Low-rank solvers for unsteady Stokes-Brinkman optimal control problem with random data

Peter Benner[a], Sergey Dolgov[a], Akwum Onwunta[a,1], Martin Stoll[a]

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

Abstract

We consider the numerical simulation of an optimal control problem constrained by the unsteady Stokes-Brinkman equation involving random data. More precisely, we treat the state, the control, the target (or the desired state), as well as the viscosity, as analytic functions depending on uncertain parameters. This allows for a simultaneous generalized polynomial chaos approximation of these random functions in the stochastic Galerkin finite element method discretization of the model. The discrete problem yields a prohibitively high dimensional saddle point system with Kronecker product structure. We develop a new alternating iterative tensor method for an efficient reduction of this system by the low-rank Tensor Train representation. Besides, we propose and analyze a robust Schur complement-based preconditioner for the solution of the saddle-point system. The performance of our approach is illustrated with extensive numerical experiments based on two- and three-dimensional examples, where the full problem size exceeds one billion degrees of freedom. The developed Tensor Train scheme reduces the solution storage by two–three orders of magnitude, depending on discretization parameters.

Keywords: Stochastic Galerkin system, iterative methods, PDE-constrained optimization, low-rank solution, tensor methods, preconditioning, Schur complement.
AMS: 35R60, 60H15, 60H35, 65N22, 65F10.

1. Introduction

The Brinkman model is a parameter-dependent combination of the Darcy and the Stokes models. It provides a unified approach to model flows of viscous fluids in a cavity and a porous media. As pointed out in [57], in practical applications, the location and number of the Darcy-Stokes interfaces might not be known a priori. Hence, the unified equations represent an advantage over the domain decomposition methods coupling the Darcy and the Stokes equations [9, 2]. The Brinkman model is typically applied in oil reservoir modeling [48] or computational fuel cell dynamics [35, 62].

The study of finite element-based solvers for the Brinkman model has, on the one hand, attracted much attention recently [48, 57, 58, 62]. It is a quite challenging task, essentially due to the high variability in the coefficients of the model, which may take very high or very small values. This feature adversely affects not only the preconditioning of the resulting linear system [57], but also the construction of stable finite element discretizations [40, 62]. On the other hand, the numerical simulation of optimization problems constrained by unsteady Brinkman equations has not yet received adequate attention. Generally speaking, optimization problems constrained by unsteady partial differential equations (PDEs) are a lot more computationally challenging because one needs to solve a system of PDEs coupled globally in time and space, and time-stepping methods quickly reach their limitations due to the enormous demand for storage [46, 55]. Yet,

[1] Corresponding author; this author's research was supported by a research grant from the International Max Planck Research School (IMPRS) for Advanced Methods in Process and System Engineering (Magdeburg).

more challenging than the aforementioned are the optimal control problems constrained by unsteady PDEs involving (countably many) parametric or uncertain inputs. This class of problems arises because the input parameters of the model, such as the viscosity or initial condition may be affected by uncertainty due, for example, to measurement errors, limited data or intrinsic variability in physical phenomenon being modeled. Hence, a convenient way to characterize the uncertainty in the problem consists in incorporating the uncertain parameters as random variables or space- and/or time-varying random fields.

In fluid mechanics, for example, the cost functionals in optimization problems are expressed in terms of flow variables (such as velocity, pressure, temperature, etc.), whereas constraints are represented by the PDE (advection-diffusion, Stokes or Stokes-Brinkman or Navier-Stokes equations, etc.) describing the flow and, whenever necessary, by topological constraints on the shape of the domain. In a broad variety of applications, the design of devices which are able to reduce drag forces, dissipations or stresses greatly enhances the efficiency of a system. For example, the reduction of vorticity and stresses in biomedical devices, and the compliance minimization in cantilevers or membranes, represent instances in which optimization techniques are called into play. In particular, in biomedical engineering, Stokes-Brinkman control could be used to model the reduction of vorticity of blood flow through intracranial aneurysms [54]. However, the value of the fluid viscosity ν may not be known precisely. Instead of guessing a value, one can model ν as a random variable defined on some complete probability space. This could be interpreted as a scenario where the volume of blood moving through the intracranial aneurysms is uncertain due to measurement error in ν or probably some other factors [51]. As aptly pointed out in a related study in the framework of a deterministic control problem [39], efficient procedures for the numerical solution of the resulting stochastic control problem is required because the model is expensive to solve, especially when solutions need to capture fine details (such as velocity and thermal boundary layers, etc.); moreover, the finite element assembling discretization procedures for the spatial domain could become expensive. The introduction of a suitable low-rank numerical scheme is thus instrumental to reduce both the storage requirements and the computational complexity. With a view to achieving these goals in this contribution, we discuss a low-rank tensor-based technique for solving high dimensional tensor product linear systems resulting from the discretization of a Stokes-Brinkman optimal control problem with stochastic inputs (SOCP).

For the numerical simulation of the SOCP, we assume that the state, the control and the target are analytic functions depending on some uncertain parameters. This allows for a simultaneous generalized polynomial chaos (PCE) approximation of these random functions [15, 16, 38, 50, 63] in the stochastic Galerkin finite element method (SGFEM) discretization of the model. However, these problems often lead to prohibitively high dimensional linear systems with Kronecker product structure.

To reduce the computational complexity, we impose the Kronecker product structure on the solution as well. More precisely, we seek an approximate solution in a low-rank tensor product representation, namely, the Tensor Train decomposition [42], also known as the Matrix Product States [29]. The tensor decomposition concept is similar to low-rank model reduction techniques, for example, the Proper Orthogonal Decomposition (POD) [34]. However, the POD solves the full problem in order to derive a reduced model. For really large-scale systems this is not feasible. Tensor methods aim to construct directly the reduced solution without a priori information. One of the most powerful tensor-based algorithms that can effectively accomplish this task is the alternating iterative method [22, 53, 60]. However, existing alternating solvers for linear systems require a positive definite matrix. Another novel contributions of this paper are the extension and adaptation of these algorithms to the saddle-points optimality system. We refer to [18, 17] for a more detailed overview of tensor methods.

This paper is structured as follows. In Section 2, we present the deterministic Stokes-Brinkman model. Section 3 introduces the Stokes-Brinkman optimal control problem with uncertain inputs and gives an overview of the SGFEM. Besides, it establishes the Kronecker-product structure of the discrete problem. Section 4 presents and analyzes our preconditioners for the corresponding saddle-point linear systems. In Section 5, we introduce the Tensor Train decomposition and alternating tensor algorithms, adjust them to the particular structure of the inverse problem and the Stokes-Brinkman model and discuss some implementation issues. Section 6 contains numerical results obtained for two- and three-dimensional examples using our approach. Finally, Section 7 gives a conclusion and outlines future research goals.

2. Deterministic Brinkman model

Let $\mathcal{D} \subset \mathbb{R}^d$ with $d \in \{1, 2, 3\}$, be a bounded open set with Lipschitz continuous simply connected boundary $\partial\mathcal{D}$. Herein, the spatial domain $\mathcal{D}$ consists of two parts, namely, a porous medium $\mathcal{D}_p$ and a viscous flow medium $\mathcal{D}_s$. That is, $\mathcal{D} = \mathcal{D}_p \cup \mathcal{D}_s$. Moreover, denote by $\mathcal{Q}$ the space-time cylinder $\mathcal{D} \times [0, T]$ and $\mathcal{T} = (0, T]$. The generalized unsteady Brinkman problem reads

$$\begin{cases} \dfrac{\partial v(\mathbf{x}, t)}{\partial t} - \nu\Delta v(\mathbf{x}, t) + K_0(\mathbf{x})v(\mathbf{x}, t) + \nabla p(\mathbf{x}, t) = u(\mathbf{x}, t), \text{ in } \mathcal{Q}, \\ -\nabla \cdot v(\mathbf{x}, t) = 0, \text{ on } \mathcal{Q}, \\ v(\mathbf{x}, t) = h(\mathbf{x}, t), \text{ on } \partial\mathcal{D} \times \mathcal{T}, \\ v(\mathbf{x}, 0) = v_0, \text{ in } \mathcal{D}, \end{cases} \tag{1}$$

where v and p are, respectively, the fluid velocity and the fluid pressure, and h is the boundary condition. The parameter ν represents the fluid viscosity. Moreover, K_0 is the *inverse permeability tensor* of the medium. We assume here that $K_0 \in L^2(\mathcal{D}) \cap L^\infty(\mathcal{D})$ and that the source term $u \in L^2(\mathcal{D})$. The challenge of this problem is that the coefficient K_0 takes two extreme values: it is very small in the viscous flow medium $\mathcal{D}_s$ so that the PDE behaves like the unsteady Stokes flow, and very big in the porous medium $\mathcal{D}_p$ in which case the PDE behaves like the unsteady Darcy equations.

In this paper, we denote by $H^k(\mathcal{D})$ the Sobolev space of functions on $\mathcal{D}$ whose derivatives up to order k are square-integrable. $H^k_0(\mathcal{D})$ denotes the closure in $H^k(\mathcal{D})$ of the set of finitely differentiable functions with compact support in $\mathcal{D}$. For some space $\mathcal{X}$ of functions on $\mathcal{D}$, let $L^2(0, T; \mathcal{X}) = L^2(0, T) \otimes \mathcal{X}$. The variational formulation of the Brinkman model (1) can thus be written in following form: find $v \in L^2(0, T; H^1_0(\mathcal{D}))$, $p \in L^2(0, T; L^2(\mathcal{D}))$ and $\partial_t v \in L^2(0, T; H^{-1}(\mathcal{D}))$, such that $v|_{t=0} = v_0$ and $a.e$ on $[0, T]$

$$\begin{cases} (\partial_t v(t), w) + \mathcal{B}(v(t), w) - \mathcal{C}(p(t), \text{div } w) = & (u, w), \ \forall w \in L^2(0, T; H^1_0(\mathcal{D})) \\ \mathcal{C}(\text{div } v, q) = & 0, \ \ \forall q \in L^2(0, T; L^2(\mathcal{D})), \end{cases}$$

where

$$\mathcal{B}(v(t), w) = (\nu\nabla v(t), \nabla w) + (K_0(\mathbf{x})v, w),$$
$$\mathcal{C}(p(t), \text{div } w) = (p(t), \nabla \cdot w),$$

and $(\cdot, \cdot)$ represents the L^2 inner product of a pair of functions on $\mathcal{D}$.

For a mixed finite element discretization of the Brinkman problem [40, 54, 57, 62] in the primal variables v and p, let $V_h \subset L^2(0, T; H^1_0(\mathcal{D}))$ and $W_h \subset L^2(0, T; L^2(\mathcal{D}))$ be finite element spaces with stable elements (i.e. elements that satisfy the *inf-sup* condition, e.g. *mini elements* as discussed in [54]) such that $V_h = \text{span}\{\phi_1, \ldots, \phi_{J_v}\}$ and $W_h = \text{span}\{\varphi_1, \ldots, \varphi_{J_p}\}$. Performing a Galerkin projection on V_h and W_h and using implicit Euler for the temporal discretization, while taking into account the boundary conditions, leads to the following equations:

$$\begin{cases} \dfrac{Mv_i - Mv_{i-1}}{\tau} + (\nu K + M_k)v_i + B^T p_i = Mu_i + g_i, \\ Bv_i = 0, \end{cases} \tag{2}$$

where τ is the time step size,

$$\begin{aligned} B &= \left[-\int_{\mathcal{D}} \varphi_k \nabla \cdot \phi_{k'}\right] && \text{is the discrete divergence operator,} \\ K &= \left[\int_{\mathcal{D}} \nabla\phi_k : \nabla\phi_{k'}\right] && \text{is the matrix representing the vector Laplacian operator,} \\ M &= \left[\int_{\mathcal{D}} \phi_k\phi_{k'}\right] && \text{is the mass matrix and} \\ M_k &= \left[\int_{\mathcal{D}} K_0\phi_k\phi_{k'}\right] && \text{is the mass matrix with the inverse permeability coefficient } K_0(\mathbf{x}). \end{aligned} \tag{3}$$

Remark 1. In the special case where $M_k = 0$ in (2), we get the unsteady Stokes problem.

3. Brinkman optimal control problem with random data

Suppose now that, even though the fluid viscosity ν is time-independent and spatially constant but that its value is not known precisely. Instead of guessing a value, we can model ν as a random variable defined on the complete probability space $(\Omega, \mathcal{F}, \mathbb{P})$. This could be interpreted as a scenario where the volume of fluid moving into a channel is uncertain due to measurement error in ν or probably some other factors [51]. Here, Ω is a sample space of events whereas, $\mathcal{F}$ denotes a σ-algebra on Ω and is endowed with an appropriate probability measure $\mathbb{P}$. The corresponding Brinkman velocity and pressure are consequently also random and the numerical solution of the associated SOCP is far more challenging. More precisely, the SOCP which we will solve in the rest of this paper consists in minimizing the cost functional of tracking-type

$$\mathcal{J} = \frac{1}{2}||v - \bar{v}||^2_{L^2(0,T;\mathcal{D})\otimes L^2(\Omega)} + \frac{\alpha}{2}||\mathrm{std}(v)||^2_{L^2(0,T;\mathcal{D})} + \frac{\beta}{2}||u||^2_{L^2(0,T;\mathcal{D})\otimes L^2(\Omega)} \tag{4}$$

subject, $\mathbb{P}$-almost surely, to the state equations[2]

$$\begin{cases} \dfrac{\partial v(\mathbf{x}, t, \omega)}{\partial t} - \nu(\omega)\Delta v(\mathbf{x}, t, \omega) + K_0(\mathbf{x})v(\mathbf{x}, t, \omega) + \nabla p(\mathbf{x}, t, \omega) = u(\mathbf{x}, t, \omega), \text{ in } \mathcal{Q} \times \Omega, \\ \hfill -\nabla \cdot v(\mathbf{x}, t, \omega) = 0, \text{ on } \mathcal{Q} \times \Omega, \\ \hfill v(\mathbf{x}, t, \omega) = h, \text{ on } \partial\mathcal{D} \times \mathcal{T} \times \Omega, \\ \hfill v(\mathbf{x}, 0, \omega) = v_0, \text{ in } \mathcal{D} \times \Omega, \end{cases}$$

where $v, \bar{v}, p : \mathcal{D} \times \mathcal{T} \times \Omega \to \mathbb{R}$ are random fields [5] representing the state (velocity), the target (or the desired state) and the pressure. The forcing term on the right hand side $u : \mathcal{D} \times \mathcal{T} \times \Omega \to \mathbb{R}$ denotes a random control function. Moreover, the positive constant β represents the parameter for the penalization of the norm of the control u, whereas α penalizes the standard deviation $\mathrm{std}(v)$ of the state v. Here, we have used the notation $L^2(\Omega) := L^2(\Omega, \mathcal{F}, \mathbb{P})$. The viscosity ν in the state equations is modeled as

$$\nu(\omega) = \nu_0 + \nu_1\xi(\omega), \qquad \nu_0, \nu_1 \in \mathbb{R}^+, \tag{5}$$

where ξ is a uniformly distributed random variable with $\xi \sim \mathcal{U}(-1, 1)$. Furthermore, we assume that the control and the target satisfy

$$u, \bar{v} \in L^2(\mathcal{D}) \otimes L^2(\mathcal{T}) \otimes L^2(\Omega), \tag{6}$$

and that, for some $\nu_{\min}, \nu_{\max} \in \mathbb{R}^+$ satisfying $0 < \nu_{\min} < \nu_{\max} < +\infty$, we have

$$\mathbb{P}\left(\omega \in \Omega : \nu(\omega) \in [\nu_{\min}, \nu_{\max}]\right) = 1. \tag{7}$$

3.1. A fully discrete problem

Two standard methods are used to discretize the optimal control problem introduced above - we can either discretize the model first and then optimize the discrete system (DTO method), or alternatively optimize first before discretizing the resulting optimality system (OTD method). The commutativity of DTO and OTD methods when applied to optimal control problems constrained by PDEs has been a subject of debate in recent times (see [36] for an overview). In what follows, we will adopt the DTO strategy because, for the optimal control problem considered in this paper, it leads to a symmetric saddle point linear system which fits in nicely with our preconditioning strategies.

[2] In this contribution, we do not consider the case of state- or control- or mixed control-state-constrained problems [21, 45, 49]. These problems can be tackled via the use of, for instance, semi-smooth Newton algorithms [20, 25, 27].

Since our optimal control problem contains random coefficients, the stochastic discretization could be effected using either a projection-based method (e.g. stochastic Galerkin method in [50]) or a sampling method (e.g. stochastic collocation method in [52]). Due to its high convergence rate, the former is our preferred method in this paper. In order to use this method, we first assume that the pressure p, the state v, the control u and the target $\bar{v}$ are analytic functions depending on the uncertain parameters. This allows for a simultaneous generalized polynomial chaos (PCE) approximation of these random functions [15, 50, 5]. Of course, $\bar{v}$ can equally be modeled deterministically. Together with the finite element method, the PCE yields an SGFEM for discretizing both the spatial and stochastic domains. More precisely, p, u, v, and $\bar{v}$ admit the following respective representations

$$\begin{aligned} p(\mathbf{x},t,\omega) &= \sum_{k=1}^{J_p}\sum_{j=1}^{P} p_{kj}(t)\varphi_k(\mathbf{x})\psi_j(\xi(\omega)), \\ u(\mathbf{x},t,\omega) &= \sum_{k=1}^{J_v}\sum_{j=1}^{P} u_{kj}(t)\phi_k(\mathbf{x})\psi_j(\xi(\omega)), \\ v(\mathbf{x},t,\omega) &= \sum_{k=1}^{J_v}\sum_{j=1}^{P} v_{kj}(t)\phi_k(\mathbf{x})\psi_j(\xi(\omega)), \\ \bar{v}(\mathbf{x},t,\omega) &= \sum_{k=1}^{J_v}\sum_{j=1}^{P} \bar{v}_{kj}(t)\phi_k(\mathbf{x})\psi_j(\xi(\omega)), \end{aligned} \tag{8}$$

where $\{\psi_j\}_{j=1}^P$ are univariate orthogonal polynomials of order $P-1$ satisfying

$$\langle\psi_1(\xi)\rangle = 1, \;\; \langle\psi_j(\xi)\rangle = 0, \; j > 1, \;\; \langle\psi_j(\xi)\psi_k(\xi)\rangle = \left\langle\psi_j^2(\xi)\right\rangle\delta_{jk}, \tag{9}$$

with

$$\langle\psi_j(\xi)\rangle = \int_{\omega\in\Omega}\psi_j(\xi(\omega))\, d\mathbb{P}(\omega) = \int_{\xi\in\Gamma}\psi_j(\xi)\rho(\xi)\, d\xi, \tag{10}$$

where ρ is the density of the random variable ξ and Γ is the support of ρ.

In spirit of [5, 55], we apply to the cost functional the trapezoidal rule for temporal discretization, and the *mini* finite elements [54], together with Legendre polynomial chaos in the SGFEM for spatial and stochastic discretizations [50], to get the following

$$\mathcal{J}(\mathbf{y},\mathbf{u}) := \frac{\tau}{2}(\mathbf{y}-\bar{\mathbf{y}})^T\boldsymbol{M}_a(\mathbf{y}-\bar{\mathbf{y}}) + \frac{\tau\alpha}{2}\mathbf{y}^T\boldsymbol{M}_b\mathbf{y} + \frac{\tau\beta}{2}\mathbf{u}^T\boldsymbol{M}_2\mathbf{u}, \tag{11}$$

where $\mathbf{y}^\top = \left[\mathbf{v}_1^\top, \mathbf{p}_1^\top, \ldots, \mathbf{v}_{n_t}^\top, \mathbf{p}_{n_t}^\top\right] \in \mathbb{R}^{JPn_t}$, $J := J_v + J_p$, and $\mathbf{u}^\top = \left[\mathbf{u}_1^\top, \ldots, \mathbf{u}_{n_t}^\top\right]$ denote the long vectors of all time snapshots of the state and control, respectively,

$$\begin{cases} \boldsymbol{M}_a = \mathrm{blkdiag}\left(\frac{1}{2}\mathcal{M}, 0, \mathcal{M}, 0, \ldots, \mathcal{M}, 0, \frac{1}{2}\mathcal{M}, 0\right), & \mathcal{M} := G_0 \otimes M, \\ \boldsymbol{M}_b = \mathrm{blkdiag}\left(\frac{1}{2}\mathcal{M}_t, 0, \mathcal{M}_t, 0, \ldots, \mathcal{M}_t, 0, \frac{1}{2}\mathcal{M}_t, 0\right), & \mathcal{M}_t := H_0 \otimes M, \\ \boldsymbol{M}_2 = \mathrm{blkdiag}\left(\frac{1}{2}\mathcal{M}, \mathcal{M}, \ldots, \mathcal{M}, \frac{1}{2}\mathcal{M}\right), \end{cases} \tag{12}$$

with M the finite element mass matrix, and

$$\begin{cases} G_0 = \mathrm{diag}\left(\left\langle\psi_1^2(\xi)\right\rangle, \left\langle\psi_2^2(\xi)\right\rangle, \ldots, \left\langle\psi_P^2(\xi)\right\rangle\right), \\ H_0 = \mathrm{diag}\left(0, \left\langle\psi_2^2(\xi)\right\rangle, \ldots, \left\langle\psi_P^2(\xi)\right\rangle\right), \end{cases} \tag{13}$$

where the Kronecker product $\otimes$ is meant in the usual sense, $A \otimes B = [A_{ij}B]$.

For an all-at-once discretization of the state equation, we use the implicit Euler together with SGFEM to get

$$\boldsymbol{K}\mathbf{y} - \boldsymbol{N}\mathbf{u} = \mathbf{g}, \tag{14}$$

where

$$\boldsymbol{K} = \begin{bmatrix} \bar{\mathcal{L}} & & & \\ -\bar{\mathcal{M}} & \bar{\mathcal{L}} & & \\ & \ddots & \ddots & \\ & & -\bar{\mathcal{M}} & \bar{\mathcal{L}} \end{bmatrix}, \ \boldsymbol{N} = \begin{bmatrix} \mathcal{N} & & & \\ & \mathcal{N} & & \\ & & \ddots & \\ & & & \mathcal{N} \end{bmatrix}, \ \mathbf{g} = \begin{bmatrix} \bar{\mathcal{M}}\mathbf{y}_0 + \mathbf{g}_1^0 \\ \mathbf{g}_2^0 \\ \vdots \\ \mathbf{g}_{n_t}^0 \end{bmatrix},$$

with

$$\mathcal{N} = G_0 \otimes N, \quad N = \begin{bmatrix} M \\ 0 \end{bmatrix}, \quad \bar{\mathcal{M}} = G_0 \otimes \tau^{-1}\bar{M}, \quad \bar{M} = \begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix}, \tag{15}$$

and, in the notation of [51],

$$\bar{\mathcal{L}} = \begin{bmatrix} \mathcal{A} & \mathcal{B}^T \\ \mathcal{B} & 0 \end{bmatrix} \tag{16}$$

represents an instance of the time-dependent Brinkman problem with

$$\mathcal{A} = G_0 \otimes A + G_1 \otimes \nu_1 K, \quad A = \tau^{-1}M + \nu_0 K + M_k, \qquad \mathcal{B} = G_0 \otimes B, \tag{17}$$

and $G_1(j,j') = \langle \xi \psi_j(\xi)\psi_{j'}(\xi)\rangle$. Note that since we are using Legendre polynomials for SGFEM discretization, G_0 is a diagonal matrix whereas G_1 is a tridiagonal matrix with zeros on the main diagonal (see e.g., [50, 51]). This implies that the matrix $\mathcal{B}$ in (17) is block-diagonal. Furthermore, since the matrices K, M and M_k are positive definite, we know that $\mathcal{A}$ in (17) is sparse and positive definite. However, $\bar{\mathcal{L}}$ is an indefinite block sparse matrix with sparse blocks.

Later it will be convenient to work with the Kronecker product representations of the system matrices. To this end, we introduce the identity matrix $I_{n_t} \in \mathbb{R}^{n_t \times n_t}$, as well as the matrix

$$C = \begin{bmatrix} 0 & & & \\ -1 & 0 & & \\ & \ddots & \ddots & \\ & & -1 & 0 \end{bmatrix}, \tag{18}$$

and observe then that

$$\boldsymbol{K} = I_{n_t} \otimes G_0 \otimes \begin{bmatrix} A & B^\top \\ B & 0 \end{bmatrix} + I_{n_t} \otimes G_1 \otimes \begin{bmatrix} \nu_1 K & 0 \\ 0 & 0 \end{bmatrix} + C \otimes G_0 \otimes \begin{bmatrix} \tau^{-1}M & 0 \\ 0 & 0 \end{bmatrix}, \tag{19}$$

and

$$\boldsymbol{N} = I_{n_t} \otimes G_0 \otimes N. \tag{20}$$

The structure of the right-hand side is problem-dependent. However, in our experiments we will use $\mathbf{y}_0 = 0$ and a static deterministic $\mathbf{g}^0$ coming from Dirichlet boundary conditions, such that $\mathbf{g} = \mathbf{g}^0 = \mathbf{e} \otimes \mathbf{e}_1 \otimes \begin{bmatrix} \mathbf{g}_v^0 \\ \mathbf{g}_p^0 \end{bmatrix}$, where $\mathbf{e}$ is the vector of all ones, and $\mathbf{e}_1$ is the first unit vector.

Now, note from (11) and (14) that the discrete Lagrangian functional of the SOCP is given by

$$\mathfrak{L} := \frac{\tau}{2}(\mathbf{y} - \bar{\mathbf{y}})^T \boldsymbol{M}_a (\mathbf{y} - \bar{\mathbf{y}}) + \frac{\tau\alpha}{2}\mathbf{y}^T \boldsymbol{M}_b \mathbf{y} + \frac{\tau\beta}{2}\mathbf{u}^T \boldsymbol{M}_2 \mathbf{u} + \boldsymbol{\lambda}^T(-\boldsymbol{K}\mathbf{y} + \boldsymbol{N}\mathbf{u} + \mathbf{g}),$$

where $\boldsymbol{\lambda}$ is the Lagrange multiplier. Hence, applying the first order conditions to $\mathfrak{L}$ yields the Karush-Kuhn-Tucker (KKT) system

$$\underbrace{\begin{bmatrix} \tau \boldsymbol{M}_1 & 0 & -\boldsymbol{K}^T \\ 0 & \beta\tau \boldsymbol{M}_2 & \boldsymbol{N}^T \\ -\boldsymbol{K} & \boldsymbol{N} & 0 \end{bmatrix}}_{:=\mathfrak{A}} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{0} \\ \mathbf{g} \end{bmatrix}, \tag{21}$$

where $\mathbf{b}_1 = \tau \boldsymbol{M}_a \bar{\mathbf{y}}$, and

$$\boldsymbol{M}_1 = \boldsymbol{M}_a + \alpha \boldsymbol{M}_b = D \otimes G_\alpha \otimes \bar{M}, \qquad \boldsymbol{M}_2 = D \otimes \mathcal{M} = D \otimes G_0 \otimes M, \tag{22}$$

$$D = \operatorname{diag}\left(\frac{1}{2}, 1 \ldots, 1, \frac{1}{2}\right) \in \mathbb{R}^{n_t \times n_t}, \qquad G_\alpha := G_0 + \alpha H_0. \tag{23}$$

We note here that if the desired state is also static and deterministic, then one gets $\bar{\mathbf{y}} = \mathbf{e} \otimes \mathbf{e}_1 \otimes \begin{bmatrix} \bar{\mathbf{v}} \\ 0 \end{bmatrix}$.

4. Preconditioning

The KKT coefficient matrix $\mathfrak{A}$ in (21) is usually ill-conditioned and thus requires a suitable preconditioner to solve (21) efficiently. A block-diagonal preconditioner, discussed in the framework of deterministic unsteady Stokes control problem [56], is written in the form $\boldsymbol{P}_1 = \operatorname{blockdiag}(\tilde{\boldsymbol{M}}_1, \beta \boldsymbol{M}_2, \tilde{\boldsymbol{S}}_1)$, where $\tilde{\boldsymbol{S}}_1 = \frac{1}{\tau}\left(\boldsymbol{K} + \boldsymbol{M}_s\right) \tilde{\boldsymbol{M}}_1^{-1} \left(\boldsymbol{K}^T + \boldsymbol{M}_s\right)^T$ is the approximate Schur complement, and $\tilde{\boldsymbol{M}}_1$ is some perturbation to $\boldsymbol{M}_1$, since the latter is rank-deficient. Here, the matrix $\boldsymbol{M}_s$ is determined via a 'matching' argument. In particular, [56] suggest the following augmentation,

$$\tilde{\boldsymbol{M}}_1 = \begin{bmatrix} D \otimes G_\alpha \otimes M & \\ & D \otimes G_\alpha \otimes \left(\|M\|_2^2 \tau \beta\right) I \end{bmatrix},$$

where I is the identity of the size of the pressure grid. However, this approach is tricky. For example, it may be quite suitable for preconditioning of MINRES, which works with the $\boldsymbol{P}_1^{-1}$-scalar product, but perform poorly in the Flexible GMRES, if we are to apply $\boldsymbol{P}_1^{-1}$ approximately. Besides, it is not obvious how to generalize it to the case when $\boldsymbol{M}_1$ is *numerically* rank-deficient, i.e. its eigenvalues form a gradually decaying sequence instead of two distinct clusters. This will occur in the low-rank tensor methods; consequently, instead of $\boldsymbol{M}_1$, we will work with its Galerkin projection in the sequel. More specifically, we proceed next to Section 4.1 to propose another preconditioner which circumvents this deficiency and yields faster convergence even with the original sparse $\boldsymbol{M}_1$.

4.1. A block-triangular preconditioner

Our point of departure is to replace the KKT coefficient matrix $\mathfrak{A}$ in (21) by $\tilde{\mathfrak{A}}$ given by

$$\tilde{\mathfrak{A}} := \mathfrak{A}\boldsymbol{\rho} = \begin{bmatrix} -\boldsymbol{K}^T & 0 & \tau \boldsymbol{M}_1 \\ \boldsymbol{N}^T & \beta\tau \boldsymbol{M}_2 & 0 \\ 0 & \boldsymbol{N} & -\boldsymbol{K} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi} & \boldsymbol{\Upsilon} \\ \boldsymbol{\Psi} & -\boldsymbol{K} \end{bmatrix},$$

where

$$\boldsymbol{\rho} = \begin{bmatrix} 0 & 0 & \boldsymbol{I} \\ 0 & \boldsymbol{I} & 0 \\ \boldsymbol{I} & 0 & 0 \end{bmatrix}, \quad \boldsymbol{\Phi} = \begin{bmatrix} -\boldsymbol{K}^T & 0 \\ \boldsymbol{N}^T & \beta\tau \boldsymbol{M}_2 \end{bmatrix}, \quad \boldsymbol{\Upsilon} = \begin{bmatrix} \tau \boldsymbol{M}_1 \\ 0 \end{bmatrix}, \quad \boldsymbol{\Psi} = \begin{bmatrix} 0 \\ \boldsymbol{N} \end{bmatrix}^T.$$

Note that the matrix $\boldsymbol{\rho}$ swaps the first and third columns of $\mathfrak{A}$ in the product $\mathfrak{A}\boldsymbol{\rho}$; it swaps the first and third rows of $\mathfrak{A}$ in the product $\boldsymbol{\rho}\mathfrak{A}$. Next, observe also that we can factorize the matrix $\tilde{\mathfrak{A}}$ as follows

$$\left[\begin{array}{cc} \boldsymbol{\Phi} & \boldsymbol{\Upsilon} \\ \boldsymbol{\Psi} & -\boldsymbol{K} \end{array}\right] = \left[\begin{array}{cc} \boldsymbol{I} & 0 \\ \boldsymbol{\Psi}\boldsymbol{\Phi}^{-1} & \boldsymbol{I} \end{array}\right] \left[\begin{array}{cc} \boldsymbol{\Phi} & \boldsymbol{\Upsilon} \\ 0 & -\boldsymbol{S}_2 \end{array}\right],$$

where

$$\boldsymbol{\Phi}^{-1} = \left[\begin{array}{cc} -\boldsymbol{K}^{-T} & 0 \\ \frac{1}{\tau\beta}\boldsymbol{M}_2^{-1}\boldsymbol{N}^T\boldsymbol{K}^{-T} & \frac{1}{\tau\beta}\boldsymbol{M}_2^{-1} \end{array}\right], \tag{24}$$

and $\boldsymbol{S}_2 = \boldsymbol{K} + \boldsymbol{\Psi}\boldsymbol{\Phi}^{-1}\boldsymbol{\Upsilon} = \boldsymbol{K} + \frac{1}{\beta}\boldsymbol{N}\boldsymbol{M}_2^{-1}\boldsymbol{N}^T\boldsymbol{K}^{-T}\boldsymbol{M}_1$. But then, from (15), (20) and (22), we obtain

$$\boldsymbol{N}\boldsymbol{M}_2^{-1}\boldsymbol{N}^T = D^{-1} \otimes G_0 \otimes \bar{M} = D^{-1} \otimes \left[\begin{array}{cc} \tau\mathcal{M} & 0 \\ 0 & 0 \end{array}\right] =: \boldsymbol{M}_{-1}. \tag{25}$$

Therefore,

$$\boldsymbol{S}_2 = \boldsymbol{K} + \boldsymbol{\Psi}\boldsymbol{\Phi}^{-1}\boldsymbol{\Upsilon} = \boldsymbol{K} + \frac{1}{\beta}\boldsymbol{M}_{-1}\boldsymbol{K}^{-T}\boldsymbol{M}_1. \tag{26}$$

We propose to right-precondition $\tilde{\mathfrak{A}}$ with the matrix

$$\boldsymbol{P}_D = \left[\begin{array}{cc} \boldsymbol{\Phi} & \boldsymbol{\Upsilon} \\ 0 & -\boldsymbol{S}_2 \end{array}\right]. \tag{27}$$

This implies that

$$\tilde{\mathfrak{A}}\boldsymbol{P}_D^{-1} = \mathfrak{A}\boldsymbol{\rho}\boldsymbol{P}_D^{-1} = \mathfrak{A}\boldsymbol{P}_2^{-1} = \left[\begin{array}{cc} \boldsymbol{I} & 0 \\ \boldsymbol{\Psi}\boldsymbol{\Phi}^{-1} & \boldsymbol{I} \end{array}\right], \tag{28}$$

where the right preconditioner $\boldsymbol{P}_2$ for the original KKT matrix $\mathfrak{A}$ satisfies

$$\boldsymbol{P}_2^{-1} = \boldsymbol{\rho}\boldsymbol{P}_D^{-1} = \left[\begin{array}{ccc} 0 & 0 & -\boldsymbol{S}_2^{-1} \\ \frac{1}{\beta\tau}\boldsymbol{M}_2^{-1}\boldsymbol{N}^T\boldsymbol{K}^{-T} & \frac{1}{\beta\tau}\boldsymbol{M}_2^{-1} & \frac{1}{\beta}\boldsymbol{M}_2^{-1}\boldsymbol{N}^T\boldsymbol{K}^{-T}\boldsymbol{M}_1\boldsymbol{S}_2^{-1} \\ -\boldsymbol{K}^{-T} & 0 & -\boldsymbol{K}^{-T}\tau\boldsymbol{M}_1\boldsymbol{S}_2^{-1} \end{array}\right]. \tag{29}$$

It can be noticed that (28) immediately implies $(\mathfrak{A}\boldsymbol{P}_2^{-1} - I)^2 = 0$; hence, such Krylov solvers as the generalized minimal residual (GMRES) method will converge in two iterations if $\boldsymbol{P}_2^{-1}$ is applied exactly, see e.g. [13, Section 8.1].

The seeming complicated structure of (29) notwithstanding, matrix-vector product with $\boldsymbol{P}_2^{-1}$ can be implemented fairly easily. For instance, suppose now that we want to solve $\mathbf{x} = \boldsymbol{P}_2^{-1}\mathbf{y}$, where $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3]^T$, $\mathbf{y} = [\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3]^T$. Then, it can easily be shown that an efficient way to implement the matrix-vector product is

$$\begin{cases} \mathbf{x}_1 = -\boldsymbol{S}_2^{-1}\mathbf{y}_3 \\ \mathbf{x}_3 = -\boldsymbol{K}^{-T}(\mathbf{y}_1 - \tau\boldsymbol{M}_1\mathbf{x}_1) \\ \mathbf{x}_2 = \tau^{-1}\beta^{-1}\boldsymbol{M}_2^{-1}(\mathbf{y}_2 - \boldsymbol{N}^T\mathbf{x}_3). \end{cases} \tag{30}$$

Next, following a state-of-the-art preconditioning strategy in [47], we approximate the Schur complement $\boldsymbol{S}_2$ in (26) with a matrix of the form

$$\begin{aligned} \tilde{\boldsymbol{S}}_2 &= (\boldsymbol{K} + \boldsymbol{M}_l)\,\boldsymbol{K}^{-T}\left(\boldsymbol{K}^T + \boldsymbol{M}_r\right). \\ &= \boldsymbol{K} + \boldsymbol{M}_l\boldsymbol{K}^{-T}\boldsymbol{M}_r + \boldsymbol{M}_l + \boldsymbol{K}\boldsymbol{K}^{-T}\boldsymbol{M}_r, \end{aligned} \tag{31}$$

where $\boldsymbol{M}_l$ and $\boldsymbol{M}_r$ are determined using also a 'matching' argument between the exact Schur complement $\boldsymbol{S}_2$ and the approximation $\tilde{\boldsymbol{S}}_2$. More precisely, we ignore the last two terms in (31) and match the first and second terms with those in (26) to get $\boldsymbol{M}_r = \beta^{-1/2}\boldsymbol{M}_1$, and $\boldsymbol{M}_l = \beta^{-1/2}\boldsymbol{M}_{-1}$, where $\boldsymbol{M}_1$ and $\boldsymbol{M}_{-1}$ are as defined, respectively, in (22) and (25). Hence, we have

$$\tilde{\boldsymbol{S}}_2 = \left(\boldsymbol{K} + \frac{1}{\sqrt{\beta}}\boldsymbol{M}_{-1}\right)\boldsymbol{K}^{-T}\left(\boldsymbol{K}^T + \frac{1}{\sqrt{\beta}}\boldsymbol{M}_1\right). \tag{32}$$

For matrix-vector products, the factors $\left(\boldsymbol{K} + \frac{1}{\sqrt{\beta}}\boldsymbol{M}_{-1}\right)$ and $\left(\boldsymbol{K}^T + \frac{1}{\sqrt{\beta}}\boldsymbol{M}_1\right)$ can be kept as sums of four Kronecker products, with the first three coming from $\boldsymbol{K}$ in (19), and the fourth corresponding to $\boldsymbol{M}_{-1}$ in (25) and $\boldsymbol{M}_1$ in (22), respectively. However, our ultimate goal is to apply $\tilde{\boldsymbol{S}}_2^{-1}$, where it appears that solving a linear system with exact factors is difficult. As a result, we instead approximate them by one Kronecker-product term: we approximate $\boldsymbol{K}$ by the first term from (19), whereas we set $\boldsymbol{M}_1 \approx I_{n_t} \otimes (1+\alpha)G_0 \otimes \bar{M}$ and $\boldsymbol{M}_{-1} \approx I_{n_t} \otimes G_0 \otimes \bar{M}$; therefore,

$$\left(\boldsymbol{K} + \frac{1}{\sqrt{\beta}}\boldsymbol{M}_{\mathtt{i}}\right) \approx I_{n_t} \otimes G_0 \otimes \begin{bmatrix} A + \eta_{\mathtt{i}} M & B^\top \\ B & 0 \end{bmatrix}, \tag{33}$$

where $\mathtt{i} \in \{-1, 1\}$, and $\eta_{-1} = 1/\sqrt{\beta}$, $\eta_1 = (1+\alpha)/\sqrt{\beta}$. Inside alternating tensor methods (cf. Section 5.5), the matrix $I_{n_t} \otimes G_0$ will be further reduced, but the concept of the one-term preconditioner remains the same.

4.2. Preconditioning of the forward Stokes-Brinkman problem

In linear systems of the form (33), I_{n_t} and G_0 can be inverted straightforwardly, while the spatial matrix may require a special treatment. To this end, we can use either the GMRES or the inexact Uzawa algorithm (see e.g. [56]), together with the block-triangular preconditioner

$$P_s = \begin{bmatrix} \tilde{A} & 0 \\ B & -S_0 \end{bmatrix}, \tag{34}$$

where $S_0 = B\tilde{A}^{-1}B^\top$ is the Schur complement and $\tilde{A} = \nu_0 K + M_k + (\tau^{-1} + \eta)M$ with $\eta = \frac{1}{\sqrt{\beta}}$ or $\eta = \frac{1+\alpha}{\sqrt{\beta}}$. So, we need P_s^{-1}, that is,

$$P_s^{-1} = \begin{bmatrix} \tilde{A}^{-1} & 0 \\ S_0^{-1}B\tilde{A}^{-1} & -S_0^{-1} \end{bmatrix}. \tag{35}$$

In what follows, we derive the approximation to the blocks of P_s^{-1}. First, to approximate $\tilde{A}$, we can use algebraic multigrid methods, since $\tilde{A}$ is symmetric and positive definite. Next, we need an approximation to the Schur complement S_0. As was pointed out in [13], the pressure mass matrix is a very effective approximation for S_0 in the case of stationary Stokes equations. However, as we are considering unsteady Stokes-Brinkman constraint, this does not apply since $\tilde{A}$ has an entirely different structure. Thus, following [56], we proceed to derive the so-called Cahouet-Chabard approximation to S_0 using a technique for the steady Navier-Stokes equation, which is based on the least squares commutator (see Chapter 8 of [13]) defined by

$$\mathbb{E} := (\mathbb{L})\nabla - \nabla(\mathbb{L}_p),$$

where $\mathbb{L} = (\tau^{-1} + \eta)I + \Delta + K_0$ and $\mathbb{L}_p = (\tau^{-1} + \eta)I_p + \Delta_p + K_{0_p}$ is defined similarly but on the pressure space. As was noted in [56], these operators are only used for the purpose of deriving matrix preconditioners and no function spaces or boundary conditions are defined here. Assuming the least squares commutator is small, we obtain the following finite element discretization of the differential operators

$$\mathbb{E}_h = (M^{-1}\tilde{A})M^{-1}B^T - M^{-1}B^T(M_p^{-1}\tilde{A}_p) \approx 0, \tag{36}$$

where $\tilde{A}, B$ and M are as defined previously, and

$$\tilde{A}_p \quad = \quad \nu_0 K_p + M_{k_p} + (\tau^{-1} + \eta)M_p. \tag{37}$$

Next, we pre-multiply the expression (36) by $B\tilde{A}^{-1}M$ and post-multiply it by $\tilde{A}_p^{-1}M_p$ to obtain

$$BM^{-1}B^T\tilde{A}_p^{-1}M_p - B\tilde{A}^{-1}B^T \approx 0, \tag{38}$$

or, equivalently,

$$S_0 \approx BM^{-1}B^T\tilde{A}_p^{-1}M_p. \tag{39}$$

Now, note that the matrix on the right hand side of (39) is not, in general, a practical choice for the Schur complement S_0 since $BM^{-1}B^T$ is not easy to work with because it is dense. Fortunately, though, $BM^{-1}B^T$ is spectrally equivalent to the Laplacian K_p defined on the pressure space [13]; that is, $K_p \sim BM^{-1}B^T$ in the sense that there exist constants c_0 and c_1 independent of h such that $0 < c_0 \leq c_1 < \infty$ with

$$c_0 \leq \frac{\langle BM^{-1}B^T\mathbf{v},\mathbf{v}\rangle}{\langle K_p\mathbf{v},\mathbf{v}\rangle} \leq c_1, \qquad \forall \mathbf{v}\in\mathbb{R}^{J_p},\ \mathbf{v}\neq\mathbf{1}.$$

This observation suggests that in general a discrete Laplacian on the pressure space is what is needed in place of $BM^{-1}B^T$ in (39). Hence, from (39), we obtain

$$S_0 \approx K_p\tilde{A}_p^{-1}M_p. \tag{40}$$

Hence, from (37) and (40), we have

$$S_0^{-1} \approx M_p^{-1}\left(\nu_0K_p + M_{k_p} + (\tau^{-1}+\eta)M_p\right)K_p^{-1}. \tag{41}$$

The inverse of the pressure Laplacian K_p^{-1} is approximated using algebraic multigrid methods, whereas the use of the Chebyshev semi-iteration will suffice for M_p^{-1}. We note here that, as pointed out in Chapter 5 of [13], the pressure Laplacian represents a Neumann problem because the pressure basis functions form a partition of unity. Indeed, this property is independent of the boundary conditions attached to the flow problem. To solve the problem of indefiniteness of K_p we just pin a boundary node in K_p (see, e.g., [7]). Afterwards, we use the AMG package provided by [8].

4.3. Spectral analysis

The effectiveness of the iterative solver for our KKT linear system (21) depends to a large extent on how well the exact Schur complement is represented by its approximation. To measure this, we need to consider the eigenvalues of the preconditioned Schur complement $\boldsymbol{S}_2^{-1}\tilde{\boldsymbol{S}}_2$. We are, however, unable to give a general estimate. Instead, we restrict our analysis to the regularization parameters.

Theorem 1. *If the system matrix $\boldsymbol{K}$ in (19) and its velocity block are invertible, then there exist constants C_1 and C_2 such that*

$$\begin{aligned}\mathrm{cond}(\boldsymbol{S}_2^{-1}\tilde{\boldsymbol{S}}_2) &\leq (1+C_1\beta^{1/2}) \quad \textit{for } \beta \textit{ sufficiently small},\\ \mathrm{cond}(\boldsymbol{S}_2^{-1}\tilde{\boldsymbol{S}}_2) &\leq (1+C_2\beta^{-1/2}) \quad \textit{for } \beta \textit{ sufficiently large},\end{aligned} \tag{42}$$

where C_1 and C_2 are independent of β.

Proof. Recall first that if

$$\boldsymbol{K}^T = \begin{bmatrix} \boldsymbol{A}^T & \boldsymbol{B}^T \\ \boldsymbol{B} & 0 \end{bmatrix},$$

where

$$\boldsymbol{B} = I_{n_t}\otimes G_0\otimes B, \tag{43}$$

$$\boldsymbol{A} = I_{n_t}\otimes G_0\otimes(\nu_0K + M_\varrho + \tau^{-1}M) + I_{n_t}\otimes G_1\otimes\nu_1K + C\otimes G_0\otimes\tau^{-1}M, \tag{44}$$

and that both $\boldsymbol{K}^T$ and $\boldsymbol{A}$ are non-singular, then

$$\boldsymbol{K}^{-T} = \begin{bmatrix} \boldsymbol{A}^{-T} - \boldsymbol{A}^{-T}\boldsymbol{B}^T\boldsymbol{S}^{-1}\boldsymbol{B}\boldsymbol{A}^{-T} & \boldsymbol{A}^{-T}\boldsymbol{B}^T\boldsymbol{S}^{-1} \\ \boldsymbol{S}^{-1}\boldsymbol{B}\boldsymbol{A}^{-T} & -\boldsymbol{S}^{-1} \end{bmatrix}, \tag{45}$$

and

$$\boldsymbol{K}\boldsymbol{K}^{-T} = \begin{bmatrix} \boldsymbol{A}\boldsymbol{A}^{-T}(I - \boldsymbol{P}_K) + \boldsymbol{P}_K & (\boldsymbol{A}\boldsymbol{A}^{-T} - I)\boldsymbol{B}^T\boldsymbol{S}^{-1} \\ 0 & I \end{bmatrix},$$

where $\boldsymbol{S} = \boldsymbol{B}\boldsymbol{A}^{-T}\boldsymbol{B}^T$, $\boldsymbol{P}_K = \boldsymbol{B}^T\boldsymbol{S}^{-1}\boldsymbol{B}\boldsymbol{A}^{-T}$, and I is an identity of suitable sizes, see e.g. [6]. Notice that $\boldsymbol{P}_K = \boldsymbol{P}_K^2$; that is, the matrix $\boldsymbol{P}_K$ is a projector and, from (43) and (44), it is also β-independent. From (22), (25) and (45), we have that

$$\beta^{-1}\boldsymbol{M}_{-1}\boldsymbol{K}^{-T}\boldsymbol{M}_1 = \begin{bmatrix} \boldsymbol{M}_\star & 0 \\ 0 & 0 \end{bmatrix}, \tag{46}$$

where

$$\boldsymbol{M}_\star = \beta^{-1}\mathtt{M}_{-1}\boldsymbol{K}_{11}\mathtt{M}_1, \tag{47}$$

$\mathtt{M}_{-1} = D^{-1} \otimes G_0 \otimes M$ and $\mathtt{M}_1 = D \otimes G_\alpha \otimes M$ are the velocity submatrices of $\boldsymbol{M}_{-1}$ and $\boldsymbol{M}_1$, as given by (25) and (22) respectively, and $\boldsymbol{K}_{11} = \boldsymbol{A}^{-T}(I - \boldsymbol{P}_K)$ denotes the (1,1) block of $\boldsymbol{K}^{-T}$. Thus, using (47), (46) and (26), we get

$$\boldsymbol{S}_2 = \boldsymbol{K} + \beta^{-1}\boldsymbol{M}_{-1}\boldsymbol{K}^{-T}\boldsymbol{M}_1 = \begin{bmatrix} \boldsymbol{A}_\star & \boldsymbol{B}^T \\ \boldsymbol{B} & 0 \end{bmatrix}, \tag{48}$$

where

$$\boldsymbol{A}_\star = \boldsymbol{A} + \boldsymbol{M}_\star. \tag{49}$$

Next, observe from (31) that

$$\tilde{\boldsymbol{S}}_2 - \boldsymbol{S}_2 = \beta^{-1/2}(\boldsymbol{M}_{-1} + \boldsymbol{K}\boldsymbol{K}^{-T}\boldsymbol{M}_1) = \begin{bmatrix} \boldsymbol{U} & 0 \\ 0 & 0 \end{bmatrix}, \tag{50}$$

where

$$\boldsymbol{U} = \beta^{-1/2}\underbrace{\left(\mathtt{M}_{-1} + \left(\boldsymbol{A}\boldsymbol{A}^{-T}(I - \boldsymbol{P}_K) + \boldsymbol{P}_K\right)\mathtt{M}_1\right)}_{:=U_1}. \tag{51}$$

Notice from (51) that U_1 is also β-independent. Now, using (45), (48) and (50), we have

$$\begin{aligned} \boldsymbol{S}_2^{-1}\tilde{\boldsymbol{S}}_2 &= \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} + \begin{bmatrix} \boldsymbol{A}_\star & \boldsymbol{B}^T \\ \boldsymbol{B} & 0 \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{U} & 0 \\ 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} I + \boldsymbol{A}_\star^{-1}(I - \boldsymbol{P}_\star)\boldsymbol{U} & 0 \\ \boldsymbol{S}_\star^{-1}\boldsymbol{B}\boldsymbol{A}_\star^{-1}\boldsymbol{U} & I \end{bmatrix}, \end{aligned} \tag{52}$$

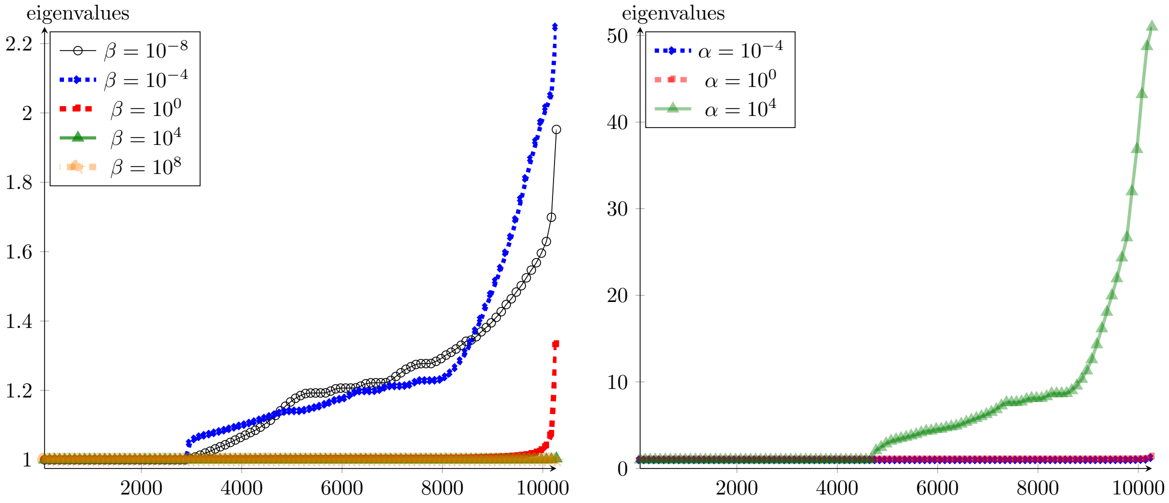
where $\boldsymbol{S}_\star = \boldsymbol{B}\boldsymbol{A}_\star^{-1}\boldsymbol{B}^T$ and

$$\boldsymbol{P}_\star = \boldsymbol{B}^T\boldsymbol{S}_\star^{-1}\boldsymbol{B}\boldsymbol{A}_\star^{-1} \tag{53}$$

is another projector. Thus, the eigenvalues of $\boldsymbol{S}_2^{-1}\tilde{\boldsymbol{S}}_2$ are contained in the set $\{1\} \cup \lambda\left(I + \boldsymbol{A}_\star^{-1}(I - \boldsymbol{P}_\star)\boldsymbol{U}\right)$.

To prove the first part of the assertion (42), suppose now that β is sufficiently small. Then, from (47), the norm of $\boldsymbol{M}_\star$ is large; this means that, in (49), $\|\boldsymbol{M}_\star\| = \|\beta^{-1}\mathtt{M}_{-1}\boldsymbol{K}_{11}\mathtt{M}_1\| \gg \|\boldsymbol{A}\|$, since $\boldsymbol{A}$ is β-independent.

Figure 1: Eigenvalue distribution of the matrix $I + \boldsymbol{A}_\star^{-1}(I - \boldsymbol{P}_\star)\boldsymbol{U}$ using the parameters $\nu_1 = 0.1, J = 642,\ P = 4,\ n_t = 4$. Left: $\alpha = 1$ and β is varied. Right: $\beta = 1$ and α is varied.



This, in turn, implies that $\beta \ll \|\mathtt{M}_{-1}\boldsymbol{K}_{11}\mathtt{M}_1\|/\|\boldsymbol{A}\|$, where the bound is completely independent of β. Hence, $\boldsymbol{A}_\star \approx \boldsymbol{M}_\star$. In particular, we have

$$0 < \widehat{c}\beta \leq \|\boldsymbol{A}_\star^{-1}\| \leq \widehat{C}\beta$$

and

$$0 < c'\beta^{-1} \leq \|\boldsymbol{S}_\star^{-1}\| \leq C'\beta^{-1},$$

from which, together with (53), we deduce that the norm of the projector $\boldsymbol{P}_\star$ is asymptotically β-independent. Finally, from (51), we have $\|\boldsymbol{U}\| = \|\beta^{-1/2}U_1\| := \tilde{C}\beta^{-1/2}$, and $\|\boldsymbol{A}_\star^{-1}(I - \boldsymbol{P}_\star)\boldsymbol{U}\| \leq C_1\beta^{1/2}$. That is, $\lambda(\boldsymbol{S}_2^{-1}\tilde{\boldsymbol{S}}_2) \in [1 - C_1\beta^{1/2}, 1 + C_1\beta^{1/2}] \to \{1\}$ when $\beta \to 0$.

On the other hand, when β is large, the norm of $\boldsymbol{M}_\star$ is small, and $\boldsymbol{A}_\star \approx \boldsymbol{A}$, a matrix independent of β. The only multiplication with β comes from $\boldsymbol{U}$; therefore, $\|\boldsymbol{A}_\star^{-1}(I - \boldsymbol{P}_\star)\boldsymbol{U}\| \leq C_2\beta^{-1/2} \to 0$ when $\beta \to \infty$. Again, the matrix $\boldsymbol{S}_2^{-1}\tilde{\boldsymbol{S}}_2$ becomes well conditioned in β in the limit, thereby completing the proof of the theorem. □

For intermediate β, we expect that $\tilde{\boldsymbol{S}}_2$ is still a good approximation to $\boldsymbol{S}_2$, and do observe that in practice. For small matrices we have illustrated the distribution of the eigenvalues of $I + \boldsymbol{A}_\star^{-1}(I - \boldsymbol{P}_\star)\boldsymbol{U}$ explicitly in Figure 1. As we can see from the left figure, as β is varied, the eigenvalues are mostly clustered between 1 and 2.2, regardless of the value of β. Note that the eigenvalues approach the maximum 2.2 for intermediate $\beta = 10^{-4}$, but remain closer to 1 for both larger and smaller values of β. This is also reflected in the experiment in Section 6.4: $\beta = 10^{-4}$ is the kink point for the error, and the maximum point for the CPU time.

On the other hand, Figure 1 (right) shows that, keeping $\beta = 1$, the eigenvalues of $I + \boldsymbol{A}_\star^{-1}(I - \boldsymbol{P}_\star)\boldsymbol{U}$ are clustered around 1 if $0 \leq \alpha \leq 1$, but drastically increase for $\alpha > 1$. Again, this observation confirms the deterioration in the performance of our solver as α increases in Section 6.5. The scenario $\alpha \gg 1$ is not of much practical interest anyway, as this would imply a very low value of the variance, in which case we lose the point of uncertainty quantification in the problem.

5. Tensor Train solver

To develop an efficient tensor-based iterative solver for our problem, we separate variables $\mathbf{x}$, ω and t, but not the inner components of $\mathbf{x}$. In what follows, we shall rely specifically on the *Tensor Train* (TT) decomposition introduced in [42] to solve our linear systems. A detailed discussion on tensor decompositions can be found in recent surveys and books [18, 17, 28]. Here, we start with the introduction to TT calculus, and finally propose the new solution algorithm for the systems like (21).

5.1. Tensor Train decomposition

By a *tensor* $\mathbf{y}$, we mean an array with many indices: $\mathbf{y} = [\mathbf{y}(i_1, \dots, i_d)]$, where $i_m = 1, \dots, n_m$, $m = 1, \dots, d$. For example, the $\mathbf{y}$-component of (21) can be seen as a 3-dimensional tensor, where i_3 enumerates basis functions in $\mathbf{x}$, i_2 defines different polynomials in the stochastic variable $\xi(\omega)$, and i_1 corresponds to different time steps, t_{i_1}. On the other hand, it is clear that the solution of a linear system is a vector. The equivalence between vectors, matrices and tensors is established via the *reshaping* operation, employed ubiquitously in tensor calculus.

Definition 1. *Given arbitrary indices $i_p, \dots, i_q$ with ranges $n_p, \dots, n_q$, by a* multi-index *we denote their big-endian lexicographic grouping,*

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

Definition 2. *Given a tensor $\mathbf{y}(i_1, \dots, i_d)$, we define the following classical* reshapes *using multi-indices:*

- *vector* $\mathbf{y} = \left[\mathbf{y}(\overline{i_1 \dots i_d})\right] \in \mathbb{R}^{n_1 \cdots n_d \times 1}$, $\quad \mathbf{y}(\overline{i_1 \dots i_d}) = \mathbf{y}(i_1, \dots, i_d)$, *and*
- *matrix* $\mathbf{Y} = \left[\mathbf{Y}(\overline{i_1 \dots i_m}, \overline{i_{m+1} \dots i_d})\right] \in \mathbb{R}^{n_1 \cdots n_m \times n_{m+1} \cdots n_d}$, $\quad \mathbf{Y}(\overline{i_1 \dots i_m}, \overline{i_{m+1} \dots i_d}) = \mathbf{y}(i_1, \dots, i_d)$.

For example, the Kronecker product $\mathbf{c} = \mathbf{a} \otimes \mathbf{b}$ of two vectors can be equivalently seen as a tensor, matrix or vector, since $\mathbf{c}(i_1, i_2) = \mathbf{c}(\overline{i_1 i_2}) = \mathbf{a}(i_1)\mathbf{b}(i_2)$.

The computational difficulty of tensors lies in their possibly large storage demand $n_1 \cdots n_d$, which grows exponentially in d if all n_k are equal. However, notice that the Kronecker product $\mathbf{a} \otimes \mathbf{b}$ above can be seen as a *low-rank* matrix, and we only need to store its factors $\mathbf{a}, \mathbf{b}$ of total cardinality $n_1 + n_2$. One of the simplest and powerful generalizations of this idea to higher dimensions is the *Tensor Train* (TT) decomposition.

Definition 3. *A tensor is said to be represented in the TT format, if*

$$\mathbf{y}(i_1, \dots, i_d) = \sum_{s_0, \dots, s_d = 1}^{r_0, \dots, r_d} \mathbf{y}^{(1)}_{s_0, s_1}(i_1) \mathbf{y}^{(2)}_{s_1, s_2}(i_2) \cdots \mathbf{y}^{(d)}_{s_{d-1}, s_d}(i_d). \tag{54}$$

The summation indices $r_0, \dots, r_d$ are called *TT ranks*, the factors $\mathbf{y}^{(m)}$, $m = 1, \dots, d$, are called *TT blocks* and have the sizes $r_{m-1} \times n_m \times r_m$. We put the rank indices to subscripts and the original indices to brackets to emphasize particular levels in (54). Omitting either group of indices, we can write equivalent compact representations as a sum of Kronecker products or a product of matrices, respectively,

$$\mathbf{y} = \sum_{s_0, \dots, s_d = 1}^{r_0, \dots, r_d} \mathbf{y}^{(1)}_{s_0, s_1} \otimes \mathbf{y}^{(2)}_{s_1, s_2} \otimes \cdots \otimes \mathbf{y}^{(d)}_{s_{d-1}, s_d} \quad \Leftrightarrow \quad \mathbf{y}(i_1, \dots, i_d) = \mathbf{y}^{(1)}(i_1) \cdots \mathbf{y}^{(1)}(i_d).$$

Here, for example, $\mathbf{y}^{(m)}_{s_{m-1}, s_m} \in \mathbb{R}^{n_m}$ is a vector, and $\mathbf{y}^{(m)}(i_m) \in \mathbb{R}^{r_{m-1} \times r_m}$ is a matrix.

In this paper, we always fix $r_0 = r_d = 1$; these dimensions are introduced only for uniformity of $\mathbf{y}^{(1)}$ and $\mathbf{y}^{(d)}$. The other TT ranks are however nontrivial, and depend on the enforced accuracy, since (54) can hold approximately. If all TT ranks are bounded, $r_m \lesssim r$, and $n_m \lesssim n$, then the TT blocks require in total $\mathcal{O}(dnr^2)$ memory, which might be much less than n^d, needed for the full vector $\mathbf{y}$. Although it is difficult in general to estimate the TT ranks theoretically, there is a reliable numerical TT-SVD procedure, which computes a quasi-optimal TT representation, using a sequence of singular value decompositions (SVD) [42].

The complexity of the TT-SVD is $\mathcal{O}(n^{d+1})$ if we compress a full tensor. However, in the course of computations we mostly need to re-compress a tensor, given already in the TT format, but with (overly) larger ranks. For example, given a matrix as a sum of Kronecker products, $\boldsymbol{A} = \sum_{q=1}^{R} A_q \otimes B_q \otimes C_q$ and a vector $\mathbf{y}$ in the format (54) with $d = 3$, the matrix-vector product can be written as follows [53, 42],

$$\mathbf{g} = \boldsymbol{A}\mathbf{y} = \sum_{s_1,s_2=1}^{r_1,r_2} \sum_{q_1,q_2=1}^{R,R} \left(A_{q_1}\mathbf{y}^{(1)}_{s_1}\right) \otimes \left(\delta_{q_1,q_2} B_{q_1}\mathbf{y}^{(2)}_{s_1,s_2}\right) \otimes \left(C_{q_2}\mathbf{y}^{(3)}_{s_2}\right), \tag{55}$$

where $\delta_{q_1,q_2} = 1$ if $q_1 = q_2$ and zero otherwise. Each bracket in the right-hand side of (55) is a larger TT block, the new rank indices are $s_1' = \overline{s_1 q_1}$, $s_2' = \overline{s_2 q_2}$, and hence the TT ranks are Rr_1, Rr_2. Similarly, a linear combination $\mathbf{y} + \mathbf{z}$ of vectors can be recast to their TT blocks $\mathbf{y}^{(m)}, \mathbf{z}^{(m)}$. However, the result might be approximated accurately enough with much smaller ranks. When applied to the TT format (55) instead of the full tensor, the TT-SVD requires $\mathcal{O}(nR^3r^3)$ operations. These properties allow to adopt classical iterative methods such as MINRES or GMRES in an *inexact* fashion, keeping all Krylov vectors in the TT format and performing the TT-SVD re-compression [33, 4, 1, 10].

5.2. Alternating iterative methods

Notwithstanding the TT truncation, the Krylov vectors may still develop rather large TT ranks – much larger than the ranks of the exact solution, in particular. Unless a very good preconditioner is available, such that the method converges in about 10 iterations, the TT-GMRES approach may become too expensive. For problems of some special forms (e.g. Lyapunov equations), one can employ ADI [59] or tensor Krylov methods [32]. For more general problems we have to employ more general alternating methods [22, 53].

The main idea behind the alternating tensor methods is to reduce the problem to the elements of a particular TT block and iterate over different TT blocks until convergence is achieved. In the mathematical community, the concept started with the *Alternating Least Squares* (ALS) method used to minimize the misfit of a tensor by a low-rank tensor model, see the surveys [30, 17]. This was later extended to the solution of linear systems [22, 44]. In quantum physics, a powerful realization of the alternation idea is the Density Matrix Renormalization Group (DMRG) algorithm [60], which is mainly used for eigenvalue problems, but also for linear systems [26]. Later on, the ALS/DMRG methods were combined with the classical gradient descent iteration: besides the ALS iteration, the TT blocks are explicitly augmented by the partial TT format of the residual surrogate. The DMRG algorithm with a single center site [61] uses the surrogate of the Krylov vector, and the Alternating Minimal Energy (AMEn) method [12] uses the actual residual, which was later adopted for eigenvalue problems as well [24, 31]. Details of these algorithms can be found in the corresponding papers.

Let us consider a linear system $\boldsymbol{A}\mathbf{y} = \mathbf{g}$ with a symmetric positive definite $\boldsymbol{A}$, such that the problem can be reformulated as $\min_{\mathbf{y}} J(\mathbf{y}) = \mathbf{y}^\top \boldsymbol{A}\mathbf{y} - 2\mathbf{y}^\top \mathbf{g}$. The ALS method plugs in the TT format (54) instead of $\mathbf{y}$, and minimizes J over a single TT block $\mathbf{y}^{(m)}$ in the course of iteration $m = 1, \ldots, d$. The optimality conditions result in a smaller linear system [22], for which we need the following

Definition 4. *Given the TT format* (54)*, we introduce the* left *resp.* right interface *matrices*

$$\boldsymbol{Y}^{<m}(\overline{i_1 \ldots i_{m-1}}, s_{m-1}) = \sum_{s_0,\ldots,s_{m-2}} \mathbf{y}^{(1)}_{s_0,s_1}(i_1) \cdots \mathbf{y}^{(m-1)}_{s_{m-2},s_{m-1}}(i_{m-1}),$$
$$\boldsymbol{Y}^{>m}(s_m, \overline{i_{m+1} \ldots i_d}) = \sum_{s_{m+1},\ldots,s_d} \mathbf{y}^{(m+1)}_{s_m,s_{m+1}}(i_{m+1}) \cdots \mathbf{y}^{(d)}_{s_{d-1},s_d}(i_d),$$

including the degenerate cases $\boldsymbol{Y}^{<1} = \boldsymbol{Y}^{>d} = 1$, and the frame *matrix*

$$\boldsymbol{Y}_m = \boldsymbol{Y}^{<m} \otimes I_{n_m} \otimes \left(\boldsymbol{Y}^{>m}\right)^\top. \tag{56}$$

The ALS method proceeds solving

$$\left(\boldsymbol{Y}_m^\top \boldsymbol{A} \boldsymbol{Y}_m\right) \mathbf{y}^{(m)} = \boldsymbol{Y}_m^\top \mathbf{g}, \quad \text{for} \quad m = 1, \ldots, d, d-1, \ldots, 1, \tag{57}$$

and so on, replacing the current TT block by the new elements $\mathbf{y}^{(m)}$ before constructing (56) for the next step. This follows from the crucial *linearity* of the TT format, $\mathbf{y} = \boldsymbol{Y}_m \mathbf{y}^{(m)}$, where $\mathbf{y}$ and $\mathbf{y}^{(m)}$ here are reshaped into vectors (see Def. 2).

However, the convergence of this algorithm is questionable. It is possible that the systems (57) remain the same within machine precision in two consecutive iterations, while the true residual of the initial linear system $\boldsymbol{A}\mathbf{y} - \mathbf{g}$ is large. The AMEn algorithm [12] was developed to circumvent this problem. In addition to the solution, we approximate the residual in the TT format, and add it to the solution after each step. This allows to increase the TT ranks if necessary, and prevent the method from a premature stagnation.

5.3. Block alternating iteration

However, the idea outlined above may not work for indefinite matrices, such as the KKT system (21). The Galerkin projections (57) obey the Poincaré Separation Theorem [23, Section 4.3], and since the spectrum has both positive and negative parts, some of the eigenvalues may interlace to zero. The projected matrices become degenerate and the calculation stops.

To avoid this problem, we store the state $\mathbf{y}$, control $\mathbf{u}$ and adjoint $\boldsymbol{\lambda}$ vectors in the so-called *block* TT format [11], and preserve the KKT structure in the reduced system. Suppose we are given a system

$$\begin{bmatrix} \tau \boldsymbol{M}_1 & 0 & -\boldsymbol{K}^\top \\ 0 & \beta\tau \boldsymbol{M}_2 & \boldsymbol{N}^\top \\ -\boldsymbol{K} & \boldsymbol{N} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \mathbf{w}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \end{bmatrix} = \begin{bmatrix} \tau \boldsymbol{M}_a \bar{\mathbf{y}} \\ 0 \\ \mathbf{g} \end{bmatrix}, \qquad \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \mathbf{w}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \boldsymbol{\lambda}, \end{bmatrix},$$

where each component $\mathbf{w}_l$, $l = 1, 2, 3$, is a vector associated with a d-dimensional tensor.

Definition 5. *A* set *of tensors is said to be represented in the* block *TT format, if for some m,*

$$\mathbf{w}_l(i_1, \ldots, i_d) = \sum_{s_0, \ldots, s_d = 1}^{r_0, \ldots, r_d} \mathbf{w}^{(1)}_{s_0,s_1}(i_1) \cdots \hat{\mathbf{w}}^{(m)}_{s_{m-1},s_m}(i_m, l) \cdots \mathbf{w}^{(d)}_{s_{d-1},s_d}(i_d). \tag{58}$$

Here, $\mathbf{w}^{(p)} \in \mathbb{R}^{r_{p-1} \times n_p \times r_p}$ are ordinary TT blocks as in (54) for $p = 1, \ldots, m-1$ and $m+1, \ldots, d$, but $\hat{\mathbf{w}}^{(m)} \in \mathbb{R}^{r_{m-1} \times n_m \times 3 \times r_m}$ is a larger block defining the components of $\mathbf{w}$.

The following two observations are crucial for us:

- the component index l appears in one arbitrary TT block;
- the component index appears in *only* one block, hence the frame matrix $\boldsymbol{W}_m$ (defined in the same guise as in Def. 4) is independent of l.

The first property implies that the initial guess can be chosen with an arbitrary position of l. Moreover, we can *move* l from one TT block to another using the SVD [11]. Indeed, let us reshape $\hat{\mathbf{w}}^{(m)}$ into a matrix $\hat{W}^{(m)} \in \mathbb{R}^{r_{m-1} n_m \times 3 r_m}$ by assigning $\hat{W}^{(m)}(\overline{s_{m-1} i_m}, \overline{l s_m}) = \hat{\mathbf{w}}^{(m)}_{s_{m-1},s_m}(i_m, l)$, and compute its truncated SVD

$$\hat{W}^{(m)} \approx U \Sigma V^\top, \qquad U \in \mathbb{R}^{r_{m-1} n_m \times r'_m}, \quad V \in \mathbb{R}^{3 r_m \times r'_m}.$$

Notice that U can overwrite the m-th TT block, by setting $\mathbf{w}^{(m)}_{s_{m-1},s'_m}(i_m) = U(\overline{s_{m-1} i_m}, s'_m)$, $s'_m = 1, \ldots, r'_m$. The remaining matrices can be seen as a tensor $\mathbf{G}_{s'_m,s_m}(l) = \Sigma V^\top(s'_m, \overline{l s_m})$ and multiplied with the next TT block as follows,

$$\hat{\mathbf{w}}^{(m+1)}_{s'_m,s_{m+1}}(i_{m+1}, l) = \sum_{s_m=1}^{r_m} \mathbf{G}_{s'_m,s_m}(l) \mathbf{w}^{(m+1)}_{s_m,s_{m+1}}(i_{m+1}),$$

or, omitting the rank indices, simply as $\hat{\mathbf{w}}^{(m+1)}(i_{m+1}, l) = \mathbf{G}(l) \mathbf{w}^{(m+1)}(i_{m+1})$. We see that we have obtained the same form as (58), with m replaced by $m+1$, and the rank r_m replaced by the new rank r'_m. In a similar way, we can move l from the m-th to the $(m-1)$-th TT block. The pseudocodes are provided in

Algorithm 1 Switching from m-th to $(m+1)$-th block TT format

```
Populate $\hat{W}^{(m)}$ by the elements $\hat{W}^{(m)}(\overline{s_{m-1}i_m}, \overline{ls_m}) = \hat{\mathbf{w}}^{(m)}_{s_{m-1},s_m}(i_m, l)$.
Compute SVD $\hat{W}^{(m)} \approx U\Sigma V^\top$, with the new rank$(U) = r'_m$.
Populate $\mathbf{w}^{(m)}$ by the elements $\mathbf{w}^{(m)}_{s_{m-1},s'_m}(i_m) = U(\overline{s_{m-1}i_m}, s'_m)$.
Populate $\mathbf{G}$ by the elements $\mathbf{G}_{s'_m,s_m}(l) = G(s'_m, \overline{ls_m})$, where $G = \Sigma V^\top$.
Compute new $\hat{\mathbf{w}}^{(m+1)}(i_{m+1}, l) = \mathbf{G}(l)\mathbf{w}^{(m+1)}(i_{m+1})$.
Replace the rank $r_m = r'_m$.
```

Algorithm 2 Switching from m-th to $(m-1)$-th block TT format

```
Populate $\hat{W}^{(m)}$ by the elements $\hat{W}^{(m)}(\overline{s_{m-1}l}, \overline{i_m s_m}) = \hat{\mathbf{w}}^{(m)}_{s_{m-1},s_m}(i_m, l)$.
Compute SVD $\hat{W}^{(m)} \approx U\Sigma V^\top$, with the new rank$(V) = r'_{m-1}$.
Populate $\mathbf{w}^{(m)}$ by the elements $\mathbf{w}^{(m)}_{s'_{m-1},s_m}(i_m) = V^\top(s'_{m-1}\overline{i_m s_m})$.
Populate $\mathbf{G}$ by the elements $\mathbf{G}_{s_{m-1},s'_{m-1}}(l) = G(\overline{s_{m-1}l}, s'_{m-1})$, where $G = U\Sigma$.
Compute new $\hat{\mathbf{w}}^{(m-1)}(i_{m-1}, l) = \mathbf{w}^{(m-1)}(i_{m-1})\mathbf{G}(l)$.
Replace the rank $r_{m-1} = r'_{m-1}$.
```

Algorithms 1 and 2, respectively. Generally, the TT ranks change after such transformations. However, in our numerical practice the ranks remained comparatively the same in different block representations.

The second property of (58) allows us to preserve the structure of the KKT system. Indeed, from (56) it follows that $\boldsymbol{W}_m \in \mathbb{R}^{n_1\cdots n_d \times r_{m-1}n_m r_m}$. By our assumption, the size of the KKT system is $3 \cdot n_1 \cdots n_d$. Therefore, instead of (57), the ALS step in the block TT format reads

$$\begin{bmatrix} \boldsymbol{W}_m^\top \tau \boldsymbol{M}_1 \boldsymbol{W}_m & 0 & -\boldsymbol{W}_m^\top \boldsymbol{K}^\top \boldsymbol{W}_m \\ 0 & \boldsymbol{W}_m^\top \beta\tau \boldsymbol{M}_2 \boldsymbol{W}_m & \boldsymbol{W}_m^\top \boldsymbol{N}^\top \boldsymbol{W}_m \\ -\boldsymbol{W}_m^\top \boldsymbol{K} \boldsymbol{W}_m & \boldsymbol{W}_m^\top \boldsymbol{N} \boldsymbol{W}_m & 0 \end{bmatrix} \hat{\mathbf{w}}^{(m)} = \begin{bmatrix} \boldsymbol{W}_m^\top \mathbf{b}_1 \\ \boldsymbol{W}_m^\top \mathbf{b}_2 \\ \boldsymbol{W}_m^\top \mathbf{b}_3 \end{bmatrix}. \tag{59}$$

The size of this system is $3 \cdot r_{m-1}n_m r_m$, the same as the number of elements in $\hat{\mathbf{w}}^{(m)}$ according to (58). After this system is solved, we use Algorithms 1, 2 to switch to the next block TT representation.

Although these algorithms allow already to adapt TT ranks to the desired accuracy, it is still useful to incorporate the residual, as proposed in the AMEn algorithm [12], to improve the convergence. The residual is computed in the same fashion as the solution. Define the block storages

$$\mathbf{z} = \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \mathbf{z}_3 \end{bmatrix} \in \mathbb{R}^{n_1\cdots n_d \cdot 3}, \quad \mathbf{Z} = \begin{bmatrix} \mathbf{z}_1 & \mathbf{z}_2 & \mathbf{z}_3 \end{bmatrix} \in \mathbb{R}^{n_1\cdots n_d \times 3}, \qquad \begin{aligned} \mathbf{z}_1 &= \tau \boldsymbol{M}_1 \mathbf{w}_1 - \boldsymbol{K}^\top \mathbf{w}_3 - \mathbf{b}_1, \\ \mathbf{z}_2 &= \beta\tau \boldsymbol{M}_2 \mathbf{w}_2 + \boldsymbol{N}^\top \mathbf{w}_3 - \mathbf{b}_2, \\ \mathbf{z}_3 &= -\boldsymbol{K}\mathbf{w}_1 + \boldsymbol{N}\mathbf{w}_2 - \mathbf{b}_3. \end{aligned} \tag{60}$$

Introducing the block TT format for $\mathbf{z}$,

$$\mathbf{z}_l(i_1, \ldots, i_d) = \mathbf{z}^{(1)}(i_1) \cdots \hat{\mathbf{z}}^{(m)}(i_m, l) \cdots \mathbf{z}^{(d)}(i_d),$$

one can use the simple ALS algorithm, and solve $\min_{\hat{\mathbf{z}}^{(m)}} \|(\mathfrak{A}\mathbf{w} - \mathbf{b}) - \mathbf{z}\|_2^2$. This problem becomes much easier with the proper *orthogonality conditions* imposed on the TT blocks.

Definition 6. *A TT block* $\mathbf{w}^{(m)}$ *is said to be* left- *resp.* right-orthogonal*, if*

$$\sum_{i_m=1}^{n_m} (\mathbf{w}^{(m)}(i_m))^\top \mathbf{w}^{(m)}(i_m) = I_{r_m}, \quad \text{or} \quad \sum_{i_m=1}^{n_m} \mathbf{w}^{(m)}(i_m)(\mathbf{w}^{(m)}(i_m))^\top = I_{r_{m-1}}.$$

Reshaping $\mathbf{w}^{(m)}$ to a matrix and computing the QR decomposition, one can make $\mathbf{w}^{(m)}$ left- or right-orthogonal in $\mathcal{O}(nr^3)$ operations. We refer to [42] for details. The crucial fact is that the orthogonality of TT blocks implies orthogonality of the frame matrix.

- If $\mathbf{w}^{(p)}$ are left-orthogonal for $p = 1, \dots, m-1$, then $\boldsymbol{W}^{<m}$ has orthogonal columns.
- If $\mathbf{w}^{(p)}$ are right-orthogonal for $p = m+1, \dots, d$, then $\boldsymbol{W}^{>m}$ has orthogonal rows.
- If both previous conditions hold, then $\boldsymbol{W}_m$ has orthogonal columns.

Now the ALS step for the residual can be easily computed as $\hat{\mathbf{z}}^{(m)} = \boldsymbol{Z}_m^\top \mathbf{Z}$, provided that $\boldsymbol{Z}_m$ is made orthogonal. Due to its low complexity, this step is never a bottleneck.

Remark 2. The switching Algorithms 1 and 2 maintain orthogonality of $\boldsymbol{W}_{m+1}$ resp. $\boldsymbol{W}_{m-1}$ automatically if the input was given with orthogonal $\boldsymbol{W}_m$. This is also beneficial for the well-posedness of (59): if $\boldsymbol{K}$ is symmetric, the spectrum of the orthogonal projection $\boldsymbol{W}_m^\top \boldsymbol{K} \boldsymbol{W}_m$ lies within the spectrum of $\boldsymbol{K}$, and if the matrices are positive definite, it holds $\operatorname{cond}(\boldsymbol{W}_m^\top \boldsymbol{K} \boldsymbol{W}_m) \le \operatorname{cond}(\boldsymbol{K})$. Moreover, in this case the error introduced to the whole tensor $\mathbf{w}$ due to the SVD is equal to the error in $\hat{W}^{(m)}$.

Having solved the reduced system for $\mathbf{w}^{(m)}$, the AMEn algorithm *enriches* it with the projected residual. For example, if we iterate increasing m, we expand the two neighboring blocks as follows,

$$\mathbf{w}^{(m)}(i_m) := \begin{bmatrix} \mathbf{w}^{(m)}(i_m) & \mathbf{z}_w^{(m)}(i_m) \end{bmatrix}, \qquad \hat{\mathbf{w}}^{(m+1)}(i_{m+1}, l) := \begin{bmatrix} \hat{\mathbf{w}}^{(m+1)}(i_{m+1}, l) \\ 0 \end{bmatrix},$$

where we compute $\hat{\mathbf{z}}_w^{(m)} = \left(\boldsymbol{W}^{<m} \otimes I_{n_m} \otimes (\boldsymbol{Z}^{>m})^\top\right)^\top \mathbf{Z}$, and $\mathbf{z}_w^{(m)}$ is obtained from $\hat{\mathbf{z}}_w^{(m)}$ by Algorithm 1. The hybrid frame matrix in $\hat{\mathbf{z}}_w^{(m)}$ is used to match the sizes of $\mathbf{w}^{(m)}$ and $\mathbf{z}_w^{(m)}$, and the zeros in $\hat{\mathbf{w}}^{(m+1)}$ are used to match its size with $\mathbf{w}^{(m)}$.

Before we outline the final procedure, we need one more trick. Although $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ are symmetric and (semi)definite, the Stokes-Brinkman matrix $\boldsymbol{K}$ is indefinite. Moreover, the sizes of $\boldsymbol{M}_1, \boldsymbol{K}$ and $\boldsymbol{M}_2$ are different. We could consider the 2×2 Stokes-Brinkman block structure and the 3×3 KKT structure on the same level, and solve the 5×5 block system. However, the second row of the Stokes-Brinkman matrix has a very particular meaning, which we can exploit to reduce the complexity.

5.4. Pressure elimination in the reduced model

The low-rank separation of space and time variables has been used for a while in the numerical simulation of the Navier-Stokes equation. The Proper Orthogonal Decomposition (POD) is a well-known approach to model reduction [34]. It reshapes the *velocity* component of the solution to a matrix $Y = [\mathbf{y}(i_1, i_2)]$, computes the truncated SVD $Y \approx U \Sigma V^\top$, and uses the columns of V for the Galerkin reduction of the velocity operators. If we were solving the continuous equation, we would have a vector-valued function $V = V(\mathbf{x}) \in \mathbb{R}^r$, where r is the number of POD terms, and the reduced solution sought in the form $y(\mathbf{x}, t) \approx V(\mathbf{x}) a(t)$. Plugging this into the Stokes-Brinkman equation, and projecting the velocity equation onto V, we have

$$\begin{cases} \frac{da}{dt} - \nu \langle V^\top, \Delta V \rangle a + \langle V^\top, K_0 V \rangle a + \langle V^\top, \nabla p \rangle & = & \langle V^\top, u \rangle, \\ \nabla \cdot V a & = & 0. \end{cases}$$

Since $a(t)$ is not fixed a priori, from the second row we have $\nabla \cdot V(\mathbf{x}) = 0$. However, then in the first row $\langle V^\top, \nabla p \rangle = -\langle \nabla \cdot V^\top, p \rangle = 0$; that is, the reduced model contains no pressure at all. In the discrete formulation, we have the system (16), and the pressure part $V^\top \mathcal{B}^\top \mathbf{p}$ is not exactly zero due to the boundary conditions. Compared to alternating methods, the POD conducts one iteration (returning with $m = 1$, since it corresponds to the time variable), and hence requires either an explicit elimination of boundary conditions [3], or nonlinear corrections [41]. Alternating iterations allow us to proceed with the following more general Gauss-Seidel-type scheme.

From now on, we focus on our Stokes-Brinkman problem with $d = 3$. When we solve (59) for the spatial TT block ($m = 3$), we consider the 5×5 Stokes-KKT structure

$$\begin{bmatrix} \tau\hat{M}_1 & 0 & 0 & -\hat{A} & -\hat{B}^\top \\ 0 & 0 & 0 & -\hat{B} & 0 \\ 0 & 0 & \beta\tau\hat{M}_2 & \hat{N}^\top & 0 \\ -\hat{A} & -\hat{B}^\top & \hat{N} & 0 & 0 \\ -\hat{B} & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{w}}^{(3)}(1) \\ \hat{\mathbf{w}}^{(3)}(2) \\ \hat{\mathbf{w}}^{(3)}(3) \\ \hat{\mathbf{w}}^{(3)}(4) \\ \hat{\mathbf{w}}^{(3)}(5) \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{b}}_\mathbf{1} \\ 0 \\ 0 \\ \hat{\mathbf{g}}_\mathbf{v} \\ \hat{\mathbf{g}}_\mathbf{p} \end{bmatrix}, \tag{61}$$

where $\hat{M}_1 = \widehat{D}_\alpha \otimes M$, $\hat{M}_2 = \widehat{D}_0 \otimes M$, $\hat{N} = \widehat{I}_0 \otimes M$, $\hat{B} = \widehat{I}_0 \otimes B$,

$$\hat{A} = \widehat{I}_0 \otimes \left(\tau^{-1}M + \nu_0 K + M_k\right) + \widehat{I}_1 \otimes \nu_1 K + \widehat{C}_0 \otimes \tau^{-1}M,$$

the reduced matrices corresponding to the time t and the event ω are computed as

$$\begin{aligned} &\widehat{I}_0 = \mathcal{W}_3^\top \left(I \otimes G_0\right)\mathcal{W}_3, \quad \widehat{I}_1 = \mathcal{W}_3^\top \left(I \otimes G_1\right)\mathcal{W}_3, \quad \widehat{C}_0 = \mathcal{W}_3^\top \left(C \otimes G_0\right)\mathcal{W}_3, \\ &\widehat{D}_0 = \mathcal{W}_3^\top \left(D \otimes G_0\right)\mathcal{W}_3, \quad \widehat{D}_\alpha = \mathcal{W}_3^\top \left(D \otimes G_\alpha\right)\mathcal{W}_3, \end{aligned} \tag{62}$$

whereas the right-hand side parts are

$$\hat{\mathbf{b}}_\mathbf{1} = \mathcal{W}_3^\top \left(D\mathbf{e} \otimes G_0\mathbf{e}_1\right) \otimes \tau\bar{\mathbf{v}}, \qquad \begin{bmatrix} \hat{\mathbf{g}}_\mathbf{v} \\ \hat{\mathbf{g}}_\mathbf{p} \end{bmatrix} = \mathcal{W}_3^\top \left(\mathbf{e} \otimes \mathbf{e}_1\right) \otimes \begin{bmatrix} \mathbf{g}_v^0 \\ \mathbf{g}_p^0 \end{bmatrix},$$

and[3] $\mathcal{W}_3 = \sum\limits_{s_1=1}^{r_1} \mathbf{w}_{s_1}^{(1)} \otimes \mathbf{w}_{s_1}^{(2)} \in \mathbb{R}^{n_t P \times r_2}$ is a chunk of the frame matrix $\boldsymbol{W}_3$. We introduce this chunk and the Kronecker structures above in order to explain the preconditioner in the next section. We see that the solution components $\hat{\mathbf{w}}^{(3)}(2)$ and $\hat{\mathbf{w}}^{(3)}(5)$ denote the state and adjoint pressures, respectively.

Our further construction is based on the following two considerations. First, the velocity and the pressure are likely to have *similar* TT blocks for time and stochastic variables. The motivation is that $\mathbf{v}$ and $\mathbf{p}$ are connected only via the matrix $I \otimes G_0 \otimes B$, which is diagonal (or even identity) in these variables. Second, the block TT format can be seen as a representation of several vectors in the common basis, comprised from the TT blocks without l. Therefore, if the only difference between the velocity and pressure is in the spatial TT block, the other TT blocks can be computed based on the velocity only. Although it is unclear whether it is allowed in general to 'freeze' some components like that, in our numerical experiments we observed that the solution is accurate enough.

So, we include only the velocities and control to the new TT block $\hat{\mathbf{w}}^{(3)} = \left[\hat{\mathbf{w}}^{(3)}(1), \hat{\mathbf{w}}^{(3)}(3), \hat{\mathbf{w}}^{(3)}(4)\right]$. Since the pressures will not change in the subsequent AMEn steps ($m = 2, 1$), their contributions to the velocity equations can be recast to the right-hand side. More precisely, we construct the TT formats

$$\delta\mathbf{b}_\mathbf{1} = \sum_{s_1,s_2} \mathbf{w}_{s_1}^{(1)} \otimes G_0 \mathbf{w}_{s_1,s_2}^{(2)} \otimes B^\top \hat{\mathbf{w}}_{s_2}^{(3)}(5), \quad \delta\mathbf{g} = \sum_{s_1,s_2} \mathbf{w}_{s_1}^{(1)} \otimes G_0 \mathbf{w}_{s_1,s_2}^{(2)} \otimes B^\top \hat{\mathbf{w}}_{s_2}^{(3)}(2), \tag{63}$$

and correct the right-hand side of (21) as follows,

$$\begin{bmatrix} \mathbf{b}_\mathbf{1} \\ 0 \\ \mathbf{g} \end{bmatrix} \quad \rightarrow \quad \begin{bmatrix} \mathbf{b}_\mathbf{1} + \delta\mathbf{b}_\mathbf{1} \\ 0 \\ \mathbf{g} + \delta\mathbf{g} \end{bmatrix}. \tag{64}$$

After that, we conduct AMEn steps $m = 2, 1, 2$ with the system of the form (59), where $\boldsymbol{K}$ contains now only the velocity equation, and hence is positive definite. As a by-product, the sizes of all submatrices in (59) are equal, which allows to use the block TT format. When we come back to $m = 3$, we drop the right-hand side corrections and solve the full system (61). If we are to stop the iteration, we return the full solution, including $\hat{\mathbf{w}}^{(3)}(2)$ and $\hat{\mathbf{w}}^{(3)}(5)$.

[3] Remember that discretization sizes in the initial and "tensor" notations match as $n_1 = n_t$, $n_2 = P$ and $n_3 = J$.

5.5. *Practical implementation*

The preconditioner developed in Section 4.1 needs to be adjusted to the local problem (61). Although the reduced matrices (62) are small, they are dense, and it is impractical to compute the blocks of (61) explicitly. However, note that all of them are single Kronecker products except $\hat{A}$. Moreover, if the norms of K and M_k are sufficiently large, and ν_1 is small, then the first term in $\hat{A}$ dominates. Therefore, we replace $\hat{A}$ by its first term $\widehat{I}_0 \otimes \left(\tau^{-1}M + \nu_0 K + M_k\right)$ during the preconditioning. This also allows to avoid the second level of preconditioning for the Stokes-Brinkman system (34). Since $\hat{B}$ contains $\widehat{I}_0$, we can assemble the Stokes-Brinkman matrix in the Kronecker form as well,

$$\hat{\mathcal{K}} = \widehat{I}_0 \otimes \begin{bmatrix} \tau^{-1}M + \nu_0 K + M_k & B^\top \\ B & 0 \end{bmatrix}.$$

In the computation of $\mathbf{x}_3$ in an analog of (30), we can solve linear systems with $\hat{\mathcal{K}}$ directly. For two-dimensional cases, this approach is faster than iterations with (34). In the same way we approximate the factors of the Schur complement (32), e.g.

$$\hat{\mathcal{K}}^\top + \hat{\mathcal{M}}_r \approx \widehat{I}_0 \otimes \begin{bmatrix} \left(\frac{1}{\tau} + \frac{1}{\sqrt{\beta}}\frac{\|\widehat{D}_\alpha\|}{\|\widehat{I}_0\|}\right) M + \nu_0 K + M_k & B^\top \\ B & 0 \end{bmatrix}, \qquad (65)$$

where we approximated $\hat{\mathcal{M}}_r = \frac{1}{\sqrt{\beta}}\widehat{D}_\alpha \otimes M$ by $\widehat{I}_0 \frac{\|\widehat{D}_\alpha\|}{\|\widehat{I}_0\|\sqrt{\beta}} \otimes M$, and $\widehat{D}_\alpha$ and $\widehat{I}_0$ are defined in (62). For three dimensions (Section 6.10), the matrices become more dense, and we have to use iterative methods, preconditioning the velocity block by a multigrid cycle. Similar rank-1 approximation is performed for the TT blocks $\mathbf{w}^{(1)}$ and $\mathbf{w}^{(2)}$. Although they are smaller than the spatial block, they are still rather large to form and solve the systems (59) directly. The crucial point here, fortunately, is that the new preconditioner does not need to invert $\boldsymbol{M}_1$.

The final block AMEn procedure for the Stokes-like structure is summarized in Algorithm 3 on page 29.

6. Numerical experiments

A systematic study of the proposed technique will be conducted on two- and three-dimensional examples. We first consider the Stokes(-Brinkman) flow constraints on $\mathcal{D} = [0,1]^2$ with the inflow boundary conditions

$$v_1|_{x_1=0} = x_2(1-x_2), \quad v_2|_{x_1=0} = 0, \qquad v|_{x_2=0} = v|_{x_2=1} = 0,$$

and 'do-nothing' boundary conditions at $x_1 = 1$. The velocity operators are discretized with the *mini* elements [54] and the pressure operators are discretized with the piecewise linear finite elements. The stiffness matrices are assembled in FEniCS 1.5.0 package [37]. For the Stokes-Brinkmann equation, the coefficient is chosen as follows:

$$K_0(\mathbf{x}) = \begin{cases} \bar{K}_0, & (x_1-0.5)^2 + (x_2-0.5)^2 \le 0.15^2, \\ 0, & \text{otherwise.} \end{cases}$$

The right-hand side and the initial condition are zeros. The desired state is the deterministic stationary solution of the forward Stokes-Brinkman problem.

The model is characterized by 8 parameters: the spatial grid size J, the number of time steps n_t, the time interval T, regularization parameters α and β, variance ν_1, a threshold for the tensor approximation and the AMEn algorithm ε, and the porosity coefficient $\bar{K}_0$. For the sake of brevity, we perform 8 experiments, fixing all parameters to their default values and varying only one of them. The default parameters are the following: one-dimensional spatial grid size $n = 64$ (so that $J = 29059$), time grid size $n_t = 2^{10}$, time interval $T = 1$, regularization parameters $\beta = 10^{-6}$ and $\alpha = 1$, variance parameter[4] $\nu_1 = 0.1$, approximation

[4] In applications involving highly heterogeneous media, such as subsurface diffusion, the variance of a random field may be several orders in magnitude. However, a highly viscous fluid is more or less homogeneous, and the 10% variance is realistic. This is the case in biomedical modeling, for example.

tolerance $\varepsilon = 10^{-6}$, and pure Stokes coefficient $\bar{K}_0 = 0$. The mean viscosity is always fixed at $\nu_0 = 1$, since the behavior of the model is the same if $\nu_0 \sim 1/T$, so we can investigate either of these parameters. The stochastic polynomial degree $P = 16$.

We investigate several kinds of discrepancies, such as the residual, the misfit w.r.t. the desired state, and so on. Therefore, it is convenient to introduce a unifying notation. All errors are measured in the Frobenius norm, i.e. given the reference $\mathbf{y}_\star$ and trial $\mathbf{y}$ vectors, we compute

$$\mathcal{E}(\mathbf{y}, \mathbf{y}_\star) = \|\mathbf{y} - \mathbf{y}_\star\|_F / \|\mathbf{y}_\star\|_F. \tag{66}$$

By 'residual', we mean the maximal relative residual among the KKT system rows:

$$\text{residual} = \max\left(\mathcal{E}(\tau \boldsymbol{M}_1 \mathbf{y} - \boldsymbol{K}^\top \boldsymbol{\lambda}, \tau \boldsymbol{M}_a \bar{\mathbf{y}});\ \mathcal{E}(\tau\beta \boldsymbol{M}_2 \mathbf{u}, \mathbf{N}^\top \boldsymbol{\lambda});\ \mathcal{E}(-\boldsymbol{K}\mathbf{y} + \boldsymbol{N}\mathbf{u}, \mathbf{g})\right).$$

Since the KKT matrix is rather ill-conditioned, we also estimate the Frobenius-norm errors of the state and control components of the solution as follows. For each experiment, we solve the system with two thresholds, ε and 0.1ε. The solution components of the latter run, denoted as $\mathbf{y}_\star$ and $\mathbf{u}_\star$, are taken as the reference ones, and the relative errors are computed by (66).

Remark 3. This error estimate can be justified similarly to the Richardson extrapolation. Suppose the true error expands as $\|\mathbf{y} - \mathbf{y}_{ex}\| = C\varepsilon^\delta + o(\varepsilon^\delta)$ for some $C > 0, \delta > 0$. Using the triangle inequality twice, we get $\|\mathbf{y} - \mathbf{y}_{ex}\| \le \|\mathbf{y} - \mathbf{y}_\star\| + \|\mathbf{y}_\star - \mathbf{y}_{ex}\|$ and $\|\mathbf{y} - \mathbf{y}_\star\| \le \|\mathbf{y} - \mathbf{y}_{ex}\| + \|\mathbf{y}_{ex} - \mathbf{y}_\star\|$, and by our assumption $\|\mathbf{y}_\star - \mathbf{y}_{ex}\| = 10^{-\delta} \cdot C\varepsilon^\delta + o(\varepsilon^\delta)$. Therefore, $(1 - 10^{-\delta})\|\mathbf{y} - \mathbf{y}_{ex}\| \le \|\mathbf{y} - \mathbf{y}_\star\| + o(\varepsilon^\delta) \le (1 + 10^{-\delta})\|\mathbf{y} - \mathbf{y}_{ex}\|$. So we can estimate both δ and $\|\mathbf{y} - \mathbf{y}_{ex}\|$ from $\|\mathbf{y} - \mathbf{y}_\star\|$. In the AMEn algorithm, the error usually depends linearly on ε, i.e. the assumption holds with $\delta = 1$, and the true error is bounded by $\frac{1}{0.9}\|\mathbf{y} - \mathbf{y}_\star\| + o(\varepsilon)$.

The complexity indicators are the CPU time, memory consumption and the number of iterations. The CPU time is measured for a sequential MATLAB R2012b program, run under Linux at Intel Xeon X5650 CPU with 2.67GHz. The TT algorithms[5] are implemented within the TT-Toolbox [43]. The memory consumption is reported as the memory compression ratio by the TT format. It is computed as the number of TT elements over the total number of degrees of freedom in the solution, i.e.

$$\%\ \text{Mem} = \frac{n_t r_1 + r_1 P r_2 + r_2 J}{J P n_t} \cdot 100.$$

By 'iterations', we mean the total number of FGMRES iterations, spent in solving the reduced systems (61) for the spatial TT block, in all AMEn steps. The FGMRES is used with the block-triangular preconditioner (30) for the KKT level only (the Stokes-like systems (65) are solved directly in two-dimensional examples).

6.1. *Performance of the new block-triangular preconditioner*

It is illustrative to compare the new preconditioner (30) with the established block-diagonal preconditioner $\boldsymbol{P}_1$ from [56], mentioned at the beginning of Section 4. We test $\boldsymbol{P}_1$ using the MINRES method, for the spatial TT block only. The comparison with $\boldsymbol{P}_2$ (30) is given in Table 1. We see that $\boldsymbol{P}_2$ provides faster convergence in terms of both iterations and time. Therefore, we use it in all the remaining experiments in this paper.

6.2. *Experiment with n_t (Figure 2)*

In the first test, we vary the number of time steps from 2^5 to 2^{12}. In addition to the solution errors $\mathcal{E}(\mathbf{y}, \mathbf{y}_\star)$, $\mathcal{E}(\mathbf{u}, \mathbf{u}_\star)$, which arise from tensor approximations, we report also the convergence of the mean value of the velocity with the time grid refinement. The mean value is computed over all variables:

$$\langle v \rangle = \frac{\tau}{T} \sum_{k,k',i=1}^{J_v, J_v, n_t} M(k,k') D(i,i) y(i,1,k') \approx \int_D \int_\Omega \frac{1}{T} \int_0^T v(\mathbf{x}, \omega, t) dt d\mathbb{P}(\omega) d\mathbf{x}.$$

[5] The MATLAB and Python codes together with precomputed 2D matrices can be downloaded from https://mpim.iwww.mpg.de/3090660/sb_supplementary.zip, and the 3D matrices are located at https://mpim.iwww.mpg.de/3090674/sb_precomputed_3d.zip.

Table 1: 2D Stokes, comparison of spatial preconditioners

	$\boldsymbol{P}_1$		$\boldsymbol{P}_2$	
β	Iterations	CPU time	Iterations	CPU time
10^{-2}	1264	6197	194	2015
10^{-4}	738	3700	201	1968
10^{-6}	196	759	108	700
10^{-8}	163	465	72	322

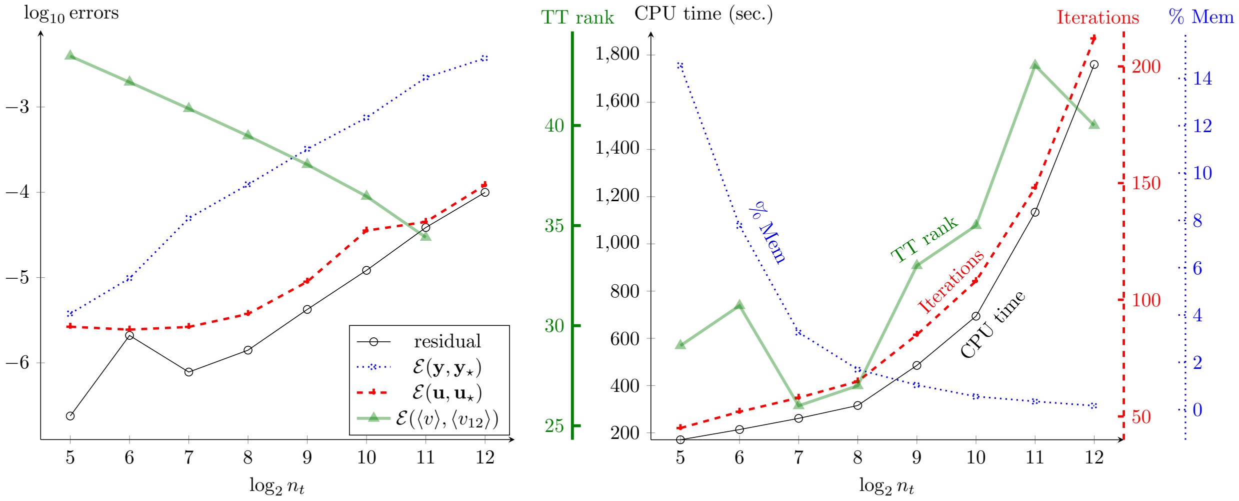


Figure 2: 2D Stokes, experiment with n_t. Left: Residual, errors w.r.t. the reference solutions, and the mean value error w.r.t. the time grid level. Right: CPU time, total number of iterations in spatial systems, memory compression ratio.

Note that $\mathbf{y}$ has the form $[\mathbf{v}, \mathbf{p}]$ w.r.t. the index k, so that the summation $k, k' = 1, \dots, J_v$ extracts only the velocity. The reference value $\langle v_{12} \rangle$ is computed on the grid $n_t = 2^{12}$. The distance from $\langle v \rangle$ decays proportionally to 2^{-n_t}, as expected for the Euler scheme.

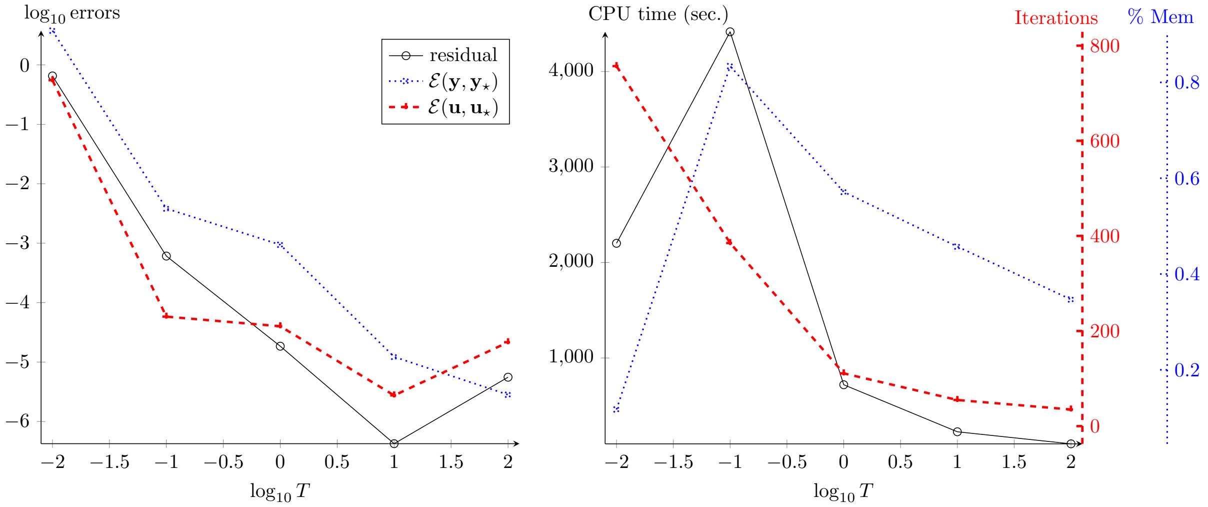
The tensor approximation errors grow proportionally to the grid size, since the matrix becomes more ill-conditioned, and one can select as an optimum the point where the discretization and solution errors intersect. To improve the overall accuracy, one has to take finer grids and decrease ε. Fortunately, the complexity increases mildly with refinement of these parameters. The dependence on ε is given in Fig. 7. Here, we notice that the CPU times and the numbers of iterations grow only as a small power of $\log n_t$, and the TT ranks scale even milder than log-linearly. This yields a sharp improvement of the memory compression ratio, which drops below 1% when n_t exceeds 2^9, and reaches 0.16% for $n_t = 2^{12}$.

The behavior of the CPU time is very close to the behavior of the iterations, hence the main reason for its growth is the deterioration of the preconditioner due to the rank-1 approximation (65). For more extreme parameters one might need a more robust preconditioner that would reflect the time derivative better.

6.3. *Experiment with T (Figure 3)*

Since the initial condition is zero, while the desired state is not for any time step, the time interval influences the model significantly. The smaller is the interval, the larger the force (in our terminology, control) that must be exerted to drive the system to the desired state. This is true not only for the physical behavior, but also for the computational efforts required to solve the system. For $T = 0.01$, the matrix becomes too ill-conditioned, and 800 iterations are not enough to compute the spatial TT block accurately enough. For larger T, both the error and the complexity decrease.

Figure 3: 2D Stokes, experiment with T. Left: Residual and errors w.r.t. the reference solutions. Right: CPU time, total number of iterations in spatial systems, memory compression ratio.



6.4. Experiment with β (Figure 4)

Although there are rigorous mathematical ways to estimate β for a given problem, such as the L-curve analysis [19] or the discrepancy principle [14], we do not follow them here for a couple of reasons. First, the value of β may be suggested by the physical considerations (i.e. the maximal force available). Second, we want to demonstrate robustness of our approach for as wide range as possible. Therefore, we vary β from 10^{-12} to 10^3.

We see that the errors are smaller for smaller β and stabilize at some levels when β increases. When β is small, the model reconstructs the deterministic Stokes solution quite accurately, as can be seen from the discrepancy $\mathcal{E}(\mathbf{v}, \bar{\mathbf{v}})$. In addition, we report the deviation of the mean solution at the final time from the desired state. This quantity is much smaller and less dependent on β than the global misfit: since the initial state is zero, the misfit in the first time steps will always be rather large, but in the latter steps the systems converges to the stationary solution. From the complexity figure, we see that the most difficult are the cases with intermediate β. The memory consumption increases with β, since the solution drives away from the rank-1 desired state.

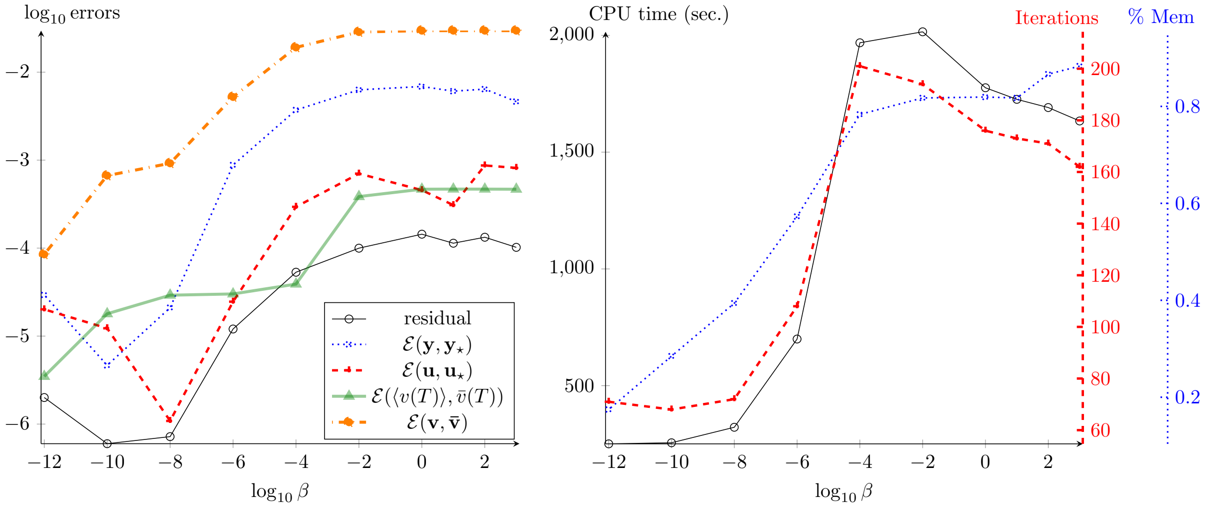
6.5. Experiment with α (Figure 5)

This parameter is supposed to penalize the standard deviation of the velocity. The (discrete) deviation is defined as follows,

$$\operatorname{std}(v) = \sqrt{\frac{\tau}{T} \sum_{k,k',i=1}^{J_v, J_v, n_t} \sum_{j=2}^{P} M(k,k') G_0(j,j) D(i,i) y^2(i,j,k')}.$$

In Fig. 5, we report the relative deviations for two variance parameters, $\nu_1 = 0.1$ and $\nu_1 = 0.9$. We see that in both cases the deviation decreases only marginally with α varying from 10^{-3} to 10^2. In particular, for $\nu_1 = 0.1$, it seems that the minimization of $\|\mathbf{v} - \bar{\mathbf{v}}\|$ with a deterministic $\bar{\mathbf{v}}$ delivers $\mathbf{v}$ with already a quasi-minimal variance as well. For larger ν_1, the deviation decreases more significantly. We could expect this effect to develop further for $\alpha > 10^3$. However, the preconditioner deteriorates rapidly with larger α. In particular, for $\alpha = 10^4$, the GMRES did not converge below the threshold $\varepsilon = 10^{-6}$ after 900 iterations. Further investigation is needed to develop reliable methods for damping the solution variance.

Figure 4: 2D Stokes, experiment with β. Left: Residual and errors w.r.t. the reference solutions, and the distance to the desired state. Right: CPU time, total number of iterations in spatial systems, memory compression ratio.



6.6. *Experiment with ν_1 (Figure 6)*

The ratio of maximal and minimal viscosities due to the stochasticity is $\nu_{\max}/\nu_{\min} = (1+\nu_1)/(1-\nu_1)$. If $\nu_1 \ll 1$, it grows almost linearly, $\nu_{\max}/\nu_{\min} \approx 1+2\nu_1$. If ν_1 is close to 1, the behavior becomes essentially nonlinear, e.g. for $\nu_1 = 0.9$ we have $\nu_{\max}/\nu_{\min} = 19$. The same can be seen in both error and complexity figures. The residuals and errors are almost stable for small ν_1, and the standard deviation grows linearly, while for $\nu_1 > 0.5$, all quantities grow faster. In particular, the distance to the desired state becomes larger since the Stokes system becomes more stiff. All three complexity indicators grow rapidly as $\nu_1 \to 1$ as well.

6.7. *Experiment with the tensor approximation tolerance (Figure 7)*

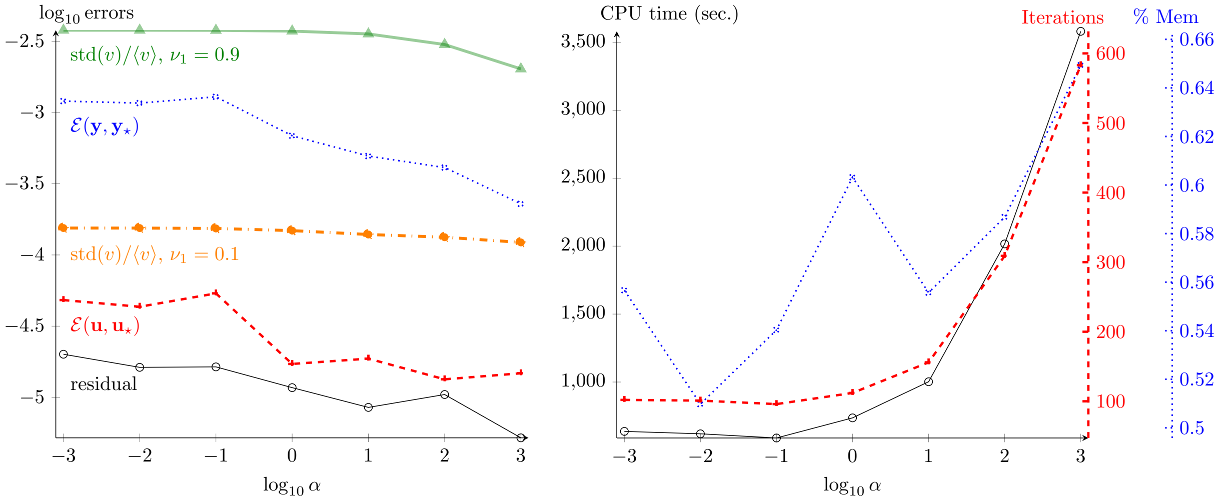
Here we confirm the consistency of the error estimate $\mathcal{E}(\mathbf{y}, \mathbf{y}_\star)$, see Remark 3. In experiments with positive definite matrices, it was observed that residuals and errors decay proportionally to ε. In this problem, this is only the case for ε between 10^{-4} and 10^{-5}. For smaller tolerances the residual and the control error are approximately proportional to $\varepsilon^{0.5}$. This may be caused by the indefiniteness of the problem and the pressure exclusion trick. We are unable to study their effects separately in the meantime, as the reduced systems (57) become degenerate if we try to apply the AMEn to an indefinite system directly.

6.8. *Experiment with n (Figure 8)*

The mesh generator in FEniCS is initialized with the number of mesh steps in one dimension n. The number of degrees of freedom for the pressure is $(n+1)^2$, since the pressure is discretized with linear elements, but together with the cubic mini elements for two components of the velocity, the total number of DoFs $J \approx 7n^2$. As in the time grid test, in addition to the residual and errors w.r.t. the reference solution, we investigate the error decay w.r.t. the grid refinement. The reference velocity for this test, $\langle v_8 \rangle$, is the mean value computed at the grid $n = 2^8$. The approximation error decays with the rate $n^{-1.4}$.

The most time-consuming stage in the scheme is the solution of the system for the spatial TT block. The sparsity of the spatial matrix allows its efficient factorization, such that the CPU time grows proportionally to n^2, i.e. linear w.r.t. the total number of spatial degrees of freedom. Interestingly, the number of iterations, TT ranks and the residual are smaller for larger n. This is due to the rank-1 approximation used for the factors of the preconditioner (32). For larger n, the norm of the discrete Laplace operator becomes larger, and the rank-1 term becomes a better approximation to the whole matrix.

Figure 5: 2D Stokes, experiment with α. Left: Residual and errors w.r.t. the reference solutions, and the relative standard deviation. Right: CPU time, total number of iterations in spatial systems, memory compression ratio.



6.9. Experiment with $\bar{K}_0$ (Figure 9)

Finally, we take $\bar{K}_0$ nonzero and investigate the Stokes-Brinkman model. For some reasons, with $n = 64$ and $\bar{K}_0 > 10^5$, the velocity matrix becomes indefinite. This might be due to the Gibbs phenomenon of the quadrature rule employed in FEniCS in computation of the stiffness matrix elements corresponding to the interface of $K_0(\mathbf{x})$. A detailed study would require interfering with the FEniCS source codes and this was not conducted. As a remedy, we perform this test with $n = 128$. This produces correct matrices up to $\bar{K}_0 = 10^6$.

We see that the scheme is quite robust in the considered range of the coefficient. The error estimates decay with increasing $\bar{K}_0$, since the system becomes closer to the Darcy model. The CPU time and the number of iterations show the chaotic behavior, but this fluctuation is only 10–20% compared to the average values.

6.10. 3D problem (Figure 10)

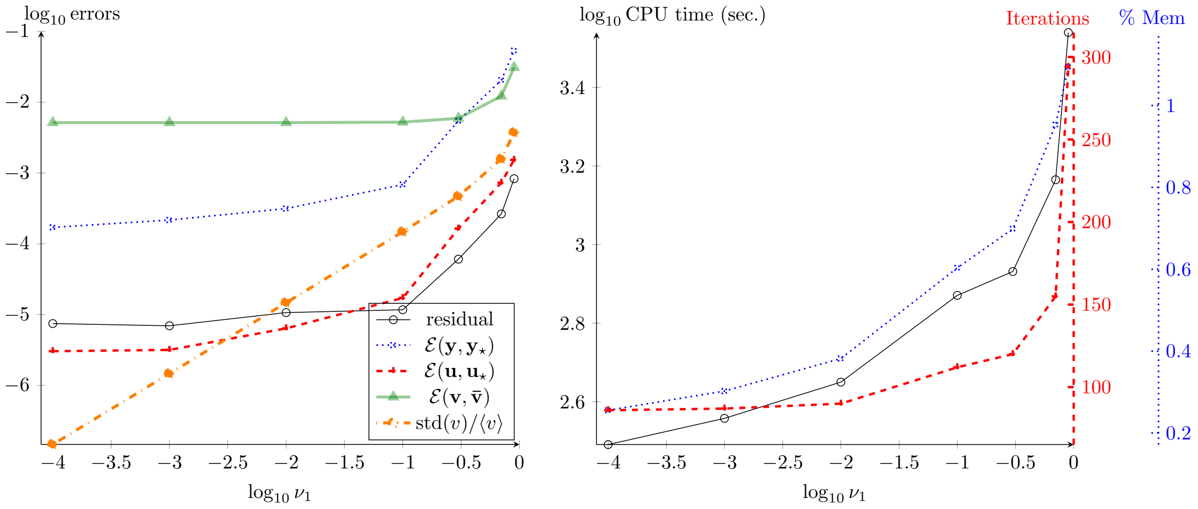
Finally, we demonstrate that our approach is suitable for larger 3D problems. We consider the three-dimensional Stokes-Brinkman problem on the domain $[0, 1] \times [0, 1] \times [0, 5]$ as constraints, with the coefficient

$$K_0(\mathbf{x}) = \begin{cases} 10^4, & (x_1 - 0.5)^2 + (x_2 - 0.5)^2 + (x_3 - 2.5)^2 \le 0.1^2, \\ 0, & \text{otherwise,} \end{cases}$$

and the inflow boundary condition $v_1|_{x_1=0} = x_2(1 - x_2)$ and zero conditions at other boundaries. The one-dimensional grid sizes are $16, 16, 32$ for x_1, x_2, x_3, respectively, which results in $J_v = 212355$ degrees of freedom for the velocity. Note that the full system size without tensor approximations would have been larger than 2 billions, which is intractable on our hardware by any means. Other parameters are the same as in the 2D tests except $\nu_1 = 0.01$ and $\varepsilon = 10^{-4}$.

Since the direct elimination is too expensive for such matrices, we used the commutator-based preconditioner (40) for the Schur complement in the Stokes matrices, and the velocity matrix was approximated by one V-cycle of the HSL MI20 algebraic multigrid [8]. The iterative method is two-level. First, we employed the block-triangular preconditioner for the KKT structure in the FGMRES method with unlimited number of iterations. Second, for all Stokes-like matrices in the preconditioning step, e.g. in (65), we used another FGMRES method with 50 iterations, preconditioned by (40) with the multigrid. That many inner iterations are needed because the commutator preconditioner deteriorates rapidly with the size of the porosity region. The KKT solver conducted in total 152 iterations, which took 148985 seconds of the CPU

Figure 6: 2D Stokes, experiment with ν_1. Left: Residual and errors w.r.t. the reference solutions, the relative standard deviation and the distance to the desired state. Right: CPU time, total number of iterations in spatial systems, memory compression ratio.



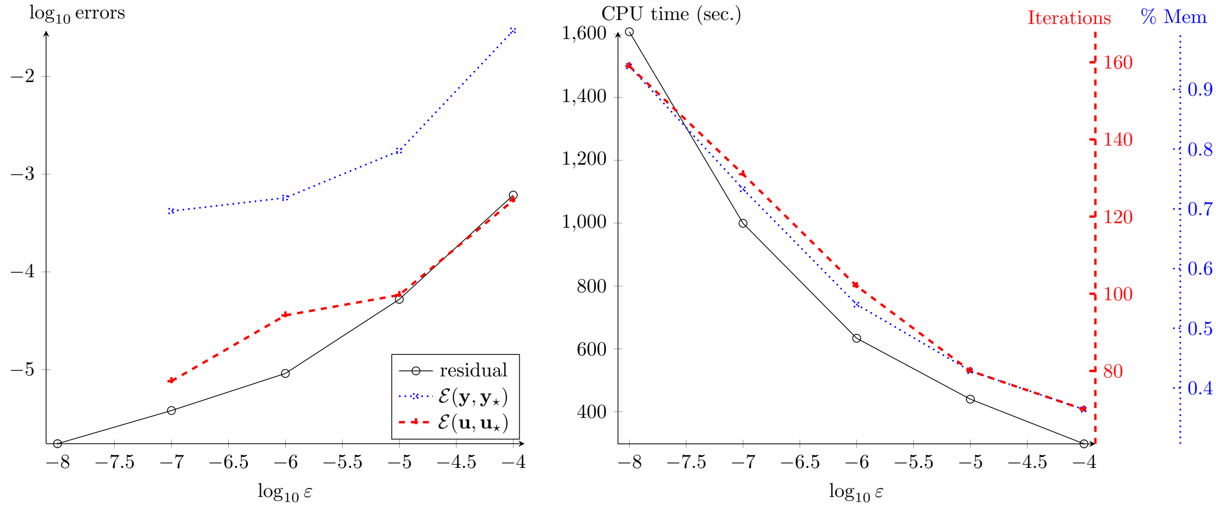
time. Nevertheless, the maximal TT rank of the solution is 8, so the TT format consumed only 0.2% of the memory required for the full solution. The final residual is $4.1 \cdot 10^{-4}$, and the misfit with the desired state $\mathcal{E}(\mathbf{v}, \bar{\mathbf{v}}) = 2.8 \cdot 10^{-3}$. The mean and the standard deviation of the solution at the final time are shown in Fig. 10. We notice a clear perturbation around the region with nonzero Brinkman coefficient. In particular, the largest deviations are attained at the interface, while in the homogeneous region the velocity is almost deterministic. The deviation of the pressure grows proportionally to the mean magnitude (note that the mean pressure is mostly negative, while the deviation is not, hence the color map in the right middle figure was reversed). The control exhibits a clear interface around the Brinkman region. Another interesting feature is that the deviation of the control is larger than its mean.

7. Conclusions and outlook

We have considered a low-rank solution to an optimal control problem constrained by Stokes-Brinkman with uncertain inputs. The discretized solution can be naturally indexed by three independent parameters, coming from the spatial, stochastic and time variables. Each of these parameters can vary in a considerable range, hence the straightforward storage of the solution consumes a vast amount of memory. By employing tensor product decomposition methods, we have reduced it by two–three orders of magnitude. However, the optimal control problem yields a saddle-point linear system, which requires a special treatment. We have extended the alternating minimal energy algorithm such that it preserves the saddle-point structure and solves this system robustly. Moreover, we have proposed a new Schur complement-based preconditioner which is free from auxiliary perturbations and provides smaller condition numbers of the preconditioned matrix. These techniques enabled the simulation of the stochastic Stokes-Brinkman optimization problem on a workstation.

Several directions of future research are possible. A natural extension is to apply our techniques to the nonlinear Navier-Stokes model. The preconditioner still needs an improvement, especially for large stochastic variance parameter ν_1, variance-penalizing regularization parameter α and many time steps. More complex models, such as those with uncertain boundary conditions and random domain, are also a challenging topic for future investigation.

Figure 7: 2D Stokes, experiment with ε. Left: Residual and errors w.r.t. the reference solutions. Right: CPU time, total number of iterations in spatial systems, memory compression ratio.



References

[1] R. Andreev and C. Tobler, *Multilevel preconditioning and low rank tensor iteration for space-time simultaneous discretizations of parabolic PDEs*, Numerical Linear Algebra with Applications, 22 (2015), pp. 317–337.

[2] H. Antil, M. Heinkenschloss, and R. H. W. Hoppe, *Domain decomposition and balanced truncation model reduction for shape optimization of the Stokes system*, Optimization Methods and Software, 26 (2011), pp. 643–669.

[3] M. J. Balajewicz, E. H. Dowell, and B. R. Noack, *Low-dimensional modelling of high-Reynolds-number shear flows incorporating constraints from the Navier-Stokes equation*, Journal of Fluid Mechanics, 729 (2013), pp. 285–308.

[4] J. Ballani and L. Grasedyck, *A projection method to solve linear systems in tensor format*, Numerical Linear Algebra with Applications, 20 (2013), pp. 27–43.

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

[6] M. Benzi, G. H. Golub, and J. Liesen, *Numerical solution of saddle point problems*, Acta Numerica, 14 (2005), pp. 1 – 137.

[7] P. Bochev and R. B. Lehoucq, *On the finite element solution of the pure Neumann problem*, SIAM Review, 47 (2005), pp. 50–66.

[8] J. Boyle, M. D. Mihajlovic, and J. A. Scott, *HSL MI20: An efficient AMG preconditioner for finite element problems in 3D*, International Journal for Numerical Methods in Engineering, 82 (2010), pp. 64–98.

[9] A. Caiazzo, V. John, and U Wilbrandt, *On classical iterative subdomain methods for the Stokes-Darcy problem*, Computational Geosciences, 18 (2014), pp. 711–728.

[10] S. V. Dolgov, *TT-GMRES: Solution to a linear system in the structured tensor format*, Russian Journal of Numerical Analysis and Mathematical Modelling, 28 (2013), pp. 149–172.

[11] S. V. Dolgov, B. N. Khoromskij, I. V. Oseledets, and D. V. Savostyanov, *Computation of extreme eigenvalues in higher dimensions using block tensor train format*, Computer Physics Communications, 185 (2014), pp. 1207–1216.

[12] S. V. Dolgov and D. V. Savostyanov, *Alternating minimal energy methods for linear systems in higher dimensions*, SIAM Journal on Scientific Computing, 36 (2014), pp. A2248–A2271.

[13] H. Elman, D. Silvester, and A. Wathen, *Finite Elements and Fast Iterative Solvers with Applications in Incompressible Fluid Dynamics*, Numerical Mathematics and Scientific Computation, Oxford University Press, New York, 2005.

[14] W. H. Engl, *Discrepancy principles for Tikhonov regularization of ill-posed problems leading to optimal convergence rates*, Journal of Optimization Theory and Applications, 52 (1987), pp. 209–215.

[15] O. G. Ernst, A. Mugler, H.-J. Starkloff, and E. Ullmann, *On the convergence of generalized polynomial chaos expansions*, ESAIM: Mathematical Modelling and Numerical Analysis, 46 (2012), pp. 317–339.

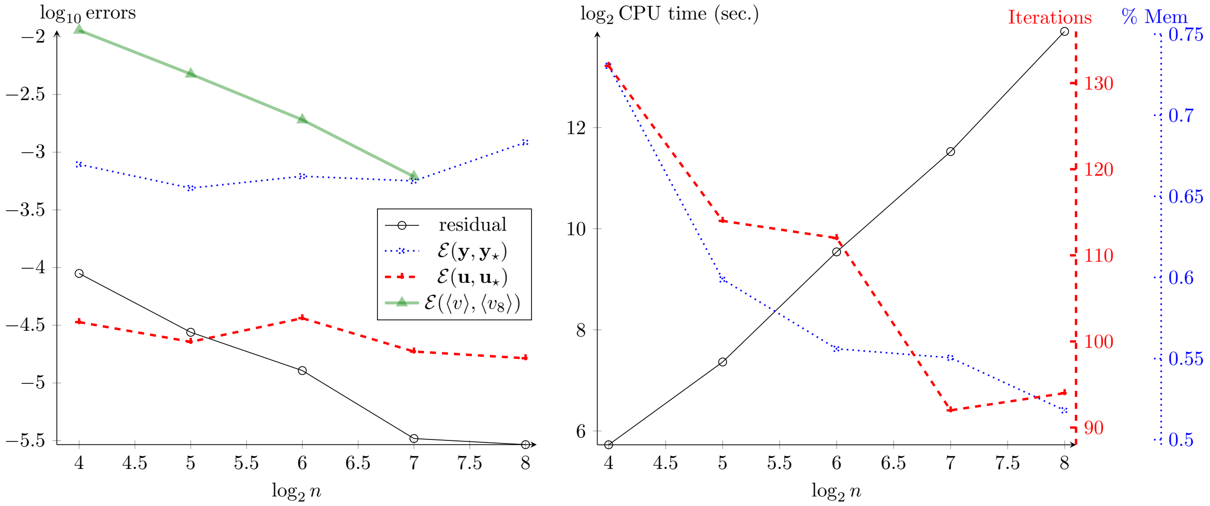
[16] R. G. Ghanem and R. M. Kruger, *Numerical solution of spectral stochastic finite element systems*, Computer Methods in Applied Mechanics and Engineering, 129 (2005), pp. 289–303.

[17] L. Grasedyck, D. Kressner, and C. Tobler, *A literature survey of low-rank tensor approximation techniques*, GAMM-Mitteilungen, 36 (2013), pp. 53–78.

[18] W. Hackbusch, *Tensor Spaces And Numerical Tensor Calculus*, Springer–Verlag, Berlin, 2012.

[19] P. C. Hansen and D. P. O'Leary, *The use of the L-curve in the regularization of discrete ill-posed problems*, SIAM Journal on Scientific Computing, 14 (1993), pp. 1487–1503.

Figure 8: 2D Stokes, experiment with n. Left: Residual and errors w.r.t. the reference solutions, and the mean value error w.r.t. the spatial grid level. Right: CPU time, total number of iterations in spatial systems, memory compression ratio.



[20] R. Herzog and E. Sachs, *Preconditioned conjugate gradient method for optimal control problems with control and state constraints*, SIAM Journal on Matrix Analysis and Applications, 31 (2010), pp. 2291 – 2317.

[21] M. Hintermüller, K. Ito, and K. Kunisch, *The primal-dual active set strategy as a semi-smooth Newton method*, SIAM Journal on Optimization, 13 (2002), pp. 865 – 888.

[22] S. Holtz, T. Rohwedder, and R. Schneider, *The alternating linear scheme for tensor optimization in the tensor train format*, SIAM Journal on Scientific Computing, 34 (2012), pp. A683–A713.

[23] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 1990.

[24] C. Hubig, I. P. McCulloch, U. Schollwöck, and F. A. Wolf, *Strictly single-site DMRG algorithm with subspace expansion*, Physical Review B, 91 (2015), p. 155115.

[25] K. Ito and K. Kunisch, *Semi-smooth Newton methods for state-constrained optimal control problems*, Systems and Control Letters, 50 (2003), pp. 221 – 228.

[26] E. Jeckelmann, *Dynamical density matrix renormalization group method*, Physical Review B, 66 (2002), p. 045114.

[27] Ch. Kanzow, *Inexact semi-smooth Newton methods for large-scale complementarity problems*, Optimization Methods and Software, 19 (2004), pp. 309 – 325.

[28] B. N. Khoromskij, *Tensor numerical methods for high-dimensional PDEs: Basic theory and initial applications*, arXiv preprint 1409.7970, 2014. ESAIM Proceedings, To appear.

[29] A. Klümper, A. Schadschneider, and J. Zittartz, *Matrix product ground states for one-dimensional spin-1 quantum antiferromagnets*, Europhysics Letters, 24 (1993), pp. 293–297.

[30] T. G. Kolda and B. W. Bader, *Tensor decompositions and applications*, SIAM Review, 51 (2009), pp. 455–500.

[31] D. Kressner, M. Steinlechner, and A. Uschmajew, *Low-rank tensor methods with subspace correction for symmetric eigenvalue problems*, SIAM Journal on Scientific Computing, 36 (2014), pp. A2346–A2368.

[32] D. Kressner and C. Tobler, *Krylov subspace methods for linear systems with tensor product structure*, SIAM Journal on Matrix Analysis and Applications, 31 (2010), pp. 1688–1714.

[33] D. Kressner and C. Tobler, *Low-rank tensor Krylov subspace methods for parametrized linear systems*, SIAM Journal on Matrix Analysis and Applications, 32 (2011), pp. 273–290.

[34] K. Kunisch and S. Volkwein, *Galerkin POD methods for parabolic problems*, Numerische Mathematik, 90 (2001), pp. 117–148.

[35] J. Larminie and A. Dicks, *Fuel cell systems explained*, vol. Second Edition, Wiley, 2013.

[36] D. Leykekhman, *Investigation of commutative properties of discontinuous Galerkin methods in PDE-constrained optimal control problems*, Journal of Scientific Computing, 53 (2012), pp. 483 – 511.

[37] A. Logg, K. A. Mardal, and G. N. Wells (Eds.), *Automated Solution of Differential Equations by the Finite Element Method*, Springer, 2012.

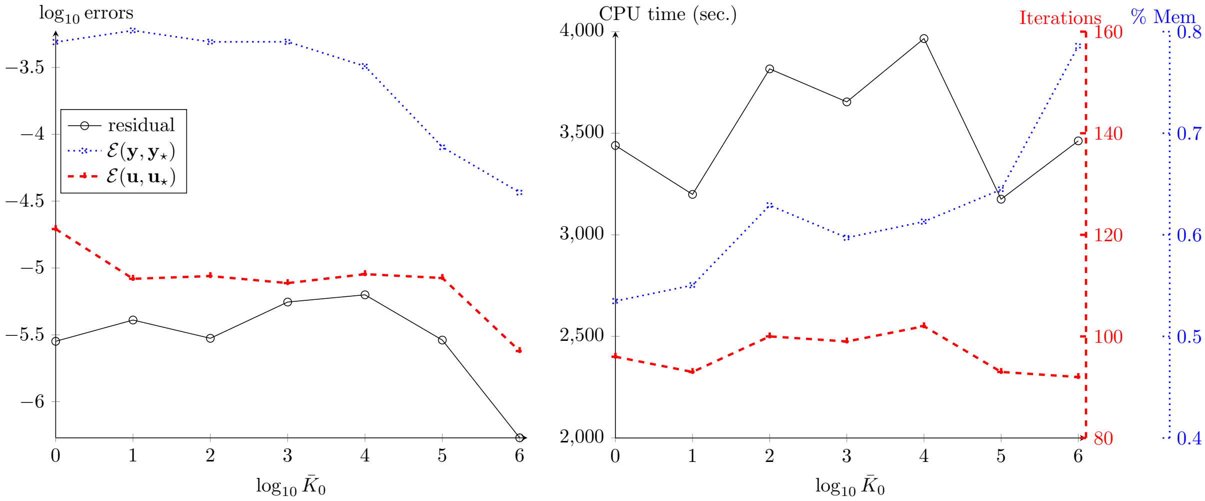
[38] G. J. Lord, C. E. Powell, and T. Shardlow, *An introduction to computational stochastic PDEs*, Cambridge University Press, 2014.

[39] A. Manzoni, A. Quarteroni, and G. Rozza, *Shape optimization for viscous flows by reduced basis methods and free-form deformation*, International Journal for Numerical Methods in Fluids, 70 (2012), pp. 646 – 670.

[40] K. A. Mardal, X. C. Tai, and R. Winther., *A mixed formulation for the Brinkman problem*, SIAM Journal on Numerical Analysis, 40 (2002), pp. 1605 – 1631.

[41] B. R. Noack, P. Papas, and P. A. Monkewitz, *The need for a pressure-term representation in empirical Galerkin models of incompressible shear flows*, Journal of Fluid Mechanics, 523 (2005), pp. 339–365.

Figure 9: 2D Stokes-Brinkmann, experiment with $\bar{K}_0$. Left: Residual and errors w.r.t. the reference solutions. Right: CPU time, total number of iterations in spatial systems, memory compression ratio.



[42] I. V. Oseledets, *Tensor-train decomposition*, SIAM Journal on Scientific Computing, 33 (2011), pp. 2295–2317.

[43] I. V. Oseledets, S. Dolgov, V. Kazeev, D. Savostyanov, O. Lebedeva, P. Zhlobich, T. Mach, and L. Song, *TT-Toolbox*. https://github.com/oseledets/TT-Toolbox.

[44] I. V. Oseledets and S. V. Dolgov, *Solution of linear systems and matrix inversion in the TT-format*, SIAM Journal on Scientific Computing, 34 (2012), pp. A2718–A2739.

[45] J. W. Pearson, M. Stoll, and A. Wathen, *Preconditioners for state constrained optimal control problems with Moreau-Yosida penalty function*, Numerical Linear Algebra with Applications, (2011), pp. 81 – 97.

[46] J. W. Pearson, M. Stoll, and A. J. Wathen, *Regularization-robust preconditioners for time-dependent PDE-constrained optimization problems*, SIAM Journal on Matrix Analysis and Applications, 33 (2012), pp. 1126–1152.

[47] J. W. Pearson and A. J. Wathen, *A new approximation of the Schur complement in preconditioners for PDE-constrained optimization*, Numerical Linear Algebra with Applications, 19 (2012), pp. 816 – 829.

[48] P. Popov, Y. Efendiev, and G. Qin, *Multiscale modeling and simulations of flows in naturally fractured Karst reservoirs*, Communications in Compuational Physics, 6 (2009), pp. 162 – 184.

[49] M. Porcelli, V. Simoncini, and M. Tani, *Preconditioning of active-set Newton methods for PDE-constrained optimal control problems*, SIAM Journal on Scientific Computing, 37 (2015), pp. S472 – S502.

[50] C. E. Powell and H. Elman, *Block-diagonal preconditioning for spectral stochastic finite-element systems*, IMA Journal of Numerical Analysis, 29 (2009), pp. 350–375.

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

[52] E. Rosseel and G. N. Wells, *Optimal control with stochastic PDE constraints and uncertain controls*, Computer Methods in Applied Mechanics and Engineering, 213-216 (2012), pp. 152–167.

[53] U. Schollwöck, *The density–matrix renormalization group*, Reviews of Modern Physics, 77 (2005), pp. 259–315.

[54] J. Sogn, *Stabilized finite element methods for the Brinkman equation on fitted and fictitious domains*, Master's Thesis, University of Oslo, 2014.

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

[56] M. Stoll and A. Wathen, *All-at-once solution of time-dependent Stokes control*, Journal of Computational Physics, 232 (2013), pp. 498–515.

[57] P. S. Vassilevski and U. Villa, *A block-diagonal algebraic multigrid preconditioner for the Brinkman problem*, SIAM Journal on Scientific Computing, 35 (2013), pp. S3 – S17.

[58] ———, *A mixed formulation for the Brinkman problem*, SIAM Journal on Numerical Analysis, 52 (2014), pp. 258 – 281.

[59] E. L. Wachspress, *The ADI Model Problem*, Springer, New York, 2013.

[60] S. R. White, *Density matrix algorithms for quantum renormalization groups*, Physical Review B, 48 (1993), pp. 10345–10356.

[61] ———, *Density matrix renormalization group algorithms with a single center site*, Physical Review B, 72 (2005), p. 180403.

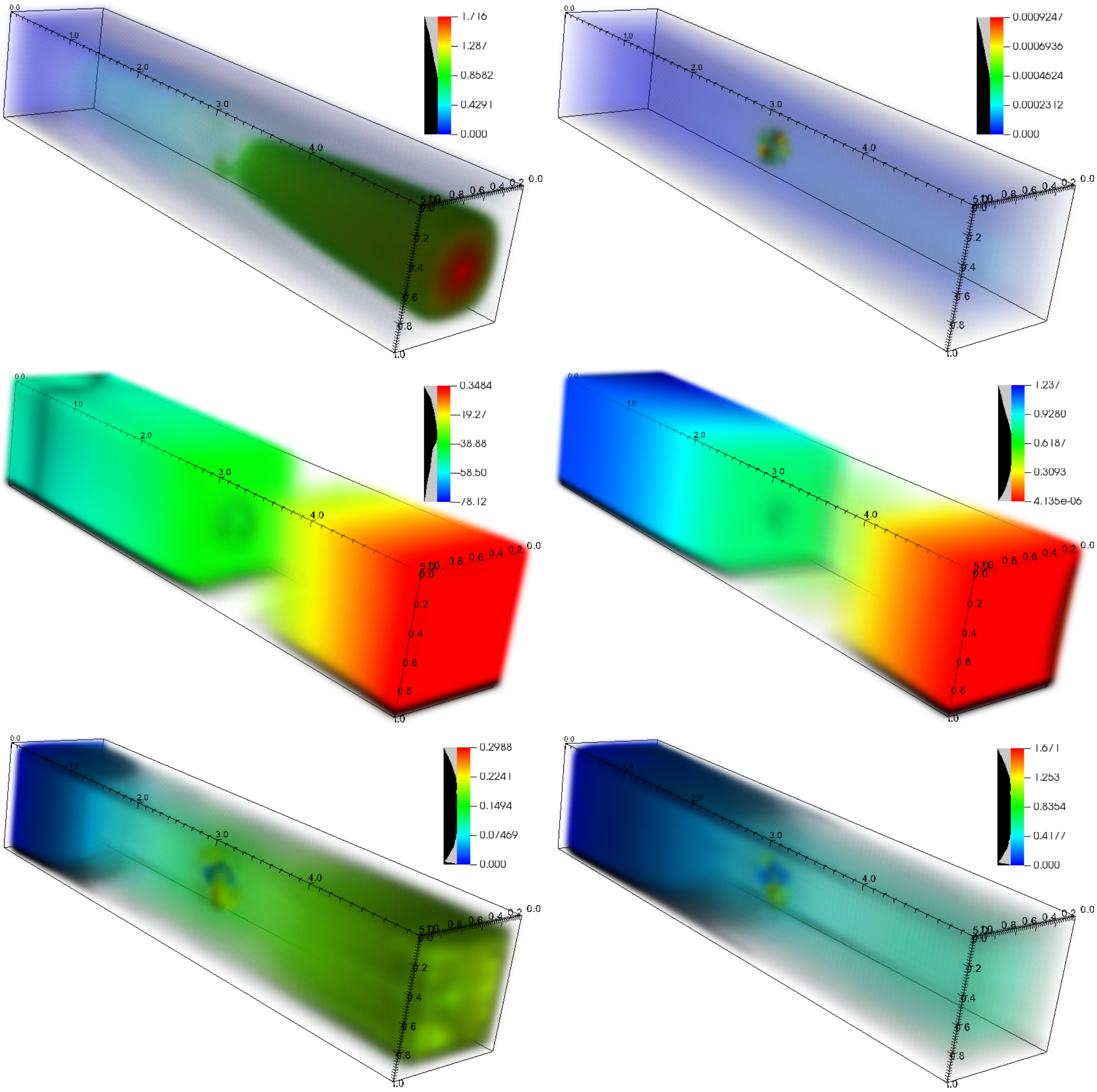
[62] X. P. Xie, J. C. Xu, and G. R. Xue, *Uniformly stable finite element methods for Darcy-Stokes-Brinkman models*, Journal of Computational Mathematics, 26 (2008), pp. 437 – 455.

[63] D. Xiu and J. Shen, *Efficient stochastic Galerkin methods for random diffusion*, Journal of Computational Physics, 228 (2009), pp. 266–281.

Algorithm 3 Block AMEn iteration for the inverse Stokes-Brinkman system (21).

```
Require: TT blocks of the matrix $\mathfrak{A}$, right-hand side $\mathbf{b}$, initial guesses $\mathbf{w}$ and $\mathbf{z}$ in the TT format (58).
Ensure: Improved approximations of the solution $\mathbf{w}$ and residual $\mathbf{z}$.
while not converged do
  for $m = 3, 2, 1$ do
    if $m = 3$ then
      Prepare and solve (61), using preconditioners (30) with (65) and possibly (41).
      Construct $\hat{\mathbf{w}}^{(3)} = \left[\hat{\mathbf{w}}^{(3)}(1), \hat{\mathbf{w}}^{(3)}(3), \hat{\mathbf{w}}^{(3)}(4)\right]$ without pressures.
      Correct the right-hand side by (64), (63).
    else
      Prepare and solve (59) with pressure parts removed from $\boldsymbol{K}, \boldsymbol{M}_1, \boldsymbol{N}$.
    end if
    if $m > 1$ then
      Use Alg. 2 for $\hat{\mathbf{w}}^{(m)}$ to move $l$ to $\mathbf{w}^{(m-1)}$.
      Compute the ALS step for the residual $\hat{\mathbf{z}}^{(m)} = \boldsymbol{Z}_m^\top \mathbf{Z}$ using (60).
      Use Alg. 2 for $\hat{\mathbf{z}}^{(m)}$ to move $l$ to $\mathbf{z}^{(m-1)}$.
      Compute the ALS step for the enrichment $\hat{\mathbf{z}}_w^{(m)} = \left(\boldsymbol{Z}^{<m} \otimes I_{n_m} \otimes (\boldsymbol{W}^{>m})^\top\right)^\top \mathbf{Z}$.
      Use Alg. 2 for $\hat{\mathbf{z}}_w^{(m)}$ to drop $l$, obtain $\mathbf{z}_w^{(m)}$.
      Enrich the solution $\hat{\mathbf{w}}^{(m-1)}(i_{m-1}, l) := \begin{bmatrix}\hat{\mathbf{w}}^{(m-1)}(i_{m-1}, l) & 0\end{bmatrix}$, $\mathbf{w}^{(m)}(i_m) := \begin{bmatrix}\mathbf{w}^{(m)}(i_m) \\ \mathbf{z}_w^{(m)}(i_m)\end{bmatrix}$.
      Make $\mathbf{w}^{(m)}$ right-orthogonal, see [42] and Def. 6.
    end if
  end for
  for $m = 1, 2, 3$ do
    if $m = 3$ then
      Prepare and solve (61), using preconditioners (30) with (65) and possibly (41).
      Construct $\hat{\mathbf{w}}^{(3)} = \left[\hat{\mathbf{w}}^{(3)}(1), \hat{\mathbf{w}}^{(3)}(3), \hat{\mathbf{w}}^{(3)}(4)\right]$ without pressures.
      Correct the right-hand side by (64), (63).
    else
      Prepare and solve (59) with pressure parts removed from $\boldsymbol{K}, \boldsymbol{M}_1, \boldsymbol{N}$.
    end if
    if $m < 3$ then
      Use Alg. 1 for $\hat{\mathbf{w}}^{(m)}$ to move $l$ to $\mathbf{w}^{(m+1)}$.
      Compute the ALS step for the residual $\hat{\mathbf{z}}^{(m)} = \boldsymbol{Z}_m^\top \mathbf{Z}$ using (60).
      Use Alg. 1 for $\hat{\mathbf{z}}^{(m)}$ to move $l$ to $\mathbf{z}^{(m+1)}$.
      Compute the ALS step for the enrichment $\hat{\mathbf{z}}_w^{(m)} = \left(\boldsymbol{W}^{<m} \otimes I_{n_m} \otimes (\boldsymbol{Z}^{>m})^\top\right)^\top \mathbf{Z}$.
      Use Alg. 1 for $\hat{\mathbf{z}}_w^{(m)}$ to drop $l$, obtain $\mathbf{z}_w^{(m)}$.
      Enrich the solution $\mathbf{w}^{(m)}(i_m) := \begin{bmatrix}\mathbf{w}^{(m)}(i_m) & \mathbf{z}_w^{(m)}(i_m)\end{bmatrix}$, $\hat{\mathbf{w}}^{(m+1)}(i_{m+1}, l) := \begin{bmatrix}\hat{\mathbf{w}}^{(m+1)}(i_{m+1}, l) \\ 0\end{bmatrix}$.
      Make $\mathbf{w}^{(m)}$ left-orthogonal, see [42] and Def. 6.
    end if
  end for
end while
```

Figure 10: 3D Stokes-Brinkman. Left: mean values at the last time step, right: standard deviations. Top: velocity, middle: pressure, bottom: control.


1.716
1.287
0.8582
0.4291
0.000
0.0009247
0.0006936
0.0004624
0.0002312
0.000
0.3484
-19.27
-38.88
-58.50
-78.12
1.237
0.9280
0.6187
0.3093
4.135e-06
0.2988
0.2241
0.1494
0.07469
0.000
1.671
1.253
0.8354
0.4177
0.000
0.0
1.0
2.0
3.0
4.0
5.0
0.8
0.6
0.4
0.2
0.0
0.2
0.4
0.6
0.8
1.0