Low-Rank Iterative Solvers for Large-Scale Stochastic Galerkin Linear Systems

Dissertation

zur Erlangung des akademischen Grades

doctor rerum naturalium

(Dr. rer. nat.)

von

Dr. rer. pol. Akwum Agwu Onwunta

geb. am **04.02.1979** in Akanu Ohafia, Nigeria

genehmigt durch die Fakultät für Mathematik

der Otto-von-Guericke-Universität Magdeburg

Gutachter: **Prof. Dr. Peter Benner**

Prof. Dr. Howard Elman

Eingereicht am: **23.03.2016**

Verteidigung am: **06.07.2016**

Abstract

Many problems in science and engineering are modelled using deterministic partial differential equations (PDEs). In practice, however, it is not always possible, for example, to measure some input data or parameters of a given model accurately; this leads to uncertainty in the simulations of the model. Hence, it is reasonable to represent such parameters in the model as random fields (or variables). This implies that the solution to the resulting stochastic model is necessarily also a random field. It is therefore of interest to quantify the influence of these uncertain parameters on the model.

In this dissertation, we study efficient low-rank iterative solvers for problems modelled via PDEs with random inputs. In particular, we employ the so-called *stochastic Galerkin finite element method* for the discretization of the considered problems. The resulting linear systems are usually very large with tensor product structure and, thus, solving them can be both time- and computer memory-consuming. Under certain assumptions, we first show, in the context of diffusion equations with stochastic coefficients, that the solution of such linear systems can be approximated with a vector of low tensor rank. We then solve the linear systems using low-rank preconditioned Krylov subspace solvers.

Next, we apply our low-rank approach to solve optimization problems governed by either steady-state or unsteady PDEs involving random coefficients and whose associated cost functionals are of tracking-type. Using diffusion equations and Stokes-Brinkman equations (each with random inputs) as constraint equations, we derive and analyze robust Schur complement-based preconditioners for solving the resulting optimality linear systems with all-at-once low-rank iterative solvers. In particular, for the model with stochastic diffusion constraint equations, we show furthermore that, with our proposed preconditioners, MINRES converges independently of all the spatial, stochastic and temporal parameters in the discretized models. Besides, in the case of the model with stochastic Stokes-Brinkman constraint equations, we develop a tensor-based low-rank algorithm to solve the optimality system. We provide extensive numerical experiments to illustrate that the low-rank scheme generally reduces the solution storage requirements by two – three orders of magnitude.

Zusammenfassung

Zahlreiche Probleme der Wissenschaft und des Ingenieurswesens werden mit deterministischen partiellen Differentialgleichungen (PDE) modelliert. In praktischen Anwendungen ist es dagegen nicht immer möglich die Eingangsdaten oder Parameter genau zu messen. Dies führt zu Unsicherheiten in der Simulation des Modells. Es ist somit nowendig, solche Parameter als Zufallsvariablen im Modell darzustellen. Dies impliziert, dass die Lösung des stochastischen Modells ebenfalls ein Zufallsfeld ist. Es ist somit von Interesse, den Einfluss der unsicheren Parameter im Modell zu quantifizieren.

In dieser Dissertation werden effiziente, niedrigrangige iterative Löser für Probleme untersucht, die mit PDEs mit zufälligen Eingangsdaten modelliert werden. Insbesondere findet die sogenannte *stochastische Galerkin finite Elemente Methode* Anwendung um die betrachteten Probleme zu diskretisieren. Die resultierenden linearen Systeme sind großskalig und weisen eine Tensorstruktur auf, sodass das Lösen sehr Zeit- und Speicherintensiv sein kann. Unter gewissen Annahmen wird im Kontext der Diffusionsgleichungen mit stochastischen Koeffizienten gezeigt, dass die Lösung solcher linearer Systeme mit einem Vektor von niedrigem Tensorrang approximiert werden kann. Die linearen Systeme werden dann mit niedrigrangigen vorkonditionierten Krylov-Unterraum Verfahren gelöst.

Des Weiteren wird der niedrigrangige Ansatz in Optimierungsproblemen angewandt, die durch stationäre oder zeitabhängige PDEs mit zufälligen Koeffizienten bestimmt sind, und deren Zielfunktionale vom Tracking-Typ sind. Für Diffusionsgleichungen oder Stokes-Brinkman Gleichungen (jeweils mit zufälligen Eingangsdaten) als Nebenbedingungen, werden robuste Schurkomplement-basierte Vorkonditionierer hergeleitet und analysiert, um die resultierenden Optimalitätssysteme mit 'all-at-once' niedrigrangigen iterativen Lösern zu lösen. Insbesondere für das Modell mit stochastischen Diffusionsgleichungen als Nebenbedingung, wird gezeigt, dass mit den vorgestellten Vorkonditionierern, das MINRES-Verfahren unabhängig von den räumlichen, zufälligen und zeitlichen Parametern im diskretisierten Modell konvergiert. Im Fall der Stokes-Brinkman Gleichungen als Nebenbedingung, wird ein tensorbasierter niedrigrangiger Algorithmus entwickelt, um das Optimalitätssystem zu lösen. Ausführliche numerische Experimente illustrieren, dass das niedrigrangige Schema den Speicherbedarf um zwei bis drei Größenordnungen reduziert.

Declaration of honour

I hereby declare that I produced this thesis without prohibited assistance and that all sources of information that were used in producing this thesis, including my own publications, have been clearly marked and referenced.

In particular, I have not wilfully:

- Fabricated data or ignored or removed undesired results.
- Misused statistical methods with the aim of drawing other conclusions than those warranted by the available data.
- Plagiarized data or publications or presented them in a distorted way.

I know that violations of copyright may lead to injunction and damage claims from the author or prosecution by the law enforcement authorities.

This work has not previously been submitted as a doctoral thesis in the same or a similar form in Germany or in any other country. It hast not previously been published as a whole. Furthermore, this dissertation is, in its entirety, unrelated to the one that I submitted for my Ph.D in Economics.

____

Location, date

____

Signature

Acknowledgements

First, I am greatly indebted to my supervisor, Prof. Dr. Peter Benner, for giving me the opportunity and creating an enabling environment for me to complete this dissertation at the Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg, Germany. His continual encouragement, invaluable pieces of advice, and unyielding support will definitely never go unnoticed. I have also been quite fortunate to be mentored by Dr. Martin Stoll. His phenomenal sense of humour and keen enthusiasm for my research were quite inspiring to me; I will cherish for a long time the atmosphere of friendliness and understanding which prevailed between us throughout the period of this work. I will not forget my gracious and unassuming colleague, Dr. Sergey Dolgov, for all the fruitful discussions we had concerning tensors and tensor-based algorithms. Quite frankly, I am very delighted to state unequivocally that Peter, Martin and Sergey were tremendously instrumental in ensuring that this work was a success. I commend them for always granting me audience each time I needed their attention despite their personal tight schedules. Besides, I have tapped quite a lot from their wealth of knowledge thanks to their remarkable willingness to collaborate with me.

I would also like to appreciate the financial support from the International Max Planck Research School (IMPRS) for Advanced Methods in Process and System Engineering (Magdeburg). The former coordinator of IMPRS, Dr. Juergen Koch, provided me with the necessary assistance that I needed to settle down when I started this work at MPI Magdeburg. For this and much more, I thank him so much. I will not forget to thank Prof. Dr. Dominique Thévenin for his words of encouragement to me in his capacity as a member of my PhD Advisory Committee (PAC). I am equally grateful to many other colleagues of mine – Dr. Lihong Feng, Dr. Xin Liang, Dr. Jens Saak, Dr. Sara Grundel, Dr. Jan Heiland, Yongjin Zhang, Petar Mlinaric, Jessica Bosch, Maximilian Behr, Cleophas Kweyu, Martin Hess, Diana Noatsch-Liebke, Janine Holzmann, Stephanie Geyer, etc – for contributing in no small measure in making me feel at home at MPI Magdeburg throughout the period of this work.

My family deserves a special place in my heart for their incredible patience and unflinching support. In particular, the love of my life (Ezinne) and our two 'sweet' boys

(Michael and Victor) have been wonderful to me. You guys rock my world! Words are indeed not enough to express my gratitude to you because you are simply amazing. I dedicate this work to you. Finally, I give all the glory to the Almighty God without whom this work would be impossible. For your amazing grace, steadfast love, and unfailing mercies upon my life, I want to say thank you Lord; you have been my help during these past years and you are and will remain my only hope in years to come.

Contents

Chapter 1

Introduction

1.1 General overview

Many problems in science and engineering involve uncertainties. The behaviours of these problems are widely predicted through the use of mathematical modelling and computer simulations. Such predictions are obtained by constructing mathematical models whose solutions describe the phenomenon of interest and then using computational methods to approximate the outputs of the models. A class of mathematical models that is usually of practical interest are partial differential equations (PDEs). Typical examples of PDEs include diffusion equations, diffusion-convection equations, Stokes equations, Navier-Stokes equations, etc., see e.g. [39, 87, 96]. In particular, if a system is modelled via a PDE, then the solution of the PDE often describes how the system behaves if some external forcing factor(s) acting on the system, as well as the intrinsic characteristics of the system (e.g. material properties), are known (or deterministic). In many applications, however, one may instead be interested in determining some unknown parameters in a model by comparing the predicted reaction with actual measurements ('inverse problems'); furthermore, one may wish to optimize certain parameters of a model in order to obtain a more desirable outcome, e.g. optimal shapes of airplane wings or the temperature control of a melting process. Such real-world problems can be mathematically formulated as *PDE-constrained optimization problems.* They are a subject of current research.

Optimization problems constrained by deterministic steady-state (or stationary) PDEs are ubiquitous in science and engineering; moreover, they are often computationally challenging. This is even more so if the constraints are deterministic time-dependent (or unsteady) PDEs, since one would then need 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 [104, 124]. Yet, more challenging than the afore-mentioned are problems governed by unsteady PDEs involving (countably many) parametric or uncertain inputs. This class of problems arises because the input parameters of the governing PDE (such as the diffusivity coefficient in the diffusion equation or the viscosity in the Navier-Stokes equation) in a given optimization model may be affected by *epistemic uncertainty*, that is, uncertainty due to incomplete knowledge that, in principle, could be remedied through additional measurements or improved measuring devices but for which such remedies are too costly or impractical to apply. For example, the highly heterogeneous subsurface properties in groundwater flow simulations can only be measured at relatively few locations, so at other locations these properties are subject to uncertainty. Incomplete knowledge can also be forced into a model due to lack of computational resources. For instance, although turbulent flows in computational fluid dynamics are generally thought of as being adequately modelled by the Navier-Stokes equations [24, 39, 51, 66, 102], in many practical situations one cannot use that model because the grids necessary to adequately approximate solutions are so fine that the resulting computational cost is prohibitive; in such cases, the unresolved scales are sometimes modelled via the addition of uncertainties into the model [84, 111]. In other situations, however, uncertainty may arise due to an inherent variability in the system that cannot be reduced by additional experimentation or improvements in measuring devices. Such uncertainties are referred to as being *aleatoric*. Examples include unexpected fluctuations induced in a flow field around an aircraft wing by wind gusts or on a structure by seismic vibrations [51].

The uncertainty encountered in the course of mathematical modeling usually propagates through the simulations of the model and quantifying its impacts on the solution of the model is frequently of great importance. By uncertainty quantification (UQ), we mean a variety of methodologies including uncertainty characterization, parameter estimation/model calibration, and error estimation. More precisely, the goal of UQ is to learn about the uncertainties in system outputs of interest, given information about the uncertainties in the system inputs [51]. Probability theory offers a natural framework to describe uncertainty, where all uncertain inputs are treated as random variables or, more generally, as random fields. Here, the approach relies on, for instance, probability density functions, expected values, variances, correlation functions, or statistical moments to provide characterizations of uncertainties. In fact, the classical approach is to model the random inputs as some idealized processes,[1] which can then be analyzed using elegant tools from classical Ito or Strantonovich calculus. In this case, the governing PDE in the optimization model is termed a *stochastic* PDE (see, for example, [72]). In recent times, there has been a growing interest in studying problems with more correlated random inputs ('colored noise') instead. In this dissertation, we consider probabilistic representations of uncertainties in problems modelled via PDEs with correlated random inputs (RPDEs)[2]. More specifically, these inputs are described by a finite-dimensional random vector, either because the problem itself can be described by a finite number of random variables or because inputs are modelled as truncated expansions of random fields. In particular, the latter is the case when we employ, for instance, a truncated *Karhunen-Lòeve expansion* to represent the random coefficients (or inputs) in the governing PDEs. In the context of correlated random inputs, the classical stochastic calculus, unfortunately, does not readily apply; therefore, other approaches are required.

The Monte Carlo method is probably the most natural and widely used technique to solve RPDEs [27]. This method generates ensembles of random realizations for the prescribed random inputs and utilizes repetitive deterministic solvers for each realization. Here, the deterministic PDE corresponding to each realization could be discretized using, for instance, the finite element method, the finite difference method or the finite volume method. The Monte Carlo method has been applied to many problems and its implementations are straightforward. It is (formally) independent of the dimensionality of the random space; that is, it is independent of the number of random variables used to characterize the random inputs. It, however, does not exploit the possible regularity that the solution might have with respect to the input parameters [128]; moreover, it exhibits a very slow convergence rate. In order to accelerate its convergence, several techniques have been developed: the multilevel Monte Carlo method [27], the quasi-Monte Carlo (QMC)

[1] Typically, these are white noises, such as Wiener processes, Poisson processes, etc.

[2] To economize notation, however, throughout this thesis we write RPDE for PDE with random inputs, whereas an optimization problem constrained by such a PDE is denoted by SOCP.

method [91], the Markov chain Monte Carlo method (MCMC) [44], the Latin hypercube sampling method [122], etc. Undoubtedly, these methods can improve the efficiency of the traditional Monte Carlo method. However, additional restrictions are imposed based on their specific designs and their applicability is limited.

Another class of methods which has received particular attention are the stochastic finite element methods (SFEM) [4, 5, 13, 59, 110]. These methods are often designed to retain the advantages of Monte Carlo simulations; in particular, they enable one to compute the full statistical characteristics of the solution, while reducing the simulation time. Two prominent variants of the SFEM are the stochastic collocation finite element method (SCFEM) [4] and the stochastic Galerkin finite element method (SGFEM)[110]. Both approaches transform the RPDE into a set of deterministic PDEs. The former samples the stochastic PDE in a set of collocation points and yields a separate deterministic PDE for each collocation point. One of the main advantages of this method is that it is nonintrusive in the sense that existing software for deterministic PDEs can readily be reused. Besides, this decoupled solution technique is highly parallelizable.

In contrast, the SGFEM applies spectral finite element theory to transform an RPDE into a set of deterministic PDEs. Because SGFEM is based on the projection of the residual onto the space of approximating polynomials, its accuracy is optimal in the L^2 sense. This can considerably reduce the number of required computations. However, the resulting system of deterministic PDEs generally exhibits coupling or decoupling between the spatial and parametric components. The coupling or decoupling of the linear systems depends primarily on the location of the randomness in the RPDEs [129]. If the RPDE is, for instance, the diffusion equation with a stochastic source term (or stochastic boundary conditions) and a deterministic diffusion coefficient, then we refer to the problem as additive noise or stochastic right-hand side problem. Here, the application of the SGFEM discretization then yields a block-diagonal global Galerkin matrix with multiple copies of one matrix of smaller size on the diagonal. This decoupled linear system may be solved using block iterative methods as discussed in [35]. On the other hand, a multiplicative noise or stochastic left-hand side problem, which is a more computationally challenging problem, occurs when we have a stochastic diffusion coefficient in the PDE, regardless of the type of source term and boundary conditions, see e.g. [110]. Depending on the type of the random diffusion coefficient and the choice of the stochastic discretization, the global Galerkin system allows decoupling only in rare cases. If and when there is decoupling, then the task is to solve a sequence of independent linear systems and this can be handled with relative ease. In this thesis, we focus mainly on the more difficult case in which the SGFEM discretization yields prohibitively high dimensional coupled deterministic PDEs, and hence large-scale tensor-product algebraic systems. These systems require specialized solution methods, and solvers for the original deterministic problem cannot be straightforwardly reused. Nevertheless, the SGFEM approach exhibits fast convergence and other nice properties [5, 110].

Regardless of the progress made so far in developing stochastic finite element-based solvers for RPDEs, it is pertinent to remark here that SOCPs have, in our opinion, not yet received adequate attention. Some of the papers on these problems include [20, 50, 59, 75, 115, 128]. While [50, 59] study the existence and the uniqueness of solutions to control problems constrained by elliptic RPDEs, the emphasis in [20, 75, 128] is on solvers based on stochastic collocation methods for SOCPs. Rosseel and Wells in [115] apply a one-shot method with both SGFEM and collocation approaches to optimal control problems constrained by elliptic RPDEs. One of their findings is that SGFEM generally exhibits superior performance compared to the stochastic collocation method, in the sense that, unlike SGFEM, the non-intrusivity property of the stochastic collocation method is lost when moments of the state variable appear in the cost functional, or when the control function is a deterministic function.

The fast convergence and other nice properties exhibited by SGFEM notwithstanding, the resulting high dimensional tensor-product algebraic systems associated with this intrusive approach unfortunately limits its attractiveness in the sense that solving the systems can be quite computer memory-consuming. Thus, for it to compete favourably with the sampling-based approaches, there is the need to develop solvers which are particularly efficient in the reduction of memory requirements of these vast linear systems during practical simulations of either stochastic forward problems or control problems. This is indeed the main focus of this dissertation. More specifically, in order to break this inherent *curse of dimensionality*, we propose and study in detail efficient low-rank preconditioned Krylov subspace methods for solving the resulting large-scale stochastic Galerkin linear systems.

For the numerical simulation of the SOCPs considered in this thesis, we assume that the state, the control and the target (or the desired state) are analytic functions depending on the uncertain parameters [94]. However, we note here that, as pointed out in [115], problems in which the control is modelled as an unknown stochastic function constitute stochastic inverse problems and they are different from those with deterministic controls [146]. In the former, the stochastic properties of the control are unknown but will be computed. So, in most cases (as we assume in this work), the mean of the computed stochastic control could be considered as optimal. Depending on the application, the mean may not, in general, be the sought optimal control, though. Besides, quantifying the uncertainty in the system response might require additional computational challenges.

1.2 Contributions and outline of the thesis

Recall that our ultimate goal in this thesis is to develop efficient low-rank iterative solvers for linear systems resulting from SGFEM discretizations. With a view to achieving this goal, we first give in Section 1.3 the basic definitions and notation on which we shall rely in the rest of this dissertation. Using the diffusion equation as a prototypical example, we proceed to discuss in Chapter 2 a finite element-based framework for the numerical simulation of problems modelled by steady-state and unsteady PDEs with uncertain inputs. Here, we specifically elaborate on discretization with SGFEM. Note that most of the materials in this chapter can be found in already existing literature. In Chapter 3, we present the first novel contributions of this dissertation, which are based mainly on [13]. More precisely, after proving the existence of a low-rank solution to the stochastic Galerkin linear system corresponding to an unsteady forward problem, we then analyze low-rank Krylov subspace solvers for the system. Furthermore, we provide numerical experiments to demonstrate that these low-rank solvers are effective in reducing the memory and the computational time requirements, especially when the fluctuations in the random data are not too large relative to their mean values.

Next, equipped with the low-rank concepts discussed in Chapter 3, we proceed in a natural way to Chapter 4 to address a relatively more challenging task – the development of efficient solvers for the saddle point linear systems resulting from the SGFEM discretization of large-scale optimization problems constrained by either stationary or unsteady diffusion equations with random inputs. Here, inspired by a state-of-the-art preconditioning strategy employed in the deterministic framework [104, 105, 124], we specifically derive and analyze robust Schur complement-based block-diagonal preconditioners which we use in conjunction with a low-rank version of the minimal residual method (MINRES) for the efficient solution of the optimality systems. More importantly, we also show here that, with our proposed preconditioners, the convergence of MINRES is independent of all the spatial, stochastic, and temporal parameters in the discretized models. Besides, we demonstrate numerically the robustness of our proposed preconditioners. Let us put our work into perspective here. Various preconditioned Krylov subspace methods for an accelerated solution of optimality systems in PDE-constrained optimization are considered in, for example, [17, 34, 39, 62, 88, 104, 105, 124]. Nevertheless, these contributions are based entirely on deterministic problems and, thus, the resulting saddle point systems are generally smaller in dimension and have fewer number of discretization parameters than the stochastic problems considered in this work. In fact, it should be borne in mind that research on the solution of SOCPs via SGFEM is still in its infancy and deemed computationally challenging; in particular, numerical results on the subject can hardly be found in the literature. To the best of our knowledge, this contribution is, to date, the first detailed study on the preconditioning of the considered class of stochastic problems and, therefore, the approach presented here evidently pushes the research frontier towards larger and more challenging problems. Most of the materials in Chapter 4 are based on the paper [14].

Finally, Chapter 5 – which is based essentially on [11] – studies the most challenging of the problems considered in this thesis, namely, an unsteady Stokes-Brinkman optimal control problem with uncertain inputs. 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 both cavity and porous media. As pointed out in [133], 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 [2, 25]. The Brinkman model is typically applied in oil reservoir modeling [108], computational fuel cell dynamics [80, 141] or biomedical engineering [120].

The study of finite element-based solvers for the Brinkman model has, on the one hand, attracted much attention recently [108, 133, 134, 141]. 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 [133], but also the construction of stable finite element discretizations [86, 141]. On the other hand, the numerical simulation of optimization problems constrained by unsteady Brinkman equations has, to the best our knowledge, not yet received any attention as at the time of writing this thesis. Therefore, in this chapter, one of our major goals is specifically to study the preconditioning of a linear system resulting from the discretization of the optimal control problem constrained by the unsteady Stokes-Brinkman flow involving random data.

As expected, the discretization of the model using SGFEM also leads to prohibitively high dimensional saddle point optimality systems with Kronecker (tensor) 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 [97], also known as the Matrix Product States [73]. The tensor decomposition concept is similar to low-rank model reduction techniques, for example, the proper orthogonal decomposition (POD) [79]. However, 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 [57, 118, 138]. However, existing alternating solvers for linear systems require a positive definite matrix. Other novel contributions of this thesis are the extension and adaptation of these algorithms to solve the optimality systems resulting from the unsteady Stokes-Brinkman SOCP. The performance of our approach is illustrated with extensive numerical experiments based on two- and three-dimensional examples. The developed Tensor Train scheme reduces the solution storage by two – three orders of magnitude. We refer to [49, 52] for a more detailed overview of tensor methods.

1.3 Preliminaries and notation

For the reader's convenience, we recall here some important concepts, as well as fix the basic notation on which we shall extensively rely in the rest of this dissertation. Our point of departure is the following definition.

Definition 1.1. *Let $X = [x_1, \ldots, x_m] \in \mathbb{R}^{n\times m}$ and $Y \in \mathbb{R}^{p\times q}$. Then, the Kronecker product $X \otimes Y \in \mathbb{R}^{np\times mq}$ and the vectorization operator* $\text{vec} : \mathbb{R}^{n\times m} \to \mathbb{R}^{nm}$ *are defined, respectively, by*

$$A := X \otimes Y = \begin{bmatrix} x_{11}Y & \ldots & x_{1m}Y \\ \vdots & \ddots & \vdots \\ x_{1n}Y & \ldots & x_{nm}Y \end{bmatrix}, \quad \text{vec}(X) = \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix}. \tag{1.1}$$

It follows from (1.1) that the vec operator essentially reshapes a matrix into a column vector by stacking the columns of the matrix. In MATLAB notation, for example, we have `vec(X)=reshape(X,n*m,1)`. More precisely, we consider the vec operator as a vector space isomorphism and denote its inverse by $\text{vec}^{-1} : \mathbb{R}^{nm} \to \mathbb{R}^{n\times m}$. Kronecker product and vec operators exhibit the following properties, see e.g., [28]:

$$\text{vec}(AXB) = (B^T \otimes A)\text{vec}(X), \tag{1.2}$$

$$(A \otimes B)(C \otimes D) = AC \otimes BD. \tag{1.3}$$

We will also need the tensor rank of a vectorized matrix, see e.g., [48].

Definition 1.2. *Let $X \in \mathbb{R}^{n\times n}$ and $\mathbf{x} = \text{vec}(X) \in \mathbb{R}^{n^2}$. Then, the tensor rank of $\mathbf{x}$ is the smallest $r \in \mathbb{Z}_+$ such that*

$$\mathbf{x} = \sum_{i=1}^{r} u_i \otimes v_i, \tag{1.4}$$

where $u_i, v_i \in \mathbb{R}^n$. In particular, the tensor rank *of the vector $\mathbf{x}$ coincides with the rank of the matrix X.*

Next, we introduce, for $d \in \mathbb{N}$, the following multi-index notation

$$\overline{i_1 i_2 \cdots i_d} = i_1 + (i_2 - 1)n_1 + \cdots + (i_d - 1)n_1 n_2 n_3 \cdots n_{d-1}, \tag{1.5}$$

where $n_k = \# i_k,\ k = 1, \ldots, d$. Note then, in particular, that if $X = [X(i,j)]_{i,j=1}^{n,m}$ and $Y = [Y(k,\ell)]_{k,\ell=1}^{p,q}$, from (1.1), one has

$$A(\overline{ik}, \overline{j\ell}) = X(i,j)Y(k,\ell), \tag{1.6}$$

where $\overline{ik} = (i-1)p + k = 1, \ldots, np$, and $\overline{j\ell} = (j-1)q + \ell = 1, \ldots, mq$. Similarly, from (1.4), we have

$$\mathbf{x} = \sum_{i=1}^{r} u_i \otimes v_i \qquad \Leftrightarrow \qquad \mathbf{x}(\overline{jk}) = \sum_{i=1}^{r} u_i(j) v_i(k), \tag{1.7}$$

where $j, k = 1, \ldots, n$, and $\overline{jk} = j + (k-1)n = 1, \ldots, n^2$.

We next introduce the necessary spaces and stochastic concepts that we will use in the context of our stochastic discretizations. That said, the triplet $(\Omega, \mathfrak{F}, \mathbb{P})$ denotes a complete probability space, where Ω is the set of elementary events, $\mathfrak{F} \subset 2^{\Omega}$ is a σ-algebra on Ω and $\mathbb{P} : \mathfrak{F} \to [0,1]$ is an appropriate probability measure. We write

$$L^2(\Omega) := L^2(\Omega, \mathfrak{F}, \mathbb{P}) = \left\{ f : \Omega \to \mathbb{R} \text{ measurable}, \ \int_{\Omega} f^2(\omega)\, d\mathbb{P}(\omega) < +\infty \right\},$$

to denote the space of all square integrable random variables, which is a Hilbert space equipped with the inner product

$$\langle f, g \rangle = \int_{\Omega} f(\omega) g(\omega)\, d\mathbb{P}(\omega), \quad f, g \in L^2(\Omega).$$

Let $\mathcal{D} \subset \mathbb{R}^d$ with $d \in \{1, 2, 3\}$, be a bounded open set with Lipschitz boundary $\partial\mathcal{D}$.

Definition 1.3. *A mapping $z : \mathcal{D} \times \Omega \to \mathbb{R}$ is called a random field if for each fixed $\mathbf{x} \in \mathcal{D}$, $z(\mathbf{x}, \cdot)$ is a random variable with respect to $(\Omega, \mathfrak{F}, \mathbb{P})$.*

We denote the mean and the covariance of z, respectively, by

$$\mathbb{E}[z](\mathbf{x}) := \langle z(\mathbf{x}, \cdot) \rangle = \int_{\Omega} z(\mathbf{x}, \omega)\, d\mathbb{P}(\omega), \quad \mathbf{x} \in \mathcal{D}, \tag{1.8}$$

and

$$C_z(\mathbf{x}, \mathbf{y}) := \langle (z(\mathbf{x}, \cdot) - \mathbb{E}[z](\mathbf{x}))(z(\mathbf{y}, \cdot) - \mathbb{E}[z](\mathbf{y})) \rangle, \quad \mathbf{x}, \mathbf{y} \in \mathcal{D}. \tag{1.9}$$

The standard deviation of z $\left(\text{std}(z) = \sqrt{\mathbb{V}\text{ar}(z)}\right)$ is given by

$$\text{std}(z) := \sigma_z = \left[\int_{\Omega} (z - \mathbb{E}(z))^2 \, d\mathbb{P}(\omega) \right]^{\frac{1}{2}}. \tag{1.10}$$

Next, we denote by $H^k(\mathcal{D})$ the Sobolev space of functions on $\mathcal{D}$ whose derivatives up to order k are square-integrable. In particular, the variational space $H_0^1(\mathcal{D}) \subset H^1(\mathcal{D})$ is defined by $H_0^1(\mathcal{D}) = \left\{ v \in H^1(\mathcal{D}) : v|_{\partial \mathcal{D}} = 0 \right\}$. Note that the dual space of $H_0^1(\mathcal{D})$, i.e., the space of all bounded linear functionals $f : H_0^1(\mathcal{D}) \to \mathbb{R}$, is denoted by

$$H^{-1}(\mathcal{D}) := H_0^1(\mathcal{D})',$$

with norm

$$||f||_{-1} := \sup_{v \in H_0^1(\mathcal{D})} \frac{|f(v)|}{||v||_1}, \;\; f \in H^{-1}(\mathcal{D}).$$

We define the Hilbert space $L^2(\mathcal{D}) \otimes L^2(\Omega)$ of second-order random fields by

$$L^2(\mathcal{D}) \otimes L^2(\Omega) = \left\{ v : \mathcal{D} \otimes \Omega \to \mathbb{R} \text{ measurable}, \int_{\Omega} \int_{\mathcal{D}} |v|^2 \, d\mathbf{x} d\mathbb{P}(\omega) < +\infty \right\},$$

and it is endowed with the norm

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

The tensor product spaces $H^1(\mathcal{D}) \otimes L^2(\Omega)$ and $H_0^1(\mathcal{D}) \otimes L^2(\Omega)$ can be defined analogously [5]. For a Hilbert space H of functions on $\mathcal{D}$ and time interval $[0, T]$, we write $L^2(0, T; H)$ for the tensor product space $L^2([0, T]) \otimes H$. However, in particular, we write $L^2(0, T; \mathcal{D})$ for $L^2(0, T; L^2(\mathcal{D}))$; we will be using these last two notations interchangeably.

Chapter 2

Numerical methods for PDEs with uncertain parameters

Consider the deterministic second-order elliptic boundary value problem

$$\begin{aligned} -\nabla \cdot (a(\mathbf{x})\nabla y(\mathbf{x})) &= u(\mathbf{x}), \quad \text{in } \mathcal{D}, \\ y(\mathbf{x}) &= 0, \qquad \text{on } \partial\mathcal{D}, \end{aligned} \tag{2.1}$$

with the source function $u \in L^2(\mathcal{D})$. The operator ∇ implies differentiation with respect to the physical coordinate $\mathbf{x}$. The model (2.1) arises, for instance, in the context of groundwater flow modeling, where the variable y is called the pressure head [27]. The parameter a is the hydraulic conductivity tensor (or diffusivity coefficient) and it characterizes how easily water can flow through the rock under a given pressure gradient. We note here that an alternative formulation of the groundwater flow model (2.1) can be obtained via the classical *Darcy's law* coupled with an incompressibility condition (see e.g. [27, 129]):

$$v + a\nabla y = g, \quad \nabla \cdot v = 0, \quad \text{in } \mathcal{D}, \tag{2.2}$$

where $u := -\nabla \cdot g$ and the quantity v represents the filtration velocity (or Darcy flux). For our purposes in this dissertation, however, we will stick to the formulation (2.1).

The problem of assessing the safety of a potential deep geological repository for radioactive waste provides a particularly good example where the model (2.1) is applied. As aptly pointed out in [27], any radionuclides leaking from such a repository could be transported back to the human environment by groundwater flowing through the rocks beneath the earth's surface and very long timescales are involved. Hence, dedicated modelling and simulation are essential in evaluating the performance of the repository.

In practice, the diffusivity coefficient a is available only at a limited number of spatial locations, but its values are required at all points of the computational domain for the simulation[1]. This fact is the primary source of uncertainty in groundwater flow calculations. Thus, understanding and quantifying the impact of this uncertainty on predictions of radionuclide transport is particularly essential for reliable repository safety assessments [27]. As already noted before, a convenient way to characterize the uncertainty in the problem consists in incorporating the uncertain diffusivity coefficient as a random variable or space-varying random field. This, in turn, implies that the solution to the resulting stochastic model is necessarily also a random field; that is, we assume that $a = a(\mathbf{x}, \omega)$ is a family of random variables $a(\mathbf{x}, \cdot)$ defined on $L^2(\Omega)$ with index variable $\mathbf{x} \in \mathcal{D}$. This assumption immediately transforms (2.1) to the following formulation. Find a function $y : \mathcal{D} \times \Omega \to \mathbb{R}$ such that, $\mathbb{P}$-almost surely, the following linear elliptic diffusion equation holds

$$\begin{aligned} -\nabla \cdot (a(\mathbf{x}, \omega) \nabla y(\mathbf{x}, \omega)) &= u(\mathbf{x}), \quad \text{in } \mathcal{D} \times \Omega, \\ y(\mathbf{x}, \omega) &= 0, \qquad \text{on } \partial\mathcal{D} \times \Omega, \end{aligned} \tag{2.3}$$

where, we assume that $u \in L^2(\mathcal{D})$ and that there exist positive constants $a_{\min}$ and $a_{\max}$ such that

$$\mathbb{P}\left(\omega \in \Omega : a(\mathbf{x}, \omega) \in [a_{\min}, a_{\max}], \forall \mathbf{x} \in \mathcal{D}\right) = 1. \tag{2.4}$$

For the weak formulation of the stochastic forward problem (2.3), we essentially seek $y \in \mathcal{V} := H_0^1(\mathcal{D}) \otimes L^2(\Omega)$ such that, $\mathbb{P}$-almost surely,

$$\mathfrak{B}(y, v) = \ell(u, v), \quad \forall v \in \mathcal{V}, \tag{2.5}$$

[1] Sometimes, the source term u is also not known exactly.

where the bilinear form $\mathfrak{B}(\cdot,\cdot)$ is given by

$$\mathfrak{B}(y,v) = \int_\Omega \int_\mathcal{D} a(\mathbf{x},\omega)\nabla y(\mathbf{x},\omega)\cdot\nabla v(\mathbf{x},\omega)\; d\mathbf{x}d\mathbb{P}(\omega),\; v,y\in\mathcal{V}, \tag{2.6}$$

and

$$\ell(u,v) = \int_\Omega \int_\mathcal{D} u(\mathbf{x})v(\mathbf{x},\omega)\; d\mathbf{x}d\mathbb{P}(\omega),\;\; v,u\in\mathcal{V}. \tag{2.7}$$

The following existence and uniqueness result of the solution y to (2.3) proved in, for instance, [59] follows from the Lax-Milgram Lemma [23].

Theorem 2.1. *Under the assumption (2.4), there exists a unique solution $y\in\mathcal{V}$ such that, $\mathbb{P}$-almost surely, (2.5) holds.*

In what follows, we discuss the different popular methods in the literature for discretizing PDEs with uncertain inputs, bearing in mind the stochastic forward problem (2.3) as our prototypical model. However, we note here that the first step to solve PDEs with uncertain inputs consists in representing the random fields in the model with a finite number of random variables. Thus, we proceed to discuss our random field representation strategy first before delving into the discretization methods.

2.1 Representation of random fields

Suppose that we have a random field $z:\mathcal{D}\times\Omega\to\mathbb{R}$ with known continuous covariance function $C_z(\mathbf{x},\mathbf{y})$. Then, $z(\mathbf{x},\omega)$ admits a proper orthogonal decomposition or Karhunen-Lòeve expansion (KLE)

$$z(\mathbf{x},\omega) = \mathbb{E}[z](\mathbf{x}) + \sigma_z \sum_{i=1}^{\infty} \sqrt{\lambda_i}\vartheta_i(\mathbf{x})\xi_i(\omega),\;\; N\in\mathbb{N}, \tag{2.8}$$

where σ_z is the standard deviation of z and the random variables $\{\xi_i\}_{i=1}^N$ are centered, normalized and mutually uncorrelated; see e.g., [110]. Here, $C_z(\mathbf{x},\mathbf{y})$ is non-negative definite and $\{\lambda_i,\vartheta_i\}$ are its corresponding eigenvalues and eigenfunctions; that is,

$$\int_\mathcal{D} C_z(\mathbf{x},\mathbf{y})\vartheta_i(\mathbf{y})\; d\mathbf{y} = \lambda_i\vartheta_i(\mathbf{x}).$$

The eigenfunctions $\{\vartheta_i\}$ form a complete orthogonal basis in $L^2(\mathcal{D})$. The eigenvalues $\{\lambda_i\}$ form a sequence of non-negative real numbers decreasing to zero and

$$\sum_{i=1}^{\infty} \lambda_i = \int_{\mathcal{D}} \mathbb{V}\mathrm{ar}[z](\mathbf{x})\, d\mathbf{x}.$$

In practical computation, one often employs a *truncated* Karhunen-Lòeve expansion (KLE):

$$z(\mathbf{x},\omega) \approx z_N(\mathbf{x},\omega) = \mathbb{E}[z](\mathbf{x}) + \sigma_z \sum_{i=1}^{N} \sqrt{\lambda_i}\vartheta_i(\mathbf{x})\xi_i(\omega), \quad N \in \mathbb{N}. \tag{2.9}$$

The series (2.9) represents the best N-term approximation of z and, by Mercer's Theorem [113, p. 245], we have

$$\sup_{\mathbf{x}\in\mathcal{D}} \mathbb{E}\left[(z - z_N)^2\right] = \sup_{\mathbf{x}\in\mathcal{D}} \sum_{i>N}^{\infty} \lambda_i \vartheta_i^2(\mathbf{x}) \to 0 \;\text{ as }\; N \to \infty.$$

In the sequel, we will employ the so-called *finite noise assumption,* which states that a random field $z(\mathbf{x},\omega)$ can be approximated with a prescribed finite number of random variables $\xi := \{\xi_1, \xi_2, \ldots, \xi_N\}$, where $N \in \mathbb{N}$ and $\xi_i(\omega) : \Omega \to \Gamma_i \subseteq \mathbb{R}$; this is, for instance, the case when we use a joint N-term KLE to approximate in (2.3) the random coefficient

$$a(\mathbf{x},\omega) \approx a_N(\mathbf{x},\xi(\omega)) = a(\mathbf{x},\xi_1(\omega),\xi_2(\omega),\ldots,\xi_N(\omega)). \tag{2.10}$$

We also make the simplifying assumption that each random variable is independent and characterized by a probability density function $\rho_i : \Gamma_i \to [0,1]$. If the distribution measure of the random vector $\xi(\omega)$ is absolutely continuous with respect to the Lebesgue measure, then there exists a joint probability density function $\rho : \Gamma \to \mathbb{R}^+$, where $\Gamma := \prod_{i=1}^N \Gamma_i \subset \mathbb{R}^N$, $\rho(\xi) = \prod_{i=1}^N \rho_i(\xi_i)$, and $\rho \in L^\infty(\Gamma)$. In particular, given the parametric representation (2.10) of $a(\mathbf{x},\omega)$, the Doob-Dynkin Lemma, cf. [5], guarantees that y, the solution corresponding to the RPDE (2.3), admits exactly the same parametrization; that is, $y(\mathbf{x},\omega) = y(\mathbf{x},\xi_1(\omega),\xi_2(\omega),\ldots,\xi_N(\omega))$. The number N has to be large enough so that the approximation error is sufficiently small. Furthermore, we can now replace the probability space $(\Omega, \mathfrak{F}, \mathbb{P})$ with $(\Omega, \mathbb{B}(\Gamma), \rho(\xi)d\xi)$, where $\mathbb{B}(\Gamma)$ denotes the Borel σ-algebra on Γ and $\rho(\xi)d\xi$ is the finite measure of the vector ξ. Besides, denoting the space of square-integrable random variables with respect to the density ρ by $L^2_\rho(\Gamma)$, we introduce the space $L^2(\mathcal{D}) \otimes L^2_\rho(\Gamma)$, equipped with the norm

$$||v||_{L^2(\mathcal{D})\otimes L^2_\rho(\Gamma)} := \left(\int_\Gamma ||v(\cdot,\xi)||^2_{L^2(\mathcal{D})}\rho(\xi)\, d\xi\right)^{\frac{1}{2}} < \infty. \tag{2.11}$$

Similarly, using equations (1.10) and (1.8) we have

$$\operatorname{std}(z) = \left[\int_\Gamma (z(\xi) - \mathbb{E}(z(\xi)))^2 \rho(\xi)\, d\xi\right]^{\frac{1}{2}} \text{ and } \mathbb{E}[z] = \langle z\rangle = \int_\Gamma z(\xi)\rho(\xi)\, d\xi < \infty. \tag{2.12}$$

Based on the fact that we can express the solution y of the stochastic elliptic problem (2.3) as $y(\mathbf{x},\xi) = y(\mathbf{x},\xi_1,\ldots,\xi_N)$, it is natural to treat $y(\mathbf{x},\xi)$, a function of d spatial variables and N random parameters, as a function of $d+N$ variables. This leads us to consider the Galerkin weak formulation (2.5) of (2.3), with respect to both physical and parameter space, in the following form: find $y \in \mathcal{V}_\rho := H^1_0(\mathcal{D}) \otimes L^2_\rho(\Gamma)$ such that, $\mathbb{P}$-almost surely,

$$\mathbb{E}\left[\int_\mathcal{D} a_N(\mathbf{x},\xi)\nabla y(\mathbf{x},\xi)\cdot\nabla v(\mathbf{x},\xi)\, d\mathbf{x}\right] = \mathbb{E}\left[\int_\mathcal{D} u(\mathbf{x})v(\mathbf{x},\xi)\, d\mathbf{x}\right],\ v \in \mathcal{V}_\rho. \tag{2.13}$$

Now, let $\mathcal{V}_{h,\rho} := \mathcal{X}_h \otimes L^2_\rho(\Gamma)$, where $\mathcal{X}_h \subset H^1_0(\mathcal{D})$ and $\dim(\mathcal{X}_h) = J$. Then, stochastic finite element methods proceed by discretizing the physical domain $\mathcal{D}$ in the usual way, leading to the semi-discrete problem: find $y_h \in \mathcal{V}_{h,\rho}$ such that, $\mathbb{P}$-almost surely,

$$\mathbb{E}\left[\int_\mathcal{D} a_N(\mathbf{x},\xi)\nabla y_h(\mathbf{x},\xi)\cdot\nabla v(\mathbf{x},\xi)\, d\mathbf{x}\right] = \mathbb{E}\left[\int_\mathcal{D} u_h(\mathbf{x})v(\mathbf{x},\xi)\, d\mathbf{x}\right],\ v \in \mathcal{V}_{h,\rho}. \tag{2.14}$$

As noted before, one could tackle (2.14) with either Monte Carlo finite element methods (MCFEM) [27] or stochastic collocation finite element methods (SCFEM) [4, 42] or SGFEM [3, 5]. In this thesis, we shall rely solely on the SGFEM for spatial and stochastic discretizations, and our exposition here particularly follows closely the framework in [110, 115]. First, however, we next give a brief overview of how to perform discretization with MCFEM and SCFEM, which will then be followed by a detailed description of SGFEM discretization.

2.2 Monte Carlo FEM

In general, Monte Carlo methods are sampling-based techniques for computing statistical quantities. In the context of RPDEs, one randomly samples the parameter vector $\xi \in \Gamma$ and computes realizations of the parametric PDE. Markov's inequality shows that the Monte Carlo method converges like $1/\sqrt{Q_{MC}}$ where Q_{MC} denotes the sample size, see e.g.,[142]. For this reason, Monte Carlo methods require a large sample size to determine 'good' approximations of the solution. To this end, let $\xi^k = \{\xi_1^k, \dots, \xi_N^k\} \in \Gamma$ be a sample of ξ. If $a_N^k(\mathbf{x}) = a_N(\mathbf{x}, \xi^k)$ is strictly positive, then each $y_h^k(\mathbf{x}) = y(\mathbf{x}, \xi^k) \in \mathcal{X}_h$ satisfies

$$\int_{\mathcal{D}} a_N^k(\mathbf{x}) \nabla y_h^k(\mathbf{x}) \cdot \nabla v(\mathbf{x}) \, d\mathbf{x} = \int_{\mathcal{D}} u(\mathbf{x}) v(\mathbf{x}) \, d\mathbf{x}, \quad v \in \mathcal{X}_h. \tag{2.15}$$

Here, discretizing (2.15) leads to a sequence of *decoupled* symmetric positive definite linear systems

$$A_k \mathbf{y}_k = \mathbf{b}, \quad k = 1, 2, \dots, Q_{MC}, \quad A_k \in \mathbb{R}^{J \times J}. \tag{2.16}$$

Each linear system in (2.16) can be solved using, for instance, the conjugate gradient method (CG) [39]. Furthermore, one can easily compute the moments of the solution as follows:

$$\mathbb{E}(y(\mathbf{x}, \cdot)^m) \approx \frac{1}{Q_{MC}} \sum_{k=1}^{Q_{MC}} y(\mathbf{x}, \xi^k)^m, \quad m = 1, 2, \dots$$

For a more detailed discussion on Monte Carlo methods and their hybrids, see e.g., [27, 51] and the references therein.

2.3 Stochastic collocation FEM

An alternative to the MCFEM is the collocation method, which samples (2.14) at a predetermined set of points $\xi^k = \{\xi_1^k, \dots, \xi_n^k\} \in \Gamma$ and constructs a high-order polynomial approximation y_{hn} to the solution function y which is then obtained by performing Lagrange interpolation. More precisely, following [47], one has

$$y_{hn}(\mathbf{x}, \xi^k) = \sum_{k=1}^{n} y_h^k(\mathbf{x}) \widehat{L}_k(\xi^k), \tag{2.17}$$

where $y_h^k(\mathbf{x}) = y(\mathbf{x}, \xi^k) \in \mathcal{X}_h$ satisfies (2.14) at $\xi^k \in \Gamma$ and $\widehat{L}_k(\xi^k)$ is a multivariate Lagrange polynomial. By construction, the approximation $y_{hn}(\mathbf{x}, \xi^k)$ is contained in the finite-dimensional subspace $\mathcal{X}_h \otimes \mathcal{Y}_n$ of the Hilbert space $H_0^1(\mathcal{D}) \otimes L_\rho^2(\Gamma)$. In particular, $\mathcal{Y}_n \subset L_\rho^2(\Gamma)$, where $\mathcal{Y}_n := \text{span}\{\widehat{L}_1(\xi^k), \ldots, \widehat{L}_n(\xi^k)\}$ and $\dim(\mathcal{Y}_n) = n$. Full tensor SCFEMs [37, 142] use Cartesian products of interpolation points on each Γ_k. Possibilities include Clenshaw-Curtis points and Gauss points. If $n_k + 1$ points are selected on Γ_k, then $n = \prod_{k=1}^N (n_k + 1)$, which quickly becomes intractable as N increases.

Sparse grid stochastic collocation methods [12, 15, 37, 142] are based on interpolation rules (such as the Stroud interpolation formulas) for high-dimensional problems. Let Z_i be a set of points on Γ_k, of size $m_i + 1$ where $m_0 = 1$ and $m_i = 2^{i-1}$ for $i \in \mathbb{N}$. For a given approximation level ℓ, the sparse grid on Γ is then defined via

$$\mathcal{H}(\ell, N) := \bigcup_{\ell \leq ||i||_1 \leq \ell + N} Z_{i_1} \times \cdots \times Z_{i_N}, \;\; i = (i_1, \ldots, i_N) \in \mathbb{N}^N.$$

The error incurred by approximating $y(\mathbf{x}, \xi)$ with $y_{hn}(\mathbf{x}, \xi^k)$ is due to interpolation. If p denotes the largest value for which polynomials of total degree p are interpolated exactly in (2.17), then sparse grid methods achieve total degree p accuracy with $\ell = p + 1$ using far fewer points than full tensor methods [9].

Just like in MCFEM, $y_h^k(\mathbf{x})$ in (2.17) solves (2.14) at $\xi^k \in \Gamma$, so that with a suitable finite element basis for $\mathcal{X}_h$, this leads to a decoupled set of linear systems. We refer the interested reader to [4, 37, 42, 47, 51, 142] for details on SCFEM and solution of the resulting linear systems.

2.4 Stochastic Galerkin FEM

The SGFEM is an intrusive approach[2] in which, like the SCFEM, one seeks y in a finite-dimensional subspace $\mathcal{X}_h \otimes \mathcal{Y}_n \subset H_0^1(\mathcal{D}) \otimes L_\rho^2(\Gamma)$, consisting of tensor products of deterministic functions defined on the spatial domain and stochastic functions defined on the probability space [51, 110]. However, unlike in SGFEM, the stochastic subspace $\mathcal{Y}_n \subset L_\rho^2(\Gamma)$

[2] Generally, SGFEM techniques are intrusive in the sense that they are non-ensemble-based methods; that is, they require the solution of discrete systems that couple all spatial and probabilistic degrees of freedom.

is spanned by multivariate Lagrangian polynomials in the SCFEM. Different classes of SGFEMs are distinguished by their choices for $\mathcal{Y}_n$. One class of SGFEMs uses tensor products of piecewise polynomials on the subdomains $\Gamma_i \subset \Gamma$ [5, 35, 93]. In this approach, the polynomial degree is fixed and approximation is improved by refining the partition of Γ. The classical, so-called *spectral* SGFEM (see e.g. [36, 43, 81, 110]) employs global polynomials of total degree n in N random variables ξ_i on Γ. In this approach, there is no partitioning of Γ and approximation is improved by increasing the polynomial degree. We shall adopt the latter method in this dissertation. To this end, suppose first that $\mathcal{X}_h \subset H_0^1(\mathcal{D})$ is a space of standard Lagrangian finite element functions on a partition $\mathbb{T}$ into triangles (or rectangles) of the domain $\mathcal{D}$ defined by

$$\mathcal{X}_h := \{v_h \in H_0^1(\mathcal{D}) : v_h \in P_k(\Xi) \ \ \forall \Xi \in \mathbb{T}\},$$

where $\Xi \in \mathbb{T}$ is a cell and P_k is the space of Lagrangian polynomials of degree k. In particular, let $\mathcal{X}_h = \mathrm{span}\{\phi_j(\mathbf{x}),\ j = 1, \dots, J\}$. Next, for the discretization of the stochastic domain we define the set $\mathcal{I}$ by

$$\mathcal{I} := \left\{ i = (i_1, \dots, i_N) \in \mathbb{N}^N : |i| = \sum_k i_k \leq n \right\},$$

and let $\mathcal{Y}_n \subset L^2_\rho(\Gamma)$ be such that $\mathcal{Y}_n := \mathrm{span}\{\psi_i(\xi) = \Pi_{k=1}^N \psi_k^{i_k}(\xi_k) : i \in \mathcal{I}\}$. Herein, $\{\psi_i(\xi)\}$ are N-variate orthogonal polynomials of degree at most n, whereas $\mathcal{I}$ is a set of all multi-indices of length N satisfying $|i| = \sum_k i_k \leq n$. It can then be shown that[3]

$$P := \dim(\mathcal{Y}_n) = \dim(\mathcal{I}) = 1 + \sum_{k=1}^{n} \frac{1}{k!} \prod_{j=0}^{k-1} (N + j) = \frac{(N+n)!}{N!n!}. \tag{2.18}$$

Hence, it turns out that there exists a bijection $\mu : \{1, \dots, P\} \to \mathcal{I}$ that assigns a unique integer i to each multi-index $\mu(i) \in \mathcal{I}$.

Note that when all the random variables ξ_i are independent and identically distributed Gaussian, the spectral approach uses a basis of multidimensional Hermite polynomials of total degree n termed the *polynomial chaos*, a terminology originally introduced by Nor-

[3] Note that if tensor product polynomials are used and a_N is linear in ξ as in (2.9), then the subspace $\mathcal{Y}_n$ possesses a basis that decouples the resulting linear system of equations [47].

bert Wiener [140] in the context of turbulence modeling. The use of Hermite polynomials ensures that the corresponding basis functions are orthogonal with respect to the Gaussian probability measure. This leads to sparse linear systems, a crucial property that must be exploited for fast solution schemes [110]. Relying on the fact that there exists a one-to-one correspondence between the probability density functions of alternative distributions and the weight functions of certain orthogonal polynomials, the concept of Hermite polynomials chaos has been extended to *generalized* polynomial chaos [143]. For instance, if uniform random variables (having support on a bounded interval) are chosen, then Legendre polynomials are the correct choice. Similarly, Jacobi polynomials go with beta-distributed random variables. When random variables with bounded images are used, the convergence and approximation properties of the resulting SGFEM are discussed in [5]. Throughout this thesis, we shall rely on Legendre polynomial chaos.

Example 2.2. *To illustrate here how the space $\mathcal{Y}_n$ is constructed [110], consider the case of uniform random variables with $N = 2$ and $n = 3$. Then $\mathcal{Y}_n$ is a set of two-dimensional Legendre polynomials (products of a univariate Legendre polynomial in ξ_1 and a univariate Legendre polynomial in ξ_2) of degree less than or equal to three. Each of the basis functions is associated with a multi-index $\nu = (\nu_1, \nu_2)$, where the components represent the degrees of the polynomials in ξ_1 and ξ_2. Since the total degree of the polynomial is three, we have the possibilities $\nu =$ (0,0), (1,0), (2,0), (3,0), (0,1), (1,1), (2,1), (0,2), (1,2), and* $(0,3)$. *Since the univariate Legendre polynomials of degrees* $0,1,2,3$ *are $L_0(x) = 1$, $L_1(x) = x$, $L_2(x) = \frac{1}{2}(3x^2-1)$, and $L_3(x) = \frac{1}{2}(5x^3-3x)$, we have*

$$\begin{aligned}\mathcal{Y}_n &= span\{\psi_i(\xi)\}_{i=0}^{9}\\ &= \{1, \xi_1, \frac{1}{2}(3\xi_1^2-1), \frac{1}{2}(5\xi_1^3-3\xi_1), \xi_2, \xi_1\xi_2, \frac{1}{2}(3\xi_1^2-1)\xi_2, \frac{1}{2}(3\xi_2^2-1),\\ &\quad \frac{1}{2}\xi_1(3\xi_2^2-1), \frac{1}{2}(5\xi_2^3-3\xi_2)\}.\end{aligned}$$

So, spectral SGFEM essentially entails performing a Galerkin projection onto $W_{hn} := \mathcal{X}_h \otimes \mathcal{Y}_n \subset H_0^1(\mathcal{D}) \otimes L_\rho^2(\Gamma)$ using basis functions r_{hn} of the form

$$r_{hn} = \sum_{j=1}^{J} \sum_{k\in\mathcal{I}} r_{jk}\phi_j(\mathbf{x})\psi_k(\xi), \tag{2.19}$$

where r_{ij} is a degree of freedom. Note, in particular, that

$$\begin{aligned}\mathbb{E}(r_{hn}) &= \left\langle \sum_{j=1}^{J} \sum_{k\in\mathcal{I}} r_{jk}\phi_j(\mathbf{x})\psi_k(\xi) \right\rangle \\ &= \sum_{j=1}^{J} \sum_{k=0}^{P-1} r_{jk}\phi_j(\mathbf{x}) \langle \psi_k(\xi)\rangle \\ &= \sum_{j=1}^{J} \sum_{k=0}^{P-1} r_{jk}\phi_j(\mathbf{x})\delta_{0k} = \sum_{j=1}^{J} r_{j0}\phi_j(\mathbf{x}), \end{aligned} \tag{2.20}$$

since

$$\langle \psi_0(\xi)\rangle = 1, \;\; \langle \psi_j(\xi)\rangle = 0, \; j > 0, \;\; \langle \psi_j(\xi)\psi_k(\xi)\rangle = \left\langle \psi_j^2(\xi)\right\rangle \delta_{jk}. \tag{2.21}$$

Now, our variational problem is to find $y_{hn} \in W_{hn}$ satisfying

$$\begin{aligned}\mathbb{E}\left[\int_{\mathcal{D}} \left(a_0 + \sigma_a \sum_{i=1}^{N} \sqrt{\lambda_i}\vartheta_i(\mathbf{x})\xi_i\right) \nabla y_{hn}(\mathbf{x},\xi)\cdot\nabla v(\mathbf{x},\xi)\, d\mathbf{x}\right] = \\ \mathbb{E}\left[\int_{\mathcal{D}} u(\mathbf{x})v(\mathbf{x},\xi)\, d\mathbf{x}\right], \;\; \forall v \in W_{hn}, \end{aligned} \tag{2.22}$$

where $a_0 = \mathbb{E}[a]$. Expanding y_{hn} and the test functions in the chosen basis in (2.22), we see that

$$y_{hn} = \sum_{k=0}^{P-1} \sum_{j=1}^{J} y_{jk}\phi_j(\mathbf{x})\psi_k(\xi) = \sum_{k=0}^{P-1} y_k\psi_k(\xi),$$

where $\{\phi_j\}$ are $\mathbf{Q}_1$ finite elements and $\{\psi_i\}$ are multi-dimensional Legendre polynomials, yields the $JP \times JP$ *coupled* linear system of equations with block structure:

$$\mathcal{K}\mathbf{y} = \mathbf{u}, \tag{2.23}$$

where

$$\mathcal{K} = \begin{bmatrix} \mathcal{K}^{(0,0)} & \mathcal{K}^{(0,1)} & \cdots & \mathcal{K}^{(0,P-1)} \\ & \ddots & & \\ \vdots & & \mathcal{K}^{(k,k)} & \vdots \\ & & \ddots & \\ \mathcal{K}^{(P-1,0)} & \mathcal{K}^{(P-1,1)} & \cdots & \mathcal{K}^{(P-1,P-1)} \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} \mathbf{y}_0 \\ \vdots \\ \mathbf{y}_k \\ \vdots \\ \mathbf{y}_{P-1} \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} \mathbf{u}_0 \\ \vdots \\ \mathbf{u}_k \\ \vdots \\ \mathbf{u}_{P-1} \end{bmatrix},$$

$\mathbf{y}_k, \mathbf{u}_k \in \mathbb{R}^J,\ k = 0, \ldots, P-1$, and the blocks $\mathcal{K}^{(p,q)}$ of the stochastic Galerkin matrix $\mathcal{K}$ are linear combinations of $N+1$ weighted stiffness matrices of dimension J, with each of them having the same sparsity pattern equivalent to that of the corresponding deterministic problem. More specifically, for $p, q = 0, \ldots, P-1$, we have

$$\mathcal{K}^{(p,q)} = \langle \psi_p(\xi)\psi_q(\xi)\rangle K_0 + \sum_{i=1}^{N} \langle \xi_i \psi_p(\xi)\psi_q(\xi)\rangle K_i, \tag{2.24}$$

and the stiffness matrices $K_i \in \mathbb{R}^{J\times J},\ i = 0, 1, \ldots, N$, are given, respectively, by

$$K_0(j,k) = \int_{\mathcal{D}} \mathbb{E}[a](\mathbf{x})\nabla\phi_j(\mathbf{x})\nabla\phi_k(\mathbf{x})\, d\mathbf{x}, \tag{2.25}$$

$$K_i(j,k) = \sigma_a\sqrt{\lambda_i}\int_{\mathcal{D}} \vartheta_i(\mathbf{x})\nabla\phi_j(\mathbf{x})\nabla\phi_k(\mathbf{x})\, d\mathbf{x},\ i > 0, \tag{2.26}$$

where $\mathbb{E}[a] > 0$ due to (2.4), so that K_0 is symmetric and positive definite. The block K_0 captures the mean information in the model and appears on the diagonal blocks of $\mathcal{K}$, whereas the other blocks $K_i,\ i = 1, \ldots, N$, represent the fluctuations in the model. In Kronecker product notation, one obtains

$$\mathcal{K} := G_0 \otimes K_0 + \sum_{i=1}^{N} G_i \otimes K_i, \quad \mathbf{u} := \mathbf{g}_0 \otimes \mathbf{u}_0, \tag{2.27}$$

where

$$\begin{cases} G_0 = \operatorname{diag}\left(\langle\psi_0^2\rangle, \langle\psi_1^2\rangle, \ldots, \langle\psi_{P-1}^2\rangle\right), \\ G_i(j,k) = \langle\xi_i\psi_j\psi_k\rangle,\ \ i = 1, \ldots, N, \end{cases} \tag{2.28}$$

and the vectors $\mathbf{g}_0 \in \mathbb{R}^P$ and $\mathbf{u}_0 \in \mathbb{R}^J$ are defined via

$$\mathbf{g}_0(i) = \langle\psi_{i-1}(\xi)\rangle,\ \ i = 1, \ldots, P, \quad \mathbf{u}_0(j) = \int_{\mathcal{D}} u(\mathbf{x})\phi_j(\mathbf{x})\, d\mathbf{x},\ \ j = 1, \ldots, J, \tag{2.29}$$

due to the orthogonality of the stochastic basis functions with respect to the probability measure of the distribution of the chosen random variables (cf. (2.21)). Observe that $\mathbf{u}_k = \mathbf{0} \in \mathbb{R}^J,\ \ k = 1, \ldots, P-1$.

Now, suppose we denote (normalized) univariate orthogonal polynomials by $\{\varphi_k\}$.

Then, recall that the sequence $\{\varphi_k\}$ satisfies the three-term recurrence relation [37]

$$\varphi_{k+1}(x) = (x - \alpha_k)\varphi_k(x) - \beta_k \varphi_{k-1}(x), \quad x \in \mathbb{R},$$

with $\varphi_0 = 1, \varphi_{-1} = 0$, it turns out that

$$G_0(j,k) = \langle \psi_j, \psi_k \rangle = \prod_{i=1}^{N} \langle \varphi_{j_i}, \varphi_{k_i} \rangle = \prod_{i=1}^{N} \delta_{j_i k_i} = \delta_{jk}, \tag{2.30}$$

and for $i > 0$, we have

$$\begin{aligned} G_i(j,k) &= \langle \xi_i \psi_j, \psi_k \rangle \\ &= \langle \xi_i \varphi_j, \varphi_k \rangle \prod_{l=1, l\neq i}^{N} \langle \varphi_{j_l}, \varphi_{k_l} \rangle \\ &= \left(\langle \varphi_{j_i+1}, \varphi_{k_i} \rangle + \alpha_{j_i} \langle \varphi_{j_i}, \varphi_{k_i} \rangle + \beta_{j_i} \langle \varphi_{j_i-1}, \varphi_{k_i} \rangle \right) \prod_{l=1, l\neq i}^{N} \langle \varphi_{j_l}, \varphi_{k_l} \rangle. \end{aligned} \tag{2.31}$$

Hence, G_0 is a diagonal matrix whereas for $k > 0$, the matrix G_k has at most three non-zero elements per row. Moreover, for symmetric density functions ρ, the coefficients α_j in the recurrence relation vanish so that the matrices G_k have a most two non-zeros per row, see e.g. [37, 110]. This is the case when using, for instance, Legendre or Hermite polynomial chaos.

The matrices G_k possess hierarchical structure. More specifically, for a polynomial chaos of order n, note that each $G_k \in \mathbb{R}^{P \times P}$ (cf. (2.18)) can be written in block form as

$$G_k = \begin{bmatrix} \widehat{G}_k & F_k^T \\ F_k & D_k \end{bmatrix}, \quad \widehat{G}_k \in \mathbb{R}^{P_a \times P_a}, \;\; F_k \in \mathbb{R}^{P_b \times P_a}, \;\; D_k \in \mathbb{R}^{P_b \times P_b}, \tag{2.32}$$

where $P = P_a + P_b = \frac{(N+n)!}{N!n!}$ and $P_a = \frac{(N+n-1)!}{N!(n-1)!}$. The matrix $\widehat{G}_k$ is defined exactly the same way as G_k and corresponds to a chaos of order $n-1$. By recursion, $\widehat{G}_k$ has a similar structure to that of (2.32). The recursion terminates with $\widehat{G}_k \in \mathbb{R}^{1\times 1}$.

It turns out that the global stochastic Galerkin matrix $\mathcal{K}$ inherits the hierarchical structure of the G_k thanks to the Kronecker product representation (2.27). Besides, $\mathcal{K}$ is symmetric and positive definite; it is highly sparse as many of the sums in (2.24) are zero.

2.5 An unsteady PDE with random inputs

In an attempt to extend our discussion on the above model problem to a time-dependent case, we consider the stochastic initial-boundary value problem: find a random function $y : [0,T] \times \mathcal{D} \times \Omega \to \mathbb{R}$, such that, $\mathbb{P}$-almost surely in Ω, the following parabolic equation holds:

$$\begin{cases} \dfrac{\partial y(t,\mathbf{x},\omega)}{\partial t} - \nabla \cdot (a(\mathbf{x},\omega)\nabla y(t,\mathbf{x},\omega)) = u(t,\mathbf{x}), \ \text{ in } \ (0,T] \times \mathcal{D} \times \Omega, \\ \qquad\qquad\qquad\qquad y(t,\mathbf{x},\omega) = 0, \ \text{ on } \ (0,T] \times \partial\mathcal{D} \times \Omega, \\ \qquad\qquad\qquad\qquad y(0,\mathbf{x},\omega) = 0, \ \text{ in } \ \mathcal{D} \times \Omega, \end{cases} \tag{2.33}$$

where the source function satisfies $u \in L^2(0,T;\mathcal{D})$ and, as before, $a(\mathbf{x},\omega)$ is assumed to be uniformly positive in $\mathcal{D} \times \Omega$. For the existence and uniqueness of (2.33), see e.g., [93]. We note here that, unlike the steady-state problem, the time-dependent model problem presents the additional challenge of solving a large coupled linear system for each time step [81, 114, 143, 144]. We will need to work with the linear systems resulting from both the stationary and the unsteady models in the sequel.

Now, using SGFEM for the spatial and the stochastic discretizations of (2.33) yields the system of ordinary differential equations [13]:

$$(G_0 \otimes M)\,\frac{d\mathbf{y}(t)}{dt} + \left(\sum_{i=0}^{N} G_i \otimes K_i\right)\mathbf{y}(t) = \mathbf{g}_0 \otimes \mathbf{u}_0, \tag{2.34}$$

where

$$M(j,k) = \int_{\mathcal{D}} \phi_j(\mathbf{x})\phi_k(\mathbf{x})\, d\mathbf{x} \tag{2.35}$$

is the finite element mass matrix. For time discretization, we use the implicit Euler method to avoid stability issues. To this end, we set $t_n = n\tau, \ n = 0,1,\ldots,n_t$, with $\tau = T/n_t$. Moreover, we define the computed numerical approximation $\mathbf{y}(t_n) := \mathbf{y}^n$, so that (2.34) yields

$$G_0 \otimes M\left(\frac{\mathbf{y}^n - \mathbf{y}^{n-1}}{\tau}\right) + \left(\sum_{i=0}^{N} G_i \otimes K_i\right)\mathbf{y}^n = (\mathbf{g}_0 \otimes \mathbf{u}_0)^n\,, \tag{2.36}$$

or, equivalently,

$$\widehat{\mathcal{K}}_\tau \mathbf{y}^n = \mathbf{b}^n, \tag{2.37}$$

where

$$\mathbf{b}^n = (G_0 \otimes M)\,\mathbf{y}^{n-1} + \tau\,(\mathbf{g}_0 \otimes \mathbf{u}_0)^n\,, \tag{2.38}$$

and

$$\begin{aligned}\widehat{\mathcal{K}}_\tau &= G_0 \otimes M + \tau \sum_{i=0}^{N} G_i \otimes K_i \\ &= G_0 \otimes \tilde{K}_0 + \sum_{i=1}^{N} G_i \otimes \tilde{K}_i, \end{aligned} \tag{2.39}$$

with $\tilde{K}_0 := M + \tau K_0,\;\; \tilde{K}_i = \tau K_i,\; i = 1, \ldots, N.$

Having completed our discussion on three different discretization approaches, we proceed next to Chapter 3 to discuss the solution methods for the linear system (2.37).

Chapter 3

Solution methods for the stochastic Galerkin system

In this chapter, we present low-rank solvers for the stochastic Galerkin linear systems obtained in Chapter 2. Here, we first recall the major existing solution methods in Section 3.1 before proceeding to propose our low-rank iterative methods in Section 3.2. The low-rank preconditioned iterative solvers presented herein are based on [13]. Numerical experiments are provided to demonstrate that these solvers are effective, especially when the fluctuations in the random data are not too large relative to their mean values.

3.1 Existing iterative solvers

The stochastic Galerkin method requires the solution of the symmetric and positive-definite (spd) Galerkin system (2.37). Once its solution is obtained, statistical quantities such as moments or a probability distribution associated with the solution process can be computed cheaply. Provided the size of the linear system is relatively small, note that one could assemble the entire Galerkin matrix and solve the linear system with, for example, Gaussian elimination. Gaussian elimination would entail $\mathcal{O}((JP)^3)$ work. For a fairly large-scale linear system, this approach is, however, inefficient in terms of memory usage since it requires assembling the full stochastic Galerkin matrix $\widehat{\mathcal{K}}_{\tau}$.

A major class of iterative solvers in the context of large-scale SGFEM are the multilevel methods. These methods essentially build a hierarchy of levels with respect to either the spatial (deterministic) discretization or the stochastic discretization (c.f (2.32)) to solve the stochastic Galerkin system. The approach is considered in [81, 114]. In addition to numerical results, [114] use local Fourier mode analysis techniques for theoretical investigations on the solver performance. Elman and Furnival [36] also consider a multilevel method based on a hierarchy of spatial grids and prove independence of the multilevel convergence rate of deterministic and stochastic discretization parameters for a random diffusion coefficient of special form. The second approach, which is based on a hierarchy of stochastic shape functions, has been considered in [68, 121].

The Krylov subspace methods are probably the most popular methods for solving large, sparse linear systems (see e.g. [39] and the references therein). The basic idea behind the Krylov subspace methods is the following. Consider, for arbitrary $\mathcal{A} \in \mathbb{R}^{m\times m}$ and $\mathbf{b} \in \mathbb{R}^m$, the linear system

$$\mathcal{A}\mathbf{x} = \mathbf{b}. \tag{3.1}$$

Suppose now that $\mathbf{x}_0$ is an initial guess for the solution $\mathbf{x}$ of (3.1), and define the initial residual $\mathbf{r}_0 = \mathbf{b} - \mathcal{A}\mathbf{x}_0$. Krylov subspace methods are iterative methods whose kth iterate $\mathbf{x}_k$ satisfies

$$\mathbf{x}_k \in \mathbf{x}_0 + \mathbb{K}_k(\mathcal{A}, \mathbf{x}_0), \quad k = 1, 2, \ldots, \tag{3.2}$$

where

$$\mathbb{K}_k(\mathcal{A}, \mathbf{x}_0) := \operatorname{span}\left\{\mathbf{r}_0, \mathcal{A}\mathbf{r}_0, \ldots, \mathcal{A}^{k-1}\mathbf{r}_0\right\} \tag{3.3}$$

denotes the kth Krylov subspace generated by $\mathcal{A}$ and $\mathbf{r}_0$. The Krylov subspaces form a nested sequence that ends with dimension $d = \dim(\mathbb{K}_m(\mathcal{A}, \mathbf{r}_0)) \leq m$, i.e.,

$$\mathbb{K}_1(\mathcal{A}, \mathbf{r}_0) \subset \ldots \subset \mathbb{K}_d(\mathcal{A}, \mathbf{r}_0) = \cdots = \mathbb{K}_m(\mathcal{A}, \mathbf{r}_0).$$

In particular, for each $k \leq d$, the Krylov subspace $\mathbb{K}_k(\mathcal{A}, \mathbf{r}_0)$ has dimension k. Because of the k degrees of freedom in the choice of the iterate $\mathbf{x}_k$, k constraints are required to make $\mathbf{x}_k$ unique. In Krylov subspace methods this is achieved by requiring that the kth residual $\mathbf{r}_k = \mathbf{b} - \mathcal{A}\mathbf{x}_k$ is orthogonal (with respect to the Euclidean inner product) to a k-dimensional space $\mathcal{C}_k$, called the constraints space:

$$\mathbf{r}_k = \mathbf{b} - \mathcal{A}\mathbf{x}_k \in \mathbf{r}_0 + \mathcal{A}\mathbb{K}_k(\mathcal{A}, \mathbf{r}_0), \tag{3.4}$$

where $\mathbf{r}_k \perp \mathcal{C}_k$. It can be shown [17] that there exists a uniquely defined iterate $\mathbf{x}_k$ of the form (3.2) and for which the residual $\mathbf{r}_k = \mathbf{b} - \mathcal{A}\mathbf{x}_k$ satisfies (3.4) if

(a) $\mathcal{A}$ is symmetric positive definite and $\mathcal{C}_k = \mathbb{K}_k(\mathcal{A}, \mathbf{r}_0)$, or

(b) $\mathcal{A}$ is nonsingular and $\mathcal{C}_k = \mathcal{A}\mathbb{K}_k(\mathcal{A}, \mathbf{r}_0)$.

In particular, (a) characterizes the conjugate gradient (CG) method [39] whereas (b) characterizes the minimal residual (MINRES) method [100] and the generalized minimal residual (GMRES) method [117].

In this thesis, the Krylov subspace solvers will be a chief cornerstone in our discussions. In particular, since $\widehat{\mathcal{K}}_\tau$ is symmetric and positive definite, we elect to focus in this chapter on the *preconditioned* conjugate gradient method[1] (PCG) to solve the system (2.37). Algorithm 3.1 shows the PCG for solving an arbitrary symmetric and positive definite system, say, (3.1), with a suitable preconditioner $\mathcal{P}$. Note that CG only requires the evaluation of matrix-vector products so that it is unnecessary to store the assembled matrix $\widehat{\mathcal{K}}_\tau$ [37, 110]. Indeed, one can perform the matrix-vector products implicitly following a procedure described by Pellissetti and Ghanem in [106]. More specifically, each matrix $\tilde{K}_k$ is assembled and the matrix-vector product is expressed as $(\widehat{\mathcal{K}}_\tau y)_j = \Sigma_{i=0}^{P-1}\Sigma_{k=0}^{N} \langle \xi_k \psi_i \psi_j \rangle \, (\tilde{K}_k y_i)$. The terms $(\tilde{K}_k y_i)$ are precomputed and then appropriately scaled as needed. This approach is efficient since most of the terms $\langle \xi_k \psi_i \psi_j \rangle$ are zero [37]. The cost of performing the matrix-vector product in this manner is essentially determined by the computation of $(\tilde{K}_k y_i)$ for $0 \le k \le N$ and $0 \le i \le P-1$, which entails $P(N+1)$ sparse matrix-vector products by matrices $\tilde{K}_k$ of order J. The implicit matrix-vector product also only requires the assembly of $N+1$ order-J stiffness matrices and the assembly of the components $\langle \xi_k \psi_i \psi_j \rangle$ of G_k.

[1] The concept of and the need for preconditioning linear systems will be made clearer in the sequel, see also e.g., [39, Chapter 2].

Algorithm 3.1 The preconditioned conjugate gradient method (PCG)

```
Choose $\mathbf{x}^{(0)}$,  compute $\mathbf{r}^{(1)} = \mathbf{b} - \mathcal{A}\mathbf{x}^{(0)}$
Solve $\mathcal{P}\mathbf{z}^{(1)} = \mathbf{r}^{(1)}$,   set $\mathbf{p}^{(0)} = \mathbf{z}^{(0)}$
for  $j = 0$ until convergence do
    $\alpha_j = \left\langle \mathbf{z}^{(j)}, \mathbf{r}^{(j)} \right\rangle / \left\langle \mathcal{A}\mathbf{p}^{(j)}, \mathbf{p}^{(j)} \right\rangle$
    $\mathbf{x}^{(j+1)} = \mathbf{x}^{(j)} + \alpha_j \mathbf{p}^{(j)}$
    $\mathbf{r}^{(j+1)} = \mathbf{r}^{(j)} - \alpha_j \mathcal{A}\mathbf{p}^{(j)}$
    < Test for convergence >
    solve $\mathcal{P}\mathbf{z}^{(j+1)} = \mathbf{r}^{(j+1)}$
    $\beta_{j+1} = \left\langle \mathbf{z}^{(j+1)}, \mathbf{r}^{(j)} \right\rangle / \left\langle \mathbf{z}^{(j)}, \mathbf{r}^{(j)} \right\rangle$
    $\mathbf{p}^{(j+1)} = \mathbf{z}^{(j)} + \beta_j \mathbf{p}^{(j)}$
end for
```

We remark here that recycled Krylov subspace methods [101] have also been employed in [65, 129] to study stochastic Galerkin linear systems. It is observed in these papers that subspace recycling results in considerable savings in terms of CPU time requirements.

Notwithstanding the advantages of the *full* solution methods presented above, we want to emphasize that the matrix dimensions quickly become prohibitively large with respect to the discretization parameters. As a consequence, one expects overwhelming memory and computational time requirements. Hence, it becomes impossible to compute the full solution to an SGFEM discretized problem. For instance, in practical applications such as groundwater flow problems, the length N of the random vector ξ is usually large due to the presence of small correlation length in the covariance function of a. This, in turn, increases the value of P in (2.18) (and hence the dimension of $\widehat{\mathcal{K}}_\tau$) quite fast, see e.g., [42]. This is a major drawback of the SGFEM. In order to break the *curse of dimensionality* associated with this problem, we propose a *low-rank approximation* to the solution of the linear system (2.37). The low-rank technique presented here only needs to store a small portion of the vectors in comparison to the full problem and we want to theoretically justify this approach in the sequel.

3.2 A low-rank solution approach

Observe first from (2.37), (2.39) and (1.2) that the stochastic Galerkin linear system (2.37) can be written as a matrix equation. That is, since

$$\left(\sum_{k=0}^{N} G_k \otimes \tilde{K}_k\right) \operatorname{vec}(Y) = \operatorname{vec}\left(\sum_{k=0}^{N} \tilde{K}_k Y G_k^T\right) = \operatorname{vec}\left(U\right),$$

where $U, Y \in \mathbb{R}^{J\times P}$, $G_k \in \mathbb{R}^{P\times P}$, $\tilde{K}_k \in \mathbb{R}^{J\times J}$, $k = 0, 1, \ldots, N$, we have

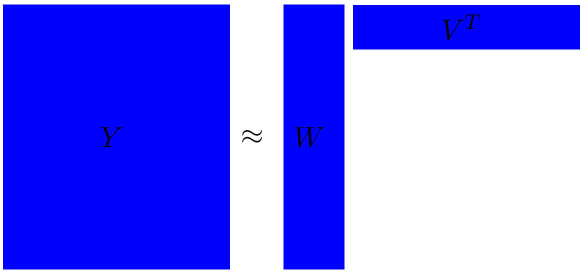
$$\sum_{k=0}^{N} \tilde{K}_k Y G_k^T = EF^T, \tag{3.5}$$

where $U = EF^T$, $E = \mathbf{u}_0 \in \mathbb{R}^{J\times r}$, $F = \mathbf{g}_0 \in \mathbb{R}^{P\times r}$, with $r = 1$ and $\mathbf{u}_0$, $\mathbf{g}_0$ as defined in (2.29). The matrix equation (3.5) can be viewed as a *generalized Sylvester equation* in the unknown Y. A fundamental challenge when solving (3.5) in the large-scale setting is storing the solution matrix Y which is typically dense even though G_k and $\tilde{K}_k$ are sparse. In practical applications, however, one generally has $r \ll J, P$. When this happens, we will refer to (3.5) as having a *low-rank* right-hand side. In this case, the resulting storage requirements for the data E and F of (3.5) are $\mathcal{O}(J+P)r$. However, the right-hand side U itself, as well as the solution matrix Y, has $\mathcal{O}(JP)$ storage requirements. Thus we see that, in the large-scale case, even storing a solution to (3.5) is computationally challenging! We shall soon see that in the large-scale, low-rank right-hand side scenario it is often the case that a solution may be approximated as $Y \approx WV^T$ with W and V both having q columns, where $q \ll J, P$, and this is what is meant by a *low-rank approximation.* If one attempts to solve for the low-rank factors W and V instead of Y, then the cost of storage for an approximate solution to (3.5) is reduced by a factor of

$$\frac{(J+P)q}{JP};$$

so, if $q \ll \min(J, P)$, then we achieve significant memory savings. The problem therefore becomes computationally tractable. A depiction of a low-rank approximation to a dense solution matrix Y when $J = P$ is given in Figure 3.1.

Figure 3.1: Approximation of a matrix Y by its low-rank components W and V.



Finally, we note here that for an arbitrary matrix $A \in \mathbb{R}^{n\times m}$, one can compute the best rank-r approximation $A_r = W_r V_r^T$ of A via the singular value decomposition (SVD) [46], where A_r is obtained by dropping all but the first r largest singular values of A, with $W_r \in \mathbb{R}^{n\times r}$ and $V_r \in \mathbb{R}^{m\times r}$. More specifically, let $A = \tilde{U}\Sigma\tilde{V}^T$ be the SVD of A. Then, we can define $W_r = \tilde{U}_r\Sigma_r$ and $V_r = \tilde{V}_r$, where Σ_r is the diagonal matrix containing the first r largest (diagonal entries of Σ) singular values of A, whereas $\tilde{U}_r$ and $\tilde{V}_r$ contain, respectively, the first r columns of $\tilde{U}$ and $\tilde{V}$. By the Eckart-Young-Mirsky theorem (see e.g. [46]), the matrix A_r is the best approximation of A in the set of all rank-r matrices with respect to the Frobenius norm.

3.2.1 Existence of low-rank solution

In what follows, we focus our attention on the solution of the system (2.37) using low-rank Krylov subspace solvers. First, however, we show, under certain conditions, that the solution of (2.37) can be approximated with a vector of low tensor rank. Our point of departure is the so-called Sherman-Morrison-Woodbury formula (see e.g. [145]), on which we shall rely to prove our main result in this chapter.

Lemma 3.1. *Let $X \in \mathbb{R}^{n\times n}$ be nonsingular and let $Y, Z \in \mathbb{R}^{n\times m}$, with $m \leq n$. Then $X + YZ^T$ is invertible if and only if $I + Z^T X^{-1} Y$ is invertible, with*

$$(X + YZ^T)^{-1} = X^{-1} - X^{-1}Y(I + Z^T X^{-1}Y)^{-1} Z^T X^{-1}. \tag{3.6}$$

We can now state our main result, which shows that the solution of the system (2.37) can indeed be approximated with a vector of low tensor rank. For this purpose, we split the matrix (2.39) as follows:

$$\widehat{\mathcal{K}}_\tau = \underbrace{G_0 \otimes \tilde{K}_0}_{:=\mathcal{L}} + \sum_{i=1}^{N} G_i \otimes \tilde{K}_i. \tag{3.7}$$

Observe then from (2.25), (2.28), (2.35) and (2.39) that $\mathcal{L}$ in (3.7) is symmetric and positive definite. Furthermore, let the stochastic matrices $G_i,\ i = 1, \ldots, N$, be decomposed in low-rank format:

$$G_i := W_i V_i^T, \;\; W_i, V_i \in \mathbb{R}^{P\times r_i}, \; i = 1, \ldots, N. \tag{3.8}$$

We illustrate the low-rank nature of these matrices in Section 3.3. Since also the stiffness matrices $\tilde{K}_i, \; i = 1, \ldots, N$, are symmetric, then each of them admits the factorization:

$$\tilde{K}_i := L_i D_i L_i^T = \tilde{L}_i L_i^T, \;\; \tilde{L}_i, L_i \in \mathbb{R}^{J\times J}, \; i = 1, \ldots, N, \tag{3.9}$$

where $\tilde{L}_i := L_i D_i, \; i = 1, \ldots, N$, with D_i and L_i (and hence $\tilde{L}_i$) being, respectively, diagonal and lower triangular matrices. The following result holds, see also [10, 13].

Theorem 3.2. *Let $\widehat{\mathcal{K}}_\tau$ denote a matrix of Kronecker product structure as in (2.39). Let $G_i, \; i = 1, \ldots, N$, have the low-rank representation (3.8) with $r = \sum_{j=1}^N r_j$, and let $\tilde{K}_i, \; i = 1, \ldots, N$, be given by the decomposition (3.9). Suppose further that $W = [W_1 \otimes \tilde{L}_1, \ldots, W_N \otimes \tilde{L}_N]$ and $V = [V_1 \otimes L_1, \ldots, V_N \otimes L_N]$. For all time steps $n \geq 2$, let the tensor rank of $\mathbf{b}^n \leq \ell$, where $\ell \ll JP$. Then, the linear system (2.37) admits the low-rank solution vector $\mathbf{y}^n$ of the form*

$$\mathbf{y}^n = \left(G_0^{-1} \otimes \tilde{K}_0^{-1}\right)(\mathbf{b}^n - W\widehat{\mathbf{y}}), \tag{3.10}$$

where the vector $\widehat{\mathbf{y}} \in \mathbb{R}^{J\cdot r}$ is the solution of

$$(I_{J\cdot r} + V^T \mathcal{L}^{-1} W)\widehat{\mathbf{y}} = V^T \mathcal{L}^{-1}\mathbf{b}^n. \tag{3.11}$$

Moreover, the tensor rank of $\mathbf{y}^n$ in (3.10) is at most

(i) $r + 1$, if $n = 1$, and

(ii) $r + \ell$, if $n \geq 2$.

Proof. Observe first from (1.3), (3.8) and (3.9) that we have the low-rank representation

$$\sum_{i=1}^N G_i \otimes \tilde{K}_i = \sum_{i=1}^N (W_i V_i^T) \otimes (\tilde{L}_i L_i^T) = \sum_{i=1}^N (W_i \otimes \tilde{L}_i)(V_i^T \otimes L_i^T) = WV^T. \tag{3.12}$$

Hence, from Lemma 3.1, (3.7) and (3.12), we note that

$$\widehat{\mathcal{K}}_\tau^{-1} = (\mathcal{L} + WV^T)^{-1} = \mathcal{L}^{-1} - \mathcal{L}^{-1}W(I_{J\cdot r} + V^T\mathcal{L}^{-1}W)^{-1}V^T\mathcal{L}^{-1},$$

so that

$$\mathbf{y}^n = \widehat{\mathcal{K}}_\tau^{-1}\mathbf{b}^n \Leftrightarrow \mathbf{y}^n = \mathcal{L}^{-1}\left[\mathbf{b}^n - W\underbrace{(I_{J\cdot r} + V^T\mathcal{L}^{-1}W)^{-1}V^T\mathcal{L}^{-1}\mathbf{b}^n}_{=\widehat{\mathbf{y}}}\right]. \tag{3.13}$$

Now, by definition, the symmetric and positive definite matrix $\mathcal{L}$ satisfies $\mathcal{L}^{-1} = G_0^{-1} \otimes \tilde{K}_0^{-1}$, which, together with (3.13), immediately yields (3.10).

To show (i), it suffices to show that the tensor rank of $\mathbf{b}^1 - W\widehat{\mathbf{y}}$ is at most $r+1$. Now, note that

$$\operatorname{rank}(\operatorname{vec}^{-1}(\mathbf{b}^1 - W\widehat{\mathbf{y}})) \leq \operatorname{rank}(\operatorname{vec}^{-1}(\mathbf{b}^1)) + \operatorname{rank}(\operatorname{vec}^{-1}(-W\widehat{\mathbf{y}})). \tag{3.14}$$

From (2.38), we see that $\mathbf{b}^1 = \tau\left(\mathbf{g}_0 \otimes \mathbf{u}_0\right)$, since $\mathbf{y}^0 = 0$ and the source term u is time-independent. But then, since the orthogonal polynomials $\{\psi_j\}$ satisfy

$$\mathbf{g}_0(j) = \langle\psi_j\rangle = \begin{cases} 1, & j = 0, \\ 0, & \text{otherwise}, \end{cases}$$

it follows from (2.29) that $\operatorname{vec}^{-1}(\mathbf{g}_0 \otimes \mathbf{u}_0) \in \mathbb{R}^{J\times P}$ is a matrix of rank 1. Hence, $\mathbf{b}^1$ is a vector of tensor rank 1. Next, we show that the tensor rank of $W\widehat{\mathbf{y}}$ is r, which, together with (3.14), completes the proof of (i). To that end, note from (3.13) that since the vector $\widehat{\mathbf{y}} \in \mathbb{R}^{Jr}$, with $r = \sum_{j=1}^N r_j$, we can reshape $\widehat{\mathbf{y}}$ to obtain the matrix $\widehat{Y} \in \mathbb{R}^{J\times r}$ via the operator vec^{-1}. More specifically, we have $\operatorname{vec}^{-1}(\widehat{\mathbf{y}}) = [\widehat{Y}_1, \ldots, \widehat{Y}_N] = \widehat{Y}$, where the submatrices $\widehat{Y}_i \in \mathbb{R}^{J\times r_i}$. Now, set $Z_i := L_i\widehat{Y}_i,\ i = 1, \ldots, N$. Observe then from (1.2) that

$$W\widehat{\mathbf{y}} = \sum_{i=1}^N (W_i \otimes L_i)\operatorname{vec}(\widehat{Y}_i) = \sum_{i=1}^N \operatorname{vec}(L_i\widehat{Y}_iW_i^T) = \sum_{i=1}^N \operatorname{vec}(Z_iW_i^T) = \sum_{i=1}^N\sum_{j=1}^{r_i} W_{ij} \otimes Z_{ij}^T.$$

But then, by assumption, $\{r_i\}_{i=1}^N$ sums up to r. Hence, the tensor rank of $W\widehat{\mathbf{y}}$ is r.

Finally, to prove the assertion (ii), suppose that, for $n \geq 2$, the tensor rank of $\mathbf{b}^n$ is at most $\ell \ll JP$. Since the tensor rank of $W\widehat{\mathbf{y}}$ is r, it trivially follows from the previous argument and the definition of $\mathbf{b}^n$ in (2.38) that (ii) holds with $\ell \geq 1$. □

Remark 3.3. *Note that, G_0 is just a $P \times P$ identity matrix if we work with orthonormal basis polynomials $\{\psi_i\}$. Hence, in this special case, (3.10) reduces to*

$$\mathbf{y}^n = \left(I_P \otimes \tilde{K}_0^{-1}\right)(\mathbf{b}^n - W\widehat{\mathbf{y}}) .$$

Remark 3.4. *Note that if we implement (3.10) in Theorem 3.2 straightforwardly, then the tensor rank tends to grow as the time step n increases. Hence, the assumption that $\forall n \geq 2$, the tensor rank of the right hand side $\mathbf{b}^n$ is at most ℓ, where $1 \leq \ell \ll JP$. In fact, in practical computations, the tensor rank of $\mathbf{y}^{n-1}$ is truncated with respect to its singular value decay to ensure that the tensor rank of $\mathbf{b}^n$ is kept under control. The decay rates of the singular values of the right hand sides and final solution (reshaped as $J \times P$ matrices) are numerically illustrated in Section 3.3.*

Remark 3.5. *We note here that Theorem 3.2 provides a theoretical evidence for the existence of low-rank tensor approximation to the solution of (2.37) as $JP \to \infty$.*

3.2.2 Preconditioning strategies

It is noteworthy that the stochastic Galerkin matrix generally suffers from poor conditioning; see, e.g. [110]. This can induce the deterioration of the rate of convergence of the Krylov subspace methods as the problem size increases. Nevertheless, this and other causes of slow convergence rates are typically remedied by the use of a suitable preconditioner $\mathcal{P}$. Conceptually, we need a matrix $\mathcal{P}$ such that $\mathcal{P}^{-1}\widehat{\mathcal{K}}_\tau$ has better spectral properties (essentially, clustered eigenvalues) and for which $\mathcal{P}^{-1}\mathbf{v}$ is cheap to compute for any vector $\mathbf{v}$ of appropriate dimension. In practice, though, we typically aim to preserve symmetry; this can certainly be achieved when $\mathcal{P}$ is symmetric positive definite [107].

Regardless of the preconditioners used, a major issue in solving (2.37) is evident. More precisely, one has to solve an enormous elliptic system in each timestep. Due to the coupled nature of the systems, this exercise can be both computer memory- and time-consuming (cf. Section 3.3). To mitigate this problem, we propose to solve (2.37) with two optimal preconditioners, together with a low-rank PCG method [78]. First, however, we introduce the preconditioners which we will use in what follows.

Mean-based preconditioner

Observe first that in general the submatrix matrix K_0 has a much more significant contribution than the other K_i's representing the random fluctuations of the system. Since K_0 only gives contributions to the main block diagonal of the global stochastic Galerkin matrix, the resulting system of linear equations will be strongly block-diagonally dominant if the random field a in (2.3) has small fluctuations away from its mean value. This important observation was exploited by Pellissetti and Ghanem in [106] to construct a block Jacobi preconditioner, which was subsequently termed the *mean-based preconditioner.* More precisely, the mean-based preconditioner is given by

$$\mathcal{P}_0 := G_0 \otimes \tilde{K}_0. \tag{3.15}$$

Now, observe that this is just the matrix $\mathcal{L}$ in Theorem 3.2 and that G_0 is a diagonal matrix due to the orthogonality of the stochastic basis functions $\{\psi_i\}$. Hence, $\mathcal{P}_0$ is a block-diagonal matrix. Moreover, by definition, $\tilde{K}_0 = M + \tau K_0$, so that $\tilde{K}_0$ is symmetric and positive definite since M and K_0 are both symmetric and positive definite from (2.35) and (2.25). So, $\mathcal{P}_0$ is positive definite and $\mathcal{P}_0^{-1} = G_0^{-1} \otimes \tilde{K}_0^{-1}$, where $G_0^{-1}(j,j) = 1/G_0(j,j) > 0$. The preconditioner then entails the approximate action of P uncoupled copies of $\tilde{K}_0^{-1}$.

Ullmann preconditioner

Ullmann in [130] points out that the mean-based preconditioner does not take into account all the information contained in $\widehat{\mathcal{K}}_\tau$ and thus proposes and analyses an optimal preconditioner which we refer to as the *Ullmann preconditioner* in the sequel. It is given by

$$\mathcal{P}_1 := \underbrace{\sum_{i=0}^{N} \frac{\operatorname{trace}(\tilde{K}_i^T \tilde{K}_0)}{\operatorname{trace}(\tilde{K}_0^T \tilde{K}_0)} G_i}_{:=G} \otimes \tilde{K}_0. \tag{3.16}$$

Observe from (3.16) and (3.15) that $\mathcal{P}_1$ can be thought of as a 'perturbed' version of $\mathcal{P}_0$ since

$$\mathcal{P}_1 = \underbrace{G_0 \otimes \tilde{K}_0}_{:=\mathcal{P}_0} + \sum_{i=1}^{N} \frac{\text{trace}(\tilde{K}_i^T \tilde{K}_0)}{\text{trace}(\tilde{K}_0^T \tilde{K}_0)} G_i \otimes \tilde{K}_0. \tag{3.17}$$

It is inspired by the first part of the following result obtained by Van Loan and Pitsianis.

Lemma 3.6. *[132] Suppose $m = m_1 m_2$, $n = n_1 n_2$, and $X \in \mathbb{R}^{m \times n}$. If $R \in \mathbb{R}^{m_2 \times n_2}$ is fixed, then the matrix $L \in \mathbb{R}^{m_1 \times n_1}$ defined by*

$$L_{i,j} := \frac{\text{trace}(X_{i,j}^T R)}{\text{trace}(R^T R)}, \quad i = 1, \ldots, m_1, \; j = 1, \ldots, n_1, \tag{3.18}$$

minimizes $||X - L \otimes R||_F$, where $X_{i,j}^T = X((i-1)m_2 + 1 : im_2, (j-1)n_2 + 1 : jn_2)$. Likewise, if $L \in \mathbb{R}^{m_1 \times n_1}$ is fixed, then the matrix $R \in \mathbb{R}^{m_2 \times n_2}$ defined by

$$R_{i,j} := \frac{\text{trace}(\tilde{X}_{i,j}^T L)}{\text{trace}(L^T L)}, \quad i = 1, \ldots, m_2, \; j = 1, \ldots, n_2, \tag{3.19}$$

minimizes $||X - L \otimes R||_F$, where $\tilde{X}_{i,j}^T = X(i : m_2 : m, j : n_2 : n)$.

Van Loan and Pitsianis further show that the matrices L defined in (3.18) and R defined in (3.19) are symmetric and positive definite provided X and R or L, respectively, are symmetric and positive definite.

Now, if we set $X = \widehat{\mathcal{K}}_\tau$ and $R = \tilde{K}_0$ in (2.37), it follows from (3.18) that the matrix G in (3.16) minimizes $||\widehat{\mathcal{K}}_\tau - G \otimes \tilde{K}_0||_F$. More interestingly, $\mathcal{P}_1$ inherits the sparsity pattern, symmetry and positive definiteness of the Galerkin matrix $\widehat{\mathcal{K}}_\tau$. Besides, unlike $\mathcal{P}_0$, it makes use of all the information in $\widehat{\mathcal{K}}_\tau$. Unfortunately, by reason of its construction, $\mathcal{P}_1$ loses the block-diagonal structure enjoyed by $\mathcal{P}_0$ which makes it more expensive to invert than $\mathcal{P}_0$.

3.2.3 Low-rank preconditioned CG method

Having presented the preconditioners, we proceed in this section to discuss the low-rank preconditioned conjugate gradient (LRPCG) method [78]. The basic idea behind LRPCG is that the iterates in the algorithm are truncated based on the decay of their singular values. Thus, at each iteration, the iterates are put in low-rank format (cf. (3.8)). The

Algorithm 3.2 Low-rank preconditioned conjugate gradient method (LRPCG)

```
Input: Matrix functions $\widehat{\mathcal{K}}_\tau, \mathcal{P} : \mathbb{R}^{J\times P} \to \mathbb{R}^{J\times P}$, right hand side $B^n \in \mathbb{R}^{J\times P}$ in low-rank format. Truncation operator $\mathcal{T}_\varepsilon$ w.r.t relative accuracy $\varepsilon$.
Output: Matrix $\mathbf{y}^n \in \mathbb{R}^{J\times P}$ fulfilling $||\widehat{\mathcal{K}}_\tau(\mathbf{y}^n) - B^n||_F \leq$ tol.
$\mathbf{y}_0^n = 0,\ R_0 = B^n,\ Z_0 = \mathcal{P}^{-1}(R_0),\ P_0 = Z_0,\ Q_0 = \widehat{\mathcal{K}}_\tau(P_0)$,
$\vartheta_0 = \langle P_0, Q_0 \rangle,\ k = 0$.
while $||R_k||_F >$ tol do
    $\omega_k = \langle R_k, P_k \rangle / \vartheta_k$
    $\mathbf{y}_{k+1}^n = \mathbf{y}_k^n + \omega_k P_k$,            $\mathbf{y}_{k+1}^n \leftarrow \mathcal{T}_\varepsilon(\mathbf{y}_{k+1}^n)$
    $R_{k+1} = B^n - \widehat{\mathcal{K}}_\tau(\mathbf{y}_{k+1}^n)$,      $Optionally : R_{k+1} \leftarrow \mathcal{T}_\varepsilon(R_{k+1})$
    $Z_{k+1} = \mathcal{P}^{-1}(R_{k+1})$
    $\beta_{k+1} = -\langle Z_{k+1}, Q_k \rangle / \vartheta_k$
    $P_{k+1} = Z_{k+1} + \beta_k P_k$,            $P_{k+1} \leftarrow \mathcal{T}_\varepsilon(P_{k+1})$
    $Q_{k+1} = \widehat{\mathcal{K}}_\tau(P_{k+1})$,            $Optionally : Q_{k+1} \leftarrow \mathcal{T}_\varepsilon(Q_{k+1})$
    $\vartheta_{k+1} = \langle P_k, Q_k \rangle$
    $k = k + 1$
end while
$\mathbf{y}^n = \mathbf{y}_k^n$
```

truncation, no doubt, introduces further error in the solution. However, the truncation tolerance can be so tightened that the error becomes negligible. More importantly, the computer memory required to store the matrices is reduced, thereby enabling large-scale computations.

First, we present LRPCG in Algorithm 3.2. We point out a few things regarding the implementation of LRPCG with respect to the solution of (2.37). Note that, in Algorithm 3.2, all vectors in $\mathbb{R}^{J\cdot P}$ (cf. (2.37)) are reshaped into $\mathbb{R}^{J\times P}$ matrices by the vec^{-1} operator. Now, recall that for each fixed time step $n = 1, 2, \ldots, n_t$, we need to solve an elliptic system using the LRPCG algorithm. In particular, for each solve, we need to evaluate $\widehat{\mathcal{K}}_\tau(X) = \widehat{\mathcal{K}}_\tau \text{vec}(X)$, where $X := \mathbf{y}_k^n$ or P_k. For this purpose, we set

$$\widehat{\mathcal{K}}_\tau \text{vec}(X) = \left(\sum_{i=0}^{N} G_i \otimes \tilde{K}_i \right) \text{vec}(X), \tag{3.20}$$

where $X \in \mathbb{R}^{J\times P}$ is of low-rank, say, r :

$$\begin{aligned} X &= WV^T,\ W \in \mathbb{R}^{J\times r},\ V \in \mathbb{R}^{P\times r},\ r \ll J, P, \\ W &= [w_1, \ldots, w_r],\ V \in [v_1, \ldots, v_r], \end{aligned}$$

so that, using (1.2), one gets

$$\mathrm{vec}(X) = \mathrm{vec}\left(\sum_{i=1}^{r} u_j v_j^T\right) = \sum_{j=1}^{r} \mathrm{vec}(u_j v_j^T) = \sum_{j=1}^{r} v_j \otimes u_j. \tag{3.21}$$

Hence, we have

$$\begin{aligned} \widehat{\mathcal{K}}_\tau \mathrm{vec}(X) &= \left(\sum_{i=0}^{N} G_i \otimes \tilde{K}_i\right) \mathrm{vec}(X) \\ &= \left(\sum_{i=0}^{N} G_i \otimes \tilde{K}_i\right)\left(\sum_{j=1}^{r} v_j \otimes u_j\right) \\ &= \sum_{i=0}^{N}\sum_{j=1}^{r} (G_i v_j) \otimes (\tilde{K}_i u_j) := \sum_{k=1}^{(N+1)r} \widehat{v}_k \otimes \widehat{u}_k \in \mathbb{R}^{JP\times 1}, \end{aligned} \tag{3.22}$$

where $\widehat{v}_k \in \mathbb{R}^P$, $\widehat{u}_k \in \mathbb{R}^J$ and we then have to reshape (3.22) to have

$$\widehat{\mathcal{K}}_\tau(X) := \mathrm{vec}^{-1}(\widehat{\mathcal{K}}_\tau \mathrm{vec}(X)) \in \mathbb{R}^{J\times P}. \tag{3.23}$$

Moreover, in order to apply any of the two preconditioners to the residual matrices R_k, that is, $\mathcal{P}^{-1}(R_k)$, we have to ensure that the R_k are in low-rank format as in (3.21), so we can obtain similar expressions as in (3.22) and (3.23), since $\mathcal{P}^{-1} := \mathcal{P}_i^{-1},\ i = 0, 1$, have the same size and Kronecker product structure as $\widehat{\mathcal{K}}_\tau$. The right hand side of (2.37), that is, $\mathbf{b}^n = (G_0 \otimes M)\,\mathbf{y}^{n-1} + \tau\,(\mathbf{g}_0 \otimes \mathbf{u}_0)$ is also reshaped such that $B^n := \mathrm{vec}^{-1}(\mathbf{b}^n) \in \mathbb{R}^{J\times P}$ and B^n is as well put in low-rank format. Finally, observe from (1.4) that the operation in (3.22) has increased the tensor rank of the resulting vector. Hence, it is important that the iterates $\mathbf{y}_k^n$ are truncated in every iteration by a truncation operator $\mathcal{T}_\varepsilon$ based on the decay of their singular values in order to keep the growth of the ranks under control. In the sequel, we describe how the truncation operation works, as well as how to exploit the low-rank format of the matrices to compute the inner products in Algorithm 3.2.

3.2.4 Truncation and matrix inner products

We start this section by assuming that the matrix of interest X is represented by two low-rank factors W and V, i.e., $X = WV^T$. Our iterative procedure starts with a low-rank decomposition of the right hand side, but the ranks of the low-rank factors increase either via the low-rank matrix vector products or vector recurrences. For this purpose, it is necessary to find new low-rank approximations $\tilde{W}$ and $\tilde{V}$ that approximate the old product $WV^T \approx \mathcal{T}_\varepsilon(X) = \tilde{W}\tilde{V}^T$, where the truncation operator $\mathcal{T}_\varepsilon$ satisfies

$$||X - \mathcal{T}_\varepsilon(X)|| \leq \varepsilon ||X||_F,$$

for some truncation tolerance ε. The columns of the matrix $\mathcal{T}_\varepsilon(X)$ have been *compressed,* and for this reason this operation is sometimes referred to as a column compression.

Kressner and Tobler discuss in [77] that one can obtain the new low-rank representation by performing skinny QR factorizations of both matrices, i.e., $W = Q_w R_w$ and $V = Q_v R_v$. We then note that $X = Q_w R_w R_v^T Q_v^T$ and an SVD of $R_w R_v^T = B \Sigma C^T$ allows us to compute a representation of lower rank. Depending on the truncation tolerance we can drop small singular values in Σ. The new low-rank factors are then obtained via

$$\tilde{W} = Q_w B(:, 1:r) \text{ and } \tilde{V} = Q_v C(:, 1:r)\Sigma(1:r, 1:r),$$

where we have used MATLAB notation. Here, the truncation rank $r' \leq r$ is chosen such that the singular values s_r satisfy

$$\sqrt{s_{r'+1}^2 + \ldots + s_r^2} \leq \varepsilon \sqrt{s_1^2 + \ldots + s_r^2},$$

where ε is the truncation tolerance. This operation leads to $X \approx \tilde{W}\tilde{V}^T$. We have implemented this approach in MATLAB but noted that the computation of the skinny QR factorization was rather slow. An alternative approach that we used, which due to some internal handling within MATLAB typically produces fast results, exploits the MATLAB function `svds` to directly compute, via a function handle, the truncated SVD of $WV^T \approx B\Sigma C^T$ without ever forming the matrix (see also [124]). Again, we drop small singular values in Σ to obtain $\tilde{V}$ and $\tilde{W}$. The computation of the truncated SVD is typically done via a procedure based on a Krylov subspace method where we require multiplication with the matrix WV^T. It is easy to see that we can perform this multiplication using the matrix factored form. This approach proved advantageous in terms of the time needed for the truncation. Alternative ways to compute the truncated SVD are possible and can be found in [6, 56, 123]. The cost of computing the truncation depends, for example in the truncated SVD approach, on the cost of multiplying with the matrix WV^T. Assuming that $W \in \mathbb{R}^{J\times r}$, then every iteration of an iterative procedure to compute the truncated SVD needs $\mathcal{O}(Jr)$ flops to compute the multiplication with W and analogously $\mathcal{O}(Pr)$ for the multiplication with V^T.

Additionally, we have to ensure that the inner products within the iterative solver are computed efficiently. To that end, suppose that

$$\begin{aligned} Y &= W_Y V_Y^T, \quad W_Y \in \mathbb{R}^{J\times r_Y},\ V_Y \in \mathbb{R}^{P\times r_Y}, \\ Z &= W_Z V_Z^T, \quad W_Z \in \mathbb{R}^{J\times r_Z},\ V_Z \in \mathbb{R}^{P\times r_Z}. \end{aligned}$$

Then, due to the properties of the trace operator[2], we see that

$$\text{trace}\left(\underbrace{\left(W_Y V_Y^T\right)^T}_{\text{Large}}\underbrace{\left(W_Z V_Z^T\right)}_{\text{Large}}\right) = \text{trace}\left(\underbrace{V_Z^T V_Y}_{\text{Small}}\underbrace{W_Y^T W_Z}_{\text{Small}}\right) \tag{3.24}$$

allows us to compute the trace of small matrices rather than of the ones from the full model. More precisely, we first compute $V_Z^T V_Y \in \mathbb{R}^{r_Z\times r_Y}$ with $(2Jr_Y r_Z)$ flops, $W_Y^T W_Z \in \mathbb{R}^{r_Y\times r_Z}$ with $(2Pr_Y r_Z)$ flops, and then the diagonal elements of the product of the two matrices with $(2r_Y r_Z)$ flops. In total, we therefore require $2(J+P+1)r_Y r_Z$ flops.

From a numerical analyst's point of view, one may object that low-rank truncations introduce an error in the LRPCG procedure as given by Algorithm 3.2, so that any orthogonality or optimality properties of the conjugate gradient method that are due to recursions would be lost, and this is certainly a valid point. In fact, as noted in [77], the residual must be explicitly calculated as $R_{k+1}^n = B^n - \widehat{\mathcal{K}}_\tau(\mathbf{y}_{k+1}^n),\ n = 0,1,2,\ldots,$ in order to ensure numerical stability, and this requires a modification in the derivation of Algorithm 3.2 so that coefficients do not depend on any recursions for the residual, unlike in the vector-based Algorithm 3.1. Despite such an objection, the method is capable of delivering quality approximations while dramatically reducing memory requirements. Having discussed the low-rank solver, we proceed to the next section to investigate its performance in conjunction with the preconditioners.

[2] Recall that $\langle Y, Z\rangle = \text{vec}\,(Y)^T\,\text{vec}\,(Z) = \text{trace}\,\left(Y^T Z\right)$.

3.3 Numerical experiments

To demonstrate the performance of the low-rank approach presented in this chapter, we consider the 2D version of our model problem (2.33). More precisely, we choose $f = 1$ and $\mathcal{D} = [-1, 1]^2$. The random input a is characterized by the covariance function

$$C_a(\mathbf{x}, \mathbf{y}) = \sigma^2 \exp\left(-\frac{|x_1 - y_1|}{\ell_1} - \frac{|x_2 - y_2|}{\ell_2} \right), \quad \forall (\mathbf{x}, \mathbf{y}) \in \mathcal{D}. \tag{3.25}$$

The eigenpairs (λ_j, ϑ_j) of the KLE of a are given explicitly in [43]. In the simulations, we set the correlation lengths $\ell_1 = \ell_2 = 1$ and the mean of the random field $\mathbb{E}(a) = 1$. Note that decreasing the correlation lengths slows down the decay of the eigenvalues in the KLE of a, and therefore more random variables are then required to sufficiently capture the randomness in the model [110]. That is, the resulting effect is an increase in the parameter N. The reverse is the case when the correlation lengths are increased.

Next, we investigate the behavior of the solvers for different values of the discretization parameters J, N, n, σ_a. Moreover, we choose $\xi = \{\xi_1, \ldots, \xi_N\}$ such that $\xi_j \sim \mathcal{U}[-1, 1]$, and $\{\psi_j\}$ are N-dimensional Legendre polynomials with support in $[-1, 1]^N$. We perform spatial discretization using $\mathbf{Q}1$ finite elements. Moreover, all the numerical experiments are performed on an Ubuntu Linux machine with 2GB RAM using MATLAB 7.14 together with a MATLAB version of `HSL MI20` [21] based on the classical AMG method as described in [127]. We implement each of the two preconditioners $\mathcal{P}_0$ and $\mathcal{P}_1$ using one V-cycle of AMG with symmetric Gauss-Seidel (SGS) smoothing to approximately invert $\tilde{K}_0$. We remark here that we apply the method as a black-box in each experiment and the set-up of the approximation to $\tilde{K}_0$ only needs to be performed once. In the experiments, the linear systems are solved for time $T = 1$ and 16 timesteps. We write DoF to mean the total degrees of freedom for the matrix $\widehat{\mathcal{K}}_\tau$; that is; $\text{DoF}(\widehat{\mathcal{K}}_\tau) = JP$. All figures are obtained with the mean-based preconditioner $\mathcal{P}_0$. Unless otherwise stated, all iterations for all solvers herein are terminated when the relative residual error, measured in the Euclidean norm, is reduced to $tol = 10^{-4}$. We remark here that the stopping iteration tolerance tol should be chosen such that the truncation tolerance $\varepsilon \leq tol$; otherwise, one would be essentially iterating on the 'noise' from the low-rank truncations, as it were.

First, in Figures 3.2, 3.3 and 3.4, we illustrate, for the 2D model problem, the singular values decay of the stochastic matrix G_1, as well as those of the right hand sides at different time steps and the final solution at $T = 1$. In these figures, we see that the decay is slow. Nevertheless, the matrix G_1 is rank deficient, which justifies its low-rank representation in Theorem 3.2. We note here that the singular values of the stochastic matrices G_k are indeed the same since the matrices are permutations of one another [110] and, hence, their ranks are equal. In particular, their rank is roughly $P/2$ for all $k > 0$. However, as already pointed out, G_0 is diagonal and of full rank P.

As an illustration of the results of Theorem 3.2, observe first from Figure 3.2 that the rank of the matrices G_k (represented here by G_1) is 32 while $P = 56$. Now, recall from the theorem that the rank of the low-rank solution is determined mainly by the ranks of the stochastic matrices G_k regardless of the dimension of the stiffness matrices $\tilde{K}_k$. More precisely, we have from the figure and the theorem that $r = \sum_{j=1}^{N} r_j = 5 \times \operatorname{rank}(G_1) = 5 \times 32 = 160$. Thus, with the truncation tolerance $\varepsilon = 10^{-10}$, for example, we see from Figure 3.2 that the tensor ranks of the right hand sides $\mathbf{b}^n$ are at most ℓ, where $\ell = 20$. Hence, one can approximate the solution to the linear systems with a solution vector whose tensor rank at each time step is at most $160 + \ell = 180$. So, it turns out that the result of the theorem is particularly important if the size of the stiffness matrices $\tilde{K}_k$ increases; that is, $J \to \infty$, while P is kept constant. As for the right hand sides, the decays at all the

Figure 3.2: Singular values decay of the stochastic matrix G_1 (left) and the right hand sides at different times $t \in \{0.125, 0.25, 0.5\}$, as well as the final solution at $t = T$ (right), $\mathrm{DoF}(\widehat{\mathcal{K}}_\tau) = 340480$ with $J = 6080,\ P = 56$ (i.e. $N = 5, n = 3), \sigma_a = 0.01$ and $tol = 10^{-8}$.

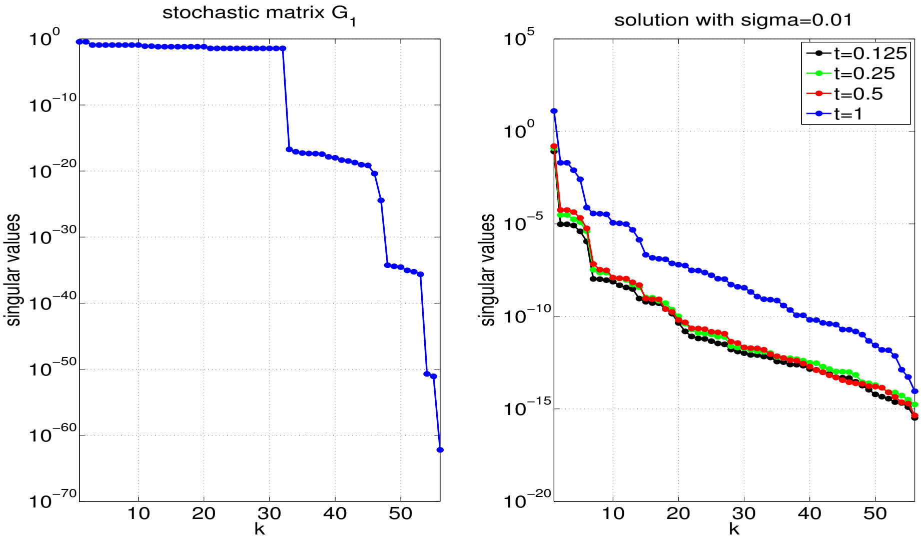


Figure 3.3: Singular values decay of the right hand sides at different times $t \in \{0.125, 0.25, 0.5\}$, as well as the final solution at $t = T$, with $\sigma_a = 0.1$ (left) and $\sigma_a = 0.5$ (right), $\mathrm{DoF}(\widehat{\mathcal{K}}_\tau) = 340480$ with $J = 6080,\ P = 56$ (i.e. $N = 5, n = 3$), and $tol = 10^{-8}$.

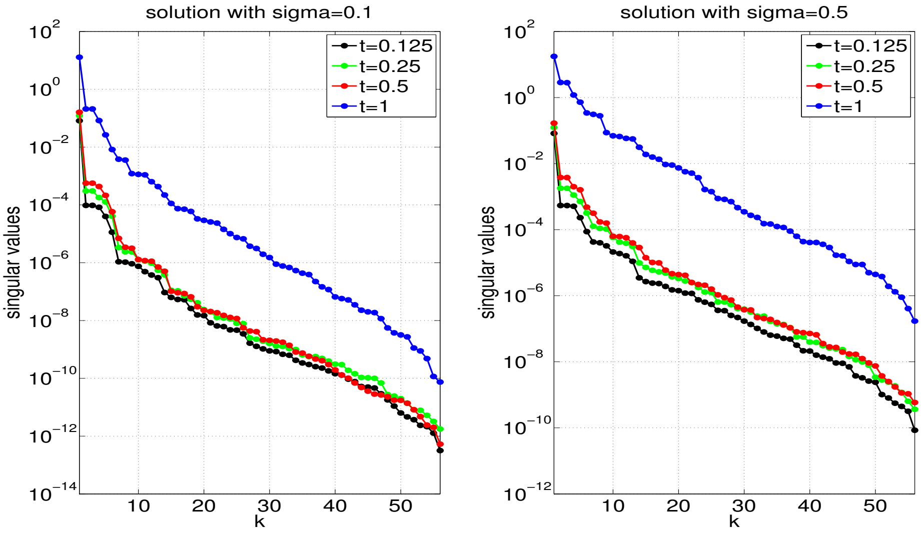
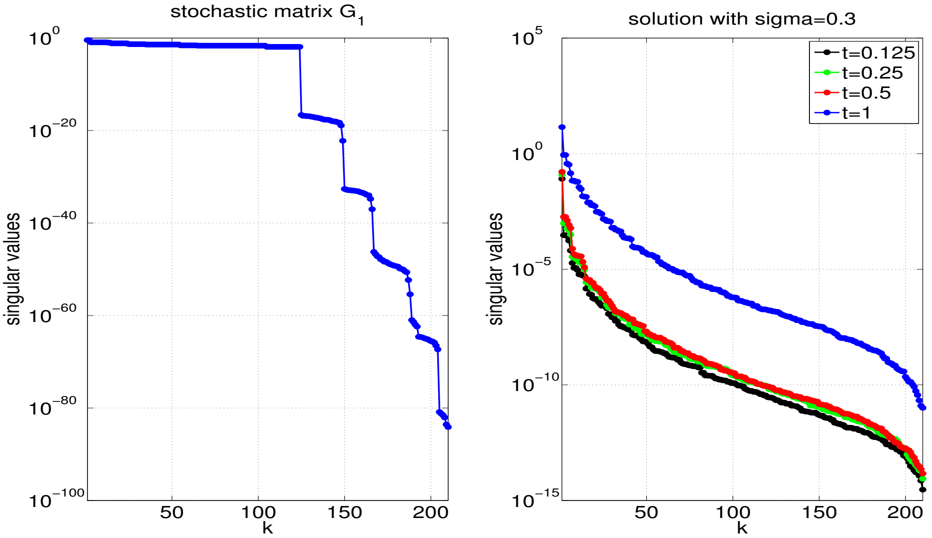


Figure 3.4: Singular values decay of the stochastic matrix G_1 (left) and the right hand sides at different times $t \in \{0.125, 0.25, 0.5\}$, as well as the final solution at $t = T$, (right), $\mathrm{DoF}(\widehat{\mathcal{K}}_\tau) = 1276800$ with $J = 6080,\ P = 210$ (i.e.$N = 6, n = 4$), $\sigma_a = 0.3$ and $tol = 10^{-8}$.



respective time steps (e.g. $t = 0.125, 0.25, 0.5$) are quite similar. Thus, we truncate the right hand sides with the same truncation tolerance. Figures 3.2 and 3.3, in particular, illustrate that, keeping other parameters fixed, increasing the variance of the random field a slows down the decay of the singular values of both the right hand sides and the final solution.

Tables 3.1, 3.2, 3.3, 3.4 and 3.5 report further the results of the simulations of the model, keeping all but one parameter constant in each table. Here, the linear systems are solved using LRPCG, as well as using the standard preconditioned CG method which we have denoted as full model (FM), that is, without low-rank truncation. As benchmarks to

Table 3.1: Simulation results showing relative errors, total CPU times (in seconds), ranks of truncated solutions, memory (in KB), and total number of iterations from preconditioned low-rank solvers (second and third columns) compared with those from standard preconditioned CG (last two columns) for $\sigma_a = 0.01$, $T = 1$, $J = 6080$, and various P; Par represents the tuple (N, n, P).

Timesteps=16	$\mathcal{P}_0$ + LRPCG	$\mathcal{P}_1$ + LRPCG	$\mathcal{P}_0$ + FM	$\mathcal{P}_1$ + FM
Truncation tol	$10^{-4}(10^{-6})$	$10^{-4}(10^{-6})$		
Par=(3,3,20)				
Ranks	6 (8)	6 (10)		
Memory	381.3 (524.2)	285.9 (571.9)	950	950
#iter	32 (32)	32 (32)	32	32
Total CPU time	20.4 (21.9)	20.7 (21.1)	119.4	123.6
Rel error	8.0e-5 (8.9e-6)	2.4e-4 (1.1e-6)		
Par=(5,3,56)				
Ranks	9 (12)	9 (16)		
Memory	527.3 (814.9)	431.4 (910.8)	2660	2660
#iter	32 (32)	32 (32)	32	32
Total CPU time	52.4 (58.0)	54.7 (58.8)	197.0	195.1
Rel error	2.2e-4 (1.2e-5)	4.0e-4 (4.1e-6)		
Par=(4,4,70)				
Ranks	8 (10)	8 (13)		
Memory	480.5 (672.6)	384.4 (768.7)	3325	3325
#iter	33 (32)	32 (33)	32	32
Total CPU time	54.5 (52.7)	54.5 (57.3)	208.5	208.3
Rel error	8.0e-5 (1.3e-5)	3.3e-4 (3.8e-6)		
Par=(6,3,84)				
Ranks	9 (14)	10 (18)		
Memory	577.9 (866.8)	481.6 (1059.4)	3990	3990
#iter	32 (32)	32 (32)	32	32
Total CPU time	139.6 (133.1)	112.5 (156.1)	228.1	229.9
Rel error	3.0e-4 (1.3e-5)	4.4e-4 (4.3e-6)		

compare the performance of the solution methods, we report the total iteration counts, the total CPU times, memory requirements (in kilobytes), the ranks of the truncated solutions and the relative error from the LRPCG solution with respect to the FM solution, measured in the Euclidean norm. By the memory requirement of a low-rank solution $X = WV^T$, we mean the sum of the two separate computer memories occupied by its factors W and V^T, since X is computed and stored in this format, unlike the solution from FM.

In Tables 3.1, 3.2, and 3.5, we show results for varying $P, J,$ and σ_a, respectively, while keeping other parameters constant. In all the tables reported in this chapter, the second and the third columns show the outputs from LRPCG while the last two are from FM (using just the MATLAB command `pcg`). Also in the second and third columns, the quantities in brackets are outputs computed with the corresponding truncation tolerance. Note that in Tables 3.1 and 3.4, we have specifically used the tuple of parameters (N, n, P). Thus, $(5, 3, 56)$, for example, implies that $N = 5, n = 3$, and $P = 56$ (cf. (2.18)).

A major observation from Tables 3.1, 3.2, 3.3 and 3.5 is that for $\sigma_a \leq 0.2$ and independently of the preconditioner used, LRPCG clearly outperforms FM in terms of CPU times and memory requirements, while maintaining fairly the same iterations as FM. From Tables 3.2 and 3.3, the efficiency as $J \to \infty$ of the LRPCG compared to FM with respect to CPU times and memory reduction is particularly noteworthy. For instance, if $J = 24448$ and $\varepsilon = 10^{-6}$, we see from Table 3.2 that the low-rank approach reduces the computational time by roughly a factor of 10 and memory required to store the solution by a factor of 4, while maintaining the same iteration counts as FM. In fact, this observation further corroborates the theoretical implication of Theorem 3.2 that the low-rank approach is of particular interest if the size of the stiffness matrices $\tilde{K}_k$ gets arbitrarily large; the FM deteriorates in this case as it suddenly struggles to cope with the increased computational complexity. Note in particular from Table 3.3 that with the FM, MATLAB indeed fails with $J = 392704$ and $P = 210$, as the size of the global stochastic Galerkin matrix $\widehat{\mathcal{K}}_\tau$ at each timestep is now increased to more than 82 million degrees of freedom. Yet, LRPCG handles this task in about 200 minutes with $\varepsilon = 10^{-6}$, $\sigma_a = 0.1$; that is, roughly 13 minutes per timestep. In this case, however, we are not able to report the relative error unlike in the other tables because the solution from FM terminates with 'out of memory', which we have denoted as 'OoM'. On the other hand, if J is relatively small and P is varied as

Table 3.2: Simulation results showing relative errors, total CPU times (in seconds), ranks of truncated solutions, memory (in KB), and total number of iterations from preconditioned low-rank solvers (second and third columns) compared with those from standard preconditioned CG (last two columns) for $N = 6$, $n = 3$, (i.e $P = 84$), $\sigma_a = 0.01$, and various J.

Timesteps=16 Truncation tol	$\mathcal{P}_0$ + LRPCG $10^{-4}(10^{-6})$	$\mathcal{P}_1$ + LRPCG $10^{-4}(10^{-6})$	$\mathcal{P}_0$ + FM	$\mathcal{P}_1$ + FM
$J = 368$				
Ranks	10 (14)	10 (17)		
Memory	42.4 (68.1)	42.7 (77.7)	241.5	241.5
#iter	32 (32)	32 (32)	32	32
Total CPU time	35.0 (41.1)	45.9 (43.5)	10.2	14.2
Rel error	3.0e-4 (1.2e-5)	4.2e-4 (6.0e-6)		
$J = 1504$				
Ranks	9 (14)	10 (17)		
Memory	148.8 (223.3)	124.1 (260.5)	987	987
#iter	32 (32)	32 (33)	32	32
Total CPU time	64.8 (66.5)	69.7 (70.0)	21.8	27.2
Rel error	3.0e-4 (1.2e-5)	4.4e-4 (6.0e-6)		
$J = 6080$				
Ranks	9 (14)	10 (18)		
Memory	577.9 (866.8)	481.6 (1059.4)	3990	3990
#iter	32 (32)	32 (32)	32	32
Total CPU time	139.6 (133.1)	112.5 (156.1)	228.1	229.9
Rel error	3.0e-4 (1.3e-5)	4.4e-4 (4.3e-6)		
$J = 24448$				
Ranks	9 (14)	10 (18)		
Memory	2299.9 (3449.8)	1916.5 (4216.4)	16044	16044
#iter	32 (32)	32 (32)	32	32
Total CPU time	352.0 (426.7)	347.8 (419.4)	3769.4	3853.4
Rel error	3.0e-4 (1.3e-5)	4.5e-4 (4.3e-6)		

in Table 3.4, then FM does better than LRPCG in terms of CPU time only. Although reported only for the case $\sigma_a = 0.01$ in Table 3.2, we also observed a similar trend as σ_a is varied and a small J is kept constant. But then, in practical applications one is usually more interested in large-scale simulations in which case (J and P are large and) LRPCG will naturally be a preferred option. Another key observation evident from all the tables is that decreasing the truncation tolerance generally reduces the relative error but, as expected, at the cost of comparatively more computational time and memory requirements. Regarding the preconditioners, we note that, compared to the Ullmann preconditioner $\mathcal{P}_1$, the mean-based preconditioner $\mathcal{P}_0$ generally yields lower ranks of the low-rank solution,

Table 3.3: Simulation results showing total CPU times (in seconds), ranks of truncated solutions, memory (in KB), and total iterations using low-rank preconditioned CG for $\mathrm{DoF}(\widehat{\mathcal{K}}_\tau) \approx 82.5 \times 10^6$ with $J = 392704$, $P = 210$ (i.e $N = 6$, $n = 4$), and various σ_a.

Timesteps=16 Truncation tol	$\mathcal{P}_0$ + LRPCG $10^{-4}(10^{-6})$	$\mathcal{P}_1$ + LRPCG $10^{-4}(10^{-6})$	$\mathcal{P}_0$ or $\mathcal{P}_1$ + FM
$\sigma_a = 0.001$			
Ranks	1 (10)	2 (12)	
Memory	3069.6 (49114.25)	6139.2 (49114.25)	
#iter	24 (32)	32 (32)	
Total CPU time	2680.7 (4775.5)	3335.4 (4944.9)	OoM
$\sigma_a = 0.01$			
Ranks	9 (12)	10 (18)	
Memory	36835.7 (55253.5)	30696.4 (67532.1)	
#iter	32 (32)	32 (32)	
Total CPU time	4157.3 (5149.9)	4115.3 (5249.8)	OoM
$\sigma_a = 0.1$			
Ranks	20 (47)	20 (52)	
Memory	89019.6 (174969.5)	82880.3 (171899.9)	
#iter	49 (49)	51 (48)	
Total CPU time	8354.5 (12419.0)	8069.3 (11801.0)	OoM

Table 3.4: Simulation results showing relative errors, total CPU times (in seconds), ranks of truncated solutions, memory (in KB), and total number of iterations from preconditioned low-rank solvers (second and third columns) compared with those from standard preconditioned CG (last two columns) for $\sigma_a = 0.01$, $J = 1504$, and various P; Par represents the tuple (N, n, P).

Timesteps=16 Truncation tol	$\mathcal{P}_0$ + LRPCG $10^{-4}(10^{-6})$	$\mathcal{P}_1$ + LRPCG $10^{-4}(10^{-6})$	$\mathcal{P}_0$ + FM	$\mathcal{P}_1$ + FM
Par=(3,3,20)				
Ranks	6 (8)	6 (10)		
Memory	95.25 (130.9)	71.4 (142.9)	235	235
#iter	32 (32)	32 (32)	32	32
Total CPU time	19.4 (17.0)	19.9 (18.1)	12.7	15.3
Rel error	8.0e-5 (8.7e-6)	2.4e-4 (1.0e-6)		
Par=(5,3,56)				
Ranks	9 (12)	9 (16)		
Memory	134.1 (207.2)	109.7 (243.8)	658	658
#iter	33 (32)	32 (33)	32	32
Total CPU time	63.8 (69.4)	69.2 (69.5)	20.0	23.6
Rel error	2.2e-4 (1.2e-5)	3.9e-4 (4.1e-6)		

less CPU times and less memory requirements for small truncation tolerance. However, both of them maintain relatively equal iteration counts either with LRPCG or FM.

In spite of the advantages enjoyed by LRPCG as outlined above, its performance is adversely affected by increase in the standard deviation σ_a of the input data. This observation is also true for FM, albeit to a lesser degree. It is indeed evident from Table 3.5 that both LRPCG and FM exhibit deteriorating performance as σ_a increases, regardless of which of the two preconditioners (that is, $\mathcal{P}_0$ or $\mathcal{P}_1$) is used. Accordingly, the decay of singular values of the solution matrices becomes slower and slower as earlier demonstrated by Figures 3.2, 3.3 and 3.4. Furthermore, as we can see from Table 3.5, even though relatively high variance limits the performance of both considered preconditioners, $\mathcal{P}_0$ tends to be more adversely affected by the increase than $\mathcal{P}_1$ in terms of both iteration counts and CPU time. This is perhaps explained by the fact that, unlike $\mathcal{P}_1$, the mean-based

Table 3.5: Simulation results showing relative errors, total CPU times (in seconds), ranks of truncated solutions, memory (in KB), and total number of iterations from preconditioned low-rank solvers (second and third columns) compared with those from standard preconditioned CG (last two columns) for $J = 6,080$, $N = 6$, $n = 3$, (i.e $P = 84$) and various σ_a.

Timesteps=16 Truncation tol	$\mathcal{P}_0$ + LRPCG $10^{-4}(10^{-6})$	$\mathcal{P}_1$ + LRPCG $10^{-4}(10^{-6})$	$\mathcal{P}_0$ + FM	$\mathcal{P}_1$ + FM
$\sigma_a = 0.01$				
Ranks	9 (14)	10 (18)		
Memory	577.9 (866.8)	481.6 (1059.4)	3990	3990
#iter	32 (32)	32 (32)	32	32
Total CPU time	139.6 (133.1)	112.5 (156.1)	228.1	229.9
Rel error	3.0e-4 (1.3e-5)	4.4e-4 (4.3e-6)		
$\sigma_a = 0.1$				
Ranks	27 (54)	21 (55)		
Memory	2070.7 (2744.9)	1348.4 (2696.8)	3990	3990
#iter	49 (49)	48 (48)	49	48
Total CPU time	206.0 (275.7)	196.4 (283.7)	342.6	352.6
Rel error	8.7e-4 (1.1e-4)	9.0e-4 (2.1e-4)		
$\sigma_a = 0.2$				
Ranks	46 (73)	49 (77)		
Memory	3033.8 (3804.3)	2985.7 (3804.3)	3990	3990
#iter	65 (72)	65 (64)	65	64
Total CPU time	200.6 (353.8)	218.6 (286.0)	508.9	524.9
Rel error	1.3e-4 (3.2e-4)	9.3e-4 (2.9e-6)		
$\sigma_a = 0.3$				
Ranks	81 (84)	84 (84)		
Memory	5393.5 (5700.6)	6308.5 (5778.8)	3990	3990
#iter	102 (242)	90 (108)	83	80
Total CPU time	911.6 (5478.7)	750.6 (1251.9)	590.1	835.1
Rel error	1.0e-3 (2.8e-4)	9.5e-4 (3.7e-4)		

preconditioner $\mathcal{P}_0$ is block-diagonal; thus, as σ_a increases, we see from (2.25), (2.26) and (2.37) that the off-diagonal blocks of the global stochastic Galerkin matrix $\widehat{\mathcal{K}}_\tau$ become more significant and they are not represented in the preconditioner. Here, we note, in particular, that the deteriorating performance of $\mathcal{P}_0$ as variance increases confirms a similar observation made in an earlier study [110] by Powell and Elman in which CG was preconditioned with $\mathcal{P}_0$, but without low-rank truncations. Due to this drawback, we remark here that we have done most of our computations using relatively small values of variance. In particular, we used $\sigma_a = 0.01$ to obtain the results in Tables 3.1, 3.2 and 3.4. We also did further experiments with $\sigma_a \in \{0.1, 0.2\}$ and $\varepsilon = 10^{-8}$ and made similar observations in the performance of LRPCG and FM.

In summary, with a view to reducing the computational time and memory requirements of the solution of arbitrarily large stochastic Galerkin linear systems, we have provided a theoretical basis for a low-rank solver to achieve these goals. More precisely, we solved the linear systems (2.37) using a low-rank conjugate gradient solver, together with two different preconditioners. In general, the combination of each of the preconditioners and the low-rank iterative solver seems quite promising for large-scale simulation of models whose random input data have comparatively low variance, as it reduces the computer memory and computational time required to solve the stochastic Galerkin linear system compared to the conventional method.

In the rest of the thesis, we proceed to develop efficient low-rank solvers for SOCPs, which are a class of higher dimensional problems. First, in Chapter 4, we discuss diffusion SOCPs after which we proceed to Chapter 5 to consider a Stokes-Brinkman SOCP.

Chapter 4

Diffusion optimal control problems with uncertain inputs

In this chapter and the next, we focus on the numerical simulation of SOCPs. As was pointed out by Rosseel and Wells in [115], the SGFEM can be applied straightforwardly to stochastic optimal control problems. However, the efficient solution of SGFEM problems can hinge on the development and application of effective preconditioners. This is, indeed, our major concern in this chapter. More specifically, we construct efficient preconditioners to be used with *all-at-once* low-rank Krylov subspace solvers for the optimality (saddle point) systems arising from SGFEM discretization of SOCPs[1].

4.1 Optimization under uncertainty

In many applications, forces or boundary conditions are to be determined such that the response of a physical or engineering system is optimal in some sense. These problems can often be formulated as the minimization of an objective functional subject to a set of constraint equations in the form of PDEs. For problems that involve uncertainty, incorporating stochastic information into a control formulation can lead to a quantification of the statistics of the system response. Mathematically speaking, stochastic PDE-constrained optimization problems or SOCPs are closely related to stochastic inverse problems, where the control variable corresponds to the parameter to be identified [64, 146].

[1] Instead of solving the linear systems iteratively for each time step in the unsteady problems considered here, we solve for all time steps at once.

In this dissertation, we will formulate our model SOCPs as follows:

$$\min_{y\in\mathcal{Y},u\in\mathcal{U}} \mathcal{J}(y,u) \text{ subject to } c(y,u)=0, \tag{4.1}$$

where the constraint equation $c(y,u)=0$ represents a PDE with uncertain coefficient(s) to be specified in the sequel, and $\mathcal{J}:\mathcal{Y}\times\mathcal{U}\to\mathbb{R}$ is a real-valued differentiable cost functional of tracking type and $\mathcal{Y},\mathcal{U}$ represent some suitably defined function spaces. Specifically, we consider this cost functional:

$$\mathcal{J}(y,u):=\frac{1}{2}||y-\bar{y}||^2_{L^2(\mathcal{D})\otimes L^2(\Omega)}+\frac{\alpha}{2}||\mathrm{std}(y)||^2_{L^2(\mathcal{D})}+\frac{\beta}{2}||u||^2_{L^2(\mathcal{D})\otimes L^2(\Omega)}. \tag{4.2}$$

In what follows, we treat the functions y,u and $\bar{y}$ as real-valued random fields representing, respectively, the *state variable,* the *control variable* and the *prescribed target system response.* Thus, the objective functional $\mathcal{J}(y,u)$ is a deterministic quantity with uncertain terms. We note here that y and u are the solutions variables; only $\bar{y}$ will be given. Moreover, $\bar{y}$ and u could be modelled deterministically. However, as mentioned earlier, problems with an unknown stochastic control constitute stochastic inverse problems and are different from control problems where the focus is on computing the optimal deterministic control. So, in most cases (as we assume in this work), the mean of the computed stochastic control could be considered as optimal. Depending on the application, the mean may not be the sought optimal control, though. Besides, the uncertainty in the system response might require additional computational challenges.

In the spirit of [115], we remark here that the control variable u could also be decomposed additively into unknown deterministic (to be computed) and known stochastic components:

$$u(\mathbf{x},\omega)=\bar{u}(\mathbf{x})+\widehat{u}(\mathbf{x},\omega),$$

where $\bar{u}:\mathcal{D}\to\mathbb{R}$ is deterministic and is the mean of u and $\widehat{u}:\mathcal{D}\times\Omega\to\mathbb{R}$ is a zero-mean stochastic part. The goal in this case is to compute $\bar{u}$, which constitutes the *signal* sent to a control device. The actual controller response is u, with $\widehat{u}$ modelling the uncertainty in the controller response for a given instruction.

The positive constant β in (4.2) represents the parameter for the penalization of the action of the control u, whereas α penalizes the standard deviation $\mathrm{std}(y)$ of the state y. If β is large, then u must be small, and y may not be very close to $\bar{y}$. In contrast, smaller values of β allow a much larger set of controls u which in turn may allow better approximation of $\bar{y}$ by y. In fact, one expects that $||u|| \to 0$ as $\beta \to \infty$. Similarly, large values of α imply very low variance of the state y, for which there is obviously little or no need for uncertainty quantification in the model. In what follows, we shall focus mainly on *distributed* control problems, in which case the control and the source term of the PDE constraint are one and the same. However, we do believe that our discussion generalizes to *boundary* control problems as well [104, 105, 124, 126].

Our aim in this chapter is to apply our low-rank approach, together with SGFEM, to two prototypical models, namely, optimization problems constrained by (a) stationary diffusion equations, (b) unsteady diffusion equations, and in each of the two cases, both the constraint equations and the objective functional have uncertain inputs. These problems pose increased computational complexity due to enormous memory requirements by the resulting often ill-conditioned linear systems representing the Karush-Kuhn-Tucker (KKT) conditions. In more specific terms, this chapter focuses on the development and analyses of efficient *block-diagonal Schur complement-based* preconditioners for use with low-rank MINRES algorithms to tackle the large-scale optimality linear systems. The materials herein are mainly from the paper [14].

4.2 A stochastic elliptic control problem

Our first SOCP consists in minimizing the cost functional $\mathcal{J}(y(\mathbf{x},\omega), u(\mathbf{x},\omega))$ defined in (4.2) subject, $\mathbb{P}$-almost surely, to the following linear elliptic diffusion equation[2]:

$$\begin{cases} -\nabla \cdot (a(\mathbf{x},\omega)\nabla y(\mathbf{x},\omega)) = u(\mathbf{x},\omega), & \text{in } \mathcal{D} \times \Omega, \\ y(\mathbf{x},\omega) = 0, & \text{on } \partial\mathcal{D} \times \Omega, \end{cases} \tag{4.3}$$

[2] In this dissertation, we do not consider the case of state- or control- or mixed control-state-constrained problems [55, 103, 109]. These problems can be tackled via the use of, for instance, semi-smooth Newton algorithms [54, 61, 67].

where $a : \mathcal{D} \times \Omega \to \mathbb{R}$ is a random coefficient field and the forcing term on the right hand side $u : \mathcal{D} \times \Omega \to \mathbb{R}$ denotes a random control function. Furthermore, we assume that

$$u \in L^2(\mathcal{D}) \otimes L^2(\Omega) \text{ a.e.}, \tag{4.4}$$

and that there exist positive constants $a_{\min}$ and $a_{\max}$ such that (2.4) is satisfied.

Recasting the above SOCP given by (4.2) and (4.3) into a saddle-point formulation, Chen and Quarteroni in [26] prove the existence and uniqueness of its solution. More precisely, the following result holds.

Theorem 4.1. *[26, Theorem 3.5] Let (4.2) and (2.4) be satisfied and let $\alpha = 0$ in (4.2). Then, there exists a unique optimal solution (y, u, f) to the SOCP (4.2) and (4.3) satisfying the stochastic optimality conditions*

$$\begin{aligned} \mathfrak{B}(y, v) &= \ell(u, v), & v &\in H_0^1(\mathcal{D}) \otimes L^2(\Omega), \\ \ell(\beta u - f, w) &= 0, & w &\in L^2(\mathcal{D}) \otimes L^2(\Omega), \\ \mathfrak{B}'(y, r) + \ell(y, r) &= \ell(\bar{y}, r), & r &\in H_0^1(\mathcal{D}) \otimes L^2(\Omega), \end{aligned}$$

where f is the adjoint variable or Lagrangian parameter associated with the optimal solution and ℓ is as given by (2.7). Here, $\mathfrak{B}'$ is the adjoint bilinear form of $\mathfrak{B}$ as defined in (2.6); that is, $\mathfrak{B}'(y, r) = \mathfrak{B}(r, y)$.

We note here that the cost functional considered in [26, 59] does not include $||\text{std}(y)||^2_{L^2(\mathcal{D})}$. But then, their results extend to the more general form of $\mathcal{J}(y, u)$ discussed in this thesis due to the Frechét differentiability of $||\text{std}(y)||^2_{L^2(\mathcal{D})}$; see, for example, [115].

As our major concern in this dissertation is to study efficient solvers resulting from the discretization of our model problems, we proceed next to recall the two common approaches in the literature to solve these optimization problems [125, 126]. The first method is the so-called *optimize-then-discretize* (OTD) approach. Here, one essentially considers the infinite-dimensional problem, writes down the first order conditions and then discretizes the first order conditions. An alternative strategy, namely, the *discretize-then-optimize* (DTO) approach involves discretizing the problem first and then building a discrete Lagrangian functional with the corresponding first order conditions. The commutativity of the DTO and OTD schemes when applied to optimal control problems constrained by PDEs has been a subject of debate in recent times (see [82] for an overview). In what follows, we will adopt the DTO strategy because, for the SOCPs considered in this dissertation, it leads to a symmetric saddle point linear system which fits in nicely with our preconditioning strategy.

Next, we note that an application of SGFEM to the cost functional (4.2) immediately yields

$$\frac{1}{2}(\mathbf{y}-\bar{\mathbf{y}})^T\mathcal{M}(\mathbf{y}-\bar{\mathbf{y}})+\frac{\alpha}{2}\mathbf{y}^T\mathcal{M}_t\mathbf{y}+\frac{\beta}{2}\mathbf{u}^T\mathcal{M}\mathbf{u}, \tag{4.5}$$

where

$$\mathcal{M}:=G_0\otimes M,\ \ \mathcal{M}_t:=H_0\otimes M,\ \ H_0:=\operatorname{diag}\left(0,\left\langle\psi_1^2\right\rangle,\ldots,\left\langle\psi_{P-1}^2\right\rangle\right), \tag{4.6}$$

with $M\in\mathbb{R}^{J\times J}$ the mass matrix and G_0 the diagonal matrix defined, respectively, in (2.35) and (2.28). Similarly, as was done in Chapter 2, a direct application of SGFEM to the state equation (4.3) yields

$$\mathcal{K}\mathbf{y}=\mathcal{M}\mathbf{u}, \tag{4.7}$$

where $\mathcal{K}$ is as given by (2.24).

Our discrete SOCP now is to minimize (4.5) subject to (4.7). The Lagrangian functional $\mathfrak{L}$ of this optimization problem is given by

$$\mathfrak{L}(\mathbf{y},\mathbf{u},\mathbf{f}):=\frac{1}{2}(\mathbf{y}-\bar{\mathbf{y}})^T\mathcal{M}(\mathbf{y}-\bar{\mathbf{y}})+\frac{\alpha}{2}\mathbf{y}^T\mathcal{M}_t\mathbf{y}+\frac{\beta}{2}\mathbf{u}^T\mathcal{M}\mathbf{u}+\mathbf{f}^T(-\mathcal{K}\mathbf{y}+\mathcal{M}\mathbf{u}-\mathbf{d}),$$

where $\mathbf{f}$ denotes the Lagrangian multiplier or adjoint associated with the constraint. Here, $\mathbf{d}:=\operatorname{diag}(G_0)\mathbf{e}_1\otimes\tilde{\mathbf{d}}$, where the vector $\tilde{\mathbf{d}}$ represents, in general, contributions from boundary conditions with respect to the spatial discretization and $\mathbf{e}_1=[1,0,\ldots,0]^T$. Now, applying the first order conditions to the Lagrangian yields, respectively, the adjoint equation, the gradient equation and the state equation:

$$
\begin{aligned}
\mathfrak{L}_{\mathbf{y}}(\mathbf{y},\mathbf{u},\mathbf{f}) = 0 \;\Rightarrow&\quad (\mathcal{M}+\alpha\mathcal{M}_t)\mathbf{y} - \mathcal{K}\mathbf{f} = \mathcal{M}\bar{\mathbf{y}},\\
\mathfrak{L}_{\mathbf{u}}(\mathbf{y},\mathbf{u},\mathbf{f}) = 0 \;\Rightarrow&\quad \beta\mathcal{M}\mathbf{u} + \mathcal{M}\mathbf{f} = 0,\\
\mathfrak{L}_{\mathbf{f}}(\mathbf{y},\mathbf{u},\mathbf{f}) = 0 \;\Rightarrow&\quad -\mathcal{K}\mathbf{y} + \mathcal{M}\mathbf{u} = \mathbf{d},
\end{aligned}
$$

or, alternatively, the following optimality system [115]

$$
\underbrace{\begin{bmatrix} \mathcal{M}_\alpha & 0 & -\mathcal{K}^T \\ 0 & \beta\mathcal{M} & \mathcal{M}^T \\ -\mathcal{K} & \mathcal{M} & 0 \end{bmatrix}}_{:=\mathcal{A}} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \mathbf{f} \end{bmatrix} = \begin{bmatrix} \mathcal{M}\bar{\mathbf{y}} \\ \mathbf{0} \\ \mathbf{d} \end{bmatrix}, \tag{4.8}
$$

where

$$
\begin{aligned}
\mathcal{M}_\alpha &= \mathcal{M} + \alpha\mathcal{M}_t\\
&= (G_0 \otimes M) + \alpha(H_0 \otimes M)\\
&= G_\alpha \otimes M, \qquad (4.9)
\end{aligned}
$$

with $G_\alpha := G_0 + \alpha H_0$, so that

$$
G_\alpha(j,k) = \begin{cases} \left\langle \psi_0^2 \right\rangle, & \text{if} \quad j = k = 0,\\ (1+\alpha)\left\langle \psi_j^2 \right\rangle, & \text{if} \quad j = k = 1, 2, \ldots, P-1,\\ 0, & \text{otherwise.} \end{cases} \tag{4.10}
$$

We note from (2.28), (4.9) and (4.10) that if $\alpha = 0$, then $G_\alpha = G_0$ and, hence, $\mathcal{M}_\alpha = \mathcal{M}$. Moreover, we assume that the parameter N in the KLE of the random input a is chosen such that $\mathcal{K}$ stays symmetric and positive definite [110]. The system (4.8) is usually of huge dimension. As a result, the use of direct solvers for this system is out of the question. In what follows, we consider efficient iterative solvers instead. First, however, we discuss some properties of the optimality system (4.8) on which we shall subsequently rely to build our theory.

4.2.1 Properties of the optimality system

Now, observe that the matrix $\mathcal{A}$ in (4.8) is of *saddle point* form:

$$\mathcal{A} = \begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix}, \tag{4.11}$$

where

$$A = \begin{bmatrix} \mathcal{M}_\alpha & 0 \\ 0 & \beta\mathcal{M} \end{bmatrix}, \quad B = [-\mathcal{K} \;\; \mathcal{M}], \tag{4.12}$$

with A being symmetric and positive definite. Moreover, B has full row rank since both $\mathcal{K}$ and $\mathcal{M}$ are invertible. Note that saddle point systems and their solvers have been extensively discussed in, for instance, [17, 39] and the references therein.

Next, we recall the following well-known result from [17, Section 3.2], which guarantees the existence of a unique solution to (4.8).

Theorem 4.2. *Suppose that the matrices A and B are as defined in (4.12). Assume that A is symmetric and positive definite. Then, the saddle point matrix $\mathcal{A}$ defined in (4.11) is invertible if and only if B^T has full column rank.*

Now, note that by the following congruence transformation

$$\begin{bmatrix} I & 0 \\ BA^{-1} & I \end{bmatrix} \begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} I & A^{-1}B^T \\ 0 & I \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & -S \end{bmatrix}, \tag{4.13}$$

where $S = BA^{-1}B^T$ is called the *Schur complement*, we know that $\mathcal{A}$ is indefinite, with n positive and m negative eigenvalues, where $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{m\times n}$, see e.g. [17, 41]. Generally speaking, unless m is very small (which is seldom the case in practice), the matrix $\mathcal{A}$ is highly indefinite, in the sense that it has many eigenvalues of both signs.

Besides indefiniteness, the saddle point matrix usually has conditioning issues due essentially to the fact that its eigenvalues vary with, for instance, the spatial discretization parameter h. The following result [17, 116] establishes the eigenvalue bounds for $\mathcal{A}$. See also [95] for related spectral results.

Theorem 4.3. *[17, Theorem 3.5] Let the matrices $\mathcal{A}, A$ and B be as in Theorem 4.2. Let θ_1 and θ_n denote the largest and smallest eigenvalues of A, and let s_1 and s_m denote the largest and smallest singular values of B. Denote by $\lambda(\mathcal{A})$ the spectrum of $\mathcal{A}$. Then*

$$\lambda(\mathcal{A}) \subset \mathbb{I}_- \cup \mathbb{I}_+, \tag{4.14}$$

where

$$\mathbb{I}_- = \left[\frac{1}{2}\left(\theta_n - \sqrt{\theta_n^2 + 4s_1^2}\right), \frac{1}{2}\left(\theta_1 - \sqrt{\theta_1^2 + 4s_m^2}\right)\right]$$

and

$$\mathbb{I}_+ = \left[\theta_n, \frac{1}{2}\left(\theta_1 + \sqrt{\theta_1^2 + 4s_m^2}\right)\right].$$

Now, observe from Theorem 4.3 that the spectral condition number $\kappa(\mathcal{A})$ of the matrix $\mathcal{A}$ given by

$$\kappa(\mathcal{A}) := \frac{\max|\lambda(\mathcal{A})|}{\min|\lambda(\mathcal{A})|} \tag{4.15}$$

grows unboundedly as either $\theta_n = \lambda_{\min}(A)$ or $s_m = s_{\min}(B)$ goes to zero (assuming that $\lambda_{\max}(A)$ and $s_{\max}(B)$ are kept constant).

4.2.2 Preconditioning the steady-state KKT system

When the optimality system (4.8) is large and sparse, iterative methods such as Krylov subspace methods are particularly attractive because their storage requirements typically depend only on the number of nonzeros in the coefficient matrix. As we have already noted in Chapter 3, an optimal Krylov subspace solver for the indefinite saddle point system is the MINRES algorithm originally proposed by Paige and Saunders in [100]. However, the twin issues of indefiniteness and poor conditioning exhibited by the saddle point KKT system (as discussed in Section 4.2.1 above) adversely affect the convergence of MINRES. Hence, we need to construct *robust* preconditioners that would accelerate the convergence of MINRES. By robust preconditioners, we mean those with which the iterative solver used is insensitive to the parameters of the discretized model. Algorithm 4.1 shows the vector-based preconditioned MINRES method [39, p. 192] for solving the saddle point

Algorithm 4.1 The preconditioned MINRES method

```
Set $\mathbf{v}^{(0)} = \mathbf{0},\ \mathbf{w}^{(0)} = \mathbf{0},\ \gamma_0 = 0$
Choose $\mathbf{x}^{(0)}$, compute $\mathbf{v}^{(1)} = \mathbf{b} - \mathcal{A}\mathbf{x}^{(0)}$
Solve $\mathcal{P}\mathbf{z}^{(1)} = \mathbf{v}^{(1)}$, set $\gamma_1 = \sqrt{\langle \mathbf{z}^{(1)}, \mathbf{v}^{(1)} \rangle}$
Set $\eta = \gamma_1,\ s_0 = s_1 = 0,\ c_0 = c_1 = 1$
for $j = 1$ until convergence do
    $\mathbf{z}^{(j)} = \mathbf{z}^{(j)}/\gamma_j$
    $\delta_j = \langle \mathcal{A}\mathbf{z}^{(j)}, \mathbf{z}^{(j)} \rangle$
    $\mathbf{v}^{(j+1)} = \mathcal{A}\mathbf{z}^{(j)} - (\delta_j/\gamma_j)\mathbf{v}^{(j)} - (\gamma_j/\gamma_{j-1})\mathbf{v}^{(j-1)}$
    solve $\mathcal{P}\mathbf{z}^{(j+1)} = \mathbf{v}^{(j+1)}$
    $\gamma_{j+1} = \sqrt{\langle \mathbf{z}^{(j+1)}, \mathbf{v}^{(j+1)} \rangle}$
    $\alpha_0 = c_j\delta_j - c_{j-1}s_j\gamma_j$
    $\alpha_1 = \sqrt{\alpha_0^2 + \gamma_{j+1}^2}$
    $\alpha_2 = s_j\delta_j + c_{j-1}c_j\gamma_j$
    $\alpha_3 = s_{j-1}\gamma_j$
    $c_{j+1} = \alpha_0/\alpha_1$
    $s_{j+1} = \gamma_{j+1}/\alpha_1$
    $\mathbf{w}^{(j+1)} = (\mathbf{z}^{(j)} - \alpha_3\mathbf{w}^{(j-1)} - \alpha_2\mathbf{w}^{(j)})/\alpha_1$
    $\mathbf{x}^{(j)} = \mathbf{x}^{(j-1)} + c_{j+1}\eta\mathbf{w}^{(j+1)}$
    $\eta = -s_{j+1}\eta$
    < Test for convergence >
end for
```

system (4.8) $\mathcal{A}\mathbf{x} = \mathbf{b}$, where $\mathbf{x}$ and $\mathbf{b}$ represent, respectively, the solution and the right hand side vectors in the system.

In what follows, we focus mainly on a block-diagonal preconditioning strategy for solving (4.8); that is, we specifically consider preconditioners $\mathcal{P}$ of the form

$$\mathcal{P} := \begin{bmatrix} A & 0 \\ 0 & BA^{-1}B^T \end{bmatrix} = \begin{bmatrix} \mathcal{M}_\alpha & 0 & 0 \\ 0 & \beta\mathcal{M} & 0 \\ 0 & 0 & S \end{bmatrix}, \tag{4.16}$$

where the Schur complement S is given by

$$S = BA^{-1}B^T = \mathcal{K}\mathcal{M}_\alpha^{-1}\mathcal{K} + \frac{1}{\beta}\mathcal{M}, \tag{4.17}$$

since $\mathcal{K}$ and $\mathcal{M}$ are symmetric. Note that many preconditioners for saddle point matrices have been proposed, such as block-triangular [22, 71], constraint [33, 69], augmented Lagrangian [45], and splitting-based [16, 119] preconditioners. For more details, we refer to the survey work of Benzi, Golub and Liesen in [17]. However, there is still motivation for further research in this area. For example, most preconditioners are not robust when β

is small. Recently, though, Pearson et al. [104, 105] have developed a new approximation of the Schur complement and used it to facilitate regularization-robust preconditioning for a broad range of deterministic optimal control problems.

The following proposition, whose proof we include herein to make our treatment more self-contained, describes the spectrum of the preconditioned matrix $\mathcal{Q} := \mathcal{P}^{-1}\mathcal{A}$.

Proposition 4.4. *[89, Proposition 1] Let the matrices $\mathcal{A}$ and $\mathcal{P}$ be as given, respectively, by (4.11) and (4.16). Then, the preconditioned matrix $\mathcal{Q}$ satisfies*

$$\mathcal{Q}(\mathcal{Q} - I)(\mathcal{Q}^2 - \mathcal{Q} - I) = 0. \tag{4.18}$$

Proof. Observe first that

$$\mathcal{Q} = \mathcal{P}^{-1}\mathcal{A} = \begin{bmatrix} I & A^{-1}B^T \\ (BA^{-1}B^T)^{-1}B & 0 \end{bmatrix}, \tag{4.19}$$

so that

$$\left(\mathcal{Q} - \frac{1}{2}I\right)^2 = \begin{bmatrix} \frac{1}{4}I + A^{-1}B^T(BA^{-1}B^T)^{-1}B & 0 \\ 0 & \frac{5}{4}I \end{bmatrix}. \tag{4.20}$$

But then, the matrix $A^{-1}B^T(BA^{-1}B^T)^{-1}B$ is a projection. Thus,

$$\left[\left(\mathcal{Q} - \frac{1}{2}I\right)^2 - \frac{1}{4}I\right]^2 = \left[\left(\mathcal{Q} - \frac{1}{2}I\right)^2 - \frac{1}{4}I\right],$$

which yields (4.18). □

Observe that if $\mathcal{Q}$ is non-singular, then (4.18) in Proposition 4.4 tells us that $\mathcal{Q}$ has only the three non-zero distinct eigenvalues $\left\{1, \frac{1\pm\sqrt{5}}{2}\right\}$ and is therefore diagonalizable. Hence, any Krylov subspace method with optimality property, such as MINRES, will terminate after at most three iterations. We note here, however, that (4.16) is only an ideal preconditioner for our saddle point system (4.8) in the sense that it is not cheap to solve the system with it. In practice, one often has to approximate its three diagonal blocks with positive definite matrices in order to use $\mathcal{P}$ with MINRES. This gives good clustering of the eigenvalues as long as the approximations are spectrally close to the exact operators. An effective approach to approximate blocks $(1,1)$ and $(2,2)$ is, for example, to approximate the mass matrices M in each of the two blocks via Chebyshev semi-iteration [137]. More specifically, for a given system involving a mass matrix $M\mathbf{x} = \mathbf{b}$, the Chebyshev semi-iteration, as given by Algorithm 4.2, is used to speed up a relaxed Jacobi iteration:

$$\mathbf{x}_{k+1} = H\mathbf{x}_k + \mathbf{g},$$

where $H = I - \theta D_0^{-1}M,\ \mathbf{g} = \theta D_0^{-1}\mathbf{b},\ D_0 = \mathrm{diag}(M)$. The optimal relaxation parameter θ must be chosen in such a way that the spectrum of the matrix H is symmetric about the origin. For instance, for a mesh of square $\mathbf{Q}_1$ elements in 2 dimensions, $\lambda(D_0^{-1}M) \subset [1/4, 9/4]$; moreover, if $\theta = 4/5$, then we get $\lambda(H) \subset [-4/5, 4/5]$; see e.g. [136].

Approximating the Schur complement S, that is, block $(3,3)$ poses more difficulty, however. One possibility [112] is to approximate S by dropping the term $\frac{1}{\beta}\mathcal{M}$ to obtain

$$S_0 := \mathcal{K}\mathcal{M}_\alpha^{-1}\mathcal{K}^T. \tag{4.21}$$

The intuitive reasoning behind this is that the first term clearly carries more information in some sense – it contains the discrete Poisson operator whereas the second term consists only of mass matrices which can be thought of as identity or natural inclusion operators in some finite element spaces. Therefore, if β is sufficently large (hence $1/\beta$ sufficiently small) one can hope that this gives a reasonable approximation.

An alternative and more robust approach, which we adopt here and in the rest of

Algorithm 4.2 Chebyshev semi-iterative algorithm for ℓ steps

```
Set $D_0 = \mathrm{diag}(M)$.
Set relaxation parameter $\theta$.
Compute $\mathbf{g} = \theta D_0^{-1}\mathbf{b}$.
Set $H = I - \theta D_0^{-1}M$ (this can be used implicitly).
Set $\mathbf{x}_0 = 0$ and $\mathbf{x}_k = H\mathbf{x}_{k-1} + g$.
Set $c_0 = 2$ and $c_1 = \theta$.
for $k = 1, \dots, l$ do
    $c_{k+1} = \theta c_k - \frac{1}{4}c_{k-1}$.
    $\vartheta_{k+1} = \theta\frac{c_k}{c_{k+1}}$.
    $\mathbf{x}_{k+1} = \vartheta_{k+1}(H\mathbf{x}_k + \mathbf{g} - \mathbf{x}_{k-1}) + \mathbf{x}_{k-1}$.
end for
```

this dissertation, was first introduced in [105] (see also [39, Chapter 5]) in the context of deterministic optimal control problems. In this case, S is approximated by a matrix S_1 of the form

$$S_1 = (\mathcal{K} + \mathcal{M}_u)\, \mathcal{M}_\alpha^{-1} (\mathcal{K} + \mathcal{M}_u)^T, \tag{4.22}$$

where $\mathcal{M}_u$ is determined by 'matching' the terms in the expressions for S_1 and S as given, respectively, by (4.22) and (4.17). More precisely, we ignore the cross terms (that is, $\mathcal{K}\mathcal{M}_\alpha^{-1}\mathcal{M}_u + \mathcal{M}_u\mathcal{M}_\alpha^{-1}\mathcal{K}$) in the expansion of S_1 to get

$$\mathcal{M}_u\mathcal{M}_\alpha^{-1}\mathcal{M}_u = \frac{1}{\beta}\mathcal{M} = \frac{1}{\beta}\mathcal{M}\mathcal{M}^{-1}\mathcal{M}. \tag{4.23}$$

Now, observe from (2.28), (4.9) and (4.10) that we have $\mathcal{M}_\alpha = G_\alpha \otimes M$. Moreover, note that ideally in (4.2), we have $\alpha \geq 0$. So, to derive an approximation to S_1, we consider first of all the case $\alpha = 0$. In this case, it is easy to see that (4.23) holds if we set

$$\mathcal{M}_u = \frac{1}{\sqrt{\beta}}\mathcal{M}, \tag{4.24}$$

since $\mathcal{M}_\alpha = \mathcal{M}$. If $\alpha > 0$, then we apply the following trick. We proceed first to replace in equation (4.10) the $(0,0)$ entry in the diagonal matrix G_α by $(1+\alpha)\left\langle \psi_0^2 \right\rangle$, so that we can then obtain

$$\mathcal{M}_\alpha = G_\alpha \otimes M \approx (1+\alpha)G_0 \otimes M = (1+\alpha)\mathcal{M}.$$

It turns out then that (4.23) holds if and only if

$$\mathcal{M}_u = \sqrt{\frac{1+\alpha}{\beta}}\mathcal{M},$$

with which we recover (4.24) for $\alpha = 0$. Hence, we have

$$S_1 = \underbrace{\left(\mathcal{K} + \sqrt{\frac{1+\alpha}{\beta}}\mathcal{M}\right)}_{:=\mathcal{Z}} \mathcal{M}_\alpha^{-1} \left(\mathcal{K} + \sqrt{\frac{1+\alpha}{\beta}}\mathcal{M}\right)^T. \tag{4.25}$$

We point out here that the expression for $\mathcal{M}_u$ implies that the ignored cross terms are

$\mathcal{O}(\beta^{-1/2})$ instead of $\mathcal{O}(\beta^{-1})$ in (4.21).

4.2.3 Spectral analysis and implementation issues

The effectiveness of the iterative solver used to solve our KKT system depends to a large extent on how well the approximation S_1 represents the exact Schur complement. To measure this, we need to consider the eigenvalues of the preconditioned Schur complement $S_1^{-1}S$. In what follows, we proceed to derive the spectrum $\lambda(S_1^{-1}S)$ of $S_1^{-1}S$ by examining the Rayleigh quotient

$$R(x) := \frac{x^T S x}{x^T S_1 x},$$

for any non-zero vector x of appropriate dimension. We shall rely on the following results on positive definite matrices.

Proposition 4.5. *[90, Theorem 2] Let $X = AB + BA$, where A and B are positive definite, Hermitian square matrices. Then, X is positive definite if*

$$\kappa(B) < \left(\frac{\sqrt{\kappa(A)} + 1}{\sqrt{\kappa(A)} - 1} \right)^2,$$

where $\kappa(\cdot)$ is as defined by (4.15).

Proposition 4.6. *Let A and B be symmetric and positive definite matrices. Then, the matrix $X = ABA$ is also symmetric and positive definite.*

Proof. Let $y = Ax$, $x \neq \mathbf{0}$. Now, note that A is invertible since it is positive definite, which implies that $y \neq \mathbf{0}$ for all $x \neq \mathbf{0}$. Thus, for all $x \neq \mathbf{0}$, we have

$$\begin{aligned} x^T ABAx &= (A^T x)^T B(Ax) \\ &= (Ax)^T B(Ax) \\ &= y^T By > 0, \end{aligned}$$

which shows that X is positive definite. It remains to prove that X is symmetric. Now, observe that

$$X^T = (ABA)^T = A^T B^T A^T = ABA = X.$$

□

We can now prove the main result of this section, which characterizes the spectrum of the preconditioned Schur complement $S_1^{-1}S$.

Theorem 4.7. *Let $\alpha \in [0, +\infty)$. Then, the eigenvalues of $S_1^{-1}S$ satisfy*

$$\lambda(S_1^{-1}S) \subset \left[\frac{1}{2(1+\alpha)}, 1\right) \quad \forall \alpha < \left(\frac{\sqrt{\kappa(\mathcal{K})}+1}{\sqrt{\kappa(\mathcal{K})}-1}\right)^2 - 1, \tag{4.26}$$

where $\mathcal{K} = \sum_{i=0}^{N} G_i \otimes K_i$ is as defined by (2.24).

Proof. Suppose that $\alpha \in [0, +\infty)$. Define the diagonal matrices Υ and $\mathcal{E}_\alpha$ by

$$\Upsilon = \operatorname{diag}(0, I_{P-1}) \text{ and } \mathcal{E}_\alpha = (I_P + \alpha\Upsilon) \otimes I_J, \tag{4.27}$$

where I_n denotes the identity matrix of dimension $n \in \mathbb{N}$. Clearly,

$$I_{JP} \preceq \mathcal{E}_\alpha \preceq (1+\alpha)I_{JP} \text{ and } I_{JP} \succeq \mathcal{E}_\alpha^{-1} \succeq (1+\alpha)^{-1}I_{JP}, \tag{4.28}$$

where, for arbitrary square matrices X and Y, we write $X \succeq Y$ if $X - Y \geq 0$, and vice versa. Moreover, from (1.3), (2.27), (2.28) and (4.9), we obtain

$$\begin{aligned}
\mathcal{M}_\alpha &= G_0 \otimes M + \alpha H_0 \otimes M \\
&= (G_0 + \alpha H_0) \otimes M \\
&= (G_0 I_P + \alpha G_0 \Upsilon) \otimes (M I_J) \\
&= (G_0 \otimes M)(I_P \otimes I_J) + (G_0 \otimes M)(\alpha\Upsilon \otimes I_J) \\
&= (G_0 \otimes M)\left[(I_P \otimes I_J) + (\alpha\Upsilon \otimes I_J)\right] \\
&= \mathcal{M}\left[(I_P + \alpha\Upsilon) \otimes I_J\right] \\
&= \mathcal{M}\mathcal{E}_\alpha = \mathcal{E}_\alpha\mathcal{M},
\end{aligned} \tag{4.29}$$

since both G_0 and $I_P + \alpha\Upsilon$ are diagonal matrices and therefore commute with each other. Now, recall from (4.25) that the approximation S_1 to the Schur complement S is given by

$$S_1 = \mathcal{K}\mathcal{M}_\alpha^{-1}\mathcal{K} + \frac{1+\alpha}{\beta}\mathcal{M}\mathcal{M}_\alpha^{-1}\mathcal{M} + \sqrt{\frac{1+\alpha}{\beta}}\left[\mathcal{K}\mathcal{M}_\alpha^{-1}\mathcal{M} + \mathcal{M}\mathcal{M}_\alpha^{-1}\mathcal{K}\right], \tag{4.30}$$

and that the preconditioned Schur complement $S_1^{-1}S$ is similar to the matrix

$$\mathcal{M}^{1/2}S_1^{-1}S\mathcal{M}^{-1/2} = (\mathcal{M}^{-1/2}S_1\mathcal{M}^{-1/2})^{-1}(\mathcal{M}^{-1/2}S\mathcal{M}^{-1/2}). \tag{4.31}$$

It therefore follows from (4.17), (4.25), (4.29), (4.30) and (4.31) that

$$\begin{aligned} S_1^{-1}S \;&\sim\; \left(\mathcal{C}\mathcal{E}_\alpha^{-1}\mathcal{C} + \frac{1+\alpha}{\beta}\mathcal{E}_\alpha^{-1} + \sqrt{\frac{1+\alpha}{\beta}}\left(\mathcal{C}\mathcal{E}_\alpha^{-1} + \mathcal{E}_\alpha^{-1}\mathcal{C}\right)\right)^{-1}\left(\mathcal{C}\mathcal{E}_\alpha^{-1}\mathcal{C} + \beta^{-1}I_{JP}\right) \\ &= \left(\beta\mathcal{C}\mathcal{E}_\alpha^{-1}\mathcal{C} + (1+\alpha)\mathcal{E}_\alpha^{-1} + \sqrt{\beta(1+\alpha)}\left(\mathcal{C}\mathcal{E}_\alpha^{-1} + \mathcal{E}_\alpha^{-1}\mathcal{C}\right)\right)^{-1}\left(\beta\mathcal{C}\mathcal{E}_\alpha^{-1}\mathcal{C} + I_{JP}\right), \end{aligned}$$

where $\sim$ implies similarity transformation and $\mathcal{C} := \mathcal{M}^{-1/2}\mathcal{K}\mathcal{M}^{-1/2}$. Now, observe that the matrix $\mathcal{C}$ is symmetric and positive definite so that $\lambda(\mathcal{C}) \subset (0, +\infty)$. Consider now the Raleigh quotient

$$R(x) := \frac{x^T\left[\beta\mathcal{C}\mathcal{E}_\alpha^{-1}\mathcal{C} + I_{JP}\right]x}{x^T\left[\beta\mathcal{C}\mathcal{E}_\alpha^{-1}\mathcal{C} + (1+\alpha)\mathcal{E}_\alpha^{-1} + \sqrt{\beta(1+\alpha)}\left(\mathcal{C}\mathcal{E}_\alpha^{-1} + \mathcal{E}_\alpha^{-1}\mathcal{C}\right)\right]x}.$$

Now, it is easy to see that $x^T\mathcal{E}_\alpha^{-1}x > 0$. Next, observe from (4.29) that

$$\begin{aligned} \mathcal{C}\mathcal{E}_\alpha^{-1} + \mathcal{E}_\alpha^{-1}\mathcal{C} &= \mathcal{M}^{-1/2}\mathcal{K}\mathcal{M}^{-1/2}\mathcal{E}_\alpha^{-1} + \mathcal{E}_\alpha^{-1}\mathcal{M}^{-1/2}\mathcal{K}\mathcal{M}^{-1/2} \\ &= \mathcal{M}^{-1/2}\mathcal{K}\mathcal{E}_\alpha^{-1}\mathcal{M}^{-1/2} + \mathcal{M}^{-1/2}\mathcal{E}_\alpha^{-1}\mathcal{K}\mathcal{M}^{-1/2} \\ &= \mathcal{M}^{-1/2}\left[\mathcal{K}\mathcal{E}_\alpha^{-1} + \mathcal{E}_\alpha^{-1}\mathcal{K}\right]\mathcal{M}^{-1/2}. \end{aligned} \tag{4.32}$$

But then, using (4.15) we see that $\kappa(\mathcal{E}_\alpha^{-1}) = 1 + \alpha$, so that Proposition 4.5 yields

$$x^T(\mathcal{K}\mathcal{E}_\alpha^{-1} + \mathcal{E}_\alpha^{-1}\mathcal{K})x > 0 \text{ for } \alpha + 1 < \left(\frac{\sqrt{\kappa(\mathcal{K})} + 1}{\sqrt{\kappa(\mathcal{K})} - 1}\right)^2. \tag{4.33}$$

Therefore, keeping in mind that both $\mathcal{M}^{-1/2}$ and $\mathcal{K}\mathcal{E}_\alpha^{-1} + \mathcal{E}_\alpha^{-1}\mathcal{K}$ are symmetric and positive definite, we see from (4.32) and Proposition 4.6 that the matrix $x^T(\mathcal{C}\mathcal{E}_\alpha^{-1} + \mathcal{E}_\alpha^{-1}\mathcal{C})x > 0$. Similarly, Proposition 4.6 guarantees that $x^T\mathcal{C}\mathcal{E}_\alpha^{-1}\mathcal{C}x > 0$ holds. Hence, the denominator of $R(x)$ is strictly positive. Thus, using (4.28), we obtain

$$R(x) \leq \frac{x^T\left[\beta\mathcal{C}\mathcal{E}_\alpha^{-1}\mathcal{C} + (1+\alpha)\mathcal{E}_\alpha^{-1}\right]x}{x^T\left[\beta\mathcal{C}\mathcal{E}_\alpha^{-1}\mathcal{C} + (1+\alpha)\mathcal{E}_\alpha^{-1} + \sqrt{\beta(1+\alpha)}\left(\mathcal{C}\mathcal{E}_\alpha^{-1} + \mathcal{E}_\alpha^{-1}\mathcal{C}\right)\right]x} < 1,$$

from which we deduce that $\lambda_{\max} := \max R(x) < 1$.

Now, recall that, for any two vectors z_1, z_2 of appropriate dimensions, the Cauchy-Schwarz Inequality implies $\langle z_1^T z_2 \rangle^2 \leq (z_1^T z_1)(z_2^T z_2)$. Thus, setting $z_1^T = x^T \mathcal{C} \mathcal{E}_\alpha^{-1/2}$ and $z_2 = \mathcal{E}_\alpha^{-1/2} x$, we obtain

$$\left(x^T \mathcal{C} \mathcal{E}_\alpha^{-1} x\right)^2 \leq \left(x^T \mathcal{C} \mathcal{E}_\alpha^{-1} \mathcal{C} x\right) \left(x^T \mathcal{E}_\alpha^{-1} x\right). \tag{4.34}$$

Similarly, if we set $z_1^T = x^T \mathcal{E}_\alpha^{-1/2}$ and $z_2 = \mathcal{E}_\alpha^{-1/2} \mathcal{C} x$, then we get

$$\left(x^T \mathcal{E}_\alpha^{-1} \mathcal{C} x\right)^2 \leq \left(x^T \mathcal{C} \mathcal{E}_\alpha^{-1} \mathcal{C} x\right) \left(x^T \mathcal{E}_\alpha^{-1} x\right), \tag{4.35}$$

so that

$$x^T (\mathcal{C} \mathcal{E}_\alpha^{-1} + \mathcal{E}_\alpha^{-1} \mathcal{C}) x \leq 2 \sqrt{\left(x^T \mathcal{C} \mathcal{E}_\alpha^{-1} \mathcal{C} x\right) \left(x^T \mathcal{E}_\alpha^{-1} x\right)}. \tag{4.36}$$

Moreover, note that since $(a+b)^2 \leq 2(a^2+b^2) \ \ \forall a, b \in \mathbb{R}$, then

$$\frac{1}{(a+b)^2} \geq \frac{1}{2(a^2+b^2)}, \ \ \forall a, b \in \mathbb{R}. \tag{4.37}$$

Hence, using (4.28), (4.36) and (4.37), one obtains

$$\begin{aligned}
R(x) &= \frac{x^T \left[\beta \mathcal{C} \mathcal{E}_\alpha^{-1} \mathcal{C} + I_{JP}\right] x}{x^T \left[\beta \mathcal{C} \mathcal{E}_\alpha^{-1} \mathcal{C} + (1+\alpha) \mathcal{E}_\alpha^{-1} + \sqrt{\beta(1+\alpha)} \left(\mathcal{C} \mathcal{E}_\alpha^{-1} + \mathcal{E}_\alpha^{-1} \mathcal{C}\right)\right] x} \\
&\geq \frac{x^T \left[\beta \mathcal{C} \mathcal{E}_\alpha^{-1} \mathcal{C} + I_{JP}\right] x}{\beta x^T \mathcal{C} \mathcal{E}_\alpha^{-1} \mathcal{C} x + (1+\alpha) x^T \mathcal{E}_\alpha^{-1} x + 2 \sqrt{\beta(1+\alpha) \left(x^T \mathcal{C} \mathcal{E}_\alpha^{-1} \mathcal{C} x\right) \left(x^T \mathcal{E}_\alpha^{-1} x\right)}} \\
&= \frac{x^T \beta \mathcal{C} \mathcal{E}_\alpha^{-1} \mathcal{C} x + x^T I_{JP} x}{\left[\beta^{1/2} (x^T \mathcal{C} \mathcal{E}_\alpha^{-1} \mathcal{C} x)^{1/2} + (1+\alpha)^{1/2} (x^T \mathcal{E}_\alpha^{-1} x)^{1/2}\right]^2} \\
&\geq \frac{x^T \beta \mathcal{C} \mathcal{E}_\alpha^{-1} \mathcal{C} x + x^T I_{JP} x}{2 \left[\beta x^T \mathcal{C} \mathcal{E}_\alpha^{-1} \mathcal{C} x + (1+\alpha) x^T \mathcal{E}_\alpha^{-1} x\right]} \\
&\geq \frac{x^T \beta \mathcal{C} \mathcal{E}_\alpha^{-1} \mathcal{C} x + x^T \mathcal{E}_\alpha^{-1} x}{2 \left[\beta x^T \mathcal{C} \mathcal{E}_\alpha^{-1} \mathcal{C} x + (1+\alpha) x^T \mathcal{E}_\alpha^{-1} x\right]} \\
&\geq \frac{x^T \mathcal{E}_\alpha^{-1} x}{2(1+\alpha) x^T \mathcal{E}_\alpha^{-1} x} = \frac{1}{2(1+\alpha)},
\end{aligned} \tag{4.38}$$

so that $\lambda_{\min} := \min R(x) \geq \frac{1}{2(1+\alpha)}$, thereby concluding the proof of the theorem. □

Note that, in the context of a deterministic optimal control problem, Pearson and Wathen in [105, Theorem 4] have independently obtained, specifically for $\alpha = 0$, a similar result to that of Theorem 4.7; see also [39, Lemma 5.2]. We, however, point out herein that, in addition to the generalization of the said result, ours yields a sharper bound than the one that these authors obtained. Moreover, with the exception of the parameter α, the result of Theorem 4.7 is independent of the discretization parameters in the system.

The following result is an immediate consequence of Theorem 4.7.

Theorem 4.8. *Let $\mathcal{A}$ be the KKT matrix given by (4.11) and define $\mathcal{P}_s$ by*

$$\mathcal{P}_s := \begin{bmatrix} A & 0 \\ 0 & S_1 \end{bmatrix},$$

where A and S_1 are given, respectively, by (4.12) and (4.25). Moreover, assume that $\alpha < \left(\frac{\sqrt{\kappa(\mathcal{K})}+1}{\sqrt{\kappa(\mathcal{K})}-1}\right)^2 - 1$, where $\mathcal{K}$ is as defined in Theorem 4.7. Then, the eigenvalues of the matrix $\mathcal{P}_s^{-1}\mathcal{A}$ satisfy

$$\lambda(\mathcal{P}_s^{-1}\mathcal{A}) = \{1\} \cup \mathcal{I}^- \cup \mathcal{I}^+, \tag{4.39}$$

where

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

and

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

Proof. First, we note that $\mathcal{P}_s^{-1}\mathcal{A}$ possesses the same eigenvalues as the symmetric matrix given by

$$\mathcal{P}_s^{-1/2}\mathcal{A}\mathcal{P}_s^{-1/2} = \begin{bmatrix} I & A^{-1/2}B^T S_1^{-1/2} \\ S_1^{-1/2}BA^{-1/2} & 0 \end{bmatrix}.$$

Now, using [41, Lemma 2.1], we know that the eigenvalues of $\mathcal{P}_s^{-1/2}\mathcal{A}\mathcal{P}_s^{-1/2}$ are either 1 or have the form $\frac{1}{2}\left(1 \pm \sqrt{1+4s^2}\right)$, where s is a singular value of $X := S_1^{-1/2}BA^{-1/2}$; in other words, s^2 is an eigenvalue of XX^T. Since $S_1^{-1}S$ is similar to XX^T, the result (4.39) follows immediately from Theorem 4.7. □

It turns out that the equality of the lengths of the intervals $\mathcal{I}^-$ and $\mathcal{I}^+$ in Theorem 4.8 is of paramount importance in establishing the convergence of MINRES [39, 107] as shown in the following.

Proposition 4.9. *[39, Theorem 4.14] Let $\mathcal{A}\mathbf{x} = \mathbf{b}$ be a saddle point linear system with $\mathcal{A}$ as given by (4.11). Let the eigenvalues of the preconditioned $\mathcal{P}^{-1}\mathcal{A}$ be contained in the intervals $[-\mu_1, -\mu_0] \cup [\nu_0, \nu_1]$ with $\mu_1 - \mu_0 = \nu_1 - \nu_0$. Then, after $2k$ steps of MINRES the residual $\mathbf{r}^{(2k)} = \mathbf{b} - \mathcal{A}\mathbf{x}^{(2k)}$ satisfies the bound*

$$||\mathbf{r}^{(2k)}||_{\mathcal{P}^{-1}} \leq 2\left(\frac{\sqrt{\mu_1\nu_1} - \sqrt{\mu_0\nu_0}}{\sqrt{\mu_1\nu_1} + \sqrt{\mu_0\nu_0}}\right)^k ||\mathbf{r}^{(0)}||_{\mathcal{P}^{-1}}, \;\; k \in \mathbb{N}. \tag{4.40}$$

Remark 4.10. *We note here that the bound (4.40) can be pessimistic, particularly if the negative and positive eigenvalues of $\mathcal{P}^{-1}\mathcal{A}$ lie in intervals of significantly different lengths [107]. However, it certainly shows that knowledge of the extreme eigenvalues of $\mathcal{P}^{-1}\mathcal{A}$ can provide useful information about the speed of convergence of MINRES. From the bound (4.40), we additionally discern that a sufficient condition for fast convergence is that μ_1/μ_0 and ν_1/ν_0 are small, since this will ensure that the eigenvalues are clustered away from the origin. The latter point is a crucial one since small eigenvalues can hinder the convergence of MINRES.*

Remark 4.11. *Due to the minimization property of MINRES [39, p. 211], we have*

$$||\mathbf{r}^{(2k+1)}||_{\mathcal{P}^{-1}} \leq ||\mathbf{r}^{(2k)}||_{\mathcal{P}^{-1}}, \;\; k \in \mathbb{N}. \tag{4.41}$$

However, the possibility that no reduction in the residual norm occurs at every other step is not precluded. Indeed, the so-called 'stair-casing' where $||\mathbf{r}^{(2k+1)}||_{\mathcal{P}^{-1}} = ||\mathbf{r}^{(2k)}||_{\mathcal{P}^{-1}}$ is often noticed in computations.

As a consequence of Theorem 4.8, Proposition 4.9 and (4.41), we immediately obtain the following result which confirms that the convergence of MINRES is independent of all the discretization and regularization parameters in the considered model save α.

Corollary 4.12. *Let the eigenvalues of $\mathcal{P}_s^{-1}\mathcal{A}$ be as given in Theorem 4.8 and let $\mathcal{A}\mathbf{x} = \mathbf{b}$ with $\mathcal{A}$ as given by (4.11). Assume, furthermore, that $\alpha < \left(\frac{\sqrt{\kappa(\mathcal{K})}+1}{\sqrt{\kappa(\mathcal{K})}-1}\right)^2 - 1$, where $\mathcal{K}$ is* *as defined in Theorem 4.7. Then, after k steps of MINRES, the residual* $\mathbf{r}^{(k)} = \mathbf{b} - \mathcal{A}\mathbf{x}^{(k)}$ *satisfies the bound*

$$||\mathbf{r}^{(k)}||_{\mathcal{P}_s^{-1}} \leq 2\left(\frac{1-1/\delta}{1+1/\delta}\right)^k ||\mathbf{r}^{(0)}||_{\mathcal{P}_s^{-1}}, \;\; k \in \mathbb{N}, \tag{4.42}$$

where $\delta = \sqrt{2(1+\alpha)}$.

The robustness of S_1 notwithstanding, we cannot implement it as it is, as this is equivalent to solving the forward problem twice per iteration due to the presence of $\mathcal{Z} := \mathcal{K} + \sqrt{\frac{1+\alpha}{\beta}}\mathcal{M}$ and its transpose in (4.25). Hence, we need to derive an appropriate approximation for $\mathcal{Z}$. To this end, observe first, from (2.27), that since

$$\begin{aligned} \mathcal{Z} &= \mathcal{K} + \sqrt{\frac{1+\alpha}{\beta}}\mathcal{M} \\ &= \left(\sum_{i=0}^{N} G_i \otimes K_i\right) + \sqrt{\frac{1+\alpha}{\beta}}(G_0 \otimes M) \\ &= \sum_{i=0}^{N} G_i \otimes \widehat{K}_i, \end{aligned} \tag{4.43}$$

with $\widehat{K}_0 := K_0 + \sqrt{\frac{1+\alpha}{\beta}}M, \;\; \widehat{K}_i = K_i, \; i = 1, \ldots, N$, one could approximate $\mathcal{Z}$ using, for example, the block-diagonal mean-based preconditioner which was considered in Chapter 3 (see also [13, 110]):

$$\mathcal{Z}_0 := G_0 \otimes \widehat{K}_0. \tag{4.44}$$

For a practical algorithm, S_1 could then be implemented using multigrid techniques for $\widehat{K}_0$ in $\mathcal{Z}_0$. As we noted in Chapter 3, (4.44) is best suited for systems for which the variance of the random input a is small relative to its mean. That is, its performance, unfortunately, deteriorates with increasing σ_a. It would therefore not be quite useful in real-world applications in which the variability in the model is reasonably high. Therefore, as an alternative to mitigate this inherent deficiency, in our numerical experiments we also consider approximate solves with $\mathcal{Z}$ (i.e., $\mathcal{Z}\mathbf{x} = \mathbf{b}$) via a preconditioned Richardson iteration as given by Algorithm 4.3. In our experience, the latter approach proved more efficient (with just a few iterations) than the former, especially as we increased the variance

Algorithm 4.3 Preconditioned Richardson iteration for $\mathcal{Z}\mathbf{x} = \mathbf{b}$.

```
Select $\mathbf{x}_0$
Set $\widehat{\mathcal{P}} := \mathcal{Z}_0$
for $k = 0, 1, 2, \ldots$ do
    $\mathbf{r}_k = \mathbf{b} - \mathcal{Z}\mathbf{x}_k$
    $\mathbf{x}_{k+1} = \mathbf{x}_k + \widehat{\mathcal{P}}^{-1}\mathbf{r}_k$
end for
```

of the random field a.

In a nutshell, we outline below the dominant operations – which we refer to as `AprecOut` – in the application of our proposed block-diagonal preconditioner $\mathcal{P}$ in (4.16).

- **(1,1)** block: 1 Chebyshev semi-iteration for the mass matrix M.
- **(2,2)** block: 1 Chebyshev semi-iteration for the mass matrix M.
- **(3,3)** block: 2 multigrid (or preconditioned Richardson iteration) operations: 1 for $\mathcal{Z}_0$ (resp. $\mathcal{Z}$) and 1 for its transpose.
- **Total**: 2 Chebyshev semi-iterations and 2 multigrid (or preconditioned Richardson iteration) operations.

Having been equipped with a suitable preconditioner, we proceed to the next section to discuss our Krylov subspace solver.

4.2.4 Low-rank solution to the steady-state problem

As we have already pointed out in Section 4.2.2, the MINRES algorithm is an optimal solver for the system (4.8). Hence, we will use it, together with (4.16), to solve (4.8). In particular, our approach is based on the low-rank version of MINRES [14, 124]. For a detailed discussion of the existence of low-rank approximation to the KKT system in the deterministic setting, we refer the interested reader to [124]. The existence result in [124] easily generalizes to the stochastic Galerkin KKT system considered herein.

In this section, we present the low-rank MINRES solver. Now, observe first that using the identity (1.2), the linear system (4.8) can be rewritten as $\mathcal{A}\mathcal{X} = \mathcal{R}$, where

$$\mathcal{A} = \begin{bmatrix} G_\alpha \otimes M & 0 & -\sum\limits_{i=0}^{N} G_i \otimes K_i \\ 0 & \beta(G_0 \otimes M) & G_0 \otimes M \\ -\sum\limits_{i=0}^{N} G_i \otimes K_i & G_0 \otimes M & 0 \end{bmatrix},$$

$$\mathcal{X} = \begin{bmatrix} \mathrm{vec}(Y) \\ \mathrm{vec}(U) \\ \mathrm{vec}(F) \end{bmatrix}, \quad \mathcal{R} = \begin{bmatrix} \mathrm{vec}(R_1) \\ 0 \\ \mathrm{vec}(R_3) \end{bmatrix},$$

and

$$Y = [y_0, \ldots, y_{P-1}], \;\; U = [u_0, \ldots, u_{P-1}], \;\; F = [f_0, \ldots, f_{P-1}],$$

$$R_1 = \mathrm{vec}^{-1}((G_0 \otimes M)\bar{\mathbf{y}}), \;\; R_3 = \mathrm{vec}^{-1}(\mathbf{d}).$$

Hence, (1.2) implies that

$$\mathcal{A}\mathcal{X} = \mathrm{vec}\left(\begin{bmatrix} MYG_\alpha^T - \sum\limits_{i=0}^{N} K_i F G_i^T \\ \beta MUG_0^T + MFG_0^T \\ -\sum\limits_{i=0}^{N} K_i Y G_i^T + MUG_0^T \end{bmatrix}\right) = \mathrm{vec}\left(\begin{bmatrix} R_1 \\ 0 \\ R_3 \end{bmatrix}\right). \tag{4.45}$$

As noted before, the low-rank approach is essentially based on the assumption that both the solution matrix $\mathcal{X}$ and the right hand side matrix $\mathcal{R}$ admit low-rank representations; that is,

$$\begin{cases} Y = W_Y V_Y^T, \;\text{ with }\; W_Y \in \mathbb{R}^{J \times r_1}, \;\; V_Y \in \mathbb{R}^{P \times r_1} \\ U = W_U V_U^T, \;\text{ with }\; W_U \in \mathbb{R}^{J \times r_2}, \;\; V_U \in \mathbb{R}^{P \times r_2} \\ F = W_F V_F^T, \;\text{ with }\; W_F \in \mathbb{R}^{J \times r_3}, \;\; V_F \in \mathbb{R}^{P \times r_3}, \end{cases} \tag{4.46}$$

where, in general, $r_{1,2,3} \ll J, P$. Substituting (4.46) in (4.45) and ignoring the vec operator, we then obtain

$$\begin{bmatrix} MW_YV_Y^TG_\alpha^T - \sum\limits_{i=0}^{N} K_iW_FV_F^TG_i^T \\ \beta MW_UV_U^TG_0^T + MW_FV_F^TG_0^T \\ -\sum\limits_{i=0}^{N} K_iW_YV_Y^TG_i^T + MW_UV_U^TG_0^T \end{bmatrix} = \begin{bmatrix} R_{11}R_{12}^T \\ 0 \\ R_{31}R_{32}^T \end{bmatrix}, \tag{4.47}$$

where $R_{11}R_{12}^T$ and $R_{31}R_{32}^T$ are the low-rank representations of the R_1 and R_3, respectively.

The attractiveness of this approach lies therefore in the fact that one can rewrite the three block rows in the left hand side in (4.47), respectively, as

$$\begin{cases} \text{(first block row)} \begin{bmatrix} MW_Y & -K_0W_F & \cdots & -K_NW_F \end{bmatrix} \begin{bmatrix} V_Y^TG_\alpha^T \\ V_F^TG_0^T \\ \vdots \\ V_F^TG_N^T \end{bmatrix}, \\ \text{(second block row)} \begin{bmatrix} \beta MW_U & MW_F \end{bmatrix} \begin{bmatrix} V_U^TG_0^T \\ V_F^TG_0^T \end{bmatrix}, \\ \text{(third block row)} \begin{bmatrix} -K_0W_Y & \cdots & -K_NW_Y & MW_U \end{bmatrix} \begin{bmatrix} V_Y^TG_0^T \\ \vdots \\ V_Y^TG_N^T \\ V_U^TG_0^T \end{bmatrix}, \end{cases} \tag{4.48}$$

so that the low-rank nature of the factors guarantees fewer multiplications with the submatrices while maintaining smaller storage requirements. More precisely, keeping in mind that

$$\mathbf{x} = \text{vec}\left(\begin{bmatrix} X_{11}X_{12}^T \\ X_{21}X_{22}^T \\ X_{31}X_{32}^T \end{bmatrix} \right) \tag{4.49}$$

Algorithm 4.4 Matrix-vector multiplication in low-rank MINRES: `Amult`

```
Input: $W_{11}, W_{12}, W_{21}, W_{22}, W_{31}, W_{32}$
Output: $X_{11}, X_{12}, X_{21}, X_{22}, X_{31}, X_{32}$
$X_{11} = [MW_{11} \quad -K_0W_{31} \quad \cdots \quad -K_NW_{31}]$
$X_{12} = [G_\alpha W_{12} \quad G_0W_{32} \quad \cdots \quad G_NW_{32}]$
$X_{21} = [\beta MW_{21} \quad MW_{31}]$
$X_{22} = [G_0W_{22} \quad G_0W_{32}]$
$X_{31} = [-K_0W_{11} \quad \cdots \quad -K_NW_{11} \quad MW_{21}]$
$X_{32} = [\ G_0W_{12} \quad \cdots \quad G_NW_{12} \quad G_0W_{22}]$
```

corresponds to the associated vector $\mathbf{x}$ from a vector-based version of MINRES, matrix-vector multiplication in our low-rank MINRES is given by Algorithm 4.4.

Note that the truncation operation is again necessary because the new computed factors could have increased ranks compared to the original factors in (4.48). Hence, a truncation of the factors $X_{ij},\ i,j = 1,2,3$, in Algorithm 4.4 is used to construct new factors; for instance,

$$[\tilde{X}_{11}, \tilde{X}_{12}] := \mathcal{T}_\varepsilon\left(X_{11}, X_{12}\right) = \mathcal{T}_\varepsilon\left([MW_{11} \quad -K_0W_{31} \quad \cdots \quad -K_NW_{31}] \begin{bmatrix} W_{12}^TG_\alpha^T \\ W_{32}^TG_0^T \\ \vdots \\ W_{32}^TG_N^T \end{bmatrix} \right),$$

where $\mathcal{T}_\varepsilon$ is the truncation operator with a prescribed tolerance ε as described in Section 3.2.4.

The inner products within the iterative low-rank solver are computed efficiently via the procedure also discussed in Section 3.2.4. In particular, using (3.24), we obtain

$$\begin{aligned} \langle \mathbf{x}, \mathbf{y} \rangle &= \operatorname{trace}\left((X_{11}X_{12}^T)^TY_{11}Y_{12}^T\right) + \operatorname{trace}\left((X_{21}X_{22}^T)^TY_{21}Y_{22}^T\right) \\ &\quad + \operatorname{trace}\left((X_{31}X_{32}^T)^TY_{31}Y_{32}^T\right) \\ &= \operatorname{trace}\left(Y_{12}^TX_{12}X_{11}^TY_{11}\right) + \operatorname{trace}\left(Y_{22}^TX_{22}X_{21}^TY_{21}\right) \\ &\quad + \operatorname{trace}\left(Y_{32}^TX_{32}X_{31}^TY_{31}\right), \end{aligned} \tag{4.50}$$

where the vector $\mathbf{y}$ is defined analogously as in (4.49).

Algorithm 4.5 shows the low-rank preconditioned minimum residual method. We note

Algorithm 4.5 Low-rank preconditioned MINRES

```
Zero-initialization of $V_{11}^{(0)}, \ldots, W_{11}^{(0)}, \ldots$, and $V_{11}^{(1)}, \ldots, W_{11}^{(1)}, \ldots$.
Choose $X_{11}^{(0)}, X_{12}^{(0)}, X_{21}^{(0)}, X_{22}^{(0)}, X_{31}^{(0)}, X_{32}^{(0)}$
Set $V_{11}, V_{12}, \ldots$ to normalized residual
Set $\left[Z_{11}^{(1)}, Z_{12}^{(1)}, Z_{21}^{(1)}, Z_{22}^{(1)}, Z_{31}^{(1)}, Z_{32}^{(1)}\right] = \texttt{Aprec}\left(V_{11}^{(1)}, V_{12}^{(1)}, V_{21}^{(1)}, V_{22}^{(1)}, V_{31}^{(1)}, V_{32}^{(1)}\right)$
Set $\eta = \gamma_1,\ s_0 = s_1 = 0,\ c_0 = c_1 = 1;\ \ \gamma_1 = \sqrt{\texttt{tracepoduct}\left(Z_{11}^{(1)}, \ldots, V_{11}^{(1)}, \ldots\right)}$
while residual norm > tolerance do
    $Z_{11}^{(j)} = Z_{11}^{(j)}/\gamma_j,\ Z_{21}^{(j)} = Z_{21}^{(j)}/\gamma_j,\ Z_{31}^{(j)} = Z_{31}^{(j)}/\gamma_j,$
    $[H_{11}, H_{12}, H_{21}, H_{22}, H_{31}, H_{32}] = \texttt{Amult}\left(Z_{11}^{(j)}, Z_{12}^{(j)}, Z_{21}^{(j)}, Z_{22}^{(j)}, Z_{31}^{(j)}, Z_{32}^{(j)}\right)$
    $\delta_j = \texttt{traceproduct}\left(H_{11}, H_{12}, H_{21}, H_{22}, H_{31}, H_{32}, Z_{11}^{(j)}, Z_{12}^{(j)}, Z_{21}^{(j)}, Z_{22}^{(j)}, Z_{31}^{(j)}, Z_{32}^{(j)}\right)$
    $\left[V_{11}^{(j+1)}, V_{12}^{(j+1)}\right] = \mathcal{T}_\varepsilon\left(\left[H_{11}\ \ -\frac{\delta_j}{\gamma_j}V_{11}^{(j)}\ \ -\frac{\gamma_j}{\gamma_{j-1}}V_{11}^{(j-1)}\right], \left[H_{12}\ \ V_{12}^{(j)}\ \ V_{12}^{(j-1)}\right]^T\right)$
    $\left[V_{21}^{(j+1)}, V_{22}^{(j+1)}\right] = \mathcal{T}_\varepsilon\left(\left[H_{21}\ \ -\frac{\delta_j}{\gamma_j}V_{21}^{(j)}\ \ -\frac{\gamma_j}{\gamma_{j-1}}V_{21}^{(j-1)}\right], \left[H_{22}\ \ V_{22}^{(j)}\ \ V_{22}^{(j-1)}\right]^T\right)$
    $\left[V_{31}^{(j+1)}, V_{32}^{(j+1)}\right] = \mathcal{T}_\varepsilon\left(\left[H_{31}\ \ -\frac{\delta_j}{\gamma_j}V_{31}^{(j)}\ \ -\frac{\gamma_j}{\gamma_{j-1}}V_{31}^{(j-1)}\right], \left[H_{32}\ \ V_{32}^{(j)}\ \ V_{32}^{(j-1)}\right]^T\right)$
    $\left[Z_{11}^{(j+1)}, Z_{12}^{(j+1)}, Z_{21}^{(j+1)}, Z_{22}^{(j+1)}, Z_{31}^{(j+1)}, Z_{32}^{(j+1)}\right] =$
    $\texttt{Aprec}\left(V_{11}^{(j+1)}, V_{12}^{(j+1)}, V_{21}^{(j+1)}, V_{22}^{(j+1)}, V_{31}^{(j+1)}, V_{32}^{(j+1)}\right)$
    $\gamma_{j+1} = \sqrt{\texttt{tracepoduct}\left(Z_{11}^{(j+1)}, \ldots, V_{11}^{(j+1)}, \ldots\right)}$
    $\alpha_0 = c_j\delta_j - c_{j-1}s_j\gamma_j$
    $\alpha_1 = \sqrt{\alpha_0^2 + \gamma_{j+1}^2}$
    $\alpha_2 = s_j\delta_j + c_{j-1}c_j\gamma_j$
    $\alpha_3 = s_{j-1}\gamma_j$
    $c_{j+1} = \alpha_0/\alpha_1$
    $s_{j+1} = \gamma_{j+1}/\alpha_1$
    $\left[W_{11}^{(j+1)}, W_{12}^{(j+1)}\right] = \mathcal{T}_\varepsilon\left(\left[Z_{11}^{(j)}\ \ -\alpha_3W_{11}^{(j-1)}\ \ -\alpha_2W_{11}^{(j)}\right], \left[Z_{12}^{(j)}\ \ W_{12}^{(j-1)}\ \ W_{12}^{(j)}\right]^T\right)$
    $\left[W_{21}^{(j+1)}, W_{22}^{(j+1)}\right] = \mathcal{T}_\varepsilon\left(\left[Z_{21}^{(j)}\ \ -\alpha_3W_{21}^{(j-1)}\ \ -\alpha_2W_{21}^{(j)}\right], \left[Z_{22}^{(j)}\ \ W_{22}^{(j-1)}\ \ W_{22}^{(j)}\right]^T\right)$
    $\left[W_{31}^{(j+1)}, W_{32}^{(j+1)}\right] = \mathcal{T}_\varepsilon\left(\left[Z_{31}^{(j)}\ \ -\alpha_3W_{31}^{(j-1)}\ \ -\alpha_2W_{31}^{(j)}\right], \left[Z_{32}^{(j)}\ \ W_{32}^{(j-1)}\ \ W_{32}^{(j)}\right]^T\right)$
    $\eta = -s_{j+1}\eta$
    if Convergence criterion fulfilled then
        Compute approximate solution
        stop
    end if
end while
```

here that in the algorithm, the square brackets [ ] without comma(s) inside them should be understood as concatenation of matrices. Besides, the functions `traceproduct` and `Amult`, given respectively by (4.50) and Algorithm 4.4, implement the inner products and matrix-vector multiplication. In principle, the preconditioner $\mathcal{P}$ should have a structure

Algorithm 4.6 Preconditioner implementation in low-rank MINRES: `Aprec`

```
Input: $W_{11}, W_{12}, W_{21}, W_{22}, W_{31}, W_{32}$
Output: $X_{11}, X_{12}, X_{21}, X_{22}, X_{31}, X_{32}$
Solve: $MX_{11} = W_{11}$      via Chebyshev semi-iteration
Solve: $G_\alpha X_{12} = W_{12}$
Solve: $\beta M X_{21} = W_{21}$      via Chebyshev semi-iteration
Solve: $G_0 X_{22} = W_{22}$
Compute $X_{31}$ and $X_{32}$ as the low-rank solution of $S_1$ with the right hand side defined
  by $W_{31}$ and $W_{32}$, using $(\mathbf{3}, \mathbf{3})$ in the procedure AprecOut at the end of Section 4.2.3.
```

that allows $\mathcal{P}^{-1}$ to benefit from low-rank format as well. This is the case with the preconditioners discussed so far, and we implement them via the function `Aprec` given by Algorithm 4.6, following also the procedure `AprecOut` outlined at the end of Section 4.2.3.

In Section 4.4, we use numerical experiments to illustrate the performance of low-rank MINRES, together with the preconditioners discussed in Section 4.2.2. Next, we proceed to Section 4.3 to present an unsteady analogue of the model problem considered so far.

4.3 A stochastic parabolic control problem

In an attempt to extend our discussion on the above model problem to a time-dependent case (see e.g. [20]), we now consider the following parabolic SOCP: Minimize

$$\mathcal{J}(t, y, u) := \frac{1}{2}||y - \bar{y}||^2_{L^2(0,T;\mathcal{D})\otimes L^2(\Omega)} + \frac{\alpha}{2}||\mathrm{std}(y)||^2_{L^2(0,T;\mathcal{D})} + \frac{\beta}{2}||u||^2_{L^2(0,T;\mathcal{D})\otimes L^2(\Omega)} \tag{4.51}$$

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

$$\begin{cases} \dfrac{\partial y(t, \mathbf{x}, \omega)}{\partial t} - \nabla \cdot (a(\mathbf{x}, \omega) \nabla y(t, \mathbf{x}, \omega)) = u(t, \mathbf{x}, \omega), & \text{in } (0, T] \times \mathcal{D} \times \Omega, \\ y(t, \mathbf{x}, \omega) = 0, & \text{on } (0, T] \times \partial\mathcal{D} \times \Omega, \\ y(0, \mathbf{x}, \omega) = y_0, & \text{in } \mathcal{D} \times \Omega, \end{cases} \tag{4.52}$$

where the random control function satisfies

$$u \in L^2(0, T; \mathcal{D}) \otimes L^2(\Omega), \quad \text{a.e,}$$

and, as before, $a(\mathbf{x}, \omega)$ is assumed to be uniformly positive in $\mathcal{D} \times \Omega$. We note here that the time-dependence of this problem introduces an additional degree of freedom which makes the system matrix here (a lot) larger than the system matrix in the steady-state case.

We use the trapezoidal rule for temporal discretization (as was done for deterministic problems in e.g. [104, 124]) and SGFEM in the spatial and the stochastic domains to get the following discrete objective function

$$\frac{\tau}{2}(\mathbf{y}-\bar{\mathbf{y}})^T\mathcal{M}_a(\mathbf{y}-\bar{\mathbf{y}})+\frac{\tau\alpha}{2}\mathbf{y}^T\mathcal{M}_b\mathbf{y}+\frac{\tau\beta}{2}\mathbf{u}^T\mathcal{M}_2\mathbf{u}, \tag{4.53}$$

where τ represents the time step size, and

$$\begin{cases}\mathcal{M}_a=\text{blkdiag}\left(\frac{1}{2}\mathcal{M},\mathcal{M},\ldots,\mathcal{M},\frac{1}{2}\mathcal{M}\right), \\ \mathcal{M}_b=\text{blkdiag}\left(\frac{1}{2}\mathcal{M}_t,\mathcal{M}_t,\ldots,\mathcal{M}_t,\frac{1}{2}\mathcal{M}_t\right),\end{cases} \tag{4.54}$$

with $\mathcal{M}$ and $\mathcal{M}_t$ as defined in (4.6). Note that, in (4.53) and (4.54), we have $\mathcal{M}_2=\mathcal{M}_a$. Here, denoting the number of time steps by n_t, we also note that

$$\mathbf{y}=\begin{bmatrix}\mathbf{y}_1\\ \vdots\\ \mathbf{y}_{n_t}\end{bmatrix},\quad \bar{\mathbf{y}}=\begin{bmatrix}\bar{\mathbf{y}}_1\\ \vdots\\ \bar{\mathbf{y}}_{n_t}\end{bmatrix}\quad \text{and } \mathbf{u}=\begin{bmatrix}\mathbf{u}_1\\ \vdots\\ \mathbf{u}_{n_t}\end{bmatrix},$$

with $\mathbf{y}_i,\bar{\mathbf{y}}_i,\mathbf{u}_i\in\mathbb{R}^{JP\times 1},\ \ i=1,\ldots,n_t$.

For an all-at-once discretization of the state equation (4.52), we use the implicit Euler method together with SGFEM to get

$$\mathcal{M}\left(\frac{\mathbf{y}^{i+1}-\mathbf{y}^i}{\tau}\right)+\mathcal{K}\mathbf{y}^{i+1}=\mathcal{M}\mathbf{u}^{i+1},$$

$$\Rightarrow\ \ \underbrace{(\mathcal{M}+\tau\mathcal{K})}_{:=\mathcal{L}_0}\mathbf{y}^{i+1}-\tau\mathcal{M}\mathbf{u}^{i+1}=\mathcal{M}\mathbf{y}^i,$$

or, equivalently,

$$\mathcal{K}_\tau\mathbf{y}-\tau\mathcal{N}\mathbf{u}=\mathbf{d},$$

where

$$\mathcal{K}_\tau := \begin{bmatrix} \mathcal{L}_0 & & & \\ -\mathcal{M} & \mathcal{L}_0 & & \\ & \ddots & \ddots & \\ & & -\mathcal{M} & \mathcal{L}_0 \end{bmatrix}, \; \mathcal{N} := \begin{bmatrix} \mathcal{M} & & & \\ & \mathcal{M} & & \\ & & \ddots & \\ & & & \mathcal{M} \end{bmatrix}, \; \mathbf{d} := \begin{bmatrix} \mathcal{M}\mathbf{y}_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

where $\mathcal{L}_0 := \mathcal{M} + \tau\mathcal{K} = G_0 \otimes (M + \tau K_0) + \tau \sum_{i=1}^{N} G_i \otimes K_i$. Observe that the matrix $\mathcal{K}_\tau$ in this case is not symmetric, unlike the matrix $\mathcal{K}$ in the stationary case.

Again, we apply first order conditions to the Lagrangian functional $\mathfrak{L}_\tau$ of this optimization problem

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

to obtain the saddle point system

$$\begin{bmatrix} \tau\mathcal{M}_1 & 0 & -\mathcal{K}_\tau^T \\ 0 & \beta\tau\mathcal{M}_2 & \tau\mathcal{N}^T \\ -\mathcal{K}_\tau & \tau\mathcal{N} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \mathbf{f} \end{bmatrix} = \begin{bmatrix} \tau\mathcal{M}_a\bar{\mathbf{y}} \\ \mathbf{0} \\ \mathbf{d} \end{bmatrix}, \tag{4.55}$$

where, from (4.54) and (4.6),

$$\begin{aligned} \mathcal{M}_1 &= \mathcal{M}_a + \alpha\mathcal{M}_b \\ &= (D \otimes \mathcal{M}) + \alpha(D \otimes \mathcal{M}_t) \\ &= D \otimes (\mathcal{M} + \alpha\mathcal{M}_t) \\ &= D \otimes G_\alpha \otimes M = D \otimes \mathcal{M}_\alpha, \end{aligned} \tag{4.56}$$

with G_α and $\mathcal{M}_\alpha$ as defined in (4.10) and (4.9), respectively. Besides,

$$D = \operatorname{diag}\left(\frac{1}{2}, 1 \ldots, 1, \frac{1}{2}\right) \in \mathbb{R}^{n_t \times n_t}. \tag{4.57}$$

We note here that

$$\mathcal{K}_\tau = (I_{n_t} \otimes \mathcal{L}_0) + (C \otimes \mathcal{M}) = I_{n_t} \otimes \left[\sum_{i=0}^{N} G_i \otimes \tilde{K}_i \right] + (C \otimes G_0 \otimes M), \tag{4.58}$$

where, as before, $\tilde{K}_0 = M + \tau K_0, \;\; \tilde{K}_i = \tau K_i, \; i = 1, \ldots, N$. The matrix $C \in \mathbb{R}^{n_t \times n_t}$ comes from the implicit Euler discretization and is given by

$$C = \begin{bmatrix} 0 & & & \\ -1 & 0 & & \\ & \ddots & \ddots & \\ & & -1 & 0 \end{bmatrix}, \tag{4.59}$$

and I_{n_t} is the identity matrix of dimension n_t. The use of other temporal discretizations is, of course, possible. The Crank-Nicolson scheme, for instance, can be written in a similar way. Moreover,

$$\mathcal{N} = I_{n_t} \otimes G_0 \otimes M, \quad \mathcal{M}_2 = D \otimes G_0 \otimes M. \tag{4.60}$$

Hence, each of the block matrices $\mathcal{K}_\tau, \mathcal{N}, \mathcal{M}_1$ and $\mathcal{M}_2$ belongs to $\mathbb{R}^{JPn_t \times JPn_t}$, since $G_i \in \mathbb{R}^{P \times P}, \; i = 0, \ldots, P-1$, and $M, K_i \in \mathbb{R}^{J \times J}, \; i = 0, \ldots, N$. So, the overall coefficient matrix in (4.55) is of dimension $3JPn_t \times 3JPn_t$.

4.3.1 Preconditioning the unsteady KKT system

As in the case of the optimality system associated with the stationary model problem, we need a good preconditioner to solve (4.55). To this end, we will proceed as before and rewrite the saddle point system (4.55) as

$$A = \begin{bmatrix} \tau \mathcal{M}_1 & 0 \\ 0 & \tau\beta \mathcal{M}_2 \end{bmatrix}, \;\; B = [-\mathcal{K}_\tau \;\; \tau \mathcal{N}], \tag{4.61}$$

in the notation of (4.11). Again, we are interested in a block-diagonal preconditioner to approximate the solution to (4.55). More precisely, we seek a preconditioner of the form

$$\widehat{\mathcal{P}} = \begin{bmatrix} A_1 & & \\ & A_2 & \\ & & S_2 \end{bmatrix},$$

with blocks $A_1 \approx \tau D \otimes G_\alpha \otimes M$ and $A_2 \approx \tau\beta D \otimes G_0 \otimes M$, and as we noted before, both approximations could be accomplished by applying a Chebyshev semi-iteration on the mass matrix M in the blocks. The matrices D, G_0 and G_α are easy to invert since they are diagonal matrices. Moreover, S_2 is an approximation to the (negative) Schur complement $S_\tau = BA^{-1}B^T$, that is,

$$S_\tau := \frac{1}{\tau}\mathcal{K}_\tau \mathcal{M}_1^{-1}\mathcal{K}_\tau^T + \frac{\tau}{\beta}\mathcal{N}\mathcal{M}_2^{-1}\mathcal{N}^T. \tag{4.62}$$

As in the time-independent case, we consider an approximation of the Schur complement of the form:

$$S_2 = \frac{1}{\tau}\left(\mathcal{K}_\tau + \widehat{\mathcal{M}}_u\right)\mathcal{M}_1^{-1}\left(\mathcal{K}_\tau + \widehat{\mathcal{M}}_u\right)^T, \tag{4.63}$$

where $\widehat{\mathcal{M}}_u$ is again determined via the 'terms-matching' procedure so that both the first and second terms in S_τ and S_2 are matched, but the cross terms in S_2 are ignored; that is, we have

$$\widehat{\mathcal{M}}_u \mathcal{M}_1^{-1}\widehat{\mathcal{M}}_u = \frac{\tau^2}{\beta}\mathcal{N}\mathcal{M}_2^{-1}\mathcal{N}^T,$$

from which we deduce that $\widehat{\mathcal{M}}_u = \gamma\mathcal{N}$, with $\gamma = \tau\sqrt{\frac{1+\alpha}{\beta}}$, by using similar arguments as before, so that

$$\begin{aligned} S_2 &= \frac{1}{\tau}\underbrace{\left(\mathcal{K}_\tau + \tau\sqrt{\frac{1+\alpha}{\beta}}\mathcal{N}\right)}_{:=\widehat{\mathcal{Z}}}\mathcal{M}_1^{-1}\left(\mathcal{K}_\tau + \tau\sqrt{\frac{1+\alpha}{\beta}}\mathcal{N}\right)^T \qquad (4.64) \\ &= \frac{1}{\tau}\left(\mathcal{K}_\tau\mathcal{M}_1^{-1}\mathcal{K}_\tau^T + \frac{\tau^2(1+\alpha)}{\beta}\mathcal{N}\mathcal{M}_1^{-1}\mathcal{N} + \tau\sqrt{\frac{1+\alpha}{\beta}}\left[\mathcal{K}_\tau\mathcal{M}_1^{-1}\mathcal{N} + \mathcal{N}\mathcal{M}_1^{-1}\mathcal{K}_\tau^T\right]\right), \end{aligned}$$

where, from (4.60), we have used the fact that $\mathcal{N} = \mathcal{N}^T$.

4.3.2 Spectral analysis and implementation issues

As in the stationary case, we have the following result regarding the eigenvalues of the preconditioned Schur complement $S_2^{-1}S_\tau$.

Theorem 4.13. *Let $\alpha \in [0, +\infty)$. Then, the eigenvalues of $S_2^{-1}S_\tau$ satisfy*

$$\lambda(S_2^{-1}S_\tau) \subset \left[\frac{1}{2(1+\alpha)}, 1\right) \quad \forall \alpha < \left(\frac{\sqrt{\kappa(\mathcal{K})}+1}{\sqrt{\kappa(\mathcal{K})}-1}\right)^2 - 1, \tag{4.65}$$

where $\mathcal{K} = \sum_{i=0}^{N} G_i \otimes K_i$ is as defined by (2.24).

Proof. Let $I_{n_t} := I$, and observe first from (4.58) that we can rewrite $\mathcal{K}_\tau$ as

$$\mathcal{K}_\tau = (I + C) \otimes (G_0 \otimes M) + I \otimes \tau \sum_{i=0}^{N} (G_i \otimes K_i) = J_0 \otimes \mathcal{M} + \tau I \otimes \mathcal{K}, \tag{4.66}$$

where

$$J_0 = I + C = \begin{bmatrix} 1 & & & \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix},$$

and $\mathcal{K}$, the coefficient matrix associated with the stationary forward problem, is positive definite. Now, using (4.27), (4.29), (4.56), (4.60), we see that

$$\begin{aligned} \mathcal{M}_1 &= D \otimes \mathcal{M}_\alpha \\ &= D \otimes \mathcal{M}\mathcal{E}_\alpha \\ &= (D \otimes \mathcal{M})(I \otimes \mathcal{E}_\alpha) \\ &= \mathcal{M}_2\mathcal{F}_\alpha = \mathcal{F}_\alpha\mathcal{M}_2, \end{aligned} \tag{4.67}$$

where $\mathcal{F}_\alpha = I \otimes \mathcal{E}_\alpha$. Observe here that since $\mathcal{F}_\alpha$ and $\mathcal{M}_2$ commute and are both invertible, one has

$$\mathcal{M}_1^{-1} = \mathcal{M}_2^{-1}\mathcal{F}_\alpha^{-1} = \mathcal{F}_\alpha^{-1}\mathcal{M}_2^{-1} = \mathcal{M}_2^{-1/2}\mathcal{F}_\alpha^{-1}\mathcal{M}_2^{-1/2}. \tag{4.68}$$

Next, define the matrix $\mathcal{X}$ by

$$\begin{aligned}\mathcal{X} &:= (D\otimes I)\mathcal{M}_2^{-1/2}\mathcal{K}_\tau\mathcal{M}_2^{-1/2} \\ &= D^{1/2}J_0D^{-1/2}\otimes I+\tau I\otimes\mathcal{M}^{-1/2}\mathcal{K}\mathcal{M}^{-1/2}. \qquad (4.69)\end{aligned}$$

Note then that $\mathcal{X}$ is similar to $J_0\otimes I+\tau I\otimes\mathcal{M}^{-1}\mathcal{K}=(D\otimes I)\mathcal{M}_2^{-1}\mathcal{K}_\tau$. Moreover, since

$$\begin{aligned}S_2^{-1}S_\tau &\sim (D\otimes I)^{-1}\mathcal{M}_2^{1/2}S_2^{-1}S_\tau\mathcal{M}_2^{-1/2}(D\otimes I) \\ &= \left[(D\otimes I)\mathcal{M}_2^{-1/2}S_2\mathcal{M}_2^{-1/2}(D\otimes I)\right]^{-1}\left[(D\otimes I)\mathcal{M}_2^{-1/2}S_\tau\mathcal{M}_2^{-1/2}(D\otimes I)\right],\end{aligned}$$

we see, from (4.62), (4.64), (4.67), (4.68) and (4.69) that

$$S_2^{-1}S_\tau\sim\left[(D\otimes I)\mathcal{M}_2^{-1/2}S_2\mathcal{M}_2^{-1/2}(D\otimes I)\right]^{-1}\left[(D\otimes I)\mathcal{M}_2^{-1/2}S_\tau\mathcal{M}_2^{-1/2}(D\otimes I)\right]=$$
$$\left[\beta\mathcal{X}\mathcal{F}_\alpha^{-1}\mathcal{X}^T+\tau^2(1+\alpha)\mathcal{F}_\alpha^{-1}+\tau\sqrt{\beta(1+\alpha)}\left(\mathcal{X}\mathcal{F}_\alpha^{-1}+\mathcal{F}_\alpha^{-1}\mathcal{X}^T\right)\right]^{-1}\left(\beta\mathcal{X}\mathcal{F}_\alpha^{-1}\mathcal{X}^T+\tau^2I\right).$$

Now, consider the Raleigh quotient

$$R(x):=\frac{x^T\left[\beta\mathcal{X}\mathcal{F}_\alpha^{-1}\mathcal{X}^T+\tau^2I\right]x}{x^T\left[\beta\mathcal{X}\mathcal{F}_\alpha^{-1}\mathcal{X}^T+\tau^2(1+\alpha)\mathcal{F}_\alpha^{-1}+\tau\sqrt{\beta(1+\alpha)}\left(\mathcal{X}\mathcal{F}_\alpha^{-1}+\mathcal{F}_\alpha^{-1}\mathcal{X}^T\right)\right]x}.$$

But then,

$$\mathcal{X}\mathcal{F}_\alpha^{-1}+\mathcal{F}_\alpha^{-1}\mathcal{X}^T=D^{1/2}(J_0+J_0^T)D^{-1/2}\otimes\mathcal{E}_\alpha^{-1}+\tau I\otimes\mathcal{M}^{-1/2}(\mathcal{K}\mathcal{E}_\alpha^{-1}+\mathcal{E}_\alpha^{-1}\mathcal{K})\mathcal{M}^{-1/2}.$$

Since the matrix $D^{1/2}(J_0+J_0^T)D^{-1/2}$ is the sum of two positive definite matrices, it is therefore positive definite. Besides, from (4.33), we know that

$$x^T(\mathcal{K}\mathcal{E}_\alpha^{-1}+\mathcal{E}_\alpha^{-1}\mathcal{K})x>0,\quad\forall\alpha<\left(\frac{\sqrt{\kappa(\mathcal{K})}+1}{\sqrt{\kappa(\mathcal{K})}-1}\right)^2-1.$$

It follows that $\mathcal{X}\mathcal{F}_\alpha^{-1}+\mathcal{F}_\alpha^{-1}\mathcal{X}^T\succ 0$. Furthermore, it is easy to check that both $\mathcal{X}\mathcal{F}_\alpha^{-1}\mathcal{X}^T$ and $\mathcal{F}_\alpha^{-1}$ are also positive definite. Hence, using (4.28), we obtain

$$R(x)\leq\frac{x^T\left[\beta\mathcal{X}\mathcal{F}_\alpha^{-1}\mathcal{X}+\tau^2(1+\alpha)\mathcal{F}_\alpha^{-1}\right]x}{x^T\left[\beta\mathcal{X}\mathcal{F}_\alpha^{-1}\mathcal{X}^T+\tau^2(1+\alpha)\mathcal{F}_\alpha^{-1}+\tau\sqrt{\beta(1+\alpha)}\left(\mathcal{X}\mathcal{F}_\alpha^{-1}+\mathcal{F}_\alpha^{-1}\mathcal{X}^T\right)\right]x}<1.$$

from which we deduce that $\lambda_{\max} := \max R(x) < 1$.

Next, we again employ the Cauchy-Schwarz Inequality as in the proof of the second part of Theorem 4.7 to obtain

$$\left(x^T \mathcal{X} \mathcal{F}_\alpha^{-1} x\right)^2 \leq \left(x^T \mathcal{X} \mathcal{F}_\alpha^{-1} \mathcal{X}^T x\right) \left(x^T \mathcal{F}_\alpha^{-1} x\right) \tag{4.70}$$

and

$$\left(x^T \mathcal{F}_\alpha^{-1} \mathcal{X}^T x\right)^2 \leq \left(x^T \mathcal{X} \mathcal{F}_\alpha^{-1} \mathcal{X}^T x\right) \left(x^T \mathcal{F}_\alpha^{-1} x\right), \tag{4.71}$$

so that

$$x^T (\mathcal{X} \mathcal{F}_\alpha^{-1} + \mathcal{F}_\alpha^{-1} \mathcal{X}^T) x \leq 2\sqrt{\left(x^T \mathcal{X} \mathcal{F}_\alpha^{-1} \mathcal{X}^T x\right) \left(x^T \mathcal{F}_\alpha^{-1} x\right)}. \tag{4.72}$$

Hence, using (4.72), (4.37) and (4.33), we get

$$\begin{aligned}
R(x) &= \frac{x^T \left[\beta \mathcal{X} \mathcal{F}_\alpha^{-1} \mathcal{X}^T + \tau^2 I\right] x}{x^T \left[\beta \mathcal{X} \mathcal{F}_\alpha^{-1} \mathcal{X}^T + \tau^2 (1+\alpha) \mathcal{F}_\alpha^{-1} + \tau \sqrt{\beta(1+\alpha)} \left(\mathcal{X} \mathcal{F}_\alpha^{-1} + \mathcal{F}_\alpha^{-1} \mathcal{X}^T\right)\right] x} \\
&\geq \frac{x^T \left[\beta \mathcal{X} \mathcal{F}_\alpha^{-1} \mathcal{X}^T + \tau^2 I\right] x}{\beta x^T \mathcal{X} \mathcal{F}_\alpha^{-1} \mathcal{X}^T x + \tau^2 (1+\alpha) x^T \mathcal{F}_\alpha^{-1} x + 2\tau \sqrt{\beta(1+\alpha) \left(x^T \mathcal{X} \mathcal{F}_\alpha^{-1} \mathcal{X}^T x\right) \left(x^T \mathcal{F}_\alpha^{-1} x\right)}} \\
&= \frac{x^T \left[\beta \mathcal{X} \mathcal{F}_\alpha^{-1} \mathcal{X}^T + \tau^2 I\right] x}{\left[\beta^{1/2} (x^T \mathcal{X} \mathcal{F}_\alpha^{-1} \mathcal{X}^T x)^{1/2} + \tau (1+\alpha)^{1/2} (x^T \mathcal{F}_\alpha^{-1} x)^{1/2}\right]^2} \\
&\geq \frac{x^T \left[\beta \mathcal{X} \mathcal{F}_\alpha^{-1} \mathcal{X}^T + \tau^2 I\right] x}{2\left[\beta x^T \mathcal{X} \mathcal{F}_\alpha^{-1} \mathcal{X}^T x + \tau^2 (1+\alpha) x^T \mathcal{F}_\alpha^{-1} x\right]} \\
&\geq \frac{\beta x^T \mathcal{X} \mathcal{F}_\alpha^{-1} \mathcal{X}^T x + \tau^2 x^T \mathcal{F}_\alpha^{-1} x}{2\left[\beta x^T \mathcal{X} \mathcal{F}_\alpha^{-1} \mathcal{X}^T x + \tau^2 (1+\alpha) x^T \mathcal{F}_\alpha^{-1} x\right]} \\
&\geq \frac{x^T \mathcal{F}_\alpha^{-1} x}{2(1+\alpha) x^T \mathcal{F}_\alpha^{-1} x} = \frac{1}{2(1+\alpha)},
\end{aligned} \tag{4.73}$$

which shows that $\lambda_{\min} := \min R(x) \geq \frac{1}{2(1+\alpha)}$, thereby concluding the proof of the theorem.

□

Remark 4.14. *Note that, we can argue exactly the same way as in Theorem 4.8 to characterize the spectrum of the preconditioned KKT system in the unsteady case, if we* *define $\mathcal{A}$ as the global coefficient matrix and $\mathcal{P}_s$ as*

$$\mathcal{P}_s = \begin{bmatrix} A & 0 \\ 0 & S_2 \end{bmatrix},$$

where A and S_2 are given by (4.61) and (4.64), respectively. Furthermore, the convergence result of Corollary 4.12 still holds.

It turns out that, if we specifically use Legendre polynomials and piecewise linear (or bilinear) approximation in the SGFEM discretization of the SOCPs considered herein, then the following result proved by Powell and Elman enables us to further bound the parameter α in Theorems 4.7 and 4.13 above.

Proposition 4.15. *[110, Lemma 3.7] Let the matrices G_k in (2.28) be defined using normalized Legendre polynomials in uniform random variables on a bounded symmetric interval $[-\nu, \nu]$, and suppose that piecewise linear (or bilinear) approximation is used for the spatial discretization, on quasi-uniform meshes. Let (λ_i, ϑ_i) be the eigenpairs associated with the N-term KLE of the random field a_N. Then $\kappa(\mathcal{K}) \leq \Phi/\Psi$, where $\Phi = c_2\mathbb{E}(a) + \eta$ and $\Psi = c_1 h^2 \mathbb{E}(a) - \eta$, with*

$$\eta = c_2 \sigma_a C_{n+1}^{\max} \sum_{i=1}^{N} \sqrt{\lambda_i} ||\vartheta_i(\mathbf{x})||_\infty,$$

where $C_{n+1}^{\max}$ is the maximal root of the Legendre polynomial of degree $n+1$, σ_a is the standard deviation of the random field a, h is the spatial discretization parameter, and c_1 and c_2 are constants independent of $h, N,$ and n.

We can now state the following result regarding the proposed approximations S_1 and S_2 given, respectively, by (4.25) and (4.64).

Corollary 4.16. *Let $\alpha \in [0, +\infty)$ and define the matrix $\widehat{S}_i$, $i = 1, 2$, by*

$$\widehat{S}_i = \begin{cases} S \textit{ as given by } (4.17), & i = 1, \\ S_{\mathcal{T}} \textit{ as given by } (4.62), & i = 2. \end{cases} \tag{4.74}$$

Then, the spectrum of $S_i^{-1}\widehat{S}_i$ satisfies

$$\lambda(S_i^{-1}\widehat{S}_i) \subset \left[\frac{1}{2(1+\alpha)}, 1\right), \quad \alpha < \tilde{\mu}^2 - 1, \quad i = 1, 2, \tag{4.75}$$

where $\tilde{\mu} = \frac{1+p+2\sqrt{p}}{p-1}$, $p \neq 1$ and $p := \Phi/\Psi$, with Φ and Ψ as defined in Proposition 4.15.

Proof. By Proposition 4.15, the condition number of the stochastic Galerkin matrix $\mathcal{K}$ is bounded by p^2. Substituting this into the bound in Theorems 4.7 and 4.13 immediately yields the result (4.75). □

Next, we derive a practical version of S_2. Observe from (4.58), (4.60) and (4.64) that

$$\begin{aligned}
\widehat{\mathcal{Z}} &:= \mathcal{K}_\tau + \gamma\mathcal{N} \\
&= [(I_{n_t} \otimes \mathcal{L}_0) + (C \otimes \mathcal{M})] + \gamma(I_{n_t} \otimes \mathcal{M}) \\
&= I_{n_t} \otimes \left[\left(G_0 \otimes (M + \tau K_0) + \tau \sum_{i=1}^{N} G_i \otimes K_i\right) + \gamma(G_0 \otimes M)\right] + (C \otimes \mathcal{M}) \\
&= I_{n_t} \otimes \left[G_0 \otimes \mathcal{Y} + \tau \sum_{i=1}^{N} G_i \otimes K_i\right] + (C \otimes G_0 \otimes M),
\end{aligned} \tag{4.76}$$

where $\mathcal{Y} = (1+\gamma)\, M + \tau K_0$. Hence, using similar arguments as in Section 4.2.2 we can now approximate $\widehat{\mathcal{Z}}$ using

$$\widehat{\mathcal{Z}}_0 := I_{n_t} \otimes G_0 \otimes \mathcal{Y}. \tag{4.77}$$

In practice, we thus approximate S_2 by applying a cheap multigrid process to $\mathcal{Y}$ in each of the diagonal blocks of $\widehat{\mathcal{Z}}_0$ and $\widehat{\mathcal{Z}}_0^T$. The expression (4.77) is admittedly not the best possible approximation to S_2 due essentially to the same reasons provided in the case of S_1 in Section 4.2.2. Besides, the absence of the term $C \otimes G_0 \otimes M$ in $\widehat{\mathcal{Z}}_0$ would likely impact negatively on the performance of $\widehat{\mathcal{Z}}_0$. Again, solves with $\widehat{\mathcal{Z}}$ via the preconditioned Richardson iteration can substantially mitigate these short-comings.

4.3.3 Low-rank tensor solver for the unsteady problem

As can be seen from (4.58), for instance, the time-dependent problem leads to an additional Kronecker product. Indeed, although the low-rank solver presented in the stationary case reduces storage problems in large-scale simulations, the low-rank factors become infeasible in higher dimensions. Further data compression can, fortunately, be achieved with more advanced high-dimensional tensor product decompositions. Together with preconditioned MINRES, we henceforth solve the linear system discussed in the rest of this thesis using an elegant and robust tensor format called *Tensor Train* (TT) format which was introduced in [97]. To that end, we proceed next to give a general overview of the TT decomposition.

First, we recall that a tensor $\mathbf{y} := \mathbf{y}(i_1, \ldots, i_d),\ i_k = 1, \ldots, n_k$ is an $n_1 \times n_2 \times \ldots \times n_d$ multi-dimensional array, where the integers $n_1, n_2, \ldots, n_d$ are called the mode sizes and d is the order of $\mathbf{y}$. The tensor $\mathbf{y}$ admits a tensor train decomposition or TT- format [29, 49, 52, 70, 97] if it can be expressed as

$$\mathbf{y}(i_1, \ldots, i_d) = \sum_{s_1 \ldots s_{d-1}=1}^{r_1 \ldots r_{d-1}} \mathbf{y}^{(1)}_{s_1}(i_1)\mathbf{y}^{(2)}_{s_1,s_2}(i_2) \cdots \mathbf{y}^{(d-1)}_{s_{d-2},s_{d-1}}(i_{d-1})\mathbf{y}^{(d)}_{s_{d-1}}(i_d), \tag{4.78}$$

where each factor $\mathbf{y}^{(k)},\ k = 1, \ldots, d$, (commonly known as *TT block* or *core*) is an $r_{k-1} \times r_k$ matrix for each fixed $i_k,\ 1 \leq i_k \leq n_k$. Moreover, the numbers r_k are called the *TT ranks.* More precisely, $\mathbf{y}^{(k)}$ is a three-dimensional array, and it can essentially be treated as an $r_{k-1} \times i_k \times r_k$ array with elements $\mathbf{y}^{(k)}(s_{k-1}, i_k, s_k) = \mathbf{y}^{(k)}_{s_{k-1},s_k}(i_k)$. Here, the boundary conditions $r_0 = r_d = 1$ are imposed on the decomposition to make the matrix-by-matrix products a scalar. The decomposition can be expressed in index form as

$$\mathbf{y}(i_1, \ldots, i_d) = \sum_{s_1 \ldots s_{d-1}=1}^{r_1 \ldots r_{d-1}} \mathbf{y}^{(1)}(s_0, i_1, s_1)\mathbf{y}^{(2)}(s_1, i_2, s_2) \cdots \mathbf{y}^{(d)}(s_{d-1}, i_d, s_d), \tag{4.79}$$

where $s_0 = s_d = 1$. It turns out that TT-decomposition yields a low-rank format for tensors as it is derived by a repeated application of low-rank approximation [97]. To see this [31], set

$$\overline{i_2 \cdots i_d} = i_2 + (i_3 - 1)n_2 + \cdots + (i_d - 1)n_2 n_3 \cdots n_{d-1}. \tag{4.80}$$

Then, by *regrouping* of indices, one can rewrite $\mathbf{y}$ as a matrix $Y_1 \in \mathbb{R}^{n_1 \times n_2 \cdots n_d}$ with $Y_1(i_1, \overline{i_2 \cdots i_d}) = \mathbf{y}(i_1, \ldots, i_d)$. Thus, applying a low-rank SVD to the matrix Y_1 yields

$$Y_1 \approx U_1 \Sigma_1 V_1^T, \quad U_1 \in \mathbb{R}^{n_1 \times r_1}, \ \ V_1 \in \mathbb{R}^{n_2 \cdots n_d \times r_1}.$$

The first factor U_1 is of moderate dimension and can be stored as $\mathbf{y}^{(1)}_{s_1}(i_1) = U_1(i_1, s_1)$, where $s_1 = 1, \ldots, r_1$ and $i_1 = 1, \ldots, n_1$. The remaining matrix $\Sigma_1 V_1^T$ depends on s_1 and $i_2 \cdots i_d$. Next, we regroup these indices as follows

$$Y_2(\overline{s_1 i_2}, \overline{i_3 \cdots i_d}) = \Sigma_1(s_1, s_1) V_1^T(s_1, \overline{i_2 \cdots i_d}),$$

and compute the next SVD:

$$Y_2 \approx U_2 \Sigma_2 V_2^T, \quad U_2 \in \mathbb{R}^{r_1 n_2 \times r_2}, \;\; V_2 \in \mathbb{R}^{n_3 \cdots n_d \times r_2}.$$

Now, U_2 can be reshaped to a 3D tensor of moderate size $\mathbf{y}^{(2)}_{s_1,s_2}(i_2) = U_2(\overline{i_2 s_1}, s_2)$, where $\overline{i_2 s_1} = i_2 + (s_1 - 1)n_2$; the decomposition is also applied to $\Sigma_2 V_2^T$. Proceeding in this manner, one eventually obtains the TT format (4.78) with the total storage of at most dnr^2 memory cells, where $r_k \leq r,\ n_k \leq n$. In particular, if r is small, then this requirement is much smaller than the storage of the full array, n^d.

For our purposes in this thesis, we henceforth narrow down our discussion on TT decomposition to the three independent variables t, ω and $\mathbf{x}$ of the solution $\mathbf{y}$ of the KKT systems. Here, we separate t, ω and $\mathbf{x}$, but not the inner components of $\mathbf{x}$. Now, note that the elements of the tensor $\mathbf{y}$ can be naturally enumerated by three indices i, j, k, corresponding to t, ω and $\mathbf{x}$, respectively. We can then consider $\mathbf{y}$ as a three-dimensional tensor with elements

$$\mathbf{y}(i, j, k) \approx \sum_{s_1, s_2 = 1}^{r_1, r_2} \mathbf{y}^{(1)}_{s_1}(i) \mathbf{y}^{(2)}_{s_1,s_2}(j) \mathbf{y}^{(3)}_{s_2}(k) \tag{4.81}$$

with r_1, r_2 as the TT ranks, $\mathbf{y}^{(m)}$, $m = 1, 2, 3$ as the *TT blocks* and $\mathbf{y}^{(1)} \in \mathbb{R}^{n_t \times r_1}$, $\mathbf{y}^{(2)} \in \mathbb{R}^{r_1 \times P \times r_2}$ and $\mathbf{y}^{(3)} \in \mathbb{R}^{r_2 \times J}$. Notice that we can fix some of the indices, e.g. $\mathbf{y}^{(2)}(j) \in \mathbb{R}^{r_1 \times r_2}$ is a matrix *slice*, $\mathbf{y}^{(2)}_{s_1,s_2} \in \mathbb{R}^P$ is a vector, and $\mathbf{y}^{(2)}_{s_1,s_2}(j)$ is a scalar. The total number of elements in all factors is $n_t r_1 + r_1 P r_2 + r_2 J = \mathcal{O}(Jr + Pr^2)$, where $r \geq r_1, r_2$, since in our case $J \sim n_t \gg P$. Therefore, if $r \ll J$, the amount of memory consumed by the TT format is much less than JPn_t, needed for the full vector $\mathbf{y}$. Particular values of r_1, r_2 depend on the accuracy we enforce in Eq. (4.81). 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 decomposition, using a sequence of singular value decompositions (SVD) [97]. In what follows we will use $\mathbf{y}(i,j,k)$ and $\mathbf{y}(\overline{ijk})$, where $\overline{ijk} = (i-1)PJ + (j-1)J + k$, interchangeably to describe the elements of a tensor $\mathbf{y}$.

Proposition 4.17. *The 3D tensor* $\mathbf{y} = \left[\mathbf{y}(\overline{ijk})\right]_{i,j,k=1}^{n_t,P,J}$ *defined in (4.81) satisfies*

$$\mathbf{y}(i,j,k) = \sum_{s_1,s_2=1}^{r_1,r_2} \mathbf{y}_{s_1}^{(1)}(i)\mathbf{y}_{s_1,s_2}^{(2)}(j)\mathbf{y}_{s_2}^{(3)}(k) \quad \Leftrightarrow \quad \mathbf{y} = \sum_{s_1,s_2=1}^{r_1,r_2} \mathbf{y}_{s_1}^{(1)} \otimes (\mathbf{y}_{s_1,s_2}^{(2)})^T \otimes (\mathbf{y}_{s_2}^{(3)})^T, \tag{4.82}$$

where the tensor $\mathbf{y}_{s_1,s_2}^{(2)} \in \mathbb{R}^{1\times P\times 1}$ *is understood as a* $1 \times P$ *matrix.*

Proof. Define the vector Z_{s_2} by

$$Z_{s_2} \equiv \sum_{s_1=1}^{r_1} \mathbf{y}_{s_1}^{(1)} \otimes (\mathbf{y}_{s_1,s_2}^{(2)})^T.$$

Observe then from (1.7) that

$$Z_{s_2} \equiv \sum_{s_1=1}^{r_1} \mathbf{y}_{s_1}^{(1)} \otimes (\mathbf{y}_{s_1,s_2}^{(2)})^T \quad \Leftrightarrow \quad Z_{s_2}(\overline{ij}) = \sum_{s_1=1}^{r_1} \mathbf{y}_{s_1}^{(1)}(i)\mathbf{y}_{s_1,s_2}^{(2)}(j). \tag{4.83}$$

Thus, using (4.83), we have

$$\mathbf{y} = \sum_{s_2=1}^{r_2} Z_{s_2} \otimes (\mathbf{y}_{s_2}^{(3)})^T \quad \Leftrightarrow \quad \mathbf{y}(\overline{ijk}) = \sum_{s_2=1}^{r_2} Z_{s_2}(\overline{ij})\mathbf{y}_{s_2}^{(3)}(k) = \sum_{s_1,s_2=1}^{r_1,r_2} \mathbf{y}_{s_1}^{(1)}(i)\mathbf{y}_{s_1,s_2}^{(2)}(j)\mathbf{y}_{s_2}^{(3)}(k),$$

thereby completing the proof of (4.82). □

The complexity of the TT-SVD is $\mathcal{O}(J^2Pn_t)$ when 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 (4.81), the matrix-vector product can be written as follows [118, 97],

$$\mathbf{g} = \boldsymbol{A}\mathbf{y} = \sum_{s_1,s_2=1}^{r_1,r_2} \sum_{q_1,q_2=1}^{R,R} \left(A_{q_1}\mathbf{y}_{s_1}^{(1)}\right) \otimes \left(\delta_{q_1,q_2}B_{q_1}(\mathbf{y}_{s_1,s_2}^{(2)})^T\right) \otimes \left(C_{q_2}(\mathbf{y}_{s_2}^{(3)})^T\right), \tag{4.84}$$

where $\delta_{q_1,q_2} = 1$ if $q_1 = q_2$ and zero otherwise. To see this, assume that the tensor $\mathbf{g}$ in

(4.84) admits TT format indexed by, say, ζ_1 and ζ_2, with

$$\mathbf{g} = \sum_{\zeta_1,\zeta_2=1}^{\rho_1,\rho_2} \mathbf{g}^{(1)}_{\zeta_1} \otimes (\mathbf{g}^{(2)}_{\zeta_1,\zeta_2})^T \otimes (\mathbf{g}^{(3)}_{\zeta_2})^T. \tag{4.85}$$

Note then that the indices ζ_1 and ζ_2 must be independent and that one should have

$$\mathbf{g}^{(1)}_{\zeta_1} = A_{q_1}\mathbf{y}^{(1)}_{s_1} \Rightarrow \zeta_1 = \overline{q_1 s_1} = q_1 + (s_1-1)R,$$

$$\mathbf{g}^{(2)}_{\zeta_1,\zeta_2} = B_{q_1}(\mathbf{y}^{(2)}_{s_1,s_2})^T\delta_{q_1,q_2} \Rightarrow \zeta_1 = \overline{q_1 s_1},\ \zeta_2 = \overline{q_2 s_2} = q_2 + (s_2-1)R,$$

$$\mathbf{g}^{(3)}_{\zeta_2} = C_{q_2}(\mathbf{y}^{(3)}_{s_2})^T \Rightarrow \zeta_2 = \overline{q_2 s_2}.$$

To explain why we introduced the indices q_1, q_2, suppose that the above TT representation of $\mathbf{g}$ were not so; then

$$\mathbf{g}^{(1)}_{\zeta_1} = A_q\mathbf{y}^{(1)}_{s_1} \Rightarrow \zeta_1 = \overline{q s_1} = q + (s_1-1)R,$$

$$\mathbf{g}^{(2)}_{\zeta_1,\zeta_2} = B_q(\mathbf{y}^{(2)}_{s_1,s_2})^T \Rightarrow \zeta_1 = \overline{q s_1},\ \zeta_2 = s_2,$$

and

$$\mathbf{g}^{(3)}_{\zeta_2} = C_q(\mathbf{y}^{(3)}_{s_2})^T = C_q(\mathbf{y}^{(3)}_{\zeta_2})^T.$$

But then, the block $\mathbf{g}^{(3)}$ is now enumerated by two indices q and ζ_2 instead of only one, thereby contradicting a unique TT representation.

Similarly, linear combination, inner product and truncation (or rounding) operators applied to tensors in TT format are given explicitly in [97]. Each bracket in the right-hand side of (4.84) is a larger TT block, the new rank indices are $s_1' = \overline{q_1 s_1}$, $s_2' = \overline{q_2 s_2}$, and hence the TT ranks are Rr_1, Rr_2. However, $\mathbf{g}$ might be approximated accurately enough with much smaller ranks. When applied to the TT format (4.84) instead of the full tensor, the TT-SVD requires $\mathcal{O}(JR^2r^2 + PR^3r^3)$ operations [97]. 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 (or TT truncation) [1, 8, 29, 78].

The matrix setup for MINRES previously discussed in Section 4.2.4 is of course a special case of the more general tensor problem. As we noted, algebraic operations (matrix, scalar products and additions) and the SVD re-compression procedure in the TT format allow to rewrite any classical iterative method, keeping all vectors in the tensor format and performing only structured operations [8, 29, 77].

As pointed out in [29], the TT-format is stable in the sense that one can always find the best approximation of tensors computed via a sequence of QR and SVD decompositions of auxiliary matrices. The TT-decomposition algorithm is implemented in the TT-toolbox [98] and comes with a number of basic linear algebra operations, such as addition, subtraction, matrix-by-vector product, etc. Unfortunately, these operations lead to prohibitive increase in the TT-ranks. Thus, one necessarily has to truncate (or round) the resulting tensor after implementing each of the operations. Generally speaking, the excessive rank growth that characterizes TT-MINRES usually slows down the convergence rate of the algorithm, though. It is, nevertheless, just fine for the size of the linear system considered for numerical experiments in this chapter. In Chapter 5, we extend and adapt alternating solvers in TT format to efficiently solve the higher dimensional problems considered therein.

There are, of course, other tensor formats such as canonical, hierarchical and Tucker formats which could be used to represent tensors [49] and hence solve our linear systems. However, our choice of TT-format (or TT toolbox) is due to its relative elegance and convenience in implementation. The details of its implementation are found in [98]. A comprehensive overview of low-rank tensor decompositions can be found in [49] and the references therein. In our numerical experiments, we use preconditioned MINRES, together with the TT toolbox, to solve the linear system (4.55).

4.4 Numerical experiments

In this section, we present some numerical results using the the same covariance function (3.25). Moreover, as in Chapter 3, we choose $\xi = \{\xi_1, \ldots, \xi_N\}$ such that $\xi_j \sim \mathcal{U}[-1, 1]$, and $\{\psi_j\}$ are N-dimensional Legendre polynomials with support in $[-1, 1]^N$. The spatial discretization uses $\mathbf{Q}_1$ finite elements. We equally investigate the behavior of the solvers (low-rank MINRES and TT-MINRES) for different values of the stochastic discretization

Table 4.1: Simulation results showing the total number of iterations from low-rank preconditioned MINRES and the total CPU times (in seconds) using the mean-based preconditioner $\mathcal{Z}_0$ in (4.44) with $\alpha = 1$, $\beta \in \{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$, $\sigma_a = 0.1$, and selected spatial (J) and stochastic (P) degrees of freedom.

LR-MINRES	# iter (t)	# iter (t)	# iter (t)	# iter (t)
J P	481	1985	8065	32513
$\beta = 10^{-2}$				
20	25 (32.8)	25 (115.4)	27 (250.5)	29 (736.6)
84	25 (119.7)	27 (380.4)	27 (582.2)	29 (1619.6)
210	25 (141.6)	27 (392.8)	27 (594.69)	29 (1673.9)
$\beta = 10^{-3}$				
20	21 (25.7)	21 (113.8)	25 (260.9)	25 (666.8)
84	21 (128.9)	23 (363.7)	25 (607.6)	25 (1438.1)
210	21 (145.6)	23 (385.5)	25 (600.8)	25 (1471.8)
$\beta = 10^{-4}$				
20	19 (8.2)	21 (17.4)	23 (67.4)	23 (618.3)
84	19 (18.8)	21 (42.5)	23 (229.7)	23 (1313.7)
210	19 (19.6)	21 (44.9)	23 (276.9)	23 (1450.0)
$\beta = 10^{-5}$				
20	17 (19.6)	17 (84.8)	21 (223.7)	21 (578.3)
84	17 (99.9)	19 (306.4)	21 (520.7)	21 (1217.2)
210	17 (115.4)	19 (313.63)	21 (515.6)	23 (1322.6)

parameters J, P, σ_a, as well as α and β. Besides, we implement each of the mean-based preconditioners $\mathcal{Z}_0$ and $\widehat{\mathcal{Z}}_0$ as given, respectively, by (4.44) and (4.77) using one V-cycle of AMG [21] with symmetric Gauss-Seidel (SGS) smoothing to approximately invert $\tilde{K}_0$. In the considered unsteady SOCP model (that is, in Section 4.3), the resulting linear systems were solved for time $T = 1$. Moreover, the target in both models is the stochastic solution of the forward model with right hand side 1 and zero Dirichlet boundary conditions[3].

Tables 4.1, 4.3, 4.4 and 4.5 show the results from the low-rank preconditioned MINRES for the model constrained by steady-state diffusion equation. In Table 4.2 we give the total dimensions of the KKT systems in (4.8) for various discretization parameters which we used to obtain the results in Tables 4.1. Herein, h is the spatial mesh size and the dimensions range between $28,000$ and 20 million.

The results in Tables 4.1, 4.4 and 4.5 were obtained with $\alpha = 1$, whereas those in Table 4.3 were computed with $\alpha = 0$. We have solved the linear systems using our pro-

[3] Note that this is not an 'inverse crime' as the right-hand side of the forward model used is deterministic, unlike in the state equation.

Table 4.2: Dimension of global coefficient matrix $\mathcal{A}$ in (4.8); here $\dim(\mathcal{A}) = 3JP$.

$J(h)$ $P(N,n)$	$481\left(\frac{1}{2^4}\right)$	$1985\left(\frac{1}{2^5}\right)$	$8065\left(\frac{1}{2^6}\right)$	$32513\left(\frac{1}{2^7}\right)$
$20\ (N=3, n=3)$	$28,860$	$119,100$	$483,900$	$1,950,780$
$84\ (N=6, n=3)$	$121,212$	$500,220$	$2,032,380$	$8,193,276$
$210\ (N=6, n=4)$	$303,030$	$1,250,550$	$5,080,950$	$20,483,190$

Table 4.3: Simulation results using the mean-based preconditioner $\mathcal{Z}_0$ in (4.44) with $\sigma_a = 0.1,\ \alpha = 0,\ \beta \in \{10^{-3}, 10^{-4}, 10^{-5}\}$, and $J = 1985$ $(h = \frac{1}{2^5})$.

LR-MINRES	# iter (t)	# iter (t)	# iter (t)
P	20	84	210
$\dim(\mathcal{A}) = 3JP$	$119,100$	$500,220$	$1,250,550$
$\beta = 10^{-3}$	19 (96.4)	21 (336.0)	21 (347.93)
$\beta = 10^{-4}$	17 (86.3)	19 (302.6)	19 (305.64)
$\beta = 10^{-5}$	15 (77.4)	17 (273.6)	17 (283.24)

posed block-diagonal preconditioner, together with the approximation S_1 for the Schur complement S. To compare their practical performances, we use both the mean-based preconditioner in (4.44) (denoted henceforth by MBP) and the preconditioned Richardson iteration in Algorithm 4.3 (with just 4 iterations, and denoted by PRI) for approximating[4] $\mathcal{Z}$ in S_1.

We observe first that Table 4.1 confirms our theoretical prediction that with a relatively low variance (here $\sigma_a = 0.1$), our proposed block-diagonal preconditioner, when used together with MBP, is robust with respect to the discretization parameters. Furthermore, Table 4.5 shows that the preconditioner performs relatively better with PRI than it does with MBP, especially as the standard deviation σ_a increases from 1% to 40%. Indeed, the iterations are clearly indicative of benign dependence of PRI on σ_a, unlike MBP. In general, the iterations obtained with MBP took slightly less CPU time, though. So, these results generally suggest that PRI should be preferred to MBP when dealing with higher fluctuations in the random input a. We remark here, though, that for $\sigma_a > 0.5$, we can no longer guarantee the positive-definiteness of the matrix $\mathcal{K}$ corresponding to the forward problem [110].

We have reported in Table 4.4 the values of the tracking term and the cost functional

[4] Recall that the PRI method uses MBP for approximate solves.

Table 4.4: Tracking term and the cost functional in the steady-state model using the mean-based preconditioner $\mathcal{Z}_0$ in (4.44) for different values of β and with $\alpha = 1,\ \sigma_a = 0.1,\ J = 1985\ (h = \frac{1}{2^5}),\ P = 84\ (N = 6, n = 3)$.

β	10^{-2}	10^{-4}	10^{-6}	10^{-10}
$\lVert y - \bar{y} \rVert^2_{L^2(\mathcal{D}) \otimes L^2_\rho(\Gamma)}$	5.1×10^{-3}	1.8×10^{-4}	1.2×10^{-4}	1.2×10^{-4}
$\mathcal{J}(y, u)$	1.4×10^{-2}	4.2×10^{-4}	2.5×10^{-4}	2.5×10^{-4}

Table 4.5: Simulation results comparing between mean-based preconditioning (MBP) and the preconditioned Richardson iteration (PRI) in approximating S_1 in low-rank preconditioned MINRES with $\alpha = 1,\ \beta = 10^{-4}$.

LR-MINRES	# iter (t)	# iter (t)	# iter (t)	# iter (t)
$J(h)$ P	$481\ \left(h = \frac{1}{2^4}\right)$	$1985\ \left(h = \frac{1}{2^5}\right)$	$8065\ \left(h = \frac{1}{2^6}\right)$	$32513\ \left(h = \frac{1}{2^7}\right)$
$\sigma_a = 0.01$ with MBP				
20	17 (7.4)	19 (16.7)	19 (53.4)	21 (544.8)
84	17 (17.0)	19 (39.0)	19 (190.0)	21 (1190.0)
210	17 (18.4)	19 (40.4)	19 (470.0)	21 (1230.2)
$\sigma_a = 0.1$ with MBP				
20	19 (8.2)	21 (17.4)	23 (67.4)	23 (618.3)
84	19 (18.6)	21 (42.5)	23 (229.7)	23 (1313.7)
210	19 (19.8)	21 (44.9)	23 (576.9)	23 (1450.0)
$\sigma_a = 0.4$ with MBP				
20	33 (13.8)	37 (28.0)	41 (115.3)	43 (1049.8)
84	35 (33.8)	41 (84.5)	45 (447.0)	47 (2610.4)
210	41 (41.9)	47 (98.4)	47 (782.3)	55 (3161.1)
$\sigma_a = 0.01$ with PRI				
20	15 (13.5)	15 (20.5)	17 (82.2)	19 (1142.4)
84	15 (31.8)	15 (57.6)	17 (332.5)	19 (2117.4)
210	15 (35.0)	15 (61.9)	17 (314.9)	19 (2777.2)
$\sigma_a = 0.1$ with PRI				
20	15 (14.2)	17 (23.6)	17 (100.4)	19 (560.0)
84	15 (32.4)	17 (66.5)	17 (350.6)	19 (2124.0)
210	15 (34.4)	17 (82.9)	17 (375.5)	19 (2463.9)
$\sigma_a = 0.4$ with PRI				
20	15 (13.6)	17 (27.1)	19 (109.2)	21 (1158.3)
84	15 (34.6)	17 (78.7)	19 (402.7)	19 (2577.4)
210	15 (34.5)	17 (80.7)	19 (414.5)	21 (2958.2)

for $\alpha = 1$ and $\sigma_a = 0.1$. As expected, the tracking term gets smaller and smaller as the regularization parameter β decreases, and the cost functional also decreases accordingly converging, respectively, to 1.2×10^{-4} and 2.5×10^{-4}.

Observe from Tables 4.1 and 4.5 that the timings for $P = 84$ and $P = 210$ are nearly constant but much higher than those for $P = 20$. Our extensive numerical experiments revealed that the timings, in general, have a strong dependence on the number of random variables N, which in turn determines P; see also [13]. Now, recall that for $P = 20$ the (stochastic) matrices $G_k \in \mathbb{R}^{P \times P}$ used in the simulations were obtained with only $N = 3$ random variables, whereas the other two cases were obtained with $N = 6$ random variables (cf. Table 4.2). So, we believe, in particular, that the timings which are roughly constant for simulations with $P = 84$ and $P = 210$ are due to the fact that both cases were computed with exactly the same value of N.

Next, in Table 4.6 we present our results for the unsteady diffusion constrained model as discussed in Section 4.3. Here, for $\alpha \in \{0, 1\}$ and different values of β, we present the outputs of our simulations showing the total CPU times and the total number of iterations from preconditioned TT-MINRES. Also, DoF=$J \cdot P \cdot n_t$ is the size of each of the 9 block matrices in KKT matrix $\mathcal{A}$; that is, $\mathcal{A}$ is of dimension 3DoF. Here, we have done the computations with $J = 1985$ ($h = \frac{1}{2^5}$), $P = 56$ ($N = 5, n = 3$), $\sigma_a = 0.1$, and various n_t.

As in the steady-state case, we see from Table 4.6 that TT-MINRES, when used together with our mean-based preconditioner as given by (4.77) is quite robust, but in general yields fewer iterations for $\alpha = 0$ than for $\alpha = 1$. We remark here that we used a smaller tolerance $tol = 10^{-3}$ in the unsteady case because MATLAB took a lot more time due to the rapid growth of TT-ranks. Although not reported here, we also got robust two-digit TT-MINRES iterations when we used $tol = 10^{-5}$; these iterations were, as expected,

Table 4.6: Simulation results using the mean-based preconditioner $\widehat{\mathcal{Z}}_0$ in (4.77) with the model with time-dependent diffusion constraint for selected parameter values and degrees of freedom.

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

even better with the preconditioned Richardson algorithm.

In a nutshell, we have derived and implemented in this chapter robust block-diagonal Schur complement-based preconditioners together with low-rank MINRES for the solution of linear systems arising from the SGFEM discretization of optimal control problems constrained by either stationary or unsteady diffusion equations with random inputs.

We proceed next to Chapter 5 to discuss also efficient low-rank solvers for a more challenging problem, namely, the unsteady Stokes-Brinkman optimal control problem with uncertain inputs.

Chapter 5

Unsteady Stokes-Brinkman optimal control problem with uncertain inputs

In this chapter, we study efficient low-rank tensor-based iterative solvers for an unsteady Stokes-Brinkman SOCP. 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 medium. In biomedical engineering it could be used to model, for instance, the reduction of vorticity of blood flow through intra cranial aneurysms [120]. 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 in some complete probability space. This could be interpreted as a scenario where the volume of blood moving through intra cranial aneurysms is uncertain due to measurement error in ν or probably some other factors.

As aptly pointed out in a related study in the framework of a deterministic control problem [85], efficient procedures for the numerical solution of the resulting Stokes-Brinkman SOCP 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 thesis, we discuss a low-rank tensor-based technique for solving high dimensional tensor product linear systems resulting from the SGFEM discretization of a Stokes-Brinkman SOCP. The materials in this chapter are based on [11]. Our point of departure is the deterministic Brinkman model.

5.1 Deterministic Brinkman model

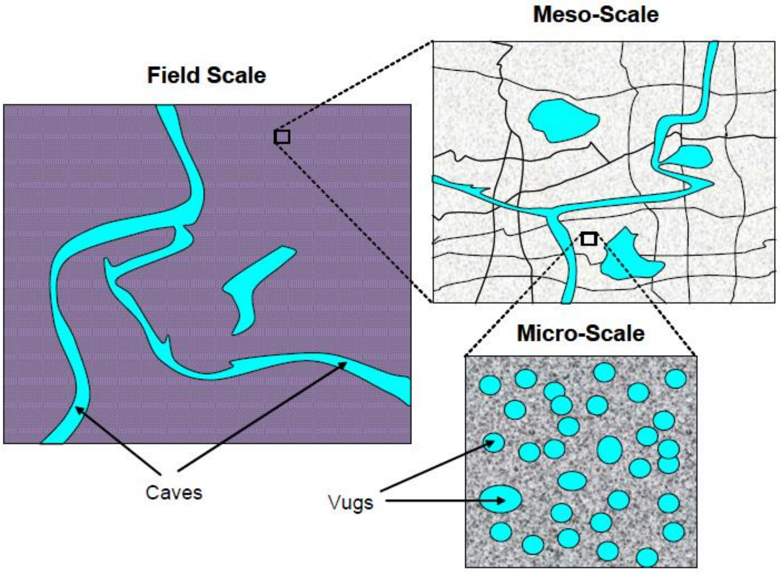
Suppose now that the spatial domain $\mathcal{D}$ consists of two parts, namely, a porous medium $\mathcal{D}_p$ and a viscous flow medium $\mathcal{D}_s$. That is, $\mathcal{D} = \mathcal{D}_p \cup \mathcal{D}_s$. The generalized unsteady Brinkman problem reads

$$\begin{cases} \dfrac{\partial v(t,\mathbf{x})}{\partial t} - \nu\Delta v(t,\mathbf{x}) + \varrho(\mathbf{x})v(t,\mathbf{x}) + \nabla p(t,\mathbf{x}) = u(t,\mathbf{x}), & \text{in } (0,T] \times \mathcal{D}, \\ -\nabla \cdot v(t,\mathbf{x}) = 0, & \text{on } (0,T] \times \mathcal{D}, \\ v(t,\mathbf{x}) = h(t,\mathbf{x}), & \text{on } [0,T] \times \partial\mathcal{D}, \\ v(0,\mathbf{x}) = v_0(\mathbf{x}), & \text{in } \mathcal{D}, \end{cases} \tag{5.1}$$

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, ϱ is the *inverse permeability tensor* of the medium. We assume here that $\varrho \in L^2(\mathcal{D}) \cap L^\infty(\mathcal{D})$ and that the source term $u \in L^2(\mathcal{D})$. The challenge of this problem is that the coefficient ϱ 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. This feature naturally arises, for instance, in fractured vuggy karst reservoirs where the model (5.1) is also widely applied; see e.g., [108] and the references therein. As illustrated in Fig. 5.1, such oil reservoirs are characterized by the presence of vugs and caves at multiple scales. The medium can be described, at each individual scale, as an ensemble of porous media with well defined properties (porosity and permeability), and a free flow region where the fluid (oil, water, gas) meets no resistance from the surrounding rock.

Popov et. al in [108] point out that the computational difficulty in such reservoirs is

Figure 5.1: Conceptual model of a fractured vuggy reservoir at multiple scales [108].



essentially attributed to the co-existence of porous and free flow regions, typically at several scales. Indeed, the presence of individual voids such as vugs and caves in a surrounding porous medium can significantly alter the permeability of the medium. Another inherent feature of such reservoirs is the fact that fractures and long range caves typically form various types of connected networks which change the effective permeability of the media by several orders of magnitude. Moreover, lack of precise knowledge of the exact position of the interface between the porous media (rock) and the and vugs/caves often poses yet another difficult problem to tackle.

For a mixed finite element discretization of the Brinkman problem [86, 120, 133, 141] in the primal variables v and p, let $V_h \subset L^2(0,T;H_0^1(\mathcal{D}))$ 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 [120]) such that $V_h = \text{span}\{\phi_1,\ldots,\phi_{J_v}\}$ and $W_h = \text{span}\{\tilde{\varphi}_1,\ldots,\tilde{\varphi}_{J_p}\}$. 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:

$$\begin{cases} \dfrac{Mv_i - Mv_{i-1}}{\tau} + (\nu K + M_\varrho)v_i + B^T p_i = Mu_i + g_i, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad Bv_i = 0, \end{cases} \tag{5.2}$$

where the mass matrix M is as defined in (2.35), $B = \left[-\int_{\mathcal{D}} \tilde{\varphi}_k \nabla \cdot \phi_{k'}\right]$ is the discrete divergence operator, $K = \left[\int_{\mathcal{D}} \nabla\phi_k : \nabla\phi_{k'}\right]$ is a matrix representing the vector Laplacian operator, and $M_\varrho = \left[\int_{\mathcal{D}} \varrho\phi_k\phi_{k'}\right]$ is the matrix associated with the term which involves the inverse permeability coefficient $\varrho(\mathbf{x})$, and τ is the size of the time step.

Remark 5.1. *In the special case where $M_\varrho = 0$ in (5.2), we get the unsteady Stokes problem.*

5.1.1 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. As noted before, instead of guessing a value, we can model ν as a random variable defined on the complete probability space $(\Omega, \mathfrak{F}, \mathbb{P})$. 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 thesis consists in minimizing the cost functional of tracking-type

$$\mathcal{J} = \frac{1}{2}||v - \bar{v}||^2_{L^2(0,T;\mathcal{D})\otimes L^2(\Omega)} + \frac{\alpha}{2}||\mathrm{std}(v)||^2_{L^2(0,T;\mathcal{D})} + \frac{\beta}{2}||u||^2_{L^2(0,T;\mathcal{D})\otimes L^2(\Omega)} \tag{5.3}$$

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

$$\begin{cases} \dfrac{\partial v(t,\mathbf{x},\omega)}{\partial t} - \nu(\omega)\Delta v(t,\mathbf{x},\omega) + \varrho(\mathbf{x})v(t,\mathbf{x},\omega) + \nabla p(t,\mathbf{x},\omega) = u(t,\mathbf{x},\omega), & \text{in } \mathcal{Q}_T \times \Omega, \\ -\nabla\cdot v(t,\mathbf{x},\omega) = 0, & \text{on } \mathcal{Q}_T \times \Omega, \\ v(t,\mathbf{x},\omega) = h(t,\mathbf{x},\omega), & \text{on } \mathcal{Q}'_T \times \Omega, \\ v(0,\mathbf{x},\omega) = v_0(\mathbf{x},\omega), & \text{in } \mathcal{D} \times \Omega, \end{cases}$$

where $\mathcal{Q}_T := (0,T] \times \mathcal{D}$ and $\mathcal{Q}'_T := [0,T] \times \partial\mathcal{D}$. Here, $v, \bar{v}, p : \mathcal{D} \times \mathcal{T} \times \Omega \to \mathbb{R}$ are random fields [13] representing the state (velocity), the target (or the desired state) and the pressure. The viscosity ν in the state equations is modeled as[1]

$$\nu(\omega) = \nu_0 + \nu_1\xi(\omega), \qquad \nu_0, \nu_1 \in \mathbb{R}^+, \tag{5.4}$$

[1] There are, of course, other ways of modeling ν, see, e.g., [84, Chapter 7]

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

$$u, \bar{v} \in L^2(0,T;\mathcal{D}) \otimes L^2(\Omega), \tag{5.5}$$

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

$$\mathbb{P}\left(\omega \in \Omega : \nu(\omega) \in [\nu_{\min}, \nu_{\max}]\right) = 1. \tag{5.6}$$

5.1.2 A fully discrete problem

As in Chapter 4, we will herein adopt the DTO strategy, together with SGFEM, for discretizing the Stokes-Brinkman problem. More specifically, we assume that p, u, v, and $\bar{v}$ admit the following respective representations:

$$\begin{cases} p(t,\mathbf{x},\omega) = \displaystyle\sum_{k=1}^{J_p}\sum_{j=0}^{P-1} p_{kj}(t)\tilde{\varphi}_k(\mathbf{x})\psi_j(\xi(\omega)), \\ u(t,\mathbf{x},\omega) = \displaystyle\sum_{k=1}^{J_v}\sum_{j=0}^{P-1} u_{kj}(t)\phi_k(\mathbf{x})\psi_j(\xi(\omega)), \\ v(t,\mathbf{x},\omega) = \displaystyle\sum_{k=1}^{J_v}\sum_{j=0}^{P-1} v_{kj}(t)\phi_k(\mathbf{x})\psi_j(\xi(\omega)), \\ \bar{v}(t,\mathbf{x},\omega) = \displaystyle\sum_{k=1}^{J_v}\sum_{j=0}^{P-1} \bar{v}_{kj}(t)\phi_k(\mathbf{x})\psi_j(\xi(\omega)), \end{cases} \tag{5.7}$$

where $\{\psi_j\}_{j=0}^{P-1}$ are univariate orthogonal polynomials of order at most P satisfying (2.21).

We apply to the cost functional (5.3) the trapezoidal rule for temporal discretization, and the *mini* finite elements [120], together with Legendre polynomial chaos in the SGFEM for spatial and stochastic discretizations [110], to get the following discrete cost functional

$$\mathcal{J}(\mathbf{y},\mathbf{u}) := \frac{\tau}{2}(\mathbf{y}-\bar{\mathbf{y}})^T \boldsymbol{M}_a(\mathbf{y}-\bar{\mathbf{y}}) + \frac{\tau\alpha}{2}\mathbf{y}^T\boldsymbol{M}_b\mathbf{y} + \frac{\tau\beta}{2}\mathbf{u}^T\boldsymbol{M}_2\mathbf{u}, \tag{5.8}$$

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

$$\begin{cases} \boldsymbol{M}_a = \text{blkdiag}\left(\frac{1}{2}\mathcal{M}, 0, \mathcal{M}, 0, \ldots, \mathcal{M}, 0, \frac{1}{2}\mathcal{M}, 0\right), & \mathcal{M} := G_0 \otimes M, \\ \boldsymbol{M}_b = \text{blkdiag}\left(\frac{1}{2}\mathcal{M}_t, 0, \mathcal{M}_t, 0, \ldots, \mathcal{M}_t, 0, \frac{1}{2}\mathcal{M}_t, 0\right), & \mathcal{M}_t := H_0 \otimes M, \\ \boldsymbol{M}_2 = \text{blkdiag}\left(\frac{1}{2}\mathcal{M}, \mathcal{M}, \ldots, \mathcal{M}, \frac{1}{2}\mathcal{M}\right), \end{cases} \tag{5.9}$$

where G_0, H_0 and M are as given by (2.28), (2.35) and (4.6), respectively.

For an all-at-once discretization of the state equation, as in Section 4.3, we use the implicit Euler method, together with SGFEM, to get

$$\boldsymbol{K}\mathbf{y} - \boldsymbol{N}\mathbf{u} = \mathbf{g}, \tag{5.10}$$

where

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

with

$$\widehat{\mathcal{N}} = G_0 \otimes N, \quad N = \begin{bmatrix} M \\ 0 \end{bmatrix}, \quad \bar{\mathcal{M}} = G_0 \otimes \tau^{-1}\bar{M}, \quad \bar{M} = \begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix}, \tag{5.11}$$

and, in the notation of [111],

$$\bar{\mathcal{L}} = \begin{bmatrix} \mathcal{A} & \mathcal{B}^T \\ \mathcal{B} & 0 \end{bmatrix} \tag{5.12}$$

represents an instance of the time-dependent Brinkman problem with

$$\mathcal{A} = G_0 \otimes A + G_1 \otimes \nu_1 K, \quad A = \tau^{-1}M + \nu_0 K + M_\varrho, \qquad \mathcal{B} = G_0 \otimes B, \tag{5.13}$$

and $G_1(j, j') = \left\langle \xi\psi_j(\xi)\psi_{j'}(\xi)\right\rangle$. Note that since we are using Legendre polynomials for the SGFEM discretization, G_0 is a diagonal matrix whereas G_1 is a tridiagonal matrix with zeros on the main diagonal (see e.g., [110, 111]). This implies that the matrices $\mathcal{A}$ and $\mathcal{B}$ in (5.13) are, respectively, block-tridiagonal and block-diagonal. Furthermore, the matrices K, M and M_ϱ are positive definite; however, $\bar{\mathcal{L}}$ is an indefinite block sparse matrix with sparse blocks.

As before, it will be convenient to work with the Kronecker product representations of the system matrices. To this end, observe that

$$\boldsymbol{K} = I_{n_t} \otimes G_0 \otimes \begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} + I_{n_t} \otimes G_1 \otimes \begin{bmatrix} \nu_1 K & 0 \\ 0 & 0 \end{bmatrix} + C \otimes G_0 \otimes \begin{bmatrix} \tau^{-1} M & 0 \\ 0 & 0 \end{bmatrix}, \tag{5.14}$$

where the matrix C is as defined in (4.59), and

$$\boldsymbol{N} = I_{n_t} \otimes G_0 \otimes N. \tag{5.15}$$

The structure of the right-hand side $\mathbf{g}$ in (5.10) 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 (5.8) and (5.10) that the discrete Lagrangian functional of the SOCP is given by

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

where $\mathbf{f}$ is the Lagrange multiplier. Hence, as before, applying the first order conditions to the Lagrangian $\mathfrak{L}$ yields, respectively, the adjoint equation, the gradient equation and the state equation:

$$\begin{aligned}
\mathfrak{L}_{\mathbf{y}}(\mathbf{y}, \mathbf{u}, \mathbf{f}) = 0 \ \Rightarrow \quad & \tau(\boldsymbol{M}_a + \alpha \boldsymbol{M}_b)\mathbf{y} - \boldsymbol{K}\mathbf{f} = \tau \boldsymbol{M}_a \bar{\mathbf{y}}, \\
\mathfrak{L}_{\mathbf{u}}(\mathbf{y}, \mathbf{u}, \mathbf{f}) = 0 \ \Rightarrow \quad & \beta\tau \boldsymbol{M}_2 \mathbf{u} + \boldsymbol{N}\mathbf{f} = \mathbf{0}, \\
\mathfrak{L}_{\mathbf{f}}(\mathbf{y}, \mathbf{u}, \mathbf{f}) = 0 \ \Rightarrow \quad & -\boldsymbol{K}\mathbf{y} + \boldsymbol{N}\mathbf{u} = \mathbf{g},
\end{aligned}$$

or, alternatively, the following KKT system

$$\underbrace{\begin{bmatrix} \tau \boldsymbol{M}_1 & 0 & -\boldsymbol{K}^T \\ 0 & \beta\tau \boldsymbol{M}_2 & \boldsymbol{N}^T \\ -\boldsymbol{K} & \boldsymbol{N} & 0 \end{bmatrix}}_{:=\mathfrak{A}} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \mathbf{f} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{0} \\ \mathbf{g} \end{bmatrix}, \tag{5.16}$$

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

$$\boldsymbol{M}_1 = \boldsymbol{M}_a + \alpha \boldsymbol{M}_b = D \otimes G_\alpha \otimes \bar{M}, \qquad \boldsymbol{M}_2 = D \otimes \mathcal{M} = D \otimes G_0 \otimes M. \tag{5.17}$$

Moreover, D and G_α are as given, respectively, by (4.57) and (4.10). We note here that if the desired state is also static and deterministic, then one gets $\bar{\mathbf{y}} = \mathbf{e} \otimes \mathbf{e}_1 \otimes \begin{bmatrix} \bar{\mathbf{v}} \\ 0 \end{bmatrix}$.

Remark 5.2. *Note that, with the exception of block (2,2), the third matrices in the Kronecker representations of all the blocks in (5.16) are themselves also block matrices.*

5.2 Preconditioning Stokes-Brinkman KKT system

As in the case of the diffusion equation in Chapter 4, the KKT coefficient matrix $\mathfrak{A}$ in (5.16) is usually ill-conditioned and thus requires a suitable preconditioner to solve (5.16) efficiently. A block-diagonal preconditioner, discussed in the framework of the deterministic unsteady Stokes control problem [126], is written in the form $\boldsymbol{P}_1 = \text{blockdiag}(\tilde{\boldsymbol{M}}_1, \beta \boldsymbol{M}_2, \tilde{\boldsymbol{S}}_1)$, where $\tilde{\boldsymbol{S}}_1 = \frac{1}{\tau}(\boldsymbol{K} + \boldsymbol{M}_s) \tilde{\boldsymbol{M}}_1^{-1} \left(\boldsymbol{K}^T + \boldsymbol{M}_s\right)^T$ is the approximate Schur complement, and $\tilde{\boldsymbol{M}}_1$ is some perturbation to $\boldsymbol{M}_1$, since $\boldsymbol{M}_1$ is rank-deficient. Here, the matrix $\boldsymbol{M}_s$ is again determined via a matching argument. In particular, in [126] the authors 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 is not obvious how to generalize it to the case in which $\boldsymbol{M}_1$ 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 5.2.1 to propose another preconditioner which circumvents this deficiency and yields faster convergence even with the original sparse $\boldsymbol{M}_1$.

5.2.1 A block-triangular preconditioner

We begin by replacing the KKT coefficient matrix $\mathfrak{A}$ in (5.16) with the matrix $\tilde{\mathfrak{A}}$ 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 the third block columns of $\mathfrak{A}$ in the product $\mathfrak{A}\boldsymbol{\rho}$; it swaps the first and the third block 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

$$\begin{bmatrix} \boldsymbol{\Phi} & \boldsymbol{\Upsilon} \\ \boldsymbol{\Psi} & -\boldsymbol{K} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & 0 \\ \boldsymbol{\Psi}\boldsymbol{\Phi}^{-1} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Phi} & \boldsymbol{\Upsilon} \\ 0 & -\boldsymbol{S}_2 \end{bmatrix}, \tag{5.18}$$

where

$$\boldsymbol{\Phi}^{-1} = \begin{bmatrix} -\boldsymbol{K}^{-T} & 0 \\ \frac{1}{\tau\beta}\boldsymbol{M}_2^{-1}\boldsymbol{N}^T\boldsymbol{K}^{-T} & \frac{1}{\tau\beta}\boldsymbol{M}_2^{-1} \end{bmatrix}, \tag{5.19}$$

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 (5.11), (5.15) and (5.17), we obtain

$$\boldsymbol{N}\boldsymbol{M}_2^{-1}\boldsymbol{N}^T = D^{-1} \otimes G_0 \otimes \bar{M} = D^{-1} \otimes \begin{bmatrix} \tau\mathcal{M} & 0 \\ 0 & 0 \end{bmatrix} =: \boldsymbol{M}_{-1}. \tag{5.20}$$

Therefore,

$$\boldsymbol{S}_2 = \boldsymbol{K} + \boldsymbol{\Psi}\boldsymbol{\Phi}^{-1}\boldsymbol{\Upsilon} = \boldsymbol{K} + \frac{1}{\beta}\boldsymbol{M}_{-1}\boldsymbol{K}^{-T}\boldsymbol{M}_1. \tag{5.21}$$

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

$$\boldsymbol{P}_D = \begin{bmatrix} \boldsymbol{\Phi} & \boldsymbol{\Upsilon} \\ 0 & -\boldsymbol{S}_2 \end{bmatrix}. \tag{5.22}$$

It follows from (5.18) that

$$\tilde{\mathfrak{A}}\boldsymbol{P}_D^{-1} = \mathfrak{A}\boldsymbol{\rho}\boldsymbol{P}_D^{-1} = \mathfrak{A}\boldsymbol{P}_2^{-1} = \begin{bmatrix} \boldsymbol{I} & 0 \\ \boldsymbol{\Psi}\boldsymbol{\Phi}^{-1} & \boldsymbol{I} \end{bmatrix}, \tag{5.23}$$

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

$$\boldsymbol{P}_2^{-1} = \boldsymbol{\rho}\boldsymbol{P}_D^{-1} = \begin{bmatrix} 0 & 0 & -\boldsymbol{S}_2^{-1} \\ \frac{1}{\beta\tau}\boldsymbol{M}_2^{-1}\boldsymbol{N}^T\boldsymbol{K}^{-T} & \frac{1}{\beta\tau}\boldsymbol{M}_2^{-1} & \frac{1}{\beta}\boldsymbol{M}_2^{-1}\boldsymbol{N}^T\boldsymbol{K}^{-T}\boldsymbol{M}_1\boldsymbol{S}_2^{-1} \\ -\boldsymbol{K}^{-T} & 0 & -\boldsymbol{K}^{-T}\tau\boldsymbol{M}_1\boldsymbol{S}_2^{-1} \end{bmatrix}. \tag{5.24}$$

It can be noticed that (5.23) immediately implies $(\mathfrak{A}\boldsymbol{P}_2^{-1} - I)^2 = 0$; hence, such Krylov solvers as the GMRES method will converge in two iterations if $\boldsymbol{P}_2^{-1}$ is applied exactly, see e.g. [39, Section 9.1].

The seemingly complicated structure of (5.24) notwithstanding, matrix-vector products 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

$$\begin{cases} \mathbf{x}_1 = -\boldsymbol{S}_2^{-1}\mathbf{y}_3 \\ \mathbf{x}_3 = -\boldsymbol{K}^{-T}(\mathbf{y}_1 - \tau \boldsymbol{M}_1 \mathbf{x}_1) \\ \mathbf{x}_2 = \tau^{-1}\beta^{-1}\boldsymbol{M}_2^{-1}(\mathbf{y}_2 - \boldsymbol{N}^T\mathbf{x}_3). \end{cases} \tag{5.25}$$

Next, following the preconditioning strategy which we employed in Chapter 4, we approximate the Schur complement $\boldsymbol{S}_2$ in (5.21) with a matrix of the form

$$\begin{aligned} \tilde{\boldsymbol{S}}_2 &= (\boldsymbol{K} + \boldsymbol{M}_l)\,\boldsymbol{K}^{-T}\left(\boldsymbol{K}^T + \boldsymbol{M}_r\right). \\ &= \boldsymbol{K} + \boldsymbol{M}_l\boldsymbol{K}^{-T}\boldsymbol{M}_r + \boldsymbol{M}_l + \boldsymbol{K}\boldsymbol{K}^{-T}\boldsymbol{M}_r, \end{aligned} \tag{5.26}$$

where $\boldsymbol{M}_l$ and $\boldsymbol{M}_r$ are again determined using the 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 (5.26) and match the first and second terms with those in (5.21) 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 (5.17) and (5.20). Hence, we have

$$\tilde{\boldsymbol{S}}_2 = \left(\boldsymbol{K} + \frac{1}{\sqrt{\beta}}\boldsymbol{M}_{-1}\right)\boldsymbol{K}^{-T}\left(\boldsymbol{K}^T + \frac{1}{\sqrt{\beta}}\boldsymbol{M}_1\right). \tag{5.27}$$

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 (5.14), and the fourth corresponding to $\boldsymbol{M}_{-1}$ in (5.20) and $\boldsymbol{M}_1$ in (5.17), respectively. However, our ultimate goal is to apply $\tilde{\boldsymbol{S}}_2^{-1}$, where solving a linear system with exact factors is indeed a very difficult computational task. As a result, we instead approximate them by one Kronecker-product term: we approximate $\boldsymbol{K}$ by the first term from (5.14), 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,

$$\left(\boldsymbol{K} + \frac{1}{\sqrt{\beta}}\boldsymbol{M}_{\mathtt{i}}\right) \approx I_{n_t} \otimes G_0 \otimes \begin{bmatrix} A + \eta_{\mathtt{i}} M & B^T \\ B & 0 \end{bmatrix}, \tag{5.28}$$

where $\mathtt{i} \in \{-1, 1\}$, and $\eta_{-1} = 1/\sqrt{\beta}$, $\eta_1 = (1+\alpha)/\sqrt{\beta}$. Inside alternating tensor methods (cf. Section 5.3.4), the matrix $I_{n_t} \otimes G_0$ will be further reduced, but the concept of the one-term preconditioner remains the same.

5.2.2 Preconditioning the forward Stokes-Brinkman problem

In linear systems of the form (5.28), 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 preconditioned Richardson iteration (c.f. Algorithm 4.3), together with the block-triangular preconditioner

$$P_{SB} = \begin{bmatrix} \tilde{A} & 0 \\ B & -S_0 \end{bmatrix}, \tag{5.29}$$

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

$$P_{SB}^{-1} = \begin{bmatrix} \tilde{A}^{-1} & 0 \\ S_0^{-1}B\tilde{A}^{-1} & -S_0^{-1} \end{bmatrix}. \tag{5.30}$$

In what follows, we derive the approximation to the blocks of P_{SB}^{-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 [39, 126], the pressure mass matrix is a very effective approximation for S_0 in the case of stationary Stokes equations. However, as we are considering an unsteady Stokes-Brinkman constraint, this does not apply since $\tilde{A}$ has an entirely different structure. Thus, following [39, Chapter 9], 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 9 of [39]) defined by

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

where $\mathbb{L} = (\tau^{-1} + \eta)I + \Delta + \varrho$ and $\mathbb{L}_p = (\tau^{-1} + \eta)I_p + \Delta_p + \varrho_p$ is defined similarly but on the pressure space. As was noted in [126], 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

$$\mathfrak{E}_h = (M^{-1}\tilde{A})M^{-1}B^T - M^{-1}B^T(M_p^{-1}\tilde{A}_p) \approx 0, \tag{5.31}$$

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

$$\tilde{A}_p \quad = \quad \nu_0 K_p + M_{\varrho_p} + (\tau^{-1} + \eta)M_p. \tag{5.32}$$

The smallness $\mathfrak{E}_h \approx 0$ should be understood in the sense that the norm of the commutator is much smaller than the norm of either term in (5.31). Next, we pre-multiply (5.31) by $B\tilde{A}^{-1}M$ and post-multiply it by $\tilde{A}_p^{-1}M_p$ to obtain

$$BM^{-1}B^T\tilde{A}_p^{-1}M_p - B\tilde{A}^{-1}B^T \approx 0, \tag{5.33}$$

or, equivalently (with $\approx$ meaning again the proximity in the norm),

$$S_0 \approx BM^{-1}B^T\tilde{A}_p^{-1}M_p. \tag{5.34}$$

Now, note that the matrix on the right hand side of (5.34) 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 [38]; 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

$$c_0 \leq \frac{\langle BM^{-1}B^T\mathbf{v}, \mathbf{v}\rangle}{\langle K_p\mathbf{v}, \mathbf{v}\rangle} \leq c_1, \qquad \forall \mathbf{v} \in \mathbb{R}^{J_p}, \ \mathbf{v} \neq \mathbf{0}.$$

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 (5.34). Hence, from (5.34), we obtain

$$S_0 \approx K_p\tilde{A}_p^{-1}M_p, \tag{5.35}$$

and from (5.32) and (5.35), we have

$$S_0^{-1} \approx M_p^{-1}\left(\nu_0 K_p + M_{k_p} + (\tau^{-1} + \eta) M_p\right) K_p^{-1}. \tag{5.36}$$

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 9 of [39], 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., [19]). Afterwards, we use the AMG package provided by [21].

5.2.3 Spectral analysis

As before, to measure how well the exact Schur complement is represented by its approximation, 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 α and β.

Theorem 5.3. *If the system matrix $\boldsymbol{K}$ in* (5.14) *and its velocity block are invertible, then there exist constants C_1 and C_2 such that*

$$\begin{aligned} &\operatorname{cond}(\boldsymbol{S}_2^{-1}\tilde{\boldsymbol{S}}_2) \le (1 + C_1\beta^{1/2}) \quad \textit{for } \beta \textit{ sufficiently small,} \\ &\operatorname{cond}(\boldsymbol{S}_2^{-1}\tilde{\boldsymbol{S}}_2) \le (1 + C_2\beta^{-1/2}) \quad \textit{for } \beta \textit{ sufficiently large,} \end{aligned} \tag{5.37}$$

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

$$\boldsymbol{B} = I_{n_t} \otimes G_0 \otimes B, \tag{5.38}$$

$$\boldsymbol{A} = I_{n_t} \otimes G_0 \otimes (\nu_0 K + M_\varrho + \tau^{-1} M) + I_{n_t} \otimes G_1 \otimes \nu_1 K + C \otimes G_0 \otimes \tau^{-1} M, \tag{5.39}$$

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

$$\boldsymbol{K}^{-T} = \begin{bmatrix} \boldsymbol{A}^{-T} - \boldsymbol{A}^{-T}\boldsymbol{B}^T\boldsymbol{S}^{-1}\boldsymbol{B}\boldsymbol{A}^{-T} & \boldsymbol{A}^{-T}\boldsymbol{B}^T\boldsymbol{S}^{-1} \\ \boldsymbol{S}^{-1}\boldsymbol{B}\boldsymbol{A}^{-T} & -\boldsymbol{S}^{-1} \end{bmatrix}, \tag{5.40}$$

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. [17]. Notice that $\boldsymbol{P}_K = \boldsymbol{P}_K^2$; that is, the matrix $\boldsymbol{P}_K$ is a projector and, from (5.38) and (5.39), it is also β-independent. From (5.17), (5.20) and (5.40), we have that

$$\beta^{-1}\boldsymbol{M}_{-1}\boldsymbol{K}^{-T}\boldsymbol{M}_1 = \begin{bmatrix} \boldsymbol{M}_\star & 0 \\ 0 & 0 \end{bmatrix}, \tag{5.41}$$

where

$$\boldsymbol{M}_\star = \beta^{-1}\mathtt{M}_{-1}\boldsymbol{K}_{11}\mathtt{M}_1, \tag{5.42}$$

$\mathtt{M}_{-1} = D^{-1} \otimes G_0 \otimes M$ 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 (5.20) and (5.17), respectively, and $\boldsymbol{K}_{11} = \boldsymbol{A}^{-T}(I - \boldsymbol{P}_K)$ denotes the (1,1) block of $\boldsymbol{K}^{-T}$. Thus, using (5.42), (5.41) and (5.21), we get

$$\boldsymbol{S}_2 = \boldsymbol{K} + \beta^{-1}\boldsymbol{M}_{-1}\boldsymbol{K}^{-T}\boldsymbol{M}_1 = \begin{bmatrix} \boldsymbol{A}_\star & \boldsymbol{B}^T \\ \boldsymbol{B} & 0 \end{bmatrix}, \tag{5.43}$$

where

$$\boldsymbol{A}_\star = \boldsymbol{A} + \boldsymbol{M}_\star. \tag{5.44}$$

Next, observe from (5.26) that

$$\tilde{\boldsymbol{S}}_2 - \boldsymbol{S}_2 = \beta^{-1/2}(\boldsymbol{M}_{-1} + \boldsymbol{K}\boldsymbol{K}^{-T}\boldsymbol{M}_1) = \begin{bmatrix} \boldsymbol{U} & 0 \\ 0 & 0 \end{bmatrix}, \tag{5.45}$$

where

$$\boldsymbol{U} = \beta^{-1/2} \underbrace{\left(\mathtt{M}_{-1} + \left(\boldsymbol{A}\boldsymbol{A}^{-T}(I - \boldsymbol{P}_K) + \boldsymbol{P}_K\right)\mathtt{M}_1\right)}_{:=U_1}. \tag{5.46}$$

Notice from (5.46) that U_1 is also β-independent. Now, using (5.40), (5.43) and (5.45), we have

$$\begin{aligned} \boldsymbol{S}_2^{-1}\tilde{\boldsymbol{S}}_2 &= \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} + \begin{bmatrix} \boldsymbol{A}_\star & \boldsymbol{B}^T \\ \boldsymbol{B} & 0 \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{U} & 0 \\ 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} I + \boldsymbol{A}_\star^{-1}(I - \boldsymbol{P}_\star)\boldsymbol{U} & 0 \\ \boldsymbol{S}_\star^{-1}\boldsymbol{B}\boldsymbol{A}_\star^{-1}\boldsymbol{U} & I \end{bmatrix}, \end{aligned} \tag{5.47}$$

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

$$\boldsymbol{P}_\star = \boldsymbol{B}^T\boldsymbol{S}_\star^{-1}\boldsymbol{B}\boldsymbol{A}_\star^{-1} \tag{5.48}$$

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 (5.37), suppose now that β is sufficiently small. Then, from (5.42) and (5.44), the norm of $\boldsymbol{M}_\star$ is much larger than the norm of $\boldsymbol{A}$, $\|\boldsymbol{M}_\star\| = \|\beta^{-1}\mathtt{M}_{-1}\boldsymbol{K}_{11}\mathtt{M}_1\| \gg \|\boldsymbol{A}\|$, since $\boldsymbol{A}$ is independent of β. That is, we take β much less than the β-independent bound $\|\mathtt{M}_{-1}\boldsymbol{K}_{11}\mathtt{M}_1\|/\|\boldsymbol{A}\|$. 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

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

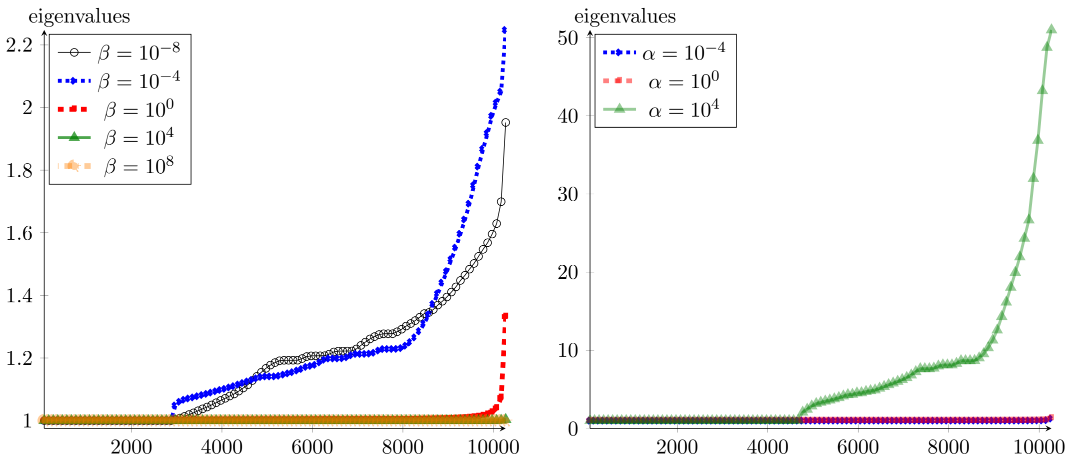
from which, together with (5.48), we deduce that the norm of the projector $\boldsymbol{P}_\star$ is asymptotically β-independent. Finally, from (5.46), we have $\|\boldsymbol{U}\| = \|\beta^{-1/2}U_1\| := \tilde{C}\beta^{-1/2}$, and $\|\boldsymbol{A}_\star^{-1}(I - \boldsymbol{P}_\star)\boldsymbol{U}\| \leq C_1\beta^{1/2}$. 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 5.2. 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 in particular that the eigenvalues approach the maximum 2.2 for intermediate $\beta = 10^{-4}$, but remain closer to 1 for both larger and smaller β. This is also depicted in the experiment in Section 5.4.4: $\beta = 10^{-4}$ is the kink point for the error, and the maximum point for the CPU time.

On the other hand, Figure 5.2 (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

Figure 5.2: 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.



increases in Section 5.4.5. 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.

5.3 A tensor train solver

5.3.1 Alternating iterative methods

Notwithstanding the TT truncation, as we noted before, 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, such as the Lyapunov equations, one can employ ADI [135] or tensor product Krylov methods [77]. For more general problems we have to employ more general *alternating methods* [57, 118].

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 Alternating Least Squares (ALS) method which is used to minimize the misfit of a tensor by a low-rank tensor model, see the surveys [49, 74]. This was later extended to the solution of linear systems [57, 99]. In quantum physics, a powerful realization of the alternation idea is the Density Matrix Renormalization Group (DMRG) algorithm [138], which is mainly used for eigenvalue problems, but also for linear systems [63]. 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 [139] uses the surrogate of the Krylov vector and the *Alternating Minimal Energy* (AMEn) method [32] uses the actual residual. The latter was later adopted for eigenvalue problems as well [60, 76].

Consider a linear system $\boldsymbol{A}\mathbf{y} = \mathbf{g}$, where $\mathbf{y}$ is sought with some initial guess in the TT format; that is,

$$\mathbf{y}(i,j,k) = \sum_{s_1,s_2=1}^{r_1,r_2} \mathbf{y}^{(1)}_{s_1}(i)\mathbf{y}^{(2)}_{s_1,s_2}(j)\mathbf{y}^{(3)}_{s_2}(k),$$

or, equivalently,

$$\mathbf{y} = \sum_{s_1,s_2=1}^{r_1,r_2} \mathbf{y}_{s_1}^{(1)} \otimes (\mathbf{y}_{s_1,s_2}^{(2)})^T \otimes (\mathbf{y}_{s_2}^{(3)})^T,$$

from Proposition 4.17. The ALS method reduces this system to the elements of a chosen TT block $\mathbf{y}^{(m)}$ in the course of iterations $m = 1,2,3$. Notice that the TT format is linear w.r.t. each particular TT block, i.e. we can write

$$\mathbf{y} = \boldsymbol{Y}_1 \mathbf{y}^{(1)} = \boldsymbol{Y}_2 \mathbf{y}^{(2)} = \boldsymbol{Y}_3 \mathbf{y}^{(3)},$$

where the *frame* matrices $\boldsymbol{Y}_m,\ m = 1,2,3$, are constructed as follows:

$$\begin{aligned}
\boldsymbol{Y}_1 &= I_{n_t} \otimes \sum_{s_2=1}^{r_2} \left(\mathbf{y}_{s_2}^{(2)}\right)^T \otimes \left(\mathbf{y}_{s_2}^{(3)}\right)^T \in \mathbb{R}^{n_t PJ \times n_t r_1}, \\
\boldsymbol{Y}_2 &= \mathbf{y}^{(1)} \otimes I_P \otimes \left(\mathbf{y}^{(3)}\right)^T \in \mathbb{R}^{n_t PJ \times r_1 P r_2}, \\
\boldsymbol{Y}_3 &= \sum_{s_1=1}^{r_1} \mathbf{y}_{s_1}^{(1)} \otimes \mathbf{y}_{s_1}^{(2)} \otimes I_J \in \mathbb{R}^{n_t PJ \times r_2 J},
\end{aligned} \tag{5.49}$$

where[2]

$$\mathbf{y}^{(1)} \in \mathbb{R}^{n_t \times r_1}, \ \ \mathbf{y}^{(2)} \in \mathbb{R}^{r_1 \times P \times r_2}, \ \text{ and } \ \mathbf{y}^{(3)} \in \mathbb{R}^{r_1 \times J}$$

are as defined in (4.81), and hence, $\mathbf{y}_{s_1}^{(1)} \in \mathbb{R}^{n_t \times 1}$, $\mathbf{y}_{s_1}^{(2)} \in \mathbb{R}^{P \times r_2}$, $\mathbf{y}_{s_2}^{(2)} \in \mathbb{R}^{r_1 \times P}$ and $\mathbf{y}_{s_2}^{(3)} \in \mathbb{R}^{1 \times J}$. In other words, each frame matrix $\boldsymbol{Y}_m$ constitutes the TT format with the block $\mathbf{y}^{(m)}$ replaced by the identity matrix of the corresponding size. To see this, fix s_1 and consider, for example, the first expression in (5.49). Here, we have

$$\boldsymbol{Y}_1(s_1) = I_{n_t} \otimes \sum_{s_2=1}^{r_2} \left(\mathbf{y}_{s_1,s_2}^{(2)}\right)^T \otimes \left(\mathbf{y}_{s_2}^{(3)}\right)^T.$$

Since $\mathbf{y}_{s_1}^{(1)} = \mathbf{y}_{s_1}^{(1)} \otimes 1 \otimes 1$, it trivially follows that

$$\mathbf{y} = \sum_{s_1=1}^{r_1} \boldsymbol{Y}_1(s_1)\mathbf{y}_{s_1}^{(1)} = \sum_{s_1,s_2=1}^{r_1,r_2} (I_{n_t}\mathbf{y}_{s_1}^{(1)}) \otimes \left(\mathbf{y}_{s_1,s_2}^{(2)}\right)^T \otimes \left(\mathbf{y}_{s_2}^{(3)}\right)^T.$$

The remaining two cases can be proven similarly. These frame matrices are used to project

[2] Note that each $\mathbf{y}^{(m)}$, $m = 1,2,3$ is reshaped into a column vector, say, $\widehat{\mathbf{y}}^{(m)}$ such that the number of columns of the frame matrix $\boldsymbol{Y}_m$ matches the length of $\widehat{\mathbf{y}}^{(m)}$.

the linear system. The ALS method updates the TT format by solving the following Galerkin linear systems:

$$\left(\boldsymbol{Y}_1^T \boldsymbol{A} \boldsymbol{Y}_1\right) \mathbf{y}^{(1)} = \boldsymbol{Y}_1^T \mathbf{g}, \tag{5.50}$$

$$\left(\boldsymbol{Y}_2^T \boldsymbol{A} \boldsymbol{Y}_2\right) \mathbf{y}^{(2)} = \boldsymbol{Y}_2^T \mathbf{g}, \tag{5.51}$$

$$\left(\boldsymbol{Y}_3^T \boldsymbol{A} \boldsymbol{Y}_3\right) \mathbf{y}^{(3)} = \boldsymbol{Y}_3^T \mathbf{g}, \tag{5.52}$$

and so on from the first step. Using the QR decompositions of the properly reshaped TT blocks, it is easy to make the frame matrices orthogonal, and therefore preserve the stability of the Galerkin systems, if $\boldsymbol{A}$ is positive definite. For example, it is enough to make $\mathbf{y}^{(1)}$ column-orthogonal and $\mathbf{y}^{(3)}$ row-orthogonal to make the whole $\boldsymbol{Y}_2$ column-orthogonal. Since this step is never a bottleneck, we always assume that the frame matrices are orthogonal, before solving (5.50)–(5.52).

However, the convergence of this algorithm is questionable. It is possible that the systems (5.50)–(5.52) remain the same within machine precision in two consecutive iterations, while the true residual of the initial linear system $\mathbf{g} - \boldsymbol{A}\mathbf{y}$ is large. The AMEn algorithm [32] was, in fact, developed to circumvent this problem. In what follows, we give a brief idea of the AMEn algorithm, adapted to 3-dimensional tensors, and then extended for saddle-point systems. To this end, note first that in addition to the solution, in AMEn we essentially approximate the residual in the TT format:

$$\mathbf{g} - \boldsymbol{A}\mathbf{y} \approx \mathbf{z} = \sum_{\zeta_1,\zeta_2=1}^{\rho_1,\rho_2} \mathbf{z}^{(1)}_{\zeta_1} \otimes (\mathbf{z}^{(2)}_{\zeta_1,\zeta_2})^T \otimes (\mathbf{z}^{(3)}_{\zeta_2})^T. \tag{5.53}$$

A very low accuracy is often sufficient for the residual (in our experiments we use $\rho_1 = \rho_2 = 2$), so we can use the simple ALS method to approximate the residual. Along the lines of (5.49), we construct the orthogonal *residual frame* matrices $\boldsymbol{Z}_m$ from (5.53) and compute $\mathbf{z}^{(m)} = \boldsymbol{Z}_m^T(\mathbf{g} - \boldsymbol{A}\mathbf{y})$ in a sequence $m = 1, 2, 3$, and so on. Since both $\boldsymbol{A}$ and $\mathbf{g}$ are given in the TT format, this computation is inexpensive. Moreover, it is enough to compute $\mathbf{z}^{(m)}$ only once after the m-th step of (5.50)–(5.52), i.e. perform one ALS iteration for $\mathbf{z}$ whenever the solution changes.

The crucial step now is the *enrichment* of the solution. Having solved (5.50), for example, we concatenate $\mathbf{y}^{(1)}$ and $\mathbf{z}^{(1)}$ as follows,

$$\mathbf{y}^{(1)}_{s_1'}(i) = \begin{cases} \mathbf{y}^{(1)}_{s_1'}(i), & s_1' = 1, \ldots, r_1, \\ \mathbf{z}^{(1)}_{s_1'-r_1}(i), & s_1' = r_1+1, \ldots, r_1+\rho_1, \end{cases}$$

and so on. The enrichment has a two-fold consequence. First, the residual can be well approximated in the basis of columns of the frame matrices, which prevents the Galerkin projection from a premature stagnation. Second, we can start from a low-rank initial guess and increase the TT ranks gradually, preventing them from becoming significantly larger than the ranks of the exact solution.

5.3.2 Block alternating iteration

The AMEn method performs well for positive definite matrices. However, the method may fail if we apply it to solve the KKT system (5.16) with the saddle-point matrix. The Galerkin projections (5.50)–(5.52) obey the Poincaré Separation Theorem [58, Section 4.3], and since the spectrum has both positive and negative parts, some of the eigenvalues may interlace with zero. Consequently, the projected matrices become degenerate and the calculation stops. To avoid this problem, we store the state $\mathbf{y}$, control $\mathbf{u}$ and adjoint $\mathbf{f}$ vectors in the *shared*, or *block* TT format [30], and preserve the KKT structure in the reduced system. Suppose that $\mathbf{y}, \mathbf{u}, \mathbf{f}$ are collected into a long vector $\hat{\mathbf{w}}^T = [\mathbf{w}_1^T, \mathbf{w}_2^T, \mathbf{w}_3^T] = [\mathbf{y}^T, \mathbf{u}^T, \mathbf{f}^T]$. The block TT format for $\hat{\mathbf{w}}$ can now be written in either of three variants:

$$\mathbf{w}_l(i,j,k) = \sum_{s_1,s_2=1}^{r_1,r_2} \hat{\mathbf{w}}^{(1)}_{s_1}(i,l)\mathbf{w}^{(2)}_{s_1,s_2}(j)\mathbf{w}^{(3)}_{s_2}(k), \tag{5.54}$$

$$\mathbf{w}_l(i,j,k) = \sum_{s_1,s_2=1}^{r_1,r_2} \mathbf{w}^{(1)}_{s_1}(i)\hat{\mathbf{w}}^{(2)}_{s_1,s_2}(j,l)\mathbf{w}^{(3)}_{s_2}(k), \tag{5.55}$$

$$\mathbf{w}_l(i,j,k) = \sum_{s_1,s_2=1}^{r_1,r_2} \mathbf{w}^{(1)}_{s_1}(i)\mathbf{w}^{(2)}_{s_1,s_2}(j)\hat{\mathbf{w}}^{(3)}_{s_2}(k,l). \tag{5.56}$$

The only difference between these three variants is in which TT block the index l ($l = 1, 2, 3$) is placed, but we need these different representations in different AMEn steps, as explained below. It is easy to switch between the representations using the SVD [30]. Given the variant (5.54), we reshape $\hat{\mathbf{w}}^{(1)}$ to a matrix $\hat{W}^{(1)} \in \mathbb{R}^{n_t \times 3r_1}$, compute the truncated SVD, namely, $\hat{W}^{(1)} \approx U\Sigma V^T$, where $U \in \mathbb{R}^{n_t \times r_1'}$, so the elements of U can be enumerated by two indices, $U(i, s_1')$, $i = 1, \dots, n_t$, $s_1' = 1, \dots, r_1'$. Therefore, $\mathbf{w}^{(1)}$ in (5.55) or (5.56) can be replaced by U. Then the matrix ΣV^T is reshaped to a matrix $R \in \mathbb{R}^{3r_1' \times r_1}$, indexed as $R(\overline{ls_1'}, s_1)$, and multiplied with $\mathbf{w}^{(2)}$ as follows:

$$\hat{\mathbf{w}}^{(2)}_{s_1', s_2}(j, l) := \sum_{s_1=1}^{r_1} R(\overline{ls_1'}, s_1)\mathbf{w}^{(2)}_{s_1, s_2}(j).$$

We notice that the result $\hat{\mathbf{w}}^{(2)}$ can overwrite $\mathbf{w}^{(2)}$ in (5.55), since it has the same form. In the same way, we can convert (5.55) to (5.56), or the other way around. Generally, the TT ranks change after such transformations. However, in the numerical practice the ranks remain comparatively the same in different block representations. The transition from one block variant to another is performed routinely in the AMEn iteration. Note that each of the variants (5.54)–(5.56) induces *only one* frame matrix $\boldsymbol{W}_m$ of the form (5.49), since the frame matrices do not depend on l:

$$\begin{aligned}
\boldsymbol{W}_1 &= I_{n_t} \otimes \sum_{s_2=1}^{r_2} \left(\mathbf{w}^{(2)}_{s_2}\right)^T \otimes \left(\mathbf{w}^{(3)}_{s_2}\right)^T, \\
\boldsymbol{W}_2 &= \mathbf{w}^{(1)} \otimes I_P \otimes \left(\mathbf{w}^{(3)}\right)^T, \\
\boldsymbol{W}_3 &= \sum_{s_1=1}^{r_1} \mathbf{w}^{(1)}_{s_1} \otimes \mathbf{w}^{(2)}_{s_1} \otimes I_J.
\end{aligned}$$

Therefore, to assemble the first reduced system (5.50), we need the first block representation (5.54), for the second system (5.51) we need (5.55), and so on. However, each frame matrix has the column size JPn_t, which coincides with the size of each of the blocks of (5.16), not the whole KKT matrix. Besides, we need a system of equations w.r.t. the index l, carried in the TT block under consideration. Thus, a natural generalization of (5.50)–(5.52) is the following

$$\begin{bmatrix} \boldsymbol{W}_m^T \tau \boldsymbol{M}_1 \boldsymbol{W}_m & 0 & -\boldsymbol{W}_m^T \boldsymbol{K}^T \boldsymbol{W}_m \\ 0 & \boldsymbol{W}_m^T \beta\tau \boldsymbol{M}_2 \boldsymbol{W}_m & \boldsymbol{W}_m^T \boldsymbol{N}^T \boldsymbol{W}_m \\ -\boldsymbol{W}_m^T \boldsymbol{K} \boldsymbol{W}_m & \boldsymbol{W}_m^T \boldsymbol{N} \boldsymbol{W}_m & 0 \end{bmatrix} \hat{\mathbf{w}}^{(m)} = \begin{bmatrix} \boldsymbol{W}_m^T \tau \boldsymbol{M}_a \bar{\mathbf{y}} \\ 0 \\ \boldsymbol{W}_m^T \mathbf{g} \end{bmatrix}, \qquad (5.57)$$

for $m = 1, 2, 3$.

Algorithm 5.1 Block AMEn iteration

```
Require: Blocks of the matrix $\mathfrak{A}$, right-hand side $\mathbf{g}$, initial guesses $w$ and $z$ in the TT format (5.54).
Ensure: Approximations of the solution $w$ and residual $z$.
while not converged do
    for $m = 1, 2, 3$ do
        Prepare and solve (5.57).
        Compute the residual $\hat{\mathbf{z}}^{(m)} = \begin{bmatrix} \boldsymbol{Z}_m^T(\mathbf{g}_1 - \mathfrak{A}_{1,:}\hat{\mathbf{w}}) \\ \boldsymbol{Z}_m^T(\mathbf{g}_2 - \mathfrak{A}_{2,:}\hat{\mathbf{w}}) \\ \boldsymbol{Z}_m^T(\mathbf{g}_3 - \mathfrak{A}_{3,:}\hat{\mathbf{w}}) \end{bmatrix}$.
        if $m < 3$ then
            Compute SVD of the solution: $\hat{\mathbf{w}}^{(m)}(i_m, l) \approx \mathbf{w}^{(m)}(i_m)\Sigma V(l)$.
            Move $l$ to the right: $\hat{\mathbf{w}}^{(m+1)}(i_{m+1}, l) = \Sigma V(l)\mathbf{w}^{(m+1)}(i_{m+1})$.
            Compute SVD of the residual: $\hat{\mathbf{z}}^{(m)}(i_m, l) \approx \mathbf{z}^{(m)}(i_m)\Sigma V(l)$.
            Define $\mathcal{W}^{(m)} := \mathcal{W}^{(m)}(\overline{s_{m-1}i_m}, s_m) = \mathbf{w}^{(m)}_{s_{m-1},s_m}(i_m)$.
            Define $\mathcal{Z}^{(m)} := \mathcal{Z}^{(m)}(\overline{s_{m-1}i_m}, s_m) = \mathbf{z}^{(m)}_{s_{m-1},s_m}(i_m)$.
            $[\mathcal{W}^{(m)} \;\; \mathcal{Z}^{(m)}] = QR$.
            $\mathbf{w}^{(m)}_{s_{m-1},s_m}(i_m) = Q(\overline{s_{m-1}i_m}, s_m)$.
        end if
    end for
    for $m = 3, 2, 1$ do
        Prepare and solve (5.57).
        Compute the residual $\hat{\mathbf{z}}^{(m)} = \begin{bmatrix} \boldsymbol{Z}_m^T(\mathbf{g}_1 - \mathfrak{A}_{1,:}\hat{\mathbf{w}}) \\ \boldsymbol{Z}_m^T(\mathbf{g}_2 - \mathfrak{A}_{2,:}\hat{\mathbf{w}}) \\ \boldsymbol{Z}_m^T(\mathbf{g}_3 - \mathfrak{A}_{3,:}\hat{\mathbf{w}}) \end{bmatrix}$.
        if $m > 1$ then
            Compute SVD of the solution: $\hat{\mathbf{w}}^{(m)}(i_m, l) \approx U(l)\Sigma\mathbf{w}^{(m)}(i_m)$.
            Move $l$ to the left: $\hat{\mathbf{w}}^{(m-1)}(i_{m-1}, l) = \mathbf{w}^{(m-1)}(i_{m-1})U(l)\Sigma$.
            Compute SVD of the residual: $\hat{\mathbf{z}}^{(m)}(i_m, l) \approx U(l)\Sigma\mathbf{z}^{(m)}(i_m)$.
            Define $\mathcal{W}^{(m)} := \mathcal{W}^{(m)}(s_{m-1}, \overline{i_m s_m}) = \mathbf{w}^{(m)}_{s_{m-1},s_m}(i_m)$.
            Define $\mathcal{Z}^{(m)} := \mathcal{Z}^{(m)}(s_{m-1}, \overline{i_m s_m}) = \mathbf{z}^{(m)}_{s_{m-1},s_m}(i_m)$.
            $[(\mathcal{W}^{(m)})^T \;\; (\mathcal{Z}^{(m)})^T] = QR$.
            $\mathbf{w}^{(m)}_{s_{m-1},s_m}(i_m) = Q(\overline{i_m s_m}, s_{m-1})$.
        end if
    end for
end while
```

After this system is solved, we use the SVD procedure outlined above to switch to the next block TT representation, compute the residual and enrich the new $\mathbf{w}^{(m)}$ (which does not contain l anymore). The residual is also kept in the block form, $\mathbf{z}^T = [\mathbf{z}_1^T, \mathbf{z}_2^T, \mathbf{z}_3^T]$, where $\mathbf{z}_l$ denotes the residual in the l-th row of the KKT system (5.16), and is approximated in the appropriate block TT representation. Its active block is computed as $\mathbf{z}^{(m)}(l) = \boldsymbol{Z}_m^T(\mathbf{g}_l - \mathfrak{A}_{l,:}\mathbf{w})$, and then the index l is replaced in the next TT block by the same SVD procedure. The pseudocode of the proposed block AMEn procedure is presented in Algorithm 5.1. For brevity, we omit the rank indices (e.g. $\mathbf{w}^{(m)}(i_m)$ is an $r_{m-1} \times r_m$ matrix, and so on), and introduce a uniform notation i_m, where $i_1 = i$, $i_2 = j$ and $i_3 = k$.

Since the block $\boldsymbol{M}_1$ is symmetric and semidefinite, the same property is inherited by the corresponding blocks in (5.57). However, $\boldsymbol{K}$ is the Stokes-Brinkman matrix, which is indefinite. 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 in what follows.

5.3.3 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 [79]. It reshapes the *velocity* component of the solution to a matrix $Y = [\mathbf{y}(\overline{ij}, k)]$, computes the truncated SVD $Y \approx U\Sigma V^T$, 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 $V = V(x) \in \mathbb{R}^r$, where r is the number of POD terms, and the reduced solution sought in the form $y(x,t) \approx V(x)e(t)$. Plugging this into the Stokes-Brinkman equation, and projecting the velocity equation onto V, we have

$$\left\{ \begin{array}{rcl} \frac{de}{dt} - \nu\langle V^T, \Delta V\rangle e + \langle V^T, \varrho V\rangle e + \langle V^T, \nabla p\rangle & = & \langle V^T, u\rangle, \\ \nabla \cdot Ve & = & 0. \end{array} \right. \tag{5.58}$$

Since $e(t)$ is not fixed a priori, from the second row of (5.58) we have $\nabla \cdot V(x) = 0$. However, then in the first row $\langle V^T, \nabla p\rangle = -\langle \nabla \cdot V^T, p\rangle = 0$; that is, the reduced model contains no pressure at all. In the discrete formulation, we have the system (5.12), and the pressure part $V^T \mathcal{B}^T \mathbf{p}$ is not exactly zero due to the boundary conditions. Nevertheless, it is often heuristically assumed that its magnitude is small [7]. If it is not the case, there are nonlinear corrections available [92]. They are important for the POD approach, since the last step there is the solution of the time-dependent reduced model. However, the alternating methods are different: we may stop the iteration at the spatial TT block and return the block TT format of the form (5.56), instead of (5.54) in the POD counterpart. Therefore, we perform the pressure exclusion trick (even if $V^T\mathcal{B}^T\mathbf{p}$ is not small) differently.

When we solve (5.57) for the spatial TT block ($m = 3$), we consider the 5×5 Stokes-KKT structure

$$\begin{bmatrix} \tau\hat{M}_1 & 0 & 0 & -\hat{A} & -\hat{B}^T \\ 0 & 0 & 0 & -\hat{B} & 0 \\ 0 & 0 & \beta\tau\hat{M}_2 & \hat{N}^T & 0 \\ -\hat{A} & -\hat{B}^T & \hat{N} & 0 & 0 \\ -\hat{B} & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{w}}^{(3)}(1) \\ \hat{\mathbf{w}}^{(3)}(2) \\ \hat{\mathbf{w}}^{(3)}(3) \\ \hat{\mathbf{w}}^{(3)}(4) \\ \hat{\mathbf{w}}^{(3)}(5) \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{b}}_\mathbf{1} \\ 0 \\ 0 \\ \hat{\mathbf{g}}_\mathbf{v} \\ \hat{\mathbf{g}}_\mathbf{p} \end{bmatrix}, \tag{5.59}$$

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

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

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

$$\begin{aligned} &\widehat{I}_0 = \mathcal{W}_3^T \left(I \otimes G_0\right) \mathcal{W}_3, \quad \widehat{I}_1 = \mathcal{W}_3^T \left(I \otimes G_1\right) \mathcal{W}_3, \quad \widehat{C}_0 = \mathcal{W}_3^T \left(C \otimes G_0\right) \mathcal{W}_3, \\ &\widehat{D}_0 = \mathcal{W}_3^T \left(D \otimes G_0\right) \mathcal{W}_3, \quad \widehat{D}_\alpha = \mathcal{W}_3^T \left(D \otimes G_\alpha\right) \mathcal{W}_3, \end{aligned} \tag{5.60}$$

whereas the right-hand side parts are

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

and $\mathcal{W}_3 = \sum\limits_{s_1=1}^{r_1} \mathbf{w}_{s_1}^{(1)} \otimes \mathbf{w}_{s_1}^{(2)} \in \mathbb{R}^{n_t P \times r_2}$ is a chunk of the frame matrix $\boldsymbol{W}_3$. We had to introduce this chunk and the Kronecker structures above in order to explain the preconditioner in the next section. Note that each of the reduced matrices in (5.60) belong to $\mathbb{R}^{r_2 \times r_2}$. Besides, we see that the solution components $\hat{\mathbf{w}}^{(3)}(2)$ and $\hat{\mathbf{w}}^{(3)}(5)$ denote the state and adjoint pressures, respectively. The new TT block is assembled from the remaining components only, $\mathbf{w}^{(3)} = \left[\hat{\mathbf{w}}^{(3)}(1), \hat{\mathbf{w}}^{(3)}(3), \hat{\mathbf{w}}^{(3)}(4)\right]$.

For the subsequent AMEn steps ($m = 2, 1$), we do not assume the pressure components to be small, but we assume that they will *not change* significantly. Therefore, their contributions to the velocity equations can be recast to the right-hand side. More precisely, we construct the TT formats

$$\delta \mathbf{b_1} = \sum_{s_1,s_2} \mathbf{w}^{(1)}_{s_1} \otimes G_0 \mathbf{w}^{(2)}_{s_1,s_2} \otimes B^T \hat{\mathbf{w}}^{(3)}_{s_2}(5), \quad \delta \mathbf{g} = \sum_{s_1,s_2} \mathbf{w}^{(1)}_{s_1} \otimes G_0 \mathbf{w}^{(2)}_{s_1,s_2} \otimes B^T \hat{\mathbf{w}}^{(3)}_{s_2}(2),$$

and correct the right-hand side of (5.16) as follows,

$$\begin{bmatrix} \mathbf{b_1} \\ 0 \\ \mathbf{g} \end{bmatrix} \quad \rightarrow \quad \begin{bmatrix} \mathbf{b_1} + \delta \mathbf{b_1} \\ 0 \\ \mathbf{g} + \delta \mathbf{g} \end{bmatrix}.$$

After that, we conduct AMEn steps $m = 2, 1, 2$ with the system of the form (5.57), where $\boldsymbol{K}$ contains now only the velocity equation, and hence is positive definite. When we come back to $m = 3$, we drop the right-hand side corrections and solve the full system (5.59). If we are to stop the iteration, we return the full solution, including $\hat{\mathbf{w}}^{(3)}(2)$ and $\hat{\mathbf{w}}^{(3)}(5)$. Due to the Galerkin projection, the accuracy depends only on how good the common TT blocks $\mathbf{w}^{(1)}$ and $\mathbf{w}^{(2)}$ represent all solution components. Although it is unclear whether it is allowed in general to 'freeze' some components, in our numerical experiments we observed that the solution is accurate enough; that is, the blocks $\mathbf{w}^{(1)}$ and $\mathbf{w}^{(2)}$ are computed accurately using only the velocity information.

5.3.4 Practical implementation

The preconditioner developed in Section 5.2.1 needs to be adjusted to the local problem (5.59). Although the reduced matrices (5.60) are small, they are dense, and it is impractical to compute the blocks of (5.59) explicitly. However, note that all of them are single Kronecker products except $\hat{A}$. Moreover, if the norms of K and M_ϱ 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 $\widehat{I_0} \otimes \left(\tau^{-1}M + \nu_0 K + M_\varrho\right)$ during the preconditioning. This also allows to avoid the second level of preconditioning for the Stokes-Brinkman system (5.29). Since $\hat{B}$ contains

$\widehat{I}_0$, we can assemble the Stokes-Brinkman matrix in the Kronecker form as well,

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

In the computation of $\mathbf{x}_3$ in (5.25), we can solve linear systems with $\hat{\mathcal{K}}$ directly. For two-dimensional cases, this approach is faster than iterations with (5.29). In the same way we approximate the factors of the Schur complement (5.27), e.g.

$$\hat{\mathcal{K}}^T + \hat{\mathcal{M}}_r \approx \widehat{I}_0 \otimes \begin{bmatrix} \left(\frac{1}{\tau} + \frac{1}{\sqrt{\beta}}\frac{\|\widehat{D}_\alpha\|}{\|\widehat{I}_0\|}\right) M + \nu_0 K + M_\varrho & B^T \\ B & 0 \end{bmatrix}, \tag{5.61}$$

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 (5.60). For three dimensional problems (Section 5.4.10), the matrices become more dense, and we have to use iterative methods, preconditioning the velocity block by a multigrid cycle. Similar approximation of the preconditioners in the form of one Kronecker product is performed for the TT blocks $\mathbf{w}^{(1)}$ and $\mathbf{w}^{(2)}$. Although they are smaller than the spatial block, they are still too large to form and solve the systems (5.57) directly. The crucial point here, fortunately, is that the new preconditioner does not need to invert $\boldsymbol{M}_1$ (cf. (5.25)).

5.4 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 *mini* elements [120] and the pressure operators are discretized with piecewise linear finite elements. The stiffness matrices are assembled in the FEniCS 1.5.0 package [83].

For the Stokes-Brinkman equation, the coefficient is chosen as follows:

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

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 $\bar{\varrho}$. 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[3] $\nu_1 = 0.1$, approximation tolerance $\varepsilon = 10^{-6}$, and pure Stokes coefficient $\bar{\varrho} = 0$. 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 is chosen as $P = 16$.

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

$$\mathcal{E}(\mathbf{y},\mathbf{y}_\star) = \|\mathbf{y}-\mathbf{y}_\star\|_F / \|\mathbf{y}_\star\|_F. \tag{5.62}$$

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

$$\text{residual} = \max\left(\mathcal{E}(\tau \boldsymbol{M}_1\mathbf{y} - \boldsymbol{K}^T\mathbf{f}, \tau\mathbf{M_a}\bar{\mathbf{y}});\ \mathcal{E}(\tau\beta\mathbf{M_2u}, \mathbf{N^Tf});\ \mathcal{E}(-\mathbf{Ky}+\mathbf{Nu},\mathbf{g})\right).$$

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

[3] 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, for example, in biomedical modeling [120].

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 (5.62).

Remark 5.4. *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. As in Chapter 4, the TT algorithms are implemented within the TT-Toolbox [98]. 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.

$$\%\ \text{Mem} = \frac{n_t r_1 + r_1 P r_2 + r_2 J}{J P n_t} \cdot 100. \tag{5.63}$$

By 'iterations', we mean the total number of FGMRES iterations, spent in solving the reduced systems (5.59) for the spatial TT block, in all AMEn steps. The FGMRES is used with the block-triangular preconditioner (5.25) for the KKT level only (the Stokes-like systems (5.61) are solved directly in two-dimensional examples).

Remark 5.5. *Note that, in the figures on the right hand sides of Figures 5.3 – 5.10, each of the multiple vertical axes corresponds only to the plot bearing the respective colour in the figure question. More precisely, the blue vertical axis correponds to the plot for proportion of memory consumption only; the black vertical axis is only for the plot for the CPU times, whereas the red vertical axis is just for the plot for iterations.*

Table 5.1: 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

5.4.1 Performance of the new block-triangular preconditioner

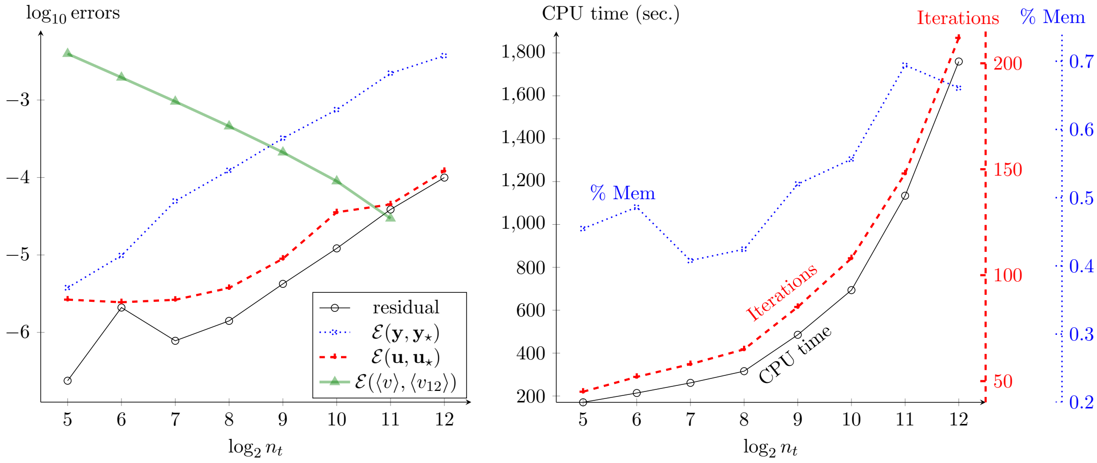
It is illustrative to compare the new preconditioner (5.25) with the established block-diagonal preconditioner $\boldsymbol{P}_1$ from [126], mentioned at the beginning of Section 5.2. We test $\boldsymbol{P}_1$ using the MINRES method, for the spatial TT block only. The comparison with $\boldsymbol{P}_2$ (5.25) is given in Table 5.1. We see that $\boldsymbol{P}_2$ provides faster convergence in terms of both iterations and time. Therefore, we use $\boldsymbol{P}_2$ in all the remaining experiments in this chapter.

5.4.2 Experiment with n_t (Figure 5.3)

In the first test, we vary the number of time steps from 2^5 to 2^{12}. In addition to general errors, 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:

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

Figure 5.3: 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 as given by (5.63).



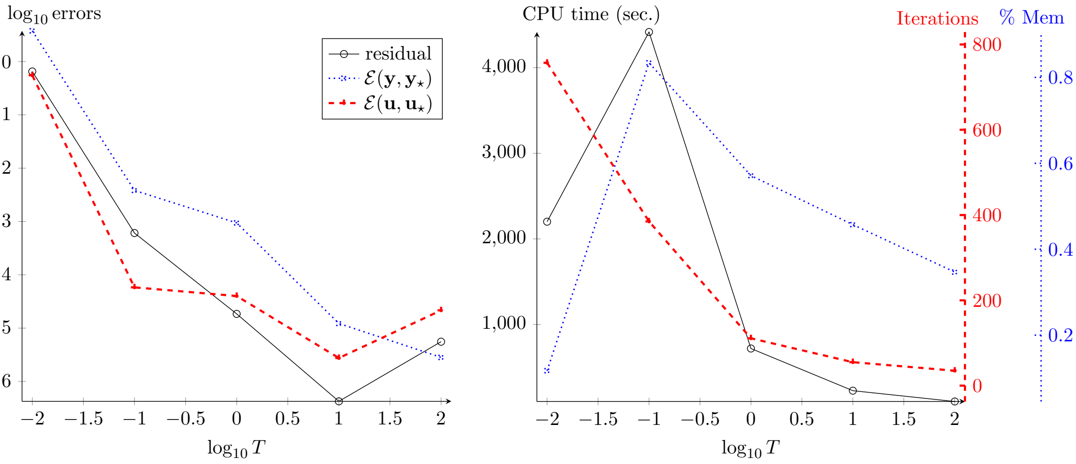
Note that $\mathbf{y}$ has the form $[\mathbf{v}, \mathbf{p}]$ w.r.t. the index k, so that the summation $k, k' = 1, \dots, J_v$ extracts only the velocity. The reference value $\langle v_{12} \rangle$ is computed on the grid $n_t = 2^{12}$. The distance from $\langle v \rangle$ decays proportionally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

The errors grow proportionally to the grid size, since the matrix becomes more ill-conditioned. However, the CPU times and the numbers of iterations grow only as a small power of $\log n_t$. The behavior of the CPU time is very close to the behavior of the iterations, while the TT ranks (and hence the memory) are almost stable w.r.t. n_t. This shows that the main reason for the increase in time is the deterioration of the quality of the preconditioner (since we use the rank-1 approximation (5.61)). A more robust (in terms of iterations) preconditioner should also involve the term related to the time derivative. However, each iteration might become more costly. Future research is needed to make the preconditioner suitable for extreme parameters.

5.4.3 Experiment with T (Figure 5.4)

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 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

Figure 5.4: 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 as given by (5.63).



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.

5.4.4 Experiment with β (Figure 5.5)

Although there are rigorous mathematical ways to estimate β for a given problem, such as the L-curve analysis [53] or the discrepancy principle [40], 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.

Figure 5.5: 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 as given by (5.63).

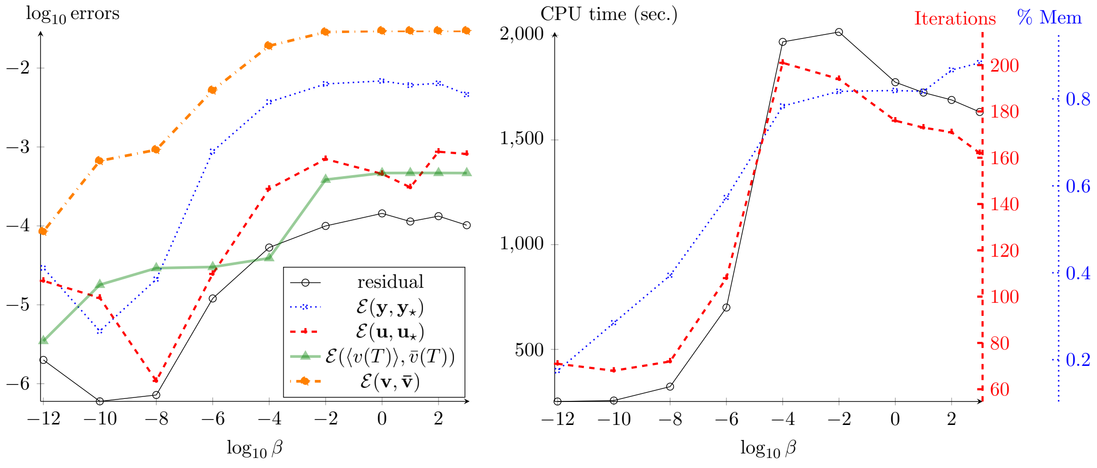
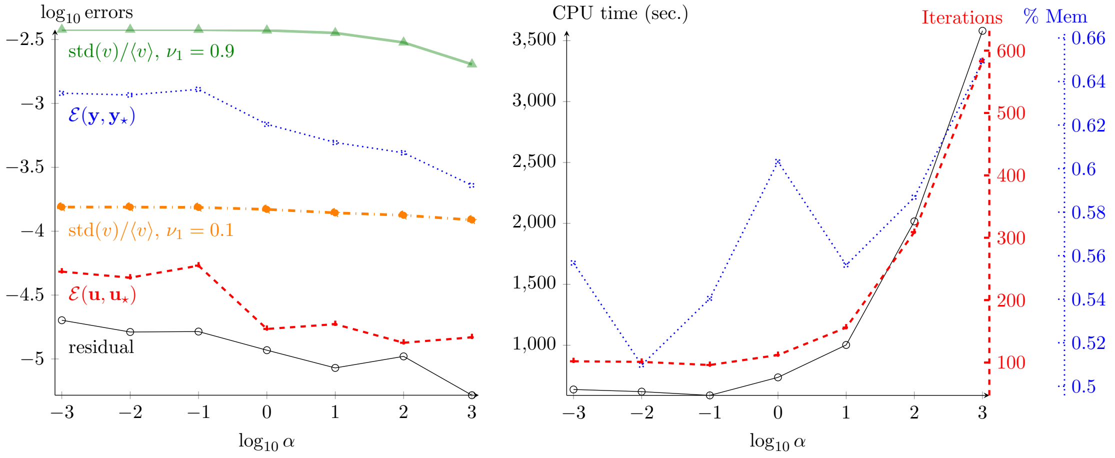


Figure 5.6: 2D Stokes, experiment with α. Left: Residual and errors w.r.t. the reference solutions, and the relative standard deviation. Right: CPU time, total number of iterations in spatial systems, memory compression ratio as given by (5.63).



5.4.5 Experiment with α (Figure 5.6)

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

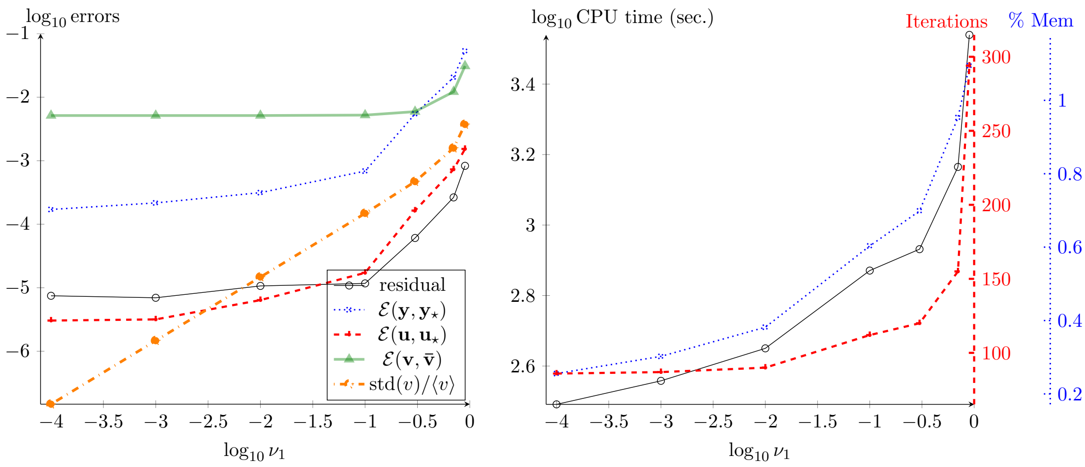
$$\operatorname{std}(v) = \sqrt{\frac{\tau}{T} \sum_{k,k',i=1}^{J_v,J_v,n_t} \sum_{j=2}^{P} M(k,k')G_0(j,j)D(i,i)y^2(i,j,k')}.$$

In Fig. 5.6, 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 $\alpha = 10^4$, 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.

5.4.6 Experiment with ν_1 (Figure 5.7)

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

Figure 5.7: 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 as given by (5.63).

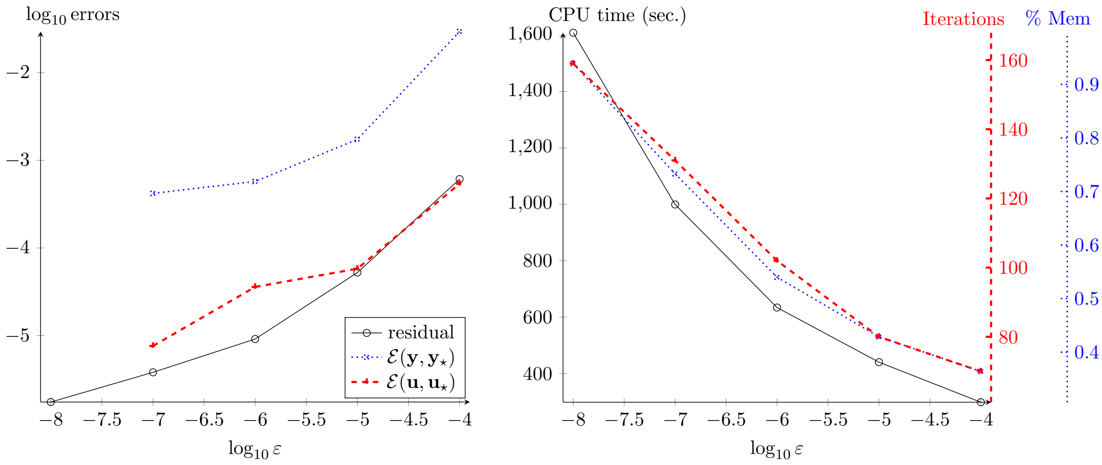


1, the behavior becomes essentially nonlinear, e.g. for $\nu_1 = 0.9$ we have $\nu_{\max}/\nu_{\min} = 19$. The same can be seen in both error and complexity figures. The residuals and errors are almost stable for small ν_1, and the standard deviation grows linearly, while for $\nu_1 > 0.5$, all quantities 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.

5.4.7 Experiment with the tensor approximation tolerance (Figure 5.8)

Here, we confirm the consistency of the error estimate $\mathcal{E}(\mathbf{y},\mathbf{y}_\star)$, see Remark 5.4. 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 $\varepsilon^{0.5}$, and the state error almost stagnates. This may be caused by the indefiniteness of the problem and the pressure exclusion trick. Unfortunately, we are unable to study their effects separately, as the reduced systems (5.50), (5.51) and (5.52) become degenerate if we try to apply the AMEn to an indefinite system directly.

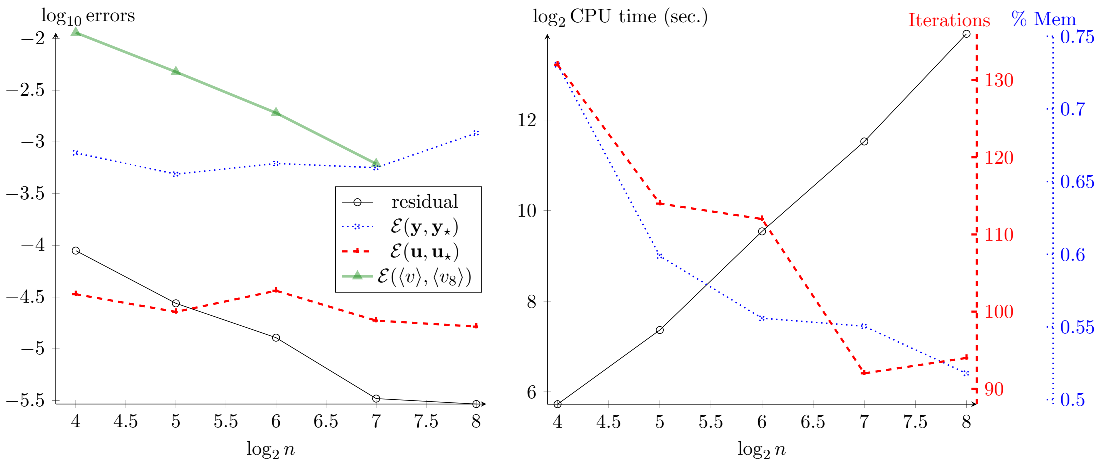
Figure 5.8: 2D Stokes, experiment with ε. Left: Residual and errors w.r.t. the reference solutions. Right: CPU time, total number of iterations in spatial systems, memory compression ratio as given by (5.63).



5.4.8 Experiment with n (Figure 5.9)

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 is $J \approx 7n^2$. 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 $n^{-1.4}$.

Figure 5.9: 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 as given by (5.63).



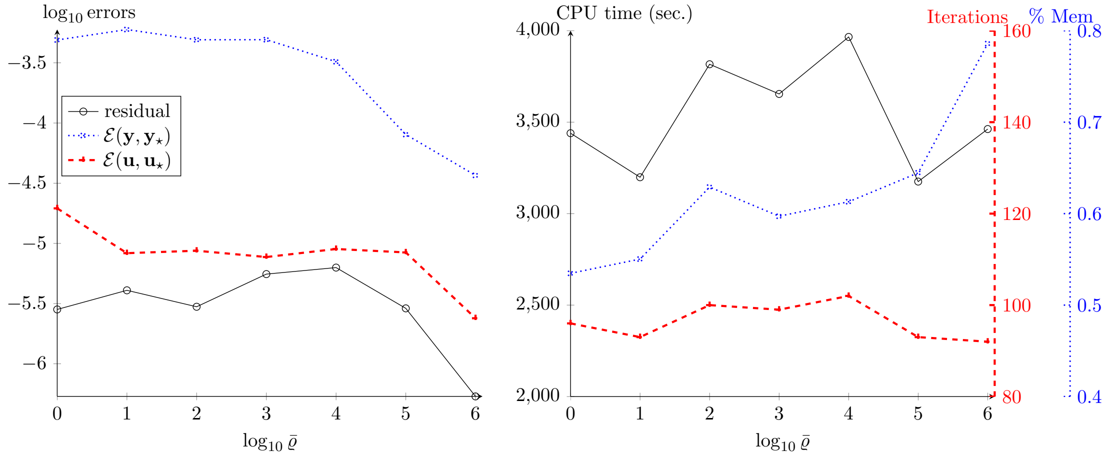
The most time-consuming stage in the scheme is the solution of the system for the spatial TT block. The sparsity of the spatial matrix allows its efficient factorization, such that the CPU time grows proportionally to n^2, i.e. linear w.r.t. the total number of spatial degrees of freedom. Interestingly, 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 (5.27). 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.

5.4.9 Experiment with $\bar{\varrho}$ (Figure 5.10)

Finally, we take $\bar{\varrho}$ nonzero and investigate the Stokes-Brinkman model. For some reasons, with $n = 64$ and $\bar{\varrho} > 10^5$, 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 $\varrho(\mathbf{x})$. 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 $\bar{\varrho} = 10^6$.

We see that the scheme is quite robust in the considered range of the coefficient. The error estimates decay with increasing $\bar{\varrho}$, since the system becomes closer to the Darcy model. The CPU time and the number of iterations show a chaotic behavior behavior due to randomization in the AMEn algorithm and CPU workload, but this fluctuation is only

Figure 5.10: 2D Stokes-Brinkmann, experiment with $\bar{\varrho}$. Left: Residual and errors w.r.t. the reference solutions. Right: CPU time, total number of iterations in spatial systems, memory compression ratio as given by (5.63).



10–20% compared to the average values. This fluctuation is always observed; here, it is however not big, i.e. the time is "constant" on average.

5.4.10 3D problem (Figure 5.11)

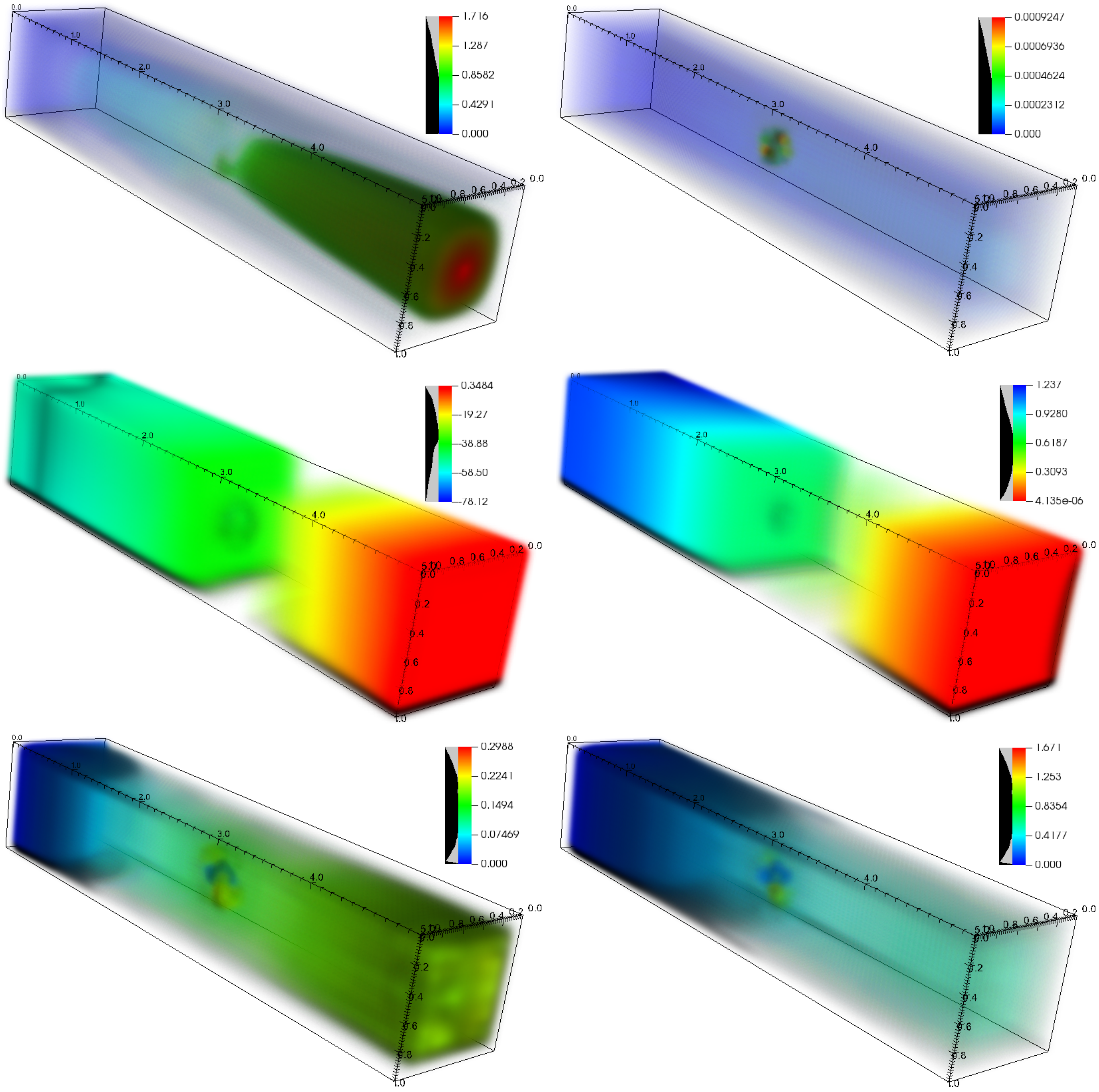
Finally, we demonstrate that our approach is suitable for larger 3D problems. We consider the three-dimensional Stokes-Brinkman problem on the domain $[0,1] \times [0,1] \times [0,5]$ as constraints, with the coefficient

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

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 billion, 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 (5.35) for the Schur complement in the Stokes matrices, and the velocity matrix was approximated by one V-cycle of the HSL MI20 algebraic multigrid [21]. 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 (5.61), we used another FGMRES method with 50 iterations, preconditioned by (5.35) 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 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. 5.11. We notice a clear perturbation around the region with nonzero

Figure 5.11: 3D Stokes-Brinkman. Left: mean values at the last time step, right: standard deviations. Top: velocity, middle: pressure, bottom: control.



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.

Chapter 6

Conclusions and outlook

The use of classical spectral SGFEM in discretizing models governed by PDEs with uncertain inputs is standard in the literature. More often than not, SGFEM discretization leads to large dimensional coupled linear systems with tensor product structure. Hence, the straightforward storage of the solution consumes a vast amount of memory. However, since Galerkin approximation yields a best approximation with respect to the energy norm, as well as a favorable framework for error estimation [18], it is worth pursuing more computationally efficient ways to simulate these models using SGFEMs. With a view to reducing the computational time and memory requirements of the solution of such arbitrarily large linear systems, we have provided a theoretical basis for a low-rank solver to achieve these goals in this dissertation. Furthermore, we have solved the linear systems (2.37) resulting from the forward problem (2.33) using a low-rank conjugate gradient iterative solver, together with two different preconditioners. In general, the combination of each of the preconditioners and the low-rank iterative solver seems quite promising for large-scale simulation of models whose random input data have comparatively low variance, as it reduces the computer memory and computational time required to solve the stochastic Galerkin linear system compared to the conventional method.

We have also considered low-rank approaches to the solution of optimal control problems constrained by either diffusion equations or Stokes-Brinkman equations with uncertain inputs. In the context of diffusion SOCPs, we have proposed robust Schur complement-based block-diagonal preconditioners to simulate the considered problems. Crucially, we have presented detailed analyses of the spectra of the derived preconditioners. Here, our approach to the solution of the KKT linear systems entails a formulation that solves the systems at once (for all time steps in the unsteady case). The all-at-once strategy often leads to a large system that cannot be solved with direct solvers. However, combining our proposed preconditioners with appropriate low-rank iterative solvers has proven efficient in accomplishing the tasks. In particular, inexact solves with the derived Schur complements via a few iterations of the preconditioned Richardson algorithm seem quite promising.

Next, the most challenging problem considered in this thesis – unsteady Stokes-Brinkman SOCP – yields a saddle-point linear system, which requires a special treatment with tensor-based techniques. We have proposed a new Schur complement-based block-triangular preconditioner which is free from auxiliary perturbations and provides smaller condition numbers of the preconditioned matrix compared to an already existing preconditioner proposed in the framework of a deterministic Stokes control problem in [126]. Furthermore, we have extended the alternating minimal energy algorithm such that it preserves the saddle-point structure and solves this system robustly. In particular, by employing the developed tensor product decomposition methods for the Stokes-Brinkman SOCP, we have reduced its solution storage requirements by two – three orders of magnitude. It is perhaps pertinent to state here that, although the low-rank approach discussed in this work introduces further error in the simulation due to the low-rank truncations, the relative tolerance of the truncation operator can be so tightened that the error will become negligible. More importantly, even though the low-rank truncation does not come free of charge, it enables the solution of unsteady UQ problems that would be otherwise intractable.

Several directions for future research are possible. To begin with, we note here that in many applications such as groundwater flow modeling [27, 131], one frequently encounters cases where the diffusivity coefficient in (2.3) is modelled as $a \approx \exp(a_N)$, where a_N is of the form (2.9). This is the so-called *stochastically nonlinear* case and it leads to a block dense linear system of the form (2.24), where the number of blocks depends *nonlinearly* on N; hence, the cost of solving the linear system becomes increasingly expensive as N increases [131]. The possible presence of a small correlation length in the covariance function associated with the random field a further exacerbates the problem since N then has to be large to control the error between a and the approximation a_N. The more random variables are needed to parameterize the uncertainty in the logarithm of the diffusivity coefficient, the higher the cost of a matrix-vector product with the SGFEM matrix and, thus, the higher the cost of one iteration (with or without a preconditioner) in the CG or MINRES algorithm. Hence, in the context of the stochastically nonlinear case, it will be reasonable to investigate the performance of our proposed low-rank iterative solvers as discussed particularly in Chapter 4. Another natural extension would be to apply the tensor techniques developed in Chapter 5 to nonlinear Navier-Stokes SOCPs. Furthermore, we admit here that the block-triangular preconditioner presented in Section 5.2.1 still needs 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.

Bibliography

[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] I. Babuška and P. Chatzipantelidis, *On solving linear elliptic stochastic partial differential equations*, Computer Methods in Applied Mechanics and Engineering, 191 (2002), pp. 4093–4122.

[4] I. Babuška, F. Nobile, and R. Tempone, *A stochastic collocation method for elliptic partial differential equations with random input data*, SIAM Journal on Numerical Analysis, 45 (2007), pp. 1005–1034.

[5] I. Babuška, R. Tempone, and G. Zouraris, *Galerkin finite element approximations of stochastic elliptic partial differential equations*, SIAM Journal on Numerical Analysis, 42 (2004), pp. 800–825.

[6] J. Baglama and L. Reichel, *Augmented implicitly restarted Lanczos bidiagonalization methods*, SIAM Journal on Scientific Computing, 27 (2005), pp. 19–42.

[7] 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.

[8] 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.

[9] V. Barthelmann, E. Novak, and K. Ritter, *High dimensional polynomial interpolation on sparse grids*, Advances in Computational Mathematics, 12 (2000), pp. 273 – 288.

[10] P. Benner and T. Breiten, *Low rank methods for a class of generalized Lyapunov equations and related issues*, Numerische Mathematik, 124 (2013), pp. 441–470.

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

[12] P. Benner and M. W. Hess, *Reduced basis modeling for uncertainty quantification of electromagnetic problems in stochastically varying domain*, Scientific Computing in Electrical Engineering SCEE 2014, Accepted, (2015).

[13] 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.

[14] ——, *Block-diagonal preconditioning for optimal control problems constrained by PDEs with uncertain inputs*, SIAM Journal on Matrix Analysis and Applications, 37 (2016), pp. 491 – 518.

[15] P. Benner and J. Schneider, *Uncertainty quantification for Maxwell's equations using stochastic collocation and model order reduction*, International Journal for Uncertainty Quantification, 5 (2015), pp. 195 – 208.

[16] M. Benzi and G. H. Golub, *A preconditioner for generalized saddle point problems*, SIAM Journal on Matrix Analysis and Applications, 26 (2004), pp. 20 – 41.

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

[18] A. Bespalov, C. E. Powell, and D. Silvester, *Energy norm a posteriori error estimation for parametric operator equations*, SIAM Journal on Scientific Computing, 36 (2013), pp. A339 – A363.

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

[20] A. Borzi and G. von Winckel, *Multigrid methods and sparse-grid collocation techniques for parabolic optimal control problems with random coefficients*, SIAM Journal on Scientific Computing, 31 (2009), pp. 2172 – 2192.

[21] 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.

[22] J. H. Bramble and J. E. Pasciak, *A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems*, Mathematics of Computation, 50 (1988), pp. 1 – 17.

[23] S. C. Brenner and L. R. Scott, *The Mathematical Theory of Finite Element Methods*, vol. Second Edition, Springer, 2012.

[24] A. N. Brooks and T. J. R. Hughes, *Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations*, Computer Methods in Applied Mechanics and Engineering, 32 (1982), pp. 199–259.

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

[26] P. Chen and A. Quarteroni, *Weighted reduced basis method for stochastic optimal control problems with elliptic PDE constraint*, SIAM/ASA Journal on Uncertainty Quantification, 2 (2014), pp. 364 – 396.

[27] K. A. Cliffe, M. B. Giles, R. Scheichl, and A. L. Teckentrup, *Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients*, Computing and Visualization in Science, 14 (2011), pp. 3–15.

[28] T. Damm, *Direct methods and ADI-preconditioned Krylov subspace methods for generalized Lyapunov equations*, Numerical Linear Algebra and Applications, 15 (2008), pp. 853–871.

[29] 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.

[30] 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.

[31] S. V. Dolgov, J. W. Pearson, D. V. Savostyanov, and M. Stoll, *Fast tensor product solvers for optimization problems with fractional differential equations as constraints*, Applied Mathematics and Computation, To appear, (2016).

[32] 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.

[33] H. S. Dollar, N. I. M. Gould, W. H. A. Schilders, and A. J. Wathen, *Implicit-factorization preconditioning and iterative solvers for regularized saddle-point systems*, SIAM Journal on Matrix Analysis and Applications, 28 (2006), pp. 170 – 189.

[34] H. S. Dollar, N. I. M. Gould, M. Stoll, and A. J. Wathen, *Preconditioning saddle-point systems with applications in optimization*, SIAM Journal on Scientific Computing, 32 (2010), pp. 249 – 270.

[35] H. Elman, O. G. Ernst, D. P. O'Leary, and M. Stewart, *Efficient iterative algorithms for the stochastic finite element method with applications to acoustic*

scattering, Computer Methods in Applied Mechanics and Engineering, 194 (2005), pp. 1037–1055.

[36] H. Elman and D. Furnival, *Solving steady-state diffusion problem using multigrid*, IMA Journal of Numerical Analysis, 27 (2007), pp. 675–688.

[37] H. Elman, C. Miller, E. Phipps, and R. S. Tuminaro, *Assessment of collocation and Galerkin approaches to linear diffussion equations with random data*, International Journal for Uncertainty Quantification, 1 (2011), pp. 19–33.

[38] 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.

[39] ——, *Finite Elements and Fast Iterative Solvers*, vol. Second Edition, Oxford University Press, 2014.

[40] 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.

[41] B. Fischer, A. Ramage, D. J. Silvester, and A. J. Wathen, *Minimum residual methods for augmented systems*, BIT, 38 (1998), pp. 527–543.

[42] P. Frauenfelder, C. Schwab, and R. A. Todor, *Finite elements for elliptic problems with stochastic coefficients*, Computer Methods in Applied Mechanics and Engineering, 194 (2005), pp. 205–228.

[43] R. G. Ghanem and P. Spanos, *Stochastic Finite Elements: A Spectral Approach*, Springer-Verlag, New York, 1996.

[44] W. Gilks, S. Richardson, and D. Spiegelhalter, *Markov Chain Monte Carlo in Practice*, Chapman & Hall, London, 1995.

[45] G. H. Golub and C. Greif, *On solving block-structured indefinite linear systems*, SIAM Journal on Scientific Computing, 24 (2003), pp. 2076 – 2092.

[46] G. H. Golub and C. H. Van Loan, *Matrix Computations*, Johns Hopkins University Press, 1996.

[47] A. D. Gordon and C. E. Powell, *Solving stochastic collocation systems with algebraic multigrid*, in Numerical Mathematics and Advanced Applications, G. Kreiss et al., ed., Springer-Verlag, 2009, pp. 377–384.

[48] L. Grasedyck, *Existence and computation of low Kronecker-rank approximations for large linear systems of tensor product structure*, Computing, 72 (2004), pp. 247–265.

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

[50] M. D. Gunzburger, H.-C. Lee, and J. Lee, *Error estimates of stochastic optimal Neumann boundary control problems*, IMA Journal on Numerical Analysis, 49 (2011), pp. 1532 – 1552.

[51] M. D. Gunzburger, C. G. Webster, and G. Zhang, *Stochastic finite element methods for equations with random input data*, Acta Numerica, 23 (2014), pp. 521–650.

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

[53] 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.

[54] 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.

[55] 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.

[56] M. E. Hochstenbach, *A Jacobi-Davidson type SVD method*, SIAM Journal on Scientific Computing, 23 (2001), pp. 606–628.

[57] 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.

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

[59] L. S. Hou, J. Lee, and H. Manouzi, *Finite element approximations of stochastic optimal control problems constrained by stochastic elliptic PDEs*, Journal of Mathematical Analysis and Applications, 384 (2011), pp. 87–103.

[60] 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.

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

[62] K. Ito, K. Kunisch, V. Schulz, and I. Gherman, *Approximate nullspace iterations for KKT systems*, SIAM Journal on Matrix Analysis and Applications, 31 (2010), pp. 1835–1847.

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

[64] B. Jin and J. Zou, *Inversion of Robin coefficient by a spectral stochastic finite element approach*, Journal of Computational Physics, 227 (2008), pp. 3282 – 3306.

[65] C. Jin, X-C. Cai., and C. Li, *Parallel domain decomposition methods for stochastic elliptic equations*, SIAM Journal on Scientific Computing, 29 (2007), pp. 2096 – 2114.

[66] H. Johnston and J. G. Liu, *Accurate, stable and efficient Navier-Stokes solvers based on explicit treatment of the pressure term*, Journal of Computational Physics, 199 (2004), pp. 221–259.

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

[68] A. Keese, *Numerical Solution of Systems with Stochastic Uncertainties: A General Purpose Framework for Stochastic Finite Elements*, PhD thesis, Fachbereich fuer Mathematik und Informatik, Technische Universitaet Braunschweig, 2004.

[69] C. Keller, N. I. M. Gould, and A. J. Wathen, *Implicit-factorization preconditioning and iterative solvers for regularized saddle-point systems*, SIAM Journal on Matrix Analysis and Applications, 21 (2000), pp. 1300 – 1317.

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

[71] A. Klawonn, *Block-triangular preconditioners for saddle point problems with a penalty term*, SIAM Journal on Scientific Computing, 19 (1998), pp. 172 – 184.

[72] P. Kloeden and E. Platen, *Numerical Solution of Stochastic Differential Equations*, Springer-Verlag, New York, 1999.

[73] 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.

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

[75] D. P. Kouri, M. Heinkenschloss, D. Ridzal, and B. G. van Bloemen Waanders, *A trust-region algorithm with adaptive stochastic collocation for PDE optimization under uncertainty*, SIAM Journal on Scientific Computing, 35 (2013), pp. A1847 – A1879.

[76] 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.

[77] 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.

[78] ——, *Low-rank tensor Krylov subspace methods for parametrized linear systems*, SIAM Journal on Matrix Analysis and Applications, 32 (2011), pp. 1288–1316.

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

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

[81] O. P. Le Maître, O. M. Knio, B. J. Debusschere, H. N. Najm, and R. G. Ghanem, *A multigrid solver for two-dimensional stochastic diffusion equations*, Computer Methods in Applied Mechanics and Engineering, 192 (2003), pp. 4723–4744.

[82] 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.

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

[84] O. P. Le Maître and O. M. Knio, *Spectral Methods for Uncertainty Quantification with Applications to Computational Fluid Dynamics*, Springer, 2010.

[85] 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.

[86] 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.

[87] K. A. Mardal and R. Winther, *Preconditioning discretizations of systems of partial differential equations*, Numerical Linear Algebra with Applications, 18 (2011), pp. 1–40.

[88] T. Mathew, M. Sarkis, and C. Schaerer, *Analysis of block matrix preconditioners for elliptic optimal control problems*, Numerical Linear Algebra with Applications, 14 (2007), pp. 257–279.

[89] M. F. Murphy, G. H. Golub, and A. J. Wathen, *A note on preconditioning for indefinite linear systems*, SIAM Journal on Scientific Computing, 21 (2000), pp. 1969 – 1972.

[90] D. W. Nicholson, *Eigenvalue bounds for AB+BA with A, B positive definite matrices*, Linear Algebra and its Applications, 24 (1979), pp. 173 – 183.

[91] H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*, SIAM, Philadelphia, 1992.

[92] 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.

[93] F. Nobile and R. Tempone, *Analysis and implementation issues for the numerical approximation of parabolic equations with random coefficients*, International Journal for Numerical Methods in Engineering, 80 (2009), pp. 979 – 1006.

[94] ——, *Analysis and implementation issues for the numerical approximation of parabolic equations with random coeffients*, International Journal of Numerical Methods in Engineering, 80/6-7 (2009), pp. 979–1006.

[95] Y. Notay, *A new analysis of block preconditioners for saddle point problems*, SIAM Journal on Matrix Analysis and Applications, 35 (2014), pp. 143 – 173.

[96] J. Ockendon, S. Howison, A. Lacey, and A. Movchan, *Applied Partial Differential Equations*, Oxford University Press, Oxford, 2003.

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

[98] I. V. Oseledets, S. Dolgov, V. Kazeev, D. Savostyanov, O. Lebedeva, P. Zhlobich, T. Mach, and L. Song, *TT-Toolbox*, 2016. https://github.com/oseledets/TT-Toolbox.

[99] 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.

[100] C. C. Paige and M. A. Saunders, *Solutions of sparse indefinite systems of linear equations*, SIAM Journal on Numerical Analysis, 12 (1975), pp. 617–629.

[101] M. L. Parks, E. de Sturler, G. Mackey, D. D. Johnson, and S. Maiti, *Recycling Krylov subspaces for sequences of linear systems*, SIAM Journal on Scientific Computing, 28 (2006), pp. 1651 – 1674.

[102] J. W. Pearson, *Preconditioned iterative methods for Navier-Stokes control problems*, Journal of Computational Physics, 292 (2015), pp. 194 – 207.

[103] 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.

[104] 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.

[105] 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.

[106] M. F. Pellissetti and R. G. Ghanem, *Iterative solution of systems of linear equations arising in the context of stochastic finite elements*, Advances in Engineering Software, 31 (2000), pp. 607–616.

[107] J. Pestana and A. J. Wathen, *Natural preconditioning and iterative methods for saddle point systems*, SIAM Review, 57 (2015), pp. 71 – 91.

[108] 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.

[109] 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.

[110] 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.

[111] 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.

[112] T. Rees, H. S. Dollar, and A. J. Wathen, *Optimal solvers for PDE-constrained optimization*, SIAM Journal on Scientific Computing, 32 (2010), pp. 271–298.

[113] F. Riesz and B. Sz.-Nagy, *Functional Analysis*, Dover, New York, 1990.

[114] E. Rosseel, T. Boonen, and S. Vandewalle, *Algebraic multigrid for stationary and time-dependent partial differential equations with stochastic coefficients*, Numerical Linear Algebra and Applications, 15 (2008), pp. 141–163.

[115] 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.

[116] T. Rusten and R. Winther, *A preconditioned iterative method for saddle point problems*, SIAM Journal on Matrix Analysis and Applications, 13 (1992), p. 887.

[117] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM Journal on Scientific and Statistical Computing, 7 (1986), pp. 856–869.

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

[119] V. Simoncini and M. Benzi, *Spectral properties of the Hermitian and skew-Hermitian splitting preconditioner for saddle point problems*, SIAM Journal on Matrix Analysis and Applications, 26 (2004), pp. 377 – 389.

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

[121] B. Sousedík and R. G. Ghanem, *Truncated hierrarchical preconditioning for the stochastic Galerkin FEM*, International Journal for Uncertainty Quantification, 4 (2014), pp. 333 – 348.

[122] M. Stein, *Large sample properties of simulations using Latin hypercube sampling*, Technometrics, 29 (1987), pp. 143 – 151.

[123] M. Stoll, *A Krylov-Schur approach to the truncated SVD*, Linear Algebra and its Applications, 436 (2012), pp. 2795–2806.

[124] 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.

[125] M. Stoll and A. Wathen, *Preconditioning for partial differential equation constrained optimization with control constraints*, Numerical Linear Algebra with Applications, 19 (2012), pp. 53–71.

[126] ——, *All-at-once solution of time-dependent Stokes control*, Journal of Computational Physics, 232 (2013), pp. 498–515.

[127] K. Stüben, *An introduction to algebraic multigrid*, in Multigrid, A. Schuller U. Trottenberg, C. Oosterlee, ed., Academic Press, 2001, pp. 413 – 532.

[128] H. Tiesler, R. M. Kirby, D. Xiu, and T. Preusser, *Stochastic collocation for optimal control problems with stochastic PDE constraints*, SIAM Journal on Control and Optimization, 50 (2012), pp. 2659 – 2682.

[129] E. Ullmann, *Solution Strategies for Stochastic Finite Element Discretizations*, PhD thesis, Technischen Universitaet Bergakademie Freiberg, 2008.

[130] ——, *A Kronecker product preconditioner for stochastic Galerkin finite element discretizations*, SIAM Journal on Scientific Computing, 32 (2010), pp. 923–946.

[131] E. Ullmann and C. E. Powell, *Solving log-transformed random diffusion problems by stochastic Galerkin mixed finite element methods*, SIAM/ASA Journal on Uncertainty Quantification, 3 (2015), pp. 509 – 534.

[132] C. F. Van Loan and N. P. Pitsianis, *Approximation with Kronecker products*, in Linear Algebra for Large Scale and Real Time Applications, M. S. Moonen and G. H. Golub, eds., Kluwer Publications, Dordrecht, 1992, pp. 293–314.

[133] 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.

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

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

[136] A. J. Wathen, *On relaxation of Jacobi iteration for consistent and generalized mass matrices*, Communications in Applied Numerical Methods, 7 (1991), pp. 93 – 102.

[137] A. J. Wathen and T. Rees, *Chebyshev semi-iteration in preconditioning for problems including the mass matrix*, Electronic Transactions in Numerical Analysis, 34 (2008), pp. 125–135.

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

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

[140] I. Wiener, *The homogeneous chaos*, American Journal of Mathematics, 60 (1938), pp. 897 – 936.

[141] 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.

[142] D. Xiu and J. S. Hesthaven, *High-order collocation methods for differential equations with random inputs*, SIAM Journal on Scientific Computing, 27 (2005), pp. 1118 – 1139.

[143] D. Xiu and G. E. Karniadakis, *A new stochastic approach to transient heat conduction modeling with uncertainty*, International Journal of Heat & Mass Transfer, 46 (2003), pp. 4681–4693.

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

[145] E. L. Yip, *A note on the stability of solving a rank-p modification of a linear system by the Sherman-Morrison-Woodbury formula*, SIAM Journal on Scientific and Statistical Computing, 7 (1986), pp. 507 – 513.

[146] N. Zabaras and B. Ganapathysubramanian, *A scalable framework for the solution of stochastic inverse problems using a sparse grid collocation approach*, Journal of Computational Physics, 227 (2008), pp. 4697 – 4735.