theta}_{kh}$  is the discretization of the  $k$ -th spatial factor in (4). Now, distinguishing inner and boundary degrees of freedom, we partition the PDE matrix into the corresponding blocks,

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{II} & \mathbf{K}_{IB} \\ \mathbf{K}_{BI} & \mathbf{K}_{BB} \end{bmatrix}.$$

The actual separation of inner (I) and boundary (B) elements is done in the spatial factors only. Then the usual finite element approach is employed: we eliminate  $\mathbf{K}_{IB}$  and  $\mathbf{K}_{BI}$  in the left hand side of (19), replacing  $\mathbf{K}$  by  $\mathbf{K}_{II}$ , and construct the right-hand side from (23) as  $\mathbf{g}_h = \mathbf{K}_{IB}\boldsymbol{\theta}_h$ .

## 4 Tensor Train decomposition approach

The solution  $y(t, \xi_1, \dots, \xi_m, \mathbf{x})$  (where  $y$  stands for  $\mathbf{v}, p, \mathbf{u}, \boldsymbol{\lambda}$  or  $\mu$ ) is a multivariate function. After discretization, independently in each variable  $t, \xi_1, \dots, \xi_m, \mathbf{x}$ , the discrete values of  $y$  can be enumerated by  $D = m + 2$  independent indices<sup>4</sup>, i.e. they form a *tensor*. Storing such a tensor directly might be prohibitively expensive. Therefore, we *decompose* tensors using the separation of variables.

Given a  $D$ -index tensor  $y(i_1, \dots, i_D)$ , its Tensor Train decomposition [34] is written as follows:

$$y(\mathbf{i}) = \sum_{s_1=1}^{r_1} \cdots \sum_{s_{D-1}=1}^{r_{D-1}} y_{s_1}^{(1)}(i_1) y_{s_1, s_2}^{(2)}(i_2) \cdots y_{s_{D-1}}^{(D)}(i_D), \quad (24)$$

where  $\mathbf{i}$  denotes the multi-index,  $\mathbf{i} = (i_1, \dots, i_D)$ . Each element of  $y$  is represented (or approximated) by a sum of products of elements of smaller tensors  $y^{(\ell)}$ ,  $\ell = 1, \dots, D$ , called TT blocks. The auxiliary summation indices  $s_1, \dots, s_{D-1}$  are called rank indices, and their ranges  $r_1, \dots, r_{D-1}$  are called TT ranks. The TT decomposition is also known as the Matrix Product States [41, 37], since, omitting  $s_1, \dots, s_{D-1}$ , we can say that an element  $y(\mathbf{i})$  is equal to a product of  $i_\ell$ -dependent matrices. The TT ranks depend on the particular tensor and accuracy, if (24) is satisfied approximately. Denoting upper bounds  $r_\ell \lesssim r$ ,  $i_\ell \lesssim N$ ,  $\ell = 1, \dots, D - 1$ , we estimate the storage cost of the TT blocks to be in  $\mathcal{O}(DNr^2)$ , which can be much less than the full amount  $N^D$ . Usually, we refer to  $r$  as the maximal TT rank.

The TT format allows to compress a given tensor up to a threshold  $\varepsilon$ , also if  $y$  is already given in the TT format, but possibly with overestimated TT ranks. This happens if we sum two arrays in the TT format, or perform the matrix multiplication. A matrix  $A$ , acting on

---

<sup>4</sup>We don't separate different *components* of  $\mathbf{x}$ , which are actually *dependent*, due to the domain geometry. Therefore, all spatial degrees of freedom are treated as one variable.

tensors of type  $y$ , can be seen as a  $2D$ -dimensional tensor and represented in a slightly different TT format,

$$A(\mathbf{i}, \mathbf{j}) = \sum_{s_1, \dots, s_{D-1}=1}^{R_1, \dots, R_{D-1}} A_{s_1}^{(1)}(i_1, j_1) \cdots A_{s_{D-1}}^{(D)}(i_D, j_D), \quad (25)$$

where  $\mathbf{i}, \mathbf{j}$  are multi-indices consisting of  $i_\ell$  and  $j_\ell$ , respectively. This is consistent with the Kronecker product ( $\otimes$ ) if  $D = 2$  and  $R = 1$ . The matrix-vector product  $\sum_{\mathbf{j}} A(\mathbf{i}, \mathbf{j})y(\mathbf{j})$  can then be implemented in the TT format block by block [34, 41].

## TT format of the convection matrix

We assume now that the velocity, in particular the previous Picard iterate, is represented by a TT decomposition (24),

$$\bar{\mathbf{v}}_h(n, j_1, \dots, j_m, k) = \bar{\mathbf{v}}^{(1)}(n) \bar{\mathbf{v}}^{(2)}(j_1) \cdots \bar{\mathbf{v}}^{(m+1)}(j_m) \bar{\mathbf{v}}^{(m+2)}(k), \quad (26)$$

with the ranks  $r_1, \dots, r_{m+1}$ . We are going to derive a matrix TT format (25) for  $\mathbf{N}[\bar{\mathbf{v}}_h]$  from (20). First, we notice from (14) that  $\mathbf{N}$  depends linearly on  $\bar{\mathbf{v}}_h$ . Therefore, we can plug (26) into (14) and distribute the summations. This gives

$$N_{0,n}(jk, j'k') = \bar{\mathbf{v}}^{(1)}(n) \quad (27)$$

$$\cdot \left[ \sum_{j''=0}^{N_\xi-1} \int_{\Gamma} \psi_j(\xi) \psi_{j'}(\xi) \psi_{j''}(\xi) \rho(\xi) d\xi \cdot \bar{\mathbf{v}}^{(2)}(j''_1) \cdots \bar{\mathbf{v}}^{(m+1)}(j''_m) \right] \quad (28)$$

$$\cdot \left[ \sum_{k''=1}^{N_v} \int_{\mathcal{D}} \phi_k(\mathbf{x}) \nabla \phi_{k'}(\mathbf{x}) \phi_{k''}(\mathbf{x}) dx \cdot \bar{\mathbf{v}}^{(m+2)}(k'') \right]. \quad (29)$$

The last term in this expression is a deterministic convection matrix  $N_s[\bar{\mathbf{v}}^{(m+2)}]$  (14), which can be assembled with the cost  $\mathcal{O}(r_{m+1}N_v)$ . The middle term can be further factorised due to the tensor product PCE space  $\mathcal{Y}_{n_\xi}$ . Indeed, introducing the matrices of triple products for all  $k = 1, \dots, m$ ,

$$H_{j''_k}(j_k, j'_k) = \int_{\Gamma_k} \psi_{j_k}(\xi_k) \psi_{j'_k}(\xi_k) \psi_{j''_k}(\xi_k) \rho_k(\xi_k) d\xi_k, \quad H_{j''_k} \in \mathbb{R}^{n_\xi \times n_\xi}, \quad (30)$$

we see that the multi-dimensional triple product in (28) can be written as a Kronecker product of  $m$  individual matrices (30). Finally, we notice that  $N_{0,n}$  acts independently on each time step  $n$ , i.e. it is diagonal w.r.t.  $n$ , and so is  $\mathbf{N}$ . Overall, given the TT format of  $\bar{\mathbf{v}}$  (26), there exists a TT format for the convection matrix with the same TT ranks,

$$\begin{aligned} \mathbf{N}[\bar{\mathbf{v}}_h] &= \sum_{s_1, \dots, s_{m+1}=1}^{r_1, \dots, r_{m+1}} \text{diag}(\bar{\mathbf{v}}_{s_1}^{(1)}) \\ &\otimes \left[ \sum_{j_1=0}^{n_\xi-1} H_{j_1} \bar{\mathbf{v}}_{s_1, s_2}^{(2)}(j_1) \right] \otimes \cdots \otimes \left[ \sum_{j_m=0}^{n_\xi-1} H_{j_m} \bar{\mathbf{v}}_{s_m, s_{m+1}}^{(m+1)}(j_m) \right] \\ &\otimes \text{blkdiag} \left( N_s[\bar{\mathbf{v}}_{s_{m+1}}^{(m+2)}], N_s[\bar{\mathbf{v}}_{s_{m+1}}^{(m+2)}] \right). \end{aligned} \quad (31)$$

Notice that the right-hand side  $\mathbf{g}_h = \mathbf{K}_{IB}\boldsymbol{\theta}_h$  in (19) must be recomputed in every Picard iteration, since  $\mathbf{K}_{IB}$  carries the corresponding (new) part  $\mathbf{N}_{IB}$ . The adjoint convection matrix can be constructed similarly, with  $W_s$  defined in (15),

$$\begin{aligned} \mathbf{W}[\bar{\mathbf{v}}_h] &= \sum_{\substack{r_1, \dots, r_{m+1} \\ s_1, \dots, s_{m+1}=1}} \text{diag}(\bar{\mathbf{v}}_{s_1}^{(1)}) \\ &\otimes \left[ \sum_{j_1=0}^{n_\xi-1} H_{j_1} \bar{\mathbf{v}}_{s_1, s_2}^{(2)}(j_1) \right] \otimes \cdots \otimes \left[ \sum_{j_m=0}^{n_\xi-1} H_{j_m} \bar{\mathbf{v}}_{s_m, s_{m+1}}^{(m+1)}(j_m) \right] \otimes W_s [\bar{\mathbf{v}}_{s_{m+1}}^{(m+2)}]. \end{aligned}$$

## Alternating linear solver

The Oseen equation given by (19) is a large linear system for  $\mathbf{y}$ , and needs to be solved keeping all the components in the TT format in order to keep the storage requirements low. A state of the art approach to this problem is alternating tensor product algorithms [48, 41, 26]. Given the system  $\mathfrak{A}\mathbf{y} = \mathbf{b}$ , we iterate over  $\ell = 1, \dots, D$ , and seek only the elements of  $y^{(\ell)}$  in each step, while the other TT blocks are fixed. Notice that the TT format (24) is *linear* with respect to the elements of each  $y^{(\ell)}$ , i.e., there exists a matrix  $Y_\ell$  (34) such that  $\mathbf{y} = Y_\ell y^{(\ell)}$ . This renders  $\mathfrak{A}\mathbf{y} = \mathbf{b}$  an overdetermined system  $\mathfrak{A}Y_\ell y^{(\ell)} = \mathbf{b}$  w.r.t. the elements of  $y^{(\ell)}$ . This system is resolved via a projection onto  $Y_\ell$ , such that  $y^{(\ell)}$  is computed from a smaller system  $(Y_\ell^\top \mathfrak{A} Y_\ell) y^{(\ell)} = Y_\ell^\top \mathbf{b}$ . Particularly efficient realizations are the Density Matrix Renormalization Group (DMRG) methods from quantum physics [48, 28] and the Alternating Minimal Energy (AMEn) algorithm [16] from the mathematical community. The DMRG approach computes two neighboring TT blocks in each step, say,  $\ell$  and  $\ell + 1$ , which allows to adapt the TT rank  $r_\ell$  to the desired accuracy. While the DMRG method was found to be extremely effective for spin Schroedinger eigenvalue problems, it may deliver insufficient accuracy for linear systems, especially with non-symmetric matrices. The AMEn method is usually faster and more robust, since it seeks only one TT block in each step, but then performs an explicit augmentation of the computed TT block of the solution by a TT block of the current residual. This allows to change TT ranks and facilitate convergence.

However, the standard AMEn method is not applicable to  $\mathfrak{A}$  directly: due to indefiniteness of  $\mathfrak{A}$ , its Galerkin projection may be degenerate. For example, consider

$$\mathfrak{A} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} \quad \text{and} \quad Y_\ell = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

One can readily verify that  $Y_\ell^\top \mathfrak{A} Y_\ell = 0$ . To cope with this issue, we employ the so-called block TT format [15] and project each *submatrix* of  $\mathfrak{A}$  separately.

Let us enumerate the components of (19) as  $y_\iota$ , where  $y_1 = \mathbf{v}_h$ ,  $y_2 = p_h$ ,  $y_3 = \mathbf{u}_h$ ,  $y_4 = \boldsymbol{\lambda}_h$  and  $y_5 = \mu_h$ . We approximate all components simultaneously by a TT format with the same blocks except the  $\ell$ -th one for some  $\ell = 1, \dots, D$ , where the enumerator  $\iota = 1, \dots, 5$  appears,

$$y_\iota(\mathbf{i}) = \sum_{s_1, \dots, s_{D-1}} y_{s_1}^{(1)}(i_1) \cdots y_{s_{\ell-1}, s_\ell}^{(\ell)}(i_\ell, \iota) \cdots y_{s_{D-1}}^{(D)}(i_D). \quad (32)$$

Using an SVD of the core  $y^{(\ell)}$ , we can turn (32) into the form

$$y_\iota(\mathbf{i}) = \sum_{s_1, \dots, s_{D-1}} y_{s_1}^{(1)}(i_1) \cdots y_{s_{\ell-1}, s_\ell}^{(\ell)}(i_\ell) \cdot y_{s_\ell, s_{\ell+1}}^{(\ell+1)}(i_{\ell+1}, \iota) \cdots y_{s_{D-1}}^{(D)}(i_D)$$

or vice versa. That is, we can *move*  $\iota$  to any particular TT block in the course of the alternating iteration [15]. When the  $\ell$ -th block carries  $\iota = 1, \dots, 5$ , the Oseen system (19) is projected onto the other TT blocks component by component, giving a system

$$\begin{bmatrix} \hat{M}_1 & 0 & -\hat{K}^* \\ 0 & \beta \hat{M}_2 & \hat{M}_3^T \\ -\hat{K} & \hat{M}_3 & 0 \end{bmatrix} y^{(\ell)} = \begin{bmatrix} 0 \\ 0 \\ \hat{g} \end{bmatrix} \quad (33)$$

on the components of the current TT block. Here,  $\hat{A} = Y_\ell^T \mathbf{A} Y_\ell$  for  $\mathbf{A} \in \{\mathbf{K}, \mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3\}$  are the submatrices projected onto the Galerkin basis  $Y_\ell$  composed from the frozen TT blocks,

$$Y_\ell(\mathbf{i}, s_{\ell-1} i'_\ell s_\ell) = y^{(1)}(i_1) \cdots y_{:, s_{\ell-1}}^{(\ell-1)}(i_{\ell-1}) \cdot I_{n_\ell}(i_\ell, i'_\ell) \cdot y_{s_\ell, :}^{(\ell+1)}(i_{\ell+1}) \cdots y^{(D)}(i_D), \quad (34)$$

where “.” stands for the full index range.

Under certain assumptions on the original system, one can prove that the block-reduced system (33) is not degenerate.

**Theorem 1.** *Suppose that the symmetric parts of  $\mathbf{K}$  and  $\mathbf{K}^*$  are positive,  $\mathbf{K} + \mathbf{K}^\top > 0$ ,  $\mathbf{K}^* + \mathbf{K}^{*\top} > 0$ . Then the reduced matrix in (33) is invertible.*

*Proof.* Due to the Poincaré theorem, the eigenvalues of an orthogonal projection of a symmetric matrix interlace with the eigenvalues of the original matrix. In particular,

$$\lambda_{\min}(\hat{K} + \hat{K}^\top) = \lambda_{\min}(Y_k^\top (\mathbf{K} + \mathbf{K}^\top) Y_k) \geq \lambda_{\min}(\mathbf{K} + \mathbf{K}^\top) > 0.$$

We also use that  $Y_k$  is orthogonal in the AMEn method. So, the symmetric part of  $\hat{K}$  (as well as of  $\hat{K}^*$ ) is positive. Moreover, by the same interlace theorem we have that  $\hat{M}_1 \geq 0$  and  $\hat{M}_2 > 0$ . Now employ [9, Theorem 3.2], which says that sufficient conditions for the KKT matrix to be invertible are that the matrix of constraints  $[-\hat{K} \quad \hat{M}_3]$  is full rank, and

$$\ker \begin{bmatrix} \hat{M}_1 & 0 \\ 0 & \beta \hat{M}_2 \end{bmatrix} \cap \ker [-\hat{K} \quad \hat{M}_3] = \{0\}.$$

The first condition is fulfilled since  $\hat{K}$  is invertible. To verify the second criterion, consider a vector in the kernel of the constraints, which has the form

$$w = \begin{bmatrix} y \\ u \end{bmatrix} = \begin{bmatrix} \hat{K}^{-1} \hat{M}_3 u \\ u \end{bmatrix}, \quad \forall u \neq 0.$$

Now check if it belongs to the kernel of the other matrix:

$$w^\top \begin{bmatrix} \hat{M}_1 & 0 \\ 0 & \beta \hat{M}_2 \end{bmatrix} w = u^\top \hat{M}_3^\top \hat{K}^{-\top} \hat{M}_1 \hat{K}^{-1} \hat{M}_3 u + \beta u^\top \hat{M}_2 u > 0,$$

since  $u^\top \hat{M}_2 u > 0$  while the first term is non-negative. Together with positive semi-definiteness of  $\text{blkdiag}(\hat{M}_1, \beta \hat{M}_2)$ , this yields that  $w$  is not in its kernel.  $\square$

The second modification to the AMEn algorithm is the unification of sizes and elimination of pressures. Note that the last ( $D$ -th) TT block carries the spatial degrees of freedom, and different components of  $y_\ell$  have different numbers of spatial basis functions:  $N_v$  for  $\mathbf{v}_h$ ,  $\mathbf{u}_h$  and  $\boldsymbol{\lambda}_h$ , and  $N_p$  for  $p_h$  and  $\mu_h$ . The Galerkin matrix  $Y_\ell$  has always the same number of rows; we make it  $N_t n_\xi^m N_v$ , i.e. consistent with  $\mathbf{v}_h$ ,  $\mathbf{u}_h$  and  $\boldsymbol{\lambda}_h$ .

The state and adjoint pressures are on the one hand more difficult, since they are of different sizes, and moreover make  $\mathbf{K}$  itself saddle-point, but on the other hand they are only connected with the other components via a Kronecker-rank-1 matrix  $\mathbf{B} = I_{N_t} \otimes I_{n_\xi}^{\otimes m} \otimes B$ . Therefore, we can compute the pressure components only within the  $D$ -th, the spatial, block. When we proceed to  $\ell < D$  (time and stochastic variables), we eliminate the pressures in a Gauss-Seidel fashion: having originally  $\mathbf{K} = \begin{bmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & 0 \end{bmatrix}$ , we project only the velocity part  $\hat{K} = Y_\ell^T \mathbf{A} Y_\ell$  in the left hand side, and cast  $Y_\ell^T \mathbf{B}^T p_h$  and  $Y_\ell^T \mathbf{B}^T \mu_h$  to the right hand side. The consequence is two-fold: the velocity part  $\mathbf{A}$  is positive definite, hence so is  $\hat{K}$ , and all sizes are now consistent.

## Preconditioning

Even the reduced system (33) can still be rather large: for  $\ell = D$  for example, it is of size  $(3N_v + 2N_p)r_{m+1}$ . Therefore, we use GMRES with the matching preconditioner [45, 5], based on approximating the Schur complement. The reduced matrix admits a (straightforwardly verifiable) decomposition

$$\begin{bmatrix} \hat{M}_1 & 0 & -\hat{K}^* \\ 0 & \beta \hat{M}_2 & \hat{M}_3^T \\ -\hat{K} & \hat{M}_3 & 0 \end{bmatrix} = \begin{bmatrix} I & * & * \\ & I & \\ & & I \end{bmatrix} \begin{bmatrix} & & -\hat{S} \\ & \beta \hat{M}_2 & \hat{M}_3^T \\ -\hat{K} & \hat{M}_3 & 0 \end{bmatrix}, \quad (35)$$

where  $\hat{S} = \hat{K}^* + \hat{M}_1 \hat{K}^{-1} \hat{M}_3 (\beta \hat{M}_2)^{-1} \hat{M}_3^T$ . Next, we use the second (anti-triangular) factor in (35) as a preconditioner. However,  $\hat{K}$  and  $\hat{S}$  must be approximated to make solving linear systems with them feasible.

The first matrix has the Kronecker form

$$\hat{K} = \hat{I} \otimes \mathcal{L} + \hat{C} \otimes \mathcal{M} + \sum_{s=1}^{r_{m+1}} \hat{D}_s \otimes \mathcal{N}_s,$$

where  $\hat{I}$ ,  $\hat{C}$  and  $\hat{D}_s$  are  $I_{N_t} \otimes I_{n_\xi}^{\otimes m}$ ,  $C\tau^{-1} \otimes I_{n_\xi}^{\otimes m}$  and the parts of the convection matrix (31), respectively, projected via  $Y_{m+2}$ . We approximate  $\hat{K}$  by the following Sylvester operator:

$$\tilde{K} = \hat{I} \otimes \left( \mathcal{L} + \sum_{s=1}^{r_{m+1}} \mathbb{E}\lambda(\hat{D}_s) \cdot \mathcal{N}_s \right) + \hat{C} \otimes \mathcal{M},$$

where  $\mathbb{E}\lambda(\hat{D}_s)$  is an average of the eigenvalues of  $\hat{D}_s$  (it is symmetric, since so is the time-stochastic part of  $\mathbf{N}$ ). Now  $\tilde{K}$  can be inverted by the Bartels-Stewart method, since  $\hat{C}$  is small and can be easily Schur-factorized.

For the Schur complement  $\hat{S}$  we use the matching factorisation,

$$\tilde{S} = \left( \hat{K}^* + \frac{1}{c} \hat{M}_1 \right) \hat{K}^{-1} \left( \hat{K} + c \cdot \hat{M}_3 (\beta \hat{M}_2)^{-1} \hat{M}_3^T \right), \quad (36)$$

where  $c > 0$  is a normalization constant. Traditionally it is proposed [45, 5] to take  $c = \sqrt{\beta}$ , and this is indeed an optimal choice for the distributed control. For the boundary control, however,  $\hat{M}_3 (\beta \hat{M}_2)^{-1} \hat{M}_3^T$  is significantly rank-deficient, and an optimal  $c$  may differ. We look for  $c$  that minimizes the residual of the system  $(\tilde{S}^{-1} \hat{S})u = \tilde{S}^{-1} f$  with a random right-hand side  $f$  after 5 GMRES iterations. Since it is difficult to differentiate the GMRES residual w.r.t.  $c$ , we employ the zero-order *golden section* optimization algorithm [12], initialized with an interval  $\log_{10} c \in [\log_{10} \sqrt{\beta} - 6, \log_{10} \sqrt{\beta} + 6]$ . Fortunately, it is sufficient to perform this procedure in the first Picard iteration only, as the optimal  $c$  does not seem to change in the latter. To solve systems with  $\tilde{S}$  efficiently, we also approximate the factors in brackets in (36) by Sylvester matrices, similarly to  $\tilde{K}$ .

The full Algorithm 1 combines the standard Picard iteration for the Navier-Stokes equation [19] and the AMEn iteration in the block TT format [5].

## 5 Numerical results

We implemented the computational codes on the basis of the Matlab TT-Toolbox [35] and the IFISS 3.3 toolbox [42] and run on one core of the `otto` cluster at MPI Magdeburg, an Intel Xeon X5650 @ 2.67GHz. The main focus of this paper is the numerical scheme, hence we vary the model and discretization parameters one by one. The default parameters, unless otherwise stated, are presented in Table 1.

Table 1: Default model and discretization parameters

$\varepsilon$	$\nu$	$\beta$	$N_t$	$T$	$h$	$n_\xi$	$\gamma$	$m$
$10^{-4}$	0.1	$10^{-2}$	$2^{10}$	30	$2^{-3}$	8	4	4

The natural indicators of computational complexity are the CPU time and the number of iterations, while the storage complexity of the TT representation is governed by the TT ranks. We show the maximal TT rank and the memory saving ratio of the TT format, compared to the full tensor representation, i.e.

$$\text{MemR} = \frac{N_t r_1 + \sum_{\ell=1}^m r_\ell r_{\ell+1} n_\xi + r_{m+1} N_v}{N_t n_\xi^m N_v}.$$

The smaller the TT ranks, the smaller is this ratio, which indicates the storage savings.

As a quantity of interest, we compute the average squared norm of vorticity  $\|\nabla \times \mathbf{v}\|^2$ , where the norm is taken in  $L^2([0, T]) \times L^2(\Omega) \times L^2(\mathcal{D})$ , i.e. the doubled first term of the functional (2).

The largest problem was solved with  $m = 8$ ,  $N_t = 1024$  and  $h = 2^{-4}$ ; this value of  $h$  gives  $N_v = 23042$  and  $N_p = 2945$ . Thus, the number of unknowns in the full KKT problem would be  $(3N_v + 2N_p) \cdot n_\xi^m \cdot N_t = 1.29 \times 10^{15}$ . Note that the number of TT elements is

---

**Algorithm 1** Block AMEn-Picard iteration for solving the stochastic inverse Navier-Stokes equations in the TT format

---

- 1: Initialize  $\mathbf{v}_h = \mathbf{u}_h = \boldsymbol{\lambda}_h = 0$ ,  $p_h = \mu_h = 0$ .
  - 2: Initialize the block TT format (32) with  $\iota$  placed in  $y^{(m+2)}$ .
  - 3: **for** iter=1,2,... **do**
  - 4:     Copy  $\bar{\mathbf{v}}_h = \mathbf{v}_h, \bar{p}_h = p_h, \bar{\mathbf{u}}_h = \mathbf{u}_h, \bar{\boldsymbol{\lambda}}_h = \boldsymbol{\lambda}_h, \bar{\mu}_h = \mu_h$ .
  - 5:     Construct the convection matrices  $\mathbf{N}[\bar{\mathbf{v}}_h]$  and  $\mathbf{W}[\bar{\mathbf{v}}_h]$  (31).
  - 6:     Construct and solve the projected system (33) for  $\ell = m + 2$ .
  - 7:     Extract individual components  $\mathbf{v}_h = y_1, p_h = y_2, \mathbf{u}_h = y_3, \boldsymbol{\lambda}_h = y_4, \mu_h = y_5$ .
  - 8:     **if**  $\|\mathbf{v}_h - \bar{\mathbf{v}}_h\| \leq \varepsilon\|\mathbf{v}_h\|$ ,  $\|p_h - \bar{p}_h\| \leq \varepsilon\|p_h\|$  and  $\|\mathbf{u}_h - \bar{\mathbf{u}}_h\| \leq \varepsilon\|\mathbf{u}_h\|$  **then**
  - 9:         Stop.
  - 10:     **end if**
  - 11:     Assemble  $y_\ell$  from  $y_1 = \mathbf{v}_h, y_2 = \mathbf{u}_h, y_3 = \boldsymbol{\lambda}_h$  only.
  - 12:     Apply the SVD to  $y^{(m+2)}$  and move  $\iota$  to  $y^{(m+1)}$ .
  - 13:     **for**  $\ell = m + 1, m, \dots, 1, 2, \dots, m + 1$  **do**
  - 14:         Construct and solve an analog of (33) with fixed pressures,
- $$\begin{bmatrix} \hat{M}_1 & 0 & -\hat{A}^* \\ 0 & \beta\hat{M}_2 & \hat{M}_3^T \\ -\hat{A} & \hat{M}_3 & 0 \end{bmatrix} y^{(\ell)} = \begin{bmatrix} -Y_\ell^T \mathbf{B}^T \mu_h \\ 0 \\ \hat{g} - Y_\ell^T \mathbf{B}^T p_h \end{bmatrix}$$
- 15:         Apply the SVD to  $y^{(\ell)}$ .
  - 16:         **if**  $\ell$  is increasing **then**
  - 17:             Move  $\iota$  to  $y^{(\ell+1)}$ .
  - 18:         **else**
  - 19:             Move  $\iota$  to  $y^{(\ell-1)}$ .
  - 20:         **end if**
  - 21:     **end for**
  - 22: **end for**
- 

$(3N_v + 2N_p)r + mn_\xi r^2 + N_t r$ ; but since the first term is dominating, it suffices to estimate the whole cost with the first term. The maximum ranks are 70 for the distributed control and 120 for the boundary control cases (see Figures 6 and 13 right, respectively). This gives roughly 5 and 9 millions of unknowns.

## 5.1 Distributed control

### 5.1.1 Discretization parameters

In the first series of tests we consider the distributed control, applied in the right hand side of (9). We start with verifying correctness of the discretization schemes. We show the results for different spatial mesh sizes in Fig. 2, and for different time step sizes in Fig. 3.

The number of spatial degrees of freedom grows quadratically with the number of steps in each dimension. Since we use a direct solver for the spatial Stokes-like problems in (36), the computational complexity is higher; in particular, it grows proportionally to  $h^{-2.5}$  in Fig. 2.



To estimate the discretization error, we take the vorticity from the finest grid  $h = 2^{-4}$  as the reference value  $\|\nabla \times \mathbf{v}_*\|$ , and compare it with the values on coarser grids. We observe the convergence order  $h^{1.5}$ , which is reasonable for the backstep domain, containing a corner with an angle  $3\pi/2$ ; see e.g., [3, 2].

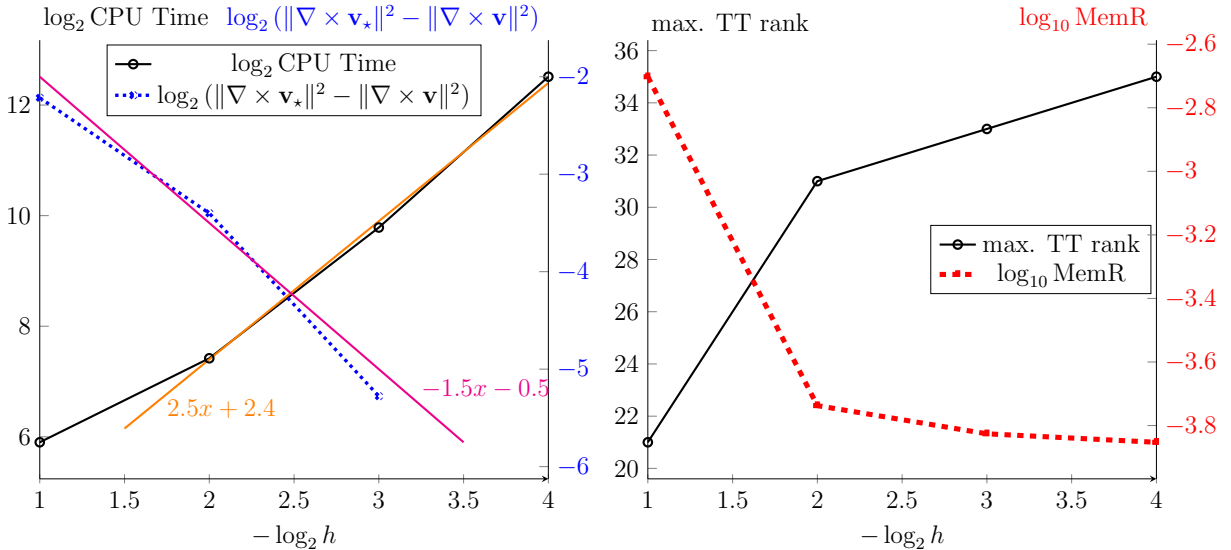
In order to keep the same number of stochastic variables throughout the spatial mesh test, we restrict it to  $m = 2$ . This is necessary since the higher KLE components in (4) are not resolved on the coarsest grid in this test, and the problem would be in a severely pre-asymptotic regime otherwise.

The TT ranks grow very moderately with the grid sizes, with the asymptotic dependence close to logarithmic. Since the full tensor storage, proportional to both  $N_v$  and  $N_t$  grows much faster, this leads actually to a decrease of the memory ratio with both spatial (Fig. 2) and time (Fig. 3) grid refinement.

The CPU time versus the time grid size demonstrates two regions. For smaller numbers of time steps, solution of the spatial problems is dominating in the total complexity, while the TT ranks grow logarithmically. This leads to a logarithmic growth of the CPU time as well. When the number of time steps is large, the temporal problems become the most time consuming, and the CPU time starts to grow linearly with  $N_t$  (the actual estimated order is 1.2 due to the additional logarithmic growth of the ranks).

The implicit Euler time discretization manifests an expected convergence of the first order. Here the reference vorticity  $\|\nabla \times \mathbf{v}_*\|$  is computed at the  $N_t = 2^{15}$  grid.

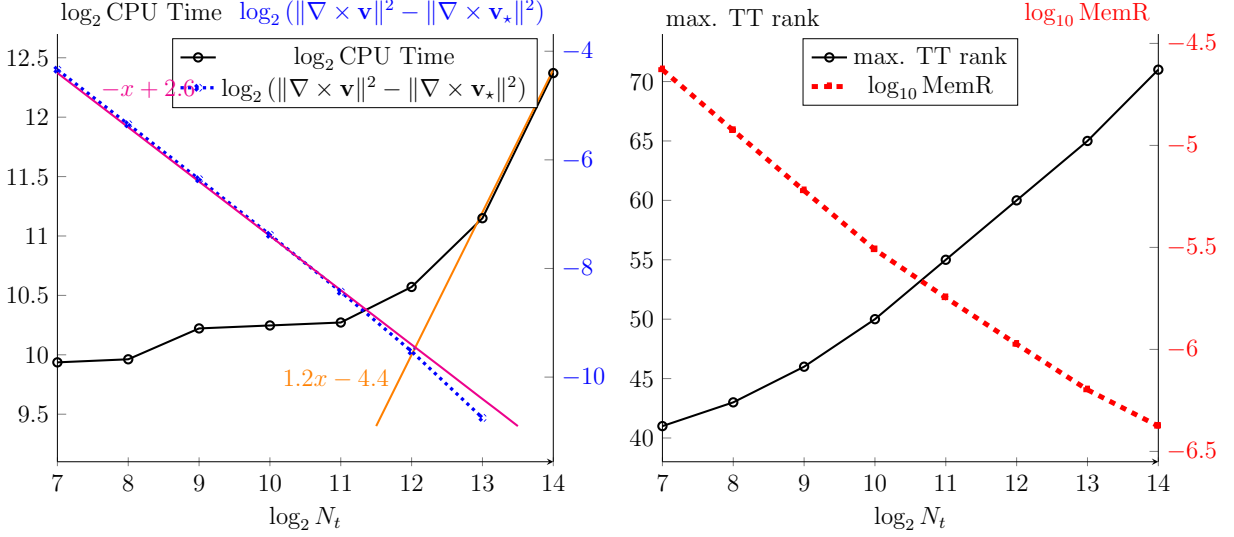
Figure 2: CPU time and convergence w.r.t. grid refinement (left) and TT rank and memory reduction ratio (right) for different spatial mesh sizes. Change of defaults:  $m = 2$ .



### 5.1.2 Robustness with respect to model parameters: viscosity, KLE decay rate and dimension

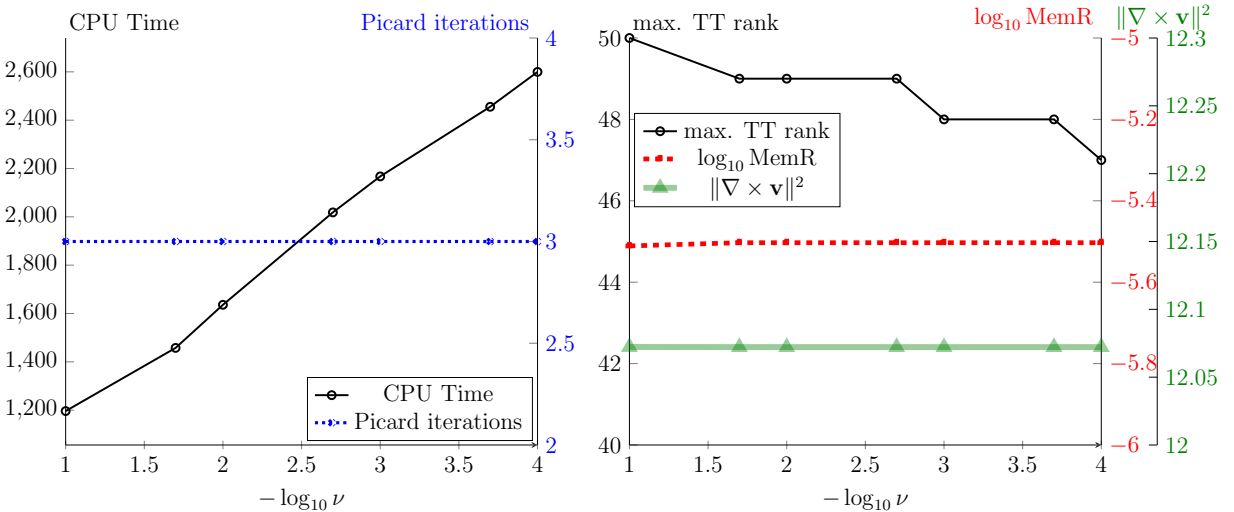
Having checked the discretization schemes, we can vary the other model parameters and see how the performance and the quantity of interest behave. In Fig. 4 we vary the viscosity from

Figure 3: CPU time and convergence w.r.t. grid refinement (left) and TT rank and memory reduction ratio (right) for different numbers of time steps.



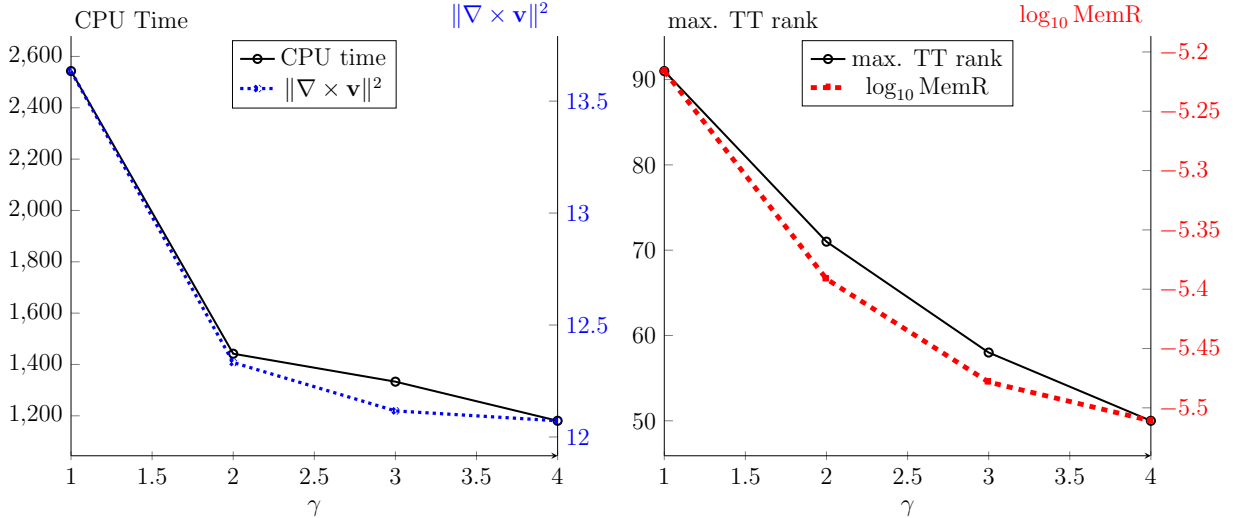
$10^{-1}$  to  $10^{-4}$  and track the computational time and the number of Picard iterations (left), as well as the storage indicators and the quantity of interest (right). We see that the CPU time grows only logarithmically with the Reynolds number, and 3 Picard iterations are enough for the nonlinear system to converge. This is also the case for other experiments with a *sufficient* control, i.e. with the distributed control and small enough  $\beta$ , that was observed previously [14] for a deterministic problem as well. The TT rank (and hence memory ratio) and the vorticity remain at almost the same level (compared to variation of other parameters and other tests), which indicates that the minimal-vorticity solution is similar to the Stokes flow, fully defined by the domain and boundary conditions, rather than the viscosity.

Figure 4: CPU time and the number of Picard iterations (left) and TT rank, memory reduction ratio and the vorticity (right) for different kinematic viscosities.



There are two parameters defining the “randomness” of the problem: the decay rate of the KLE coefficients  $\gamma$ , and the total number of KLE components after truncation  $m$ , see (5). The smaller is  $\gamma$ , the slower is the decay, and hence the larger is the number of “effective” dimensions. This makes the problem more difficult to solve, as we observe in Fig. 5.

Figure 5: CPU time and the vorticity (left) and TT rank and the memory reduction ratio (right) for different KLE decay rates  $\gamma$ .



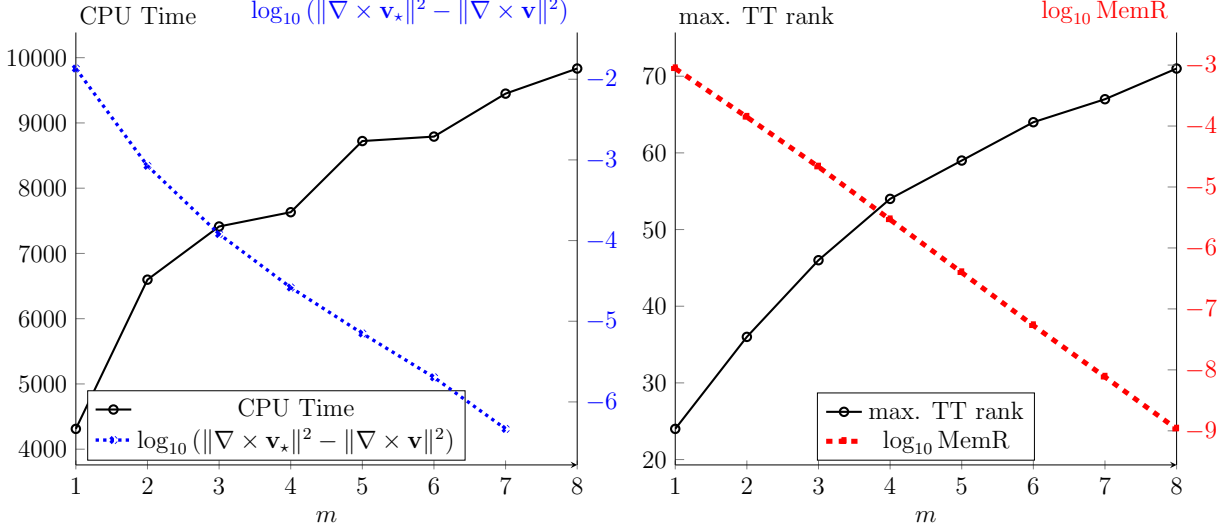
Increasing the number of stochastic variables faces the same issue as decreasing the number of spatial grid points: a coarse spatial grid may not resolve the latter KLE terms. In order to vary  $m$  up to 8, we employ the spatial grid with  $h = 2^{-4}$ . The results are shown in Fig. 6. The CPU times grows milder than linearly with  $m$  since (a) the most time-consuming stage is still the solution of the spatial problems, and (b) the model saturates with larger  $m$  due to the decaying KLE series. The latter phenomenon is also illustrated by convergence of the vorticity norm with  $m$ .

This example illustrates how the TT decomposition gets rid of the curse of dimensionality. The total number of degrees of freedom in a full tensor would grow exponentially in  $m$ , while the TT ranks (and hence the TT storage) grow milder than linearly. This leads to an exponential decay of the memory reduction ratio, which reaches 9 orders of magnitude for the largest dimension.

### 5.1.3 Influence of the regularization

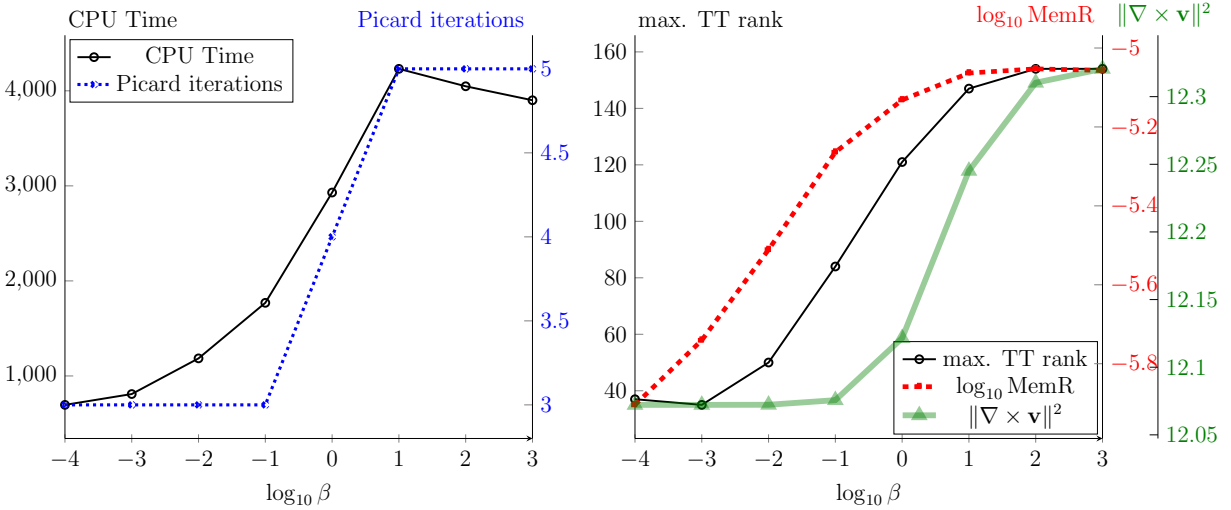
In Fig. 7, we investigate how the model and the scheme depend on the regularization parameter  $\beta$ . Setting  $\beta \gg 1$  means solving almost an uncontrolled problem, while  $\beta < 1$  corresponds to a control with a certain “power”. We see that the CPU time and TT ranks grow significantly with  $\beta$ , when the model switches from effectively Stokes to an essentially nonlinear Navier-Stokes regime. This is also reflected by a larger number of Picard iterations, needed for the system to converge. The norm of vorticity starts growing fast when  $\beta$  exceeds 1. On the other hand, for very large  $\beta > 10$  we observe a slight decrease of the CPU time due to a smaller

Figure 6: CPU time and the vorticity (left) and TT rank and the memory reduction ratio (right) for different numbers of stochastic variables  $m$ . Change of defaults:  $h = 2^{-4}$ .



number of the local GMRES iterations: the KKT matrix becomes anti-diagonally dominant, and the preconditioner (36) becomes more efficient.

Figure 7: CPU time and the number of Picard iterations (left) and TT rank, memory reduction ratio and vorticity (right) for different regularization parameters.



Figures 8 and 9 show the first two moments of the velocity and control for the final time. For comparison, the uncontrolled flow is shown in Figure 10. We can notice that the uncontrolled flow is more aligned to the top of the domain, while developing an eddy below the step. The controlled minimal vorticity flow reflects the regime of a larger viscosity. Moreover, the reattachment point shrinks significantly compared to that in Figure 10.

Figure 8: Plots of the mean (top) and variance (bottom) of the stream function for an unsteady distributed control flow with  $\nu = 1/50$  at  $t = 10$

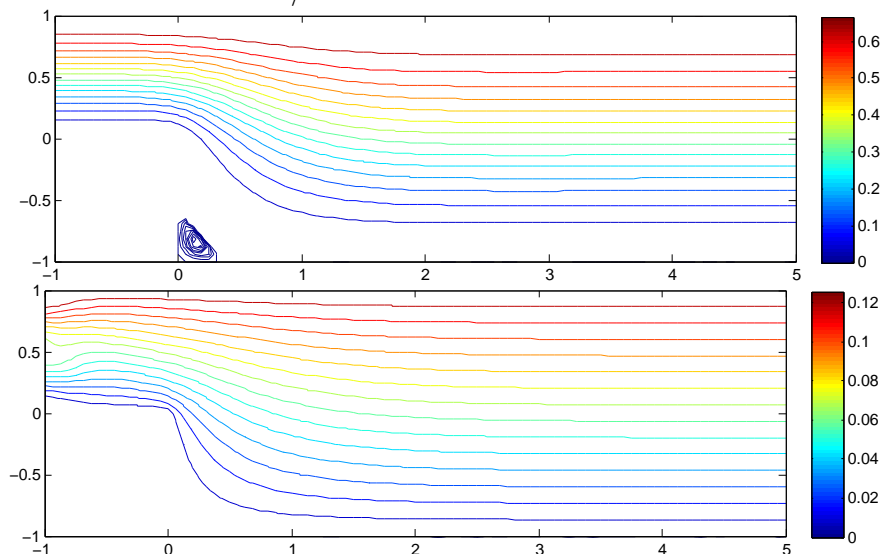
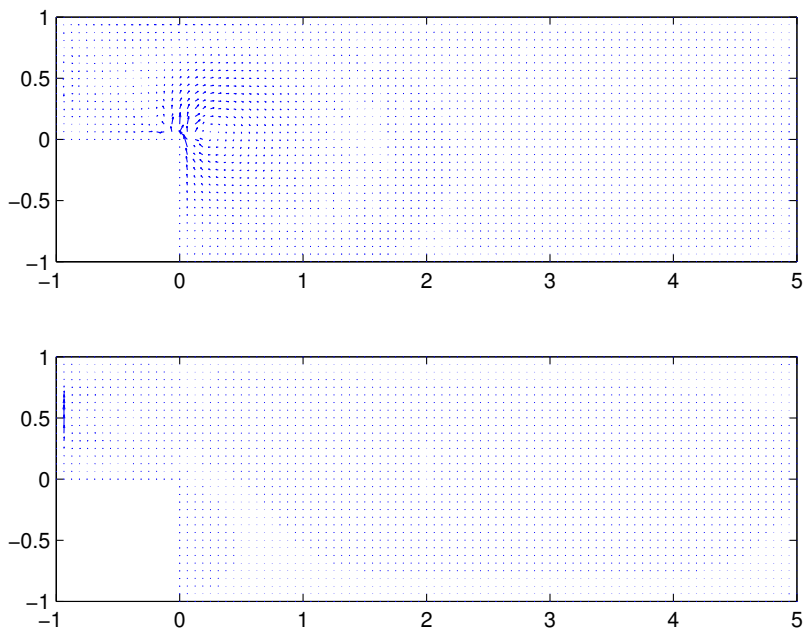


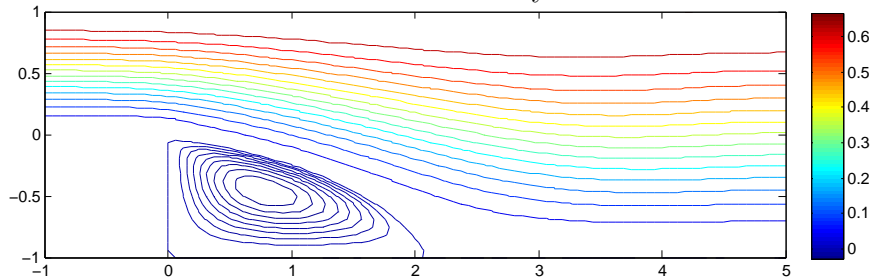
Figure 9: Plots of the mean (top) and variance (bottom) of the control for an unsteady distributed control flow with  $\nu = 1/50$ ,  $n_t = 1024$ .



## 5.2 Time-dependent boundary control problem

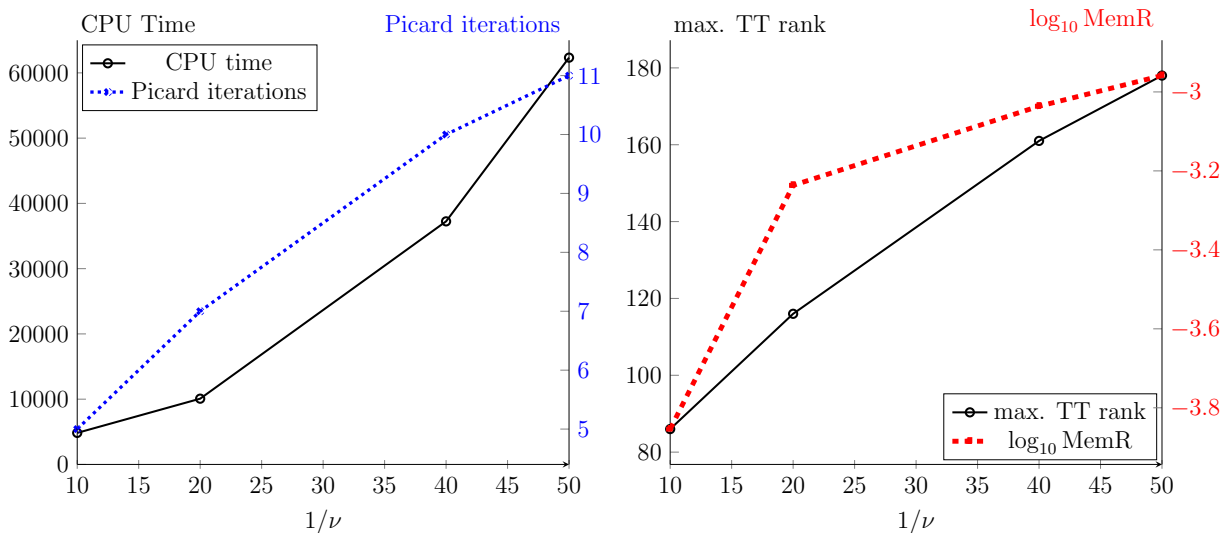
Now we switch to the boundary control, applied as a Neumann boundary condition on the step wall (9). The problem is more difficult since the partial control cannot fully steer the flow to the Stokes regime. Strongly nonlinear effects that remain in the flow inflate the TT ranks and slow down the convergence.

Figure 10: Stream function for an unsteady uncontrolled flow at  $t = 10$ .



In particular, in Fig. 11 we show how the scheme behaves for different viscosities. We see that, compared to the distributed control case, even the case  $\nu = 1/50$  is difficult to solve due to very large TT ranks. On the other hand, changing the regularization parameter (Fig. 12) does not influence the cost and storage complexities as much as in the distributed control case. In fact, taking small  $\beta$  in the partial control case leads to a poor performance of the Schur complement preconditioner, and hence a large CPU time due to a large number of local GMRES iterations. Nevertheless, smaller vorticity norm and number of Picard iterations for  $\beta < 1$  indicates that the target functional is optimized as expected.

Figure 11: CPU time and the number of Picard iterations (left) and TT rank and memory reduction ratio (right) for different viscosities with the boundary control.



Concerning the stochastic parameters, we see qualitatively the same behaviour as in the distributed control case: all complexity indicators increase for smaller  $\gamma$  and larger  $m$ , although the CPU time and TT ranks saturate with  $m$ , see Fig. 13.

### 5.3 Stationary boundary control problem

One of the sources of the rank inflation in the previous section is the simultaneous storage of all time snapshots in a single TT representation. In the next tests we consider the stationary Navier-Stokes equations as constraints. In Fig. 14, we vary the viscosity, and in Fig. 15

Figure 12: CPU time and the number of Picard iterations (left) and TT rank and vorticity (right) for different regularization parameters of the boundary control.

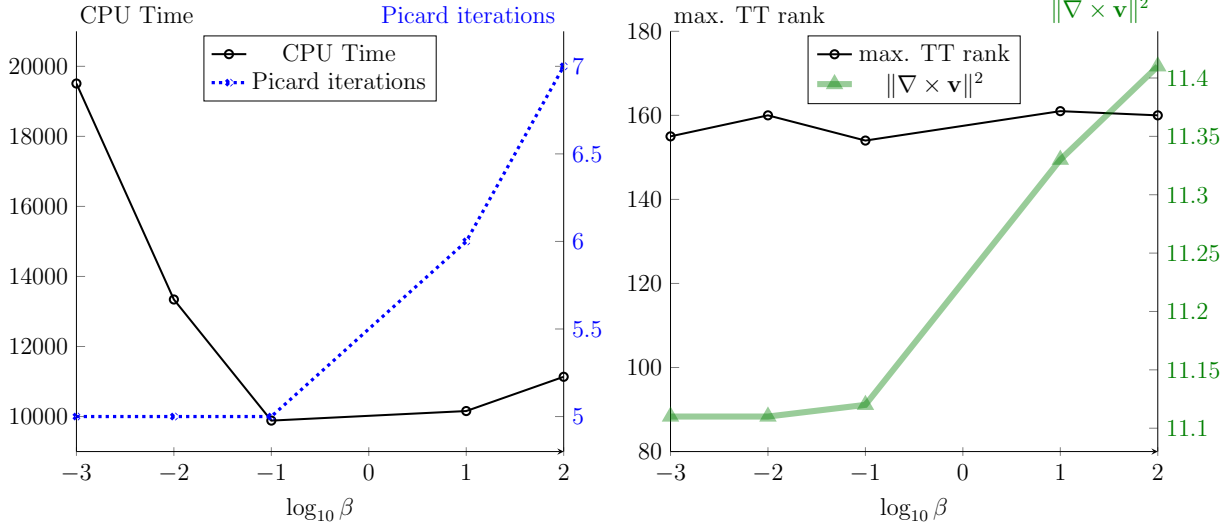
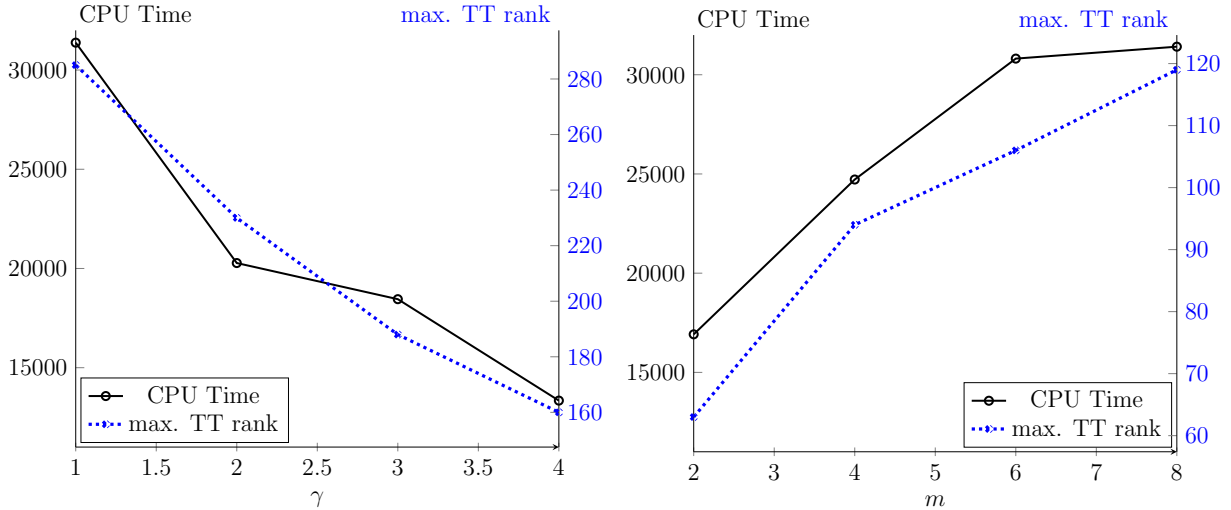


Figure 13: CPU time and the TT rank for different KLE decays (left) and KLE dimensions (right), boundary control.



we investigate the influence of the regularization parameter. We see that the results reflect qualitatively the behaviour of the time-dependent problem with the boundary control, i.e. the stochastic components are also strongly coupled due to the nonlinearity, and it needs more Picard iterations and larger TT ranks for the solution to converge in a low viscosity regime. Nonetheless, the TT ranks (and hence the computational times) are much smaller than in the time-dependent case.

Finally, Figures 16 and 17, depict the moments of the velocity and control of the stationary boundary controlled flow. The boundary controlled flow still manifests an eddy around the corner. Nonetheless, it corresponds to a smaller vorticity than in the uncontrolled regime (Fig. 10), where there is a rather sharp interface between the main stream and the corner area. We

Figure 14: CPU time and number of Picard iterations (left) and TT rank and memory reduction ratio (right) for different viscosities, boundary control, stationary problem.

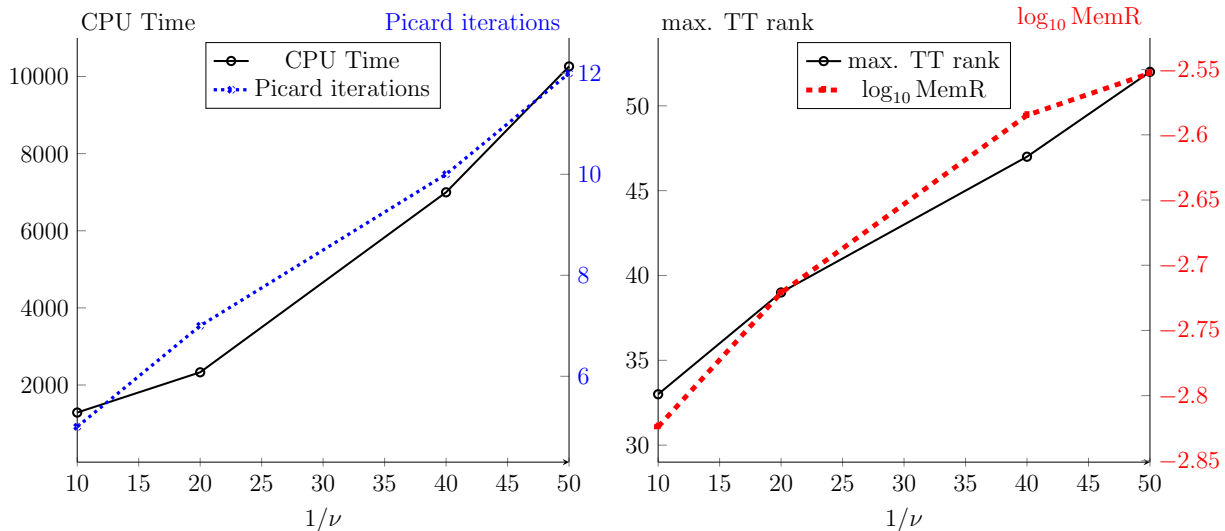
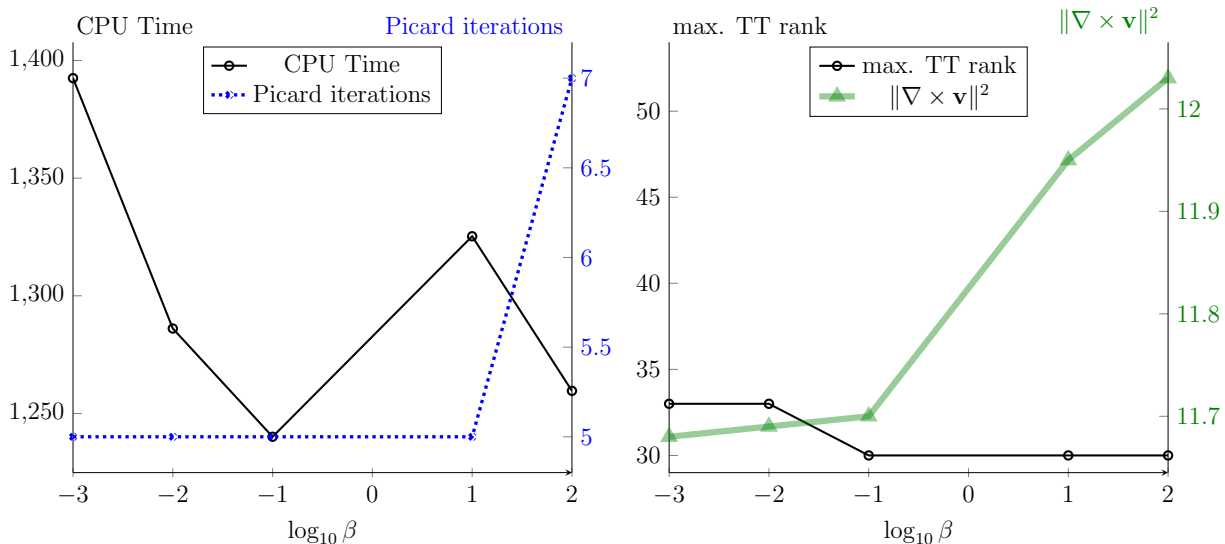


Figure 15: CPU time and number of Picard iterations (left) and TT rank and vorticity (right) for different regularization parameters, boundary control, stationary problem.



also remark that the number of Picard iterations remains generally below 10, with just a few exceptions.

## 6 Conclusions and outlook

We demonstrated the applicability of low-rank tensor decompositions to the solution of optimal control problems constrained by unsteady Navier-Stokes equations with stochastic inputs. This problem has a threefold challenge: a nonlinear time-dependent PDE, an optimization



Figure 16: Plots of the mean (top) and variance (bottom) of the stream function for a stationary boundary control flow with  $\nu = 1/50$ .

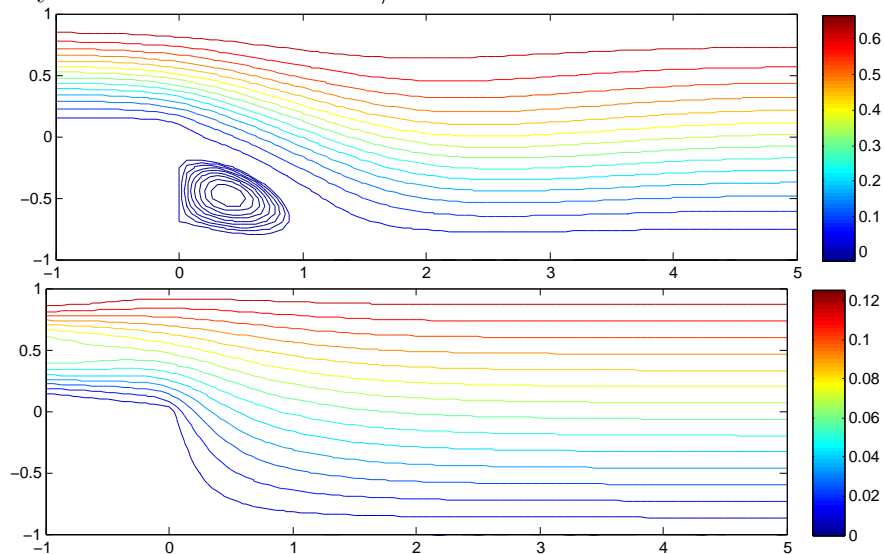
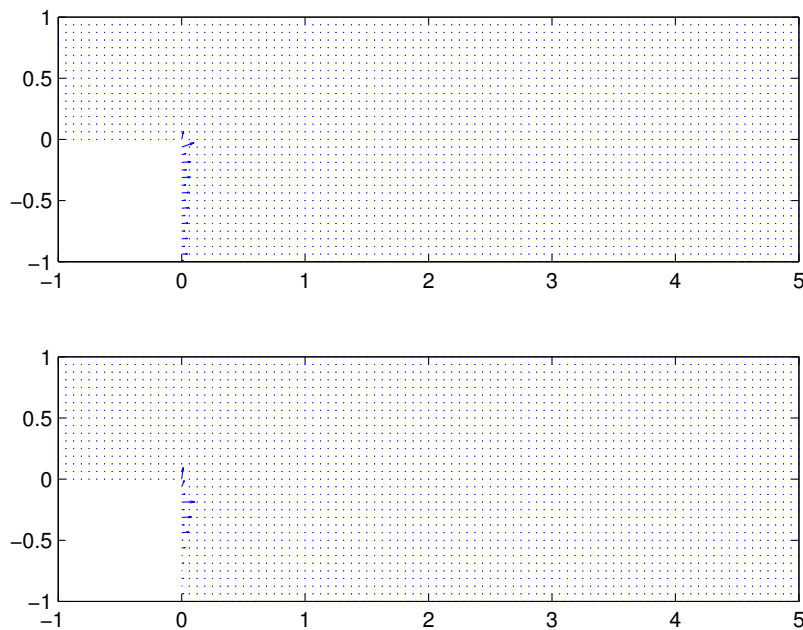


Figure 17: Plots of the mean (top) and variance (bottom) of control for a stationary boundary control flow with  $\nu = 1/50$ .



problem using a Lagrangian approach, and random inputs. In a classical stochastic Galerkin discretization one needs to multiply the numbers of degrees of freedom coming from space, time and stochastic quantities. In particular, the outer optimization implies storing all time snapshots, and random quantities are introduced as independent variables.

We never store or compute all elements of such a solution as this would quickly become

infeasible even on large computers. Instead we compress them into the Tensor Train representation. A crucial part of the scheme is an alternating iterative solver, which computes the TT factors directly. Although known in the multilinear algebra community, this idea required substantial modifications in order to be applicable to the OPNS. We preserve the saddle-point structure in the reduced model and accommodate components of different sizes, such as the boundary control. Moreover, given a low-rank representation of the previous solution, we assemble the linearized operator also in a low-rank form. The scheme provides a significant reduction of complexity, up to 9 orders of magnitude for the largest number of random variables.

## References

- [1] A. ALI, M. HINZE, AND E. ULLMANN, *Multilevel Monte Carlo analysis for optimal control of elliptic PDEs with random coefficients*, arXiv:1602.08943, (2016).
- [2] I. BABUŠKA AND B. Q. GUO, *The hp version of the finite element method for domains with curved boundaries*, *SIAM Journal on Numerical Analysis*, 25 (1988), pp. 837–861.
- [3] ———, *Regularity of the solution of elliptic problems with piecewise analytic data. Part I. Boundary value problems for linear elliptic equation of second order*, *SIAM Journal on Mathematical Analysis*, 19 (1988), pp. 172–203.
- [4] I. BABUŠKA, R. TEMPONE, AND G. ZOURARIS, *Galerkin finite element approximations of stochastic elliptic partial differential equations*, *SIAM Journal on Numerical Analysis*, 42 (2004), pp. 800–825.
- [5] 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 (2016), pp. 26–54.
- [6] P. BENNER AND M. W. HESS, *Reduced Basis Modeling for Uncertainty Quantification of Electromagnetic Problems in Stochastically Varying Domain*, in *Scientific Computing in Electrical Engineering*, A. Bartel, M. Clemens, M. Guenther, and E. J. W. ter Maten, eds., Springer, 2016, pp. 215 – 222.
- [7] P. BENNER, A. ONWUNTA, AND M. STOLL, *Low-rank solution of unsteady diffusion equations with stochastic coefficients*, *SIAM/ASA Journal on Uncertainty Quantification*, 3 (2015), pp. 622 – 649.
- [8] ———, *Block-diagonal preconditioning for optimal control problems constrained by PDEs with uncertain inputs*, *SIAM Journal on Matrix Analysis and Applications*, 37 (2016), pp. 491 – 518.
- [9] M. BENZI, G. H. GOLUB, AND J. LIESEN, *Numerical solution of saddle point problems*, *Acta Numer*, 14 (2005), pp. 1–137.

- [10] A. BESPALOV, C. E. POWELL, AND D. SILVESTER, *Energy norm a posteriori error estimation for parametric operator equations*, SIAM Journal on Scientific Computing, 36 (2013), pp. A339 – A363.
- [11] A. BORZI AND G. VON WINCKEL, *Multigrid methods and sparse-grid collocation techniques for parabolic optimal control problems with random coefficients*, SIAM Journal on Scientific Computing, 31 (2009), pp. 2172 – 2192.
- [12] W. CHENEY AND D. KINCAID, *Numerical Mathematics and Computing*, Brooks/Cole Publishing Co., Pacific Grove, CA, USA, 6th ed., 2007.
- [13] A. J. CHORIN AND F. LU, *Discrete approach to stochastic parametrization and dimension reduction in nonlinear dynamics*, PNAS, 112 (2015), pp. 9804–9809.
- [14] S. DOLGOV AND M. STOLL, *Low-rank solution to an optimization problem constrained by the Navier–Stokes equations*, SIAM Journal on Scientific Computing, 39 (2017), pp. A255–A280.
- [15] S. V. DOLGOV, B. N. KHOROMSKIJ, I. V. OSELEDETS, AND D. V. SAVOSTYANOV, *Computation of extreme eigenvalues in higher dimensions using block tensor train format*, Computer Physics Communications, 185 (2014), pp. 1207–1216.
- [16] S. V. DOLGOV AND D. V. SAVOSTYANOV, *Alternating minimal energy methods for linear systems in higher dimensions*, SIAM Journal on Scientific Computing, 36 (2014), pp. A2248–A2271.
- [17] H. ELMAN, O. G. ERNST, D. P. O’LEARY, AND M. STEWART, *Efficient iterative algorithms for the stochastic finite element method with applications to acoustic scattering*, Computer Methods in Applied Mechanics and Engineering, 194 (2005), pp. 1037–1055.
- [18] H. ELMAN AND D. FURNIVAL, *Solving steady-state diffusion problem using multigrid*, IMA Journal of Numerical Analysis, 27 (2007), pp. 675–688.
- [19] H. ELMAN, D. SILVESTER, AND A. WATHEN, *Finite Elements and Fast Iterative Solvers*, vol. Second Edition, Oxford University Press, 2014.
- [20] R. G. GHANEM AND P. SPANOS, *Stochastic Finite Elements: A Spectral Approach*, Springer-Verlag, New York, 1996.
- [21] M. D. GUNZBURGER, C. G. WEBSTER, AND G. ZHANG, *Stochastic finite element methods for equations with random input data*, Acta Numerica, 23 (2014), pp. 521–650.
- [22] W. HACKBUSCH, *Tensor Spaces And Numerical Tensor Calculus*, Springer–Verlag, Berlin, 2012.
- [23] R. HERZOG AND E. SACHS, *Preconditioned conjugate gradient method for optimal control problems with control and state constraints*, SIAM Journal on Matrix Analysis and Applications, 31 (2010), pp. 2291 – 2317.

- [24] M. HINTERMÜLLER, K. ITO, AND K. KUNISCH, *The primal-dual active set strategy as a semi-smooth Newton method*, SIAM Journal on Optimization, 13 (2002), pp. 865 – 888.
- [25] MICHAEL HINZE, *Optimal and instantaneous control of the instationary Navier-Stokes equations*, Habilitation, TU Berlin, 2000.
- [26] S. HOLTZ, T. ROHWEDDER, AND R. SCHNEIDER, *The alternating linear scheme for tensor optimization in the tensor train format*, SIAM Journal on Scientific Computing, 34 (2012), pp. A683–A713.
- [27] K. ITO AND K. KUNISCH, *Semi-smooth Newton methods for state-constrained optimal control problems*, Systems and Control Letters, 50 (2003), pp. 221 – 228.
- [28] E. JECKELMANN, *Dynamical density matrix renormalization group method*, Physical Review B, 66 (2002), p. 045114.
- [29] CH. KANZOW, *Inexact semi-smooth Newton methods for large-scale complementarity problems*, Optimization Methods and Software, 19 (2004), pp. 309 – 325.
- [30] B. N. KHOROMSKIJ, *Tensor numerical methods for multidimensional PDEs: theoretical analysis and initial applications*, ESAIM Proceedings, 48 (2015), pp. 1–28.
- [31] D. P. KOURI, M. HEINKENSCHLOSS, D. RIDZAL, AND B. G. VAN BLOEMEN WAANDERS, *A trust-region algorithm with adaptive stochastic collocation for PDE optimization under uncertainty*, SIAM Journal on Scientific Computing, 35 (2013), pp. A1847 – A1879.
- [32] O. P. LE MAÎTRE, O. M. KNIO, B. J. DEBUSSCHERE, H. N. NAJM, AND R. G. GHANEM, *A multigrid solver for two-dimensional stochastic diffusion equations*, Computer Methods in Applied Mechanics and Engineering, 192 (2003), pp. 4723–4744.
- [33] F. NOBILE AND R. TEMPONE, *Analysis and implementation issues for the numerical approximation of parabolic equations with random coefficients*, International Journal for Numerical Methods in Engineering, 80 (2009), pp. 979 – 1006.
- [34] I. V. OSELEDETS, *Tensor-train decomposition*, SIAM Journal on Scientific Computing, 33 (2011), pp. 2295–2317.
- [35] I. V. OSELEDETS, S. DOLGOV, V. KAZEEV, D. SAVOSTYANOV, O. LEBEDEVVA, P. ZHLOBICH, T. MACH, AND L. SONG, *TT-Toolbox*, 2016. <https://github.com/oseledets/TT-Toolbox>.
- [36] J. W. PEARSON, M. STOLL, AND A. WATHEN, *Preconditioners for state constrained optimal control problems with Moreau-Yosida penalty function*, Numerical Linear Algebra with Applications, (2011), pp. 81 – 97.
- [37] D. PEREZ-GARCIA, F. VERSTRAETE, M. M. WOLF, AND J. I. CIRAC, *Matrix product state representations*, Quantum Info. Comput., 7 (2007), pp. 401–430.

- [38] M. PORCELLI, V. SIMONCINI, AND M. TANI, *Preconditioning of active-set Newton methods for PDE-constrained optimal control problems*, SIAM Journal on Scientific Computing, 37 (2015), pp. S472 – S502.
- [39] C. E. POWELL AND H. ELMAN, *Block-diagonal preconditioning for spectral stochastic finite-element systems*, IMA Journal of Numerical Analysis, 29 (2009), pp. 350–375.
- [40] E. ROSSEEL AND G. N. WELLS, *Optimal control with stochastic PDE constraints and uncertain controls*, Computer Methods in Applied Mechanics and Engineering, 213-216 (2012), pp. 152–167.
- [41] U. SCHOLLWÖCK, *The density–matrix renormalization group*, Reviews of Modern Physics, 77 (2005), pp. 259–315.
- [42] D. SILVESTER, H. ELMAN, AND A. RAMAGE, *Incompressible Flow and Iterative Solver Software (IFISS) version 3.4*, August 2015. <http://www.manchester.ac.uk/ifiss/>.
- [43] J. SOGN, *Stabilized finite element methods for the Brinkman equation on fitted and fictitious domains*, Master’s Thesis, University of Oslo, 2014.
- [44] M. STOLL AND T. BREITEN, *A low-rank in time approach to PDE-constrained optimization*, SIAM Journal on Scientific Computing, 37 (2015), pp. B1 – B29.
- [45] M. STOLL AND A. WATHEN, *All-at-once solution of time-dependent Stokes control*, Journal of Computational Physics, 232 (2013), pp. 498–515.
- [46] H. TIESLER, R. M. KIRBY, D. XIU, AND T. PREUSSER, *Stochastic collocation for optimal control problems with stochastic PDE constraints*, SIAM Journal on Control and Optimization, 50 (2012), pp. 2659 – 2682.
- [47] F. TRÖLTZSCH, *Optimal Control of Partial Differential Equations: Theory, Methods and Applications*, American Mathematical Society, 2010.
- [48] S. R. WHITE, *Density matrix algorithms for quantum renormalization groups*, Physical Review B, 48 (1993), pp. 10345–10356.
- [49] I. WIENER, *The homogeneous chaos*, American Journal of Mathematics, 60 (1938), pp. 897 – 936.
- [50] D. XIU AND G. E. KARNIADAKIS, *A new stochastic approach to transient heat conduction modeling with uncertainty*, International Journal of Heat & Mass Transfer, 46 (2003), pp. 4681–4693.