

Low-rank solution of an optimal control problem constrained by random Navier-Stokes equations

Simulation

Low-rank solution of an optimal control problem constrained by random Navier-Stokes equations

Peter Benner¹ | Sergey Dolgov² | Akwum Onwunta^{3,5}  | Martin Stoll⁴

¹Computational Methods in Systems and Control Theory, Max-Planck-Institut für Dynamik komplexer technischer Systeme, Magdeburg, Germany

²Department of Mathematical Sciences, University of Bath, Bath, UK

³Department of Computer Science, University of Maryland, College Park, Maryland

⁴Faculty of Mathematics, Professorship Scientific Computing, Technische Universität Chemnitz, Chemnitz, Germany

⁵Max-Planck-Institut für Dynamik komplexer technischer Systeme, Magdeburg

Correspondence

Akwum Onwunta, Department of Computer Science, University of Maryland, College Park, MD 20742.
Email: onwunta@cs.umd.edu

Funding information

EPSRC Fellowship, Grant/Award Number: EP/M019004/1; U. S. Department of Energy Office of Advanced Scientific Computing Research, Grant/Award Number: DE-SC000930

[Correction added on 12 June 2020 after first online publication: A second affiliation for Akwum Onwunta has been added in this version.]

Summary

We develop a low-rank tensor decomposition algorithm for the numerical solution of a distributed optimal control problem constrained by two-dimensional time-dependent Navier-Stokes equations with a stochastic inflow. The goal of optimization is to minimize the flow vorticity. The inflow boundary condition is assumed to be an infinite-dimensional random field, which is parametrized using a finite- (but high-) dimensional Fourier expansion and discretized using the stochastic Galerkin finite element method. This leads to a prohibitively large number of degrees of freedom in the discrete solution. Moreover, the optimality conditions in a time-dependent problem require solving a coupled saddle-point system of nonlinear equations on all time steps at once. For the resulting discrete problem, we approximate the solution by the tensor-train (TT) decomposition and propose a numerically efficient algorithm to solve the optimality equations directly in the TT representation. This algorithm is based on the alternating linear scheme (ALS), but in contrast to the basic ALS method, the new algorithm exploits and preserves the block structure of the optimality equations. We prove that this structure preservation renders the proposed block ALS method well posed, in the sense that each step requires the solution of a nonsingular reduced linear system, which might not be the case for the basic ALS. Finally, we present numerical experiments based on two benchmark problems of simulation of a flow around a von Kármán vortex and a backward step, each of which has uncertain inflow. The experiments demonstrate a significant complexity reduction achieved using the TT representation and the block ALS algorithm. Specifically, we observe that the high-dimensional stochastic time-dependent problem can be solved with the asymptotic complexity of the corresponding deterministic problem.

KEYWORDS

iterative methods, low-rank solution, PDE-constrained optimization, preconditioning, saddle-point system, Schur complement, stochastic Galerkin system, tensor train format

1 | INTRODUCTION

We consider the numerical simulation of a stochastic optimal control problem (SOCP). More precisely, our goal in this work is to efficiently solve the optimal control of an incompressible flow problem governed by time-dependent Navier-Stokes equations with random inputs. This problem can be computationally challenging due to the nonlinear constraint.¹ The computational complexity associated with the problem further stems from the uncertain inputs.^{2,3} A viable solution approach to optimization problems with stochastic constraints employs the spectral stochastic Galerkin finite element method (SGFEM). In particular, the steady-state Navier-Stokes equations were solved with SGFEM in References 4-7. However, this intrusive approach* leads to the so-called *curse of dimensionality*, in the sense that the number of expansion coefficients of the discrete solution grows exponentially in the number of random variables and quickly becomes intractable for direct calculation.^{2,8,9}

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. Since the weak solution of the parametrized PDE is often a holomorphic function with respect to the parameters,¹¹ a suitable family of orthogonal polynomials is the mostly used ansatz. One approach to reduce the curse of dimensionality is to sparsify explicitly the set of polynomials, by bounding, for example, the total degree.^{4,6,12-14} Alternatively, one can consider the full Cartesian space of polynomials with bounded individual degree, but never store the expansion coefficients explicitly. Instead, one approximates the tensor of coefficients by a low-rank decomposition. We develop a low-rank technique based on recent advances in numerical tensor methods^{15,16} for the efficient solution of the SOCP. Our aim in this article is to lift the curse of dimensionality inherent in the SOCP and allow for the efficient simulations of the model on 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.

In the last decade, the use of SGFEM discretization with low-rank approximations to solve high-dimensional and parametrized problems was studied extensively (see References 2,7,17-19 and the references therein). In particular, References 18 and 19 approximate the solution in the canonical tensor format. Later on, compression with the tensor train and hierarchical tensor formats was carried out. On the other hand, Reference 19 focuses on the convergence rate of approximations by finite sums of rank-1 tensors solution of the problems. Adaptive tensor-train method presented in Reference 17 is aimed at minimizing the error resulting from the numerical approximation of the solution of the SGFEM discretization of parameterized PDEs. In Reference 2, the authors consider the numerical solution of Stokes-Brinkman optimal control problem with random inputs. By employing the SGFEM discretization and the tensor-train format, the article efficiently approximates the numerical solution of the high-dimensional linear system by significantly reducing the storage requirements of the original problem. In the present article, the authors extend the ideas developed in Reference 2 to the more difficult case of optimization problems constrained by Navier-Stokes equations with random inputs. Moreover, we prove well posedness of the approach suggested in Reference 2 under the problem formulation under consideration.

Alternative approaches to tackle optimization problems with stochastic constraints include stochastic collocation schemes,²⁰⁻²² as well as Monte Carlo methods.²³ These methods are essentially sampling based and nonintrusive. However, for optimization problems, the SGFEM exhibits superior performance compared with the stochastic collocation method.⁹ This is due to the fact that, unlike SGFEM, the nonintrusivity 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.²²

Low-rank tensor methods for optimal control problems with uncertain inputs have also been considered in Reference 24, where the authors focus mainly on a solver for a stochastic optimal control problem with a low-dimensional but pointwise constrained control. Such problems can be tackled via, for instance, semismooth Newton algorithms.²⁵⁻²⁷ In our work, however, we do not consider the case of state- or control- or mixed control-state-constrained problems.²⁸⁻³⁰

The rest of the article is organized as follows. First, we present in Section 2 the PDE-constrained optimization problem that we would like to solve, as well as the necessary mathematical concepts and notation on which we shall rely in the rest

*SGFEM techniques are "intrusive" in the sense that the software packages for the associated deterministic problems cannot be directly reused, since all spatial and stochastic degrees of freedom are coupled.

of our discussion. Next, we proceed to Section 3 to discuss the SGFEM discretization of the problem. Section 4 introduces the TT decomposition, the (block) ALS algorithms, and the 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.

2 | PROBLEM STATEMENT AND MATHEMATICAL DESCRIPTION

This article investigates the distributed optimal control of an incompressible flow problem constrained by time-dependent Navier-Stokes equations with uncertain inflow in an open bounded spatial domain $D \subset \mathbb{R}^2$ with a piecewise-Lipschitz boundary $\Upsilon = \partial D$ and an outward normal vector \vec{n} . We denote the spatial coordinates as $\mathbf{x} = (x_1, x_2) \in D$. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a complete probability space, where Ω is the set of outcomes, $\mathcal{F} \subseteq 2^\Omega$ is the σ -algebra of events, and $\mathbb{P} : \mathcal{F} \rightarrow [0, 1]$ is a uniform probability measure. Moreover, we consider a finite time interval $(0, T)$ and seek a control function $\mathbf{u}(\mathbf{x}, \omega, t) : D \times \Omega \times (0, T) \rightarrow \mathbb{R}^2$, a velocity field $\mathbf{v}(\mathbf{x}, \omega, t) : D \times \Omega \times (0, T) \rightarrow \mathbb{R}^2$, and a pressure $p(\mathbf{x}, \omega, t) : D \times \Omega \times (0, T) \rightarrow \mathbb{R}$ that solve the minimization problem

$$\min_{\mathbf{u}, \mathbf{v} \in L^2(D) \times L^2(\Omega) \times L^2(0, T)} \mathcal{J}(\mathbf{u}, \mathbf{v}) = \frac{1}{2} \|\nabla \times \mathbf{v}\|^2 + \frac{\beta}{2} \|\mathbf{u}\|^2, \tag{1}$$

with the norm defined as

$$\|\mathbf{v}\|^2 = \int_0^T \int \int_D \|\mathbf{v}(\mathbf{x}, \omega, t)\|_2^2 \, d\mathbf{x} d\mathbb{P}(\omega) \, dt,$$

and \mathbf{u}, \mathbf{v} , and p are constrained, \mathbb{P} -almost surely, through the nonstationary Navier-Stokes equations:

$$\begin{cases} \partial_t \mathbf{v} - \nu \Delta \mathbf{v} + (\mathbf{v} \cdot \nabla) \mathbf{v} + \nabla p = \mathbf{u}, & \text{in } D \times \Omega \times (0, T), \\ -\nabla \cdot \mathbf{v} = 0, & \text{in } D \times \Omega \times (0, T), \\ \mathbf{v} = \boldsymbol{\theta}, & \text{on } \Upsilon_{\text{in}} \times \Omega \times (0, T), \quad \text{inflow} \\ \mathbf{v} = 0, & \text{on } \Upsilon_w \times \Omega \times (0, T), \quad \text{wall} \\ \nu \nabla \mathbf{v} \cdot \vec{n} - p \vec{n} = 0, & \text{on } \Upsilon_{\text{out}} \times \Omega \times (0, T), \quad \text{outlet} \\ \mathbf{v}(\mathbf{x}, \omega, 0) = 0, & \text{in } D \times \Omega. \end{cases} \tag{2}$$

Here, we let Υ be decomposed into disjoint parts corresponding to inflow Υ_{in} , solid wall Υ_w , and outflow Υ_{out} . The time derivative is denoted by ∂_t . Naturally, we assume that $\|\mathbf{v}\| < \infty$, $\|\mathbf{u}\| < \infty$ and $\|p\| < \infty$. Each of the functions $\mathbf{v}, \mathbf{u}, p$ is a random field, since so is the inflow boundary condition $\boldsymbol{\theta}(\mathbf{x}, \omega, t) : \Upsilon_{\text{in}} \times \Omega \times (0, T) \rightarrow \mathbb{R}^2$, specified below. The regularization parameter β in (1) balances between minimization of the vorticity and penalization of the control magnitude. The cost $\mathcal{J}(\mathbf{u}, \mathbf{v})$ is a deterministic functional with stochastic arguments \mathbf{u}, \mathbf{v} , and the parameter ν is a deterministic kinematic viscosity.

Our domains of interest are the flow around a cylinder illustrated in Figure 1 and the backward facing step shown in Figure 2 (cf Reference 31). These are standard benchmarks and illustrate many of the difficulties that arise when solving Stokes and especially Navier-Stokes control problems.^{32,33} Furthermore, in our experiments, we use a zero initial condition and apply instead an exponentially growing stochastic inflow

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

(see Figures 1 and 2), where $\kappa(\mathbf{x}, \omega)$ is a random field on $\Upsilon_{\text{in}} \times \Omega$. The mean field of κ is set to the corresponding benchmark function associated with each of the domains in Figures 1 and 2. The (square integrable) remainder is assumed to have a convergent Fourier expansion, parametrized by independent identically distributed (i.i.d.) random variables $\{\xi_1, \xi_2, \dots\}$. For concreteness, we assume uniformly distributed $\xi_\ell \sim \mathcal{U}(-1, 1)$, although we can generalize this to any zero-mean i.i.d. probability measure. In particular, for the obstacle domain (Figure 1) we assume a parabolic mean inflow,

$$\kappa(\mathbf{x}, \omega) = (1 + x_2)(1 - x_2) + \underbrace{\sum_{\ell=1}^{\infty} \ell^{-\gamma-1/2} \sin(\pi \ell x_2)}_{=: \theta_\ell(\mathbf{x})} \xi_\ell(\omega). \tag{3}$$

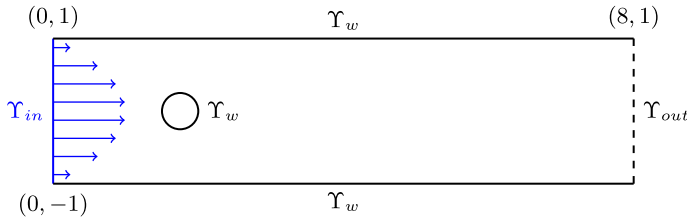


FIGURE 1 A flow with uncertain inflow past a circular obstacle [Colour figure can be viewed at wileyonlinelibrary.com]

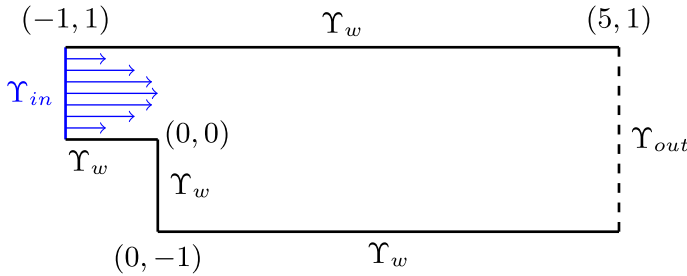


FIGURE 2 A flow with uncertain inflow in a backward step domain [Colour figure can be viewed at wileyonlinelibrary.com]

The parameter γ governs the rate of decay of the stochastic Fourier expansion (SFE), and hence it can be referred to as the smoothness of the random field. The offset $1/2$ is chosen to make the decay rate of the expansion (3) similar to that of the Karhunen-Löve expansion of typical one-dimensional covariance functions, for example, Matérn.¹⁷ In computational practice, we truncate (3) after $m \in \mathbb{N}$ terms such that the error is sufficiently small:

$$\kappa_m(\mathbf{x}, \omega) = (1 + x_2)(1 - x_2) + \sum_{\ell=1}^m \ell^{-\gamma-1/2} \sin(\pi \ell x_2) \xi_\ell(\omega). \tag{4}$$

In the backward step domain problem (Figure 2), we use a similar benchmark inflow field³³

$$\kappa_m(\mathbf{x}, \omega) = 4x_2(1 - x_2) + \underbrace{\sum_{\ell=1}^m \frac{1}{2} \ell^{-\gamma-1/2} \sin(2\pi \ell x_2) \xi_\ell(\omega)}_{=: \theta_\ell(\mathbf{x})}. \tag{5}$$

In both Equations (4) and (5), the random vector $\xi := (\xi_1, \xi_2, \dots, \xi_m) : \Omega \rightarrow \Gamma \subset \mathbb{R}^m$ is characterized by the joint probability density function $\rho(\xi) = \frac{1}{2^m}$ on the Cartesian joint image $\Gamma = [-1, 1]^m$. The expectation of any measurable function $f(\xi)$ is then given by the integral $\mathbb{E}[f] = \int_{\Gamma} f(\xi) \rho(\xi) d\xi < \infty$. The assumption that $\kappa(\mathbf{x}, \omega)$ admits a stochastic representation as above allows us to transform the stochastic optimal control problem into a deterministic problem that now depends on the parameter vector $\xi := (\xi_1, \xi_2, \dots, \xi_m) \in \mathbb{R}^m$. For brevity, since we consider m as an a priori model parameter, we always denote κ_m by κ in the rest of the article. Parametrization of (1) and (2) using (4), (5) leads to the following finite dimensional deterministic optimal control problem:

$$\min_{\mathbf{u}, \mathbf{v}} \mathcal{J}(\mathbf{u}, \mathbf{v}) = \frac{1}{2} \int_0^T \int_{\Gamma} \int_D \|\nabla \times \mathbf{v}\|_2^2 d\mathbf{x} \rho(\xi) d\xi dt + \frac{\beta}{2} \int_0^T \int_{\Gamma} \int_D \|\mathbf{u}\|_2^2 d\mathbf{x} \rho(\xi) d\xi dt, \tag{6}$$

where now $\mathbf{u}(\mathbf{x}, \xi, t), \mathbf{v}(\mathbf{x}, \xi, t) : D \times \Gamma \times (0, T) \rightarrow \mathbb{R}^2$, subject to

$$\begin{cases} \partial_t \mathbf{v} - \nu \Delta \mathbf{v} + (\mathbf{v} \cdot \nabla) \mathbf{v} + \nabla p = \mathbf{u}, & \text{in } D \times \Gamma \times (0, T), \\ -\nabla \cdot \mathbf{v} = 0, & \text{in } D \times \Gamma \times (0, T), \end{cases} \tag{7}$$

and the corresponding boundary and initial conditions from (2).

For the numerical simulation of the SOCP given by (6) and (7), we will adopt the so-called *optimize-then-discretize* (OTD) strategy, where 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 velocity λ and pressure μ .^{34,35} We observe 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.³¹ Since the Picard iteration has a larger radius of convergence compared with the Newton iteration, our choice in this contribution is the Picard iteration. We apply the so-called Karush-Kuhn-Tucker procedure^{31(chap8.2)} to (6) and (7) to obtain the following linear optimality system:³⁶

$$\begin{aligned} \partial_t \mathbf{v} - \nu \Delta \mathbf{v} + (\bar{\mathbf{v}} \cdot \nabla) \mathbf{v} + \nabla p &= \mathbf{u} & \mathbf{x} \in D, \\ \nabla \cdot \mathbf{v} &= 0 & \mathbf{x} \in D, \\ \mathbf{v} &= \boldsymbol{\theta} & \mathbf{x} \in Y_{\text{in}}, \\ \mathbf{v} &= 0 & \mathbf{x} \in Y_w, \\ \nu \nabla \mathbf{v} \cdot \bar{\mathbf{n}} - p \bar{\mathbf{n}} &= 0 & \mathbf{x} \in Y_{\text{out}}, \\ \mathbf{v}(\mathbf{x}, \boldsymbol{\xi}, 0) &= 0 & \mathbf{x} \in D, \end{aligned} \quad (8)$$

$$\begin{aligned} -\partial_t \lambda - \nu \Delta \lambda - (\bar{\mathbf{v}} \cdot \nabla) \lambda + (\nabla \bar{\mathbf{v}})^T \lambda + \nabla \mu &= -\nabla \times (\nabla \times \mathbf{v}) & \mathbf{x} \in D, \\ \nabla \cdot \lambda &= 0 & \mathbf{x} \in D, \\ \lambda &= 0 & \mathbf{x} \in Y_w \cup Y_{\text{in}}, \\ \nu \nabla \lambda \cdot \bar{\mathbf{n}} - \mu \bar{\mathbf{n}} &= 0 & \mathbf{x} \in Y_{\text{out}}, \\ \lambda(\mathbf{x}, \boldsymbol{\xi}, T) &= 0 & \mathbf{x} \in D, \end{aligned} \quad (9)$$

$$\beta \mathbf{u} + \lambda = 0 \quad \mathbf{x} \in D, \quad (10)$$

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 | STOCHASTIC GALERKIN DISCRETIZATION OF THE SOLUTION

A finite dimensional system is obtained by making (8)-(10) Galerkin-orthogonal to a finite polynomial subspace in the parameters $\boldsymbol{\xi}$, the so-called *generalized polynomial chaos*,³⁷ as well as a finite element basis in \mathbf{x} . In many variables, the polynomial chaos can be introduced in different ways. The most straightforward approach is to use a Cartesian product of univariate subspaces. Choosing a maximal individual degree d , we construct a space

$$\mathcal{Y}_d := \text{span} \left(\psi_j(\boldsymbol{\xi}) := \prod_{\ell=1}^m \psi_{j_\ell}^{j_\ell}(\xi_{\ell}) : j_\ell = 0, \dots, d-1, \ell = 1, \dots, m \right), \quad (11)$$

where $\psi_{j_\ell}^{j_\ell}$ is a univariate polynomial of degree j_ℓ , and j is the total index of the multivariate polynomial, constructed from individual indices of univariate polynomials,

$$j = j_1 d^{m-1} + j_2 d^{m-2} + \dots + j_{m-1} d + j_m. \quad (12)$$

Note that $j = 0, \dots, d^m - 1$, where d^m is indeed the cardinality of the Cartesian space. We assume that the polynomials $\psi_j(\boldsymbol{\xi})$ are orthogonal with respect to the given probability measure, that is, $\mathbb{E}[\psi_j \psi_{j'}] = 0$ for any $j_\ell \neq j'_\ell$, comprising j and j' , respectively. In particular, we use Legendre polynomials that are orthogonal with respect to the uniform measure. Similarly, Jacobi polynomials can be used for beta-distributed random variables, or Hermite polynomials can be used for the Gaussian measure, as introduced originally by Norbert Wiener³⁸ in the context of turbulence modeling. The convergence with respect to the degree d can be established by the standard analysis of the univariate polynomial approximation.³⁹ Note that this requires a sufficient smoothness of the solution. It holds specifically for a flow with minimized vorticity as intended by (1), but the convergence rate may deteriorate for a turbulent solution.

In this article, we cope with the exponential growth of the cardinality d^m using low-rank tensor decompositions (see Section 4) for approximating directly the *discrete* expansion coefficients of the solution in a given full Cartesian basis. This allows us to get rid of the curse of dimensionality, since we never store the full tensor of coefficients explicitly, but only its low-rank factors. Another choice, which partially alleviates the curse of dimensionality, is the total degree polynomial space. The cardinality of the total degree space is $(m+d)/(m!d!)$, which, for a fixed d , depends polynomially on m . However, this cardinality might still be rather large, which, together with a complicated sparsity pattern of the Galerkin matrix, makes the numerical solution expensive. By contrast, iterative tensor product methods (see Algorithm 1) require subsequent solution of small reduced systems of equations on each tensor factor, which can be more efficient than assembling and solving a single but large Galerkin system for the total degree space.⁴⁰

For the spatial discretization, we use the stable P_2 - P_1 Taylor-Hood finite element ansatz,³³ consisting of biquadratic elements $\{\phi_k(\mathbf{x})\}_{k=1}^{N_v}$ for the velocity, and bilinear elements $\{\varphi_k(\mathbf{x})\}_{k=1}^{N_p}$ for the pressure. Specifically, we approximate the velocity functions $\mathbf{v}, \mathbf{u}, \lambda$ in a subspace

$$V_h = \text{span} \left\{ \begin{bmatrix} \phi_k(\mathbf{x}) \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \phi_k(\mathbf{x}) \end{bmatrix} \right\}_{k=1}^{N_v} \subset H_0^1(D)^2,$$

where $H_0^1(D)$ is the Sobolev space of functions with derivatives in $L^2(D)$, satisfying the homogeneous Dirichlet boundary conditions on Υ_w . The pressure functions p, μ are approximated in a subspace

$$Q_h = \text{span} \{\varphi_k(\mathbf{x})\}_{k=1}^{N_p} \subset L^2(D).$$

Finally, the time derivative is discretized using the implicit Euler scheme on a uniform time grid $\{t_n\}_{n=1}^{N_t}$, $t_n = \tau n$, $\tau = T/N_t$. The solutions can then be approximated by the following expansions:

$$\begin{aligned} \mathbf{v}(\mathbf{x}, \xi, t_n) &:= \begin{bmatrix} v^1(\mathbf{x}, \xi, t_n) \\ v^2(\mathbf{x}, \xi, t_n) \end{bmatrix} \approx \mathbf{v}_h(\mathbf{x}, \xi, t_n) := \sum_{j=0}^{d^m-1} \sum_{k=1}^{N_v} \begin{bmatrix} \hat{v}^1(k, j, n) \\ \hat{v}^2(k, j, n) \end{bmatrix} \phi_k(\mathbf{x}) \psi_j(\xi), \\ p(\mathbf{x}, \xi, t_n) &\approx p_h(\mathbf{x}, \xi, t_n) := \sum_{j=0}^{d^m-1} \sum_{k=1}^{N_p} \hat{p}(k, j, n) \varphi_k(\mathbf{x}) \psi_j(\xi), \end{aligned} \quad (13)$$

and similarly for $\mathbf{u} \approx \mathbf{u}_h$, $\lambda \approx \lambda_h$ and $\mu \approx \mu_h$, where \hat{v} and \hat{p} are sought expansion coefficients. The weak variational formulation is obtained by projecting (8)-(10) onto the same spaces V_h, Q_h , combined with the polynomial subspace \mathcal{Y}_d defined in (11). This gives the following optimality equations:

$$\begin{aligned} \mathbb{E} \int_D \left[\frac{\mathbf{v}_h - \mathbf{v}_h(\mathbf{x}, \xi, t_{n-1})}{\tau} \cdot \mathbf{v}' + \nu \nabla \mathbf{v}_h : \mathbf{v}' + (\bar{\mathbf{v}}_h \cdot \nabla) \mathbf{v}_h \cdot \mathbf{v}' \right. \\ \left. - p_h(\nabla \cdot \mathbf{v}') - \mathbf{u}_h \cdot \mathbf{v}' \right] d\mathbf{x} \psi'(\xi) = 0 \quad \forall \mathbf{v}' \in V_h, \end{aligned} \quad (14)$$

$$\mathbb{E} \int_D [(\nabla \cdot \mathbf{v}_h) p'(\mathbf{x})] d\mathbf{x} \psi'(\xi) = 0 \quad \forall p' \in Q_h,$$

$$\begin{aligned} \mathbb{E} \int_D \left[\frac{\lambda_h - \lambda_h(\mathbf{x}, \xi, t_{n+1})}{\tau} \cdot \mathbf{v}' + \nu \nabla \lambda_h : \mathbf{v}' - (\bar{\mathbf{v}}_h \cdot \nabla) \lambda_h \cdot \mathbf{v}' \right. \\ \left. + [(\nabla \bar{\mathbf{v}}_h) \lambda_h] \cdot \mathbf{v}' - \mu_h(\nabla \cdot \mathbf{v}') - \nabla \mathbf{v}_h \cdot \nabla \mathbf{v}' \right] d\mathbf{x} \psi'(\xi) = 0 \quad \forall \mathbf{v}' \in V_h, \end{aligned} \quad (15)$$

$$\mathbb{E} \int_D [(\nabla \cdot \lambda_h) p'(\mathbf{x})] d\mathbf{x} \psi'(\xi) = 0 \quad \forall p' \in Q_h,$$

$$\mathbb{E} \int_D [\beta \mathbf{u}_h \cdot \mathbf{v}' + \lambda_h \cdot \mathbf{v}'] d\mathbf{x} \psi'(\xi) = 0 \quad \forall \mathbf{v}' \in V_h, \quad (16)$$

for all $\psi'(\xi) \in \mathcal{Y}_d$, where we let

$$\begin{aligned} \mathbf{v}_h &= \mathbf{v}_h(\mathbf{x}, \xi, t_n), \bar{\mathbf{v}}_h = \bar{\mathbf{v}}_h(\mathbf{x}, \xi, t_n), \mathbf{u}_h = \mathbf{u}_h(\mathbf{x}, \xi, t_n), \\ \lambda_h &= \lambda_h(\mathbf{x}, \xi, t_n), p_h = p_h(\mathbf{x}, \xi, t_n), \mu_h = \mu_h(\mathbf{x}, \xi, t_n) \end{aligned}$$

unless stated otherwise.

To write (14)-(16) in a matrix form, we collect the expansion coefficients from (13) into *component* coefficient vectors

$$\begin{aligned} \hat{\mathbf{v}}(j, n) &= [\hat{\mathbf{v}}^1(1, j, n) \dots \hat{\mathbf{v}}^1(N_v, j, n), \hat{\mathbf{v}}^2(1, j, n) \dots \hat{\mathbf{v}}^2(N_v, j, n)]^\top, \\ \hat{p}(j, n) &= [\hat{p}(1, j, n) \dots \hat{p}(N_p, j, n)]^\top, \\ \hat{\mathbf{u}}(j, n) &= [\hat{\mathbf{u}}^1(1, j, n) \dots \hat{\mathbf{u}}^1(N_v, j, n), \hat{\mathbf{u}}^2(1, j, n) \dots \hat{\mathbf{u}}^2(N_v, j, n)]^\top, \\ \hat{\lambda}(j, n) &= [\hat{\lambda}^1(1, j, n) \dots \hat{\lambda}^1(N_v, j, n), \hat{\lambda}^2(1, j, n) \dots \hat{\lambda}^2(N_v, j, n)]^\top, \\ \hat{\mu}(j, n) &= [\hat{\mu}(1, j, n) \dots \hat{\mu}(N_p, j, n)]^\top, \end{aligned}$$

as well as *total* vectors

$$\begin{aligned} \hat{\mathbf{w}} &= [(\hat{\mathbf{v}}(j, n)^\top, \hat{p}(j, n)^\top)^\top], \quad j = 0, \dots, d^m - 1, \quad n = 1, \dots, N_t, \\ \hat{\mathbf{u}} &= [\hat{\mathbf{u}}(j, n)], \quad j = 0, \dots, d^m - 1, \quad n = 1, \dots, N_t, \end{aligned} \tag{17}$$

$$\hat{\boldsymbol{\eta}} = [(\hat{\lambda}(j, n)^\top, \hat{\mu}(j, n)^\top)^\top], \quad j = 0, \dots, d^m - 1, \quad n = 1, \dots, N_t.$$

Moreover, we need to introduce the following matrices resulting from the weak bilinear forms with the basis functions.

- The Laplace matrix for V_h : $[L]_{k,k'} = \int_D \nabla \phi_k \cdot \nabla \phi_{k'} \, d\mathbf{x}, k, k' = 1, \dots, N_v$.
- The mass matrix for V_h : $[M]_{k,k'} = \int_D \phi_k \phi_{k'} \, d\mathbf{x}, k, k' = 1, \dots, N_v$.
- The partial discrete divergence operator: $[B^z]_{k,k'} = \int_D \phi_k \frac{\partial \phi_{k'}}{\partial x_z} \, d\mathbf{x}, z = 1, 2, k = 1, \dots, N_p, k' = 1, \dots, N_v$.
- The convection matrix depending on the given velocity component vector $\hat{\mathbf{v}}$:

$$[F_{\hat{\mathbf{v}}}]_{k,k'} = \int_D \phi_k \left[\sum_{k''=1}^{N_v} \begin{bmatrix} \hat{\mathbf{v}}^1(k'') \\ \hat{\mathbf{v}}^2(k'') \end{bmatrix} \phi_{k''}(\mathbf{x}) \right] \cdot \nabla \phi_{k'} \, d\mathbf{x}, \quad k, k' = 1, \dots, N_v. \tag{18}$$

- The weak derivative (adjoint) matrix depending on the given velocity:

$$[W_{\hat{\mathbf{v}}}^{z,z'}]_{k,k'} = \int_D \phi_k \left[\sum_{k''=1}^{N_v} \hat{\mathbf{v}}^z(k'') \frac{\partial \phi_{k''}}{\partial x_{z'}} \right] \phi_{k'} \, d\mathbf{x}, \quad k, k' = 1, \dots, N_v, \quad z, z' = 1, 2. \tag{19}$$

Now the system (14)-(16) can be expressed in the following compact matrix form on the total vectors (17) of solution coefficients:

$$\underbrace{\begin{bmatrix} \mathcal{M}_1 & 0 & -\mathcal{K}^* \\ 0 & \beta \mathcal{M}_2 & \mathcal{M}_3^\top \\ -\mathcal{K} & \mathcal{M}_3 & 0 \end{bmatrix}}_{=: \mathbf{A}} \underbrace{\begin{bmatrix} \hat{\mathbf{w}} \\ \hat{\mathbf{u}} \\ \hat{\boldsymbol{\eta}} \end{bmatrix}}_{=: \mathbf{y}} = \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \hat{\mathbf{g}} \end{bmatrix}}_{=: \mathbf{b}}. \tag{20}$$

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

follows:

$$\begin{aligned} \mathcal{K} &= \mathcal{L} \otimes I_{d^m} \otimes I_{N_t} + \mathcal{M} \otimes I_{d^m} \otimes C\tau^{-1} + \mathcal{F}_{\hat{\mathbf{v}}}, \\ \mathcal{K}^* &= \mathcal{L} \otimes I_{d^m} \otimes I_{N_t} + \mathcal{M} \otimes I_{d^m} \otimes C^\top \tau^{-1} + \mathcal{W}_{\hat{\mathbf{v}}} - \mathcal{F}_{\hat{\mathbf{v}}}, \end{aligned} \tag{21}$$

and the mixing terms read

$$\begin{aligned} \mathcal{M}_1 &= \begin{bmatrix} L & 0 & 0 \\ 0 & L & 0 \\ 0 & 0 & 0 \end{bmatrix} \otimes I_{d^m} \otimes I_{N_t}, & \mathcal{M}_3 &= \begin{bmatrix} M & 0 \\ 0 & M \\ 0 & 0 \end{bmatrix} \otimes I_{d^m} \otimes I_{N_t}, \\ & & \mathcal{M}_2 &= \begin{bmatrix} M & 0 \\ 0 & M \end{bmatrix} \otimes I_{d^m} \otimes I_{N_t}, \end{aligned} \tag{22}$$

where² I_{N_t} and I_{d^m} are identity matrices of the corresponding sizes,

$$C = \begin{bmatrix} 1 & & & & \\ -1 & 1 & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & -1 & 1 \end{bmatrix}$$

comes from the implicit Euler time scheme,

$$\mathcal{L} = \begin{bmatrix} \nu L & 0 & (B^1)^\top \\ 0 & \nu L & (B^2)^\top \\ B^1 & B^2 & 0 \end{bmatrix} \quad \text{and} \quad \mathcal{M} = \begin{bmatrix} M & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

are the full spatial Stokes and mass matrices, respectively, and $\mathcal{F}_{\hat{\mathbf{v}}}, \mathcal{W}_{\hat{\mathbf{v}}}$ are matrices derived from the nonlinear terms, evaluated at the coefficients $\hat{\mathbf{v}}$ of the previous Picard iterate $\bar{\mathbf{v}}$. These matrices can be also constructed in a Kronecker product form using (18) and (19), provided a compatible decomposition of $\hat{\mathbf{v}}$ is available. This will be introduced in the next section.

The right-hand side vector $\hat{\mathbf{g}}$ depends on the boundary conditions. The stochastic expansion of $\kappa(\mathbf{x}, \xi)$ yields the following tensor form for the inflow function,

$$\hat{\theta} = \begin{bmatrix} \hat{\theta}_\ell^1 \\ 0 \\ 0 \end{bmatrix} \otimes \sum_{\ell=0}^m \left[\left(\bigotimes_{\ell'=1}^{\ell-1} \mathbf{e}_1 \right) \otimes \mathbf{e}_2 \otimes \left(\bigotimes_{\ell'=\ell+1}^m \mathbf{e}_1 \right) \right] \otimes \mathbf{g}^t, \tag{23}$$

where the vector $\hat{\theta}_\ell^1$ is the discretized spatial function $\theta_\ell(\mathbf{x})$ of the inflow (4) or (5), $[\hat{\theta}_\ell^1]_k = \theta_\ell(x_1^k, x_2^k)$ for $\mathbf{x}^k \in \Upsilon_{\text{in}}$, and 0 otherwise,

$$\mathbf{e}_1 = [1 \ 0 \ \dots \ 0]^\top \in \mathbb{R}^d \quad \text{and} \quad \mathbf{e}_2 = [0 \ 1 \ 0 \ \dots \ 0]^\top \in \mathbb{R}^d$$

are the unit vectors resulting from expanding a constant and ξ_ℓ as given by (4) in \mathcal{Y}_d , and the vector \mathbf{g}^t is the discretized time profile of the inflow, $[\mathbf{g}^t]_n = 1 - \exp(-\tau n)$. Now we partition the PDE matrix into blocks, corresponding to inner and boundary degrees of freedom,

$$\mathcal{K} = \begin{bmatrix} \mathcal{K}_{\text{II}} & \mathcal{K}_{\text{IB}} \\ \mathcal{K}_{\text{BI}} & \mathcal{K}_{\text{BB}} \end{bmatrix},$$

or, more precisely, (B) denotes the nodes associated with the centers \mathbf{x}^k of biquadratic elements from V_h belonging to the inflow boundary, $\mathbf{x}^k \in \Upsilon_{\text{in}}$, and (I) denotes all interior and outflow nodes. Then the usual finite element approach is employed: we eliminate \mathcal{K}_{IB} and \mathcal{K}_{BI} in the left-hand side of (20), replacing \mathcal{K} by \mathcal{K}_{II} , and construct the right-hand side from (23) as $\hat{\mathbf{g}} = \mathcal{K}_{\text{IB}} \hat{\theta}$.

4 | TENSOR TRAIN DECOMPOSITION

Separation of the total index enumerating all degrees of freedom of, for example, the discrete solution in (20) back into individual indices corresponding to independent variables can be extended from (12) to the space and time variables as well. Each of the total vectors (17) $\hat{\mathbf{y}} = [\hat{\mathbf{y}}(i)]$, where \mathbf{y} stands for \mathbf{w} , \mathbf{u} , or $\boldsymbol{\eta}$, can also be seen as a $(m + 2)$ -dimensional *tensor* with elements $\hat{\mathbf{y}}(k, j_1, \dots, j_d, n)$, provided that the total index i is defined consistently,

$$i = (k - 1)d^m N_t + j_1 d^{m-1} N_t + \dots + j_m n_t + n.$$

The particular *order* of indices is not important for the concept in general. However, the actual numerical complexity of the low-rank decompositions introduced further might depend on the order, and certain approaches exist for determining it adaptively.^{41,42} For our problem, the particular ordering chosen above is good enough due to the decay of the stochastic expansion of $\kappa(\mathbf{x}, \boldsymbol{\xi})$, and the corresponding decrease of the “influence” of ξ_ℓ for larger ℓ , as well as the largest dimensions corresponding to \mathbf{x} and t being enumerated by the first and the last indices. To alleviate the curse of dimensionality, we represent $\hat{\mathbf{y}}$ by the so-called tensor train (TT) decomposition.⁴³

$$\hat{\mathbf{y}}(k, j_1, \dots, j_m, n) = \sum_{s_0=1}^{r_0} \dots \sum_{s_m=1}^{r_m} y_{s_0}^{(0)}(k) y_{s_0, s_1}^{(1)}(j_1) \dots y_{s_{m-1}, s_m}^{(m)}(j_m) y_{s_m}^{(m+1)}(n). \tag{24}$$

In practice, the exact equality (24) holds very rarely, and one seeks an *approximation* of $\hat{\mathbf{y}}$ by the right-hand side of (24). The auxiliary summation indices s_0, \dots, s_m are called rank indices, and their ranges r_0, \dots, r_m are called *TT ranks*. The factors on the right-hand side of (24) are called *TT blocks*. The trailing TT blocks are matrices, that is, $y^{(m+1)} \in \mathbb{R}^{r_m \times n_t}$, and $y^{(0)}$ can be a matrix of size $(2N_v + N_p) \times r_0$ or $2N_v \times r_0$, depending on what vector of (17) is approximated. The middle TT blocks can be seen in three equivalent forms:

- $y^{(\ell)} \in \mathbb{R}^{r_{\ell-1} \times d \times r_\ell}$ is a three-dimensional *tensor*. Introducing the maximal TT rank $r := \max_{\ell=0, \dots, m} r_\ell$, we can conclude that the TT blocks contain at most $m d r^2 + (2N_v + N_p + N_t) r$ elements, that is, a *linear* amount w.r.t. the dimensionality m . The TT ranks depend on the particular tensor and approximation accuracy. In numerical practice, we aim for applications that allow a sufficiently accurate TT approximation with r being *much* smaller than the $\mathcal{O}(d^m)$ cardinality of the original tensor.
- For a fixed $j_\ell, y^{(\ell)}(j_\ell) \in \mathbb{R}^{r_{\ell-1} \times r_\ell}$ is a *matrix*. In fact, we can omit s_ℓ in (24) and consider the right-hand side as a product of matrices, parametrized by k, j_ℓ and n . For this reason, the TT decomposition is also known as the matrix product states^{44,45} in quantum physics.
- For fixed $s_{\ell-1}, s_\ell, y_{s_{\ell-1}, s_\ell}^{(\ell)} \in \mathbb{R}^d$ is a *vector*. Similarly, we can say that $y_{s_0}^{(0)}$ and $y_{s_m}^{(m+1)}$ are vectors. This allows us to rewrite (24) using Kronecker products:

$$\hat{\mathbf{y}} = \sum_{s_0, \dots, s_m=1}^{r_0, \dots, r_m} y_{s_0}^{(0)} \otimes y_{s_0, s_1}^{(1)} \otimes \dots \otimes y_{s_m}^{(m+1)}. \tag{25}$$

Note that the left-hand side is now a *vector* too, compatible with (17).

The latter notation (25) can be easily generalized to break the curse of dimensionality of *matrices* acting on $\hat{\mathbf{y}}$, such as those in (20). Namely, we introduce a matrix TT decomposition

$$\mathbf{A} = \sum_{s_0, \dots, s_m=1}^{R_0, \dots, R_m} A_{s_0}^{(0)} \otimes A_{s_0, s_1}^{(1)} \otimes \dots \otimes A_{s_m}^{(m+1)} \tag{26}$$

with some *matrix TT ranks* R_0, \dots, R_m . For example, each of the mixing matrices (22) is a particular case of the matrix TT decomposition with $R_0 = \dots = R_m = 1$ and $A^{(m+1)} = I_{N_t}$, $A^{(\ell)} = I_d$ for $\ell = 1, \dots, m$, and $A^{(0)}$ being one of the leftmost block matrices in (22). However, (26) can be extended also to (21) with larger R_0, \dots, R_m , as we show in Section 4.1.

Equipped with (25) and (26), we can implement the so-called *tensor arithmetics*⁴³ efficiently. Linear combinations of vectors in the form (25), scalar, Hadamard, or matrix-vector products $\mathbf{A}\hat{\mathbf{y}}$, can be constructed in the TT format exactly, block by block, without ever expanding the Kronecker products explicitly. However, TT ranks of such exact representation can be still unnecessarily large for the desired accuracy, for example, TT ranks of $\mathbf{A}\hat{\mathbf{y}}$ are equal to the products $R_\ell r_\ell$. In this case, the TT decomposition can be *recompressed* down to quasi-optimal TT ranks for a given threshold $\epsilon > 0$ using the truncated singular value decompositions (SVD) applied to the matrix forms of TT blocks, with the total complexity being linear in m and polynomial in r and $R := \max_\ell R_\ell$. We refer to References 43 and 44 for details on these procedures.

4.1 | TT format of the convection matrix

Since we can represent any tensor in a TT decomposition with sufficiently large TT ranks, we assume now that the expansion coefficients (13) of the velocity $\bar{\mathbf{v}}_h$ of the previous Picard iterate are written similarly to (25) as follows:

$$\hat{\mathbf{v}} = \sum_{s_0, \dots, s_m}^{r_0, \dots, r_m} v_{s_0}^{(0)} \otimes v_{s_0, s_1}^{(1)} \otimes \dots \otimes v_{s_{m-1}, s_m}^{(m)} \otimes v_{s_m}^{(m+1)}. \tag{27}$$

Note that $v_{s_0}^{(0)} \in \mathbb{R}^{2N_v}$ contains coefficients corresponding to the V_h space.

Considering only the nonlinear term $(\bar{\mathbf{v}}_h \cdot \nabla) \mathbf{v}'$ from (14), we can write the matrix element of $\mathcal{F}_{\hat{\mathbf{v}}}$, replacing \mathbf{v}_h by $\phi_k(\mathbf{x})\psi_j(\xi)$ and \mathbf{v}' by $\phi_{k'}(\mathbf{x})\psi_{j'}(\xi)$, as follows:

$$[\mathbf{F}_{\hat{\mathbf{v}}}^n]_{kj, k'j'} = \mathbb{E} \int_D \phi_k \psi_j \left[\sum_{k'', j''} \left[\hat{\mathbf{v}}^1(k'', j'', n) \phi_{k''} \psi_{j''} \right] \cdot \nabla \phi_{k'} dx \psi_{j'}, \right.$$

where n is the current time step. Due to linearity of the TT format (27) w.r.t. $v^{(0)}$, we can replace

$$\begin{bmatrix} \hat{\mathbf{v}}^1(k'', j'', n) \\ \hat{\mathbf{v}}^2(k'', j'', n) \end{bmatrix} = \sum_{s_0, \dots, s_m}^{r_0, \dots, r_m} \begin{bmatrix} v_{s_0}^{(0)1}(k'') \\ v_{s_0}^{(0)2}(k'') \end{bmatrix} v_{s_0, s_1}^{(1)}(j''_1) \dots v_{s_m}^{(m+1)}(n),$$

and, using (18), we obtain

$$[\mathbf{F}_{\hat{\mathbf{v}}}^n]_{jj'} = \sum_{s_0, \dots, s_m}^{r_0, \dots, r_m} F_{v_{s_0}^{(0)}} \mathbb{E} \left[\sum_{j''=0}^{d^m-1} \psi_j \cdot \psi_{j''} v_{s_0, s_1}^{(1)}(j''_1) \dots v_{s_{m-1}, s_m}^{(m)}(j''_m) \cdot \psi_{j'} \right] v_{s_m}^{(m+1)}(n).$$

Due to the Cartesian structure of (11), we can introduce a matrix of trilinear form

$$H_{j''_\ell}(j_\ell, j'_\ell) = \int_{-1}^1 \psi_\ell^{j_\ell}(\xi_\ell) \psi_\ell^{j'_\ell}(\xi_\ell) \psi_\ell^{j''_\ell}(\xi_\ell) \frac{1}{2} d\xi_\ell, \quad H_{j''_\ell} \in \mathbb{R}^{d \times d}, \tag{28}$$

such that

$$\mathbf{F}_{\hat{\mathbf{v}}}^n = \sum_{s_0, \dots, s_m}^{r_0, \dots, r_m} F_{v_{s_0}^{(0)}} \otimes \left[\sum_{j''_1=0}^{d-1} H_{j''_1} v_{s_0, s_1}^{(1)}(j''_1) \right] \otimes \dots \otimes \left[\sum_{j''_m=0}^{d-1} H_{j''_m} v_{s_{m-1}, s_m}^{(m)}(j''_m) \right] \cdot v_{s_m}^{(m+1)}(n).$$

Finally, we notice that $\mathbf{F}_{\hat{\mathbf{v}}}^n$ applies to each individual time step n , and hence the entire matrix is block diagonal w.r.t. the time dimension,

$$\mathcal{F}_{\hat{\mathbf{v}}} = \sum_{s_0, \dots, s_m=1}^{r_1, \dots, r_m} \begin{bmatrix} F_{v_{s_0}^{(0)}} & 0 & 0 \\ 0 & F_{v_{s_0}^{(0)}} & 0 \\ 0 & 0 & 0 \end{bmatrix} \otimes \dots \otimes \left[\sum_{j''_\ell=0}^{d-1} H_{j''_\ell} v_{s_{\ell-1}, s_\ell}^{(\ell)}(j''_\ell) \right] \dots \otimes \text{diag}(v_{s_m}^{(m+1)}). \tag{29}$$

Note that this is exactly the matrix TT decomposition (26) with TT ranks $R_0 = r_0, \dots, R_m = r_m$. Adding the two linear terms in (21), we can conclude that if the velocity coefficients are represented in the TT format (27) with the maximal TT rank r , the forward Navier-Stokes matrix \mathcal{K} can be represented in the matrix TT format with the maximal TT rank $R = r + 2$.

The right-hand side $\hat{\mathbf{g}} = \mathcal{K}_{\text{IB}} \hat{\boldsymbol{\theta}}$ in (20) must be recomputed in every Picard iteration, since \mathcal{K}_{IB} carries the corresponding (new) part \mathcal{F}_{IB} . This can be achieved using the fast TT-structured matrix-vector product.⁴³

The adjoint convection matrix can be constructed similarly, using (19):

$$\mathcal{W}_{\hat{\mathbf{v}}} = \sum_{s_0, \dots, s_m=1}^{r_1, \dots, r_m} \begin{bmatrix} W_{v_{s_0}^{(0)}}^{1,1} & W_{v_{s_0}^{(0)}}^{1,2} & 0 \\ W_{v_{s_0}^{(0)}}^{2,1} & W_{v_{s_0}^{(0)}}^{2,2} & 0 \\ 0 & 0 & 0 \end{bmatrix} \otimes \dots \otimes \left[\sum_{j_\ell''=0}^{d-1} H_{j_\ell'' v_{s_{\ell-1}, s_\ell}^{(\ell)}}(j_\ell'') \right] \dots \otimes \text{diag}(v_{s_m}^{(m+1)}). \quad (30)$$

4.2 | Alternating linear solver

The discretized Oseen equation given by (20) is a large linear system in \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 to use alternating tensor product algorithms.^{44,46,47} Given the linear system $\mathbf{A}\mathbf{y} = \mathbf{b}$, we iterate over $\ell = 0, \dots, m + 1$ and seek only the elements of $\mathbf{y}^{(\ell)}$ in each step, while the other TT blocks are fixed. Again, since the TT format (24) is linear w.r.t. the elements of each TT block, there exists a matrix Y_ℓ (37) such that $\mathbf{y} = Y_\ell \cdot \text{vec}(\mathbf{y}^{(\ell)})$, where $\text{vec}(\mathbf{y}^{(\ell)})$ is a vector of all elements of the TT block $\mathbf{y}^{(\ell)}$. This renders $\mathbf{A}\mathbf{y} = \mathbf{b}$ an overdetermined system $\mathbf{A}Y_\ell \cdot \text{vec}(\mathbf{y}^{(\ell)}) = \mathbf{b}$ w.r.t. the elements of $\mathbf{y}^{(\ell)}$. This system is resolved via a projection onto Y_ℓ , such that $\mathbf{y}^{(\ell)}$ is computed from a smaller system $(Y_\ell^\top \mathbf{A} Y_\ell) \text{vec}(\mathbf{y}^{(\ell)}) = Y_\ell^\top \mathbf{b}$. Due to the linearity of the TT format, this method was called the alternating linear scheme (ALS).⁴⁷ However, the simple ALS algorithm can converge very slowly, and the TT ranks of the representation are fixed and cannot be adapted if it turns out that they were underestimated. More efficient extensions reduce the problem to computing two TT blocks $\mathbf{y}^{(\ell)}, \mathbf{y}^{(\ell+1)}$ simultaneously (the density matrix renormalization group [DMRG]^{46,48}) or augment the projection matrices Y_ℓ with a partial TT approximation of the residual (the alternating minimal energy (AMEn) algorithm⁴⁹).

However, the standard ALS method is not immediately applicable to our matrix \mathbf{A} from (20). First, the matrix \mathbf{A} is indefinite, and hence its Galerkin projection can be degenerate.

Example 1. Consider

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

One can readily verify that $Y_\ell^\top \mathbf{A} Y_\ell = 0$.

Instead of projecting the entire matrix \mathbf{A} , we employ the so-called block TT format⁵⁰ and project each *submatrix* of \mathbf{A} separately, which will be explained below.

Second, the velocity and pressure vectors have different sizes. Fortunately, this issue can be circumvented by decoupling of the momentum and kinematic equations, which is a well-established technique in computational fluid dynamics.⁵¹ This stems from the Lagrange multiplier role of the pressure in the divergence-free constraint for the velocity. Moreover, the empirical Galerkin projection of the pressure gradient (which is implicitly realized in the ALS method) is often exactly zero for closed flows.^{52,53} In general, this is not the case, and the pressure correction should be chosen judiciously⁵² in the standard model reduction framework with a precomputed basis. However, the ALS algorithm, carrying out several iterations, can be seen as an “iterative” model reduction, where the basis $\mathbf{v}^{(0)}$ is refined in the course of computations. This procedure can be also combined with the outer Picard iteration. This allows us to design a simpler Chorin-type approach³⁶ that is more suitable for the TT approximations.

First, fixing the velocity coefficients, the first row of (20) becomes an overdetermined equation on the pressure coefficients,

$$-\mathbf{B}^\top \hat{\mathbf{p}} = \mathbf{K} \hat{\mathbf{v}} + \hat{\mathbf{g}}_v - \mathbf{M}_3 \hat{\mathbf{u}},$$

where

$$\begin{aligned} \mathbf{B} &= [B^1 \ B^2] \otimes I_{d^m} \otimes I_{N_t}, \\ \mathbf{K} &= \begin{bmatrix} \nu L & 0 \\ 0 & \nu L \end{bmatrix} \otimes I_{d^m} \otimes I_{N_t} + \begin{bmatrix} M & 0 \\ 0 & M \end{bmatrix} \otimes I_{d^m} \otimes C\tau^{-1} + \mathbf{F}_{\hat{\mathbf{v}}}, \end{aligned} \tag{31}$$

$$\begin{aligned} \mathbf{F}_{\hat{\mathbf{v}}} &= \sum_{s_0, \dots, s_m=1}^{r_1, \dots, r_m} \begin{bmatrix} F_{\nu s_0}^{(0)} & 0 \\ 0 & F_{\nu s_0}^{(0)} \end{bmatrix} \otimes \dots \otimes \begin{bmatrix} \sum_{j_\ell''=0}^{d-1} H_{j_\ell''} \nu_{s_{\ell-1}, s_\ell}^{(\ell)}(j_\ell'') \end{bmatrix} \dots \otimes \text{diag}(\nu_{s_m}^{(m+1)}), \\ \mathbf{M}_3 &= \begin{bmatrix} M & 0 \\ 0 & M \end{bmatrix} \otimes I_{d^m} \otimes I_{N_t}, \end{aligned}$$

and $\hat{\mathbf{g}}_v$ is a subvector of $\hat{\mathbf{g}}$ corresponding to the velocity elements only. Solving this system in the least squares sense gives the pressure Poisson equation,⁵¹

$$\hat{p} = (\mathbf{B}\mathbf{B}^\top)^{-1} \mathbf{B} (-\mathbf{K}\hat{\mathbf{v}} - \hat{\mathbf{g}}_v + \mathbf{M}_3 \hat{\mathbf{u}}), \tag{32}$$

and similarly

$$\hat{\mu} = (\mathbf{B}\mathbf{B}^\top)^{-1} \mathbf{B} (-\mathbf{K}^* \hat{\lambda} + \mathbf{M}_1 \hat{\mathbf{v}}), \tag{33}$$

where \mathbf{K}^* and \mathbf{M}_1 are the “velocity” submatrices of \mathcal{K}^* and \mathcal{M}_1 , respectively, constructed similarly to (31). Now we cast these pressure components to the right-hand sides of the corresponding velocity equations and obtain a KKT system with blocks of the same size,

$$\begin{bmatrix} \mathbf{M}_1 & 0 & -\mathbf{K}^* \\ 0 & \mathbf{M}_2 & \mathbf{M}_3^\top \\ -\mathbf{K} & \mathbf{M}_3 & 0 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{v}} \\ \hat{\mathbf{u}} \\ \hat{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{B}^\top \hat{\mu} \\ 0 \\ \hat{\mathbf{g}}_v + \mathbf{B}^\top \hat{p} \end{bmatrix}. \tag{34}$$

The consequence is 2-fold: all solution components are of the same size $2N_v \cdot d^m N_t$ now, and hence are suitable for the block storage introduced below, and moreover, the matrices \mathbf{K}, \mathbf{K}^* , coming from the discretization of diffusion-convection equations, are positive definite.

Let us now focus on the block AMEn algorithm for solving (34). We denote the components of the solution by $\hat{\mathbf{y}}_i$, that is, $\hat{\mathbf{y}}_1 = \hat{\mathbf{v}}, \hat{\mathbf{y}}_2 = \hat{\mathbf{u}}$, and $\hat{\mathbf{y}}_3 = \hat{\lambda}$. We approximate all components simultaneously by a TT format with all the same TT blocks except the ℓ th block for some $\ell = 0, \dots, m + 1$, which actually carries the enumerator $i = 1, 2, 3$:

$$\hat{\mathbf{y}}_i = \sum_{s_0, \dots, s_m=1}^{r_0, \dots, r_m} y_{s_0}^{(0)} \otimes \dots \otimes \hat{y}_{s_{\ell-1}, s_\ell, i}^{(\ell)} \otimes y_{s_\ell, s_{\ell+1}}^{(\ell+1)} \otimes \dots \otimes y_{s_m}^{(m+1)}. \tag{35}$$

Without loss of generality, we can let $s_{-1} = s_{m+1} = 1$ to make $\hat{y}_{s_{\ell-1}, s_\ell, i}^{(\ell)}$ well defined for $\ell = 0, m + 1$.

The block TT decomposition (35) can be initialized with i placed in any TT block, but it can also be moved to a different TT block in the course of computations.⁵⁰ For example, suppose we need to replace i into $y^{(\ell+1)}$. We can introduce a matrix $\hat{Y}^{(\ell)}$ with elements

$$\hat{Y}^{(\ell)}(s_{\ell-1} j_\ell, i s_\ell) := \hat{y}_{s_{\ell-1}, s_\ell, i}^{(\ell)},$$

with j_ℓ replaced by k or n as appropriate and compute its SVD

$$\hat{Y}^{(\ell)} = U \Sigma V^\top \approx U_{r'_\ell} \Sigma_{r'_\ell} V_{r'_\ell}^\top,$$

truncated up to a rank r'_ℓ and/or an accuracy tolerance $\|\Sigma - \Sigma_{r'_\ell}\| < \varepsilon \|\Sigma\|$. Next, we construct new TT blocks

$$y_{s_{\ell-1}, s_{\ell}}^{(\ell)}(j_{\ell}) = U(s_{\ell-1}j_{\ell}, s'_{\ell}), \quad \hat{y}_{s'_{\ell}, s_{\ell+1}, t}^{(\ell+1)} = \sum_{s_{\ell}=1}^{r_{\ell}} \Sigma(s'_{\ell}, s'_{\ell}) V^{\top}(s'_{\ell}, t s_{\ell}) y_{s_{\ell}, s_{\ell+1}}^{(\ell+1)}, \quad (36)$$

where $s'_{\ell} = 1, \dots, r'_{\ell}$. Note that these TT blocks constitute a block TT decomposition similar to (35), but with ℓ replaced by $\ell + 1$. Similarly we can move t into $y^{(\ell-1)}$.

Thus, in the ℓ th step of the ALS algorithm, we can always ensure that the ℓ th TT block carries t as shown in (35). We can construct the so-called frame matrix

$$Y_{\ell} = \left[\sum_{s_0, \dots, s_{\ell-2}} y_{s_0}^{(0)} \otimes \dots \otimes Y_{s_{\ell-2}}^{(\ell-1)} \right] \otimes I \otimes \left[\sum_{s_{\ell+1}, \dots, s_m} Y_{s_{\ell+1}}^{(\ell+1)} \otimes \dots \otimes y_{s_m}^{(m+1)} \right], \quad (37)$$

where

$$Y_{s_{\ell-2}}^{(\ell-1)} = \left[y_{s_{\ell-2}, 1}^{(\ell-1)} \dots y_{s_{\ell-2}, r_{\ell-1}}^{(\ell-1)} \right], \quad Y_{s_{\ell+1}}^{(\ell+1)} = \left[y_{1, s_{\ell+1}}^{(\ell+1)} \dots y_{r_{\ell}, s_{\ell+1}}^{(\ell+1)} \right],$$

which can be seen as the block TT format (35) with $\hat{y}^{(\ell)}$ replaced by the identity. Note that (37) is a particular case of the matrix TT format (26), and hence it can be multiplied with the TT representations of \mathbf{K} , \mathbf{M}_1 , \mathbf{M}_2 , or \mathbf{M}_3 efficiently.⁴⁷ This allows us to generalize the ALS algorithm to a *block ALS* algorithm² by projecting the KKT system of momentum equations (34) block by block,

$$\begin{bmatrix} \hat{\mathbf{M}}_1 & 0 & -\hat{\mathbf{K}}^* \\ 0 & \beta \hat{\mathbf{M}}_2 & \hat{\mathbf{M}}_3^T \\ -\hat{\mathbf{K}} & \hat{\mathbf{M}}_3 & 0 \end{bmatrix} \text{vec}(\hat{y}^{(\ell)}) = \begin{bmatrix} Y_{\ell}^{\top} \mathbf{B}^{\top} \hat{\mu} \\ 0 \\ Y_{\ell}^{\top} (\hat{\mathbf{g}}_v + \mathbf{B}^{\top} \hat{p}) \end{bmatrix}, \quad (38)$$

which defines the elements of $\hat{y}^{(\ell)}$. Here, $\hat{A} = Y_{\ell}^{\top} \mathbf{A} Y_{\ell}$ for $\mathbf{A} \in \{\mathbf{K}, \mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3\}$ are the submatrices projected onto the frame matrix (37) in the Galerkin sense.

Since \mathbf{K}, \mathbf{K}^* are discretized diffusion-convection operators, we can prove that the block-reduced system (38) is nonsingular, and hence that, in contrast to the simple ALS algorithm, the block ALS method is well posed.

Theorem 1. *Suppose that the symmetric parts of \mathbf{K} and \mathbf{K}^* are positive definite, $\mathbf{K} + \mathbf{K}^{\top} > 0$, $\mathbf{K}^* + \mathbf{K}^{*\top} > 0$. Then the reduced matrix in (38) 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,

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

where $\lambda_{\min}(\cdot)$ is the minimal (real) eigenvalue of a matrix. We also use the fact that Y_k is orthogonal, which is ensured by SVD (36) in the course of the ALS iteration[†]. So, the symmetric part of $\hat{\mathbf{K}}$ (as well as of $\hat{\mathbf{K}}^*$) is positive definite. Moreover, by the same interlacing theorem we have that $\hat{\mathbf{M}}_1 \geq 0$ and $\hat{\mathbf{M}}_2 > 0$. Now we employ [54, theorem 3.2]: for the KKT matrix to be invertible, it is sufficient to have a full rank matrix of constraints $[-\hat{\mathbf{K}} \ \hat{\mathbf{M}}_3]$ and

$$\ker \underbrace{\begin{bmatrix} \hat{\mathbf{M}}_1 & 0 \\ 0 & \beta \hat{\mathbf{M}}_2 \end{bmatrix}}_{\hat{\mathbf{M}}_{12}} \cap \ker [-\hat{\mathbf{K}} \ \hat{\mathbf{M}}_3] = \{0\},$$

where $\ker(\cdot)$ is the kernel of a matrix. The first condition is fulfilled since $\hat{\mathbf{K}}$ is invertible. To verify the second criterion, consider a vector in the kernel of the constraints, which can be written as

[†]Indeed, the new TT block $y^{(\ell)}$ in (36) is produced from an orthogonal matrix of singular vectors U , and hence $\sum_{s_{\ell-1}} (y_{s_{\ell-1}, s_{\ell}}^{(\ell)})^T y_{s_{\ell-1}, s_{\ell}}^{(\ell)} = (U^{\top} U)_{s_{\ell}, s_{\ell}} = I_{s_{\ell}, s_{\ell}}$. So, we just need to initialize (35) with the same property. See Reference 43 for more details.

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

Now let us 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 nonnegative. Together with positive semidefiniteness of \hat{M}_{12} , this yields that w is not in its kernel. ■

As Theorem 4.1 shows, the block ALS algorithm avoids mixing of different blocks in (34) and hence excludes Example 1. A rigorous convergence proof of ALS is a much more difficult problem, and usually only a local convergence can be established,⁵⁵ while the algorithm converges rapidly from a fairly general initial guess in practice. Nevertheless, we believe that the consistency provided by Theorem 1 is an important part of justification of the approach.

Remark 1. In principle, the Chorin iteration (32)-(34) can be performed as is, but we observed a faster convergence if the TT blocks $v^{(0)}, \lambda^{(0)}$ of the velocity variables are made divergence free. This can be satisfied efficiently by modifying only the reduced KKT system (38) for $\ell = 0$, that is, we solve

$$\left[\begin{array}{cc|cc|cc} \hat{M}_1 & 0 & 0 & -\hat{K}^* & -\hat{B}^\top & \\ 0 & 0 & 0 & -\hat{B} & 0 & \\ \hline 0 & 0 & \beta \hat{M}_2 & \hat{M}_3^\top & 0 & \\ -\hat{K} & -\hat{B}^\top & \hat{M}_3 & 0 & 0 & \\ -\hat{B} & 0 & 0 & 0 & 0 & \end{array} \right] \begin{bmatrix} \hat{v} \\ \delta_p \\ \hat{u} \\ \hat{\lambda} \\ \delta_\mu \end{bmatrix} = \begin{bmatrix} \mathcal{Y}_0^\top \begin{pmatrix} \mathbf{B}^\top \hat{\mu} \\ 0 \end{pmatrix} \\ 0 \\ \mathcal{Y}_0^\top \begin{pmatrix} \hat{\mathbf{g}}_v + \mathbf{B}^\top \hat{p} \\ \hat{\mathbf{g}}_p \end{pmatrix} \end{bmatrix}, \quad (39)$$

where $\hat{B} = I_{r_0} \otimes [B^1 \ B^2]$ is the projected divergence matrix, and $\hat{\mathbf{g}}_p$ is the subvector of $\hat{\mathbf{g}}$ corresponding to pressure elements only and assemble the new TT block $\hat{y}^{(0)} = [\hat{v} \ \hat{u} \ \hat{\lambda}]$ without the auxiliary pressure vectors δ_p, δ_μ (they vanish as the iteration converges). All other ALS steps, that is, $\ell = 1, \dots, m + 1$, are carried out solving only the momentum equations (38).

4.3 | Preconditioning

Although the reduced system is much smaller than the original problem, it might still be rather large for a “naive” treatment, for example, (39) is of size $(3N_v + 2N_p)r_0$. We solve it using GMRES, together with a preconditioner introduced in References 2 and 56, which is based on the Schur complement approximation. The reduced matrix (39) admits a (straightforwardly verifiable) decomposition

$$\begin{bmatrix} \hat{\mathcal{M}}_1 & 0 & -\hat{\mathcal{K}}^* \\ 0 & \beta \hat{\mathcal{M}}_2 & \hat{\mathcal{M}}_3^\top \\ -\hat{\mathcal{K}} & \hat{\mathcal{M}}_3 & 0 \end{bmatrix} = \begin{bmatrix} I & * & * \\ & I & \\ & & I \end{bmatrix} \begin{bmatrix} -\hat{S} \\ \beta \hat{\mathcal{M}}_2 & \hat{\mathcal{M}}_3^\top \\ -\hat{\mathcal{K}} & \hat{\mathcal{M}}_3 & 0 \end{bmatrix}, \quad (40)$$

where $\hat{S} = \hat{\mathcal{K}}^* + \hat{\mathcal{M}}_1 \hat{\mathcal{K}}^{-1} \hat{\mathcal{M}}_3 (\beta \hat{\mathcal{M}}_2)^{-1} \hat{\mathcal{M}}_3^\top$ is the Schur complement. We precondition (39) by the second (antitriangular) factor from (40). However, $\hat{\mathcal{K}}$ and \hat{S} are still large and must be approximated.

The first matrix has the Kronecker form

$$\hat{\mathcal{K}} = \mathcal{L} \otimes I_{r_0} + \mathcal{M} \otimes \hat{C} + \sum_{s=1}^{r_0} \mathcal{F}_s \otimes \hat{H}_s,$$

where \hat{C}, \hat{H}_s are partial projections of the Euler and convection (29) matrices onto $\sum_{s_1, \dots, s_m} Y_{s_1}^{(1)} \otimes \dots \otimes Y_{s_m}^{(m+1)}$. We approximate $\hat{\mathcal{K}}$ by the following Sylvester operator:

$$\tilde{\mathcal{K}} = \left(\mathcal{L} + \sum_{s=1}^{r_0} \frac{\text{tr}(\hat{H}_s)}{r_0} \cdot \mathcal{F}_s \right) \otimes I_{r_0} + \mathcal{M} \otimes \hat{C}, \quad (41)$$

where $\text{tr}(\hat{H}_s)/r_0$ is an average of the eigenvalues of \hat{H}_s . Now $\tilde{\mathcal{K}}$ can be inverted by the Bartels-Stewart method, since \hat{C} is small and can be easily Schur factorized.

The Schur complement \hat{S} is approximated by a factorized form,⁵⁶

$$\tilde{S} = \left(\hat{\mathcal{K}}^* + \frac{1}{c} \hat{\mathcal{M}}_1 \right) \hat{\mathcal{K}}^{-1} (\hat{\mathcal{K}} + c \cdot \hat{\mathcal{M}}_3 (\beta \hat{\mathcal{M}}_2)^{-1} \hat{\mathcal{M}}_3^T), \quad (42)$$

where $c > 0$ is a normalization constant. Traditionally, it is proposed^{2,56} to set $c = \sqrt{\beta}$, and this has proved to be the optimal choice for the tracking objective function. However, for the vorticity norm function, the optimal c might 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 five GMRES iterations. Since it is difficult to differentiate the GMRES residual with respect to c , we employ the zero-order *golden section* optimization algorithm,⁵⁷ initialized with an interval $\log_{10} c \in [\log_{10} \sqrt{\beta} - 6, \log_{10} \sqrt{\beta} + 6]$. It is sufficient to carry out this procedure only in the first Picard iteration, since the optimal c does not change much in the latter iterations. We also approximate the factors in the brackets in (42) similarly to (41). The full procedure is summarized in Algorithm 1.

Algorithm 1. Block ALS iteration for solving parametric inverse Navier-Stokes equations in the TT format

- 1: Initialize $\hat{v} = \hat{u} = \hat{\lambda} = 0$ and the block TT format (35) with ι in $\hat{y}^{(0)}$.
 - 2: **for** iter=1,2,...until convergence **do**
 - 3: Construct nonlinear terms $\mathcal{F}_{\hat{v}}$ and $\mathcal{W}_{\hat{v}}$ as shown in (29), (30).
 - 4: Compute the pressure components as shown in (32) and (33).
 - 5: Assemble and solve the projected system (39).
 - 6: Compute SVD of $\hat{y}^{(0)}$ and move ι to $y^{(1)}$ as shown in (36).
 - 7: **for** $\ell = 1, 2, \dots, m+1, m, \dots, 1$ **do**
 - 8: Assemble and solve the projected system (38).
 - 9: Compute SVD of $\hat{y}^{(\ell)}$.
 - 10: **if** $\sim \ell$ is increasing **then**
 - 11: Move ι to $y^{(\ell+1)}$ as shown in (36).
 - 12: **else**
 - 13: Move ι to $y^{(\ell-1)}$ analogously.
 - 14: **end if**
 - 15: **end for**
 - 16: **end for**
-

5 | NUMERICAL RESULTS

This research made use of the Balena High Performance Computing (HPC) Service at the University of Bath. We have implemented the computational codes on the basis of the MATLAB TT-Toolbox⁵⁸ and the T-IFISS 1.1 toolbox⁵⁹ and run on one core of Balena, an Intel E5-2650 v2 CPU.

5.1 | Flow around an obstacle

In the first experiment, we benchmark Algorithm 1 on the Karman vortex problem of simulating a flow in the obstacle domain shown in Figure 1.

5.1.1 | Estimation of errors

The model is influenced by different sources of error. However, one is typically interested in the total error and the corresponding numerical cost. We begin with investigating different types of error, in order to equilibrate them in the complexity experiment. Throughout these experiments, the model parameters are fixed to the following

values: viscosity $\nu = 10^{-3}$, final time $T = 20$, regularization parameter $\beta = 10^{-2}$, and the decay rate of the stochastic expansion $\gamma = 2.5$. Moreover, the polynomial degree in the stochastic Galerkin basis (11) is set to $d = 8$.

Spatial discretization error.

In order to determine how the solution converges with the spatial grid refinement, we further fix the number of stochastic variables $m = 0$ (ie, solve the deterministic problem), the number of time steps $N_t = 2048$, and the TT truncation threshold $\varepsilon = 10^{-5}$ in Algorithm 1. We compute the mean square vorticity $\|\nabla \times \mathbf{v}\|_{L^2(D)}^2$ and the mean square mass of the velocity, $\|\mathbf{v}\|_{L^2(D)}^2$, for different numbers of spatial grid refinements n_{ref} . The results are shown in Table 1(left).

We deduce the empirical convergence rate $\epsilon_{n_{\text{ref}}} = C \cdot 2^{-p \cdot n_{\text{ref}}}$ using Runge's rule:

$$p = \log_2 \left[\frac{Q_{n_{\text{ref}}} - Q_{n_{\text{ref}}+1}}{Q_{n_{\text{ref}}+1} - Q_{n_{\text{ref}}+2}} \right], \quad C = \frac{|Q_{n_{\text{ref}}} - Q_{n_{\text{ref}}+1}|}{|Q_{n_{\text{ref}}+2}|} \cdot \frac{2^{p \cdot n_{\text{ref}}}}{1 - 2^{-p}},$$

where $Q_{n_{\text{ref}}}$ is a quantity of interest ($\|\nabla \times \mathbf{v}\|_{L^2(D)}^2$ or $\|\mathbf{v}\|_{L^2(D)}^2$) computed with the refinement level n_{ref} . We see that the largest relative error is attained for the vorticity and is of the form

$$\epsilon_{n_{\text{ref}}} = 1.85 \cdot 10^{-3} \cdot 2^{-1.9 \cdot n_{\text{ref}}}. \quad (43)$$

This agrees with the second order of approximation of the P_1 finite elements for twice differentiable functions.

The number N_v of the P_2 basis functions for each velocity component, the number N_p of the P_1 basis functions for the pressure, as well as the total number of spatial degrees of freedom in the KKT system $N_x = 6N_v + 2N_p$ are shown in Table 2.

Time discretization error.

Now we fix $n_{\text{ref}} = 2$ and vary the number of time steps in the Euler scheme instead. From Table 1(right), we confirm the first order of convergence of the implicit Euler scheme:

$$\epsilon_{N_t} = 0.5404 \cdot \frac{1}{N_t}. \quad (44)$$

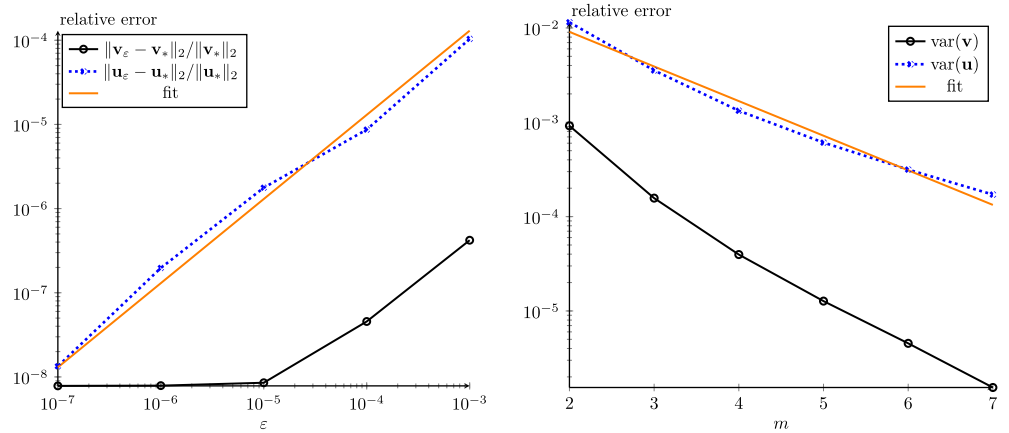
TABLE 1 Vorticity and mass of the solution for different spatial grid levels (left) and numbers of time steps (right) and the empirical convergence rate parameters p and C

n_{ref}	$\ \nabla \times \mathbf{v}\ _{L^2(D)}^2$	$\ \mathbf{v}\ _{L^2(D)}^2$	N_t	$\ \nabla \times \mathbf{v}\ _{L^2(D)}^2$	$\ \mathbf{v}\ _{L^2(D)}^2$
1	38.3504	8.058752	1024	38.3746	8.06114
2	38.3645	8.059017	2048	38.3645	8.05901
3	38.3682	8.059087	4096	38.3594	8.05795
p	1.89	1.91	p	1.00	1.00
C	$1.85 \cdot 10^{-3}$	$1.69 \cdot 10^{-4}$	C	0.5404	0.5404

TABLE 2 Sizes of the space discretization matrices for different spatial grid levels

n_{ref}	1	2	3
N_v	3604	14 152	56 080
N_p	934	3604	14 152
N_x	23 492	92 120	364 784

FIGURE 3 Relative errors in the solution for different TT approximation thresholds (left) and numbers of stochastic variables (right) [Colour figure can be viewed at wileyonlinelibrary.com]



TT approximation error.

In this experiment, we fix $n_{\text{ref}} = 2$, $N_t = 2048$, and vary the TT approximation threshold ϵ in Algorithm 1 from 10^{-3} to 10^{-7} . We compare the velocity \mathbf{v}_ϵ and the control \mathbf{u}_ϵ , computed for the given tolerance ϵ , with the reference solutions \mathbf{v}_* , \mathbf{u}_* , computed with $\epsilon = 10^{-8}$. The results are shown in Figure 3(left).

We see that the dominant error occurs in the control component. The least squares log-linear fit gives the following dependence:

$$\epsilon_\epsilon = 0.13 \cdot \epsilon. \quad (45)$$

Stochastic parametrization error.

The exact stochastic inflow is an infinite dimensional random field. For computational purposes, we parametrize it by a truncated stochastic expansion of κ . This, however, introduces an additional error, corresponding to the truncated tail of the expansion. In Figure 3(right), we vary the number of random variables m and plot the relative difference of the velocity and the control variance with respect to the reference solutions $\text{var}(\mathbf{v}_*)$, $\text{var}(\mathbf{u}_*)$, computed with $m = 8$. Again, the error is dominated by the control component. Although the decay rate in the inflow is proportional to m^{-3} , the convergence of the variance looks more like exponential. The least squares log-linear fit gives

$$\epsilon_m = \exp(-0.8448 \cdot m - 3.011). \quad (46)$$

5.1.2 | Total error-performance test

Equipped with the quantitative error estimates (43)-(46), we can equilibrate all errors. Namely, we vary only n_{ref} and compute the corresponding space discretization error $\epsilon_{n_{\text{ref}}}$. Then we assume $\epsilon_{N_t} = \epsilon_\epsilon = \epsilon_m = \epsilon_{n_{\text{ref}}}$ and set the other parameters through the inverses of the dependencies (44)-(46):

$$\begin{aligned} N_t &= 2^{\lceil \log_2(0.5404) - \log_2(\epsilon) \rceil}, \\ \epsilon &= 7.7 \cdot \epsilon, \\ m &= \lceil -2.7256 \cdot \log_{10}(\epsilon) - 3.5639 \rceil. \end{aligned}$$

In Figure 4, we show the total CPU times vs the equilibrated error ϵ (realized by different n_{ref}), as well as the maximal TT rank of the solution.

We can observe that the CPU time grows proportionally to ϵ^{-2} . Qualitatively, this is the complexity of the deterministic problem: due to the second order of approximation in space, the number of spatial degrees of freedom for a two-dimensional space $N_x = \mathcal{O}(h^{-2})$ is of the order of the reciprocal of the discretization error, whereas the first order of approximation in time implies $N_t = \mathcal{O}(\epsilon^{-1})$. The total cost of the deterministic time-dependent problem is $\mathcal{O}(N_x N_t) = \mathcal{O}(\epsilon^{-2})$. This demonstrates that the tensor product methods can solve the stochastic time-dependent problem with the

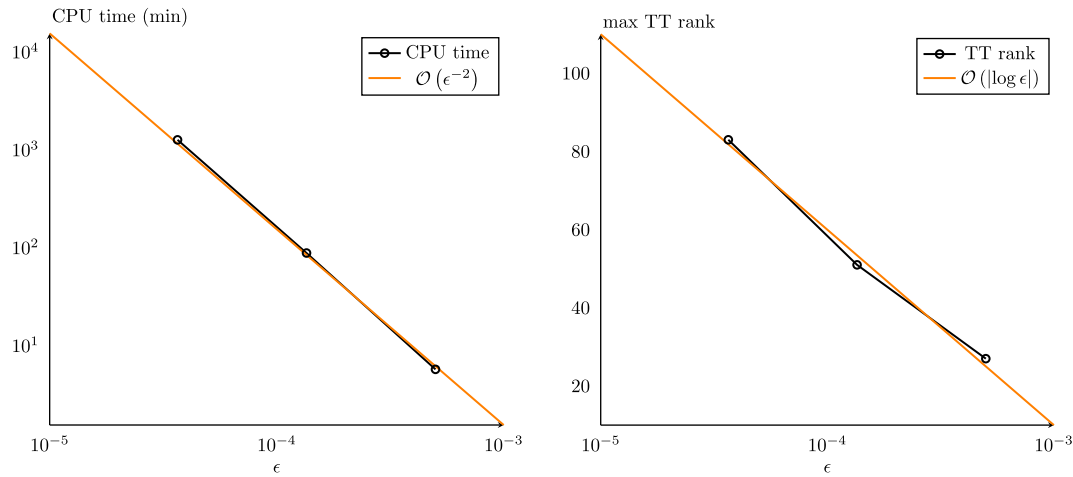


FIGURE 4 Total CPU time (left) and TT rank (right) vs total approximation error [Colour figure can be viewed at wileyonlinelibrary.com]

asymptotic complexity of the deterministic problem (or the time-dependent problem with the asymptotic complexity of the stationary problem³⁶).

On the other hand, the cost of a straightforward solution of the stochastic problem is $\mathcal{O}(N_x N_t N_\xi)$, where N_ξ is the number of degrees of freedom introduced in the stochastic variables. A traditional Monte Carlo approach would involve $N_\xi = \mathcal{O}(\epsilon^{-2})$, and hence the total cost of the order of ϵ^{-4} . Quasi Monte Carlo or sparse polynomial chaos techniques can reduce N_ξ to $\mathcal{O}(\epsilon^{-1})$, and the total cost to $\mathcal{O}(\epsilon^{-3})$, which is still asymptotically larger than the cost of the TT approach.

5.1.3 | Influence of the control

The reduction power of the tensor decompositions stems from their low storage, in particular, low TT ranks for the desired accuracy. The tensor ranks are hugely related to the structure of the solution. In particular, it is hard to expect low TT ranks for a turbulent flow, which develops for high Reynolds numbers in absence of the control. It is the inverse problem, designed in order to impose an additional structure, that allows to keep the TT ranks of the controlled solution low. In Figure 5, we compare mean flow streamlines in the forward problem (uncontrolled flow) and in the inverse problem. We see that the vorticity-minimizing control can efficiently enforce the laminar structure of the flow. The variance of the controlled flow velocity is shown in Figure 6, and the mean control is shown in Figure 7. As expected, the control is localized around the obstacle.

To investigate the influence of the regularization parameter on the flow, we increase the viscosity to $\nu = 10^{-2}$ such that the optimal control problem with a large parameter $\beta = 10^2$ becomes feasible. The velocity magnitudes for $\beta = \infty$, $\beta = 10^2$, and $\beta = 10^{-2}$ are shown in Figure 8. Notice how the flow becomes more laminar as more control is applied.

5.1.4 | Influence of the model parameters

Finally, we investigate how the performance of the TT algorithm is affected by different model parameters. We vary the regularization parameter β , the rate of decay in the stochastic expansion of κ of the inflow γ , and the viscosity ν . The computational times and TT ranks are shown in Figures 9, 10, and 11, respectively. The unchanged parameters are set to their default values listed in the beginning of Section 5.1.1. Moreover, we set the spatial discretization level $n_{\text{ref}} = 2$, and the other approximation parameters accordingly, as derived in Section 5.1.2.

We see that the lowest cost is attained at intermediate values of the regularization parameter. For smaller β , the velocity is driven closer to the stationary field. However, our scheme computes the difference between the stationary field and the actual velocity. Similarly to the solution error in iterative methods, the smaller this difference is in magnitude, the more

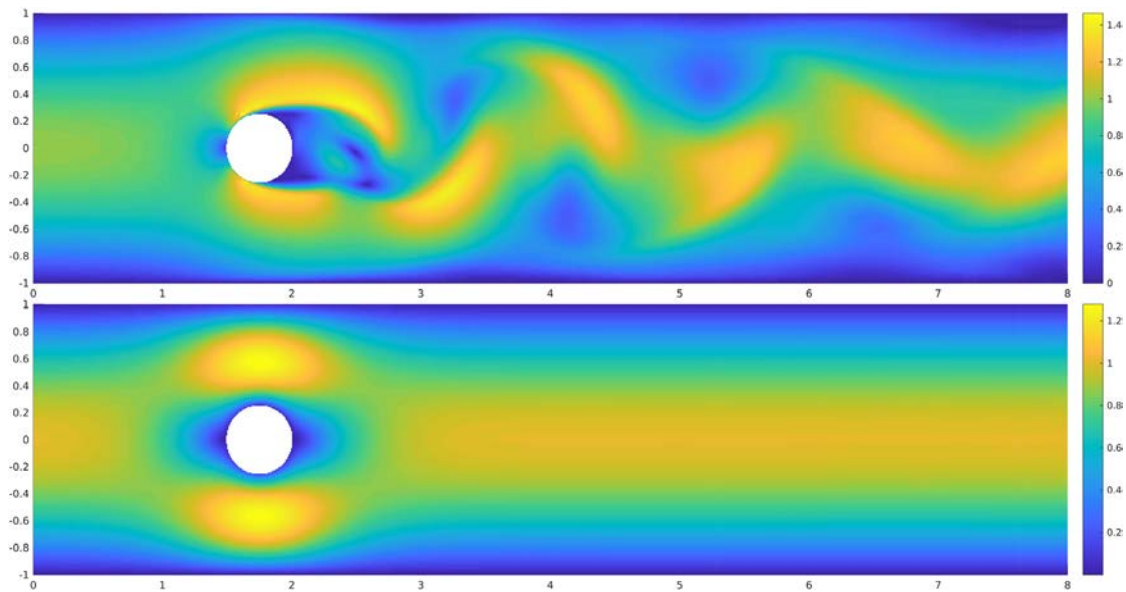


FIGURE 5 Velocity magnitude of the mean uncontrolled (top) and controlled (bottom) flows with $\beta = 10^{-2}$ at $t = 20$, $\nu = 10^{-3}$ [Colour figure can be viewed at wileyonlinelibrary.com]

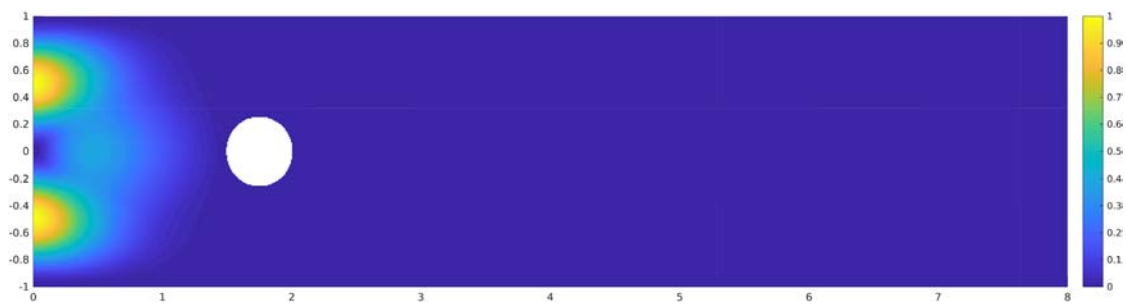


FIGURE 6 Magnitude of the variance of the controlled velocity at $t = 20$, $\nu = 10^{-3}$ [Colour figure can be viewed at wileyonlinelibrary.com]

complicated and oscillatory structure it has. This leads to larger TT ranks in the limit of small β , and consequently larger computing cost.

On the other hand, when β is large, the control becomes too weak to drive the solution away from the uncontrolled flow. The solution starts to manifest turbulent behavior, which increases the TT ranks as well. Moreover, a stronger influence of the nonlinear term leads to more Picard iterations, such that the total computational time increases further. The best scenario for the TT solver is therefore the balanced regularization, which corresponds to a sufficient, but not excessive control.

The rate of decay in the Fourier expansion (4) is governed by the covariance function of the random field. Highly correlated field can be approximated by fewer independent random quantities, which corresponds to a faster decay, that is, larger γ . Smaller γ means that more stochastic variables have a strong influence on the solution. Figure 10 shows that the computational cost is inversely proportional to the decay rate.

On the other hand, from Figure 11, we can observe that the TT scheme is quite robust with respect to the Reynolds number. In particular, the solution has the same TT ranks for the viscosity ranging from 10^{-4} to 10^{-2} . The computational time grows logarithmically with the Reynolds number. This is mainly due to the GMRES solver for (39), which needs more iterations for more convection-dominated problems.

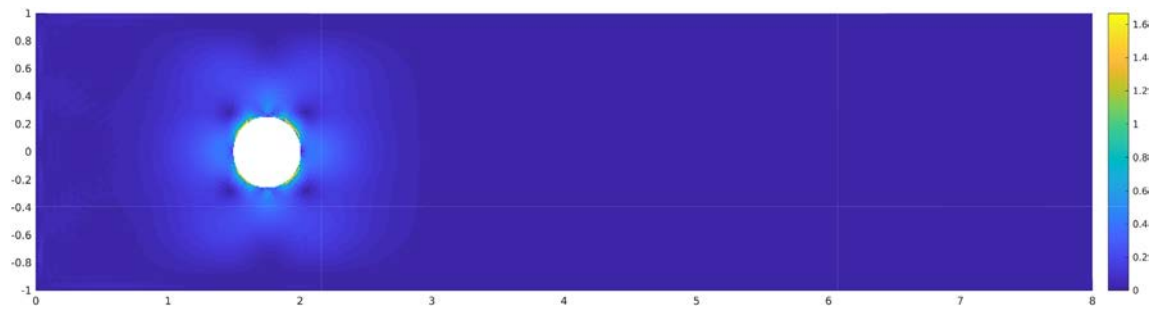


FIGURE 7 Magnitude of the mean control signal at $t = 20$, $\nu = 10^{-3}$ [Colour figure can be viewed at [wileyonlinelibrary.com](https://onlinelibrary.wiley.com/doi/10.1002/nd.4843)]

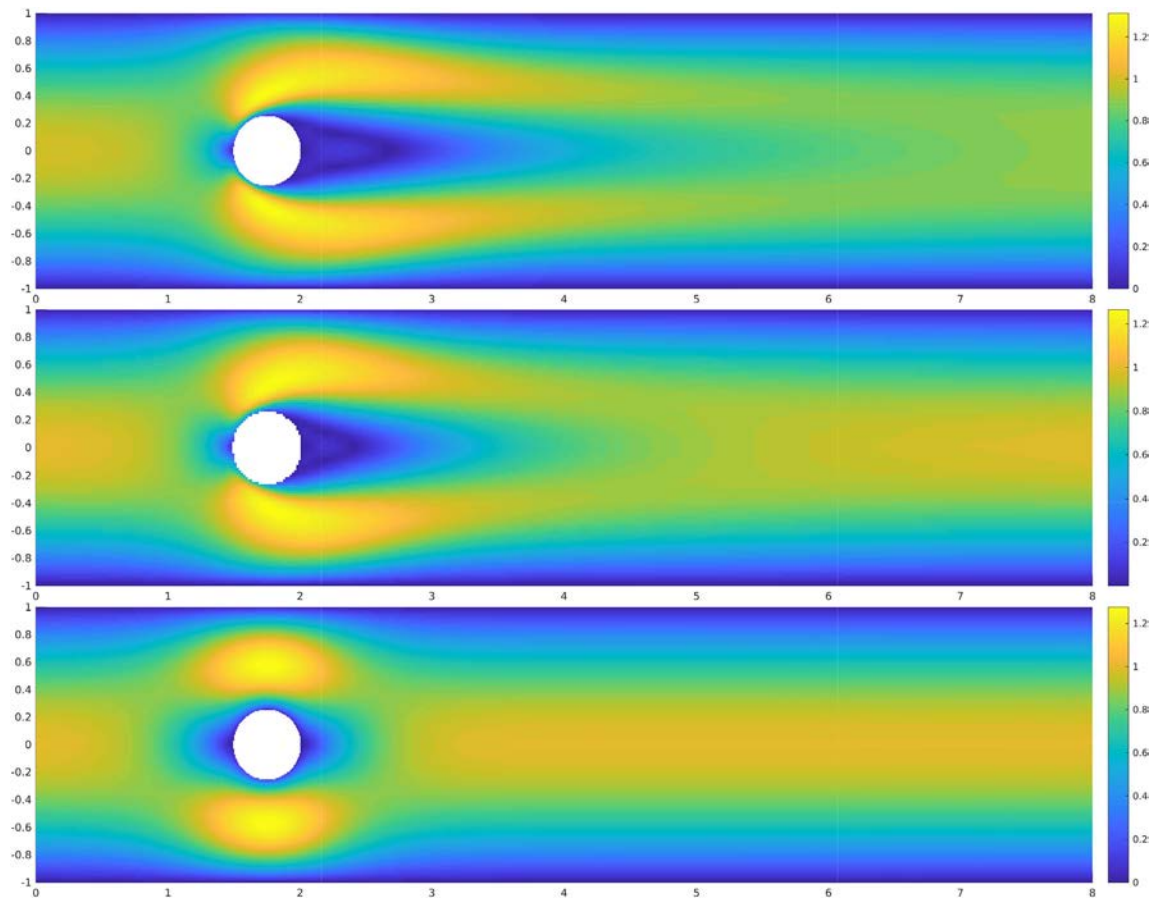


FIGURE 8 Velocity magnitude of the mean uncontrolled (top) and controlled flows with $\beta = 10^2$ (middle) and $\beta = 10^{-2}$ (bottom) at $t = 20$, $\nu = 10^{-2}$ [Colour figure can be viewed at [wileyonlinelibrary.com](https://onlinelibrary.wiley.com/doi/10.1002/nd.4843)]

5.2 | Backward step domain with uncertain inflow

Additionally, we test our solver on another benchmark problem, the backward step domain (see Figure 2). As most results for this problem are qualitatively very similar to the flow around an obstacle, we only report a small number of test results. The inflow field at the leftmost boundary Y_{in} is given in (5). We choose the same other parameters as in the first experiment: viscosity $\nu = 10^{-3}$, final time $T = 20$, regularization parameter $\beta = 10^{-2}$, and the stochastic expansion decay rate $\gamma = 2.5$.

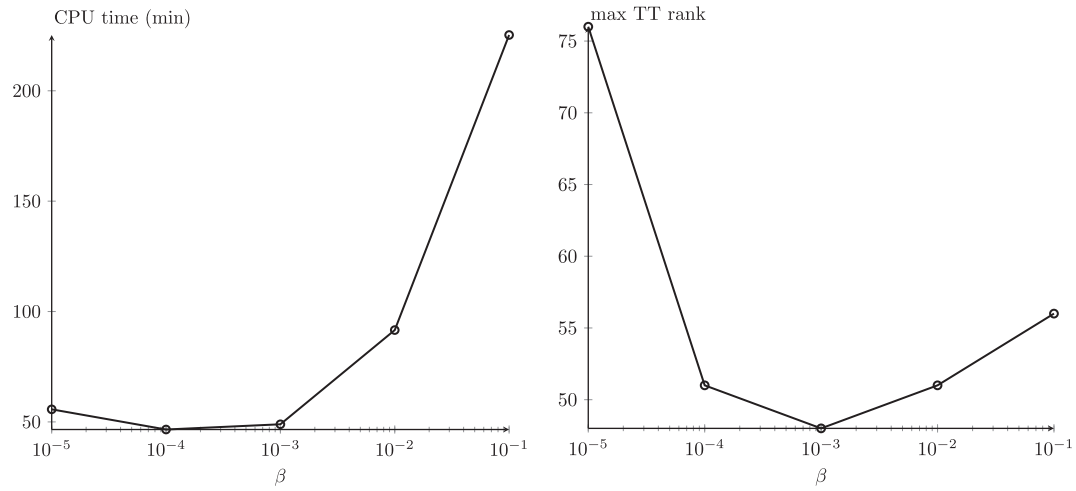


FIGURE 9 Total CPU time (left) and TT rank (right) vs regularization parameter β

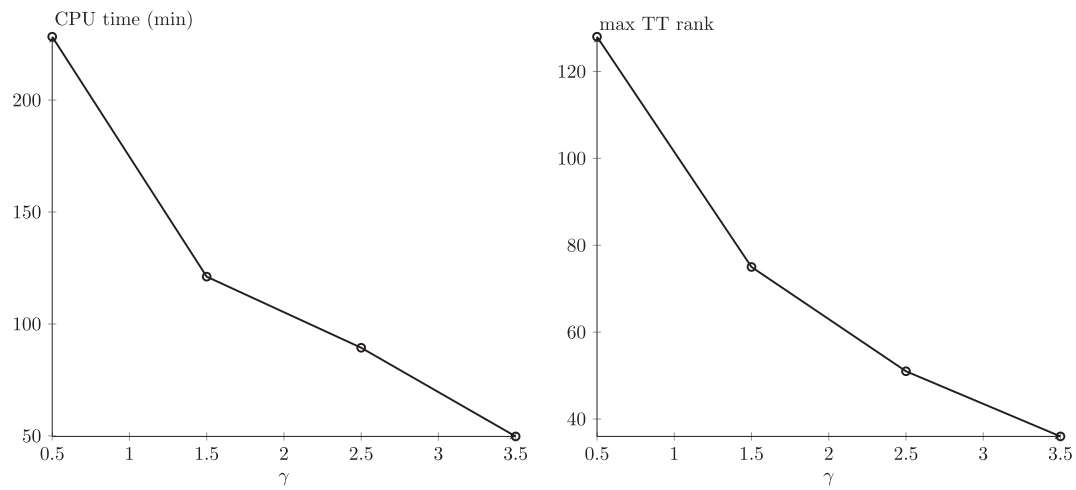


FIGURE 10 Total CPU time (left) and TT rank (right) vs stochastic expansion of κ decay rate γ

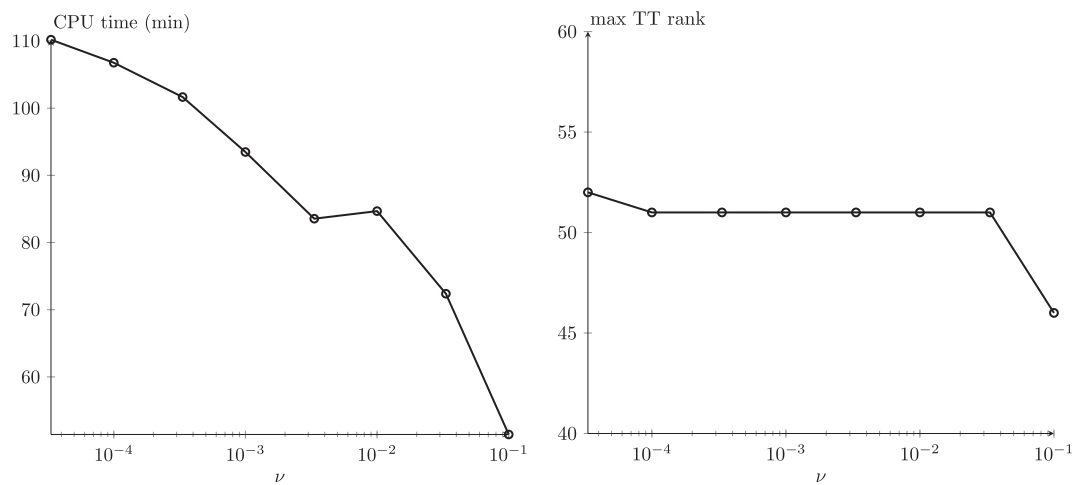


FIGURE 11 Total CPU time (left) and TT rank (right) vs viscosity ν

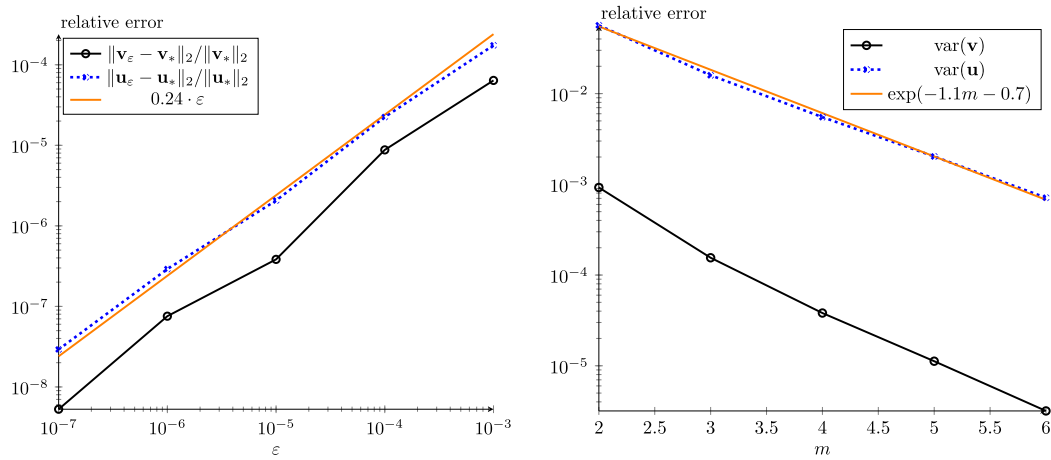


FIGURE 12 Relative errors in the solution for different TT approximation thresholds (left) and numbers of stochastic variables (right) [Colour figure can be viewed at wileyonlinelibrary.com]

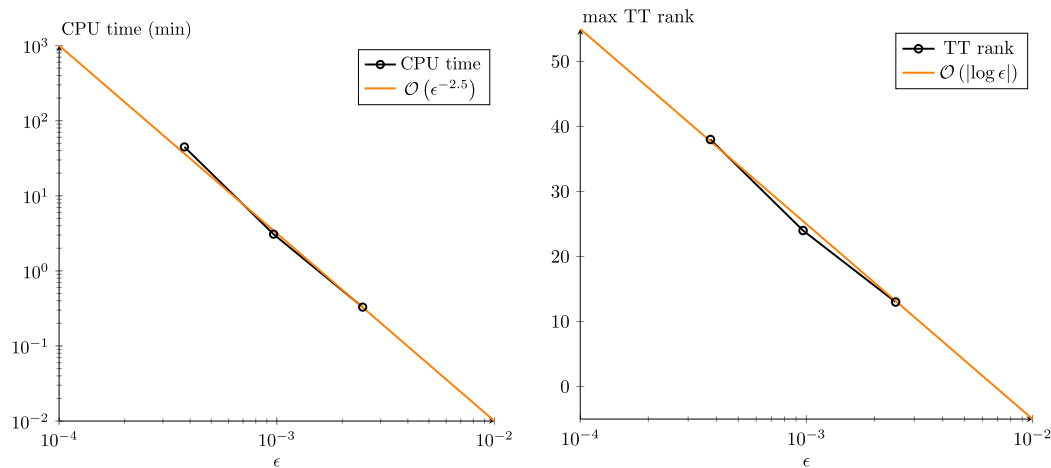


FIGURE 13 Total CPU time (left) and TT rank (right) vs total approximation error [Colour figure can be viewed at wileyonlinelibrary.com]

From the vorticity and the mass of the deterministic solution computed for different spatial mesh refinement levels and numbers of time steps, we infer the following empirical convergence rates:

$$\epsilon_{n_{\text{ref}}} = 6.37 \cdot 10^{-3} \cdot 2^{-1.36 \cdot n_{\text{ref}}}, \quad \epsilon_{N_t} = 0.5404 \cdot \frac{1}{N_t}.$$

Notice that the spatial discretization order is lower than 2, which can be attributed to the reentrant corner in the domain.

The errors of the TT approximation and stochastic parametrization are shown in Figure 12.

The fitted error expressions are shown in the corresponding plots. Now we can equilibrate all errors to the spatial discretization error for $n_{\text{ref}} = 1, 2, 3$ and measure the computational complexity with respect to the total error (see Figure 13).

We see that the slope is higher than in the obstacle example. This is due to the slower convergence of the spatial discretization, whereas the TT rank still depends logarithmically on the accuracy. Higher order finite elements in space might make the scheme more efficient. Nevertheless, even the simple discretization demonstrates that the low-rank decompositions can perform uncertainty quantification with the same asymptotic cost as the solution of a deterministic problem.

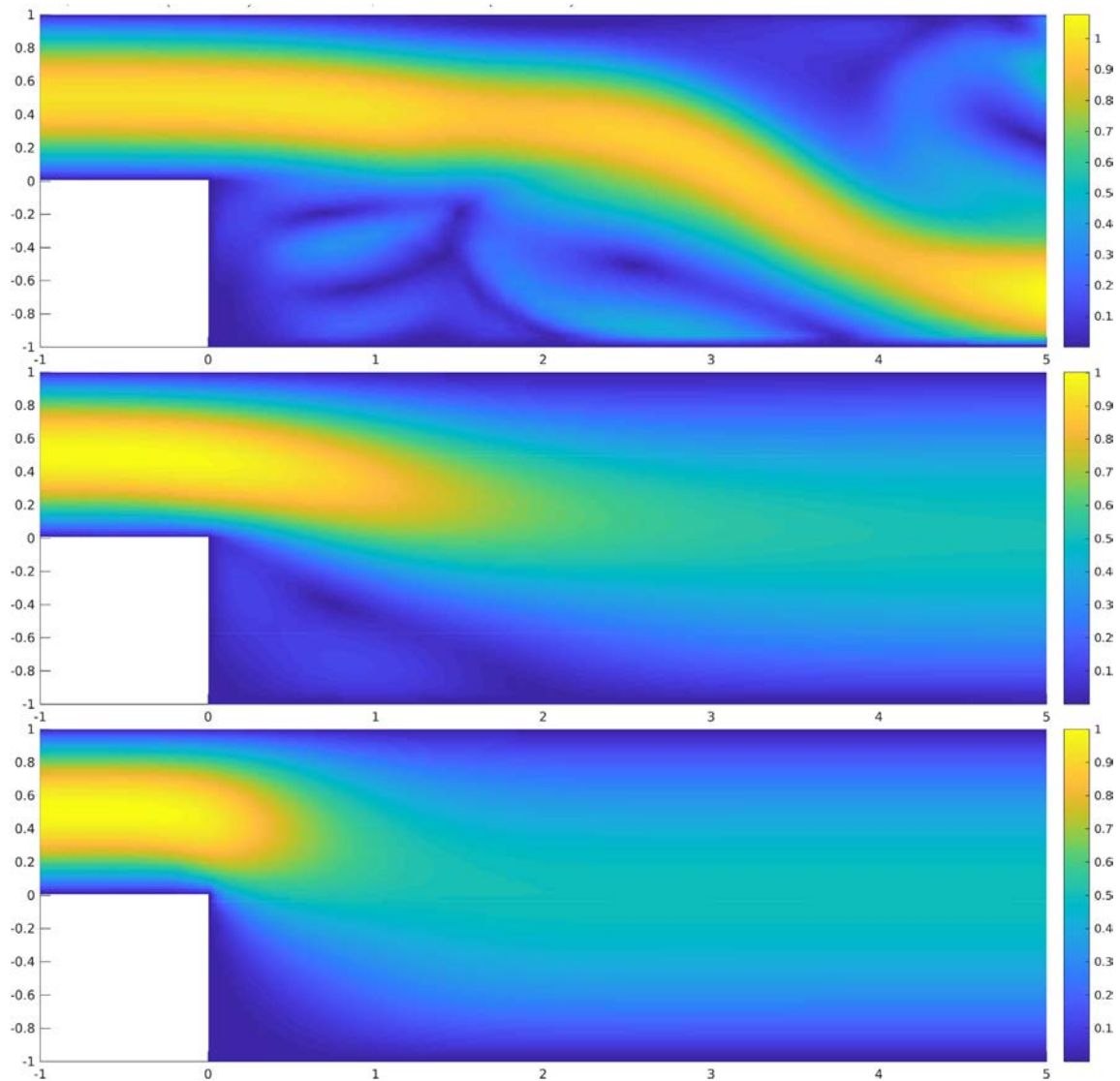


FIGURE 14 Mean velocity magnitude in the backstep domain, uncontrolled flow (top), controlled with $\beta = 10^2$ (middle) and with $\beta = 10^{-2}$ (bottom) [Colour figure can be viewed at [wileyonlinelibrary.com](https://onlinelibrary.wiley.com/terms-and-conditions)]

The mean velocities are shown in Figure 14. Again, we can see stabilization of the flow when a stronger control is applied.

6 | CONCLUSIONS AND OUTLOOK

We demonstrated the applicability of low-rank tensor decompositions to the solution of two benchmark optimal control problems constrained by two-dimensional unsteady Navier-Stokes equations with stochastic inputs. This problem has a 3-fold challenge: a nonlinear time-dependent PDE, an optimization problem using a Lagrangian approach, and random inputs. As particular model problems, we consider the minimization of vorticity of a von Kármán flow around an obstacle, as well as a backward step domain problem. Each of the two models problems has an uncertain inflow condition.

We discretize our problem using the tensor product stochastic Galerkin FEM, in which case the numbers of degrees of freedom coming from space, time, and all individual random variables multiply. The largest size of the problem to solve for $n_{\text{ref}} = 2$, $m = 8$, and $N_t = 2048$ in the full representation would be $N_x \cdot 8^m \cdot N_t \approx 10^{15}$, which exceeds our memory capacity by several orders of magnitude.


We avoid storing or computing such excessive amounts of data by approximating the solution in the TT decomposition. A crucial part of the scheme is an alternating linear scheme solver, which computes the TT factors directly. This algorithm required substantial modifications in order to be applicable to the SOCP: we preserve the saddle-point structure in the reduced model and split the pressure and the velocity components in order to equilibrate the sizes of the tensors involved in the alternating algorithm. The largest size of the reduced problem we actually needed to solve in Algorithm 1 for the TT rank $r = 60$ was $N_x \cdot r \approx 5 \cdot 10^6$. This is still rather large and requires preconditioning. Nevertheless, this is significantly smaller than the complexity one could expect from sampling-based methods, let alone the full discrete problem. We observed an empirical complexity rate of $\mathcal{O}(\epsilon^{-2})$ for the total error ϵ , which indicates that the stochastic time-dependent problem can be solved with the cost of a deterministic problem.

For future research, we plan to tackle more general nonlinear problems, as well as boundary and constrained controls. We expect difficulties in applying the tensor decompositions considered in this article to discontinuous functions, such as the indicator function of an active set. However, nonlinear problems involving smooth functions seem to be suitable for the low-rank approach.⁶⁰

ACKNOWLEDGMENTS

This work was started when A.O and M.S were at Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg, Germany. S.D. gratefully acknowledges funding by the EPSRC Fellowship EP/M019004/1. A.O. gratefully acknowledges funding by the US Department of Energy Office of Advanced Scientific Computing Research, Applied Mathematics program, under Award Number DE-SC0009301.

ORCID

Akwum Onwunta  <https://orcid.org/0000-0003-2110-8881>

REFERENCES

1. Chorin AJ, Lu F. *Discrete approach to stochastic parametrization and dimension reduction in nonlinear dynamics*. *Proc Natl Acad Sci USA*. 2015;112:9804-9809.
2. Benner P, Dolgov S, Onwunta A, Stoll M. *Low-rank solvers for unsteady Stokes-Brinkman optimal control problem with random data*. *Comput Methods Appl Mech Eng*. 2016;304:26-54.
3. Benner P, Onwunta A, Stoll M. *Block-diagonal preconditioning for optimal control problems constrained by PDEs with uncertain inputs*. *SIAM J Matrix Anal Appl*. 2016;37:491-518.
4. Powell CE, Silvester DJ. *Preconditioning steady-state Navier-Stokes equations with random data*. *SIAM J Sci Comput*. 2012;34:A2482-A2506.
5. Sousedik B, Ghanem RG. *Truncated hierarchical preconditioning for the stochastic Galerkin FEM*. *Int J Uncertainty Quantif*. 2014;4:333-348.
6. Sousedik B, Elman HC. *Stochastic Galerkin methods for the steady-state Navier-Stokes equations*. *J Comput Phys*. 2016;316:435-452. <https://doi.org/10.1016/j.jcp.2016.04.013>.
7. Lee K, Elman HC, Sousedik B. *A low-rank solver for the Navier-Stokes equations with uncertain viscosity*. *SIAM/ASA J Uncertainty Quantif*. 2019;7:1275-1300. <https://doi.org/10.1137/17M1151912>.
8. Benner P, Onwunta A, Stoll M. *Low-rank solution of unsteady diffusion equations with stochastic coefficients*. *SIAM/ASA J Uncertainty Quantif*. 2015;3:622-649.
9. Rosseel E, Wells GN. *Optimal control with stochastic PDE constraints and uncertain controls*. *Comput Methods Appl Mech Eng*. 2012;213-216:152-167.
10. Bepalov A, Powell CE, Silvester D. *Energy norm a posteriori error estimation for parametric operator equations*. *SIAM J Sci Comput*. 2013;36:A339-A363.
11. Cohen A, DeVore R, Schwab C. *Convergence rates of best N-term Galerkin approximations for a class of elliptic SPDEs*. *Found Comput Math*. 2010;10:615-646. <https://doi.org/10.1007/s10208-010-9072-2>.
12. Frauenfelder P, Schwab C, Todor RA. *Finite elements for elliptic problems with stochastic coefficients*. *Comput Methods Appl Mech Eng*. 2005;194:205-228.
13. Matthies H, Keese A. *Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations*. *Comput Methods Appl Mech Eng*. 2005;194:1295-1331. <https://doi.org/10.1016/j.cma.2004.05.027>.
14. Bieri M, Schwab C. *Sparse high order FEM for elliptic SPDEs*. *Comput Methods Appl Mech Eng*. 2009;198:1149-1170. <https://doi.org/10.1016/j.cma.2008.08.019>.
15. Hackbusch W. *Tensor Spaces And Numerical Tensor Calculus*. Berlin, Germany: Springer-Verlag; 2012.
16. Khoromskij BN. *Tensor numerical methods for multidimensional PDEs: theoretical analysis and initial applications*. *ESAIM Proc*. 2015;48:1-28. <https://doi.org/10.1051/proc/201448001>.
17. Eigel M, Pfeiffer M, Schneider R. *Adaptive stochastic Galerkin FEM with hierarchical tensor representations*. *Numerische Mathematik*. 2017;136:765-803.

18. Espig M, Hackbusch W, Litvinenko A, Matthies HG, Wahnert P. *Efficient low-rank approximation of the stochastic Galerkin matrix in tensor formats*. *Comput Math Appl*. 2014;67:818-829.
19. Khoromskij BN, Schwab C. *Tensor-structured Galerkin approximation of parametric and stochastic elliptic PDEs*. *SIAM J Sci Comput*. 2011;33:364-385.
20. Borzi A, von Winckel G. *Multigrid methods and sparse-grid collocation techniques for parabolic optimal control problems with random coefficients*. *SIAM J Sci Comput*. 2009;31:2172-2192.
21. Kouri DP, Heinkenschloss M, Ridzal D, van Bloemen Waanders BG. *A trust-region algorithm with adaptive stochastic collocation for PDE optimization under uncertainty*. *SIAM J Sci Comput*. 2013;35:A1847-A1879.
22. Tiesler H, Kirby RM, Xiu D, Preusser T. *Stochastic collocation for optimal control problems with stochastic PDE constraints*. *SIAM J Control Optim*. 2012;50:2659-2682.
23. Ali AA, Ullmann E, Hinze M. *Multilevel Monte Carlo analysis for optimal control of elliptic PDEs with random coefficients*. *SIAM/ASA J Uncertainty Quantif*. 2017;5:466-492.
24. Garreis S, Ulbrich M. *Constrained optimization with low-rank tensors and applications to parametric problems with PDEs*. *SIAM J Sci Comput*. 2017;39:A25-A54. <https://doi.org/10.1137/16M1057607>.
25. Herzog R, Sachs E. *Preconditioned conjugate gradient method for optimal control problems with control and state constraints*. *SIAM J Matrix Anal Appl*. 2010;31:2291-2317.
26. Ito K, Kunisch K. *Semi-smooth Newton methods for state-constrained optimal control problems*. *Syst Control Lett*. 2003;50:221-228.
27. Kanzow C. *Inexact semi-smooth Newton methods for large-scale complementarity problems*. *Optim Methods Softw*. 2004;19:309-325.
28. Hintermüller M, Ito K, Kunisch K. *The primal-dual active set strategy as a semi-smooth Newton method*. *SIAM J Optim*. 2002;13:865-888.
29. Pearson JW, Stoll M, Wathen A. *Preconditioners for state constrained optimal control problems with Moreau-Yosida penalty function*. *Numer Linear Algebra Appl*. 2011;21:81-97.
30. Porcelli M, Simoncini V, Tani M. *Preconditioning of active-set Newton methods for PDE-constrained optimal control problems*. *SIAM J Sci Comput*. 2015;37:S472-S502.
31. Elman H, Silvester D, Wathen A. *Finite Elements and Fast Iterative Solvers*. Second ed. Oxford: Oxford University Press; 2014.
32. Schäfer M, Turek S, Durst F, Krause E, Rannacher R. *Benchmark computations of laminar flow around a cylinder*. *Flow Simulation with High-Performance Computers II*. Braunschweig/Wiesbaden: Springer; 1996:547-566.
33. Elman H, Silvester D, Wathen A. *Finite Elements and Fast Iterative Solvers with Applications in Incompressible Fluid Dynamics*. Numerical Mathematics and Scientific Computation. New York, NY: Oxford University Press; 2005.
34. Tröltzsch F. *Optimal Control of Partial Differential Equations: Theory, Methods and Applications*. Providence, RI: American Mathematical Society; 2010.
35. M. Hinze, *Optimal and Instantaneous Control of the Instationary Navier-Stokes Equations*. Habilitation, TU Berlin; 2000.
36. Dolgov S, Stoll M. *Low-rank solution to an optimization problem constrained by the Navier-Stokes equations*. *SIAM J Sci Comput*. 2017;39:A255-A280. <https://doi.org/10.1137/15M1040414>.
37. Xiu D, Karniadakis GE. *A new stochastic approach to transient heat conduction modeling with uncertainty*. *Int J Heat Mass Transf*. 2003;46:4681-4693.
38. Wiener I. *The homogeneous chaos*. *Am J Math*. 1938;60:897-936.
39. Babuška I, Tempone R, Zouraris G. *Galerkin finite element approximations of stochastic elliptic partial differential equations*. *SIAM J Numer Anal*. 2004;42:800-825.
40. Dolgov S, Khoromskij BN, Litvinenko A, Matthies HG. *Polynomial Chaos Expansion of random coefficients and the solution of stochastic partial differential equations in the Tensor Train format*. *SIAM J. Uncertainty Quantif*. 2015;3:1109-1135. <https://doi.org/10.1137/140972536>.
41. Ballani J, Grasedyck L. *Tree adaptive approximation in the hierarchical tensor format*. *SIAM J Sci Comput*. 2014;36:A1415-A1431. <https://doi.org/10.1137/130926328>.
42. Barcza G, Legeza O, Marti KH, Reiher M. *Quantum-information analysis of electronic states of different molecular structures*. *Phys Rev A*. 2011;83:012508. <https://doi.org/10.1103/PhysRevA.83.012508>.
43. Oseledets IV. *Tensor-train decomposition*. *SIAM J Sci Comput*. 2011;33:2295-2317. <https://doi.org/10.1137/090752286>.
44. Schollwöck U. *The density-matrix renormalization group*. *Rev Mod Phys*. 2005;77:259-315. <https://doi.org/10.1103/RevModPhys.77.259>.
45. Perez-Garcia D, Verstraete F, Wolf MM, Cirac JI. *Matrix product state representations*. *Quant Info Comput*. 2007;7:401-430.
46. White SR. *Density matrix algorithms for quantum renormalization groups*. *Phys Rev B*. 1993;48:10345-10356. <https://doi.org/10.1103/PhysRevB.48.10345>.
47. Holtz S, Rohwedder T, Schneider R. *The alternating linear scheme for tensor optimization in the tensor train format*. *SIAM J Sci Comput*. 2012;34:A683-A713. <https://doi.org/10.1137/100818893>.
48. Jeckelmann E. *Dynamical density matrix renormalization group method*. *Phys Rev B*. 2002;66:045114. <https://doi.org/10.1103/PhysRevB.66.045114>.
49. Dolgov SV, Savostyanov DV. *Alternating minimal energy methods for linear systems in higher dimensions*. *SIAM J Sci Comput*. 2014;36:A2248-A2271. <https://doi.org/10.1137/140953289>.
50. Dolgov SV, Khoromskij BN, Oseledets IV, Savostyanov DV. *Computation of extreme eigenvalues in higher dimensions using block tensor train format*. *Comput Phys Commun*. 2014;185:1207-1216. <https://doi.org/10.1016/j.cpc.2013.12.017>.
51. Johnston H, Liu JG. *Accurate, stable and efficient Navier-Stokes solvers based on explicit treatment of the pressure term*. *J Comput Phys*. 2004;199:221-259.

52. Noack BR, Papas P, Monkewitz PA. *The need for a pressure-term representation in empirical Galerkin models of incompressible shear flows*. *J Fluid Mech*. 2005;523:339-365. <https://doi.org/10.1017/S0022112004002149>.
53. Balajewicz MJ, Dowell EH, Noack BR. *Low-dimensional modelling of high-Reynolds-number shear flows incorporating constraints from the Navier-Stokes equation*. *J Fluid Mech*. 2013;729:285-308. <https://doi.org/10.1017/jfm.2013.278>.
54. Benzi M, Golub GH, Liesen J. *Numerical solution of saddle point problems*. *Acta Numer*. 2005;14:1-137.
55. Rohwedder T, Uschmajew A. *On local convergence of alternating schemes for optimization of convex problems in the tensor train format*. *SIAM J Numer Anal*. 2013;51:1134-1162.
56. Stoll M, Wathen A. *All-at-once solution of time-dependent Stokes control*. *J Comput Phys*. 2013;232:498-515.
57. Cheney W, Kincaid D. *Numerical Mathematics and Computing*. 6th ed. Pacific Grove, CA: Brooks/Cole Publishing Co.; 2007.
58. I. V. Oseledets, S. Dolgov, V. Kazeev, D. Savostyanov, O. Lebedeva, P. Zhlobich, T. Mach, and L. Song, *TT-Toolbox*, 2016. <https://github.com/oseledets/TT-Toolbox>.
59. D. Silvester, A. Bespalov, Q. Liao, and L. Rocchi, *T-IFISS: version 1.1*. <https://doi.org/10.1016/j.camwa.2020.03.005>.
60. S. Dolgov and J. W. Pearson, *Preconditioners and tensor product solvers for optimal control problems from chemotaxis*; 2018. <http://arxiv.org/abs/1806.08539>.

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of this article.

How to cite this article: Benner P, Dolgov S, Onwunta A, Stoll M. Low-rank solution of an optimal control problem constrained by random Navier-Stokes equations. *Int J Numer Meth Fluids*. 2020;92:1653–1678. <https://doi.org/10.1002/flid.4843>