ulm university universität
uulm

Universität Ulm
Fakultät für Mathematik und Wirtschaftswissenschaften
Institut für Numerische Mathematik

Reduced Basis Methods for Partial Differential Equations with Stochastic Influences

Dissertation zur Erlangung des Doktorgrades
Dr. rer. nat.
der Fakultät für Mathematik und Wirtschaftswissenschaften
der Universität Ulm

vorgelegt von
Bernhard Wieland
aus Stuttgart–Bad Cannstatt

April 2013

Dekan:	Prof. Dr. Dieter Rautenbach
Erstgutachter:	Prof. Dr. Karsten Urban
Zweitgutachter:	Jun.-Prof. Dr. Bernard Haasdonk
Abgabe der Doktorarbeit:	15. April 2013
Tag der Promotion:	26. Juni 2013

Abstract

This thesis is concerned with the development of reduced basis methods for parametrized partial differential equations (PPDEs) with stochastic influences. We consider uncertainties in the operator, right-hand side, boundary conditions and in the underlying domain. We are particularly interested in situations where the PPDE has to be evaluated quite often for various instances of the deterministic parameters and the stochastic influences. In the stochastic framework, such a situation occurs, e.g., in Monte Carlo simulations to compute statistical quantities such as mean, variance, or other moments.

For the efficient application of the reduced basis method, it is necessary to develop affine decompositions with respect to the stochastic influences. We therefore extend the methodology of the empirical interpolation for the application in the stochastic setting, in particular for noisy input data. Alternatively, we also use a truncated Karhunen–Loève (KL) expansion to resolve and affinely decompose the stochasticity. We derive a-posteriori error bounds for the state variable and output functionals, including also the KL-truncation errors. Non-standard dual problems are introduced for the approximation and analysis of special quadratic outputs which can in particular be applied to efficiently approximate statistical quantities such as mean and moments. We provide new error bounds for such outputs, outperforming standard approximations.

To reduce the number of affine terms and hence for the improvement of the efficiency of the reduced simulations, we generalized the partitioning concepts for explicitly given deterministic parameter domains to arbitrary input functions with possibly unknown, high-dimensional, or even without direct parameter dependencies. No a-priori information about the input is necessary.

We use all the presented methods for the application to PPDEs with stochastic influences on stochastic and additionally parametrized domains.

Zusammenfassung

Diese Arbeit befasst sich mit der Entwicklung von Reduzierten–Basis–Methoden für parametrisierte partielle Differentialgleichungen mit stochastischen Einflüssen. Diese können sowohl im Operator, in der rechten Seite, in den Randbedingungen als auch im zugrundeliegenden Gebiet auftreten. Besonders interessant im Zusammenhang mit Modellreduktion sind Problemstellungen, bei denen die Differentialgleichung für viele Realisierungen eines Parameters und der stochastischen Einflüsse gelöst werden muss. Die Berechnung von statistischen Größen wie Erwartungswert, Varianz oder Momente wird oft über Monte–Carlo Simulationen vollzogen und birgt demzufolge hohes Reduktionspotenzial.

Zur effizienten Anwendung der Reduzierten–Basis–Methoden ist es notwendig, affine Zerlegungen bezüglich der zufälligen Einflüsse zu entwickeln. Wir erweitern dazu die Methodik der empirischen Interpolation auf den stochastischen Fall, um insbesondere auch verrauschte Eingabedaten verarbeiten zu können. Alternativ betrachten wir zur Auflösung und affinen Zerlegung der Stochastizität zudem die auf endlich viele Terme begrenzte Karhunen–Loève (KL) Entwicklung. Unter Einbeziehung des Abschneidefehlers werden a-posteriori Fehlerschranken für die Zustandsgröße und die Ausgabefunktionale hergeleitet. Neben dem Erwartungswert betrachten wir insbesondere quadratische statistische Größen wie Varianz und zweites Moment. Für die Approximation solcher Ausgaben entwickeln wir neue duale Formulierungen mit deren Hilfe wir effiziente und rigorose Fehlerschranken berechnen können, die gegenüber gewöhnlichen Methoden deutliche Verbesserungen aufweisen.

Zur Reduktion der Anzahl affiner Terme und damit zur Verbesserung der Effizienz der Reduzierten–Basis–Methode verallgemeinern wir bestehende Konzepte zur Partitionierung von deterministischen Parametergebieten auf beliebige Eingabefunktionen. Sowohl Abhängigkeiten von unbekannten Parametern als auch komplett parameterunabhängige oder hochdimensionale Parameter können damit betrachtet werden, wobei keinerlei a-priori Information notwendig ist.

Die vorgestellten Methoden werden abschließend auf parametrisierte partielle Differentialgleichungen angewendet, die auf stochastischen und zusätzlich parametrisierten Gebieten definiert werden.

Contents

List of Figures

List of Tables

List of Algorithms

Symbols and Acronyms

Symbols

$\mathfrak{A}$	σ-algebra on Ω
α	coercivity constant
α_{LB}	lower bound of the coercivity constant
β	inf-sup constant
β_{LB}	lower bound of the inf-sup constant
c	coefficient function
$\mathfrak{c}$	coefficient function
$\mathbb{C}$	covariance
D	spatial domain
Δ	error bound for the primal solution
Δ_{M,M^+}	EIM error estimator for M affine terms using M^+ coefficients
Δ_{KL}	KL truncation error bound for the primal formulation
δ_{KL}	KL truncation error bound
$\tilde{\Delta}^{(\cdot)}_{\mathrm{KL}}$	KL truncation error bound for dual formulations
$\tilde{\Delta}^{(\cdot)}$	error bound for dual solutions
Δ_{RB}	RB error bound for the primal formulation
$\tilde{\Delta}^{(\cdot)}_{\mathrm{RB}}$	RB error bound for dual formulations
Δ^s	error bound for the output s
$\tilde{D}$	random spatial domain
$\mathbb{E}$	expectation
γ	continuity constant
γ_{UB}	upper bound of the continuity constant
H^1	Hilbert space of weakly differentiable functions

H_0^1	subset of H^1 with zero boundary conditions
J	number of subdomains of the parameter domain $\mathcal{P}$
J_T	continuity constant of a trilinear form
λ_k	eigenvalue of the Karhunen-Loève expansion
L_∞	space of bounded functions
$\ll$	much smaller than
$\subset\subset$	much smaller subset
L_2	space of square integrable functions
$\mathbb{M}_1$	first moment
$\mathbb{M}_2$	second moment
$\mathcal{M}$	family of input functions
μ	deterministic parameter, $\mu \in \mathcal{P}$
$\mathcal{N}$	dimension of the detailed solution
$\mathbb{N}$	set of real numbers
N	dimension of the reduced space
n_{train}	number of training snapshots
Ω	set of random events
ω	random event $\omega \in \Omega$
$\otimes$	matrix tensor product
p	dimension of the parameter domain $\mathcal{P} \subset \mathbb{R}^p$
$\mathcal{P}$	parameter domain
$\mathcal{P}^j$	subdomain of the parameter domain $\mathcal{P}$
$\mathbb{P}$	probability measure on the σ-algebra $\mathfrak{A}$
ρ_0	continuity constant of a bilinear form
ρ_1	continuity constant of a trilinear form
$\mathbb{R}$	set of real umbers
$\mathcal{S}_{\text{EIM},M}$	struct containing the EIM data for M basis functions
$\mathcal{S}_{\text{RB},N}$	struct containing the RB data for N basis functions
$\cdot_K$	subscript or superscript indicating KL truncations
$\cdot_N$	subscript or superscript indicating reduced quantities
$\mathfrak{u}$	arbitrary function in X
$\mathbb{V}$	variance
X	Hilbert space

ξ	random variable with zero mean and unit variance
Ξ_{train}	set of training snapshots
$\tilde{X}_N^{(i)}$	reduced space for the ith dual problem
X_N	reduced space for the primal problem

Acronyms

BRR	Brezzi-Rappaz-Raviart
cf.	confer (compare)
DEIM	Discrete Empirical Interpolation Method
e.g.	exempli gratia (for example)
EIM	Empirical Interpolation Method
FE	Finite Element
FEM	Finite Element Method
FS	Fixed Shapes
FS IPM	Fixed Shapes Implicit Partitioning Method
i.e.	id est (that is)
IPM	Implicit Partitioning Method
KL	Karhunen–Loève
LSEIM	Least–Squares Empirical Interpolation Method
MC	Monte Carlo
MS	Moving Shapes
MS IPM	Moving Shapes Implicit Partitioning Method
OEIM	Operator Empirical Interpolation Method
PC	Polynomial Chaos
PDE	Partial Differential Equation
POD	Proper Orthogonal Decomposition
POIM	Proper Orthogonal (Empirical) Interpolation Method
PPDE	Parametrized Partial Differential Equation
RB	Reduced Basis
RBM	Reduced Basis Method
SPDE	Stochastic Partial Differential Equation
SVD	Singular Value Decomposition

Chapter 1

Introduction

Reduced basis methods for partial differential equations with stochastic influences?

1.1 Motivation

Several problems in science, medicine, economics and engineering are modeled by partial differential equations (PDEs). Often, such models contain uncertainties in terms of imprecise, unknown, or stochastic input. One could think for example of coefficients of the PDE that are based upon inaccurate or noisy measurements. Furthermore, even the underlying spatial domain may be obtained by defective measurements, e.g., by scanning or X-raying. Especially for sensitive systems, it may be of interest to simulate how a small perturbation of the input influences the solution, e.g., to determine tolerances for the accuracy of measurements or to derive requirements for actual mechanical implementations. Also, unknown spatial coefficients are often modeled stochastically. Examples include the porosity structure of Li-ion batteries, fuel cells, or the modeling of groundwater flows. Another application is given by inverse problems, where for given (measured) outputs, the distribution of a random input parameter is desired. Eventually, it is quite common in financial mathematics to model unknown data using stochastic processes, e.g., mortality rates for life insurance simulations or the market price behavior for risk analysis in the banking sector. Generally spoken, uncertainty or randomness is more or less everywhere.

In addition to such uncertainties, many problems also depend on a number of

deterministic parameters, i.e., one has a parametrized PDE (PPDE). Examples include model parameters such as material properties, parametric geometries, or forces. We are particularly interested in situations where the PPDE with stochastic influences has to be evaluated quite often for various instances of the deterministic parameters and the stochastic influences. In the stochastic framework, such a situation occurs, e.g., in Monte Carlo simulations to compute statistical quantities such as mean, variance, or other moments. For the deterministic parameters, one might think of parameter studies or optimization. Such a many-query situation requires the numerical solution of the PDE for many instances of the parameter and stochastic influence, which is infeasible in particular for more complex PDEs. Hence, model reduction is desired.

It should be noted that we are *not* concerned with stochastic PDEs involving the Itô calculus. This is the reason why we use the term *PDEs with stochastic influences*, even though this might be a bit lengthy.

1.2 The Reduced Basis Method

The reduced basis method (RBM) has intensively been studied for the numerical solution of PPDEs. One basic idea is an offline-online decomposition combined with a rigorous a-posteriori error control.

In the offline stage, computationally expensive evaluations are performed. The reduced basis (RB) is formed by solving the complex PPDE for certain parameter values, so-called snapshots. The selection is usually based upon a Greedy algorithm [14, 73, 98]. Basically, the snapshot that corresponds to the largest error bound is selected for the basis extension. The detailed solutions are obtained using a fine discretization, e.g., finite elements, finite differences, or finite volumes. Hence, high-dimensional systems have to be solved. The error bounds can be used to control the size of the reduced model.

For a new parameter, the reduced system is then used in the online stage for a highly efficient simulation. The dimension of the system reduces to the number of selected snapshots. The error bounds confirm the approximation quality in the reduced setting.

Hence, besides so-called multi-query problems, where solutions of a PPDE have

to be evaluated repeatedly for different parameter values, a typical application of the RBM is given by real-time settings. In such cases, even very high offline costs can be accepted.

For the efficiency of the RBM, it is required that the problem allows for an affine decomposition with respect to the parameter, i.e., for a separation of spatial and parametric terms. Since many problems do not naturally show such properties, the empirical interpolation method (EIM) has been developed to generate affine approximations of the coefficients of the PDE [7, 86] or directly of arbitrary differential operators [19, 20, 27, 43]. Certainly, the additional approximation error has to be considered and included in the analysis.

The RBM has been studied for wide classes of problems and many applications have been developed in the recent past by a growing number of researchers. Besides linear elliptic [73, 77] and parabolic equations [40, 76], also more complicated quadratically nonlinear problems have been reduced [24, 86, 96]. In the latter case, the error analysis is based upon the well-known Brezzi-Rappaz-Raviart theory [13, 16]. Furthermore, for special classes of coupled systems, e.g., saddle point problems and especially the Stokes equations, RB theory has been introduced [33, 34, 75]. A lot of work has been done on RBMs for problems on parametrized geometries for several different applications, e.g., [33, 63, 78, 88], to mention just a few. Additionally, the RBM can be used for both parameter optimization [26] and parameter dependent optimal control problems [39, 59]. Recently, it has been started to consider also RBMs for parametrized variational inequalities [44]. So far, the RBM can be efficiently applied only to stationary inequalities. However, work is going on to extend the results to instationary problems for the application to option pricing in financial mathematics. In this context, RBMs based upon weak formulations in space and time are considered, which have already been successfully applied to time-periodic problems [83]. Also for other instationary problems, it could be shown that such formulations lead to additional reduction capabilities [90, 106].

Furthermore, to improve the efficiency of the RBM, several different domain decomposition methodologies have been developed. While for the parameter domain partitioning, it is enough to generate separate reduced bases on each subdomain [30, 31, 41], the processing of separated time domains [25] and spatial domains

[28, 53, 67, 94] requires special treatments at the intersecting boundaries. Also, alternatives to the Greedy basis selection have been developed such that the basis can be adapted to the current parameter in the online stage [61].

So far, not much work on RBMs regarding stochastic problems has been done. In [12], a specific problem with stochastic Robin-type boundary conditions is studied. However, the analysis presented there does not cover the case of general stochastic influences, e.g., in terms of random spatial coefficients. In this sense, the present work will generalize and extend the findings in [12].

For the sake of completeness, let us also mention some further related work. In [11], an RB control variate technique for variance reduction is introduced. Furthermore, the terminology "reduced basis" is also used in the context of model reduction via Krylov subspaces, e.g., in [70, 79], also for stochastic problems.

1.3 Objective

The aim of this work is to develop a general framework of reduced basis methods for PPDEs with stochastic influences that can be applied to wide classes of problems. We want to consider both linear and non-linear problems, in particular with a focus on quadratic non-linearities. The methods are meant to deal with various instances of uncertainties, including stochastic influences

(a) in the coefficients of the PDE, i.e., in the operator,

(b) in the right-hand side,

(c) in the boundary conditions, and

(d) in the domain.

Furthermore, in the context of randomness and uncertainties, one often depends on noisy input data.

For the efficient application of the RBM, it is necessary to develop affine decompositions with respect to the random input, i.e., spatial and random influences have to be separated. Therefore, one objective of the thesis is to generalize the EIM methodology for the application to stochastic and noisy data. Alternatively, to make use of the specific properties of stochastic inputs, it is also desired to connect the RBM with the Karhunen–Loève (KL) expansion and polynomial chaos.

A main issue of problems with random input is the approximation of linear and non-linear statistical outputs such as mean, variance, and other moments. Hence, besides the approximation of linear output functionals, we want to focus on the development of RBMs that are in particular adapted to the approximation of such statistical quantities.

1.4 Outline of the Work

We start with an introduction of different known techniques to solve PDEs with stochastic influences in Chapter 2. These methods will serve as detailed solutions underlying the RBM. We focus on two different classes: formulations weak in space and strong in probability such as Monte Carlo methods as well as formulations weak in space and probability, e.g., stochastic Galerkin methods and stochastic collocation methods. For a simple illustrative problem, we provide the necessary ingredients for the modeling of the stochasticity and the application of the methods, namely the Karhunen–Loève expansion and polynomial chaos.

In Chapter 3, we consider the construction of affine decompositions with respect to deterministic parameters *and* random influences. After a short introduction about the application of affine decompositions in the context of the RBM, we generalize the framework of the EIM to the stochastic case and consider in particular approximations in the presence of noise. The proper orthogonal decomposition (POD) is applied on the given input data. We show that the replacement of the usual EIM basis, using now the POD eigenmodes, leads to improved affine approximations in mean-squared sense. Connections of the method to the so-called discrete EIM (DEIM) are derived and we show that we obtain the same results with less run-time complexity, allowing now to apply the EIM error estimators to both methods. In a second step, we introduce a least–squares EIM that uses more knots than basis functions. We show that the method generates close to optimal affine approximations.

It is common to partition parameter domains and construct separate reduced bases on each subdomain for more efficient online simulations. Known partitioning methods relate on explicit descriptions of compact parameter domains. In Chapter 4, we develop implicit partitioning methods (IPMs) that can be applied to all

classes of parameter domains, even to unknown parameters or non-parametric input data. We develop three different approaches, all connected to the EIM. For two of the methods, the partitioning is based upon the EIM approximation error, where for the first method, the partition also depends on the number of used affine terms for the EIM approximation. The subdividing scheme of third method relates on the EIM coefficients and enables tree based assignment procedures. We provide several examples and demonstrate that, applied to compact parameter domains, the IPMs generate better results than explicit partitioning methods for wide classes of problems.

In Chapter 5, we develop the RBM for linear PPDEs with stochastic influences in coefficients, right-hand side, and boundary conditions. We assume the availability of an affine decomposition with respect to the deterministic parameter and apply the KL expansion to the stochastic terms. We develop error bounds for the state variable and for linear random outputs that also take the KL truncation error into account. Using additional non-standard dual problems, we can also derive good approximations and error bounds for nonlinear statistical outputs such as second moment and variance. We show that the approach can, to some extend, also be applied to higher moments. We can furthermore derive that parts of the KL truncation error do not influence the RB approximation of the statistical outputs such that the developed bounds clearly outperform direct approaches. We illustrate the results using an example of heat transfer in a two-dimensional porous medium, where the porosity and the boundary conditions are modeled using spatial stochastic processes.

In Chapter 6, the results of Chapter 5 are generalized to quadratically nonlinear problems. It is shown that the error analysis, especially of the statistical outputs, can be adopted in a very similar form. The used dual formulations remain linear such that the complexity for the dual solutions correspond to just one Newton iteration of the primal problem. Hence, the improved error bounds become highly efficient. We demonstrate this effect for the example of a convection-diffusion problem in a porous medium.

Chapter 7 combines the results of Chapters 3 to 6 for the application of stochastic PPDEs on stochastic, parametric domains. Using a diffeomorphic mapping from a fixed reference domain to the original domain, we show how the problem

can be transformed such that all parametric and stochastic dependencies are contained in the coefficients of the PDE. Two known methods to build such mappings are described and compared. For two different cases, purely stochastic domains as well as stochastic *and* parametric domains, we show how the RBM can be applied. In the first case, we use a special form of the KL expansion to derive that the RBMs of Chapters 5 and 6 can be used. We furthermore provide a method to apply the IPMs of Chapter 4, maintaining still the good approximation results of the KL expansion. In the case of stochastic and parametric domains, we show that the EIM can be used in combination with techniques known from deterministic problems. Naturally, the IPM can also be applied. We illustrate the different approaches using the example of a plate where a random hole appears on the bottom side.

Finally, in Chapter 8, we briefly describe further applications of the presented methods to instationary problems as well as to formulations weak in space and probability. Additionally, areas of future research are provided.

Chapter 3 is based upon joint work with K. Urban and the main results have already been published in [92] in a very similar form. We added a section about affine decompositions in the context of the RBM.

Chapter 5 is based upon joint work with B. Haasdonk and K. Urban and the main results have already been published in [45] in a very similar form. We added sections about higher moments, non-coercive problems, and showed that some assumptions regarding stochastic independence can be weakened such that more general classes of problems can be considered.

Chapter 6 is based upon joint work with K. Urban and the main results have already been published in [93] in a very similar form. We showed that some assumptions regarding stochastic independence can be weakened such that more general classes of problems can be considered.

Chapter 2

Solutions of PDEs with Stochastic Influences

Popular techniques to solve PDEs with stochastic influences include perturbation methods [64, 95] and second order analysis [48, 50]. Both methods are based upon an expansion of the random quantities in a Taylor series about their respective mean values. Hence, good results can only be obtained for small perturbations, i.e., under specific smoothness conditions of the uncertain behavior.

Another technique is the Neumann series approach, where the inverse of the uncertain operator is approximated by its Neumann series [4, 35]. For example, the method has been applied to examine the response variability arising from spatially uncertain material properties [80, 105].

In this chapter, we introduce two different solution concepts for PDEs with stochastic influences. Formulations weak in space and strong in probability are considered as well as formulations weak in space *and* probability. We start with a simple example in Section 2.1 that will be used to illustrate the general problem and the different techniques. In Sections 2.2 and 2.3, we introduce the Karhunen–Loève (KL) and the Polynomial Chaos (PC) expansion, respectively, that are used for the modeling of the stochastic influences, i.e., of spatial stochastic processes. In Section 2.4, we briefly describe the Monte Carlo (MC) method as an example of the weak-strong concept. In Sections 2.5 and 2.6, we introduce two methods based upon the weak-weak formulation, stochastic Galerkin methods and stochastic collocation methods.

2.1 Mathematical Formulation

2.1.1 Model Problem

Let $(\Omega, \mathfrak{A}, \mathbb{P})$ be a probability space, where Ω denotes a set of elementary events, $\mathfrak{A}$ a σ-algebra on Ω and $\mathbb{P}$ a probability measure on $\mathfrak{A}$, and let $D \subset \mathbb{R}^d$ denote a bounded spatial domain. Furthermore, let c denote a real-valued second order spatial stochastic process, i.e., $c : D \times \Omega \to \mathbb{R}$, $(x;\omega) \mapsto c(x;\omega)$. For each $\omega \in \Omega$, the trajectory $c(\omega) := c(\cdot;\omega) : D \mapsto \mathbb{R}$ is supposed to be in $L_2(D)$. We assume the existence of constants $c^-, c^+ \in \mathbb{R}$, independent of x and ω, such that $0 < c^- \leq c(x;\omega) \leq c^+ < \infty$. Hence, for any bounded spatial stochastic process $d : D \times \Omega \to \mathbb{R}$ with $d(\omega) := d(\cdot;\omega) \in L_2(D)$, we consider the linear elliptic problem,

$$\begin{cases} -\nabla \cdot \big(c(x;\omega)\, \nabla u(x;\omega)\big) &= d(x;\omega) \quad \text{in } D, \\ \quad u(x;\omega) &= 0 \qquad \text{on } \partial D. \end{cases} \tag{2.1}$$

The coefficient $c(x;\omega)$ my describe the random diffusivity or conductivity of the underlying system. Then, the solution $u(x;\omega)$ of the PDE denotes the corresponding concentration or temperature.

2.1.2 D-weak/Ω-strong Formulation

In weak or variational formulations, PDEs and their solutions are not considered pointwise, as it would be the case using strong formulations. Instead, both the operator and the right-hand side are multiplied by some test function in a previously specified test space. Then, the integral over the given domain is considered. A solution is called weak if it solves this integral formulation of the problem for all test functions.

In this section, we consider solutions that are weak in space but strong in probability. Hence, the integral in the variational formulation is taken only over the spatial domain D and the solutions are considered pointwise in Ω. For each realization of the underlying stochastic processes c and d, we obtain a respective deterministic spatial variational problem. Exemplarily, we now derive the variational formulation of (2.1) and provide existence and uniqueness results.

Let us consider the Hilbert space $H^1(D) \subset L_2(D)$ on the spatial domain D with the inner product

$$(w, v)_{H^1} = \int_D w(x)v(x) + \nabla w(x) \cdot \nabla v(x)\, dx,$$

and let us denote the subspace of functions in $H^1(D)$ vanishing in the trace sense at the boundary of D by $H_0^1 := H_0^1(D) := \{v \in H^1(D) \,|\, v = 0 \text{ on } \partial D\}$. Furthermore, let the bilinear form $a : H_0^1 \times H_0^1 \times \Omega \to \mathbb{R}$ be defined by

$$a(w, v; \omega) := \int_D c(x; \omega)\nabla w(x) \cdot \nabla v(x)\, dx. \tag{2.2}$$

The bilinear form a is uniformly coercive and uniformly continuous for all $\omega \in \Omega$, i.e., there are constants $\alpha_0 > 0$ and $\gamma_\infty < \infty$ such that

$$\alpha(\omega) := \inf_{v \in H_0^1} \frac{a(v, v; \omega)}{\|v\|_{H^1}^2} \geq \alpha_0, \quad \text{(uniform coercivity)},$$

$$\gamma(\omega) := \sup_{w \in H_0^1} \sup_{v \in H_0^1} \frac{a(w, v; \omega)}{\|w\|_{H^1}\|v\|_{H^1}} \leq \gamma_\infty, \quad \text{(uniform continuity)}.$$

This can be easily shown using the Poincaré inequality [2] and the fact that $c(x; \omega)$ is strictly positive and bounded from above and below by constants independent of x and ω. Next, we define the linear form $f : H_0^1 \times \Omega \to \mathbb{R}$ by

$$f(v; \omega) := \int_D d(x; \omega)v(x)\, dx. \tag{2.3}$$

Since $d(\omega) \in L_2(D)$ for all $\omega \in \Omega$, f is bounded, i.e., f is continuous.

The D-weak/Ω-strong formulation of (2.1) is now given as follows. For any random event $\omega \in \Omega$, find $u(\omega) \in H_0^1$ such that

$$a(u(\omega), v; \omega) = f(v; \omega), \quad \forall v \in H_0^1(D). \tag{2.4}$$

Hence, using the D-weak/Ω-strong formulation, the PDE can be solved separately for any realization $\omega \in \Omega$. In some way, one could consider these solutions as "pointwise" in Ω.

It remains to provide results concerning the existence and uniqueness of solutions of (2.4).

Proposition 2.1. *For each $\omega \in \Omega$, the variational problem* (2.4) *admits a unique solution $u(\omega) \in H_0^1$ depending continuously on the right-hand side f.*

Proof. The result follows from Lax-Milgram Theorem, using the uniform coercivity and the uniform continuity of a as well as the continuity of f [2]. □

2.1.3 D-weak/Ω-weak Formulation

In this section, we consider formulations weak in space and probability. Let us first denote the space of all square integrable random variables on Ω by $L_2(\Omega)$. We define the corresponding inner product of two random variables $\xi, \eta \in L_2(\Omega)$ by their correlation, i.e.

$$(\xi, \eta)_{L_2(\Omega)} := \mathbb{E}[\xi\,\eta] := \int_\Omega \xi(\omega)\eta(\omega)\mathbb{P}(d\omega).$$

For D-weak/Ω-weak formulations, we now consider the tensor product Hilbert space $H_0^1(D) \otimes L_2(\Omega)$. The inner product on $H_0^1(D) \otimes L_2(\Omega)$ is given by

$$\begin{aligned}(w, v)_{H_0^1(D)\otimes L_2(\Omega)} &= \mathbb{E}\left[\,(w, v)_{H^1}\right] \\ &= \int_\Omega \int_D w(x;\omega)v(x;\omega) + \nabla w(x;\omega)\cdot\nabla v(x;\omega)\,dx\,\mathbb{P}(d\omega).\end{aligned}$$

Similar to the D-weak/Ω-strong case, we define a bilinear form $a : (H_0^1(D) \otimes L_2(\Omega)) \times (H_0^1(D) \otimes L_2(\Omega)) \to \mathbb{R}$ and a linear form $f : H_0^1(D) \otimes L_2(\Omega) \to \mathbb{R}$ by

$$a(w, v) := \mathbb{E}\left[\int_D c(x;\cdot)\nabla w(x;\cdot)\cdot\nabla v(x;\cdot)\,dx\right], \tag{2.5a}$$

$$f(v) := \mathbb{E}\left[\int_D d(x;\cdot)v(x;\cdot)\,dx\right], \tag{2.5b}$$

respectively. Using again the positivity and boundedness of c as well as the Poincaré inequality, it can easily be shown that a is coercive and continuous in the D-weak/Ω-weak sense. In other words, we have

$$\alpha := \inf_{v\in H_0^1(D)\otimes L_2(\Omega)} \frac{a(v, v)}{\|v\|^2_{H_0^1(D)\otimes L_2(\Omega)}} > 0,$$

$$\gamma := \sup_{w,v\in H_0^1(D)\otimes L_2(\Omega)} \frac{a(w, v)}{\|w\|_{H_0^1(D)\otimes L_2(\Omega)}\|v\|_{H_0^1(D)\otimes L_2(\Omega)}} < \infty.$$

The forms a and f no longer depend on realizations ω since the uncertainties are implied in the Hilbert space $H_0^1(D) \otimes L_2(\Omega)$, i.e., in the arguments of a and f. Consequently, the coercivity and continuity constants are as well independent of specific realizations and we do not need a "uniform" definition. Furthermore, it is clear that f is again continuous.

The D-weak/Ω-weak formulation of (2.1) is given as follows. Find $u \in H_0^1(D)\otimes L_2(\Omega)$ such that

$$a(u, v) = f(v), \quad \forall v \in H_0^1(D) \otimes L_2(\Omega). \tag{2.6}$$

Hence, the solution of (2.6) contains the complete uncertainty information. It is possible to directly evaluate statistical quantities such as mean and correlations. Additionally, solutions of specific realizations can still be evaluated. However, we will see in Sections 2.4 and 2.5 that solutions of (2.6) can be computationally very expensive, also compared to the complexity of multiple solutions of (2.4).

We close the section providing results concerning the existence and uniqueness of solutions of (2.6).

Proposition 2.2. *The variational problem* (2.6) *admits a unique solution* $u \in H_0^1 \otimes L_2(\Omega)$ *depending continuously on the right-hand side* f.

Proof. Follows from Lax-Milgram Theorem, using the coercivity and the continuity of a as well as the continuity of f [2]. □

2.2 Karhunen–Loève Expansion

The approximation of spatial or time-dependent stochastic processes with high accuracy requires the sampling at many points in space or time and increases the computational costs, e.g., of Monte Carlo methods. The objective of the Karhunen–Loève (KL) expansion is the separation of random and spatial or time dependencies. This facilitates not only the sampling procedure but is also a key requirement of solution procedures for PDEs with stochastic influences such as stochastic finite elements.

In this section, we introduce the main concept of the KL expansion. It has been investigated separately by K. Karhunen [60] and M. Loève [65] and is closely connected to Proper Orthogonal Decomposition (POD) [62] or Singular Value Decomposition (SVD). We first introduce the theoretical concept of the KL expansion and of the KL truncation error. For this part, we follow the concept of [35]. However, we generalize the results from an L_2-based formulation to arbitrary Hilbert spaces. We then introduce the so-called method of snapshots that provides a procedure to efficiently construct the KL expansion without the knowledge of a covariance function, using just a finite number of random samples [82, 47]. Finally, we generalize the results of the first two sections to obtain joint KL expansions for vector-valued processes or of several scalar-valued but correlated processes. We follow the concept of [46], generalizing again to arbitrary Hilbert spaces.

2.2.1 Theoretical Aspects

As in Section 2.1.1, let $(\Omega, \mathfrak{A}, \mathbb{P})$ be a probability space and let $D \subset \mathbb{R}^d$ denote a spatial domain. For some appropriate Hilbert space X on D with inner product $(\cdot,\cdot)_X$, let $c : D \times \Omega \to \mathbb{R}$ now denote a second order real-valued spatial stochastic process with trajectories $c(\omega) \in X = X(D)$ for each $\omega \in \Omega$. We split c into its expectation $\bar{c}(x) := \mathbb{E}[c(x;\cdot)]$ and a fluctuating part $\tilde{c}(x,\omega) = c(x;\omega) - \bar{c}(x)$ such that $\mathbb{E}[\tilde{c}(x;\cdot)] = 0$, i.e.,

$$c(x;\omega) = \bar{c}(x) + \tilde{c}(x;\omega). \tag{2.7}$$

Furthermore, let c be a second order process, i.e., square integrable with respect to the probability measure $\mathbb{P}$. Then, its covariance function

$$\mathbb{C}(x_1, x_2) := \mathbb{E}[\tilde{c}(x_1;\cdot)\tilde{c}(x_2;\cdot)] \tag{2.8}$$

is bounded by the Cauchy-Schwarz inequality and symmetric positive definite. Hence, the eigenvalues λ_k, $k \in \mathbb{N}$, of the covariance integral kernel $T : X \to X$,

$$(Tv)(x) := (\mathbb{C}(x,\cdot), v)_X, \quad v \in X, \tag{2.9}$$

are strictly positive and the corresponding eigenfunctions $c_k \in X$, $k \in \mathbb{N}$, can be orthonormalized such that $(c_k, c_l)_X = \delta_{k,l}$, where δ denotes the Kronecker delta. The subsequent theorem provides a decomposition of the stochastic process in the desired form.

Theorem 2.3 (Karhunen–Loève Expansion)**.** *Let $c(x;\omega)$, $\mathbb{C}(x_1,x_2)$ and $(Tv)(x)$ be as defined in (2.7), (2.8), and (2.9), respectively. Then, it holds that*

$$\tilde{c}(x;\omega) = \sum_{k=1}^{\infty} \sqrt{\lambda_k}\xi_k(\omega)c_k(x), \tag{2.10}$$

where λ_k and c_k, $k \in \mathbb{N}$, denote the eigenvalues and eigenfunctions of T, respectively, and ξ_k, $k \in \mathbb{N}$, are uncorrelated random variables with zero mean and unit variance. They are given by

$$\xi_k(\omega) := \frac{1}{\sqrt{\lambda_k}} (\tilde{c}(\cdot;\omega), c_k)_X . \tag{2.11}$$

Proof. Using the spectral theorem, it is clear that $\tilde{c}(x;\omega)$ can be expanded as a linear combination of the eigenfunctions. Hence, we can assume that $\tilde{c}(x;\omega)$ is of

the form of (2.10) and it remains to show that the random variables ξ_k fulfill the proposed properties. Using (2.10), the covariance can be written as

$$\begin{aligned}\mathbb{C}(x_1, x_2) &= \mathbb{E}\left[\tilde{c}(x_1;\cdot)\tilde{c}(x_2;\cdot)\right] \\ &= \sum_{k=1}^{\infty}\sum_{l=1}^{\infty}\mathbb{E}\left[\xi_k(\cdot)\xi_l(\cdot)\right]\sqrt{\lambda_k\lambda_l}c_k(x_1)c_l(x_2).\end{aligned}$$

We use this form of the covariance function and apply the operator T on the eigenfunction c_n. The orthonormality of the eigenfunction yields

$$\begin{aligned}\lambda_n c_n(x) = (Tc_n)(x) &= (\mathbb{C}(x,\cdot), c_n)_X \\ &= \sum_{k=1}^{\infty}\sum_{l=1}^{\infty}\mathbb{E}\left[\xi_k(\cdot)\xi_l(\cdot)\right]\sqrt{\lambda_k\lambda_l}c_k(x)\,(c_l, c_n)_X \\ &= \sum_{k=1}^{\infty}\mathbb{E}\left[\xi_k(\cdot)\xi_n(\cdot)\right]\sqrt{\lambda_k\lambda_n}c_k(x).\end{aligned}$$

Taking the inner product of both left-hand and right-hand side with the eigenfunction c_m yields

$$\begin{aligned}\lambda_n(c_n, c_m)_X = \lambda_n\delta_{m,n} &= \sum_{k=1}^{\infty}\mathbb{E}\left[\xi_k(\cdot)\xi_n(\cdot)\right]\sqrt{\lambda_k\lambda_n}(c_k, c_m)_X \\ &= \mathbb{E}\left[\xi_m(\cdot)\xi_n(\cdot)\right]\sqrt{\lambda_m\lambda_n}.\end{aligned}$$

Since $\lambda_k > 0$, $k \in \mathbb{N}$, we directly obtain $\mathbb{E}\left[\xi_m(\cdot)\xi_n(\cdot)\right] = \delta_{m,n}$. Hence, all random variables ξ_k are uncorrelated and have unit variance. Considering the expectation of $\tilde{c}$ which is known to be zero,

$$\mathbb{E}[\tilde{c}(x;\cdot)] = \sum_{k=1}^{\infty}\sqrt{\lambda_k}\mathbb{E}[\xi_k(\cdot)]c_k(x) = 0,$$

we obtain $\mathbb{E}[\xi_k] = 0$. To show (2.11), we consider the inner product of $\tilde{c}$ with the eigenfunction c_l.

$$(\tilde{c}(\cdot;\omega), c_l)_X = \sum_{k=1}^{\infty}\sqrt{\lambda_k}\xi_k(\omega)(c_k, c_l)_X = \sqrt{\lambda_l}\xi_l(\omega)$$

which proves the claim. □

For numerical purposes, one needs a finite approximation of the KL expansion. We assume that the eigenvalues are sorted in descending order, i.e., $\lambda_1 \geq \lambda_2 \geq \ldots$, and truncate the series after K terms. The truncation error is denoted by

$$\varepsilon_K(x;\omega) := \sum_{k=K+1}^{\infty} \sqrt{\lambda_k}\xi_k(\omega)c_k(x).$$

It is straightforward to derive the mean squared truncation error as the sum over the remaining eigenvalues,

$$\mathbb{E}\left[\, \|\varepsilon_K\|_X^2 \,\right] = \sum_{k=K+1}^{\infty}\sum_{l=K+1}^{\infty} \sqrt{\lambda_k\lambda_l}\mathbb{E}\left[\xi_k\xi_l\right](c_k,c_l)_X = \sum_{k=K+1}^{\infty} \lambda_k. \tag{2.12}$$

2.2.2 Method of Snapshots

In many cases, the covariance function $\mathbb{C}$ is not given analytically and it has to be approximated by Monte Carlo procedures. Let $c(\cdot,\omega_n)$, $1 \leq n \leq n_{\text{train}}$, be n_{train} instances of the stochastic process. Then, one uses the Monte Carlo approximation

$$\mathbb{C}_{\text{MC}}(x_1,x_2) := \frac{1}{n_{\text{train}}}\sum_{n=1}^{n_{\text{train}}} \tilde{c}(x_1;\omega_n)\tilde{c}(x_2;\omega_n). \tag{2.13}$$

We define the covariance operator T_{MC} analogously to (2.9), using $\mathbb{C}_{MC}$ instead of $\mathbb{C}$.

In discretized form, the covariance functions $\mathbb{C}$ and $\mathbb{C}_{\text{MC}}$ can be represented by positive (semi-)definite matrices which we denote covariance matrices. Then, the evaluation of T and T_{MC} reduces to a matrix-vector product. Let $\mathcal{N}$ be the number of degrees of freedom of the discretization. If n_{train} is smaller than $\mathcal{N}$, the rank of the $\mathcal{N}$-dimensional covariance matrix is at most n_{train} and the method of snapshots [82] provides an alternative procedure to evaluate the non-zero eigenvalues and the corresponding eigenfunctions. We define the n_{train}-dimensional matrix $\widehat{\mathbb{C}} = (\widehat{\mathbb{C}}_{n,m})_{n,m=1}^{n_{\text{train}}}$ by

$$\widehat{\mathbb{C}}_{n,m} := \frac{1}{n_{\text{train}}}\left(\tilde{c}(\cdot;\omega_n),\tilde{c}(\cdot;\omega_m)\right)_X \tag{2.14}$$

and denote its eigenvalues by $\hat{\lambda}_k$ with corresponding ℓ_2-orthonormalized eigenvectors $\boldsymbol{v}_k \in \mathbb{R}^{n_{\text{train}}}$, $k = 1,\ldots,n_{\text{train}}$. The ith component of $\boldsymbol{v}_k$ is denoted by $v_k^{(i)}$. We define the functions

$$\hat{c}_k(x) := \sum_{n=1}^{n_{\text{train}}} v_k^{(n)}\tilde{c}(x;\omega_n), \tag{2.15}$$

$k = 1, \ldots, n_{\text{train}}$, and show that $\hat{\lambda}_k$ coincide with the non-zero eigenvalues of $\mathbb{C}_{MC}$, where $\hat{c}_k$ denote the corresponding eigenfunctions. We evaluate the covariance operator T_{MC} at $\hat{c}_k$. Using the definitions of $\mathbb{C}_{\text{MC}}$ in (2.13) and of $\hat{c}_k$ in (2.15), we obtain

$$\begin{aligned}
\left(T_{\text{MC}}\hat{c}_k\right)(x) &= \left(\mathbb{C}_{\text{MC}}(x,\cdot), \hat{c}_k\right)_X \\
&= \left(\frac{1}{n_{\text{train}}} \sum_{n=1}^{n_{\text{train}}} \tilde{c}(x;\omega_n)\tilde{c}(\cdot;\omega_n), \sum_{m=1}^{n_{\text{train}}} v_k^{(m)} \tilde{c}(\cdot;\omega_m) \right)_X \\
&= \sum_{n=1}^{n_{\text{train}}} \tilde{c}(x;\omega_n) \sum_{m=1}^{n_{\text{train}}} v_k^{(m)} \frac{1}{n_{\text{train}}} \left(\tilde{c}(\cdot;\omega_n), \tilde{c}(\cdot;\omega_m)\right)_X .
\end{aligned}$$

The latter part of the right-hand side is just the definition of $\widehat{\mathbb{C}}_{n,m}$ as introduced in (2.14). Using the eigenvalue properties of $\widehat{\mathbb{C}}$ yields

$$\begin{aligned}
\left(T_{\text{MC}}\,\hat{c}_k\right)(x) &= \sum_{n=1}^{n_{\text{train}}} \tilde{c}(x;\omega_n) \sum_{m=1}^{n_{\text{train}}} \widehat{\mathbb{C}}_{n,m} v_k^{(m)} \\
&= \sum_{n=1}^{n_{\text{train}}} \tilde{c}(x;\omega_n) \hat{\lambda}_k v_k^{(n)} \\
&= \hat{\lambda}_k \hat{c}_k .
\end{aligned}$$

Hence, $\hat{\lambda}_k$ is eigenvalue and $\hat{c}_k$ eigenfunction of T_{MC}. It is easy to show that the eigenfunctions $\hat{c}_k$ are orthogonal. The inner product is given by

$$\left(\hat{c}_k, \hat{c}_l\right)_X = \sum_{n=1}^{n_{\text{train}}} \sum_{m=1}^{n_{\text{train}}} v_k^{(n)} v_l^{(m)} \left(\tilde{c}(\cdot;\omega_n), \tilde{c}(\cdot;\omega_n)\right)_X$$

and using the definition of $\hat{\mathbb{C}}$ and its eigenvalue properties, we obtain

$$\begin{aligned}
\left(\hat{c}_k, \hat{c}_l\right)_X &= n_{\text{train}} \sum_{n=1}^{n_{\text{train}}} \sum_{m=1}^{n_{\text{train}}} v_k^{(n)} \hat{\mathbb{C}}_{n,m} v_l^{(m)} \\
&= \lambda_l \, n_{\text{train}} \, \boldsymbol{v}_k^T \boldsymbol{v}_l \;=\; \lambda_l \, n_{\text{train}} \delta_{k,l}.
\end{aligned}$$

Hence, after normalization and sorting the eigenvalues in descending order, we obtain

$$c_k = \frac{1}{\sqrt{\lambda_k n_{\text{train}}}} \hat{c}_k .$$

As a consequence, it is possible to obtain the relevant eigenvalues and eigenfunctions of the covariance operator by solving only a smaller problem. The remaining

eigenvectors that correspond to zero eigenvalues do not contain important information for the representation of the stochastic since the mean squared truncation error from (2.12), using just the first n_{train} eigenfunctions, is obviously zero.

2.2.3 Multi–Component KL Expansion

So far, we introduced the KL expansion for scalar functions. Often, it is desirable to generate expansions also for vector-valued processes [46]. Let X^r, $r \in \mathbb{N}$, denote the space of r-dimensional functions, where each component is a function in X. For $\boldsymbol{c}, \boldsymbol{d} \in X^r$, let

$$(\boldsymbol{c}, \boldsymbol{d})_{X^r} = \sum_{j=1}^{r} (c^j, d^j)_X$$

be the inner product on X^r, where c^j denotes the jth component of $\boldsymbol{c}$. Now, for some r-dimensional random process $\boldsymbol{c} : D \times \Omega \to \mathbb{R}^r$, $\boldsymbol{c}(\cdot;\omega) \in X^r$, with $\bar{\boldsymbol{c}}(x) = \mathbb{E}[\boldsymbol{c}(x;\cdot)]$ and $\tilde{\boldsymbol{c}}(x;\omega) = \boldsymbol{c}(x;\omega) - \bar{\boldsymbol{c}}(x)$, the objective is to generate expansions analogously to (2.10),

$$\tilde{\boldsymbol{c}}(x;\omega) = \sum_{k=1}^{\infty} \sqrt{\lambda_k}\xi_k(\omega)\boldsymbol{c}_k(x), \tag{2.16}$$

where all $\boldsymbol{c}_k \in X^r$ are orthonormal with respect to $(\cdot,\cdot)_{X^r}$ and $\xi_k(\omega)$ are real scalar-valued random variables with zero mean and unit variance. Such an expansion can also be useful to jointly model several scalar-valued but correlated processes. Then, the correlation is already included in the expansion and one does not need any further processing of the different random variables of each process. Furthermore, the total number of terms needed for a good approximation of the processes may be smaller for the resulting joint expansion than for separate expansions.

For the construction of the multi-component KL expansion, we define the covariance function $\mathbb{C}$ similarly to (2.8) by

$$\begin{aligned} \mathbb{C}^{i,j}(x_1, x_2) &:= \mathbb{E}\left[\tilde{c}^i(x_1;\cdot)\tilde{c}^j(x_2;\cdot)\right], \\ \mathbb{C}(x_1, x_2) &:= \mathbb{E}\left[\tilde{\boldsymbol{c}}(x_1;\cdot)\tilde{\boldsymbol{c}}(x_2;\cdot)^T\right] \in \mathbb{R}^{r\times r}, \end{aligned}$$

and the covariance operator $T : X^r \to X^r$ is given as

$$(T\boldsymbol{c})(x) := \left((\mathbb{C}^{i,\cdot}(x,\cdot), \boldsymbol{c})_{X^r}\right)_{i=1}^{r}, \quad \boldsymbol{c} \in X^{r\cdot}$$

Let λ_k, $k \in \mathbb{N}$, be the eigenvalues of T and $\boldsymbol{c}_k \in X^r$ the corresponding orthonormalized eigenfunctions. With the random variables

$$\xi_k(\omega) := \frac{1}{\sqrt{\lambda_k}} \left(\tilde{\boldsymbol{c}}(\cdot;\omega), \boldsymbol{c}_k\right)_{X^r},$$

the fluctuating part $\tilde{\boldsymbol{c}}$ of the stochastic process $\boldsymbol{c}$ is given by (2.16). For the proof of the representation, we use

$$\begin{aligned}
\lambda_n \boldsymbol{c}_n(x) = (T\boldsymbol{c}_n)(x) &= \left(\left(\mathbb{C}^{i,\cdot}(x,\cdot), \boldsymbol{c}_n\right)_{X^r}\right)_{i=1}^r \\
&= \left(\left(\mathbb{E}\left[\tilde{c}^i(x;\cdot)\tilde{\boldsymbol{c}}(\cdot;\cdot)\right], \boldsymbol{c}_n\right)_{X^r}\right)_{i=1}^r \\
&= \sum_{k=1}^{\infty}\sum_{l=1}^{\infty} \mathbb{E}\left[\xi_k(\cdot)\xi_l(\cdot)\right] \sqrt{\lambda_k\lambda_l}\boldsymbol{c}_k(x) \left(\boldsymbol{c}_l, \boldsymbol{c}_n\right)_{X^r}
\end{aligned}$$

and the remaining part is equivalent to the proof of Theorem 2.3. Furthermore, it is clear that the mean squared truncation error is given analogously to (2.12) by $\sum_{k=K+1}^{\infty} \lambda_k$.

It is still possible to apply the method of snapshots analogously to the scalar-valued case. We define the n_{train}-dimensional matrix $\widehat{\mathbb{C}}$ by

$$\widehat{\mathbb{C}}_{n,m} := \frac{1}{n_{\text{train}}} \left(\tilde{\boldsymbol{c}}(\cdot;\omega_n), \tilde{\boldsymbol{c}}(\cdot;\omega_m)\right)_{X^r}$$

and evaluate its eigenvalues $\hat{\lambda}_k$ and eigenvectors $\boldsymbol{v}_k \in \mathbb{R}^{n_{\text{train}}}$, $k = 1, \ldots, n_{\text{train}}$. Then, the eigenvalues $\hat{\lambda}_k$ coincide with the non-zero eigenvalues of the Monte Carlo approximation T_{MC} of T and the corresponding orthonormalized eigenfunctions of T_{MC} are given by

$$\boldsymbol{c}_k(x) := \frac{1}{\sqrt{\lambda_k n_{\text{train}}}} \sum_{n=1}^{n_{\text{train}}} v_k^{(n)} \tilde{\boldsymbol{c}}(x;\omega_n).$$

The proof works analogously to the scalar-valued case.

2.3 Polynomial Chaos Expansion

It remains to model the random variables $\xi_k : \Omega \to \mathbb{R}$, $k \in \mathbb{N}$. Certainly, equation (2.11) is not appropriate for numerical purposes since it already requires the knowledge of the specific realization of $c(\cdot;\omega)$. Furthermore, the evaluation of the inner product in (2.11) can be expensive. Even with the respective density functions of ξ_k at hand, they may be difficult to simulate since they are only uncorrelated but

not necessarily independent. Hence, it would be desirable to represent ξ_k using a set of independent random variables with known density function.

The Polynomial Chaos (PC) expansion, first introduced by Norbert Wiener in 1938 [101], provides such a method. Let $\{\eta_i : \Omega \to \mathbb{R}\}_{i=1}^{\infty}$ be a set of uncorrelated standard normally distributed random variables. It is shown that the space of all polynomials in $\{\eta_i\}_{i=1}^{\infty}$ is dense in $L_2(\Omega)$, i.e., dense in the space of all second order random variables. Hence, any second order random variable ξ can be represented by a series of orthogonal polynomials in $\{\eta_i\}_{i=1}^{\infty}$ [35].

One possible choice of orthogonal polynomials are Hermite polynomials [101]. For the n-dimensional vector of coordinates $\boldsymbol{\eta} = (\eta_i)_{i=1}^{n}$ and a subset $\{\eta_{i_1}, \ldots, \eta_{i_p}\}$, we denote the Hermite polynomials of degree p by

$$H_p(\eta_{i_1}, \ldots, \eta_{i_p}) := e^{\frac{1}{2}\boldsymbol{\eta}^T\boldsymbol{\eta}} (-1)^p \frac{\partial^p}{\partial\eta_{i_1} \ldots \partial\eta_{i_p}} e^{-\frac{1}{2}\boldsymbol{\eta}^T\boldsymbol{\eta}}, \tag{2.17}$$

where η_{i_r} is not necessarily different to η_{i_s} for $r \neq s$. Hermite polynomials are orthogonal with respect to the weighting function $w_H(\boldsymbol{\eta}) = (2\pi)^{-n/2} e^{-\frac{1}{2}\boldsymbol{\eta}^T\boldsymbol{\eta}}$, i.e.,

$$\int_{\mathbb{R}^n} H_p(\eta_{i_1}, \ldots, \eta_{i_p}) H_q(\eta_{j_1}, \ldots, \eta_{j_q})\, w_H(\boldsymbol{\eta})\, d\boldsymbol{\eta} = h_{i_1,\ldots,i_p} \cdot \delta_{\{i_1,\ldots,i_p\},\{j_1,\ldots,j_q\}},$$

where $h_{i_1,\ldots,i_p}$ denotes the norm of $H_p(\eta_{i_1}, \ldots, \eta_{i_p})$,

$$h_{i_1,\ldots,i_p} = \int_{\mathbb{R}^n} \left| H_p(\eta_{i_1}, \ldots, \eta_{i_p}) \right|^2 w_H(\boldsymbol{\eta})\, d\boldsymbol{\eta}.$$

Since the weighting function w_H corresponds to the Gaussian probability density function, Hermite polynomials in standard normally distributed random variables $\{\eta_i\}_{i=1}^{\infty}$ are also orthogonal with respect to the Gaussian probability measure, i.e.,

$$\mathbb{E}\left[H_p(\eta_{i_1}, \ldots, \eta_{i_p}) H_q(\eta_{j_1}, \ldots, \eta_{j_q}) \right] = h_{i_1,\ldots,i_p} \cdot \delta_{\{i_1,\ldots,i_p\},\{j_1,\ldots,j_q\}}.$$

Now, $h_{i_1,\ldots,i_p}$ denotes the second moment of $H_p(\eta_{i_1}(\omega), \ldots, \eta_{i_p}(\omega))$ and any second order random variable $\xi \in L_2(\Omega)$ can be expanded as

$$\xi(\omega) = a_0 H_0 + \sum_{p=1}^{\infty} \sum_{i_1 \geq \ldots \geq i_p \geq 1} a_{i_1,\ldots,i_p} H_p(\eta_{i_1}(\omega), \ldots, \eta_{i_p}(\omega)), \tag{2.18}$$

where a_0 and $a_{i_1 \ldots i_p}$ denote deterministic coefficients independent of ω,

$$a_0 = \mathbb{E}\left[\xi\right], \quad a_{i_1 \ldots i_p} = \frac{\mathbb{E}\left[\xi H_p(\eta_{i_1}, \ldots, \eta_{i_p})\right]}{h_{i_1,\ldots,i_p}}.$$

Hence, the PC expansion can be seen as the projection of $\xi : \Omega \to \mathbb{R}$ into the space of polynomials with respect to $\{\eta_i : \Omega \to \mathbb{R}\}_{i=1}^{\infty}$.

For numerical purposes, it is necessary to truncate the infinite PC series (2.18). Therefore, we specify a maximal degree r of the polynomials and restrict to the maximal number n of independent random variables. Then, the total number of remaining terms is given by $P + 1 = \binom{n+r}{n}$. Now, it is common to rewrite the truncated version of (2.18) in the form

$$\xi(\omega) = \sum_{p=0}^{P} \hat{a}_p \hat{H}_p(\boldsymbol{\eta}(\omega)), \tag{2.19}$$

where each coefficient $\hat{a}_p$ and each polynomial $\hat{H}_p(\boldsymbol{\eta}(\omega))$ in (2.19) corresponds to a specific coefficient $a_{i_1 \dots i_p}$ and polynomial $H_p(\eta_{i_1}(\omega), \dots, \eta_{i_p}(\omega))$ in (2.18), respectively. We assume that the entities in (2.19) appear in the particular order that is indicated in (2.18), i.e., first the polynomial of degree 0, then n polynomials of degree 1 and so on. Hence,

$$\hat{a}_0 = a_0, \dots, \ \hat{a}_n = a_n, \ \hat{a}_{n+1} = a_{1,1}, \ \hat{a}_{n+2} = a_{2,1}, \ \hat{a}_{n+3} = a_{2,2}, \ \hat{a}_{n+4} = a_{3,1}, \dots$$

and analogously for the polynomials.

It is also possible to use other than normally distributed random variables to model second order processes. As observed before, the weighting function w_H of Hermite polynomials corresponds to the probability density function of the Gaussian random variables. Similarly, e.g., the weighting functions of Laguerre, Jacobi, and Legendre polynomials correspond to the probability density functions of gamma, beta, and uniformly distributed random variables, respectively [103]. Hence, analogous derivations of (2.18) and (2.19) can be done using other appropriate polynomials and random variables. It has been shown that the convergence rate of (2.19) depends on the selected polynomials and random variables. Hence, it depends on the specific problem which representation provides optimal convergence. For more information, we refer to [103].

2.4 Monte Carlo Method

The most straightforward example for D-weak/Ω-strong formulations is the Monte Carlo (MC) method [69, 72]. For each random realization of the random input,

we obtain a deterministic problem and can hence use known deterministic solvers.

Considering the example given in Section 2.1.1, we create a discretized subspace $X \subset H_0^1(D)$ with the basis $\{\varphi_1, \ldots, \varphi_{\mathcal{N}}\}$, e.g., by using the finite element method. For each random realization $c(x;\omega)$ and $d(x;\omega)$, we construct a discretized formulation of (2.4), i.e., the matrix $A(\omega) := (a(\varphi_j, \varphi_i; \omega))_{i,j=1}^{\mathcal{N}}$ and the right-hand side $F(\omega) := (f(\varphi_i;\omega))_{i=1}^{\mathcal{N}}$, where a and f are given in (2.2) and (2.3), respectively. Now, let $\underline{u} \in \mathbb{R}^{\mathcal{N}}$ be the solution of the linear system $A(\omega)\underline{u}(\omega) = F(\omega)$. Then, the solution $u \in X$ of the discretized version of (2.4) is given by

$$u(x;\omega) = \sum_{i=1}^{\mathcal{N}} \underline{u}_i(\omega)\varphi_i(x).$$

Using the KL expansions of $c(x;\omega)$ and $d(x;\omega)$, it is possible to efficiently evaluate the system components $A(\omega)$ and $F(\omega)$. Let c_k, $k \in \mathbb{N}$, be the kth eigenfunction of the KL expansion of c with corresponding eigenvalue λ_k and random variable ξ_k. We assume that c and d are sufficiently precise approximated by using only the first K terms of the corresponding KL expansions. For $w, v \in X$, let $a_k(w, v)$ be given by

$$a_k(w, v) := \int_D c_k(x)\nabla w(x) \cdot \nabla v(x)\, dx,$$

$k = 0, \ldots, K$, where c_0 is given by the mean of c for notational convenience and we denote $\lambda_0 = 1$, $\xi_0 = 1$. Now, we define $A_k := (a_k(\varphi_j, \varphi_i))_{i,j=1}^{\mathcal{N}}$. Then $A(\omega)$ can be constructed as $A(\omega) = \sum_{k=0}^{K} \sqrt{\lambda_k}\xi_k(\omega)A_k$. Analogously, we can construct $F(\omega)$.

For the evaluation of statistical outputs such as mean or variance of the solution u or of any from u derived output of interest $s(u)$, we solve the discretized version of (2.4) for a large set of random realizations of c and d. Then, the MC approximation of the mean and the variance of $s(u)$, i.e., the sample mean and sample variance, are given by

$$\mathbb{E}_{\mathrm{MC}}\left[s(u)\right] = \frac{1}{M}\sum_{m=1}^{M} s(u(\omega_m)), \tag{2.20}$$

$$\mathbb{V}_{\mathrm{MC}}\left[s(u)\right] = \frac{1}{M-1}\sum_{m=1}^{M} \left(s(u(\omega_m)) - \mathbb{E}_{\mathrm{MC}}\left[s(u)\right]\right)^2, \tag{2.21}$$

respectively, where M denotes the number of samples used for the approximation and ω_m, $m = 1, \ldots, M$, the respective underlying random events.

The advantages of the Monte Carlo method include simplicity concerning the implementation. Not only that well known deterministic solvers can be used, it is also clear that parallelization techniques can directly be applied. Furthermore, the convergence rate of the sample mean and variance with respect to the number of used samples M is independent of the dimensionality of the random space, i.e., independent of the number of random variables used to characterize the random inputs in the KL and PC expansion [15, 32].

On the other hand, the convergence is rather slow. The error decreases only in the order of $\mathcal{O}(1/\sqrt{M})$. Hence, it depends on the actual dimension of the probability space if the Monte Carlo method outperforms other techniques that are presented in the subsequent sections.

Several modifications of the Monte Carlo method have been introduced to improve the convergence of the statistical outputs. E.g., using the quasi–Monte Carlo method, the random selection of the samples is replaced by a deterministic sequence of properly chosen points, so-called quasi-random or low-discrepancy sequences [15]. Recently, another Monte Carlo approach has been introduced for stochastic PDEs, called multilevel Monte Carlo method [8, 21], where the PDE is solved for several spatial discretizations. Instead of the straightforward MC application, as for example given in (2.20), the MC mean is evaluated based on a very coarse grid and "updated" by an MC mean of the difference of the outputs of different grids. E.g.,

$$\mathbb{E}_{\mathrm{MLMC}}\left[s(u_{\mathcal{N}})\right] = \frac{1}{M_0}\sum_{m=1}^{M_0} s(u_{\mathcal{N}_0}(\omega_m)) + \frac{1}{M_1}\sum_{m=1}^{M_1}\left(s(u_{\mathcal{N}}(\omega_m)) - s(u_{\mathcal{N}_0}(\omega_m))\right),$$

where $u_{\mathcal{N}}$ and $u_{\mathcal{N}_0}$, $\mathcal{N}_0 < \mathcal{N}$, denote the solutions of a PDE based upon discretizations with $\mathcal{N}$ and $\mathcal{N}_0$ degrees of freedom, respectively. It can be shown that the computational costs compared to the straightforward MC application can be reduced [21].

2.5 Stochastic Galerkin Method

The stochastic Galerkin method has first been proposed by R. G. Ghanem and P. D. Spanos in [35] and denotes the first D-weak/Ω-weak formulation. It has been further discussed for example in [69] and [103] that form the basis for the following

discussion. The method is also known as stochastic finite element method. However, this denomination is also used in other contexts, even for D-weak/Ω-strong formulations. Hence, we prefer the non-ambiguous name.

As for the D-weak/Ω-strong formulation, we use a discretized space X with the basis $\{\varphi_1, \dots, \varphi_{\mathcal{N}}\}$. Additionally, we now discretize the space of second order random variables $L_2(\Omega)$. We use the KL expansions of the coefficients c and d and model the occurring random variables using PC expansions. In the combination with the truncated KL expansion, the number n of used random variables $\boldsymbol{\eta} = (\eta_1, \dots, \eta_n)$ usually coincides with the total number K of used KL terms [35]. Hence, for a maximal degree r of the polynomial chaos, we obtain $P + 1 = \binom{K+r}{K}$ orthogonal basis functions $\hat{H}_p(\boldsymbol{\eta}(\omega))$, $p = 0, \dots, P$, as defined in (2.19). These basis functions span the discretized subspace $S = S(\Omega)$ of $L_2(\Omega)$, i.e., the discretized version of the D-weak/Ω-weak formulation (2.6) is based upon the $\mathcal{N} \cdot (P+1)$-dimensional subspace $X(D) \otimes S(\Omega) \subset H_0^1(D) \otimes L_2(\Omega)$. A basis of $X \otimes S$ is given by $\{\varphi_i \cdot \hat{H}_p \,|\, i = 1, \dots, \mathcal{N}, p = 0, \dots, P\}$. In the following, we assume that the polynomial chaos functions $\hat{H}_p$ are normalized and therefore orthonormal.

Using the definitions of Section 2.1.3, it is clear that we can define the deterministic stiffness matrix $A \in \mathbb{R}^{\mathcal{N}\cdot(P+1)\times\mathcal{N}\cdot(P+1)}$ and the right-hand side $F \in \mathbb{R}^{\mathcal{N}\cdot(P+1)}$ of the discretized problem (2.6) by

$$A := \Big(a(\varphi_j \hat{H}_q, \varphi_i \hat{H}_p)\Big)_{\substack{i,\,j\,=\,1,\dots,\mathcal{N} \\ p,\,q\,=\,0,\dots,P}}, \tag{2.22a}$$

$$F := \Big(f(\varphi_i \hat{H}_p)\Big)_{\substack{i\,=\,1,\dots,\mathcal{N} \\ p\,=\,0,\dots,P}}, \tag{2.22b}$$

respectively. Now, let $\underline{u} \in \mathbb{R}^{\mathcal{N}\cdot(P+1)}$ be the solution of the linear system $A\underline{u} = F$. Then, the solution $u \in X \otimes S$ of the discretized version of (2.6) is given by

$$u(x;\omega) = \sum_{i=1}^{\mathcal{N}} \sum_{p=0}^{P} \underline{u}_{i,p} \varphi_i(x) \hat{H}_p(\boldsymbol{\eta}(\omega)).$$

The evaluation of the mean and the variance of u is straightforward. Since, by definition (2.17) and the orthogonality property of the Hermite polynomials, $\hat{H}_0(\boldsymbol{\eta}) = 1$ and $\mathbb{E}[\hat{H}_p(\boldsymbol{\eta})] = 0$ for $p > 0$, we have

$$\mathbb{E}[u(x;\cdot)] = \sum_{i=1}^{\mathcal{N}} \underline{u}_{i,0} \varphi_i(x).$$

Due to the orthonormality of the polynomial chaos basis functions $\hat{H}_p$, the correlation function of u is given by

$$\mathbb{C}_u(x_1, x_2) = \mathbb{E}[u(x_1;\cdot)u(x_2;\cdot)] = \sum_{i=1}^{\mathcal{N}}\sum_{j=1}^{\mathcal{N}}\sum_{p=1}^{P} \underline{u}_{i,p}\underline{u}_{j,p}\varphi_i(x_1)\varphi_j(x_2).$$

The variance of u at a specific point is given by $\mathbb{V}[u(x;\cdot)] = \mathbb{C}_u(x,x)$.

For linear output functionals $s(u)$, the derivation of mean and variance of s is straightforward, e.g., $\mathbb{E}[s(u)] = s(\mathbb{E}[u]) = \sum_{i=1}^{\mathcal{N}} \underline{u}_{i,0} s(\varphi_i(x))$. For nonlinear outputs, the evaluation may be more involved and may require the knowledge of higher moments of $\hat{H}_p$. However, it is still possible to evaluate sample mean and sample variance as introduced in Section 2.4 since $u(x;\omega)$ can be evaluated pointwise for random realizations $\boldsymbol{\eta}(\omega)$.

2.5.1 The Stiffness Matrix

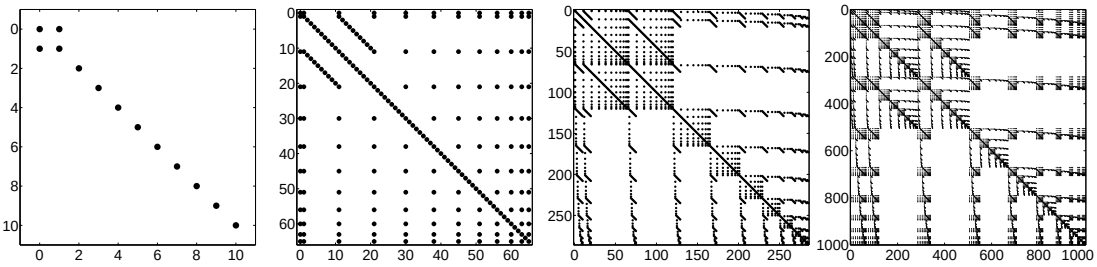
Let us now take a closer look to the stiffness matrix A for stochastic Galerkin methods. We use the KL expansion (2.10) of the coefficient c as introduced in (2.7). The KL sum is truncated after K terms and we model the arising random variables ξ_k using the PC expansion. For notational convenience, we set $c_0(x) := \bar{c}(x)$, $\lambda_0 = 1$ and $\xi_0 = 1$. Then, c is given by

$$c(x;\omega) = \sum_{k=0}^{K} \sqrt{\lambda_k}\xi_k(\omega)c_k(x) = \sum_{k=0}^{K} \sqrt{\lambda_k}c_k(x)\left(\sum_{r=0}^{P} \hat{a}_{k,r}\hat{H}_r(\boldsymbol{\eta}(\omega))\right).$$

The components $A_{(i,p),(j,q)} = a(\varphi_j\hat{H}_q, \varphi_i\hat{H}_p)$ of the stiffness matrix A can therefore be written as

$$\begin{aligned} A_{(i,p),(j,q)} &= \sum_{k=0}^{K} \sqrt{\lambda_k}\,\mathbb{E}\left[\int_D \xi_k c_k(x)\nabla\varphi_j(x)\hat{H}_q(\boldsymbol{\eta})\cdot\nabla\varphi_i(x)\hat{H}_p(\boldsymbol{\eta})\,dx\right] \\ &= \sum_{k=0}^{K} \sqrt{\lambda_k}\left(\int_D c_k(x)\nabla\varphi_j(x)\cdot\nabla\varphi_i(x)dx\right)\cdot\mathbb{E}\left[\xi_k\hat{H}_q(\boldsymbol{\eta})\hat{H}_p(\boldsymbol{\eta})\right]. \end{aligned}$$

Hence, we see that we can separate A into components with different dependencies. On the one hand, we have parts depending just on quantities in the space X of functions on the spatial domain D. On the other hand, we have parts depending only on the quantities in the space S of second order random variables. We



(a) $K = 10,\ r = 1,$ $P = 10.$ (b) $K = 10,\ r = 2,$ $P = 65.$ (c) $K = 10,\ r = 3,$ $P = 285.$ (d) $K = 10,\ r = 4,$ $P = 1000.$

Figure 2.1: Sparsity pattern of the matrix A_1^S for $r = 1, 2, 3, 4$, respectively.

therefore define the corresponding matrices as

$$A_k^X := \left(\int_D c_k(x) \nabla \varphi_j(x) \cdot \nabla \varphi_i(x)\, dx \right)_{i,j=1}^{\mathcal{N}} \in \mathbb{R}^{\mathcal{N} \times \mathcal{N}}, \tag{2.23a}$$

$$\begin{aligned} A_k^S &:= \left(\mathbb{E}\left[\xi_k \hat{H}_q(\boldsymbol{\eta}) \hat{H}_p(\boldsymbol{\eta}) \right] \right)_{p,q=0}^{P} \\ &= \left(\sum_{r=0}^{P} \hat{a}_{k,r} \mathbb{E}\left[\hat{H}_r(\boldsymbol{\eta}) \hat{H}_q(\boldsymbol{\eta}) \hat{H}_p(\boldsymbol{\eta}) \right] \right)_{p,q=0}^{P} \in \mathbb{R}^{(P+1)\times(P+1)}, \end{aligned} \tag{2.23b}$$

$k = 0, \ldots, K$, such that the stiffness matrix is given by the sum of matrix tensor products

$$A := \sum_{k=0}^{K} \sqrt{\lambda_k} \left(A_k^X \otimes A_k^S \right). \tag{2.23c}$$

Hence, this formulation separates random and spatial influences.

The matrices A_k^X correspond to the stiffness matrices of the respective deterministic discretizations and hence show the known sparsity pattern. E.g., using the finite element methods with a linear Lagrange basis to discretize $H_0^1(D)$, the matrices A_k^X are tridiagonal.

The construction of the matrices A_k^S include the evaluations of the mean values of $\mathbb{E}[\hat{H}_r(\boldsymbol{\eta}) \hat{H}_q(\boldsymbol{\eta}) \hat{H}_p(\boldsymbol{\eta})]$, $r, p, q = 0, \ldots, P$. However, this can be done analytically since the random variables in $\boldsymbol{\eta}(\omega)$ are mutually independent with known moments. Furthermore, the values of $\mathbb{E}[\hat{H}_r(\boldsymbol{\eta}) \hat{H}_q(\boldsymbol{\eta}) \hat{H}_p(\boldsymbol{\eta})]$ are actually independent of the current problem and can hence be evaluated and stored once and reused for many different problems. The sparsity pattern of the matrix A_k^S, $k = 1$, is given in Figure 2.1 for a KL expansion with $K = 10$ terms and four different maximal polynomial

degrees $r = 1, 2, 3$, and 4, respectively. For $r = 1$, we obtain 13 non-zeros terms which denotes about 10.74% of the entries whereas for $r = 4$, we obtain 15931 non-zeros terms, i.e., 1.590% of the entries.

Using the tensor product formulation (2.23c), we can describe the shape of $A_k := A_k^X \otimes A_k^S$ as a block matrix of the sparsity pattern of A_k^X, where each block shows the sparsity pattern of A_k^S. Obviously, A_k can analogously be constructed vice versa, i.e., as a block matrix of the shape of A_k^S, where each block has the pattern of A_k^X. In any case, it is clear that a complete decoupling of random and spatial influences is not possible.

Provided that the solutions u are sufficiently smooth in the random space, stochastic Galerkin methods exhibit fast convergence rates with increasing order of the KL and PC expansions. The resolution of the random space S is very high whereas Monte Carlo methods need many simulations to obtain a similar approximation quality. However, in contrast to Monte Carlo methods, the use of a larger number of random variables and higher order polynomials strongly increases the computational effort. The dimension of A_k^S and therefore of the stiffness matrix A grows exponentially fast in the number K of KL terms and with the maximal degree r of the polynomial chaos, recalling that $P + 1 = \binom{K+r}{K}$. Hence, it depends on the actual choice of K and r if Galerkin methods outperform the Monte Carlo method that converges rather slow.

2.6 Stochastic Collocation Method

In this section, we briefly describe the idea of stochastic collocation methods [6, 9, 10, 102] that can be seen as a generalization of the stochastic Galerkin method. The objective is to combine the advantages of both Monte Carlo methods and stochastic Galerkin methods. The main idea is to decouple random and spatial dependencies such that the implementation can be done using basically deterministic solvers as for the Monte Carlo method but maintain the high resolution of S as obtained using stochastic Galerkin methods.

Stochastic collocation methods are also based upon D-weak/Ω-weak formulations, i.e., for the example provided in Section 2.1.1, solutions in the tensor product space $H_0^1(D) \otimes L_2(\Omega)$ are desired. As before, the method performs a Galerkin ap-

proximation in space and one obtains a discrete subset X of $H_0^1(D)$, e.g., using finite elements. Additionally, the method takes advantage of multivariate polynomial interpolations. The random space $L_2(\Omega)$ is approximated using a collocation in the zeros of suitable tensor product orthogonal polynomials. Hence, the approximation S of $L_2(\Omega)$ is again spanned by orthogonal polynomials.

In contrast to stochastic Galerkin methods, the solution procedure requires only evaluations of the corresponding deterministic problems at each interpolation point. Naturally, this leads to uncoupled problems as in the Monte Carlo approach. At the same time, the fast convergence for sufficiently smooth processes can be conserved [6, 102].

The effectivity of such methods depends on proper choices of interpolation points since the overall complexity corresponds to the solution of M deterministic problems, where M is the number of selected knots. Hence, the objective is to choose as few points as possible. Referring to the KL expansion (2.10), a "point" in $L_2(\Omega)$ can be considered to be represented as one random realization of the random variables $\xi_1, \ldots, \xi_K$. Hence, the space to be represented can be transformed to the multidimensional cube $[0,1]^K \subset \mathbb{R}^K$.

Several possibilities for appropriate interpolation point selections have been introduced. Besides the straightforward K-dimensional tensor product of a set of knots in the one-dimensional interval $[0,1]$, e.g., sparse grids based upon the Smolyak algorithm or Stroud's cubature methods have been proposed. For more details, see [102] and the references therein.

Chapter 3

Affine Decompositions of Parametric Stochastic Processes

This chapter is based upon joint work with K. Urban and the main results have already been published in [92] in a very similar form. We added Section 3.1 about affine decompositions in the context of the RBM.

We consider parameter dependent spatial stochastic processes in the context of PDEs and model order reduction. For a given parameter, a random sample of such a process specifies a sample coefficient function of a PDE, e.g., characteristics of porous media such as Li-ion batteries or random influences in biomechanical systems. To apply the Reduced Basis Method (RBM) to parametrized systems with stochastic or deterministic parameter dependencies, it is necessary to get affine decompositions of the systems in parameter and space [45, 73].

For deterministic problems, it is common to use the Empirical Interpolation Method (EIM) [7, 86] for parametric coefficients and the Discrete EIM (DEIM) [19, 20] as well as the Operator EIM (OEIM) [27, 43] for discrete operator approximations. For stochastic coefficients, one can apply the Karhunen–Loève expansion [60, 65] where the terms with stochastic dependencies are assumed to satisfy certain distributions and are modeled using polynomial chaos expansions [101, 103].

In this chapter, we extend the EIM to parametrized spatial stochastic processes. The goal is to develop efficiently computable affine decompositions of not only parameter dependent but also stochastic systems that separate spatial dependencies

from parametric and probabilistic influences without any assumptions on the distribution of non-spatial terms. We will use the basic concept of the EIM together with ideas from Proper Orthogonal Decomposition (POD). We emphasize that the presented methods are not limited to stochastic functions but work analogously on noisy input data or on other hardly decomposable functions.

We start the chapter introducing the necessity and applicability of affine decompositions in the context of reduced methods. In Section 3.2, we provide necessary information about the POD, EIM, Operator EIM, and DEIM that will be used to introduce two new approaches to construct affine decompositions of parametric, stochastic, and possibly non-smooth processes. In Section 3.3, we introduce a Proper Orthogonal Interpolation Method (POIM) that is based on the EIM and the POD and replaces the L_∞-based basis selection by an L_2-'optimal' basis. We show a connection to the DEIM and provide new error estimates that can be used for both methods. We then introduce a Least-Squares EIM (LSEIM) in Section 3.4 that uses more knots than basis functions. A similar approach as already been presented in [71]. In Section 3.5 we provide a numerical example and show that these methods can be used to obtain close to optimal approximations of random and also noisy input data.

3.1 Affine Decompositions in the Context of the RBM

In this section, we show how affine decompositions can be used to efficiently solve parametric PDEs using a small set of basis functions. The objective is to assemble and solve the system independently of the dimension of the actual full discretization but depending only on the size of the reduced basis.

We consider again the example problem given in Section 2.1.1 and the corresponding D-weak/Ω-strong formulation of Section 2.1.2. For now, ω may denote either a deterministic parameter or a stochastic event. As in Section 2.4, we denote $\mathcal{N}$ as the dimension of the full discretized problem, i.e., as the size of the corresponding basis $\{\varphi_1, \dots, \varphi_\mathcal{N}\}$ of the discretized Hilbert space X. Furthermore, let $N \ll \mathcal{N}$ denote the size of a 'reduced basis' $\{\zeta_1, \dots, \zeta_N\}$, $\operatorname{span}\{\zeta_1, \dots, \zeta_N\} =: X_N \subset X$, where $\zeta_n = \sum_{i=1}^{\mathcal{N}} z_{i,n}\, \varphi_i$.

We assume the availability of affine decompositions of the bilinear form a from (2.2) and of the linear form f from (2.3) given by

$$a(w, v; \omega) = \sum_{m=1}^{M^a} \theta_m^a(\omega)\, a_m(w, v), \tag{3.1}$$

$$f(v; \omega) = \sum_{m=1}^{M^f} \theta_m^f(\omega)\, f_m(v), \tag{3.2}$$

respectively, where $M^a, M^f \ll \mathcal{N}$. As described in Section 2.4, the stiffness matrix of the discretized system is given by $A(\omega) := (a(\varphi_j, \varphi_i; \omega))_{i,j=1}^{\mathcal{N}}$ and the right-hand side can be evaluated as $F(\omega) := (f(\varphi_i; \omega))_{i=1}^{\mathcal{N}}$. Furthermore, we define the ω-independent matrices $A_m := (a_m(\varphi_j, \varphi_i))_{i,j=1}^{\mathcal{N}}$, $m = 1, \ldots, M^a$, and the vectors $F_m(\omega) := (f_m(\varphi_i))_{i=1}^{\mathcal{N}}$, $m = 1, \ldots, M^f$.

The reduced problem formulation (2.4) reads as follows: For any $\omega \in \Omega$, find $u_N(\omega) \in X_N$ such that

$$a(u_N(\omega), v; \omega) = f(v; \omega), \quad \forall v \in X_N.$$

Using the reduced basis stiffness matrix $(A_N(\omega) := a(\zeta_k, \zeta_n; \omega))_{n,k=1}^{N}$ and right-hand side $F_N(\omega) := (f(\zeta_n; \omega))_{n=1}^{N}$, the reduced basis solution $u_N(\omega)$ is given by $u_N(\omega) = \sum_{n=1}^{N} \underline{u}_{N,n}(\omega)\zeta_n$, where $\underline{u}_N(\omega) \in \mathbb{R}^N$ denotes the solution of $A_N(\omega)\underline{u}_N = F_N(\omega)$. Hence, it suffices to solve a linear equation of dimension $N \ll \mathcal{N}$. For adequately chosen reduced basis functions, we expect $u(\omega) \approx u_N(\omega)$.

However, for each parameter or random sample $\omega \in \Omega$, we have to assemble a new reduced basis stiffness matrix and right-hand side. The straightforward construction of $A_N(\omega)$ involves the evaluation of

$$a(\zeta_k, \zeta_n; \omega) = \sum_{j=1}^{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} z_{j,k} z_{i,n}\, a(\varphi_j\, \varphi_i; \omega).$$

In other words, using the reduced basis coefficient matrix $Z_N := (z_{i,n})_{\substack{i=1,\ldots,\mathcal{N},\\ n=1,\ldots,N}}$, we have $A_N(\omega) = Z_N^T A(\omega) Z_N$, and analogously, we can evaluate $F_N(\omega) = Z_N^T F(\omega)$. Hence, the assembling of the reduced system is not independent of $\mathcal{N}$ and therefore not efficient. Using directly the definitions of a and f in (2.2) and (2.3), the construction of the system involves the integration over the domain D which also depends on the fine discretization, i.e., on $\mathcal{N}$.

Let us now describe the application of the affine decompositions (3.1) and (3.2) of a and f for the efficient construction of the system. Using the above definitions of the ω-independent quantities A_m and F_m, we define the corresponding reduced basis quantities $A_{N,m} := Z_N^T A_m Z_N \in \mathbb{R}^{N\times N}$ and $F_{N,m} := Z_N^T F_m \in \mathbb{R}^N$. Since these components are also ω-independent, they have to be evaluated only once and can be stored for further use. For each new parameter or random sample $\omega \in \Omega$, using (3.1) and (3.2), we can now assemble

$$A_N(\omega) = \sum_{m=1}^{M^a} \theta_m^a(\omega) A_{N,m}, \quad F_N(\omega) = \sum_{m=1}^{M^f} \theta_m^f(\omega) F_{N,m}, \tag{3.3}$$

with the computational complexities $\mathcal{O}(M^a N^2)$ and $\mathcal{O}(M^f N)$, respectively. Hence, using affine decompositions, the assembling of the system can be performed independently of $\mathcal{N}$, and the reduced solution can be obtained efficiently.

3.2 Preliminaries

In this section, we briefly review some of the basic known facts on POD and EIM that are needed in order to describe our new approaches.

3.2.1 Problem Formulation

Let $(\Omega, \mathfrak{A}, \mathbb{P})$ be a probability space, $\mathcal{P} \subset \mathbb{R}^p$ be a set of deterministic parameters, and let $D \subset \mathbb{R}^d$ denote a spatial domain. Furthermore, let $c : D \times (\mathcal{P} \times \Omega) \to \mathbb{R}$ denote a real-valued parameter dependent spatial stochastic process. For each pair $(\mu, \omega) \in \mathcal{P} \times \Omega$, we assume to obtain a trajectory $c(\mu, \omega) \in X \subset L_\infty(D) \cap C^0(D)$ for some appropriate Hilbert space X on D.

Let now $c(\mu, \omega)$ denote a coefficient or right-hand side in some PDE. Provided that $c(\mu, \omega)$ is an affine function of the parameters and the spatial variables, it is also possible to get an affine approximation of the bilinear form a and the linear form f. In general, however, this requirement is not fulfilled, in particular in the presence of stochastic influences. The objective of this chapter is thus (i) to find an affine approximation of $c(\mu, \omega)$ of the form

$$c(x; \mu, \omega) \approx \sum_{m=1}^{M} \theta_m(\mu, \omega)\, q_m(x) \tag{3.4}$$

with so-called *collateral basis* functions $q_m \in X$, $m = 1, \dots, M$, (ii) to construct efficient evaluation procedures for the coefficients $\theta_m(\mu, \omega) \in \mathbb{R}$, $m = 1, \dots, M$, and (iii) the derivation of effective a-posteriori error estimators to choose $M \in \mathbb{N}$ possibly small in order to guarantee a certain accuracy in (3.4).

Suppose an affine decomposition in the deterministic parameter is already given, i.e.,

$$c(x; \mu, \omega) = \sum_{q=1}^{Q} \theta_q(\mu)\, c_q(x; \omega),$$

where $\theta_q(\mu)$ can be evaluated efficiently, possibly analytically. We can evaluate the respective KL expansions of the stochastic functions c_q, truncate each expansion, and obtain a decomposition of the desired form,

$$c(x; \mu, \omega) \approx \sum_{q=1}^{Q} \sum_{k=0}^{K_q} \theta_q(\mu) c_{q,k}(\omega)\, \sqrt{\lambda_{q,k}} c_{q,k}(x).$$

Otherwise, more involved algorithms are necessary.

3.2.2 Proper Orthogonal Decomposition (POD)

As already mentioned, the POD can be seen as the deterministic equivalent of the KL expansion. Similarly, one evaluates the eigenfunctions and eigenvalues of a covariance operator to determine an orthonormal basis and to estimate the approximation quality of the corresponding subspace. In the deterministic context, the POD is often formulated as an optimization problem based upon a set of training snapshots:

For some training set $\Xi_{\text{train}} \subset \mathcal{P} \times \Omega$ of cardinality n_{train} and corresponding trajectories $c(\mu, \omega)$, $(\mu, \omega) \in \Xi_{\text{train}}$, the POD space V_M^{POD} of dimension M is defined via the following optimization problem

$$V_M^{\text{POD}} := \arg \inf_{\substack{V_M \subset X_{\text{train}} \\ \dim V_M = M}} \left(\frac{1}{n_{\text{train}}} \sum_{(\mu,\omega) \in \Xi_{\text{train}}} \inf_{w_M \in V_M} \| c(\mu, \omega) - w_M \|_2^2 \right), \tag{3.5}$$

where $X_{\text{train}} := \operatorname{span}\{c(\mu, \omega) | (\mu, \omega) \in \Xi_{\text{train}}\}$. It yields hierarchical spaces, i.e., $V_{M-1} \subset V_M$, and is L_2-optimal in the sense that the average squared L_2-error of the representation of the training trajectories is minimized.

As for the KL expansion, a hierarchical basis of V_M^{POD} is given by the eigenfunctions v_m of decreasing eigenvalues λ_m, $m = 1, \ldots, M$, of the covariance operator $\mathbb{C}_{\mathrm{POD}} : D \times D \to \mathbb{R}$ defined as

$$\mathbb{C}_{\mathrm{POD}}(x_1, x_2) := \frac{1}{n_{\mathrm{train}}} \sum_{(\mu,\omega)\in\Xi_{\mathrm{train}}} c(x_1; \mu, \omega)\, c(x_2; \mu, \omega), \quad x_1, x_2 \in D.$$

Analogously to the KL expansion, the average squared approximation error of the trajectories in the training set is given by

$$\frac{1}{n_{\mathrm{train}}} \sum_{(\mu,\omega)\in\Xi_{\mathrm{train}}} \|c(\mu,\omega) - c_M^{\mathrm{POD}}(\mu,\omega)\|_2^2 \;=\; \sum_{m>M} \lambda_m,$$

where $c_M^{\mathrm{POD}}(\mu,\omega)$ denotes the orthogonal projection of $c(\mu,\omega)$ onto V_M^{POD}. For more details, see for example [62]. As for the KL expansion, it is also possible to apply the method of snapshots for the evaluation of the eigenvalues and the construction of the eigenfunctions of $\mathbb{C}_{\mathrm{POD}}$.

However, using the eigenfunctions v_m, $m = 1, \ldots, M$, as collateral basis, it is not possible to efficiently evaluate the corresponding coefficients $\theta_m(\mu,\omega)$, $m = 1, \ldots, M$. In contrast to the KL expansion, these coefficients do not satisfy a certain probability distribution and can not be modeled using PC expansion. Hence, it is not possible to directly apply the POD for our purpose.

3.2.3 Empirical Interpolation Method (EIM)

We briefly review the EIM as introduced for example in [7] and [86]. In these publications, it has been used to derive affine decompositions of parametric functions. Here, we use the parametric stochastic specification that we consider in this work.

The main idea of the EIM is to use a collateral basis such that the affine approximation of a new function c requires only the values of c at a set of interpolation points of the same size as the basis. The construction of the basis ensures that the approximation is exact at the knots and that the coefficients are efficiently evaluable.

EIM: Offline-phase

A general form of the EIM offline procedure is described in Algorithm 3.1. It generates the so-called collateral basis $Q_M = \{q_1, \ldots, q_M\}$ of cardinality M and

Algorithm 3.1 Offline – Empirical Interpolation Method.

1 **for** $M = 1$ **to** $M_{\max}$ **do**
2 $\quad c = \mathsf{getNextBasisFunction}(Q_{M-1}, T_{M-1}, \Xi_{\text{train}})$
3 $\quad c_{M-1}^{\text{EIM}} = \mathsf{getApproximation}(Q_{M-1}, T_{M-1}, c)$
4 $\quad r_M = c - c_{M-1}^{\text{EIM}}$
5 $\quad t_M = \arg\operatorname{ess\,sup}_{x \in D} |r_M(x)|, \quad T_M = \{T_{M-1}, t_M\}$
6 $\quad q_M = r_M / r_M(t_M), \quad Q_M = \{Q_{M-1}, q_M\}$
7 **end for**

the corresponding set of interpolation points $T_M = \{t_1, \ldots, t_M\}$, $M \leq M_{\max}$, where $M_{\max}$ denotes the maximal allowed number of affine terms. We will describe the main steps below. The ingredient of the algorithm is a training set $\Xi_{\text{train}} \subset \mathcal{P} \times \Omega$ such that the space $\operatorname{span}\{c(\mu, \omega) \,|\, (\mu, \omega) \in \Xi_{\text{train}}\}$ sufficiently covers the family of functions $\{c(\mu, \omega) \,|\, (\mu, \omega) \in \mathcal{P} \times \Omega\}$. Furthermore, we start with an empty set of basis functions $Q_0 = \{\}$ and an empty set of interpolation points $T_0 = \{\}$.

We start with the procedure that computes the affine approximation in line 3 of Algorithm 3.1. In the first step of the loop, for an empty basis Q_0, the procedure $\mathsf{getApproximation}(Q_0, T_0, c)$ returns zero, i.e., $c_0^{\text{EIM}} = 0$ for all functions $c \in X$. Otherwise, for any non-empty basis Q_M, $\mathsf{getApproximation}(Q_M, T_M, c)$ computes the coefficients $\boldsymbol{\theta}_M(c) = (\theta_j(c))_{j=1}^M$ by solving the linear system

$$\sum_{j=1}^{M} \theta_j(c) q_j(t_i) = c(t_i),\ i = 1, \ldots, M, \tag{3.6}$$

and returns the approximation $c_M^{\text{EIM}} = \sum_{j=1}^M \theta_j(c) q_j$. By construction, this approximation is exact at the knots $t_i, i = 1, \ldots, M$. Denoting $B_M := (q_j(t_i))_{i,j=1}^M$ and $\boldsymbol{c}_M := (c(t_i))_{i=1}^M$ allows to rewrite (3.6) as $B_M \boldsymbol{\theta}_M(c) = \boldsymbol{c}_M$ such that

$$c_M^{\text{EIM}} = Q_M \boldsymbol{\theta}_M(c) = Q_M B_M^{-1} \boldsymbol{c}_M. \tag{3.7}$$

Here, $Q_M = \{q_1, \ldots, q_M\}$ is associated with the "matrix" where each column refers to one basis function.

The procedure $\mathsf{getNextBasisFunction}(Q_{M-1}, T_{M-1}, \Xi_{\text{train}})$ in line 2 evaluates EIM approximations $c_{M-1}^{\text{EIM}}(\mu, \omega)$ of all trajectories $c(\mu, \omega), (\mu, \omega) \in \Xi_{\text{train}}$, and returns the trajectory that is so far worst approximated in the L_∞-sense. Hence, in the first step, the procedure returns the training function with the largest magnitude.

In line 4, the residual is evaluated. The next knot t_M is defined in line 5 in order to supremize the residual, i.e., as that point where c is so far worst approximated. Hence, the interpolation point selection procedure is based upon the L_∞-error. The next collateral basis function q_M is added in line 6, defined as the L_∞-normalized residual. We denote the approximation space at step M by $W_M^{\mathrm{EIM}} := \mathrm{span}\{q_1, \dots, q_M\}$.

As mentioned before, the approximation is exact at the knots, i.e., the residual r_M is zero at $t_1, \dots, t_{M-1}$. This implies that the linear system (3.6) is lower triangular with diagonal unity, i.e., $(B_M)_{j,j} = q_j(t_j) = 1$ and $(B_M)_{i,j} = q_j(t_i) = 0$ for $i < j$. The computational complexity of the evaluation of the EIM coefficients $\boldsymbol{\theta}_M$ is thus $\mathcal{O}(M^2)$.

EIM: Online-phase

In the online phase, sketched in Algorithm 3.2, we affinely approximate a new trajectory $c(\mu, \omega)$ for $(\mu, \omega) \in \mathcal{P} \times \Omega$. We choose an $M < M_{\max}$ that is assumed to be sufficiently large for a good approximation quality. Additionally, we define M^+ with $M < M^+ \leq M_{\max}$ that is used for the error estimation.

We then call $\mathsf{getCoefficients}(M^+, c(\mu, \omega))$ that evaluates the trajectory at the knots $(t_i)_{i=1}^{M^+}$ and returns the solution $\boldsymbol{\theta}_{M^+}(\mu, \omega)$ of the lower triangular linear system

$$\sum_{j=1}^{M^+} \theta_j(\mu, \omega) q_j(t_i) = c(t_i; \mu, \omega),\ i = 1, \dots, M^+. \tag{3.8}$$

For an efficient application of the EIM, we require that evaluations of trajectories at the knots $(t_i)_{i=1}^{M^+}$ are fast, ideally of complexity $\mathcal{O}(M^+)$. Due to the lower triangular form of the linear system (3.8), the solutions show a hierarchical structure, i.e., $\boldsymbol{\theta}_{M+1} = (\boldsymbol{\theta}_M, \theta_{M+1})$.

We use only the first M coefficients to evaluate the approximation $c_M^{\mathrm{EIM}}(\mu, \omega)$ of the given trajectory, see line 4 of Algorithm 3.2. This evaluation is not independent of the dimension $\mathcal{N}$ of a given discretization of the function space X. However, as we have seen in Section 3.1, it is not even necessary in the RBM context to evaluate $c_M^{\mathrm{EIM}}(\mu, \omega)$ in the online phase at all, only the coefficients $\boldsymbol{\theta}_M$ are used. Hence, line 4 of Algorithm 3.2 can be skipped.

One usually uses the additional coefficients to get error estimators. Under the

Algorithm 3.2 Online – Empirical Interpolation Method.

1 choose M and M^+ such that $M < M^+ \leq M_{\max}$

2 select a trajectory $c(\mu, \omega)$ for some $(\mu, \omega) \in \mathcal{P} \times \Omega$

3 $\boldsymbol{\theta}_{M^+}(\mu, \omega) = \mathsf{getCoefficients}(M^+, c(\mu, \omega))$

4 evaluate approximation

$$c_M^{\mathrm{EIM}}(\mu, \omega) = \sum_{j=1}^{M} \theta_j(\mu, \omega) q_j \tag{3.9}$$

5 evaluate the L_∞-error estimator

$$\Delta_{M,M^+}^{\mathrm{EIM}}(\mu, \omega) = \sum_{j=M+1}^{M^+} |\theta_j(\mu, \omega)| \tag{3.10}$$

assumption that the trajectory $c(\mu, \omega)$ is in $W_{M^+}^{\mathrm{EIM}}$, the quantity $\Delta_{M,M^+}^{\mathrm{EIM}}(\mu, \omega)$ from (3.10) provides a rigorous upper bound of the L_∞-error. The respective bound for the L_2-error could be given by $\sum_{j=M+1}^{M^+} \|q_j\|_2 |\theta_j(\mu, \omega)|$. However, the assumption $c(\mu, \omega) \in W_{M^+}^{\mathrm{EIM}}$ usually does not hold and $\Delta_{M,M^+}^{\mathrm{EIM}}$ just provides a non-rigorous (but in practice very good) estimate. For more details on EIM error estimators and more accurate bounds, see [86].

3.2.4 Empirical Interpolation of Differential Operators

The DEIM [20] and the empirical operator interpolation [27, 43] work in a similar context. Both methods generate affine decompositions of discretized differential operators. As opposed to the EIM, the basis function selection is based upon operator evaluations and the knots represent indices of the discrete operator. In the online phase, the discrete operator evaluations are approximated instead of trajectories $c(\mu, \omega)$. Hence, Algorithms 3.1 and 3.2 can directly be used for the empirical operator interpolation, considering t_i to be indices and c to be operator evaluations.

In this context, the evaluation of the operator — typically nonlinear and/or time dependent — at an index t_i involves the evaluation of the solution of the equation from a previous time step or iteration at several points. It is required that the

Algorithm 3.3 Offline – DEIM.

1 **for** $M = 1$ **to** $M_{\max}$ **do**
2 select v_M as next basis function
3 $c_{M-1}^{\mathrm{DEIM}} = \mathsf{getApproximation}(V_{M-1}, T_{M-1}, v_M)$
4 $r_M = v_M - c_{M-1}^{\mathrm{DEIM}}$
5 $t_M = \arg\operatorname{ess\,sup}_{x \in D} |r_M(x)|, \quad T_M = \{T_{M-1}, t_M\}$
6 $V_M = \{V_{M-1}, v_M\}$
7 **end for**

number of such points is constant and much smaller than the number of degrees of freedom of the discretization. This property is also called H-independence [27]. Typical discretization techniques such as finite element, finite volume, or finite difference methods fulfill this requirement. In the following we do not explicitly address this topic. However, it is important to keep in mind that the evaluation at an index t_i can be expensive.

The DEIM implies further modifications of the presented algorithms. At the start of the method, one applies a POD on the discrete operator generating eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots$ and corresponding orthonormal eigenfunctions $v_1, v_2, \cdots$. The further steps are sketched in Algorithm 3.3.

Instead of $\mathsf{getNextBasisFunction}()$ in Algorithm 3.1, we directly select the M-th POD eigenfunction in iteration M. Furthermore, in the DEIM context, we do not add the residual to the collateral basis, but the eigenfunction itself. Hence, line 6 of Algorithm 3.1 reduces to $V_M = \{V_{M-1}, v_M\}$ and the approximation space reads $W_M^{\mathrm{DEIM}} := \operatorname{span}\{v_1, \ldots, v_M\}$. Lines 3 to 5 remain necessary to determine the knots.

Due to the different selection method, the linear systems (3.6), solved in line 3 of Algorithm 3.3, and (3.8), solved online, become full. The complexity increases to $\mathcal{O}(M^3)$ and $\mathcal{O}((M^+)^3)$, respectively. Furthermore, the error estimator introduced in line 5 of Algorithm 3.2 is not valid anymore since in this context, it does not hold that the coefficients are hierarchical, i.e., $\boldsymbol{\theta}_{M+1} \neq (\boldsymbol{\theta}_M, \theta_{M+1})$. There are some non-rigorous a-priori average-error estimates, see [20].

3.3 A Proper Orthogonal (Empirical) Interpolation Method (POIM)

In this section, we propose a Proper Orthogonal Interpolation Method (POIM) that is based on the EIM and POD. The main idea is to replace the basis selection based upon the L_∞-error by some L_2-'optimal' procedure. Even though the method is motivated by stochastic problems, it can be applied to deterministic formulations as well and may lead to improved approximations in that case, too.

The method has some similarities to the DEIM, even though the DEIM originally applies to differential operators. In fact, we show that we can modify the DEIM according to the POIM methodology, making it faster but still producing the same approximations. Furthermore, we show that the provided a-posteriori error estimates for the POIM can also be applied to the DEIM.

3.3.1 Outline of the Method

We adopt the concept of the DEIM and apply the POD to our problem in a first step. In other words, we define a training set $\Xi_{\text{train}} \subset \mathcal{P} \times \Omega$, evaluate trajectories $c(\mu, \omega), (\mu, \omega) \in \Xi_{\text{train}}$, and compute POD eigenvalues $\lambda_1, \ldots, \lambda_{M_{\max}}$ and eigenfunctions $v_1, \ldots, v_{M_{\max}}$, using either the method of snapshots or the direct approach.

As for the DEIM, we select in each iteration the respective POD eigenfunction as next basis function and evaluate its approximation to define the residual and the knot. However, in contrast to the DEIM, we do not directly add the POD eigenfunction to the collateral basis, but we use the L_∞-normalized residual q_M, as described in Algorithm 3.4, line 6. This part of the algorithm has been adopted from the EIM and ensures that the linear systems (3.6) and (3.8) are still lower triangular. Therefore, the procedure $\mathsf{getApproximation}(Q_M, T_M, \cdot)$ is identical to the one used in Algorithm 3.1 and the online phase of the POIM is identical to the online phase of the EIM provided in Algorithm 3.2.

It is clear that the approximation space W_M^{POIM} is still L_2-optimal in the sense of (3.5). In other words, we have

$$W_M^{\text{POIM}} = \operatorname{span}\{q_1, \ldots, q_M\} = \operatorname{span}\{v_1, \ldots, v_M\} = W_M^{\text{DEIM}}, \tag{3.11}$$

Algorithm 3.4 Offline – POIM.

1 **for** $M = 1$ **to** $M_{\max}$ **do**
2 select v_M as next basis function
3 $c_{M-1}^{\mathrm{POIM}} = \mathsf{getApproximation}(Q_{M-1}, T_{M-1}, v_M)$
4 $r_M = v_M - c_{M-1}^{\mathrm{POIM}}$
5 $t_M = \arg\operatorname{ess\,sup}_{x \in D} |r_M(x)|, \quad T_M = \{T_{M-1}, t_M\}$
6 $q_M = r_M / r_M(t_M), \quad Q_M = \{Q_{M-1}, q_M\}$
7 **end for**

which can be easily shown by induction over M. The basis Q_M is not orthonormal and the knots still depend on the L_∞-error of the residual r_M. However, since r_M is a linear combination of the first M POD eigenfunctions, it is typically smooth and the knot should be adequately chosen.

3.3.2 Error Estimators

We can directly apply the error estimator defined in Algorithm 3.2, line 5, i.e., we solve the lower triangular system (3.8) in $\mathcal{O}((M^+)^2)$ for some $M^+ > M$ and use the additional coefficients $\theta_{M+1}, \ldots, \theta_{M^+}$ to evaluate $\Delta_{M,M^+}^{\mathrm{EIM}}(\mu, \omega)$.

3.3.3 Application within the DEIM Context

As indicated in Section 3.2.4, the concepts of EIM and DEIM differ only slightly, using operator evaluations instead of trajectories and indices instead of interpolation points. Hence, the POIM can directly be used to approximate operators as well. In view of (3.11), the approximation spaces of the DEIM and the POIM coincide. In the following two lemmas, we show that both methods also produce the same approximations.

Lemma 3.1. *Let c be an arbitrary function and let c_M^{POIM}, c_M^{DEIM} be approximations using M basis functions generated by the POIM and the DEIM, respectively, using the same interpolation points. Then, $c_M^{\mathrm{POIM}} = c_M^{\mathrm{DEIM}}$.*

Proof. Let $Q_M = \{q_1, \ldots, q_M\}$ denote the matrix of POIM-basis functions and $V_M = \{v_1, \ldots, v_M\}$ the matrix of DEIM-basis functions, where each column of the respective matrices refers to one basis function. Since both bases span the

same space, there exists a matrix $\Psi_M \in \mathbb{R}^{M\times M}$ such that $Q_M = V_M \cdot \Psi_M$. Due to the construction of Q_M in Algorithm 3.4, Ψ_M is upper triangular. Let $T_M = (t_1, \ldots, t_M)$ denote the selected knots. We define

$$B_M^{\mathrm{POIM}} := (q_j(t_i))_{i,j=1}^M, \quad B_M^{\mathrm{DEIM}} := (v_j(t_i))_{i,j=1}^M \in \mathbb{R}^{M\times M},$$

and $\mathbf{c}_M := (c(t_i))_{i=1}^M \in \mathbb{R}^M$. Since $Q_M = V_M \cdot \Psi_M$, it is also clear that we can write $B_M^{\mathrm{POIM}} = B_M^{\mathrm{DEIM}} \cdot \Psi_M$. Then, using the form of (3.7) for the respective linear systems, we obtain

$$\begin{aligned} c_M^{\mathrm{POIM}} &= Q_M \cdot \left(B_M^{\mathrm{POIM}}\right)^{-1} \mathbf{c}_M \\ &= V_M \Psi_M \cdot \left(B_M^{\mathrm{POIM}}\right)^{-1} \mathbf{c}_M \\ &= V_M \cdot \left(B_M^{\mathrm{DEIM}}\right)^{-1} \mathbf{c}_M \;=\; c_M^{\mathrm{DEIM}} \end{aligned}$$

which proves the claim. □

We furthermore note that the upper triangular matrix Ψ_M is hierarchical in the sense that Ψ_{M-1} is given as the restriction of Ψ_M to the first $M-1$ rows and columns. This is clear from the construction of Q_M in Algorithm 3.4, line 4 as a linear combination of $\{Q_{M-1}, v_M\}$. Since $\mathrm{span}\{Q_{M-1}\} = \mathrm{span}\{V_{M-1}\}$, q_M can also be written as a linear combination of the basis V_M.

It remains to show that the knots produced by the different methods coincide.

Lemma 3.2. *The DEIM in Algorithm 3.3 and the POIM in Algorithm 3.4 generate the same set of interpolation points.*

Proof. Let $(t_i^{\mathrm{POIM}})_{i=1}^M$ denote the POIM-knots and $(t_i^{\mathrm{DEIM}})_{i=1}^M$ the DEIM-knots. The proof is now done by induction. Since for both methods, the approximation procedures $\mathsf{getApproximation}(\cdot)$ return zero for empty basis sets Q_0 or V_0, respectively, we have that $r_1 = v_1$ for both methods and therefore $t_1^{\mathrm{POIM}} = t_1^{\mathrm{DEIM}}$. Let the assertion be true for $M-1$. Then, Lemma 3.1 provides that both methods return the same approximation, i.e., $c_{M-1}^{\mathrm{POIM}} = c_{M-1}^{\mathrm{DEIM}}$. Hence, both methods use the same residual to evaluate the next knot such that $t_M^{\mathrm{POIM}} = t_M^{\mathrm{DEIM}}$. □

As a consequence of the two results above, we can use the POIM instead of the DEIM, generating the same approximations, but solving only a triangular system. Hence, the online complexity reduces to $\mathcal{O}(M^2)$. Furthermore, we can now use

the EIM a-posteriori error estimates for the DEIM as well. At the same time, the DEIM a-priori error estimates are still valid since neither the approximation space is changed nor the actual approximations.

Even if an orthonormal basis would be needed and the DEIM is directly applied, we can now implement the DEIM more efficiently, including also the evaluation of a-posteriori error estimates. We first solve the triangular system (3.8) for coefficients $\boldsymbol{\theta}_{M^+}^{\mathrm{POIM}}$ which also includes the evaluation of $\boldsymbol{\theta}_{M}^{\mathrm{POIM}}$ due to the hierarchical behavior of the coefficients. It holds that

$$\boldsymbol{\theta}_{M}^{\mathrm{DEIM}} = \Psi_M \boldsymbol{\theta}_{M}^{\mathrm{POIM}}, \qquad \boldsymbol{\theta}_{M^+}^{\mathrm{DEIM}} = \Psi_{M^+} \boldsymbol{\theta}_{M^+}^{\mathrm{POIM}}. \tag{3.12}$$

Since Ψ_M is upper triangular, the complexities of the evaluations in (3.12) are $\mathcal{O}(M^2)$ and $\mathcal{O}((M^+)^2)$, respectively. Hence, the DEIM coefficients can be evaluated with a total complexity of $\mathcal{O}(2M^2)$. Furthermore, we can still apply the error estimator (3.10) with the POIM coefficients $\theta_{M+1}^{\mathrm{POIM}}, \ldots, \theta_{M^+}^{\mathrm{POIM}}$. The computational complexity of the error estimator is therefore $\mathcal{O}(2(M^+)^2)$. We do not need to store two sets of basis functions but only the orthonormal basis V_{M^+} and the two triangular matrices Ψ_{M^+} and $B_{M^+}^{\mathrm{POIM}}$ of the POIM.

3.4 A Least-Squares Empirical Interpolation Method (LSEIM)

In this section, we introduce a Least–Squares Empirical Interpolation Method (LSEIM) that uses more knots than basis functions and solves a least-squares problem to evaluate $\boldsymbol{\theta}_M$. This can be combined with both EIM and POIM.

3.4.1 Outline of the Method

The general concept of the LSEIM offline procedure is described in Algorithm 3.5. The main steps are described below. We again initialize the algorithm with an empty basis $Q_0 = \{\}$ and an empty set of knots $T_0 = \{\}$. Furthermore, we denote the number of used knots in step M by I_M and set $I_0 = 0$.

The procedure getNextBasisFunction() in line 2 returns either the so far worst approximated snapshot, as described for the EIM in Section 3.2.3, or the M-th POD eigenfunction, if the LSEIM is combined with the POIM.

Algorithm 3.5 Offline – LSEIM.

1 **for** $M = 1$ **to** $M_{\max}$ **do**
2 $\quad c = \mathsf{getNextBasisFunction}()$
3 $\quad c_{M-1}^{\mathrm{LSEIM}} = \mathsf{getApproximation}(Q_{M-1}, T_{I_{M-1}}, c)$
4 $\quad r_M = c - c_{M-1}^{\mathrm{LSEIM}}$
5 $\quad (t_i)_{i=I_{M-1}+1}^{I_M} = \mathsf{getNextKnots}(r_M), \quad T_{I_M} = \{T_{I_{M-1}}, (t_i)_{i=I_{M-1}+1}^{I_M}\}$
6 $\quad q_M = \mathsf{get}L_2\mathsf{orthonormal}(r_M), \quad Q_M = \{Q_{M-1}, q_M\}$
7 **end for**

For the LSEIM-approximation in line 3, we solve the least-squares problem

$$\sum_{i=1}^{I_M} \left(\sum_{j=1}^{M} \theta_j(c) q_j(t_i) - c(t_i) \right)^2 \to \min \tag{3.13}$$

for the coefficients $\boldsymbol{\theta}_M \in \mathbb{R}^M$ and evaluate $c_M^{\mathrm{LSEIM}} = Q_M \boldsymbol{\theta}_M$. Since the approximation and thus the residual r_M are no longer exact at the knots, the system is full and the complexity of solving (3.13) increases to $\mathcal{O}(I_M M^2)$.

There is no unique way to determine the number and location of the new knots $(t_i)_{i=I_{M-1}+1}^{I_M}$ in line 5. For the examples in Section 3.5, we used a constant number of two new knots per basis function, defined by the essential infimum and the essential supremum of the residual, respectively: $t_{I_M-1} := \arg\operatorname{ess\,inf}_{x\in D} r_M(x)$ and $t_{I_M} := \arg\operatorname{ess\,sup}_{x\in D} r_M(x)$ with $I_M = 2M$.

It is also possible to use iterative and adaptive selection methods. A natural procedure would be to first add the basis function and iteratively add knots in a second step. The actual number of knots, i.e., the number of iterations, can also be determined in several different ways. One choice could be to add knots until the approximations of the functions in the training set are close to optimal in the sense of their L_2-projections into the space $W_M^{\mathrm{POD}} = \operatorname{span}(Q_M)$. Alternatively, one could also measure the error between the L_2-optimal and LSEIM coefficients $\boldsymbol{\theta}_M^{L_2}$ and $\boldsymbol{\theta}_M^{LSEIM}$ which might be cheaper. A third way could be to just minimize the approximation error for the last basis function. In any case, it is crucial to adequately specify the error term 'close to optimal', i.e., the error tolerance. This can be difficult and my depend on the actual problem. Hence, we prefer the above mentioned simple method and we will see in Section 3.5 that it works very well in practice.

To extend the L_2-orthonormal basis in line 6, we add the L_2-projection of the residual on $W_{M-1}^{\text{LSEIM}} := \text{span}\{q_1, \dots, q_{M-1}\}$ to the basis. Analogously to Lemmas 3.1 and 3.2, we can show that this is equivalent to add L_∞-normalized residuals. We just replace the solution of $B_M \boldsymbol{\theta}_M = \boldsymbol{c}_M$ in the proof of Lemma 3.1 by the solution of a minimization problem of the form (3.13). However, since the system is full anyway, we prefer the L_2-orthonormal basis.

Once M is fixed in the online phase, one can compute and store the QR-decomposition and solve (3.13) in $\mathcal{O}(I_M M)$ for any new right-hand side. Under the assumption that the number of selected knots per iteration is $\mathcal{O}(1)$, i.e., $I_M \in \mathcal{O}(M)$, the cost increases only moderately. A drawback in the online application is the necessity to evaluate trajectories $c(\mu, \omega)$ at additional knots to get new right-hand sides, which can be expensive. However, we hope to reduce the number M of affine terms such that the overall cost decreases. Furthermore, within the RBM context, the total online complexity to assemble the system and to compute solution and error bounds is $\mathcal{O}(I_M M + MN^2 + N^3 + M^2 N^2)$, where N is the dimension of the reduced space (cf. [73]). Thus, a small M becomes more important than a decrease of the number of knots.

3.4.2 Error Estimators

It is not possible to directly adopt the error estimators used for the EIM and POIM since $\boldsymbol{\theta}_{M+1} \neq (\boldsymbol{\theta}_M, \theta_{M+1})$. Instead, we separately solve (3.13) for M and M^+ and denote the solutions by $(\theta_j^M)_{j=1}^M$ and $(\theta_j^{M^+})_{j=1}^{M^+}$, respectively. Since Q_M is L_2-orthonormal, the L_2-error estimator is given by

$$\Delta_{M,M^+}^{\text{LSEIM}} := \sum_{j=1}^{M} \left| \theta_j^{M^+} - \theta_j^M \right| + \sum_{j=M+1}^{M^+} \left| \theta^{M_j^+} \right| \tag{3.14}$$

whereas the respective L_∞-error estimator is given by $\sum_{j=1}^M \|q_j\|_\infty |\theta_j^{M^+} - \theta_j^M| + \sum_{j=M+1}^{M^+} \|q_j\|_\infty |\theta^{M_j^+}|$. The computational complexity increases compared to EIM and DEIM, even though it still is $\mathcal{O}((M^+)^2)$ for given QR-decompositions and $I_M \in \mathcal{O}(M)$.

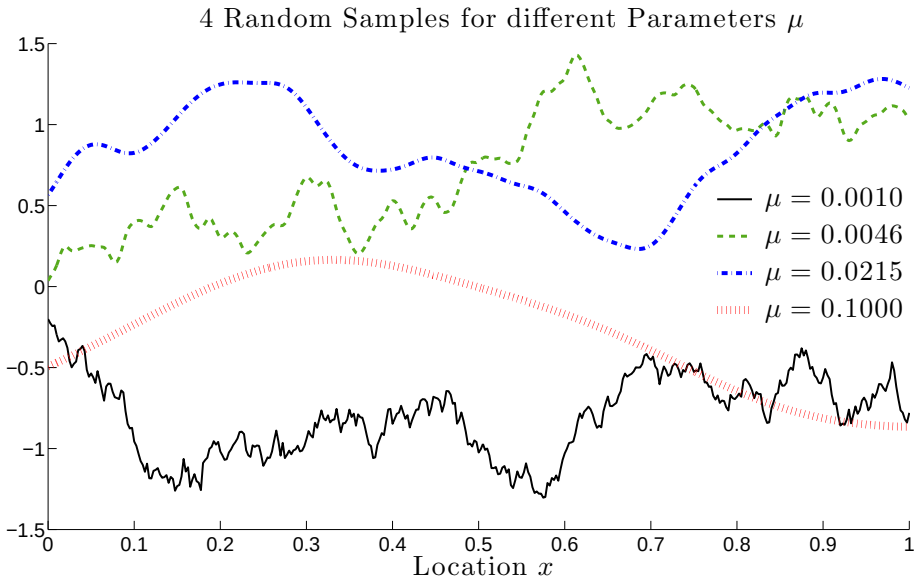


Figure 3.1: Four random trajectories $c(\mu, \omega)$ as defined in (3.15) for different smoothing parameter configurations.

3.5 Numerical Example

We consider a Wiener process $W : \mathbb{R} \times \Omega \to \mathbb{R}$ with probability space $(\Omega, \mathfrak{A}, \mathbb{P})$ such that $W(x;\omega) - W(y;\omega)$ is normally distributed with zero mean and variance $|x - y|$. The variance at $x = 0$ is assumed to be zero. Furthermore, we apply a parameter dependent smoothing filter $F(x, y; \mu) = \frac{1}{\sqrt{2\pi}\mu} \exp(-\frac{1}{2}\frac{(x-y)^2}{\mu^2})$ with deterministic parameters $\mu \in \mathcal{P} = [10^{-3}, 10^{-1}]$. The objective is to evaluate affine approximations of processes $c(\mu, \omega) : [0, 1] \to \mathbb{R}$ of the form

$$c(x; \mu, \omega) = \int_{x-1/2}^{x+1/2} F(x, y; \mu)\, W(y; \omega) dy. \tag{3.15}$$

Thus, the trajectories are continuous with increasing smoothness for larger μ. Hence, we will approximate a set of functions with different smoothness properties. Figure 3.1 shows random trajectories for four values of μ, logarithmically equally spaced on $\mathcal{P}$.

In the RBM context, we use $c(\mu, \omega)$ as a stochastic coefficient of some PDE, e.g., $\nabla \cdot (c(\mu, \omega)\nabla u(\mu, \omega)) = f$. Here, $c(\mu, \omega)$ is constructed to exemplarily represent both the case of random functions and the case of noisy input data.

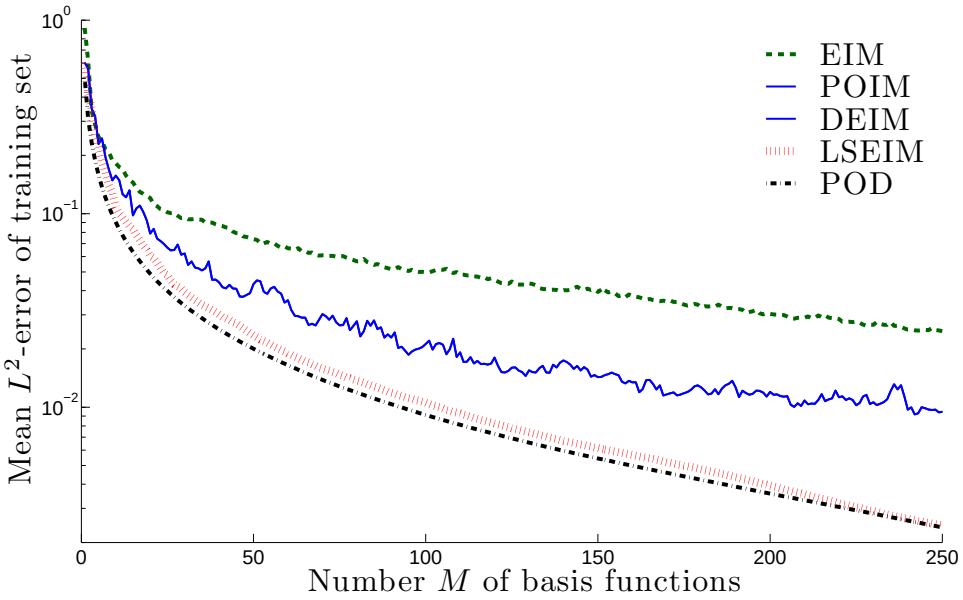


Figure 3.2: Average L_2-error of training trajectories.

We used a discretization of $\mathcal{N} = 400$ equidistant subintervals of the domain $D = [0, 1]$. For the construction of trajectories $c(\mu, \omega)$, we generated samples of the Wiener process W on the interval $[-1/2, 3/2]$ and evaluated (3.15). We used a training set $\Xi_{\text{train}} \subset \mathcal{P} \times \Omega$ with a total of 3000 samples, divided on 30 logarithmically spaced parameters $\mu \in \mathcal{P}$. This training set has been used to perform the POD, EIM, DEIM, POIM and LSEIM. We used POD eigenfunctions for the generation of the LSEIM basis.

Figure 3.2 shows the average L_2-error of all training trajectories $c(\mu, \omega)$, $(\mu, \omega) \in \Xi_{\text{train}}$. In this context, the POD provides the minimal error that can not be improved, i.e., the error of the L_2-projection on the L_2-optimal POD basis in the sense of (3.5). We can see that the average EIM-error convergence rate is far from optimal whereas the LSEIM almost reaches the minimum. Even though the POIM uses the same basis as the LSEIM, the error is noticeably larger. Thus, the coefficients are not adequately evaluated. For an error tolerance of 10^{-2}, 105 basis functions and 210 knots are needed for the LSEIM whereas the POIM needs 240 knots and basis functions and the EIM more than 350. In this case, the LSEIM needs even less knots than the POIM and would considerably save online time within an RBM. As shown in Section 3.3.3, the POIM and DEIM produce the

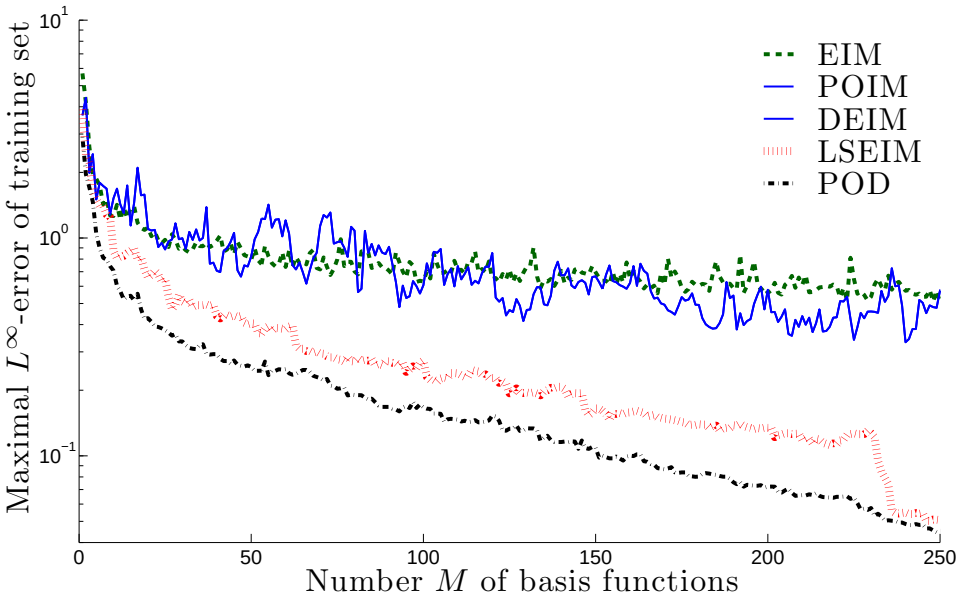


Figure 3.3: Maximal L_∞-error of training trajectories.

same results.

Figure 3.3 shows the maximal L_∞-error convergence of all considered methods. Here, the EIM and the POIM show a similar behavior. The errors decrease very slowly and significant variations can be observed. For the POIM, it is clear that the low convergence rate is caused by imprecise coefficients since the LSEIM still produces better results using the same basis. Even though the construction of the EIM is based on maximum L_∞-error minimization, the convergence is not monotonic either, since inappropriate basis functions may be selected.

Table 3.1: Effectivities of the L_∞-error estimators for 3200 test trajectories, $1 \leq M \leq \mathcal{N}-8$, and $M^+ = M+8$

	Minimal	Average	Maximal	$\% < 1$
EIM	0.373	3.025	9.148	0.022 %
POIM	0.320	3.411	14.849	0.014 %
LSEIM	0.446	2.431	6.992	0.024 %

In Table 3.1, we provide the effectivities of the introduced L_∞-error estimators, i.e., the ratio $\Delta_{M,M^+}/|c_M - c|$ of error estimator and real error. We used a test

set $\Xi_{\text{test}} \subset \mathcal{P} \times \Omega$ with a total of 3200 samples, divided on 32 logarithmically spaced parameters $\mu \in \mathcal{P}$. For all error estimators, we used 8 additional coefficients, i.e., $M^+ = M + 8$, and the table shows the minimal, average, and maximal effectivities of all test trajectories and all $M \leq \mathcal{N} - 8$. We can see that the error is not rigorous since effectivities less than one occur. However, the percentage of ineffective estimators, given in the last column, is very low. For higher accuracy, we could increase M^+. In most cases, the estimators denote error bounds and the effectivities are rather small, where the LSEIM yields slightly better results than the EIM and the POIM, respectively.

3.6 Conclusions

We demonstrated that it is useful to add POD eigenfunctions instead of snapshots to generate the EIM basis if these may be non-smooth. We proved that the described method produces the same approximation as the DEIM with less computational cost and provided error estimators for both methods. Furthermore, we showed that using more knots than basis functions improves the approximation quality and arrives at close to optimal results.

Chapter 4

Implicit Partitioning Methods for Unknown Parameter Domains

In the context of RBM for PDEs with deterministic parameter dependencies, it is common to split the parameter domain into several parts and construct separate reduced bases for each parameter subdomain [30, 31, 41]. It is assumed that the variation of the parametric coefficients of the PDEs and therefore the variation of the corresponding solutions become small on each subdomain. Then, only small numbers of basis functions are needed and the online cost of the RBM decreases.

In this chapter, we generalize the partitioning concepts developed for deterministic and compact parameter domains to arbitrary, possibly unknown parameter domains. No explicit description of the parameter domain — if existent at all — will be required, and no particular information about the problem is needed. Furthermore, we will show that our new implicit partitioning methods also outperform the existing methods for wide classes of problems even in the setting of known parameter domains.

In Section 4.1, we briefly introduce two different partitioning procedures for known, explicitly given parameter domains. The first method, the so-called p-Partitioning [41], requires the availability of an affine decomposition in the parameter whereas the second method, the hp-Partitioning [30, 31], is based upon the EIM and generates affine decompositions, i.e., collateral EIM bases, and partitions simultaneously. In Section 4.2, we introduce the general concept of unknown parameter domains and of affine decompositions with respect to unknown param-

eters. Furthermore, we introduce some necessary assumptions and requirements for our new implicit partitioning methods.

As the *hp*-Partitioning, the here introduced Implicit Partitioning Method (IPM) generates affine decompositions and partitions in parallel. We will develop two different concepts for the IPM. In Section 4.3, we introduce an IPM where the form of the subdomains is not fixed but depends on the used collateral basis size. The method is therefore called Moving Shapes (MS) IPM. Next, in Section 4.4, we develop IPMs where the forms of the subdomains are supposed to be stationary. These methods are called Fixed Shapes (FS) IPM. Finally, in Section 4.6, we provide several numerical examples and compare the different methods.

4.1 Preliminaries

We start introducing the partitioning concepts for known, deterministic, and compact parameter domains. Let $D \subset \mathbb{R}^d$ denote a bounded spatial domain and let $\mathcal{P} \subset \mathbb{R}^p$ be a compact parameter domain which is for now assumed to be a p-dimensional hypercube. Furthermore, let $c : D \times \mathcal{P} \to \mathbb{R}$, $(x; \mu) \mapsto c(x; \mu)$, denote a parametrized coefficient of an arbitrary PDE. Suppose detailed solutions of the PDE on a discrete Hilbert space X of dimension $\mathcal{N}$ are available, based upon any discretization scheme such as finite elements or finite differences. Let $X_N \subset X$ denote a reduced space of dimension N. Then, for the partitioning, we assume the availability of rigorous and efficiently evaluable error bounds $\Delta(\mu)$ of the error between the detailed and the reduced solution of the PDE for the parameter $\mu \in \mathcal{P}$.

We define $N_{\max}$ as the largest allowed basis size such that a certain maximal online run time for a reduced solution is not exceeded. At the same time, an error tolerance ε_{tol} is desired. Hence, the objective is to divide $\mathcal{P}$ into multiple subdomains and generate individual reduced bases such that

(i) the dimension of all reduced spaces is smaller than $N_{\max}$,

(ii) the maximal error on each subdomain does not exceed ε_{tol},

(iii) each parameter $\mu \in \mathcal{P}$ can be assigned efficiently to the right subdomain,

whereas the number of subdomains should be as small as possible.

Algorithm 4.1 p-Partitioning($\mathcal{P}^j$, $N_{\max}$, ε_{tol}, J)

```
1  create Ξ^j_train from P^j
2  for N = 1 to N_max do
3      S^j_RB,N = addBasisFunction(S^j_RB,N−1, Ξ^j_train)
4      Δ_N,max = getMaxErrorBound(S^j_RB,N, Ξ^j_train)
5      if Δ_N,max < ε_tol then
6          return S^j_RB,N, P^j
7      end if
8  end for
9  {P^{J+i} | i = 1, ..., 2^p} = refinePartition(P^j)
10 J_new = J + 2^p
11 for i = 1 to 2^p do
12     p-Partitioning(P^{J+i}, N_max, ε_tol, J_new)
13 end for
```

4.1.1 p-Partitioning

We first introduce the so-called p-Partitioning [41]. The "p" refers to "parameter" and distinguishes the method from other concepts such as time domain partitioning (t-Partitioning) [25] and a diversity of domain decomposition and related methods [1, 53, 58, 66]. For the p-Partitioning, it is assumed that the PDE already allows for an affine decomposition in the parameter $\mu \in \mathcal{P}$ which is either given or approximated using the EIM or similar techniques as described in Chapter 3.

The method starts with a coarse uniform grid on the p-dimensional hypercube $\mathcal{P}$ which defines the initial parameter domain partition. Hence, each subdomain itself defines a p-dimensional hypercube. For each initial subdomain, we call the procedure described in Algorithm 4.1, where $\mathcal{P}^j \subset \mathcal{P}$ denotes the current subdomain and J the total number of created subdomains. The algorithm recursively generates structs $\mathcal{S}^j_{\text{RB},N}$ for each subdomain $\mathcal{P}^j$ that include all the RB-related data, e.g., the basis itself, the RB system matrices and vectors, and the data that is necessary to evaluate the error bounds. We briefly describe the main steps.

In line 1, we create an appropriate set of training parameters $\Xi^j_{\text{train}} \subset \mathcal{P}^j$. In [41], an adaptive training set extension procedure is used for the Ξ^j_{train}. However, we can also assume a fixed training set without changing the theoretical aspects

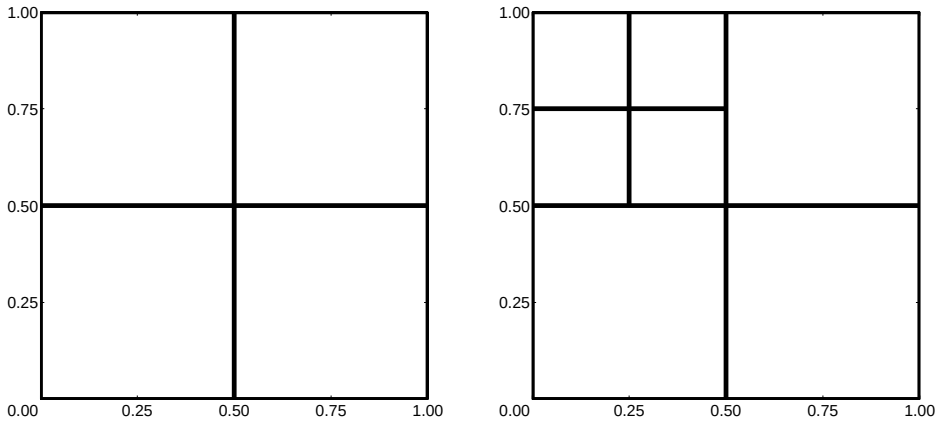


Figure 4.1: Two refinement steps using the p-Partitioning procedure for $\mathcal{P} = [0,1]^2$.

of the p-Partitioning.

From line 2 to 8, the reduced basis for the actual subdomain $\mathcal{P}^j$ is constructed. We first add a basis function to the reduced space in line 3 and update the RB-related data in $\mathcal{S}^j_{\mathrm{RB},N}$. For the basis selection, it is common to use a Greedy approach, i.e., the solution for the parameter in the training set that is so far worst approximated is added [73, 98]. For instationary problems, not the complete trajectory is used but only the first POD eigenfunction, based upon the error trajectory of the solutions at all time steps for the selected parameter. This procedure is denoted as POD-Greedy [42]. Next, in line 4, we evaluate the maximal error bound over all training samples. If this error bound is small enough, we return the actual subdomain and the corresponding reduced basis. Both are then stored for later use in the online stage. Otherwise, we repeat the procedure until $N_{\max}$ is reached.

If the error still exceeds the tolerance $\varepsilon_{\mathrm{tol}}$ for $N = N_{\max}$, the variation of the solutions on the current subdomain is too large. Hence, the current subdomain and the corresponding basis are discarded and we perform the refinement step in line 9. We divide the hypercube $\mathcal{P}^j \subset \mathbb{R}^p$ into 2^p "subhypercubes" of identical sizes, i.e., the edge length of the new hypercubes is half of the length of the edges of $\mathcal{P}^j$. Figure 4.1 shows two exemplary refinement steps for a two dimensional parameter domain $\mathcal{P} = [0,1]^2$.

Next, we set the number of subdomains from J to $J+2^p$ and recursively call the

procedure p-Partitioning$(\cdot)$ for all new subdomains. Note that the number J also includes already discarded subdomains. However, the indices j in Algorithm 4.1 and hence the total number J are only specified to facilitate the understanding of the procedure. In practice, the indexing will be done in a tree-based scheme. The algorithm returns only the RB data for leaf-subdomains, and all other reduced bases are not stored.

The assignment of a new parameter $\mu \in \mathcal{P}$ to the appropriate subdomain in the online stage can be done using the tree structure of the partition. For any point in a hypercube $\mathcal{P} \subset \mathbb{R}^p$ with 2^p subdomains as described above, it is possible to identify the subdomain where the point is located in $\mathcal{O}(p)$. This procedure can be repeated iteratively. Hence, assuming a well balanced partition tree of depth $\mathcal{O}(\log J)$, the assignment complexity reads $\mathcal{O}(\log J \cdot p)$.

One basic disadvantage of the proposed partitioning method is the increase of the offline run-time. During the refinement procedure, many reduced bases are discarded after $N_{\max}$ iterations. Each iteration requires the computation of a large number of reduced solutions and one detailed solution and is therefore expensive. Hence, it is desired to detect at an early stage if $N_{\max}$ basis functions will not suffice to adequately represent the solutions on the current subdomain. The maximal error for $N_{\max}$ basis functions can be predicted by extrapolating $\Delta_{N,\max}$, where the decay of the error is often assumed to be exponentially fast. The basis extension is stopped and the partition is directly refined as soon as the prediction indicates that we will not reach the error tolerance. Hence, Algorithm 4.1 is changed in the following way. After line 7, we add the following part:

$\Delta^{\text{pred}}_{N_{\max},\max} = \mathsf{getPredictedMaxErrorBoundAt}N_{\max}(\Delta_{1,\max}, \ldots, \Delta_{N-1,\max})$
if $\Delta^{\text{pred}}_{N_{\max},\max} > \varepsilon_{\text{tol}}$ **then**
  break
end if

Many superfluous computations are hereby avoided.

Compared to straightforward basis constructions without partitioning, the storage complexity increases. However, it is now possible to control the online complexity by choosing $N_{\max}$ as desired, although for instationary problems, the minimal choice of ε_{tol} is not independent of $N_{\max}$. If ε_{tol} is chosen too small, it can be neces-

sary to use more than $N_{\max}$ basis functions to cover the complexity of a trajectory over time even for a single parameter.

4.1.2 *hp*-Partitioning

Independently of the *p*-Partitioning, another similar method, called "*hp* certified RBM", has been proposed in [30]. The term "*hp*" is adopted from the finite element (FE) theory, where "*h*" refers to the mesh size and "*p*" to the polynomial degree of the local FE basis function which both are determined and refined adaptively. In the context of parameter domain partitioning, the "*h*" analogously represents the refinement of the partition and the "*p*" stands for the improvement of the basis on a subdomain, i.e., the selection of further reduced basis functions.

In [30] and [29], the *hp*-Partitioning has been introduced for stationary and instationary problems, respectively, for already affine problems. The methods differ only slightly from the *p*-Partitioning of Section 4.1.1. The main distinction are two different procedures for the splitting into subdomains, leading to theoretical convergence results for some special cases. In [31], the *hp*-Partitioning is introduced for non-affine problems and is connected to the EIM. Here, the *p*-refinement step refers to the selection of an additional collateral basis function for the EIM (cf. Section 3.2.1). In this section, we only describe the latter method since it shows some similarities to our implicit partitioning methods. Furthermore, we introduce the two splitting techniques that have also been used in the other publications about *hp*-Partitioning, the so-called anchor point splitting scheme and the gravity center splitting scheme.

In contrast to the *p*-Partitioning from Section 4.1.1, the *hp*-Partitioning is divided into two completely separate parts, the *h*-part with the refinement of the partition and the *p*-part with the basis construction. We introduce two separate error tolerances $\varepsilon^h_{\text{tol}}$ and $\varepsilon^p_{\text{tol}}$ and two maximal numbers $M^h_{\max}$ and $M^p_{\max}$ of collateral EIM basis functions for the *h*-part and the *p*-part. The error tolerances refer now to the EIM error. The *h*-indexed quantities are only employed to make the subdividing scheme cheaper whereas the *p*-indexed quantities refer to the actual desired values.

Algorithm 4.2 describes the general *h*-part of the *hp*-Partitioning that is common to both splitting schemes. Given an initial partition, we call the procedure

Algorithm 4.2 *hp*-Partitioning($\mathcal{P}^j$, $M^h_{\max}$, $\varepsilon^h_{\text{tol}}$, J)

1 create Ξ^j_{train} from $\mathcal{P}^j$
2 **for** $M = 1$ **to** $M^h_{\max}$ **do**
3 $\quad \{\mathcal{S}^j_{\text{EIM},M}, \mu^j_M\} = \mathsf{addBasisFunction}(\mathcal{S}^j_{\text{EIM},M-1}, \Xi^j_{\text{train}})$
4 $\quad \varepsilon_{M,\max} = \mathsf{getMaxError}(\mathcal{S}^j_{\text{EIM},M}, \Xi^j_{\text{train}})$
5 $\quad$ **if** $\varepsilon_{M,\max} < \varepsilon^h_{\text{tol}}$ **then**
6 $\quad\quad$ **return** $\mathcal{S}^j_{\text{EIM},M}, \mathcal{P}^j$
7 $\quad$ **end if**
8 **end for**
9 $\{\mathcal{P}^{J+i} \mid i = 1, \ldots, J_{\text{add}}\} = \mathsf{refinePartition}(\mathcal{P}^j, \mu^j_1, \ldots, \mu^j_{M_{\max}})$
10 $J_{\text{new}} = J + J_{\text{add}}$
11 **for** $i = 1$ **to** J_{add} **do**
12 $\quad$ *hp*-$\mathsf{Partitioning}(\mathcal{P}^{J+i}, M^h_{\max}, \varepsilon^h_{\text{tol}}, J_{\text{new}})$
13 **end for**

for each initial subdomain. The refinement and basis construction works again recursively. The relatively large error tolerance $\varepsilon^h_{\text{tol}}$ is used and only a small number $M^h_{\max}$ of maximal basis functions per subdomain is allowed. In that way, the construction of superfluous bases functions for subdomains that are discarded anyway is avoided. The total number of current subdomains is denoted by J.

Compared to Algorithm 4.1, the main difference is that we do not construct the RB system but EIM collateral bases and structs $\mathcal{S}^j_{\text{EIM},M}$ containing the complete EIM data, where j refers again to the subdomain and M to the number of basis functions. Hence, the procedure $\mathsf{addBasisFunction}(\cdot)$ in line 3 performs one iteration of the offline EIM construction as described in Algorithm 3.1, line 2 to 6. Additionally, it now returns the parameter that corresponds to the just selected basis function. These parameters are used for the new refinement procedures in line 9. Since no error estimators for the EIM can be evaluated during the construction of the collateral basis, the exact L_∞-error is evaluated in line 4 and used as termination condition in line 5.

Before we introduce the different refinement procedures that can be used in line 9, we briefly provide the second step of the *hp*-Partitioning, the *p*-part. The actual basis on each subdomain is constructed analogously to the EIM Algorithm

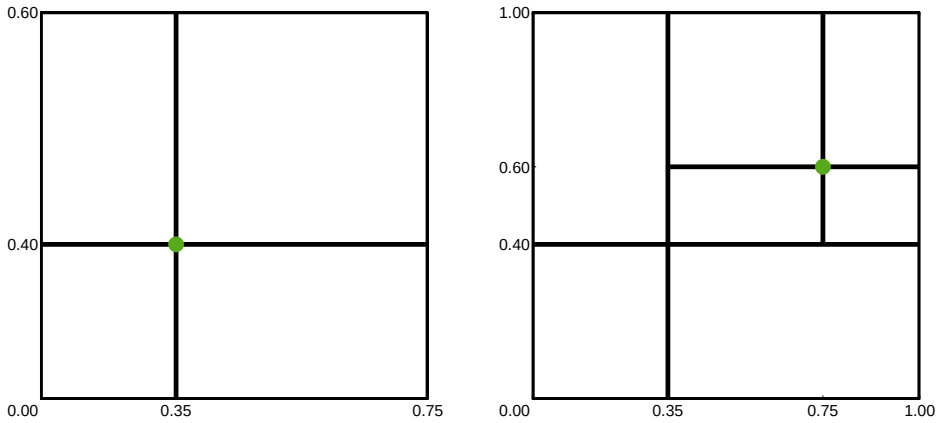


Figure 4.2: Two refinement steps using the gravity center splitting scheme for $\mathcal{P} = [0,1]^2$. Gravity centers $\bar{\mu}^1 = [0.35, 0.40]$ for the first step (left) and $\bar{\mu}^2 = [0.75, 0.60]$ for the second step (right).

3.1. For each final subdomain, we call Algorithm 3.1 and iterate until the small error tolerance $\varepsilon_{\text{tol}}^p$ or the maximal number $M_{\max}^p$ is reached.

Gravity Center Splitting Scheme

For the gravity center refinement procedure, it is assumed that that the parameter domain $\mathcal{P} \subset \mathbb{R}^p$ and each subdomain are given by a p-dimensional hypercube. In the refinement step, we cut the current subdomain $\mathcal{P}^j$ into $J_{\text{add}} = 2^p$ subhypercubes. As opposed to the p-Partitioning, these new subdomains are *not* equally sized. The splitting is now based on the so-called "gravity center" $\bar{\mu}^j$ which is evaluated using the parameters that correspond to the selected basis functions of the EIM in the subdomain $\mathcal{P}^j$,

$$\bar{\mu}^j := \frac{1}{M_{\max}^h} \sum_{M=1}^{M_{\max}^h} \mu_M^j.$$

Now, the gravity center denotes the (only) point of $\mathcal{P}^j$ that all 2^p new subdomains share, i.e., the coordinates of $\bar{\mu}^j$ define the splitting positions of $\mathcal{P}^j$. Figure 4.2 exemplarily shows two refinement steps using the gravity center splitting scheme for the square $\mathcal{P} = [0,1]^2$. First, the square is split based upon the gravity center $\bar{\mu}^1 = [0.35, 0.40]$. The subdomain in the upper right corner is then divided based

upon the gravity center $\bar{\mu}^2 = [0.75, 0.60]$.

As for the p-Partitioning, the online assignment of a new parameter $\mu \in \mathcal{P} \subset \mathbb{R}^p$ to the appropriate subdomain is done using a tree search. Only the gravity centers have to be stored to completely define the final partition as well as the partition tree. In each step, the identification of the next subdomain is of complexity $\mathcal{O}(p)$. Thus, for a well balanced tree of depth $\mathcal{O}(\log J)$, the assignment complexity reads $\mathcal{O}(\log J \cdot p)$ again.

Anchor Point Splitting Scheme

The anchor point splitting scheme divides the current parameter domain $\mathcal{P}^j$ into $J_{\text{add}} = 2$ subdomains, independently of its shape and dimension. For the splitting, it is assumed that one can define a distance measure $d : \mathcal{P} \times \mathcal{P} \to \mathbb{R}$ on the parameter domain. The two subdomains are then specified by the proximity to the parameters μ_1^j and μ_2^j — the so-called anchor points — that have been returned by the procedure addBasisFunction$(\cdot)$ in line 3 of Algorithm 4.2 and correspond to the two first selected EIM basis functions in the subdomain $\mathcal{P}^j$. Then, the new subdomains in line 9 of Algorithm 4.2 are defined in the following way,

$$\begin{aligned} \mathcal{P}^{J+1} &:= \{\mu \in \mathcal{P}^j \,|\, d(\mu, \mu_1^j) < d(\mu, \mu_2^j)\}, \\ \mathcal{P}^{J+2} &:= \{\mu \in \mathcal{P}^j \,|\, d(\mu, \mu_2^j) \leq d(\mu, \mu_1^j)\}. \end{aligned} \tag{4.1}$$

Each parameter $\mu \in \mathcal{P}^j$ is associated with the closest anchor point. Figure 4.3 exemplarily shows two refinement steps using the anchor point splitting scheme for the square $\mathcal{P} = [0, 1]^2$. In the first step (left), the anchor points $\mu_1^1 = [0.1, 0.1]$ and $\mu_2^1 = [0.9, 0.9]$ have been used such that the cross section of the new subdomains is given by the diagonal from the upper left to the lower right corner of $\mathcal{P}$. In the second step, the anchor points $\mu_1^2 = [0.1, 0.1]$ and $\mu_2^2 = [0.8, 0.1]$ lead to the separation parallel to the y-coordinate at $x = 0.45$.

Since only two anchor points are needed for the next refinement step, it is enough to set $M_{\max}^h = 2$. Furthermore, the two subdomains can inherit the basis function of the "parent" domain that corresponds to their respective anchor point. In other words, for the domain $\mathcal{P}^j$ with the two "child" subdomains $\mathcal{P}^{J+1}$ and $\mathcal{P}^{J+2}$ as defined in (4.1), we have $\mu_1^{J+1} := \mu_1^j$ and $\mu_1^{J+2} := \mu_2^j$. Thus, only one more iteration has to be performed for each new subdomain. In the example in Figure 4.3, we already applied this simplification and used $\mu_1^2 = \mu_1^1$.

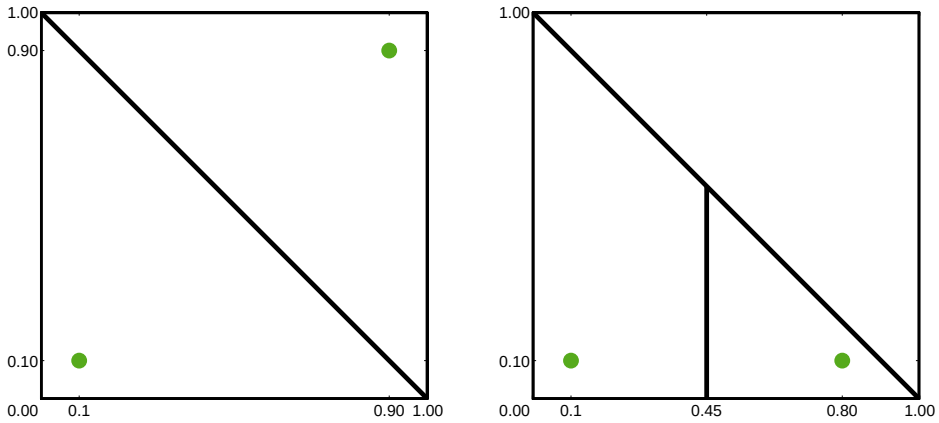


Figure 4.3: Two refinement steps using the anchor point splitting scheme for $\mathcal{P} = [0,1]^2$. Anchor points for the first (left) and second refinement step (right).

As before, we use a tree search in the online stage to find the appropriate subdomain for a given new parameter $\mu \in \mathcal{P}$. We iteratively select the nearest anchor point and "move" to the corresponding subdomain until a final subdomain is reached. For $\mathcal{P} \subset \mathbb{R}^p$, one can use the Euclidean distance measure. Then, the evaluation of the distances to the anchor points is of complexity $\mathcal{O}(p)$. Assuming a balanced tree of depth $\mathcal{O}(\log J)$, the total tree search is again of complexity $\mathcal{O}(\log J \cdot p)$.

Compared to the p-Partitioning and the hp-Partitioning based upon the gravity center splitting, the anchor point splitting produces the most flexible shapes and is the cheapest since during the refinement procedure only $M^h_{\max} = 2$ basis functions are needed for each subdomain. Furthermore, only one basis extension is required per subdomain by reusing the anchor points.

However, for the optimal application of the hp-Partitioning, the choice of $\varepsilon^h_{\text{tol}}$ is crucial. Using a tolerance that is too large, the resulting partition may not be fine enough and it may be impossible to reach the tolerance $\varepsilon^p_{\text{tol}}$ with $M^p_{\max}$ basis functions in the p-part. Then, more refinement steps are necessary and the so far constructed bases have to be discarded. Still, it is possible to apply the error prediction method as presented for the p-Partitioning to decrease the number of superfluous computations.

4.2 Partitioning of Unknown Parameter Domains

4.2.1 Unknown Parameter Domains

Let us start with the illustration of the concept of unknown parameter domains using some practical examples. First, one may consider coefficient functions of PDEs that are based upon measurements. On the one hand, underlying parameters can be hidden since the information of the system that produces the measured outcome is not completely accessible. On the other hand, the measured input functions could be completely non-parametric and merely belong to a common class of functions in terms of boundedness, regularity, and/or similar shape. Another application of unknown parameters are stochastic inputs, where the "parameter domain" can be seen as a set Ω of stochastic events that does not imply a feasible metric. Hence, the theory of compact parameter domains does not apply. As an example of such events, one may consider the porosity structure of any physical medium such as sandstone (cf. Sections 5.8 and 6.5) or Li-ion batteries.

In general, any input function in discretized form can be adopted to an $\mathcal{N}$-dimensional parameter setting, where $\mathcal{N}$ denotes the number of degrees of freedom of the discretization. However, since the online parameter assignment of the presented partitioning methods of Section 4.1 depends on the dimension of the parameter domain, they are inappropriate for such a setting.

In the following, it is assumed that the input coefficient functions can be obtained without the detailed knowledge of any underlying parameter or stochastic event. Hence, no information about the parameter domain is required, and therefore, no distance measures on the parameter domain can be assumed to exist. We now define the family of possible input functions by

$$\mathcal{M} := \{c(\mu) : D \to \mathbb{R} \,|\, \mu \in \mathcal{P}\}, \tag{4.2}$$

where $D \subset \mathbb{R}^d$ denotes a bounded spatial domain. The parameter $\mu \in \mathcal{P}$ can also be interpreted as a reference to an arbitrary real life event that underlies the function $c(\mu)$, or just as an index to the associated $c(\mu) \in \mathcal{M}$. Alternatively, it could also be seen as a parameter vector of the possibly infinite dimension of $\mathcal{M}$. In any case, μ is not a parameter in the classical sense and the p- or hp-Partitioning are not applicable.

Another interpretation could be to consider the whole function $c(\mu)$ as a parameter, i.e., to consider a parameter function $\mu(x)$ in a certain function space $\mathcal{M}$. The subsequent theory and methods remain valid for such cases.

4.2.2 Affine Decomposition for Unknown Parameters

For the application of the EIM even for unknown parameter domains or arbitrary sets of functions, and for the applicability of partitioning methods, we postulate:

Assumption 4.1. *A mechanism is available that delivers arbitrarily many functions $c(\mu)$, $\mu \in \mathcal{P}$, from the family of functions $\mathcal{M}$ as defined in* (4.2)*. For any given $\varepsilon > 0$, it is possible to create a finite training set of functions $\mathcal{M}_{\text{train}} \subset \mathcal{M}$ of cardinality $n_{\text{train}} \in \mathbb{N}$ that sufficiently covers the variety of $\mathcal{M}$ up to the maximal error tolerance ε, i.e.,*

$$\sup_{c(\mu)\in\mathcal{M}} \inf_{v\in span(\mathcal{M}_{\text{train}})} \|c(\mu) - v\|_X \leq \varepsilon \tag{4.3}$$

for a given norm $\|\cdot\|_X$. Furthermore, let $\mathcal{M}$ be replaced by any subset $\mathcal{M}^0 \subset\subset \mathcal{M}$ with significantly less variation, i.e., of less complexity. Then, $\mathcal{M}_{\text{train}}$ can be replaced by a subset $\mathcal{M}^0_{\text{train}} \subset\subset \mathcal{M}_{\text{train}}$ of significantly less cardinality $n^0_{\text{train}} \ll n_{\text{train}}$ such that (4.3) *still holds.*

Now, the offline and online EIM Algorithms 3.1 and 3.2 can directly be adopted for our case. Instead of a training parameter set Ξ_{train} for the Greedy step (in line 2 of Algorithm 3.1), we can directly use the training functions $\mathcal{M}_{\text{train}}$. Let $Q_M = \{q_1, \ldots, q_M\}$ be a given collateral basis and let $T_M = \{t_1, \ldots, t_M\}$ be the EIM interpolation points. For any function $c = c(\mu) \in \mathcal{M}$, we can evaluate the coefficients $\boldsymbol{\theta}_M(c) = (\theta_i(c))_{i=1}^M$ for the affine approximation $c_M^{\text{EIM}} = \sum_{m=1}^M \theta_m(c) q_i$ using the linear system (3.6) without the knowledge of a possibly underlying parameter. We evaluate the vector $\boldsymbol{c}_M := (c(t_i))_{i=1}^M$ and the triangular matrix $B_M = (q_j(t_i))_{i,j=1}^M$ such that $\boldsymbol{\theta}_M(c) = B_M^{-1}\boldsymbol{c}_M$.

4.2.3 Implicit Partitioning Problem Formulation

We now formulate the tasks and the main idea of the IPM. Input functions that are based upon unknown parameters naturally do not directly admit for an affine

decomposition. Hence, the partitioning is connected to the EIM as we have already seen for the *hp*-Partitioning. We define the implicit partitioning problem.

Problem 4.2 (Implicit Partitioning Problem)**.** For a family of input functions $\mathcal{M}$ that suffices Assumption 4.1, create a partition of the parameter domain,

(a) without the use of an explicit description of either $\mathcal{P}$ or $\mathcal{M}$,

(b) without an explicit description of the partitions and subdomains,

(c) with efficient and suitable assignments of new input functions $c(\mu)$.

For each subdomain, create separate affine decompositions with respect to the unknown parameter as described in Section 4.2. The partition is supposed to be fine enough such that

(d) the affine approximations are precise up to a tolerance ε_{tol},

(e) the number of collateral basis functions per subdomain does not exceed $M_{\max}$.

The basic idea of the following implicit partitioning methods is the construction of several EIM bases that cover different parts of the family of input functions $\mathcal{M}$. As opposed to the *p*- and *hp*-Partitioning, the splitting of the parameter domain is based upon the proximity of functions in $\mathcal{M}$ to the spaces spanned by the different collateral EIM bases and not on geometrical aspects of the parameter domain. In the offline stage, during the construction of the collateral basis functions, the proximity can directly be based upon the approximation error. In the online stage, we have to use the error estimates to fulfill the efficiency requirement of the Implicit Partitioning Problem 4.2(c).

Under Assumption 4.1, it is possible to generate a training set of functions $\mathcal{M}_{\text{train}} \subset \mathcal{M}$ of cardinality $n_{\text{train}} \in \mathbb{N}$ that sufficiently covers the complexity of $\mathcal{M}$. Furthermore, the second part of Assumption 4.1 assures that a partitioning based upon a training set $\mathcal{M}_{\text{train}}$ is possible under the condition that $\mathcal{M}$ itself can be split into several parts of less complexity.

In fact, the presented implicit partitioning methods can rather be seen as a partitioning of the family $\mathcal{M}$ or of the space spanned by $\mathcal{M}$. Thus, functions $c(\mu)$ are assigned to an appropriate subspace of $\operatorname{span}(\mathcal{M})$ rather than μ is assigned to a subdomain of $\mathcal{P}$. However, for an easier understanding, we often stay in the

parameter setting and still refer to parameters and subdomains. We construct the structs $\mathcal{S}^j_{\mathrm{EIM},M}$, $j = 1, \ldots, J$, that contain the complete EIM data for each subdomain, respectively. The term "struct" is adopted from programming languages like C where a struct denotes a single structured data type that unites a set of components of different data types. Here, $\mathcal{S}^j_{\mathrm{EIM},M}$ also defines the subspaces $\mathcal{M}^j$ of dimension M which correspond to the parameter subdomains $\mathcal{P}^j$, $j = 1, \ldots, J$. In the following, we just refer to subdomain j and mean the subdomain defining components $\mathcal{P}^j$, $\mathcal{M}^j$, or $\mathcal{S}^j_{\mathrm{EIM},M}$. For a better illustration of the methods, we also use parametric functions for explicitly given parameter domains.

4.3 Moving Shapes IPM

We introduce different implicit partitioning procedures. As mentioned before, the common approach is the construction of several EIM bases that are supposed to cover different parts of the family of input functions $\mathcal{M}$. The first procedure, the Moving Shapes (MS) Implicit Partitioning Method (IPM), simultaneously generates the number of J EIM bases for a previously fixed number J of subdomains. It is desired that the partition is formed such that the complexity of $\mathcal{M}$ is equally distributed on the J different subdomains and the least possible number of basis functions is obtained. This is achieved by letting the subdomains reshape in each iteration instead of using a fixed partition. Thus, the actual partition depends on the used number M of basis functions.

4.3.1 Outline of the Method

The MS IPM is described in Algorithms 4.3 and 4.4. Let J denote the desired number of subdomains and let $\varepsilon_{\mathrm{tol}} > 0$ be the desired approximation error tolerance. Furthermore, let the set of training parameters be given by $\{\mu_1, \ldots, \mu_{n_{\mathrm{train}}}\}$ such that the set of training functions reads $\mathcal{M}_{\mathrm{train}} = \{c(\mu_n) \,|\, n = 1, \ldots, n_{\mathrm{train}}\}$. Algorithm 4.3 generates J structs $\mathcal{S}^j_{\mathrm{EIM},M}$, $j = 1, \ldots, J$, $M \in \mathbb{N}$, containing the EIM data for the corresponding subdomains. Since the number of subdomains and the error tolerance $\varepsilon_{\mathrm{tol}}$ are (at least for now) fixed, we do not set a maximal number of basis functions per subdomain, differently to the *hp*-Partitioning where $M_{\max}$ and $\varepsilon_{\mathrm{tol}}$ were fixed and J was flexible.

Algorithm 4.3 MovingShapesIPM($\mathcal{M}_{\text{train}}$, ε_{tol}, J)

1 **set** $M = 0$
2 **repeat**
3 $M = M + 1$
4 **if** M==1 **then**
5 $\mathcal{S}^0_{\text{EIM},J+1}$ = doInitialEIM($\mathcal{M}_{\text{train}}$, $J+1$)
6 $\{\mathcal{S}^1_{\text{EIM},1}, \ldots, \mathcal{S}^J_{\text{EIM},1}\}$ = initialFirstBasisFunction($\mathcal{S}^0_{\text{EIM},J+1}$, J)
7 **else**
8 **for** $j = 1$ **to** J **do**
9 $\mathcal{S}^j_{\text{EIM},M}$ = addBasisFunction($\mathcal{S}^j_{\text{EIM},M-1}$, $\mathcal{M}^j_{\text{train}}$)
10 **end for**
11 **end if**
12 $\{\mathcal{I}^1_M, \ldots, \mathcal{I}^J_M\}$ = getOfflineAssignment($\mathcal{S}^1_{\text{EIM},M}, \ldots, \mathcal{S}^J_{\text{EIM},M}, \mathcal{M}_{\text{train}}$)
13 **for** $j = 1$ **to** J **do**
14 $\mathcal{M}^j_{\text{train}} = \{c(\mu) \in \mathcal{M}_{\text{train}} \,|\, \mu \in \mathcal{I}^j_M\}$
15 $\varepsilon^j_{M,\max}$ = getMaxError($\mathcal{S}^j_{\text{EIM},M}$, $\mathcal{M}^j_{\text{train}}$)
16 **end for**
17 **until** $\max\limits_{j\in\{1,\ldots,J\}} \{\varepsilon^j_{M,\max}\} < \varepsilon_{\text{tol}}$
18 **return** $\{\mathcal{S}^1_{\text{EIM},M}, \ldots, \mathcal{S}^J_{\text{EIM},M}\}$

We start the description of the MS IPM with the initialization of the EIM structs $\mathcal{S}^j_{\text{EIM},1}$, $j = 1, \ldots, J$, in the first iteration of the loop in Algorithm 4.3, for $M = 1$. In line 5, we perform $J+1$ steps of the normal EIM, as described in Section 4.2.2, based upon the training set $\mathcal{M}_{\text{train}}$ and without any partitioning. We refer to this step as initial EIM and denote the resulting EIM struct by $\mathcal{S}^0_{\text{EIM},J+1}$. Then, in line 6, we discard the first basis function of $\mathcal{S}^0_{\text{EIM},J+1}$ and distribute the remaining J functions that have been selected by the initial EIM to the EIM structs $\mathcal{S}^j_{\text{EIM},1}$, $j = 1, \ldots, J$, as initial basis functions, respectively.

Neglecting the first basis function of the initial EIM is not crucial to the basis assignment. In our experiments, it led to a more balanced initial distribution of the complexity to the subdomains.

In line 12 of Algorithm 4.3, we call the procedure getOfflineAssignment$(\cdot)$ that is further described in Algorithm 4.4. For each subdomain j, the procedure returns a

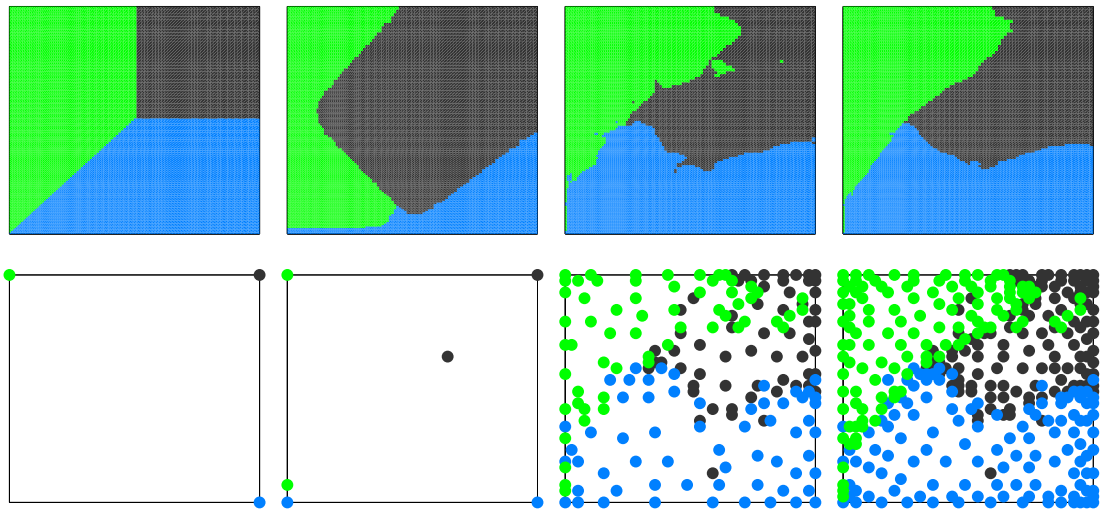
Algorithm 4.4 getOfflineAssignment($\mathcal{S}^1_{\mathrm{EIM},M}, \ldots, \mathcal{S}^J_{\mathrm{EIM},M}$, $\mathcal{M}_{\mathrm{train}}$)

```
1  I^1 = ... = I^J = {}
2  for n = 1 to n_train do
3      for j = 1 to J do
4          ε^j_M(μ_n) = getError(S^j_{EIM,M}, c(μ_n))
5      end for
6      i = arg inf_j {ε^j_M(μ_n) | j = 1, ..., J}
7      I^i_M = I^i_M ∪ {μ_n}
8  end for
```

set of assigned parameters $\mathcal{I}^j_M$ that refer to the corresponding functions in $\mathcal{M}_{\mathrm{train}}$, where the assignment is based upon the EIM approximation error. In detail, for a given parameter μ_n, $n \in \{1, \ldots, n_{\mathrm{train}}\}$, we evaluate the EIM approximation error of the corresponding function $c(\mu_n)$ in all subdomains. This is performed in Algorithm 4.4, line 3 to 5. Then, the parameter is assigned to the subdomain that best approximates $c(\mu_n)$ in line 6 and 7. Note that we distinguish $\mathcal{P}^j$ used for the p- and hp-Partitioning from $\mathcal{I}^j$. While $\mathcal{I}^j$ denotes a discrete set of parameters, $\mathcal{P}^j$ provides the explicit description of the complete subdomain j.

The further steps work very similar to Algorithm 4.2, but simultaneously for all subdomains. In line 15 of Algorithm 4.3, we evaluate the maximal error on each subdomain, or more precisely, the maximal error out of the set of currently assigned functions $\mathcal{M}^j_{\mathrm{train}} := \{c(\mu) \in \mathcal{M}_{\mathrm{train}} \,|\, \mu \in \mathcal{I}^j_M\}$. In line 17, we check if all maximal errors already fall below the tolerance $\varepsilon_{\mathrm{tol}}$. We do not stop the basis extensions until convergence on all subdomains is obtained, i.e., even if the tolerance is reached on a certain subdomain, we add more basis functions if the error on other subdomains still exceeds the desired value. For $M > 1$, the basis extension is done in line 9. As for the hp-Partitioning, we select the so far worst approximated function of the subdomain. Here, this means that for subdomain j, the next basis function is selected out of the set of currently assigned functions $\mathcal{M}^j_{\mathrm{train}}$ that are represented by the corresponding parameters $\mathcal{I}^j_{M-1}$.

A new effect in comparison to the hp-Partitioning is that the basis extension also changes the shape of the partitions since the assignment of parameters is based upon the EIM approximation error. The selection of a new basis function $c(\mu^j_M)$



(a) $M = 1$ (b) $M = 2$ (c) $M = 60$ (d) $M = 120$

Figure 4.4: MS IPM subdomains (top row) and selected parameters for basis extension (bottom row) for four different basis sizes M.

for some μ_M^j located close to the boundary of the subdomain j yields a movement of the respective shape towards the just selected parameter. Functions $c(\mu)$ for μ close to μ_M^j will be assigned to subdomain j in the next iteration.

This effect is illustrated in Figure 4.4. It provides the result of the MS IPM for an explicitly given parametric function $c : \mathcal{D} \times \mathcal{P} \to \mathbb{R}$ on the spatial domain $\mathcal{D} = [0, 1]^2$ and with parameters $\mu = (\mu_1, \mu_2) \in \mathcal{P} = [0.3, 0.7]^2$, given by

$$c(x; \mu) = e^{-50\left((x_1-\mu_1)^2+(x_2-\mu_2)^2\right)}. \tag{4.4}$$

We used a uniform discretization of $\mathcal{D}$ with $\mathcal{N} = 2601$ degrees of freedom and $n_{\text{train}} = 1600$ logarithmically distributed parameter samples. In detail, Figure 4.4 shows the partitions of the parameter domain after $M = 1$, 2, 60, and 120 iterations in the top row with a resolution of $40 \cdot 40$ pixels. The respective parameters that have been selected for the bases extensions are provided in the bottom row. It can be seen that the shapes of the subdomains change especially during the first iterations. Later, the changes are rather small and the shapes seem to converge. In the second step, for $M = 2$, the black part selected a basis function that corresponds to the parameter from the lower left corner of the subdomain for $M = 1$. Therefore, it "takes over" huge parts of the other subdomains. In the iterations between $M = 60$ and $M = 120$, the basis extensions are mostly based upon

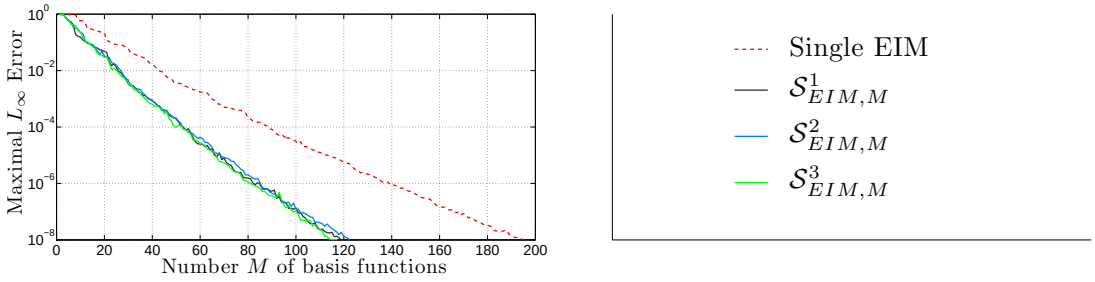


Figure 4.5: Convergence of the MS IPM for $J = 3$ compared to a single EIM.

parameters inside the subdomains and therefore, the boundaries do not change significantly.

The objective of the subdomain reshaping is a more effective use of the basis functions. For fixed shapes, the first basis functions are usually selected at the border of subdomains. Consequently, adjacent subdomains would select basis functions that cover the same area. Furthermore, the reshaping results in a good distribution of the complexity of $\mathcal{M}$ on the different subdomains. The subdomains are likely to be formed such that the respective numbers of basis functions necessary for a given approximation tolerance differs only very slightly. In Figure 4.5, we confirm this assumption for the given example. The figure shows the error convergence of a single EIM without partitioning and the convergence result using the MS IPM and $J = 3$ subdomains. More examples are provided in Section 4.6.

It can be observed in Figure 4.4(b) that two subdomains can select basis functions close to each other in the same step (see the green and blue subdomain). This is not optimal since both functions cover again the same part of $\mathcal{M}$ and less than possible information is therefore added in this iteration. However, it is very difficult to avoid such cases. The straightforward approach would be to successively extend the bases. Before the next subdomain selects a parameter, a reassignment is performed based upon the new approximation errrors. However, this procedure does not work properly. Especially at the beginning of the procedure, the extension of only one basis by one function yields very unbalanced shapes. The larger basis outperforms the others on most of the parameter domain and therefore covers a too large area. Certainly, it is possible to develop more sophisticated methods to avoid such cases. However, a general heuristic that works as a black box for all kind of input functions is not known but would be desired in the case of unknown

Algorithm 4.5 getOnlineAssignment($\mathcal{S}^1_{\mathrm{EIM},M^+}, \ldots, \mathcal{S}^J_{\mathrm{EIM},M^+}$, $c(\mu)$, M, M^+)

```
1  for j = 1 to J do
2      θ^j_{M+}(μ) = getCoefficients(S^j_{EIM,M+}, c(μ))
3      Δ^j_{M,M+}(μ) = Σ_{m=M+1}^{M+} |θ^j_m(μ)|
4  end for
5  i = arg inf_j {Δ^j_{M,M+}(μ) | j = 1, ..., J}
6  return {i, θ^i_{M+}(μ)}
```

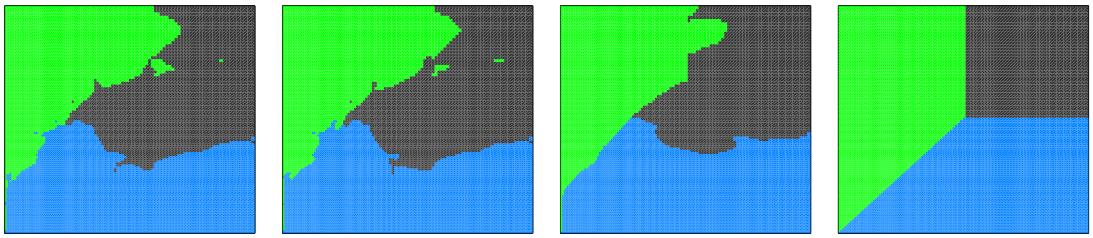
parameters.

Even though we obtain very balanced convergence rates, we can not completely prevent that two subdomains partially cover the same part of the parameter domain. It can be seen in Figure 4.4(c) and 4.4(d) that some of the selected basis functions are separated and enclosed by a different subdomain. However, for other values of M, these basis functions are within their respective subdomain and therefore necessary to obtain best approximation qualities with a minimal basis size. Furthermore, it is not possible to discard such functions from the basis even for values of M where they are separated from their subdomain. In other words, they still play an important role for the approximation quality.

Let $\mathcal{N}$ be the number of degrees of freedom of the discretized functions in $\mathcal{M}$. Then, the complexity of an iteration in the offline stage of the MS IPM consists of $\mathcal{O}(JM^2 \cdot n_{\mathrm{train}})$ for the computation of the approximations of the training samples in $\mathcal{M}$, $\mathcal{O}(JM^2 \cdot n_{\mathrm{train}} \cdot \mathcal{N})$ for the evaluation of the EIM errors, and $\mathcal{O}(J \cdot n_{\mathrm{train}})$ to assign the training snapshots to the subdomains. Thus, the total complexity is given by $\mathcal{O}(JM^2 n_{\mathrm{train}} \mathcal{N})$.

4.3.2 Online Assignment

In the online stage, it is not possible to evaluate the exact EIM approximation errors independently of the dimension $\mathcal{N}$. Hence, the assignment is now based upon the EIM error estimator. The straightforward procedure is given in Algorithm 4.5. For a new parameter μ and an input function $c(\mu)$, we evaluate the coefficients $\boldsymbol{\theta}^j_{M^+}(\mu)$ and error bounds $\Delta^j_{M,M^+}(\mu)$ for all $j = 1, \ldots, J$. Then, we select the subdomain with the smallest error estimator. The algorithm returns the selected subdomain i and the corresponding coefficients $\boldsymbol{\theta}^i_{M^+}(\mu)$ that can be used for the



(a) "True" partition (b) Online assignment using Alg. 4.5 (c) Online assignment using Alg. 4.6 and $\varepsilon_{\text{tol}} = 10^{-4}$ (d) First assignment trial in Alg. 4.6 using $\arg\max |\theta_1^j|$

Figure 4.6: MS IPM online assignments for $M = 60$ and $M^+ = 66$

further processing of the input function $c(\mu)$.

It is clear that the assigned subdomain is not necessarily optimal in the sense of the real error. However, it is not essential that we hit the best subdomain but to select a sufficiently precise approximation. Figure 4.6(b) shows the online assignment based upon the smallest error estimator for the example provided in (4.4). The result is shown for $M = 60$ basis functions and the error estimator uses 6 additional coefficients, i.e., $M^+ = 66$. In comparison to the "true" partition in Figure 4.6(a), based upon the exact EIM approximation error, only minor deviations can be observed. The use of more than 6 coefficients for the error estimates would furthermore lead to results closer to the "true" partition.

Online Complexity

The online complexity for the assignment of a new parameter $\mu \in \mathcal{P}$ to the appropriate subdomain according to Algorithm 4.5 consists of $\mathcal{O}(JM)$ for the evaluation of $c(\mu) \in \mathcal{M}$ at the interpolation points, $\mathcal{O}(JM^2)$ for the computation of the coefficients and the error bounds, and $\mathcal{O}(J)$ for the actual assignment to the subdomain. Thus, the total complexity reads $\mathcal{O}(JM^2)$, where it has been assumed that $M^+ = \mathcal{O}(M)$.

Compared to the *p*- and *hp*-Partitioning, the online complexity increased significantly. For both methods, the parameter assignment is independent of the computation of the coefficients and error bounds which yields an additive term of $\mathcal{O}(M^2)$ instead of a multiplication. Furthermore, the number of subdomains

Algorithm 4.6 getFastOnlineAssignment($\mathcal{S}^1_{\mathrm{EIM},M^+}, \ldots, \mathcal{S}^J_{\mathrm{EIM},M^+}$, $c(\mu), M, M^+$)

```
1  for j = 1 to J do
2      θ_1^j(μ) = getCoefficients(S^j_{EIM,1}, c(μ), 1)
3  end for
4  {j_1, ..., j_J} = sortCoefficientsDescending(|θ_1^1(μ)|, ..., |θ_1^J(μ)|)
5  for k = 1 to J do
6      θ^{j_k}_{M+}(μ) = getCoefficients(S^{j_k}_{EIM,M+}, c(μ), M+)
7      Δ^{j_k}_{M,M+}(μ) = Σ_{i=M+1}^{M+} |θ_i^{j_k}(μ)|
8      if Δ^{j_k}_{M,M+}(μ) < ε_tol then
9          return {j_k, θ^{j_k}_{M+}(μ)}
10     end if
11 end for
```

J enters only logarithmically. Yet, the assignment according to Algorithm 4.5 is independent of the dimension of the parameter domain.

Nevertheless, the online complexity of the MS IPM is acceptable. On the one hand, the number M of basis functions decreases with increasing number J of subdomains. In the current example, the run-time is approximately constant in J. On the other hand, the main complexity in the context of RB methods commonly amounts to $\mathcal{O}(M^2 \cdot N^2 + N^3)$, where N denotes the number of basis functions for the reduced basis. Since separate reduced bases are constructed for each subdomain, N is decreasing in J, too. Hence, the most expensive computations in the RB context decrease significantly.

Improved Online Complexity

The key requirement of the assignment is to obtain approximations that fulfill a certain error tolerance $\varepsilon_{\mathrm{tol}}$ and not to find the best subdomain. As a consequence, it is possible to break the loop over j in Algorithm 4.5 as soon as $\Delta^j_{M,M^+}(\mu)$ falls below $\varepsilon_{\mathrm{tol}}$ for any value of j. Then, the average online complexity is already reduced to half. In Algorithm 4.6, we present a heuristic that provides a more suitable search order of the subdomains than just checking the error estimators step by step.

The EIM is generated in a form such that the importance of the basis functions decreases in M. In other words, the coefficients of the first basis functions are usually larger than the following ones. At the same time, using a collateral basis that does not fit to the input data, the coefficients are rather equally distributed over all basis functions and the first coefficients are therefore comparatively small.

We use this effect for the search order heuristic. In line 2 of Algorithm 4.6, we evaluate only the first coefficients θ_1^j of the affine approximations of a given input function $c(\mu)$ for all subdomains $j = 1, \ldots, J$. In line 4, we sort these coefficients in descending order with respect to their absolute values and return an ordered list of subdomains. Then, we iteratively check if the error estimator of the subdomains fall below the tolerance ε_{tol}, starting with the subdomain with the largest coefficent θ_1^j. Once we find the first subdomain that approximates $c(\mu)$ sufficiently precise, we return the subdomain and the corresponding coefficients $\boldsymbol{\theta}_{M^+}^{j_k}(\mu)$ for the affine approximation.

Figure 4.6(c) shows the online assignment based upon Algorithm 4.6 for the error tolerance $\varepsilon_{\text{tol}} = 10^{-4}$. Again, we used $M = 60$ basis functions and an additional number of 6 coefficients for the error estimators. The partition reveals some larger deviations compared to the "true" partition 4.6(a) and the direct assignment 4.6(b) based upon Algorithm 4.5, respectively. However, for all parameters, both the error estimator and the true error fall below ε_{tol}.

In Figure 4.6(d), the result of the heuristic of Algorithm 4.6 is provided. It shows the first assignment attempt, i.e., the assignment based upon the largest first coefficient. We can see that large parts coincide with the assignment in Figure 4.6(c). In fact, 80.8% of the parameters in Figure 4.6(d) are associated to the same subdomain as in Figure 4.6(c) and are therefore directly assigned after just one iteration. Hence, in most cases, the online complexity reduces to $\mathcal{O}(J + M^2)$. For another 17.6%, we need two attempts until a subdomain is found that approximates the corresponding function sufficiently well. For only 1.6% of the parameters, we have to evaluate the coefficients for all subdomains.

We conclude that the alternative assignment procedure determines appropriate subdomains very fast. In several examples (see also Section 4.6), we observed that the great majority of the parameters are assigned in the first attempt. Furthermore, for examples where the first coefficient is not sufficient for a fast assignment,

it is also possible to use more than one. In fact, this procedure is in some way the opposite to the assignment based upon the error bounds. Instead of the smallest error bound, i.e., the smallest coefficients $|\theta_m^j|$ for some large values of m, we take the largest coefficients $|\theta_m^j|$ for some small m.

4.3.3 Refinement Procedure

So far, we fixed the number of subdomains in advance, whereas for many applications, a certain maximal number $M_{\max}$ of basis functions is desired and the necessary number of subdomains is unknown. Thus, we start the MS IPM with an initial guess J_0 of needed subdomains. Once we detect that $M_{\max}$ will be reached but the error still exceeds the tolerance ε_{tol}, we need a refinement of the partition. In contrast to the p- and hp-Partitioning, it is not possible for the MS IPM to directly divide a subdomain into several parts. Hence, a refinement now yields to a complete restart of the procedure with an increased number of subdomains.

It is too expensive to perform the complete MS IPM until $M_{\max}$ is reached before a refinement is performed. However, we can adopt the ideas of both p- and hp-Partitioning. As for the hp-Partitioning in Section 4.1.2, it is possible to define additional quantities $M_{\max}^h \ll M_{\max}$ and $\varepsilon_{\text{tol}}^h \gg \varepsilon_{\text{tol}}$. Then, we perform the MS IPM using the new tolerance, the new maximal number of basis functions, and an initial number J of subdomains. Once $M_{\max}^h$ is reached but the error does not fall below $\varepsilon_{\text{tol}}^h$, we increase J by one and iterate the procedure. After convergence, we finally restart the MS IPM using the before detected number of subdomains and the actually desired quantities ε_{tol} and $M_{\max}$. However, the procedure may still be expensive and it is very difficult to define $\varepsilon_{\text{tol}}^h$ and $M_{\max}^h$ such that the final number of subdomains is indeed sufficient. Hence, it may happen that additional expensive refinements are needed.

Alternatively, we can adopt the prediction methodology from the p-Partitioning presented in Section 4.1.1. We start the MS IPM as described in Algorithm 4.3 with an initial number J of subdomains. Let $M_J(\varepsilon_{\text{tol}})$ denote the number of basis functions that are necessary to reach the tolerance ε_{tol} for the given number J of subdomains. After each basis extension, before line 17 of Algorithm 4.3, we predict $M_J(\varepsilon_{\text{tol}})$ by extrapolating the maximal errors of the previous steps. We denote the prediction of $M_J(\varepsilon_{\text{tol}})$ by $M_J^{\text{pred}}(\varepsilon_{\text{tol}})$. If $M_J^{\text{pred}}(\varepsilon_{\text{tol}}) \leq M_{\max}$, we proceed the basis

extension. Otherwise, we increase the number of subdomains to some $J_{\text{new}} > J$ and restart the MS IPM.

For the efficiency of this refinement procedure, it is crucial to appropriately select the new number of subdomains. In the ideal case, a perfectly well separable family of functions $\mathcal{M}$, twice as many subdomains would lead to a halved number of necessary basis functions and the relation $J_0 M_{J_0}(\varepsilon_{\text{tol}}) = J_1 M_{J_1}(\varepsilon_{\text{tol}})$ would hold. Hence, the new number of subdomains should be determined by $J_{\text{new}} = J M_J^{\text{pred}}(\varepsilon_{\text{tol}})/M_{\max}$. However, this ideal case is very unrealistic and provides only a lower bound for the actually needed number of subdomains. Instead, we assume a nonlinear dependence and use

$$J_{\text{new}} = J \cdot \left(\frac{M_J^{\text{pred}}(\varepsilon_{\text{tol}})}{M_{\max}} \right)^{\alpha} \tag{4.5}$$

for some $\alpha > 1$. The exponent α depends on the separability of $\mathcal{M}$. We therefore start with a rather small α, e.g., $\alpha = 2$. If further refinement steps are necessary, α can be increased step by step. Alternatively, it would also be possible to determine an appropriate choice of α using the *hp* methodology. For some $\varepsilon_{\text{tol}}^h \gg \varepsilon_{\text{tol}}$ and several numbers of subdomains $J \in \{J_1, \ldots, J_n\}$, $n \in \mathbb{N}$, we determine $M_J(\varepsilon_{\text{tol}})$ or $M_J^{\text{pred}}(\varepsilon_{\text{tol}})$ and fit α such that

$$J_p \cdot \left(M_{J_p}(\varepsilon_{\text{tol}}) \right)^{\alpha} \approx J_q \cdot \left(M_{J_q}(\varepsilon_{\text{tol}}) \right)^{\alpha}, \quad p, q = 1, \ldots, n.$$

4.4 Fixed Shapes IPM

In the previous section, we developed a partitioning method for unknown parameter domains that is very flexible and automatically adapts the shapes to the given problem. The convergence in the different subdomains is well-balanced, the online assignments are adequate, and, for the majority of parameters, fast.

However, the refinement procedure can be relatively expensive since the bases on all subdomains are discarded and a complete restart is necessary. Furthermore, it is common in the EIM context to adaptively determine the number M of basis functions. Coefficients are added until the error estimator is precise enough. This can be done without an increased complexity. Since the use of more basis functions may yield a shift to a different subdomain, this adaptive selection of M is difficult in the context of moving shapes.

The objective of the Fixed Shapes (FS) IPM in this section is the development of an adaptive implicit partitioning method that fulfills the requirements of the Implicit Partitioning Problem 4.2 but allows a fast refinement procedure and an adaptive use of the number of basis functions M. Furthermore, the assignment of parameters are supposed to be based upon a tree based structure. Altogether, we could decrease offline and online complexity.

We develop two methods that fulfill different aspects of the above mentioned requirements. They have in common that the subdomains do not move with increasing M. We first present a procedure where the assignment is still based upon the approximation error. Then, in Section 4.4.2, an alternative is presented that is based upon the heuristic that has been already used in Algorithm 4.6, i.e., upon the first approximation coefficients.

4.4.1 Error Based FS IPM

For the following approach, we do not reassign the parameters to the subdomains in each iteration anymore. The assignment based upon the minimal approximation error of the first iteration is fixed for all further steps. Thus, the subdomains are independent of each other and also the generation of the EIM collateral bases can be performed independently. Furthermore, a subdomain can again be subdivided into several new subdomains with the same procedure and we can construct the partition in a tree-based scheme.

The detailed procedure of this FS IPM is described in Algorithm 4.7. It reveals strong similarities to the p-Partitioning of Algorithm 4.1 and the hp-Partitioning of Algorithm 4.2. We assume that the initial assignment for an arbitrary number J of subdomains and a training set $\mathcal{M}_{\text{train}}$ has been performed analogously to the initial step of the MS IPM, producing J disjoint sub-training sets $\mathcal{M}^j_{\text{train}} \subset \mathcal{M}_{\text{train}}$ and the initial EIM structs $\mathcal{S}^j_{\text{EIM},M_0}$, $j = 1, \dots, J$, $M_0 = 1$. We use the more general notation with an arbitrary initial M_0 in Algorithm 4.7 for later reference. For now, we constantly set $M_0 \equiv 1$. Algorithm 4.7 is started independently for each subdomain.

From line 1 to 7 of Algorithm 4.7, the already known EIM basis extension on subdomain j is performed, using always the same set of training samples. Once the maximal error falls below the tolerance ε_{tol}, the EIM struct $\mathcal{S}^j_{\text{EIM},M}$ is returned

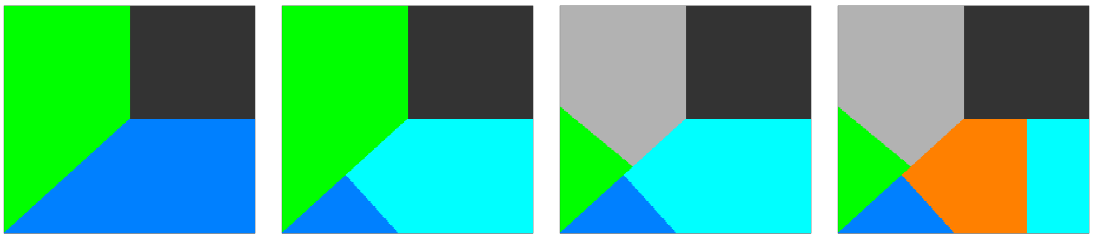
Algorithm 4.7 FixedShapesIPM($\mathcal{S}^j_{\mathrm{EIM},M_0}$, $\mathcal{M}^j_{\mathrm{train}}$, $M_{\max}$, $\varepsilon_{\mathrm{tol}}$, J)

```
1  for M = M_0 + 1 to M_max do
2      S^j_{EIM,M} = addBasisFunction(S^j_{EIM,M-1}, M^j_train)
3      ε^j_{M,max} = getMaxError(S^j_{EIM,M}, M^j_train)
4      if ε^j_{M,max} < ε_tol then
5          return S^j_{EIM,M}
6      end if
7  end for
8  {S^{J+1}_{EIM,1}, ..., S^{J+J_add}_{EIM,1}} = initialFirstBasisFunction(S^j_{EIM,J_add+1}, J_add)
9  {I^{J+1}, ..., I^{J+J_add}} = getOfflineAssignment(S^{J+1}_{EIM,1}, ..., S^{J+J_add}_{EIM,1}, M^j_train)
10 J_new = J + J_add
11 for i = 1 to J_add do
12     M^{J+i}_train = {c(μ) ∈ M_train | μ ∈ I^{J+i}}
13     FixedShapesIPM(S^{J+i}_{EIM,M_0}, M^{J+i}_train, M_max, ε_tol, J_new)
14 end for
```

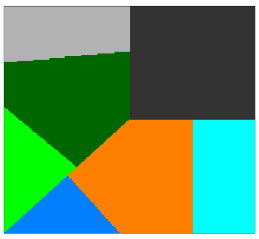
in line 5.

When $M_{\max}$ is reached without convergence, a refinement procedure has do be performed. Let J_{add} denote the number of new subdomains. As in the initial step of the MS IPM, we use the first $J_{\mathrm{add}} + 1$ selected basis functions of the current subdomain to initialize the new EIM structs in line 8. Again, we omit the first basis for a better distribution of the subdomains. In line 9, we assign the functions in $\mathcal{M}_{\mathrm{train}}$ to the appropriate subdomain based upon the approximation error as described in Algorithm 4.4. From line 11 to 14, we recursively start Algorithm 4.7 for each new subdomain.

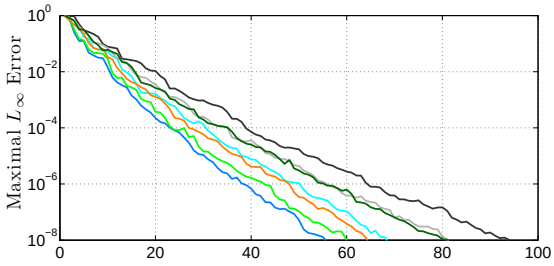
Since the shapes are fixed, we do not have to evaluate the approximation error for all input functions on all subdomains in each iteration. Hence, the offline runtime decreases. To further improve the offline complexity, it is now possible to adapt both the *hp* and the prediction methodology to the current algorithm. In other words, it is possible to create the subdomains based upon a comparatively large error tolerance $\varepsilon^h_{\mathrm{tol}} \gg \varepsilon_{\mathrm{tol}}$ and perform the basis extension in the second step. Furthermore, J_{add} can be chosen adaptively using the predicted number of needed basis functions M_1^{pred} on the current domain and Equation (4.5).



(a) initial partition (b) refinement 1 (c) refinement 2 (d) refinement 3



(e) refinement 4



(f) error decay in final subdomains

Figure 4.7: Initial partition for $J = 3$ (a). Refinement steps using the FS IPM of Algorithm 4.7 and $J_{\text{add}} = 2$ ((b) – (e)). Error decay in the final subdomains (f).

In Figure 4.7, the refinements for the example provided in (4.4) with a desired accuracy of $\varepsilon_{\text{tol}} = 10^{-8}$ and a maximal number of basis function $M_{\max} = 100$ is shown. We started with the initial partition already used for the MS IPM for $J = 3$ and used a constant number of $J_{\text{add}} = 2$. After four refinements, the partition is fine enough. Figure 4.7(f) shows the convergence in the final subdomains. It can be observed that the convergence rate is not balanced between the different subdomains anymore. For the desired accuracy, the subdomains need between 56 and 95 basis functions.

Online Procedure

In the online assignment, it is not straightforward to use the tree structure of the partition for efficient assignments. At a certain node of the tree of subdomains, we can evaluate the EIM error estimators for all child subdomains and proceed at the subdomain with the best result. However, for accurate results, it is not enough to only use small numbers M and M^+. In each node, M should be chosen such that the accuracy ε_{tol} is already reached. Otherwise, it is possible that we select

branches where the error can never fall below the desired tolerance. Hence, large values of $M > M_{\max}$ will be needed in the tree search. Thus, also offline run-time and the storage complexity increases.

Alternatively, we only store the leaf subdomains and apply the online assignment that has already been used for the MS IPM. It is also possible to use the heuristic of Algorithm 4.6 for a more efficient assignment. However, due to the unbalanced convergence rates, the subdomains that converge faster may cover parts that are too large. Hence, the adaptive basis size selection is not always possible in this case. In any way, compared to the MS IPM, the offline complexity has been significantly reduced.

4.4.2 Coefficient Based FS IPM

For both the MS IPM and the error based FS IPM, we tried to accelerate the assignment using the heuristic introduced in Algorithm 4.6 which is based upon the first coefficients of the affine approximations. In Figure 4.6(d), the result of this heuristic has been shown. It can be observed that the heuristic partition based upon the largest first coefficient almost exactly coincides with the initial true error based partition as show in Figure 4.4(a) and Figure 4.7(a). Hence, it seems to be a natural idea to use the heuristic not only in the online stage for a more efficient assignment but for the complete partitioning procedure. In other words, we could already base the splitting of a subdomain upon the largest coefficient.

However, it is not always possible to directly apply the heuristic. Consider the example problem used in [7] for the introduction of the EIM,

$$c(x;\mu) = (x_1 + \mu_1)^2 + (x_2 + \mu_2)^2, \tag{4.6}$$

$x \in \mathcal{D} = [0,1]^2$, $\mu \in \mathcal{P} = [0.01, 1]^2$. Independently of μ, the maximum of c is located at $x_{\max} = (1,1)$. Hence, each subdomains selects the same first EIM interpolation knot $t_1 = x_{\max} = \arg\max_{x\in\mathcal{D}} q_1^j(x)$, where q_1^j denotes the first basis function in subdomain j. Then, for the approximation of a function $c(\mu)$, $\mu \in \mathcal{P}$, the first coefficients in all subdomains are equal to $c(x_{\max}, \mu)$ due to the L_∞-normalization of the basis functions. Hence, we follow a slightly different approach and use a flexible and adaptive number of coefficients for the assignment.

Algorithm 4.8 getCoefficientBasedAssignment($\mathcal{S}^1_{\mathrm{EIM},M_0}, \ldots, \mathcal{S}^J_{\mathrm{EIM},M_0}$, $c(\mu)$, M_0)

1 **for** $j = 1$ **to** J **do**
2 $\quad \boldsymbol{\theta}^j_{M_0}(\mu) = \mathsf{getCoefficients}(\mathcal{S}^j_{\mathrm{EIM},M_0}, c(\mu), M_0)$
3 **end for**
4 **return** $i = \arg\max_j \{\|\boldsymbol{\theta}^j_{M_0}(\mu)\|_1, j = 1, \ldots, J\}$

Generally, for a given number M_0 of used coefficients and a given function $c(\mu)$, the coefficient based assignment procedure is given in Algorithm 4.8. We evaluate the vector $\boldsymbol{\theta}^j_{M_0}(\mu)$ of the first M_0 approximation coefficients for all subdomains j and return the subdomain where the sum of the respective coefficients in absolute values is maximal.

Since the assignment is independent of the number of degrees of freedom, the procedure $\mathsf{getCoefficientBasedAssignment}(\cdot)$ can be used in both offline and online stage. Furthermore, the assignment is independent of the actually used number of basis function. Thus, the subdomains are now completely fixed and online and offline shapes exactly coincide.

The main structure of the offline phase of the coefficient based FS IPM works analogously to the error based FS IPM in Algorithm 4.7, i.e., we build a tree of subdomains. A leaf subdomain will be refined if the given error tolerance $\varepsilon_{\mathrm{tol}}$ is not reached with a maximal number $M_{\max}$ of basis functions. The only change occurs in line 9 of Algorithm 4.7. Instead of the error based assignment procedure $\mathsf{getOfflineAssignment}(\cdot)$, we call

$$\{\mathcal{I}^{J+i}, \mathcal{S}^{J+i}_{\mathrm{EIM},M_0} \,|\, i = 1, \ldots, J_{\mathrm{add}}\} = \mathsf{refineCoefficientBased}(\mathcal{S}^{J+1}_{\mathrm{EIM},1}, \ldots, \mathcal{S}^{J+J_{\mathrm{add}}}_{\mathrm{EIM},1}, \mathcal{M}^j_{\mathrm{train}}, \mathcal{S}^j_{\mathrm{EIM},M_{\max}}, J_{\mathrm{add}}+1).$$

that is provided in Algorithm 4.9 and described in the following. It automatically detects the necessary number M_0 of used coefficients for an appropriate splitting of the domain. It returns not only the sets of assigned parameters $\mathcal{I}^{J+i}$ but also the to M_0 basis functions updated EIM structs $\mathcal{S}^{J+i}_{\mathrm{EIM},M_0}$, $i = 1, \ldots, J_{\mathrm{add}}$. Besides the initial EIM structs and the training functions, the input of the procedure also includes the EIM struct of the parent subdomain and the index of its last basis function that has been used as initial basis for a child subdomain.

Algorithm 4.9 starts trying to use a single coefficient for the assignment. From line 2 to 6, we perform the assignment based upon Algorithm 4.8. In line 7 of

Algorithm 4.9 refineCoefficientBased($\mathcal{S}^1_{\mathrm{EIM},1}, \dots, \mathcal{S}^J_{\mathrm{EIM},1}$, $\mathcal{M}_{\mathrm{train}}$, $\mathcal{S}^0_{\mathrm{EIM},M_{\max}}$, J_{init})

```
1  for M_0 = 1 to M_0^max do
2      I^1 = ... = I^J = {}
3      for n = 1 to n_train do
4          j = getCoefficientBasedAssignment(S^1_{EIM,M_0}, ..., S^J_{EIM,M_0}, c(μ_n), M_0)
5          I^j = I^j ∪ {μ_n}
6      end for
7      if I^1, ..., I^J ≠ ∅ then
8          return {I^j, S^j_{EIM,M_0} | j = 1, ..., J}
9      end if
10     {S^j_{EIM,M_0+1} | j = 1, ..., J} = doMSIPMStep(S^1_{EIM,M_0}, ..., S^J_{EIM,M_0}, M_train)
11 end for
12 for j = 1 to J do
13     if I^j = ∅ then
14         J_init = J_init + 1
15         S^j_{EIM,1} = newInitialBasisFunction(S^0_{EIM,M_max}, J_init)
16     end if
17 end for
18 refineCoefficientBased(S^1_{EIM,1}, ..., S^J_{EIM,1}, M_train, S^0_{EIM,M_max}, J_init)
```

Algorithm 4.8, we check if the assignment worked. More precisely, if none of the sets of parameters $\mathcal{I}^j$ is empty, we accept the new partition and return the result. For each subdomain, we also store the used number M_0 of coefficients for the assignment.

If at least one of the sets is empty, we discard the assignment. This happens for example if the coefficients of the different subdomains all have the same magnitude. Hence, we try to use more coefficients. We perform one step of the MS IPM in line 10 to determine one additional basis function for each subdomain. Then, the assignment procedure is iterated.

Suppose a maximal number $M_0^{\max}$ of allowed coefficients is reached without an appropriate assignment, we reset the subdomains with new initial bases from line 12 to 17. For each subdomain j with $\mathcal{I}^j = \emptyset$, we replace its initial basis by the next function of the parent subdomain that has not been used for any initial

basis in line 15. If $J_{\text{init}} > M_{\max}$, we can not assign a new initial basis and the algorithm would have to be stopped. Otherwise, we restart the whole procedure refineCoefficientBased$(\cdot)$ in line 18 with $M_0 = 1$. In other words, we discard all the selected basis functions except the first ones and use only the (partially new) initial EIM structs.

As mentioned before, we build a tree of subdomains such that we can perform a very efficient tree search in the online stage. Let $c(\mu)$ be a new input function. At each node, we evaluate the first M_0 approximation coefficients $\boldsymbol{\theta}^j_{M_0}$ of all child subdomains j which is of complexity $\mathcal{O}(J_{\text{add}} M_0^2)$. We move to the child subdomain where $\|\boldsymbol{\theta}^j_{M_0}\|_1$ is maximal until a leaf subdomain is reached. Hence, the assignment of a function to the right leaf subdomain can be achieved with complexity $\mathcal{O}(\log(J) \cdot M_0^2)$. Usually, M_0 is very small, even 1 for most examples and nodes, and the assignment is very fast.

At most nodes in the tree, only M_0 basis functions have to be stored. Only for leaf subdomains, the complete EIM structs are stored. For additional accelerations of the offline stage, the method can be combined with the *hp* methodology and/or the prediction techniques.

There is no guarantee that the partitioning converges by using more coefficients or by resetting the initial basis functions as suggested. However, in the examples in Section 4.6, we see that the method works well, even for the very unfavorable example introduced in (4.6). As for the error based FS IPM, we do not have the balanced convergence of the MS IPM. However, to avoid subdomains with only very few input functions, we could also replace the condition in line 7 and reject partitions where the distribution of the parameters is very unbalanced. E.g., a partition would be accepted only if all subdomains contain at least a postulated percentage of the parameters. In the examples below, we required that each subdomain obtains at least 5% of the parameters.

4.5 Combinations

It is possible to combine the different implicit partitioning methods in several ways. To save offline run time but still generate flexible shapes, it could be useful to first perform some steps of the error based FS IPM and generate a tree of subdomains.

At some step, we then switch to the MS IPM starting with the initial bases on the generated leaf subdomains. In the online stage, we would then use the assignment as given for the MS IPM.

In a similar way, it is possible to combine the coefficient based FS IPM and the MS IPM. Again, a small tree can be built resulting in a rough partition. Then, on each leaf subdomain j, we could generate a new partition using the MS IPM, respectively. To some extend, we would keep the flexible shapes and balanced convergence in this way. Furthermore, we would now save online and offline run time. In the offline stage, we do not need to evaluate all approximation errors on all input functions and all subdomains in the "tree phase". For the "MS IPM phase", the partitioning needs less refinement steps. In the online stage, we first perform an efficient tree search to find the appropriate leaf subdomain. Then, the more expensive online assignment based upon Algorithm 4.5 on the final partition is only used for a smaller number of subdomains.

If in any case, the coefficient based FS IPM does not terminate but produces inappropriate subdomains, it is therefore still possible to proceed with the MS IPM to still obtain the desired accuracy with less than $M_{\max}$ basis functions.

4.6 Numerical Examples and Comparisons

In this section, we consider three different examples to illustrate the different properties of the presented partitioning methods. For all examples, explicitly given parameter domains have been used. An additional example for stochastic input data can be found in Section 7.6.3. We compare the results of the implicit methods with the *hp* anchor point and *hp* gravity center methods and discuss advantages and disadvantages.

The desired L_∞ error tolerance in the construction of the partitions is given by $\varepsilon_{\text{tol}} = 10^{-8}$ for all examples. For the tree based refinement steps of the FS IPM, J_{add} has been set to 2 for all cases to facilitate the comparison to the other methods and to guarantee efficient tree structures. For the coefficient based FS IPM, we rejected partitions if one of the two subdomain obtained less than 5% of the parameters of the parent subdomain. The maximal number of coefficients used for the assignment has been set to $M_0^{\max} = 6$.

No error prediction techniques to accelerate the offline process have been used in order not to generate more subdomains than necessary and to obtain "optimal" partitioning results. Consequently, we also used $\varepsilon_{\text{tol}}^h = \varepsilon_{\text{tol}}^p$ and $M_{\max}^h = M_{\max}^p$ for the *hp*-Partitioning.

Example 1

We first consider the example adopted from [31] and already provided in (4.4). For the spatial domain $\mathcal{D} = [0,1]^2$ and the explicitly given parameter domain $\mathcal{P} = [0.3, 0.7]^2$, the input function $c : \mathcal{D} \times \mathcal{P} \to \mathbb{R}$ is given by

$$c(x;\mu) = e^{-50\left((x_1-\mu_1)^2+(x_2-\mu_2)^2\right)}.$$

For the discretization of the spatial domain, we used a uniform grid with edge length 0.02 such that the number of degrees of freedom is given by $\mathcal{N} = 2601$. The parameter samples for the offline stage are selected using a logarithmically distributed grid with 72 parameters in each direction and $n_{\text{train}} = 5184$ samples.

For all refinement steps of the coefficient based FS IPM, it was sufficient to use only $M_0 = 1$ coefficient for the assignment. Since the results of the coefficient based and the error based FS IPM were almost identical in numbers and shapes of subdomains, we omitted the error based results.

In Figure 4.8, we compare the efficiency of the implicit methods with the *hp* results. For given maximal number of basis functions $M_{\max}$, the respective numbers J of generated subdomains are displayed in a logarithmic scale. A single EIM on the complete parameter domain $\mathcal{D}$ needs $M = 199$ basis functions for the error tolerance $\varepsilon_{\text{tol}} = 10^{-8}$. We can see that the differences of the numbers of needed subdomains for the shown methods are very small. Only the *hp* gravity center method generated better results than the MS IPM. However, this method is far less flexible since only partitions with $J = 4, 16$ and 64 subdomains could be obtained. Hence, it would also generate more than necessary subdomains for most values $M_{\max}$. Furthermore, the determination of the gravity center is based upon the total number of $M_{\max}$ parameters and may be less appropriate for $M_{\max}^h \ll M_{\max}$ in real applications. The coefficient based FS IPM generated very similar results to the *hp* anchor point method with about the same offline and online complexity but without any knowledge of the parameters.

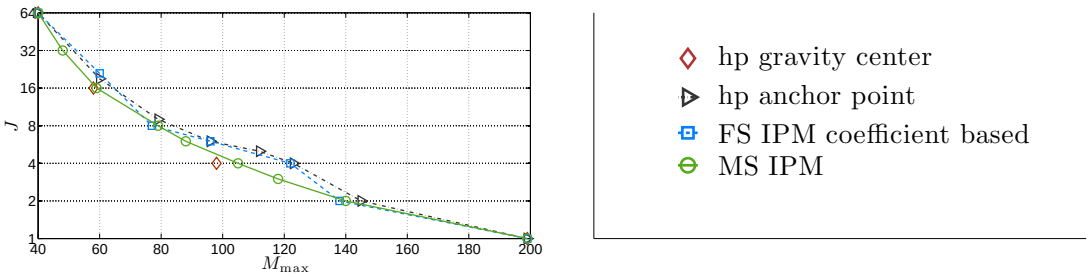
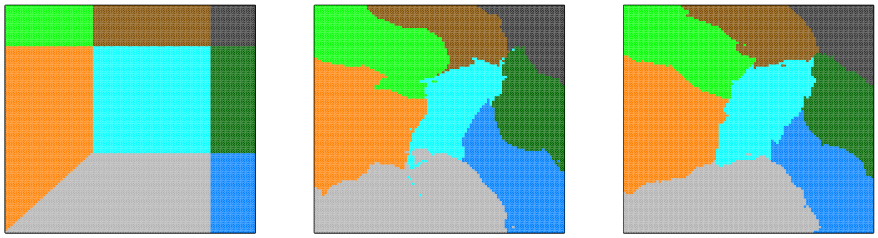
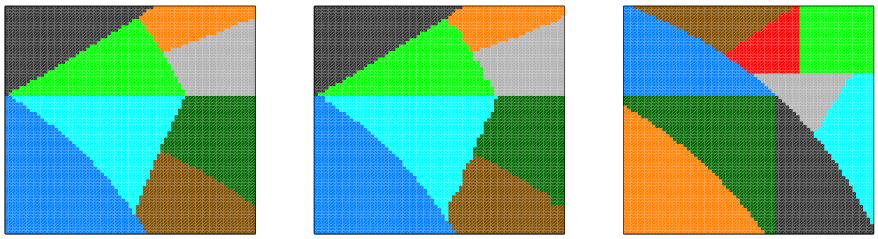


Figure 4.8: Comparison: number of subdomains J necessary for a given maximal number of affine terms $M_{\max}$ for Example 1.



(a) $M = 1$ (b) $M = 33$ (c) $M = 78$

Figure 4.9: Moving partitions for Example 1 using the MS IPM for $J = 8$, leading to $M = 33$ for $\varepsilon_{\text{tol}} = 10^{-4}$ and $M = 78$ for $\varepsilon_{\text{tol}} = 10^{-8}$.



(a) FS IPM coeff. based (b) FS IPM error based (c) hp anchor point

Figure 4.10: Partitioning results for Example 1 and $M_{\max} = 80$ using different tree-based methods.

In Figure 4.9, we see the partitioning result of the MS IPM for the given number of $J = 8$ subdomains and three different numbers M of basis functions. Comparing the initial partition for $M = 1$ with the partitions for $M = 33$ and $M = 78$, we see again that in later basis extension steps, only few reshaping occurs. For the error tolerance $\varepsilon_{\text{tol}} = 10^{-4}$, the different subdomains needed between $M = 31$ and $M = 33$ basis functions. Hence, the convergence rate is very well-balanced. The error tolerance $\varepsilon_{\text{tol}} = 10^{-8}$ has been reached between $M = 75$ and $M = 78$.

We tested the MS IPM online stage using a test sample set of 10,000 parameters and the fast assignment of Algorithm 4.6. The first assignment trial, i.e., the partition based upon the largest first coefficient, coincided in over 98% of the samples with the partitioning result of the MS IPM for $M = 1$. In other words, the first assignment trial partition looked almost like Figure 4.9(a).

The following list shows the necessary assignment trials until an appropriate subdomain has been selected for $M = 33$, $M^+ = 36$ and $\varepsilon_{\text{tol}} = 10^{-4}$. The first line provides the number of trials, the second line the number of samples in % that have been assigned to an appropriate partition in the corresponding step.

1.	2.	3.	4.	5.	6.	7.	8.	-
77.99%	13.33%	3.25%	4.30%	0.56%	0.05%	0.00%	0.00%	0.52%

We can see that the great majority has been assigned in the first two steps. Hence, the procedure is very efficient. On average, we needed less than 1.39 assignment trials per sample, even though for 0.52% of the samples, the error estimator was larger than ε_{tol} on all subdomains. Here, a larger number M would be needed. For a second example with a large number $J = 64$ of subdomains and $M = 40$, $M^+ = 44$, the average number of assignment trials was still less than 2.

Figure 4.10 compares the partitioning result of both tree structured implicit methods with the result *hp* anchor point method for a given $M_{\max} = 80$ and $\varepsilon_{\text{tol}} = 10^{-8}$. The coefficient based and the error based FS IPM generate almost identical partitions with $J = 8$ subdomains whereas the *hp* method needed $J = 9$ subdomains in this case.

Example 2

We now consider the example that has been used in [7] to introduce the idea of the EIM and which has briefly been mentioned in Section 4.4.2. For a spatial

domain $\mathcal{D} = [0,1]^2$ and the parameter domain $\mathcal{P} = [0.01,1]^2$, the input functions $c : \mathcal{D} \times \mathcal{P} \to \mathbb{R}$ are defined by

$$c(x;\mu) = (x_1 + \mu_1)^2 + (x_2 + \mu_2)^2.$$

As for the Example 1, we use a uniform grid on $\mathcal{D}$ with edge length 0.02 and $\mathcal{N} = 2601$ degrees of freedom. The parameter domain $\mathcal{P}$ is sampled using a logarithmically distributed grid with 72 parameters in each direction leading to $n_{\text{train}} = 5184$.

We already mentioned that the maximum of c is located at $x_{\max} = (1,1)$ independently of μ. Since the first interpolation knot is located at the maximum of the first basis function, all subdomains select the same first knot and therefore, the first approximation coefficient $\theta_1^j(\mu)$ is identical for all subdomains j. Hence, the coefficient based FS IPM needs at least two coefficients for the assignments. On average over all performed runs and nodes, it selected about 2.8 coefficients for the assignments. Especially in the lower parts of the constructed trees, i.e., for large numbers of subdomains, we had to reset the initial basis functions to obtain appropriate partitions.

In Figure 4.11, we compare again the numbers of generated subdomains for different values of $M_{\max}$, where the numbers of subdomains are plotted logarithmically. For this example, the MS IPM clearly outperforms the other methods. On average, the coefficient based FS IPM and the hp methods produced similar numbers of subdomains. For the hp gravity center method, only the very few numbers of $J = 4, 7, 16$ and 25 could be reached at all.

In Figure 4.12, we compare the partitions generated by the different methods for a desired number of $M_{\max} = 55$ basis functions. We observe that the shapes and numbers of subdomains differ significantly, where the MS IPM in Figure 4.12(a) seems to divide the parameter domain in the best way. It is also interesting to see that some of the subdomains of the coefficient based FS IPM in Figure 4.12(b) are divided into several parts that are not connected. E.g., the "black subdomain" consists of parameters in the lower left and lower right part of the parameter domain. For the construction of this partition, two resets of the initial partition were necessary and the average number of coefficients for the assignment was 2.75.

It turned out that it is also possible to use a constant number $M_0 > 1$ of coefficients for the assignment for this example. If a partition in a node is rejected,

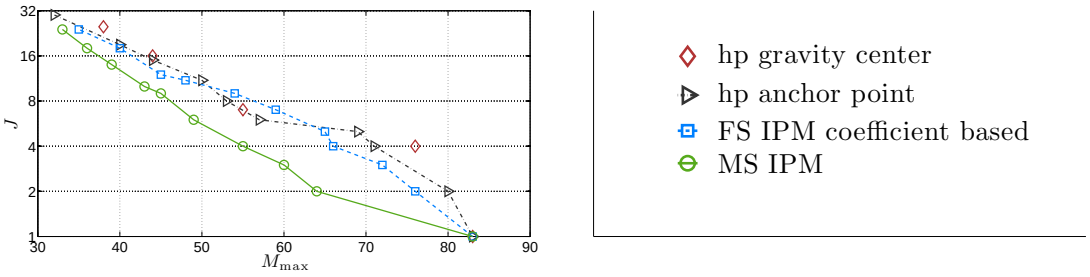
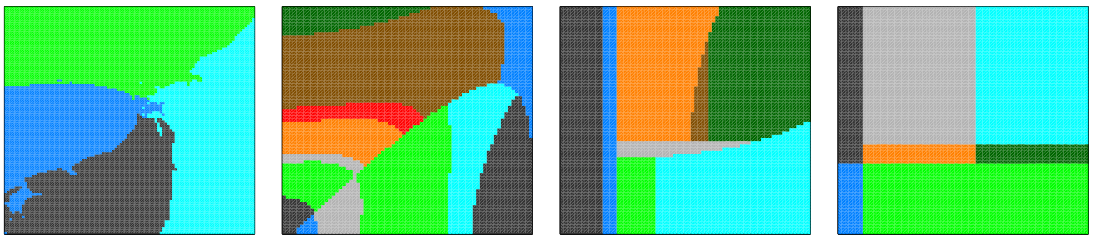
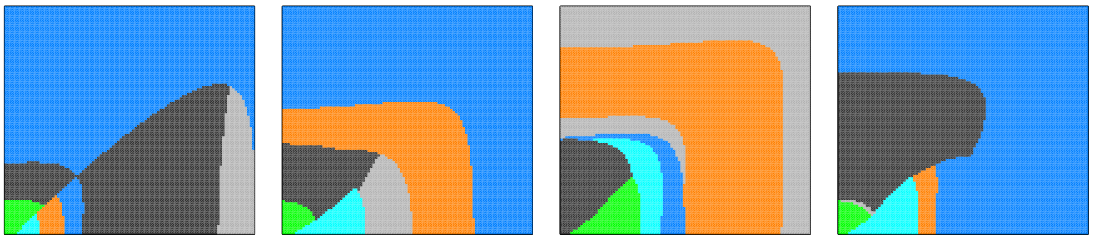


Figure 4.11: Comparison: number of subdomains J necessary for a given maximal number of affine terms $M_{\max}$ for Example 2.



(a) MS IPM final: $J = 4$ (b) FS IPM coefficient based: $J = 9$ (c) hp anchor point: $J = 8$ (d) hp gravity center: $J = 7$

Figure 4.12: Partitioning result for Ex. 2 and desired $M_{\max} = 55$ using different partitioning methods.



(a) based on $M_0 = 2$ coefficients (b) based on $M_0 = 3$ coefficients (c) based on $M_0 = 4$ coefficients (d) based on $M_0 = 5$ coefficients

Figure 4.13: Partitioning result for Ex. 2 and using the coefficient based FS IPM for constant M_0 and $J = 6$.

we directly substitute the initial basis as described in Algorithm 4.9. Figure 4.13 shows the partitioning result after five refinement steps, i.e., for $J = 6$ subdomains, for different numbers M_0. For this example, the initial bases have been reset only if no parameters have been assigned, i.e., if $\mathcal{I}^j = \emptyset$ for some subdomain j. It can be seen that the shapes of the subdomains strongly depend on the used number of coefficients. From the appearance of the shapes, we would guess that $M_0 = 3$ leads to the best results, whereas the shapes in Figure 4.12(b) show some similarities to the result of Figure 4.13(a) for $M_0 = 2$. Indeed, the FS IPM with constant $M_0 = 3$ produced better results for many cases. E.g., for $M_{\max} = 40$, it needed only $J = 12$ and for $M_{\max} = 60$ only $J = 5$ subdomains whereas the "regular" FS IPM needed $J = 18$ and $J = 7$ subdomains, respectively. However, it is not possible to know the proper number a priori. Hence, the procedure proposed in Algorithm 4.9 seems to be more appropriate, especially for unknown parameter domains.

Example 3

In the last example of this section, we consider a special parameter dependency. For the spatial domain $\mathcal{D} = [0,1]^2$ and the explicitly given parameter domain $\mathcal{P} = [0,1]^2$, the input function $c : \mathcal{D} \times \mathcal{P} \to \mathbb{R}$ is given by

$$c(x;\mu) = e^{-50\left(\left(x_1 - 4\left(\mu_1 - \frac{1}{2}\right)^2 - \mu_2^2\right)^2 + x_2^2\right)}.$$

Now, for parameters $\mu \in \mathcal{P}$ on the elliptic curves

$$4\left(\mu_1 - \tfrac{1}{2}\right)^2 + \mu_2^2 \equiv const,$$

the input functions $c(\mu)$ are identical. Hence, it is desirable that the partitioning methods detect this dependency and adjust the splitting of the subdomains accordingly.

For the discretization of the spatial domain, we used again a uniform grid with edge length 0.02 and obtain $\mathcal{N} = 2601$ degrees of freedom. The parameter samples for the offline stage are now selected using a uniform grid on $\mathcal{P} = [0,1]^2$ with 72 parameters in each direction. Hence, we obtain $n_{\text{train}} = 5184$ uniformly distributed samples.

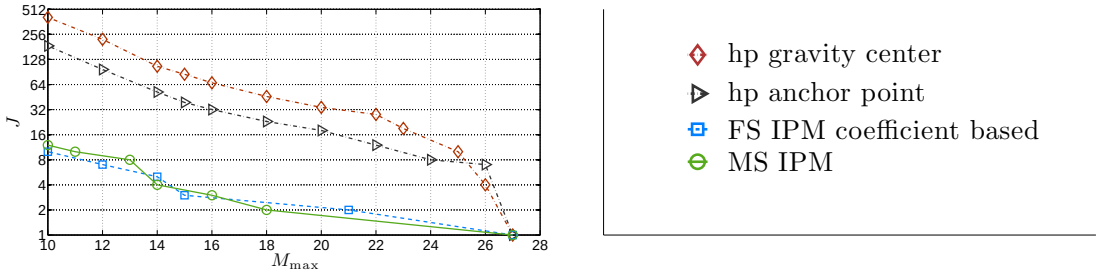
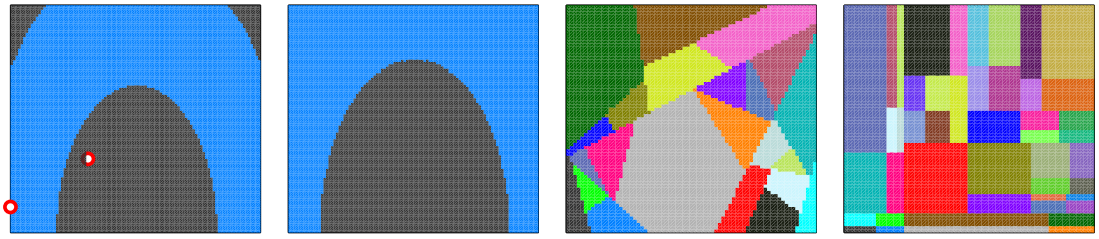
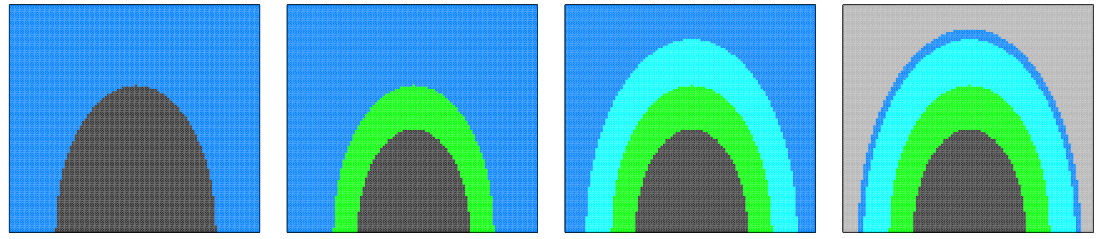


Figure 4.14: Comparison: number of subdomains J necessary for a given maximal number of affine terms $M_{\max}$ for Example 3.



(a) MS IPM initial: $J = 2$ (b) MS IPM final: $J = 2$ (c) hp anchor point: $J = 23$ (d) hp gravity center: $J = 46$

Figure 4.15: Partitioning result for Example 3 and desired $M_{\max} = 18$ using different partitioning methods.



(a) refinement 1 (b) refinement 2 (c) refinement 3 (d) refinement 4

Figure 4.16: Tree structured refinement steps for Example 3 using the coefficient based FS IPM.

A single EIM on the complete subdomain converged for $M = 27$ basis functions. For $M_{\max} = 18$, the partitioning results of the MS IPM and the two *hp* methods are provided in Figure 4.15. For the MS IPM, two subdomains are sufficient. We provide the initial partition for $M = 1$ and the final partition for $M = 18$ in Figures 4.15(a) and 4.15(b), respectively. In Figure 4.15(a), we also marked the two parameters that have been used for the initial basis. It can directly be seen that the subdomains are defined by the band around the ellipse of parameters on which the initial parameters are located.

Contrarily, the *hp* results in Figures 4.15(c) and 4.15(d) do not properly detect the geometric parameter dependency of the input functions. Not even the symmetry along the axis $\mu_1 = \frac{1}{2}$ has been used. As a consequence, a huge number of subdomains is needed. At the same time, many subdomains cover the same part of the family of input functions. Especially for the *hp* anchor point method, one can see that many small subdomains have been created along one parameter ellipse.

Also in Figure 4.14, it can be observed that the *hp* methods do not detect the more special parameter dependency. On average, the anchor point method needs about 10 times more subdomains to appropriately cover the complexity of $\mathcal{M}$. The number of subdomains created by the gravity center splitting procedure is an additional factor of around two larger.

The number and shapes of the subdomains created for the MS IPM and the FS IPM differ only slightly. In Figure 4.16, the tree structure of the coefficient based FS IPM is provided. In each step, one of the subdomains is divided into two parts. We see that shapes of the subdomains keep their elliptic appearance. For the parameter assignments, one approximation coefficient was sufficient.

4.7 Conclusions

We developed implicit partitioning methods that assign parametric input functions to an appropriate subdomain without the knowledge of the actual parameter or any other additional information. On each subdomain, an EIM is performed that creates affine approximations of the input functions with respect to the unknown parameter. The methods automatically detect complex parametric structures such as symmetries or other patterns of the parametric dependency. Hence, for wide

Table 4.1: Comparison of the different partitioning methods.

	MS	FS: $\mathcal{I}^j_{\text{first}}$	FS: $\boldsymbol{\theta}^j_{M_0}$	*hp*
implicit	+	+	+	−
offline complexity	−	+	+	+
online complexity	−	−	+	+
adaptive basis size	−	o	+	+
balanced convergence	+	−	−	−
flexible shapes	+	o	o	−
robustness	+	+	−	+

classes of problems, the implicit methods outperform other partitioning methods even for known explicitly given parameters.

In Table 4.1, a summary of the advantages and disadvantages of the methods is provided and the behavior is compared to the *hp*-Partitioning methods. In the first column, the characteristics of the MS IPM are illustrated. Next, the properties of the error based and the coefficient based FS IPM are shown. In the last row, we compare the implicit methods with the *hp* methods that need explicitly given parameter dependencies.

As mentioned before, a tree structure is desired offline and online for a fast construction of the partition and for fast assignments. The only implicit method that fulfills both requirements is the coefficient based FS IPM. The online assignment for the MS IPM and the error based IPM can be accelerated using the heuristic of Algorithm 4.6. Since the shapes are not absolutely fixed for the error based FS IPM, it can not be guaranteed that the adaptive choice of the number of used basis function is possible.

Only the MS IPM creates partitions where the convergence rate of the EIM in all subdomains is well-balanced. Furthermore, the shapes of the subdomains show the largest amount of flexibility and adaptively try to optimize the distribution of the parameters. The shapes generated by the error and coefficient based FS IPM are less adapted to the problem. Nevertheless, symmetries and other regular parametric patterns are detected and used for a more efficient partitioning as shown in Example 3 in Section 4.6.

The coefficient based FS IPM is less robust in the sense that it can not be guaranteed that the splitting into several subdomains works appropriately. It may be necessary to adjust for example the maximal number $M_0^{\max}$ of coefficients for the assignment or the minimal percentage of parameters that are required in each subdomain in the splitting phase. Hence, it is more difficult to use the method as a black box, even though it worked quite well for our examples.

Chapter 5

RBM for Linear Parametric PDEs with Stochastic Influences

This chapter is based upon joint work with B. Haasdonk and K. Urban and the main results have already been published in [45] in a very similar form. We added sections about higher moments, non-coercive problems, and showed that some assumptions regarding stochastic independence can be weakened such that more general classes of problems can be considered.

In this chapter, we introduce the RB methodology for parametrized partial differential equations (PPDEs) with stochastic influences. We consider problems that are already affine with respect to the deterministic parameter. Furthermore, strong solutions in probability are used such that the problem is solved in a Monte Carlo context. One might now think that the RB approach for deterministic problems can immediately be used in this context as well, viewing the stochasticity, i.e., stochastic events or inputs, as additional parameters. However, unlike for deterministic parameters, we have generally no distance measure in the probability space at our disposal, and so the ideas cannot be transferred directly. A basic assumption of the RBM is a smooth dependence of the solution of the PPDE with respect to the parameter, which cannot be assured due to the lack of the distance measure. Furthermore, the dimension of the parameter space crucially influences the efficiency of the RBM. In the case of stochastic influences, the parameter space may be infinite-dimensional.

As a way out, we propose using a Karhunen–Loève (KL) expansion (cf. Section 2.2) of the stochastic process and appropriately truncating it. Even though the resulting expansion coefficients are still random variables, i.e., functions with respect to the stochastic event, we treat them in some way as parameters that can be modeled using polynomial chaos (PC) expansions (cf. Section 2.3). The KL truncation error of course has to be be analyzed. The KL expansion shows some resemblance to the empirical interpolation method (cf. Section 3.2.3) in order to obtain an affine decomposition of random and spatial variables, where the random variables correspond to the parameter dependent EIM coefficients. Consequently, our analysis is in some parts similar to the EIM analysis in, e.g., [86].

Particularly in the presence of stochastic influences, one is interested not only in a good approximation of the state, i.e., the solution of the PPDE, but also in accurate outputs, together with corresponding statistical quantities such as expectation or variance. The latter requires the computation of quadratic output functionals. Different RBMs for quadratic outputs have been studied. These methods use expanded formulations that eliminate the nonlinearity [54] or introduce special dual problems [56]. Due to the KL truncation effects, however, these approaches cannot be used directly for our problem at hand. Hence, we introduce two more modified dual linear problems in order to derive a posteriori error bounds also for the above-mentioned statistical quantities. These error estimates can then be used in a standard Greedy approach [98] for the offline snapshot selection.

The remainder of the chapter is organized as follows. In Section 5.1, we collect known facts on variational problems with stochastic influences, the KL expansion, and the RBM. We restrict ourselves to linear coercive problems. Section 5.3 contains our a posteriori error analysis for the primal and dual solutions as well as linear and quadratic outputs. In Section 5.4 and 5.5, we introduce the error analysis for statistical quantities such as moments and variances. Note that since the operator has stochastic influences, we cannot derive a deterministic PDE for linear moments such as the expectation even for linear PDEs. In Section 5.6, the methodology and error analysis are expanded to non-coercive but inf-sup stable problems. The offline-online decomposition is presented in Section 5.7 as well as a method to compute coercivity lower bounds adjusted to stochastic problems. Our numerical experiments are described in Section 5.8.

5.1 Problem Formulation

In this section, we collect the basic features of the problem under consideration.

5.1.1 Variational Problems with Stochastic Influences

Let $D \subset \mathbb{R}^d$ be an open, bounded domain, $\mathcal{P} \subset \mathbb{R}^p$ a set of deterministic parameters, and $(\Omega, \mathfrak{A}, \mathbb{P})$ a probability space. For some $X \subset H^1(D)$, accounting also for the corresponding boundary conditions, let $a : X \times X \times \mathcal{P} \times \Omega \to \mathbb{R}$ be a possibly nonsymmetric form that is bilinear, continuous, and coercive with respect to the first two arguments, and let $f : X \times \mathcal{P} \times \Omega \to \mathbb{R}$ be a form with $f(\cdot; \mu, \omega) \in H^{-1}(D)$, $(\mu, \omega) \in \mathcal{P} \times \Omega$, that is stochastically independent of $a(\cdot, \cdot; \mu, \omega)$ such that the variational problem

$$a(u, v; \mu, \omega) = f(v; \mu, \omega), \qquad v \in X, \tag{5.1}$$

admits a unique solution $u(\mu, \omega) = u(\cdot; \mu, \omega) \in X$ for all $(\mu, \omega) \in \mathcal{P} \times \Omega$. As an example, think of a linear elliptic second order PDE whose coefficients and right-hand side depend on deterministic parameters $\mu \in \mathcal{P}$ and stochastic inputs $\omega \in \Omega$. In particular, we have in mind the case in which a coefficient function on D depends on stochastic influences modeled by ω. A formulation of the type (5.1) is also called D-weak/Ω-strong [12], and the difference from a variational approach with respect to both terms, e.g., stochastic Galerkin methods [69], should be noted. As already mentioned in the introduction, the direct view of ω — which represents an underlying stochastic event — as an additional parameter is *not* entirely possible. One should think of it merely as an uncertainty; i.e., $a(\cdot, \cdot; \cdot, \omega)$ is a random variable or a stochastic process. Nevertheless, we sometimes refer to ω as the stochastic parameter.

In order to achieve computational efficiency of an RBM for (5.1), we assume that both terms in (5.1) allow for an affine decomposition with respect to the deterministic parameter μ, namely,

$$a(w, v; \mu, \omega) = \sum_{q=1}^{Q^a} \theta_q^a(\mu) \big[\bar{a}_q(w, v) + a_q(w, v; \omega)\big], \tag{5.2}$$

$$f(v; \mu, \omega) = \sum_{q=1}^{Q^f} \theta_q^f(\mu) \big[\bar{f}_q(v) + f_q(v; \omega)\big], \tag{5.3}$$

with $Q^a,\ Q^f \geq 1$, $\theta_q^a, \theta_q^f : \mathcal{P} \to \mathbb{R}$, $\bar{a}_q$, $a_q(\cdot,\cdot;\omega) : X \times X \to \mathbb{R}$, and $\bar{f}_q$, $f_q(\cdot;\omega) : X \to \mathbb{R}$ bounded for all $\omega \in \Omega$. Note that $\bar{a}_q$ and $\bar{f}_q$ denote the expectations of the terms in brackets; $a_q(\cdot,\cdot;\omega)$ and $f_q(\cdot;\omega)$ denote the respective fluctuating parts. We assume that all parts a_q, f_q are stochastically independent. In general, we do not require any further assumption on these terms. However, in Section 5.7, some restrictions are introduced in order to use an alternative method for the computation of coercivity lower bounds. In cases in which a and f do not allow for a decomposition in the form of (5.2) and (5.3), respectively, a standard tool to derive affine approximations of nonaffine functions is the *empirical interpolation method* (EIM) [7]. A possible use of the EIM would require a technically more involved error analysis which is not discussed in this chapter; cf. Chapter 3, Chapter 7, and [86].

In order to describe the well-posedness of (5.1), one usually defines the coercivity and continuity constants, respectively, as

$$\alpha(\mu,\omega) := \inf_{v\in X} \frac{a(v,v;\mu,\omega)}{\|v\|_X^2}, \qquad \gamma(\mu,\omega) := \sup_{w\in X}\sup_{v\in X} \frac{a(w,v;\mu,\omega)}{\|w\|_X\|v\|_X}. \tag{5.4}$$

We assume that for some $0 < \alpha_0, \gamma_\infty < \infty$, we have

$$\alpha(\mu,\omega) \geq \alpha_0 > 0 \qquad \text{(uniform coercivity)}, \tag{5.5a}$$

$$\gamma(\mu,\omega) \leq \gamma_\infty < \infty \qquad \text{(uniform continuity)} \tag{5.5b}$$

for all $(\mu,\omega) \in \mathcal{P}\times\Omega$. Under these assumptions, the Lax–Milgram theorem guarantees the well-posedness of (5.1). Next, for $(\mu,\omega) \in \mathcal{P}\times\Omega$, we define parameter-dependent inner products and energy norms as

$$(w,v)_{\mu,\omega} := a(w,v;\mu,\omega), \qquad \|w\|_{\mu,\omega}^2 := (w,w)_{\mu,\omega}, \quad v,w \in X. \tag{5.6}$$

In many situations, one is not (or not only) interested in the state $u(\mu,\omega)$ or the error in the energy norm, but in some quantity of interest in terms of a linear continuous functional $\ell : X \times \mathcal{P} \times \Omega \to \mathbb{R}$. Again, we assume that ℓ is affine, i.e.,

$$\ell(v;\mu,\omega) = \sum_{q=1}^{Q^\ell} \theta_q^\ell(\mu)\big[\bar{\ell}_q(v) + \ell_q(v;\omega)\big] \tag{5.7}$$

with $Q^\ell \geq 1$, $\theta_q^\ell : \mathcal{P} \to \mathbb{R}$, and $\bar{\ell}_q$, $\ell_q(\cdot;\omega) : X \to \mathbb{R}$ bounded and linear for all $\omega \in \Omega$. It is assumed that all parts ℓ_q are stochastically independent as well as that ℓ is independent of a. If ℓ is deterministic, we set $\ell_q \equiv 0$. The output $s : \mathcal{P} \times \Omega \to \mathbb{R}$ is given as

$$s(\mu,\omega) := \ell(u(\mu,\omega);\mu,\omega). \tag{5.8}$$

If $\ell = f$, the output coincides with the right-hand side; this is called the *compliant case*. In the noncompliant case, it is fairly standard to consider a *dual problem* of finding $p^{(1)} = p^{(1)}(\mu,\omega)$ such that for given $(\mu,\omega) \in \mathcal{P} \times \Omega$ one has

$$a(v, p^{(1)};\mu,\omega) = -\ell(v;\mu,\omega), \qquad v \in X. \tag{5.9}$$

The superscript $^{(1)}$ in (5.9) is motivated by the fact that we will introduce further dual problems later on.

5.1.2 Karhunen–Loève Expansion

As already stated in the introduction of the chapter, we consider the well-known *Karhunen–Loève (KL) expansion* (cf. Section 2.2 and [60, 65]). Let us briefly recall the main facts. Let $\kappa : D \times \Omega \to \mathbb{R}$ be a spatial stochastic process with zero mean and existing covariance operator $\mathrm{Cov}_\kappa(x,y) := \mathbb{E}\big[\kappa(x;\cdot)\,\kappa(y;\cdot)\big]$, $x, y \in D$. Let $(\lambda_k, \kappa_k(x))$, $k = 1, \ldots, \infty$, be the eigenvalue/eigenfunction-pairs of the covariance operator; then the KL expansion reads

$$\kappa(x;\omega) = \sum_{k=1}^{\infty} \sqrt{\lambda_k}\,\xi_k(\omega)\,\kappa_k(x), \tag{5.10}$$

where $\xi_k : \Omega \to \mathbb{R}$ are uncorrelated random variables with zero mean and variance 1. The eigenvalues are ordered $\lambda_1 \geq \lambda_2 \geq \cdots \geq 0$, and for numerical purposes, we assume a fast decay. One of the main reasons we consider the KL expansion is now obvious since the above equation allows for a separation of the stochastic and the spatial terms. This is very similar to an affine expansion of a form with respect to a deterministic parameter as is common in RBMs. Here, we can use the deterministic and purely space-dependent terms for calculations in the offline phase so that the stochastic influences enter only through the coefficients in the KL expansion and are thus scalar quantities.

Since the KL expansion requires zero-mean random variables, the affine decompositions in (5.2), (5.3), and (5.7) are obtained by a separation into the deterministic expectations $\bar{a}_q$, $\bar{f}_q$, $\bar{\ell}_q$ and the zero-mean stochastic parts. We apply the KL expansion to the factors a_q, f_q, and ℓ_q. For $b \in \{a, f, \ell\}$, we get, using the appropriate arguments and our assumptions regarding stochastic independence,

$$b(\cdot;\mu,\omega) = \sum_{q=1}^{Q^b} \theta_q^b(\mu) \left[\bar{b}_q(\cdot) + \sum_{k=1}^{\infty} \xi_{q,k}^b(\omega)\, b_{q,k}(\cdot) \right], \tag{5.11}$$

where for notational convenience $b_{q,k}$ also contains $\sqrt{\lambda_{q,k}^b}$ from the spectral decomposition of the corresponding covariance operator.

For numerical purposes, one usually restricts the infinite sums to some finite numbers $K_q^b < \infty$ of terms. It is well known that the KL approximation is optimal in a certain sense [60, 65]. For $b \in \{a, f, \ell\}$ we obtain the truncated forms

$$b^K(\cdot;\mu,\omega) := \sum_{q=1}^{Q^b} \theta_q^b(\mu) \left[\bar{b}_q(\cdot) + \sum_{k=1}^{K_q^b} \xi_{q,k}^b(\omega)\, b_{q,k}(\cdot) \right]. \tag{5.12}$$

Here and in the following, an index or superscript K indicates that the expression is, or is derived from, a truncated form. We do not distinguish the dependencies on K_q^b, $q = 1, \ldots, Q^b$, $b \in \{a, f, \ell\}$. The truncated primal and dual problems read, for $(\mu, \omega) \in \mathcal{P} \times \Omega$,

$$a^K(u_K(\mu,\omega), v; \mu, \omega) = f^K(v; \mu, \omega), \qquad v \in X, \tag{5.13}$$

$$a^K(v, p_K^{(1)}(\mu,\omega); \mu, \omega) = -\ell^K(v; \mu, \omega), \qquad v \in X, \tag{5.14}$$

with solutions $u_K = u_K(\mu, \omega)$ and $p_K^{(1)} = p_K^{(1)}(\mu, \omega)$, respectively.

5.1.3 Output of Interest

Often, one is interested in the state $u(\mu, \omega)$ as well as in the output functional

$$s(\mu, \omega) := \ell(u(\mu, \omega); \mu).$$

Furthermore, we may be interested in the squared functional

$$s^2(\mu, \omega) := (\ell(u(\mu, \omega), \mu))^2.$$

Besides these random outputs, we want to evaluate some statistical quantities such as first and second moments of $s(\mu,\cdot)$, denoted by

$$\mathbb{M}_1(\mu) := \mathbb{E}[s(\mu,\cdot)] \quad \text{and} \quad \mathbb{M}_2(\mu) := \mathbb{E}[s^2(\mu,\cdot)],$$

respectively. Additionally, we need the squared first moment $\mathbb{M}_1^2(\mu) = (\mathbb{E}[s(\mu,\cdot)])^2$ to evaluate the variance $\mathbb{V}(\mu,\omega)$, given by

$$\mathbb{V}(\mu) = \mathbb{M}_2(\mu) - \mathbb{M}_1^2(\mu).$$

5.2 Reduced Basis Approximation

We consider an RB approximation with respect to our parameters $(\mu,\omega) \in \mathcal{P}\times\Omega$. To this end, we first consider the detailed approximation of the primal and dual problems, e.g., by a finite element discretization on a sufficiently fine grid. The corresponding spaces are usually again denoted by X, indicating that the detailed approximation and the exact solution are (numerically) indistinguishable. We assume that $\dim(X) = \mathcal{N}$, where $\mathcal{N}$ is assumed to be "large". Consequently, as is typical in the RBMs, the error analysis will address only the error of the reduced to the detailed solution.

The primal and dual RB spaces are then appropriate subspaces

$$X_N \subset X,\ \dim(X_N) = N \ll \mathcal{N}, \qquad \tilde{X}_N^{(1)} \subset X,\ \dim(\tilde{X}_N^{(1)}) = \tilde{N}^{(1)} \ll \mathcal{N}.$$

Here and in what follows, an index N indicates that the expression denotes or is based on reduced systems. We do not explicitly indicate the dependencies on the different dimensions of the reduced systems. Nevertheless, the dimensions of the reduced spaces X_N and $\tilde{X}_N^{(\cdot)}$ defined below may be different. We obtain a truncated primal-dual RB formulation. For $(\mu,\omega) \in \mathcal{P}\times\Omega$, determine $u_{N,K} = u_{N,K}(\mu,\omega) \in X_N$, $p_{N,K}^{(1)} = p_{N,K}^{(1)}(\mu,\omega) \in \tilde{X}_N^{(1)}$ such that

$$a^K(u_{N,K}, v; \mu,\omega) = f^K(v;\mu,\omega), \qquad v \in X_N, \tag{5.15}$$

$$a^K(v, p_{N,K}^{(1)}; \mu,\omega) = -\ell^K(v;\mu,\omega), \qquad v \in \tilde{X}_N^{(1)}. \tag{5.16}$$

We will comment later on the specific construction of X_N and $\tilde{X}_N^{(1)}$.

5.3 A posteriori Error Analysis

Now, we focus on the introduction of a posteriori error bounds for the primal and dual problems as well as for linear and quadratic output functionals. We will follow considerations partly similar to those in [86].

5.3.1 Notation

We start by fixing some notations for the subsequent analysis. In many cases, where it should be clear from the setting, we will omit the parameter (μ, ω) for notational convenience. Let

$$e_{\mathrm{RB}}(\mu,\omega) := u_K(\mu,\omega) - u_{N,K}(\mu,\omega), \tag{5.17a}$$
$$\tilde{e}^{(1)}_{\mathrm{RB}}(\mu,\omega) := p^{(1)}_K(\mu,\omega) - p^{(1)}_{N,K}(\mu,\omega) \tag{5.17b}$$

be the primal and dual RB errors, respectively, where again u_K and $p^{(1)}_K$ denote the solutions of (5.13) and (5.14), respectively. The corresponding residuals read

$$r_{\mathrm{RB}}(v;\mu,\omega) := f^K(v;\mu,\omega) - a^K(u_{N,K},v;\mu,\omega) = a^K(e_{\mathrm{RB}},v;\mu,\omega), \tag{5.18a}$$
$$\tilde{r}^{(1)}_{\mathrm{RB}}(v;\mu,\omega) := -\ell^K(v;\mu,\omega) - a^K(v,p^{(1)}_{N,K};\mu,\omega) = a^K(v,\tilde{e}^{(1)}_{\mathrm{RB}};\mu,\omega). \tag{5.18b}$$

Assuming the availability of a computable lower bound $0 < \alpha_{\mathrm{LB}}(\mu,\omega) \leq \alpha(\mu,\omega)$ of the coercivity constant, it is fairly standard to derive RB error bounds in terms of the following quantities:

$$\Delta_{\mathrm{RB}}(\mu,\omega) := \frac{1}{\alpha_{\mathrm{LB}}(\mu,\omega)} \sup_{v\in X} \frac{r_{\mathrm{RB}}(v;\mu,\omega)}{\|v\|_X}, \tag{5.19a}$$
$$\tilde{\Delta}^{(1)}_{\mathrm{RB}}(\mu,\omega) := \frac{1}{\alpha_{\mathrm{LB}}(\mu,\omega)} \sup_{v\in X} \frac{\tilde{r}^{(1)}_{\mathrm{RB}}(v;\mu,\omega)}{\|v\|_X}. \tag{5.19b}$$

Following the arguments of standard RB a posteriori error analysis [73], the terms Δ_{RB} and $\tilde{\Delta}^{(1)}_{\mathrm{RB}}$ account for the error caused by restricting X to X_N or $\tilde{X}^{(1)}_N$, i.e., the RB error, given the truncated KL forms in (5.13, 5.14).

Next, we investigate the KL truncation error. In view of the definition of a^K, f^K, and ℓ^K, we see that any truncation error depends on the random variable ω and thus on the particular realization. This dependence is somehow unsatisfactory since all derived bounds would depend on a realization of a random variable.

Thus, we propose replacing the random variables $\xi_{q,k}^b(\omega)$, $k > K_q^b$, $b \in \{a, f, \ell\}$, by some ω-independent quantity. If the probability density functions of the random variables have finite support or the problem that underlies the PDE restricts their variations, we can use rigorous upper bounds ξ_{UB}^b, i.e., $|\xi_{q,k}^b(\omega)| \leq \xi_{\mathrm{UB}}^b$, $b \in \{a, f, \ell\}$, for all $\omega \in \Omega$. In many cases, however, it is also appropriate to use quantiles instead. For some $0 < \rho < 1$, we define ξ_{UB}^b such that $|\xi_{q,k}^b(\omega)| \leq \xi_{\mathrm{UB}}^b$ holds with probability $1 - \rho$, where ρ should be sufficiently small to be negligible in the following analysis. Hence, we can define the error terms for the primal and dual problems as

$$\delta_{\mathrm{KL}}(v; \mu, \omega) := \sum_{q=1}^{Q^a} |\theta_q^a(\mu)| \sum_{k=K_q^a+1}^{\infty} \xi_{\mathrm{UB}}^a \, |a_{q,k}(u_{N,K}(\mu, \omega), v)|, \tag{5.20a}$$

$$\tilde{\delta}_{\mathrm{KL}}^{(1)}(v; \mu, \omega) := \sum_{q=1}^{Q^a} |\theta_q^a(\mu)| \sum_{k=K_q^a+1}^{\infty} \xi_{\mathrm{UB}}^a \, |a_{q,k}(v, p_{N,K}^{(1)}(\mu, \omega))|, \tag{5.20b}$$

as well as for the right-hand sides $b \in \{f, \ell\}$,

$$\delta_{\mathrm{KL}}^b(v; \mu) := \sum_{q=1}^{Q^b} |\theta_q^b(\mu)| \sum_{k=K_q^b+1}^{\infty} \xi_{\mathrm{UB}}^b \, |b_{q,k}(v)|. \tag{5.20c}$$

Note, that δ_{KL} and $\tilde{\delta}_{\mathrm{KL}}^{(1)}$ still depend on ω via the RB solutions $u_{N,K}$ and $p_{N,K}^{(1)}$. The right-hand side terms δ_{KL}^f and $\delta_{\mathrm{KL}}^\ell$ are deterministic and thus depend only on $\mu \in \mathcal{P}$. For numerical realizations, the terms in (5.20) are usually truncated at some $K_{\max}$, where $K_q^b < K_{\max} \ll \mathcal{N} < \infty$. In a fashion similar to that for the RB error, we set

$$\Delta_{\mathrm{KL}}(\mu, \omega) := \frac{1}{\alpha_{\mathrm{LB}}(\mu, \omega)} \sup_{v \in X} \frac{\delta_{\mathrm{KL}}(v; \mu, \omega)}{\|v\|_X}, \tag{5.21a}$$

$$\tilde{\Delta}_{\mathrm{KL}}^{(1)}(\mu, \omega) := \frac{1}{\alpha_{\mathrm{LB}}(\mu, \omega)} \sup_{v \in X} \frac{\tilde{\delta}_{\mathrm{KL}}^{(1)}(v; \mu, \omega)}{\|v\|_X}, \tag{5.21b}$$

as well as

$$\Delta_{\mathrm{KL}}^b(\mu, \omega) := \frac{1}{\alpha_{\mathrm{LB}}(\mu, \omega)} \sup_{v \in X} \frac{\delta_{\mathrm{KL}}^b(v; \mu)}{\|v\|_X}, \quad b \in \{f, \ell\}. \tag{5.21c}$$

Remark 5.1. Since the definition of δ_{KL} in (5.20) includes absolute values, it is not linear in v and we can not define $\|\delta_{\mathrm{KL}}(\cdot; \mu, \omega)\|_{X'}$. Therefore, we use the (for

linear forms equivalent) formulation $\sup_{v\in X} \delta_{\mathrm{KL}}(v;\mu,\omega)/\|v\|_X$ in (5.21). Still, it is possible to efficiently evaluate the estimates of the truncation errors. More details are provided in Section 5.7.2.

Remark 5.2. Certainly, the error bounds in (5.20) and (5.21) can be defined without replacing the random variables $\xi_{q,k}^b(\omega)$, $k > K_q^b$, $b \in \{a, f, \ell\}$, by some upper bound or quantile. This might be reasonable in some applications, especially if one is interested only in statistical outputs such as mean or variance. Note that in this case, the absolute values in the definitions would be omitted and the δ_{KL}-forms remain linear. Then, the KL truncation error bounds would obviously be much smaller since the upper bounds ξ_{UB}^b, $b \in \{a, f, \ell\}$, already represent the worst case scenario of the truncation.

5.3.2 Primal and Dual Errors

We start by estimating primal and dual errors involving both KL and RB truncation, i.e.,

$$e(\mu,\omega) := u(\mu,\omega) - u_{N,K}(\mu,\omega), \tag{5.22a}$$

$$\tilde{e}^{(1)}(\mu,\omega) := p^{(1)}(\mu,\omega) - p_{N,K}^{(1)}(\mu,\omega), \tag{5.22b}$$

where u and $p^{(1)}$ denote the detailed primal and dual solutions of (5.1) and (5.9), respectively. For better readability and notational compactness, we omit the parameters μ and ω in the following whenever it does not affect the meaning.

Proposition 5.3. *Setting*

$$\Delta(\mu,\omega) := \Delta_{\mathrm{RB}}(\mu,\omega) + \Delta_{\mathrm{KL}}(\mu,\omega) + \Delta_{\mathrm{KL}}^f(\mu,\omega),$$

we get $\|e(\mu,\omega)\|_X \le \Delta(\mu,\omega)$ *for all* $(\mu,\omega) \in \mathcal{P}\times\Omega$.

Proof. We have for any $v \in X$ that

$$\begin{aligned} a(e,v) &= a(u,v) - a(u_{N,K},v) \\ &= \big(f(v) - f^K(v)\big) + \big(a^K(u_{N,K},v) - a(u_{N,K},v)\big) + \big(f^K(v) - a^K(u_{N,K},v)\big). \end{aligned}$$

The last term coincides with the residual $r_{\mathrm{RB}}(v) = a^K(e_{\mathrm{RB}}, v)$ in (5.18). Testing with $v = e$ and using the coercivity of a yields

$$\begin{aligned}\|e\|_X &\leq {\alpha_{\mathrm{LB}}}^{-1} \frac{a(e,e)}{\|e\|_X} \\ &\leq \frac{|f(e) - f^K(e)|}{\alpha_{\mathrm{LB}} \|e\|_X} + \frac{|a^K(u_{N,K}, e) - a(u_{N,K}, e)|}{\alpha_{\mathrm{LB}} \|e\|_X} + \frac{|r_{\mathrm{RB}}(e)|}{\alpha_{\mathrm{LB}} \|e\|_X} \\ &\leq \Delta^f_{\mathrm{KL}} + \Delta_{\mathrm{KL}} + \Delta_{\mathrm{RB}}\end{aligned}$$

by standard RB estimates, using the definitions of the bounds in (5.19, 5.21). □

Corollary 5.4. *Setting*

$$\tilde{\Delta}^{(1)}(\mu,\omega) = \tilde{\Delta}^{(1)} := \tilde{\Delta}^{(1)}_{\mathrm{RB}} + \tilde{\Delta}^{(1)}_{\mathrm{KL}} + \Delta^{\ell}_{\mathrm{KL}}$$

yields the estimate $\|\tilde{e}^{(1)}(\mu,\omega)\|_X \leq \tilde{\Delta}^{(1)}(\mu,\omega)$ *for all* $(\mu,\omega) \in \mathcal{P} \times \Omega$.

Proof. In a way similar to the above we get for any $v \in X$ that

$$\begin{aligned}a(v, \tilde{e}^{(1)}) &= a(v, p^{(1)}) - a(v, p^{(1)}_{N,K}) \\ &= \big(\ell^K(v) - \ell(v)\big) + \big(a^K(v, p^{(1)}_{N,K}) - a(v, p^{(1)}_{N,K})\big) - \big(\ell^K(v) + a^K(v, p^{(1)}_{N,K})\big),\end{aligned}$$

and using $v = \tilde{e}^{(1)}$ yields the desired estimate. □

The next step is to investigate the effectivity of the above estimators. To this end, we define the Riesz representations of primal and dual residuals as

$$\big(\mathcal{E}_{\mathrm{RB}}(\mu,\omega), v\big)_X = r_{\mathrm{RB}}(v;\mu,\omega), \quad v \in X, \tag{5.23a}$$

$$\big(\tilde{\mathcal{E}}^{(1)}_{\mathrm{RB}}(\mu,\omega), v\big)_X = \tilde{r}^{(1)}_{\mathrm{RB}}(v;\mu,\omega), \quad v \in X, \tag{5.23b}$$

for $\mu \in \mathcal{P}$ and $\omega \in \Omega$. Since $\mathcal{E}_{\mathrm{RB}}$ is the Riesz representative of r_{RB}, we have that $\|\mathcal{E}_{\mathrm{RB}}(\mu,\omega)\|_X = \|r_{\mathrm{RB}}(\mu,\omega)\|_{X'}$, and thus by definition

$$\begin{aligned}\|\mathcal{E}_{\mathrm{RB}}(\mu,\omega)\|_X &= \alpha_{\mathrm{LB}}(\mu,\omega)\, \Delta_{\mathrm{RB}}(\mu,\omega), \\ \|\tilde{\mathcal{E}}^{(1)}_{\mathrm{RB}}(\mu,\omega)\|_X &= \alpha_{\mathrm{LB}}(\mu,\omega)\, \tilde{\Delta}^{(1)}_{\mathrm{RB}}(\mu,\omega).\end{aligned}$$

Analogously, we define the Riesz representations of the KL residuals by

$$\big(\mathcal{E}_{\mathrm{KL}}(\mu,\omega), v\big)_X = r(v;\mu,\omega) - r_{\mathrm{RB}}(v;\mu,\omega), \tag{5.24a}$$

$$\big(\tilde{\mathcal{E}}^{(1)}_{\mathrm{KL}}(\mu,\omega), v\big)_X = \tilde{r}(v;\mu,\omega) - \tilde{r}^{(1)}_{\mathrm{RB}}(v;\mu,\omega), \tag{5.24b}$$

where the detailed residuals are defined as

$$\begin{aligned} r(v;\mu,\omega) &:= f(v;\mu,\omega) - a(u_{N,K}, v;\mu,\omega), \\ \tilde{r}(v;\mu,\omega) &:= -\ell(v;\mu,\omega) - a(v, p^{(1)}_{N,K};\mu,\omega). \end{aligned}$$

We obtain that

$$\begin{aligned} \|\mathcal{E}_{\mathrm{KL}}\|_X &= \|r - r_{\mathrm{RB}}\|_{X'} = \|f - a(u_{N,K},\cdot) - f^K + a^K(u_{N,K},\cdot)\|_{X'} \\ &\leq \|f - f^K\|_{X'} + \|a(u_{N,K},\cdot) - a^K(u_{N,K},\cdot)\|_{X'} \\ &\leq \alpha_{\mathrm{LB}}(\mu,\omega)(\Delta^f_{\mathrm{KL}} + \Delta_{\mathrm{KL}}), \end{aligned} \tag{5.25}$$

and similarly $\|\tilde{\mathcal{E}}^{(1)}_{\mathrm{KL}}\|_X \leq \alpha_{\mathrm{LB}}(\Delta^{\ell}_{\mathrm{KL}} + \tilde{\Delta}^{(1)}_{\mathrm{KL}})$. Finally, in order to estimate the effectivities

$$\eta(\mu,\omega) := \frac{\Delta(\mu,\omega)}{\|e(\mu,\omega)\|_X}, \qquad \tilde{\eta}^{(1)}(\mu,\omega) := \frac{\tilde{\Delta}^{(1)}(\mu,\omega)}{\|\tilde{e}^{(1)}(\mu,\omega)\|_X}, \tag{5.26}$$

we define the following quantities:

$$c(\mu,\omega) := \frac{\Delta_{\mathrm{KL}}(\mu,\omega) + \Delta^f_{\mathrm{KL}}(\mu,\omega)}{\Delta_{\mathrm{RB}}(\mu,\omega)}, \tag{5.27a}$$

$$\tilde{c}^{(1)}(\mu,\omega) := \frac{\tilde{\Delta}^{(1)}_{\mathrm{KL}}(\mu,\omega) + \Delta^{\ell}_{\mathrm{KL}}(\mu,\omega)}{\tilde{\Delta}^{(1)}_{\mathrm{RB}}(\mu,\omega)}. \tag{5.27b}$$

Proposition 5.5. *If $c(\mu,\omega) \in [0,1)$, we get*

$$\eta(\mu,\omega) \leq \frac{\gamma_{\mathrm{UB}}(\mu,\omega)}{\alpha_{\mathrm{LB}}(\mu,\omega)} \frac{1 + c(\mu,\omega)}{1 - c(\mu,\omega)},$$

where $\gamma_{\mathrm{UB}}(\mu,\omega) \geq \gamma(\mu,\omega)$ is an upper continuity bound.

Proof. It is straightforward to see that for $v \in X$ we have

$$\begin{aligned} a(e,v) &= r(v;\mu,\omega) \\ &= (r(v;\mu,\omega) - r_{\mathrm{RB}}(v;\mu,\omega)) + r_{\mathrm{RB}}(v;\mu,\omega) \\ &= (\mathcal{E}_{\mathrm{KL}}(\mu,\omega), v)_X + (\mathcal{E}_{\mathrm{RB}}(\mu,\omega), v)_X \\ &= (\mathcal{E}_{\mathrm{KL}}(\mu,\omega) + \mathcal{E}_{\mathrm{RB}}(\mu,\omega), v)_X. \end{aligned}$$

Thus, with $v = \mathcal{E}_{\mathrm{RB}} - \mathcal{E}_{\mathrm{KL}}$, we get

$$\begin{aligned} a(e, \mathcal{E}_{\mathrm{RB}} - \mathcal{E}_{\mathrm{KL}}) &= (\mathcal{E}_{\mathrm{KL}} + \mathcal{E}_{\mathrm{RB}}, \mathcal{E}_{\mathrm{RB}} - \mathcal{E}_{\mathrm{KL}})_X \\ &= \|\mathcal{E}_{\mathrm{RB}}\|^2_X - \|\mathcal{E}_{\mathrm{KL}}\|^2_X, \end{aligned}$$

and hence

$$\begin{aligned}\|\mathcal{E}_{\mathrm{RB}}\|_X^2 - \|\mathcal{E}_{\mathrm{KL}}\|_X^2 \;&=\; a(e, \mathcal{E}_{\mathrm{RB}} - \mathcal{E}_{\mathrm{KL}}) \;\leq\; \gamma_{\mathrm{UB}}\, \|e\|_X \,(\|\mathcal{E}_{\mathrm{RB}}\|_X + \|\mathcal{E}_{\mathrm{KL}}\|_X) \\ &=\; \gamma_{\mathrm{UB}}\, \|e\|_X \frac{\|\mathcal{E}_{\mathrm{RB}}\|_X^2 - \|\mathcal{E}_{\mathrm{KL}}\|_X^2}{\|\mathcal{E}_{\mathrm{RB}}\|_X - \|\mathcal{E}_{\mathrm{KL}}\|_X}.\end{aligned}$$

Therefore, by the above estimates,

$$\|e\|_X \;\geq\; \frac{1}{\gamma_{\mathrm{UB}}}(\|\mathcal{E}_{\mathrm{RB}}\|_X - \|\mathcal{E}_{\mathrm{KL}}\|_X) \;\geq\; \frac{\alpha_{\mathrm{LB}}}{\gamma_{\mathrm{UB}}}(\Delta_{\mathrm{RB}} - \Delta_{\mathrm{KL}} - \Delta_{\mathrm{KL}}^f).$$

This finally implies that

$$\eta \;=\; \frac{\Delta}{\|e\|_X} \;\leq\; \frac{\gamma_{\mathrm{UB}}}{\alpha_{\mathrm{LB}}} \frac{\Delta_{\mathrm{RB}} + \Delta_{\mathrm{KL}} + \Delta_{\mathrm{KL}}^f}{\Delta_{\mathrm{RB}} - \Delta_{\mathrm{KL}} - \Delta_{\mathrm{KL}}^f} \;=\; \frac{\gamma_{\mathrm{UB}}}{\alpha_{\mathrm{LB}}} \frac{1+c}{1-c},$$

which proves the claim. □

Completely analogously we can estimate the dual effectivity as follows.

Corollary 5.6. *If $\tilde{c}^{(1)}(\mu,\omega) \in [0,1)$, we get*

$$\tilde{\eta}^{(1)}(\mu,\omega) \;\leq\; \frac{\gamma_{\mathrm{UB}}(\mu,\omega)}{\alpha_{\mathrm{LB}}(\mu,\omega)} \frac{1+\tilde{c}^{(1)}(\mu,\omega)}{1-\tilde{c}^{(1)}(\mu,\omega)}.$$

□

Finally, for later reference, we note another result. Defining

$$\eta_0(\mu,\omega) \;:=\; \sqrt{\frac{\gamma_{\mathrm{UB}}(\mu,\omega)}{\alpha_{\mathrm{LB}}(\mu,\omega)}}\left(\frac{1+c(\mu,\omega)}{1-c(\mu,\omega)}\right), \tag{5.28}$$

we get the following estimate for the effectivity with respect to the energy norm.

Corollary 5.7. *If $c(\mu,\omega) \in [0,1)$, we get*

$$\frac{\sqrt{\alpha_{\mathrm{LB}}(\mu,\omega)}\Delta(\mu,\omega)}{\|e(\mu,\omega)\|_{\mu,\omega}} \;\leq\; \eta_0(\mu,\omega).$$

Proof. In the proof of Proposition 5.5, we replace $\|e\|_X$ by $\|e\|_{\mu,\omega}{\gamma_{\mathrm{UB}}}^{-1/2}$. □

5.3.3 Output Error

Now we consider the approximation $\ell^K(u_{N,K};\mu,\omega)$ to the output $\ell(u;\mu,\omega) = s(\mu,\omega)$. As already known from the RB a posteriori error analysis of linear output functionals [73], we add a correction term and consider

$$s_{N,K}(\mu,\omega) \;:=\; \ell^K(u_{N,K};\mu,\omega) - r_{\mathrm{RB}}(p^{(1)}_{N,K};\mu,\omega) \tag{5.29}$$

and define the output error estimator by

$$\Delta^s(\mu,\omega) \;:=\; \alpha_{\mathrm{LB}}\Delta\tilde{\Delta}^{(1)} + \delta_{\mathrm{KL}}(p^{(1)}_{N,K}) + \delta^f_{\mathrm{KL}}(p^{(1)}_{N,K}) + \delta^\ell_{\mathrm{KL}}(u_{N,K}). \tag{5.30}$$

Then, we obtain the following estimate.

Theorem 5.8. *$|s(\mu,\omega) - s_{N,K}(\mu,\omega)| \le \Delta^s(\mu,\omega)$ holds for all $\mu\in\mathcal{P}$ and $\omega\in\Omega$.*

Proof. By standard arguments, we get (omitting the argument (μ,ω))

$$\begin{aligned} s - s_{N,K} &= \ell(u) - \ell^K(u_{N,K}) + r_{\mathrm{RB}}(p^{(1)}_{N,K}) \\ &= \ell(u) - \ell^K(u_{N,K}) + f^K(p^{(1)}_{N,K}) - a^K(u_{N,K},p^{(1)}_{N,K}) \\ &= [\ell^K(u) - \ell^K(u_{N,K})] + [f(p^{(1)}_{N,K}) - a^K(u_{N,K},p^{(1)}_{N,K})] \\ &\quad + [\ell(u) - \ell^K(u)] - [f(p^{(1)}_{N,K}) - f^K(p^{(1)}_{N,K})]. \end{aligned}$$

For the first term on the right-hand side, we have

$$\ell^K(u) - \ell^K(u_{N,K}) = -a^K(u,p^{(1)}_K) + a^K(u_{N,K},p^{(1)}_K) = -a^K(e,p^{(1)}_K).$$

Using $f(p^{(1)}_{N,K}) = a(u,p^{(1)}_{N,K})$, we get for the first two terms

$$\begin{aligned} &[\ell^K(u) - \ell^K(u_{N,K})] + [f(p^{(1)}_{N,K}) - a^K(u_{N,K},p^{(1)}_{N,K})] \\ &= -a^K(e,p^{(1)}_K) + a(u,p^{(1)}_{N,K}) - a^K(u_{N,K},p^{(1)}_{N,K}) \\ &= -a^K(e,p^{(1)}_K) + a^K(u-u_{N,K},p^{(1)}_{N,K}) + [a(u,p^{(1)}_{N,K}) - a^K(u,p^{(1)}_{N,K})] \\ &= -a^K(e,p^{(1)}_K - p^{(1)}_{N,K}) + [a(u,p^{(1)}_{N,K}) - a^K(u,p^{(1)}_{N,K})] \\ &= -a^K(e,\tilde{e}^{(1)}_{\mathrm{RB}}) + [a(u,p^{(1)}_{N,K}) - a^K(u,p^{(1)}_{N,K})] \\ &= -\tilde{r}^{(1)}_{\mathrm{RB}}(e) + [a(u,p^{(1)}_{N,K}) - a^K(u,p^{(1)}_{N,K})]. \end{aligned}$$

Using $\ell(u) - \ell^K(u) = \ell(e+u_{N,K}) - \ell^K(e+u_{N,K})$ and $a(u,p^{(1)}_{N,K}) - a^K(u,p^{(1)}_{N,K}) = a(e+u_{N,K},p^{(1)}_{N,K}) - a^K(e+u_{N,K},p^{(1)}_{N,K})$ and putting all this together yields

$$\begin{aligned} s - s_{N,K} = &- \tilde{r}^{(1)}_{\mathrm{RB}}(e) \;+\; [a(e,p^{(1)}_{N,K}) - a^K(e,p^{(1)}_{N,K})] \;+\; [\ell(e) - \ell^K(e)] \\ &+ [\ell(u_{N,K}) - \ell^K(u_{N,K})] \;-\; [f(p^{(1)}_{N,K}) - f^K(p^{(1)}_{N,K})] \\ &+ [a(u_{N,K},p^{(1)}_{N,K}) - a^K(u_{N,K},p^{(1)}_{N,K})]. \end{aligned} \tag{5.31}$$

Using the triangle inequality, we estimate the first three terms separately, i.e.,

$$\begin{aligned} |\tilde{r}^{(1)}_{\mathrm{RB}}(e;\mu,\omega)| &\le \|e\|_X \sup_{v\in X}(\tilde{r}^{(1)}_{\mathrm{RB}}(v)/\|v\|_X) \le \alpha_{\mathrm{LB}}\,\Delta\,\tilde{\Delta}^{(1)}_{\mathrm{RB}}, \\ |a(e,p^{(1)}_{N,K}) - a^K(e,p^{(1)}_{N,K})| &\le \|e\|_X \sup_{v\in X}(\tilde{\delta}^{(1)}_{\mathrm{KL}}(v)/\|v\|_X) \le \alpha_{\mathrm{LB}}\,\Delta\,\tilde{\Delta}^{(1)}_{\mathrm{KL}}, \\ |\ell(e) - \ell^K(e)| &\le \|e\|_X \sup_{v\in X}(\delta^{\ell}_{\mathrm{KL}}(v)/\|v\|_X) \le \alpha_{\mathrm{LB}}\,\Delta\,\Delta^{\ell}_{\mathrm{KL}}, \end{aligned}$$

by Proposition 5.3. Furthermore, $|\ell(u_{N,K}) - \ell^K(u_{N,K})| \le \delta^{\ell}_{\mathrm{KL}}(u_{N,K})$, $|f(p^{(1)}_{N,K}) - f^K(p^{(1)}_{N,K})| \le \delta^{f}_{\mathrm{KL}}(p^{(1)}_{N,K})$ and $|a(u_{N,K},p^{(1)}_{N,K}) - a^K(u_{N,K},p^{(1)}_{N,K})| \le \delta_{\mathrm{KL}}(p^{(1)}_{N,K})$. We put everything together, which yields the desired result. □

The above analysis shows two effects. First, the RB and KL error terms $\Delta_{\mathrm{RB}}, \Delta_{\mathrm{KL}}, \Delta^{f}_{\mathrm{KL}}$ and $\tilde{\Delta}^{(1)}_{\mathrm{RB}}, \tilde{\Delta}^{(1)}_{\mathrm{KL}}, \Delta^{\ell}_{\mathrm{KL}}$ appear in pairwise products in the first term of (5.30). In order to obtain the full order of approximation, RB and KL error terms should thus be of comparable sizes. Second, as opposed to the deterministic case, we obtain the additional additive terms $\delta_{\mathrm{KL}}(p^{(1)}_{N,K})$, $\delta^{f}_{\mathrm{KL}}(p^{(1)}_{N,K})$, and $\delta^{\ell}_{\mathrm{KL}}(u_{N,K})$ as we see from the estimates of $|a(u,p^{(1)}_{N,K}) - a^K(u,p^{(1)}_{N,K})|$, $|f(p^{(1)}_{N,K}) - f^K(p^{(1)}_{N,K})|$, and $|\ell(u) - \ell^K(u)|$.

Finally, we investigate the effectivity of the output error bound for the special case of a compliant output, i.e., $\ell = f$, and symmetric bilinear form a. For this case, we have $p^{(1)}_{N,K} = -u_{N,K}$, $\tilde{N}^{(1)} = N$ and $\Delta^s = \alpha_{\mathrm{LB}}\,\Delta^2 + \delta^{\mathrm{comp}}_{\mathrm{KL}}$, $\delta^{\mathrm{comp}}_{\mathrm{KL}} := \delta_{\mathrm{KL}}(u_{N,K}) + 2\delta^{f}_{\mathrm{KL}}(u_{N,K})$.

Proposition 5.9. *In the compliant case with a symmetric bilinear form a and for $\eta_0(\mu,\omega)$ from (5.28), we assume that $\alpha_{\mathrm{LB}}(\mu,\omega)\Delta(\mu,\omega)^2 \ge \eta_0(\mu,\omega)^2\delta^{comp}_{\mathrm{KL}}(\mu,\omega)$. Then, the effectivity $\eta^s(\mu,\omega) := \frac{\Delta^s(\mu,\omega)}{|s(\mu,\omega)-s_{N,K}(\mu,\omega)|}$ is bounded by*

$$\eta^s(\mu,\omega) \le \eta_0(\mu,\omega)^2 \frac{\alpha_{\mathrm{LB}}(\mu,\omega)\Delta(\mu,\omega)^2 + \delta^{comp}_{\mathrm{KL}}(\mu,\omega)}{\alpha_{\mathrm{LB}}(\mu,\omega)\Delta(\mu,\omega)^2 - \eta_0(\mu,\omega)^2\delta^{comp}_{\mathrm{KL}}(\mu,\omega)}. \tag{5.32}$$

Proof. Following the proof of Theorem 5.8 yields for $\ell = f$ and $p^{(1)}_{N,K} = -u_{N,K}$

$$\begin{aligned} s - s_{N,K} &= f(u) - 2\,f^K(u_{N,K}) + a^K(u_{N,K},u_{N,K}) \\ &= a(u,u) + 2[f(u_{N,K}) - f^K(u_{N,K})] - 2f(u_{N,K}) + a(u_{N,K},u_{N,K}) \\ &\quad - [a(u_{N,K},u_{N,K}) - a^K(u_{N,K},u_{N,K})] \\ &= a(e,e) + 2[f(u_{N,K}) - f^K(u_{N,K})] - [a(u_{N,K},u_{N,K}) - a^K(u_{N,K},u_{N,K})]. \end{aligned}$$

Hence, we can estimate

$$\begin{aligned} a(e,e) &= s - s_{N,K} - 2[f(u_{N,K}) - f^K(u_{N,K})] + [a(u_{N,K}, u_{N,K}) - a^K(u_{N,K}, u_{N,K})] \\ &\leq |s - s_{N,K}| + \delta_{\mathrm{KL}}^{\mathrm{comp}}. \end{aligned}$$

Using Corollary 5.7, we get

$$\frac{\alpha_{\mathrm{LB}}}{\eta_0^2}\Delta^2 \;\leq\; \|e\|_{\mu,\omega}^2 \;=\; a(e,e) \;\leq\; |s - s_{N,K}| + \delta_{\mathrm{KL}}^{\mathrm{comp}}$$

which implies $|s - s_{N,K}| \geq \frac{\alpha_{\mathrm{LB}}}{\eta_0^2}\Delta^2 - \delta_{\mathrm{KL}}^{\mathrm{comp}}$. This yields

$$\frac{\Delta^s}{|s - s_{N,K}|} \leq \frac{\alpha_{\mathrm{LB}}\Delta^2 + \delta_{\mathrm{KL}}^{\mathrm{comp}}}{\frac{\alpha_{\mathrm{LB}}}{\eta_0^2}\Delta^2 - \delta_{\mathrm{KL}}^{\mathrm{comp}}},$$

which proves the claim. □

The assumption $\alpha_{\mathrm{LB}}(\mu,\omega)\Delta(\mu,\omega)^2 \geq \eta_0(\mu,\omega)^2\delta_{\mathrm{KL}}^{\mathrm{comp}}(\mu,\omega)$ is rather restrictive and can be validated only a posteriori. It requires either the energy norm error effectivity η_0 or the KL truncation error $\delta_{\mathrm{KL}}^{\mathrm{comp}}$ to be small. However, the effectivity bound is consistent with the deterministic case in the sense that for large K, it converges to the energy norm error effectivity bound η_0^2 as provided in Corollary 5.7, where c is approaching zero at the same time.

5.3.4 Quadratic Output

As a next step, we consider quadratic output functions of the form

$$s^2(\mu,\omega) \;:=\; [\ell(u(\mu,\omega);\mu)]^2,$$

where ℓ is an ω-independent linear functional. If ℓ were stochastic itself, the subsequently constructed error bounds would include terms depending on the magnitude of $s_{N,K}$ (cf. Remark 5.11). Also, it is readily seen that just squaring the output $s_{N,K}$ from (5.29) is not sufficient. In fact, since

$$s^2 - (s_{N,K})^2 \;=\; (s - s_{N,K})(s + s_{N,K}) \leq \Delta^s \cdot (s + s_{N,K}), \tag{5.33}$$

the right-hand side does not have the desirable "square" effect, as is typical in RBMs. Hence, we follow a different path by introducing an additional dual problem, namely, determining $p_K^{(2)}(\mu,\omega) \in X$ such that

$$a^K(v, p_K^{(2)}(\mu,\omega);\mu,\omega) = -2\, s_{N,K}(\mu,\omega) \cdot \ell(v;\mu) =: -\ell^{(2)}(v;\mu,\omega), \quad v \in X. \tag{5.34}$$

Of course, the solution of (5.34) reads $p_K^{(2)} = 2\, s_{N,K}\, p_K^{(1)}$, which, however, is useless in the RB context since we have a different parameter-dependent right-hand side and thus different RB spaces. Hence, we consider an RB space $\tilde{X}_N^{(2)} \subset X$, $\dim(\tilde{X}_N^{(2)}) = \tilde{N}^{(2)}$ and determine some $p_{N,K}^{(2)}(\mu,\omega) \in \tilde{X}_N^{(2)}$ such that

$$a^K(v, p_{N,K}^{(2)}(\mu,\omega);\mu,\omega) \;=\; -\ell^{(2)}(v;\mu,\omega), \quad v \in \tilde{X}_N^{(2)}. \tag{5.35}$$

We can apply the analysis performed in Section 5.3.2 and just need to adjust the notation. The dual error reads $\tilde{e}_{\mathrm{RB}}^{(2)} := p_K^{(2)} - p_{N,K}^{(2)}$, the residual as $\tilde{r}_{\mathrm{RB}}^{(2)}(v) := a^K(v, \tilde{e}_{\mathrm{RB}}^{(2)})$ and the RB bounds as $\tilde{\Delta}_{\mathrm{RB}}^{(2)} := {\alpha_{\mathrm{LB}}}^{-1} \sup_{v\in X} \left(\tilde{r}_{\mathrm{RB}}^{(2)}(v)/\|v\|_X\right)$. The KL truncation term $\tilde{\delta}_{\mathrm{KL}}^{(2)}$ is defined analogously to (5.20b) by replacing $p_{N,K}^{(1)}$ by $p_{N,K}^{(2)}$, and analogously to (5.21), $\tilde{\Delta}_{\mathrm{KL}}^{(2)} := {\alpha_{\mathrm{LB}}}^{-1} \sup_{v\in X} \left(\tilde{\delta}_{\mathrm{KL}}^{(2)}(v)/\|v\|_X\right)$. The terms $\delta_{\mathrm{KL}}^{\ell^{(2)}}(v;\mu)$ and $\Delta_{\mathrm{KL}}^{\ell^{(2)}}(\mu,\omega)$ vanish since ℓ is deterministic. Then, Proposition 5.3 and Corollary 5.4 yield the following estimate for $\tilde{e}^{(2)} := p^{(2)} - p_{N,K}^{(2)}$:

$$\|\tilde{e}^{(2)}(\mu,\omega)\|_X \;\le\; \tilde{\Delta}^{(2)}(\mu,\omega) \;:=\; \tilde{\Delta}_{\mathrm{RB}}^{(2)}(\mu,\omega) + \tilde{\Delta}_{\mathrm{KL}}^{(2)}(\mu,\omega). \tag{5.36}$$

We consider the approximation $[\ell(u_{N,K}(\mu,\omega);\mu,\omega)]^2$. Similar to the definition of $s_{N,K}$ in Section 5.3.3, we add correction terms and consider

$$s_{N,K}^{[2]}(\mu,\omega) := \left(\ell(u_{N,K})\right)^2 - \left(r_{\mathrm{RB}}(p_{N,K}^{(1)})\right)^2 - r_{\mathrm{RB}}(p_{N,K}^{(2)}). \tag{5.37}$$

It is important to keep in mind that we distinguish the squared approximation $(s_{N,K})^2 = s_{N,K} \cdot s_{N,K}$ from the approximation $s_{N,K}^{[2]}$ of the square of s. In fact, it is easy to see that we can also write $s_{N,K}^{[2]}$ in terms of $s_{N,K} = \ell(u_{N,K}) - r_{\mathrm{RB}}(p_{N,K}^{(1)})$,

$$s_{N,K}^{[2]}(\mu,\omega) = (s_{N,K})^2 + 2 s_{N,K} \cdot r_{\mathrm{RB}}(p_{N,K}^{(1)}) - r_{\mathrm{RB}}(p_{N,K}^{(2)}), \tag{5.38}$$

i.e., we have two additional correction terms. For $\tilde{X}_N^{(2)} = \tilde{X}_N^{(1)}$, the correction terms in (5.38) would cancel out. We define the quadratic output error bound

$$\Delta^{s^2}(\mu,\omega) := (\Delta^s)^2 \;+\; \alpha_{\mathrm{LB}} \Delta \tilde{\Delta}^{(2)} + \delta_{\mathrm{KL}}(p_{N,K}^{(2)}) + \delta_{\mathrm{KL}}^f(p_{N,K}^{(2)}) \tag{5.39}$$

and obtain the following result.

Theorem 5.10. *$\left|s^2(\mu,\omega) - s_{N,K}^{[2]}(\mu,\omega)\right| \le \Delta^{s^2}(\mu,\omega)$ holds for all $\mu \in \mathcal{P}$, $\omega \in \Omega$.*

Proof. With (5.38), the output error is given by

$$\begin{aligned} s^2 - s_{N,K}^{[2]} &= s^2 - (s_{N,K})^2 - 2s_{N,K}\, r_{\mathrm{RB}}(p_{N,K}^{(1)}) + r_{\mathrm{RB}}(p_{N,K}^{(2)}) \\ &= (s - s_{N,K})^2 + 2s_{N,K}(s - s_{N,K}) - 2s_{N,K}\, r_{\mathrm{RB}}(p_{N,K}^{(1)}) + r_{\mathrm{RB}}(p_{N,K}^{(2)}). \end{aligned}$$

Using $s_{N,K} = \ell(u_{N,K}) - r_{\mathrm{RB}}(p_{N,K}^{(1)})$ yields

$$2s_{N,K}(s - s_{N,K}) = 2s_{N,K}\left(\ell(u) - \ell(u_{N,K}) + r_{\mathrm{RB}}(p_{N,K}^{(1)})\right).$$

Putting these together, replacing $2s_{N,K}\ell$ by $\ell^{(2)}$, we have

$$s^2 - s_{N,K}^{[2]} = (s - s_{N,K})^2 \;+\; \ell^{(2)}(u) - \ell^{(2)}(u_{N,K}) + r_{\mathrm{RB}}(p_{N,K}^{(2)}). \tag{5.40}$$

From Theorem 5.8, we know that $(s - s_{N,K})^2 \leq (\Delta^s)^2$. The second part of (5.40) can be estimated analogously to Theorem 5.8 by replacing ℓ by $\ell^{(2)}$ and $p^{(1)}$ by $p^{(2)}$. Since $\ell = \ell^K$, we obtain

$$\begin{aligned} &\ell^{(2)}(u) - \ell^{(2)}(u_{N,K}) + r_{\mathrm{RB}}(p_{N,K}^{(2)}) \\ &= -\,\tilde{r}_{\mathrm{RB}}^{(2)}(e) \;+\; [a(e, p_{N,K}^{(2)}) - a^K(e, p_{N,K}^{(2)})] \\ &\quad -\,[f(p_{N,K}^{(2)}) - f^K(p_{N,K}^{(2)})] \;+\; [a(u_{N,K}, p_{N,K}^{(2)}) - a^K(u_{N,K}, p_{N,K}^{(2)})], \end{aligned} \tag{5.41}$$

which can be bounded by $\alpha_{\mathrm{LB}}\Delta\tilde{\Delta}_{\mathrm{RB}}^{(2)} + \alpha_{\mathrm{LB}}\Delta\tilde{\Delta}_{\mathrm{KL}}^{(2)} + \delta_{\mathrm{KL}}^{f}(p_{N,K}^{(2)}) + \delta_{\mathrm{KL}}(p_{N,K}^{(2)})$. □

If Δ^s is already small, the first part of the error bound Δ^{s^2} will be comparatively negligible. The second part of the error bound is of the same form as Δ^s in (5.30). Hence, we can hope that Δ^{s^2} is approximately of the same order as Δ^s.

Remark 5.11. If ℓ were stochastic itself, we would have to take the respective truncation error of the right-hand side $\ell^{(2)}$ into account. Since $\ell^{(2)} = s_{N,K}\ell$, this error directly depends on $s_{N,K}$. Hence, we would have to add the terms $\alpha_{\mathrm{LB}}\Delta \cdot s_{N,K}\Delta_{\mathrm{KL}}^{\ell}$ and $s_{N,K}\delta_{\mathrm{KL}}^{\ell}(p_{N,K}^{(2)})$ to the error bound in (5.39).

5.4 Statistical Output Error Analysis

In this section, we consider first and second moments of the linear output functional $s(\mu, \omega) = \ell(u(\mu, \omega); \mu)$,

$$\mathbb{M}_1(\mu) := \mathbb{E}\left[s(\mu, \cdot)\right], \quad \mathbb{M}_2(\mu) := \mathbb{E}\left[s^2(\mu, \cdot)\right], \quad \mathbb{V}(\mu) := \mathbb{M}_2(\mu) - \left(\mathbb{M}_1(\mu)\right)^2.$$

We assume again that the functional ℓ is deterministic, i.e., that there is no explicit dependence on the stochastic parameter ω but the randomness of the output functional s is only through u. We start with the following lemma.

Lemma 5.12. *Assuming independence of a and f as stated in Section 5.1.1, we have*

$$\mathbb{E}\left[a(u_{N,K}, p^{(i)}_{N,K}) - a^K(u_{N,K}, p^{(i)}_{N,K})\right] = 0, \quad \mathbb{E}\left[f(p^{(i)}_{N,K}) - f^K(p^{(i)}_{N,K})\right] = 0,$$

$i = 1, 2, 3$, where $p^{(3)}_{N,K}(\mu, \omega)$ is given in (5.45) and ℓ is assumed to be deterministic.

Proof. Since $u_{N,K}$ and $p^{(i)}_{N,K}$ depend only on truncated forms, they depend only on the random variables $\left\{\xi^a_{q,k}\right\}^{k=1,\dots,K^a_q}_{q=1,\dots,Q^a}$ and $\left\{\xi^f_{q,k}\right\}^{k=1,\dots,K^f_q}_{q=1,\dots,Q^f}$. Since $\xi^b_{q,k}$ and $\xi^{b'}_{q',k'}$ are uncorrelated for $(q,k,b) \neq (q',k',b')$, both $u_{N,K}$ and $p^{(i)}_{N,K}$ are uncorrelated to $\left\{\xi^a_{q,k}\right\}^{k>K^a_q}_{q=1,\dots,Q^a}$ and $\left\{\xi^f_{q,k}\right\}^{k>K^f_q}_{q=1,\dots,Q^f}$. We thus obtain

$$\begin{aligned}
&\mathbb{E}\left[a(u_{N,K}, p^{(i)}_{N,K}) - a^K(u_{N,K}, p^{(i)}_{N,K})\right] \\
&= \mathbb{E}\left[\sum_{q=1}^{Q^a} \sum_{k=K^a_q+1}^{\infty} \theta^a_q(\mu)\xi^a_{q,k}(\cdot) a_{q,k}(u_{N,K}, p^{(i)}_{N,K})\right] \\
&= \sum_{q=1}^{Q^a} \sum_{k=K^a_q+1}^{\infty} \theta^a_q(\mu) \underbrace{\mathbb{E}\left[\xi^a_{q,k}(\cdot)\right]}_{=0} \mathbb{E}\left[a_{q,k}(u_{N,K}, p^{(i)}_{N,K})\right] = 0
\end{aligned}$$

and, analogously, $\mathbb{E}[f(p^{(i)}_{N,K}) - f^K(p^{(i)}_{N,K})] = 0$. □

Remark 5.13. In the proof of Lemma 5.12 we see that the assumption in Section 5.1.1 that all parts a_q and f_q in (5.2) and (5.3), respectively, are stochastically independent is too strong. It suffices to assume that $u_{N,K}$ and $p^{(i)}_{N,K}$ are uncorrelated to $\left\{\xi^b_{q,k}\right\}^{k>K^b_q}_{q=1,\dots,Q^b}$, $b \in \{a, f\}$, i.e., that $\left\{\xi^b_{q,k}\right\}^{k=1,\dots,K^b_q}_{q=1,\dots,Q^b}$, $b \in \{a, f\}$, are uncorrelated to $\left\{\xi^b_{q,k}\right\}^{k>K^b_q}_{q=1,\dots,Q^b}$, $b \in \{a, f\}$. We have seen in Section 2.2.3 that it is possible to obtain joint KL expansions for different, possibly correlated, processes $a_q, a_{q'}$ or $f_q, f_{q'}$, $q \neq q'$, or even $a_q, f_{q'}$. In this case, the respective random variables are identical. For our case, we would have $\xi^a_{q,k} = \xi^a_{q',k}$, $\xi^f_{q,k} = \xi^f_{q',k}$, or $\xi^a_{q,k} = \xi^f_{q',k}$. Using the same truncation values for correlated terms, i.e., $K^a_q = K^a_{q'}$, $K^f_q = K^f_{q'}$, or $K^a_q = K^f_{q'}$, it is still certified that Lemma 5.12 holds. Therefore, it is possible to deal with completely dependent terms and it is thus also possible to apply the subsequent theory to very general problem classes.

Hence, for the presented a-posteriori analysis, it is sufficient to require that the following assumption holds.

Assumption 5.14. *The sets of random variables*

$$\left\{\xi_{q,k}^{b}\right\}_{q=1,\dots,Q^b}^{k=1,\dots,K_q^b},\ b\in\{a,f\},\qquad \textit{and}\qquad \left\{\xi_{q,k}^{b}\right\}_{q=1,\dots,Q^b}^{k>K_q^b},\ b\in\{a,f\},$$

are uncorrelated from each other.

5.4.1 First and Second Moments

The straightforward estimate for the first moment $\mathbb{M}_1(\mu)$ is given by $\mathbb{M}_{1,NK}(\mu) := \mathbb{E}\left[s_{N,K}(\mu,\cdot)\right]$, and we define the error bound

$$\Delta^{\mathbb{M}_1}(\mu) := \mathbb{E}\left[\alpha_{\mathrm{LB}}\Delta\tilde{\Delta}^{(1)}\right]. \tag{5.42}$$

Corollary 5.15. $|\mathbb{M}_1(\mu) - \mathbb{M}_{1,NK}(\mu)| \le \Delta^{\mathbb{M}_1}(\mu)$ *holds for all* $\mu\in\mathcal{P}$.

Proof. Equation (5.31), Lemma 5.12, and $\ell = \ell^K$ yield

$$\begin{aligned}\mathbb{M}_1 - \mathbb{M}_{1,NK} &= \mathbb{E}\left[-\tilde{r}_{\mathrm{RB}}^{(1)}(e) \ +\ a(e,p_{N,K}^{(1)}) - a^K(e,p_{N,K}^{(1)})\right]\\ &\quad + \mathbb{E}\left[a(u_{N,K},p_{N,K}^{(1)}) - a^K(u_{N,K},p_{N,K}^{(1)})\right] - \mathbb{E}\left[f(p_{N,K}^{(1)}) - f^K(p_{N,K}^{(1)})\right]\\ &= \mathbb{E}\left[-\tilde{r}_{\mathrm{RB}}^{(1)}(e) \ +\ a(e,p_{N,K}^{(1)}) - a^K(e,p_{N,K}^{(1)})\right].\end{aligned}$$

Following the proof of Theorem 5.8, we obtain the desired result. □

Analogously, the straightforward estimate for the second moment $\mathbb{M}_2(\mu)$ is given by $\mathbb{M}_{2,NK}(\mu) := \mathbb{E}\big[s_{N,K}^{[2]}(\mu,\cdot)\big]$ and we define the error bound

$$\Delta^{\mathbb{M}_2}(\mu) := \mathbb{E}\left[(\Delta^s)^2 \ +\ \alpha_{\mathrm{LB}}\Delta\tilde{\Delta}^{(2)}\right]. \tag{5.43}$$

Corollary 5.16. $|\mathbb{M}_2(\mu) - \mathbb{M}_{2,NK}(\mu)| \le \Delta^{\mathbb{M}_2}(\mu)$ *holds for all* $\mu\in\mathcal{P}$.

Proof. Equations (5.40) and (5.41), Lemma 5.12, and $\ell = \ell^K$ yield

$$\begin{aligned}\mathbb{M}_2 - \mathbb{M}_{2,NK} &= \mathbb{E}\left[(s - s_{N,K})^2\right] - \mathbb{E}\left[\tilde{r}_{\mathrm{RB}}^{(2)}(e) \ +\ a(e,p_{N,K}^{(2)}) - a^K(e,p_{N,K}^{(2)})\right]\\ &\quad - \mathbb{E}[f(p_{N,K}^{(2)}) - f^K(p_{N,K}^{(2)})] + \mathbb{E}[a(u_{N,K},p_{N,K}^{(2)}) - a^K(u_{N,K},p_{N,K}^{(2)})]\\ &= \mathbb{E}\left[(s - s_{N,K})^2\right] - \mathbb{E}\left[\tilde{r}_{\mathrm{RB}}^{(2)}(e) \ +\ a(e,p_{N,K}^{(2)}) - a^K(e,p_{N,K}^{(2)})\right].\end{aligned}$$

Following the proof of Theorem 5.10, we obtain the desired result. □

5.4.2 Squared First Moment

In order to get an estimation of the variance, it remains to find an estimation for the the squared first moment. We follow the same approach as in Section 5.3.4 and introduce a third dual problem with right-hand side $\ell^{(3)}(v;\mu) := 2\mathbb{M}_{1,NK}(\mu)\,\ell(v;\mu)$. The dual and the corresponding reduced systems are then given by

$$a^K(v, p_K^{(3)};\mu,\omega) \;=\; -\ell^{(3)}(v;\mu), \quad v \in X, \tag{5.44}$$

$$a^K(v, p_{N,K}^{(3)};\mu,\omega) \;=\; -\ell^{(3)}(v;\mu), \quad v \in \tilde{X}_N^{(3)}, \tag{5.45}$$

respectively, where $\tilde{X}_N^{(3)} \subset X$ denotes the RB space of dimension $\dim(\tilde{X}_N^{(3)}) = \tilde{N}^{(3)}$. The error analysis is now mainly straightforward, following Section 5.3.4. We denote the new dual error by $\tilde{e}_{\mathrm{RB}}^{(3)} := p_K^{(3)} - p_{N,K}^{(3)}$ and the residual by $\tilde{r}_{\mathrm{RB}}^{(3)}(v) := a^K(v, \tilde{e}_{\mathrm{RB}}^{(3)})$ to define the RB bound $\tilde{\Delta}_{\mathrm{RB}}^{(3)} := {\alpha_{\mathrm{LB}}}^{-1} \sup_{v\in X}(\tilde{r}_{\mathrm{RB}}^{(3)}(v)/\|v\|_X)$. The KL truncation term $\tilde{\delta}_{\mathrm{KL}}^{(3)}$ is defined analogously to (5.20b) by replacing $p_{N,K}^{(1)}$ by $p_{N,K}^{(3)}$, and analogously to (5.21), $\tilde{\Delta}_{\mathrm{KL}}^{(3)} := {\alpha_{\mathrm{LB}}}^{-1} \sup_{v\in X}(\tilde{\delta}_{\mathrm{KL}}^{(3)}(v)/\|v\|_X)$. Then, Proposition 5.3 and Corollary 5.4 yield the following estimate for $\tilde{e}^{(3)} := p^{(3)} - p_{N,K}^{(3)}$:

$$\|\tilde{e}^{(3)}(\mu,\omega)\|_X \;\le\; \tilde{\Delta}^{(3)}(\mu,\omega) \;:=\; \tilde{\Delta}_{\mathrm{RB}}^{(3)}(\mu,\omega) + \tilde{\Delta}_{\mathrm{KL}}^{(3)}(\mu,\omega). \tag{5.46}$$

We define the approximation of the squared first moment, adding some correction terms. Analogously to (5.38), we consider

$$\mathbb{M}_{1,NK}^{[2]}(\mu) = (\mathbb{M}_{1,NK})^2 + 2\mathbb{M}_{1,NK} \cdot \mathbb{E}\left[r_{\mathrm{RB}}(p_{N,K}^{(1)})\right] - \mathbb{E}\left[r_{\mathrm{RB}}(p_{N,K}^{(3)})\right]. \tag{5.47}$$

Note the distinction between the squared approximation $(\mathbb{M}_{1,NK})^2 = \mathbb{M}_{1,NK} \cdot \mathbb{M}_{1,NK}$ and the direct approximation $\mathbb{M}_{1,NK}^{[2]}$ of the squared first moment. The error bound is given by

$$\Delta^{\mathbb{M}_1^2}(\mu) \;:=\; (\Delta^{\mathbb{M}_1})^2 \;+\; \mathbb{E}\left[\alpha_{\mathrm{LB}}\Delta\tilde{\Delta}^{(3)}\right]. \tag{5.48}$$

Theorem 5.17. $\left|\mathbb{M}_1^2(\mu) - \mathbb{M}_{1,NK}^{[2]}(\mu)\right| \le \Delta^{\mathbb{M}_1^2}(\mu)$ *holds for all* $\mu \in \mathcal{P}$.

Proof. Analogously to Theorem 5.10, the output error is given by

$$\mathbb{M}_1^2 - \mathbb{M}_{1,NK}^{[2]} = (\mathbb{M}_1 - \mathbb{M}_{1,NK})^2 \;+\; \mathbb{E}\left[\ell^{(3)}(u) - \ell^{(3)}(u_{N,K}) + r_{\mathrm{RB}}(p_{N,K}^{(3)})\right].$$

From Corollary 5.15, we know that $(\mathbb{M}_1 - \mathbb{M}_{1,NK})^2 \leq (\Delta^{\mathbb{M}_1})^2$. Analogously to Theorem 5.8, using $\ell = \ell^K$ and replacing ℓ by $\ell^{(3)}$ and $p^{(1)}$ by $p^{(3)}$, we obtain

$$\begin{aligned}
&\mathbb{E}\left[\ell^{(3)}(u) - \ell^{(3)}(u_{N,K}) + r_{\mathrm{RB}}(p^{(3)}_{N,K})\right] \\
&= \mathbb{E}\left[-\tilde{r}^{(3)}_{\mathrm{RB}}(e) \;+\; a(e, p^{(3)}_{N,K}) - a^K(e, p^{(3)}_{N,K})\right] \\
&\quad - \mathbb{E}\left[f(p^{(3)}_{N,K}) - f^K(p^{(3)}_{N,K})\right] + \mathbb{E}\left[a(u_{N,K}, p^{(3)}_{N,K}) - a^K(u_{N,K}, p^{(3)}_{N,K})\right] \\
&= \mathbb{E}\left[-\tilde{r}^{(3)}_{\mathrm{RB}}(e) \;+\; a(e, p^{(3)}_{N,K}) - a^K(e, p^{(3)}_{N,K})\right],
\end{aligned}$$

where the last equation is obtained by Lemma 5.12. The result can be bounded analogously to Theorem 5.8 by $\mathbb{E}[\alpha_{\mathrm{LB}}\Delta\tilde{\Delta}^{(3)}_{\mathrm{RB}} + \alpha_{\mathrm{LB}}\Delta\tilde{\Delta}^{(3)}_{\mathrm{KL}}]$. □

5.4.3 Variance

It is straightforward to define

$$\mathbb{V}_{NK}(\mu) := \mathbb{M}_{2,NK}(\mu) - \mathbb{M}^{[2]}_{1,NK}(\mu), \tag{5.49}$$

and it is furthermore clear that $|\mathbb{V} - \mathbb{V}_{NK}| \leq \mathbb{E}[\Delta^{s^2}] + \Delta^{\mathbb{M}_1^2}$ is an upper bound for the error. However, we can derive more precise error bounds. Denoting $\tilde{r}^{(2-3)}_{\mathrm{RB}}(v) := a^K(v, \tilde{e}^{(2)}_{\mathrm{RB}} - \tilde{e}^{(3)}_{\mathrm{RB}})$ and $\tilde{\Delta}^{(2-3)}_{\mathrm{RB}} := {\alpha_{\mathrm{LB}}}^{-1} \sup_{v\in X} \left(\tilde{r}^{(2-3)}_{\mathrm{RB}}(v)/\|v\|_X\right)$ as well as defining the KL truncation term $\tilde{\delta}^{(2-3)}_{\mathrm{KL}}$ by (5.20b), replacing $p^{(1)}_{N,K}$ by $(p^{(2)}_{N,K} - p^{(3)}_{N,K})$, and analogously to (5.21), $\tilde{\Delta}^{(2-3)}_{\mathrm{KL}} := {\alpha_{\mathrm{LB}}}^{-1} \sup_{v\in X} \left(\tilde{\delta}^{(2-3)}_{\mathrm{KL}}(v)/\|v\|_X\right)$, we obtain $\|\tilde{e}^{(2)} - \tilde{e}^{(3)}\|_X \leq \tilde{\Delta}^{(2-3)} := \tilde{\Delta}^{(2-3)}_{\mathrm{RB}} + \tilde{\Delta}^{(2-3)}_{\mathrm{KL}}$ and the variance error bound

$$\Delta^{\mathbb{V}}(\mu) \;:=\; \mathbb{E}\left[(\Delta^s)^2\right] + (\Delta^{\mathbb{M}_1})^2 \;+\; \mathbb{E}\left[\alpha_{\mathrm{LB}}\Delta\tilde{\Delta}^{(2-3)}\right]. \tag{5.50}$$

Theorem 5.18. *$|\mathbb{V}(\mu) - \mathbb{V}_{NK}(\mu)| \leq \Delta^{\mathbb{V}}(\mu)$ holds for all $\mu \in \mathcal{P}$.*

Proof. From Theorems 5.10 and 5.17 we know that

$$\begin{aligned}
\mathbb{V} - \mathbb{V}_{NK} = \mathbb{E}\left[(s - s_{N,K})^2\right] \;-\; &(\mathbb{M}_1 - \mathbb{M}_{1,NK})^2 \\
+\; &\mathbb{E}\left[\ell^{(2)}(u) - \ell^{(2)}(u_{N,K}) + r_{\mathrm{RB}}(p^{(2)}_{N,K})\right] \\
-\; &\mathbb{E}\left[\ell^{(3)}(u) - \ell^{(3)}(u_{N,K}) + r_{\mathrm{RB}}(p^{(3)}_{N,K})\right],
\end{aligned}$$

and the first two terms can be bounded by $\mathbb{E}\left[(\Delta^s)^2\right]$ and $(\Delta^{\mathbb{M}_1})^2$, respectively. From (5.41), Lemma 5.12, and Theorem 5.17, we have for $i = 2, 3$

$$\mathbb{E}\left[\ell^{(i)}(u) - \ell^{(i)}(u_{N,K}) + r_{\mathrm{RB}}(p^{(i)}_{N,K})\right] = \mathbb{E}\left[-\tilde{r}^{(i)}_{\mathrm{RB}}(e) + a(e, p^{(i)}_{N,K}) - a^K(e, p^{(i)}_{N,K})\right].$$

We subtract the two expressions and again follow the proof of Theorem 5.8. The claim follows directly using the definitions above. □

In our numerical experiments, we have observed that it is sufficient to use the same reduced space for the two additional dual problems (5.35) and (5.45), i.e., $\tilde{X}_N^{(2)} = \tilde{X}_N^{(3)}$. Then, it holds that $p_{N,K}^{(3)}(\mu,\omega) = p_{N,K}^{(2)}(\mu,\omega)\mathbb{M}_{1,NK}(\mu)/s_{N,K}(\mu,\omega)$, and it is sufficient to solve only one additional dual problem. Hence, we consider

$$a^K(v, p_{N,K}^{(4)}(\mu,\omega);\mu,\omega) \;=\; -2\ell(v;\mu), \quad v \in \tilde{X}_N^{(2)}, \tag{5.51}$$

such that $p_{N,K}^{(2)} = s_{N,K} \cdot p_{N,K}^{(4)}$ and $p_{N,K}^{(3)} = \mathbb{M}_{1,NK} \cdot p_{N,K}^{(4)}$. For a faster evaluation of the variance error bound (5.50), we could use $p_{N,K}^{(2)} - p_{N,K}^{(3)} = (s_{N,K} - \mathbb{M}_{1,NK})\, p_{N,K}^{(4)}$. Furthermore, defining $\tilde{\delta}_{\mathrm{KL}}^{(4)}$, $\tilde{\Delta}_{\mathrm{KL}}^{(4)}$, $\tilde{\Delta}_{\mathrm{RB}}^{(4)}$, and $\tilde{\Delta}^{(4)}$ analogously to $\tilde{\delta}_{\mathrm{KL}}^{(1)}$, $\tilde{\Delta}_{\mathrm{KL}}^{(1)}$, $\tilde{\Delta}_{\mathrm{RB}}^{(1)}$, and $\tilde{\Delta}^{(1)}$, respectively, we obtain, e.g., $\tilde{\Delta}_{\mathrm{RB}}^{(2-3)} = |s_{N,K} - \mathbb{M}_{1,NK}|\, \tilde{\Delta}_{\mathrm{RB}}^{(4)}$. Analogously, we can construct the error terms $\tilde{\delta}_{\mathrm{KL}}^{(i)}$, $\tilde{\Delta}_{\mathrm{KL}}^{(i)}$, $\tilde{\Delta}_{\mathrm{RB}}^{(i)}$, and $\tilde{\Delta}^{(i)}$, $i \in \{2, 3, 2-3\}$, using the respective term for $i = 4$. Still, it is possible to use two different RB spaces such that both dual problems (5.35) and (5.45) have to be solved. The theory does not change for that case.

5.5 Higher Moments

Often, it is desirable to evaluate higher moments, i.e., $\mathbb{E}\left[(s(\mu,\omega))^n\right]$, $n > 2$. To some extend, it is possible to extend the proposed scheme of Section 5.3.4 and use additional dual problems to improve the approximation of such outputs. However, it is not completely straightforward to derive the appropriate estimates and we do not have a simple general form for arbitrary moments. Furthermore, the error bounds do not show the same nice form as for the quadratic outputs and contain terms directly dependent on $s_{N,K}$. Exemplarily, we provide results for the third and fourth moment in this section.

5.5.1 Third Moment

We follow the approach of Section 5.3.4 and introduce a new dual problem with right-hand side $\ell^{(5)}(v;\mu,\omega) := 3\left(s_{N,K}(\mu,\omega)\right)^2 \ell(v;\mu)$. The dual and the corre-

sponding reduced systems are then given by

$$a^K(v, p_K^{(5)}; \mu, \omega) \;=\; -\ell^{(5)}(v; \mu, \omega), \quad v \in X, \tag{5.52}$$

$$a^K(v, p_{N,K}^{(5)}; \mu, \omega) \;=\; -\ell^{(5)}(v; \mu, \omega), \quad v \in \tilde{X}_N^{(5)}, \tag{5.53}$$

respectively, where $\tilde{X}_N^{(5)} \subset X$ denotes the RB space of dimension $\dim(\tilde{X}_N^{(5)}) = \tilde{N}^{(5)}$. We define the RB and KL residuals $\tilde{r}_{\mathrm{RB}}^{(5)}(v)$ and $\tilde{\delta}_{\mathrm{KL}}^{(5)}$ analogously to Section 5.3.1, replacing $p_{N,K}^{(1)}$ by $p_{N,K}^{(5)}$, and obtain the corresponding new RB and KL bounds $\tilde{\Delta}_{\mathrm{RB}}^{(5)}$ and $\tilde{\Delta}_{\mathrm{KL}}^{(5)}$. Let $p^{(5)}(\mu, \omega)$ be the solution of the untruncated version of (5.52) and let $\tilde{e}^{(5)} := p^{(5)} - p_{N,K}^{(5)}$. Then, Proposition 5.3 and Corollary 5.4 yield the following estimate:

$$\|\tilde{e}^{(5)}(\mu, \omega)\|_X \;\leq\; \tilde{\Delta}^{(5)}(\mu, \omega) \;:=\; \tilde{\Delta}_{\mathrm{RB}}^{(5)}(\mu, \omega) + \tilde{\Delta}_{\mathrm{KL}}^{(5)}(\mu, \omega). \tag{5.54}$$

We define an approximation of the the cubed output $(s(\mu, \omega))^3$, adding additional correction terms. We consider

$$\begin{aligned} s_{N,K}^{[3]}(\mu, \omega) &:= (s_{N,K})^3 + 3(s_{N,K})^2 r_{\mathrm{RB}}(p_{N,K}^{(1)}) - r_{\mathrm{RB}}(p_{N,K}^{(5)}) \\ &= \big(\ell(u_{N,K})\big)^3 - \big(r_{\mathrm{RB}}(p_{N,K}^{(1)})\big)^3 - r_{\mathrm{RB}}(p_{N,K}^{(5)}) - 3 s_{N,K} \big(r_{\mathrm{RB}}(p_{N,K}^{(1)})\big)^2. \end{aligned} \tag{5.55}$$

Comparing the approximation of the second equation with the approximation of the squared output in (5.37), it may be surprising that the last term is not removed by an additional correction term. However, we will show that this is not (cf. Remark 5.21).

We define the cubic output error bound

$$\Delta^{s^3}(\mu, \omega) := (\Delta^s)^3 + 3|s_{N,K}|(\Delta^s)^2 + \alpha_{LB} \Delta \tilde{\Delta}^{(5)} + \delta_{\mathrm{KL}}^f(p_{N,K}^{(5)}) + \delta_{\mathrm{KL}}(p_{N,K}^{(5)}) \tag{5.56}$$

and obtain the following result.

Proposition 5.19. $\big|s^3(\mu, \omega) - s_{N,K}^{[3]}(\mu, \omega)\big| \leq \Delta^{s^3}(\mu, \omega)$ *holds for all* $(\mu, \omega) \in \mathcal{P} \times \Omega$.

Proof. With (5.55), the cubic output error is given by

$$s^3 - s_{N,K}^{[3]} = s^3 - (s_{N,K})^3 - 3(s_{N,K})^2 r_{\mathrm{RB}}(p_{N,K}^{(1)}) + r_{\mathrm{RB}}(p_{N,K}^{(5)}).$$

It is straightforward to rewrite the first two terms, i.e., the error without correction terms, as

$$\begin{aligned} s^3 - (s_{N,K})^3 &= (s - s_{N,K})^3 + 3 s s_{N,K} (s - s_{N,K}) \\ &= (s - s_{N,K})^3 + 3(s_{N,K})^2 (s - s_{N,K}) + 3 s_{N,K} (s - s_{N,K})^2. \end{aligned}$$

From Theorem 5.8, we know that $|s-s_{N,K}|^3 \leq (\Delta^s)^3$, and $|3s_{N,K}(s-s_{N,K})^2| \leq 3|s_{N,K}|(\Delta^s)^2$. It remains to estimate the second term with the correction terms. We obtain

$$\begin{aligned} & 3(s_{N,K})^2(s-s_{N,K}) - 3(s_{N,K})^2 r_{\mathrm{RB}}(p_{N,K}^{(1)}) + r_{\mathrm{RB}}(p_{N,K}^{(5)}) \\ = \;& 3(s_{N,K})^2(\ell(u)-\ell(u_{N,K})) + r_{\mathrm{RB}}(p_{N,K}^{(5)}) \\ = \;& -\left(\ell^{(5)}(u) - \ell^{(5)}(u_{N,K}) + r_{\mathrm{RB}}(p_{N,K}^{(5)})\right) \end{aligned}$$

which can be estimated analogously to Theorem 5.10 by $\alpha_{LB}\Delta\tilde{\Delta}^{(5)} + \delta_{\mathrm{KL}}^f(p_{N,K}^{(5)}) + \delta_{\mathrm{KL}}(p_{N,K}^{(5)})$. □

The estimate for the third moment $\mathbb{M}_3(\mu)$ is given by $\mathbb{M}_{3,NK}(\mu) := \mathbb{E}\big[s_{N,K}^{[3]}(\mu,\cdot)\big]$, and we define the error bound

$$\Delta^{\mathbb{M}_3}(\mu) := \mathbb{E}\left[(\Delta^s)^3 + 3|s_{N,K}|(\Delta^s)^2 + \alpha_{LB}\Delta\tilde{\Delta}^{(5)}\right]. \tag{5.57}$$

Corollary 5.20. *$|\mathbb{M}_3(\mu) - \mathbb{M}_{3,NK}(\mu)| \leq \Delta^{\mathbb{M}_3}(\mu)$ holds for all $\mu \in \mathcal{P}$.*

Proof. We use the results of Theorem 5.19 and Lemma 5.12. Analogously to Corollary 5.15 and Corollary 5.16, we derive

$$\begin{aligned} \mathbb{M}_3 - \mathbb{M}_{3,NK} = \;& \mathbb{E}\left[(s-s_{N,K})^3 + 3s_{N,K}(s-s_{N,K})^2\right] \\ & - \mathbb{E}\left[\tilde{r}_{\mathrm{RB}}^{(5)}(e) \;+\; a(e, p_{N,K}^{(5)} - a^K(e, p_{N,K}^{(5)})\right] \end{aligned}$$

which directly leads to the desired result. □

Certainly, the term $3|s_{N,K}|(\Delta^s)^2$ in the error bounds for the cubed output and the third moment are unsatisfactory. However, the error bound still outperforms straightforward estimations. In the following remark, we show the problem that occurs while trying to remove the term.

Remark 5.21. To avoid the terms $3|s_{N,K}|(\Delta^s)^2$ in (5.56) and (5.57), we would like to introduce a new dual problem with the right-hand side $\ell^{(6)}(v) = 3s_{N,K}\ell(v)\ell(v)$. Obviously, this is not possible. However, assuming we could make the impossible come true, we would add the correction terms $3s_{N,K}\big(r_{\mathrm{RB}}(p_{N,K}^{(1)})\big)^2$ and $r_{\mathrm{RB}}(p_{N,K}^{(6)})$ in (5.55). Then, the approximation would be consistent to (5.37) and the error bound for $\mathbb{M}_{3,NK}$ would be of the form $\mathbb{E}\big[(\Delta^s)^3 + \alpha_{LB}\Delta(\tilde{\Delta}^{(5)} + \tilde{\Delta}^{(6)})\big]$.

5.5.2 Fourth Moment

It is clear that the problem of the given approach that occurred in the derivation of good error bounds for the third moment will be rather more critical for higher moments. We will briefly show this aspect for the fourth moment, where a straightforward estimation of $s^4(\mu,\omega)$ could be given by $\left(s^{[2]}_{N,K}\right)^2$ such that

$$s^4 - \left(s^{[2]}_{N,K}\right)^2 \;=\; (s^2 - s^{[2]}_{N,K})^2 \,+\, 2s^{[2]}_{N,K}(s^2 - s^{[2]}_{N,K}) \;\leq\; (\Delta^{s^2})^2 \,+\, 2s^{[2]}_{N,K}\,\Delta^{s^2}. \quad (5.58)$$

Alternatively, let us introduce the additional dual problem, already in reduced form, following the idea of the previous sections:

$$a^K(v, p^{(6)}_{N,K}; \mu, \omega) \;=\; -\ell^{(6)}(v;\mu,\omega), \quad v \in \tilde{X}^{(6)}_N, \tag{5.59}$$

where $\ell^{(6)}(v;\mu,\omega) := 4s_{N,K}(\mu,\omega)\, s^{[2]}_{N,K}(\mu,\omega)\, \ell(v;\mu)$. We then define an approximation of s^4, using again two correction terms. We obtain

$$s^{[4]}_{N,K}(\mu,\omega) := \left(s^{[2]}_{N,K}\right)^2 + 2s^{[2]}_{N,K} \cdot r_{\mathrm{RB}}(p^{(2)}_{N,K}) - r_{\mathrm{RB}}(p^{(6)}_{N,K})$$

which is exactly analogous to the approximation $s^{[2]}_{N,K}$ in (5.38). Analogously to the error of the squared output in the proof of Theorem 5.10, we have

$$s^4 - s^{[4]}_{N,K} = (s^2 - s^{[2]}_{N,K})^2 \,+\, 2s^{[2]}_{N,K}(s^2 - s^{[2]}_{N,K}) - 2s^{[2]}_{N,K} \cdot r_{\mathrm{RB}}(p^{(2)}_{N,K}) - r_{\mathrm{RB}}(p^{(6)}_{N,K}).$$

A reformulation of the second term yields

$$\begin{aligned} 2s^{[2]}_{N,K}(s^2 - s^{[2]}_{N,K}) = 2s^{[2]}_{N,K}(s - s_{N,K})^2 + 4s_{N,K}s^{[2]}_{N,K}\ell(u) - 4s_{N,K}s^{[2]}_{N,K}\ell(u_{N,K}) \\ + 2s^{[2]}_{N,K} r_{\mathrm{RB}}(p^{(2)}_{N,K}). \end{aligned}$$

Together, using $\ell^{(6)}(v) = 4\, s_{N,K}\, s^{[2]}_{N,K}\, \ell(v)$, we have

$$s^4 - s^{[4]}_{N,K} = (s^2 - s^{[2]}_{N,K})^2 + 2s^{[2]}_{N,K}(s - s_{N,K})^2 + \left(\ell^{(6)}(u) - \ell^{(6)}(u_{N,K}) + r_{\mathrm{RB}}(p^{(6)}_{N,K})\right)$$

which is estimated, using the same techniques as in Theorems 5.10 and 5.19, by

$$\Delta^{s^4}(\mu,\omega) := (\Delta^{s^2})^2 + 2s^{[2]}_{N,K}(\Delta^s)^2 \,+\, \alpha_{\mathrm{LB}}\Delta\tilde{\Delta}^{(6)} + \delta^f_{\mathrm{KL}}(p^{(6)}_{N,K}) + \delta_{\mathrm{KL}}(p^{(6)}_{N,K}). \quad (5.60)$$

Compared to the straightforward error bound in (5.58), we replaced the term $2s^{[2]}_{N,K}\,\Delta^{s^2}$ by a more precise bound based upon the solution of (5.59) (recall that Δ^{s^2} already contains $(\Delta^s)^2$).

However, differently to the bound for $s^{[3]}_{N,K}$ of the previous section, the second term of (5.60) contains the factor $s^{[2]}_{N,K}$ instead of $s_{N,K}$. Thus, for good results, Δ^s might be required to be very small.

5.6 Inf-Sup Stable Problems

It is possible to maintain the presented error analysis in analogous form if a is not coercive but only inf-sup stable. For the primal problem, we require

$$\beta(\mu,\omega) := \inf_{v\in X}\sup_{w\in X}\frac{a(v,w;\mu,\omega)}{\|v\|_X\|w\|_X} > \beta_0 > 0 \quad \forall \mu,\omega \in \mathcal{P}\times\Omega.$$

Then, existence and uniqueness of the solution of (5.1) are still valid [3]. Analogously, we require the inf-sup stability of the dual problem (5.9). Let $\beta_{\mathrm{LB}}(\mu,\omega)$ be a lower bound of both primal and dual inf-sup constants. The bounds in $\Delta_{\mathrm{KL}}, \tilde{\Delta}^{(i)}_{\mathrm{KL}}, \Delta_{\mathrm{RB}}$ and $\tilde{\Delta}^{(i)}_{\mathrm{RB}}$, $i = 1,\dots,4$, are now redefined replacing α_{LB} by β_{LB}. For the proof of the error bound for the primal solution $u_{N,K}$ in Proposition 5.3, we obtain

$$\begin{aligned}\|e\|_X &\leq \frac{1}{\beta_{\mathrm{LB}}}\sup_{w\in X}\frac{a(e,w)}{\|w\|_X}\\ &\leq \sup_{w\in X}\frac{f(w)-f^K(w)}{\beta_{\mathrm{LB}}\|w\|_X} + \sup_{w\in X}\frac{a^K(u_{N,K},w)-a(u_{N,K},w)}{\beta_{\mathrm{LB}}\|w\|_X} + \sup_{w\in X}\frac{r_{\mathrm{RB}}(w)}{\beta_{\mathrm{LB}}\|w\|_X}\\ &\leq \Delta^f_{\mathrm{KL}} + \Delta_{\mathrm{KL}} + \Delta_{\mathrm{RB}}.\end{aligned}$$

Analogously, the error bound of the dual solution $p^{(1)}_{N,K}$ in Corollary 5.4 can be proven in the inf-sup case which directly implies the error bounds for the further dual solution $p^{(i)}_{N,K}$, $i = \{2,3,4\}$. Since Equation (5.25) remains valid in the inf-sup stable case, replacing again α_{LB} by β_{LB}, it is straightforward that the effectivity bounds for the primal and dual error bounds in Proposition 5.5 and Corollary 5.6 still hold.

Besides the replacement of α_{LB} by β_{LB}, we do not need any further changes to prove the output error bounds for $s_{N,K}$ and $s^{[2]}_{N,K}$ in Theorem 5.8 and Theorem 5.10, respectively. Analogously, the statistical output bounds $\Delta^{\mathbb{M}_1}$, $\Delta^{\mathbb{M}_1^2}$, $\Delta^{\mathbb{M}_2}$ and $\Delta^{\mathbb{V}}$ remain valid. Also, the effectivity bound of Δ^s for symmetric bilinear forms and compliant outputs can directly be adopted.

5.7 Offline-Online Decomposition

In this section, we describe the offline and online procedures and provide corresponding run-time and storage complexities. We start with the description of a

method to evaluate lower bounds for the coercivity constant. For this method, we assume the bilinear form a to be parametrically coercive with respect to the deterministic parameter; i.e., $\theta_q^a(\mu) > 0$ for all $\mu \in \mathcal{P}$ and $\bar{a}_q(v,v) + a_q(v,v;\omega) \geq 0$, $v \in X$, for all $\omega \in \Omega$ and $1 \leq q \leq Q^a$.

5.7.1 Coercivity Lower Bound

From the deterministic case, we know the following methods to determine lower bounds $\alpha_{\mathrm{LB}}(\mu,\omega)$ for $\alpha(\mu,\omega)$: the min-θ approach [73] and the successive constraint method (SCM) [57]. The latter approach is less restrictive and could be directly applied to the stochastic parameter case and also to inf-sup stable problems. However, it requires much more effort, online as well as offline. The min-θ approach requires the bilinear form a to be parametrically coercive with respect to the deterministic *and* the stochastic parameters. Therefore, the extension of the method to our case is not possible. We would require $\xi_{q,k}(\omega)$ to be positive.

To partially maintain the advantage of the min-θ approach, we propose a combination of both methods. We fix some parameter $\bar{\mu} \in \mathcal{P}$ and get the inequality

$$\alpha(\mu,\omega) = \inf_{v\in X} \frac{a(v,v;\mu,\omega)}{\|v\|_X^2} \geq \inf_{v\in X} \frac{a(v,v;\mu,\omega)}{a(v,v;\bar{\mu},\omega)} \cdot \inf_{v\in X} \frac{a(v,v;\bar{\mu},\omega)}{\|v\|_X^2}. \tag{5.61}$$

If a is parametrically coercive, we apply the min-θ approach on the first term. Precisely, for $\theta_{\min}(\mu) := \min_{1\leq q\leq Q^a}\{\theta_q^a(\mu)/\theta_q^a(\bar{\mu})\}$, we obtain ω-independent lower bounds

$$\frac{a(v,v;\mu,\omega)}{a(v,v;\bar{\mu},\omega)} \quad \geq \quad \theta_{\min}(\mu) \quad \forall v \in X,\ \forall(\mu,\omega) \in \mathcal{P}\times\Omega$$

analogously to [73]. For the approximation of the second term, we first apply the SCM to the truncated form and obtain μ-independent lower bounds

$$\frac{a^K(v,v;\bar{\mu},\omega)}{\|v\|_X^2} \quad \geq \quad \alpha_{\mathrm{SCM}}^K(\omega) \quad \forall v \in X,\ \forall \omega \in \Omega.$$

To take the truncation error into account, we consider the parameter independent truncation error

$$\Delta_{\mathrm{KL}}^{\alpha} := \sup_{v\in X} \left(\sum_{q=1}^{Q^a} \theta_q^a(\bar{\mu}) \sum_{k=K+1}^{K_{\max}} \xi_{UB} \frac{a_{q,k}(v,v)}{\|v\|_X^2} \right) \tag{5.62}$$

such that $-\Delta_{\mathrm{KL}}^{\alpha}\|v\|_X^2 \leq a(v,v;\bar{\mu},\omega) - a^K(v,v;\bar{\mu},\omega)$. Hence, we define $\alpha_{\mathrm{SCM}}(\omega) := \alpha_{\mathrm{SCM}}^K(\omega) - \Delta_{\mathrm{KL}}^{\alpha}$ and obtain the coercivity lower bound $\alpha_{LB}(\mu,\omega) := \theta_{\min}(\mu) \cdot \alpha_{\mathrm{SCM}}(\omega)$. It is essential that K be large enough to obtain a positive α_{SCM}.

Both $\alpha_{\mathrm{SCM}}(\omega)$ and $\theta_{\min}(\mu)$ can be evaluated independently. Therefore, it might be useful to store α_{SCM} for many random realizations and reuse these values in combination with different μ. This is possible if the same random realizations can be used for several parameters. Then $\alpha_{\mathrm{LB}}(\mu,\omega)$ can be evaluated very quickly in the online stage.

5.7.2 Assembling of the Error Bounds

We exemplarily show the construction of the error bound of the primal solution $\Delta(\mu,\omega) = \Delta_{\mathrm{RB}}(\mu,\omega) + \Delta_{\mathrm{KL}}(\mu,\omega) + \Delta_{\mathrm{KL}}^f(\mu,\omega)$ from Proposition 5.3. Let $\zeta_1,\dots,\zeta_N$ be the basis of the primal reduced space X_N. The error bounds can be evaluated using the Riesz representatives $\mathcal{A}_{q,k,n}$, $q = 1,\dots,Q^a$, $k = 1,\dots,K_{\max}^a$, $n = 1,\dots,N$, and $\mathcal{F}_{q,k}$, $q = 1,\dots,Q^f$, $k = 1,\dots,K_{\max}^f$, given by

$$\big(\mathcal{A}_{q,k,n}, v\big)_X = a_{q,k}(\zeta_n, v), \qquad \big(\mathcal{F}_{q,k}, v\big)_X = f_{q,k}(v), \qquad v \in X.$$

These quantities are independent of the parameters and can be evaluated in the offline stage. However, we only store the pairwise inner products

$$\big(\mathcal{A}_{q,k,n}, \mathcal{A}_{q',k',n'}\big)_X, \quad \big(\mathcal{A}_{q,k,n}, \mathcal{F}_{q',k'}\big)_X, \quad \big(\mathcal{F}_{q,k}, \mathcal{F}_{q',k'}\big)_X. \tag{5.63}$$

We start with the construction of the RB-part of the error bound, $\Delta_{\mathrm{RB}}(\mu,\omega) = \frac{1}{\alpha_{\mathrm{LB}}(\mu,\omega)}\|r_{\mathrm{RB}}(\cdot;\mu,\omega)\|_{X'}$, which implies the evaluation of the dual norm of the residual. Let $\mathcal{R}(\mu,\omega)$ be the Riesz representative of $r_{\mathrm{RB}}(\cdot;\mu,\omega)$,

$$\big(\mathcal{R}(\mu,\omega), v\big)_X = r_{\mathrm{RB}}(v;\mu,\omega) = f^K(v;\mu,\omega) - a^K(u_{N,K}(\mu,\omega), v;\mu,\omega), \quad v \in X.$$

For $u_{N,K}(\mu,\omega) = \sum_{n=1}^N \bar{u}_n(\mu,\omega)\zeta_n$, we can evaluate $\mathcal{R}(\mu,\omega)$ as

$$\mathcal{R}(\mu,\omega) = \sum_{q=1}^{Q^f}\sum_{k=1}^{K_q^f} \theta_q^f(\mu)\xi_{q,k}^f(\omega)\mathcal{F}_{q,k} - \sum_{q=1}^{Q^a}\sum_{k=1}^{K_q^a}\sum_{n=1}^{N} \theta_q^a(\mu)\xi_{q,k}^a(\omega)\bar{u}_n(\mu,\omega)\mathcal{A}_{q,k,n}.$$

The dual norm of the residual is now given by $\|r_{\mathrm{RB}}(\cdot;\mu,\omega)\|_{X'} = \|\mathcal{R}(\mu,\omega)\|_X = \sqrt{(\mathcal{R}(\mu,\omega),\mathcal{R}(\mu,\omega))_X}$ which can now be evaluated using the stored inner products of (5.63) in $\mathcal{O}((Q^f K^f + Q^a K^a N)^2)$, independently of $\mathcal{N}$.

As mentioned in Remark 5.1, the definition of δ_{KL} in (5.20) includes absolute values and we can not define $\|\delta_{\mathrm{KL}}(\cdot;\mu,\omega)\|_{X'}$. However, it is still possible to efficiently estimate the truncation error for the bilinear form, using the inner products of Riesz representatives. We obtain

$$\begin{aligned}
\sup_{v\in X}\frac{\delta_{\mathrm{KL}}(v;\mu,\omega)}{\|v\|_X} &= \sup_{v\in X}\sum_{q=1}^{Q^a}\sum_{k=K_q^a+1}^{K_{\max}^a}\left|\theta_q^a(\mu)\xi_{\mathrm{UB}}\right|\left|\sum_{n=1}^{N}\bar{u}_n(\mu,\omega)\frac{(\mathcal{A}_{q,k,n},v)_X}{\|v\|_X}\right| \\
&\leq \sum_{q=1}^{Q^a}\sum_{k=K_q^a+1}^{K_{\max}^a}\left|\theta_q^a(\mu)\xi_{\mathrm{UB}}\right|\sup_{v\in X}\left|\sum_{n=1}^{N}\bar{u}_n(\mu,\omega)\frac{(\mathcal{A}_{q,k,n},v)_X}{\|v\|_X}\right| \\
&= \sum_{q=1}^{Q^a}\sum_{k=K_q^a+1}^{K_{\max}^a}\left|\theta_q^a(\mu)\xi_{\mathrm{UB}}\right|\left\|\sum_{n=1}^{N}\bar{u}_n(\mu,\omega)\mathcal{A}_{q,k,n}\right\|_X, \qquad (5.64)
\end{aligned}$$

which can be evaluated in $\mathcal{O}(Q^a(K_{\max}^a-K^a)N^2)$, where the different values K_q^a have been replaced by some K^a for notational reaasons. It is clear that the presented bound does not directly coincide with the definition of Δ_{KL} in (5.21) since an additional estimate has been performed in the second line. However, the direct evaluation of Δ_{KL} is difficult. Consider the bound

$$\delta_{\mathrm{KL}}(v;\mu,\omega) = \sum_{q=1}^{Q^a}\sum_{k=K_q^a+1}^{K_{\max}^a}\underbrace{\sigma_{q,k}(\mu,\omega)}_{\pm 1}\,\theta_q^a(\mu)\xi_{\mathrm{UB}}^a\sum_{n=1}^{N}\bar{u}_n(\mu,\omega)\,(\mathcal{A}_{q,k,n},v)_X\,,$$

where, compared to the definition of $\delta_{\mathrm{KL}}(v;\mu,\omega)$ in (5.20), the absolute values have been replaced by an appropriate sign function $\sigma_{q,k}(\mu,\omega)\in\{-1,+1\}$. Then, the bound $\Delta_{\mathrm{KL}}(\mu,\omega)$ can be written as

$$\sup_{v\in X}\frac{\delta_{\mathrm{KL}}(v;\mu,\omega)}{\|v\|_X} = \left\|\sum_{q=1}^{Q^a}\sum_{k=K_q^a+1}^{K_{\max}^a}\sigma_{q,k}(\mu,\omega)\,\theta_q^a(\mu)\xi_{\mathrm{UB}}^a\sum_{n=1}^{N}\bar{u}_n(\mu,\omega)\mathcal{A}_{q,k,n}\right\|_X.$$

For given values of $\sigma_{q,k}(\mu,\omega)$, $q=1,\dots,Q^a$, $k=K_q^a,\dots,K_{\max}^a$, this bound can be efficiently evaluated in $\mathcal{O}\big((Q^a(K_{\max}^a-K^a)N)^2\big)$. However, to find the correct bound, it would be necessary to test all possibilities of $\sigma_{q,k}(\mu,\omega)$, i.e., it would be necessary to evaluate the bound $2^{Q^a(K_{\max}^a-K^a)}$ times, leading to a complexity of $\mathcal{O}\big((Q^a(K_{\max}^a-K^a)N)^2\cdot 2^{Q^a(K_{\max}^a-K^a)}\big)$. Even though the evaluation is independent of $\mathcal{N}$, we prefer the much cheaper bound in (5.64).

As mentioned in Remark 5.2, it is also possible to use random variables instead of the upper bounds ξ_{UB}. Then, the evaluation of Δ_{KL} is straightforward,

$$\Delta_{\mathrm{KL}}(\mu,\omega) \;=\; \left\| \sum_{q=1}^{Q^a} \sum_{k=K_q^a+1}^{K_{\max}^a} \theta_q^a(\mu)\xi_{q,k}^a \sum_{n=1}^{N} \bar{u}_n(\mu,\omega)\mathcal{A}_{q,k,n} \right\|_X ,$$

which is of complexity $\mathcal{O}\big((Q^a(K_{\max}^a - K^a)N)^2\big)$.

5.7.3 Online Procedure

We first summarize the run-time complexity to solve a reduced system and evaluate the corresponding outputs and bounds. Assuming the availability of all necessary terms, the complexity is the same for all primal and dual problems. For notational compactness, we do not distinguish between Q^b, K^b, $K_{\max}^b$ for $b \in \{a, f, \ell\}$, but just use Q, K, and $K_{\max}$, respectively. In the same way, we just use N instead of N, $\tilde{N}^{(1)}$, $\tilde{N}^{(2)}$, and $\tilde{N}^{(3)}$.

The complexity to assemble a reduced system for a new parameter pair reads $\mathcal{O}(QKN^2)$; the solution is then obtained in $\mathcal{O}(N^3)$ operations. For the output evaluation, we need to assemble some additional matrices and vectors — again with complexity $\mathcal{O}(QKN^2)$ — to evaluate the residuals. The actual evaluation is then of complexity $\mathcal{O}(N^2)$. For the error bounds, we first evaluate the coercivity lower bound. The complexity depends on the chosen method, optimally $\mathcal{O}(Q)$. For the Δ_{KL}- and Δ_{RB}-error bounds, we use the previously evaluated and stored Riesz representative inner products and compute the bounds in $\mathcal{O}(Q(K_{\max}-K)N^2)$ and $\mathcal{O}(Q^2K^2N^2)$, respectively. For the δ_{KL}-error bounds, we just need $\mathcal{O}(Q(K_{\max}-K))$ matrix-vector and vector-vector multiplications; the total complexity is therefore $\mathcal{O}(Q(K_{\max}-K)N^2)$.

Suppose we use M random realizations to evaluate the Monte Carlo estimates for any given deterministic parameter; the overall run-time complexity for the computation of the statistical outputs is $\mathcal{O}(M(N^3+(Q^2K^2+Q(K_{\max}-K))N^2))$, including the complexity for the evaluation of the error bounds.

If we are interested in both second moment and variance, the online procedure works as follows. We solve the primal and first dual problems for M realizations and some fixed μ. For all realizations, we store $s_{N,K}$, which is later used to solve the second and third dual problem (5.35) and (5.45). For the quadratic output

evaluations, we additionally store $r_{\mathrm{RB}}(p^{(1)}_{N,K})$ as well as the primal solutions $u_{N,K}$ needed for the computation of the respective last terms in (5.38) and (5.47). Furthermore, for the corresponding error bounds (5.39) and (5.50), we store Δ and Δ^s. Hence, the overall storage complexity is $\mathcal{O}((N+4)M)$.

Using the same reduced space for the second and third dual problems (5.35) and (5.45), it is possible to evaluate all statistical outputs with storage complexity $\mathcal{O}(M)$. For some fixed μ, the basic concept is to solve (5.51) for each random realization at the same time as the primal and first dual problems (5.15) and (5.16). It is clear that the evaluation of $s^{[2]}_{N,K}$ in (5.38) and the second moment $\mathbb{M}_{2,NK} = \mathbb{E}[s^{[2]}_{N,K}]$ as well as its error bounds Δ^{s^2} from (5.39) and $\Delta^{\mathbb{M}_2} = \mathbb{E}[\Delta^{s^2}]$ can be obtained with storage complexity $\mathcal{O}(1)$. As a consequence of the use of (5.51), we have $\mathbb{E}[r_{\mathrm{RB}}(p^{(3)}_{N,K})] = \mathbb{M}_{1,NK}\mathbb{E}[r_{\mathrm{RB}}(p^{(4)}_{N,K})]$, and the evaluation of $\mathbb{M}^{[2]}_{1,NK}$ in (5.47) is of storage complexity $\mathcal{O}(1)$, too, and hence the evaluation of $\mathbb{V}_{NK} = \mathbb{M}_{2,NK} - \mathbb{M}^{[2]}_{1,NK}$. Analogously, $\mathbb{E}[\alpha_{\mathrm{LB}}\Delta\tilde{\Delta}^{(3)}] = |\mathbb{M}_{1,NK}| \cdot \mathbb{E}[\alpha_{\mathrm{LB}}\Delta\tilde{\Delta}^{(4)}]$, and hence the storage complexity to evaluate $\Delta^{\mathbb{M}_1^2}$ in (5.48) is constant. Therefore, using only the less precise variance error bound $|\mathbb{V} - \mathbb{V}_{NK}| \leq \Delta^{\mathbb{M}_2} + \Delta^{\mathbb{M}_1^2}$, it would even be possible to solve all problems with storage complexity $\mathcal{O}(1)$. However, for the variance error bound presented in (5.50), we additionally store $s_{N,K}$ and $\alpha_{\mathrm{LB}}\Delta\tilde{\Delta}^{(4)}$ for each realization with storage complexity $\mathcal{O}(M)$ to enable the evaluation of $\mathbb{E}[\alpha_{\mathrm{LB}}\Delta\tilde{\Delta}^{(2-3)}] = \mathbb{E}\big[|s_{N,K} - \mathbb{M}_{1,NK}| \cdot \alpha_{\mathrm{LB}}\Delta\tilde{\Delta}^{(4)}\big]$.

5.7.4 Greedy Basis Selection

To generate the bases of the reduced spaces, we perform a Greedy algorithm as it is well known in the RB context [98, 73]. For a training parameter set $\Xi_{\mathrm{train}} \subset \mathcal{P} \times \Omega$ and some initial basis, given by an arbitrary single snapshot, we solve the reduced primal and dual problems (5.15), (5.16), (5.35), and (5.45) and evaluate the error bounds for the desired outputs. For each problem, we select the parameter pair for which the RB error part of the desired output error bound is maximal and add the corresponding solution of the unreduced problem to the respective basis. We iterate the procedure until the error bounds fall below an intended tolerance for all training parameters.

To generate Ξ_{train}, we use the random variables of the KL expansion (5.11) for the actual sampling of Ω. Hence, we have to sample a high-dimensional space for

the Greedy algorithm. Since the "importance" of KL random variables ξ_k decrease in k, the most obvious sampling procedure would be a Monte Carlo approach. Alternatively to the sampling, it is also possible to use optimization algorithms to find a good random sample for the basis extension. For more information, we refer to [91].

Next, we are going to describe how to specify the KL truncation, precisely the numbers of affine terms used for the approximation, K^b, $b \in \{a, f, \ell\}$, and the number of terms used to estimate the truncation error, $K^b_{\max}$, $b \in \{a, f, \ell\}$. We integrate the specification into the Greedy algorithm. For different truncation lengths and very large $K_{\max}$ values, we solve the reduced system and evaluate the KL error bounds for all training parameters. K^b, $b \in \{a, f, \ell\}$, are chosen as the minimal numbers such that the KL error bounds do not exceed a given tolerance, respectively. This tolerance should be rather small compared to the allowed output errors. Similarly, we make $K^b_{\max}$, $b \in \{a, f, \ell\}$, as small as possible such that we underestimate the KL error bounds only negligibly. Since the KL truncation errors do not depend on the dimension of the RB spaces, K^b and $K^b_{\max}$, $b \in \{a, f, \ell\}$, are likely to be appropriate for all reduced spaces and can be fixed for all further computations. However, it would also be possible to make further adjustments during the Greedy algorithm.

Suppose that Ξ_{train} consists of n_{train} deterministic parameters and M_{train} random realizations for each of the parameters. Then, the Greedy complexity is $\mathcal{O}(Nn_{\text{train}})$ times the online complexity to find the "optimal" parameters in each iteration, i.e., $\mathcal{O}(Nn_{\text{train}}M_{\text{train}}(N^3 + (Q^2K^2 + Q(K_{\max} - K))N^2))$, plus $\mathcal{O}(QK_{\max}N\mathcal{N})$ to solve for the corresponding detailed solutions. Furthermore, the construction of the reduced system matrices and vectors is of complexity $\mathcal{O}(QK_{\max}N^2\mathcal{N})$ and the evaluation of the used Riesz representatives and the pairwise inner products is of complexity $\mathcal{O}(Q^2K^2_{\max}N^2\mathcal{N})$.

We store these RB system matrices and vectors as well as the Riesz representative inner products that are used to construct the Δ_{KL}- and Δ_{RB}-error bounds. Hence the total storage complexity is $\mathcal{O}((Q^2K^2 + Q(K_{\max} - K))N^2)$.

Especially for stochastic problems, it is not clear if the parameter range is sufficiently covered by the random training set Ξ_{train}. However, since we evaluate a posteriori error bounds, we detect such cases in the online stage and could still

extend both Ξ_{train} and the basis.

5.8 Numerical Realization and Experiments

In this section, an example of a two-dimensional porous medium is chosen to illustrate the different aspects of the proposed methods. We consider heat transfer in a wet sandstone with porosity modeled by a random function $\kappa(x;\omega)$ that represents the rate of pore space within some control volume. We construct κ generating $\mathcal{N}$ standard normally distributed random variables and applying a Gaussian smoothing filter of the form $\exp(-\|x-y\|^2/\sigma^2)$, where $\sigma = 1/5$. Additionally, we perform a Wiener process–like algorithm on the $\mathcal{N}$ new variables. Hence, $\kappa(\cdot;\omega)$ is (at least) almost surely everywhere continuous, and hence $\kappa(\cdot;\omega) \in L_2(D)$. Furthermore, our model depends on a deterministic parameter $\mu \in \mathcal{P} = [0.01, 1]$ that denotes the global water saturation in the pores. Hence, the proportion of air in the pores is given by $(1-\mu)$. Let $c_{\text{s}} = 2.40$ be the heat conductivity constant of pure (theoretically imporous) sandstone and let $c_{\text{w}} = 0.60$, $c_{\text{a}} = 0.03$ be the respective heat conductivity constants of water and air. With this notation, the total heat conductivity of a wet sandstone is assumed to be

$$\begin{aligned} c(x;\mu,\omega) &= c_{\text{s}} \cdot (1-\kappa(x;\omega)) \;+\; (\mu c_{\text{w}} + (1-\mu)c_{\text{a}})\kappa(x;\omega) \\ &= c_{\text{s}} \;+\; (-c_{\text{s}} + \mu c_{\text{w}} + (1-\mu)c_{\text{a}})\kappa(x;\omega). \end{aligned} \tag{5.65}$$

We consider a domain $D = (0,1)^2 \subset \mathbb{R}^2$ and impose homogeneous Dirichlet boundary conditions on some boundary part Γ_{D} and nonhomogeneous Neumann boundary conditions on the opposite "output" boundary Γ_{out}, where the right-hand side of the boundary condition is a random function $g(\omega) : [0,1] \to \mathbb{R}$, stochastically independent of κ, representing some random loss of heat at the output boundary and modeled by a smoothed Wiener bridge process. On the other boundaries, we impose homogeneous Neumann conditions, representing isolated parts of the sandstone. For a given $\mu \in \mathcal{P}$ and some random realization of κ, we are interested in the average temperature at the "output" boundary Γ_{out}, denoted by $s(\mu,\omega)$.

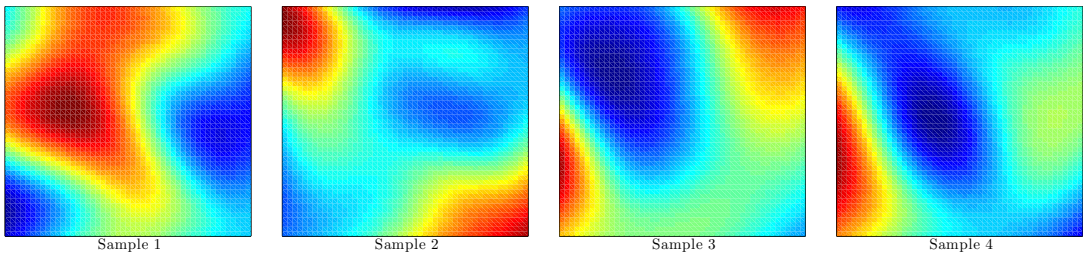


Figure 5.1: Four random realizations of κ

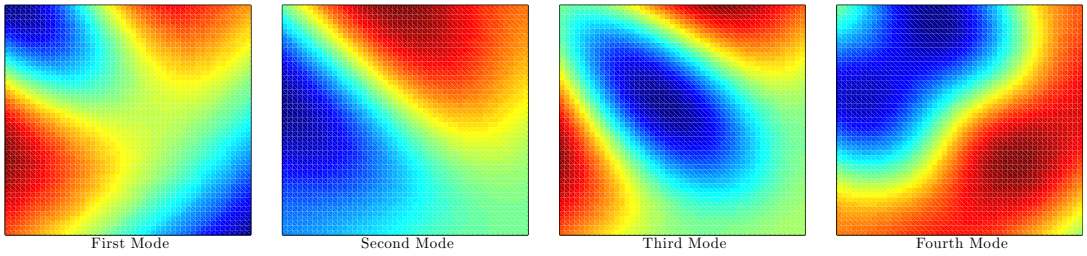


Figure 5.2: First four modes of $\tilde{\kappa}$

Now, the PDE reads as follows: for given $(\mu, \omega) \in \mathcal{M}$, find $u(\mu, \omega)$ such that

$$\begin{cases} -\nabla \cdot \big(c(\mu,\omega)\ \nabla u(\mu,\omega)\big) &= 0 \quad \text{in } D, \\ u(\mu,\omega) &= 0 \quad \text{on } \Gamma_{\mathrm{D}}, \\ n \cdot \big(c(\mu,\omega)\ \nabla u(\mu,\omega)\big) &= 0 \quad \text{on } \Gamma_{\mathrm{N}}, \\ n \cdot \big(c(\mu,\omega)\ \nabla u(\mu,\omega)\big) &= g(\omega) \quad \text{on } \Gamma_{\mathrm{out}}. \end{cases} \tag{5.66}$$

In the weak form, we compute $u(\mu, \omega) \in X$ such that $a(u(\mu,\omega), v; \mu, \omega) = f(v; \omega)$ for all $v \in X$, where $a(w, v; \mu, \omega) = \int_D c(\mu, \omega) \nabla w \cdot \nabla v$ and $f(v; \omega) = \int_{\Gamma_{\mathrm{out}}} g(\omega) v$. For the functional $\ell(v) = \int_{\Gamma_{\mathrm{out}}} v$, the noncompliant output is given by

$$s(\mu, \omega) := \ell(u(\mu, \omega)) = \int_{\Gamma_{\mathrm{out}}} u(\mu, \omega).$$

The affine decomposition of the bilinear form a in μ is straightforward. Let $\bar{\kappa}(x)$ denote the mean of $\kappa(x; \cdot)$ and $\tilde{\kappa}(x; \omega) := \kappa(x; \omega) - \bar{\kappa}(x)$ its stochastic part with zero mean. We define $\theta_1(\mu) :\equiv c_s$ and $\theta_2(\mu) := -c_{\mathrm{s}} + \mu c_{\mathrm{w}} + (1 - \mu) c_{\mathrm{a}}$. Then, using the notation of (5.2), $\bar{a}_1(w, v) = \int_D \nabla w \cdot \nabla v$, whereas $a_1(w, v; \omega) \equiv 0$ vanishes. For the second affine term, we have $\bar{a}_2(w, v) = \int_D \bar{\kappa} \nabla w \cdot \nabla v$ and $a_2(w, v; \omega) = \int_D \tilde{\kappa}(\omega) \nabla w \cdot \nabla v$. In the same way, we denote by $\bar{g}(x)$ the mean

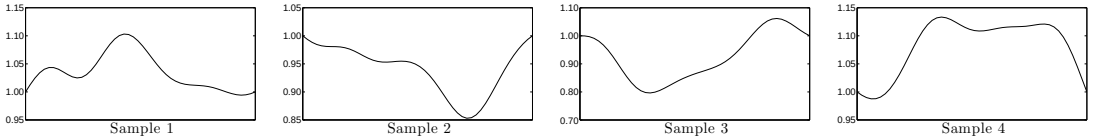


Figure 5.3: Four random realizations of g

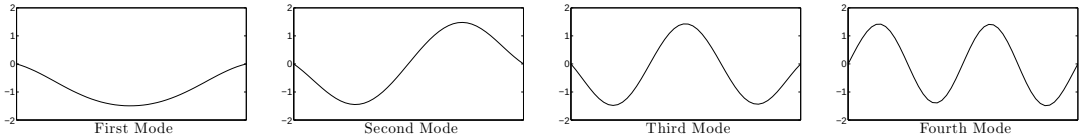
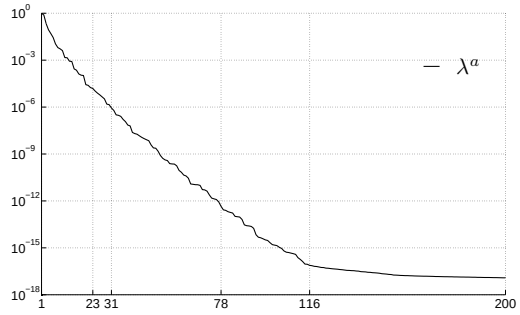


Figure 5.4: First four modes of $\tilde{g}$

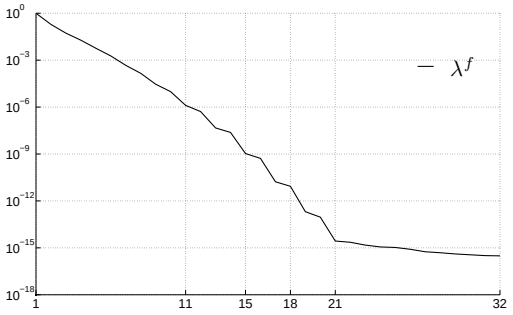
of $g(x;\cdot)$ and by $\tilde{g}(x;\omega)$ its stochastic part and define $\bar{f}_1(v) = \int_{\Gamma_{\text{out}}} \bar{g}v$ as well as $f_1(v;\omega) = \int_{\Gamma_{\text{out}}} \tilde{g}(\omega)v$, where $\theta_1^f = 1$. Using KL expansions of $\tilde{\kappa}$ and $\tilde{g}$, we directly obtain affine decompositions of a_2 and f_1 in ω, respectively. Since ℓ is independent of μ and ω, we put all forms into the framework of (5.11) with $Q^a = 2$, $Q^f = 1$, and $Q^\ell = 1$, where $\xi_{1,k}^a(\omega) = 0$ for all $k \geq 1$, and therefore $K_1^a = 0$ in (5.12).

Figure 5.1 shows four random realizations of κ and Figure 5.2 the first four eigenmodes of the KL expansion of $\tilde{\kappa}$. Its eigenvalues are provided in Figure 5.5(a). The expectation of κ is supposed to be constant in space, $\bar{\kappa}(x) \equiv 0.33$. We assume the random coefficients $\xi_{2,k}^a(\omega)$ to be standard normally distributed. Since $\kappa(x;\omega)$ is restricted to $[0,1]$, whereas $\xi_{2,k}^a(\omega)$ are unbounded, we dismiss realizations that do not satisfy the physical constraints. However, this can be done easily online, and this happens with a probability of less than $2.5 \cdot 10^{-6}$ in our model. Then, $c(x;\mu,\omega) > \mu c_{\text{w}} + (1-\mu)c_{\text{a}} > 0.0357 > 0$, and the PDE is uniformly coercive. Figure 5.3 shows four random realizations of g and Figure 5.4 the first four eigenmodes of the KL expansion of $\tilde{g}$. Its eigenvalues are provided in Figure 5.5(b). The expectation of g is constant in space, $\bar{g}(x) = 1$. The random coefficients $\xi_{1,k}^f(\omega)$ are assumed to be standard normally distributed. Here, we do not restrict g to a certain interval. However, negative values of g are very unlikely.

For the detailed approximations, we choose a finite element (FE) space X with linear Lagrange elements and $\mathcal{N} = 4841$ degrees of freedom. Furthermore, we use $K_{\text{detail}}^a = 78$ and $K_{\text{detail}}^f = 18$ terms to assemble the detailed forms a and f, respectively. These numbers of terms are already precise enough compared to the FE error.



(a) Eigenvalues of the KL expansion of $\tilde{\kappa}$ and KL truncation values $K^a = 23$, $K^a_{\max} = 31$, and $K^a_{\text{detail}} = 78$.



(b) Eigenvalues of the KL expansion of $\tilde{g}$ and KL truncation values $K^f = 11$, $K^f_{\max} = 15$, and $K^f_{\text{detail}} = 18$.

Figure 5.5: Eigenvalues and truncation values of the Karhunen–Loève expansions.

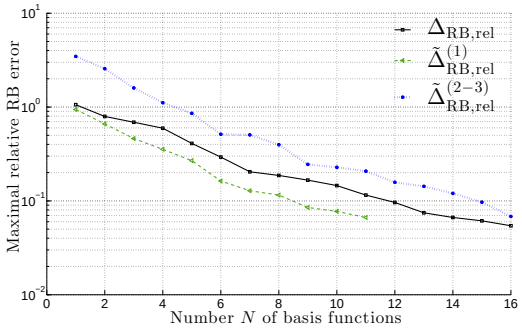
The bilinear form a with the affine decomposition introduced before is not parametrically coercive since $\theta^a_2(\mu) < 0$. However, since $\bar{a}_2(\cdot) = 0.33 \cdot \bar{a}_1(\cdot)$, resorting the affine terms to

$$a(\cdot;\mu,\omega) = \theta^a_1(\mu)\big(\bar{a}_1(\cdot) - \bar{a}_2(\cdot) - a_2(\cdot;\omega)\big) + (\theta^a_1(\mu) + \theta^a_2(\mu))\big(\bar{a}_2(\cdot) + a_2(\cdot;\omega)\big)$$

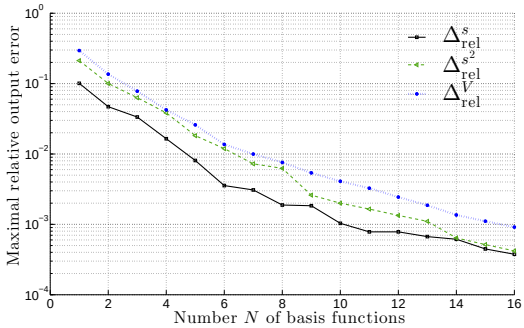
leads to a decomposition that fulfills the requirements of the method proposed in Section 5.7.1 to evaluate coercivity lower bounds. That is, we first create several random samples of the sandstone in the online stage and store the respective α_{SCM}. Then, for all water saturations $\mu \in \mathcal{P}$, we use the same samples and can reuse α_{SCM}.

Using the initial basis of the Greedy algorithm, we specify the KL truncation as described in Section 5.7.4. For a relative error tolerance $\varepsilon_{\text{tol}} = 10^{-3}$, we choose K^a and K^f such that the respective truncation errors, especially the δ_{KL}-parts, do not exceed $0.1\varepsilon_{\text{tol}}$. This leads to $K^a = 23$, $K^a_{\max} = 31$, $K^f = 11$, and $K^f_{\max} = 15$, as marked in Figures 5.5(a) and 5.5(b). For the KL error bounds, we use the upper bound $\xi_{\text{UB}} := 5.2$ such that $|\xi_{q,k}| > \xi_{\text{UB}}$ with a probability of less than $2.5 \cdot 10^{-7}$.

As mentioned, we use the same space for the second and third dual spaces, $\tilde{X}^{(2)}_N = \tilde{X}^{(3)}_N$, and solve only the additional dual problem (5.51). Figure 5.6(a) shows the decay of the maximal relative error bounds of the primal and dual solutions u and $p^{(1)}$, and of the difference of the additional dual solutions $p^{(2)} - p^{(3)}$ that is used for the construction of the variance. In Figure 5.6(b) we provide the



(a) Relative error decay of primal solution u and dual solution $p^{(1)}$, and of $p^{(2)} - p^{(3)}$.



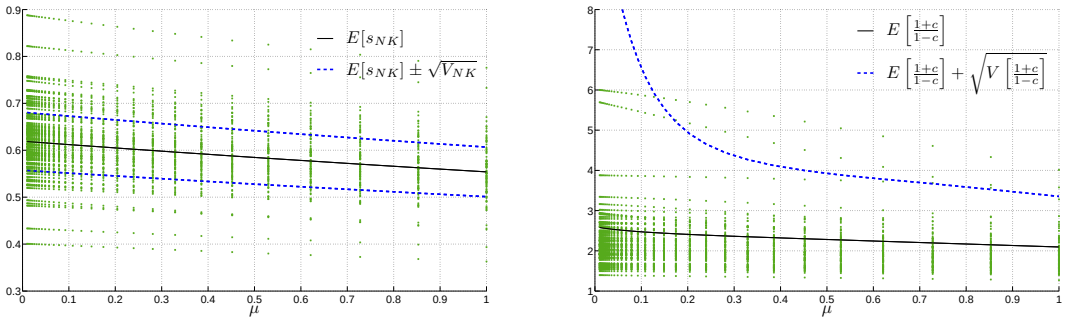
(b) Relative error decay of outputs s, s^2, and $\mathbb{V}$ without δ_{KL}-contributions, respectively.

Figure 5.6: Greedy error decay.

decay of the error bounds of the desired outputs. We omit the δ_{KL}-parts since they do not decrease with the number of basis functions and could therefore have a negative effect on the basis selection procedure. It turns out that $(N, \tilde{N}^{(1)}, \tilde{N}^{(2)}) = (16, 11, 16)$ is sufficient for relative error below the tolerance for all outputs.

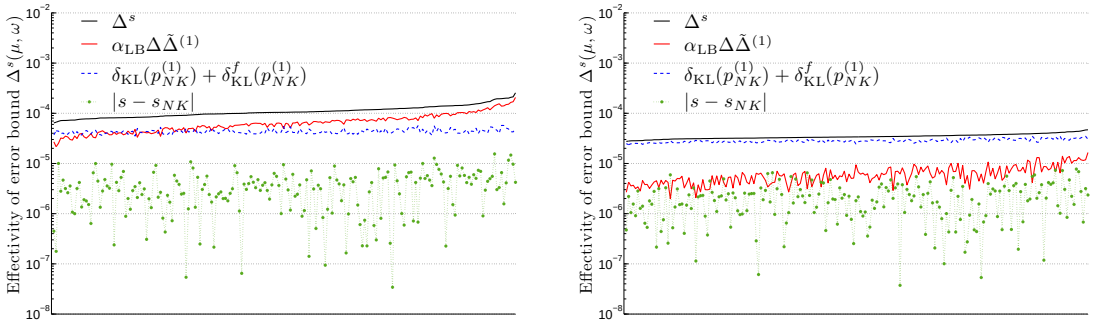
On our reference system, a 3.06 GHz Intel Core 2 Duo processor, 4 GB RAM, we used Comsol 3.5.0.608 (3.5a) to construct and store the FE system components and MATLAB 7.8.0 (R2009a) to implement and run both the detailed and reduced models. For the solutions, we used the MATLAB *mldivide* function which automatically adapts to the structure of the system, e.g., sparsity patterns. Solving the detailed problem with $\mathcal{N} = 4841$ degrees of freedom, we needed about 0.211 seconds per sample on average, whereas the reduced problem could be solved in about 0.00603 seconds per sample, including the solution of all primal and dual problems and the evaluation of all outputs and error bounds. Hence, we gain a speedup by a factor of about 35. To show that the number of reduced basis functions is independent of the degrees of freedom of the detailed problem, we started another Greedy algorithm using $\mathcal{N} = 19121$. Again, the error bounds fell below the desired error tolerance for $(N, \tilde{N}^{(1)}, \tilde{N}^{(2)}) = (16, 11, 16)$. On average, the computation of the larger detailed problem needed about 0.837 seconds per sample. Since the size of the reduced system did not change, we gain a speedup by a factor greater than 138.

The result of the reduced computation is shown in Figure 5.7(a). For each



(a) output $s_{N,K}$ (b) effectivity bound factor $\frac{1+c}{1-c}$

Figure 5.7: Means of output $s_{N,K}$ and of effectivity bound factor $\frac{1+c}{1-c}$, their standard deviations, and 100 random samples for a test set of 30 logarithmically distributed values of μ, respectively



(a) 200 random samples for $\mu = 0.01$ (b) 200 random samples for $\mu = 1.00$

Figure 5.8: Error bound Δ^s, split into its δ_{KL} and Δ parts, and actual output error for 200 random samples and two values of μ.

parameter of a test set of 30 logarithmically distributed values of μ, we evaluated the output s, its mean, and the variance $\mathbb{V}$ using 10000 random samples. In Figure 5.7(a), we plotted the mean and standard deviations of $s_{N,K}$ as well as 100 random samples for each parameter of the test set.

In Figure 5.8, we show the errors and error bounds for the output s for two values of μ and 200 random samples each. The samples are sorted according to Δ^s. We see that the error bound is effective. The average effectivity $\Delta^s/|s - s_{N,K}|$ is about 200. We furthermore separated the error bound into its different parts. One can see that the δ_{KL} part hardly varies since it is not directly dependent on the current random realization. While for $\mu = 0.01$, $\alpha_{\mathrm{LB}}\Delta\tilde{\Delta}^{(1)}$ contributes most to Δ^s, the δ_{KL} parts contribute most for $\mu = 1.00$. Hence, adaptive choices of K^a and K^f could improve the error bounds and reduce the run-time and will be a part of future work.

In Figure 5.9 we compare our variance evaluation method and corresponding error bounds with two other evaluation procedures based upon the use of the sample variance $\mathbb{E}[(s_{N,K})^2] - (\mathbb{E}_{NK})^2$. For the "direct" bound, we follow (5.33) and replace s by $(s - s_{N,K}) + s_{N,K}$, which can be estimated by $\Delta^s + |s_{N,K}|$. Analogously, we obtain $|\mathbb{M}_1| \leq \Delta^{\mathbb{M}_1} + |\mathbb{M}_{1,NK}|$, which leads us to the "direct" variance error bound

$$|\mathbb{V} - \mathbb{V}_{NK}| \quad \leq \quad \mathbb{E}[\Delta^s(\Delta^s + 2|s_{N,K}|)] + \Delta^{\mathbb{M}_1}(\Delta^{\mathbb{M}_1} + 2|\mathbb{M}_{1,NK}|).$$

For the "sophisticated" bound, we refer the reader to Appendix A or [12]. We see that our variance approximations and the corresponding error estimates in fact give sharper bounds. The direct error bound is about 160 times larger; the sophisticated error bound still is about 12 times larger on average.

Compared to the deterministic problems, the effectivity bound $\eta(\mu, \omega)$ from (5.26) contains an additional factor of the form $(1 + c)/(1 - c)$, where c is given by (5.27). Figure 5.7(b) shows the average factor, its standard deviation, and 100 random samples for each parameter of the test set. We can see that the additional factor takes an average value of about 2.4. Hence, compared to the deterministic case, the effectivity upper bound increases only moderately in most cases. However, there are cases in which $c(\mu, \omega) \approx 1$ and the effectivity bound becomes inappropriate or, for $c(\mu, \omega) > 1$, even nonexistent. This can be avoided using larger K.

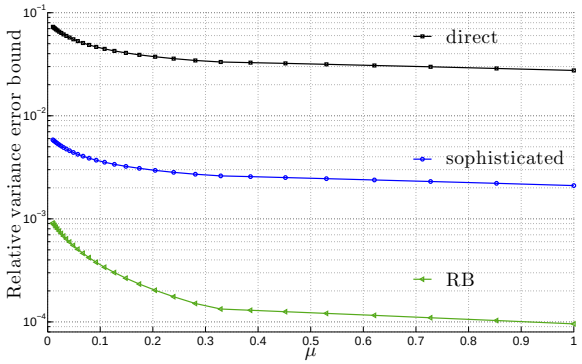


Figure 5.9: Different relative error bounds for variance $\mathbb{V}(\mu)$.

5.9 Conclusions and Outlook

We presented a general RB framework for linear coercive PPDEs with stochastic influences. Efficient a posteriori error bounds have been developed for the state and output functionals, also dealing with additional KL-truncation errors. We furthermore introduced a new error analysis for special quadratic and statistical outputs such as second moment and variance using additional nonstandard dual problems. We showed that parts of the KL-truncation errors vanish for such outputs. Furthermore, the current framework has been adapted to noncoercive inf-sup stable problems.

Chapter 6

RBM for Quadratically Nonlinear Parametric PDEs with Stochastic Influences

This chapter is based upon joint work with K. Urban and the main results have already been published in [93] in a very similar form. We showed that some assumptions regarding stochastic independence can be weakened such that more general classes of problems can be considered.

Deterministic parametrized quadratically nonlinear problems and RBM have been studied for affine problems [96] and non-affine problems [18]. The analysis is based on the Brezzi-Rappaz-Raviart (BRR) theory [13, 16]. RBM for stochastic parametrized linear problems have been studied in [12, 45] and in the previous chapter. The evaluation of statistical outputs such as second moment or variance requires good approximation procedures for quadratic output functionals which have already been developed in Chapter 5 for this special case and a linear PDE setting. In general, quadratic output functionals in the RB context are introduced in [55, 56].

In this chapter, we combine the methods for quadratic deterministic and linear stochastic problems for the case of a given affine decomposition with respect to the deterministic parameter. The affine decomposition with respect to the stochastic dependency is obtained using the KL expansion. Consequently, especially the

error analysis of state and linear output functional is very similar to [96] and [18], whereas the analysis of quadratic and statistical outputs is strongly based upon the results of chapter 5 and [45].

We begin in Section 6.1 with the introduction of the general variational formulation and its Frèchet derivative for the class of problems we are dealing with. Furthermore, we briefly describe the KL expansion and introduce the desired random and statistical outputs of interest. In Section 6.2, we present the nonlinear primal RB formulation of the problem and appropriate linear dual RB problems used for the RB approximation of the different outputs of interest. The a-posteriori analysis of the error of state and outputs is developed in Section 6.3. The offline-online decomposition is briefly described in Section 6.4, where also evaluation procedures for the inf-sup and continuity constants are presented, which are needed for the evaluation of the error bounds. Finally, in Section 6.5, we provide numerical experiments for a stationary quadratic convection-diffusion problem.

6.1 Problem Formulation

6.1.1 Variational Formulation

Let $D \subset \mathbb{R}^d$ denote an open, bounded, spatial domain, $\mathcal{P} \subset \mathbb{R}^p$ a set of deterministic parameters, and $(\Omega, \mathfrak{A}, \mathbb{P})$ a probability space. For some subspace $X \subset H^1(D)$ of dimension $\dim(X) = \mathcal{N}$, let $a_0 : X \times X \times \mathcal{P} \times \Omega \to \mathbb{R}$ be a bilinear form with respect to the first two arguments, $a_1 : X \times X \times X \times \mathcal{P} \times \Omega \to \mathbb{R}$ a trilinear form with respect to the first three arguments, and let $f : X \times \mathcal{P} \times \Omega \to \mathbb{R}$ be linear and bounded. We assume uniformly boundedness of a_0 and a_1, i.e., for $(\mu, \omega) \in \mathcal{P} \times \Omega$, there are continuity constants $0 < \rho_0(\mu, \omega) < \bar{\rho}_0 < \infty$ and $0 < \rho_1(\mu, \omega) < \bar{\rho}_1 < \infty$ such that

$$|a_0(u, v; \mu, \omega)| \;\leq\; \rho_0(\mu, \omega)\, \|u\|_X\, \|v\|_X, \quad u, v \in X, \tag{6.1}$$

$$|a_1(u, w, v; \mu, \omega)| \;\leq\; \rho_1(\mu, \omega)\, \|u\|_X\, \|w\|_X\, \|v\|_X, \quad u, w, v \in X. \tag{6.2}$$

For $(\mu, \omega) \in \mathcal{P} \times \Omega$ and $w, v \in X$, we define

$$g(w, v; \mu, \omega) \;:=\; a_0(w, v; \mu, \omega) + a_1(w, w, v; \mu, \omega) - f(v; \mu, \omega) \tag{6.3}$$

and solve the nonlinear, parametrized, and random variational problem

$$g(u(\mu,\omega),v;\mu,\omega) \;=\; 0 \quad \forall v \in X. \tag{6.4}$$

For the moment, we assume the existence of a solution of (6.4) for each pair (μ,ω). A detailed proof is given in Section 6.3, following the well known Brezzi–Rappaz–Raviart (BRR) theory [13].

6.1.2 Affine Decomposition via Karhunen–Loève Expansion

In order to achieve computational efficiency of an RBM, we assume g to allow for an affine decomposition in the deterministic parameter μ, namely

$$g(w,v;\mu,\omega) \;=\; \sum_{q=1}^{Q} \theta_q(\mu)\big[\bar{g}_q(w,v) + g_q(w,v;\omega)\big], \tag{6.5}$$

where $\bar{g}_q : X \times X \to \mathbb{R}$, $q = 1,\dots,Q$, are bounded and denote the expectations of the terms in brackets, and $g_q : X \times X \times \Omega \to \mathbb{R}$, $q = 1,\dots,Q$, have zero mean and represent the fluctuating parts. To separate also stochastic and spatial dependencies, we express $g_q(w,v;\omega)$ using Karhunen–Loève expansions [60, 65], and obtain

$$g_q(w,v;\omega) \;=\; \sum_{k=0}^{\infty} \xi_{q,k}(\omega)\, g_{q,k}(w,v), \quad q = 1,\dots,Q. \tag{6.6}$$

The random variables $\xi_{q,k} : \Omega \to \mathbb{R}$ are uncorrelated and have zero mean and unit variance. The bilinear forms $g_{q,k} : X \times X \to \mathbb{R}$, $q = 1,\dots,Q$, $k = 1,\dots,K$, are bounded and the magnitude is typically assumed to decrease exponentially fast in k. For numerical purposes, the infinite sums are usually restricted by some sufficiently large $K < \infty$, leading to truncated forms g_q^K and thereby g^K. The corresponding solution of the truncated form of (6.4) is denoted by $u_K(\mu,\omega)$.

In practice, we may have different numbers Q of affine terms for a_0, a_1 and f, and we may truncate each of the respective decomposed forms at different values of K. However, for notational convenience, we do not explicitly specify all dependencies but indicate them just by Q and K, respectively. Furthermore, an index or superscript K indicates that the expression denotes or is based upon truncated systems.

6.1.3 Newton Iteration

We iteratively solve (6.4) or the respective truncated problem using Newton's method. The Frèchet derivative of g at some point $z \in X$ is given by

$$dg(u, v; \mu, \omega)[z] \;=\; a_0(u, v; \mu, \omega) + a_1(u, z, v; \mu, \omega) + a_1(z, u, v; \mu, \omega) \tag{6.7}$$

and the respective truncated form is denoted by dg^K. For some initial guess $u_K^{[0]}(\mu, \omega)$, we solve

$$dg^K(du_K^{[i]}(\mu, \omega), v; \mu, \omega)[u_K^{[i]}(\mu, \omega)] \;=\; -g^K(u_K^{[i]}(\mu, \omega), v; \mu, \omega), \quad \forall v \in X \tag{6.8}$$

and evaluate the Newton update $u_K^{[i+1]}(\mu, \omega) = u_K^{[i]}(\mu, \omega) + du_K^{[i]}(\mu, \omega)$.

6.1.4 Output of Interest

As in Chapter 5, we are not only interested in the state $u(\mu, \omega)$ but also in some output functional

$$s(\mu, \omega) := \ell(u(\mu, \omega); \mu),$$

where $\ell : X \times \mathcal{P} \to \mathbb{R}$ denotes a parametric linear form. Furthermore, we may be interested in the squared functional $s^2(\mu, \omega) := (\ell(u(\mu, \omega), \mu))^2$.

Besides these random outputs, we again want to evaluate some statistical quantities such as first and second moment of $s(\mu, \omega)$, denoted by $\mathbb{M}_1(\mu) := \mathbb{E}[s(\mu, \cdot)]$ and $\mathbb{M}_2(\mu) := \mathbb{E}[s^2(\mu, \cdot)]$, respectively. Additionally, we have to provide the squared first moment $\mathbb{M}_1^2(\mu) = (\mathbb{E}[s(\mu, \cdot)])^2$ to evaluate the variance, given by

$$\mathbb{V}(\mu) = \mathbb{M}_2(\mu) - \mathbb{M}_1^2(\mu).$$

6.2 Reduced Basis System

In this section, we introduce reduced primal and dual systems that are used to derive good approximations of the desired random and statistical outputs of interest. For the construction of the dual problems, we combine the ideas of [18, 96], where dual problems for quadratically nonlinear problems with linear outputs are derived, and the methods of Chapter 5, where dual formulations for linear problems in combination with quadratic and statistical outputs have been introduced.

6.2.1 Primal-Dual Formulation for Linear Outputs

We create a reduced basis from solutions $\zeta_n := u_K(\mu_n, \omega_n)$ for some appropriate parameter set $\{\mu_n, \omega_n\}_{n=1}^N \in (\mathcal{P} \times \Omega)^N$, $N \ll \mathcal{N}$. The reduced space is given by $X_N = \text{span}(\{\zeta_n\}_{n=1}^N) \subset X$. Due to the affine decomposition of g and dg, it is possible to assemble and approximately solve the reduced system

$$g^K(u_{N,K}(\mu,\omega), v; \mu, \omega) \;=\; 0, \quad v \in X_N. \tag{6.9}$$

for each $(\mu, \omega) \in \mathcal{P} \times \Omega$ with computational complexity $\mathcal{O}(QKN^3I)$, independent of $\mathcal{N}$, where I denotes the number of Newton iterations. We also introduce a linear dual problem in full and truncated, reduced form,

$$dg(v, p^{(1)}(\mu,\omega); \mu, \omega)[\tfrac{1}{2}(u(\mu,\omega)+u_{N,K}(\mu,\omega))] \;=\; -\ell(v;\mu), \quad v \in X, \tag{6.10}$$

$$dg^K(v, p^{(1)}_{N,K}(\mu,\omega); \mu, \omega)[u_{N,K}(\mu,\omega)] \;=\; -\ell(v;\mu), \quad v \in \tilde{X}^{(1)}_N, \tag{6.11}$$

with solutions $p^{(1)}(\mu,\omega) \in X$ and $p^{(1)}_{N,K}(\mu,\omega) \in \tilde{X}^{(1)}_N$, respectively. The superscript $^{(1)}$ is motivated by the fact that we will introduce further dual problems later on. The reduced dual space $\tilde{X}^{(1)}_N$ of dimension $\tilde{N}^{(1)} \ll \mathcal{N}$ is constructed analogously to X_N as the span of solutions of (6.10) or of the corresponding truncated system for appropriate parameter pairs $(\mu, \omega) \in \mathcal{P} \times \Omega$. The complexity to solve the dual problem corresponds to just one Newton iteration of the primal problem. Here and in the following, an index or superscript N indicates that the expression denotes or is based on reduced systems. We do not explicitly indicate the dependencies on the different dimensions of the primal and dual reduced systems. For notational simplicity, we also omit the parameter pair (μ, ω) in many cases, where it does not affect the understanding.

Let $r_{\text{RB}}(v; \mu, \omega) := g^K(u_{N,K}(\mu,\omega), v; \mu, \omega)$ be the residual of the reduced primal problem for some $v \in X$. We define the RB approximation of the linear output $s(\mu, \omega)$ and its corresponding linear statistical output, the first moment $\mathbb{M}_1(\mu)$, by

$$s_{N,K}(\mu,\omega) \;:=\; \ell(u_{N,K};\mu) + r_{\text{RB}}(p^{(1)}_{N,K}; \mu, \omega), \tag{6.12}$$

$$\mathbb{M}_{1,NK}(\mu) \;:=\; \mathbb{E}[s_{N,K}(\mu,\cdot)], \tag{6.13}$$

respectively, where $r_{\text{RB}}(p^{(1)}_{N,K}(\mu,\omega); \mu, \omega)$ has been added as a correction term to improve the approximation. In Section 6.3, we will provide error bounds to show that this choice leads to good results.

6.2.2 Dual Formulations for Quadratic Outputs

As mentioned in Section 6.1.4, we are also interested in the squared output $s^2(\mu,\omega)$. Since the straightforward approximation $(s_{N,K}(\mu,\omega))^2$ does not lead to accurate results (cf. Section 5.3.4), we define $\ell^{(2)}(\mu,\omega) := 2s_{N,K}(\mu,\omega)\ell(v;\mu)$ and introduce the additional linear dual problems, full and reduced,

$$dg(v,p^{(2)}(\mu,\omega);\mu,\omega)[\tfrac{1}{2}(u(\mu,\omega)+u_{N,K}(\mu,\omega))] \;=\; -\ell^{(2)}(v;\mu,\omega),\; v\in X, \tag{6.14}$$

$$dg^K(v,p^{(2)}_{N,K}(\mu,\omega);\mu,\omega)[u_{N,K}(\mu,\omega)] \;=\; -\ell^{(2)}(v;\mu,\omega),\; v\in \tilde{X}^{(2)}_N, \tag{6.15}$$

with solutions $p^{(2)}(\mu,\omega)\in X$ and $p^{(2)}_{N,K}(\mu,\omega)\in \tilde{X}^{(2)}_N$, respectively, using some appropriate reduced dual space $\tilde{X}^{(2)}_N$ of dimension $\tilde{N}^{(2)} \ll \mathcal{N}$. Analogously to Chapter 5, the RB approximation of the quadratic output $s^2(\mu,\omega)$ and its corresponding statistical output, the second moment $\mathbb{M}_2(\mu)$, are then defined by

$$s^{[2]}_{N,K}(\mu,\omega) \;:=\; (s_{N,K})^2 + 2s_{N,K}\, r_{\mathrm{RB}}(p^{(1)}_{N,K};\mu,\omega) - r_{\mathrm{RB}}(p^{(2)}_{N,K};\mu,\omega), \tag{6.16}$$

$$\mathbb{M}_{2,NK}(\mu) \;:=\; \mathbb{E}\left[s^{[2]}_{N,K}(\mu,\cdot)\right], \tag{6.17}$$

i.e., we add two additional correction terms compared to the straightforward approximation.

6.2.3 Dual Formulation for the Variance Approximation

To develop good approximations of the variance $\mathbb{V}(\mu) = \mathbb{M}_2(\mu) - \mathbb{M}_1^2(\mu)$, it remains to find RB estimates of $\mathbb{M}_1^2(\mu)$. We define $\ell^{(3)}(\mu,\omega) := 2\mathbb{M}_{1,NK}(\mu,\omega)\ell(v;\mu)$ and introduce the additional linear dual problems, full and reduced,

$$dg(v,p^{(3)}(\mu,\omega);\mu,\omega)[\tfrac{1}{2}(u(\mu,\omega)+u_{N,K}(\mu,\omega))] \;=\; -\ell^{(3)}(v;\mu,\omega),\; v\in X, \tag{6.18}$$

$$dg^K(v,p^{(3)}_{N,K}(\mu,\omega);\mu,\omega)[u_{N,K}(\mu,\omega)] \;=\; -\ell^{(3)}(v;\mu,\omega),\; v\in \tilde{X}^{(3)}_N, \tag{6.19}$$

with solutions $p^{(3)}(\mu,\omega)\in X$ and $p^{(3)}_{N,K}(\mu,\omega)\in \tilde{X}^{(3)}_N$, respectively, using some appropriate reduced dual space $\tilde{X}^{(3)}_N$ of dimension $\tilde{N}^{(3)} \ll \mathcal{N}$. The RB approximations of the squared first moment $\mathbb{M}_1^2(\mu)$ and the variance $\mathbb{V}(\mu)$ are then given by

$$\mathbb{M}^{[2]}_{1,NK}(\mu) \;:=\; (\mathbb{M}_{1,NK})^2 + 2\mathbb{M}_{1,NK}\mathbb{E}\left[r_{\mathrm{RB}}(p^{(1)}_{N,K})\right] - \mathbb{E}\left[r_{\mathrm{RB}}(p^{(3)}_{N,K})\right], \tag{6.20}$$

$$\mathbb{V}_{NK}(\mu) \;:=\; \mathbb{E}\left[s^{[2]}_{N,K}(\mu,\cdot)\right] \;-\; \mathbb{M}^{[2]}_{1,NK}(\mu), \tag{6.21}$$

respectively. Analogously to (6.16), we added two correction terms.

In our numerical experiments, we have observed that it is sufficient to use the same reduced space for the second and third dual problem, i.e., $\tilde{X}_N^{(2)} = \tilde{X}_N^{(3)}$. Hence, we just solve

$$dg^K(v, p_{N,K}^{(4)}(\mu,\omega);\mu,\omega)[u_{N,K}(\mu,\omega)] \;=\; -2\ell(v;\mu,\omega), \quad \forall v \in \tilde{X}_N^{(2)} \tag{6.22}$$

for $p_{N,K}^{(4)}(\mu,\omega) \in \tilde{X}_N^{(2)}$ such that $p_{N,K}^{(2)} = s_{N,K} \cdot p_{N,K}^{(4)}$ and $p_{N,K}^{(3)} = \mathbb{M}_{1,NK} \cdot p_{N,K}^{(4)}$.

6.3 A-Posteriori Analysis

Parts of the following analysis are based on the Brezzi-Rappaz-Raviart (BRR) theory [13, 16] which has already been used in the RB context for affine deterministic problems, e.g., in [96], and non-affine deterministic problems, e.g., in [18]. Consequently, especially the analysis in Sections 6.3.2 to 6.3.4 is very similar to parts of the mentioned publications. The analysis of quadratic and statistical outputs is based upon the results in Chapter 5, where the linear stochastic case has been discussed.

Under the assumption that solutions $u(\mu,\omega)$ of (6.4) and $u_{N,K}(\mu,\omega)$ of (6.9) exist, we define the inf-sup constant $\beta(\mu,\omega)$ as

$$\beta(\mu,\omega) \;:=\; \inf_{w\in X} \sup_{v\in X} \frac{dg(w,v;\mu,\omega)[u_{N,K}(\mu,\omega)]}{\|w\|_X \|v\|_X}. \tag{6.23}$$

We furthermore assume the existence of some $\beta_0 > 0$ such that $\beta(\mu,\omega) > \beta_0$ for all $(\mu,\omega) \in \mathcal{P} \times \Omega$. Existence and uniqueness of solutions of the dual problems (6.11), (6.15) and (6.19) follows immediately. We furthermore assume the availability of a positive lower bound $\beta_{\mathrm{LB}}(\mu,\omega)$ of the inf-sup constant $\beta(\mu,\omega)$ and an efficient evaluation procedure, compare Section 6.4.2.

6.3.1 Notation

We first introduce some notation for the subsequent analysis. Let

$$\begin{aligned} e_{\mathrm{RB}}(\mu,\omega) \;&:=\; u_K(\mu,\omega) - u_{N,K}(\mu,\omega), \\ \tilde{e}_{\mathrm{RB}}^{(i)}(\mu,\omega) \;&:=\; p_K^{(i)}(\mu,\omega) - p_{N,K}^{(i)}(\mu,\omega), \quad i = 1,2,3, \end{aligned}$$

denote the error between the reduced primal and dual solutions and the corresponding solutions of the full but truncated systems, respectively. Furthermore, let

$$\begin{aligned} e(\mu,\omega) &:= u(\mu,\omega) - u_{N,K}(\mu,\omega), \\ \tilde{e}^{(i)}(\mu,\omega) &:= p^{(i)}(\mu,\omega) - p^{(i)}_{N,K}(\mu,\omega), \quad i=1,2,3, \end{aligned}$$

denote the total error of the reduced primal and dual solutions, respectively. We define the RB residuals

$$\begin{aligned} r_{\mathrm{RB}}(v;\mu,\omega) &:= g^K(u_{N,K},v;\mu,\omega) &&= g^K(e_{\mathrm{RB}},v;\mu,\omega), \\ \tilde{r}^{(i)}_{\mathrm{RB}}(v;\mu,\omega) &:= dg^K(v,p^{(i)}_{N,K})[u_{N,K}] + \ell^{(i)}(v) &&= dg^K(v,\tilde{e}^{(i)}_{\mathrm{RB}})[u_{N,K}], \quad i=1,2,3, \end{aligned}$$

as a "measure" of the error that results from the basis reduction. Additionally, we define some KL "residuals" indicating the truncation errors $g - g^K$ and $dg - dg^K$. To obtain truncation bounds independent of the actual random realization, we replace the random variables $\xi_{q,k}$, $k > K$ by some ϱ-quantile $\xi^{\varrho}_{\mathrm{UB}}$, i.e., we define some $0 \leq \varrho \ll 1$ such that $|\xi_{q,k}| \leq \xi^{\varrho}_{\mathrm{UB}}$ holds with probability $1-\varrho$. We define

$$\begin{aligned} \delta_{\mathrm{KL}}(v;\mu,\omega) &:= \sum_{q=1}^{Q} |\theta_q(\mu)| \sum_{k=K+1}^{\infty} \xi^{\varrho}_{\mathrm{UB}} \, |g_{q,k}(u_{N,K},v)|\,, \\ \tilde{\delta}^{(i)}_{\mathrm{KL}}(v;\mu,\omega) &:= \sum_{q=1}^{Q} |\theta_q(\mu)| \sum_{k=K+1}^{\infty} \xi^{\varrho}_{\mathrm{UB}} \, |dg_{q,k}(v,p^{(i)}_{N,K})[u_{N,K}]|, \quad i=1,2,3. \end{aligned}$$

For numerical purposes, the possibly infinte sums in the above definitions will be truncated as well at some large $K_{\max} > K$ such that the additional truncation error is negligible.

Since we replaced the random variables $\xi_{q,k}(\omega)$ by its ϱ-quantile $\xi^{\varrho}_{\mathrm{UB}}$, the KL residuals δ_{KL} and $\tilde{\delta}^{(i)}_{\mathrm{KL}}$ are not residuals in the classical sense but represent corresponding quantiles, i.e., $\delta_{\mathrm{KL}}(v) \geq |(g - g^K)(u_{N,K},v)|$ and $\tilde{\delta}^{(i)}_{\mathrm{KL}}(v) \geq |(dg - dg^K)(v,p^{(i)}_{N,K})[u_{N,K}]|$ holds with a certain probability. In many cases, the random variables $\xi_{q,k}(\omega)$ are bounded since the underlying problem restricts their variations. Then, we can choose $\varrho = 0$ and obtain rigorous bounds. Otherwise, ϱ should be sufficiently small to be negligible in the following analysis.

Based on the introduced residuals, we define RB and KL bounds for $i \in \{1, 2, 3\}$,

$$\Delta_{\mathrm{RB}}(\mu,\omega) := \frac{1}{\beta_{\mathrm{LB}}} \sup_{v\in X} \left(\frac{r_{\mathrm{RB}}(v)}{\|v\|_X} \right), \quad \tilde{\Delta}^{(i)}_{\mathrm{RB}}(\mu,\omega) := \frac{1}{\beta_{\mathrm{LB}}} \sup_{v\in X} \left(\frac{\tilde{r}^{(i)}_{\mathrm{RB}}(v)}{\|v\|_X} \right), \tag{6.24}$$

$$\Delta_{\mathrm{KL}}(\mu,\omega) := \frac{1}{\beta_{\mathrm{LB}}} \sup_{v\in X} \left(\frac{\delta_{\mathrm{KL}}(v)}{\|v\|_X} \right), \quad \tilde{\Delta}^{(i)}_{\mathrm{KL}}(\mu,\omega) := \frac{1}{\beta_{\mathrm{LB}}} \sup_{v\in X} \left(\frac{\tilde{\delta}^{(i)}_{\mathrm{KL}}(v)}{\|v\|_X} \right). \tag{6.25}$$

Before we provide the actual error bounds for the state and the outputs, we introduce a so-called proximity indicator $\tau(\mu,\omega)$ which can be seen as a dimensionless measure of the residuals. Similarly to [18, 96], we define

$$\tau(\mu,\omega) \;:=\; 4\frac{\rho_1(\mu,\omega)}{\beta_{\mathrm{LB}}(\mu,\omega)} \left(\Delta_{\mathrm{RB}}(\mu,\omega) + \Delta_{\mathrm{KL}}(\mu,\omega)\right), \tag{6.26}$$

where $\rho_1(\mu,\omega)$ is given by (6.2). For $\tau(\mu,\omega) < 1$, we furthermore define

$$d(\mu,\omega) \;:=\; \left(1 + \sqrt{1-\tau(\mu,\omega)}\right)^{-1} \tag{6.27}$$

which will appear as a factor in the upcoming error bounds. It is easy to see that $d(\mu,\omega)$ is decreasing in $\tau(\mu,\omega)$ and takes values in the interval $[1/2, 1)$.

6.3.2 Primal Solution Error

For $\tau(\mu,\omega) < 1$, we define the bound

$$\Delta(\mu,\omega) \;:=\; 2d(\mu,\omega) \left(\Delta_{\mathrm{RB}}(\mu,\omega) + \Delta_{\mathrm{KL}}(\mu,\omega)\right). \tag{6.28}$$

Since $d(\mu,\omega)$ approaches $1/2$ for small τ, the bound $\Delta(\mu,\omega)$ approaches $\Delta_{\mathrm{RB}}(\mu,\omega) + \Delta_{\mathrm{KL}}(\mu,\omega)$ which corresponds to the bound in the linear case, cf. Chapter 5. To show that $\Delta(\mu,\omega)$ is indeed an upper bound for the error of the reduced primal solution $u_{N,K}$, we need the following statement, which has been introduced and proved almost analogously for deterministic problems in [18, 96].

Lemma 6.1. *For $(\mu,\omega) \in \mathcal{P} \times \Omega$ and $\tau(\mu,\omega) < 1$, let $u_{N,K}(\mu,\omega)$ be a solution of (6.9). We define the operator $\Phi : X \times \mathcal{P} \times \Omega \to X$ by*

$$\begin{aligned} &dg(\Phi(w;\mu,\omega), v;\mu,\omega)[u_{N,K}(\mu,\omega)] \\ &\quad= \; dg(w,v;\mu,\omega)[u_{N,K}(\mu,\omega)] - g(w,v;\mu,\omega) \quad \forall v \in X, \end{aligned}$$

for a given $w \in X$. Then, Φ has a unique fixed point $w^*(\mu,\omega)$ in the ball $B(u_{N,K}(\mu,\omega), r(\mu,\omega)) \subset X$ where the radius $r(\mu,\omega)$ is in the interval

$$r(\mu,\omega) \in \left[\Delta(\mu,\omega),\ \frac{\beta_{\mathrm{LB}}(\mu,\omega)}{2\rho_1(\mu,\omega)}\right).$$

Proof. We omit all parameter dependencies for notational convenience. First, we show the identity

$$g(w_2,v) - g(w_1,v) \ = \ dg(w_2-w_1,v)[\tfrac{1}{2}(w_2+w_1)]. \tag{6.29}$$

Using just the definition of g in (6.3), we have

$$g(w_2,v) - g(w_1,v) = a_0(w_2-w_1,v) + a_1(w_2,w_2,v) - a_1(w_1,w_1,v).$$

With the definition of dg in (6.7), we obtain

$$\begin{aligned} dg(w_2-w_1,v)[\tfrac{1}{2}(w_2+w_1)] &= a_0(w_2-w_1,v) + \tfrac{1}{2}a_1(w_2-w_1,w_2+w_1,v) \\ &\quad + \tfrac{1}{2}a_1(w_2+w_1,w_2-w_1,v) \\ &= a_0(w_2-w_1,v) + a_1(w_2,w_2,v) - a_1(w_1,w_1,v), \end{aligned}$$

where the last equation is obtained by expanding the "a_1" terms. Together, we obtain the identity (6.29).

Using again the definition of dg in (6.7) and the continuity assumption (6.2), it is straightforward to show the inequality

$$\begin{aligned} dg(w,v)[z_2] - dg(w,v)[z_1] \ &= \ a_1(w,z_2-z_1,v) + a_1(z_2-z_1,w,v) \\ &\leq \ 2\rho_1\|w\|_X\|v\|_X\|z_2-z_1\|_X. \end{aligned} \tag{6.30}$$

To prove the Lemma, we apply these results and use the Banach fixed point theorem. We first show that Φ is a contraction on $\bar{B}(u_{N,K},r)$ for some $r>0$. For $w_1,w_2 \in \bar{B}(u_{N,K},r)$, we know that $\frac{1}{2}(w_2+w_1) \in \bar{B}(u_{N,K},r)$. Using the definition of Φ and (6.29), we obtain

$$\begin{aligned} dg(\Phi[w_2]-\Phi[w_1],v)[u_{N,K}] \ &= \ dg(w_2-w_1,v)[u_{N,K}] - (g(w_2,v)-g(w_1,v)) \\ &= \ dg(w_2-w_1,v)[u_{N,K}] - dg(w_2-w_1,v)[\tfrac{1}{2}(w_2+w_1)] \\ &= \ dg(w_2-w_1,v)[u_{N,K} - \tfrac{1}{2}(w_2+w_1)]. \end{aligned}$$

Hence, applying (6.30) and the fact that $\frac{1}{2}(w_2 + w_1) \in \bar{B}(u_{N,K}, r)$,

$$\begin{aligned} |dg(\Phi[w_2] - \Phi[w_1], v)[u_{N,K}]| &\leq 2\rho_1 \|w_2 - w_1\|_X \|v\|_X \|u_{N,K} - \tfrac{1}{2}(w_2 + w_1)\|_X \\ &\leq 2r\rho_1 \|w_2 - w_1\|_X \|v\|_X. \end{aligned}$$

We use this result and the inf-sup constant (6.23),

$$\begin{aligned} \|\Phi[w_2] - \Phi[w_1]\|_X &\leq \frac{1}{\beta_{\mathrm{LB}}} \sup_{v \in X} \frac{dg(\Phi[w_2] - \Phi[w_1], v)[u_{N,K}]}{\|v\|_X} \\ &\leq \frac{2r\rho_1}{\beta_{\mathrm{LB}}} \|w_2 - w_1\|_X. \end{aligned}$$

Hence, Φ is a contraction for $0 < r < \beta_{\mathrm{LB}}/2\rho_1$.

Next, we show that there is a radius $r \in (0, \beta_{\mathrm{LB}}/2\rho_1)$ such that Φ maps the ball $\bar{B}(u_{N,K}, r)$ into itself. For $w \in \bar{B}(u_{N,K}, r)$, it holds with (6.29) that

$$\begin{aligned} dg(\Phi[w] - u_{N,K}, v) &= dg(w - u_{N,K}, v)[u_{N,K}] - g(w, v) \\ &= dg(w - u_{N,K}, v)[u_{N,K}] - (g(w, v) - g(u_{N,K}, v)) - g(u_{N,K}, v) \\ &= dg(w - u_{N,K}, v)[u_{N,K}] - dg(w - u_{N,K}, v)[\tfrac{1}{2}(w + u_{N,K})] \\ &\quad - g(u_{N,K}, v). \end{aligned}$$

Using again (6.30), we obtain

$$\begin{aligned} |dg(w - u_{N,K}, v)[u_{N,K}] - dg(w - u_{N,K}, v)[\tfrac{1}{2}(w + u_{N,K})]| &\leq \rho_1 \|w - u_{N,K}\|_X^2 \|v\|_X \\ &\leq \rho_1 r^2 \|v\|_X. \end{aligned}$$

Furthermore, it is clear that

$$|g(u_{N,K}, v)| \leq |(g - g^K)(u_{N,K}, v)| + |g^K(u_{N,K}, v)| \leq \delta_{\mathrm{KL}}(v) + |r_{\mathrm{RB}}(v)|.$$

Hence, using again the inf-sup constant (6.23), we get

$$\|\Phi[w] - u_{N,K}\|_X \leq \frac{1}{\beta_{\mathrm{LB}}} \sup_{v \in X} \frac{dg(\Phi[w] - u_{N,K}, v)}{\|v\|_X} \leq \frac{\rho_1 r^2}{\beta_{\mathrm{LB}}} + (\Delta_{\mathrm{KL}} + \Delta_{\mathrm{RB}}).$$

Therefore, Φ maps $\bar{B}(u_{N,K}, r)$ into itself for all r with $\rho_1 r^2 \beta_{\mathrm{LB}}^{-1} + \Delta_{\mathrm{KL}} + \Delta_{\mathrm{RB}} < r$, which holds for $r \in [\Delta, \beta_{\mathrm{LB}}/(2\rho_1 d)]$. Since $d < 1$ by (6.27), Φ has a unique fixed point on $B(u_{N,K}, r)$ for $r \in [\Delta, \beta_{\mathrm{LB}}/(2\rho_1))$. □

Proposition 6.2. *For $\tau(\mu, \omega) < 1$, $(\mu, \omega) \in \mathcal{P} \times \Omega$, there exists a unique solution $u(\mu, \omega) \in B\big(u_{N,K}(\mu, \omega), \frac{\beta_{\mathrm{LB}}(\mu,\omega)}{2\rho_1(\mu,\omega)}\big)$ of (6.4) such that $\|e(\mu, \omega)\|_X \leq \Delta(\mu, \omega)$.*

Proof. The proof follows directly from Lemma 6.1. Since the fixed point of Φ solves (6.4), we have existence and uniqueness in $B(u_{N,K}, \frac{\beta_{\mathrm{LB}}}{2\rho_1})$. Furthermore, the fixed point is in the ball $B(u_{N,K}, \Delta)$ which leads to the error bound. □

At the beginning of Section 6.3, we assumed the existence of solutions $u(\mu,\omega)$ of (6.4) and $u_{N,K}(\mu,\omega)$ of (6.9). With Proposition 6.2, we can prove existence and local uniqueness of $u(\mu,\omega)$ a-posteriori, solving just the reduced problem and evaluating $\tau(\mu,\omega)$. However, the reduced basis has to be sufficiently large to fulfill the requirement $\tau(\mu,\omega) < 1$. This reflects the fact that we can not expect well-posedness of the nonlinear problem for all parameters μ and ω.

6.3.3 Dual Solution Error

For the dual solutions $p^{(i)}_{N,K}(\mu,\omega)$ of (6.11), (6.15) and (6.19), we define the bounds $\tilde{\Delta}^{(i)}(\mu,\omega)$, $i \in \{1,2,3\}$, by

$$\begin{aligned}\tilde{\Delta}^{(i)}(\mu,\omega) \;&:= 2d(\mu,\omega)\left(\tilde{\Delta}^{(i)}_{\mathrm{RB}}(\mu,\omega)+\tilde{\Delta}^{(i)}_{\mathrm{KL}}(\mu,\omega)\right)\\ &\quad + 2d(\mu,\omega)\frac{\rho_1(\mu,\omega)}{\beta_{\mathrm{LB}}(\mu,\omega)}\Delta(\mu,\omega)\|p^{(i)}_{N,K}(\mu,\omega)\|_X. \qquad (6.31)\end{aligned}$$

The last term of (6.31) can also be expressed in terms of τ and d and we obtain the alternative notation $\tilde{\Delta}^{(i)} = 2d(\tilde{\Delta}^{(i)}_{\mathrm{RB}} + \tilde{\Delta}^{(i)}_{\mathrm{KL}}) + d^2\tau\|p^{(i)}_{N,K}\|_X$.

Proposition 6.3. *For $\tau(\mu,\omega) < 1$, it holds that $\|\tilde{e}^{(i)}(\mu,\omega)\|_X \le \tilde{\Delta}^{(i)}(\mu,\omega)$ for $i \in \{1,2,3\}$, $(\mu,\omega) \in \mathcal{P}\times\Omega$.*

Proof. It is straightforward that

$$\begin{aligned}&dg(v,\tilde{e}^{(i)})[u_{N,K}]\\ &= dg(v,p^{(i)})[\tfrac{1}{2}(u+u_{N,K})] - dg(v,p^{(i)})[\tfrac{1}{2}(u-u_{N,K})] - dg(v,p^{(i)}_{N,K})[u_{N,K}].\end{aligned}$$

Let us consider the first and last term.

$$\begin{aligned}&\left|dg(v,p^{(i)})[\tfrac{1}{2}(u+u_{N,K})] - dg(v,p^{(i)}_{N,K})[u_{N,K}]\right|\\ &= \left|\ell^{(i)}(v) - dg^K(v,p^{(i)}_{N,K})[u_{N,K}] - (dg-dg^K)(v,p^{(i)}_{N,K})[u_{N,K}]\right|\\ &\le \left|\tilde{r}^{(i)}_{\mathrm{RB}}(v)\right| + \left|\tilde{\delta}^{(i)}_{\mathrm{KL}}(v)\right|.\end{aligned}$$

For the middle term, we use $p^{(i)} = \tilde{e}^{(i)} + p_{N,K}^{(i)}$ and inequality (6.30) to obtain

$$\left|dg(v, p^{(i)})[\tfrac{1}{2}(u - u_{N,K})]\right| \;\leq\; \rho_1 \|e\|_X \left(\|\tilde{e}^{(i)}\|_X + \|p_{N,K}^{(i)}\|_X\right) \|v\|_X.$$

We combine these results to estimate the error $\tilde{e}^{(i)}$. Using the inf-sup condition (6.23), we obtain

$$\begin{aligned}\|\tilde{e}^{(i)}\|_X &\leq \frac{1}{\beta_{\mathrm{LB}}} \sup_{v \in X} \frac{dg(v, \tilde{e}^{(i)})[u_{N,K}]}{\|v\|_X} \\ &\leq \left(\tilde{\Delta}_{\mathrm{RB}}^{(i)} + \tilde{\Delta}_{\mathrm{KL}}^{(i)}\right) + \frac{\rho_1}{\beta_{\mathrm{LB}}} \Delta\big(\|\tilde{e}^{(i)}\|_X + \|p_{N,K}^{(i)}\|_X\big),\end{aligned}$$

i.e.,

$$\|\tilde{e}^{(i)}\|_X \left(1 - \frac{\rho_1}{\beta_{\mathrm{LB}}} \Delta\right) \;\leq\; \left(\tilde{\Delta}_{\mathrm{RB}}^{(i)} + \tilde{\Delta}_{\mathrm{KL}}^{(i)}\right) + \frac{\rho_1}{\beta_{\mathrm{LB}}} \Delta \|p_{N,K}^{(i)}\|_X.$$

Since $(1 - \frac{\rho_1}{\beta_{\mathrm{LB}}}\Delta) = (1 - \frac{1}{2}d\tau) = (1 - \frac{1}{2}\frac{\tau}{1+\sqrt{1-\tau}}) = (\frac{2}{2} - \frac{1-\sqrt{1-\tau}}{2}) = \frac{1+\sqrt{1-\tau}}{2} = \frac{1}{2d}$, using the definitions of τ in (6.26) and d in (6.27), the claim is proven. □

6.3.4 Linear Output Error

In the subsequent sections, we provide bounds for the errors between the outputs defined in Section 6.1.4 and its approximations. In all proofs, we omit the parameters (μ, ω) for notational compactness. In this section, we provide error bounds for the approximations of the linear output $s(\mu, \omega)$ and the first moment $\mathbb{M}_1(\mu)$. However, we start with some assumptions and statements that will be used in the proofs of all output error bounds.

Assumption 6.4. *The sets of random variables*

$$\left\{\xi_{q,k}\right\}_{\substack{k=1,\dots,K \\ q=1,\dots,Q}} \qquad \textit{and} \qquad \left\{\xi_{q,k}\right\}_{\substack{k>K \\ q=1,\dots,Q}}$$

from (6.6) *are uncorrelated from each other.*

Assumption 6.4 is fulfilled for example if the different g_q from (6.5) are stochastically independent or uncorrelated. However, it has already been stated in Remark 5.13 that it is also possible to deal with correlated terms g_q and $g_{q'}$, using joint KL expansions (cf. Section 2.2.3). Hence, Assumption 6.4 can easily be fulfilled for all kinds of problems.

Lemma 6.5. *Under Assumption 6.4, we have*

$$\mathbb{E}\left[g(u_{N,K}, p^{(i)}_{N,K}) - g^K(u_{N,K}, p^{(i)}_{N,K})\right] = 0, \quad i = 1, 2, 3.$$

Proof. Since $u_{N,K}$ and $p^{(i)}_{N,K}$ depend only on truncated forms, they depend only on the random variables $\left\{\xi_{q,k}\right\}_{\substack{k=1,\dots,K\\q=1,\dots,Q}}$. Since, by Assumption 6.4, $\left\{\xi_{q,k}\right\}_{\substack{k=1,\dots,K\\q=1,\dots,Q}}$ is uncorrelated to $\left\{\xi_{q,k}\right\}_{\substack{k>K\\q=1,\dots,Q}}$, both $u_{N,K}$ and $p^{(i)}_{N,K}$ are uncorrelated to $\left\{\xi_{q,k}\right\}_{\substack{k>K\\q=1,\dots,Q}}$. We thus obtain

$$\begin{aligned}
&\mathbb{E}\left[g(u_{N,K}, p^{(i)}_{N,K}) - g^K(u_{N,K}, p^{(i)}_{N,K})\right] \\
&= \mathbb{E}\left[\sum_{q=1}^{Q}\sum_{k=K+1}^{\infty}\theta_q(\mu)\xi_{q,k}(\cdot)g_{q,k}(u_{N,K}, p^{(i)}_{N,K})\right] \\
&= \sum_{q=1}^{Q}\sum_{k=K+1}^{\infty}\theta_q(\mu)\underbrace{\mathbb{E}\left[\xi_{q,k}(\cdot)\right]}_{=0}\mathbb{E}\left[g_{q,k}(u_{N,K}, p^{(i)}_{N,K})\right] \;=\; 0
\end{aligned}$$

which proves the claim. □

Lemma 6.6. *Let $u(\mu,\omega)$ be the solution of (6.4), $u_{N,K}(\mu,\omega)$ the solution of (6.9) and $p^{(i)}(\mu,\omega)$, $i = 1, 2, 3$, the solutions of (6.10), (6.14) and (6.18), respectively. For $i \in \{1, 2, 3\}$, it holds that $\ell^{(i)}(u) - \ell^{(i)}(u_{N,K}) = g(u_{N,K}, p^{(i)})$.*

Proof. Since $\ell^{(i)}(u) - \ell^{(i)}(u_{N,K}) = \ell^{(i)}(e)$ and using the respective dual formulation (6.10), (6.14) or (6.18), we have

$$\begin{aligned}
\ell^{(i)}(u) - \ell^{(i)}(u_{N,K}) &= -dg(e, p^{(i)})[\tfrac{1}{2}(u + u_{N,K})] \\
&= -a_0(e, p^{(i)}) - \tfrac{1}{2}a_1(e, u+u_{N,K}, p^{(i)}) - \tfrac{1}{2}a_1(u+u_{N,K}, e, p^{(i)}) \\
&= -a_0(u, p^{(i)}) - a_1(u, u, p^{(i)}) \\
&\quad + a_0(u_{N,K}, p^{(i)}) + a_1(u_{N,K}, u_{N,K}, p^{(i)}) \\
&= -f(p^{(i)}) + a_0(u_{N,K}, p^{(i)}) + a_1(u_{N,K}, u_{N,K}, p^{(i)}) \\
&= g(u_{N,K}, p^{(i)}),
\end{aligned}$$

which proves the postulated equality. □

Let us now introduce the bound for the error between the linear output $s(\mu,\omega)$ and its approximation $s_{N,K}(\mu,\omega)$ defined in (6.12). We define the bound $\Delta^s(\mu,\omega)$ by

$$\Delta^s(\mu,\omega) \;:=\; \frac{\beta_{\mathrm{LB}}(\mu,\omega)}{2d(\mu,\omega)}\Delta(\mu,\omega)\tilde{\Delta}^{(1)}(\mu,\omega) \;+\; \delta_{\mathrm{KL}}(p^{(1)}_{N,K}(\mu,\omega);\mu,\omega). \tag{6.32}$$

Proposition 6.7. *For $\tau(\mu,\omega) < 1$, it holds that $|s(\mu,\omega) - s_{N,K}(\mu,\omega)| \leq \Delta^s(\mu,\omega)$.*

Proof. From Lemma 6.6, we know that $\ell(u) - \ell(u_{N,K}) = g(u_{N,K}, p^{(1)})$. Hence, with $s_{N,K}$ from (6.12), we obtain

$$\begin{aligned} s - s_{N,K} &= g(u_{N,K}, p^{(1)}) - g^K(u_{N,K}, p^{(1)}_{N,K}) \\ &= g^K(u_{N,K}, p^{(1)}) - g^K(u_{N,K}, p^{(1)}_{N,K}) + (g - g^K)(u_{N,K}, p^{(1)}) \\ &= g^K(u_{N,K}, \tilde{e}^{(1)}) + (g - g^K)(u_{N,K}, \tilde{e}^{(1)}) + (g - g^K)(u_{N,K}, p^{(1)}_{N,K}). \end{aligned}$$

We use the definition of the bounds introduced in Section 6.3.1 and estimate

$$\begin{aligned} |s - s_{N,K}| &\leq |r_{\mathrm{RB}}(\tilde{e}^{(1)})| + \delta_{\mathrm{KL}}(\tilde{e}^{(1)}) + \delta_{\mathrm{KL}}(p^{(1)}_{N,K}) \\ &\leq \beta_{\mathrm{LB}}\Delta_{\mathrm{RB}}\|\tilde{e}^{(1)}\|_X + \beta_{\mathrm{LB}}\Delta_{\mathrm{KL}}\|\tilde{e}^{(1)}\|_X + \delta_{\mathrm{KL}}(p^{(1)}_{N,K}) \\ &\leq \beta_{\mathrm{LB}}(\Delta_{\mathrm{RB}} + \Delta_{\mathrm{KL}})\tilde{\Delta}^{(1)} + \delta_{\mathrm{KL}}(p^{(1)}_{N,K}), \end{aligned}$$

which proves the claim. □

With Proposition 6.7 and Lemma 6.5 at hand, it is clear that we can easily define a good bound for the error between the first moment $\mathbb{M}_1(\mu)$ and its approximation $\mathbb{M}_{1,NK}(\mu)$ as defined in (6.13). We define the bound $\Delta^{\mathbb{M}_1}(\mu)$ by

$$\Delta^{\mathbb{M}_1}(\mu) := \mathbb{E}\left[\frac{\beta_{\mathrm{LB}}(\mu,\cdot)}{2d(\mu,\cdot)}\Delta(\mu,\cdot)\tilde{\Delta}^{(1)}(\mu,\cdot)\right], \tag{6.33}$$

i.e., compared to $\mathbb{E}[\Delta^s]$, the last term has been omitted.

Corollary 6.8. *Under Assumption 6.4, for $\mu \in \mathcal{P}$ and $\tau(\mu,\cdot) < 1$, it holds that $|\mathbb{M}_1(\mu) - \mathbb{M}_{1,NK}(\mu)| \leq \Delta^{\mathbb{M}_1}(\mu)$.*

Proof. From the proof of Proposition 6.7 and Definition (6.13), we know that

$$\begin{aligned} \mathbb{M}_1(\mu) - \mathbb{M}_{1,NK}(\mu) = \mathbb{E}\left[g^K(u_{N,K}, \tilde{e}^{(1)}) + (g - g^K)(u_{N,K}, \tilde{e}^{(1)})\right] \\ + \mathbb{E}\left[(g - g^K)(u_{N,K}, p^{(1)}_{N,K})\right]. \end{aligned}$$

Since Assumption 6.4 holds, we can apply Lemma 6.5 and the last term vanishes. Following the proof of Proposition 6.7 directly leads to the desired result. □

6.3.5 Quadratic Output Error

We continue with the quadratic outputs $s^2(\mu,\omega)$ and $\mathbb{M}_2(\mu)$ and start with the bound for the error between the squared output $s^2(\mu,\omega)$ and its approximation $s^{[2]}_{N,K}(\mu,\omega)$ from (6.16). We define the bound $\Delta^{s^2}(\mu,\omega)$ by

$$\begin{aligned}\Delta^{s^2}(\mu,\omega) \;&:=\; \big(\Delta^s(\mu,\omega)\big)^2 \\ &\quad+\; \frac{\beta_{\mathrm{LB}}(\mu,\omega)}{2d(\mu,\omega)}\Delta(\mu,\omega)\tilde{\Delta}^{(2)}(\mu,\omega) \;+\; \delta_{\mathrm{KL}}(p^{(2)}_{N,K}(\mu,\omega);\mu,\omega). \qquad (6.34)\end{aligned}$$

Proposition 6.9. *For $\tau(\mu,\omega)<1$, it holds that $|s^2(\mu,\omega)-s^{[2]}_{N,K}(\mu,\omega)|\le\Delta^{s^2}(\mu,\omega)$.*

Proof. With the definition of $s^{[2]}_{N,K}$ in (6.16), the output error is given by

$$\begin{aligned}s^2-s^{[2]}_{N,K} \;&=\; s^2-(s_{N,K})^2-2s_{N,K}\,r_{\mathrm{RB}}(p^{(1)}_{N,K})+r_{\mathrm{RB}}(p^{(2)}_{N,K}) \\ &=\; (s-s_{N,K})^2+2s_{N,K}(s-s_{N,K})-2s_{N,K}\,r_{\mathrm{RB}}(p^{(1)}_{N,K})+r_{\mathrm{RB}}(p^{(2)}_{N,K}).\end{aligned}$$

Using $s_{N,K}=\ell(u_{N,K})-r_{\mathrm{RB}}(p^{(1)}_{N,K})$ from (6.12) yields

$$2s_{N,K}(s-s_{N,K}) \;=\; 2s_{N,K}\left(\ell(u)-\ell(u_{N,K})+r_{\mathrm{RB}}(p^{(1)}_{N,K})\right).$$

Together, replacing $2s_{N,K}\ell$ by $\ell^{(2)}$, we have

$$s^2-s^{[2]}_{N,K} \;=\; (s-s_{N,K})^2 \;+\; \ell^{(2)}(u)-\ell^{(2)}(u_{N,K})+r_{\mathrm{RB}}(p^{(2)}_{N,K}). \qquad (6.35)$$

From Proposition 6.7, we know that $(s-s_{N,K})^2\le(\Delta^s)^2$. The second part of (6.35) can be estimated analogously to Proposition 6.7 by replacing ℓ by $\ell^{(2)}$ as well as $p^{(1)}$ by $p^{(2)}$ and with Lemma 6.6. We obtain

$$\begin{aligned}&\big|\ell^{(2)}(u)-\ell^{(2)}(u_{N,K})+r_{\mathrm{RB}}(p^{(2)}_{N,K})\big| \\ &=\; \big|g^K(u_{N,K},\tilde{e}^{(2)})+(g-g^K)(u_{N,K},\tilde{e}^{(2)})+(g-g^K)(u_{N,K},p^{(2)}_{N,K})\big| \\ &\le\; |r_{\mathrm{RB}}(\tilde{e}^{(2)})|+\delta_{\mathrm{KL}}(\tilde{e}^{(2)})+\delta_{\mathrm{KL}}(p^{(2)}_{N,K}) \\ &\le\; \frac{\beta_{\mathrm{LB}}}{2d}\Delta\tilde{\Delta}^{(2)}+\delta_{\mathrm{KL}}(p^{(2)}_{N,K})\end{aligned}$$

which proves the claim. □

Since the second moment $\mathbb{M}_2(\mu)$ and its approximation $\mathbb{M}_{2,NK}(\mu)$ defined in (6.17) are just the expectations of $s^2(\mu,\cdot)$ and $s^{[2]}_{N,K}(\mu,\cdot)$, respectively, it is clear

that we can define the bound $\Delta^{\mathbb{M}_2}(\mu)$ by the expectation of $\Delta^{s^2}(\mu,\cdot)$ and omitting again the last term, i.e.,

$$\Delta^{\mathbb{M}_2}(\mu) \;:=\; \mathbb{E}\left[\left(\Delta^{s}(\mu,\omega)\right)^2 \;+\; \frac{\beta_{\mathrm{LB}}(\mu,\omega)}{2d(\mu,\omega)}\Delta(\mu,\omega)\tilde{\Delta}^{(2)}(\mu,\omega)\right]. \tag{6.36}$$

Corollary 6.10. *Under Assumption 6.4, for $\mu \in \mathcal{P}$ and $\tau(\mu,\cdot) < 1$, it holds that $|\mathbb{M}_2(\mu) - \mathbb{M}_{2,NK}(\mu)| \leq \Delta^{\mathbb{M}_2}(\mu)$.*

Proof. From the proof of Proposition 6.9 and Definition (6.17), we know that

$$\begin{aligned}\mathbb{M}_1^2(\mu) - \mathbb{M}_{1,NK}^{[2]}(\mu) = \mathbb{E}\left[(s - s_{N,K})^2\right] + \mathbb{E}\left[g^K(u_{N,K},\tilde{e}^{(2)}) + (g-g^K)(u_{N,K},\tilde{e}^{(2)})\right] \\ + \mathbb{E}\left[(g-g^K)(u_{N,K},p_{N,K}^{(2)})\right].\end{aligned}$$

Since Assumption 6.4 holds, we can apply Lemma 6.5 and the last term vanishes. Following the proof of Proposition 6.9 directly leads to the desired result. □

6.3.6 Variance Output Error

We start with the bound for the error between squared first moment $\mathbb{M}_1^2(\mu)$ and its approximation $\mathbb{M}_{1,NK}^{[2]}(\mu)$. We define the bound $\Delta^{\mathbb{M}_1^2}(\mu)$ by

$$\Delta^{\mathbb{M}_1^2}(\mu) \;:=\; (\Delta^{\mathbb{M}_1}(\mu))^2 + \mathbb{E}\Big[\frac{\beta_{\mathrm{LB}}(\mu,\cdot)}{2d(\mu,\cdot)}\Delta(\mu,\cdot)\tilde{\Delta}^{(3)}(\mu,\cdot)\Big]. \tag{6.37}$$

Proposition 6.11. *Under Assumption 6.4, for $\mu \in \mathcal{P}$ and $\tau(\mu,\cdot) < 1$, it holds that $|\mathbb{M}_1^2(\mu) - \mathbb{M}_{1,NK}^{[2]}(\mu)| \leq \Delta^{\mathbb{M}_1^2}(\mu)$.*

Proof. Analogously to Proposition 6.9, the output error is given by

$$\mathbb{M}_1^2 - \mathbb{M}_{1,NK}^{[2]} \;=\; (\mathbb{M}_1 - \mathbb{M}_{1,NK})^2 \;+\; \mathbb{E}\left[\ell^{(3)}(u) - \ell^{(3)}(u_{N,K}) + r_{\mathrm{RB}}(p_{N,K}^{(3)})\right].$$

From Corollary 6.8, we know $(\mathbb{M}_1 - \mathbb{M}_{1,NK})^2 \leq (\Delta^{\mathbb{M}_1})^2 = (\mathbb{E}\left[\Delta^s\right])^2$. We estimate the remaining term analogously to Proposition 6.7, replacing ℓ by $\ell^{(3)}$ as well as $p^{(1)}$ by $p^{(3)}$ and with Lemma 6.6. Using Lemma 6.5, we obtain

$$\mathbb{E}\left[\ell^{(3)}(u) - \ell^{(3)}(u_{N,K}) + r_{\mathrm{RB}}(p_{N,K}^{(3)})\right] = \mathbb{E}\left[g^K(u_{N,K},\tilde{e}^{(3)}) + (g-g^K)(u_{N,K},\tilde{e}^{(3)})\right]$$

which can be estimated by $\mathbb{E}\big[\frac{\beta_{\mathrm{LB}}}{2d}\Delta\tilde{\Delta}^{(3)}\big]$ analogously to Proposition 6.7. □

From the above results, it is clear that the variance error could directly be bounded by

$$|\mathbb{V}(\mu) - \mathbb{V}_{NK}(\mu)| \;\leq\; \Delta^{\mathbb{M}_2}(\mu) + \Delta^{\mathbb{M}_1^2}(\mu). \tag{6.38}$$

However, we can derive more precise error bounds. Analogously to Section 6.3.1, we define dual RB and KL residuals $\tilde{r}_{\mathrm{RB}}^{(2-3)}(v;\mu,\omega)$ and $\tilde{\delta}_{\mathrm{KL}}^{(2-3)}(v;\mu,\omega)$, replacing $p_{N,K}^{(i)}$ by $(p_{N,K}^{(2)} - p_{N,K}^{(3)})$,

$$\begin{aligned}
\tilde{r}_{\mathrm{RB}}^{(2-3)}(v;\mu,\omega) \;&:=\; dg^K(v, p_{N,K}^{(2)} - p_{N,K}^{(3)})[u_{N,K}] + \ell^{(i)}(v),\\
\tilde{\delta}_{\mathrm{KL}}^{(2-3)}(v;\mu,\omega) \;&:=\; \sum_{q=1}^{Q} |\theta_q(\mu)| \sum_{k=K+1}^{\infty} \xi_{\mathrm{UB}}^{\varrho} \, |dg_{q,k}(v, p_{N,K}^{(2)} - p_{N,K}^{(3)})[u_{N,K}]|.
\end{aligned}$$

The corresponding bounds read

$$\begin{aligned}
\tilde{\Delta}_{\mathrm{RB}}^{(2-3)}(\mu,\omega) \;&:=\; \beta_{\mathrm{LB}}^{-1}(\mu,\omega) \sup_{v\in X} \big(\tilde{r}_{\mathrm{RB}}^{(2-3)}(v;\mu,\omega)/\|v\|_X\big),\\
\tilde{\Delta}_{\mathrm{KL}}^{(2-3)}(\mu,\omega) \;&:=\; \beta_{\mathrm{LB}}^{-1}(\mu,\omega) \sup_{v\in X} \big(\tilde{\delta}_{\mathrm{KL}}^{(2-3)}(v;\mu,\omega)/\|v\|_X\big).
\end{aligned}$$

As a consequence of Proposition 6.3, we obtain

$$\|\tilde{e}^{(2)} - \tilde{e}^{(3)}\|_X \leq \tilde{\Delta}^{(2-3)} \;:=\; 2d\left(\tilde{\Delta}_{\mathrm{RB}}^{(2-3)} + \tilde{\Delta}_{\mathrm{KL}}^{(2-3)}\right) + 2d\frac{\rho_1}{\beta_{\mathrm{LB}}}\Delta\|p_{N,K}^{(2)} - p_{N,K}^{(3)}\|_X.$$

Thus, we can define the variance error bound $\Delta^{\mathbb{V}}(\mu)$ by

$$\Delta^{\mathbb{V}}(\mu) := \mathbb{E}\left[(\Delta^s(\mu,\cdot))^2\right] + (\Delta^{\mathbb{M}_1}(\mu))^2 \;+\; \mathbb{E}\left[\frac{\beta_{\mathrm{LB}}(\mu,\cdot)}{2d(\mu,\cdot)}\Delta(\mu,\cdot)\tilde{\Delta}^{(2-3)}(\mu,\cdot)\right]. \tag{6.39}$$

Proposition 6.12. *Under Assumption 6.4, for $\mu \in \mathcal{P}$ and $\tau(\mu,\cdot) < 1$, it holds that $|\mathbb{V}(\mu) - \mathbb{V}_{NK}(\mu)| \leq \Delta^{\mathbb{V}}(\mu)$.*

Proof. From Propositions 6.9 and 6.11, we know

$$\begin{aligned}
\mathbb{V} - \mathbb{V}_{NK} \;=\; &\mathbb{E}\left[(s - s_{N,K})^2\right] \;-\; (\mathbb{M}_1 - \mathbb{M}_{1,NK})^2\\
&+\; \mathbb{E}\left[\ell^{(2)}(u) - \ell^{(2)}(u_{N,K}) + r_{\mathrm{RB}}(p_{N,K}^{(2)})\right]\\
&-\; \mathbb{E}\left[\ell^{(3)}(u) - \ell^{(3)}(u_{N,K}) + r_{\mathrm{RB}}(p_{N,K}^{(3)})\right]
\end{aligned}$$

and the first two terms can be bounded by $\mathbb{E}\left[(\Delta^s)^2\right]$ and $(\Delta^{\mathbb{M}_1})^2$, respectively. From Lemma 6.6 and the definition of the residual r_{RB}, we know that

$$\ell^{(i)}(u) - \ell^{(i)}(u_{N,K}) + r_{\mathrm{RB}}(p_{N,K}^{(i)}) \;=\; g(u_{N,K}, p^{(i)}) - g^K(u_{N,K}, p_{N,K}^{(i)}), \quad i = 2,3.$$

We subtract the two expressions and follow again the proof of Proposition 6.7. The claim follows directly, using the above definitions and Lemma 6.5. □

6.4 Offline-Online Decomposition

The aim of the RBM are online evaluation procedures of state, outputs and corresponding error bounds independent of the dimension $\mathcal{N}$ of X. In this section, we describe the offline-online decomposition and provide the respective complexities.

For the $\mathcal{N}$-independence, it is of crucial importance to efficiently evaluate the continuity constant $\rho_1(\mu,\omega)$ from (6.2) and the inf-sup constant $\beta_{\mathrm{LB}}(\mu,\omega)$ from (6.23). We start with an evaluation procedure for the continuity constant.

6.4.1 Continuity Constant

The derivation of the continuity constant $\rho_1(\mu,\omega)$ from (6.2) is commonly done using Hölder's inequality and applying the Sobolev embedding theorem [23, 89], where the existence of a so-called Sobolev embedding constant ρ_X with $\|v\|_4 \leq \rho_X\|v\|_X$ for all $v \in X$ is shown. However, the actual derivation of $\rho_1(\mu,\omega)$ depends on the specific form of the trilinear form a_1. Here, we exemplarily provide the derivation strategy for a specific trilinear form that also (but not only) covers the example problem discussed in Section 6.5. Let a_1 be given by

$$a_1(u,w,v;\mu,\omega) \;:=\; \int_D \vec{\nu}(\mu,\omega)\cdot\nabla u w v \;=\; \int_D \nu_1(\mu,\omega)u_x w v + \int_D \nu_2(\mu,\omega)u_y w v,$$

where $\nu(\mu,\omega) : D\times\mathcal{P}\times\Omega \to \mathbb{R}^2$ denotes some parametric spatial stochastic process. For the first part, omitting ν for one moment, we apply Hölder's inequality twice,

$$\begin{aligned}\int_D u_x w v \;&\leq\; \left[\int_D (u_x)^2\right]^{1/2}\left[\int_D (wv)^2\right]^{1/2}\\ &\leq\; \left[\int_D (u_x)^2\right]^{1/2}\left[\int_D (ww)^2\right]^{1/4}\left[\int_D (vv)^2\right]^{1/4}.\end{aligned}$$

Analogously, we estimate the second part. For $\bar{\nu}(\mu,\omega) := \max_{i\in\{1,2\}}\|\nu_i(\mu,\omega)\|_\infty$, we directly obtain the bound $a_1(u,w,v;\mu,\omega) \leq \bar{\nu}(\mu,\omega)(\|u_x\|_2 + \|u_y\|_2)\|w\|_4\|v\|_4$. Using Young's inequality, we can easily show that $\|u_x\|_2+\|u_y\|_2 \leq \sqrt{2}\|u\|_X$. Hence,

$$a_1(u,w,v;\mu,\omega) \;\leq\; \sqrt{2}\bar{\nu}(\mu,\omega)\|u\|_X\|w\|_4\|v\|_4.$$

Now, we apply the Sobolev embedding theorem and obtain the desired continuity constant $\rho_1(\mu,\omega) := \sqrt{2}\bar{\nu}(\mu,\omega)\rho_X^2$. Suppose $\vec{\nu}$ allows for an affine decomposition

in the parameters (μ, ω), it is clear that $\bar{\nu}$ and therefore ρ_1 can be decomposed as well with the same number of affine terms. Hence, the online evaluation of $\rho_1(\mu, \omega)$ can be done efficiently.

It remains to compute the Sobolev embedding constant ρ_X which involves the solution of a nonlinear eigenproblem of the form

$$\int_D \phi^3 v \;=\; \lambda \cdot (\phi, v)_X, \quad \forall v \in X, \quad \|\phi\|_X = 1. \tag{6.40}$$

The solution of (6.40) can be obtained using e.g., fixed point or homotopy procedures [96]. The Sobolev embedding constant ρ_X is then given by $(\lambda_{\max})^{1/4}$. The evaluation can be done offline.

6.4.2 Inf-Sup Constant

For the evaluation of the inf-sup constant, we refer to the successive constraint method (SCM) [57] that can almost directly be applied to the stochastic case. However, due to the KL truncation, we have to subtract a correction term. Let $\beta_{\mathrm{LB}}^K(\mu, \omega)$ be a lower bound of the inf-sup constant with respect to the truncated form $dg^K(w, v; \mu, \omega)[u_{N,K}(\mu, \omega)]$. We furthermore define

$$\Delta_{\mathrm{KL}}^{\beta}(\mu, \omega) \;:=\; \sup_{w \in X} \sup_{v \in X} \left(\sum_{q=1}^{Q} |\theta_q(\mu)| \sum_{k=K+1}^{K_{\max}} \xi_{\mathrm{UB}}^{\varrho} \, \frac{|dg_{q,k}(v, w)[u_{N,K}]|}{\|w\|_X \|v\|_X} \right)$$

and obtain the lower bound (cf. [18], Section 5.7.1)

$$\beta_{\mathrm{LB}}(\mu, \omega) \;:=\; \beta_{\mathrm{LB}}^K(\mu, \omega) - \Delta_{\mathrm{KL}}^{\beta}(\mu, \omega) \;\leq\; \beta(\mu, \omega).$$

In [57], it is shown that the online evaluation of $\beta_{\mathrm{LB}}^K(\mu, \omega)$ is independent of $\mathcal{N}$. However, it involves the solution of a linear program with about $(QKN)^2/2$ degrees of freedom. One can show that the online evaluation of $\Delta_{\mathrm{KL}}^{\beta}(\mu, \omega)$ is of complexity $\mathcal{O}(Q(K_{\max} - K)N)$. The combined offline evaluations for $\beta_{\mathrm{LB}}^K(\mu, \omega)$ and $\Delta_{\mathrm{KL}}^{\beta}(\mu, \omega)$ include $QK_{\max}N$ eigenvalue problems of the full dimension $\mathcal{N}$.

6.4.3 Offline Complexity

To generate the reduced basis, we use a Greedy algorithm as it is well known in the RB context [97, 73]. Suppose we use a training set of n_{train} samples, the basis

selection procedure needs $\mathcal{O}(N \cdot n_{\text{train}})$ times the online run-time. Furthermore, the evaluation of the actual basis is of complexity $\mathcal{O}(IQK_{\text{detail}}N\mathcal{N})$, where I is the number of used Newton iterations, assuming that the detailed computation uses K_{detail} terms of the KL expansion. The complexity to compute the matrices and vectors of the reduced system is $\mathcal{O}(QK_{\max}N^3)$. For the evaluation of the Δ_{KL} and Δ_{RB} error bounds, we evaluate $\mathcal{O}(QK_{\max}N^2)$ Riesz representatives, one for each affine term of the residuals, and its pairwise inner products. Thus, the complexity reads $\mathcal{O}(Q^2K_{\max}^2N^4\mathcal{N})$. We store the reduced system matrices and vectors and the Riesz representative inner products, i.e., the storage complexity is $\mathcal{O}(Q^2K_{\max}^2N^4)$.

6.4.4 Online Complexity

Let us summarize the online run-time complexity to assemble and solve the reduced system for one parameter pair $(\mu, \omega) \in \mathcal{P} \times \Omega$ and to evaluate outputs and error bounds. Let I denote again the number of Newton iterations. In each iteration, we have to assemble and solve the reduced primal system which is of complexity $\mathcal{O}(QKN^3)$ and $\mathcal{O}(N^3)$, respectively. The evaluation of the residuals r_{RB} — needed as correction terms for the outputs — is done in $\mathcal{O}(QKN^3)$ as well. Furthermore, we need to assemble and solve the linear dual problems with complexity $\mathcal{O}(QKN^3 + N^3)$, i.e., the complexity of just one Newton iteration. For the error bounds, we first evaluate β_{LB}, solving a linear program with about $(QKN)^2/2$ degrees of freedom. The evaluation of ρ_1 can be done in $\mathcal{O}(QK)$. For the δ_{KL}-error bounds, we need $Q(K_{\max} - K)$ matrix-vector multiplications, i.e., the complexity is $\mathcal{O}(Q(K_{\max} - K)N^2)$. For the Δ_{KL} and Δ_{RB} error bounds, we have to assemble the inner products of the Riesz representatives with the total complexity $\mathcal{O}\big((Q^2K^2+Q(K_{\max}-K))N^4\big)$, where the Δ_{KL} bounds are evaluated analogously to Section 5.7.2. Hence, the overall complexity reads $\mathcal{O}(IQKN^3) + \mathcal{O}\big((Q^2K^2+Q(K_{\max}-K))N^4\big)$.

The storage complexity is $\mathcal{O}(QK_{\max}N^3)$ for all reduced matrices and vectors, and $\mathcal{O}(Q^2K_{\max}^2N^4)$ for the Riesz inner products. Suppose we use M realizations to evaluate Monte Carlo statistical outputs. Then, we have an additional storage complexity of $\mathcal{O}(M)$ to store certain RB outputs or we need to reevaluate the respective quantities when needed. However, using the less precise variance error bound (6.38), it is possible to evaluate all quantities with an additional storage

complexity of just $\mathcal{O}(1)$. For more details, we refer to Section 5.7.3.

6.5 Numerical Experiment

In this section, we consider a two-dimensional stationary convection-diffusion process in a porous medium. We model the concentration or mass of a physical quantity transported through a wet sandstone. The diffusivity depends on the porosity, modeled by some spatial stochastic process, and the water saturation of the sandstone, given by some deterministic parameter. The nonlinear convective term includes the gradient of the concentration together with a given dominant direction and a scalar intensity factor given by another deterministic parameter.

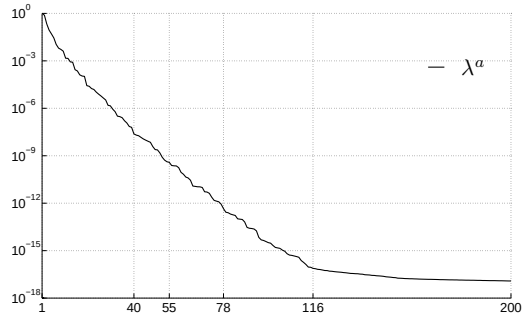
Let $D = (0,1)^2 \subset \mathbb{R}^2$ denote the physical domain of the sandstone and $(\Omega, \mathfrak{A}, \mathbb{P})$ some probability space. The porosity, i.e., the rate of pore space within some control volume, is denoted by the spatial stochastic process $\kappa : D \times \Omega \to [0,1]$ and is assumed to be smooth in space. Furthermore, the global water saturation in the pores is given by $\mu_1 \in [0.05, 1.00]$. Let $\eta_\mathrm{s} = 0.04$ be the diffusivity constant of pure (theoretically imporous) sandstone and $\eta_\mathrm{w} = 3.10$, $\eta_\mathrm{a} = 1.20$ the respective diffusivity constants of water and air. With these notations, the diffusivity of a wet sandstone is assumed to be

$$\begin{aligned} \eta(x;\mu,\omega) &= \eta_\mathrm{s} \cdot (1 - \kappa(x;\omega)) + (\mu_1 \eta_\mathrm{w} + (1-\mu_1)\eta_\mathrm{a})\kappa(x;\omega) \\ &= \eta_\mathrm{s} + (-\eta_\mathrm{s} + \mu_1 \eta_\mathrm{w} + (1-\mu_1)\eta_\mathrm{a})\kappa(x;\omega). \end{aligned} \tag{6.41}$$

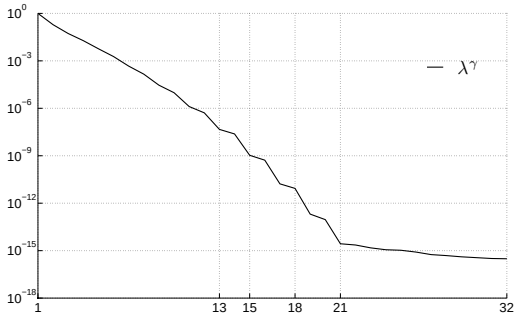
We denote the scaled dominant convection direction by $\vec{\nu}(\mu_2) = \frac{\mu_2}{\sqrt{2}}\binom{1}{1}$, where $\mu_2 \in [0.2, 1.0]$. Finally, we introduce a random zero mean Neumann outlet condition $\gamma(\omega)$ at one part of the boundary. For $\mu := (\mu_1, \mu_2) \in \mathcal{P} := [0.05, 1.00] \times [0.20, 1.00]$, the PDE reads as follows: for given $(\mu, \omega) \in \mathcal{P} \times \Omega$, find $u(\mu, \omega)$ such that

$$\begin{cases} -\nabla \cdot \big(\eta(\mu_1,\omega)\, \nabla u(\mu,\omega)\big) + \vec{\nu}(\mu_2) \cdot \nabla u\, u &= 0 \quad \text{in } D, \\ u(\mu,\omega) &= 0 \quad \text{on } \Gamma_\mathrm{D}, \\ n \cdot \big(\eta(\mu_1,\omega)\, \nabla u(\mu,\omega)\big) &= 0 \quad \text{on } \Gamma_\mathrm{N}, \\ n \cdot \big(\eta(\mu_1,\omega)\, \nabla u(\mu,\omega)\big) &= \gamma(\omega) \quad \text{on } \Gamma_\mathrm{out}. \end{cases} \tag{6.42}$$

In the weak form, this leads to the trilinear form $a_1(w,z,v;\mu) = \int_D \vec{\nu}(\mu_2) \cdot \nabla w\, z\, v$, the bilinear form $a_0(w,v;\mu,\omega) = \int_D \eta(\mu_1,\omega) \nabla w \nabla v$, and the linear form $f(v;\omega) =$



(a) Eigenvalues of the KL expansion of $\tilde{\kappa}$ and KL truncation values $K^{\kappa}=40$, $K^{\kappa}_{\max}=55$, and $K^{\kappa}_{\text{detail}}=78$.



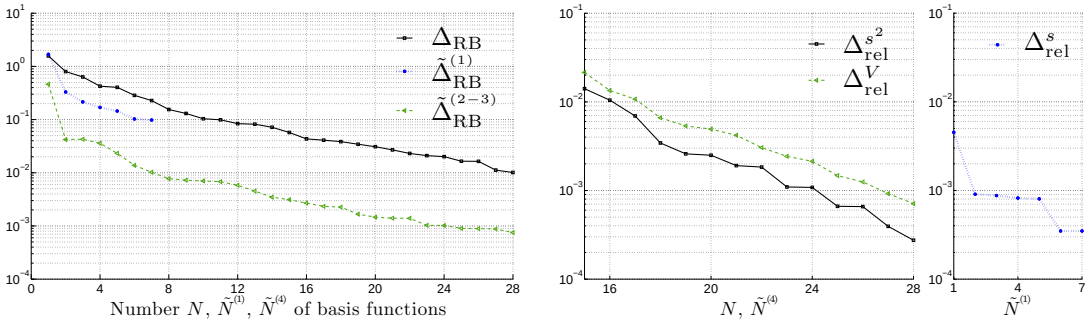
(b) Eigenvalues of the KL expansion of $\tilde{g}$ and KL truncation values $K^{\gamma}=13$, $K^{\gamma}_{\max}=15$, and $K^{\gamma}_{\text{detail}}=18$.

Figure 6.1: Eigenvalues and truncation values of the Karhunen–Loève expansions.

$\int_{\Gamma_{\text{out}}} \gamma(\omega)v$. We define $\theta_1(\mu) := \eta_s$ and $\theta_2(\mu) := -\eta_{\text{s}} + \mu_1\eta_{\text{w}} + (1-\mu_1)\eta_{\text{a}}$ using (6.41), as well as $\theta_3(\mu) := \mu_2$, and $\theta_4(\mu) := 1$. Hence, the affine decompositions with respect to μ of a_0, a_1 and f are given by

$$\begin{aligned} a_0(w,v;\mu,\omega) &= \theta_1(\mu)\int_D \nabla w \nabla v \;+\; \theta_2(\mu)\int_D \kappa(\omega)\nabla w \nabla v, \\ a_1(w,z,v;\mu) &= \theta_3(\mu)\int_D \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}\cdot \nabla w\, z\, v, \\ f(v;\omega) &= \theta_4(\mu)\int_{\Gamma_{\text{out}}} \gamma(\omega)\, v. \end{aligned}$$

As for the numerical example in Chapter 5, let $\bar{\kappa}(x)$ denote the mean of the porosity $\kappa(x;\cdot)$ and $\tilde{\kappa}(x;\omega) := \kappa(x;\omega) - \bar{\kappa}(x)$ its stochastic part with zero mean and the KL expansion $\tilde{\kappa}(x;\omega) = \sum_{k=1}^{K_{\kappa,\max}} \xi^{\kappa}_k(\omega)\kappa_k(x)$, where $\bar{\kappa}(x) \equiv 0.62$ is supposed to be constant in space. We use the same model for $\tilde{\kappa}$ as in the numerical example of Chapter 5. Four random realizations and the first four KL modes of $\tilde{\kappa}$ have already been provided in Figure 5.1 and 5.2, respectively. Analogously, we have the KL expansion for the zero mean outlet given by $\gamma(x;\omega) = \sum_{k=1}^{K_{\gamma,\max}} \xi^{\gamma}_k(\omega)\gamma_k(x)$, where γ is again adopted from Chapter 5. Four random realizations and the first four KL modes of γ have been provided in Figure 5.3 and 5.4, respectively. The KL eigenvalues of $\tilde{\kappa}$ and γ are plotted in Figure 6.1. For the detailed solution, we use $K^{\kappa}_{\text{detail}} = 78$ terms to specify $\tilde{\kappa}$ and $K^{\gamma}_{\text{detail}} = 18$ terms to specify γ. In the reduced setting, $K^{\kappa} = 40$ and $K^{\gamma} = 13$ terms are used, respectively. The error is



(a) Error decay of primal solution u and dual solution $p^{(1)}$, and of $(p^{(2)} - p^{(3)})$

(b) Relative error decay of outputs s, s^2 and $\mathbb{V}$ without δ_{KL}-contributions, respectively

Figure 6.2: Greedy error decay

measured using $K^{\kappa}_{\max} = 55$ and $K^{\gamma}_{\max} = 15$, respectively, such that the additional truncation error is negligible compared to the given error tolerance. In total, the affine decomposition of g with respect to μ and ω consists of $3+K^{\kappa}_{\mathrm{detail}}+K^{\gamma}_{\mathrm{detail}} = 99$ terms, the affine decomposition of dg of $2+K^{\kappa}_{\mathrm{detail}} = 80$ terms, and the respective truncated forms of $3+K^{\kappa}+K^{\gamma} = 44$ and $2+K^{\kappa} = 32$.

The output of interest is assumed to be the average concentration at the "output" boundary Γ_{out}, i.e., for $\ell(v) = \int_{\Gamma_{\mathrm{out}}} v$, we define the output $s(\mu,\omega) := \ell(u(\mu,\omega))$. Furthermore, we are interested in its mean, second moment and variance.

For the detailed solution, we use a finite element space $X \subset \{v \in H^1(D) \,|\, v = 0 \text{ on } \Gamma_D\}$ with linear Lagrange elements and $\mathcal{N} = 3191$ degrees of freedom. For the corresponding H^1-norm $\|\cdot\|_X$, we evaluate the Sobolev embedding constant $\rho_X = \sup_{v\in X} \|v\|_4/\|v\|_X$ as described in Section 6.4.1 and obtain $\rho_x = 0.60077$.

For the basis construction, we use a greedy algorithm such that $\tilde{X}^{(2)}_N = \tilde{X}^{(3)}_N$. Figure 6.2(a) shows the decay of the maximal RB error bounds of the primal and dual solutions u and $p^{(1)}$ as well as the difference of the additional dual solutions $p^{(2)}-p^{(3)}$. For $(N, \tilde{N}^{(1)}, \tilde{N}^{(2)}) = (28, 7, 28)$, the error bounds of the desired outputs fall below the given tolerance $tol = 10^{-3}$ for all (μ,ω) in the training sample. The decay of the output error bounds is provided in Figure 6.2(b), omitting the δ_{KL}-parts that do not decrease in N and are therefore ignored in the greedy procedure. We simultaneously created X_N and $\tilde{X}^{(2)}_N$ that are used for the reduced solution of s^2 and $\mathbb{V}$, assuming that $\tilde{N}^{(1)}$ is already large enough such that the terms $(\Delta^s)^2$

	average error bound	factor
simple	$5.193 \cdot 10^{-3}$	$55.85 \cdot \Delta^{\mathbb{V}}$
sophisticated	$9.366 \cdot 10^{-4}$	$10.07 \cdot \Delta^{\mathbb{V}}$
$\Delta^{\mathbb{V}}$	$9.299 \cdot 10^{-5}$	$1.000 \cdot \Delta^{\mathbb{V}}$

Table 6.1: Comparison of different variance error bounds for a test set of 256 parameters, using 10.000 random samples for each parameter.

and $(\Delta^{\mathbb{M}_1})^2$ in the respective error bounds Δ^{s^2} and $\Delta^{\mathbb{V}}$ are sufficiently small. For $N \geq 15$ primal basis functions, we obtained $\tau < 1$ for all (μ, ω) in the training set. Then, we created $\tilde{X}_N^{(1)}$ such that Δ^s and $\Delta^{\mathbb{M}_1}$ indeed become sufficiently small. Since N was already large, only a small number of $\tilde{N}^{(1)} = 7$ basis functions were needed.

To compare detailed and reduced solutions, we used a 3.06 GHz Intel Core 2 Duo processor, 4 GB RAM. We used MATLAB 7.9.0 (R2012a) to run reduced simulations and MATLAB 7.9.0 with the link to Comsol 3.5a for the detailed computations. For the (rather small) detailed system with $\mathcal{N} = 3.191$, we already achieved a speedup factor of about 26 from full to reduced simulations, where in the reduced case, the evaluations of all error bounds are included. Tests with finer meshes and hence larger $\mathcal{N}$ for the full solutions showed that the desired error tolerance can still be reached with the same numbers of basis functions. E.g., for $\mathcal{N} = 12.555$ and $(N, \tilde{N}^{(1)}, \tilde{N}^{(2)}) = (28, 7, 28)$, the speedup factor was about 78.

In Table 6.1, we compare the presented method to evaluate variances $\mathbb{V}_{NK}$ and the error bound $\Delta^{\mathbb{V}}$ with two alternative procedures. Neither of the two needs additional dual problems. The simplest method just uses the estimations

$$|s^2 - (s_{N,K})^2| = |(s - s_{N,K})(s + s_{N,K})| \leq \Delta^s(\Delta^s + 2|s_{N,K}|)$$

and analogously $|\mathbb{M}_1^2 - (\mathbb{M}_{1,NK})^2| \leq \Delta^{\mathbb{M}_1}(\Delta^{\mathbb{M}_1} + 2|\mathbb{M}_{1,NK}|)$. For the more sophisticated method, we refer to [12] or Appendix A. Both methods already use the good approximations and bounds for $s(\mu, \omega)$ and $\mathbb{M}_1(\mu)$ from Section 6.3.4. We see that our variance evaluation and the error bounds produce much sharper results. Compared to the "simple" method, the bound is about 56 times smaller, compared to the "sophisticated" method, it still is more than 10 times smaller. The costs, on

the other hand, increase only moderately. The evaluation of the additional dual problem (6.22) corresponds to just one Newton iteration of the primal problem.

Chapter 7

Application of the RBM to PDEs on Stochastic Domains

For the quality of the solution of a given PDE, it is essential to adequately model the underlying domain. In many cases, however, the description of the domain is obtained by imperfect or defective measurements, e.g., by scanning or X-raying. Further digital image processing to detect the boundaries may yield further errors (cf., e.g., [5, 68], [100, Ch. 5]). Also, deviations from the description of a domain to the mechanical implementation can play an important role, e.g., in very sensitive aerodynamic systems. Hence, it may be of interest to investigate how perturbations of the boundary affect the solution of the PDE, also to define tolerances for the actual production process of mechanical systems. Furthermore, for some applications, the boundary of a domain is directly modeled using stochastic parameters. E.g., in bone fracture healing simulations, the shape of the bones could be modeled stochastically [81, 99]. The stochastic description of the bones could be obtained using sample data and the method of snapshots (cf. Section 2.2).

Besides straightforward but expensive Monte Carlo procedures, several techniques to solve PDEs on stochastic domains have been developed. E.g., in [17], a "fictitious domain" is used that encloses the the stochastic domain. The PDE is solved on the larger domain and the boundary conditions of the stochastic domain are modeled using Lagrange multipliers. This yields a saddle-point formulation, where the stiffness matrix is independent of the stochasticity. A different approach has been introduced for example in [49, 50]. The variation of the random bound-

ary, compared to its known mean, is assumed to be small. For a given two-point correlation function, deterministic formulations for mean, variance, and correlation functions of the solution and output functionals, respectively, are derived, using "second order shape calculus".

In this chapter, we follow an ansatz that has already been used for both stochastic boundaries [84, 104] and also for parametric domains [63, 87], in the latter references already in the context of RBMs. The random and/or parametric domain is projected to a fixed and deterministic reference domain, using a bijective mapping. Then, the weak formulation of the PDE on the original domain is transformed to the reference domain using the change of variables formula, i.e., the spatial variable is substituted by the mapping from the reference domain. The stochasticity and/or parametric dependencies are thereby shifted into the coefficients of the PDE. For the detailed solution, Monte Carlo approaches or stochastic Galerkin methods can now be employed as described in Chapter 2.

The chapter is organized as follows. In Section 7.1, we provide an exemplary PDE to explain the projection of a boundary value problem to a reference domain and to introduce the requirements for the domain mapping function. Two different procedures to construct such bijective mappings for parametric and stochastic boundaries are introduced in Section 7.2, the Laplace equation based mapping [85, 104] and the transfinite element mapping [36, 37]. In Section 7.3, we discuss the problem of preserving of affine decompositions from the original to the mapped weak formulation. For the special case of stochastic but non-parametric domains, we show in Section 7.4 how the RBMs from Chapter 5 and 6 can be applied. For stochastic *and* parametric domains, alternative RBM procedures and their requirements are briefly described in Section 7.5. In Section 7.6, we provide numerical examples for both cases. We demonstrate that we can also use the IPMs from Chapter 4 to decrease the online costs for both non-parametric and parametric, stochastic boundaries.

7.1 Preliminaries

We start with the formulation of the exemplary problem that will be used throughout the whole chapter. For now, we do not specify any parametric or stochastic

dependence of the domain or any other quantity.

7.1.1 Model Problem

Let $\tilde{D} \subset \mathbb{R}^d$, $d \in \mathbb{N}$, be an arbitrary bounded spatial domain. In the context of stochastic and parametric domains, $\tilde{D}$ can be seen as a random realization for a given parameter. However, for the moment, we do not explicitly indicate such a dependence. Let $\tilde{a}$ and $\tilde{g} \in L_2(\tilde{D})$ be real valued, bounded functions on $\tilde{D}$ and $\tilde{b}, \tilde{c} : \tilde{D} \to \mathbb{R}^d$ functions in $(L_2(\tilde{D}))^d$, where each component is bounded as well. Furthermore, let $\partial\tilde{D}$ denote the boundary of $\tilde{D}$ with the boundary segments $\tilde{\Gamma}_{\mathrm{D}}$ and $\tilde{\Gamma}_{\mathrm{N}}$, $\tilde{\Gamma}_{\mathrm{D}} \cup \tilde{\Gamma}_{\mathrm{N}} = \partial\tilde{D}$, $\tilde{\Gamma}_{\mathrm{D}} \cap \tilde{\Gamma}_{\mathrm{N}} = \emptyset$. We define the real valued function $\tilde{h} \in L_2(\tilde{\Gamma}_{\mathrm{N}})$ on the segment $\tilde{\Gamma}_{\mathrm{N}}$ to describe a Neumann boundary outlet condition of the following quadratically nonlinear, stationary PDE,

$$\begin{cases} -\nabla \cdot \big(\tilde{a}(\tilde{x})\nabla\tilde{u}(\tilde{x})\big) + \tilde{b}(\tilde{x}) \cdot \nabla\tilde{u}(\tilde{x}) + \tilde{c}(\tilde{x}) \cdot \nabla\tilde{u}(\tilde{x})\tilde{u}(\tilde{x}) &= \tilde{g}(\tilde{x}),\ \tilde{x} \in \tilde{D}, \\ \tilde{u}(\tilde{x}) &= 0,\ \tilde{x} \in \tilde{\Gamma}_{\mathrm{D}}, \\ \tilde{n}(\tilde{x}) \cdot \big((\tilde{a}(\tilde{x})\ \nabla\tilde{u}(\tilde{x})\big) &= \tilde{h}(\tilde{x}),\ \tilde{x} \in \tilde{\Gamma}_{\mathrm{N}}, \end{cases} \tag{7.1}$$

where $\tilde{n}(\tilde{x})$ denotes the outward normal on $\tilde{\Gamma}_{\mathrm{N}}$. For some appropriate Hilbert space $\tilde{X} = X(\tilde{D}) \subset H^1(\tilde{D})$, accounting also for the boundary conditions, the weak form of (7.1) can be formulated in the following way: find $\tilde{u} \in \tilde{X}$ such that

$$\tilde{a}_0(\tilde{u}, \tilde{v}) + \tilde{a}_1(\tilde{u}, \tilde{u}, \tilde{v}) = \tilde{f}(\tilde{v}), \quad \tilde{v} \in \tilde{X}, \tag{7.2}$$

where, for $\tilde{u}, \tilde{w}, \tilde{v} \in \tilde{X}$, we used

$$\begin{aligned} \tilde{a}_0(\tilde{u}, \tilde{v}) &:= \int_{\tilde{D}} \tilde{a}(\tilde{x})\nabla\tilde{u}(\tilde{x}) \cdot \nabla\tilde{v}(\tilde{x})d\tilde{x} + \int_{\tilde{D}} \tilde{b}(\tilde{x}) \cdot \nabla\tilde{u}(\tilde{x})\tilde{v}(\tilde{x})d\tilde{x}, \\ \tilde{a}_1(\tilde{u}, \tilde{w}, \tilde{v}) &:= \int_{\tilde{D}} \tilde{c}(\tilde{x}) \cdot \nabla\tilde{u}(\tilde{x})\tilde{w}(\tilde{x})\tilde{v}(\tilde{x})d\tilde{x}, \\ \tilde{f}(\tilde{v}) &:= \int_{\tilde{D}} \tilde{g}(\tilde{x})\tilde{v}(\tilde{x})d\tilde{x} + \int_{\tilde{\Gamma}_{\mathrm{N}}} \tilde{h}(\tilde{x})\tilde{v}(\tilde{x})d\tilde{x}. \end{aligned}$$

7.1.2 Projection to a Reference Domain

We now show how the problem can be projected to an appropriate reference domain $D \subset \mathbb{R}^d$ with Lipschitz boundary. Let $T : D \to \tilde{D}$ be a diffeomorphism, i.e., a bijective and continuously differentiable mapping, where the inverse function

$T^{-1} : \tilde{D} \to D$ is also continuously differentiable. We denote the Jacobian matrix of T by $J_T : D \to \mathbb{R}^{d\times d}$,

$$J_T(x) \;=\; \begin{pmatrix} \frac{\partial T_1}{\partial x_1} & \dots & \frac{\partial T_1}{\partial x_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial T_d}{\partial x_1} & \dots & \frac{\partial T_d}{\partial x_d} \end{pmatrix}(x),$$

where T_i, $i \in \{1, \dots, d\}$, denotes the i-th component of the mapping T. Furthermore, the determinant of the Jacobian matrix is denoted by $\det J_T(x)$ and is called Jacobian determinant. In literature, $\det J_T(x)$ is also just referred to as the Jacobian. However, to avoid confusions, we keep using the full expression. We can bijectively extend T on the boundary of D, i.e., we have $\partial\tilde{D} = T(\partial D)$, and we define the boundary segments $\Gamma_{\mathrm{D}} := T^{-1}(\tilde{\Gamma}_{\mathrm{D}})$ and $\Gamma_{\mathrm{N}} := T^{-1}(\tilde{\Gamma}_{\mathrm{N}})$.

Let us proceed with the transformation of the coefficient functions of the PDE. For any $\mathfrak{c} \in \{a, b, c, g, h\}$, we define $\mathfrak{c}(x) := \tilde{\mathfrak{c}}(T(x))$. Since T is a diffeomorphism, we can assume that the characteristics of the transformed coefficients are preserved, e.g., for $\tilde{\mathfrak{c}} \in L_2(\tilde{D})$, we assume that $\mathfrak{c} \in L_2(D)$. Furthermore, for parametric and/or stochastic coefficients that allow for an affine decomposition, the affinity is also maintained. However, the orthogonality, if given, of the spatial terms in the affine decomposition is not preserved.

For any function $\tilde{\mathfrak{u}} \in \{\tilde{u}, \tilde{w}, \tilde{v}\}$, $\tilde{\mathfrak{u}} \in \tilde{X} = X(\tilde{D})$, we define $\mathfrak{u} = \tilde{\mathfrak{u}}(T(x))$ analogously to the coefficient functions of the PDE. Again, it is clear that $\mathfrak{u} \in X := X(D) \subset H^1(D)$ since T is a diffeomorphism. Suppose higher derivatives of order $r \in \mathbb{N}$ would be necessary for the PDE in weak form, we would have to require T to be a C^r-diffeomorphism to guarantee that the regularity of $\tilde{\mathfrak{u}} \in \tilde{X}$ is preserved by the transformation. In other words, T and T^{-1} would be required to be r times continuously differentiable.

Let now $\nabla_{\tilde{x}}$ denote the gradient with respect to the spatial variable $\tilde{x} \in \tilde{D} \subset \mathbb{R}^d$ on the original domain, and let ∇_x denote the gradient with respect to the spatial variable $x \in D$ on the reference domain. Using the chain rule, the gradient of $\mathfrak{u} \in X$ is given by

$$\nabla_x \mathfrak{u}(x) = \nabla_x \tilde{\mathfrak{u}}(T(x)) = \left(\sum_{j=1}^{d} \frac{\partial T_j}{\partial x_i}(x) \, \frac{\partial \tilde{\mathfrak{u}}}{\partial \tilde{x}_j}\big(T(x)\big) \right)_{i=1,\dots,d}$$

$$
\begin{aligned}
&= \begin{pmatrix} \frac{\partial T_1}{\partial x_1} & \dots & \frac{\partial T_d}{\partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial T_1}{\partial x_d} & \dots & \frac{\partial T_d}{\partial x_d} \end{pmatrix}(x) \begin{pmatrix} \frac{\partial \tilde{\mathfrak{u}}}{\partial \tilde{x}_1} \\ \vdots \\ \frac{\partial \tilde{\mathfrak{u}}}{\partial \tilde{x}_d} \end{pmatrix}\big(T(x)\big) \\
&= J_T^T(x) \nabla_{\tilde{x}} \tilde{\mathfrak{u}}(T(x)),
\end{aligned}
$$

where $J_T^T(x)$ denotes the transposed Jacobian matrix of the mapping T. Since T is bijective, $J_T(x)$ is a regular matrix for all $x \in D$ and we obtain

$$
\nabla_{\tilde{x}} \tilde{\mathfrak{u}}(T(x)) = J_T^{-T}(x) \nabla_x \mathfrak{u}(x),
$$

where $J_T^{-T}(x)$ denotes the transposed inverse Jacobian matrix of the mapping T. Now, we can map the weak formulation of the PDE into the reference domain. Using integration by substitution, i.e., integration by substitution, we obtain

$$
\begin{aligned}
\int_{\tilde{D}} \tilde{a}(\tilde{x}) \nabla_{\tilde{x}} \tilde{u}(\tilde{x}) \cdot \nabla_{\tilde{x}} \tilde{v}(\tilde{x}) d\tilde{x} &= \int_D \tilde{a}(T(x)) \nabla_{\tilde{x}} \tilde{u}(T(x)) \cdot \nabla_{\tilde{x}} \tilde{v}(T(x)) |\det J_T(x)| dx \\
&= \int_D \big(\big(|\det J_T| J_T^{-1} J_T^{-T} \big)(x)\, a(x) \nabla_x u(x) \big) \cdot \nabla_x v(x) dx.
\end{aligned}
$$

Analogously, we get

$$
\begin{aligned}
\int_{\tilde{D}} \tilde{b}(\tilde{x}) \cdot \nabla_{\tilde{x}} \tilde{u}(\tilde{x}) \tilde{v}(\tilde{x}) d\tilde{x} &= \int_D \big(\big(|\det J_T| J_T^{-1} \big)(x)\, b(x) \big) \cdot \nabla_x u(x)\, v(x) dx, \\
\int_{\tilde{D}} \tilde{c}(\tilde{x}) \cdot \nabla_{\tilde{x}} \tilde{u}(\tilde{x}) \tilde{u}(\tilde{x}) \tilde{v}(\tilde{x}) d\tilde{x} &= \int_D \big(\big(|\det J_T| J_T^{-1} \big)(x)\, b(x) \big) \cdot \nabla_x u(x)\, u(x)\, v(x) dx, \\
\int_{\tilde{D}} \tilde{g}(\tilde{x}) \tilde{v}(\tilde{x}) d\tilde{x} &= \int_D |\det J_T(x)| g(x) v(x) dx.
\end{aligned}
$$

Since T is assumed to be bijectively extendable on the boundary of D, we can also transform the boundary integral

$$
\int_{\tilde{\Gamma}_{\mathrm{N}}} \tilde{h}(\tilde{x}) \tilde{v}(\tilde{x}) d\tilde{x} = \int_{\Gamma_{\mathrm{N}}} |\det J_T(x)| h(x) v(x) dx.
$$

In order to abbreviate the notation, we define

$$
a_T(x) := |\det J_T(x)| J_T^{-1}(x) J_T^{-T}(x) a(x) \in \mathbb{R}^{d \times d}, \tag{7.3a}
$$

$$
b_T(x) := |\det J_T(x)| J_T^{-1}(x) b(x) \in \mathbb{R}^d, \tag{7.3b}
$$

$$
c_T(x) := |\det J_T(x)| J_T^{-1}(x) c(x) \in \mathbb{R}^d, \tag{7.3c}
$$

$$
g_T(x) := |\det J_T(x)| g(x) \in \mathbb{R}, \tag{7.3d}
$$

$$
h_T(x) := |\det J_T(x)| h(x) \in \mathbb{R}, \tag{7.3e}
$$

and obtain the transformed forms

$$a_0(u,v) := \int_D a_T(x)\nabla_x u(x)\cdot\nabla_x v(x)dx + \int_D b_T(x)\cdot\nabla_x u(x)v(x)dx, \tag{7.4a}$$

$$a_1(u,w,v) := \int_D c_T(x)\cdot\nabla_x u(x)w(x)v(x)dx, \tag{7.4b}$$

$$f(v) := \int_D g_T(x)v(x)dx + \int_{\Gamma_N} h_T(x)v(x)dx. \tag{7.4c}$$

Hence, problem (7.2), projected on the reference domain D, reads: find $u \in X$ such that

$$a_0(u,v) + a_1(u,u,v) = f(v), \quad v \in X. \tag{7.5}$$

7.2 Construction of the Domain Mapping

Let $\mathcal{P} \subset \mathbb{R}^P$ be a set of deterministic parameters and $(\Omega, \mathfrak{A}, \mathbb{P})$ a probability space. In this section, we briefly describe two methods to construct bijective mappings $T : D \to \tilde{D}$ for parametric and stochastic domains $\tilde{D} = \tilde{D}(\mu,\omega)$, $(\mu,\omega) \in \mathcal{P}\times\Omega$. We define the randomness and the parametric dependence of the domain $\tilde{D}(\mu,\omega)$ by its boundary $\partial\tilde{D}(\mu,\omega)$. We assume the existence of a random parametrized function $\rho : \partial D \times \mathcal{P} \times \Omega \mapsto \mathbb{R}^d$ on the boundary of the reference domain D such that $\rho(\partial D;\mu,\omega) = \partial\tilde{D}(\mu,\omega)$. Furthermore, we assume that ρ is already affine in the deterministic parameter, i.e,

$$\rho(x;\mu,\omega) = \sum_{q=1}^{Q} \theta_q(\mu)\big[\rho_{q,0}(x) + \rho_q(x;\omega)\big], \tag{7.6}$$

where $\rho_{q,0}$ denote the expectations of the terms in brackets and $\rho_q(\cdot;\omega)$ the respective fluctuating parts. Hence, the mean of $\rho(x;\mu,\cdot)$, denoted by $\mathbb{E}[\rho(\partial D;\mu,\cdot)] = \sum_{q=1}^{Q}\theta_q(\mu)\rho_{q,0}(x)$, defines the expected boundary of $\tilde{D}(\mu;\cdot)$, i.e.,

$$\mathbb{E}\left[\partial\tilde{D}(\mu,\cdot)\right] = \mathbb{E}\left[\rho(\partial D;\mu,\cdot)\right].$$

For non-parametric stochastic domains, the natural choice of the reference domain D would be the expectation of $\tilde{D}(\cdot)$, i.e., $\mathbb{E}[\rho(x;\cdot)] = x$. In the parametric, stochastic case, the expectation for a certain parameter $\bar{\mu} \in \mathcal{P}$ may be appropriate. However, for many application, it is also desirable to select a reference domain that is as simple as possible, e.g., squares, cubes, or hypercubes.

For each term $\rho_q(x;\omega)$ in (7.6), we assume the existence of a Karhunen–Loève expansion. Hence, ρ is given by

$$\rho(x;\mu,\omega) = \sum_{q=1}^{Q} \theta_q(\mu) \left[\rho_{q,0}(x) + \sum_{k=1}^{K_q^{\text{detail}}} \xi_{q,k}(\omega)\rho_{q,k}(x) \right], \tag{7.7}$$

where the KL sum has already been truncated at the values K_q^{detail}, $q = 1,\dots,Q$, that are sufficiently large to adequately represent the boundary for the detailed simulations.

7.2.1 Laplace Equation Based Mapping

In this section, we construct a mapping based upon solutions of the Laplace equation. We follow the description of [104]. More details can be found in [85]. For each affine term $\rho_{q,k}$, $q = 1,\dots,Q$, $k = 0,\dots,K_q^{\text{detail}}$, of (7.7), we solve a linear boundary value problem on the reference domain, where the boundary conditions are given by $\rho_{q,k}$, i.e.,

$$\begin{aligned} \Delta T_{q,k}(x) &= 0, \quad x \in D, \\ T_{q,k}(x) &= \rho_{q,k}(x), \quad x \in \partial D, \end{aligned} \tag{7.8}$$

In fact, the boundary value problem (7.8) is solved component wise, i.e., we have to solve the number of $Q(K^{\text{detail}}+1)d$ PDEs, where the dependence of K^{detail} on q has been omitted for notational convenience. However, the operator of the system does not change for the different boundary conditions. Hence, the solutions of (7.8) can be evaluated very efficiently. Furthermore, it is not necessary to use the same fine grid as for the detailed solutions. The only requirement is that the boundary of the reference domain is sufficiently well discretized such that the boundary conditions for all affine terms of (7.7), i.e., for all random and parameter samples, can be approximated adequately.

Due to the linearity of (7.8), we obtain affine decompositions of both T and J_T,

$$T(x;\mu,\omega) = \sum_{q=1}^{Q} \theta_q(\mu) \left[T_{q,0}(x) + \sum_{k=0}^{K_q^{\text{detail}}} \xi_{q,k}(\omega)T_{q,k}(x) \right], \tag{7.9}$$

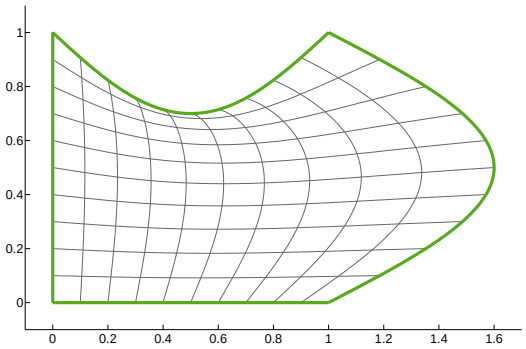
$$J_T(x;\mu,\omega) = \sum_{q=1}^{Q} \theta_q(\mu) \left[J_{T\,q,0}(x) + \sum_{k=0}^{K_q^{\text{detail}}} \xi_{q,k}(\omega)J_{T\,q,k}(x) \right], \tag{7.10}$$

where $J_{T\,q,k}(x)$ denotes the Jacobian matrix of $T_{q,k}(x)$. Even though (7.9) is of the same form as (7.7) and we still have $\mathbb{E}[\xi_{q,k}] = 0$, the sums over k do not specify KL decompositions. As opposed to the terms $\rho_{q,k}$, $k \geq 1$, we do not have orthogonality of the functions $T_{q,k}$, $k \geq 1$. Certainly, the correlation matrices of the terms in brackets can be easily evaluated and a new KL expansion of the respective terms of T can be obtained (cf. Section 2.2). However, as we can see in (7.3), also affine decompositions for more complicated terms are necessary. We will go into more details in Sections 7.4 and 7.5.

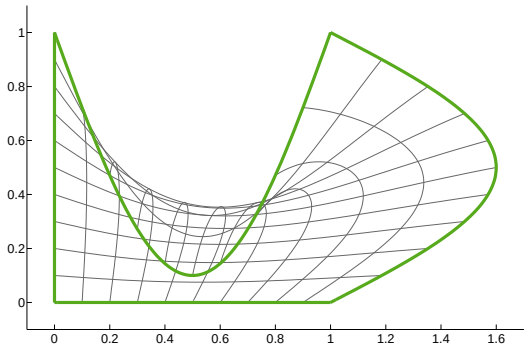
Using the weak form of (7.8), it is clear that the mapping components $T_{q,k}$ are differentiable. However, there is no guarantee that T is a diffeomorphism, i.e., that T is invertible. Often, this is only the case for small perturbations of the random and/or parametric boundary. Nevertheless, it is possible to verify that the inverse mapping exists a posteriori, using the Jacobian determinant. It is sufficient that $|\det J_T(x)| \neq 0$ for all $x \in D$. Since T is assumed to be continuously differentiable, this means that no change of the sign of $\det J_T(x)$ is permitted on D.

Figures 7.1(a) and 7.1(b) show the result of the Laplace equation based mapping for $D = [0,1]^2$ and two different deformations similar to [36, Fig. 8(a)]. The distortions of the upper and right boundary are given by sine functions with different amplitudes, respectively. It can be seen that the mapping is suitable for the small deformation, whereas the large deformations yields a degenerated result. This can also be confirmed by the sign of the respective Jacobian determinants of J_T, provided in Figures 7.1(c) and 7.1(d). Green color indicates a positive sign and black color a negative sign. Since the sign in Figure 7.1(c) is constantly positive, the mapping from Figure 7.1(a) is invertible. Conversely, the mapping from Figure 7.1(b) is not bijective since the sign of $\det J_T$ changes on D. Since T is continuously differentiable, $\det J_T = 0$ on the border between the green an the black area of Figure 7.1(d).

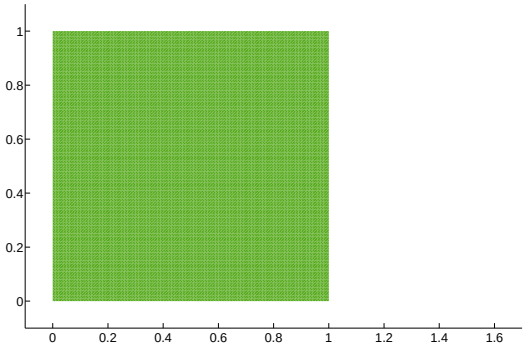
An advantage of the Laplace equation based mapping is its flexibility with respect to the reference domain that admits arbitrary shapes. Furthermore, it is easily possible to divide the domain into several parts and to solve (7.8) separately on each subdomain. Certainly, the boundary conditions have to be chosen such that the mapping matches on the inner boundaries. In this way, it is possible to enforce values of the mapping in the inner parts of the domain. However, it is not



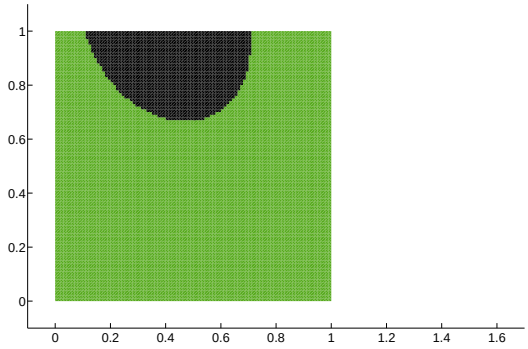
(a) Mapping T for a small deformation.



(b) Mapping T for a large deformation.



(c) Sign of the Jacobian determinant $\det J_T$ for the mapping in Figure 7.1(a).



(d) Sign of the Jacobian determinant $\det J_T$ for the mapping in Figure 7.1(b).

Figure 7.1: Two Laplace equation based mapping results for $D = [0, 1]^2$.

clear if the mapping is still differentiable at the intersections of the subdomains.

In the next section, a method is introduced that generates such mappings, i.e., it is possible to enforce the mapping T to take certain values at the inner parts, still keeping the differentiability.

7.2.2 Transfinite Element Mapping

The subsequent domain mapping has been introduced in the publications of Gordon and Hall [36, 37]. Let $\tilde{D}(\mu, \omega)$ be a parametrized and stochastic domain in $\mathbb{R}^d$. In [36, 37], only the cases $d \in \{2, 3\}$, are considered. Here, we additionally provide the formulation of the mapping for arbitrary dimensions $d \in \mathbb{N}$.

The transfinite element mapping is again based upon the availability of a bound-

ary mapping function $\rho : \partial D \times \mathcal{P} \times \Omega \to \mathbb{R}^d$ such that $\rho(\partial D; \mu, \omega) = \partial \tilde{D}(\mu, \omega)$. It is assumed that the reference domain is given by the unit hypercube of dimension d, i.e., $D = [0,1]^d$. The parametrically stochastic mapping $T(\mu, \omega) : D \to \tilde{D}(\mu, \omega)$ is constructed such that $T(\partial D; \mu, \omega) = \rho(\cdot; \mu, \omega)$. Furthermore, additional hyperplanes can be defined, called "constant generalized coordinates" [36], where T is required to take specific predetermined values. The hyperplanes, i.e., lines for $d = 2$ or planes for $d = 3$, are orthogonal to the coordinate axes. The values of the mapping on the hyperplanes may depend on the current realization of the boundary $\rho(\mu, \omega)$, $(\mu, \omega) \in \mathcal{P} \times \Omega$.

If a given random and parametrized domain can be decomposed into several parts with different properties, e.g., material properties, it is then possible to use this method to fix the boundaries between these different parts at a constant location in the reference domain. On the other hand, the method can be useful for the construction of bijective mappings on complicated domains, where a one-to-one correspondence is hard to achieve.

Let us now introduce the ingredients of the transfinite element mapping. We start with the mathematical description of the constant generalized coordinates. Next, we define so-called blending functions that are used to define d projectors P_q, $q = 1, \ldots, d$, that propagate the deformation of the boundary segments and generalized coordinates along the x_q-axis. Finally, the mapping is constructed as a combination of these projectors.

Constant Generalized Coordinates. We denote a constant generalized coordinate, i.e., the hyperplane orthogonal to a specific coordinate axes, by the intersection point with the corresponding coordinate axis. E.g., for a two-dimensional domain, a line parallel to the x_2-axis and through the point $(\bar{x}_1, 0)$, given by the set $\{x = (x_1, x_2) \in D \,|\, x_1 = \bar{x}_1\}$, is just denoted by $\bar{x}_1$. We partition the x_q-axis with points $\bar{x}_q^{(i)}$, $i = 1, \ldots, I_q - 1$, $I_q \in \mathbb{N}$, such that

$$0 < \bar{x}_q^{(1)} < \ldots < \bar{x}_q^{(I_q - 1)} < 1,$$

and obtain $I_q - 1$ generalized coordinate lines orthogonal to the x_q-axis, additional to the two boundary segments $\bar{x}_q^{(0)} = 0$ and $\bar{x}_q^{(I_q)} = 1$.

In the following, we assume the knowledge of the boundary mapping function $\rho(x; \mu, \omega)$ not only on the boundary of D but also on the generalized coordinates.

We require that T coincides with ρ on all generalized coordinates, i.e., for all points on a general coordinate line $\bar{x}_q^{(i)}$, we require

$$T(x_1, \ldots, \bar{x}_q^{(i)}, \ldots, x_d; \mu, \omega) = \rho(x_1, \ldots, \bar{x}_q^{(i)}, \ldots, x_d; \mu, \omega)$$

for all $i = 0, \ldots, I_q$, $q = 1, \ldots, d$. For $I_1 = \ldots = I_d = 1$, we would obtain the usual condition that ρ coincides with T just on the boundary of the reference domain, $T(\partial D; \mu, \omega) = \rho(\cdot; \mu, \omega)$.

In most practical applications, the values of ρ on the additional hyperplanes depend only on few values of ρ on the boundary of D. The determination of the location of the hyperplanes and the values of the mapping clearly depends on the current problem and the desired properties of the mapping. Thus, we do not go into more detail about good choices of $\bar{x}_q^{(i)}$ at this step. For more information, see [36] and the examples below.

Blending Functions. To interpolate between the generalized coordinates, we define so-called blending functions. For each coordinate direction x_q, $q = 1, \ldots, d$, and for each boundary segment or constant generalized coordinate $\bar{x}_q^{(i)}$, $i = 0, \ldots, I_q$, we define an interpolating function $\varphi_q^{(i)} : \mathbb{R} \to \mathbb{R}$ as a function in x_q. The functions $\varphi_q^{(i)}$ take the value one at the respective generalized coordinate $\bar{x}_q^i$ and zero at the other generalized coordinates orthogonal to the x_q-axis. In other words,

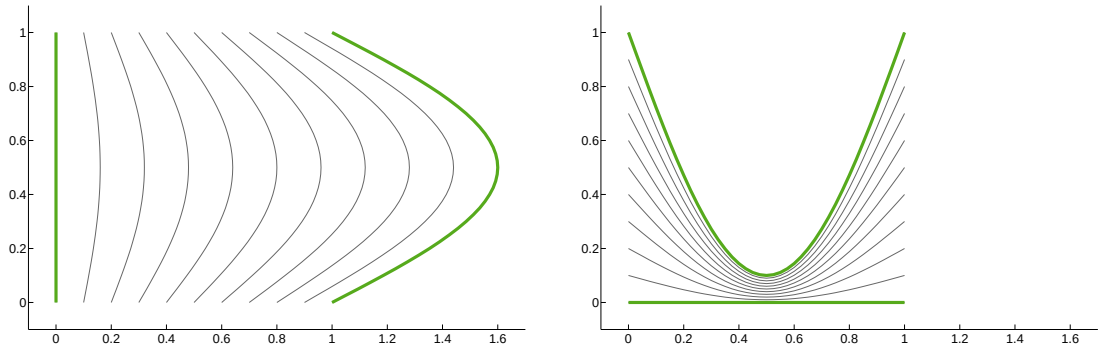
$$\varphi_q^{(i)}(\bar{x}_q^{(j)}) \;=\; \delta_{ij}, \quad i, j = 0, \ldots, I_q, \quad q = 1, \ldots, d.$$

The blending functions are independent of μ and ω, i.e., of the current realization of the boundary. We can use for example the simple Lagrange polynomial interpolation of the form

$$\varphi_q^{(i)}(x_q) \;=\; \prod_{j \neq i} \left(x_q - \bar{x}_q^{(j)}\right) \Big/ \prod_{j \neq i} \left(\bar{x}_q^{(i)} - \bar{x}_q^{(j)}\right). \tag{7.11}$$

However, to avoid an oscillating behavior of the blending functions that would later be carried over to the mapping, spline interpolations may be preferable for the construction of the interpolation.

Projector Definition. We use the blending functions to define projectors $P_q[\rho]$ for each coordinate direction x_q, $q = 1, \ldots, d$. For notational clarity, we omit the



(a) $P_1[\rho](x)$ (b) $P_2[\rho](x)$

Figure 7.2: Projectors $P_1[\rho](x)$ and $P_2[\rho](x)$ for the example from Figure 7.1(b).

dependence of the projectors on the parameters μ and the random events ω which is implicitly carried by the boundary mapping ρ. For $x = (x_1, \ldots, x_d)^T \in D$, the projector $P_q[\rho] : D \to \mathbb{R}^d$ is given by

$$P_q[\rho](x) := \sum_{i=0}^{I_q} \rho(x_1, \ldots, \bar{x}_q^{(i)}, \ldots, x_d)\ \varphi_q^{(i)}(x_q), \quad q = 1, .., d. \tag{7.12}$$

It coincides with ρ on the constant generalized coordinates $x_q^{(i)}$, $i = 0, .., I_q$. Furthermore, the behavior of ρ on the constant generalized coordinates is interpolated in x_q-direction using the corresponding blending functions $\varphi_q^{(i)}$, $i = 0, \ldots, I_q$.

Let us consider again the example boundary mapping from Figure 7.1(b). Using no additional generalized coordinates but only the boundary mapping $\rho : \partial D \to \mathbb{R}^2$, Figure 7.2 shows the values of the projectors P_1 and P_2. The blending functions are simply linear interpolation functions, i.e., $\varphi_q^{(1)} = 1 - x_q$, $\varphi_q^{(2)} = x_q$, $q = 1, 2$.

Mapping. Certainly, an appropriate mapping can not be constructed by just adding the different Projectors. For the assembling of the mapping, we define products and boolean sums of projectors. Denote the product of operators by $P_q P_p[\rho] = P_q[P_p[\rho]]$. In other words, we obtain

$$P_p P_q[\rho](x) = \sum_{i=0}^{I_p} \sum_{j=0}^{I_q} \rho(x_1, \ldots, \bar{x}_p^{(i)}, \ldots, \bar{x}_q^{(j)}, \ldots, x_d)\ \varphi_p^{(i)}(x_p) \varphi_q^{(j)}(x_q).$$

Using the definition of the projector in (7.12), it is straightforward to show that $P_p P_q[\rho](x^*) = P_q(x^*)$ for $x^* \in D$ with $x_p^* = \bar{x}_p^{(m)}$. Then, the product of the $k \leq d$

projectors P_{q_n}, $n = 1, \ldots, k$, is given by

$$\left(\prod_{n=1}^{k} P_{q_n}\right)[\rho](x) := \sum_{i_{q_1}=1}^{I_{q_1}} \cdots \sum_{i_{q_k}=1}^{I_{q_k}} \left(\rho(x)\Big|_{x_{q_m}=\bar{x}_{q_m}^{(i_{q_m})}}^{m=1,\ldots,k} \prod_{n=1}^{k} \varphi_q^{(i_{q_n})}(x_{q_n})\right) \tag{7.13}$$

and it is clear that (7.13) coincides with ρ on all intersection points of the corresponding generalized coordinates, i.e., for $x_0 \in \{x \in D \,|\, x_{q_n} = \bar{x}_{q_n}^{(i_{q_n})}, n = 1, \ldots, k, i_{q_n} \in \{0, \ldots, I_{q_n}\}\}$, we have $(\prod_{n=1}^{k} P_{q_n})[\rho](x_0) = \rho(x_0)$.

Considering the product of all projectors P_q, $q = 1, \ldots, d$, the result coincides with ρ at the $(I_1+1)\cdot\ldots\cdot(I_d+1)$ points $(\bar{x}_1^{(i_1)}, \ldots, \bar{x}_d^{(i_d)})^T$. Between these points, ρ is interpolated using the product of d blending functions. For the example provided in Figure 7.2, this means that $P_1P_2[\rho]$ coincides with ρ at the four corners of D. In between, we linearly interpolate. Hence, we obtain $P_1P_2[\rho](x) = x$ for all $x \in D$.

Next, we define the "boolean sum" of two projectors P_i and P_j by

$$(P_p \oplus P_q)[\rho](x) := (P_p + P_q)\,[\rho](x) - (P_pP_q)[\rho](x).$$

The boolean sum coincides with ρ on all generalized coordinates

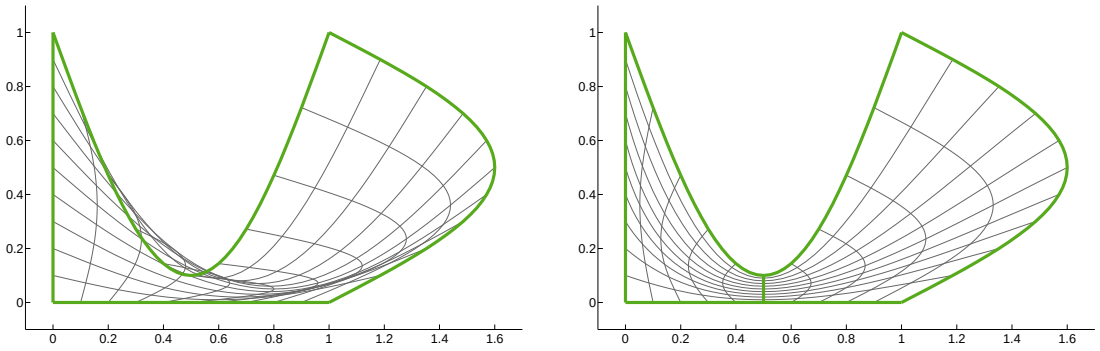
Since $P_pP_q[\rho](x^*) = P_q(x^*)$ for $x^* \in D$ with $x_p^* = \bar{x}_p^{(m)}$, $(P_p \oplus P_q)[\rho]$ coincides with ρ on all generalized coordinates $\bar{x}_p^{(i)}$, $i = 1, \ldots, I_p$, and $\bar{x}_q^{(j)}$, $j = 1, \ldots, I_q$. For the proof, we consider an arbitrary point $x^* \in D$ such that $x_p^* = \bar{x}_p^{(m)}$. From above, we know that $P_pP_q[\rho](x^*) = P_q(x^*)$ which leads to $(P_p \oplus P_q)[\rho](x^*) = P_p[\rho](x^*) = \rho(x^*)$ by definition of P_p.

Hence, the following boolean sum of all d operators coincides with ρ on all constant generalized coordinates and therefore defines the mapping T:

$$T(x) := \left(\bigoplus_{q=1}^{d} P_q\right)[\rho](x) := \sum_{k=1}^{d} (-1)^{k+1} \left(\sum_{1 \le q_1 < \ldots < q_k \le d} \prod_{n=1}^{k} P_{q_n}\right)[\rho](x).$$

In two dimensions, i.e., for d=2, this formula reduces to

$$\begin{aligned}(P_1 \oplus P_2)\,[\rho](x) &= (P_1 + P_2)\,[\rho](x) - (P_1P_2)\,[\rho](x) \\ &= \sum_{i=0}^{I_1} \rho(\bar{x}_1^{(i)}, x_2)\; \varphi_1^{(i)}(x_1) + \sum_{j=0}^{I_2} \rho(x_1, \bar{x}_2^{(j)})\; \varphi_2^{(j)}(x_2) \\ &\quad - \sum_{i=0}^{I_1} \sum_{j=0}^{I_2} \rho(\bar{x}_1^{(i)}, \bar{x}_2^{(j)})\; \varphi_1^{(i)}(x_1)\varphi_2^{(j)}(x_2).\end{aligned}$$



(a) No additional generalized coordinates.

(b) One additional generalized coordinate $\bar{x}_1^{(1)} = 0.5$.

Figure 7.3: Two transfinite element mapping results for the example from Figure 7.1(b).

For d=3, we have

$$P_1 \oplus P_2 \oplus P_3 := (P_1 + P_2 + P_3) - (P_1P_2 + P_1P_3 + P_2P_3) + (P_1P_2P_3)\,.$$

Due to the availability of an affine decomposition of ρ at the boundary ∂D as provided in (7.7) with $\mathcal{O}(QK^{\text{detail}})$ terms, it can be assumed that ρ is also affine with respect to the parameter pair $(\mu, \omega) \in \mathcal{P} \times \Omega$ on the additional generalized coordinates. Then, it is clear that T can be decomposed accordingly with $\mathcal{O}(QK^{\text{detail}})$ terms. We define $T_{q,k}(x) := (\bigoplus_{q=1}^{d} P_q)[\rho_{q,k}](x)$ and obtain the same form of T as provided in (7.9) for the Laplace based mapping.

In Figure 7.3(a), we provide the mapping result for the example from Figure 7.1(b), using no additional generalized coordinate and the projectors as illustrated in Figure 7.2 with linear blending functions. Obviously, the mapping is not invertible.

For Figure 7.3(b), we use the additional generalized coordinate $\bar{x}_1^{(1)} = 0.5$ and $\rho(\bar{x}_1^{(1)}, x_2) = (\bar{x}_1^{(1)}, 0.1 \cdot x_2)^T$, where 0.9 is the maximal deviation of the upper boundary of D. For the definition of the blending functions, we used the Lagrange interpolation (7.11) such that $\varphi_1^{(i)}$, $i = 0, 1, 2$, are quadratic and $\varphi_2^{(i)}$, $i = 0, 1$, linear functions. The quadratic effect can be directly seen in the propagation of the deformation of the right boundary in x_1-direction. To reduce the strong deviations in x_1 direction on the left-hand side of $\bar{x}_1^{(1)} = 0.5$, it would be possible

to use splines instead of Lagrange interpolation.

Computation of the Jacobian Matrix. It is clear that the Jacobian matrix of the mapping T can analogously be expressed as it has been provided in (7.10) for the Laplace based mapping, where $J_{Tq,k} = (\frac{\partial T_{q,k}}{\partial x_1} \cdots \frac{\partial T_{q,k}}{\partial x_d})$, where each component is a d-dimensional vector. We exemplarily construct $J_{Tq,k}$ for $d = 2$. Given the blending functions and their derivatives, we can easily evaluate $J_{Tq,k}$ — and therefore $J_T(\mu, \omega)$ for each random parameter pair $(\mu, \omega) \in \mathcal{P} \times \Omega$ — using just values of $\rho_{q,k}$ and of its derivative at the constant generalized coordinates $\bar{x}_p^{(i)}$, $p = 1, \ldots, d$. We obtain

$$\begin{aligned}
\frac{\partial T_{q,k}}{\partial x_1}(x) = &\sum_{i=0}^{I_1} \rho_{q,k}(\bar{x}_1^{(i)}, x_2) \frac{\partial \varphi_1^{(i)}}{\partial x_1}(x_1) \; + \; \sum_{j=0}^{I_2} \frac{\partial \rho_{q,k}}{\partial x_1}(x_1, \bar{x}_2^{(j)}) \; \varphi_2^{(j)}(x_2) \\
&- \sum_{i=0}^{I_1} \sum_{j=0}^{I_2} \rho_{q,k}(\bar{x}_1^{(i)}, \bar{x}_2^{(j)}) \; \frac{\partial \varphi_1^{(i)}}{\partial x_1}(x_1) \varphi_2^{(j)}(x_2), \\
\frac{\partial T_{q,k}}{\partial x_2}(x) = &\sum_{i=0}^{I_1} \frac{\partial \rho_{q,k}}{\partial x_2}(\bar{x}_1^{(i)}, x_2) \; \varphi_1^{(i)}(x_1) \; + \; \sum_{j=0}^{I_2} \rho_{q,k}(x_1, \bar{x}_2^{(j)}) \; \frac{\partial \varphi_2^{(j)}}{\partial x_2}(x_2) \\
&- \sum_{i=0}^{I_1} \sum_{j=0}^{I_2} \rho_{q,k}(\bar{x}_1^{(i)}, \bar{x}_2^{(j)}) \; \varphi_1^{(i)}(x_1) \frac{\partial \varphi_2^{(j)}}{\partial x_2}(x_2).
\end{aligned}$$

For $d = 3$, the computation works analogously. Since T allows for an affine decomposition with $\mathcal{O}(QK^{\text{detail}})$ terms, J_T can be decomposed analogously.

7.3 Affine Decomposition of the Transformed Problem

Let us consider the coefficient functions in (7.3) which are now dependent on parameters $\mu \in \mathcal{P}$ and stochastic events $\omega \in \Omega$. On the one hand, this dependence is contained in the Jacobian matrix J_T and therefore in J_T^{-1} and $\det J_T$. On the other hand, the transformed coefficients $\mathfrak{c} \in \{a, b, c, g, h\}$, $\mathfrak{c}(x; \mu, \omega) := \tilde{\mathfrak{c}}(T(x; \mu, \omega); \mu, \omega)$, are implicitly dependent on (μ, ω) via the mapping T and may also carry further parametric and stochastic influences. In any case, the trilinear, bilinear, and linear forms in (7.4) are now dependent on $(\mu, \omega) \in \mathcal{P} \times \Omega$ and affine representations are

required. In this section, we investigate the affinity with respect to the deterministic parameter $\mu \in \mathcal{P}$.

In the preceding section, we derived affine mappings T based upon affine boundary mappings ρ. We furthermore assume that $\tilde{\mathfrak{c}}(\mu, \omega)$ is already affine in μ, i.e.,

$$\tilde{\mathfrak{c}}(\tilde{x}; \mu, \omega) = \sum_{q=1}^{Q^{\mathfrak{c}}} \theta_q^{\mathfrak{c}}(\mu) \tilde{\mathfrak{c}}_q(\tilde{x}; \omega), \quad \tilde{\mathfrak{c}} \in \{\tilde{a}, \tilde{b}, \tilde{c}, \tilde{g}, \tilde{h}\}, \tag{7.14}$$

where ω may indicate both the dependence of the coefficient $\tilde{\mathfrak{c}}$ on the stochastic domain and possibly further probabilistic dependencies. We now map the coefficients to the reference domain. Considering only the affine decomposition of T with respect to the deterministic paramater μ, i.e.,

$$T(x; \mu, \omega) = \sum_{p=1}^{Q^T} \theta_p^T(\mu) T_p(x; \omega), \tag{7.15}$$

we define $\mathfrak{c}_{q,p}(x; \omega) := \tilde{\mathfrak{c}}_q(T_p(x; \omega); \omega)$ and obtain

$$\begin{aligned} \mathfrak{c}(x; \mu, \omega) := \tilde{\mathfrak{c}}(T(x; \omega); \mu, \omega) &= \sum_{q=1}^{Q^{\mathfrak{c}}} \theta_q^{\mathfrak{c}}(\mu) \tilde{\mathfrak{c}}_q(T(x; \mu, \omega); \omega) \\ &= \sum_{q=1}^{Q^{\mathfrak{c}}} \theta_q^{\mathfrak{c}}(\mu) \sum_{p=1}^{Q^T} \theta_p^T(\mu) \mathfrak{c}_{q,p}(x; \omega). \end{aligned} \tag{7.16}$$

Hence, the mapped coefficient functions $\mathfrak{c} \in \{a, b, c, g, h\}$ are affine with respect to μ as well with $Q^{\mathfrak{c}} Q^T$ terms.

The availability of affine decompositions of T, J_T, and of the coefficient functions $\mathfrak{c} \in \{a, b, c, g, h\}$ is not sufficient for the definition of affine decompositions of the trilinear, bilinear, and linear forms in (7.4). As we can see in (7.3), also the terms $|\det J_T|$, $|\det J_T| J_T^{-1}$, and $|\det J_T| J_T^{-1} J_T^{-T}$ are involved.

Theoretically, the Jacobian determinant can be decomposed with respect to the deterministic parameter μ using $\mathcal{O}((Q^T)^d)$ terms, since, e.g., for $d = 2$, we have

$$\det J_T(x; \mu, \omega) = \left(\frac{\partial T_1}{\partial x_1} \frac{\partial T_2}{\partial x_2} - \frac{\partial T_1}{\partial x_2} \frac{\partial T_2}{\partial x_1} \right)(x; \mu, \omega).$$

Then, the affine decomposition of $\det J_T(x; \mu, \omega) \mathfrak{c}(x; \mu, \omega)$ would already include $\mathcal{O}\big(Q^{\mathfrak{c}} (Q^T)^{(d+1)}\big)$ terms which might be inapplicably large. Furthermore, for $d = 2$,

we obtain

$$|\det J_T(x;\mu,\omega)| \cdot J_T^{-1}(x;\mu,\omega) = \operatorname{sign}(\det J_T(x;\mu,\omega)) \begin{pmatrix} \frac{\partial T_2}{\partial x_2} & -\frac{\partial T_1}{\partial x_2} \\ -\frac{\partial T_2}{\partial x_1} & \frac{\partial T_1}{\partial x_1} \end{pmatrix} (x;\mu,\omega)$$

which is affine with $\mathcal{O}(Q^T)$ terms since the sign of the Jacobian determinant is constant. However, for $d > 2$, we loose this affinity. Furthermore, it is not possible to directly construct an affine decomposition of the term $|\det J_T(x;\mu,\omega)| \cdot J_T^{-1}(x;\mu,\omega) J_T^{-T}(x;\mu,\omega)$.

Summarizing, we can not directly use the affinity of T, J_T, and $\mathfrak{c} \in \{a, b, c, g, h\}$ to construct affine decompositions of the forms in (7.4). In the following sections, we will show how the RBM can still be efficiently applied. We will therefore consider two different cases of the parametric and stochastic dependence of the domain.

7.4 RBM for Stochastic, Non-Parametric Domains

In this section, we consider domains $\tilde{D}(\omega) \subset \mathbb{R}^d$, $\omega \in \Omega$, where the deformation of the boundary is purely stochastic and does not carry any deterministically parametric dependence. We will show that the coefficients $\mathfrak{c}_T$, $\mathfrak{c} \in \{a, b, c, g, h\}$, allow for affine decompositions with respect to the deterministic parameter μ. Therefore, the linear, bilinear, and trilinear forms in (7.4) are affine in μ and we can show that it is possible to apply the theory of the Chapters 5 and 6.

For a given reference domain $D \subset \mathbb{R}^d$, we assume the availability of a KL expansion of the boundary mapping $\rho(\omega) : \partial D \to \tilde{\partial} D(\omega)$ and an appropriate diffeomorphic mapping $T(\omega) : D \to \tilde{D}(\omega)$,

$$\rho(x;\omega) = \rho_0(x) + \sum_{k=1}^{K^{\text{detail}}} \xi_k(\omega)\rho_k(x), \tag{7.17}$$

$$T(x;\omega) = T_0(x) + \sum_{k=1}^{K^{\text{detail}}} \xi_k(\omega)T_k(x). \tag{7.18}$$

We furthermore assume that the possibly parametric and stochastic coefficients $\tilde{a}, \tilde{b}, \tilde{c}, \tilde{g}, \tilde{h}$ from (7.1) allow for affine decompositions with respect to a deterministic parameter $\mu \in \mathcal{P}$ of the form (7.14). Again, ω may indicate both the dependence on the stochastic domain and possibly further probabilistic dependencies.

We now map the coefficients to the reference domain. For $\mathfrak{c} \in \{a, b, c, g, h\}$, we define $\mathfrak{c}_q(x;\omega) := \tilde{\mathfrak{c}}_q(T(x;\omega);\omega)$ and obtain

$$\mathfrak{c}(x;\mu,\omega) := \tilde{\mathfrak{c}}(T(x;\omega);\mu,\omega) = \sum_{q=1}^{Q^{\mathfrak{c}}} \theta_q^{\mathfrak{c}}(\mu)\tilde{\mathfrak{c}}_q(T(x;\omega);\omega) = \sum_{q=1}^{Q^{\mathfrak{c}}} \theta_q^{\mathfrak{c}}(\mu)\mathfrak{c}_q(x;\omega). \quad (7.19)$$

Since the mapping T and its Jacobian matrix J_T do not depend on the deterministic parameter μ, the coefficients a_T, b_T, c_T, g_T, h_T, defined in (7.3), are therefore already affine with respect to μ. For $\mathfrak{c} \in \{a, b, c, g, h\}$, we denote by $\mathfrak{c}_{T,q}(x;\omega)$ the respective affine term analogously to (7.3) such that

$$\mathfrak{c}_T(x;\mu,\omega) := \sum_{q=1}^{Q^{\mathfrak{c}}} \theta_q^{\mathfrak{c}}(\mu)\mathfrak{c}_{T,q}(x;\omega).$$

E.g., we have $a_{T,q}(x;\omega) = |\det J_T(x)| J_T^{-1}(x) J_T^{-T}(x) a_q(x;\omega)$. Hence, for the efficient application of the RBM, it remains to get affine decompositions of the coefficients with respect to ω.

Since the random functions $\mathfrak{c}_{T,q}(x;\omega)$, $q = 1, \dots, Q^{\mathfrak{c}}$, $\mathfrak{c} \in \{a, b, c, g, h\}$, all depend on the stochastic domain, they are stochastically dependent. For the application of the RBM, it is not appropriate to apply the KL decomposition on each term separately. The resulting random variables that appear in respective KL expansions would be correlated. Hence, we would violate Assumption 5.14 for linear problems or Assumption 6.4 for nonlinear problems. The RBM and the a-posteriori analysis from Sections 5.3 to 5.5 or from Section 6.3 could not be applied.

As a consequence, following Remark 5.13, it is necessary to evaluate a single joint KL expansions for all affine terms of the coefficient functions $\mathfrak{c}_{T,q}(x;\omega)$, $q = 1, \dots, Q^{\mathfrak{c}}$, $\mathfrak{c} \in \{a, b, c, g, h\}$ (cf. Section 2.2.3). Let $\mathfrak{c}_{T,q,0}(x)$ denote the mean of $\mathfrak{c}_{T,q}(x;\omega)$. Then, the complete affine decomposition for all coefficients reads

$$\mathfrak{c}_T(x;\mu,\omega) := \sum_{q=1}^{Q^{\mathfrak{c}}} \theta_q^{\mathfrak{c}}(\mu) \left[\mathfrak{c}_{T,q,0}(x) + \sum_{k=1}^{K^{\text{joint}}} \sqrt{\lambda_k^{\text{joint}}}\, \xi_k^{\text{joint}}(\omega)\mathfrak{c}_{T,q,k}(x) \right], \quad (7.20)$$

where λ_k^{joint} and $\xi_k^{\text{joint}}(\omega)$ do neither depend on q nor on the specific coefficient $\mathfrak{c} \in \{a, b, c, g, h\}$.

The multi-component KL expansion can easily be constructed using the method of snapshots. Since we already have a KL expansion of ρ and T, it is possible to generate arbitrarily many samples of the stochastic domain.

It is now straightforward to define the affine formulations of the linear, bilinear, and trilinear forms in (7.4). As mentioned before, we only have to require that each KL sum is truncated at the same value to apply the RBMs from Chapter 5 and Chapter 6.

7.5 RBM for Stochastic and Parametric Domains

Let us now consider the case of stochastic *and* parametric domains. We assume the availability of an affine boundary mapping $\rho(\mu,\omega) : \partial D \to \partial\tilde{D}(\mu,\omega)$ as provided in (7.7), leading to the affine decomposition of the mapping $T(\mu,\omega) : D \to \tilde{D}(\mu,\omega)$ of the form (7.9). Furthermore, it can be assumed that the coefficients $\tilde{\mathfrak{c}} \in \{\tilde{a},\tilde{b},\tilde{c},\tilde{g},\tilde{h}\}$ allow for affine decompositions of the form (7.14).

Using only the affine representation of T with respect to the parameter μ as provided in (7.15), we obtain

$$\mathfrak{c}(x;\mu,\omega) = \sum_{q=1}^{Q^{\mathfrak{c}}} \theta_q^{\mathfrak{c}}(\mu) \sum_{p=1}^{Q^T} \theta_p^T(\mu)\mathfrak{c}_{q,p}(x;\omega), \quad \mathfrak{c} \in \{a,b,c,g,h\},$$

analogously to (7.16). As in Section 7.4, we apply a single joint KL expansion on all components $\mathfrak{c}_{q,p}(x;\omega)$, $p = 1,\dots,Q^T$, $q = 1,\dots,Q^{\mathfrak{c}}$, $\mathfrak{c} = a,b,c,g,h$, and $T_p(x;\omega)$, $p = 1,\dots,Q^T$, which yields a complete affine representation

$$\begin{aligned}
\mathfrak{c}(x;\mu,\omega) &= \sum_{q=1}^{Q^{\mathfrak{c}}} \theta_q^{\mathfrak{c}}(\mu) \sum_{p=1}^{Q^T} \theta_p^T(\mu) \left[\mathfrak{c}_{q,p,0}(x) + \sum_{k=1}^{K^{\text{joint}}} \sqrt{\lambda_k^{\text{joint}}}\, \xi_k^{\text{joint}}(\omega)\mathfrak{c}_{q,p,k}(x) \right], \\
T(x;\mu,\omega) &= \sum_{p=1}^{Q^T} \theta_p^T(\mu) \left[T_{p,0}(x) + \sum_{k=1}^{K^{\text{joint}}} \sqrt{\lambda_k^{\text{joint}}}\, \xi_k^{\text{joint}}(\omega) T_{p,k}(x) \right],
\end{aligned} \tag{7.21}$$

$\mathfrak{c} \in \{a,b,c,g,h\}$. The term $\mathfrak{c}_{q,p,0}(x)$ denotes the mean of $\mathfrak{c}_{q,p}(x;\omega)$ and $T_{p,0}(x)$ denotes the mean of $T_p(x;\omega)$. Hence, it is sufficient to generate the random variables ξ_k^{joint}, $k = 1,\dots,K^{\text{joint}}$, for the complete description of a random sample. Otherwise, the modeling of the correlated terms a,b,c,g,h and T would be difficult. It is clear that the Jacobian matrix $J_T(x;\mu,\omega)$ is analogously decomposed to $T(x;\mu,\omega)$ in (7.21).

However, we still can not create affine approximations of the coefficient functions a_T, b_T, c_T, g_T, h_T, as we have shown in Section 7.3, and the RBM can not directly

be applied. Hence, it is now necessary to apply the EIM or the POIM from Chapter 3. Note that it is not necessary to apply the LSEIM even if the original stochastic input is based upon noisy data. Since we already applied the KL expansion, we already obtained a "smoothing effect" of the data. The KL expansion can be truncated such that noisy data is removed or smoothed, keeping the information that is necessary for an appropriate detailed solution.

The EIM or POIM is now based upon samples of a_T, b_T, c_T, g_T, h_T that can be generated using the components in (7.21). It has been mentioned in Chapter 3 that the evaluation of the input functions at the interpolation points is required to be independent of the discretization. In our case, it is clear that the evaluation of $\mathfrak{c}(x;\mu,\omega)$ in (7.21) is of complexity $\mathcal{O}(Q^{\mathfrak{c}}Q^T K^{\text{joint}})$ and the complexity for $T(x;\mu,\omega)$ is of complexity $\mathcal{O}(Q^T K^{\text{joint}})$. Hence, the overall complexity for the evaluation of a coefficient function a_T, b_T, c_T, g_T, h_T at an interpolation point also reads $\mathcal{O}(Q^{\mathfrak{c}}Q^T K^{\text{joint}})$ and the requirement of discretization independent evaluations is fulfilled.

To remove redundancies in the affine approximations generated by the EIM or POIM, it can be useful to apply the multi-component EIM or POIM for quantities that appear in the same linear form [86]. E.g., for $a_T(x;\mu,\omega) \in \mathbb{R}^{d\times d}$, it is useful to generate only joint affine approximation for all d^2 components.

Suppose the affine approximations become too large, it is also possible to apply the implicit partitioning methods of Chapter 4 to reduce the online costs. For this case, the p-Partitioning of Section 4.1.1 or the hp methods of Section 4.1.2 are not appropriate since each random variable ξ_k^{joint} would be considered as a parameter. Therefore, the dimension of the parameter domain would be too large.

Given the affine approximation of our coefficients, using the EIM or POIM, we can apply the RBM. However, it is not possible to apply the methods from Chapter 5 and Chapter 6. Instead, we refer to [86], where the RBM is used in combination with the EIM. Most of the error analysis takes similar forms, replacing the error analysis that occurs due to KL truncation by the analysis due to the "truncation" of the EIM affine decomposition. However, it is not possible anymore to use the statistical output error analysis of Sections 5.4, 5.5, and 6.3. In detail, the requirements of Lemma 5.12 and Lemma 6.4 can not be fulfilled. Hence, we do not obtain the improved error bounds for the statistical moments. For the error

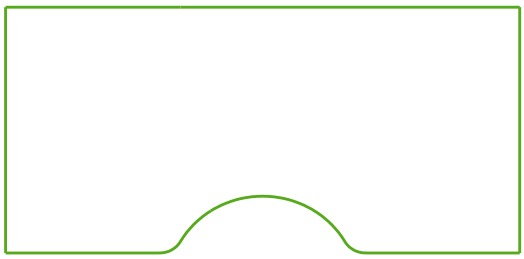


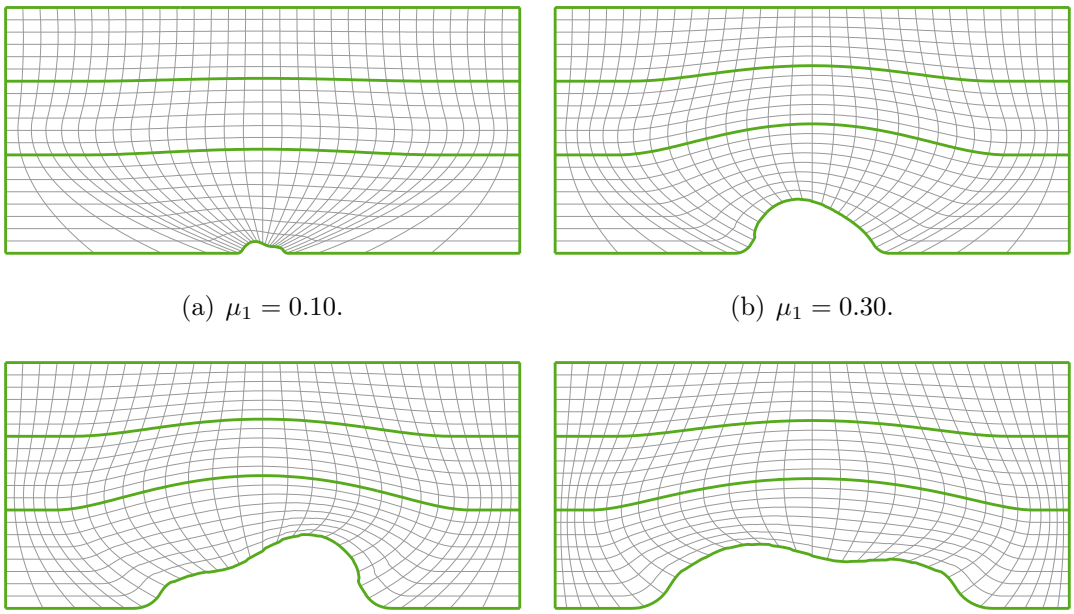
Figure 7.4: Expected shape of the random domain $\tilde{D}(\mu, \omega)$ for $\mu = 0.4$.

bound of the variance, we would have to fall back on the evaluation procedure in Appendix A.

7.6 Numerical Examples

We illustrate the different approaches of Sections 7.4 and 7.5 using the example of a plate where a hole appears on the bottom side. The plate is represented by the domain $\tilde{D} \subset (0, 2) \times (0, 1) \subset \mathbb{R}^2$. The radius of the hole is modeled by a deterministic parameter $\mu_1 \in \mathcal{P}_1 := [0.1, 0.7] \subset \mathbb{R}$ whereas the shape of its boundary is modeled stochastically. In detail, we consider a circular hole, as shown in Figure 7.4 for the example $\mu_1 = 0.4$, which denotes the expected shape of the random boundary. Then, we use smoothed Wiener processes as described in Section 3.5 to define the deviations of the hole in x_1- and x_2-direction. To certify the smoothness of the boundary, the Wiener process is transformed such that the deviation and its derivative on the very left and very right side of the hole is set to zero.

Description of the Mapping T. We define the rectangular reference domain $D := (0, 2) \times (0, 1) \subset \mathbb{R}^2$ and use the transfinite element procedure to construct an appropriate mapping $T : D \to \tilde{D}(\mu_1, \omega)$. Figure 7.5 shows four random samples of the domain $\tilde{D}(\mu_1, \omega)$ for different values of μ_1 and the result of our mapping $T(\mu_1, \omega)$ for a uniform grid of 31 times 21 points. The grid does not coincide with the used discretization which may require local refinements for stability reasons. The bold, green lines mark the generalized coordinates. The left, right, and top boundary of the domain is fixed, i.e., $T(x; \mu_1, \omega) = x$. Besides the bottom bound-



(a) $\mu_1 = 0.10$. (b) $\mu_1 = 0.30$.

(c) $\mu_1 = 0.50$. (d) $\mu_1 = 0.70$.

Figure 7.5: Four random samples of of $\tilde{D}(\mu_1, \omega)$ for different values of μ with the corresponding mapping $T(\mu_1, \omega) : D \to \tilde{D}(\mu_1, \omega)$ for a uniform grid on D.

ary, we defined two additional generalized coordinates $\bar{x}_2^{(1)} = 0.4$ and $\bar{x}_2^{(2)} = 0.7$.

Let us first describe the transformation of the bottom boundary. It can be seen in Figure 7.5 that points to the left and to the right of the hole are shifted inwards, i.e., towards the hole. In detail, points x on the reference boundary segment $[0.0, 0.3] \times \{0\} \subset D$ are mapped to $\tilde{x} \in [0.0, 1.0-\mu_1] \times \{0\}$ on $\tilde{D}(\mu_1, \omega)$ and analogously $x \in [1.7, 2.0] \times \{0\} \mapsto \tilde{x} \in [1.0+\mu_1, 2.0] \times \{0\}$. Hence, this transformation only depends on the deterministic parameter and not on the stochastic boundary. It leads to a higher resolution at the most significant regions. Furthermore, it enables bijective mappings for cases, where the boundary on the hole goes upwards in almost vertical direction, as it can be observed for example in Figure 7.5(c) on the right side of the hole.

To emphasize this effect, especially for large μ_1, the points on the generalized coordinates $\bar{x}_2^{(1)} = 0.4$ and $\bar{x}_2^{(2)} = 0.7$ are shifted outwards which can best be observed in Figure 7.5(d). Again, the shift depends only on the deterministic parameter and is larger at the lower generalized coordinate $\bar{x}_2^{(1)}$ than at $\bar{x}_2^{(2)}$. It is distributed over the whole x_1-axis.

Furthermore, the two generalized coordinates generate an upward displacement of points in the inner part of the domain. The magnitude of the displacement depends on the random deviation in x_2-direction at the center of the hole $x_{\text{center}} = (1,0)^T$, i.e., on $\rho_2(x_{\text{center}}; \mu_1, \omega)$. Even though this is not optimal since the maximal deviation on the bottom boundary is not necessarily given on x_{center} as we can observe for example in Figure 7.5(c), it provides accurate mappings and is easy to evaluate. The determination of the overall maximum would contradict the assumption of evaluations independent of $\mathcal{N}$. The range on the generalized coordinates where the upward shift occurs is restricted to the inner part of the domain, in particular to points $x \in \bar{x}_2^{(i)}$ with $x_1 \in [0.3, 1.7]$, $i = 1, 2$.

Problem Description. Let us now describe the considered PDE on the stochastic and parametric domain. We introduce an additional parameter $\mu_0 \in \mathcal{P}_0 := [0.1, 1.0] \subset \mathbb{R}$ and define $\tilde{a}(\tilde{x}; \mu_0) := \mu_0$ which describes the constant heat conductivity of the plate $\tilde{D}(\mu_1, \omega)$. For $\mu = (\mu_0, \mu_1) \in \mathcal{P} := \mathcal{P}_0 \times \mathcal{P}_1$ and $\omega \in \Omega$, we define

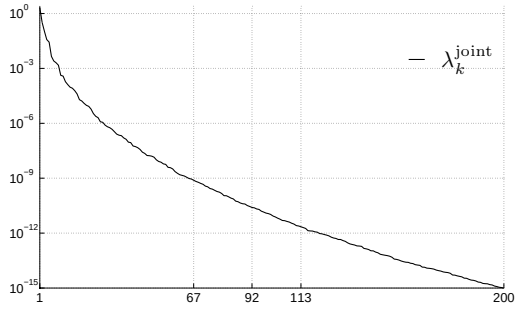
$$\begin{cases} -\nabla \cdot \big(\tilde{a}(\mu_0) \nabla \tilde{u}(\tilde{x}; \mu, \omega)\big) &= 0, & \tilde{x} \in \tilde{D}(\mu_1, \omega), \\ \tilde{u}(\tilde{x}; \mu, \omega) &= 0, & \tilde{x} \in \tilde{\Gamma}_{\text{D}}(\mu_1, \omega), \\ \tilde{n}(\tilde{x}; \mu, \omega) \cdot \big((\tilde{a}(\mu_0)\, \nabla \tilde{u}(\tilde{x}; \mu, \omega)\big) &= \tilde{h}(\tilde{x}; \mu_1), & \tilde{x} \in \tilde{\Gamma}_{\text{N}}(\mu_1, \omega), \end{cases} \tag{7.22}$$

where $\tilde{\Gamma}_{\text{D}}$ denotes the left and $\tilde{\Gamma}_{\text{N}}$ the remaining boundary of $\tilde{D}$. We set $\tilde{h}(\tilde{x}; \mu_1) = 0$ on the top and right boundary. On the bottom part, we set $\tilde{h}(\tilde{x}; \mu_1) = 1$ outside the hole and 0 otherwise.

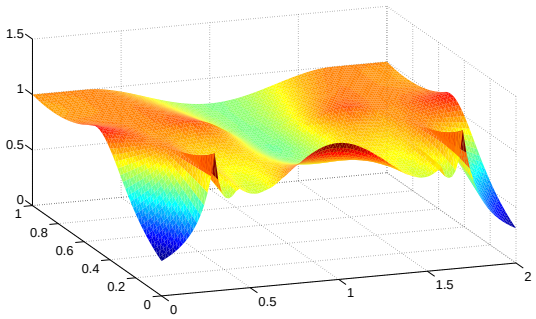
Let Γ_{out} denote the deterministic and non-parametric right boundary of $\tilde{D}(\mu_1, \omega)$. We define the output functionals $\tilde{\ell} : X(\tilde{D}(\mu_1, \omega)) \to \mathbb{R}$ and $\ell : X(D) \to \mathbb{R}$ by

$$\ell(v) = \int_{\Gamma_{\text{out}}} v, \quad v \in X(D),$$
$$\tilde{\ell}(\tilde{v}) = \int_{\Gamma_{\text{out}}} \tilde{v}, \quad v \in X(\tilde{D}(\mu_1, \omega)),$$

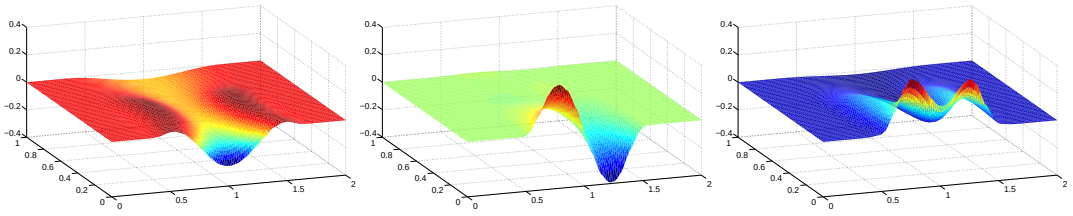
respectively. For $v(x) = \tilde{v}(T(x))$, it is clear that $\ell(v) = \tilde{\ell}(\tilde{v})$ since Γ_{out} does not depend on $(\mu, \omega) \in \mathcal{P}$. The desired output is given by $s(\mu, \omega) = \ell(u(\mu, \omega))$, where $u(\mu, \omega)$ denotes the (weak) solution of the PDE (7.22), transformed to the reference domain. In other words, the output corresponds to the average of the solution $u(\mu, \omega)$ at the output boundary Γ_{out}.



(a) Eigenvalues of the joint KL expansion and KL truncation values $K^{\text{joint}} = 67$, $K^{\text{joint}}_{\text{max}} = 92$, and $K^{\text{joint}}_{\text{detail}} = 113$.



(b) Expectation of $a_{T,1,1}(x;\cdot)$



(c) First three modes (from left to right) of $a_{T,1,1}(x;\cdot)$ using the joint KL expansion.

Figure 7.6: Result of the joint KL expansion of $a_T(x;\omega)$ and $h_T(x;\omega)$.

7.6.1 The Non–Parametric Case

For this section, we set $\mu_1 = 0.4$. Hence, the domain does not carry any parametric dependence anymore and we use $\mu = \mu_0$ and $\mathcal{P} = \mathcal{P}_0$. Since the PDE is affine in μ, we can apply the RBM as proposed in Section 7.4.

Construction of the Affine Representation (7.20)**.** As described in Sections 7.1.2 and 7.4, we first have to employ the multi-component KL expansion on the parameter independent parts of the coefficients

$$\begin{aligned} a_T(x;\mu,\omega) &:= |\det J_T(x;\omega)| J_T^{-1}(x;\omega) J_T^{-T}(x;\omega)\tilde{a}(\mu), \\ h_T(x;\omega) &:= |\det J_T(x;\omega)| \tilde{h}(T(x;\omega)), \end{aligned}$$

to obtain affine decompositions with respect to ω. Since $\tilde{a}(\mu_0) = a(\mu) = \mu$ is constant in space, we can just omit this term for the following considerations and use the notation $a_T(x;\omega)$ instead. Now, the application of the multi-component KL expansion is straightforward.

Figure 7.6 shows some results of this joint KL expansion. In Figure 7.6(a), the eigenvalues are provided. For our reduced model, we use the first $K^{\text{joint}} = 67$ modes to approximate the coefficients a_T and h_T. The KL error can be measured using an additional number of 25 terms, where the detailed solution has been obtained using the first 113 terms. Exemplarily, we also provide some KL results for $a_{T,1,1}(x;\omega)$, i.e., for the first component of the matrix $a_T(x;\omega)$. In Figure 7.6(b), the expectation of $a_{T,1,1}(x;\omega)$ is provided. Even though T has been constructed continuously differentiable, it can be observed that the coefficient is only continuous. However, since a_T is based upon the Jacobian matrix J_T, it is clear that only continuity can be guaranteed. Figure 7.6(c) shows the first 3 modes of $a_{T,1,1}(x;\omega)$. It is important to mention that these modes do not necessarily coincide with the eigenmodes of a direct KL expansion of $a_{T,1,1}(x;\omega)$.

Reduced Simulations. On our reference system, a 3.06 GHz Intel Core 2 Duo processor, 4 GB RAM, we used Comsol 3.5.0.608 (3.5a) to construct and store the FE system components and MATLAB 8.0.0.783 (R2012b) to implement and run both the detailed and reduced models. For the solutions, we used the MATLAB *mldivide* function which automatically adapts to the structure of the system, e.g., sparsity patterns. For the detailed solutions, we needed a discretization with $\mathcal{N} = 25,794$ degrees of freedom to accurately resolve the stochastic boundary.

For the evaluation of the lower bound of the coercivity constant, we applied the method proposed in Section 5.7.1. We assumed that we are interested in the random outputs $s(\mu,\omega)$ and $s^2(\mu,\omega)$ as well as in the statistical outputs $\mathbb{M}_1(\mu)$, $\mathbb{M}_2(\mu)$, and $\mathbb{V}(\mu)$. For the desired relative error tolerance $\varepsilon_{\text{tol}} = 10^{-3}$, the greedy converged for $(N, \tilde{N}^{(1)}, \tilde{N}^{(2)}) = (43, 38, 43)$ basis functions, where the same space has been used for the additional dual problems, i.e., $\tilde{X}_N^{(2)} = \tilde{X}_N^{(3)}$. Suppose we were not interested in the random squared output $s^2(\mu,\omega)$ but only in the second moment, $(N, \tilde{N}^{(1)}, \tilde{N}^{(2)}) = (41, 38, 41)$ basis functions would have been sufficient.

For the solution of the detailed problem and for the evaluation of the desired random outputs, the average run-time was about 1.873 seconds per random sample. For the reduced simulation, we needed only about 0.04798 seconds per sample, where the evaluations included the outputs and the error bounds. Hence, the reduced simulation is about 39 times faster.

Let us take a closer look at the online run-time of the reduced system. Using the MATLAB *profile* function, we could observe that the assembling of the reduced systems, the computation of the solutions, and the evaluation of the outputs took only about 2.6% of the online run-time. For the evaluation of the coercivity lower bound via the SCM, we needed about 17% of the time, where the splitting $\alpha_{\mathrm{LB}}(\mu,\omega) = \theta_{\min}(\mu) \cdot \alpha_{\mathrm{SCM}}(\omega)$ as described in Section 5.7.1 has been applied. The highest complexity is originated by the evaluation of the error bounds. Almost 80% of the run-time has been needed for their evaluation. This effect can be explained by the relatively large number K^{joint} of used KL terms which enters quadratically into the complexity of the error bounds but only linearly in the assembling of the reduced system. In Section 7.6.3, we will briefly describe how the IPMs of Chapter 4 can be used to decrease the number of affine terms for our case.

7.6.2 The Parametric Case

As we have described in Sections 7.3 and 7.5, we can not directly obtain affine approximations of the transformed coefficients a_T and h_T in the parametric and stochastic boundary case. Hence, the respective affine decompositions have to be generated using the EIM or POIM (cf. Chapter 3). However, for their application, in particular for the efficient evaluation of the functions at the interpolation points, the availability of an affine representation of T and of the (already transformed) coefficients a and h of the form (7.21) is still required.

Construction of the Affine Representation (7.21)**.** On the bottom boundary, the coefficient $\tilde{h}(\tilde{x};\mu_1)$ is not affine in μ_1 since $\tilde{h}(\tilde{x};\mu_1) = 1$ outside the hole and zero otherwise. However, the mapping T is constructed such that this non-affinity is "canceled out" by using a constant location of the hole on the reference domain, i.e., $h(x)$ is independent of μ. The other coefficient in the PDE (7.22), $\tilde{a}(\mu_0) = \mu_0$, is constant in space and linear in μ_0. Hence, it is sufficient to generate an affine decomposition of T to obtain a representation of the form (7.21).

For our example, the boundary mapping $\rho : \partial D \to \partial\tilde{D}(\mu_1,\omega)$ is constructed affine in μ with only two terms, where $\theta_1(\mu_1) = 1$ and $\theta_2(\mu_1) = \mu_1$. Furthermore, the projection of the generalized coordinates is also decomposed into two affine terms in μ_1 with the same parameter functions θ_1 and θ_2. Hence, the overall

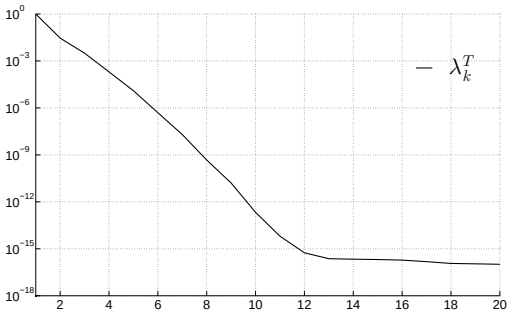


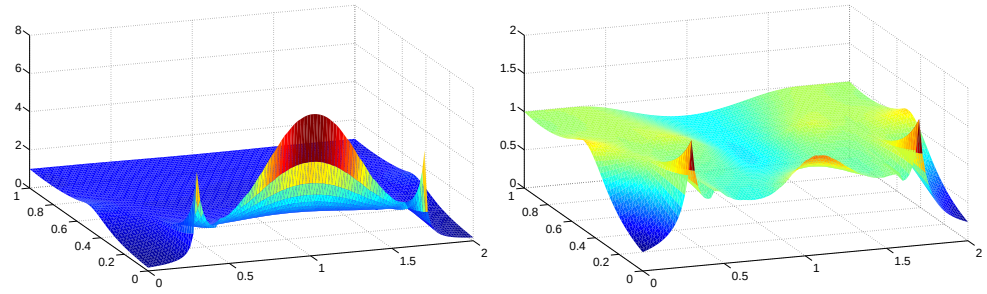
Figure 7.7: Eigenvalues of the KL expansion of T.

number Q^T of affine terms of the domain mapping $T(\mu_1, \omega) : D \to \tilde{D}(\mu_1, \omega)$ with respect to μ_1 is also two.

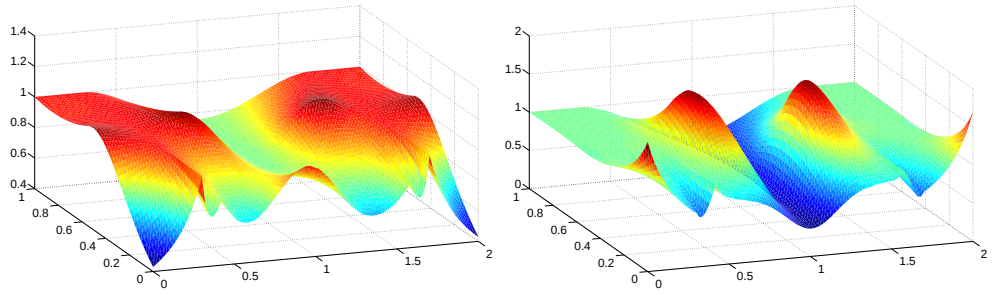
Since the first parameter function $\theta_1 = 1$ is independent of μ_1 and since $\theta_2 = \mu_1$, it is clear that the first affine term T_1 coincides with the mapping T for $\mu_1 = 0$. Thus, the image $T_1(D)$ describes a rectangular domain with no hole anymore. As a consequence, it is independent of the random boundary, i.e., of ω. It only contains the parameter independent shifts of the bottom boundary and the two generalized coordinates in horizontal direction.

We now apply the KL expansion on the second affine term T_2. Since T maps to values in $\mathbb{R}^2$, we actually apply a multi-component KL expansion on the two components of T_2. Figure 7.7 shows the normalized eigenvalues of the joint KL expansion of the two components of T_2. Compared to the eigenvalue decay in Figure 7.6(a) of the joint KL expansion of the coefficients a_T and h_T that depend on the Jacobian matrix and its determinant, the convergence is very fast. We take T_1, the expectation of T_2, and its first 13 modes to obtain a very accurate affine approximation of $T(\mu_1, \omega)$ with 15 terms. Hence, it is possible to efficiently apply the EIM on the coefficients a_T and h_T, respectively.

EIM Approximations of $a_T(\mu, \omega)$. We just consider the first component $a_{T,1,1}$ of a_T for the upcoming examples, although, for actual applications, it would be more efficient to generate a joint EIM approximation of all matrix components. We start with a POIM, i.e., an EIM where the basis functions are determined by a previously performed POD (cf. Section 3.3). For 20 equally distributed parameters $\mu_1 \in \mathcal{P}_1 = [0.1, 0.7]$, we generated 250 random samples each, and used the



(a) $\mu_1 = 0.1$ (b) $\mu_1 = 0.3$



(c) $\mu_1 = 0.5$ (d) $\mu_1 = 0.7$

Figure 7.8: Random samples of $a_{T,1,1}(\mu, \omega)$ for four different values of μ_1.

resulting $n_{\text{train}} = 5000$ samples to compute the POD eigenmodes via the method of snapshots. Since μ_2 appears just as a multiplicative factor in the coefficient, it can be omitted for the moment. Figure 7.8 exemplarily shows four typical random samples of $a_{T,1,1}(\mu, \omega)$ for different parameters μ_1. It is obvious that it is not trivial to find an affine approximation.

We have shown in Section 2.2 that the mean squared KL approximation error for a given truncation value K is given by the sum over the remaining eigenvalues (cf. Equation (2.12)). Considering the eigenvalues of the joint KL expansion in 7.6(a), we can derive that the L_2-approximation error of the coefficients a_T and h_T should not exceed $\varepsilon_{\text{tol}} = 10^{-5} \approx \left(\sum_{k > K_{\max}^{\text{joint}}} \lambda_k^{\text{joint}} \right)^{1/2}$ for the application of the RBM.

Figure 7.9 shows the approximation error of the POIM applied on $a_{T,1,1}(\mu, \omega)$. We provide both the maximal L_∞-error that is usually considered in the EIM context and the mean L_2-error for a better comparison to the results of the previous section. It can be observed that both errors decay with the same convergence rate and the values roughly coincide. For the desired accuracy of $\varepsilon_{\text{tol}} = 10^{-5}$, we need

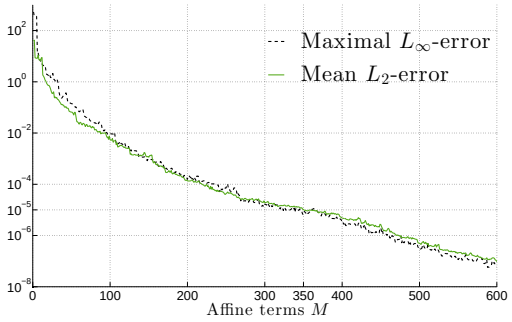


Figure 7.9: Maximal L_∞- and average L_2-error of the POIM applied on $a_{T,1,1}$.

about 350 terms.

Application of the Coefficient Based FS IPM. The number of affine terms increased significantly compared to the $K_{\max}^{\text{joint}} = 92$ terms in the non-parametric case. Thus, it is not clear if the RBM can still be efficiently applied. Furthermore, the dimensions of the reduced spaces X_N, $\tilde{X}_N^{(1)}$, and $\tilde{X}_N^{(2)}$ will also increase due to the additional parameter dependence of the domain. Hence, we would like to perform a partitioning of the parameter domain $\mathcal{P} \times \Omega$ to reduce the online costs.

Let us first specify our parameter setting in more detail. For now, we do not consider the heat conductivity parameter μ_0. Hence, we only obtain a single deterministic parameter $\mu_1 \in [0.1, 0.7]$. Furthermore, we consider the random variables ξ_k^T of the KL expansion of T as additional parameters. We assume that each random variable ξ_k^T takes values in a bounded interval $I_k := (a_k, b_k) \subset \mathbb{R}$. Then, we can define a compact parameter domain of the dimension $p = 14$.

We could now try to use explicit partitioning procedures such as the p- or the hp-Partitioning. However, both the p-Partitioning and the hp gravity center splitting scheme divide the parameter domain into $2^p = 16,384$ subdomains in each iteration. This is certainly not appropriate. The hp anchor point splitting scheme divides the parameter domain into two subdomains based upon the distance to the anchor point. Hence, it can also be applied to parameter domains of higher dimensions. Nevertheless, it is not clear if this splitting is really appropriate for the stochastic case, since the random variables are usually not equally distributed on the intervals I_k. Furthermore, the significance of the random variables for the problem decreases for larger k which is not considered by the splitting scheme.

We therefore prefer an implicit partitioning procedure of Chapter 4. Since the evaluation of the coefficient at an interpolation point is rather expensive ($\mathcal{O}(15)$), the MS IPM could become too expensive. Hence, we apply the coefficient based FS IPM, where the online assignment can be achieved using a binary tree search. For the refinement, we use Algorithm 4.9 that adaptively selects the number of necessary coefficients. We set the maximal number to 12 and reject a partition if one of the child subdomains obtains less than 5% of the functions of the parent subdomain.

For the desired accuracy $\varepsilon_{\text{tol}} = 10^{-5}$, we tested 4 different numbers $M_{\max}$ of allowed affine terms. For $M_{\max} = 200$, we needed 5 subdomains and 13 subdomains were sufficient for $M_{\max} = 150$. For $M_{\max} = 100$, already 51 subdomains have been created. However, we now obtain the same approximation quality as for the non-parametric case with about the same number of affine terms. In fact, the average number of affine terms over the 51 subdomains is already about 63, i.e., much lower than 92. For most subdivisions, one coefficient was enough for the assignment and in no situation, more than 3 coefficients have been used. Thus, the online assignment is very efficient.

It could be argued that it is possible to apply explicit partitioning methods just on the deterministic parameter domain $\mathcal{P}_1$ without considering the stochastic influences. However, using implicit methods and taking the stochasticity into account, we can go on with the partitioning and obtain affine approximations with even less terms than for the non-parametric case. E.g., using 124 subdomains, $M_{\max} = 72$ could be reached.

7.6.3 Application of the FS IPM to the Non-Parametric Case

To demonstrate that it is also useful to implicitly partition only stochastic influences, without any deterministic parameters, we created 2000 random samples for the fixed parameter $\mu_1 = 0.4$ and started the coefficient based FS IPM for $M_{\max} = 50$. We reached the desired accuracy $\varepsilon_{\text{tol}} = 10^{-5}$ using 25 leaf subdomains, where the average number of needed affine terms already decreased to 34. Hence, we could significantly decrease the number of affine terms.

It remains to show that it is still possible to apply the RBM theory of Chapter 5 that has been used in the case of non-parametric domains before. We perform separate KL expansions on each leaf subdomain using appropriate samples and construct independent reduced systems. It can be assumed that the number of KL modes that are necessary for appropriate approximations of the coefficients roughly coincide with the number of affine terms of the EIM, i.e., that $K_{\max} \approx M_{\max}$. In the online stage, the actual simulations can be performed independently on each subdomain.

However, for the evaluation of adequate statistical outputs such as mean or variance, we have to bring together the outputs of all subdomains. Furthermore, to apply Monte Carlo approximations, the availability of a probability measure is required that provides the probability that a certain subdomain is selected.

It is easy to construct such a discrete probability measure. We simulate a large number of random boundaries, using only the 13 random coefficients of the KL expansion of T. For each sample, we determine the appropriate subdomain and store the number of samples that have been assigned to each subdomain. Since this assignment is very fast compared to the run-time complexities of both full and reduced simulations, we can evaluate very accurate approximations of the probability measure. Furthermore, it is possible to do the construction already in the offline stage.

7.7 Conclusions

We showed the applicability of the RBM theory introduced in Chapters 5 and 6 to PPDEs on stochastic domains. We furthermore showed that the IPMs from Chapter 4 can be applied to such problems to increase the efficiency of the RBM.

For parametric *and* stochastic domains, we used the POIM from Chapter 3 to generate affine approximations of the parametric and stochastic coefficients. Now, the RB-EIM methodology that has been developed for deterministic problems can directly be applied to stochastic problems. We showed that the IPM can lead to a significant improvement of the online run-time complexity, especially for such complicated problems.

Chapter 8

Further Stochastic RBM Settings and Conclusions

In this chapter, we briefly describe further possible applications of the RBM to instationary, stochastic problems as well as to D-weak/Ω-weak settings (cf. Section 2.1.3), and we provide some ideas for future work. Finally, we summarize and discuss the main contributions of this work.

8.1 Instationary Problems

So far, we considered only stationary problems. However, it is straightforward to apply the introduced methodology also for instationary problems, as considered for example in [27, 38, 40]. On the one hand, the results of Chapters 3 and 4 directly provide connections of stochastic problems to the EIM such that the results of, e.g., [27, 38], for deterministic, non-affine, instationary problems could directly be applied. On the other hand, for affine problems with respect to deterministic parameters, as for example considered in [40], the methods provided in Chapters 5 and 6 can easily be extended.

Certainly, various other instationary frameworks with stochastic influences can be considered. E.g., stochastic PDEs (SPDEs) in the sense of Itô calculus play an important role in financial mathematics. Some work about model reduction in that context has been done in [22, 51, 52, 74], where, however, the stochasticity is not directly considered and partial integro-differential equations are solved. Hence,

studies about the RBM for Itô SPDEs, including the stochasticity, could be part of future work.

8.2 D-weak/Ω-weak RBMs

In Chapters 5 to 7, we considered RBMs in the context of Monte Carlo simulations, i.e., PDEs in a D-weak/Ω-strong context. However, as introduced in Chapter 2, other techniques to solve PDEs with stochastic influences are known. In this section, we will briefly describe how RBMs for deterministic parameters as well as the results in this work can applied to D-weak/Ω-weak formulations.

8.2.1 RBM for Stochastic Galerkin Methods

We have introduced stochastic Galerkin methods in Section 2.5 for the example problem (2.1). The KL expansion is employed on the stochastic coefficients of the PDE and the resulting random variables are modeled using polynomial chaos representations. Considering a finite element space X of dimension $\mathcal{N}$ and the space of the polynomial chaos S of dimension $P+1$, we obtain a weak formulation on the space $X \otimes S$ of dimension $\mathcal{N} \cdot (P+1)$.

Using the bilinear form $a : (X \otimes S) \times (X \otimes S) \to \mathbb{R}$ and the linear form $f : (X \otimes S) \to \mathbb{R}$ from (2.5), we derived the corresponding deterministic stiffness matrix $A \in \mathbb{R}^{\mathcal{N}(P+1)\times\mathcal{N}(P+1)}$ and the right-hand side $F \in \mathbb{R}^{\mathcal{N}(P+1)}$ in (2.22). In the following, we focus on the bilinear form a and the stiffness matrix A. All considerations can be analogously applied on the right-hand side.

We assume an additional, affine parameter dependence of the coefficient c that enters the bilinear form a, i.e., $c(x;\mu,\omega) = \sum_{q=1}^{Q} \theta_q(\mu) c_q(x;\omega)$. It is clear that for each component c_q, we can perform a KL expansion and evaluate the corresponding bilinear forms a_q and the stiffness matrices A_q analogously to (2.5) and (2.22), respectively.

Since all the system components are deterministic in the D-weak/Ω-weak setting, it is straightforward to apply the RBM. Let $u_n \in X \otimes S$, $n = 1, \dots, N$, denote a basis of the reduced space $(X \otimes S)_N$. For $A_{N,q} \in \mathbb{R}^{N\times N}$, $(A_{N,q})_{n,m} := a_q(u_m, u_n)$, we can assemble the reduced stiffness matrix $A_N(\mu) = \sum_{q=1}^{Q} \theta_q(\mu) A_{N,q}$ in $\mathcal{O}(QN^2)$ for each new parameter and the reduced solution can be obtained in $\mathcal{O}(N^3)$, as it

is well known from deterministic RBMs (cf., e.g., [73, 77]). Also, the further characteristics of deterministic RBMs can directly be applied to the problem. Linear output functionals and even expectations can be easily obtained independently of the dimension of the detailed problem. It is straightforward to define dual problems. Error bounds of the solution and the outputs can be evaluated, assuming coercivity or inf-sup stability of the bilinear form a, using the Riesz representatives of the components $a_q(u_n, \cdot)$.

It would be of interest and could be part of future work to compare reducibility of the D-weak/Ω-weak formulation with the Monte Carlo based settings developed in this work. Similar to RBMs for "space-time" formulations [83, 90], where weak solutions in space and time are considered, it can be hoped that the online run-time can be significantly reduced compared to the Monte Carlo approach.

However, this online run-time reduction is at the expense of extremely high offline costs. Recall that, in the non-parametric case, we have $P + 1 = \binom{K+r}{K}$, where K denotes the number of used KL terms and r denotes the maximal degree r of the polynomial chaos. Hence, the dimension increases exponentially fast in K. Suppose the coefficients $c_q(x;\omega)$, $q = 1, \dots, Q$, are stochastically independent and suppose each coefficient is approximated using K random variables. Then, the dimension of the full problem increases to $\mathcal{N} \cdot \binom{QK+r}{QK}$. E.g., for $K = 10$ and $r = 3$, the dimension would increases from 1,717 to 10,626. Hence, depending on the actual application, the RBM for stochastic Galerkin methods may rather be of theoretical interest than useful for actual applications.

8.2.2 RBM for Stochastic Collocation Methods

We have briefly introduced stochastic collocation methods in Section 2.6. Basically, the idea is to find D-weak/Ω-weak solutions, where the random space $S = L_2(\Omega)$ is approximated using polynomial interpolation. An interpolation point corresponds to a random realization of the random variables in the KL expansion. Hence, the approximation requires only evaluations of D-weak/Ω-strong solutions and the quality depends on the choice and number of interpolation points.

It is clear that the number of interpolation points is limited due to the high costs using the full finite element discretization. Hence, it seems to be a natural idea to combine the RBM as introduced in Chapters 5 to 7 with the stochastic

collocation method.

For a fine grid of interpolation points and an initial reduced basis, we can efficiently evaluate the reduced solutions on the whole grid and evaluate the corresponding error bounds. If the solutions are sufficiently accurate, we can stop the basis extension and use the RB approximations to generate the approximation of the space $X \otimes S$. Otherwise, we can use the Greedy approach — as we have described in Section 5.7.4 — to select the next basis function and iterate the procedure.

It could be part of future work to investigate how the RB and KL truncation error propagates in the interpolation of the random space S, i.e., how interpolation and RB/KL errors interact. Furthermore, it could be interesting how the approximation of outputs, random and statistical, can be optimized and if the dual problems can still be applied.

8.3 Conclusions

We presented a reduced basis framework for general linear and quadratically nonlinear parametric partial differential equations with stochastic influences. No specific assumptions regarding the random input has been used such that possible applications include random differential operators, right-hand sides, and boundary conditions. It is demonstrated that the RB methodology allows us to deal with large nonlinear parameterized systems involving significant stochastic deviations.

For problems that allow for an affine decomposition with respect to the deterministic parameter, we used the Karhunen-Loève (KL) expansion of the respective random terms to resolve and affinely decompose the stochasticity. The KL expansion is truncated for an additional reduction of the complexity. We derived efficient a-posteriori error bounds for the state and output functionals, also dealing with additional KL-truncation errors. Using additional non-standard dual problems, we furthermore introduced a new analysis for special quadratic outputs which could in particular be applied to efficiently approximate statistical quantities such as mean, second moment and variance. We provided new error bounds for such outputs, outperforming standard approximations, and showed that parts of the KL-truncation errors vanish. To some extend, the methodology could be used for higher moments

as well. However, the bounds loose some aspects of their efficiency. Hence, it depends on the actual problem if the evaluation of the additional dual problems pays off. Especially for the non-linear problems, where the non-linearity dominates the system, the additional linear dual problems are comparatively cheap, since the complexity corresponds just to one Newton iteration.

For stochastic problems that are not affine with respect to the parameter, we developed new forms of the EIM which are especially useful for noisy or non-smooth input data. Hence, apart from the stochastic context, there may be several further applications as well. We demonstrated that it is useful to add POD eigenfunctions instead of snapshots to generate the EIM basis. We proved that the described method produces the same approximation as the DEIM with less computational cost. It is shown that the EIM error estimators can therefore be used for both methods. Furthermore, the method has been extended to a least-squares problem, using more interpolation points than basis functions. In this way, we could improve the approximation quality and arrived at close to optimal results.

To reduce the number of affine terms and the dimension of the reduced space, we generalized the partitioning concepts for explicitly given deterministic, compact parameter domains to arbitrary input functions with possibly unknown or even without direct parameter dependencies. In other words, no a-priori information about the input is necessary. The so-called implicit partitioning methods are no more based upon distance measures in the parameter domain but upon the approximation quality of several different empirical interpolations. It is shown that the methods can efficiently be applied to both stochastic and deterministic problems and outperform the known methods for wide classes of problems. Furthermore, it is described how the methods can be used to decrease the online costs even for cases where no EIM is used for the generation of affine decompositions.

Eventually, we use all the presented methods and apply them to PPDEs with stochastic influences on parametric and/or stochastic domains. In other words, stochasticity can now be obtained in both coefficients and in the domain. Furthermore, it is briefly described how the methods can be used, e.g., for instationary problems and formulations weak in space and probability.

Appendix A

Alternative Variance Error Bound

The following error analysis for an approximated Variance is adopted from [12]. It is used for the numerical examples in Chapters 5 and 6 and is referred to as sophisticated variance error bound.

Let $s(\omega)$ be a random variable with expectation $\mathbb{M}_1$ and variance $\mathbb{V}$. Furthermore, let $s_{N,K}(\omega)$ be an approximation of $s(\omega)$ such that $|s(\omega) - s_{N,K}(\omega)| \leq \Delta^s(\omega)$ as for example provided in Chapters 5 and 6. The expectation of the approximated random variable and of the error bound are denoted by $\mathbb{M}_{1,NK}$ and $\Delta^{\mathbb{M}_1}$, respectively. The variance of the approximated random variable $s_{N,K}$ is denoted by $\mathbb{V}_{NK}$ and serves as an estimation of the variance $\mathbb{V}$. We define

$$\begin{aligned} s^-(\omega) &:= s_{N,K}(\omega) - \Delta^s(\omega), & s^+ &:= s_{N,K}(\omega) + \Delta^s(\omega), \\ \mathbb{M}_1^- &:= \mathbb{M}_{1,NK} - \Delta^{\mathbb{M}_1}, & \mathbb{M}_1^+ &:= \mathbb{M}_{1,NK} + \Delta^{\mathbb{M}_1}, \end{aligned}$$

such that $s^-(\omega) \leq s(\omega) \leq s^+(\omega)$ and $\mathbb{M}_1^- \leq \mathbb{M}_1 \leq \mathbb{M}_1^+$. Next, we define

$$\sigma^-(\omega) = \mathbb{M}_1^- - s^+(\omega), \qquad \sigma^+(\omega) = \mathbb{M}_1^+ - s^-(\omega),$$

and obtain $\sigma^-(\omega) \leq \mathbb{M}_1 - s(\omega) \leq \sigma^+(\omega)$. Let $V^-(\omega)$ and $V^+(\omega)$ be given by

$$\begin{aligned} V^-(\omega) &:= \begin{cases} \min\{|\sigma^-(\omega)|, |\sigma^+(\omega)|\}, & \text{if } \sigma^+(\omega) > \sigma^-(\omega), \\ 0, & \text{else}, \end{cases} \\ V^+(\omega) &:= \max\{|\sigma^-(\omega)|, |\sigma^+(\omega)|\}. \end{aligned}$$

It can be concluded that $(V^-(\omega))^2 \leq (\mathbb{M}_1 - s(\omega))^2 \leq (V^+(\omega))^2$ such that

$$\mathbb{E}\left[\left(V^-(\cdot)\right)^2\right] \leq \mathbb{V} \leq \mathbb{E}\left[\left(V^+(\cdot)\right)^2\right].$$

Then, it is clear that a bound for the approximation error $|\mathbb{V}-\mathbb{V}_{NK}|$ of the variance is given by

$$|\mathbb{V} - \mathbb{V}_{NK}| \;\leq\; \max\left\{\left|\mathbb{V}_{NK} - \mathbb{E}\left[\left(V^{-}(\cdot)\right)^{2}\right]\right|, \left|\mathbb{V}_{NK} - \mathbb{E}\left[\left(V^{+}(\cdot)\right)^{2}\right]\right|\right\}.$$

Bibliography

[1] F. Albrecht, B. Haasdonk, S. Kaulmann, and M. Ohlberger. The localized reduced basis multiscale method. In A. Handlovičová, Z. Minarechové, and D. Ševčovič, editors, *Proceedings of ALGORITMY 2012, Vysoke Tatry, Podbanske, September 9-14, 2012*, pages 393–403, 2012.

[2] W. Arendt and K. Urban. *Partielle Differenzialgleichungen. Eine Einführung in analytische und numerische Methoden.* Spektrum Akademischer Verlag Heidelberg, 2010.

[3] I. Babuška. Error-bounds for finite element method. *Numer. Math.*, 16:322–333, 1970/1971.

[4] I. M. Babuška and P. Chatzipantelidis. On solving elliptic stochastic partial differential equations. *Comput. Methods Appl. Mech. Engrg.*, 191(37-38):4093 – 4122, 2002.

[5] I. M. Babuška and J. Chleboun. Effects of uncertainties in the domain on the solution of Neumann boundary value problems in two spatial dimensions. *Math. Comp.*, 71(240):1339–1370 (electronic), 2002.

[6] I. M. Babuška, F. Nobile, and R. Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM J. Numer. Anal.*, 45(3):1005–1034 (electronic), 2007.

[7] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera. An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. *C. R. Math. Acad. Sci. Paris*, 339(9):667–672, 2004.

[8] A. Barth, A. Lang, and C. Schwab. Multilevel Monte Carlo method for parabolic stochastic partial differential equations. *BIT*, 53(1):3–27, 2013.

[9] J. Beck, F. Nobile, L. Tamellini, and R. Tempone. Implementation of optimal Galerkin and collocation approximations of PDEs with random coefficients. In *CANUM 2010,* 40^{e} *Congrès National d'Analyse Numérique*, volume 33 of *ESAIM Proc.*, pages 10–21. EDP Sci., Les Ulis, 2011.

[10] J. Beck, R. Tempone, F. Nobile, and L. Tamellini. On the optimal polynomial approximation of stochastic PDEs by Galerkin and collocation methods. *Math. Models Methods Appl. Sci.*, 22(9):1250023, 33, 2012.

[11] S. Boyaval. A fast Monte Carlo method with a reduced basis of control variates applied to uncertainty propagation and Bayesian estimation. *Comput. Methods Appl. Mech. Engrg.*, 241–244(0):190–205, 2012.

[12] S. Boyaval, C. Le Bris, Y. Maday, N. C. Nguyen, and A. T. Patera. A reduced basis approach for variational problems with stochastic parameters: application to heat conduction with variable Robin coefficient. *Comput. Methods Appl. Mech. Engrg.*, 198(41-44):3187–3206, 2009.

[13] F. Brezzi, J. Rappaz, and P.-A. Raviart. Finite-dimensional approximation of nonlinear problems. I. Branches of nonsingular solutions. *Numer. Math.*, 36(1):1–25, 1980/81.

[14] A. Buffa, Y. Maday, A. T. Patera, C. Prud'homme, and G. Turinici. *A priori* convergence of the greedy algorithm for the parametrized reduced basis method. *ESAIM Math. Model. Numer. Anal.*, 46(3):595–603, 2012.

[15] R. E. Caflisch. Monte Carlo and quasi-Monte Carlo methods. In *Acta numerica, 1998*, volume 7 of *Acta Numer.*, pages 1–49. Cambridge Univ. Press, Cambridge, 1998.

[16] G. Caloz and J. Rappaz. Numerical analysis for nonlinear and bifurcation problems. In *Handbook of numerical analysis, Vol. V*, Handb. Numer. Anal., V, pages 487–637. North-Holland, Amsterdam, 1997.

[17] C. Canuto and T. Kozubek. A fictitious domain approach to the numerical solution of PDEs in stochastic domains. *Numer. Math.*, 107(2):257–293, 2007.

[18] C. Canuto, T. Tonn, and K. Urban. A posteriori error analysis of the reduced basis method for nonaffine parametrized nonlinear PDEs. *SIAM J. Numer. Anal.*, 47(3):2001–2022, 2009.

[19] S. Chaturantabut and D. Sorensen. Discrete empirical interpolation for nonlinear model reduction. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 4316–4321, dec. 2009.

[20] S. Chaturantabut and D. C. Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM J. Sci. Comput.*, 32(5):2737–2764, 2010.

[21] K. A. Cliffe, M. B. Giles, R. Scheichl, and A. L. Teckentrup. Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients. *Comput. Vis. Sci.*, 14(1):3–15, 2011.

[22] R. Cont, N. Lantos, and O. Pironneau. A reduced basis for option pricing. *SIAM J. Financial Math.*, 2:287–316, 2011.

[23] M. Cwikel and E. Pustylnik. Sobolev type embeddings in the limiting case. *J. Fourier Anal. Appl.*, 4(4-5):433–446, 1998.

[24] S. Deparis. Reduced basis error bound computation of parameter-dependent Navier-Stokes equations by the natural norm approach. *SIAM J. Numer. Anal.*, 46(4):2039–2067, 2008.

[25] M. Dihlmann, M. Drohmann, and B. Haasdonk. Model reduction of parametrized evolution problems using the reduced basis method with adaptive time partitioning. In D. Aubry, P. Diez, B. Tie, and N. Pares, editors, *Adaptive Modeling and Simulation 2011*, pages 156–167. CIMNE, 2011.

[26] M. Dihlmann and B. Haasdonk. Certified PDE-constrained parameter optimization using reduced basis surrogate models for evolution problems.

Preprint, SimTech – Cluster of Excellence, Universität Stuttgart, February 2013.

[27] M. Drohmann, B. Haasdonk, and M. Ohlberger. Reduced basis approximation for nonlinear parametrized evolution equations based on empirical operator interpolation. *SIAM Journal on Scientific Computing*, 34(2):A937–A969, 2012.

[28] J. L. Eftang, D. B. P. Huynh, D. J. Knezevic, E. M. Rønquist, and A. T. Patera. Port reduction in component-based static condensation for parametrized problems: Approximation and a posteriori error estimation. In F. Breitenecker and I. Troch, editors, *Proceedings MATHMOD 2012, 7th Vienna International Conference on Mathematical Modelling*, volume 7, pages 695–699, 2012.

[29] J. L. Eftang, D. J. Knezevic, and A. T. Patera. An *hp* certified reduced basis method for parametrized parabolic partial differential equations. *Math. Comput. Model. Dyn. Syst.*, 17(4):395–422, 2011.

[30] J. L. Eftang, A. T. Patera, and E. M. Rønquist. An "*hp*" certified reduced basis method for parametrized elliptic partial differential equations. *SIAM J. Sci. Comput.*, 32(6):3170–3200, 2010.

[31] J. L. Eftang and B. Stamm. Parameter multi-domain hp empirical interpolation. In Thesis: J. L. Eftang. Reduced Basis Methods for Parametrized Partial Differential Equations. 2011., May 2011.

[32] G. S. Fishman. *Monte Carlo: Concepts, algorithms, and applications*. Springer Series in Operations Research. Springer-Verlag, New York, 1996.

[33] A.-L. Gerner and K. Veroy. Reduced basis *a posteriori* error bounds for the Stokes equations in parametrized domains: a penalty approach. *Math. Models Methods Appl. Sci.*, 21(10):2103–2134, 2011.

[34] A.-L. Gerner and K. Veroy. Certified Reduced Basis Methods for Parametrized Saddle Point Problems. *SIAM J. Sci. Comput.*, 34(5):A2812–A2836, 2012.

[35] R. G. Ghanem and P. D. Spanos. *Stochastic finite elements: A spectral approach*. Springer-Verlag, New York, 1991.

[36] W. J. Gordon and C. A. Hall. Construction of curvilinear co-ordinate systems and applications to mesh generation. *Internat. J. Numer. Methods Engrg.*, 7:461–477, 1973.

[37] W. J. Gordon and C. A. Hall. Transfinite element methods: blending-function interpolation over arbitrary curved element domains. *Numer. Math.*, 21:109–129, 1973/74.

[38] M. A. Grepl. *Reduced-Basis Approximation and A Posteriori Error Estimation for Parabolic Partial Differential Equations*. Phd in mechanical engineering, Massachusetts Institute of Technology, June 2005.

[39] M. A. Grepl and M. Kärcher. Reduced basis a posteriori error bounds for parametrized linear-quadratic elliptic optimal control problems. *C. R. Math. Acad. Sci. Paris*, 349(15-16):873–877, 2011.

[40] M. A. Grepl and A. T. Patera. A posteriori error bounds for reduced-bias approximations of parametrized parabolic partial differential equations. *M2AN Math. Model. Numer. Anal.*, 39(1):157–181, 2005.

[41] B. Haasdonk, M. Dihlmann, and M. Ohlberger. A training set and multiple bases generation approach for parameterized model reduction based on adaptive grids in parameter space. *Math. Comput. Model. Dyn. Syst.*, 17(4):423–442, 2011.

[42] B. Haasdonk and M. Ohlberger. Reduced basis method for finite volume approximations of parametrized linear evolution equations. *M2AN Math. Model. Numer. Anal.*, 42(2):277–302, 2008.

[43] B. Haasdonk, M. Ohlberger, and G. Rozza. A reduced basis method for evolution schemes with parameter-dependent explicit operators. *ETNA, Electronic Transactions on Numerical Analysis*, 32:145–161, 2008.

[44] B. Haasdonk, J. Salomon, and B. Wohlmuth. A Reduced Basis Method for Parametrized Variational Inequalities. *SIAM J. Numer. Anal.*, 50(5):2656–2676, 2012.

[45] B. Haasdonk, K. Urban, and B. Wieland. Reduced basis methods for parametrized partial differential equations with stochastic influences using the Karhunen-Loève expansion. *SIAM/ASA J. Uncertainty Quantification*, 1:79–105, 2013.

[46] H. Hadinejad-Mahram, D. Dahlhaus, and D. Blömker. Karhunen-Loève expansion of vecor random processes. Technical report, Communications Technology Laboratory, Swiss Federal Institute of Technology Zürich, 2002.

[47] R. A. Handler, K. D. Housiadas, and A. N. Beris. Karhunen-Loeve representations of turbulent channel flows using the method of snapshots. *Internat. J. Numer. Methods Fluids*, 52(12):1339–1360, 2006.

[48] H. Harbrecht. A finite element method for elliptic problems with stochastic input data. *Appl. Numer. Math.*, 60(3):227–244, 2010.

[49] H. Harbrecht. On output functionals of boundary value problems on stochastic domains. *Math. Methods Appl. Sci.*, 33(1):91–102, 2010.

[50] H. Harbrecht, R. Schneider, and C. Schwab. Sparse second moment analysis for elliptic problems in stochastic domains. *Numer. Math.*, 109(3):385–414, 2008.

[51] P. Hepperger. Option pricing in Hilbert space-valued jump-diffusion models using partial integro-differential equations. *SIAM J. Financial Math.*, 1:454–489, 2010.

[52] P. Hepperger. Hedging electricity swaptions using partial integro-differential equations. *Stochastic Process. Appl.*, 122(2):600–622, 2012.

[53] D. B. P. Huynh, D. J. Knezevic, and A. T. Patera. A static condensation reduced basis element method: approximation and *a posteriori* error estimation. *ESAIM Math. Model. Numer. Anal.*, 47(1):213–251, 2013.

[54] D. B. P. Huynh and A. T. Patera. Reduced basis approximation and a posteriori error estimation for stress intensity factors. *Internat. J. Numer. Methods Engrg*, 72(10):1219–1259, 2007.

[55] D. B. P. Huynh and A. T. Patera. Reduced basis approximation and a posteriori error estimation for stress intensity factors. *International Journal for Numerical Methods in Engineering*, 72(10):1219–1259, 2007.

[56] D. B. P. Huynh, J. Peraire, A. T. Patera, and G. R. Liu. Real-time reliable prediction of linear-elastic mode-i stress intensity factors for failure analysis. In *Proceedings of the 6th Singapore-MIT Alliance Annual Symposium, Cambridge, MA*, 2006.

[57] D. B. P. Huynh, G. Rozza, S. Sen, and A. T. Patera. A successive constraint linear optimization method for lower bounds of parametric coercivity and inf-sup stability constants. *C. R. Math. Acad. Sci. Paris*, 345(8):473 – 478, 2007.

[58] L. Iapichino, A. Quarteroni, and G. Rozza. A reduced basis hybrid method for the coupling of parametrized domains represented by fluidic networks. *Comput. Methods Appl. Mech. Engrg.*, 221/222:63–82, 2012.

[59] M. Kärcher and M. A. Grepl. A certified reduced basis method for parametrized elliptic optimal control problems. Preprint, RWTH Aachen University, 2012.

[60] K. Karhunen. Über lineare Methoden in der Wahrscheinlichkeitsrechnung. *Ann. Acad. Sci. Fennicae. Ser. A. I. Math.-Phys.*, 1947(37):79, 1947.

[61] S. Kaulmann and B. Haasdonk. Online greedy reduced basis construction using dictionaries. Preprint, SimTech – Cluster of Excellence, Universität Stuttgart, March 2013.

[62] K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM J. Numer. Anal.*, 40(2):492–515, 2002.

[63] T. Lassila and G. Rozza. Parametric free-form shape design with PDE models and reduced basis method. *Comput. Methods Appl. Mech. Engrg.*, 199(23-24):1583–1592, 2010.

[64] W. K. Liu, T. Belytschko, and A. Mani. Random field finite elements. *Internat. J. Numer. Methods Engrg.*, 23(10):1831–1845, 1986.

[65] M. Loève. *Probability theory. II.* Springer-Verlag, New York, fourth edition, 1978. Grad. Texts in Math. 46.

[66] Y. Maday and E. M. Rønquist. A reduced-basis element method. In *Proceedings of the Fifth International Conference on Spectral and High Order Methods (ICOSAHOM-01) (Uppsala)*, volume 17, pages 447–459, 2002.

[67] Y. Maday and E. M. Rønquist. The reduced basis element method: application to a thermal fin problem. *SIAM J. Sci. Comput.*, 26(1):240–258 (electronic), 2004.

[68] S. Mallat and W. L. Hwang. Singularity detection and processing with wavelets. *IEEE Trans. Inform. Theory*, 38(2, part 2):617–643, 1992.

[69] H. G. Matthies and A. Keese. Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations. *Comput. Methods Appl. Mech. Engrg.*, 194(12-16):1295–1331, 2005.

[70] P. S. Mohan, P. B. Nair, and A. J. Keane. Multi-element stochastic reduced basis methods. *Comput. Methods Appl. Mech. Engrg.*, 197(17-18):1495–1506, 2008.

[71] N. C. Nguyen and J. Peraire. An interpolation method for the reconstruction and recognition of face images. In *VISAPP (2)*, pages 91–96, 2007.

[72] M. Papadrakakis and V. Papadopoulos. Robust and efficient methods for stochastic finite element analysis using Monte Carlo simulation. *Comput. Methods Appl. Mech. Engrg.*, 134(3-4):325–340, 1996.

[73] A. T. Patera and G. Rozza. Reduced Basis Approximation and A Posteriori Error Estimation for Parametrized Partial Differential Equations. Version 1.0, MIT, Cambridge, MA, 2006.

[74] O. Pironneau. Calibration of options on a reduced basis. *J. Comput. Appl. Math.*, 232(1):139–147, 2009.

[75] D. V. Rovas. *Reduced-Basis Output Bound Methods for Parametrized Partial Differential Equations*. PhD thesis, Massachusetts Institute of Technology, February 2003.

[76] D. V. Rovas, L. Machiels, and Y. Maday. Reduced-basis output bound methods for parabolic problems. *IMA J. Numer. Anal.*, 26(3):423–445, 2006.

[77] G. Rozza, D. B. P. Huynh, and A. T. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations: Application to transport and continuum mechanics. *Arch. Comput. Methods Eng.*, 15(3):229–275, 2008.

[78] G. Rozza and K. Veroy. On the stability of the reduced basis method for Stokes equations in parametrized domains. *Comput. Methods Appl. Mech. Engrg.*, 196(7):1244–1260, 2007.

[79] S. K. Sachdeva, P. B. Nair, and A. J. Keane. Hybridization of stochastic reduced basis methods with polynomial chaos expansions. *Probabilistic Engineering Mechanics*, 21(2):182–192, 2006.

[80] M. Shinozuka and G. Deodatis. Response variability of stochastic finite element systems. *Journal of Engineering Mechanics*, 114(3):499–519, 1988.

[81] U. Simon, P. Augat, M. Utz, and L. Claes. A numerical model of the fracture healing process that describes tissue development and revascularisation. *Computer Methods in Biomechanics and Biomedical Engineering*, 14(1):79–93, 2011. PMID: 21086207.

[82] L. Sirovich. Turbulence and the dynamics of coherent structures. I. Coherent structures. *Quart. Appl. Math.*, 45(3):561–571, 1987.

[83] K. Steih and K. Urban. Space-time reduced basis methods for time-periodic partial differential equations. In *Proceedings MATHMOD 2012, 7th Vienna International Conference on Mathematical Modelling (accepted)*, 2012.

[84] D. M. Tartakovsky and D. Xiu. Stochastic analysis of transport in tubes with rough walls. *J. Comput. Phys.*, 217(1):248–259, 2006.

[85] J. F. Thompson, Z. U. A. Warsi, and C. W. Mastin. *Numerical grid generation*. North-Holland Publishing Co., New York, 1985. Foundations and applications.

[86] T. Tonn. *Reduced-Basis Method (RBM) for Non-Affine Elliptic Parametrized PDEs (Motivated by Optimization in Hydromechanics)*. PhD thesis, Ulm University, Ulm, Germany, 2012.

[87] T. Tonn and K. Urban. A reduced-basis method for solving parameter-dependent convection-diffusion problems around rigid bodies. In *Proceedings of the ECCOMAS CFD*, 2006.

[88] T. Tonn, K. Urban, and S. Volkwein. Optimal control of parameter-dependent convection-diffusion problems around rigid bodies. *SIAM J. Sci. Comput.*, 32(3):1237–1260, 2010.

[89] N. Trudinger. On imbeddings into orlicz spaces and some applications. *Indiana Univ. Math. J.*, 17:473–483, 1968.

[90] K. Urban and A. T. Patera. A new error bound for reduced basis approximation of parabolic partial differential equations. *C. R. Math. Acad. Sci. Paris*, 350(3-4):203–207, 2012.

[91] K. Urban, S. Volkwein, and O. Zeeb. Greedy sampling using nonlinear optimization. Preprint, Ulm University, 2012.

[92] K. Urban and B. Wieland. Affine decompositions of parametric stochastic processes for application within reduced basis methods. In F. Breitenecker and I. Troch, editors, *Proceedings MATHMOD 2012, 7th Vienna International Conference on Mathematical Modelling*, volume 7, pages 716–721, 2012.

[93] K. Urban and B. Wieland. Reduced basis methods for quadratically nonlinear partial differential equations with stochastic influences. In J. Eberhardsteiner, H. J. Böhm, and F. G. Rammerstorfer, editors, *CD-ROM Proceedings*

of the 6th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2012), September 10-14, 2012, Vienna, Austria. Vienna University of Technology, Austria, September 2012.

[94] S. Vallaghé and A. T. Patera. The static condensation reduced basis element method for a mixed-mean conjugate heat exchanger model. Preprint, MIT, Cambridge, MA, 2012 August.

[95] E. Vanmarcke and M. Grigoriu. Stochastic finite element analysis of simple beams. *Journal of Engineering Mechanics*, 109(5):1203–1214, 1983.

[96] K. Veroy and A. T. Patera. Certified real-time solution of the parametrized steady incompressible Navier-Stokes equations: rigorous reduced-basis a posteriori error bounds. *Internat. J. Numer. Methods Fluids*, 47(8-9):773–788, 2005.

[97] K. Veroy, C. Prud'homme, and A. T. Patera. Reduced-basis approximation of the viscous Burgers equation: rigorous a posteriori error bounds. *C. R. Math. Acad. Sci. Paris*, 337(9):619–624, 2003.

[98] K. Veroy, C. Prud'homme, D. V. Rovas, and A. T. Patera. A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations. In *AIAA paper 2003-3847, Proceedings of the 16th AIAA Computational Fluid Dynamics Conference*, 2003.

[99] T. Wehner, L. Claes, F. Niemeyer, D. Nolte, and U. Simon. Influence of the fixation stability on the healing time—a numerical study of a patient-specific fracture healing process. *Clinical Biomechanics*, 25(6):606–612, 2010.

[100] B. Wieland. Speech signal noise reduction with wavelets. Diploma Thesis, Ulm University, Ulm, Germany, October 2009.

[101] N. Wiener. The homogeneous chaos. *Amer. J. Math.*, 60(4):897–936, 1938.

[102] D. Xiu and J. S. Hesthaven. High-order collocation methods for differential equations with random inputs. *SIAM J. Sci. Comput.*, 27(3):1118–1139, 2005.

[103] D. Xiu and G. E. Karniadakis. The Wiener–Askey polynomial chaos for stochastic differential equations. *SIAM J. Sci. Comput.*, 24(2):619–644, 2002.

[104] D. Xiu and D. M. Tartakovsky. Numerical methods for differential equations in random domains. *SIAM J. Sci. Comput.*, 28(3):1167–1185 (electronic), 2006.

[105] F. Yamazaki, M. Shinozuka, and G. Dasgupta. Neumann expansion for stochastic finite element analysis. *Journal of Engineering Mechanics*, 114(8):1335–1354, 1988.

[106] M. Yano, A. T. Patera, and K. Urban. A space-time certified reduced basis method for Burgers' equation. Preprint, Ulm University, July 2012.

Lebenslauf

Persönliche Daten

Bernhard Wieland
Geb. am 22. Juli 1983 in Stuttgart–Bad Cannstatt

Schulbildung

09/1990–08/1994	Grundschule Eichbergschule, Leinfelden–Echterdingen
09/1994–07/2003	Immanuel–Kant–Gymnasium, Leinfelden–Echterdingen Abschluss: Abitur

Studium

10/2003–10/2009	Mathematik mit Nebenfach Informatik, Universität Ulm Abschluss: Diplom Diplomarbeit: *Speech Signal Noise Reduction with Wavelets*
08/2007–05/2008	Applied Mathematics, Florida Institute of Technology, Melbourne, FL, USA Abschluss: Master of Science
seit 10/2009	Promotionsstudium, Institut für Numerische Mathematik, Universität Ulm

Stipendien und Auszeichnungen

07/2003	Sozialpreis des Vereins der Freunde des Immanuel–Kant–Gymnasiums für besonderes soziales Engagement
07/2003	Preis für herausragende Leistungen im Abitur
08/2007–05/2008	Fulbright Stipendium
12/2009–11/2012	Promotionsstipendium nach dem Landesgraduiertenförderungsgesetz (LGFG)

Universitäre Beschäftigungen

08/2004–07/2007	Studentische Hilfskraft an der Universität Ulm
08/2007–05/2008	Teaching Assistant in *Calculus II*, Florida Institute of Technology, Melbourne, FL, USA
07/2008–09/2009	Wissenschaftliche Hilfskraft am Institut für Numerische Mathematik, Universität Ulm

seit 10/2009	Wissenschaftlicher Mitarbeiter am Ulmer Zentrum für Wissenschaftliches Rechnen, Universität Ulm
seit 12/2012	Wissenschaftlicher Mitarbeiter am Institut für Numerische Mathematik, Universität Ulm

Universitäre Gremien

10/2005–09/2007	Mitglied der Studienkommission Mathematik, Fakultät für Mathematik und Wirtschaftswissenschaften, Universität Ulm
10/2006–09/2007	Mitglied des Fakultätsrats der Fakultät für Mathematik und Wirtschaftswissenschaften, Universität Ulm

Berufserfahrung

02/2007–04/2007	Praktikum bei der Voith Paper AG, Heidenheim: Entwicklung numerischer Methoden zur Approximation der Einschwingzeit linearer Kontrollsysteme
06/2009–07/2009	Praktikum bei der d-fine GmbH: Projektarbeit zur Verbesserung der Berechnung von Marktpreisrisiken einer großen deutschen Bank

Ulm, 15. April 2013

Publikationen und Vorträge

Publikationen

01/2012	K. Urban and B. Wieland. Affine decompositions of parametric stochastic processes for application within reduced basis methods. In *Proceedings MATHMOD 2012, 7th Vienna International Conference on Mathematical Modelling*, 2012.
09/2012	K. Urban and B. Wieland. Reduced basis methods for quadratically nonlinear partial differential equations with stochastic influences. In J. Eberhardsteiner, H. J. Böhm, and F. G. Rammerstorfer, editors, *CD-ROM Proceedings of the 6th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2012), September 10-14, 2012, Vienna, Austria*. Vienna University of Technology, Austria, September 2012.
03/2013	B. Haasdonk, K. Urban, and B. Wieland. Reduced basis methods for parametrized partial differential equations with stochastic influences using the Karhunen-Loève expansion. *SIAM/ASA J. Uncertainty Quantification*, 1:79–105, 2013.
in Bearbeitung	B. Wieland. Implicit Partitioning Methods for Unknown Parameter Domains.

Ausgewählte Vorträge

07/2010	Reduced Basis Methods for Parametric PDEs with Stochastic Influences. Summer School Optimal Control of PDEs, Cortona, Italy.
12/2010	Reduced Basis Methods for PDEs with Stochastic Influences. Workshop on Reduced Basis Methods, Ulm.
10/2011	Reduced Basis Methods for PDEs on Stochastic Domains. Summer School on Reduced Basis Methods, Günzburg Reisensburg.
02/2012	Affine Decompositions of Parametric Stochastic Processes for Application within Reduced Basis Methods. MathMod, Vienna Conference on Mathematical Modelling, Vienna, Austria.
08/2012	An Implicit Partitioning Method for Unknown Parameter Domains

	(in the context of RBMs with stochastic influences). Workshop on Reduced Basis Methods, Freudenstadt
09/2012	Reduced Basis Methods for quadratically nonlinear PDEs with stochastic influences. ECCOMAS 2012, 6th European Conference on Computational Methods in Applied Sciences and Engineering, Vienna, Austria.
10/2012	An Implicit Partitioning Method for Unknown Parameter Domains. Second International Workshop on Model Reduction for Parametrized Systems (MoRePaS II), Schloss Reisensburg, Günzburg.
01/2013	Reduced Basis Methods for parametrized PDEs with stochastic influences. 29th GAMM-Seminar Leipzig on Numerical Methods for Uncertainty Quantification, Max Planck Institute for Mathematics in the Sciences, Leipzig.

Danksagungen

An dieser Stelle möchte ich mich bei allen herzlich bedanken, die durch fachliche und persönliche Unterstützung am Gelingen dieser Doktorarbeit beigetragen haben.

Zuerst geht mein besonderer Dank an Prof. Dr. Karsten Urban, der bei mir das Interesse am Themengebiet der Reduzierten–Basis–Methoden geweckt hat und der es mir ermöglichte, mich in den letzten Jahren intensiv damit zu beschäftigen. Viele seiner Anregungen aus zahlreichen Diskussionen sind in die Arbeit eingeflossen. Zudem war seine intensive Betreuung und Wertschätzung meiner Forschungsarbeit stets zusätzliche Motivation.

Mein Dank gilt weiterhin Jun.-Prof. Dr. Bernard Haasdonk, der sich trotz der räumlichen Distanz bereit erklärt hat, als Zweitbetreuer und -gutachter zu fungieren. Bei zahlreichen Gelegenheiten gab er mir wertvolle Hinweise und trug dazu bei, Probleme und Fragestellungen aus neuen Blickwinkeln zu betrachten.

Zudem bedanke ich mich bei Dr.-Ing. Ulrich Simon. Die Betreuung der Übung seiner Vorlesungen machte mir stets großen Spaß. Dabei profitierte ich insbesondere von seinem großen Fachwissen und lernte bei zahlreichen Diskussionen selbst viel über mechanische Problemstellungen und Lösungsansätze. Das von ihm entgegengebrachte Vertrauen, ihn bei zahlreichen Firmenkontakten des Ulmer Zentrums für Wissenschaftliches Rechnen begleiten zu dürfen, freute mich besonders, und führte zu vielen interessanten Einblicken.

Weiter möchte ich mich bei allen Mitarbeitern des Instituts für Numerische Mathematik und des Ulmer Zentrums für Wissenschaftliches Rechnen der Universität Ulm bedanken. Die Kollegialität, Hilfsbereitschaft und das fachliche Wissen waren eine Stütze bei der täglichen Arbeit und viele Freundschaften sind dabei entstanden. Für viele wertvolle Diskussionen bedanke ich mich bei Timo Tonn, der mir

insbesondere am Anfang meiner Promotion sehr hilfreich war, und der gesamten „RB–Runde“. Besonders danke ich auch Julia Springer, Theresa Springer, Kristina Steih, Silke Glas, Antonia Mayerhofer, Steffen Baumann, Mladjan Radic und Oliver Zeeb für das Korrekturlesen dieser Arbeit.

Schließlich möchte ich mich von ganzem Herzen bei meinen Eltern Elfriede und Wolfgang Wieland bedanken, die es mir ermöglicht haben, Mathematik zu studieren und mir darüberhinaus zu allen Zeiten eine große Unterstützung waren.

Danke!

Erklärung

Hiermit versichere ich, Bernhard Wieland, dass ich die vorliegende Arbeit selbständig angefertigt habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie die wörtlich oder inhaltlich übernommenen Stellen als solche kenntlich gemacht habe. Ich erkläre außerdem, dass diese Arbeit weder im In- noch im Ausland in dieser oder ähnlicher Form in einem anderen Promotionsverfahren vorgelegt wurde.

Ulm, 3. Juli 2013

Bernhard Wieland