B-Splines on Sparse Grids for Uncertainty Quantification

Vom Stuttgarter Zentrum für Simulationswissenschaften (SC SimTech) und
der Fakultät für Informatik, Elektrotechnik und Informationstechnik
der Universität Stuttgart zur Erlangung der Würde eines Doktors
der Naturwissenschaften (Dr. rer. nat.) genehmigte Abhandlung

Vorgelegt von

Michael F. Rehme

aus Tübingen

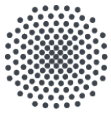
Hauptberichter: Prof. Dr. Dirk Pflüger

Mitberichter: Prof. Dr. Stephen Roberts

Tag der mündlichen Prüfung: 22. Juli 2021

Institut für Parallele und Verteilte Systeme der Universität Stuttgart

2021



University of Stuttgart
Germany

Submitted to the University of Stuttgart

Involved institutions and departments:
Cluster of Excellence in Simulation Technology
Institute for Parallel and Distributed Systems
Chair of Scientific Computing
Chair of Simulation of Large Systems
Mathematical Sciences Institute of the
Australian National University

Australian
National
University

Michael F. Rehme
Scientific Computing
Institute for Parallel and Distributed Systems
University of Stuttgart
Universitätsstr. 38
70569 Stuttgart
Germany

D 93 (dissertation)

Parts of this thesis were funded by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2075 390740016. We acknowledge the support by the Stuttgart Center for Simulation Science (SimTech).

Contents

Lists of Figures, Tables, Algorithms, and Theorems

List of Figures

List of Tables

List of Algorithms

List of Theorems

List of Symbols and Acronyms

Symbol	Meaning	Page with first occurrence
b^n	Uniform standard B-spline of degree n, defined on knots $(0, 1, \dots, n+1)$	44
$b^{n,\mathrm{nak}}_{\ell,k}$	Not-a-knot B-spline of degree n, level ℓ and index k	50
$b^{n,\mathrm{m}}_{\ell,k}$	Modified not-a-knot B-spline of degree n, level ℓ and index k	30
$b^{n,\mathrm{e}}_{\ell,k}$	Extended not-a-knot B-spline of degree n, level ℓ and index k	54
$b^{n,\mathrm{bnk}}_{\ell,k}$	Boundaryless not-a-knot B-spline of degree n, level ℓ and index k	55
$C(\boldsymbol{\ell}, \boldsymbol{i})$	Hierarchical children of gridpoint $x_{\boldsymbol{\ell},\boldsymbol{i}}$	37
D	Dimensionality $D \in \mathbb{N}$ (number of input parameters)	22
$\mathbb{E}$	Expected value	69
f	Objective function / model $f : [0,1]^D \to \mathbb{R}$	24
$\hat{f}_\ell$	Full grid interpolant of level ℓ	24
$\tilde{f}_\ell$	Sparse grid interpolant of level ℓ	26
$\tilde{f}$	Sparse grid interpolant with respect to an arbitrary level-index set	39
$h_{\boldsymbol{\ell}}$	Grid width $h_{\boldsymbol{\ell}} \coloneqq \mathbf{2}^{-\boldsymbol{\ell}}$	24
$H^{\mathbf{2}}_{\mathrm{mix}}$	Sobolev space with dominating mixed derivatives	23
$H^{\mathbf{2}}_{0,\mathrm{mix}}$	Functions $f \in H^{\mathbf{2}}_{\mathrm{mix}}$ which vanish on the boundary	23
I	Finite set of hierarchical level-index pairs $(\boldsymbol{\ell}, \boldsymbol{i})$	37
$I_{\boldsymbol{\ell}}$	Set of odd hierarchical indices of level $\boldsymbol{\ell}$	25
I^{ref}	Set of level-index pairs $(\boldsymbol{\ell}, \boldsymbol{i})$ of refineable grid points	37
$K_{\boldsymbol{\ell}}$	Full set of indices of level $\boldsymbol{\ell}$	24
$K^{\mathrm{bnk}}_{\boldsymbol{\ell}}$	Full set of indices of level $\boldsymbol{\ell}$ for boundaryless not-a-knot B-splines	55
$L_{\ell,k}$	Lagrange polynomial of level ℓ and index k	50
$\mathbb{N}$	Natural numbers without zero	22
$\mathbb{N}_0$	Natural numbers including zero	23
N_{eff}	Effective sample size	79
$N(\mu, \sigma^2)$	Normal distribution with mean μ and standard deviation σ	75

Symbol	Meaning	Page with first occurrence
$O(\cdot)$	Big-O Landau notation	29
$p_{\boldsymbol{\ell},\boldsymbol{k}}$	Hierarchical piecewise polynomial C^0 element of level $\boldsymbol{\ell}$ and index $\boldsymbol{k}$	29
$\mathbb{P}^n$	Space of all polynomials of degree $m \leq n$ on $[0,1]$	50
$\mathbb{R}$	Real numbers	22
$\mathbb{R}_+$	Positive real numbers	112
$S^n_{\boldsymbol{\xi}}$	Spline space of degree n with knots $\boldsymbol{\xi}$	46
$S^{n,\text{poly}}_{\ell}$	Space of polynomial recreating splines of degree n and level ℓ	55
s	Standard deviation	69
$U(\ell,u)$	Uniform distribution on the interval $[\ell,u]$	75
$\mathbb{V}$	Variance	69
$V_{\boldsymbol{\ell}}$	Nodal space of level $\boldsymbol{\ell}$	24
$V^{\text{s,b}}_{\ell,D}$	Regular D-dimensional sparse grid space with boundary of level ℓ	33
$V^{\text{s}}_{\ell,D}$	Regular D-dimensional sparse grid space without boundary of level ℓ	34
V^{s}	Arbitrary sparse grid space (potentially spatially adaptive)	37
$W_{\boldsymbol{\ell}}$	Hierarchical subspace of level $\boldsymbol{\ell}$	25
$\boldsymbol{X}$	Random variable	68
$x_{\boldsymbol{\ell},\boldsymbol{i}}$	Grid point $x_{\boldsymbol{\ell},\boldsymbol{i}} \coloneqq \boldsymbol{i} \cdot h_{\boldsymbol{\ell}}$	24
$\mathbb{Z}$	Integer numbers	43
$\alpha_{\boldsymbol{\ell},\boldsymbol{k}}$	Hierarchical surplus for basis function of level $\boldsymbol{\ell}$ and index $\boldsymbol{k}$	26
$\varphi_{\boldsymbol{\ell},\boldsymbol{k}}$	Hierarchical basis function of level $\boldsymbol{\ell}$ and index $\boldsymbol{k}$	24
Λ^{nak}	$\Lambda^{\text{nak}}(n) \coloneqq \lceil \log_2(n) \rceil$	50
Λ	$\Lambda(n) \coloneqq \lceil \log_2(n+2) \rceil$	54
$\boldsymbol{\xi}$	General knot sequence $\boldsymbol{\xi} \coloneqq (\xi_0, \ldots, \xi_{z+n})$	43
$\boldsymbol{\xi}^{n,\text{nak}}_{\ell}$	Not-a-knot knot sequence of degree n and level ℓ	49
$\boldsymbol{\xi}^{n,\text{bnk}}_{\ell}$	Boundaryless not-a-knot knot sequence of degree n and level ℓ	55
Ω	$\Omega \subset \mathbb{R}^D$	22
$\Omega_{\boldsymbol{\ell}}$	Set of full grid points of level $\boldsymbol{\ell}$	24
$\Omega^{\text{s,b}}_{\ell,D}$	Set of regular D-dimensional sparse grid points of level ℓ with boundary	33
$\Omega^{\text{s}}_{\ell,D}$	Set of regular D-dimensional nonboundary sparse grid points of level ℓ	34
$\Omega^{\text{s,1}}_{\ell,D}$	Set of regular D-dimensional sparse grid points of level ℓ, reduced boundary	35
Ω^{s}	Arbitrary sparse grid, regular or adaptive	37
ϱ_X	Probability density function of the random variable $\boldsymbol{X}$	68
$\lvert\cdot\rvert_1$	ℓ_1 norm	22
$\lvert\cdot\rvert_2$	Euclidean norm	22
$\lvert\cdot\rvert_\infty$	Maximum norm	22

Symbol	Meaning	Page with first occurrence
$\|\cdot\|_p$	L^p norm	23
ABC	Approximate Bayesian Computation	119
ANOVA	Analysis of Variance	81
HMC	Hamiltonian Monte Carlo	80
MC	Monte Carlo	69
MCMC	Markov Chain Monte Carlo	79
MH	Metropolis-Hastings	79
NRMSE	Normalized Root Mean Square Error	60
NUTS	No-U-Turn Sampler	80
PASUL	Python Active Subspaces Utility Library	89
PCE	Polynomial Chaos Expansion	72
RMSE	Root Mean Square Error	60
UQ	Uncertainty Quantification	17

Abstract/Zusammenfassung

Abstract

Uncertainty Quantification (UQ) generalizes computer simulations by introducing stochastic terms. This allows for propagation, characterization, and inference of uncertainties, based on evaluations of the underlying simulation. However, for large-scale problems the simulation is usually computationally demanding, which makes standard UQ approaches infeasible. Reliable surrogate models are needed to approximate diverse quantities of interest from few simulation runs. In this data-scarce regime, adaptive discretization is key to the extraction of as much information as possible from few data points. Therefore, we use spatially adaptive sparse grids, which we combine with B-spline basis functions to allow for higher-order approximation, access to derivatives, and numerically exact integration.

In this thesis we propose the use of B-splines and spatially adaptive sparse grids to create accurate and versatile surrogates for UQ. First, we develop new B-spline basis functions that for the first time achieve optimal approximation order on non-boundary sparse grids. Second, we combine and customize numerous numerical techniques to enable the surrogates to tackle a wide range of UQ tasks. In particular, the surrogates provide forward propagation of uncertainties through the calculation of stochastic moments, sensitivity analysis and dimension reduction through active subspaces, and optimization and efficient sampling based on gradient information. Third, we demonstrate the combined potential of B-splines and sparse grids for UQ with four real-world applications, where our approach outperforms state-of-the-art methods. The applications include an otherwise unrealizable decision-making tool for COVID-19 tests, improved parameter calibration for an engine test-bench, and forward UQ investigations of a tsunami prediction and a subsurface CO_2 storage.

Kurzzusammenfassung

Die Quantifizierung von Unsicherheiten (UQ) ist ein Fachgebiet an der Schnittstelle von Stochastik, Informatik und Ingenieurswissenschaften, das sich mit dem Einschätzen von Risiken, sowie dem Charakterisieren und Reduzieren von Unsicherheiten beschäftigt. Indem Computersimulationen realer Prozesse um stochastische Terme ergänzt werden, tritt eine ganzheitliche Betrachtung an die Stelle individueller spezifischer Ergebnisse. Klassische UQ Methoden basieren auf Auswertungen der zugrundeliegende Simulation, was aber für umfangreiche Probleme sehr rechenaufwändig werden kann. Deshalb werden verlässliche Surrogat-Modelle benötigt um die vielfältigen Zielgrößen ausgehend von wenigen Simulationsauswertungen zu approximieren. Adaptive Diskretisierung ist der Schlüssel, um trotz dieser Datenknappheit möglichst viel Information aus wenigen Datenpunkten zu gewinnen, deshalb verwenden wir räumlich adaptive dünne Gitter. Diese kombinieren wir mit B-spline Basisfunktionen, um Approximationen höherer Ordnung, Zugang zu Ableitungen und numerisch exakte Integration zu ermöglichen. Allerdings müssen wir, um den Fluch der Dimensionalität zu vermeiden, die Randpunkte der dünnen Gitter auslassen, obwohl diese für die volle Approximationsqualität der Basis eigentlich zwingend notwendig wären.

In dieser Arbeit kombinieren wir B-splines und räumlich adaptive dünne Gitter um präzise und vielseitige Surrogate für UQ zu erstellen. Zuerst entwickeln wir neue B-spline Basisfunktionen, welche erstmalig auch auf randlosen dünnen Gittern die optimale Approximationsordnung erreichen. Dann passen wir mehrere numerische Verfahren so an, dass sie unseren Surrogaten ermöglichen ein breites Spektrum von UQ-Aufgaben zu bewältigen; insbesondere die Vorwärts-Ausbreitung von Unsicherheiten durch das Berechnen stochastischer Momente, Sensitivitätsanalyse und Dimensionsreduktion durch aktive Unterräume, sowie Optimierung und effiziente Stichprobennahme basierend auf Gradienteninformation. Schließlich demonstrieren wir das kombinierte Potential von B-splines und dünnen Gittern für UQ anhand von vier realen Anwendungen aus unterschiedlichen Fachgebieten. Diese beinhalten eine ansonsten nicht verwirklichbare Entscheidungshilfe für COVID-19 Tests, verbesserte Parameterkalibrierung für einen Motorteststand, sowie Vorwärts-UQ Untersuchungen für eine Tsunamivorhersage und einen unterirdischen CO_2 Speicher, bei denen unsere Surrogate den Stand der Technik übertreffen.

Preface

While preparing and writing this dissertation over the last four years, I was supported by many people, whom I would like to thank. First of all, thanks goes to Dirk Pflüger for his excellent supervision, encouragement, and the opportunity to conduct research on an exciting topic. While I was completely free to pursue my own research interests, he was always ready to guide me when I got stuck, or needed a little push. Second, I want to thank the co-reporter of this thesis, Stephen Roberts for his collaboration on the tsunami application, and his great hospitality during my stay with him at the Australian National University in Canberra.

Further I am grateful to Stefan Zimmer, for creative discussions and answering countless questions, no matter how foolish they were, to Fabian Franzelin, for introducing me to uncertainty quantification and multi-pitch climbing, to Julian Valentin, the "B-spline cardinal" for paving the way for B-splines on sparse grids, to my collaborators Michael Schick, David John, Timo de Wolff, Martin-Immanuel Bittner and Janin Heuer for enabling me to work on cutting edge research topics, and to all members of the Scientific Computing and the Simulation of Large Systems groups from the University of Stuttgart.

I also want to thank Stephan Renninger for countless enjoyable discussions, which not even once dealt with my actual work, and last but not least my beloved girlfriend Naomi, my parents, and my family for their never ending support.

Stuttgart, April, 2021

1

Introduction

Only 80 years ago, during the second World War, the first large-scale computer simulations were performed by John von Neumann and Stanislaw Ulam [Bor19]. Since then, computer simulations have become one of the major tools in many scientific fields, and are by now considered the third *pillar of science*, next to theory and experiments [Bun14]. With continuously increasing computational power at decreasing cost [Sch97], as well as many theoretical improvements from mathematics and computer sciences, the accuracy and significance of simulations is increasing rapidly. Still, exact simulation of real world systems can quickly turn into a Sisyphean task. Computer simulations will always be only simplified approximations of a tiny portion of reality, where few key parameters are identified, ideal assumptions are presumed and minor phenomena are ignored. Therefore, quantifying uncertainty and estimating confidence is essential for thorough investigations.

The field of Uncertainty Quantification (UQ) as we know it today, was established in the mid 1990s and has constantly grown, and gained in importance ever since [Sul15]. Uncertainty quantification expands traditional simulations with stochastic descriptions, and thus allows to take into account otherwise neglected effects that cannot be handled explicitly. Conversely, by deriving probabilistic properties of a model from observations, UQ enables parameter calibration and, more generally, solutions for inverse problems. The main challenge in UQ is that brute-force stochastic simulation techniques are infeasible for large-scale problems. Therefore, surrogates, i.e., replacement models which recreate the original up to sufficient accuracy, are required.

Complex simulations are often very expensive in terms of required hardware and runtimes, thus, it is important to keep the amount of simulation runs for surrogate creation as low as possible. In this data-scarce regime, adaptive discretization methods offer great cost-benefit ratio by iteratively choosing the most promising next evaluation point

based on the current data-set. Spatially adaptive sparse grids are such a discretization approach for the creation of surrogates for expensive and high-dimensional objectives. In contrast to full grids, i.e., uniform isotropic Cartesian products, their number of inner grid points grows multiple orders of magnitude slower, mitigating the so-called *curse of dimensionality*. The quality of sparse grid surrogates depends mainly on the underlying basis functions. Originally hat functions were used, but they were soon generalized to piecewise polynomial C^0 elements which achieve higher-order approximation. Because the C^0 elements are not differentiable, B-splines were then adapted to sparse grids, resulting in differentiable higher-order approximations.

So far, boundary grid points are necessary to achieve the full approximation potential, but because the number of boundary points of sparse grids grows exponentially in the number of dimensions, the boundary must be omitted. To compensate for this, so-called modified basis functions have been used until now, but these only work well for objectives with zero second derivatives. Universally applicable higher-order basis functions on non-boundary sparse grids have not been available until now. In this thesis we introduce the theoretical foundation for such basis functions, and then demonstrate their practical usefulness for uncertainty quantification based on four real-world applications, where we take advantage of several beneficial properties of B-spline basis functions. Their higher-order approximation quality results in accurate representations. Furthermore, splines can be integrated and differentiated numerically exact, because they are piecewise polynomials. The former is essential for the calculation of stochastic moments, and we use the latter for optimization, sensitivity analysis, and sophisticated sampling.

The outline of this thesis is as follows. We introduce sparse grids in Chap. 2, starting from univariate hierarchical basis functions. Generalizing these to multi-dimensional settings, we are able to define regular sparse grids. An overview over the number of grid points then motivates to generally omit boundary points, before we introduce spatial adaptivity which further reduces the required number of grid points.

B-spline basis functions are the core of our approaches, and we introduce them in Chap. 3. We establish the necessary theory and describe the state-of the-art, before we introduce two new types of B-spline basis functions for non-boundary sparse grids, *extended not-a-knot B-splines* and *boundaryless not-a-knot B-splines*. Despite their different motivation and definition, we prove that both bases in fact span the same space. To demonstrate the new basis function's approximation quality, we then perform a numerical comparison with established basis functions, where the new approaches always perform comparable or better than the other bases.

In Chap. 4 we introduce the required stochastic theory, and explain how UQ benefits from our surrogates. We define the polynomial chaos expansion, a popular UQ surrogate

technique, which we will later use for comparison. We then present two UQ techniques for which B-splines can be used to advantage. First, Bayesian inference for the solution of inverse problems, where we enable a modern sampling technique through our surrogate's derivatives. Second, active subspaces, a dimension-reduction technique based on sensitivity analysis, which requires integration of the surrogate's squared derivatives.

Then we demonstrate the accuracy and versatile applicability of B-splines on sparse grids for UQ based on four real-world applications. In Chap. 5 we create a surrogate for a subsurface carbon dioxide storage benchmark. By comparing our results to four other state-of-the art UQ techniques, we demonstrate how well even complex simulations and their stochastic moments can be represented using our approach.

Then, in Chap. 6, we present the results of a collaboration with Bosch Research, where we perform Bayesian inference for a real DC motor test-bench. Where so far Metropolis-Hastings and Approximate Bayesian Computation have been used for sampling the posterior, the gradients provided by the B-spline surrogate enable the more advanced Hamiltonian Monte Carlo sampling.

In Chap. 7 we perform a comprehensive forward UQ study for the simulation of tsunami run-ups. The study includes calculating the expected run-up, narrowing down the probable scenarios, and gradient-based optimization revealing the worst-case scenario.

In the last application, in Chap. 8, we perform a simulation comparing the effectiveness of different pool testing strategies with respect to COVID-19 characteristics. An interactive website allows the general public to receive recommendations customized for their individual environments. As the simulation would take hours of run-time to complete, the website utilizes a surrogate model which is almost exact with respect to stochastic fluctuations.

We end with concluding remarks and the appendix, which contains extensive proofs and additional details on the applications, that would have interrupted the flow of reading in the main text.

2

Sparse Grids

> *The curse of dimensionality [is] a malediction that has plagued the scientist from the earliest days.*
>
> — Richard E. Bellman [Bel61]

Uniform, isotropic tensor product grids, also called *full grids*, are a simple and widely used approach for the discretization of hypercubes. While full grids sample the entire domain evenly, their number of grid points grows exponentially in the number of dimensions. This prevents their practical usage for domains in more than three or four dimensions, a behavior known as the *curse of dimensionality* [Bel61].

Sparse grids are a discretization scheme designed to mitigate the curse of dimensionality. For increasing dimension, their number of grid points increases significantly slower than that of their full grid counterparts, whereas the asymptotic error decay for interpolation is only deteriorated by a logarithmic factor. The underlying methodology of sparse grids was first described by Smolyak in the 1960s [Smo63], in a form that is by now known as the *combination technique* [Gri90]. It is based on multiple approximations, each much coarser than the corresponding full grid. Still, their superposition reaches more accuracy than any of the component grids by itself. In the 1990s Zenger introduced sparse grids based on a hierarchical basis [Zen91]. Further seminal work was performed by Bungartz [Bun92; Bun98; Bun04], and Griebel [Gri90]. Since then, sparse grids have been applied in a wide range of disciplines [Peh15], among others for

interpolation [Sic11], quadrature [Ger98], partial differential equations [Wid08; Nob08], data-mining [Gar01; Pfl10], uncertainty quantification [Fra15; Fra17], density estimation [Peh14], optimization [Val16; Val19], classification [Pfl10; Pfa16], clustering [Pfa19] and economics [Bun12; Sch18].

In this chapter, we introduce sparse grids based on hierarchical basis functions. We first define univariate hierarchical subspaces in Sec. 2.1 for arbitrary basis functions, and then present the traditionally used bases in Sec. 2.2. Using tensor products, we generalize the univariate concepts to arbitrary dimensions in Sec. 2.3. An estimate on the importance of individual subspaces then motivates regular sparse grids in Sec. 2.4, which we define with and without boundary points. The former provide more accuracy than the latter, however, the number of boundary grid points grows exponentially in the number of dimensions. Thus, for increasing dimension only non-boundary grids can be applied. In Sec. 2.5, we reduce the required number of grid points for accurate approximations even further by defining spatially adaptive sparse grids. In contrast to regular sparse grids, they are automatically refined according to the quantity of interest. We mainly follow established literature [Bun04; Pfl10; Val19] for this introductory chapter and summarize the concepts, which we will need later on. The formulation and derivation of Thm. 2.2 are new contributions of this thesis.

Notation. Let $\boldsymbol{v} \in \mathbb{R}^D$ be a D-dimensional vector for $D \in \mathbb{N}$. Throughout this thesis, the term dimension D refers to the overall number of input parameters of a function. This should not be confused with other notions of dimension such as the number of spatial dimensions or the number of degrees of freedom. The ℓ_1-norm $|\boldsymbol{v}|_1$, the Euclidean norm $|\boldsymbol{v}|_2$, and the maximum norm $|\boldsymbol{v}|_\infty$ of $\boldsymbol{v}$ are defined as

$$|\boldsymbol{v}|_1 := \sum_{d=1}^{D} |v_d|, \qquad |\boldsymbol{v}|_2^2 := \sum_{d=1}^{D} v_d^2, \qquad |\boldsymbol{v}|_\infty := \max_{1 \le d \le D} |v_d|. \tag{2.1}$$

Bold numbers represent vectors with constant entries, e.g., $\mathbf{1}$ is the D-dimensional unit vector $(1, \dots, 1)^T \in \mathbb{R}^D$. Operations for D-dimensional multi-indices are performed component-wise,

$$\boldsymbol{\ell} \le \boldsymbol{k} \iff \ell_d \le k_d \ \forall\ 1 \le d \le D, \tag{2.2}$$

$$\max(\boldsymbol{\ell}, \boldsymbol{k}) = (\max(\ell_1, k_1), \dots, \max(\ell_D, k_D))^T. \tag{2.3}$$

Let (Ω, F, μ) be a measure space with the set $\Omega \subset \mathbb{R}^D$, a σ-algebra F, and a measure μ on (Ω, F). For measurable functions $f : \Omega \subset \mathbb{R}^D \to \mathbb{R}$ and $0 < p < \infty$, we define the L^p norm as

$$\|f\|_p := \left(\int_\Omega |f(\boldsymbol{x})|^p d\boldsymbol{x} \right)^{\frac{1}{p}}, \tag{2.4}$$

and $L^p(\Omega)$ as the space of all measurable functions f for which it holds $\|f\|_p < \infty$. For multi-indices $\boldsymbol{k} \in \mathbb{N}_0^D$, mixed derivatives are defined as

$$\frac{\partial^{|\boldsymbol{k}|_1}}{\partial \boldsymbol{x}^{\boldsymbol{k}}} f = \frac{\partial^{|\boldsymbol{k}|_1}}{\partial x_1^{k_1} \cdots \partial x_D^{k_D}} f. \tag{2.5}$$

We define the Sobolev semi-norm $|f|_{H^{\boldsymbol{k}}_{\text{mix}}}$ and Sobolev norm $\|f\|_{H^{\boldsymbol{m}}_{\text{mix}}}$ as

$$|f|_{H^{\boldsymbol{k}}_{\text{mix}}} := \left\| \frac{\partial^{|\boldsymbol{k}|_1}}{\partial \boldsymbol{x}^{\boldsymbol{k}}} f \right\|_2, \quad \|f\|^2_{H^{\boldsymbol{m}}_{\text{mix}}} := \sum_{\boldsymbol{0} \leq \boldsymbol{k} \leq \boldsymbol{m}} |f|^2_{H^{\boldsymbol{k}}_{\text{mix}}}, \tag{2.6}$$

where $\boldsymbol{k}, \boldsymbol{m} \in \mathbb{N}_0^D$. The *Sobolev space* with dominating mixed derivatives $H^{\boldsymbol{2}}_{\text{mix}}$ is the space of functions for which mixed partial derivatives of second order exist and are finite,

$$H^{\boldsymbol{2}}_{\text{mix}} := \{ f : \Omega \subset \mathbb{R}^D \to \mathbb{R} \mid \|f\|^2_{H^{\boldsymbol{2}}_{\text{mix}}} < \infty \}, \tag{2.7}$$

and $H^{\boldsymbol{2}}_{0,\text{mix}}$ is the space of all functions $f \in H^{\boldsymbol{2}}_{\text{mix}}$, which vanish on the boundary $\partial\Omega$. Furthermore, we define the gradient ∇f of a function $f : \mathbb{R}^D \to \mathbb{R}$ as

$$\nabla f := \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_D} \end{pmatrix}, \tag{2.8}$$

and the Jacobian $\boldsymbol{J} f$ of a function $f : \mathbb{R}^D \to \mathbb{R}^m$ as

$$\boldsymbol{J} f := \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_D} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_D} \end{pmatrix}, \tag{2.9}$$

provided that the derivatives in Equations (2.8) and (2.9) exist. Numerically, we can only handle compact sub-volumes $\Omega \subset \mathbb{R}^D$, and sparse grids can only be defined on tensor-products of finite intervals. To simplify notation and implementation, we therefore restrict ourselves to the unit interval $[0, 1]^D$ without loss of generality.

2.1 Univariate Interpolation

We now first introduce univariate nodal bases and the concept of hierarchical bases. Then, we show their equivalence with respect to interpolation.

2.1.1 Interpolation in the Nodal Basis

Discretizing the unit interval $[0,1]$ uniformly with *grid width* $h_\ell := 2^{-\ell}$ for *level* $\ell \in \mathbb{N}_0$ results in Ω_ℓ, the *full grid* of level ℓ, which consists of $2^\ell + 1$ grid points $x_{\ell,i}$,

$$\Omega_\ell := \{x_{\ell,i} \mid i \in K_\ell\}, \quad x_{\ell,i} := i \cdot h_\ell, \tag{2.10}$$

where $K_\ell := \{0, \dots, 2^\ell\}$ is the full index set of level ℓ, and we call $i \in K_\ell$ the *index*. With each grid point $x_{\ell,i}$, we associate a basis function $\varphi_{\ell,i} : [0,1] \to \mathbb{R}$, and for all $\ell \in \mathbb{N}_0$ we require the functions $\{\varphi_{\ell,i} \mid i \in K_\ell\}$ to be linearly independent. We imagine a one-to-one correspondence between basis functions and grid points, and therefore call this the *nodal basis*. The space of all linear combinations of the nodal basis functions of level ℓ is

$$V_\ell := \operatorname{span}\{\varphi_{\ell,i} \mid i \in K_\ell\}. \tag{2.11}$$

Any function $f : [0,1] \to \mathbb{R}$ in the full grid space V_ℓ can be interpolated on the full grid Ω_ℓ. This means that we can determine coefficients $c_{\ell,i} \in \mathbb{R}$ such that

$$\hat{f}_\ell := \sum_{i'=0}^{2^\ell} c_{\ell,i'} \varphi_{\ell,i'}, \quad \hat{f}_\ell(x_{\ell,i}) = f(x_{\ell,i}), \quad \forall x_{\ell,i} \in \Omega_\ell. \tag{2.12}$$

The interpolation problem (2.12) is well-defined and yields unique coefficients $c_{\ell,i'}$, because the basis functions are linearly independent. The function $\hat{f}_\ell$ is called the *full grid interpolant* of f of level ℓ. Interpolation on the full grid is illustrated in Fig. 2.1.

2.1.2 Interpolation in the Hierarchical Basis

The nodal full grid basis consists of $2^\ell + 1$ equally important basis functions. We now introduce an equivalent hierarchical basis of decreasing importance. This will later allow us to balance the total number of grid points and the resulting approximation quality. Because the full grids Ω_ℓ are nested, i.e., each grid point with an even index on level ℓ is equal to a grid point with an odd index on level $(\ell - 1)$, we can re-order the grid points

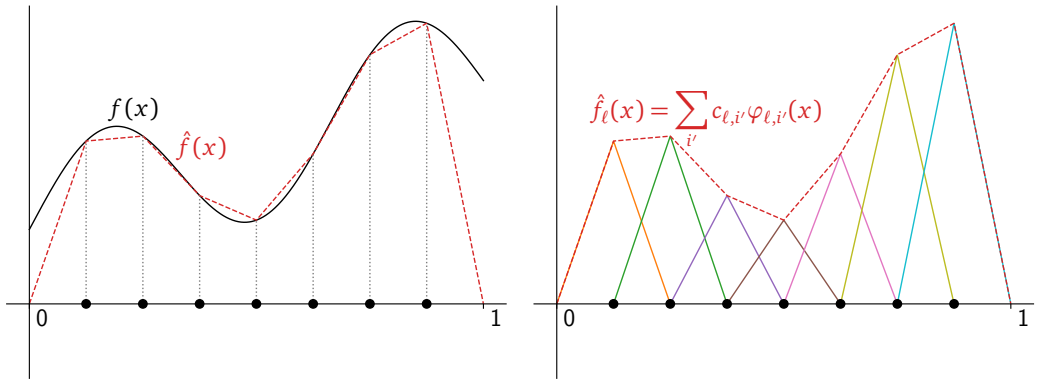


FIGURE 2.1 One-dimensional piecewise linear interpolation of the function f with nodal hat basis functions, on a non-boundary full grid, resulting in the full grid interpolant $\hat{f}_\ell$ of level 3.

hierarchically. Let I_ℓ be the one-dimensional hierarchical index set of level ℓ,

$$I_\ell := \begin{cases} \{0,1\} & \ell = 0, \\ \{i \in K_\ell \mid i \text{ odd}\} & \ell > 0. \end{cases} \tag{2.13}$$

With this we decompose the full grid Ω_ℓ into hierarchical subgrids,

$$\Omega_\ell = \bigcup_{\ell'=0}^{\ell} \{x_{\ell',i} \mid i \in I_{\ell'}\}. \tag{2.14}$$

The basis functions corresponding to each hierarchical subgrid span the corresponding hierarchical subspace W_ℓ of level ℓ,

$$W_\ell := \operatorname{span}\{\varphi_{\ell,i} \mid i \in I_\ell\}. \tag{2.15}$$

Analogously to the decomposition of the full grid Ω_ℓ into subgrids in Eq. (2.14), we decompose the full grid space V_ℓ into the direct sum of hierarchical subspaces W_ℓ. This decomposition is called *hierarchical splitting*.

LEMMA 2.1 (univariate hierarchical splitting)
If the hierarchical subspaces $W_{\ell'}$ are subspaces of V_ℓ for $\ell' \leq \ell$ and the hierarchical basis functions $\varphi_{\ell',i'}$ are linearly independent for $\ell' \leq \ell$, $i' \in I_{\ell'}$, then

$$V_\ell = \bigoplus_{\ell'=0}^{\ell} W_{\ell'}. \tag{2.16}$$

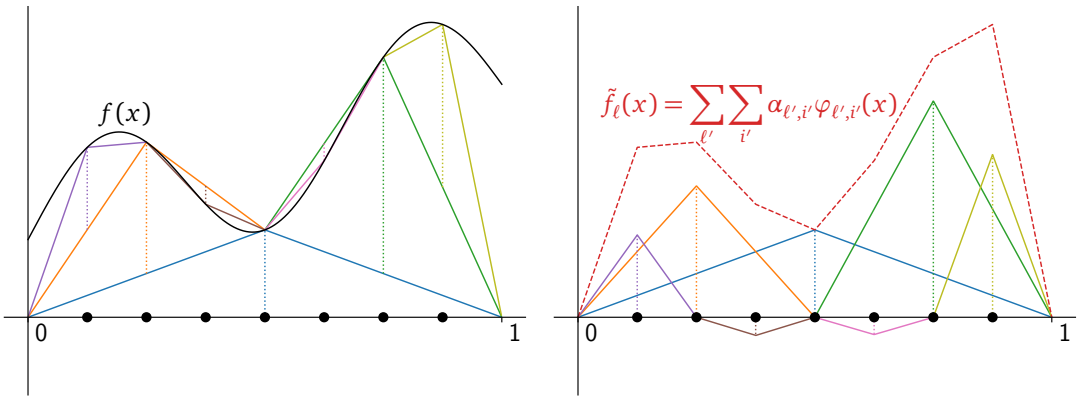


FIGURE 2.2 One-dimensional piecewise linear interpolation of the same function f as in Fig. 2.1, but with hierarchical hat basis functions on a non-boundary sparse grid of level 3. The resulting approximation $\tilde{f}_\ell$ is identical to $\hat{f}_\ell$ in V_ℓ. The left image also illustrates why the hierarchical interpolation coefficients are called "surpluses".

PROOF See [Val19]. ∎

Under the requirements of this lemma, it holds $V_\ell = V_{\ell-1} \oplus W_\ell$. Therefore, the spaces are called *nested*. Following Lm. 2.1, any function $f : [0,1] \to \mathbb{R}$ in the full grid space V_ℓ can be represented in the hierarchical basis. As in Eq. (2.12), we can calculate coefficients $\alpha_{\ell',i'} \in \mathbb{R}$, such that

$$\tilde{f}_\ell := \sum_{\ell'=0}^{\ell} \sum_{i' \in I_{\ell'}} \alpha_{\ell',i'} \varphi_{\ell',i'}, \quad \tilde{f}_\ell(x_{\ell,i}) = f(x_{\ell,i}), \quad \forall x_{\ell,i} \in \Omega_\ell. \tag{2.17}$$

It is important to note that $\tilde{f}_\ell$ is the same function as $\hat{f}_\ell$ from Eq. (2.12), but evolved in a different basis for the same space. This fact is illustrated in Figures 2.1 and 2.2. In a sparse grid context, the coefficients $\alpha_{\ell',i'}$ are called *hierarchical surpluses*. The process of calculating the surpluses from objective values at the grid points is generally called *interpolation*, but in a sparse grid context also the term *hierarchization* is used.

Hierarchical splitting via prolongation. We now derive a criterion for hierarchical splitting which will become particularly useful in Chap. 3, when we introduce new hierarchical B-spline bases. Let $\{\varphi_{\ell,k} \mid k \in K_\ell\}$ be basis functions, which are linearly independent for each $\ell \geq 0$, but for which it is not known whether the corresponding hierarchical basis $\{\varphi_{\ell',i'} \mid 0 \leq \ell' \leq \ell, i' \in I_{\ell'}\}$ is linearly independent. Let further

$$\text{span}\{\varphi_{\ell-1,k} \mid k \in K_{\ell-1}\} = V_{\ell-1} \subset V_\ell = \text{span}\{\varphi_{\ell,i} \mid i \in K_\ell\} \text{ for all } \ell \geq 1, \tag{2.18}$$

then there exists the prolongation matrix $\boldsymbol{P} \in \mathbb{R}^{(2^\ell+1)\times(2^{\ell-1}+1)}$ of level ℓ, which has entries $p_{i,k}$, such that

$$\varphi_{\ell-1,k} = \sum_{i\in K_\ell} p_{i,k}\ \varphi_{\ell,i}, \tag{2.19}$$

for all $k \in K_{\ell-1}$. Now for any given linear combination $f := \sum_{k\in K_{\ell-1}} c_k \varphi_{\ell-1,k}$, where $\boldsymbol{c} := (c_k)_{k\in K_{\ell-1}}$ is the vector of coefficients, we can calculate $\tilde{\boldsymbol{c}} := \boldsymbol{P}\boldsymbol{c}$. Using Eq. (2.19) and exploiting the finiteness of the sums yields

$$f = \sum_{k\in K_{\ell-1}} c_k \varphi_{\ell-1,k} = \sum_{k\in K_{\ell-1}} c_k \sum_{i\in K_\ell} p_{i,k}\ \varphi_{\ell,i} = \sum_{i\in K_\ell} \underbrace{\sum_{k\in K_{\ell-1}} p_{i,k} c_k}_{\tilde{c}_i}\ \varphi_{\ell,i} = \sum_{i\in K_\ell} \tilde{c}_i \varphi_{\ell,i}. \tag{2.20}$$

Let now $g := \sum_{k\in K_{\ell-1}} c_k \varphi_{\ell-1,k} + \sum_{j\in I_\ell} d_j \varphi_{\ell,j}$ be a linear combination of the nodal basis functions of level $\ell-1$ and the hierarchical basis functions of level ℓ with coefficients $\boldsymbol{c} := (c_k)_{k\in K_{\ell-1}}$ and $\boldsymbol{d} := (d_j)_{j\in I_\ell}$. We apply Eq. (2.20) and combine the resulting sums exploiting $I_\ell \subset K_\ell$,

$$\begin{aligned} g &= \sum_{k\in K_{\ell-1}} c_k \varphi_{\ell-1,k} + \sum_{j\in I_\ell} d_j \varphi_{\ell,j} = \sum_{i\in K_\ell} \tilde{c}_i \varphi_{\ell,i} + \sum_{j\in I_\ell} d_j \varphi_{\ell,j} = \sum_{i\in K_\ell} (\tilde{c}_i + \tilde{d}_i)\varphi_{\ell,i} \\ &= \sum_{i\in K_\ell} a_i \varphi_{\ell,i}, \end{aligned} \tag{2.21}$$

where

$$a_i := (\tilde{c}_i + \tilde{d}_i), \quad \tilde{d}_i := \begin{cases} d_i & \text{if } i \in I_\ell, \\ 0 & \text{else.} \end{cases} \tag{2.22}$$

Next, we split the matrix $\boldsymbol{P}$ into matrices $\boldsymbol{P}_1 \in \mathbb{R}^{(2^{\ell-1}+1)\times(2^{\ell-1}+1)}$ and $\boldsymbol{P}_2 \in \mathbb{R}^{(2^{\ell-1})\times(2^{\ell-1}+1)}$, where $\boldsymbol{P}_1$ contains the rows of $\boldsymbol{P}$ corresponding to basis functions of level ℓ, which are not part of the hierarchical basis, $\{\varphi_{\ell,k} \mid k \in K_\ell \backslash I_\ell\}$ and $\boldsymbol{P}_2$ contains the rows of $\boldsymbol{P}$ corresponding to basis functions of level ℓ, which are part of the hierarchical basis, $\{\varphi_{\ell,k} \mid k \in I_\ell\}$. With this, the identity matrix $\mathbf{1} \in \mathbb{R}^{(2^{\ell-1})\times(2^{\ell-1})}$, and the zero matrix $\mathbf{0} \in \mathbb{R}^{(2^{\ell-1}+1)\times(2^{\ell-1})}$, we define

$$\boldsymbol{A} := \begin{pmatrix} \boldsymbol{P}_1 & \mathbf{0} \\ \boldsymbol{P}_2 & \mathbf{1} \end{pmatrix} \in \mathbb{R}^{(2^\ell+1)\times(2^\ell+1)}. \tag{2.23}$$

Combining the coefficients $\boldsymbol{c}$ and $\boldsymbol{d}$ into a vector $(\boldsymbol{c}\ \boldsymbol{d})^T$ we can formulate the transformation of Eq. (2.21) as a matrix operation

$$\boldsymbol{A} \cdot \begin{pmatrix} \boldsymbol{c} \\ \boldsymbol{d} \end{pmatrix} = \begin{pmatrix} (a_i)_{i \in K_\ell \setminus I_\ell} \\ (a_i)_{i \in I_\ell} \end{pmatrix}. \tag{2.24}$$

If $\boldsymbol{P}_1$ is regular, then $\boldsymbol{A}$ is regular because we can explicitly state its inverse,

$$\boldsymbol{A}^{-1} := \begin{pmatrix} \boldsymbol{P}_1^{-1} & \boldsymbol{0} \\ -\boldsymbol{P}_2 \boldsymbol{P}_1^{-1} & \boldsymbol{1} \end{pmatrix}, \tag{2.25}$$

Consequently, any linear combination of the nodal basis of level ℓ can be represented as a linear combination of the nodal basis of level $\ell - 1$ and the hierarchical basis of level ℓ, by applying the inverse of $\boldsymbol{A}$. This is summarized in the following theorem.

THEOREM 2.2 (hierarchical splitting via prolongation)
Let $V_{\ell-1} \subset V_\ell$. If $\boldsymbol{P}_1$, defined as above, is regular, then

$$V_\ell = \bigoplus_{\ell'=0}^{\ell} W_{\ell'}. \tag{2.26}$$

In particular, the hierarchical basis functions $\{\varphi_{\ell',i'} \mid 0 \leq \ell' \leq \ell, i' \in I_{\ell'}\}$ are linearly independent.

Note, that this theorem is also valid if no boundary points are used, i.e., the indices 0 and 2^ℓ are removed from K_ℓ. The same derivation can be used, only the matrices in Eq. (2.23) must be defined accordingly smaller.

2.2 Basis Functions

Hat functions. We introduce sparse grids based on generic hierarchical basis functions $\varphi_{\ell,i}$. Traditionally however, piecewise linear hat functions are used [Bun04],

$$\varphi_{\ell,i} = b^1_{\ell,i}(x) := \max(1 - |x/h_\ell - i|,\ 0). \tag{2.27}$$

This basis fulfills the following *fundamental property*, also known as *incremental property*,

$$\begin{aligned} \varphi_{\ell,i}(x_{\ell',i'}) &= 0, \quad \ell' < \ell, i' \in I_{\ell'}, \\ \varphi_{\ell,i}(x_{\ell,i'}) &= \delta_{i,i'}, \qquad i' \in I_\ell, \end{aligned} \tag{2.28}$$

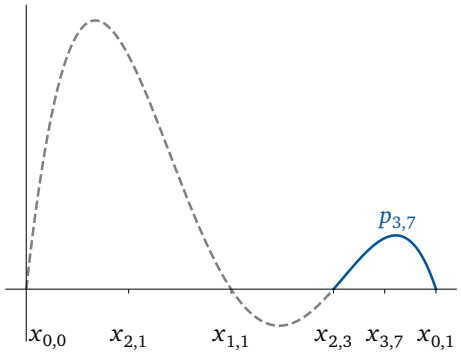
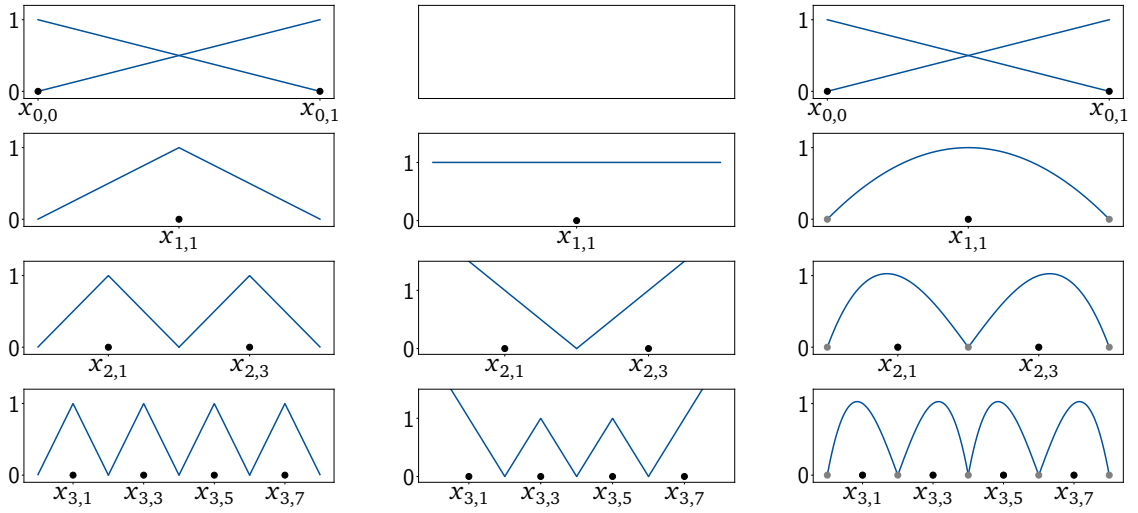


FIGURE 2.3 The hierarchical C^0 element $p_{3,7}$ of degree 4 is defined as the Lagrange polynomial in $x_{3,7}$ with respect to $\{x_{0,0}, x_{1,1}, x_{2,3}, x_{3,7}, x_{0,1}\}$ truncated to $(x_{2,3}, x_{0,1})$.

which enables efficient hierarchization in multivariate settings through the unidirectional principle. This scheme decomposes a linear operator in D dimensions into multiple applications of corresponding univariate linear operators [Bal94]. The unidirectional principle is in particular applied for fast hierarchization. The major disadvantage of the piecewise linear hat function basis (2.27) is that their approximation order, i.e., the convergence of the L^2 interpolation error, is limited to $O(h_\ell^{-2})$.

Hierarchical C^0 elements. In a first attempt to improve the approximation order, Bungartz introduced piecewise D-polynomial C^0 elements $p_{\ell,i}$ [Bun98]. These basis functions are hierarchical piecewise polynomials of arbitrary degree. They are defined as Lagrange polynomials with respect to certain grid points of the same and lower levels, but truncated to $(x_{\ell,i-1}, x_{\ell,i+1})$, see Fig. 2.3 for an illustration. Consequently, these basis functions satisfy the fundamental property Eq. (2.28), and the univariate principle can be applied. While the C^0 elements provide better approximation order than simple hat functions, they are not continuously differentiable. This is a significant drawback, as there are many relevant applications that require gradient information. For example, in this thesis we make use of gradients for active subspaces in Sec. 4.4, Bayesian inference in Chap. 6, and optimization in Chap. 7. Therefore, in recent years, B-splines have become increasingly popular with sparse grids. We introduce them comprehensively in Chap. 3.

Modified basis functions. The so far mentioned basis functions share the property that for level $\ell > 0$ they evaluate to zero on the domain's boundary. In higher dimensions, when non-boundary sparse grids must be used, this constrains the approximation quality at the boundary. Therefore, Pflüger introduced modified basis functions, which extrap-



A Hierarchical hat functions

B Modified hierarchical hat functions

C Hierarchical C^0 elements.

FIGURE 2.4 Classical hierarchical basis functions for sparse grids of level $\ell = 0, \ldots, 3$.

olate heuristically towards the boundary [Pfl10]. For linear hat functions, this means that the left-most and right-most functions of each level are linearly continued towards the boundary, resulting in the modified piecewise linear basis $b^{1,\mathrm{m}}_{\ell,i}$. See Sec. 3.2.1 for more details on this, and Fig. 2.4 for an illustration. For the hierarchical C^0 elements, the modification consists of omitting the boundary support points $x_{0,0}$ and $x_{0,1}$, and reducing the polynomial degree by two. We denote modified hierarchical C^0 elements by $p^{\mathrm{m}}_{\ell,i}$.

2.3 Multivariate Interpolation

The concepts from Sec. 2.1 and the basis functions from Sec. 2.2 are easily generalized towards arbitrary D-dimensional spaces by using tensor products. Level and index become multi-indices

$$\boldsymbol{\ell} := (\ell_d)_{d=1}^D \in \mathbb{N}_0^D, \quad \boldsymbol{i} := (i_d)_{d=1}^D \in \mathbb{N}_0^D, \tag{2.29}$$

and the grid width and grid points are defined component-wise,

$$h_{\boldsymbol{\ell}} := \left(2^{-\ell_d}\right)_{d=1}^D, \quad x_{\boldsymbol{\ell},\boldsymbol{i}} := \left(i_d h_{\ell_d}\right)_{d=1}^D. \tag{2.30}$$

Summarizing all grid points of level $\boldsymbol{\ell}$ now yields the D-dimensional full grid of level $\boldsymbol{\ell} \in \mathbb{N}_0^D$,

$$\Omega_{\boldsymbol{\ell}} := \{x_{\boldsymbol{\ell},\boldsymbol{i}} \mid \boldsymbol{i} = \mathbf{0}, \ldots, \mathbf{2}^{\boldsymbol{\ell}}\}, \tag{2.31}$$

where the notation $\boldsymbol{i} = \mathbf{0}, \ldots, \mathbf{2}^{\boldsymbol{\ell}}$ abbreviates all indices $\boldsymbol{i} = (i_d)_{d=1}^D$ with $0 \leq i_d \leq 2^{\ell_d}$ for $d = 1, \ldots, D$. The univariate basis functions are generalized through tensor products,

$$\varphi_{\boldsymbol{\ell},\boldsymbol{i}} : [0,1]^D \to \mathbb{R}, \quad \varphi_{\boldsymbol{\ell},\boldsymbol{i}}(\boldsymbol{x}) := \prod_{d=1}^{D} \varphi_{\ell_d,i_d}(x_d), \tag{2.32}$$

where $\boldsymbol{x} := (x_d)_{d=1}^D \in [0,1]^D$, and the multivariate full grid space $V_{\boldsymbol{\ell}}$ of level $\boldsymbol{\ell}$ is accordingly defined as

$$V_{\boldsymbol{\ell}} := \operatorname{span}\{\varphi_{\boldsymbol{\ell},\boldsymbol{i}} \mid \boldsymbol{i} = \mathbf{0}, \ldots, \mathbf{2}^{\boldsymbol{\ell}}\}. \tag{2.33}$$

The multivariate hierarchical index set is defined via Cartesian products, $I_{\boldsymbol{\ell}} := I_{\ell_1} \times \cdots \times I_{\ell_D}$. Generalizing Eq. (2.14) and Eq. (2.15), we decompose the full grid $\Omega_{\boldsymbol{\ell}}$ into hierarchical subgrids,

$$\Omega_{\boldsymbol{\ell}} = \bigcup_{\boldsymbol{\ell}'=\mathbf{0}}^{\boldsymbol{\ell}} \{x_{\boldsymbol{\ell}',\boldsymbol{i}} \mid \boldsymbol{i} \in I_{\boldsymbol{\ell}'}\}, \tag{2.34}$$

and define hierarchical subspaces of level $\boldsymbol{\ell}$,

$$W_{\boldsymbol{\ell}} := \operatorname{span}\{\varphi_{\boldsymbol{\ell},\boldsymbol{i}} \mid \boldsymbol{i} \in I_{\boldsymbol{\ell}}\}, \tag{2.35}$$

with which we can state the multivariate hierarchical splitting.

LEMMA 2.3 (multivariate hierarchical splitting)
Let $\boldsymbol{\ell} \in \mathbb{N}_0^D$. If and only if the hierarchical subspaces $W_{\boldsymbol{\ell}'}$ are subspaces of $V_{\boldsymbol{\ell}}$ for $\boldsymbol{\ell}' \leq \boldsymbol{\ell}$ and the hierarchical basis functions $\varphi_{\boldsymbol{\ell}',\boldsymbol{i}'}$ are linearly independent for $\boldsymbol{\ell}' \leq \boldsymbol{\ell}$, $\boldsymbol{i}' \in I_{\boldsymbol{\ell}}$ then

$$V_{\boldsymbol{\ell}} = \bigoplus_{\boldsymbol{\ell}'=0}^{\boldsymbol{\ell}} W_{\boldsymbol{\ell}'}. \tag{2.36}$$

PROOF See [Val19]. ∎

Under the assumptions of Lm. 2.3 any function $f : [0,1]^D \to \mathbb{R}$ in $V_{\boldsymbol{\ell}}$ can be represented equivalently in the full basis and in the hierarchical basis, i.e., there exist coeffi-

cients $c_{\ell,\boldsymbol{i}'} \in \mathbb{R}$ and $\alpha_{\boldsymbol{\ell}',\boldsymbol{i}'} \in \mathbb{R}$ such that

$$\hat{f}_\ell := \sum_{\boldsymbol{i}'=\boldsymbol{0}}^{\boldsymbol{2}^\ell} c_{\ell,\boldsymbol{i}'} \varphi_{\ell,\boldsymbol{i}'}, \tag{2.37}$$

$$\tilde{f}_\ell := \sum_{\boldsymbol{\ell}'=\boldsymbol{0}}^{\boldsymbol{\ell}} \sum_{\boldsymbol{i}' \in I_{\boldsymbol{\ell}'}} \alpha_{\boldsymbol{\ell}',\boldsymbol{i}'} \varphi_{\boldsymbol{\ell}',\boldsymbol{i}'}, \tag{2.38}$$

$$\hat{f}_\ell(x_{\ell,\boldsymbol{i}}) = \tilde{f}_\ell(x_{\ell,\boldsymbol{i}}) = f(x_{\ell,\boldsymbol{i}}), \quad \forall x_{\ell,\boldsymbol{i}} \in \Omega_\ell, \tag{2.39}$$

and $\hat{f}_\ell = \tilde{f}_\ell$ in V_ℓ.

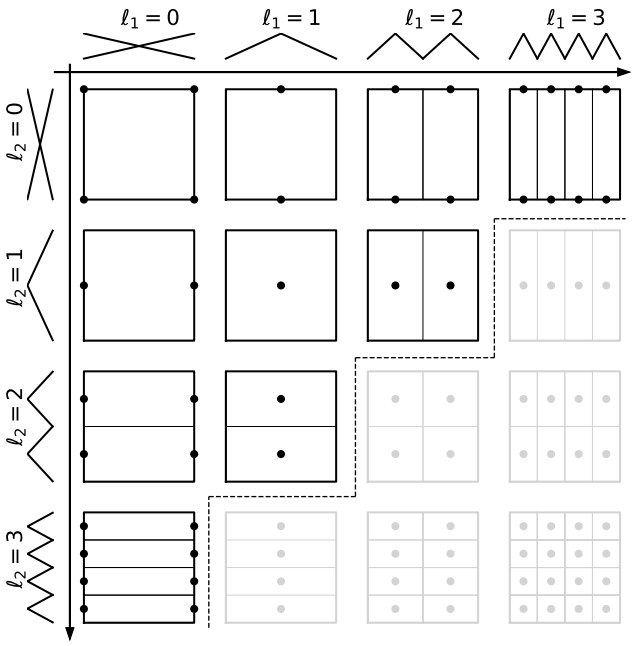
The total number of grid points of Ω_ℓ, the D-dimensional full grid of level $\boldsymbol{\ell}$, is $|\Omega_\ell| = \prod_{d=1}^{D}(2^{\ell_d}+1)$. Usually the same level $\ell \in \mathbb{N}_0$ is chosen in each dimension, resulting in a total of $(2^\ell+1)^D$ grid points. This exponential growth of the number of grid points with respect to the dimension D is the previously mentioned curse of dimensionality [Bel61]. It makes execution and storage needs unfeasible even for moderately high dimensional settings.

2.4 Regular Sparse Grids

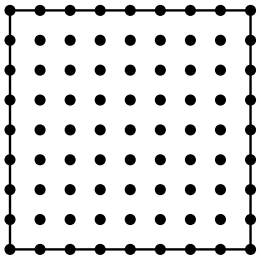
We introduced the hierarchical decomposition in Sec. 2.1.2 to allow estimates on the importance of individual basis functions. Because of the hierarchy, this is directly related to the size of the interpolation coefficients. In multi-dimensional settings, the coefficients often decrease with the level sum $|\boldsymbol{\ell}|_1$ [Smo63; Zen91]. For example, in [Bun04; Gar12] it is shown that for functions $f \in H^{\boldsymbol{2}}_{0,\text{mix}}$ each component of Eq. (2.38) can be estimated by

$$\left\| \sum_{\boldsymbol{i}' \in I_{\boldsymbol{\ell}'}} \alpha_{\boldsymbol{\ell}',\boldsymbol{i}'} \varphi_{\boldsymbol{\ell}',\boldsymbol{i}'} \right\|_2 \in O(2^{-|\boldsymbol{\ell}'|_1}). \tag{2.40}$$

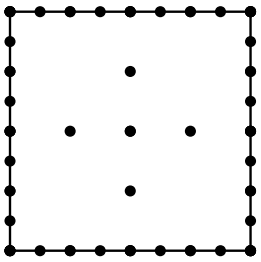
Motivated by this estimate on the importance of the individual hierarchical components, we now define sparse grids. Sparse grids omit grid points whose corresponding basis functions have small support and contribute only little to the overall function approximation according to Eq. (2.40).



A Scheme of hierarchical subspaces W_ℓ in two dimensions up to level $\ell = (3, 3)$.



B Regular full grid of level $\ell = 3$.



C Regular sparse grid of level $\ell = 3$.

FIGURE 2.5 Decomposition of the full grid space V_ℓ on the full grid Ω_ℓ into hierarchical subspaces. Illustrated with piecewise linear hat basis functions.

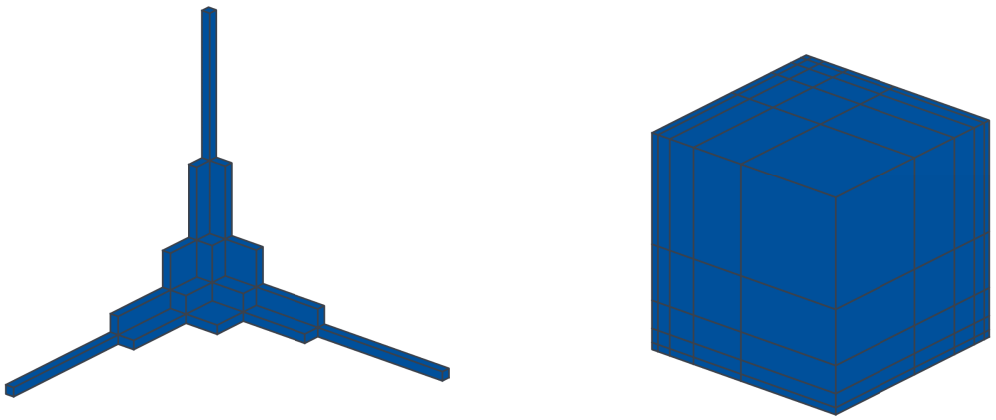
DEFINITION 2.4 (regular sparse grid with boundary)
The D-dimensional *regular sparse grid* of level $\ell \in \mathbb{N}_0$ including boundary is defined as

$$\Omega^{s,b}_{\ell,D} := \bigcup_{|\ell'|_1 \leq \ell} \{x_{\ell',\boldsymbol{i}} \mid \boldsymbol{i} \in I_{\ell'}\}, \tag{2.41}$$

where s and b do not denote exponents, but stand for *sparse* and *boundary*. The corresponding D-dimensional *regular sparse grid space* $V^{s,b}_{\ell,D}$ of level ℓ is defined as

$$V^{s,b}_{\ell,D} := \bigoplus_{|\ell'|_1 \leq \ell} W_{\ell'}. \tag{2.42}$$

Figure 2.5 illustrates this decomposition of the full grid into hierarchical subgrids and the resulting sparse grid. The definition of I_ℓ as Cartesian product in Eq. (2.13) allows a straightforward definition of sparse grids without boundary through omitting the zeroth level.



A Regular non-boundary sparse grid of level 5. **B** Full grid of level 5.

FIGURE 2.6 The number of grid points of the subgrids forming a non-boundary sparse grid $\Omega^s_{\ell,D}$ and a full grid Ω_ℓ in three-dimensional space, represented as volumes. Own representation based on an illustration by Stephen Roberts.

DEFINITION 2.5 (regular sparse grid without boundary)
The D-dimensional *regular non-boundary sparse grid* of level $\ell \in \mathbb{N}$, is defined as

$$\Omega^s_{\ell,D} := \bigcup_{1 \le \ell', |\ell'|_1 \le \ell} \{x_{\ell',i} \mid i \in I_{\ell'}\}, \tag{2.43}$$

and the corresponding D-dimensional *regular non-boundary sparse grid space* $V^s_{\ell,D}$ of level ℓ, is defined as

$$V^s_{\ell,D} := \bigoplus_{1 \le \ell', |\ell'|_1 \le \ell} W_{\ell'}. \tag{2.44}$$

For fixed dimension D, the number of grid points of a regular sparse grid of level ℓ grows like $O(h_\ell^{-1}(\log h_\ell^{-1})^{D-1})$. This is significantly less than $O(h_\ell^{-D})$, the growth rate of the corresponding full grid, as is illustrated in Fig. 2.6. Still, under some smoothness assumptions, cf. Eq. (2.40), the L^2-error of a hat-function sparse grid interpolant with homogeneous zero boundary conditions decays like $O(h_\ell^2(\log h_\ell^{-1})^{D-1})$, which deteriorates only slightly from the full grid error decay of $O(h_\ell^2)$ [Bun04]. This major statement on sparse grids shows that sparse grids indeed mitigate the curse of dimensionality. Thus, they allow for approximations that would be infeasible based on full grids. We now go into more detail on the number of grid points.

LEMMA 2.6 (number of sparse grid points)
The number of grid points of a regular D-dimensional sparse grid with boundary of level ℓ is

$$|\Omega^{\mathrm{s,b}}_{\ell,D}| = \sum_{d=0}^{D} 2^d \binom{D}{d} |\Omega^{\mathrm{s}}_{\ell,D-d}|, \tag{2.45}$$

where the number of inner grid points, which is the number of grid points of a regular D-dimensional non-boundary sparse grid of level ℓ, is

$$|\Omega^{\mathrm{s}}_{\ell,D}| = \sum_{d=0}^{\ell+1} 2^d \binom{D-1+d}{D-1}. \tag{2.46}$$

PROOF See [Bun04]. ∎

Lemma 2.6 tells us that the boundary of a regular sparse grid consists of sparse grids of lower level on each facet of the unit hypercube $[0,1]^D$. However, the number of facets of a hypercube grows exponentially in the number of dimensions. To mitigate this growth, and to reduce the number of boundary points, in practice boundary sparse grids are set up a little different from Eq. (2.41). The major problem of $\Omega^{\mathrm{s,b}}_{\ell,D}$ is that boundary points are introduced already on level zero and inner points only from level one onwards. The discretization on the boundary is thus always a level finer than the discretization along the main axis. Enforcing on the boundary the same level of discretization as on the corresponding hyperplane through the grid point on level one, $x_{\mathbf{1},\mathbf{1}}$, defines *sparse grids with aligned boundary*,

$$\Omega^{\mathrm{s,1}}_{\ell,D} := \bigcup_{\boldsymbol{\ell}' \in L^{\mathrm{s,1}}_{\ell,D}} \{x_{\boldsymbol{\ell}',\boldsymbol{i}} \mid \boldsymbol{i} \in I_{\boldsymbol{\ell}'}\}, \quad L^{\mathrm{s,1}}_{\ell,D} := \{\boldsymbol{\ell}' \in \mathbb{N}_0^D \mid |\max(\boldsymbol{\ell}', \mathbf{1})|_1 \leq \ell\}. \tag{2.47}$$

A sparse grid with aligned boundary can be thought of as having no level zero in the subspace scheme, but introducing the boundary points together with the midpoint on level one. For moderate dimensionalities the number of boundary grid points of a sparse grid with aligned boundary is still manageable and the benefit of better boundary resolution outweighs the cost of additional points. However, for increasing dimension the boundary grid points must be omitted to prevent the curse of dimensionality, cf. Tab. 2.1.

The level of boundary points and inner points can even be completely decoupled, see [Val19] for more details. However, for high dimension or expensive objective functions, the boundary will always pose a severe limitation. We therefore do not pursue this approach, but in the next chapter our main focus is to derive basis functions which compensate for omitting all boundary points.

D	$\lvert\Omega_{\boldsymbol{\ell}}\rvert$	$\lvert\Omega^{\mathrm{s,b}}_{\ell+D-1,D}\rvert$	$\lvert\Omega^{\mathrm{s,1}}_{\ell,D}\rvert$	$\lvert\Omega^{\mathrm{s}}_{\ell,D}\rvert$
1	33	33	33	31
2	1,089	385	257	129
3	35,937	3,809	1,505	351
4	$1.2 \cdot 10^6$	35,585	7,681	769
5	$3.9 \cdot 10^7$	$3.2 \cdot 10^5$	36,033	1,471
6	$1.3 \cdot 10^9$	$2.9 \cdot 10^6$	$1.6 \cdot 10^5$	2,561
7	$4.3 \cdot 10^{10}$	$2.5 \cdot 10^7$	$6.8 \cdot 10^5$	4,159
8	$1.4 \cdot 10^{12}$	$2.3 \cdot 10^8$	$2.7 \cdot 10^6$	6,401
9	$4.6 \cdot 10^{13}$	$2.0 \cdot 10^9$	$1.1 \cdot 10^7$	9,439
10	$1.5 \cdot 10^{15}$	$1.8 \cdot 10^{10}$	$4.3 \cdot 10^7$	13,441

TABLE 2.1 Number of grid points for a full grid $\Omega_{\boldsymbol{\ell}}$, a sparse grid with boundary $\Omega^{\mathrm{s,b}}_{\ell+D-1,D}$, a sparse grid with aligned boundary $\Omega^{\mathrm{s,1}}_{\ell,D}$ and a non-boundary sparse grid $\Omega^{\mathrm{s}}_{\ell,D}$ of level $\ell = 5$, respectively. The level shift for $\Omega^{\mathrm{s,b}}_{\ell+D-1,D}$ ensures the same inner grid as $\Omega^{\mathrm{s}}_{\ell,D}$ and enables a meaningful comparison. For increasing dimension D the number of boundary sparse grid points exceeds the number of inner grid points by several orders of magnitude.

2.5 Adaptive Sparse Grids

Regular sparse grids mitigate the curse of dimensionality by selecting only the most important hierarchical subspaces. However, regular sparse grids still discretize the objective domain uniformly. For heterogeneous functions too few grid points are spent in regions of interest, while too many grid points lie in regions of little significance. The only way to improve a regular sparse grid's sampling is to increase the level ℓ, which introduces many hierarchical subgrids at once. Often the objective function f, which must be evaluated at each grid point, is very expensive, and more grid points mean longer computation times.

2.5.1 Dimensional Adaptivity

One way to potentially reduce the number of sparse grid points, while keeping comparable approximation quality, is dimensional adaptivity [Ger03], which is mainly used in the context of the combination technique. Instead of level-wise adding multiple hierarchical subspaces at once, the subspaces are added individually. The next subspace is chosen by balancing the expected improvement of the approximation and the cost in terms of the number of additional grid points. Only admissible subspaces can be added, i.e., subspaces for which all predecessor subspaces are already part of the sparse grid. If the objective function is very sensitive to a subset of parameters and rather independent of others,

dimensional adaptivity can reduce the number of necessary grid points. However, each admissible subspace contains twice as many grid points as one of its predecessors and thus dimensional adaptivity still introduces many grid points in regions of little interest.

2.5.2 Spatial Adaptivity

Truly locally customized sparse grids are obtained by using spatial adaptivity, which does not add whole hierarchical subspaces, but only refines individual grid points. From the hierarchical perspective it is only natural not to think level-wise of sparse grids, but instead to view them as a tree-like structure. This tree then must not be balanced, but the depth of each branch can be customized. Thus, in contrast to regular and dimensionally adaptive sparse grids, spatially adaptive sparse grids are capable to resolve specific local features better than the remaining domain. Spatially adaptive sparse grids were first introduced by Zenger [Zen91] and further studied in particular by Bungartz [Bun04; Bun08] and Pflüger [Pfl10].

We generalize the definitions of regular sparse grids and regular sparse grid spaces using an arbitrary finite set I of level-index pairs $(\boldsymbol{\ell}, \boldsymbol{i})$, where $\boldsymbol{\ell} \in \mathbb{N}_0^D$, and $\boldsymbol{i} \in I_{\boldsymbol{\ell}}$. The sparse grid Ω^{s} and the according sparse grid space V^{s} corresponding to I are

$$\Omega^{\mathrm{s}} := \{x_{\boldsymbol{\ell},\boldsymbol{i}} \mid (\boldsymbol{\ell}, \boldsymbol{i}) \in I\}, \quad V^{\mathrm{s}} := \operatorname{span}\{\varphi_{\boldsymbol{\ell},\boldsymbol{i}} \mid (\boldsymbol{\ell}, \boldsymbol{i}) \in I\}. \tag{2.48}$$

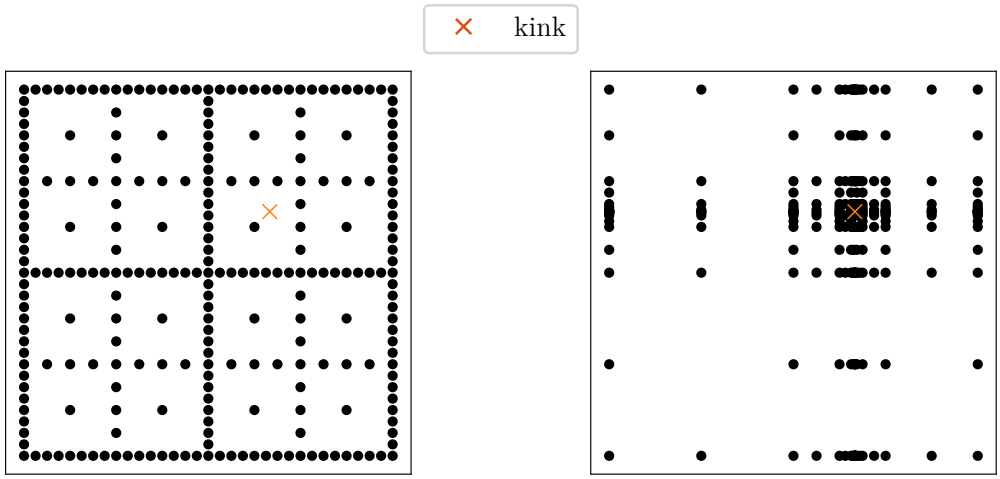
We define the *hierarchical children* $C(\boldsymbol{\ell}, \boldsymbol{i})$ of a grid point $x_{\boldsymbol{\ell},\boldsymbol{i}}$ as all points $x_{\boldsymbol{\ell}',\boldsymbol{i}'}$ for which there exists $1 \leq t \leq D$, such that

$$\begin{aligned} &\ell_d = \ell'_d,\ i_d = i'_d \quad \forall d \in \{1,\dots D\}\backslash\{t\}, \\ &\ell'_t = \ell_t + 1, \\ &i'_t \in \begin{cases} \{1\} & \ell_t = 0, \\ \{2i_t - 1,\ 2i_t + 1\} & \ell_t > 0. \end{cases} \end{aligned} \tag{2.49}$$

For a given sparse grid Ω^{s}, we define the *set of refineable grid points* $I^{\mathrm{ref}} \subset I$ as the set of level-index pairs of grid points for which not all hierarchical children are yet in Ω^{s},

$$I^{\mathrm{ref}} := \{(\boldsymbol{\ell}, \boldsymbol{i}) \in I \mid C(\boldsymbol{\ell}, \boldsymbol{i}) \not\subset \Omega^{\mathrm{s}}\}. \tag{2.50}$$

Given an objective function $f : [0,1]^D \to \mathbb{R}$, we can now create a spatially adaptive sparse grid. The first step is to initially scan the domain using a regular sparse grid Ω^{s} of small level ℓ. Next we identify one or multiple grid points $x_{\boldsymbol{\ell}^*,\boldsymbol{i}^*} \in \Omega^{\mathrm{s}}$, which have major influence on the quantity of interest. Then the grid is refined by adding for each $x_{\boldsymbol{\ell}^*,\boldsymbol{i}^*}$ all



A Regular sparse grid of level 5 with 257 grid points. L^2-approximation error: $8.9 \cdot 10^{-4}$.

B Spatially adaptive sparse grid with 250 grid points. L^2-approximation error: $6.3 \cdot 10^{-6}$.

FIGURE 2.7 Interpolating $f(x,y) := \left|x - \frac{2}{3}\right| \left|y - \frac{2}{3}\right|$ with linear basis functions on aligned boundary sparse grids. The objective function has a kink at $x = \frac{2}{3}, y = \frac{2}{3}$, which is harder to approximate. The spatially adaptive grid focuses on resolving around the kink, which results in a better approximation.

$\leq 2D$ hierarchical children $C(\boldsymbol{\ell}^*, \boldsymbol{i}^*)$ that are not yet part of the sparse grid. The number of grid points to be refined in each step must be chosen depending on the problem at hand to balance exploration and exploitation. On the one hand, a too small number can lead to a greedy behavior, refining around the same local feature over and over again. On the other hand, for a too large number the resulting sparse grid becomes close to regular.

When using boundary grids, additionally to all children $C(\boldsymbol{\ell}^*, \boldsymbol{i}^*)$, we also add all *ancestor boundary grid points* $A(\boldsymbol{\ell}^*, \boldsymbol{i}^*)$ to keep the grid consistent. These are

$$\begin{aligned} A(\boldsymbol{\ell}^*, \boldsymbol{i}^*) := \{ x_{\boldsymbol{\ell},\boldsymbol{i}} \mid \boldsymbol{\ell}, \boldsymbol{i} \in \mathbb{N}_0^D, \\ \exists t \in \{1, \dots, D\} \text{ s.t. } \ell_t = 0, \\ \exists n \in \mathbb{N} \text{ s.t. } (\boldsymbol{\ell}^*, \boldsymbol{i}^*) \in C^n(\boldsymbol{\ell}, \boldsymbol{i}) \}, \end{aligned} \tag{2.51}$$

with $C^1(\boldsymbol{\ell}, \boldsymbol{i}) := C(\boldsymbol{\ell}, \boldsymbol{i})$, and $C^n(\boldsymbol{\ell}, \boldsymbol{i}) := \bigcup_{x_{\boldsymbol{\ell}',\boldsymbol{i}'} \in C^{n-1}(\boldsymbol{\ell},\boldsymbol{i})} C(\boldsymbol{\ell}', \boldsymbol{i}')$ for $n > 1$ containing the children of the n-th generation. The refinement procedure is iterated until a threshold for the maximum number of grid points is exceeded, see Alg. 2.1 for an algorithmic description and Fig. 2.7 for an illustration.

Let Ω^s be a sparse grid with respect to any feasible level-index set I. Note that this includes any regular or adaptive sparse grid. We denote the according sparse grid

```
   f, objective function
   M, maximum number of grid points
   T, number of grid points to be refined in each step
   ℓ_0, initial level
 1 function Ω^s, α = SpatialRefinement(f, M, T, ℓ_0)
 2   Ω^s ← regular sparse grid of level ℓ_0
 3   while |Ω^s| < M do
 4     α ← interpolation coefficients for f w.r.t. Ω^s
 5     I^ref ← {(ℓ, i) ∈ I | C(ℓ, i) ⊄ Ω^s}
 6     for 1 ≤ t < T do
 7       (ℓ*, i*) ← argmax_{(ℓ,i)∈I^ref} |α_{ℓ,i}|
 8       Ω^s ← Ω^s ∪ C(ℓ*, i*)
 9       I^ref ← I^ref \ {(ℓ*, i*)}
10       if boundary grid then
11         Ω^s ← Ω^s ∪ A(ℓ*, i*)
12   α ← interpolation coefficients for f w.r.t. Ω^s
13   return Ω^s, α
```

ALGORITHM 2.1 Creation of a sparse grid with surplus adaptive spatial refinement.

representation of an objective function $f : [0,1]^D \to \mathbb{R}$ as $\tilde{f}$, and define it as

$$\tilde{f} := \sum_{(\boldsymbol{\ell}',\boldsymbol{i}')\in I} \alpha_{\boldsymbol{\ell}',\boldsymbol{i}'}\varphi_{\boldsymbol{\ell}',\boldsymbol{i}'}, \quad \text{such that} \quad \tilde{f}(x_{\boldsymbol{\ell},\boldsymbol{i}}) = f(x_{\boldsymbol{\ell},\boldsymbol{i}}), \quad \forall(\boldsymbol{\ell},\boldsymbol{i}) \in I. \tag{2.52}$$

Surplus refinement. The most popular criterion for the identification of $x_{\boldsymbol{\ell}^*,\boldsymbol{i}^*}$, the grid point to be refined, is *surplus-based refinement* [Bun04; Pfl10]. Because of the hierarchical basis, larger surpluses $|\alpha_{\boldsymbol{\ell},\boldsymbol{i}}|$ imply a worse local approximation. Consequently, $x_{\boldsymbol{\ell}^*,\boldsymbol{i}^*}$ is chosen as

$$(\boldsymbol{\ell}^*,\boldsymbol{i}^*) = \operatorname*{arg\,max}_{(\boldsymbol{\ell},\boldsymbol{i})\in I^{\text{ref}}} |\alpha_{\boldsymbol{\ell},\boldsymbol{i}}|. \tag{2.53}$$

Surplus refinement for multiple outputs. In our applications, objective functions will often have multiple output quantities, i.e.,

$$\begin{aligned} f : \mathbb{R}^D &\to \mathbb{R}^m \\ \boldsymbol{x} &\mapsto f(\boldsymbol{x}) = (f^1(\boldsymbol{x}), \ldots, f^m(\boldsymbol{x}))^T. \end{aligned} \tag{2.54}$$

For each output component $f^k(\boldsymbol{x})$ we need to create an individual sparse grid representation, i.e., calculate coefficients $\alpha^k_{\boldsymbol{\ell}',\boldsymbol{i}'}$. However, we use the same underlying sparse grid

for all output components and refine it with respect to the overall maximal surplus,

$$(\boldsymbol{\ell}^*, \boldsymbol{i}^*) = \underset{(\boldsymbol{\ell},\boldsymbol{i})\in I^{\text{ref}},\ k=1,\ldots,m}{\arg\max} |\alpha^k_{\boldsymbol{\ell},\boldsymbol{i}}|. \tag{2.55}$$

Using the same underlying sparse grid for all outputs allows us to construct the QR decomposition of the interpolation matrix only once, and then apply it consecutively to all outputs. Thus, creating surrogates for arbitrarily many outputs is only linearly more expensive than for a single output. The above criterion can easily be generalized for objective functions with outputs of different magnitudes by adding weight terms.

We also experiment with other refinement criteria than surplus refinement, which is discussed in Chap. 5.

SG^{++}. All the sparse grid concepts introduced in this chapter, as well as the new B-spline basis functions from the next chapter, and the new UQ approaches are available within the sparse grid framework SG^{++}, which is open-source and freely available [SGpp; Pfl10]. All numerical results and comparisons in this thesis were obtained through the author's own implementations within this framework.

3

B-Splines

> *A technique used [. . .] during World War Two to construct templates for airplanes by passing thin wooden planks ('splines') through points laid out on the floor.*
>
> — Robin Forrest [Bar95]

In the last section, we presented hierarchical polynomial C^0 elements, which were designed to improve the approximation quality of simple hat basis functions on sparse grids. However, the approximation power of C^0 elements is still limited because of their non-differentiability. Global polynomials would be a differentiable higher-order approximation alternative, but to improve the accuracy of the approximation and to increase the number of degrees of freedom, the polynomial degree must be increased. This can lead to Runge's phenomenon [Run01], which introduces oscillations. Interpolation in Chebyshev points, i.e., using Chebyshev polynomials, avoids this and in fact minimizes the effect of Runge's phenomenon. However, Chebyshev polynomials are still globally defined, which makes adapting to local features of the objective function difficult. Furthermore, Faber showed that for every given sequence of sets of interpolation nodes, there exists a continuous function for which the interpolation process on those nodal sets diverges [Fab14]. Consequently, there is no universal global scheme of interpolation that suits all problems.

Therefore, we now turn our attention to *splines*, which combine the piecewise design of C^0 elements and the higher-order approximation and differentiability of global

polynomials by dividing the parameter space into smaller intervals and defining local piecewise polynomials on each segment. This approach enables local adaptivity and avoids Runge's phenomenon. The transition at the intersection of intervals is smooth enough to guarantee higher-order approximation quality.

Splines were first studied by Schoenberg in 1946 [Sch46]. Because of temporary business secrecy, his work was independently continued by de Boor [Bir65; DeB78], Bézier [Béz66], and de Casteljau [DeC63] around 1960. This coincides with the development of the Finite Element Method (FEM), which was advantageously applied in combination with B-splines [Höl03]. The combination of the two techniques remains fruitful to this day, and has been further improved through non-uniform rational B-splines (NURBS) [Pie12] and isogeometric analysis (IGA) [Hug05]. B-splines have by now become a standard tool in numerical mathematics [Qua10] and are used in numerous fields. In recent years, B-splines were applied to sparse grids for high-dimensional approximation and optimization [Pfl10; Sic11; Val18; Val19], and in this chapter we continue this work. In particular, we lay the theoretical foundation for the usage of B-splines with non-boundary sparse grids, which are crucial in higher dimensionalities.

Section 3.1 recapitulates the essential B-spline definitions and theorems. It concludes with the definition of hierarchical B-splines for sparse grids in Sec. 3.1.1 and boundary treatment in Sec. 3.1.2. Section 3.2 then focuses on B-splines on non-boundary sparse grids. This section is a major contribution to the theory of B-splines on sparse grids. We recapitulate the status quo, namely modified B-splines in Sec. 3.2.1, which can lead to a significant decline of the approximation order. We then introduce two new B-spline bases for the usage on sparse grids, neither of which suffers from this decline, hierarchical extended B-splines in Sec. 3.2.2 and hierarchical boundaryless not-a-knot B-splines in Sec. 3.2.3. The chapter closes with a comparison of the approximation quality of the introduced basis functions in Sec. 3.3.

This chapter and the Appendix A.1 contain the major theoretical contributions of this thesis: the new extended not-a-knot B-spline basis, and the new boundaryless not-a-knot B-spline basis, as well as the proof for hierarchical splitting. These bases have already been presented in publications [Reh18; Reh21b; Reh21c], but the formal proof of hierarchical splitting, as well as the thorough comparison with established bases, are shown for the first time in this thesis. The boundaryless not-a-knot B-splines are based on a discussion with Dr. Stefan Zimmer (University of Stuttgart, Germany).

3.1 B-Splines

Splines are defined through knot sequences, which divide the area of interest into sections, on each of which the spline is a polynomial of a specified degree. A major advantage of splines is that degrees of freedom can be introduced by adding more knots to the sequence, while the polynomial degree stays constant. This makes them more flexible and preferable over global polynomials. Many different types of splines exist, but one has by now become established as the basis for the space of splines. Its advantages are, among others, a simple and elegant theory based on recursiveness, optimal approximation order, and fast numerical algorithms. We introduce these basis splines, or *B-splines*, in the same manner as Höllig [Höl13].

Let $\boldsymbol{\xi} := (\xi_0, \dots, \xi_{z+n})$ be a *knot sequence*, that is, a non-decreasing sequence of real numbers ξ_k for $k \in \{0, \dots, z+n\}$ and $z, n \in \mathbb{N}$. This notation is meant to imply that z splines of degree n will be defined through the knot sequence, which determines a partitioning of a subset $[\xi_0, \xi_{z+n}) \subset \mathbb{R}$ into knot intervals $[\xi_k, \xi_{k+1})$, $k = 0, \dots, z+n-1$. The knots need not be unique and the maximal number of repetitions of ξ_k in the sequence $\boldsymbol{\xi}$ is called its *multiplicity*.

DEFINITION 3.1 (B-spline)
For a knot sequence $\boldsymbol{\xi}$, the B-spline $b^n_{k,\boldsymbol{\xi}}$ of index k and degree n is defined by the Cox-de-Boor recursion [Cox72; De 72],

$$b^n_{k,\boldsymbol{\xi}}(x) := \begin{cases} \dfrac{x-\xi_k}{\xi_{k+n}-\xi_k} b^{n-1}_{k,\boldsymbol{\xi}}(x) + \dfrac{\xi_{k+n+1}-x}{\xi_{k+n+1}-\xi_{k+1}} b^{n-1}_{k+1,\boldsymbol{\xi}}(x) & n \geq 1, \\ \chi_{[\xi_k,\xi_{k+1})}(x) & n = 0, \end{cases} \tag{3.1}$$

where terms with zero denominator are discarded, and the characteristic function $\chi_{[\xi_k,\xi_{k+1})}$ evaluates to one in the interval $[\xi_k, \xi_{k+1})$ and zero elsewhere.

Each B-spline $b^n_{k,\boldsymbol{\xi}}$ is uniquely determined by its knots and degree. Its support is $[\xi_k, \xi_{k+n+1}]$, and on each nonempty knot-interval $[\xi_k, \xi_{k+1})$, the B-spline is a non-negative polynomial of degree $\leq n$. These properties can be shown by induction [Höl13]. When iterating over all splines of a knot sequence, we write $k \sim \boldsymbol{\xi}$, which can be read as $k = 0, \dots, z-1$. The simplest and most common knot sequence is the uniform knot sequence $\boldsymbol{\xi}^{\mathrm{u}}$ which has knots $\xi^{\mathrm{u}}_k := kh$ for index $k \in \mathbb{Z}$ and segment length $h \in \mathbb{R}$.

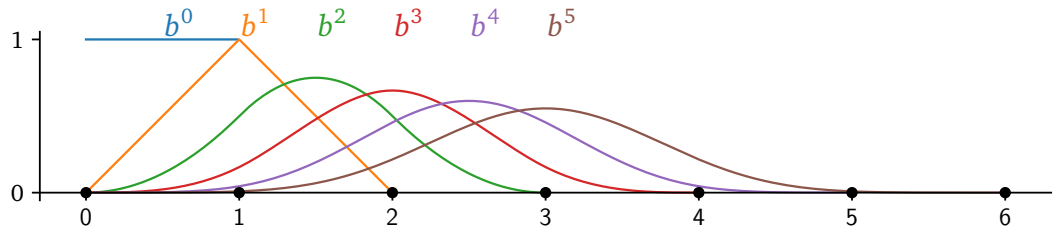


FIGURE 3.1 The standard uniform B-splines $b^0, \ldots, b^5$.

DEFINITION 3.2 (cardinal splines)
The *standard uniform B-spline* b^n is defined on the knots $0, 1, \ldots, n+1$, i.e., $b^n := b^n_{0,\xi^u}$ with grid width $h = 1$. The uniform B-splines for an arbitrary uniform knot sequence ξ^u with grid width h are scaled translates of b^n,

$$b^n_{k,\xi^u} = b^n((x - \xi^u_k)/h), \quad k \sim \xi^u. \tag{3.2}$$

Their linear combinations $\sum_{k\sim\xi^u} c_k b^n_{k,\xi^u}$ are called *cardinal splines* of degree n and grid width h. The standard uniform B-splines of degrees 0 to 5 can be seen in Fig. 3.1.

The recursive definition of B-splines (3.1) directly implies a recursive formula for their derivatives as the weighted difference of two B-splines of lower degree. Consequently, the derivative of a B-spline of degree n with knots $\xi_k, \ldots, \xi_{k+n+1}$ is given by

$$\frac{d}{dx} b^n_{k,\xi}(x) = \frac{n}{\xi_{k+n} - \xi_k} b^{n-1}_{k,\xi}(x) - \frac{n}{\xi_{k+n+1} - \xi_{k+1}} b^{n-1}_{k+1,\xi}(x), \tag{3.3}$$

where again terms with zero denominator are discarded. It follows from this formula that $b^n_{k,\xi}$ is $(n-\mu)$-times continuously differentiable at a knot ξ_i of multiplicity $\mu \le n$. In particular, B-splines are continuous if all knots have multiplicity $\le n$. This property is further formalized through the definition of n-regularity, which is illustrated in Fig. 3.2.

DEFINITION 3.3 (knot sequence regularity)
The parameter interval $D^n_\xi := [\xi_n, \xi_z]$ is the maximal interval on which the B-splines $b^n_{k,\xi}, k \sim \xi$ form a partition of unity. A knot sequence ξ is n-regular, if each B-spline $b^n_{k,\xi}, k \sim \xi$, is continuous and nonzero at some points in D^n_ξ.
In particular, for a knot sequence ξ to be n-regular all knot multiplicities must be $\le n$, $\xi_n < \xi_{n+1}$, and $\xi_{z-1} < \xi_z$.

To better resolve details or singularities, a given B-spline representation can be refined by inserting an additional knot. Böhm's algorithm gives the coefficients for the resulting spline in a consistent representation.

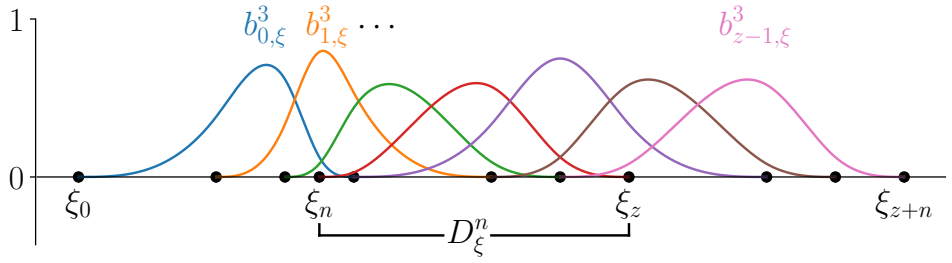


FIGURE 3.2 All relevant (i.e., non-zero) cubic B-splines on the knot sequence $\xi = (\xi_0, \ldots, \xi_{z+n})$, and the parameter interval D^n_ξ.

THEOREM 3.4 (Böhm's algorithm)
Let $\xi = (\xi_0, \ldots, \xi_{z+n})$ be a given knot sequence and $\zeta = (\zeta_0, \ldots, \zeta_{z+n+1})$ be the knot sequence obtained by inserting a knot η in ξ into the interval $[\xi_j, \xi_{j+1})$. If

$$f = \sum_{k=0}^{z-1} c_k b^n_{k,\xi} = \sum_{i=0}^{z} \tilde{c}_i b^n_{i,\zeta}, \tag{3.4}$$

then the new coefficients $(\tilde{c}_i)_{i=0}^z$ can be expressed in terms of the coefficients $(c_k)_{k=0}^{z-1}$,

$$\tilde{c}_i := \begin{cases} c_i & \text{if } 0 \le i \le j-n, \\ \frac{\eta-\xi_i}{\xi_{i+n}-\xi_i} c_i + \frac{\xi_{i+n}-\eta}{\xi_{i+n}-\xi_i} c_{i-1} & \text{if } j-n+1 \le i \le j, \\ c_{i-1} & \text{if } j+1 \le i \le z. \end{cases} \tag{3.5}$$

PROOF See [Boe80]. ∎

Multiple knots can be inserted by iteratively applying Böhm's algorithm, which is illustrated in Fig. 3.3.

One of the major theorems in spline theory is the Marsden identity. It states that B-splines can represent all polynomials of degree $\le n$, and thus are indeed a generalization of global polynomials.

THEOREM 3.5 (Marsden identity)
Let ξ be an n-regular knot sequence. Any polynomial of degree $\le n$, restricted to $x \in D^n_\xi$, can be represented as a linear combination of B-splines. In particular,

$$(x-y)^n = \sum_{k=0}^{z-1} \psi^n_{k,\xi}(y) b^n_{k,\xi}(x), \tag{3.6}$$

where $\psi^n_{k,\xi}(y) := (\xi_{k+1} - y) \cdots (\xi_{k+n} - y)$.

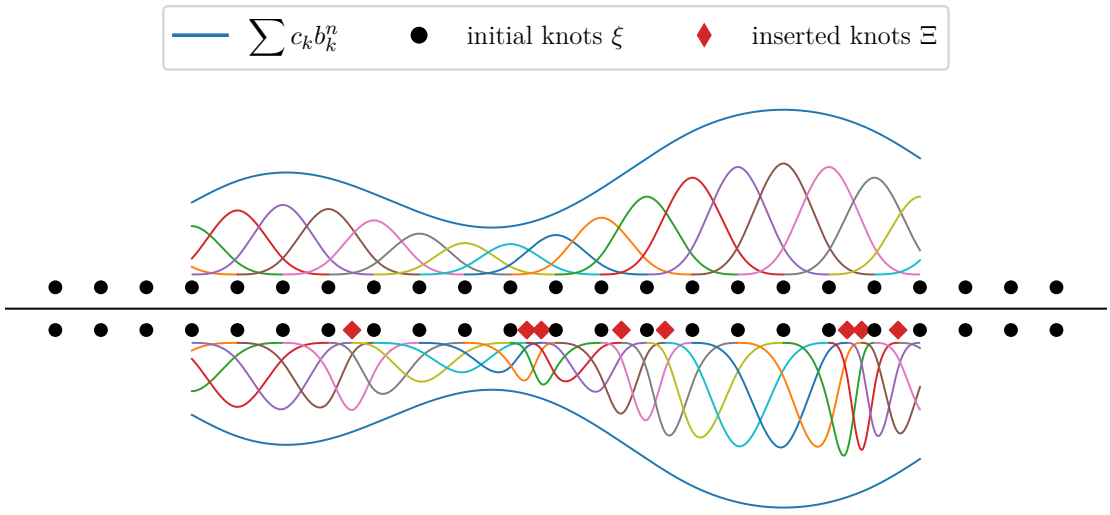


FIGURE 3.3 Illustration of Böhm's algorithm (Thm. 3.4). On top is a function approximated on D_ξ^n by cubic B-splines on a knot sequence ξ. Mirrored at the center line is the same function approximated by cubic B-splines on the knot sequence $\zeta = \xi \cup \Xi$, where Ξ are additional knots inserted using Böhm's algorithm.

PROOF See [Höl13]. ∎

We now define the space of splines.

DEFINITION 3.6 (spline space)

Let $n \in \mathbb{N}$, and ξ be an n-regular knot sequence. We use the term S_ξ^n for the space of splines of degree $\leq n$. It consists of all functions on the parameter interval D_ξ^n that are

- polynomials of degree $\leq n$ on the nondegenerate knot intervals of D_ξ^n, and
- $(n-\mu)$ times continuously differentiable at an interior knot of D_ξ^n with multiplicity μ.

The name-giving basis property of B-splines is confirmed by the following theorem.

THEOREM 3.7 (basis splines)

The B-splines $b_{k,\xi}^n$, $k \sim \xi$, restricted to D_ξ^n form a basis for the spline space S_ξ^n. Consequently, any spline $s \in S_\xi^n$ can be written as a linear combination of the B-splines corresponding to ξ,

$$s(x) = \sum_{k \sim \xi} c_k b_{k,\xi}^n(x), \quad x \in D_\xi^n, \tag{3.7}$$

with uniquely determined coefficients $c_k \in \mathbb{R}$.

PROOF See [Höl13]. ∎

For the remainder of the thesis we construct spline approximations of functions

f through *interpolation*. This requires interpolation points $(x_i)_{i=0}^{z-1} \in D_\xi^n$. Then the interpolation matrix $\boldsymbol{A} = (a_{i,k})_{i,k=0}^{z-1}$ can be constructed, where each entry is defined as $a_{i,k} := b_{k,\xi}^n(x_i)$. If the interpolation points are chosen such that $\boldsymbol{A}$ is regular, then the *interpolant* $\tilde{f}$ evaluated at $x \in D_\xi^n$ is given by

$$\tilde{f}(x) = \sum_{k=0}^{z-1} c_k b_{k,\xi}^n(x), \tag{3.8}$$

where $(c_k)_{k=0}^{z-1} := \boldsymbol{A}^{-1}\boldsymbol{b}$, and $\boldsymbol{b} := (f(x_i))_{i=0}^{z-1}$. By construction, it holds $f(x_i) = \tilde{f}(x_i)$ for all interpolation points $(x_i)_{i=0}^{z-1}$.

Unlike for global polynomials, the local support of the B-spline basis implies that there must be a restriction on the placement of the interpolation points for well-posedness of spline interpolation problems. This restriction is characterized by the Schoenberg-Whitney conditions.

THEOREM 3.8 (Schoenberg-Whitney conditions)
For a spline space S_ξ^n with finite knot sequence $\xi = (\xi_0, \dots, \xi_{z+n})$, and a point sequence $x_0 < \cdots < x_{z-1}$ of interpolation points in the parameter domain D_ξ^n, there exists a unique interpolating spline $s = \sum_{k=0}^{z-1} c_k b_{k,\xi}^n \in S_\xi^n$ for arbitrary values $y_0 = f(x_0), \dots, y_{z-1} = f(x_{z-1})$, if and only if

$$\xi_k < x_k < \xi_{k+n+1}, \tag{3.9}$$

for $k = 0, \dots, z-1$.

PROOF See [Sch53; Höl13]. ∎

The error of spline approximations converges with optimal order with respect to the degree n. This is formalized in the following theorem, which uses the notation C^{n+1} for the space of $(n+1)$-times continuously differentiable functions.

THEOREM 3.9 (error of full grid spline interpolation)
Let the knot sequence $\xi = (\xi_0, \dots, \xi_{n+z})$ and interpolation points $(x_i)_{i=0}^{z-1}$ fulfill the Schoenberg-Whitney conditions. The error of a spline interpolant $s = \sum_{k\sim\xi} c_k b_{k,\xi}^n \in S_\xi^n$ with respect to a smooth function $f \in C^{n+1}$ can be estimated by

$$|f(x) - s(x)| \le c(n, \|\boldsymbol{A}^{-1}\|_\infty)\|f^{(n+1)}\|_\infty h^{n+1}, \quad x \in D_\xi^n, \tag{3.10}$$

where the constant c depends on the degree and the maximum norm of the inverse of the interpolation matrix $\boldsymbol{A}$, h is the maximum length of knot intervals, and it is assumed that the parameter interval $D_\xi^n = [\xi_n, \xi_z]$ contains all interpolation points as well as for each

B-spline a largest interval in its support.

PROOF See [Höl13]. ∎

Multivariate B-splines are defined via tensor products.

DEFINITION 3.10 (multivariate B-splines)
The D-variate B-spline of degree $(n_1, \dots, n_D)$ with respect to the knot sequences $\xi = (\xi_1, \dots, \xi_D)$ is

$$b_{\boldsymbol{k},\xi}^{(n_1,\dots,n_D)}(\boldsymbol{x}) = \prod_{d=1}^{D} b_{k_d,\xi_d}^{n_d}(x_d). \tag{3.11}$$

Its knots in the d-th coordinate direction are $\xi_{d,k_d}, \dots, \xi_{d,k_d+n_d+1}$.

Usually, we use the same degree n in all coordinate directions, and simply write n instead of $(n, \dots, n)$. The above tensor product structure allows us to focus on the univariate case for the rest of this chapter, the multivariate counterparts then follow automatically [Val19].

3.1.1 Hierarchical B-Splines

For the usage on sparse grids, we need to define hierarchical B-splines. This can be done in a straightforward way by using a knot sequence that matches the sparse grid points. Let $\xi_\ell^{n,\mathrm{u}} := (\xi_{\ell,0}^{n,\mathrm{u}}, \dots, \xi_{\ell,2^\ell+2n}^{n,\mathrm{u}})$ be the uniform knot sequence with grid width $h_\ell = 2^{-\ell}$, where $\xi_{\ell,k}^{n,\mathrm{u}} := (k-n)h_\ell$. This results in *hierarchical uniform B-splines* $b_{\ell,k}^n$ of degree n, level ℓ and index k,

$$b_{\ell,k}^n := b_{k+\frac{n-1}{2},\xi_\ell^{n,\mathrm{u}}}^n, \quad k = 0, \dots, 2^\ell. \tag{3.12}$$

Hierarchical subspaces spanned by B-splines are defined analogously to Eq. (2.15). They are combined into sparse grids following Equations (2.42) and (2.44). From here on, we will introduce B-spline variations and knot sequences in a hierarchical manner, because in this thesis we will apply them on sparse grids, and not in the nodal context.

For even degrees Eq. (3.12) is not applicable, because the index $k + \frac{n-1}{2}$ would have to be interpreted as the midpoint between two consecutive grid points. This runs contrary to the concept of hierarchical splitting from Lm. 2.1, which is the theoretical foundation of sparse grids. Therefore, as is common, from now on we will only use odd degrees [Val19]. This still includes the basic linear case $n = 1$ as well as the most popular cubic case $n = 3$.

Note on hierarchical B-splines. The term 'hierarchical B-splines' is unfortunately also used for B-splines on grids of varying grid width which are generated by iteratively halving

the grid width for subsections of the domain, cf. [Höl13]. This kind of hierarchical splines is not used in this thesis, and by "hierarchical" we always refer to basis functions according to Equations (2.13) and (2.15)

3.1.2 Not-a-knot B-Splines

Usually, B-splines are introduced for bi-infinite knot sequences on the whole real axis. In the previous section, however, we have introduced B-splines for finite knot sequences ξ. This was reasonable considering that we ultimately want to work with B-splines primarily on sparse grids which are defined on finite hypercubes. However, the finite uniform knot sequences do not fulfill the Schoenberg-Whitney conditions of Thm. 3.8. Generally speaking, the problem is that the dimension of the spline space is z, but, when interpolating in the knots, there are only $z-(n-1)$ interpolation points inside $D^n_{\xi^{\mathrm{u}}}$. Consequently, $(n-1)$ additional conditions are required. The Schoenberg-Whitney conditions require in particular that all interpolation points are inside D^n_ξ. However, for the so far introduced hierarchical B-splines it holds $D^n_{\xi^{n,\mathrm{u}}_\ell} \subsetneq [0,1]$, as the first $\frac{n-1}{2}$ and last $\frac{n-1}{2}$ sparse grid points are outside $D^n_{\xi^{n,\mathrm{u}}_\ell}$.

The most common approaches to resolve such boundary problems and to validate the Schoenberg-Whitney conditions are boundary knots of higher multiplicity at 0 and 1 [Höl13], specifying derivative values at the boundaries [DeB78], or the not-a-knot condition [De 66; Höl13; Val19]. The first two options require the boundary points, which is unfortunate for the usage on sparse grids, as we discussed following Lm. 2.6. Therefore, we focus on the not-a-knot condition which requires continuity of the n-th derivative at the $\frac{n-1}{2}$ left-most and $\frac{n-1}{2}$ right-most knots inside $D^n_{\xi^{\mathrm{u}}}$, excluding the boundary points.

Requiring continuity at these points means that a spline s must be a single polynomial of degree n on $[0, x_{\ell,(n+1)/2}]$ and $[x_{\ell,2^\ell-(n+1)/2}, 1]$. It leaves no degrees of freedom at the $(n-1)$ knots which are affected by the condition. The condition is thus equivalent to excluding these knots completely from the knot sequence ξ, but keeping them in the set of interpolation nodes. Consequently, the hierarchical uniform not-a-knot sequence of level ℓ and degree n, $\xi^{n,\mathrm{nak}}_\ell := (\xi^{n,\mathrm{nak}}_{\ell,0}, \dots, \xi^{n,\mathrm{nak}}_{\ell,2^\ell+n+1})$, is defined as

$$\xi^{n,\mathrm{nak}}_{\ell,k} := \begin{cases} \xi^{n,\mathrm{u}}_{\ell,k} & k = 0,\dots,n, \\ \xi^{n,\mathrm{u}}_{\ell,k+(n-1)/2} & k = n+1,\dots,2^\ell, \\ \xi^{n,\mathrm{u}}_{\ell,k+n-1} & k = 2^\ell+1,\dots,2^\ell+n+1. \end{cases} \tag{3.13}$$

This definition is only applicable if there are at least $(n-1)$ inner knots that can be removed

from the uniform knot sequence, which is the case for level $\ell \geq \Lambda^{\text{nak}}(n) := \lceil \log_2(n) \rceil$. If $\ell < \Lambda^{\text{nak}}(n)$ we use $\xi^{n,\text{nak}}_{\ell,k} := \xi^{n,\text{u}}_{\ell,k}$ and Lagrange polynomials

$$L_{\ell,k}(x) := \prod_{\substack{0 \leq m \leq 2^\ell, \\ m \neq k}} \frac{x - \xi^{n,\text{u}}_{\ell,m}}{\xi^{n,\text{u}}_{\ell,k} - \xi^{n,\text{u}}_{\ell,m}}, \quad k = 0, \ldots, 2^\ell. \tag{3.14}$$

This ensures a basis for the polynomial space on the first levels. Finally, the *hierarchical not-a-knot B-spline basis* $b^{n,\text{nak}}_{\ell,k}$ of degree n, level ℓ and index $0 \leq k \leq 2^\ell$ is defined as

$$b^{n,\text{nak}}_{\ell,k}(x) := \begin{cases} b^n_{k,\xi^{n,\text{nak}}_\ell}(x) & \ell \geq \Lambda^{\text{nak}}(n), \\ L_{\ell,k}(x) & \ell < \Lambda^{\text{nak}}(n). \end{cases} \tag{3.15}$$

These functions fulfill the Schoenberg-Whitney conditions and form a basis of the spline space $S^{n,\text{nak}}_\ell := S^n_{\xi^{n,\text{nak}}_\ell}$. An illustration of hierarchical not-a-knot B-splines can be found later in the overview Fig. 3.6B.

Hierarchical splitting. The main motivation for the development of the above basis was having a B-spline basis which can be used on sparse grids. The following theorem verifies that this goal has now been accomplished, i.e., Lm. 2.1 is satisfied. In particular, it shows that the span of hierarchical not-a-knot B-splines includes the polynomial space of highest achievable degree with respect to the number of basis functions. This is an elemental property for the approximation quality of the basis.

THEOREM 3.11 (hierarchical splitting for not-a-knot B-splines)
Let $V^{n,\text{nak}}_\ell$ be the nodal not-a-knot space as in Eq. (2.33) *and $W^{n,\text{nak}}_\ell$ be the hierarchical not-a-knot subspaces as in Eq.* (2.15)*. Then it holds*

$$S^{n,\text{nak}}_\ell = V^{n,\text{nak}}_\ell = \bigoplus_{\ell'=0}^{\ell} W_{\ell'}. \tag{3.16}$$

Furthermore, if $0 \leq \ell < \Lambda^{\text{nak}}(n)$, then $\mathbb{P}_{2^\ell} \subseteq S^{n,\text{nak}}_\ell$, and if otherwise $\ell \geq \Lambda^{\text{nak}}(n)$, then $\mathbb{P}_n \subseteq S^{n,\text{nak}}_\ell$, where $\mathbb{P}_n$ is the polynomial space of degree n on $[0,1]$.

PROOF See [Val19]. ∎

The approximation quality of splines persists on sparse grids. If the objective function is sufficiently smooth, the L^2-error of the interpolation with not-a-knot B-splines of degree n on a D-dimensional regular sparse grid of level ℓ decays asymptotically like

$$O(h_\ell^{n+1}(\log_2 h_\ell^{-1})^{D-1}) = O(2^{-(n+1)\ell}\ell^{D-1}), \tag{3.17}$$

see [Sic11] for more details.

3.2 Non-Boundary B-Splines

As we have seen in Sec. 2.4, the number of boundary points of a sparse grid still grows exponentially for increasing dimensionality. Thus, for higher dimensionalities the boundary points must be omitted and non-boundary grids must be used. However, without boundary points and the according basis functions, the B-spline basis does not span the space of splines anymore. So far, in a sparse grid context, this problem was addressed by using *modified B-splines*, which extrapolate towards the boundary by enforcing zero second derivatives. This works well if the objective function shares this property, but in general it does not. In particular the space of modified B-splines does not even include the monomials x^m for $m > 1$. Therefore, we now introduce two new spline bases for non-boundary sparse grids, which have previously been published in [Reh18; Reh21b]. The two bases are motivated by exact representation of polynomials, a reasonable minimum requirement for spline approximation. In practice, the new bases perform significantly better also for non-polynomial objective functions.

3.2.1 Modified B-Splines

To address an objective function with natural boundary conditions Pflüger introduced modified basis functions [Pfl10], which were further developed by Valentin [Val19]. The idea is to modify the hierarchical not-a-knot B-spline basis in such a way that the second derivative vanishes on the boundary. Hierarchical modified not-a-knot B-splines $b_{\ell,k}^{n,\mathrm{m}}$ of degree n, level $\ell \geq 1$, and index k are defined as,

$$b_{\ell,k}^{n,\mathrm{m}}(x) := \begin{cases} 1 & \ell = 1,\ k = 1, \\ b_{\ell,k}^{n,\mathrm{nak}}(x) + 2b_{\ell,k-1}^{n,\mathrm{nak}}(x) & \ell \geq 2,\ k = 1,\ n = 1, \\ b_{\ell,k}^{n,\mathrm{nak}}(x) - \frac{\frac{d^2}{dx^2} b_{\ell,k}^{n,\mathrm{nak}}(0)}{\frac{d^2}{dx^2} b_{\ell,k-1}^{n,\mathrm{nak}}(0)} b_{\ell,k-1}^{n,\mathrm{nak}}(x) & \ell \geq 2,\ k \in \{1, \ldots, \frac{n+1}{2}\}, n > 1, \\ b_{\ell,2^\ell-k}^{n,\mathrm{m}}(1-x) & \ell \geq 2,\ k \in \{2^\ell - \frac{n+1}{2}, \ldots, 2^\ell - 1\}, \\ b_{\ell,k}^{n,\mathrm{nak}}(x) & \text{otherwise.} \end{cases} \tag{3.18}$$

Note that for degree $n = 1$ the second derivative always vanishes. Therefore, the modification is defined as the linear continuation of the left-most and right-most B-splines. An illustration of hierarchical modified B-splines can be found later in the overview Fig. 3.6C.

3.2.2 Extended B-Splines

Extending B-splines is a technique for maintaining the polynomial representation property when omitting grid points. It was first presented in the context of weighted and extended B-splines (*WEB-splines*) [Höl01; Höl03], and then generalized for usage in hierarchical subdivision [Mar17]. The idea of the extension is to transfer the contribution of omitted B-splines to close remaining ones. This is made possible by Marsden's identity Thm. 3.5, which allows to represent the coefficients of omitted B-splines as linear combinations of remaining B-splines. Thus, the ability to represent polynomials exactly can be preserved. Following the approach of WEB-splines, we now introduce hierarchical extended not-a-knot B-splines for nonboundary sparse grids. First, we derive extended not-a-knot B-splines for fixed level ℓ, the generalization to a hierarchical basis then follows immediately.

Let $\mathbb{P}^n$ be the space of polynomials of degree $\leq n$. Höllig and Hörner simplify Marsden's identity (Thm. 3.5) for the uniform knot sequence. They deduce that the coefficients for the representation of any polynomial of degree $\leq n$ with respect to x are themselves again polynomials of degree $\leq n$, but with respect to the index k [Höl03]. The same reasoning with respect to the not-a-knot knot sequence yields that for all $m = 0, \ldots, n$ there exist polynomials $q_m \in \mathbb{P}^n$, such that for the coefficients $q_m(k), k \in K_\ell$ it holds

$$x^m = \sum_{k \in K_\ell} q_m(k) b_{\ell,k}^{n,\text{nak}}, \tag{3.19}$$

where $K_\ell = \{0, \ldots, 2^\ell\}$ is the full index set of level ℓ. We divide the index set K_ℓ into indices of outer boundary splines $K_\ell^{\text{o}} := \{0, 2^\ell\}$ and indices of inner splines $K_\ell^{\text{i}} := K_\ell \backslash K_\ell^{\text{o}} = \{1, \ldots, 2^\ell - 1\}$, such that

$$x^m = \sum_{i \in K_\ell^{\text{i}}} q_m(i) b_{\ell,i}^{n,\text{nak}} + \sum_{j \in K_\ell^{\text{o}}} q_m(j) b_{\ell,j}^{n,\text{nak}}. \tag{3.20}$$

The polynomials q_m are of degree $\leq n$, thus we can compute $q_m(j)$ from the values at any $(n+1)$ inner indices. In particular, we can determine q_m as the polynomial interpolant of the values $q_m(i),\ i \in K_\ell^{\text{i}}(j)$, where $K_\ell^{\text{i}}(j)$ are the $(n+1)$ inner indices $i \in K_\ell^{\text{i}}$ that are closest to j. Thus, there exist unique coefficients $e_{i,j} \in \mathbb{R}$, such that

$$q_m(j) = \sum_{i \in K_\ell^{\text{i}}(j)} e_{i,j} q_m(i). \tag{3.21}$$

We formulate the according system of linear equations, and solve it to obtain explicit values for $e_{i,j}$, which we call the *extension coefficients*. They are stated in Tab. 3.1. Any basis of $\mathbb{P}^n$ can be formulated as a linear combination of the monomials x^m. Therefore,

Degree n	Extension coefficients $[e_{1,0}, \dots, e_{n+1,0}]$
1	$[2, -1]$
3	$[5, -10, 10, -4]$
5	$\begin{cases} [8, -28, 42, -35, 20, -6] & \ell = 3, \\ [8, -28, 56, -70, 56, -21] & \ell > 3 \end{cases}$

TABLE 3.1 Extension coefficients $e_{i,j}$ for degrees $n \in \{1, 3, 5\}$. Only the coefficients for the extension at the left boundary are shown, i.e., $j = 0$. The right boundary is treated symmetrically. For degree 5 and level 3, the left and right extensions overlap, resulting in a special case.

when formulating the system of linear equations which yields the extension coefficients for an arbitrary basis, the resulting system is only a linear transformation of the system with respect to monomials. Consequently, the values of the extension coefficients are independent of our derivation based on monomials.

Note that it is not strictly necessary to use $K_\ell^{\mathrm{i}}(j)$, the inner indices which are closest to the outer index j, as any $(n+1)$ inner indices would do. However, using the closest indices makes the resulting basis more stable [Höl03]. See Fig. 3.4 for an illustration of the extension procedure.

We now represent an arbitrary polynomial $P \in \mathbb{P}^n$ using the extension coefficients. We call $K_\ell^{\mathrm{o}}(i) \coloneqq \{j \in K_\ell^{\mathrm{o}} \mid i \in K_\ell^{\mathrm{i}}(j)\}$ the dual of $K_\ell^{\mathrm{i}}(j)$. Following Eq. (3.20) it holds

$$P = \sum_{m=0}^{n} p_m x^m = \sum_{m=0}^{n} \sum_{i \in K_\ell^{\mathrm{i}}} p_m q_m(i) b_{\ell,i}^{n,\mathrm{nak}} + \sum_{m=0}^{n} \sum_{j \in K_\ell^{\mathrm{o}}} p_m q_m(j) b_{\ell,j}^{n,\mathrm{nak}} \tag{3.22}$$

for uniquely defined coefficients $p_m,\ q_m(i),\ q_m(j) \in \mathbb{R}$. Exploiting the finiteness of the sets K_ℓ^{i} and K_ℓ^{o}, we interchange the sums,

$$P = \sum_{i \in K_\ell^{\mathrm{i}}} \left(\sum_{m=0}^{n} p_m q_m(i) \right) b_{\ell,i}^{n,\mathrm{nak}} + \sum_{j \in K_\ell^{\mathrm{o}}} \left(\sum_{m=0}^{n} p_m q_m(j) \right) b_{\ell,j}^{n,\mathrm{nak}}. \tag{3.23}$$

By the definition of the extension coefficients $e_{i,j}$ in Eq. (3.21), and because $K_\ell^{\mathrm{o}}(i)$ is the dual of $K_\ell^{\mathrm{i}}(j)$ it holds

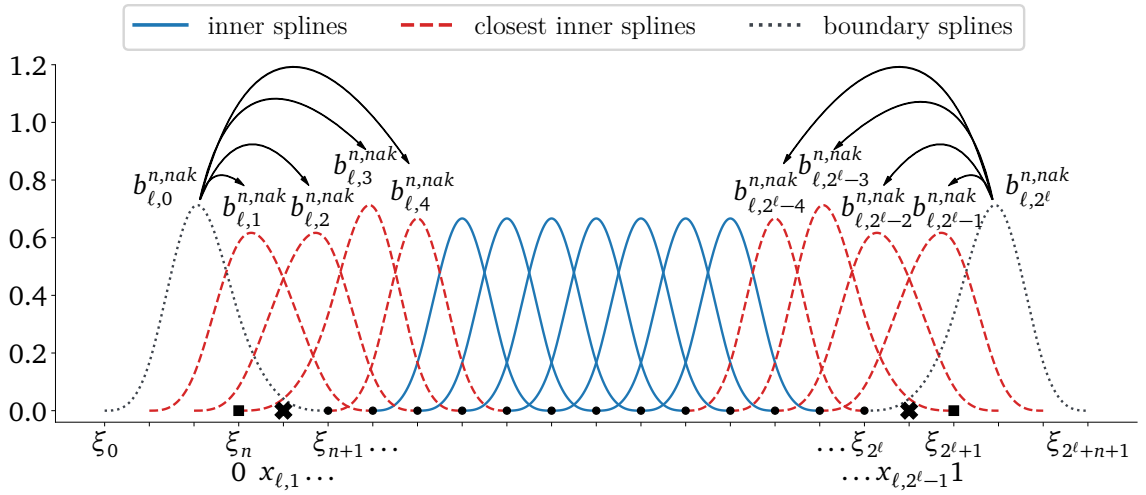


FIGURE 3.4 Schematic illustration of the extension of not-a-knot B-splines of degree $n = 3$ on a one-dimensional regular grid of level $\ell = 4$. The boundary splines with indices $j \in K_\ell = \{0, 16\}$ are added to the $n+1$ closest inner splines $K_\ell^{\mathrm{i}}(0) = \{1, 2, 3, 4\}$ and $K_\ell^{\mathrm{i}}(16) = \{12, 13, 14, 15\}$, indicated with arrows.

$$P = \sum_{i \in K_\ell^{\mathrm{i}}} \underbrace{\left(\sum_{m=0}^{n} p_m q_m(i) \right)}_{=:\beta_i} \underbrace{\left(b_{\ell,i}^{n,\mathrm{nak}} + \sum_{j \in K_\ell^{\mathrm{o}}(i)} e_{i,j} b_{\ell,j}^{n,\mathrm{nak}} \right)}_{=:b_{\ell,i}^{n,\mathrm{e}}} \tag{3.24}$$

$$= \sum_{i \in K_\ell^{\mathrm{i}}} \beta_i b_{\ell,i}^{n,\mathrm{e}}, \tag{3.25}$$

with coefficients $\beta_i \in \mathbb{R}$. Consequently, we define the extended not-a-knot B-spline $b_{\ell,i}^{n,\mathrm{e}}$ of degree n, level ℓ and index i as

$$b_{\ell,i}^{n,\mathrm{e}} := \begin{cases} b_{\ell,i}^{n,\mathrm{nak}} + \sum_{j \in K_\ell^{\mathrm{o}}(i)} e_{i,j} b_{\ell,j}^{n,\mathrm{nak}} & \ell \geq \Lambda(n), \\ L_{\ell,i}(x) & \ell < \Lambda(n), \end{cases} \tag{3.26}$$

where $\Lambda(n) := \lceil \log_2(n+2) \rceil$. Again Lagrange polynomials are employed on lower levels. This ensures the polynomial basis property, as long as there are not enough inner knots for the extension, i.e., less than $(n+1)$ inner knots. The corresponding knot sequence of degree n and level ℓ is $\xi_\ell^{n,\mathrm{e}} := (\xi_{\ell,1}^{n,\mathrm{nak}}, \ldots, \xi_{\ell,2^\ell+n}^{n,\mathrm{nak}})$. With this we have derived the following corollary.

COROLLARY 3.12 (extended not-a-knot B-splines recreate polynomials)
If $\ell \geq \Lambda(n)$, *then* $\mathbb{P}^n \subset \operatorname{span}\{b_{\ell,i}^{n,\mathrm{e}} \mid i \in K_\ell^{\mathrm{i}}\}$.

With this corollary we have achieved our goal of non-boundary B-spline basis func-

tions spanning the space of splines. It remains to prove the correctness of the hierarchical splitting for the hierarchical extended B-spline basis. In the next section we first introduce another approach for polynomial representing non-boundary B-splines, and then prove that both new bases span the same space.

3.2.3 Boundaryless not-a-knot B-Splines

The motivation for not-a-knot B-splines (see Sec. 3.1.2) is that dim $S^n_{\xi^{\mathrm{u}}} = z$ for uniform knots, while $D^n_{\xi^{\mathrm{u}}}$ contains only $z-(n-1)$ interpolation points. Therefore, another $(n-1)$ additional conditions are needed. Not-a-knot B-splines fix this issue by removing the $\frac{n-1}{2}$ left-most and $\frac{n-1}{2}$ right-most knots from the uniform knot sequence. Unfortunately, omitting the boundary points makes the problem reappear. Following the idea of the not-a-knot condition, we now introduce the boundaryless not-a-knot knot sequence $\xi^{n,\mathrm{bnk}}_\ell$, which revalidates the Schoenberg-Whitney conditions after omitting the boundary points. The approach itself is simple and intuitive. We remove the two boundary interpolation nodes of the not-a-knot sequence and compensate for this by omitting two more knots on top of the not-a-knot condition. Consequently, the $\frac{n+1}{2}$ left-most and $\frac{n+1}{2}$ right-most knots are removed from the uniform knot sequence, and the boundaryless not-a-knot knot sequence $\xi^{n,\mathrm{bnk}}_\ell = (\xi^{n,\mathrm{bnk}}_{\ell,0}, \dots, \xi^{n,\mathrm{bnk}}_{\ell,2^\ell+n-1})$ is defined through

$$\xi^{n,\mathrm{bnk}}_{\ell,k} := \begin{cases} \xi^{n,\mathrm{u}}_{\ell,k} & k = 0,\dots,n, \\ \xi^{n,\mathrm{u}}_{\ell,k+(n+1)/2} & k = n+1,\dots,2^\ell-2, \\ \xi^{n,\mathrm{u}}_{\ell,k+n+1} & k = 2^\ell-1,\dots,2^\ell+n-1. \end{cases} \tag{3.27}$$

This definition is only applicable if there are at least $(n+1)$ inner knots in $\xi^{n,\mathrm{nak}}_\ell$, which is the case for $\ell \geq \Lambda(n)$. The boundaryless not-a-knot B-spline basis $b^{n,\mathrm{bnk}}_{\ell,k}$ of degree n, level ℓ and index k is given by

$$b^{n,\mathrm{bnk}}_{\ell,k}(x) := \begin{cases} b^n_{k,\xi^{n,\mathrm{bnk}}_\ell}(x) & \ell \geq \Lambda(n), \\ L_{\ell,k}(x) & \ell < \Lambda(n). \end{cases} \tag{3.28}$$

We define the space of polynomial recreating splines of degree n and level $\ell \geq 1$,

$$S^{n,\mathrm{poly}}_\ell := \mathrm{span}\{b^{n,\mathrm{bnk}}_{\ell,k} \mid k \in K^{\mathrm{bnk}}_\ell := \{0,\dots,2^\ell-2\}\}, \tag{3.29}$$

and prove that the new basis indeed preserves the polynomial space $\mathbb{P}^n$ if $\ell \geq \Lambda(n)$, and otherwise preserves the polynomial space of highest achievable degree, given the number of available grid points.

LEMMA 3.13 (space of polynomial recreating splines)
If $\ell \geq \Lambda(n)$*, then* $\mathbb{P}^n \subseteq S_\ell^{n,\text{poly}}$*. If otherwise* $0 < \ell < \Lambda(n)$*, then* $\mathbb{P}^{2^\ell - 1} \subseteq S_\ell^{n,\text{poly}}$.

PROOF For $0 < \ell < \Lambda(n)$, the boundaryless not-a-knot B-splines $b_{\ell,k}^{n,\text{bnk}}$ are defined to be Lagrange polynomials and thus span the according polynomial space of polynomial degree $2^\ell - 1$. Let $\ell \geq \Lambda(n)$, then from Eq. (3.27) it follows immediately that $D^n_{\xi_\ell^{n,\text{bnk}}} = [0,1]$. Furthermore, the knot-sequence is n-regular, and thus the lemma follows from Marsden's identity (Thm. 3.5). ■

Hierarchical boundaryless not-a-knot B-splines are defined in the usual way with a slight adjustment. Because we have removed an extra knot on the left side, the indexing is off and the basis function associated with grid point $x_{\ell,k}$ is $b_{\ell,k-1}^{n,\text{bnk}}$. In order to correct this, we introduce one auxiliary knot and its according basis function for the definition of hierarchical boundaryless not-a-knot B-splines. It does not matter where this knot lies, as long as it is to the left of all other knots. For simplicity, we use $\xi_{\ell,-1}^{n,\text{u}}$, which is the left-next point of the underlying uniform knot sequence, cf. Eq. (3.27). By adding this knot and its according basis functions the indices of the basis functions to the right of it are increased by one, such that now the grid point $x_{\ell,k}$ consistently corresponds to the basis function $b_{\ell,k}^{n,\text{bnk}}$. The boundaryless not-a-knot sequence including the auxiliary knot is illustrated in Fig. 3.5, hierarchical boundaryless not-a-knot B-splines are illustrated in Fig. 3.6E.

In the context of sparse grids, it has already been shown that hierarchical uniform B-splines and hierarchical not-a-knot B-splines of degrees $n \in \{1,3,5\}$ are linearly independent [Val19]. The same holds for hierarchical boundaryless not-a-knot B-splines. We now show that they span the same space $S_\ell^{n,\text{poly}}$ as the full basis of level ℓ.

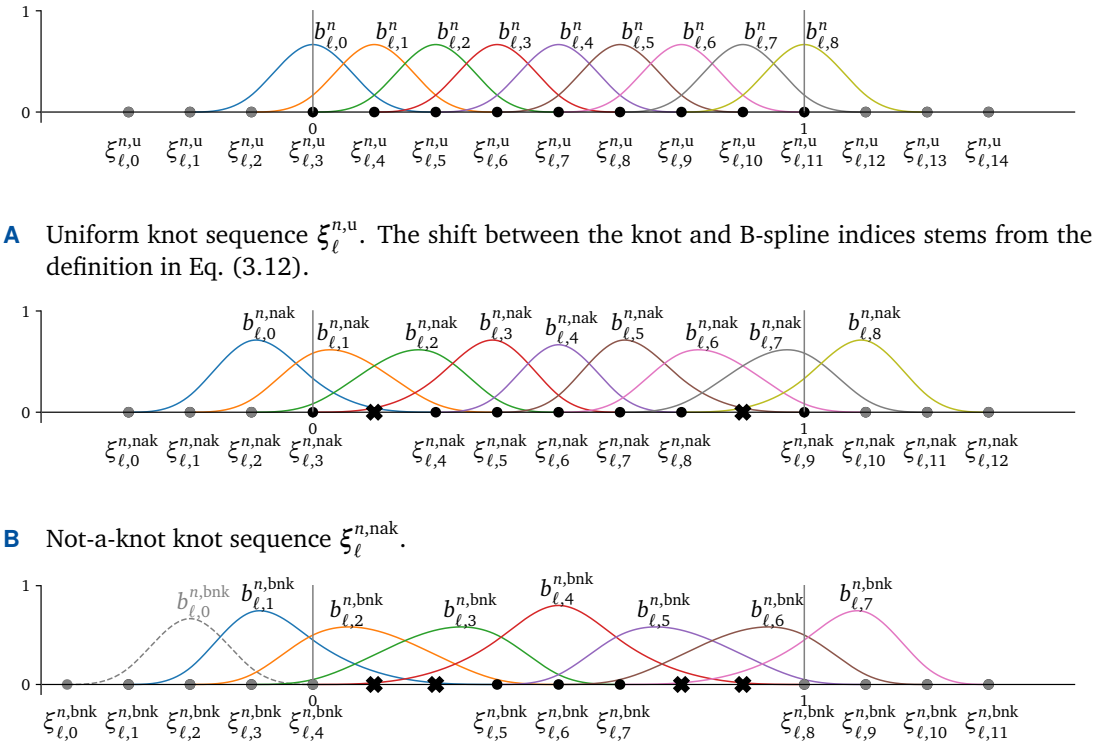
THEOREM 3.14 (hierarchical boundaryless nak B-splines are linearly independent)
For degree $n \in \{1,3,5\}$ *the hierarchical boundaryless not-a-knot splines*

$$\{b_{\ell',k}^{n,\text{bnk}} \mid 1 \leq \ell' \leq \ell,\ k \in I_{\ell'}\} \tag{3.30}$$

are a basis of $S_\ell^{n,\text{poly}}$.

PROOF Uses Thm. 2.2. Details in Appendix A.1.1. ■

Extended not-a-knot B-splines are defined by adding outer basis functions to the left-most and right-most inner basis functions. It is astonishing that the resulting basis is independent of the actual position of the outer knots. The basis solely depends on the inner knots and the degree. We use this property to prove that extended and boundaryless not-a-knot B-splines span the same spline space.



A Uniform knot sequence $\xi^{n,\mathrm{u}}_{\ell}$. The shift between the knot and B-spline indices stems from the definition in Eq. (3.12).

B Not-a-knot knot sequence $\xi^{n,\mathrm{nak}}_{\ell}$.

C Boundaryless not-a-knot knot sequence $\xi^{n,\mathrm{bnk}}_{\ell}$. The auxiliary knot and auxiliary basis function for the index correction are shown in dashed gray.

FIGURE 3.5 Overview of knot sequences of level $\ell = 3$, and nodal B-splines of degree $n = 3$ on the unit interval $[0, 1]$. Knots which are removed because of the not-a-knot condition are indicated with black crosses.

THEOREM 3.15 (extended and boundaryless nak B-splines span the same space) *Let* $S^{n,\mathrm{e}}_{\ell} := \operatorname{span}\{b^{n,\mathrm{e}}_{\ell,i} \mid i \in \{1, \ldots, 2^{\ell} - 1\}\}$. *Then* $S^{n,\mathrm{e}}_{\ell} = S^{n,\mathrm{poly}}_{\ell}$.

PROOF See Appendix A.1.2. ∎

With this we have achieved our goal. The extended not-a-knot B-spline basis, as well as the boundaryless not-a-knot B-spline basis, for the first time, allow differentiable higher-order approximations on non-boundary sparse grids. In terms of numerical approximation there is no difference between the two bases when using regular grids. However, the basis functions are different and so are their surpluses. Thus, when using spatially adaptive grids the two bases perform slightly differently.

Interpolation condition. It is not immediately clear to what extent the adjustments of extended and boundaryless not-a-knot B-splines influence their practical applicability. In case of the extended not-a-knot B-splines, adding outer basis functions to inner ones, and

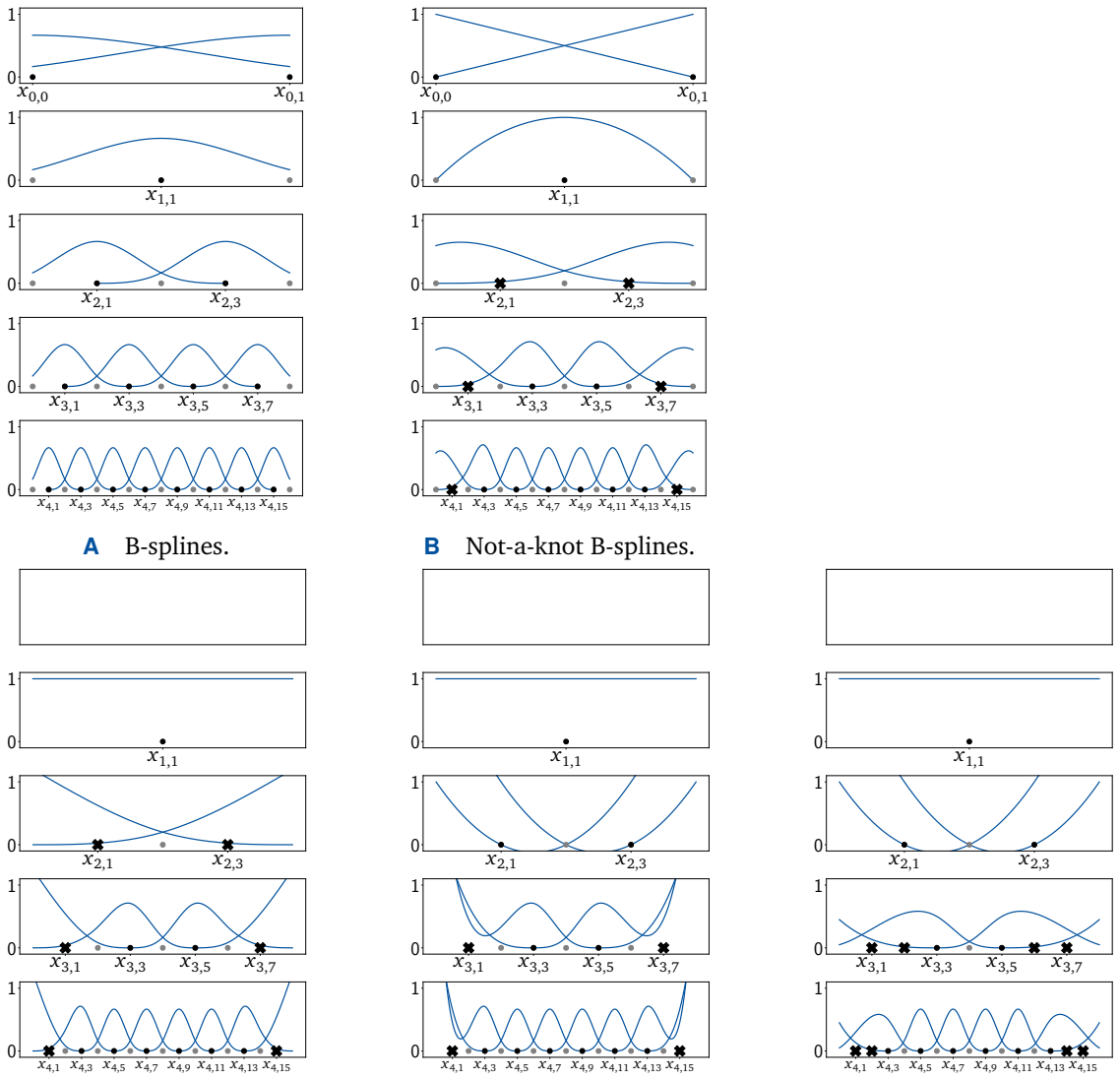


FIGURE 3.6 Hierarchical cubic B-spline bases of level $0, \ldots, 4$. Basis functions including the boundary in the upper row, non-boundary bases in the lower row. The not-a-knot condition is indicated with crosses.

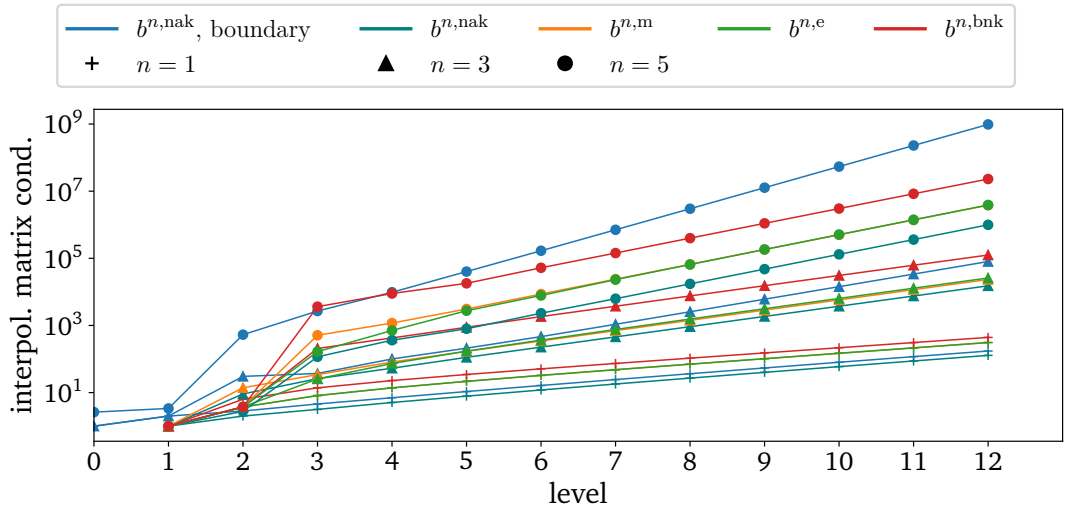


FIGURE 3.7 Interpolation matrix condition for univariate B-splines of degrees $n \in \{1, 3, 5\}$ on regular sparse grids. On level 0 only boundary basis functions exist. For levels 1 and 2 Lagrange polynomials are used by the respective bases, which causes the jumps in between level 2 and 3. From level 3 onwards the condition of all non-boundary basis functions of the same degree increase similarly except for a constant factor.

thus increasing their support, might severely impair the condition $\kappa(\boldsymbol{A})$ of the interpolation matrix $\boldsymbol{A}$, which is defined as

$$\kappa(\boldsymbol{A}) := \|\boldsymbol{A}\|_2 \, \|\boldsymbol{A}^{-1}\|_2. \tag{3.31}$$

However, this is not the case. In Fig. 3.7, we see the condition of the univariate interpolation matrix with respect to a univariate regular sparse grid for not-a-knot B-splines on a boundary grid, modified, extended, and boundaryless not-a-knot B-splines. For each degree $n \in \{1, 3, 5\}$, the interpolation conditions of all non-boundary basis functions increase similarly and differ only by a constant factor. Not-a-knot B-splines on non-boundary grids always have the smallest condition. However, they are only included as a reference for unadjusted behavior. They are not usable for applications in practice as will be demonstrated in the next section. Extended not-a-knot B-splines generally have a smaller interpolation condition than boundaryless not-a-knot B-splines. For increasing degree, the condition of not-a-knot B-splines on boundary grids becomes comparatively worse until, for degree $n = 5$, the boundary points and according basis functions impair the interpolation condition more than the adjustments of extended and boundaryless not-a-knot B-splines.

Fundamental property. Apart from our newly introduced hierarchical extended and boundaryless not-a-knot B-splines, there exist other B-spline-related basis functions that were specifically designed for usage on sparse grids. Recall that hierarchical C^0 elements are defined to fulfill the same fundamental property as linear hat functions, i.e.,

$$\begin{aligned} \varphi_{\ell,i}(x_{\ell',i'}) &= 0 \qquad \ell' < \ell, i' \in I_{\ell'}, \\ \varphi_{\ell,i}(x_{\ell,i'}) &= \delta_{i,i'} \qquad\quad i' \in I_{\ell}, \end{aligned} \tag{3.32}$$

cf. Sec. 2.2. This property allows fast hierarchization through the univariate principle, without the need to solve a system of linear equations. B-splines have larger support, which on the one hand contributes to their improved approximation quality. On the other hand it makes solving a system of linear equations necessary when preparing a sparse grid surrogate. Therefore, weakly fundamental B-splines $b^{n,\mathrm{wf}}$ and modified weakly fundamental B-splines $b^{n,\mathrm{wf,m}}$ have been developed [Val19]. These bases fulfill the fundamental property (3.32) by construction and can make use of the quick hierarchization. However, the support of the basis functions is increased, and some basis functions must be constructed and stored individually. Furthermore, although no system of linear equations must be solved, either a breadth first search must be performed, which reintroduces complexity and run-time, or some additional grid points must be added to enable the unidirectional principle. The remaining advantage of weakly fundamental splines is their reduced memory requirement. In the next section we demonstrate that, with respect to accuracy, weakly fundamental splines are generally outperformed by extended and boundaryless not-a-knot B-splines and therefore we do not use them in practice.

3.3 Numerical Comparison

We now compare the approximation quality of the basis functions which we have introduced in the last sections. For this we use straightforward objective functions whose smoothness and behavior is known, and we demonstrate how this influences the approximation quality of the individual bases.

Let $f : \Omega \subset \mathbb{R}^D \to \mathbb{R}$ be an objective function. We interpolate it using sparse grids and various basis functions resulting in a surrogate $\tilde{f}$, respectively. Then, we compute the Root Mean Square Error (RMSE), also known as discrete L^2-error, and the more informative Normalized Root Mean Square Error (NRMSE),

$$\mathrm{RMSE}(f,\tilde{f}) := \left(\sum_{m=1}^{M}\left(f(\boldsymbol{x}_m)-\tilde{f}(\boldsymbol{x}_m)\right)^2\right)^{1/2}, \tag{3.33}$$

$$\mathrm{NRMSE}(f,\tilde{f}) := \frac{RMSE(f,\tilde{f})}{\max_m(f(\boldsymbol{x}_m))-\min_m(f(\boldsymbol{x}_m))}, \tag{3.34}$$

based on $M = 10\,000$ uniformly distributed random samples $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_M \in \Omega$.

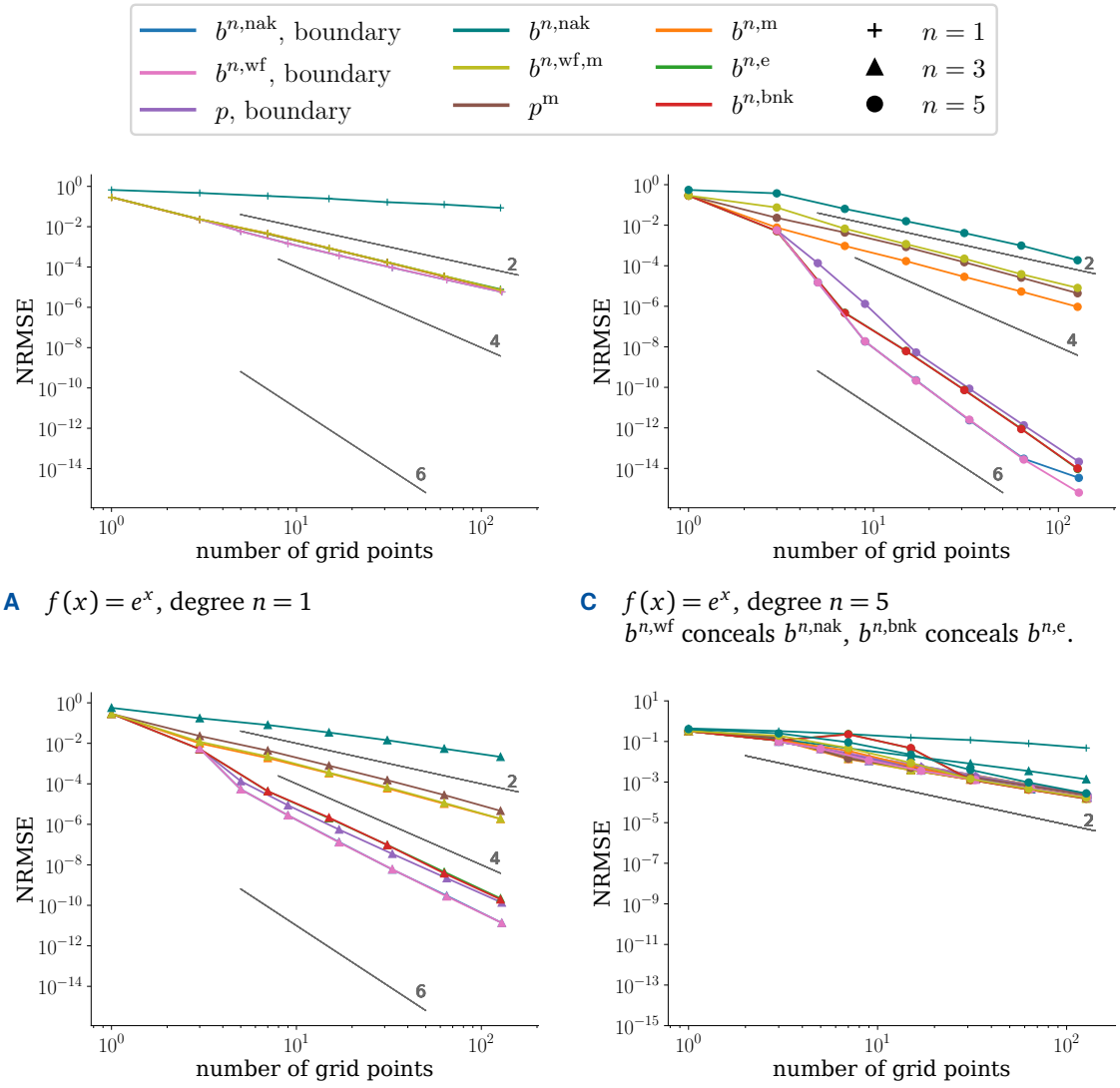
We first use regular sparse grids for univariate objective functions to demonstrate the influence of the objective's smoothness. Then, we use regular sparse grids for multivariate functions to demonstrate the influence of high dimensionality. Finally, we show the additional improvements of spatial adaptivity. Our comparison includes classical hierarchical C^0 elements p on boundary grids, and their modified counterparts p^{m} for non-boundary grids, not-a-knot B-splines $b^{n,\mathrm{nak}}$ on boundary grids and on non-boundary grids, as well as their modified counterparts for non-boundary grids $b^{n,\mathrm{m}}$. Furthermore, we compare with weakly fundamental not-a-knot B-splines, again for boundary grids $b^{n,\mathrm{wf}}$, and modified for non-boundary grids $b^{n,\mathrm{wf,m}}$. Finally, we include the two new bases, which were introduced in the last section: extended not-a-knot B-splines $b^{n,\mathrm{e}}$ and boundaryless not-a-knot B-splines $b^{n,\mathrm{bnk}}$. Both were designed to outperform and replace the modification of basis functions. All mentioned basis functions are applied with degrees $n \in \{1,3,5\}$.

Note that similar basis functions, like $b^{n,\mathrm{e}}$ and $b^{n,\mathrm{bnk}}$, perform alike on regular grids and thus in the plots one conceals the other. We indicate this in the text accordingly. However, on spatially adaptive grids the different coefficients lead to slightly different refinement behavior. Therefore, although two bases theoretically span the same space, the quality of their adaptive interpolants usually differs.

3.3.1 Univariate Functions

We use two univariate objective functions to demonstrate the direct link between smoothness of the objective function and the quality of the approximations. The results are displayed in Fig. 3.8.

Smooth function. Our first objective function, $f(x) = e^x$, is infinitely smooth. In one dimension, there are only two boundary points, independent of the sparse grid's level and thus they do not yet have a notable negative influence. In fact, the basis functions on boundary grids perform the best overall and reach optimal convergence rates. The extended B-splines and boundaryless not-a-knot B-splines have the same optimal order of convergence, and their NRMSE is in between the boundary splines and the boundary C^0 elements. The modified functions perform much worse, because the exponential function does not have zero second derivatives. The modified bases are all stuck at



A $f(x) = e^x$, degree $n = 1$

C $f(x) = e^x$, degree $n = 5$
$b^{n,\text{wf}}$ conceals $b^{n,\text{nak}}$, $b^{n,\text{bnk}}$ conceals $b^{n,\text{e}}$.

B $f(x) = e^x$, degree $n = 3$
$b^{n,\text{wf}}$ conceals $b^{n,\text{nak}}$, $b^{n,\text{bnk}}$ conceals $b^{n,\text{e}}$.

D $f(x) = |x - \frac{1}{3}|$, degrees $n \in \{1, 3, 5\}$

FIGURE 3.8 Convergence of regular sparse grid approximations for univariate objective functions. Convergence orders in gray.
A-C: $f(x) = e^x$ is in C^∞ but optimal convergence rates are achieved only by boundary basis functions and the new bases $b^{n,\text{e}}$ and $b^{n,\text{bnk}}$.
D: $f(x) = |x - \frac{1}{3}|$ is in C^0, convergence is limited overall.

the convergence order of linear functions, independent of their degree. Here, and for all examples to come, the not-a-knot B-splines without boundary points or boundary adjustment perform considerably worse than all other bases. This clearly demonstrates the necessity for boundary adjustment when using non-boundary sparse grids.

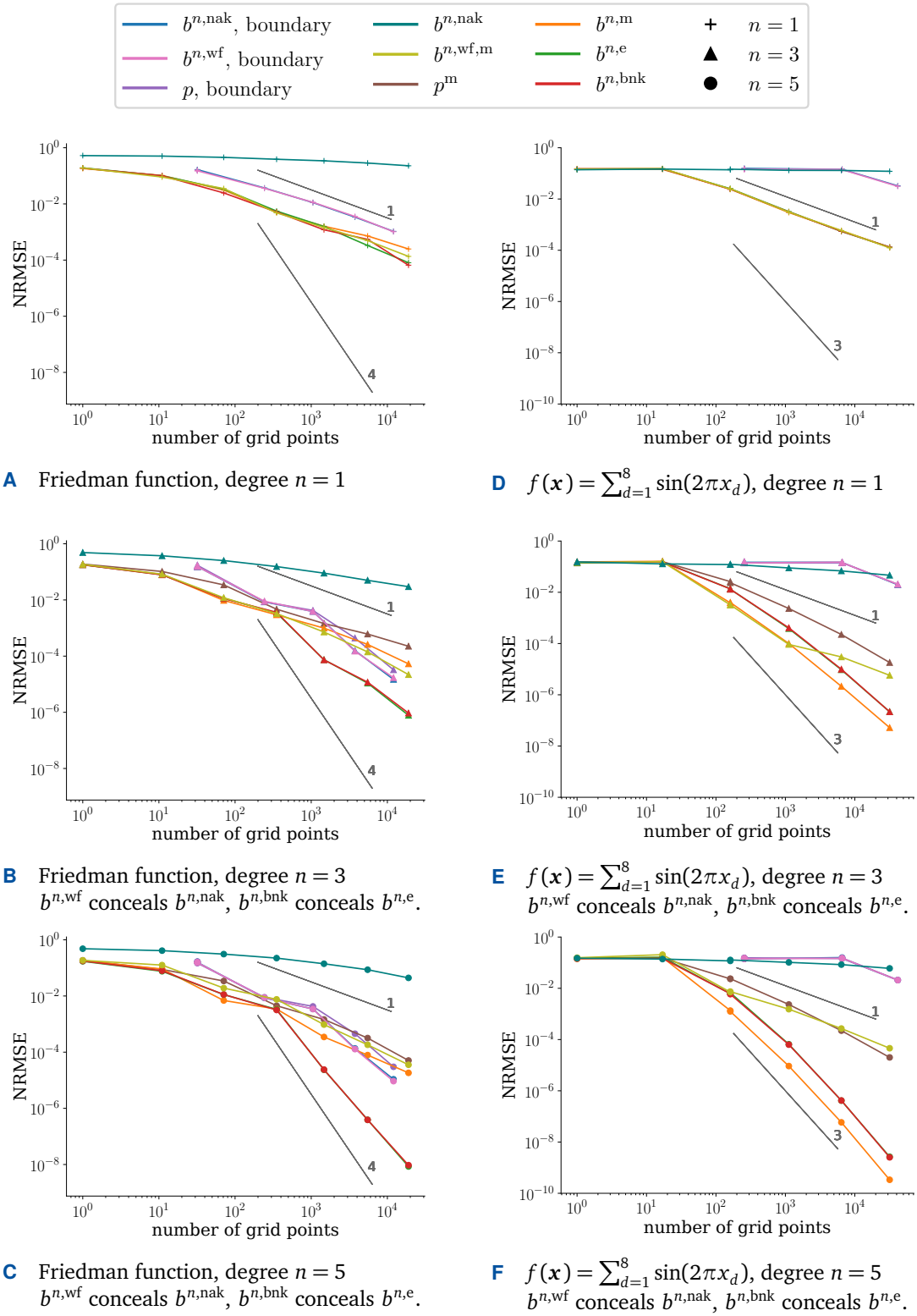
Non-differentiable function. The absolute value function $f(x) = |x - \frac{1}{3}|$ is continuous but has a non-differentiable kink in $x = \frac{1}{3}$. Figure 3.8D shows that this leads to a convergence rate of h^{-2} for all basis functions, independent of their degree, and demonstrates the necessity for sufficient smoothness of the objective function. However, this limited convergence can be overcome through spatial adaptivity as can be seen later in Fig. 3.10.

3.3.2 Multivariate Functions

The next two objective functions demonstrate that for increasing dimension, extended and boundaryless not-a-knot B-splines outperform the other basis functions. The results are displayed in Fig. 3.9.

Friedman. The 5-dimensional benchmark function $f(\boldsymbol{x}) = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5$ was introduced by Friedman [Fri83; Fri91]. Five dimensions are already enough for non-boundary sparse grids to significantly outperform their boundary counterparts. In Figures 3.9A to 3.9C extended and boundaryless not-a-knot B-splines outperform all other basis functions notably. Their NRMSE is up to four orders of magnitude smaller than that of the other bases.

Zero second derivatives. The function $f(\boldsymbol{x}) = \sum_{d=1}^{8} \sin(2\pi x_d)$ has zero second derivatives at the boundary of the domain $[0, 1]^8$. This is the ideal scenario for modified B-spline basis functions, and indeed they perform best, as can be seen in Figures 3.9D to 3.9F. However, the new extended B-splines and boundaryless not-a-knot B-splines show the same order of convergence. This demonstrates the generality of the new basis functions. If no knowledge on the derivatives is available, e.g., for black-box simulations, the new basis functions can always be applied, without risking a significant drawback, making modified B-splines obsolete in practice. The boundary sparse grids have become unusable in this 8-dimensional space. They spend large amounts of grid points on the boundary without notably resolving the inner domain. The other basis functions do not suffer as much from the higher dimensionality. Still, their convergence order decreases.



A Friedman function, degree $n = 1$

D $f(\boldsymbol{x}) = \sum_{d=1}^{8} \sin(2\pi x_d)$, degree $n = 1$

B Friedman function, degree $n = 3$
$b^{n,\mathrm{wf}}$ conceals $b^{n,\mathrm{nak}}$, $b^{n,\mathrm{bnk}}$ conceals $b^{n,\mathrm{e}}$.

E $f(\boldsymbol{x}) = \sum_{d=1}^{8} \sin(2\pi x_d)$, degree $n = 3$
$b^{n,\mathrm{wf}}$ conceals $b^{n,\mathrm{nak}}$, $b^{n,\mathrm{bnk}}$ conceals $b^{n,\mathrm{e}}$.

C Friedman function, degree $n = 5$
$b^{n,\mathrm{wf}}$ conceals $b^{n,\mathrm{nak}}$, $b^{n,\mathrm{bnk}}$ conceals $b^{n,\mathrm{e}}$.

F $f(\boldsymbol{x}) = \sum_{d=1}^{8} \sin(2\pi x_d)$, degree $n = 5$
$b^{n,\mathrm{wf}}$ conceals $b^{n,\mathrm{nak}}$, $b^{n,\mathrm{bnk}}$ conceals $b^{n,\mathrm{e}}$.

FIGURE 3.9 Convergence of regular sparse grid approximations for multivariate objective functions. Convergence orders in gray.
A-C: The new bases $b^{n,\mathrm{e}}$ and $b^{n,\mathrm{bnk}}$ outperform all other bases for the 5-dimensional Friedman function.
D-F: The non-boundary splines $b^{n,\mathrm{m}}$, $b^{n,\mathrm{e}}$ and $b^{n,\mathrm{bnk}}$ outperform all others for the 8-dimensional trigonometric function.

3.3.3 Spatial Adaptivity

Now that we have seen the regular convergence behavior, we demonstrate that spatial adaptivity can significantly improve the approximation quality. To do so, we reconsider two of last section's objective functions: The univariate absolute function, which is not differentiable, and the 8-dimensional sum of sines, which results in slower convergence rates because of its higher dimensionality. The results are displayed in Fig. 3.10.

Non-differentiable function. The non-differentiability of $f(x) = |x - \frac{1}{3}|$ limits the convergence of all basis functions on regular sparse grids. Spatial adaptivity overcomes this. Instead of spreading grid points over the whole domain, it consecutively refines towards the non-differentiable kink. With this, lower-order basis functions are able to resolve the function up to numerical precision, which is shown in Figures 3.10A to 3.10C. Higher-order functions still struggle, because close to the discontinuity their approximation is deteriorated by oscillations introduced by the Gibbs phenomenon.

Sum of sines. For regular sparse grids, the highest achievable convergence rate is limited, but spatial adaptivity employs grid points much more efficiently. After an initial scan of the domain, spatial adaptivity boosts the convergence so significantly that it even exceeds the optimal convergence rates of regular grids. Figures 3.10D to 3.10F show that modified, extended, and boundaryless not-a-knot B-splines perform all equally well, leaving the other basis functions behind. The NRMSE is several orders of magnitude better than for regular grids in Fig. 3.9.

Summary. In this chapter we presented the status-quo of B-spline basis functions for sparse grids. We then introduced two new bases, extended not-a-knot B-splines, and boundaryless not-a-knot B-splines, which are both designed to maintain the polynomial recreation property on nonboundary sparse grids. With this, higher-order differentiable basis functions for non-boundary sparse grids are available for the first time. In the final section of this chapter we compared the new bases with established sparse grid basis functions, and the new bases always performed comparable or better.

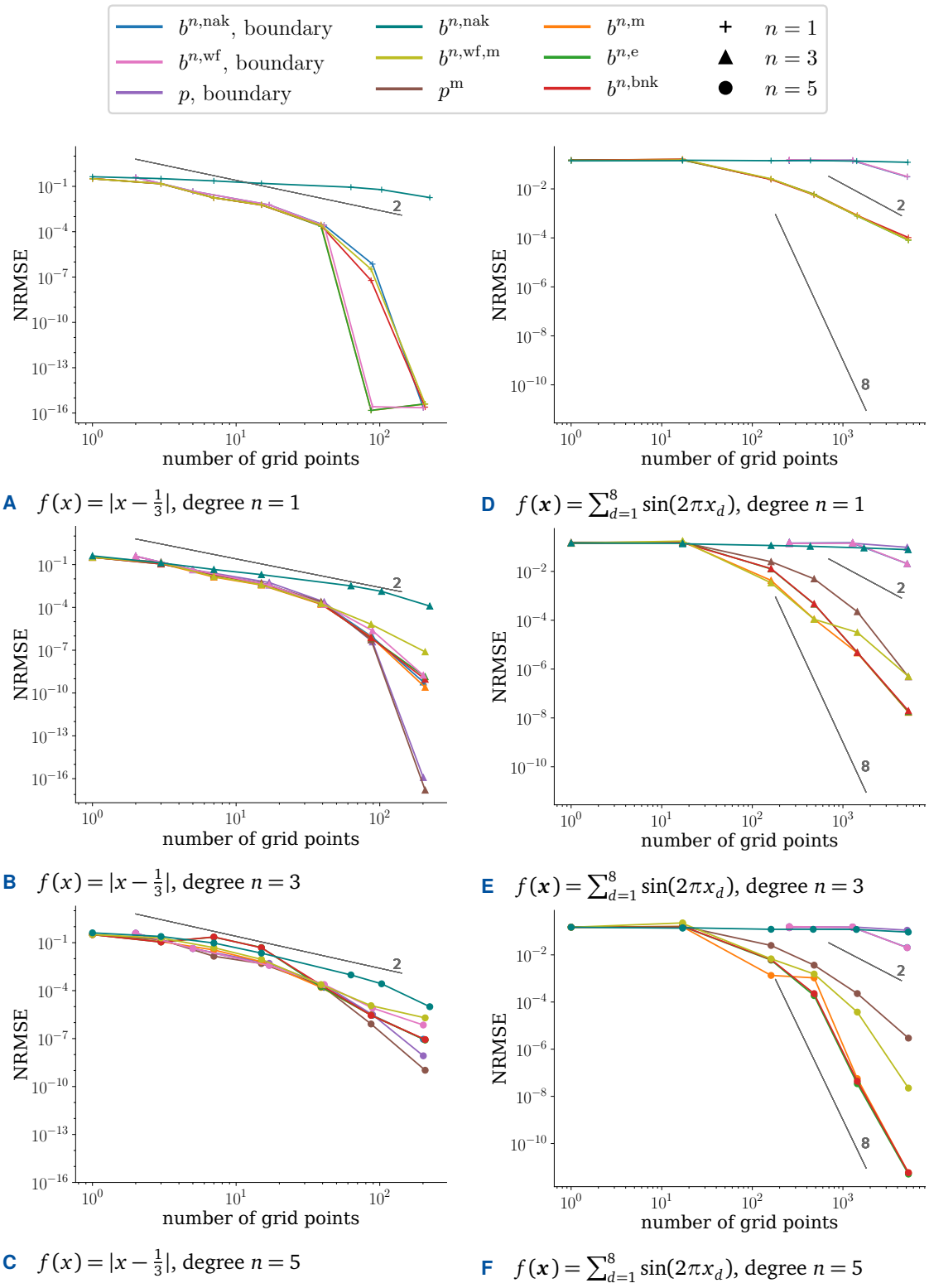


FIGURE 3.10 Convergence of spatially adaptive sparse grid approximations. Convergence orders in gray.
A-C: Spatial adaptivity allows accurate representation of the C^0 function for lower order bases.
D-F: Spatial adaptivity significantly improves the approximation order for the 8-dimensional triangular function.

4

Uncertainty Quantification

> *All models are wrong. The practical question is how wrong do they have to be to not be useful.*
>
> — George E. P. Box [Box87]

Uncertainty Quantification (UQ) is a broad term, which may be summarized as the treatment of models with stochastic methods to incorporate uncertainties, primarily but not exclusively, of the input parameters. This is particularly important for the assessment of risks, estimating the reliability of systems, and characterizing and reducing uncertainties. Stochastic formulations allow handling many effects, which otherwise would have to be neglected due to complexity or insufficient information. Uncertainty quantification proceeds in two directions: *forward*, which means that uncertainties are propagated through the model to "determine the effect of input uncertainties on response metrics of interest" [Eld09a], and *inverse*, where potential original states are assigned the likelihood of resulting in observed outcomes. Both directions have in common that the focus lies on probability distributions instead of scalar values, allowing a holistic understanding instead of specific isolated results.

Simulations in the context of UQ are often very expensive in terms of computation time and hardware resources. Therefore, a detailed investigation solely based on evaluations of the simulation is usually inefficient, or might even be unfeasible. This problem can be dealt with by creating a *surrogate*, that is, a sufficiently accurate approximation for the original model.

In this chapter, we first establish the stochastic notation in Sec. 4.1, before we introduce two expansion methods in Sec. 4.2. The first, stochastic collocation, allows forward propagation of uncertainty through our surrogates. The second, polynomial chaos expansion, is a popular surrogate technique in UQ, which we use for comparison in our applications. In Sec. 4.3 we then introduce Bayesian inference, which is the most prominent technique for inverse UQ. Bayesian inference relies primarily on efficient sampling. We reference a state-of-the-art sampling algorithm, and explain how it can be enabled through our surrogates, accelerating this difficult task. Finally, in Sec. 4.4 we first define active subspaces, an emerging dimension reduction technique based on sensitivity analysis, before we present a new integration algorithm based on active subspaces and B-splines.

This chapter mainly summarizes the stochastic basics required for the UQ applications in the upcoming chapters, and Sec. 4.2 has already been published as part of [Reh21b]. Section 4.4 contains the second major contribution to the theory of B-splines on sparse grids for UQ: the new integration algorithm for functions with a one-dimensional active subspace. The algorithm has already been published in [Reh19], but the formulated proof for Thm. 4.4 has never been presented before.

4.1 Introduction to UQ

Before we can present the new methods, we must establish the probabilistic notation.

Basic probability theory. Let $\boldsymbol{X}$ be a random variable with values in $\Omega \subseteq \mathbb{R}^D$, and a probability density function ϱ_X such that the probability of any event $A \subseteq \Omega$ is

$$P(A) = P(X \in A) = \int_A \varrho_X(\boldsymbol{x}) d\boldsymbol{x}, \tag{4.1}$$

and it holds

$$\int_\Omega \varrho_X(\boldsymbol{x}) d\boldsymbol{x} = 1, \quad \varrho_X > 0. \tag{4.2}$$

We restrict ourselves to the hypercube case,

$$\Omega := \bigotimes_{d=1}^{D} \Omega_d := \bigotimes_{d=1}^{D} [\ell_d, u_d], \tag{4.3}$$

where $\ell_d, u_d \in \mathbb{R}$, and $\ell_d < u_d$ for $d = 1, \dots, D$. Random variables $\boldsymbol{X}$ and $\boldsymbol{Y}$ are independent if and only if their joint density $\varrho_{X,Y}$ satisfies

$$\varrho_{X,Y}(\boldsymbol{x}, \boldsymbol{y}) = \varrho_X(\boldsymbol{x})\varrho_Y(\boldsymbol{y}), \tag{4.4}$$

for all $\boldsymbol{x}, \boldsymbol{y} \in \Omega$. The *expected value*, or *mean* of $\boldsymbol{X}$ is defined as

$$\mathbb{E}[\boldsymbol{X}] := \int_\Omega \boldsymbol{x}\, \varrho_X(\boldsymbol{x}) d\boldsymbol{x}. \tag{4.5}$$

The *variance* of $\boldsymbol{X}$ is defined as the expected value of the squared deviation from the mean of $\boldsymbol{X}$,

$$\mathbb{V}[\boldsymbol{X}] := \mathbb{E}[(\boldsymbol{X} - \mathbb{E}[\boldsymbol{X}])^2], \tag{4.6}$$

which can be expanded to

$$\mathbb{V}[\boldsymbol{X}] = \mathbb{E}[\boldsymbol{X}^2] - \mathbb{E}[\boldsymbol{X}]^2. \tag{4.7}$$

The *standard deviation* of $\boldsymbol{X}$ is given by

$$s[\boldsymbol{X}] := +\sqrt{\mathbb{V}[\boldsymbol{X}]}. \tag{4.8}$$

Given a data-set of $M \in \mathbb{N}$ values, for example stemming from measurements or model evaluations the p-th *percentile* is the smallest value in this set, such that p percent of the data are less than or equal to that value.

Monte Carlo. The oldest and most simple approach in UQ is the Monte Carlo (MC) method, which is still relevant and widely used today. It was introduced by John von Neumann and Stanislaw Ulam, who worked for the nuclear weapons project in the 1940s [Bor19]. Being secret, the project required a code name, and because of the underlying random sampling, it was named after the casino in Monaco, where Ulam's uncle used to gamble [Met87]. One of the main purposes of the Monte Carlo method is integration, which makes it popular for the calculation of stochastic moments. The above stochastic quantities can be approximated by drawing a set of M samples $\boldsymbol{x}_1, \dots, \boldsymbol{x}_M$ from the distribution ϱ_X and calculating

$$\mathbb{E}_{\mathrm{MC}}[\boldsymbol{X}] := \frac{1}{M} \sum_{m=1}^{M} \boldsymbol{x}_m, \tag{4.9}$$

$$\mathbb{V}_{\mathrm{MC}}[\boldsymbol{X}] \coloneqq \frac{1}{M}\sum_{m=1}^{M}(\boldsymbol{x}_m - \mathbb{E}_{\mathrm{MC}}[\boldsymbol{X}])^2\,, \tag{4.10}$$

$$s_{\mathrm{MC}}[\boldsymbol{X}] \coloneqq +\sqrt{\mathbb{V}_{\mathrm{MC}}[\boldsymbol{X}]}. \tag{4.11}$$

The law of large numbers states that the above terms almost surely converge towards $\mathbb{E}[\boldsymbol{X}]$, $\mathbb{V}[\boldsymbol{X}]$, and $s[\boldsymbol{X}]$ respectively [Sul15]. In practice, often one does not know the density from which to sample explicitly. Instead, samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M$ are provided in the form of measurements or data, and one assumes that these samples follow an unknown distribution. Monte Carlo calculations are intuitive and simple to apply, which makes them very popular. However, Monte Carlo integration only converges at a rate of $O(\sqrt{M})$ [Caf98]. This means that to halve the error for M samples, $4M$ samples are needed, which is too slow for many practical applications.

Multivariate objectives. When dealing with time-dependent problems, the objective is often formulated as a multivariate function, mapping from the parameter space to the discretized space of $T \in \mathbb{N}$ time steps,

$$\begin{aligned} f : \Omega \subseteq \mathbb{R}^D &\to \mathbb{R}^T \\ \boldsymbol{X} &\mapsto (f_1(\boldsymbol{X}), \ldots, f_T(\boldsymbol{X}))\,. \end{aligned} \tag{4.12}$$

For such multivariate quantities, we define the stochastic moments component-wise, i.e.,

$$\begin{aligned} \mathbb{E}[f] &= (\mathbb{E}[f_1], \ldots, \mathbb{E}[f_T]) \in \mathbb{R}^T, \\ \mathbb{V}[f] &= (\mathbb{V}[f_1], \ldots, \mathbb{V}[f_T]) \in \mathbb{R}^T, \\ s[f] &= (s[f_1], \ldots, s[f_T]) \in \mathbb{R}^T. \end{aligned} \tag{4.13}$$

Sources of uncertainty. Uncertainty is a very broad term and can arise from a wide variety of sources. Classically it is classified into two categories, *aleatoric* uncertainty and *epistemic* uncertainty. Aleatoric uncertainty is inherent randomness, the knowledge of which cannot be determined sufficiently enough. However, aleatoric uncertainty can often be described by data, and thus it can still be taken into account. Epistemic uncertainty refers to a lack of knowledge, or intentional simplifications. Usually both kinds of uncertainties are present in real-world applications. A further distinction of uncertainty sources includes *observational uncertainty* stemming from noise and inaccurate measurements, *structural model uncertainty* coming from a lack of knowledge or missing relevant physical processes when modeling the computer simulation, and *numerical uncertainty* which is introduced by inaccuracies of numerical methods.

A key element of UQ is the focus on distributions and probabilities, instead of scalar

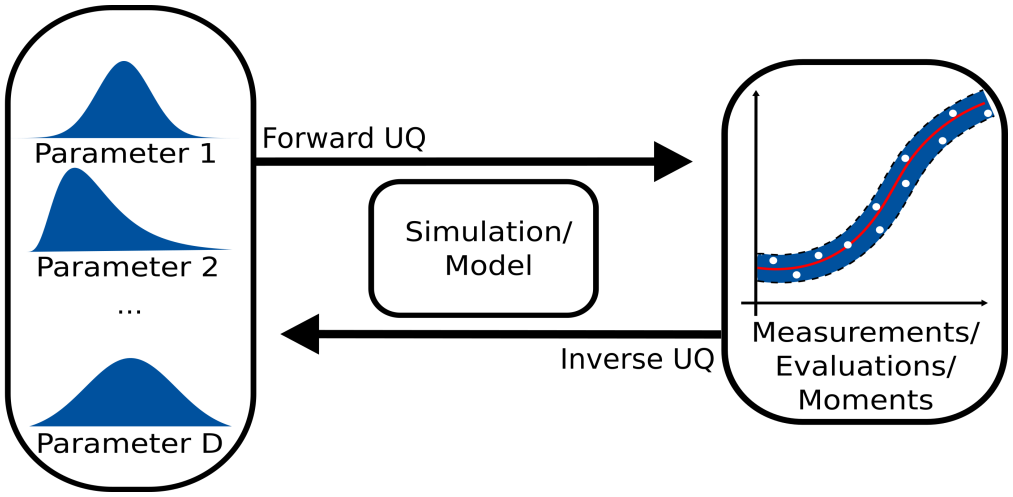


FIGURE 4.1 Abstract illustration of uncertainty quantification. Uncertain parameters and their according distributions can be propagated through a model to obtain a stochastic description of the outcome (forward UQ). Conversely, parameter distributions can be estimated from measurements or evaluations (inverse UQ).

values. Rather than specific input-output pairs, integral quantities and the overall likelihood of events are considered. In general two approaches are distinguished, which we will discuss now in more detail: forward UQ and inverse UQ. Both are illustrated in Fig. 4.1.

Forward UQ. Let $f : \Omega \subseteq \mathbb{R}^D \to \mathbb{R}^m$ be a black-box function, which maps input parameters to a scalar or vector-valued output. We will refer to f as the model, the simulation, or the objective function contextually. In forward UQ, each input parameter is assigned a probability density function, which summarizes the according uncertainty from various sources. The distributions are propagated through the model, which results in stochastic metrics describing the effect of the input uncertainties on response metrics of interest. Of particular interest are stochastic moments, which describe the expected outcome and deviation thereof. We perform forward UQ through stochastic collocation, which we introduce in the next section.

Inverse UQ. Inverse UQ is concerned with determining causal factors from observed results. We want to determine unknown model input parameters $\boldsymbol{x}$ based on data comprised of observed model outputs $Y = f(\boldsymbol{x})$. This task is often complicated by noise which overlays the observations, i.e., $Y' = f(\boldsymbol{x}) + \varepsilon$ for a noise term ε. Therefore, instead of finding an explicit solution, it makes more sense to find the parameter values which are most likely to have resulted in the given observations Y'. The most popular approach for

inverse UQ is Bayesian inference, which we introduce in Sec. 4.3 and apply in Chap. 6.

4.2 Expansion Methods

We now present the two most popular approaches for forward uncertainty propagation based on surrogates, stochastic collocation and polynomial chaos expansion.

4.2.1 Stochastic Collocation

Let $f : \Omega \subseteq \mathbb{R}^D \to \mathbb{R}^m$ be the objective function. *Stochastic collocation* describes the process of replacing it by a surrogate $\tilde{f}$, and approximating the stochastic moments of f through the moments of $\tilde{f}$. In this thesis we create surrogates with B-splines and sparse grids,

$$f \approx \tilde{f} := \sum_{i \sim I} \alpha_i b_i. \tag{4.14}$$

From this we approximate the mean and variance using Gauss-Legendre quadrature,

$$\mathbb{E}[f] \approx \mathbb{E}[\tilde{f}] = \int_\Omega \tilde{f}(\boldsymbol{x}) \varrho_f(\boldsymbol{x}) d\boldsymbol{x} \approx \sum_k \tilde{f}(\boldsymbol{x}_k) \varrho_f(\boldsymbol{x}_k) \omega_k, \tag{4.15}$$

$$\mathbb{V}[f] \approx \mathbb{V}[\tilde{f}] = \mathbb{E}[\tilde{f}^2] - \mathbb{E}[\tilde{f}]^2, \tag{4.16}$$

where $\boldsymbol{x}_k$ are the points and ω_k the weights of the quadrature rule. B-splines are piecewise polynomials of degree n and thus exactly integrable using quadrature rules of appropriate order, i.e. $\lceil \frac{n+1}{2} \rceil$, on each knot segment. Depending on the distribution ϱ_f this quadrature order may not suffice and must be increased appropriately.

4.2.2 Polynomial Chaos Expansion

Polynomial Chaos Expansion (PCE) is one of the most popular techniques in UQ for the creation of surrogates and the calculation of stochastic moments. The underlying idea of projecting a full-complexity model onto orthogonal polynomial basis functions was originally established by Wiener [Wie38]. We now introduce PCE briefly and later on compare our surrogates based on B-splines, sparse grids, and stochastic collocation to this technique. We describe generalized PCE based on the Wiener-Askey scheme [Xiu02; Eld09b], which is the most common PCE approach. The scheme can be seen in Tab. 4.1. It shows polynomial bases which are optimal for standard distributions. Optimal here means orthogonal with respect to the weight function, which differs from the density by a constant factor that normalizes it to unit integral. If other distribution types are required, nonlinear variable transformations like Rosenblatt [Ros52] and Nataf [Der86] can be applied,

Distribution	Density function	Polynomial Basis	Weight function	Support
Normal	$\frac{1}{\sqrt{2\pi}}e^{\frac{-x^2}{2}}$	Hermite $He_n(x)$	$e^{\frac{-x^2}{2}}$	$(-\infty,\infty)$
Uniform	$\frac{1}{2}$	Legendre $P_n(x)$	1	$[-1,1]$
Beta	$\frac{(1-x)^\alpha(1+x)^\beta}{2^{\alpha+\beta+1}B(\alpha+1,\beta+1)}$	Jacobi $P_n^{(\alpha,\beta)}(x)$	$(1-x)^\alpha(1+x)^\beta$	$[-1,1]$
Exponential	e^{-x}	Laguerre $L_n(x)$	e^{-x}	$[0,\infty)$
Gamma	$\frac{x^\alpha e^{-x}}{\Gamma(\alpha+1)}$	gen. Laguerre $L_n^{(\alpha)}(x)$	$x^\alpha e^{-x}$	$[0,\infty)$

TABLE 4.1 Connection between standard continuous probability density functions and the Wiener-Askey scheme of continuous polynomial bases which are orthogonal with respect to the corresponding weight function [Eld09b].

but convergence rates are typically decreased by this [Eld09b]. Alternatively orthogonal polynomials can be generated numerically to match the given distributions [Eld09a].

The orthogonal polynomial basis functions from the Wiener-Askey scheme are used to approximate the functional form between random inputs and stochastic outputs as a series expansion. This expansion takes the form

$$f(\boldsymbol{X}) = \gamma_0\Phi_0 + \sum_{i_1=1}^{\infty}\gamma_{i_1}\Phi_1(\boldsymbol{X}_{i_1}) + \sum_{i_1=1}^{\infty}\sum_{i_2=1}^{i_1}\gamma_{i_1,i_2}\Phi_2(\boldsymbol{X}_{i_1},\boldsymbol{X}_{i_2}) + \ldots, \tag{4.17}$$

where the dimension of the random vector is unbounded and each additional set of nested summations adds an additional order of polynomials to the expansion. The order-based indexing can be replaced by term-based indexing to simplify the representation [Eld09b], resulting in

$$f(\boldsymbol{X}) = \sum_{k=0}^{\infty}\gamma_k\Psi_k(\boldsymbol{X}), \tag{4.18}$$

with a direct correspondence between $\gamma_{i_1,i_2,\ldots}$ and γ_k, and between $\Phi_j(\boldsymbol{X}_{i_1},\boldsymbol{X}_{i_2},\ldots)$ and $\Psi_k(\boldsymbol{X})$, see [Eld09b] for more details. The PCE coefficients γ_k are often calculated using spectral projection,

$$\gamma_k = \frac{\langle f,\Psi_k\rangle_{\varrho_f}}{\langle\Psi_k,\Psi_k\rangle_{\varrho_f}} = \frac{1}{\langle\Psi_k,\Psi_k\rangle_{\varrho_f}}\int_\Omega f\Psi_k\varrho_f d\boldsymbol{x}, \tag{4.19}$$

where $\langle\cdot,\cdot\rangle_{\varrho_f}$ is the scalar product with respect to the density ϱ_f. The spectral projection in Eq. (4.19) requires the calculation of high-dimensional integrals. For this task often sparse grids in form of the combination technique are applied [Eld09b]. Alternatively,

the PCE coefficients can be calculated via regression.

A major advantage of PCE is, that once the expansion has been calculated, stochastic moments can be extracted without further costly integration. Because of the orthonormality of the polynomial basis functions, because $\mathbb{E}[g \cdot h] = \langle g, h \rangle_{\varrho_f}$, and because $\Psi_0 \equiv 1$ it holds

$$\mathbb{E}[f] = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma_k \Psi_k\right] = \sum_{k=0}^{\infty} \gamma_k \mathbb{E}[1 \cdot \Psi_k] = \sum_{k=0}^{\infty} \gamma_k \underbrace{\mathbb{E}[\Psi_0 \cdot \Psi_k]}_{=\delta_{0,k}} = \gamma_0. \tag{4.20}$$

Using this, we derive a simplified formula for the variance of PCE as well,

$$\begin{aligned} \mathbb{V}[f] &= \mathbb{E}\left[(f - \mathbb{E}[f])^2\right] = \mathbb{E}\left[\left(\sum_{k=0}^{\infty} \gamma_k \Psi_k - \gamma_0\right)^2\right] = \mathbb{E}\left[\left(\sum_{k=1}^{\infty} \gamma_k \Psi_k\right)^2\right] \\ &= \sum_{k=1}^{\infty} \gamma_k^2 \underbrace{\mathbb{E}[\Psi_k^2]}_{=1} = \sum_{k=1}^{\infty} \gamma_k^2. \end{aligned} \tag{4.21}$$

In practice the infinite expansion (4.18) must be truncated. Traditionally polynomials up to a specific total order are used. It follows then from Eq. (4.21) that truncated PCE must underestimate the variance.

4.2.3 Numerical Comparison

We now compare PCE and stochastic collocation using a benchmark for UQ approximation methods. The benchmark is a real-world application, modeled by the U.S. office of nuclear waste isolation [Har83]. A borehole is drilled through an aquifer above a nuclear waste repository, through this repository and into an aquifer below. The input parameters describing this scenario and their distributions are defined in Tab. 4.2. The response $Q \in \mathbb{R}$ is the flow in m^3/yr,

$$Q = \frac{2\pi T_u (H_u - H_\ell)}{\ln(r/r_w)\left(1 + \frac{2LT_u}{\ln(r/r_w) r_w^2 K_w} + \frac{T_u}{T_\ell}\right)}. \tag{4.22}$$

On the basis of this benchmark, we now compare our approach based on sparse grids, B-splines, and stochastic collocation to PCE. We use the DAKOTA library [Dak14], an industry standard for UQ computations, to perform the PCE. We also add the results of simple Monte Carlo quadrature to the comparison, to demonstrate the urgent need for more advanced UQ methods. The objective function is smooth, therefore we use degree $n = 5$ for all B-spline bases. A first comparison has already been published in [Reh21b].

Name	Description	Distribution
r_w	radius of borehole (m)	$N(\mu = 0.1, \sigma = 0.0161812)$
r	radius of influence (m)	$\log N(\mu = 7.71, \sigma = 1.0056)$
T_u	upper aquifer transmissivity (m^2/yr)	$U[63070, 115600]$
H_u	upper aquifer potentiometric head (m)	$U[990, 1110]$
T_ℓ	lower aquifer transmissivity (m^2/yr)	$U[63.1, 116]$
H_ℓ	lower aquifer potentiometric head (m)	$U[700, 820]$
L	borehole length (m)	$U[1120, 1680]$
K_w	borehole hydraulic conductivity (m/yr)	$U[9855, 12045]$

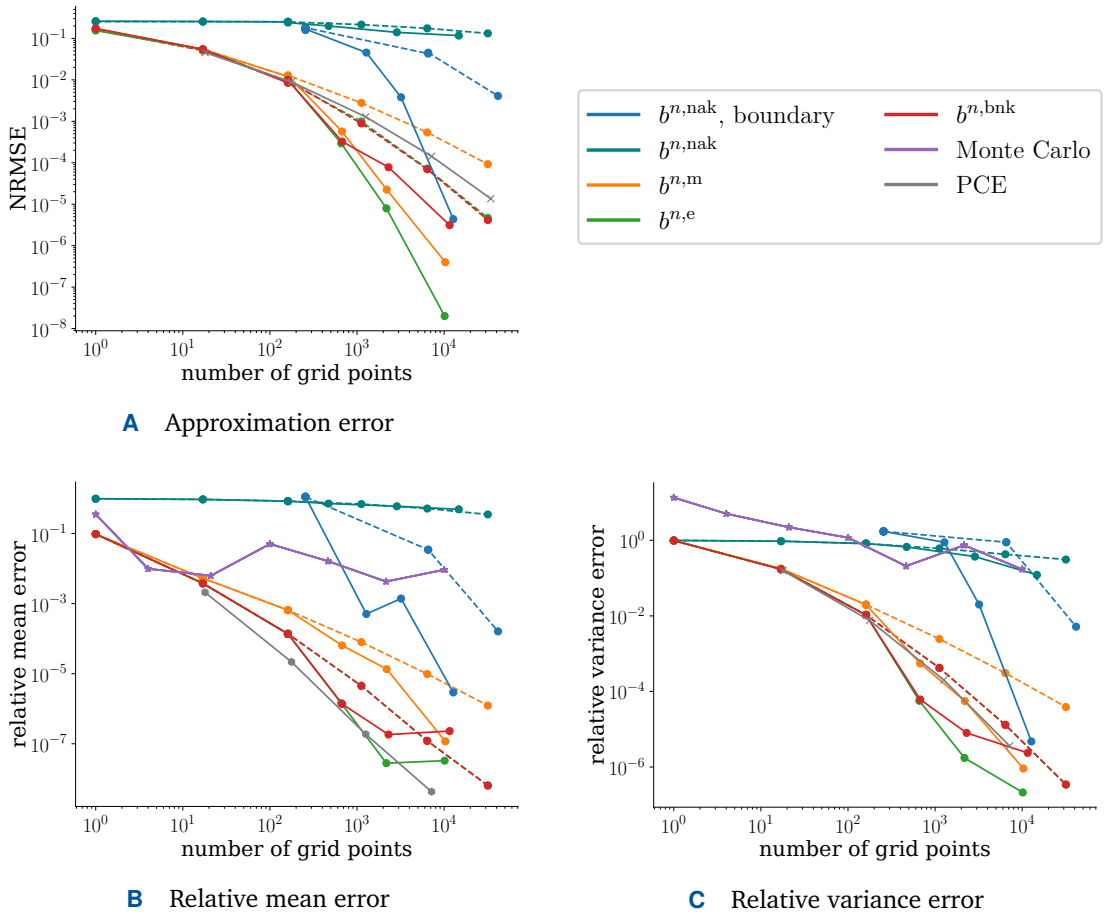
TABLE 4.2 The eight input parameters and according distributions for the borehole model Eq. (4.22) from [Har83].

Now we add boundaryless not-a-knot B-splines, which were not yet developed at the time of publication. On regular grids they perform exactly like the extended splines, as is to be expected. On adaptive grids they behave slightly different, because their surpluses differ and thus different grid points are chosen for refinement. For mean and variance we use a reference solution computed with extended not-a-knot B-splines of degree 5 on a spatially adaptive sparse grid of 35,000 grid points. We verify this reference solution by calculating a similar one using DAKOTA's PCE based on a sparse grid of level 5 with 34,290 grid points. The difference between both results for mean and variance is smaller than 10^{-11}, respectively.

Looking at the NRMSE in Fig. 4.2A, we see that extended B-splines and the boundaryless not-a-knot B-splines perform best among the regular grids. In particular, they outperform PCE. The boundary not-a-knot B-splines on regular grids are far behind, because they cannot adequately resolve the inner domain of an 8-dimensional space. All sparse grid methods significantly improve when using spatial adaptivity, and extended B-splines on spatially adaptive sparse grids perform best among the studied methods. Note that not-a-knot B-splines without boundary points or any boundary treatment do not converge at all, which demonstrates once again the urgent need for compensation when omitting the boundary points.

B-splines are numerically exact integrable, and no additional error is introduced when performing stochastic collocation. Therefore, the errors for the stochastic quantities look similar to the NRMSE. The polynomial chaos expansion slightly outperforms the other methods for the approximation of the mean in Fig. 4.2B. This is to be expected, as the mean approximation does not depend on the whole expansion, but only its first coefficient, cf. Eq. (4.20).

Figure 4.2C shows that for the variance approximation, B-splines on adaptive sparse



A Approximation error

B Relative mean error

C Relative variance error

FIGURE 4.2 Errors for the borehole model (4.22) with not-a-knot B-splines ($b^{n,\text{nak}}$) with and without boundary points, modified ($b^{n,\text{m}}$), extended ($b^{n,\text{e}}$), and boundaryless not-a-knot B-splines ($b^{n,\text{bnk}}$) of degree $n = 5$. Regular sparse grids are represented with dashed lines, adaptive sparse grids with solid lines. For comparison results from PCE and Monte Carlo are included.

grids outperform PCE, because PCE by definition underestimates this quantity, as we explained following Eq. (4.21). The Monte Carlo results are several orders of magnitudes worse than the results for the more advanced techniques, which confirms the need for the latter.

4.3 Bayesian Inference

Parameter calibration is the task of identifying the parameters of a model $f : \mathbb{R}^D \to \mathbb{R}$ such that the output resembles a real process that is captured in form of experimental measurements or observations. The real process is often expensive in terms of required materials

and execution time, therefore only few measurements $Y = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_M\}$, $M \in \mathbb{N}$ are available. Additionally, the measurements are usually perturbed by noise, which makes the parameter calibration problem ill-posed in the sense of Hadamard, meaning that neither existence, nor uniqueness, nor stability of the solution can be guaranteed [Had02].

One method to find a set of parameters fitting the observations is *regression*. The idea is to find the parameter configuration $\boldsymbol{x}^* \in \mathbb{R}^D$, which minimizes the difference between the data Y and the model evaluation $f(\boldsymbol{x}^*)$. To make the problem well-posed and inhibit noise influence, a regularization term can be added. A popular example for this approach is the least squares formulation with a Tikhonov regularization term C [Tik77],

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} \left(\frac{1}{2} \|Y - f(\boldsymbol{x})\|_2^2 + \lambda C(\boldsymbol{x}) \right). \tag{4.23}$$

However, the quality of regression approaches heavily depends on the choice of regularization, and the result is only a single parameter configuration $\boldsymbol{x}^*$ with no information about the remaining parameter space. Furthermore, depending on the structure of the problem, it might be difficult to avoid local minima.

Therefore, another more informative approach is a central aspect of UQ, *Bayesian inference*. Instead of proposing only a single solution $\boldsymbol{x}^*$, every feasible parameter combination is assigned the relative probability of matching the observations. This information is summarized in a probability distribution. Even the measurement noise can be comprised in the Bayesian formulation, and then be inferred along, resulting in another distribution describing the noise behavior. The general idea of Bayesian inference is that starting from a prior belief, additional information or new observations are incorporated, and the belief is updated according to Bayes' rule, eventually resulting in the posterior distribution. Formally, we define the conditional probability $P(A \mid B)$, that is, the likelihood of event A occurring given that B is true, as

$$P(A \mid B) := \frac{P(A \cap B)}{P(B)}, \tag{4.24}$$

where we assume that $P(B) \neq 0$. Let now $\boldsymbol{X}$ be a random variable describing the parameters of the model f. We want to compute a distribution describing how likely $\boldsymbol{X}$ is to take a value $\boldsymbol{x}$, given observations Y. This is called the *posterior distribution* $\pi(\boldsymbol{X} = \boldsymbol{x} \mid Y)$. Any information that we have a priori can be incorporated in the *prior distribution* $\pi(\boldsymbol{X})$, which is an initial guess for the unknown posterior distribution. The *likelihood* $\pi(Y \mid \boldsymbol{X} = \boldsymbol{x})$ returns the plausibility of the measurements Y given parameter $\boldsymbol{x}$. Bayes formula brings

together these terms [Bay63],

$$\pi(\boldsymbol{X} \mid Y) = \frac{\pi(Y \mid \boldsymbol{X})\ \pi(\boldsymbol{X})}{\pi(Y)}, \tag{4.25}$$

where $\pi(Y) := \int \pi(\boldsymbol{X})\pi(Y \mid \boldsymbol{X}) d\boldsymbol{x}$ is the *marginal likelihood*, a normalization factor which is usually ignored in practice, because it is constant with respect to $\boldsymbol{x}$.

Hierarchical modeling. The full power of Bayesian inference is achieved by using hierarchical models [Gel06b; Gel13]. Assume we have a set of experiments with outcomes $\boldsymbol{y}_j$, $j = 1, \dots, M$. Let further $\boldsymbol{X}^j = (\boldsymbol{X}_1^j, \dots, \boldsymbol{X}_D^j) \in \Omega \subseteq \mathbb{R}^D$ be the parameter combinations for each outcome $\boldsymbol{y}_j$, with likelihood $\pi(\boldsymbol{y}_j \mid \boldsymbol{X}^j)$. If the parameter combinations $\boldsymbol{X}^j$ are exchangeable, that is, if the joint distribution is independent of their ordering, then each observation $\boldsymbol{y}_j$ can be regarded as outcome of the same experiment, based on the respective realization of an underlying joint probability model. Thus, we model the distribution of each parameter $\boldsymbol{X}_d$ conditionally on hyper-parameters $\boldsymbol{\theta}$, which themselves are given a probabilistic specification. The hyper-parameters $\boldsymbol{\theta}$ then describe a family of distributions, and each parameter combination $\boldsymbol{X}^j$ is an instance of such a distribution. Because the underlying distribution is completely described by the hyper-parameters, it suffices to infer $\pi(\boldsymbol{\theta} \mid Y)$ to obtain $\pi(\boldsymbol{X} \mid Y)$. We make this more clear with a quick example.

Let $f : \mathbb{R} \to \mathbb{R}$ depend only on a single parameter x, and let $\boldsymbol{y}_1 = f(x^1)$ and $\boldsymbol{y}_2 = f(x^2)$ be two experimental outcomes for unknown realizations x^1 of $\boldsymbol{X}^1$ and x^2 of $\boldsymbol{X}^2$. In simple Bayesian modeling, one would infer one distribution for each parameter $\boldsymbol{X}^1$ and $\boldsymbol{X}^2$ respectively. In hierarchical Bayesian modeling in contrast, one would assume that $\boldsymbol{X}^1$ and $\boldsymbol{X}^2$ are both instances of a common underlying probability model. For example, one could assume that they are both normally distributed, but with slightly different means μ_1 and μ_2, and standard deviations σ_1 and σ_2. Consequently, we assume that there are underlying distributions μ and σ such that μ_1, μ_2 are realizations of μ, and σ_1, σ_2 are realizations of σ. For example μ and σ could be uniformly distributed, $\mu \sim U[\theta_1, \theta_2]$, $\sigma \sim U[\theta_3, \theta_4]$ with hyper-parameters $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3, \theta_4) \in \mathbb{R}^4$.

Markov Chain Monte Carlo. In general the posterior distribution $\pi(\boldsymbol{X} \mid Y)$ cannot be evaluated directly and must be approximated numerically based on samples. Following Eq. (4.25) it holds

$$\pi(\boldsymbol{X} \mid Y) \propto \pi(Y \mid \boldsymbol{X})\pi(\boldsymbol{X}). \tag{4.26}$$

The right-hand side of Eq. (4.26) can be evaluated point-wise for any $\boldsymbol{x} \in \Omega$. Making use of this we generate a set of samples that is distributed following $\pi(\boldsymbol{X} \mid Y)$ to estimate the

density numerically. However, sampling soon becomes very expensive, in particular for high dimensional parameter spaces, where often the distribution is concentrated locally, and therefore negligibly small in most parts of the space. For efficient sampling, the next sample point must be determined iteratively based on the present set of samples, such that it contributes much new information.

One of the most widely used strategies for sampling distributions is Markov Chain Monte Carlo (MCMC) [Bro11]. A *Markov chain* $(\boldsymbol{x}^1, \boldsymbol{x}^2, \dots)$ is a progression of points in the parameter space, generated by repeatedly applying a random map, the *Markov transition*. For the number of samples going to infinity, the Markov chain samples converge towards the target distribution [Ros95]. MCMC samples are typically autocorrelated, i.e., consecutive samples are close to each other and do not provide independent information. Therefore, the quality of $C \in \mathbb{N}$ chains of $N \in \mathbb{N}$ samples each is not measured in the number of total samples CN, but with the *effective sample size* $N_{\text{eff}} \in \mathbb{N}$. The effective sample size quantifies the number of exact independent samples from the target distribution, which provide the same estimation power as the N autocorrelated samples. It is defined as [Gel13]

$$N_{\text{eff}} \coloneqq \frac{CN}{1 + 2\sum_{t=1}^{\infty} \rho_t}, \tag{4.27}$$

where ρ_t is the lag-t autocorrelation over the history of the Markov chain. This is the correlation between two chains offset by t positions (i.e., a "lag" in time-series terminology). The lag-t autocorrelation can usually not be quantified analytically, but it can be estimated from the drawn samples [Gel13]. In practice, sampling libraries provide an estimate for N_{eff} [Sal16].

Metropolis-Hastings. For applying MCMC, it remains to define the Markov transition. One of the most popular approaches is the Metropolis-Hastings (MH) algorithm [Met53; Has70], which consists of two steps: The *proposal*, which is a stochastic perturbation of the current state, and the *correction*, which rejects bad proposals. Let $g(\boldsymbol{x} \mid \boldsymbol{x}^t)$ be an arbitrary initial density that suggests the next sample, based on the current sample $\boldsymbol{x}^t$. Starting from an initial point $\boldsymbol{x}^1$, iteratively a new proposal $\boldsymbol{x}^*$ is generated by sampling from the jumping distribution at time t, $g(\boldsymbol{x}^* \mid \boldsymbol{x}^t)$. Then the ratio of densities is calculated,

$$r \coloneqq \frac{\pi(\boldsymbol{x}^* \mid Y)}{\pi(\boldsymbol{x}^t \mid Y)}, \tag{4.28}$$

and the next sample is set to be

$$\boldsymbol{x}^{t+1} = \begin{cases} \boldsymbol{x}^* & \text{with probability } \min(r, 1), \\ \boldsymbol{x}^t & \text{otherwise.} \end{cases} \tag{4.29}$$

Often a Gaussian distribution is used for the initial density g, making the Markov chain a random walk.

The Metropolis-Hastings algorithm is easy to understand and implement, therefore it is still widely applied. However, it scales bad with dimensionality and complexity of the target distribution [Bet17]. To efficiently sample the distribution, large jumps into unexplored regions of high probability mass are necessary. However, for simple random jumping, almost all proposals will be neglected. Additionally, the samples are correlated by construction, thus the effective sample size can be significantly smaller than the actual number of samples. To make up for this, at least partially, often the first samples are disregarded and considered as *tuning samples* or *burn-in*, which narrows down the region of interest.

Still, jump proposals exploiting the geometry of the distribution and following contours of high probability result in a much more efficient sampling than the Metropolis-Hastings algorithm. One strategy, which accomplishes this favorable goal is Hamiltonian Monte Carlo (HMC).

Hamiltonian Monte Carlo and NUTS. Hamiltonian Monte Carlo is an MCMC technique for efficiently sampling probability distributions. It was originally developed in 1987 [Dua87], but only recently has become known to a broad audience [Nea11; Bet17], when high-performance implementations became available [Sal16; Car17]. The idea of HMC is to construct a vector field aligned with the target probability density function by using its gradients. This is done using differential geometry, which classically is used for the description of the physics of Hamiltonian mechanics, e.g., gravitation and orbits. Following the Hamiltonian vector field, the sampler rapidly moves through the parameter space, while being constrained to relevant regions. Several parameters need to be tuned, such that this can be done efficiently. In particular, it must be decided how long to follow the trajectories. The No-U-Turn Sampler (NUTS) is an implementation of HMC, which jumps to the next trajectory after it followed the current one for a semicircle, and starts to go back towards already explored space [Hof14].

In comparison to the classical Metropolis-Hastings algorithm using a Gaussian random walk proposal strategy, HMC significantly reduces the autocorrelation between its samples. Exploiting the gradient information, HMC maintains a high probability of samples being accepted event though making large jumps through the parameter space. In

fact, for NUTS the effective sample size N_{eff} can even be larger than the actual number of samples N, if the lag-t autocorrelation in Eq. (4.27) becomes negative [Sta18]. This is caused by NUTS simulating the Hamiltonian forward and backward in time at random, doubling the time at each iteration, which potentially leads to antithetical sampling.

HMC pays for its efficient sampling with the requirement for gradients of the posterior. Commonly, automatic differentiation is used to evaluate these gradients [Mar19], but if the objective is a black-box function, then this approach is not feasible. In Chap. 6 we use B-splines on sparse grids to approximate a black box simulation and then estimate the parameters matching real-world measurements using NUTS based on the B-spline surrogate's derivatives. For this we use the NUTS implementation of the pymc3 library [Sal16]. Further information on Hamiltonian Monte Carlo and NUTS can be found in [Nea11; Bet17].

4.4 Active Subspaces

An important part of UQ modeling is *sensitivity analysis* [Chr02; Sal08], which is "the study of how uncertainty in the output of a model can be apportioned to different sources of uncertainty in the model input" [Sal04]. It allows for better understanding, testing, and improving models [Pan97]. Sensitivity analysis identifies inputs that cause significant uncertainty in the output, and thus points out where further research or a reparametrization is necessary. Conversely, the analysis also identifies insignificant parameters which have almost no influence on the quantity of interest. Such parameters can be omitted to reduce the dimension, and thus the overall complexity of the model. In this section, we present the beneficial combination of active subspaces, an emerging sensitivity analysis tool used for dimension reduction, and B-splines on sparse grids.

So far, in the context of sparse grids, sensitivity analysis has mainly been performed through Analysis of Variance (ANOVA) [Gri05; Feu10]. ANOVA is based on the concept of disassembling a function into individual terms, each of which only depends on a subset of the input parameters,

$$f(x_1,\dots,x_D) = f_0 + \sum_{j_1} f_{j_1}(x_{j_1}) + \sum_{j_1<j_2} f_{j_1,j_2}(x_{j_1},x_{j_2}) + \cdots + f_{1,\dots,D}(x_1,\dots,x_D). \tag{4.30}$$

The first component f_0 is constant and represents the function's average, the following components depend on an increasing number of parameters, where the one-dimensional components f_{j_1} represent the variances with respect to the corresponding dimension, and so forth, until the last component $f_{1,\dots,D}$, which is fully D-dimensional. If the terms of the decomposition (4.30) decay rapidly, then only superpositions of low-dimensional

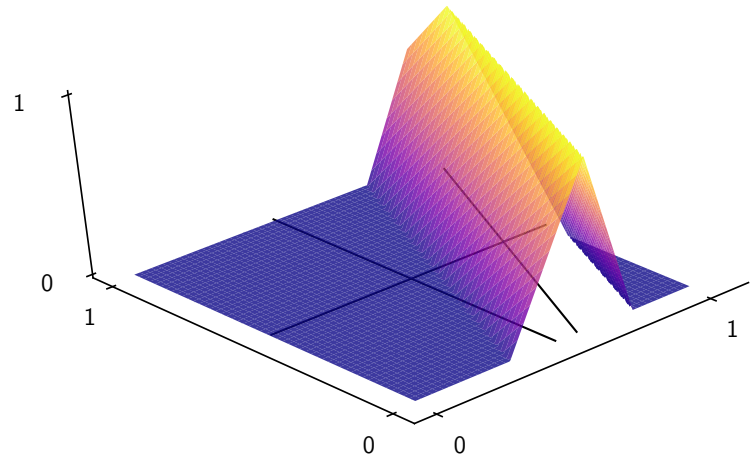


FIGURE 4.3 The function f (4.31) is a one-dimensional hat function, rotated by about 27 degrees. ANOVA fails to find the intrinsic one-dimensional structure because it is not axis-parallel (indicated with black lines).

functions govern the essential features. Thus, high-dimensional terms can be omitted without losing significant information.

ANOVA works well, if all essential features of the function f are axis-parallel. However, non-axis-parallel features can not be found by ANOVA. The following example function,

$$f(x_1, x_2) := \max(1 - |5y - 2.5|, 0), \quad y = x_1 \cos(0.15\pi) - x_2 \sin(0.15\pi), \tag{4.31}$$

is a one-dimensional hat function rotated by about 27 degrees, illustrated in Fig. 4.3. Because of the rotation, ANOVA fails to find the simple one-dimensional structure [Sch19]. In contrast to ANOVA, the concept of active subspaces is well suited for the detection of such features. Active subspaces provide a new basis for the parameter space based on estimating the directions of most expected change, accordingly rotating the domain and assigning each rotated direction a relevance estimate. We now introduce active subspaces following Constantine [Con15a].

Let $f : [0,1]^D \to \mathbb{R}$ be differentiable and Lipschitz continuous, where the restriction to the unit hypercube is without loss of generality. The input parameters $\boldsymbol{X} = (\boldsymbol{X}_1, \ldots, \boldsymbol{X}_D)$ are random variables and have an associated probability density $\varrho_{\boldsymbol{X}} \colon \Omega \subseteq \mathbb{R}^D \to \mathbb{R}$. Lipschitz continuity implies that the gradient's norm is bounded by a constant $L \in \mathbb{R}$,

$$\|\nabla_{\boldsymbol{X}} f(\boldsymbol{x})\|_2 \leq L \text{ for all } \boldsymbol{x} \in [0,1]^D. \tag{4.32}$$

Let $\boldsymbol{C} \in \mathbb{R}^{D\times D}$ be the average of the outer product of the function's gradient with itself,

$$\boldsymbol{C} := \int_{[0,1]^D} (\nabla_{\boldsymbol{x}} f)(\nabla_{\boldsymbol{x}} f)^T \varrho_{\boldsymbol{X}} d\boldsymbol{x}. \tag{4.33}$$

This means that each entry of $\boldsymbol{C} = (C_{i,j})_{i,j=1}^D$ is the average of the product of partial derivatives,

$$C_{i,j} := \int_{[0,1]^D} \frac{\partial f}{\partial x_i}\frac{\partial f}{\partial x_j} \varrho_{\boldsymbol{X}} d\boldsymbol{x}, \quad i,j = 1,\dots,D. \tag{4.34}$$

The matrix $\boldsymbol{C}$ is positive semidefinite,

$$\boldsymbol{v}^T \boldsymbol{C} \boldsymbol{v} = \int_{[0,1]^D} (\boldsymbol{v}\nabla_{\boldsymbol{x}} f)^2 \varrho_{\boldsymbol{X}} d\boldsymbol{x} \geq 0 \text{ for all } \boldsymbol{v} \in \mathbb{R}^D, \tag{4.35}$$

and since $\boldsymbol{C}$ is also symmetric, it has a real eigenvalue decomposition,

$$\boldsymbol{C} = \boldsymbol{W}\boldsymbol{\Lambda}\boldsymbol{W}^T, \quad \boldsymbol{\Lambda} = \operatorname{diag}(\lambda_1,\dots,\lambda_D), \quad \lambda_1 \geq \cdots \geq \lambda_D \geq 0, \tag{4.36}$$

where $\boldsymbol{W} \in \mathbb{R}^{D\times D}$ is the orthogonal matrix of normalized eigenvectors $\boldsymbol{w}_1,\dots,\boldsymbol{w}_D$. Without loss of generality, we assume $\boldsymbol{\Lambda}$ and $\boldsymbol{W}$ to be sorted decreasingly according to the eigenvalues $\lambda_1,\dots,\lambda_D$. The following lemma describes the relationship of the mean squared directional derivatives and the eigendecomposition of $\boldsymbol{C}$.

LEMMA 4.1 (mean squared derivative and eigendecomposition)
The mean squared directional derivative of f with respect to the eigenvector $\boldsymbol{w}_k$ is equal to the corresponding eigenvalue,

$$\mathbb{E}\left[\left((\nabla_{\boldsymbol{x}} f)^T \boldsymbol{w}_k\right)^2\right] = \int_{[0,1]^D} \left((\nabla_{\boldsymbol{x}} f)^T \boldsymbol{w}_k\right)^2 \varrho_{\boldsymbol{X}} d\boldsymbol{x} = \lambda_k, \quad i = 1,\dots,D. \tag{4.37}$$

PROOF By the definition of the matrix $\boldsymbol{C}$ in Eq. (4.33),

$$\lambda_k = \boldsymbol{w}_k^T \boldsymbol{C} \boldsymbol{w}_k = \boldsymbol{w}_k^T \left(\int_{[0,1]^D} (\nabla_{\boldsymbol{x}} f)(\nabla_{\boldsymbol{x}} f)^T \varrho_{\boldsymbol{X}} d\boldsymbol{x}\right) \boldsymbol{w}_k = \int_{[0,1]^D} \left((\nabla_{\boldsymbol{x}} f)^T \boldsymbol{w}_k\right)^2 \varrho_{\boldsymbol{X}} d\boldsymbol{x}. \tag{4.38}$$

■

The eigenvectors in $\boldsymbol{W}$ define a new basis through a rotation of the domain of f. It follows from Lm. 4.1, that if $\lambda_k = 0$ for some $k \in \{1,\dots,D\}$, then f is constant

along the direction $\boldsymbol{w}_k$. Using the rotated D-dimensional basis $\boldsymbol{W}$, no information is lost when omitting $\boldsymbol{w}_k$. This property motivates the term inactive subspace, introduced by Russi [Rus10], for directions $\boldsymbol{w}_k$ with no or only small average change. We now formalize this by separating the eigenvalues and eigenvectors. Let $1 \leq t < D$,

$$\boldsymbol{\Lambda} = \begin{pmatrix} \boldsymbol{\Lambda}_1 & \\ & \boldsymbol{\Lambda}_2 \end{pmatrix}, \quad \boldsymbol{\Lambda}_1 := \operatorname{diag}(\lambda_1, \dots, \lambda_t), \quad \boldsymbol{\Lambda}_2 := \operatorname{diag}(\lambda_{t+1}, \dots, \lambda_D), \tag{4.39}$$

$$\boldsymbol{W} = (\boldsymbol{W}_1 \quad \boldsymbol{W}_2), \quad \boldsymbol{W}_1 := (\boldsymbol{w}_1, \dots, \boldsymbol{w}_t), \quad \boldsymbol{W}_2 := (\boldsymbol{w}_{t+1}, \dots, \boldsymbol{w}_D). \tag{4.40}$$

Next, we define new variables $\boldsymbol{X}^1$ and $\boldsymbol{X}^2$,

$$\boldsymbol{X}^1 := \boldsymbol{W}_1^T \boldsymbol{X}, \quad \boldsymbol{X}^2 := \boldsymbol{W}_2^T \boldsymbol{X}. \tag{4.41}$$

Any input $\boldsymbol{X}$ can now be expressed as a combination of $\boldsymbol{X}^1$ and $\boldsymbol{X}^2$,

$$\boldsymbol{X} = \underbrace{\boldsymbol{W}\boldsymbol{W}^T}_{Id}\boldsymbol{X} = \boldsymbol{W}_1\boldsymbol{W}_1^T\boldsymbol{X} + \boldsymbol{W}_2\boldsymbol{W}_2^T\boldsymbol{X} = \boldsymbol{W}_1\boldsymbol{X}^1 + \boldsymbol{W}_2\boldsymbol{X}^2, \tag{4.42}$$

because $\boldsymbol{W}$ is orthogonal. With this we extend Lm. 4.1.

LEMMA 4.2 (decomposition into active and inactive subspaces)
The mean squared gradients of f with respect to $\boldsymbol{X}^1$ and $\boldsymbol{X}^2$ satisfy

$$\int_{[0,1]^D} (\nabla_{\boldsymbol{X}^1} f)^T (\nabla_{\boldsymbol{X}^1} f) \varrho_{\boldsymbol{X}} d\boldsymbol{x} = \lambda_i + \dots + \lambda_t, \tag{4.43}$$

$$\int_{[0,1]^D} (\nabla_{\boldsymbol{X}^2} f)^T (\nabla_{\boldsymbol{X}^2} f) \varrho_{\boldsymbol{X}} d\boldsymbol{x} = \lambda_{t+1} + \dots + \lambda_D. \tag{4.44}$$

PROOF See [Con15a]. ∎

It follows from Lm. 4.2 that, on average, small perturbations of $\boldsymbol{X}^1$ change f more than small perturbations of $\boldsymbol{X}^2$. How much more is determined by the eigenvalues in $\boldsymbol{\Lambda}_1$ and in $\boldsymbol{\Lambda}_2$. If there is a significant drop between the values λ_t and λ_{t+1}, then the inputs $\boldsymbol{X}^2$ can be omitted, reducing the dimension, but losing only little information. With this motivation, $\boldsymbol{X}^1$ are called the *active variables* and $\boldsymbol{X}^2$ are called the *inactive variables*. Furthermore, the range of the eigenvectors in $\boldsymbol{W}_1$ is called *active subspace*, and the range of the eigenvectors in $\boldsymbol{W}_2$ is called *inactive subspace*. The method of active subspaces is mainly used for dimension reduction, but the sizes of the eigenvalues $\lambda_1, \dots, \lambda_D$ also provide a sensitivity analysis of the input parameters [Jef15; Con17a]. Error estimates and more details are available in the literature [Con15a]. The focus of this work is the

calculation of the matrix $\boldsymbol{C}$.

Related methods. The eigendecomposition of $\boldsymbol{C}$ is traditionally approximated by using Monte Carlo quadrature with $M \in \mathbb{N}$ samples $\boldsymbol{x}_1, \dots, \boldsymbol{x}_M$ following ϱ_X [Con15a],

$$\boldsymbol{C} \approx \hat{\boldsymbol{C}} = \frac{1}{M} \sum_{i=1}^{M} (\nabla_X f(\boldsymbol{x}_i))(\nabla_X f(\boldsymbol{x}_i))^T = \hat{\boldsymbol{W}} \hat{\boldsymbol{\Lambda}} \hat{\boldsymbol{W}}^T. \tag{4.45}$$

This is equivalent to applying a Singular Value Decomposition (SVD) to the gradient samples [Rus10; Con15a],

$$\frac{1}{\sqrt{D}} (\nabla_X f(\boldsymbol{x}_1), \dots, \nabla_X f(\boldsymbol{x}_M)) = \hat{\boldsymbol{W}} \sqrt{\hat{\boldsymbol{\Lambda}}} \hat{\boldsymbol{V}}. \tag{4.46}$$

A closely related method is Principal Component Analysis (PCA) [Wol87], also called discrete Karhunen-Loève transformation [Sul15]. This method linearly transforms data to a new orthogonal coordinate system, such that the greatest variance is associated with the first coordinate, the second greatest variance with the second coordinate and so forth. Through this, correlated variables are linearly transformed into relatively uncorrelated variables [Flu88]. Just like active subspaces, PCA can be computed using either an eigenvalue decomposition of the covariance matrix, or by using SVD. The main difference is, that PCA seeks a low-dimensional linear parametrization of a random vector, such that its covariance is well-approximated and the resulting parameters are uncorrelated. In contrast, active subspaces help to approximate a scalar valued function of several variables, by identifying directions along which the scalar-valued function changes more on average in terms of its gradient.

Approximating active subspaces. The most expensive part of active subspace detection is the calculation of the matrix $\boldsymbol{C}$. If the gradient of f is explicitly known, or can be sampled, Monte Carlo quadrature can be used to approximate $\boldsymbol{C}$ as indicated in Eq. (4.45). However, Monte Carlo quadrature converges very slowly. Furthermore, computer simulations do in general not provide access to their gradients at all. Therefore, the gradients must be approximated. This has so far been done with linear or quadratic ridge functions [Con15a; Con17b; Gre18], but if f is not of linear or quadratic shape, these methods can only approximate f and ∇f up to a certain degree. We therefore, for the first time, use B-splines and sparse grids to approximate the matrix $\boldsymbol{C}$. Parts of this have already been published in [Reh19]. B-splines can approximate complex functions more accurately than simple linear or quadratic ridge functions. Because B-splines can be differentiated and integrated numerically exact, the accuracy of the approximation is incorporated into

the calculation of the matrix $\boldsymbol{C}$. Additionally, the sparse grid is beneficial for the calculation of the potentially high-dimensional integrals. We demonstrate the improved active subspace approximation for two example functions in Sec. 4.4.2, but first we present a new theoretical result on integration with respect to an active subspace.

4.4.1 One-Dimensional Active Subspaces Integration

We now introduce a new integration algorithm for the special case of one-dimensional active subspaces. The use case might seem limited at first, but in practice one-dimensional active subspaces are commonly found, even in complex simulations. Notable examples include simulations for solar cells [Con15c], lithium-ion batteries [Con17b], or airfoil shapes [Gre18].

Assume $\lambda_1 > 0$ and $\lambda_2 = \cdots = \lambda_D = 0$, then f has an exact one-dimensional active subspace spanned by $\boldsymbol{W}_1 = \boldsymbol{w}_1$, and there exists a function g such that,

$$\begin{aligned} g &: [a,b] \to \mathbb{R}, \\ g(y) &= g(\boldsymbol{W}_1^T \boldsymbol{x}) = f(\boldsymbol{x}), \end{aligned} \tag{4.47}$$

where $\boldsymbol{x} \in [0,1]^D$, and $a := \min_{\boldsymbol{x}} \boldsymbol{W}_1^T \boldsymbol{x}$, $b := \max_{\boldsymbol{x}} \boldsymbol{W}_1^T \boldsymbol{x}$. We approximate g using not-a-knot boundary B-splines on a one-dimensional spatially adaptive sparse grid with level-index set I, resulting in the surrogate $\tilde{g}$,

$$f(\boldsymbol{x}) = g(y) \approx \tilde{g}(y) = \sum_{(\ell,i)\in I} \alpha_{\ell,i} b_{\ell,i}^{n,\text{nak}}(y). \tag{4.48}$$

In general g is unknown, so we need to approximate it from evaluations of f. For every grid point $x_{\ell,i}$, $(\ell,i) \in I$ we solve $\boldsymbol{x}' = \arg\min_{\boldsymbol{x}} \|\boldsymbol{W}_1^T \boldsymbol{x} - x_{\ell,i}\|_2$ and interpolate in the pairs $(x_{\ell,i},\ f(\boldsymbol{x}'))$.

Now we want to integrate $\tilde{g}$ subject to the volume of the inactive subspace. Calculating this $(D-1)$-dimensional volume numerically is an expensive task, only slightly cheaper than the original D-dimensional integral, because the shape of the inactive subspace is that of a zonotope and not a simple hypercube [Con15a]. However, we combine Ramsay's definition of M-splines [Ram88] and Schoenberg's theorem on simplex volumes [Cur66], and show that the volume of the complicated $(D-1)$-dimensional inactive subspace can be formulated exactly as a sum of B-splines [Reh19]. This is achieved by decomposing the domain into simplices and applying the upcoming Thm. 4.4, which formulates the volume of a projected simplex in terms of a scaled B-spline. First we recall the definition of a simplex, then we state the theorem.

DEFINITION 4.3 (simplex)
A *simplex* σ in $\mathbb{R}^D$ is the convex hull of $D+1$ vertices $\boldsymbol{v}^0, \ldots, \boldsymbol{v}^D \in \mathbb{R}^D$. The vertices must be affinely independent, i.e., $\boldsymbol{v}^1 - \boldsymbol{v}^0, \ldots, \boldsymbol{v}^D - \boldsymbol{v}^0$ must be linearly independent. The simplex can then formally be defined as

$$\sigma := \left\{ t_0 \boldsymbol{v}^0 + \cdots + t_D \boldsymbol{v}^D \;\middle|\; \sum_{d=0}^{D} t_d = 1 \text{ and } t_d \geq 0 \;\text{ for all } d = 0, \ldots, D \right\}. \tag{4.49}$$

The volume V_σ of a simplex σ is [Ste66]

$$V_\sigma = \frac{|\det \boldsymbol{A}|}{D!}, \quad \text{where } \boldsymbol{A} := [\boldsymbol{v}^1 - \boldsymbol{v}^0, \ldots, \boldsymbol{v}^D - \boldsymbol{v}^0] \in \mathbb{R}^{DxD}. \tag{4.50}$$

THEOREM 4.4 (The volume of a projected simplex is an M-spline)
The density function $\Phi(x_1)$ obtained by projecting orthogonally onto the first coordinate x_1 the volume of a D-dimensional simplex σ of volume V_σ, so located that its $D+1$ vertices $\boldsymbol{v}^0, \ldots, \boldsymbol{v}^D$ project orthogonally into the knot sequence $\boldsymbol{\xi}^\sigma := (\xi_0^\sigma, \ldots, \xi_D^\sigma)$, is given by

$$\Phi(x_1) = V_\sigma \cdot M^D_{0,\xi^\sigma}(x_1), \tag{4.51}$$

where the M-spline M^D_{0,ξ^σ} can be represented as a scaled B-spline,

$$M^D_{0,\xi^\sigma} = \frac{D}{\xi_D^\sigma - \xi_0^\sigma} b^D_{0,\xi^\sigma}. \tag{4.52}$$

PROOF See Appendix A.2.1. ∎

This theorem, which is illustrated in Fig. 4.4, tells us that the volume of a D-dimensional simplex, projected onto a one-dimensional line, is equal to a scaled B-spline. Thus, by triangulating the unit cube with simplices, we can exactly represent the volume of the $(D-1)$-dimensional hyperplane $\{\boldsymbol{x} \in [0,1]^D \mid \boldsymbol{W}_1^T \boldsymbol{x} = y\}$ as a sum of M-splines. Let S_D be the group of all permutations of $\{1, \ldots, D\}$. Every permutation τ in S_D defines a simplex $\sigma_\tau \subset [0,1]^D$ via

$$\sigma_\tau := \{\boldsymbol{x} = (x_1, \ldots, x_D) \in [0,1]^D \mid 0 \leq x_{\tau(1)} \leq x_{\tau(2)} \leq \cdots \leq x_{\tau(D)} \leq 1\}. \tag{4.53}$$

By construction these simplices triangulate the hypercube $[0,1]^D$ and have equal volume

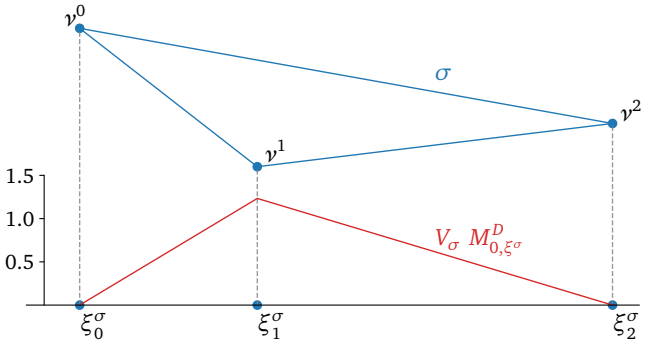


FIGURE 4.4 The volume of the two-dimensional simplex σ is projected onto the x-axis. This equals the scaled M-spline defined on the knot sequence $\boldsymbol{\xi}^\sigma = (\xi^\sigma_0, \xi^\sigma_1, \xi^\sigma_2)$ of projected corners $\boldsymbol{\nu}^0, \boldsymbol{\nu}^1, \boldsymbol{\nu}^2$ as described in Thm. 4.4.

V_1. It follows from Thm. 4.4 that $V(y)$ is given by

$$V(y) = V_1 \sum_{\tau \in S_D} M^D_{0,\xi_{\sigma_\tau}}(y). \tag{4.54}$$

Combining these observations results in the new integration algorithm,

$$\int_{[0,1]^D} f(\boldsymbol{x}) d\boldsymbol{x} = \int_a^b g(y) V(y) dy \tag{4.55}$$

$$\approx \int_a^b \sum_{(\ell,i)\in I} \alpha_{\ell,i}\ b^{n,\mathrm{nak}}_{\ell,i}(y) V_1 \sum_{\tau \in S_D} M^D_{0,\xi_{\sigma_\tau}}(y) dy \tag{4.56}$$

$$= V_1 \sum_{(\ell,i)\in I} \alpha_{\ell,i} \sum_{\tau \in S_D} \int_a^b b^{n,\mathrm{nak}}_{\ell,i}(y) M^D_{0,\xi_{\sigma_\tau}}(y) dy. \tag{4.57}$$

Without Thm. 4.4 the $(D-1)$-dimensional volume $V(y)$ must be numerically approximated, which is expensive and introduces additional inaccuracies. In our new formulation (4.57) however, the volume is exactly captured in form of splines, which are integrated numerically exact using Gaussian quadrature of appropriate order. The complexity of calculating integrals has been eliminated and the method's accuracy solely depends on how well $\tilde{g}$ approximates g.

Practical limitations. The major limitation of the algorithm in practice is the triangulation of the D-dimensional unit hypercube. Our decomposition Eq. (4.53) relies on $D!$ simplices. This is not optimal, but minimal triangulation of the unit cube is a compli-

cated task. Optimal triangulations have so far only been found for up to $D = 7$ [Hug96]. For higher dimensionalities only a lower bound for the number of simplices is known, $(D+1)^{\frac{D-1}{2}}$ [Gla12]. Furthermore, if the initial assumption $\lambda_2 = \cdots \lambda_D = 0$ does not hold then Eq. (4.55) does not hold either. However, according to Lm. 4.2 if $\lambda_2, \ldots, \lambda_D$ are relatively small in comparison to λ_1 then Eq. (4.57) remains a good approximation for the integral of f. The contributions of $\boldsymbol{w}_2, \ldots, \boldsymbol{w}_D$ can be interpreted as noise and interpolating the pairs $(x_{\ell,i},\ f(\boldsymbol{x}'))$ might become unstable. This instability should be able to be overcome by using regression with a suitable regularization term instead of interpolation.

4.4.2 Numerical Results

We now demonstrate the recognition of active subspaces and the new integration algorithm with two example functions. Hereby, we compare cubic B-splines on spatially adaptive sparse grids to the Python Active Subspaces Utility Library (PASUL) [Pas16].

PASUL. PASUL is one of the most popular numerical active subspace libraries. It provides routines for the detection of active subspaces based on Monte Carlo quadrature. If the derivatives of the objective function are not available, PASUL uses approximations based on linear or quadratic ridge functions. These are based on $M \in \mathbb{N}$ samples, drawn from the density ϱ_X. The objective function f is evaluated at each sample, and either a global linear or global quadratic model is least-squares fitted accordingly [Con15a; Pas16]. The linear model has coefficients $c \in \mathbb{R}$ and $\boldsymbol{b} \in \mathbb{R}^D$,

$$f(\boldsymbol{x}) \approx c + \boldsymbol{b}^T \boldsymbol{x}, \tag{4.58}$$

and provides the constant gradient approximation $\nabla f(\boldsymbol{x}) \approx \boldsymbol{b}$. The quadratic model has coefficients $c \in \mathbb{R}$, $\boldsymbol{b} \in \mathbb{R}^D$ and a symmetric matrix $\boldsymbol{H} \in \mathbb{R}^{D \times D}$,

$$f(\boldsymbol{x}) \approx \frac{1}{2} \boldsymbol{x}^T \boldsymbol{H} \boldsymbol{x} + \boldsymbol{b}^T \boldsymbol{x} + c. \tag{4.59}$$

The gradient is then approximated via $\nabla f(\boldsymbol{x}) \approx \boldsymbol{H}\boldsymbol{x} + \boldsymbol{b}$. These simple models have been used in several real world applications, among others for determining the active subspace of a hypersonic scramjet simulation [Con15b], or that of an airfoil shape [Gre18]. Once the active subspace is approximated, PASUL can approximate the function g on the active variables using a least-squares fitted polynomial approximation. In the following comparison we tried all polynomial degrees in $\{1, \ldots, 10\}$ and added the best result to our comparison. The PASUL library also provides a routine which integrates the objective function over the active subspace with respect to the volume of the inactive subspace

$V(y)$. Quadrature nodes are constructed on the active variables using a Delaunay triangulation. The quadrature weights are the relative frequencies within a histogram based on sampling the original variables, mapping them to the active variables, and assigning each sample to the triangle onto which it is projected. In case of a one-dimensional active subspace, the triangulation is reduced to intervals [Pas16].

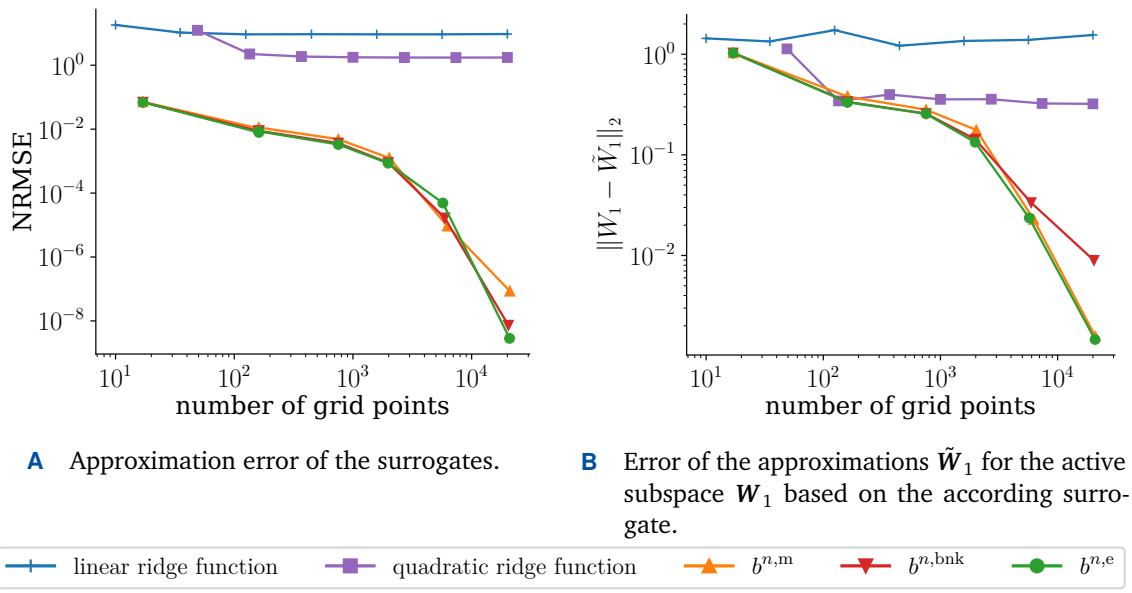
We assume uniform distributions and use two closed form example functions with clear active subspaces for demonstrational purpose. The derivatives of these example functions can easily be derived, but we use that only for the calculation of reference values. For the detection of active subspaces, we assume not to have derivative knowledge, as would be the case in practice. The first example function demonstrates that active subspaces of general dimensionality are well approximated by B-splines on sparse grids. The second example function demonstrates the integration algorithm for functions with a one-dimensional active subspace.

Let now $f_1 : [0,1]^8 \to \mathbb{R}$ be defined as

$$f_1(x) = \sqrt{a_1 x_1 + a_2 x_2} + (a_3 x_3 + a_4 x_4)^3 - \sin(a_5 x_5 + a_6 x_6) - \cos(a_7 x_7 + a_8 x_8), \tag{4.60}$$

with coefficients $\boldsymbol{a} = (a_1, \ldots, a_8) \in \mathbb{R}^8$ randomly chosen under condition that $|\boldsymbol{a}|_2 = 2$ [1]. This function consists of a square root, a polynomial part, and two trigonometric parts, which all rely on different subsets of the input parameters, resulting in f_1 having a four-dimensional active subspace. Figure 4.5A shows the NRMSE of approximating f_1 with PASUL's linear and quadratic ridge functions, and with B-splines on sparse grids. Even thought the function f_1 is relatively simple, both ridge functions are severely restricted, and neither even reaches an NRMSE of 10^{-1} before the error stagnates. This is to be expected, because the approximation qualities of the linear and quadratic approaches are strictly limited. In contrast, the NRMSE of the B-spline approaches converges towards zero, and shows no stagnation. The quality of the response surface is directly related to the quality of the derived active subspace, as can be seen in Fig. 4.5B. Again the quality of the ridge function approximations soon stagnates. The classical approach, based on linear or quadratic ridge functions, and Monte Carlo quadrature vividly reaches its limits. The B-spline approaches converge towards zero, yielding good approximations of the four-dimensional active subspace.

[1] The results in Fig. 4.5 are based on
$\boldsymbol{a}$=[1.2314893703893552, 0.38523139441115317, 0.6394417061820445, 0.36059074362452487, 0.33945470566336716, 0.6960580203613244, 0.19537123690744182, 1.076207757948391]



A Approximation error of the surrogates.

B Error of the approximations $\tilde{\boldsymbol{W}}_1$ for the active subspace $\boldsymbol{W}_1$ based on the according surrogate.

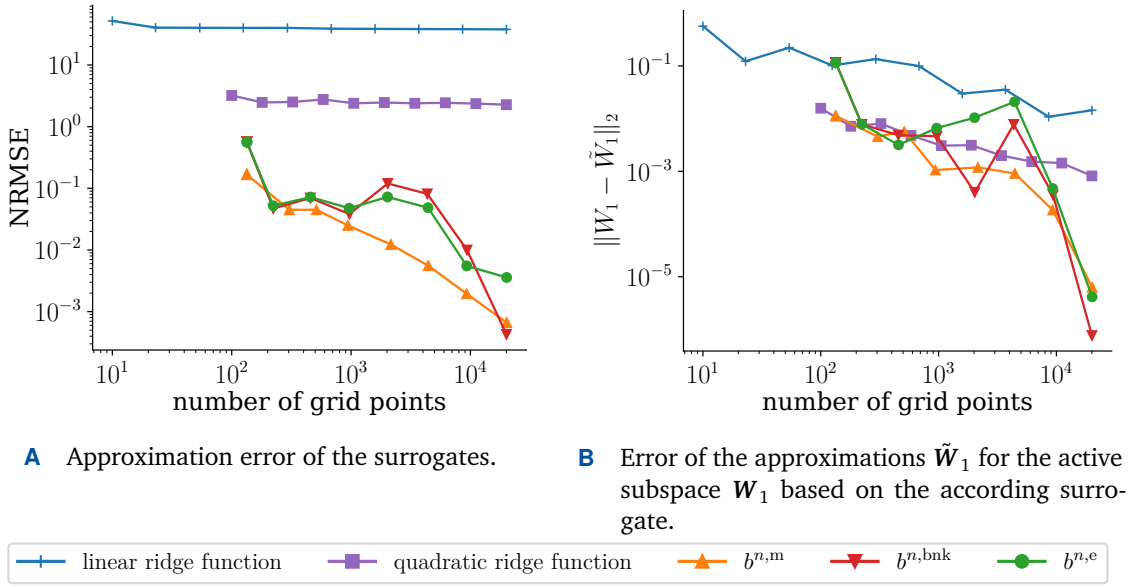
FIGURE 4.5 Errors of the surrogates and resulting active subspace approximations for the example function f_1, based on a linear ridge function, quadratic ridge function, and cubic B-splines on adaptive sparse grids.

The second objective function $f_2 : [0,1]^8 \to \mathbb{R}$ is given by

$$f_2(\boldsymbol{x}) = \frac{\sin(\gamma \sum_{i=1}^{8} x_i + 1)}{\gamma \sum_{i=1}^{8} x_i + 1}, \tag{4.61}$$

where we choose $\gamma = 0.75$. This function has a one-dimensional active subspace. We now investigate the quality of the active subspace approximation, before we apply the new integration algorithm based on the detected active subspace. Figure 4.6A shows the NRMSE of approximating f_2, again with linear and quadratic ridge functions, and B-splines on sparse grids. The behavior is the same as in the previous example. The ridge functions barely converge and the error soon stagnates, whereby the quadratic ridge function is better than the linear one. The error of the B-spline approximations converges towards zero. Again, the NRMSE behavior is related to the quality of active subspace detection, which can be seen in Fig. 4.6B. After the pre-convergence phase, all B-spline approximations have an active subspace approximation error multiple magnitudes smaller than the ridge functions.

Now that we have found the one-dimensional active subspace based on our approximations, we want to exploit this information to integrate the function f_2. The



A Approximation error of the surrogates.

B Error of the approximations $\tilde{\boldsymbol{W}}_1$ for the active subspace $\boldsymbol{W}_1$ based on the according surrogate.

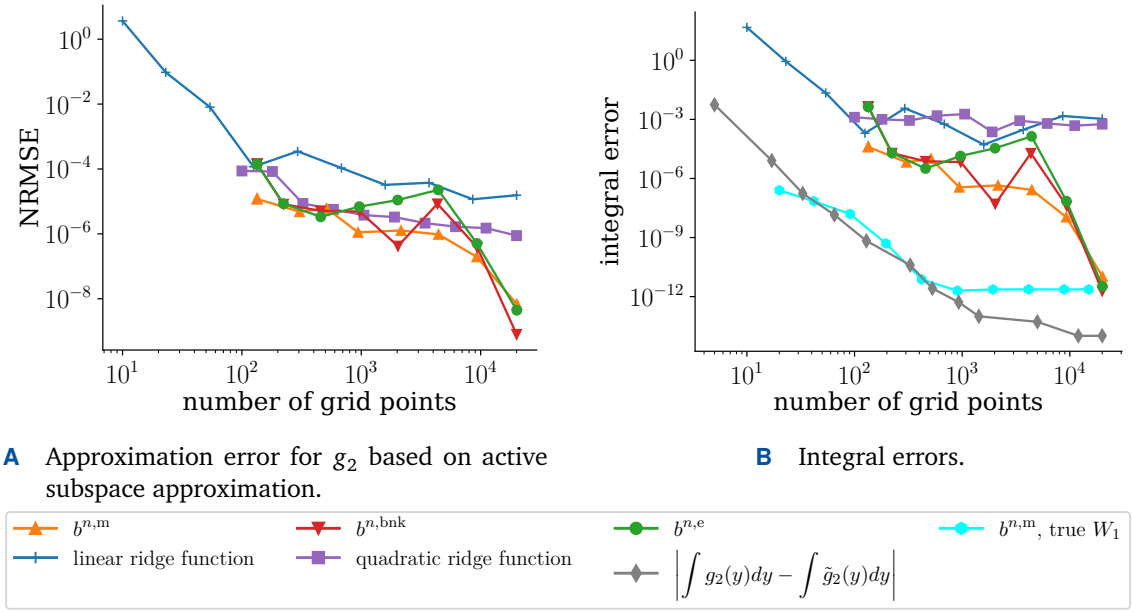
FIGURE 4.6 Errors of the surrogates and resulting active subspace approximations for the example function f_2, based on a linear ridge function, quadratic ridge function, and cubic B-splines on adaptive sparse grids.

corresponding univariate function $g_2 : [0, \sqrt{8}] \to \mathbb{R}$ is given by

$$g_2(y) = \frac{\sin(\gamma\sqrt{8}y + 1)}{\gamma\sqrt{8}y + 1}. \tag{4.62}$$

In Fig. 4.7A we show the error of approximating g_2 from evaluations of f_2 and the approximated active subspace. The results directly depend on the quality of the approximated active subspace. Thus, the integration error resembles the behavior in Fig. 4.6B. After the pre-convergence phase, the B-spline-based approximations outperform the polynomial approximations.

Using Eq. (4.62) and (4.57) we calculate a numerically exact reference value for the integral $\int_{[0,1]^8} f_2(\boldsymbol{x})d\boldsymbol{x}$. In Fig. 4.7B the error of approximating this integral is shown. The integration algorithm (4.57) is numerically exact, therefore the only error source for the B-spline-based integrations is the quality of the approximation of g_2. Indeed, for the B-spline approaches the integral convergence plot looks almost exactly like Fig. 4.7A. To emphasize the numerical exactness of the integration algorithm we provided the algorithm with the exact active subspace W_1. Additionally, we plotted the error $\left|\int_{[0,\sqrt{8}]} g_2(y)dy - \int_{[0,\sqrt{8}]} \tilde{g}_2(y)dy\right|$, where $\tilde{g}_2$ is the B-spline approximation of g_2 on a univariate sparse grid. The two errors converge alike, demonstrating that indeed the integration algorithm reduces the complexity of the original high-dimensional integral



A Approximation error for g_2 based on active subspace approximation.

B Integral errors.

FIGURE 4.7 Errors of the approximations $\tilde{g}_2$ calculated from f_2 using an approximation for the active subspace (left). Errors for approximating the integral $\int_{[0,1]^8} f_2(\boldsymbol{x})d\boldsymbol{x}$ with a univariate integral based on the approximations for g_2 with a linear and a quadratic ridge function, and cubic B-splines on adaptive sparse grids, the error for approximating the integral with the spline-based quadrature algorithm supplied with the exact active subspace $\boldsymbol{W}_1$, and the error for integrating g_2 with not a knot B-splines on a one-dimensional adaptive sparse grid (right).

to the complexity of univariate integration, where the error is solely determined by the approximation of the active subspace.

5

Application I
Subsurface CO_2 Storage

> *If we are to have any hope of keeping global temperature [increases] down below 2°C then we desperately need to develop ways to capture and store carbon dioxide.*
>
> — Stuart Haszeldine [McK]

The progressive industrialization of the last centuries has led to a drastic increase in global energy demand. This in turn resulted in vast amounts of waste, among others carbon dioxide (CO_2), which must be stored securely. Storing CO_2 in geological formations is an interim technology with high potential to mitigate CO_2 emissions, and to weaken the greenhouse effect [IPC05; IPC11]. In order to minimize the risks involved, any potential storage site must be investigated extensively, whereby investigations are complicated by the fact that subsurface properties, like porosity or permeability, are observable only to a limited extent. Consequently, a lot of uncertainty is present when modeling such processes. This uncertainty can usually not be described analytically, but instead measurements and observations are used to describe a data-driven model.

To compare multiple different UQ methodologies for such scenarios, Köppel et al. established a data-driven benchmark for probabilistic risk assessment for a subsurface CO_2 storage [Köp18]. The results and underlying model are publicly available [KöpDa]. In this chapter we run the same benchmark with B-splines on sparse grids using different

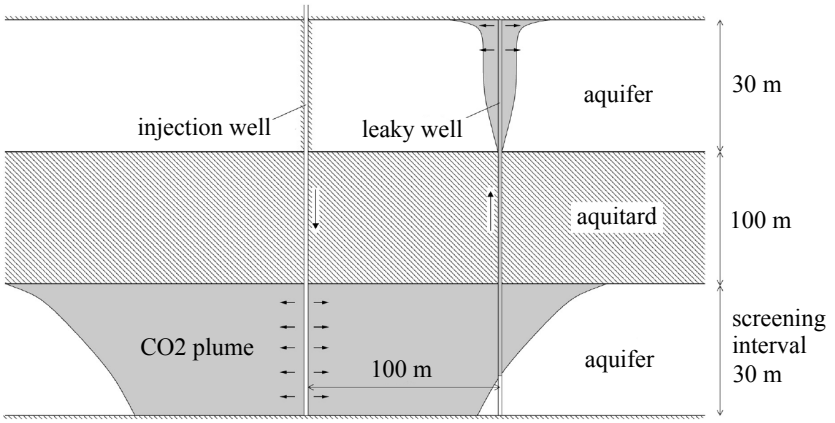


FIGURE 5.1 Cross-section of the CO_2 storage benchmark's physical domain from [Ebi07]. CO_2 is injected at high pressure through the central injection well, spreads in the lower aquifer and partially escapes over time through the leaky well.

spatial refinement strategies, and add the results to the comprehensive comparison.

We first introduce the CO_2 subsurface storage benchmark in Sec. 5.1, before we present two alternative refinement criteria for spatial adaptivity in Sec. 5.2. In Sec. 5.3 we first show how different B-splines perform in the benchmark, then describe the other UQ methodologies, and finally compare them to our approach.

This chapter is based on a collaboration with Dr. Fabian Franzelin and Prof. Dr. Sergey Oladyshkin (both SimTech, University of Stuttgart, Germany), and continues the comparison from [Köp18]. The presented results have already been published in [Reh21a]. The collaborators provided the previous results and the underlying benchmark in form of a black-box simulation. The author of this thesis implemented the new spatial adaptivity criteria for B-splines, and added B-splines on sparse grids to the overall comparison, demonstrating the high competitiveness of this approach. In particular the new results outperform the previous ones based on sparse grids with hierarchical C^0 elements.

5.1 Carbon Dioxide Storage Model

Carbon dioxide can be captured at industrial facilities, compressed into a liquid state, and pumped into deep underground formations. There it dissolves into the local formation water through a variety of geochemical reactions [Roc04]. This approach, known as carbon capture and storage (CCS), is considered a major factor in the battle against climate change [Has18]. In practice, it is of absolute importance to accurately estimate the behavior of a storage site before a decision can be made on its usage. Otherwise,

Name	Description	Range	Unit
Φ	reservoir porosity	$[0, 0.8]$	–
q	CO_2 injection rate	$[0.6, 1.4] \cdot 10^{-4}$	m^3/s
θ	relative permeability degree	$[2, 4]$	–

TABLE 5.1 Input parameters for the CO_2 storage benchmark.

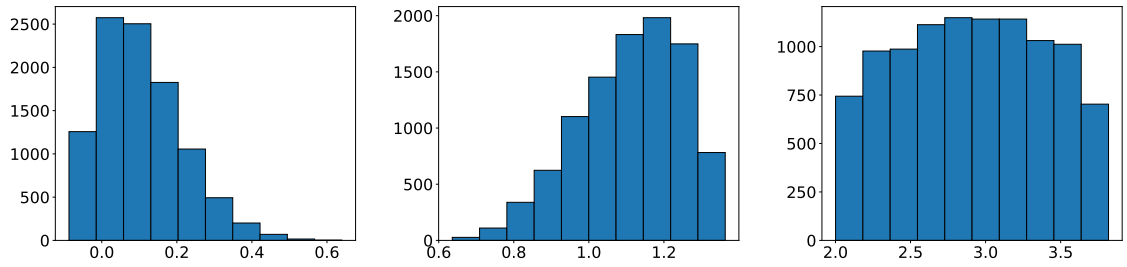
the high pressure used to pump CO_2 into the reservoir could cause structural failures leading to earthquakes, pollution of drinking water, or the CO_2 could escape back into the atmosphere. The investigation is complicated by the fact that the raw data obtained from such storage sites can hardly be described by theoretical stochastic distributions. Additionally, external driving forces and material properties are observable only to a limited extent and at high costs [Köp18]. Thus, data-driven UQ approaches must be used to incorporate observations and measurements.

To give an overview of the available methods and to compare their performance, Köppel et al. set up a benchmark for carbon dioxide storage and implemented it with four different forward UQ methods. The benchmark domain is illustrated in Fig. 5.1. Carbon dioxide is pumped through the central injection well into an aquiferous reservoir, separated from an upper aquifer by an aquitard. The CO_2 starts spreading in the lower aquifer and, after some time, reaches a leaky well. Through this well some CO_2 escapes back to the atmosphere over time. The relevant quantities in this scenario are the expected value and standard deviation of the total amount of escaping CO_2 over a period of 100 days. With this information the suitability of the potential storage site can then be assessed.

The benchmark is modeled as a multiphase flow problem in porous media. The original model, which is based on the multiphase version of Darcy's law and the mass balance of the brine (water) and the CO_2 (gas) phase, was derived by Class et al. [Cla09]. It was then modified, resulting in a system of partial differential equations [Nor11]. A coordinate transformation finally resulted in the following PDE formulation [Ola11; Köp18],

$$\Phi \frac{\partial S_g}{\partial \tau} - \frac{1}{r} \frac{\partial}{\partial r} \left(q C_p f_g(\theta) \right) - q = 0, \tag{5.1}$$

which is formulated in a radial coordinate system depending on the coordinate r, which is the distance from the injection well in meters. The other components of Eq. (5.1) are the saturation of the gas-phase S_g, the time of the simulation τ, a pressure constant C_p, and the fractional flow function f_g, which depends on the saturation of liquid- and



A Reservoir porosity Φ [-] **B** Injection rate $q \times 10^{-4}$ [m^3/s] **C** Relative permeability degree θ [-]

FIGURE 5.2 Histograms of the 10 000 Monte Carlo samples forming the data-driven distributions for the parameters of the CO_2 storage benchmark.

gas-phase. As initial conditions, the aquifer is filled with brine, and CO_2 is injected at a constant rate at the center of the physical domain. The lateral boundary conditions are constant Dirichlet conditions and equal to the initial conditions. All other boundaries are no-flow boundaries. The hyperbolic transport equation (5.1) depends on a spatial as well as a temporal coordinate. In the physical space a semi-discrete central-upwind finite volume scheme is applied [Kur05]. The temporal space is discretized using second order Runge-Kutta.

There are three uncertain parameters in Eq. (5.1), which we summarize in Tab. 5.1. The reservoir porosity Φ models uncertainty of material properties, the CO_2 injection rate q models uncertainty in the boundary conditions, and the relative permeability degree θ models uncertainty from the non-linearity of the conceptual model, i.e., uncertainty from the conceptualization of the model. No explicit distributions are known for these uncertain parameters, instead a set of 10 000 samples is provided with the benchmark and a data-driven approach is used to incorporate the uncertainty information. The reference stochastic moments were calculated using the Monte Carlo method (cf. Equations (4.9) and (4.10)), and for consistency we too use Monte Carlo to compute the moments of our surrogate with respect to the data. Histograms of this data are shown in Fig. 5.2.

The multiphase flow problem is strongly nonlinear and solving it for a given parameter combination is computationally demanding. This is also the reason for the small number of parameters of the benchmark, which were designed to still allow for a Monte Carlo based reference solution. For the full derivation of Eq. (5.1) and more details on the model and the simulation we refer to [Kur05; Ola11; Köp18]. An overview of the current status of modeling CO_2 storage in deep saline aquifers is given in [Cel15].

5.2 Spatial Adaptivity

In addition to surplus-based refinement, cf. Sec. 2.5, for this application we gave two other spatial adaptivity refinement strategies a try. These refinement criteria were derived and employed by Franzelin in the context of the same CO_2 storage benchmark [Fra17].

Let $\tilde{f} = \sum_{(\boldsymbol{\ell},\boldsymbol{i})\in I} \alpha_{\boldsymbol{\ell},\boldsymbol{i}} \varphi_{\boldsymbol{\ell},\boldsymbol{i}}$ be the current interpolant of the objective function f, and ϱ be a probability density function. The idea of the first refinement strategy is to minimize the functional

$$J(I) \coloneqq \int \left(f(\boldsymbol{x}) - \tilde{f}(\boldsymbol{x})\right)^2 \varrho(\boldsymbol{x}) d\boldsymbol{x}, \tag{5.2}$$

i.e., to minimize the weighted L^2-error. To do so, we determine

$$\begin{aligned} (\boldsymbol{\ell}^*, \boldsymbol{i}^*) &= \operatorname*{arg\,max}_{(\boldsymbol{\ell},\boldsymbol{i})\in I^{\text{ref}}} | J(I \backslash \{(\boldsymbol{\ell},\boldsymbol{i})\}) | \\ &= \operatorname*{arg\,max}_{(\boldsymbol{\ell},\boldsymbol{i})\in I^{\text{ref}}} |\alpha_{\boldsymbol{\ell},\boldsymbol{i}}| \left(\int \varphi_{\boldsymbol{\ell},\boldsymbol{i}}(\boldsymbol{x})^2 \varrho(\boldsymbol{x}) d\boldsymbol{x} \right)^{\frac{1}{2}}, \end{aligned} \tag{5.3}$$

and refine by iteratively adding the children $C(\boldsymbol{\ell}^*, \boldsymbol{i}^*)$ to the sparse grid. This strategy is called *L^2-based refinement*. The second criterion is derived analogously by setting $J(I) \coloneqq \mathbb{E}[f - \tilde{f}]$, which leads to *expected-value-based refinement* [Fra15],

$$(\boldsymbol{\ell}^*, \boldsymbol{i}^*) = \operatorname*{arg\,max}_{(\boldsymbol{\ell},\boldsymbol{i})\in I^{\text{ref}}} | \ \alpha_{\boldsymbol{\ell},\boldsymbol{i}} \ \mathbb{E}[\varphi_{\boldsymbol{\ell},\boldsymbol{i}}] \ |. \tag{5.4}$$

We set $\mathbb{E}[\varphi_{\boldsymbol{\ell},\boldsymbol{i}}]$ to $\mathbb{E}_{\varrho}[\varphi_{\boldsymbol{\ell},\boldsymbol{i}}]$ if the probability density function ϱ is analytically known. In data-driven scenarios, where only samples of an unknown probability density function are available, we use $\mathbb{E}_{\text{MC}}[\varphi_{\boldsymbol{\ell},\boldsymbol{i}}]$.

The two refinement strategies promise improvements for the approximation of their corresponding quantities. However, they require multiple integrations in each refinement step, and they are thus much more expensive than surplus-based refinement.

5.3 Numerical Results

The subsurface storage simulation uses 250 time steps to discretize the time period of 100 days. It returns $(f_1, \ldots, f_{250})$, where f_t is the amount of escaping CO_2 at time t. We create one surrogate $\tilde{f}_t$ for each time step, all based on the same underlying adaptive sparse grid, and measure the average L^2-error of the quantities of interest over all time

steps,

$$\varepsilon_{\mathbb{E}} = \left(\sum_{t=1}^{250} \left(\mathbb{E}_{\mathrm{MC}}(f_t) - \mathbb{E}_{\mathrm{MC}}(\tilde{f}_t) \right)^2 \right)^{\frac{1}{2}} \frac{1}{250}, \tag{5.5}$$

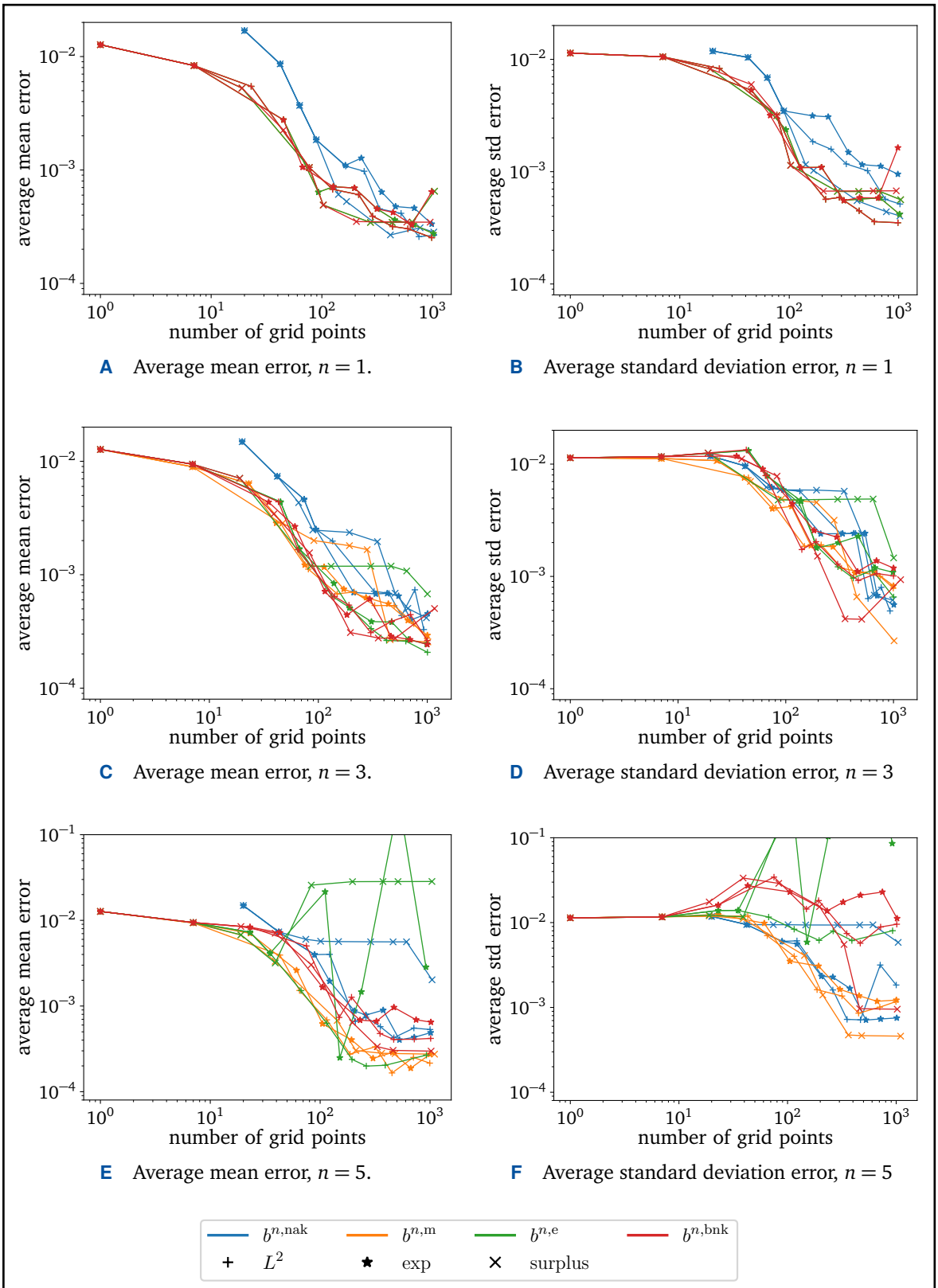
$$\varepsilon_s = \left(\sum_{t=1}^{250} \left(s_{\mathrm{MC}}(f_t) - s_{\mathrm{MC}}(\tilde{f}_t) \right)^2 \right)^{\frac{1}{2}} \frac{1}{250}. \tag{5.6}$$

5.3.1 Results for B-spline Bases

Figure 5.3 shows the convergence for boundary not-a-knot B-splines, as well as modified, extended and boundaryless not-a-knot B-splines of degrees $n \in \{1, 3, 5\}$ using surplus-based, weighted L^2, and expected-value-based spatial refinement respectively. For degrees $n = 1$ and $n = 3$ the non-boundary splines outperform the boundary approach, which requires too many grid points on the boundary. For degree $n = 5$ some quintic bases become unstable. This is caused by a discontinuity of the underlying subsurface flow simulation, which is due to the CO_2 displacement front resembling a shock. The discontinuity also generally prevents higher-order splines from reaching optimal convergence order. Instead, all stable bases perform rather similar. At an error of around $5 \cdot 10^{-4}$ the convergence stagnates as the accuracy of the Monte Carlo reference solution is reached.

Linear B-splines generally perform better at computing the standard deviation, while the stable quintic B-splines perform slightly better in computing the expected value. The cubic B-splines lie in between. This can be traced back to the discontinuity. Closer to the discontinuity a smooth surrogate of degree 5 oscillates. While in the calculation of the expected value these oscillations cancel each other, for the calculation of the standard deviation this is not the case. Squaring the integrand means that each oscillation is taken into account and deteriorates the result. The higher the degree, the worse the oscillations of the surrogate. Thus, the smooth surrogate of degree 5 is superior for mean calculations, but inferior for standard deviation. This also explains why cubic B-splines lie in between the others. They are not as smooth as the quintic splines, but in contrast to linear splines they oscillate close to discontinuities.

Overall, the smallest expected value error is achieved by modified B-splines of degree 5 using weighted L^2 refinement. The overall smallest standard deviation error is achieved by all non-boundary B-splines of degree 1 using weighted L^2 refinement, because all boundaryless B-splines behave alike for degree $n = 1$, cf. Sec. 3.2. We compare these results to other UQ methodologies in the next section.



A Average mean error, $n = 1$.

B Average standard deviation error, $n = 1$

C Average mean error, $n = 3$.

D Average standard deviation error, $n = 3$

E Average mean error, $n = 5$.

F Average standard deviation error, $n = 5$

FIGURE 5.3 Average mean error and average standard deviation error for approximations of the subsurface storage benchmark using four different spline types, degrees $n \in \{1, 3, 5\}$ and three different refinement criteria (L^2-based, expected-value-based, surplus-based).

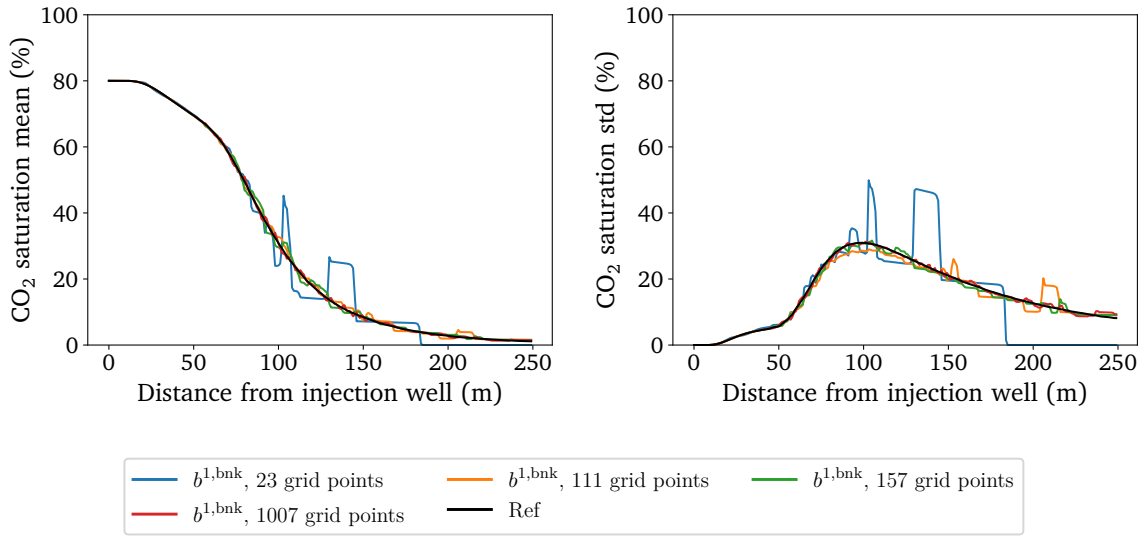


FIGURE 5.4 Mean and standard deviation of CO_2 saturation at $t = 100$ days based on boundaryless not-a-knot B-splines of degree 1 on L^2-based spatially adaptive grid points of increasing refinement in comparison to the reference Monte Carlo solution using 10^4 samples. The CO_2 saturation decreases with the distance from the injection well.

Figure 5.4 shows the approximated mean and standard deviation of the benchmark exemplarily for linear boundaryless not-a-knot B-splines on L^2-based spatially adaptive sparse grids of increasing size at the end of the simulation time. It can be seen that already for a sparse grid of only 111 points the characteristics of the objectives are adequately resolved. This shows that our method is applicable even in extremely data-scarce regimes. Still, as is to be expected, for increasing grid size the approximation then gets closer to the reference.

Assessment of the refinement criteria. For the application in this chapter we used the weighted L^2-based and expected-value-based refinement strategies from [Fra17]. As can be seen in Fig. 5.3, the differences between the different criteria and surplus-based refinement are marginal. However, the L^2 as well as the expected-value-based refinement require multiple integrations in each refinement step and are thus more expensive than the traditional surplus-based refinement which needs no additional calculations at all. Therefore, we do not pursue these two approaches, but in the upcoming applications we only use surplus-based refinement. We interpret these results however not so much as failure of the L^2-based and expected-value-based refinement, but rather as a confirmation of the quality of the traditional surplus criterion, which prioritizes the ares of most

importance for a great variety of quantities of interest.

5.3.2 Comparison of UQ Methods

We now compare the best B-spline surrogates from the previous section with all methods from the original comparison [Köp18]. First we give a brief description for each comparative method.

Arbitrary polynomial chaos expansion. A variant of PCE (cf. Sec. 4.2.2), which does not rely on analytical standard distributions from the Wiener-Askey scheme. Instead, it adapts to arbitrary probability distributions given in form of data [Ola12]. Statistical moments are derived from the dataset and used to numerically construct an orthonormal polynomial basis. Then an expansion is constructed just like with the generalized PCE in Eq. (4.17).

Spatially adaptive sparse grids with hierarchical C^0 elements. Sparse grids were already present in the original comparison. There, piecewise d-polynomial C^0-elements on boundary grids and modified C^0-elements on non-boundary grids were used, cf. Sec. 2.2. The sparse grids were refined using the same three spatial adaptivity criteria that we presented in Sec. 5.2, of which weighted L^2-refinement gave the best results.

Kernel greedy interpolation. Linear combinations of radial basis functions are a popular method for the creation of surrogates [Wen95]. In this comparison they are represented in the form of C^2 Wendland kernels [Wen04]. In contrast to sparse grids, these basis functions do not belong to a structured mesh. Instead, starting from a large set of potential evaluation points, a vectorial kernel orthogonal greedy algorithm iteratively picks the most promising points. The kernel functions are centered at these scattered points and the coefficients of the surrogate are calculated using interpolation.

Hybrid stochastic Galerkin. An intrusive method, which does not construct a surrogate. Instead, the stochastic domain is decomposed into stochastic elements (cubic subsets of the domain), each of which gets its own individual orthonormal polynomial PCE basis [Bür14]. This stochastic discretization yields a representation of the original partial differential equation as linear combination of deterministic coefficients and polynomials with random arguments. For the application at hand, a uniform distribution is assumed and Legendre polynomials are used.

Figure 5.5 shows the results of the overall comparison. Our surrogates perform highly competitive and show the smallest mean error for moderate numbers of model

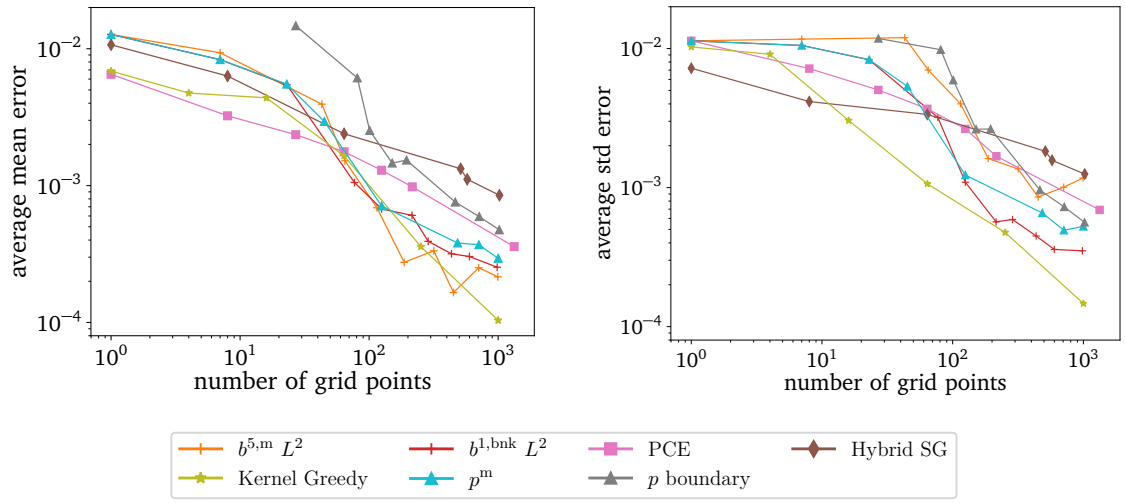


FIGURE 5.5 Comparison of average mean and standard deviation error of B-splines on sparse grids with the four methods from the original comparison: arbitrary polynomial chaos expansion, Kernel Greedy Interpolation, Hybrid Stochastic Galerkin, modified C^0 elements p^{m} on a non-boundary sparse grid, and C^0 elements p on a boundary sparse grid.

runs. In particular, for both quantities of interest our B-splines approaches outperform the former sparse grid results which were based on (modified) hierarchical C^0 elements. For standard deviation all methods fall short for the Kernel Greedy Interpolation. However, the graph does not take into account that for this method actually a shape parameter δ must be tuned. This is usually done based on additional model runs which would increase the cost of Kernel Greedy Interpolation, and thus deteriorate its performance in the comparison. These additional model runs were not taken into account in the comparison, althought multiple fixed δ were tested a-priori.

Summary. We used surrogates based on B-splines and sparse grids to approximate an expensive benchmark for subterranean CO_2 storage. Using stochastic collocation, we calculated the quantities of interest and compared the results with established UQ methods. The good results of our methods are, to some extent, caused by the flexibility in the degree. Quintic B-splines perform better for the standard deviation of this simulation, while the mean is captured better with linear basis functions. Choosing the degree depending on the problem at hand and the desired quantity of interest resulted in one of the best results of the comparison. The authors of the original study also explain that the uncertain parameters of the problem strongly influence the non-linearity of the underlying flow problem. This leads to Gibbs phenomena and oscillations of the corresponding surrogates. These effects are weakened considerably by the use of spline functions of

moderate degree. The application also demonstrates that in practice higher degrees can be dangerous. While we have seen their improved convergence behavior for smooth objectives in Sec. 3.3, the discontinuities in this application led to unstable behavior.

6

Application II
Bayesian Inference for a DC Engine

> *Under Bayes' theorem, no theory is perfect. Rather, it is a work in progress, always subject to further refinement and testing.*
>
> — Nate Silver [Sil12]

In order to better understand the behavior of essential electrical and mechanical components, and to create virtual prototypes, Bosch Research is interested in quantitative ways to determine the effect of parameter uncertainties in simulation and measurement. To mimic real counterparts as closely as possible, measurements capturing their behavior must be transferred to calibrateable simulations. In recent years, a test-bench for an electric direct current (DC) motor has been established to explore the technical possibilities of such a digital pendant [Gla16; Gla17; Joh19]. To account for varying material properties and production tolerances, it is important that uncertainties are taken into account and the reliability of the results is formulated explicitly. Therefore, hierarchical Bayesian modeling is used to infer the probability of model parameters from the test-bench measurements.

The most expensive part of Bayesian inference is the sampling of the posterior. So far, in context of the DC motor test-bench, mainly Metropolis-Hastings has been used [Gla16]. Recently, Approximate Bayesian Computation was used and improved

these results [Joh21a; Joh21b]. The NUTS algorithm, which we introduced in Sec. 4.3 is another modern and efficient sampling approach. The idea of applying NUTS for the test-bench inference problem was considered highly promising, but so far it was put aside because the simulation does not provide any derivative information, and basic approximations thereof are too expensive. This chapter describes a collaboration with Bosch Research, in which we demonstrate the powerful coupling of the NUTS algorithm and B-splines on spatially adaptive sparse grids, which provide good approximations for the mandatory derivatives.

In a first step, in Sec. 6.1 we use an artificial setup for which the analytical solution is known, and for which measurements are created synthetically by adding Gaussian noise to evaluations. This allows us to accurately measure the quality of the surrogate, its Jacobian and the inference results, as well as the influence of noise. However, the real benefit of our approach is demonstrated in Sec. 6.2, where we use real test-bench measurements and a complex black-box simulation modeling the motor behavior. With this we demonstrate that NUTS enabled through the derivatives of our surrogates is applicable in practice, and that it delivers results fulfilling industry requirements.

This original chapter is based on a collaboration with Dr. Michael Schick, and Dr. David John (both Bosch Research). New contributions are the usage of NUTS based on the derivatives of a surrogate from B-splines on sparse grids, and the resulting parameter calibration. The collaborators contributed the DC motor models, the real-world benchmark measurements, and the ABC results, which have previously been published in [Joh21a; Joh21b].

6.1 Artificial Problem

A DC motor is classically composed of a rotor between the two poles of a magnetic field, called the stator. The rotor is surrounded by the armature winding, which consists of several coils. When electrical current flows through the armature winding, the rotor becomes magnetic and starts to turn according to the permanent magnetic field of the stator. The commutator, located on the rotor, changes the poles of the rotor after each turn of 180 degrees leading to continuous torque. Assuming no mechanical load, this physical behavior can be described by an ODE system derived from one equation for the electrical side and one equation for the mechanical side. The two output quantities of the system are the armature current I(A) and the rotational speed ω(rad/s) over time t(s),

Name	Description	Distribution			Unit
R	resistance	$N(\mu_R,\ \sigma_R^2)$	$\mu_R = 9.0$	$\sigma_R = 0.9$	Ω
c_g	motor constant	$N(\mu_{c_g}, \sigma_{c_g}^2)$	$\mu_{c_g} = 3.0$	$\sigma_{c_g} = 0.3$	Nm/A
c_m	motor constant	$N(\mu_{c_m}, \sigma_{c_m}^2)$	$\mu_{c_m} = 0.5$	$\sigma_{c_m} = 0.05$	V s/rad
L	inductance	$N(\mu_L,\ \sigma_L^2)$	$\mu_L = 0.11$	$\sigma_L = 0.011$	H
J	torque of inertia	$N(\mu_J,\ \sigma_J^2)$	$\mu_J = 0.1$	$\sigma_J = 0.01$	kg m^2
D	friction	$N(\mu_D,\ \sigma_D^2)$	$\mu_D = 0.1$	$\sigma_D = 0.01$	kg m/s

TABLE 6.1 Input parameters for the DC motor ODE system (6.1). For each input parameter we set up a distribution to parametrize the DC motor ODE system. The specification of these distributions is chosen arbitrarily in order to investigate the proposed inference methods. We assume all distributions to be independent.

$$\begin{aligned} \frac{dI(t)}{dt} &= -\frac{R}{L}I(t) - \frac{c_m}{L}\omega(t) + \frac{V}{L}, \\ \frac{d\omega(t)}{dt} &= \frac{c_g}{J}I(t) - \frac{D}{J}\omega(t), \end{aligned} \tag{6.1}$$

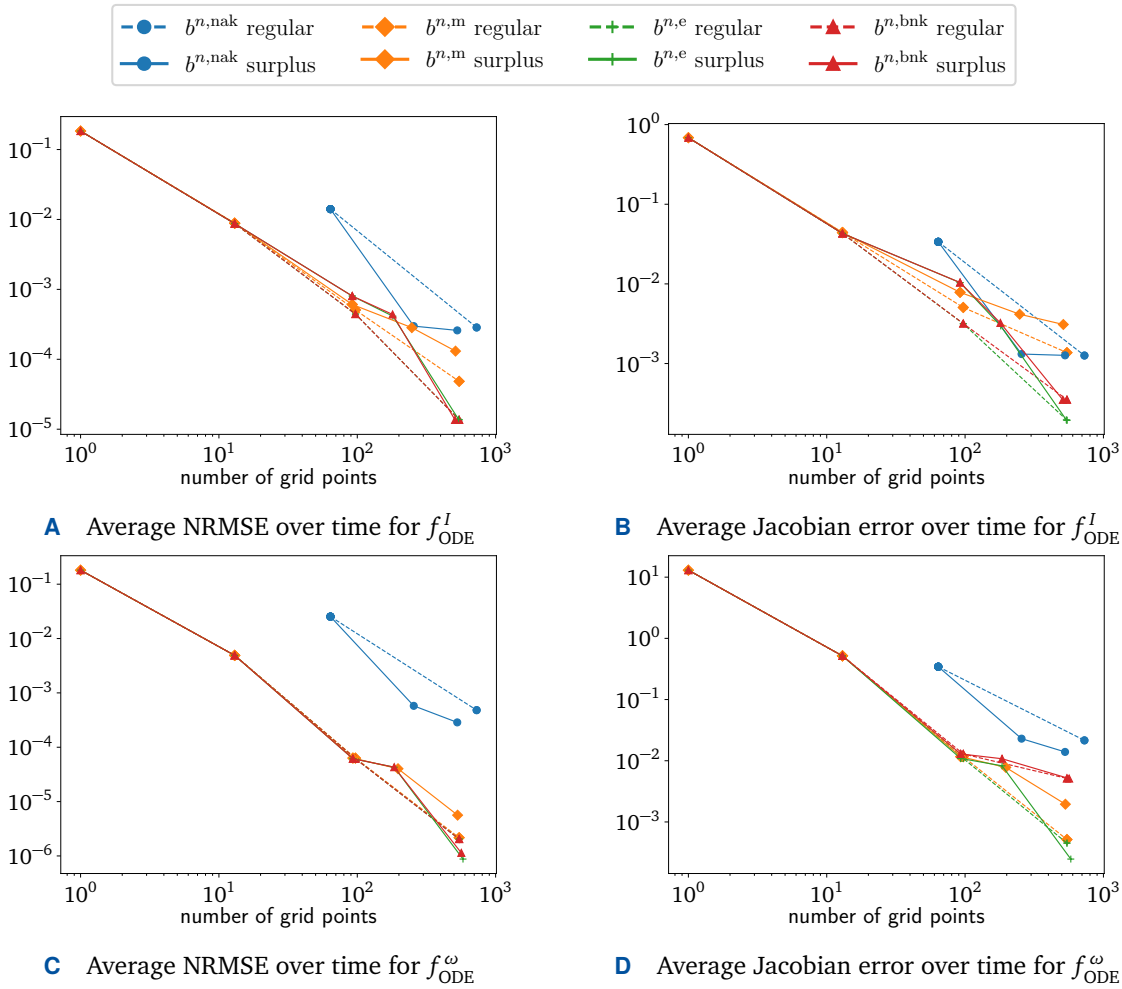
where we use a constant voltage of $V = 12V$, and initial values $I(0) = \omega(0) = 0$. The six input parameters are summarized in Tab. 6.1. Note that the following inference does not depend on this particular choice of distributions and that the presented method works for non-normal distributions too. Still, in practice normal distributions are often the default choice and thus they are used for the test-bench, too. For further details on this ODE system and the general modeling of electrical motors we refer to the literature [Tol18].

We set up the objective function to be the numerical solution of the initial value problem (6.1) on the time domain $[0, 6]$ seconds, which is discretized using $N_t = 601$ equidistant time steps. Given values for all six input parameters, we use the `odeint` routine from the SciPy library [Vir20], which is based on the FORTRAN library `odepack` [Hin83], to solve Eq. (6.1). The resulting objective function is

$$\begin{aligned} f_{\text{ODE}} = (f_{\text{ODE}}^I,\ f_{\text{ODE}}^\omega) : \mathbb{R}^6 &\to \mathbb{R}^{2\times N_t}, \\ \boldsymbol{X} &\mapsto \boldsymbol{I}, \boldsymbol{\omega}, \end{aligned} \tag{6.2}$$

where $\boldsymbol{X} = (R, c_g, c_m, L, J, D)$ is the vector of input random variables, and $\boldsymbol{I} = (I_0, \ldots, I_{N_t-1})$, and $\boldsymbol{\omega} = (\omega_0, \ldots, \omega_{N_t-1})$ are the discretized outputs.

In this first scenario an analytical solution for Eq. (6.1) is known, and it is stated in Appendix B.1. However, as analytical solutions are generally not available, we perform the inference using the numerical approach (6.2), which lets us incorporate the numerical uncertainty stemming from inaccuracies of the solver. Still, we make use of the analytical



A Average NRMSE over time for f^I_{ODE}

B Average Jacobian error over time for f^I_{ODE}

C Average NRMSE over time for f^ω_{ODE}

D Average Jacobian error over time for f^ω_{ODE}

FIGURE 6.1 Average errors over time for the approximations of the artificial DC motor functions f^I_{ODE} and f^ω_{ODE}, and their Jacobians for cubic B-splines on regular and adaptive sparse grids.

solution to verify the accuracy of the surrogate and its Jacobian.

6.1.1 Surrogate Convergence

Using the analytical solution of the coupled ODE system (6.1) and its Jacobian, we create a reference data set of 1 000 evaluations. We interpolate the objective function f_{ODE} with various cubic B-splines on regular and spatially adaptive sparse grids, to find out which combination works best. Higher degrees do not provide any better convergence, indicating a limited smoothness of the objective function.

The resulting average NRMSE and Jacobian error for both outputs are illustrated in Fig. 6.1. All non-boundary B-splines outperform not-a-knot B-splines including the

boundary, because in this six-dimensional scenario the boundary splines cannot adequately resolve the inner domain. Among the non-boundary splines the extended B-splines perform slightly best, and therefore we will use these in the upcoming inference section. Note, that the error of the surrogate and its Jacobian can usually not be determined in practice. Still, all basis functions approximate the objective well and none would significantly impair the results, when choosing ignorantly.

6.1.2 Inference for the Artificial Motor Model

Synthetic measurement data. We now assume the parameter distributions in Tab. 6.1 to be unknown, and want to infer them from M noisy measurements. In this artificial scenario we will use perturbed evaluations of f_{ODE} itself. Such synthetic data is comprehensible and allows a detailed comparison of the inference results to reference values. To simplify notation, let $(\cdot)$ from now on indicate that we apply something for both outputs, I and ω. We create synthetic measurements $\boldsymbol{y}_i^{(\cdot)}, i = 1, \dots, M$ by drawing a sample $\boldsymbol{x}_i$ from the joint product distribution of the parameters, evaluating the objective function, and adding noise,

$$\boldsymbol{y}_i^{(\cdot)} = f_{\text{ODE}}^{(\cdot)}(\boldsymbol{x}_i) + \varepsilon^{(\cdot)}, \tag{6.3}$$

where $\varepsilon^{(\cdot)}$ is a sample of independent and identically normally distributed Gaussian noise with standard deviation $\sigma^I = 0.1$ and $\sigma^\omega = 0.5$ respectively. This noise emulates the measurement errors and inaccuracies which are inevitable when working with real physical set-ups. We summarize the measurements in the data-set Y,

$$Y := \{\boldsymbol{y}_i^I, \boldsymbol{y}_i^\omega \mid i = 1, \dots, M\}. \tag{6.4}$$

The synthetic measurements can be seen in Fig. 6.2.

Hierarchical Bayesian model. We formulate the stochastic inverse problem as a hierarchical model, assuming Gaussian distributions for each parameter from Tab. 6.1 and inferring the hyper-parameters. In order to infer the unknown probability distribution $\pi(\boldsymbol{X})$, where $\boldsymbol{X} = (R, c_g, c_m, L, J, D)$, we first set up the hierarchical distributions for each input parameter,

$$\pi(\boldsymbol{X}_d \mid \tilde{\mu}_{\boldsymbol{X}_d}, \tilde{\sigma}_{\boldsymbol{X}_d}) \sim N(\tilde{\mu}_{\boldsymbol{X}_d}, \tilde{\sigma}^2_{\boldsymbol{X}_d}), \quad d = 1, \dots, 6, \tag{6.5}$$

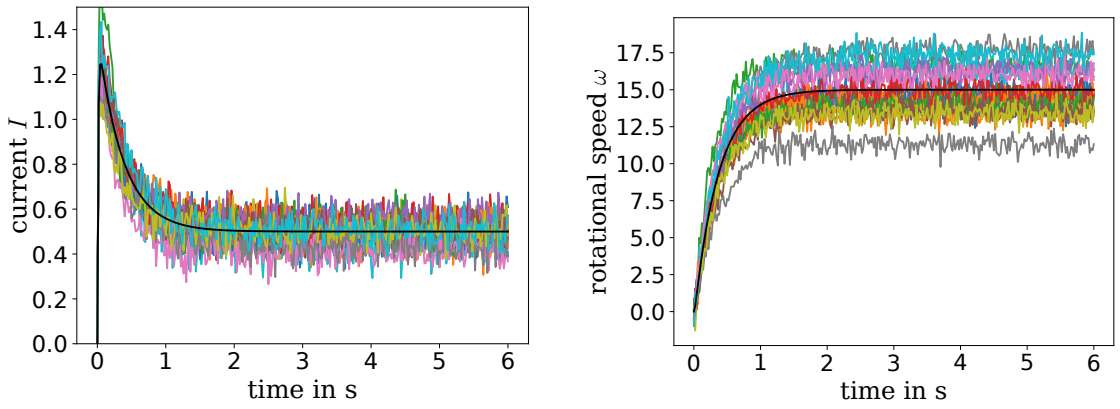


FIGURE 6.2 Twenty synthetic noisy measurements of the motor model (6.2), each color represents one time series. The black time series are $I(\mu_R, \dots, \mu_D)$ and $\omega(\mu_R, \dots, \mu_D)$ with no noise.

where $\tilde{\mu}_{\boldsymbol{X}_d} \in \mathbb{R}$, and $\tilde{\sigma}_{\boldsymbol{X}_d} \in \mathbb{R}_+$ are hyper-parameters for $d = 1, \dots 6$, and we summarize the hyper-parameters in a vector,

$$\boldsymbol{\theta} = \left(\tilde{\mu}_R, \tilde{\mu}_{c_g}, \tilde{\mu}_{c_m}, \tilde{\mu}_L, \tilde{\mu}_J, \tilde{\mu}_D, \tilde{\sigma}_R, \tilde{\sigma}_{c_g}, \tilde{\sigma}_{c_m}, \tilde{\sigma}_L, \tilde{\sigma}_J, \tilde{\sigma}_D\right). \tag{6.6}$$

Because we assumed the distributions for the parameters $\boldsymbol{X}_d$ to be independent, we approximate the desired overall distribution as the product distribution,

$$\pi(\boldsymbol{X}) = \prod_{d=1}^{6} \pi(\boldsymbol{X}_d \mid \boldsymbol{\theta}). \tag{6.7}$$

The N_t-dimensional likelihoods for each measurement $\boldsymbol{y}_i^{(\cdot)}$ are

$$\pi(\boldsymbol{y}_i^{(\cdot)} \mid \boldsymbol{x}_i, \sigma_i^{(\cdot)}) = N\left(\boldsymbol{y}_i^{(\cdot)} - \tilde{f}_{\text{ODE}}^{(\cdot)}(\boldsymbol{x}_i),\ \Sigma_i^{(\cdot)}\right), \tag{6.8}$$

where $\boldsymbol{x}_i$ is a realization of $\boldsymbol{X}$, $\Sigma_i^{(\cdot)} = \text{diag}(\sigma_i^{(\cdot)}) \in \mathbb{R}^{N_t \times N_t}$ are the noise terms along time discretization, and $\tilde{f}_{\text{ODE}}^{(\cdot)}$ are the sparse grid surrogates for f_{ODE}^{I} and f_{ODE}^{ω}, respectively. We summarize the noise terms as $\boldsymbol{\sigma} = (\sigma_i^I, \sigma_i^\omega)_{i=1}^M$ and assume the likelihood for all measurements Y to be

$$\pi(Y \mid \boldsymbol{X}, \boldsymbol{\sigma}) = \prod_{i=1}^{M} N\left(\boldsymbol{y}_i^I - \tilde{f}^I(\boldsymbol{x}_i),\ \Sigma_i^I\right)\ N\left(\boldsymbol{y}_i^\omega - \tilde{f}^\omega(\boldsymbol{x}_i),\ \Sigma_i^\omega\right). \tag{6.9}$$

We set up $\pi_0(\boldsymbol{\theta})$, the prior of the hyper-parameters, as

$$\pi_0(\tilde{\mu}_{X_d}) \sim U(0.9\ \mu_{X_d},\ 1.1\ \mu_{X_d}), \quad \pi_0(\tilde{\sigma}_{X_d}) \sim \text{HalfCauchy}(1), \quad d = 1, \dots, 6. \tag{6.10}$$

The HalfCauchy distribution is the doubled right half of the Cauchy distribution and generally recommended as a weakly informative prior for hierarchical Bayesian modeling [Gel06a; Pol12]. The measurement noises $\boldsymbol{\sigma}$ are hierarchically modeled as

$$\pi(\sigma_i^I \mid \tilde{\boldsymbol{\sigma}}) \sim N(0, \tilde{\sigma}_i^I), \quad \pi(\sigma_i^\omega \mid \tilde{\boldsymbol{\sigma}}) \sim N(0, \tilde{\sigma}_i^\omega), \quad i = 1, \dots, M, \tag{6.11}$$

with hyper-parameters $\tilde{\boldsymbol{\sigma}} := (\tilde{\sigma}_i^I, \tilde{\sigma}_i^\omega)_{i=1}^M$ and their priors $\pi_0(\tilde{\boldsymbol{\sigma}})$

$$\pi_0(\tilde{\sigma}_i^I) \sim N(0.1, 0.1), \quad \pi_0(\tilde{\sigma}_i^\omega) \sim N(0.1, 0.1), \quad i = 1, \dots, M. \tag{6.12}$$

Consequently, to approximate the posterior distribution $\pi(\boldsymbol{\theta}, \boldsymbol{\sigma} \mid Y)$, we need to sample the hyper-parameters $\boldsymbol{\theta}$, the measurement noise standard deviations $\boldsymbol{\sigma}$, and the realizations $\boldsymbol{x}_i, i = 1, \dots, M$. This is a total of $12 + 2M + 6M$ parameters to sample from. Bayes inference for such problems has already been performed multiple times [Gla16; Joh21a; Joh21b]. However, so far only two input parameters were used, while our scenario comprises six parameters.

Inference Results. The two objective quantities I and ω are each approximated with an individual surrogate created spatially adaptive with extended not-a-knot B-splines of degree 3. The surrogates are set up to be refined until they exceed 500 grid points, resulting in 539 grid points for I and 581 grid points for ω, starting from level 1 and refining 10 points per iteration. We compare Metropolis-Hastings using $N = 1\,000$ and $N = 10\,000$ to NUTS using $N = 100$ and $N = 200$ samples, both based on the above surrogates. For each sampling set-up we sample two chains, each using N tuning samples as burn-in, before actually sampling N samples.

First we look at the inferred densities, illustrated in Fig. 6.3. For most hyper-parameters, the NUTS results for $N = 200$ differ only slightly from the $N = 100$ results, indicating that already for only 100 samples the sampler has almost converged. In contrast, some of the Metropolis-Hastings densities have two peaks, indicating that the two chains did not converge against a common target, and a larger number of tuning samples would be necessary. The Metropolis-Hastings results are particularly worse than NUTS for the noise terms σ^I and σ^ω. The peaks for single chains for μ_R and μ_L can be explained by luck with respect to the random starting value, because the second chain using a different starting value did not converge noticeably towards the peak respectively.

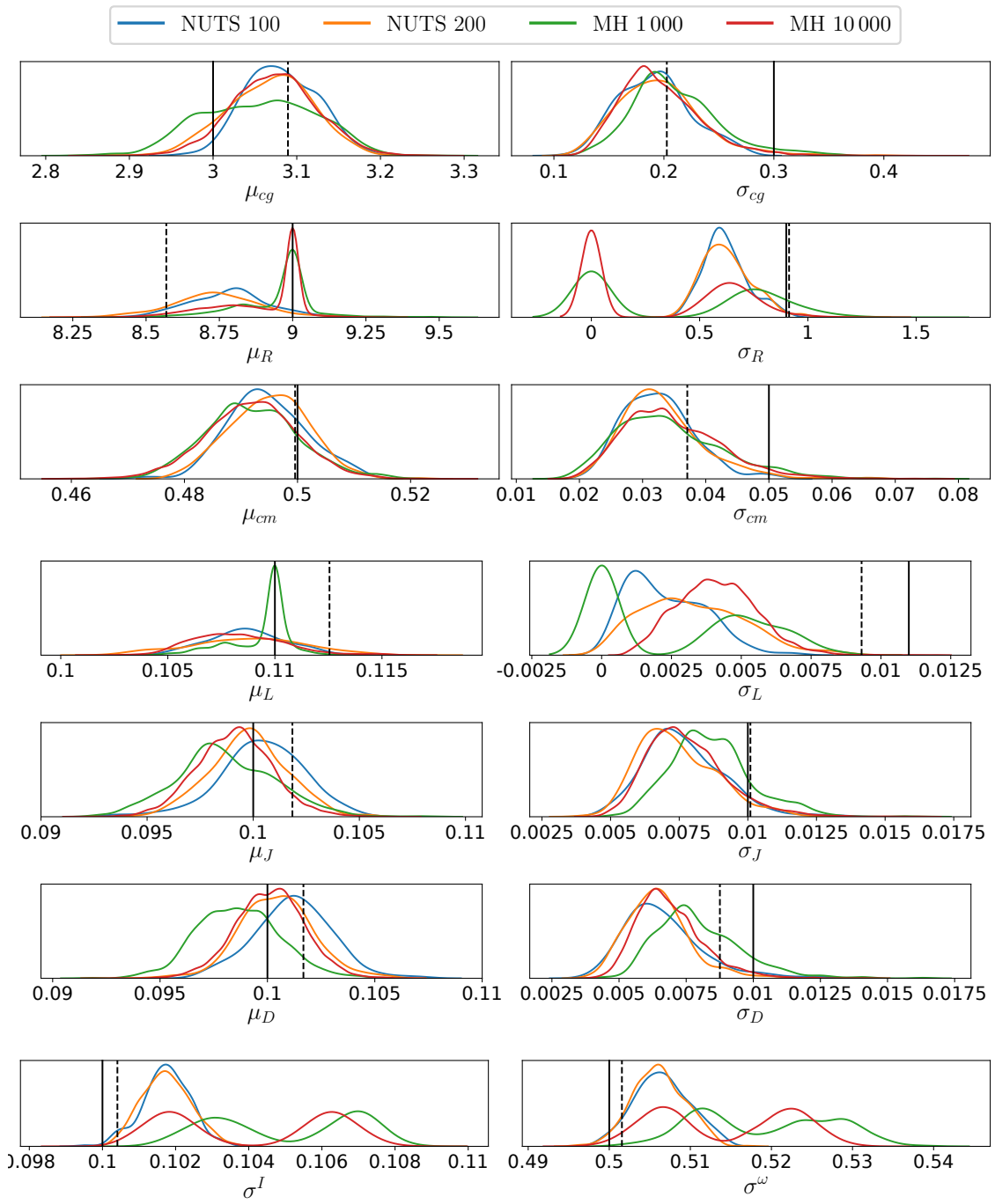


FIGURE 6.3 Gaussian kernel density estimates of the inference samples of Metropolis-Hastings and NUTS for N samples and an equal number of tuning samples. For each hyper-parameter the true maximum likelihood without noise is indicated with a solid line, the mean or standard deviation of the samples are indicated with dashed lines.

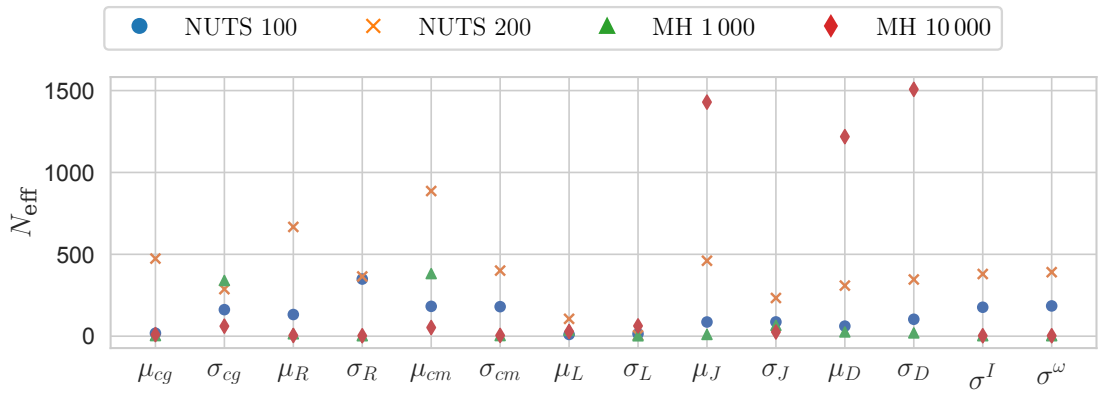


FIGURE 6.4 Number of effective samples N_{eff} of Metropolis-Hastings and NUTS for N samples and an equal number of tuning samples for the inference of the hyperparameters of the artificial motor.

Second, we look at the number of effective samples in Fig. 6.4. For Metropolis-Hastings the number of effective samples is very inconsistent across the different hyperparameters. For few of them it is very large, but for most others it is rather close to zero. NUTS conversely has a noticeable number of effective samples for almost all hyperparameters and NUTS 200 outperforms the other methods for most of them. Note, that the number of effective samples for NUTS is sometimes even larger than N, which can be explained by negative lag-t autocorrelation, cf. Sec. 4.3. Overall NUTS delivers significantly better results for a small fraction of the number of samples of Metropolis-Hastings, where the usage of NUTS became possible only through the good Jacobian approximations of our surrogate based on B-splines and sparse grids.

6.2 Motor Test-Bench and Simulation

The real motor test-bench is set up around a windshield wiper electric motor. In production these motors have varying properties due to tolerances, material uncertainties and different suppliers. The motors need to be able to cope with rapidly changing and varying load due to rain, wind or snow. Therefore, the test-bench is capable of varying the load torque T(Nm) via a magnetic powder brake, imitating such demands. Furthermore, in practice the voltage can vary significantly because of different batteries, battery age and the electric consumption of other devices. Therefore, the voltage U(V) of the test-bench is attached to a controllable power source. The quantities of interest measuring motor performance are the current I(A) and the rotational speed ω(rad/s), which is recorded at a shaft connecting a worm gear at the powder brake to the motor. For more details on

Name	Description	Distribution			Unit
U	voltage	$N(\mu_U, \sigma_U^2)$	$\mu_U = 13.5$	$\sigma_U = 0.7$	V
T	load torque	$N(\mu_T, \sigma_T^2)$	$\mu_T = 2.5$	$\sigma_T = 0.2$	Nm

TABLE 6.2 Input parameters of the black box DC motor model and the distributions used for creating the measurements. We assume the distributions to be independent.

the test-bench see [Gla16].

Real measurements were created by repeatedly running the test-bench motor for varying voltage and load torque inputs, where the two input parameters take values that are sampled from the Gaussian normal distributions specified in Tab. 6.2. The resulting raw measurements were then preprocessed. Outliers, stemming from unreliabilities of the powder break were detected and removed, the starting points of the measurements were aligned, the data was cut off after 6 seconds, filtered to remove some noise, and down-sampled to reduce data size, see [Joh21a] for more details. This preprocessing resulted in a data-set of 100 noisy measurements of current I and rotational speed ω,

$$Y := \{\boldsymbol{y}_i^I, \boldsymbol{y}_i^\omega \mid i = 1, \dots, 100\}, \tag{6.13}$$

where for a fixed index i the discrete measurement signals

$$\boldsymbol{y}_i^I := [y_i^I(t_0), \dots, y^I(t_{600})], \tag{6.14}$$

$$\boldsymbol{y}_i^\omega := [y_i^\omega(t_0), \dots, y^\omega(t_{600})], \tag{6.15}$$

are each of size $N_t := 601$ on equidistant time steps $t_0, \dots, t_{600}$ in $[0, 6]$ seconds. The measurements can be seen in Fig. 6.5.

Simulation. The simulation model of the test-bench is an updated and more complex version of the basic ODE system (6.1). A model of the worm gear is added to the mechanical part of the ODE, and a detailed thermal model interconnection of the electrical and mechanical part is introduced. Instead of assuming constant stationary temperatures, a warm up phase is modeled, where the temperature increases from an initial state to a stationary level over time. Furthermore, several linear and nonlinear dependencies in between resistance, temperature, friction, current and velocity are introduced. We do not go into detail for this model, but use a black-box implementation. The corresponding objective function is

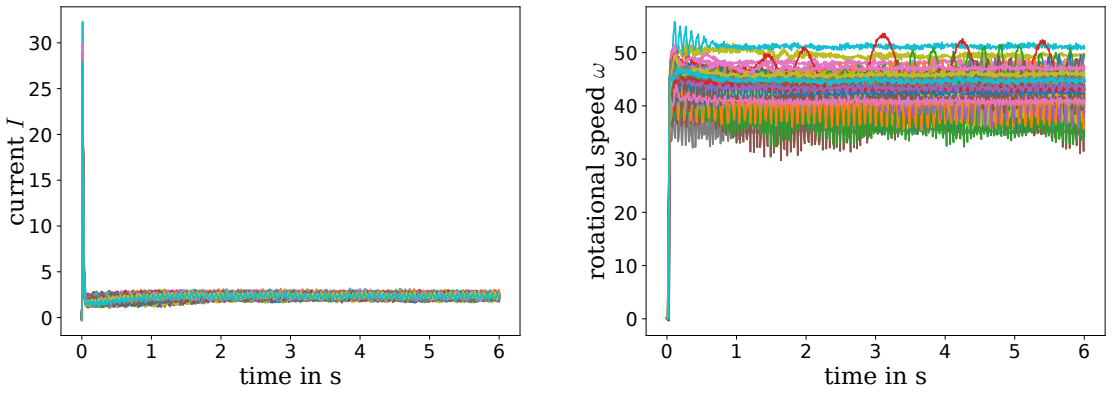


FIGURE 6.5 The 100 measurements of the motor test-bench, each color represents one time series.

$$f = (f^I, f^\omega) \colon \mathbb{R}^2 \to \mathbb{R}^{2\times N_t} \qquad U,\ T \mapsto \boldsymbol{I}, \boldsymbol{\omega}, \tag{6.16}$$

mapping the voltage U and load torque T to current $\boldsymbol{I} \coloneqq (I_0, \dots, I_{600})$ and angular velocity $\boldsymbol{\omega} \coloneqq (\omega_0, \dots, \omega_{600})$ discretized over time.

6.2.1 Surrogate Convergence

We approximate the simulation outputs (6.16) with various B-spline basis functions on regular and spatially adaptive sparse grids. The results can be seen in Fig. 6.6. The NRMSE stagnates around 10^{-5} for I and around 10^{-6} for ω, because our surrogates reach the accuracy of the simulation itself, which is limited due to tolerances of the underlying solver. All splines deliver comparable results, but not-a-knot B-splines on boundary grids converge slightly faster than the others. This is to be expected for a two-dimensional parameter space, where the number of boundary points does not dominate to such extend as in higher dimensional spaces. Most spatially adaptive sparse grids outperform their regular counterparts. In particular, for I the not-a-knot B-splines on adaptive boundary grids need less than a third of the regular grid points to reach maximum accuracy. The underlying grids of the surrogates we use for the upcoming inference are displayed in Fig. 6.7.

6.2.2 Inference for the Real Motor Model

Hierarchical Bayesian model. As in the artificial scenario in Sec. 6.1.2, we model the input parameters hierarchically. We want to infer the unknown probability distribution

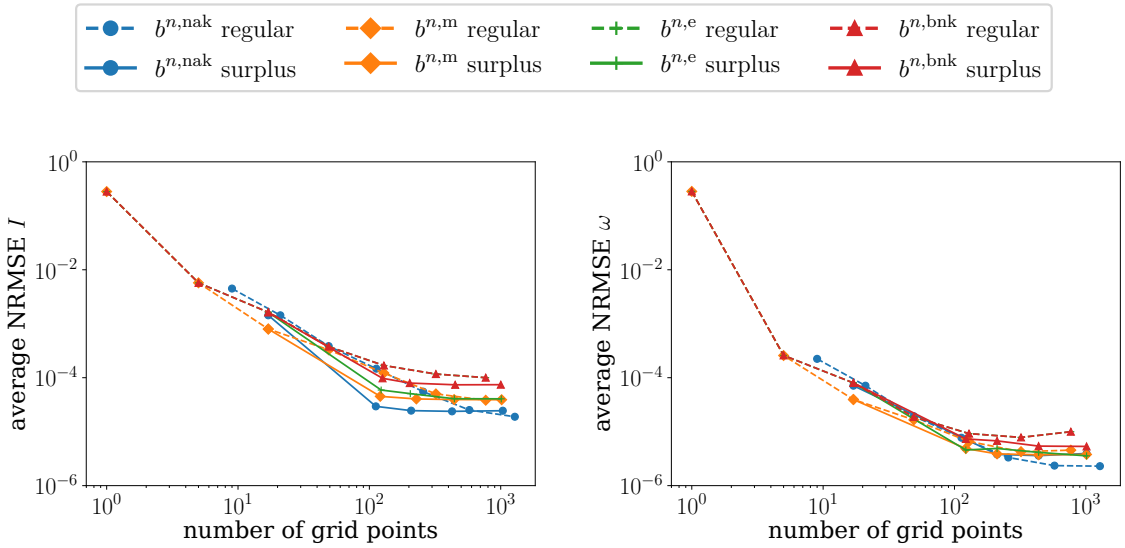


FIGURE 6.6 Approximation errors for the DC motor test-bench simulation for cubic splines on regular and spatially adaptive sparse grids. NRMSE calculated based on 1 000 reference points and averaged over time.

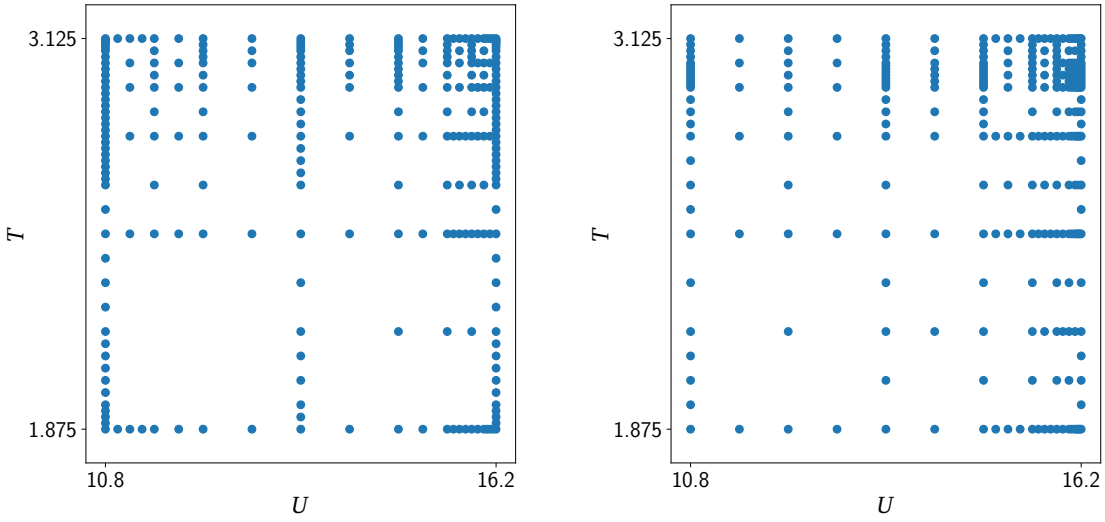


FIGURE 6.7 Spatially adaptive sparse grids for cubic not-a-knot B-splines including the boundary. For 263 grid points (I, left) and 266 grid points (ω, right) maximum accuracy is achieved.

$\pi(\boldsymbol{X})$, where now $\boldsymbol{X} = (U, T)$. For both inputs we assume normal distributions,

$$\pi(U \mid \tilde{\mu}_U, \tilde{\sigma}_U) \sim N(\tilde{\mu}_U, \tilde{\sigma}_U^2), \quad \pi(T \mid \tilde{\mu}_T, \tilde{\sigma}_T) \sim N(\tilde{\mu}_T, \tilde{\sigma}_T^2), \tag{6.17}$$

where $\tilde{\mu}_U, \tilde{\mu}_T \in \mathbb{R}$, and $\tilde{\sigma}_U, \tilde{\sigma}_T \in \mathbb{R}^+$ are hyper-parameters, which we summarize in a vector $\boldsymbol{\theta} := (\tilde{\mu}_U, \tilde{\sigma}_U, \tilde{\mu}_T, \tilde{\sigma}_T)$. We then approximate the desired distribution by

$$\pi(\boldsymbol{X}) = \pi(U \mid \boldsymbol{\theta})\ \pi(T \mid \boldsymbol{\theta}). \tag{6.18}$$

The likelihoods for each measurement $\boldsymbol{y}_i^{(\cdot)}$ are defined as in the artificial scenario,

$$\pi(\boldsymbol{y}_i^{(\cdot)} \mid \boldsymbol{x}_i, \sigma_i^{(\cdot)}) = N\left(\boldsymbol{y}_i^{(\cdot)} - \tilde{f}^{(\cdot)}(\boldsymbol{x}_i),\ \Sigma_i^{(\cdot)}\right), \tag{6.19}$$

as are the likelihoods for all measurements Y,

$$\pi(Y \mid \boldsymbol{X}, \boldsymbol{\sigma}) = \prod_{i=1}^{M} N\left(\boldsymbol{y}_i^I - \tilde{f}^I(\boldsymbol{x}_i),\ \Sigma_i^I\right)\ N\left(\boldsymbol{y}_i^\omega - \tilde{f}^\omega(\boldsymbol{x}_i),\ \Sigma_i^\omega\right). \tag{6.20}$$

We set up the prior of the hyper-parameters, $\pi_0(\boldsymbol{\theta})$, as

$$\begin{aligned} \pi_0(\tilde{\mu}_U) &\sim U(0.8\ \mu_U,\ 1.2\ \mu_U), \quad \pi_0(\tilde{\sigma}_U) \sim U(0.5\ \sigma_U,\ 1.5\ \sigma_U), \\ \pi_0(\tilde{\mu}_T) &\sim U(0.75\ \mu_T,\ 1.25\ \mu_T), \quad \pi_0(\tilde{\sigma}_T) \sim U(0.375\ \sigma_T,\ 1.625\ \sigma_T). \end{aligned} \tag{6.21}$$

The measurement noises $\boldsymbol{\sigma}$, are hierarchically approximated by

$$\pi(\sigma_i^I \mid \tilde{\boldsymbol{\sigma}}) \sim N(0, \tilde{\sigma}_i^I), \quad \pi(\sigma_i^\omega \mid \tilde{\boldsymbol{\sigma}}) \sim N(0, \tilde{\sigma}_i^\omega), \quad i = 1, \ldots, M, \tag{6.22}$$

with hyper-parameters $\tilde{\boldsymbol{\sigma}} := (\tilde{\sigma}_i^I, \tilde{\sigma}_i^\omega)_{i=1}^M$ for which we assume priors $\pi_0(\tilde{\boldsymbol{\sigma}})$

$$\pi_0(\tilde{\sigma}_i^I) \sim N(0.25, 0.1), \quad \pi_0(\tilde{\sigma}_i^\omega) \sim N(1.0, 0.1), \quad i = 1, \ldots, M. \tag{6.23}$$

Consequently, to approximate the posterior distribution $\pi(\boldsymbol{\theta}, \boldsymbol{\sigma} \mid Y)$ we need to sample the hyper-parameters $\boldsymbol{\theta}$, the measurement noise standard deviations $\boldsymbol{\sigma}$, and the realizations $\boldsymbol{x}_i, i = 1, \ldots, M$. This is a total of $4 + 2M + 2M = 404$ parameters to sample from.

The test-bench measurements and the simulation model have already been used in previous publications [Joh21b; Joh21a]. There Approximate Bayesian Computation (ABC) methods were used to approximate the posterior. We summarize the ABC approach in the next paragraph, and then compare with their results.

Approximate Bayesian Computation. The aim of ABC methods is to calculate an approximation $\hat{\pi}(\boldsymbol{X} \mid Y)$ of the true posterior $\pi(\boldsymbol{X} \mid Y)$. This is done using a distance measure between simulated values and observed data. A new sample is accepted if the distance is smaller than a specified threshold δ. The crux is to appropriately define the distance measure and the threshold. To reduce the problem's complexity and increase the acceptance rate, summary statistics are used. This means that the time series are reduced to few key values, such that instead of the distance between full time series, only the distance between the summary statistics is taken. For the results which we use in our comparison, the summary statistics of the data Y are its Monte Carlo mean and standard deviation,

$$S(Y) = (\mathbb{E}_{MC}\,[f^I],\ \mathbb{E}_{MC}\,[f^\omega],\ s_{MC}\,[f^I],\ s_{MC}\,[f^\omega]). \tag{6.24}$$

The summary statistics of the generative model are the according means and standard deviations with respect to $\pi(\boldsymbol{X} \mid \boldsymbol{\theta})$, calculated with Gauss-Hermite sparse grid quadrature. The distance measure is then the L^2-norm of the difference of the summary statistics, and δ is set to 1.5. Unfortunately the summary statistics prevent the method from inferring the noise terms σ^I, σ^W, which must be estimated a priori and then are kept constant throughout the inference. For more information on ABC we refer to [Naj16; Joh21b].

To speed up the sampling process within ABC, a polynomial chaos expansion surrogate is used. The coefficients are calculated based on a level 5 sparse grid of 181 points yielding an error of 10^{-3} for the current I and 10^{-4} for the angular velocity ω, which is both an order of magnitude larger than the errors of our sparse grid surrogates, cf. Fig. 6.6. The improved performance of our approach can be explained by the different underlying ideas of the two concepts. In contrast to PCE which approximates a function well with respect to a distribution, surplus adaptive sparse grids approximate the function itself. Because only little is known about the distributions before sampling, PCE cannot be applied effectively in this scenario.

Inference results. For the test-bench we use the same sampling strategy as in the artificial scenario, i.e., for each set-up, we sample two chains, each using a burn-in of N tuning samples before drawing the actual N samples. Bosch Research provided us with their results based on ABC, which was set up to create 1 500 samples [Joh21a; Joh21b]. By construction the ABC samples are uncorrelated, and thus it is reasonable to compare this number to the N_{eff} of NUTS and Metropolis-Hastings. In Fig. 6.8 it can be seen that because the MH samples are highly correlated, 5 000 tuning samples and equally many actual samples are required to produce a number of effective samples similar to the 1 500 ABC samples. For NUTS it only takes 500 tuning and actual samples to reach a

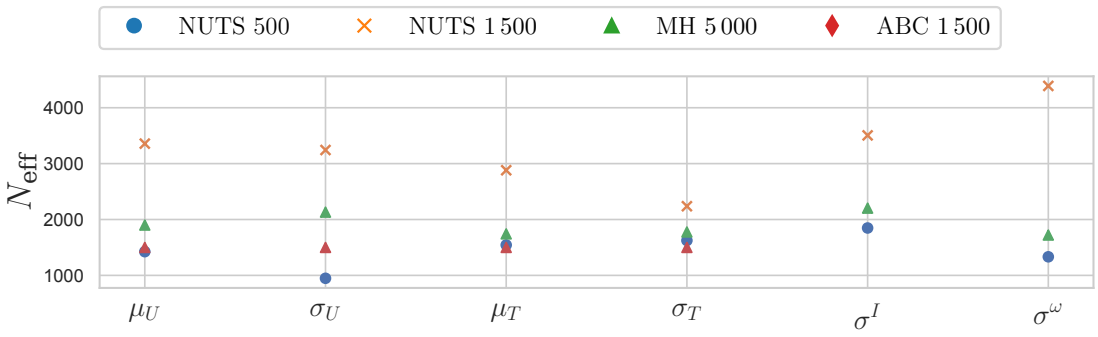


FIGURE 6.8 Number of effective samples for inference of the test-bench hyper-parameters. ABC cannot infer the noise terms σ^I and σ^ω.

comparable number of effective samples. Just like in the artificial scenario, the negative autocorrelation mentioned in Sec. 4.3 make NUTS highly effective. In fact, when creating 1 500 samples with NUTS, their approximation quality is significantly beyond that of ABC.

The inferred posterior distributions can be seen in Fig. 6.9. Because the number of samples was chosen such that their N_{eff} is roughly the same, NUTS 500 and MH 5 000 perform very comparable. This number of samples is indeed already sufficient to accurately represent the posterior, as can be seen from the NUTS 1 500 results. While the number of effective samples is significantly higher, it only confirms the results of the previous two, and does not add much new information to the posterior estimates. The ABC results however differ notably from the others. For μ_U and μ_T the MH and NUTS results are focused close to the true values. ABC's posterior is much broader, and while it does capture the true value, it is much less certain. For σ_U and σ_T ABC is a bit more focused than the other methods. As mentioned in Sec. 6.2.2, ABC cannot infer the noise hyper-parameters σ^I and σ^ω, but must use a-priori estimations, which are indicated as Dirac distributions in the plots. Creating artificial data and perturbing it using the different noise estimates from Fig. 6.9 shows that NUTS and MH are close to the truth, but ABC significantly underestimates the noise. This less accurate noise estimation directly influences all inference results, and likely explains the broader posteriors for μ_U and μ_T.

Run-times. We intentionally postpone a detailed discussion of run-times for this section. While an evaluation of the sparse grid surrogate is about 10 000 times faster than an evaluation of the black-box simulation, averaged over 1 000 evaluations each, our choice of pymc3 as sampling library comes with a major drawback. The objective function must be handed to pymc3 in form of a theano [The16] operator and wrapping the SG^{++}

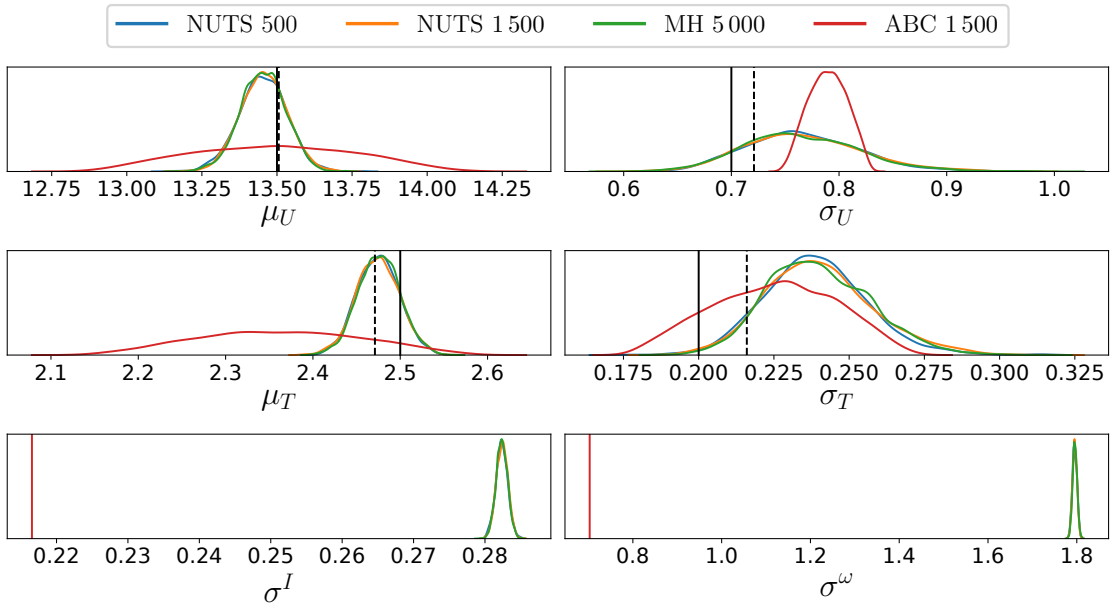


FIGURE 6.9 Gaussian kernel density estimates of the inference samples of Metropolis-Hastings, NUTS, and ABC. For each hyper-parameter the underlying maximum likelihood without noise is indicated with a solid line, the mean or standard deviation of the samples are indicated with dashed lines. For the noise terms σ^I, σ^ω no true information is available. For ABC the noise terms are not inferred but estimated a-priori and thus indicated here as Dirac distribution.

evaluation for theano slows down the overall evaluation significantly. The same holds for evaluations of the gradients. For M measurements, NUTS and MH both require $2M$ evaluations of the surrogate, but NUTS requires an additional $2M$ gradient evaluations. Consequently, the slow-down from the wrapper affects NUTS much worse, and so in fact NUTS and MH both require roughly the same amount of time to reach a similar number of effective samples, although NUTS requires considerably less total samples.

Summary. This chapter shows a first investigation regarding the combined potential of sparse grids, B-splines and the NUTS algorithm for parameter calibration. For this we demonstrated that the surrogate derivatives are accurate enough to allow using the NUTS algorithm for a complex black-box function. Furthermore, we showed that NUTS requires significantly fewer samples than Metropolis-Hastings to reach a similar accuracy. While the wrapper slow-down prevents a comparable superiority of NUTS with respect to run-time, a re-implementation using non-wrapped NUTS, and preferably in C/C++ instead of Python, would significantly outperform MH and allow a fair run-time comparison with ABC. Also, we only used a two-dimensional set-up for the real test-bench. This allowed comparability because all work on the test-bench so far has been done for similar set-ups.

However, as we have shown in Sec. 6.1 our approach can handle larger input parameter spaces too.

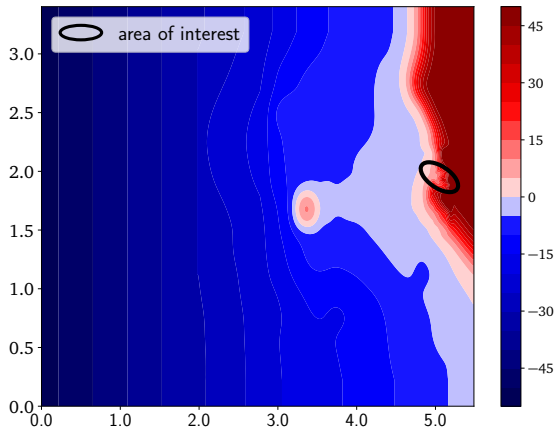
7

Application III
Forward UQ for a Tsunami Simulation

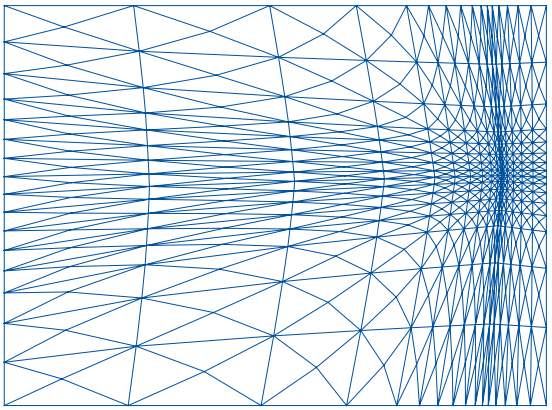
> *A Powerful Wave*
> *Relentless Wall of Water*
> *Kept Moving Inland.*
>
> — Haiku by R. Gasperson [Gas]

In this chapter we conduct a comprehensive forward UQ study to gain a better understanding for a tsunami wave simulation. Starting from records of an actual tsunami wave, we vary the shape of the incoming wave, and observe the resulting impact on the run-up at the coast. The underlying wave simulation is computationally very expensive, and more than a handful of direct calculations are infeasible. Therefore, a surrogate is required. Unfortunately, because of shocks in the underlying solution, the smoothness of the objective is only of first order. We still use higher-order basis functions for the surrogate, because they provide gradients which allow us to optimize the objective. Moreover, although the approximation error converges at a limited rate, the surrogate reaches the accuracy of the simulation itself. We compare our B-splines on sparse grids with the UQ library DAKOTA, and our approach reaches slightly better results for all quantities of interest.

This chapter is based on a collaboration with Prof. Dr. Stephen Roberts (Australian National University, Canberra, Australia). The presented results have already been published



A Bathymetry (m) of the simulation domain (km), including a part of the coast of Okushiri, the Monai valley (indicated with an ellipse), and Muen island. The tsunami wave is coming in from the western boundary on the left.



B Underlying triangulation of the simulation domain with 16×16 triangles for illustrational purpose. In practice 64×64 triangles were used, but they would not be visible at this scale. The triangulation is refined around Monai valley.

FIGURE 7.1

in [Reh21c]. The collaborator provided the underlying tsunami wave simulation, which is based on previous work [Baa17]. The author of this thesis adjusted the parametrization of the waves, prepared the simulation data, created the surrogate, and conducted the forward UQ study.

7.1 The Tsunami Model

The Hokkaido-Nansei-Oki Tsunami. On the 12th of July 1993 an earthquake near the island Hokkaido west of the Japanese mainland caused the Hokkaido-Nansei-Oki tsunami. While run-ups reached up to 3.5 meters on the Japanese mainland, in the Monai valley on Okushiri island the run-up height reached almost 30 meters. It was suggested that Muen island, located in front of the Monai valley might have caused this through a lens effect. An experimental study was conducted to investigate the lens effect hypothesis and a 1/400 scale model of the region was constructed in form of a 205m long water tank [Mat01]. The bathymetry of the domain, Muen island and the Monai valley can be seen in Fig. 7.1A. The experiments reproduced the run-ups and found them to be independent of the presence of Muen island. Still, following this thorough investigation, accurate experimental data is now available, and the setting is used as a benchmark for the simulation of tsunamis [Nie05; Syn07].

Benchmark implementation. The experimental data is used to model the benchmark tsunami in ANUGA [Rob15], a fluid dynamics simulation framework of the Australian National University and Geoscience Australia, which uses the finite-volume method to solve the non-linear shallow water equations. Inside ANUGA, the domain is discretized as an unstructured mesh of triangular cells. The governing equations are solved within each cell to track water depth and horizontal momentum over time. The simulation is initialized with the domain's geometry, the initial water level, and the incoming tsunami wave in form of a boundary condition. This initial incoming wave is then propagated through the domain. The resulting quantity of interest is the time-dependent average height of the water layer measured on top of Monai valley. This simulation with ANUGA is based on previous work [Baa17], where it was shown that a multi-fidelity combination technique surrogate of mixed resolutions converges towards a high resolution combination technique surrogate. In contrast to this, our investigation focuses on the convergence of a spatially adaptive fixed-resolution surrogate towards the actual simulation, as well as a thorough forward UQ analysis. We discretize the computational domain by 64×64 adaptive triangles, which is illustrated in Fig. 7.1B. The mesh is refined around Monai valley to capture the height of the water layer most accurately in the area of interest. The underlying code is publicly available [Hno20].

7.1.1 Wave Shape Parametrization

We assume that the total energy of the tsunami wave is known, but its actual shape is uncertain. Based on this we investigate how the shape of the incoming tsunami waves influences the run-up. The varying wave shapes are based on the original tsunami wave shape that was captured in the experiments. We decompose this original wave into Gaussian bumps and vary the shape of each bump, forming new wave shapes related to the original. Finally, the energy of the new wave is normalized to the energy of the original wave. This parametrization is a continuation of previous work by de Baar et al. [Baa17].

Formally, let $W(t)$ be the original incoming wave, depending on time $t \in [0, 22.5]$ in seconds. We define the initial residual $r_0(t) := W(t)$ and then iteratively fit the largest possible Gaussian bump. The d-th Gaussian bump is described by its height η_d, its width τ_d and its center ν_d, which are given by

$$\eta_d, \tau_d, \nu_d := \operatorname{argmin}_{\eta,\tau,\nu} \int_0^{22.5} \left| r_{d-1}(t) - \eta \exp\left(-\frac{(t-\tau)^2}{2\nu^2}\right)\right|^2 dt. \tag{7.1}$$

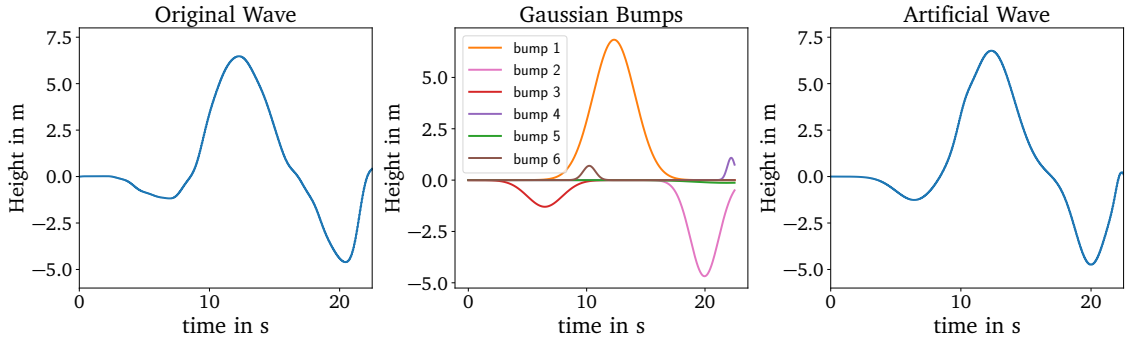


FIGURE 7.2 The original wave W (left) is decomposed into $D = 6$ Gaussian bumps (center), which form the artificial wave W_p^n (right), here with $\boldsymbol{\theta} = \mathbf{1}$. To create different wave shapes, the height of the d-th Gaussian bump is multiplied by the wave shape parameter $\theta_d \in [0.5, 1.5]$, then the energy of the wave is normalized.

The according d-th residual is then defined as

$$r_d(t) := r_{d-1}(t) - \eta_d \exp\left(-\frac{(t-\tau_d)^2}{2\nu_d^2}\right). \tag{7.2}$$

This process is iterated for $d = 1, \ldots, D$, until the total number of D Gaussian bumps is reached. The new incoming wave W_p is then parameterized as

$$W_p(t, \boldsymbol{\theta}) := \sum_{d=1}^{D} \theta_d \eta_d \exp\left(-\frac{(t-\tau_d)^2}{2\nu_d^2}\right), \tag{7.3}$$

where $\boldsymbol{\theta} := [\theta_1, \ldots, \theta_D] \in [0.5, 1.5]^D$ are uncertain wave shape parameters. The d-th parameter θ_d determines an increased or decreased contribution of the d-th Gaussian bump, depending on whether $\theta_d < 1$, or $1 < \theta_d$. Finally, we normalize the new wave W_p, by rescaling it such that is holds the same amount of energy as the original wave W,

$$W_p^n(t, \boldsymbol{\theta}) := W_p(t, \boldsymbol{\theta}) \frac{\|W(t)\|_2}{\|W_p(t, \boldsymbol{\theta})\|_2}, \tag{7.4}$$

where $\|\cdot\|_2$ is the L^2-norm with respect to time t. For the experiments we use $D = 6$ Gaussian bumps, for which the average L^2-difference over time between the original incoming wave and the artificial wave is only 0.51mm. See Fig. 7.2 for an illustration of the wave decomposition.

Inside the ANUGA simulation, the time domain $[0, 22.5]$ is discretized with 451 uniform time steps $t_m := m \frac{22.5}{450}$, $m = 0, \ldots, 450$. Our objective function is therefore the

mapping

$$f : \mathbb{R}^D \to \mathbb{R}^{451}, \quad \boldsymbol{\theta} \mapsto \big(f_0(\boldsymbol{\theta}), \ldots, f_{450}(\boldsymbol{\theta})\big)^T, \tag{7.5}$$

where $f_m(\boldsymbol{\theta}) = W_p^n(t_m, \boldsymbol{\theta})$.

We assume that the shape parameters $\boldsymbol{\theta}$ are i.i.d. and follow a truncated normal distribution centered at the original wave, i.e., $\theta_d \sim N(\mu, \sigma)$ with mean $\mu = 1.0$ and standard deviation $\sigma = 0.125$. We truncate this distribution to the interval $[0.5, 1.5]$ which contains more than 99,99% of the distributions mass.

The aim of our study is to understand how the uncertainty in the wave shape parameters $\boldsymbol{\theta}$ influences the height of the tsunami run-ups. A single evaluation of the ANUGA tsunami simulation requires one hour[1], therefore we replace the simulation by a surrogate on which we then perform the UQ analysis.

7.2 Numerical Results

The underlying objective function f from Eq. (7.5) corresponds to a 1 : 400 scale model of the real tsunami wave. To make the results of this chapter more intuitively understandable, we scale them by a factor of 400, resulting in values which correspond to actual real tsunami heights in meters. For comparison and verification we use the UQ toolbox DAKOTA [Dak14]. The major advantage of our approach, implemented in SG^{++}, over DAKOTA is that once we have prepared the surrogate, all the following quantities can directly be accessed and no further evaluation of the expensive objective function is necessary. DAKOTA uses polynomial chaos expansion (cf Sec. 4.2.2) for the approximation of the objective function and for the calculation of stochastic moments. It does however not allow optimization based on the PCE surrogate. Instead, the objective function must be evaluated again.

7.2.1 Accuracy of the Surrogate

We measure the approximation error for all methods based on the same set of 10 000 reference wave shape parameters, randomly sampled following the joint parameter distribution. The average L^2-error over time can be seen in Fig. 7.3. All non-boundary sparse grid approaches perform alike. This is the same effect we have seen in Sec. 3.3.1,

[1] Raijin supercomputer at NCI Australia, a Fujitsu Primergy-Lenovo NextScale system. Evaluations were distributed, whereby each individual simulation was performed using one node with 2 × 8 core Intel Xeon E5-2670 (Sandy Bridge) 2.6GHz

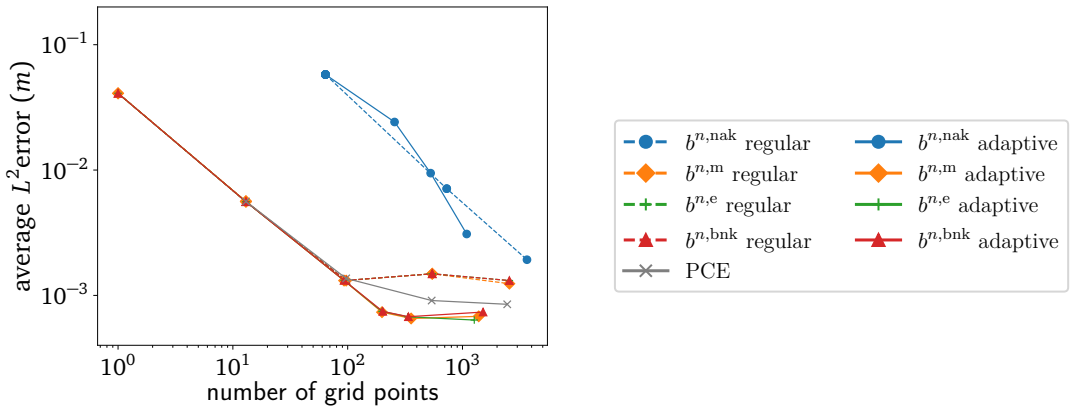


FIGURE 7.3 Average L^2-error in meters for not-a-knot boundary B-splines, modified B-splines, extended not-a-knot B-splines, and boundaryless not-a-knot B-splines on regular and spatially adaptive Sparse Grids, and polynomial chaos expansion.

where insufficient smoothness of the objective prevents higher-order basis functions to reach their full potential. Here, because of shocks in the underlying solution, the tsunami simulation itself is only first order accurate, hence higher-order B-splines and the PCE do not converge with optimal order. Nevertheless, we are using cubic splines, which although not reaching optimal order, still slightly outperform linear ones. What is even more important is that they provide gradients, which we make use of in order to optimize the surrogate in the next section.

We refine each surrogate until it exceeds 1 000 grid points. The boundary sparse grid spends too many grid points on the boundary and too few inside the domain and thus has the largest approximation error. The B-splines on non-boundary sparse grids perform better, and spatial adaptivity lets their error fall below that of the PCE, thus showing the best overall approximation quality. All methods stagnate at an average L^2-error of around 10^{-3} meters, i.e., one millimeter. This stagnation is caused by small inaccuracies of the underlying parametrization. As we mentioned in the previous section, the six-dimensional decomposition into Gaussian bumps represents the original wave only up to an error in the order of a millimeter. The error is then propagated through the simulation and prevents a more accurate surrogate. This also explains well the only slight improvement from spatial adaptivity. Just when the domain is sufficiently scanned to actually make use of the adaptivity, maximum accuracy is already reached.

7.2.2 UQ Analysis

By propagating the uncertainty of the wave shape parameters through the simulation, we can better understand how run-ups behave typically and in extreme cases. For this forward uncertainty quantification we calculate three stochastic key quantities. As all B-spline variants on non-boundary sparse grids perform similarly, we exemplarily show only the UQ results for extended not-a-knot B-splines. They are summarized in Fig. 7.4A.

Percentiles. The evaluation of our surrogate is computationally cheap, and we make use of this to propagate the uncertainty in the simplest manner. We evaluate the surrogate on a large parameter set following the according distribution to quantify the variation of the outcomes. Using a parameter set of 100 000 samples drawn from the distribution, we derive the 5-th and 95-th percentile component-wise with respect to time. These percentiles are defined as the values below which 5% or 95% of the values can be found respectively.

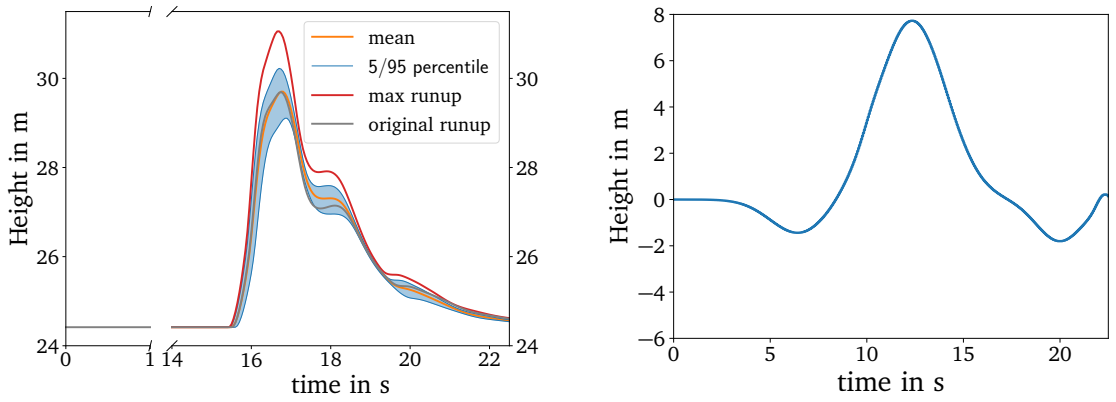
Expected value. The second UQ key quantity we compute is the expected value, which for time-dependent functions is defined component-wise as in Eq. (4.13),

$$\mathbb{E}[f] = \big(\mathbb{E}[f_0], \ldots, \mathbb{E}[f_{450}]\big). \tag{7.6}$$

We compute the mean using stochastic collocation to integrate the surrogate with respect to the probability distribution of the wave shape parameters. DAKOTA's PCE can return the mean immediately without further computations following Eq. (4.20). We measure the accuracy of the calculated means using a reference solution based on a boundaryless not-a-knot B-spline surrogate of degree $n = 3$ on a regular sparse grid of level 6 with 10 625 grid points. Our adaptive surrogate based on 1 267 grid points has an average L^2-difference of 0.0028 m to the reference mean and its worst deviation for any time step is 0.017 m. This is slightly more accurate than the DAKOTA PCE which uses 2 465 function evaluations to approximate the mean with an average L^2-difference of 0.0030 m and worst deviation over all time steps of 0.019 m.

Shape optimization. An important question that arises in the context of our tsunami simulation is which wave shape leads to the largest run-up and thus to the most destructive tsunami wave. To determine the shape, we have to optimize the objective function over time,

$$\boldsymbol{\theta}^{\max} := \arg\max_{\boldsymbol{\theta}} (\max_{m} \tilde{f}_m(\boldsymbol{\theta})). \tag{7.7}$$



A UQ analysis of the tsunami model resulting in percentiles, mean and the overall maximum.

B Shape of the optimal incoming wave $\boldsymbol{\theta}^{\max}$ resulting in the maximum run-up of 31.06m.

FIGURE 7.4 Summary of the forward UQ analysis.

Because we used cubic splines for our surrogate, we have access to the surrogate's gradients. This allows us to optimize the surrogate using the gradient descent algorithm following [Val19]. We start at $\boldsymbol{\theta}^0 := [\theta_1^0, \dots, \theta_D^0]$, the grid point with maximum function value. Then, in each iteration step, the gradient descent algorithm chooses a search direction s_j oriented like the normalized gradient, to determine $\boldsymbol{\theta}^{j+1} := [\theta_1^{j+1}, \dots, \theta_D^{j+1}]$,

$$\boldsymbol{\theta}^{j+1} := \boldsymbol{\theta}^j + \delta_j \frac{\nabla \tilde{f}(\boldsymbol{\theta}^j)}{\|\nabla \tilde{f}(\boldsymbol{\theta}^j)\|_2}. \tag{7.8}$$

The step size δ_j is determined using the Armijo line search algorithm [Noc06]. The optimization of our surrogate results in a maximum height of 30.92m. This run-up value is larger than the value of any of the 100 000 evaluations we used to calculate the percentiles. It turns out that the optimal configuration lies on the boundary of the domain, and therefore, for this quantity of interest, the boundary not-a-knot B-splines perform even better resulting in

$$\begin{aligned} \boldsymbol{\theta}^{\max} &= [1.5,\ 0.5,\ 1.5,\ 0.671875,\ 0.5,\ 0.5], \\ \max_m \tilde{f}_m(\boldsymbol{\theta}^{\max}) &= 31.05672\text{m}. \end{aligned} \tag{7.9}$$

The optimal incoming wave can be seen in Fig. 7.4B. The DAKOTA toolbox does not offer optimization based on its PCE surrogate. Instead, it uses the mesh adaptive direct search algorithm (MADS) [Aud06] based on quadratic polynomials, which requires another 2 500 evaluations of the objective function. This is a significant drawback in comparison to SG^{++}, which performs all operations on the same initially constructed

surrogate. DAKOTA's optimization then results in shape parameters corresponding to a run-up of 31.0206m, which is a little smaller than our result with boundary.

Summary. All three key quantities of the comprehensive forward UQ analysis are summarized in Fig. 7.4A. This image gives a good overview of how different incoming wave shapes can influence the tsunami run-up behavior. Using non-boundary sparse grids we were able to represent the expensive tsunami simulation up to its own numerical accuracy using only around 100 evaluations. We then calculated percentiles, the expected value, and optimized the surrogate using the differentiability of the B-spline bases. For all these key quantities our results are slightly better than those of the industry standard UQ library DAKOTA. In particular, our approach does not require an expensive re-sampling for optimization, but simply uses the already prepared surrogate.

8

Application IV
Pool Testing for COVID-19

> *Testing is our way out.*
>
> — Paul Romer [Rom20])

8.1 Introduction

In the beginning of the year 2020 the coronavirus SARS-CoV-2, which causes the COVID-19 disease, spread around the world. It was declared a pandemic by the World Health Organization (WHO) on March 11, 2020 [Cuc20]. The virus is highly infectious, and until submission of this work caused at least 128, 235, 377 infections worldwide, of which more than 2, 804, 036 people died [JHU]. One of the characteristics of the virus are people being contagious without the appearance of symptoms. This explains in part the exponential growth in the number of infections in the early phase of the pandemic, where infections were passed on unnoticed. Rapid and repeated testing of the population became key to the containment of the disease [McC20; Pet20a; Pet20b]. Massive testing allows targeted countermeasures for clusters of infection, and early interruption of infection chains. The speed of testing is crucial, but among others it is limited by supply of and access to diagnostic tests and laboratory capacities, which make a regular periodical population wide screenings of a countries' population using individual tests unfeasible.

One way to significantly increase testing capacities is pooled testing. This approach

dates back to the second world war, when Dorfman first applied it for syphilis testing [Dor43]. Today it is regularly applied in many fields such as HIV testing [Pil04; Pil05] or blood screening [Seo15]. The idea of pool testing is to group the samples of $k \in \mathbb{N}$ individuals into one joint pool. Then only a single test is used for the whole pool. If, on the one hand, the result is negative, all k samples are declared negative and $k-1$ tests have been saved. If, on the other hand, the result is positive further testing is required to divide the group into good and infected samples. This follow-up testing can be done by a variety of different approaches.

After first studies demonstrated that the diagnostic tests used for COVID-19 are sensitive enough to allow pooled testing [Yel20; Hog20], the author of this thesis collaborated with two research groups to create a decision-making-tool for comparison of pool testing approaches. The results were published in a study, which showed that testing strategies based on pooling increase the speed and throughput of screening for COVID-19 significantly [Wol20], and a provisional patent application for the method [Pfl20] was filed. The tool provides practical guidance to public health authorities and governments on how to best deploy limited testing resources to maximize the number of people tested in the shortest amount of time possible. Common pool testing methods are compared with respect to COVID-19 specifics and multiple quantities of interest. Efficiency is measured in identified cases per test, given a limited testing capacity. Furthermore, an interactive website [Cov20] allows the specification of relevant parameters resulting in individual recommendations. The available computing resources for the website are limited and only little waiting time is tolerable for users. Therefore, it is impossible to run multiple hours of full scale simulations, and instead a response surface based on B-splines and sparse grids has been realized.

In this chapter we summarize this work on pool testing for COVID-19. In Sec. 8.2 we introduce the different pool testing strategies, explain the modeling approach and present the results of the study. Then, in Sec. 8.3 we create B-spline and sparse grid based response surfaces, which replace the compute-intensive simulation with an error of less than 1% with respect to stochastic fluctuations.

The results in this chapter have already been published in [Wol20]. They were obtained in a collaboration with Prof. Dr. Timo de Wolff and Janin Heuer (both University of Braunschweig, Germany), and Dr. Martin-Immanuel Bittner (Arctoris, Oxford, United Kingdom). The collaborators contributed medical expertise, parts of the simulation framework, conceptualization, and validation. The author of this thesis contributed the major part of the simulation framework and the response surfaces for the website. In this thesis we add a new discussion on the accuracy of the response surfaces and the stochastic fluctuations.

8.2 Simulation of Pool-Based Testing

The study uses a simulation approach to identify the most effective testing strategy. It is based on a newly developed general framework for the comparison of mass testing procedures for infection with COVID-19. The underlying code is publicly available [Cod20], and can easily be adapted to other scenarios and diseases.

The comparison includes the following six pool testing strategies, which are illustrated in Fig. 8.1.

Individual testing. Each sample gets tested individually. This is the commonly used approach which serves as the baseline for our comparison.

Two level pooling. If a pool is tested positive, all contained samples are tested individually. This is the original pooling strategy proposed by Dorfman in 1943 [Dor43].

Binary splitting. A binary tree approach. If a pool of size k is tested positive, it is split into two pools of size $k/2$, and a pooled test is performed on each of the two new sets. The procedure is repeated recursively until each sample is either identified to be positive or negative [Sob59].

Recursive binary splitting (RBS). A recent improvement to the binary splitting algorithm that tries to trade off between the total number of tests and the recursive depth [Che15]. If a pool tests positive at one level of the tree hierarchy, then a single positive case is identified via binary search and removed from the pool. The method is called recursively with the remaining samples. See Appendix B.2 for more details.

Purim. A matrix-based pooling approach [Far20]. Individuals are represented as entries of a matrix. Columns and rows of the matrix form groups which are tested in a first testing round. In a second round cross-sections of positively tested rows and columns are tested individually.

Sobel R-1. A decision tree approach based on the assumption of a binomial distribution of the test results. Pool sizes are adapted according to the minimization of the expected number of remaining tests [Sob59]. The method is stochastically optimal if the infection rate is known and can be seen as the upper bound for the efficiency of pool testing strategies. See Appendix B.2 for more details.

Using the above six strategies we cover all fundamental principles of group testing in a representative comparison. While there are many more pool testing strategies in general,

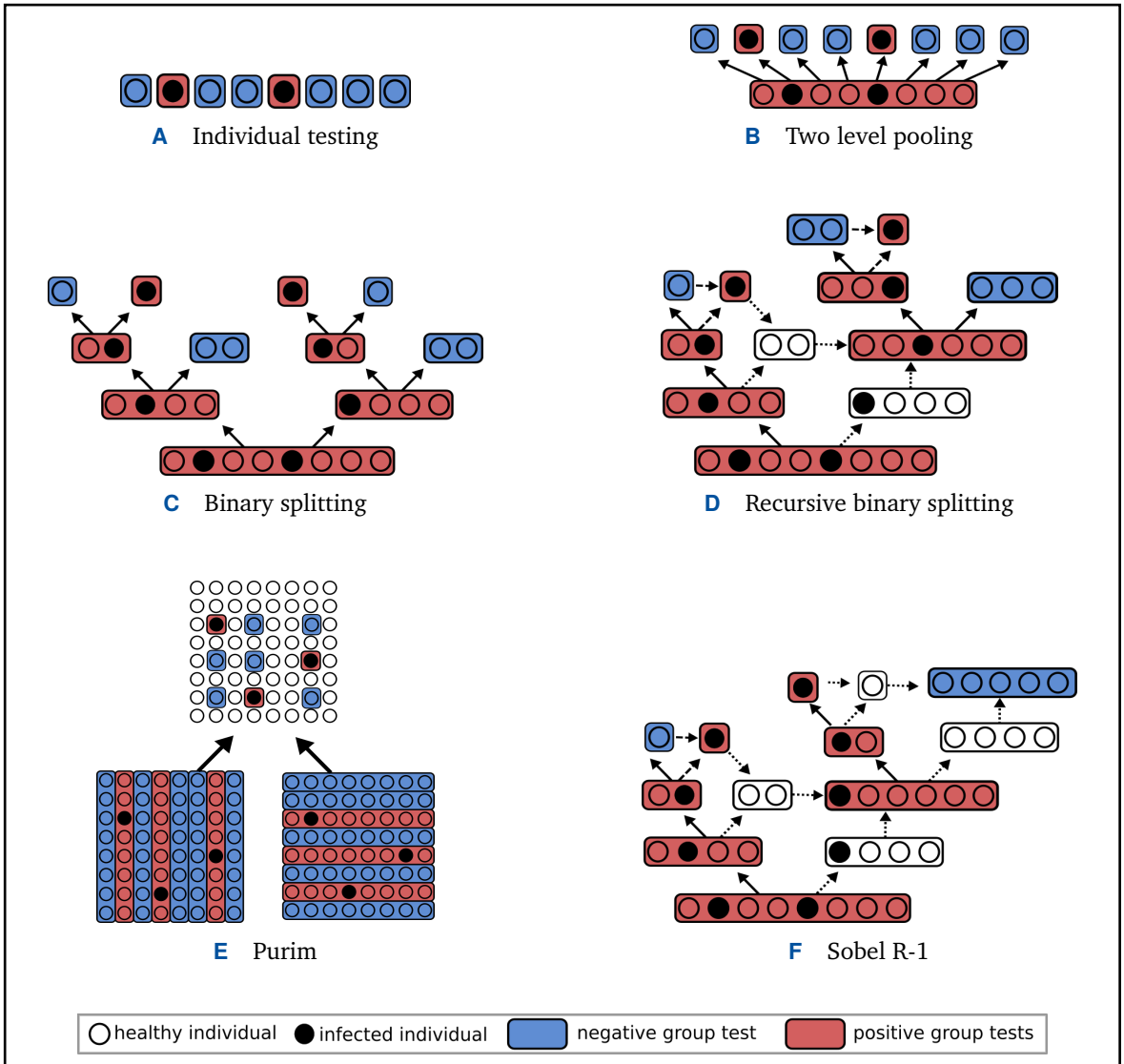


FIGURE 8.1 Illustrations of the six compared pool testing approaches based on [Wol20].

most of them can be interpreted as modifications of at least one of the above methods [Du00].

Model. We model the testing procedure with seven simulation parameters and one meta parameter. The inputs and outputs of the simulation are summarized in Tab. 8.1. Inside the simulation all members of the given population are represented individually, and each is infected with the probability of the given infection rate I. Then the pool testing

Inputs	Description	Range	Unit	Outputs
I	infection rate	$[0, 100]$	%	time to test population
s_1	test sensitivity	$[0, 100]$	%	number of used tests
s_2	test specificity	$[0, 100]$	%	number true positives
k	pool size	$[1, 32]$	individuals	number false positives
P	population size	$[1, \infty)$	individuals	identified cases per test
c	daily test capacity	$[1, \infty)$	samples	
T	test duration	$[1, \infty)$	h	
r	simulation runs	$[1, \infty)$	repetitions	

TABLE 8.1 Input parameters for the pool testing simulation, and the according input ranges. The outputs are returned as expected values and standard deviations over r simulation runs.

strategies are applied, whereby the test sensitivity s_1 and specificity s_2 are determined by the user. We allow pools of up to $k = 32$ individuals, following clinical studies which have shown reliable results for up to this size [Yel20]. The simulation uses time steps of the length of a given test duration T. Because each individual is infected with the given probability, and thus the distribution of infected individuals over the whole population is random, there is stochastic uncertainty present. To account for this, the whole simulation is repeated r times, and the final outputs are the means and standard deviations calculated from the repetitions.

Optimal pool size. There is a major correlation between infection rate and group size. For small infection rates most pools are tested negative. Thus, the pooling methods are most efficient when the pools are as large as possible. However, when the infection rate increases smaller pools become more efficient until the optimal pool size is one, and the method resembles individual testing. Therefore, we optimize the group size for the given infection rate and each method. This has been done before for specific testing strategies [Han20; Xio19], but we do so in a comparative benchmark. Because of the long run-times of the actual simulation, we create one response surface for each testing method based on cubic boundary not-a-knot B-splines and spatially adaptive sparse grids. From this we tabulate the expected time to test a reference population of $P = 100\,000$ individuals given the infection rate for all pool sizes $k \in \{1, \dots, 32\}$, see Fig. 8.2. The simulation then looks up the optimal pool size for each method and infection rate.

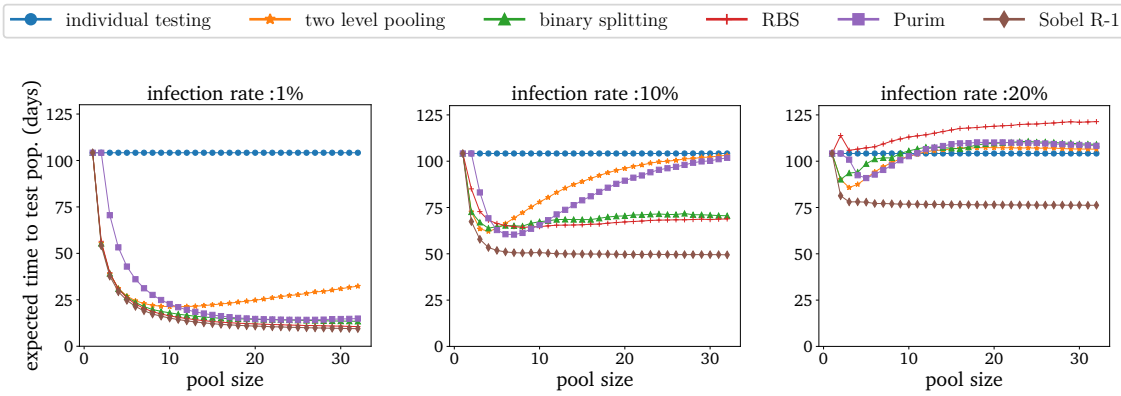
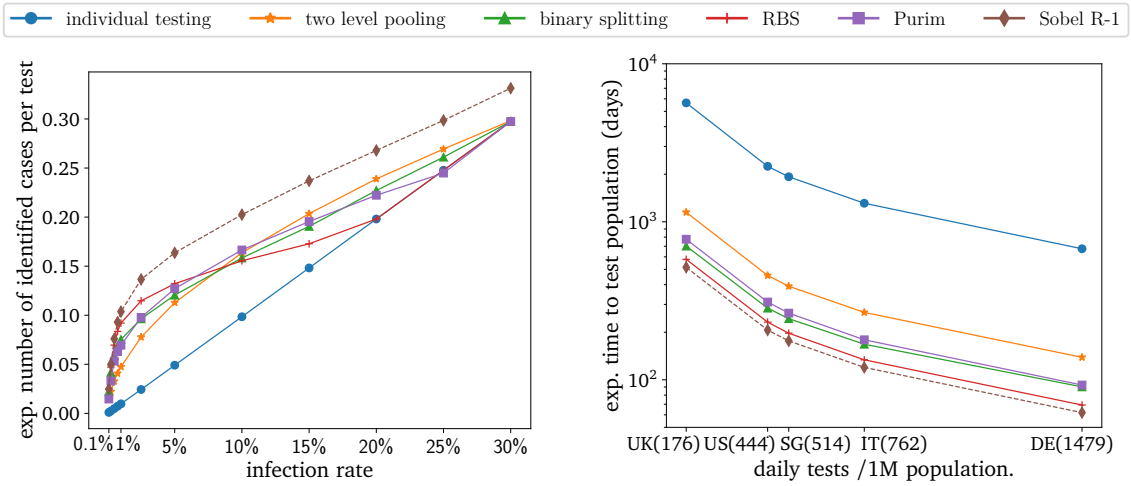


FIGURE 8.2 Expected time to test the population for pool sizes $k \in \{1, \ldots, 32\}$ and infection rates of 1%, 10%, and 20%. The results are obtained from a response surface. From this the optimal pool sizes are determined. Sobel R-1 internally optimizes the pool size, where the initial pool size only serves as an upper bound. Therefore, large initial pool sizes do not impair its efficiency.

8.2.1 Results

Our simulations show that for infection rates of up to at least 20% all pooled testing strategies are significantly more efficient in terms of the expected number of identified sick individuals per test than individual testing, see Fig. 8.3A. This can be explained by the fact that for small infection rates most pools are tested negative, and optimal pool sizes are large. Pooled methods can then rule out most groups using only a single test. Turning this argument around for large infection rates, when many pools are positive, pool testing could result in extra work compared to individual testing if pool sizes are chosen too large. However, because we optimize the pool size, this effect does not occur in our simulations. Instead, for each pooling approach, there is a certain infection rate from which the optimal group size is one, and in fact individual testing is used.

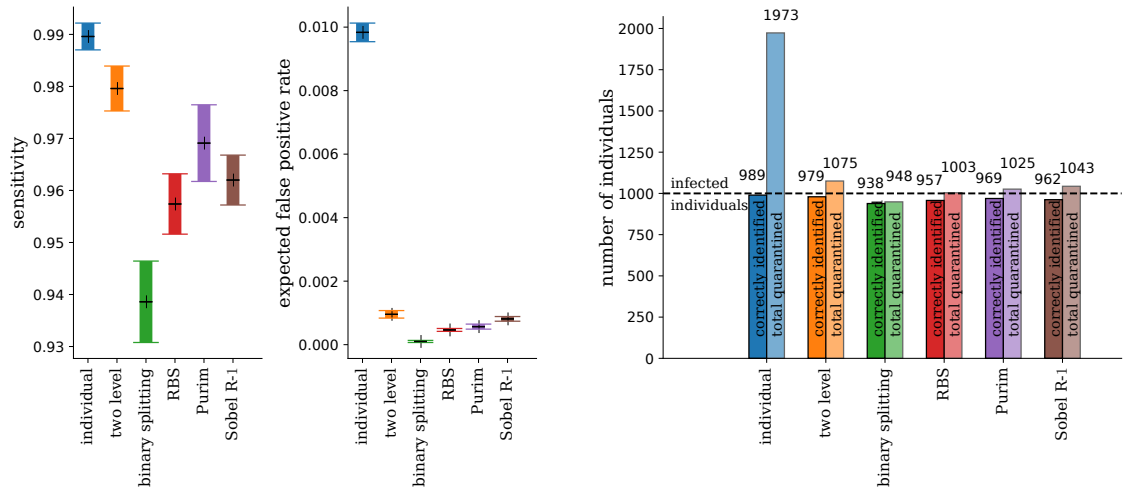
Scenarios. We consider five realistic country-based scenarios as of April 2020, to reflect a broad variety of population sizes and testing capacities (United Kingdom, United States, Singapore, Italy, Germany). They can be seen in Fig. 8.3B. We assume a sensitivity of $s_1 = 99\%$ based on data reported by LabCorp to the United States Food and Drug Administration and an estimated false-positive rate of 1% [Lab20]. The simulations show, among others, that testing 10% of the US population for an infection rate of 1% would take 225 days when testing individually, but only between 21 and 46 days using pooled testing; a speedup by a factor of up to 10.8. This would enable to repeatedly test frontline medical staff and public workers in order to maintain stability of the medical system and the economy.



A Expected number of identified cases per test for different infection rates using optimized group sizes.

B Expected time to test the whole population depending on the daily test capacity per 1 million population as of early April 2020 (infection rate 1%).

FIGURE 8.3 Efficiency of pool testing approaches. Measured in identified cases per test and time to test the population.



A Expected true positives and false positives for the six testing strategies.

B Numbers of correctly identified cases (left bars) and quarantined individuals (right bars) for the six testing strategies.

FIGURE 8.4 Accuracy of pooled testing approaches, calculated for a reference population of 100 000 individuals and an infection rate of 1%.

Pool test accuracy. A minor downside of pooled testing is that a screening campaign based on a hierarchical approach on average identifies fewer cases than individual testing. The expected rate of identified cases (true positives) therefore drops by between 1% and 5% in total, see Fig. 8.4A. However, referring to Fig. 8.3A the improvement in terms of identified cases per test is so significant, that although few infected individuals are missed, overall the pooling strategies are still significantly more efficient. Additionally, the likelihood of incorrectly classifying a subject as infected is reduced from 1% to almost 0%, because multiple sequential tests reduce the false positive rate with every additional test. Thus, unnecessary quarantines can be prevented almost entirely; an important consideration for the economy, see Fig. 8.4B.

Apart from efficiency, a major consideration for the choice of pooling strategy is the potential sequential depth, i.e., the maximum number of times one sample is added to different pools. The durability of the sample might be limited, so that only a certain sequential depth is feasible. For our maximum pool size of 32 individuals, binary splitting requires at most six sequential steps of testing of a single sample. Purim and two level testing can always be carried out in two sequential testing steps. For recursive binary splitting and Sobel-R1, re-pooling the batch sizes can lead to large numbers of sequential stages, they are thus restricted in their practical applicability.

Overall our study provides a clear rationale for adoption of pool-based testing to increase speed and throughput of testing for COVID-19. Our simulations indicate that population-wide testing is only possible by making use of pool-based strategies. Otherwise, testing even 10% of the US population would take more than seven months as of April 2020.

8.3 Response Surface

The pool-testing simulation is publicly available [Cod20], however, its usage requires basic programming skills. To make the simulation available for decision-makers, doctors and the interested public, we have additionally provided a website [Cov20], which allows the specification of custom scenarios and returns the key information of the simulation. The web-server is however limited in its computational resources and not able to perform full scale simulations within an acceptable response time. Therefore, the computational intensive tasks are outsourced to a response surface.

In a first internal step the response surface optimizes the group sizes for the specified infection rate. In a second step the actual results depending on the optimal group size are queried and returned to the user. The whole process takes only a fraction of a second and is not noticeable to the user, who receives immediate results for his or her request.

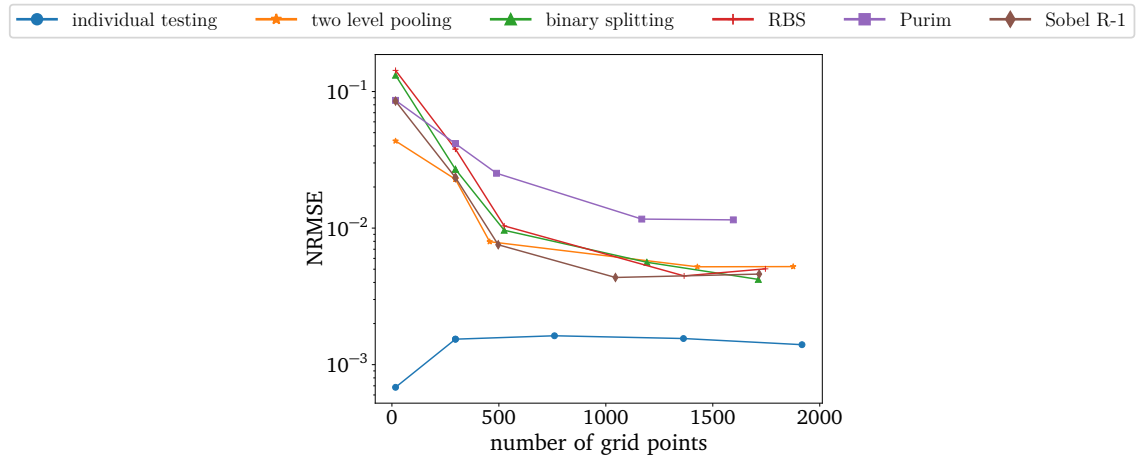


FIGURE 8.5 Convergence of the response surfaces for the pool testing simulation, based on a reference population of 100 000 individuals and 10 repetitions of the simulation. The response surfaces are created with spatially adaptive sparse grids and not-a-knot boundary B-splines of degree 3.

For the response surfaces we use spatially adaptive sparse grids including the boundary and cubic not-a-knot B-splines. We create individual response surfaces for each of the six methods, allowing method-specific adaptive refinement. Figure 8.5 shows the NRMSE of the response surfaces for increasing numbers of grid points and all six testing approaches. All methods but individual testing start with good convergence rates but soon stagnate and for around 1 500 grid points achieve the best possible approximation. For individual testing the same effect occurs, but the underlying objective function is linear. Therefore, the response surface immediately reaches the point of the best possible approximation. This limit to the approximation accuracy for each method can be explained by stochastic fluctuations.

Stochastic fluctuations. Individuals of the virtual population are infected randomly with respect to the infection rate. To account for outliers, the outputs of the simulation in Tab. 8.1 are calculated as expected values over r runs. This converges towards the actual expected value for increasing r. However, the outputs of the simulation, which are used as inputs for learning the response surface, still contain stochastic fluctuations. We need to investigate the size of these fluctuations to find a trade-off between run-time and accuracy.

There is a hard limit on the overall achievable accuracy given by the population size. As our simulation treats members of the population individually, it can not take fractional individuals into account. Given a population of 10^a individuals for $a \in \mathbb{N}_0$, the

infection rate can only be taken into account up to its a-th decimal. Therefore, we use a reference population of fixed size, and scale the results to the actual input population P. This ensures enough accuracy for smaller populations, it prevents extreme run-times for very large populations, and it provides the same accuracy for all inputs.

The stochastic fluctuations mainly depends on two of the input parameters, the population size P and the number of simulation runs r. We now measure this relation for various combinations of these key parameters. Let x_i, $i \in \{1, \dots, 100\}$ be 100 randomly chosen points in the parameter space for a fixed population size P, and a fixed number of repetitions r. For every testing strategy we evaluate the simulation ten times at each point, resulting in evaluation sets $(w_{1,1}, \dots, w_{1,10}), \dots, (w_{100,1}, \dots, w_{100,10})$. From this we approximate the stochastic fluctuations ε_i at each point x_i as the largest variation between two measurements

$$\varepsilon_i := \max_j w_{i,j} - \min_j w_{i,j}. \tag{8.1}$$

We then approximate the overall noise ε for population size P and r repetitions as

$$\varepsilon := \max_i \varepsilon_i. \tag{8.2}$$

This is a rough estimate on how much the simulation evaluations vary, which serves as an upper bound on up to which order of magnitude the response surface can learn meaningful values at all. The quality of the response surfaces, measured as NRMSE, matches the approximative noise ε, as can be seen in Fig. 8.6. In particular, it is not possible for any response surface to provide more accurate results. It should be noted that for noisy data overfitting is a potential risk. However, as can be seen in Fig. 8.6 this is not the case here. Instead, the error stagnates just as expected.

Based on this investigation we have decided to use a reference population of $P = 100\,000$ individuals, $r = 10$ simulation runs, a reference daily test capacity of $c = 1\,000$ individuals, and a reference test duration of $T = 5$ hours to train the response surface. The outputs are then scaled accordingly. With this setup the stochastic fluctuations and thus the overall error are significantly smaller than 1%.

Summary. We have developed a valuable decision support tool to improve the testing for COVID-19. We found that at no extra cost testing for the disease can be made significantly more efficient by using pooled testing. To the best of our knowledge, our study [Wol20] presented the first comprehensive, comparative assessment of optimized testing strategies for effective large-scale screening for infection with COVID-19. At the point of writing the study, other research groups had investigated two level pooling with

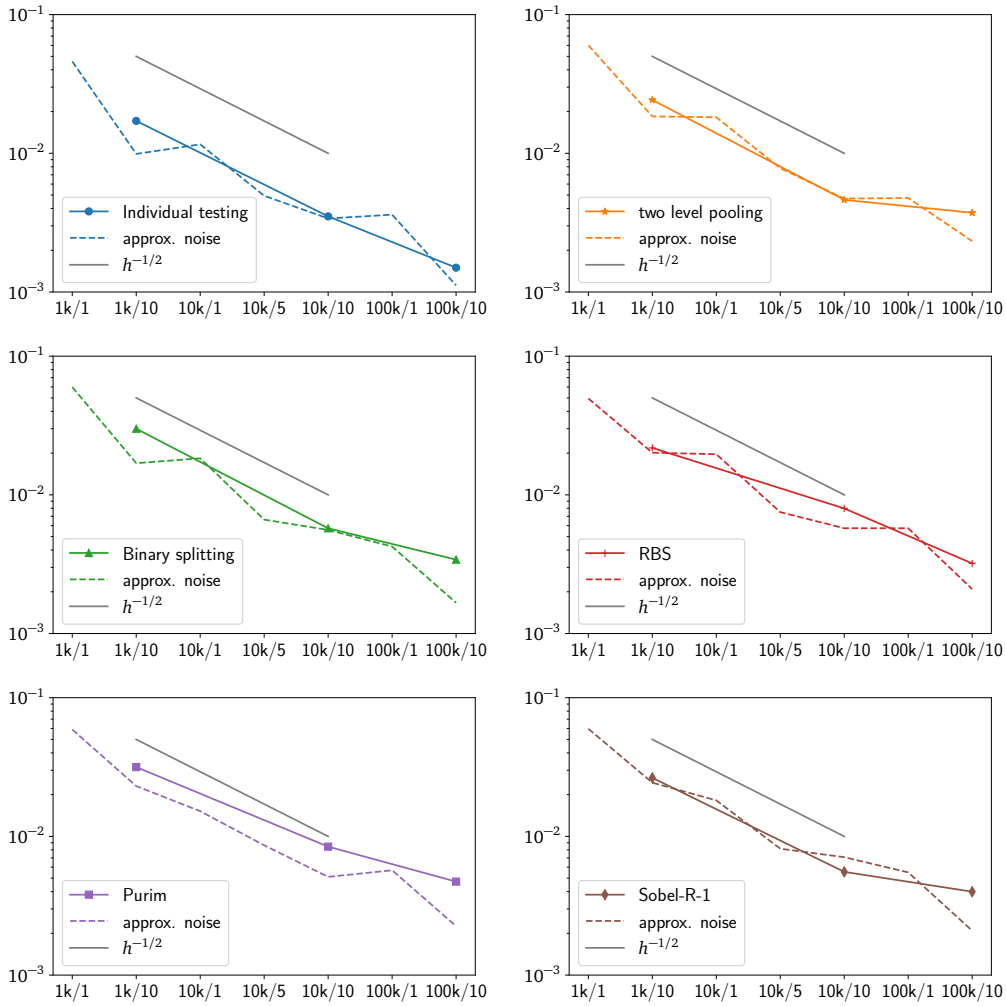


FIGURE 8.6 Approximate stochastic noise of the simulation, here illustrated for cases identified per test. The x-axis consists of population/number of repetitions pairs, where the population is given in 1 000 (k). For a subset of these pairs the NRMSE of response surfaces with maximum accuracy is plotted. The convergence rate of $O(\sqrt{N})$ illustrates the dependency of the accuracy on the underlying Monte Carlo method.

optimized pool size, matrix-based approaches for pooled screening and binary splitting for COVID-19 [Far20; Täu20; Sin20; Sha20; Nor20; Nar20]. However, the presented study was the only one that includes models of all these central categories of approaches, benchmarking and comparing them with each other. In particular, our comparison was the only one comparing tree-based approaches with a matrix-based approach. To make the results available and allow custom requests, an interactive online tool is provided. This is only possible, because the online tool can rely on a response surface which approximates the simulation with an error of similar size as the underlying stochastic fluctuations.

Pooled testing has been sufficiently applied in practice by now, for example in Ghana [Sib20] where otherwise no wide-range testing would have been possible. Another example is the Chinese Province Qingdao where, after first infections were reported in October 2020, two stage pooling was used to test all 10.9 million residents [Nej20]. All infected individuals were found and quarantined and the likely mass-outbreak of Covid-19 was prevented without a lockdown.

9

Conclusion

In this thesis we have introduced surrogates with B-splines on sparse grids for uncertainty quantification. We have investigated the potential of this approach using several real-world applications, and compared the results with established UQ methods. The various application areas have been as heterogeneous as UQ itself. They have required, among others, differentiation, integration, optimization, and efficient discretization, which demonstrates the great versatility and general applicability of our approach.

The main contributions of this thesis can be summarized as follows:

- We have presented two new equivalent non-boundary B-spline bases, and we have proved that they span the desired function space, allowing higher-order approximations on spatially adaptive non-boundary sparse grids for arbitrary objectives. We have compared our new bases with state-of-the-art hierarchical bases for sparse grids, and the new bases have always performed comparable or better.

- Using stochastic collocation we have demonstrated that our surrogates can perform accurate forward UQ by propagating uncertainties through simulations, and calculating stochastic key quantities.

- We have used the gradients of our surrogates to enable the otherwise unfeasible usage of the efficient NUTS algorithm. With this we have performed inverse UQ and efficiently sampled the posterior of hierarchical Bayesian models.

- Exploiting the integrability and differentiability of our surrogates, we have detected active subspaces, and outperformed the status-quo numerical toolbox for this task, PASUL. Furthermore, we have developed a new integration algorithm for functions with a one-dimensional active subspace based on a spline representation of otherwise difficult to compute high-dimensional volume.

We demonstrated the practical usability of our surrogate approach with four applications from different fields. In the first application we performed a benchmark calculation for subterranean storage of carbon dioxide. This required the accurate representation of an expensive objective with a discontinuity, and the calculation of its stochastic moments. By comparing our results with four other state-of-the-art UQ methods we demonstrated that our approach is highly competitive.

In our second application we performed inverse UQ, by using the NUTS algorithm to sample the posterior from Bayesian inference. Although this algorithm is very efficient, it requires gradient information, which for complex simulations is generally not available. By creating a surrogate and using its derivatives, we were able to apply NUTS for real-world measurements from a DC motor test-bench resulting in accurate parameter calibration fulfilling industry requirements.

We then performed a thorough forward UQ analysis for a simulation of tsunami run-ups. For this we created a surrogate which reached the accuracy of the underlying simulation itself, calculated stochastic moments and percentiles, and optimized the wave shape, leading to the maximum run-up.

The last application dealt with a recent topic of interest, pool testing for COVID-19, where we simulated and compared multiple testing strategies. In the according publication only a small selection of scenarios could be presented, therefore an interactive website has been created to allow for individual recommendations. This was only possible, because the expensive actual simulations could be replaced by a surrogate, which is accurate almost up to stochastic fluctuations.

This thesis has demonstrated the great potential of the combination of B-splines and sparse grids. Of course, in research solving one problem presents a dozen new ones, and many starting points for future work exist.

- All investigated applications allowed direct evaluation, thus we have been able to focus on interpolation. Our methods, however, can easily be generalized to become a data-driven approach by using regression instead.

- The probability distributions of the parameters in our applications have either been specified as standard closed form distributions, or we have used Monte Carlo quadrature to incorporate samples. B-splines on sparse grids could also be used to smoothly approximate the unknown distribution from samples, i.e., to perform a kernel density estimation.

- The extension of B-splines in Sec. 3.2 has been motivated by weighted and extended B-splines (WEB-splines). It should be possible to generalize WEB-splines themselves in a hierarchical manner, and thus to use them on sparse grids. This would allow

more accurate sparse grid approximations on domains with difficult geometry.

- B-splines on sparse grids are of course not restricted to applications from UQ. The newly introduced bases and methods can also be used for applications from other fields that require accurate and smooth approximation from few data points, derivative information, integration, optimization, or collocation.

A

Appendix: Proofs

This appendix chapter contains proofs which were too long to be included in the main text, where they would have disturbed the flow of reading.

A.1 Proofs for Chapter 3

A.1.1 Proof of the linear independence of hierarchical boundaryless not-a-knot B-Splines

THEOREM 3.14 (hierarchical boundaryless nak B-splines are linearly independent) *For degree $n \in \{1,3,5\}$ the hierarchical boundaryless not-a-knot splines*

$$\{b_{\ell',k}^{n,\mathrm{bnk}} \mid 1 \leq \ell' \leq \ell,\ k \in I_{\ell'}\} \tag{3.30}$$

are a basis of $S_\ell^{n,\mathrm{poly}}$.

PROOF For level $\ell < \Lambda(n)$ the boundaryless not-a-knot B-spline basis is defined via Lagrange polynomials, only from level $\ell = \Lambda(n)$ onwards B-splines are used. The statement is clearly true for the Lagrange polynomials on the first levels $\ell < \Lambda(n)-1$. For $\ell \geq \Lambda(n)-1$ we show the statement by using Thm. 2.2. We first consider degree $n = 1$. The single Lagrange polynomial of level $\ell = \Lambda(n)-1 = 1$ is representable through the nodal boundaryless not-a-knot B-spline basis of level $\ell = \Lambda(n) = 2$, resulting in the prolongation matrix

$$\boldsymbol{P} = \begin{pmatrix} 1 & 2 & 1 \end{pmatrix}. \tag{A.1}$$

The matrix $\boldsymbol{P}_1$ is simply $\boldsymbol{P}_1 = (1)$, which is regular, and thus Thm. 2.2 can be applied.

Let now $\ell \geq \Lambda(n)$. We begin by using Böhm's algorithm (Thm. 3.4) to represent the nodal boundaryless not-a-knot B-spline basis of level ℓ through the nodal boundaryless not-a-knot B-spline basis of level $\ell+1$, by iteratively inserting all new knots of $\xi_{\ell+1}^{n,\mathrm{bnk}}$ into $\xi_{\ell}^{n,\mathrm{bnk}}$. We obtain the prolongation matrix $\boldsymbol{P} \in \mathbb{R}^{(2^{\ell}-1)\times(2^{\ell+1}-1)}$, which has entries $p_{i,k}$, such that

$$b_{\ell,k}^{n,\mathrm{bnk}} = \sum_{i \in K_{\ell+1}^{\mathrm{bnk}}} p_{i,k}\; b_{\ell+1,i}^{n,\mathrm{bnk}} \tag{A.2}$$

for all $k \in K_{\ell}^{\mathrm{bnk}}$. For degree $n = 1$ and level $\ell = 2$, the matrix $\boldsymbol{P}$ is given by

$$\boldsymbol{P} = \begin{pmatrix} 1 & 0.5 & 0.25 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0.75 & 1 & 0.75 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0.25 & 0.5 & 1 \end{pmatrix}. \tag{A.3}$$

Therefore, the matrix $\boldsymbol{P}_1$ is given by

$$\boldsymbol{P}_1 = \begin{pmatrix} 0.5 & 0.5 & 0 \\ 0 & 1 & 0 \\ 0 & 0.5 & 0.5 \end{pmatrix}. \tag{A.4}$$

We perform two Gauss elimination steps on $\boldsymbol{P}_1$. First, we subtract the second row 0.5 times from the first row. Second, we perform the analogous operation on the last row. These operations can be written as multiplications with Frobenius matrices $\boldsymbol{F}_1$ and $\boldsymbol{F}_2$. Let the result be $\tilde{\boldsymbol{P}_1} = \boldsymbol{F}_1\boldsymbol{F}_2\boldsymbol{P}_1$, then $\tilde{\boldsymbol{P}_1}$ is diagonally dominant and thus regular. It follows that $\boldsymbol{P}_1$ is regular because we can explicitly construct its inverse, $\boldsymbol{P}_1^{-T} = \tilde{\boldsymbol{P}_1}^{-T}\boldsymbol{F}_1\boldsymbol{F}_2$. Let now level $\ell > 2$, then the matrix $\boldsymbol{P}_1$ is given by

$$\boldsymbol{P}_1 = \begin{pmatrix} 0.5 & 0.5 & & & \\ & 1 & \ddots & & \\ & & & 1 & \\ & & & 0.5 & 0.5 \end{pmatrix} \tag{A.5}$$

The same strategy of a partial Gaussian elimination of the first and last row can be applied. The result are diagonally dominant matrices, which are thus regular.

For degrees $n = 3$ and $n = 5$ an analogous procedure of partial Gaussian elimination can be performed. For brevity, we do not state them explicitly because many more eliminations steps are necessary. But ultimately the matrices become diagonally dominant. Thus, they are regular, which again implies the regularity of $\boldsymbol{P}_1$. Consequently, Thm. 2.2

can be applied for $n \in \{1,3,5\}$. ■

A short note on the degree. We believe that Thm. 3.14 holds for all degrees $n \in \mathbb{N}$. Unfortunately, the previous proof technique using partial Gaussian elimination does no longer work for degree $n > 5$. Because in this thesis we focus on spline degrees $n \in \{1,3,5\}$, this implies no restriction to our work. However, for a generally valid theory, it remains to show P_1's regularity for all $n \in \mathbb{N}$.

A.1.2 Proof that extended and boundaryless not-a-knot B-Splines span the same space

We'll need truncated power functions for the proof of the next theorem.

DEFINITION A.1 (truncated power function)
For $u \in \mathbb{R}$ the truncated power function with exponent $n \in \mathbb{N}$ is defined as

$$u_+^n := \begin{cases} 0 & u < 0, \\ u^n & u \geq 0. \end{cases} \tag{A.6}$$

Before B-splines were developed, splines were constructed using truncated power functions. This approach is known to be open to severe ill-conditioning [Boo72]. Still, for the upcoming proof we can make use of the representation of a spline as linear combination of truncated power functions. A B-spline of degree n can be written as

$$b_{k,\xi}^n(x) = (\xi_{k+n+1} - \xi_k)(-1)^n((x-\cdot)_+^n[\xi_k, \ldots, \xi_{k+n+1}]) \tag{A.7}$$

where $f[\xi_k, \ldots, \xi_{k+n+1}]$ are the divided differences of a function f, see [DeB78; Str94] for more details.

THEOREM 3.15 (extended and boundaryless nak B-splines span the same space)
Let $S_\ell^{n,\mathrm{e}} := \operatorname{span}\{b_{\ell,i}^{n,\mathrm{e}} \mid i \in \{1, \ldots, 2^\ell - 1\}\}$. *Then* $S_\ell^{n,\mathrm{e}} = S_\ell^{n,\mathrm{poly}}$.

PROOF For level $\ell < \Lambda(n)$ $S_\ell^{n,\mathrm{e}}$ and $S_\ell^{n,\mathrm{poly}}$ are both spanned by Lagrange polynomials and thus equal. Let now $\ell \geq \Lambda(n)$.

Extended not-a-knot B-splines are linear combinations of not-a-knot B-splines including the boundary B-splines. Therefore, $S_\ell^{n,e} \subseteq S_{\xi_\ell^{n,\mathrm{nak}}}^n$. The boundaryless not-a-knot knot-sequence is defined by removing knots from the not-a-knot sequence. It follows immediately from Böhm's algorithm that $S_\ell^{n,\mathrm{poly}} \subseteq S_{\xi_\ell^{n,\mathrm{nak}}}^n$. Let $s = \sum_{k \sim \xi_\ell^{n,\mathrm{nak}}} c_k b_{\ell,k}^{n,\mathrm{nak}} \in S_{\xi_\ell^{n,\mathrm{nak}}}^n$ be a spline with coefficients $c_k \in \mathbb{R}$. We now show that $s \in S_\ell^{n,e} \iff s \in S_\ell^{n,\mathrm{poly}}$.

According to Marsden's identity, the extension coefficients $e_{i,j}$ depend only on the degree n and not on the particular choice of the underlying polynomial basis, cf. Sec. 3.2.2.

Therefore, s is an extended spline in $S_\ell^{n,e}$ if and only if its first coefficient c_0 and its last coefficient c_{2^ℓ} are correctly represented by the linear combination of extension coefficients and the next $n+1$ inner coefficients. We can formulate this as

$$s \in S_\ell^{n,e} \iff \tag{A.8a}$$

$$\left(-1, e_{1,0}, \ldots, e_{n+1,0}\right) \begin{pmatrix} c_0 \\ \vdots \\ c_{n+1} \end{pmatrix} = 0 \quad \wedge \tag{A.8b}$$

$$\left(e_{2^\ell-n-1,2^\ell}, \ldots, e_{2^\ell-1,2^\ell}, -1\right) \begin{pmatrix} c_{2^\ell-n-1} \\ \vdots \\ c_{2^\ell} \end{pmatrix} = 0. \tag{A.8c}$$

For x in the left-most inner knot-segment $[\xi_{\ell,n}^{n,\mathrm{nak}}, \xi_{\ell,n+1}^{n,\mathrm{nak}})$, s is a polynomial of degree n,

$$s\big|_{[\xi_{\ell,n}^{n,\mathrm{nak}}, \xi_{\ell,n+1}^{n,\mathrm{nak}})}(x) = a_0 + a_1 x + \cdots + a_n x^n, \tag{A.9}$$

where $s\big|_{[\xi_{\ell,n}^{n,\mathrm{nak}}, \xi_{\ell,n+1}^{n,\mathrm{nak}})}(x)$ is the polynomial which equals s on the knot segment $[\xi_{\ell,n}^{n,\mathrm{nak}}, \xi_{\ell,n+1}^{n,\mathrm{nak}})$ with coefficients $a_m \in \mathbb{R}$. Note that the polynomial itself is globally defined and thus can be evaluated for all $x \in \mathbb{R}$.

We now look at x in the next knot-segment $[\xi_{\ell,n+1}^{n,\mathrm{nak}}, \xi_{\ell,n+2}^{n,\mathrm{nak}})$. Already knowing s on the previous segment, there is only one degree of freedom left, which is introduced by $\xi_{\ell,n+1}^{n,\mathrm{nak}}$ and according to Eq. (A.7) can be represented by a truncated power function multiplied with a coefficient $d_{\mathrm{left}} \in \mathbb{R}$,

$$s\big|_{[\xi_{\ell,n+1}^{n,\mathrm{nak}}, \xi_{\ell,n+2}^{n,\mathrm{nak}})}(x) = s\big|_{[\xi_{\ell,n}^{n,\mathrm{nak}}, \xi_{\ell,n+1}^{n,\mathrm{nak}})}(x) + d_{\mathrm{left}}(x - \xi_{\ell,n+1}^{n,\mathrm{nak}})_+^n. \tag{A.10}$$

The spline s on $[\xi_{\ell,n}^{n,\mathrm{nak}}, \xi_{\ell,n+2}^{n,\mathrm{nak}})$ is now represented in two different bases, in the B-splines $b_{\ell,k}^{n,\mathrm{nak}}$ with coefficients $c_0, \ldots, c_{n+1}$, and in the monomial and truncated power function basis with coefficients $a_0, \ldots, a_n, d_{\mathrm{left}}$. Transforming from one basis to the other, the coefficients $c_0, \ldots, c_{n+1}$ of the spline s can be uniquely identified from $a_0, \ldots, a_n, d_{\mathrm{left}}$. In particular there exists a linear transformation matrix $\boldsymbol{A} \in \mathbb{R}^{n+1 \times n+1}$, such that

$$\boldsymbol{A} \begin{pmatrix} a_0 \\ \vdots \\ a_n \\ d_{\mathrm{left}} \end{pmatrix} = \begin{pmatrix} c_0 \\ \vdots \\ c_n \\ c_{n+1} \end{pmatrix}. \tag{A.11}$$

Using this, Eq. (A.8b) becomes

$$\left(-1, e_{1,0}, \ldots, e_{n+1,0}\right) \boldsymbol{A} \begin{pmatrix} a_0 \\ \vdots \\ a_n \\ d_{\text{left}} \end{pmatrix} = 0. \tag{A.12}$$

We now look at the product of the extension coefficients and the matrix $\boldsymbol{A}$. Let this be

$$\left(v_0, \ldots, v_{n+1}\right) \coloneqq \left(-1, e_{1,0}, \ldots, e_{n+1,0}\right) \boldsymbol{A}. \tag{A.13}$$

Consider any monomial x^m for $m \in \{0, \ldots, n\}$. According to Cor. 3.12 $x^m \in S_\ell^{n,e}$. Because the corresponding coefficients are $a_0 = \cdots = a_{m-1} = a_{m+1} = \cdots = a_n = d_{\text{left}} = 0$ and $a_m = 1$, it follows from Eq. (A.12)

$$0 = \left(-1, e_{1,0}, \ldots, e_{n+1,0}\right) \boldsymbol{A} \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} = \left(v_0, \ldots, v_{n+1}\right) \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} = v_m. \tag{A.14}$$

Consequently, $v_m = 0$ for all $m \in \{0, \ldots, n\}$. We apply an analogous procedure to Eq. (A.8c) based on $(d_{\text{right}}, a_{2^\ell - n - 1}, \ldots, a_{2^\ell})^T$, for $d_{\text{right}} \in \mathbb{R}$. Combining the two, Eq. (A.8) becomes

$$s \in S_\ell^{n,e} \iff v_{n+1} d_{\text{left}} = 0 \ \wedge \ v_{2^\ell - n - 1} d_{\text{right}} = 0. \tag{A.15}$$

We focus again one the first of the two right hand sides. Following from Eq. (A.10), the matrix $\boldsymbol{A}$ has the following structure

$$\boldsymbol{A} = \begin{pmatrix} a_{0,0} & \ldots & a_{0,n} & 0 \\ \vdots & & \vdots & \vdots \\ a_{n,0} & \ldots & a_{n,n} & 0 \\ a_{n+1,0} & \ldots & a_{n+1,n} & w \end{pmatrix} \tag{A.16}$$

for entries $a_{i,j} \in \mathbb{R}$ and $w \neq 0$. Therefore, $v_{n+1} = e_{n+1,0} w$. As in [Höl03], $e_{n+1,0}$ can be written as Lagrange polynomial in the knots, which implies in particular $e_{n+1,0} \neq 0$. Therefore, it holds $v_{n+1} \neq 0$. Analogously we derive $v_{2^\ell - n - 1} \neq 0$, and by combining both,

Eq. (A.15) becomes

$$s \in S_\ell^{n,e} \iff d_{\text{left}} = 0 \wedge d_{\text{right}} = 0. \tag{A.17}$$

This means that $s \in S_\ell^{n,e}$ if and only if the coefficients for the basis at $\xi_{\ell,n+1}^{n,\text{nak}}$ and $\xi_{\ell,2^\ell-n-1}^{n,\text{nak}}$ are zero, i.e., the spline is smooth at these knots. Consequently, extended not-a-knot B-splines do not depend on the position of outer knots, but only on the smoothness at inner knots and their degree, which is exactly the definition of the boundaryless not-a-knot sequence $\xi_\ell^{n,\text{bnk}}$, see Eq. (3.27). Therefore, the two bases span the same space.

■

A.2 Proofs for Chapter 4

The upcoming proof is based on Schoenberg's original work [Cur66], which makes use of the following result of Peano [Dav63; Cur66],

LEMMA A.2 (divided difference and M-spline)
Let $f \in C^D$ be a D times continuously differentiable function, and $f[\xi_0, \xi_1, \ldots, \xi_D]$ be the divided difference of f with respect to $\xi_0, \xi_1, \ldots, \xi_D$. Then

$$f[\xi_0, \xi_1, \ldots, \xi_D] = \frac{1}{D!} \int_{\xi_0}^{\xi_D} M_{0,\xi}^D(x) f^{(D)}(x) dx, \tag{A.18}$$

where $\xi = (\xi_0, \ldots, \xi_D)$, and the M-spline $M_{0,\xi}^D$ is defined as in Thm. 4.4.

Furthermore, we need the following lemma by Hermite and Gennochi [Nör24; Cur66],

LEMMA A.3 (divided difference as simplex integral)
Let f be an analytic function, then

$$f[\xi_0, \ldots, \xi_D] = \int_{\tau_D} \cdots \int f^{(D)}(\xi_0 t_0 + \cdots + \xi_D t_D) dt_1 \cdots dt_D, \tag{A.19}$$

where the integration is performed over the simplex

$$\tau_D := \left\{ t_1 \geq 0, \cdots, t_D \geq 0 \mid 1 \geq \sum_{d=1}^{D} t_d, \text{ and } t_0 = 1 - t_1 - \cdots - t_D \right\}. \tag{A.20}$$

A.2.1 Proof that the volume of a projected simplex is an M-spline

THEOREM 4.4 (The volume of a projected simplex is an M-spline)
The density function $\Phi(x_1)$ obtained by projecting orthogonally onto the first coordinate x_1 the volume of a D-dimensional simplex σ of volume V_σ, so located that its $D+1$ vertices $\boldsymbol{v}^0,\dots,\boldsymbol{v}^D$ project orthogonally into the knot sequence $\boldsymbol{\xi}^\sigma := (\xi_0^\sigma,\dots,\xi_D^\sigma)$, is given by

$$\Phi(x_1) = V_\sigma \cdot M^D_{0,\xi^\sigma}(x_1), \tag{4.51}$$

where the M-spline M^D_{0,ξ^σ} can be represented as a scaled B-spline,

$$M^D_{0,\xi^\sigma} = \frac{D}{\xi_D^\sigma - \xi_0^\sigma} b^D_{0,\xi^\sigma}. \tag{4.52}$$

PROOF The vertices of σ are $\boldsymbol{v}^d = (v_1^d,\dots,v_D^d)$ for $d = 0,\dots,D$. The requirements of the theorem state that projecting these vertices onto their first coordinate results in the knot sequence $\boldsymbol{\xi}^\sigma = (\xi_0^\sigma,\dots,\xi_D^\sigma)$. This means $v_1^0 = \xi_0^\sigma,\ v_1^1 = \xi_1^\sigma,\cdots,v_1^D = \xi_D^\sigma$.

Let now f be an analytic function. By applying Cavalieri's principle, we obtain

$$\int_\sigma \cdots \int f^{(D)}(x_1)dx_1\dots dx_D = \int_{\xi_0^\sigma}^{\xi_D^\sigma} \Phi(x_1) f^{(D)}(x_1)dx_1. \tag{A.21}$$

We apply integration by substitution to the n-fold integral on the left-hand side and change to new variables $t_1,\dots,t_D$ defined by

$$\begin{aligned} x_1 &= v_1^0\left(1-\sum_{d=1}^D t_d\right) + v_1^1 t_1 + \cdots + v_1^D t_D, \\ &\vdots \\ x_D &= v_D^0\left(1-\sum_{d=1}^D t_d\right) + v_D^1 t_1 + \cdots + v_D^D t_D. \end{aligned} \tag{A.22}$$

The absolute value of the determinant of the Jacobian of the substitution's partial derivatives is

$$\left|\det \frac{\partial(x_1,\dots,x_D)}{\partial(t_1,\dots,t_D)}\right| = \left|\det \begin{pmatrix} v_1^1 - v_1^0 & v_1^2 - v_1^0, & \cdots & , v_1^D - v_1^0 \\ v_2^1 - v_2^0, & \cdots & & , v_2^D - v_2^0 \\ \vdots & & & \\ v_D^1 - v_D^0, & \cdots & & , v_D^D - v_D^0 \end{pmatrix}\right| = |\det \boldsymbol{A}|, \tag{A.23}$$

for the matrix $\boldsymbol{A} \coloneqq [\boldsymbol{v}^1 - \boldsymbol{v}^0, \ldots, \boldsymbol{v}^D - \boldsymbol{v}^0] \in \mathbb{R}^{DxD}$. We make use of the formula for the volume of simplices in Eq. (4.50), and obtain

$$|\det \boldsymbol{A}| = D! V_\sigma. \tag{A.24}$$

The integral on the left-hand side of (A.21) is transformed into the integral on the right-hand side of (A.19) multiplied by $D!V_\sigma$. Inserting (A.19) into (A.21) yields

$$f[\xi_0^\sigma, \ldots, \xi_D^\sigma] D! V_\sigma = \int_{\xi_0^\sigma}^{\xi_D^\sigma} \Phi(x_1) f^{(D)}(x_1) dx_1. \tag{A.25}$$

We insert (A.18) into Eq. (A.25), resulting in

$$V_\sigma \int_{\xi_0^\sigma}^{\xi_D^\sigma} M^D_{0,\xi_\sigma}(x_1) f^{(D)}(x_1) dx_1 = \int_{\xi_0^\sigma}^{\xi_D^\sigma} \Phi(x_1) f^{(D)}(x_1) dx_1, \tag{A.26}$$

which holds for all analytic functions f. By the continuity of $\Phi(x_1)$, we conclude $V_\sigma M^D_{0,\xi_\sigma} = \Phi$. ■

B

Application Details

B.1 Additional Details for Chapter 6

The analytical solution for the ODE system (6.1) for initial values $I(0) = \omega(0) = 0$ is [Sto18],

$$
\begin{aligned}
I(t) &= c_1 e^{\lambda_1 t} v_{11} + c_2 e^{\lambda_2 t} v_{21} + \frac{V(\lambda_1 v_{21} - \lambda_2 v_{11})}{L\lambda_1\lambda_2(v_{11} - v_{21})}, \\
\omega(t) &= c_1 e^{\lambda_1 t} + c_2 e^{\lambda_2 t} + \frac{V(\lambda_1 - \lambda_2)}{L\lambda_1\lambda_2(v_{11} - v_{21})},
\end{aligned}
\tag{B.1}
$$

where

$$
\begin{aligned}
\lambda_1 &= -\frac{LD - \mu + JR}{2JL}, \\
\lambda_2 &= -\frac{LD + \mu + JR}{2JL}, \\
v_1 &= \begin{pmatrix} v_{11} \\ v_{12} \end{pmatrix} = \begin{pmatrix} \frac{d}{c_g} - \frac{LD - \mu + JR}{2Lc_g} \\ 1 \end{pmatrix}, \\
v_2 &= \begin{pmatrix} v_{21} \\ v_{22} \end{pmatrix} = \begin{pmatrix} \frac{d}{c_g} - \frac{LD + \mu + JR}{2Lc_g} \\ 1 \end{pmatrix}, \\
c_1 &= -\frac{V}{L\lambda_1(v_{21} - v_{11})}, \\
c_2 &= \frac{V}{L\lambda_2(v_{21} - v_{11})}, \\
\mu &= \sqrt{(JR - LD)^2 - 4c_m c_g JL}.
\end{aligned}
$$

B.2 Additional Details for Chapter 8

B.2.1 Recursive Binary Splitting

```
   S, a set of samples, at least one infected
 1 function DIG(S)
 2   T ← S
 3   while |T| > 1 do
 4     Select a subset T′ of size ⌈|T|/2⌉ from T.  Test T′
 5     if T′ is infected then
 6       T ← T′
 7     else
 8       Identify all samples in T′ as good
 9       T ← T\T′
10   Identify the single sample in T as infected
11   Return all unidentified samples of S
```

ALGORITHM B.1 Binary splitting procedure DIG identifies one infected sample and an unspecified number of good samples from a sample set S [Che15].

```
   S, a set of samples which are either good or infected
   c ∈ {1,∗}, status indicator                    ⇝ c = 1: S contains at least one infected,
 1                                                    c = ∗: no information available
 2 function RBS(S,c)
 3   if |S| == 1 then
 4     if c == ∗ then Test and identify single sample in S, exit
 5     else if c == 1 then Identify single sample in S as infected, exit
 6   else if |S| > 1 then
 7     if c == ∗ then
 8       Test S
 9       if S is not contaminated then
10         Identify all samples in S as good, exit
11   Partition S into subsets S1, S2, s.t.  |S1| = ⌈|S|/2⌉ and |S2| = |S| − ⌈|S|/2⌉
12   Test S1 and S2
13   if S1 is good and S2 is infected then
14     S′2 = DIG(S2)            ⇝ identifies one infected and an unspecified number of good samples
15     if S′2 ≠ ∅ then RBS(S′2,∗)
16   else if S1 is infected and S2 is good then
17     S′1 = DIG(S1)            ⇝ identifies one infected and an unspecified number of good samples
18     if S′1 ≠ ∅ then RBS(S′1,∗)
19   else if S1 and S2 are infected then
20     RBS(S1,1)
21     RBS(S2,1)
```

ALGORITHM B.2 Recursive Binary Splitting (RBS) classifies all samples of a sample set S to be either good or infected [Che15].

B.2.2 Sobel R-1

The procedure R-1, introduced by Sobel and Groll, is an optimal pool testing algorithm in the sense that the expected number of tests to identify all infected samples is minimal, see [Sob59] for the proof of the optimality. At every stage of the algorithm, the current n samples are organized in two sets. The *infected set* I of size $m \geq 0$, for which it is known that it contains at least one infected sample, and the *binomial set* B, which contains the remaining $n - m \geq 0$ samples. Sobel and Groll assume a binomial distribution of the test results with q the probability of a sample being good, and p the probability of being infected. Then the expected number of remaining pool tests is given recursively by

$$H(n) = 1 + \min_{1 \leq k \leq n} (q^k \, H(n-k) + (1-q^k) \, G(k,n)), \tag{B.2}$$

for $m = 0$, and

$$G(m,n) = 1 + \min_{1 \leq k \leq m-1} \left[\left(\frac{q^k - q^m}{1-q^m} \right) G(m-k, n-k) + \left(\frac{1-q^k}{1-q^m} \right) G(k,n) \right], \tag{B.3}$$

for $n \geq m \geq 2$. The boundary cases are defined as

$$H(0) = 0, \quad G(1,n) = H(n-1) \text{ for } n \geq 1. \tag{B.4}$$

The algorithm refers to $m > 1$ as the *G-situation*, and to $m = 0$ as the *H-situation*.

In the algorithm the optimal group size k for the next pool test is determined. This is done by testing all feasible k using the G or H formula respectively. This step can be precalculated in advance, such that the algorithm only has to look up the optimal group size for the next pool test in a table. The algorithm starts with $n = |S|$ and $m = 0$.

```
   S, a set of samples
 1 function R-1(S)
 2   B ← S                                                  ⇝ binomial set
 3   I ← ∅                                                  ⇝ infected set
 4   while B ≠ ∅ or I ≠ ∅ do
 5     if |I| == 0 then                                     ⇝ H-situation, determine infected set I
 6       k ← optimal group size for next pool test
 7       T ← subset of size k from B
 8       Test T
 9       if T is good then
10         Identify samples in T as good and remove them
11       else if T is infected then
12         I ← I ∪ T
13     else                                                 ⇝ G-situation, identify one infected sample in I
14       while |I| > 1 do
15         k ← optimal group size for next pool test
16         T ← subset of size k from I
17         Test T
18         if T is good then
19           Identify samples in T as good and remove them
20         else if T is infected then
21           B ← B ∪ I\T
22       Identify single sample in I as infected and remove it
```

ALGORITHM B.3 Sobel R-1 algorithm [Sob59].

Own Publications

[Reh18] **Rehme**, M. F.; **Zimmer**, S.; **Pflüger**, D.: *Hierarchical extended B-splines for approximations on sparse grids*, Proceedings of the 5th Workshop on Sparse Grids and Applications, 2018, to be published, Springer

[Reh19] **Rehme**, M. F.; **Pflüger**, D.: *Active subspaces with B-spline surrogates on sparse grids*, 3rd International Conference on Uncertainty Quantification in Computational Sciences and Engineering, Crete, Greece, 2019, doi:https://doi.org/10.7712/120219.6336.18612

[Reh21a] **Rehme**, M. F.; **Franzelin**, F.; **Pflüger**, D.: *B-splines on sparse grids for surrogates in uncertainty quantification*, Reliability Engineering and System Safety, 2021, ISSN: 0951-8320, doi:https://doi.org/10.1016/j.ress.2021.107430

[Reh21b] **Rehme**, M. F.; **Pflüger**, D.: *Stochastic collocation with hierarchical extended B-splines on sparse grids*, Approximation Theory XVI, Nashville, 2019, ed. by **Fasshauer**, G. E.; **Neamtu**, M.; **Schumaker**, L. L., Springer International Publishing, 2021, doi:https://doi.org/10.1007/978-3-030-57464-2_12

[Reh21c] **Rehme**, M. F.; **Roberts**, S. G.; **Pflüger**, D.: *Uncertainty quantification for the Hokkaido Nansei-Oki tsunami using B-splines on adaptive sparse grids*, Proceedings of the 20th Computational Techniques and Applications Conference 2020, 2021, doi:https://doi.org/10.21914/anziamj.v62.16121

[Wol20] **Wolff**, T. de; **Pflüger**, D.; **Rehme**, M. F.; **Heuer**, J.; **Bittner**, M.-I.: *Evaluation of pool-based testing approaches to enable population-wide screening for Covid-19*, PLOS ONE 15, 2020, doi:10.1371/journal.pone.0243692

Bibliography

[Aud06] **Audet**, C.; **Dennis Jr**, J. E.: *Mesh adaptive direct search algorithms for constrained optimization*, SIAM Journal on optimization 17.1, 2006, pp. 188–217

[Baa17] **Baar**, J. H. de; **Roberts**, S. G.: *Multifidelity sparse-grid-based uncertainty quantification for the Hokkaido Nansei-Oki tsunami*, Pure and Applied Geophysics 174.8, 2017, pp. 3107–3121

[Bal94] **Balder**, R.: *Adaptive Verfahren für elliptische und parabolische Differentialgleichungen auf dünnen Gittern*, PhD thesis, Technische Universität München, 1994

[Bar95] **Bartels**, R. H.; **Beatty**, J. C.; **Barsky**, B. A.: *An introduction to splines for use in computer graphics and geometric modeling*, Morgan Kaufmann, 1995

[Bay63] **Bayes**, T.: *LII. an essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, FRS communicated by Mr. Price, in a letter to John Canton, AMFR S*, Philosophical transactions of the Royal Society of London 53, 1763, pp. 370–418

[Bel61] **Bellman**, R.: *Adaptive Control Processes: A Guided Tour*, Rand Corporation. Research studies, Princeton University Press, 1961

[Bet17] **Betancourt**, M.: *A conceptual introduction to Hamiltonian Monte Carlo*, arXiv preprint arXiv:1701.02434, 2017

[Béz66] **Bézier**, P.: *Definition numerique des courbes et surface*, Automatisme 11.4, 1966, pp. 625–632

[Bir65] **Birkhoff**, G.; **De Boor**, C. R.: *Piecewise polynomial interpolation and approximation*, Approximation of functions, 1965, pp. 164–190

[Boe80] **Boehm**, W.: *Inserting new knots into B-spline curves*, Computer-Aided Design 12.4, 1980, pp. 199–201

[Boo72] **Boor**, C. de: *On calculating with B-splines*, Journal of Approximation Theory 6.1, 1972, pp. 50–62, ISSN: 0021-9045, doi:10.1016/0021-9045(72)90080-9

[Bor19] **Borrelli**, A.; **Wellmann**, J.: *Computer simulations then and now: an introduction and historical reassessment*, N.T.M. 27, 2019, pp. 407–417, doi:10.1007/s00048-019-00227-6

[Box87] **Box**, G. E.; **Draper**, N. R.: *Empirical model-building and response surfaces.* John Wiley & Sons, 1987

[Bro11] **Brooks**, S.; **Gelman**, A.; **Jones**, G.; **Meng**, X.-L.: *Handbook of Markov chain Monte Carlo*, CRC press, 2011

[Bun04] **Bungartz**, H.-J.; **Griebel**, M.: *Sparse grids*, Acta Numerica 13, 2004, pp. 147–269, ISSN: 1474-0508, doi:10.1017/S0962492904000182

[Bun08] **Bungartz**, H.-J.; **Pflüger**, D.; **Zimmer**, S.: *Adaptive sparse grid techniques for data mining*, Modeling, simulation and optimization of complex processes, Springer, 2008, pp. 121–130

[Bun12] **Bungartz**, H.-J.; **Heinecke**, A.; **Pflüger**, D.; **Schraufstetter**, S.: *Option pricing with a direct adaptive sparse grid approach*, Journal of Computational and Applied Mathematics 236.15, 2012, pp. 3741–3750

[Bun14] **Bungartz**, H.-J.; **Zimmer**, S.; **Buchholz**, M.; **Pflüger**, D.: *Modeling and simulation: an application-oriented introduction*, Springer-Verlag Berlin Heidelberg, 2014, isbn: 978-3-662-51842-7

[Bun92] **Bungartz**, H.-J.: *Dünne Gitter und deren Anwendung bei der adaptiven Lösung der dreidimensionalen Poisson-Gleichung*, PhD thesis, Technische Universität München, 1992

[Bun98] **Bungartz**, H.-J.: *Finite elements of higher order on sparse grids*, Habilitation thesis, Technische Universität München, 1998

[Bür14] **Bürger**, R.; **Kröker**, I.; **Rohde**, C.: *A hybrid stochastic Galerkin method for uncertainty quantification applied to a conservation law modelling a clarifier-thickener unit*, ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik 94.10, 2014, pp. 793–817

[Caf98] **Caflisch**, R. E. et al.: *Monte Carlo and Quasi-Monte Carlo methods*, Acta numerica 1998, 1998, pp. 1–49

[Car17] **Carpenter**, B. et al.: *Stan: a probabilistic programming language*, Journal of statistical software 76.1, 2017

[Cel15] **Celia**, M. A.; **Bachu**, S.; **Nordbotten**, J.; **Bandilla**, K.: *Status of CO2 storage in deep saline aquifers with emphasis on modeling approaches and practical simulations*, Water Resources Research 51.9, 2015, pp. 6846–6892

[Che15] **Cheng**, Y.; **Du**, D.-Z.; **Zheng**, F.: *A new strongly competitive group testing algorithm with small sequentiality*, Annals of Operations Research 229.1, 2015, pp. 265–286

[Chr02] **Christopher Frey**, H.; **Patil**, S. R.: *Identification and review of sensitivity analysis methods*, Risk analysis 22.3, 2002, pp. 553–578

[Cla09] **Class**, H. et al.: *A benchmark study on problems related to CO2 storage in geologic formations*, Computational Geosciences 13.4, 2009, pp. 409–434

[Cod20] *Publicly available code for scenario-based mass testing for COVID-19*, https://github.com/SC-SGS/covid19-pooling, Accessed on 2021-04-01

[Con15a] **Constantine**, P. G.: *Active subspaces: Emerging ideas for dimension reduction in parameter studies*, SIAM, 2015

[Con15b] **Constantine**, P. G.; **Emory**, M.; **Larsson**, J.; **Iaccarino**, G.: *Exploiting active subspaces to quantify uncertainty in the numerical simulation of the HyShot II scramjet*, Journal of Computational Physics 302, 2015, pp. 1–20

[Con15c] **Constantine**, P. G.; **Zaharatos**, B.; **Campanelli**, M.: *Discovering an active subspace in a single-diode solar cell model*, Statistical Analysis and Data Mining: The ASA Data Science Journal 8.5-6, 2015, pp. 264–273

[Con17a] **Constantine**, P. G.; **Diaz**, P.: *Global sensitivity metrics from active subspaces*, Reliability Engineering & System Safety 162, 2017, pp. 1–13

[Con17b] **Constantine**, P. G.; **Doostan**, A.: *Time-dependent global sensitivity analysis with active subspaces for a lithium ion battery model*, Statistical Analysis and Data Mining: The ASA Data Science Journal 10.5, 2017, pp. 243–262

[Cov20] *Online tool for scenario-based mass testing for COVID-19*, `https://ipvs.informatik.uni-stuttgart.de/sgs/cgi-bin/JA/covid19/`, Accessed on 2021-04-01

[Cox72] **Cox**, M. G.: *The numerical evaluation of B-splines*, IMA J. Appl. Math. 10.2, 1972, pp. 134–149

[Cuc20] **Cucinotta**, D.; **Vanelli**, M.: *WHO declares COVID-19 a pandemic*, Acta Bio Medica Atenei Parmensis 91.1, 2020, pp. 157–160, doi:10.23750/abm.v91i1.9397

[Cur66] **Curry**, H. B.; **Schoenberg**, I. J.: *On pólya frequency functions IV: the fundamental spline functions and their limits*, Journal danalyse mathématique 17.1, 1966, pp. 71–107

[Dak14] **Adams**, B. M.; **al**, M. S. E. et: *Dakota, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis*, Sandia Technical Report, SAND2014-4633, July 2014. Updated May 2019

[Dav63] **Davis**, P. J.: *Interpolation and approximation*, 1963

[De 66] **De Boor**, C.: *The method of projection as applied to the numerical solution of two point boundary value problems using cubic splines*, PhD thesis, Citeseer, 1966

[De 72] **De Boor**, C.: *On calculating with B-splines*, J. Approx. Theory 6.1, 1972, pp. 50–62

[DeB78] **De Boor**, C.: *A practical guide to splines*, vol. 27, Springer New York, 1978

[DeC63] **De Casteljau**, P.: *Courbes et surfaces à pôles*, André Citroën, Automobiles SA, Paris 66, 1963

[Der86] **Der Kiureghian**, A.; **Liu**, P.-L.: *Structural reliability under incomplete probability information*, Journal of Engineering Mechanics 112.1, 1986, pp. 85–104

[Dor43] **Dorfman**, R.: *The detection of defective members of large populations*, The Annals of Mathematical Statistics 14.4, 1943, pp. 436–440

[Du00] **Du**, D.; **Hwang**, F. K.: *Combinatorial group testing and its applications*, vol. 12, World Scientific, 2000

[Dua87] **Duane**, S.; **Kennedy**, A. D.; **Pendleton**, B. J.; **Roweth**, D.: *Hybrid Monte Carlo*, Physics letters B 195.2, 1987, pp. 216–222

[Ebi07] **Ebigbo**, A.; **Class**, H.; **Helmig**, R.: *CO2 leakage through an abandoned well: problem-oriented benchmarks*, Computational Geosciences 11.2, 2007, pp. 103–115

[Eld09a] **Eldred**, M.: *Recent advances in non-intrusive polynomial chaos and stochastic collocation methods for uncertainty analysis and design*, 50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference 17th AIAA/ASME/AHS Adaptive Structures Conference 11th AIAA No, 2009, p. 2274

[Eld09b] **Eldred**, M.; **Burkardt**, J.: *Comparison of non-intrusive polynomial chaos and stochastic collocation methods for uncertainty quantification*, 47th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition, 2009, p. 976

[Fab14] **Faber**, G.: *Über die interpolatorische Darstellung stetiger Funktionen*, Jahresber. Deutsch. Math. Verein 23, 1914, pp. 192–210

[Far20] **Fargion**, B. I.; **Fargion**, D.; **Lucentini**, P. G. D. S.; **Habib**, E.: *Purim: a rapid method with reduced cost for massive detection of COVID-19*, arXiv preprint arXiv:2003.11975, 2020

[Feu10] **Feuersänger**, C.: *Sparse grid methods for higher dimensional approximation.* PhD thesis, University of Bonn, 2010

[Flu88] **Flury**, B.: *Common principal components & related multivariate models*, John Wiley & Sons, Inc., 1988

[Fra15] **Franzelin**, F.; **Diehl**, P.; **Pflüger**, D.: *Non-intrusive uncertainty quantification with sparse grids for multivariate peridynamic simulations*, Meshfree Methods for Partial Differential Equations VII, Springer, 2015, pp. 115–143

[Fra17] **Franzelin**, F.: *Data-Driven Uncertainty Quantification for Large-Scale Simulations*, PhD thesis, University of Stuttgart, IPVS, 2017

[Fri83] **Friedman**, J. H.; **Grosse**, E.; **Stuetzle**, W.: *Multidimensional additive spline approximation*, SIAM J. Sci. Comput. 4.2, 1983, pp. 291–301

[Fri91] **Friedman**, J. H. et al.: *Multivariate adaptive regression splines*, Ann. Stat. 19.1, 1991, pp. 1–67

[Gar01] **Garcke**, J.; **Griebel**, M.; **Thess**, M.: *Data mining with sparse grids*, Computing 67.3, 2001, pp. 225–253, doi:10.1007/s006070170007

[Gar12] **Garcke**, J.: *Sparse grids in a nutshell*, Sparse Grids and Applications, Springer, 2012, pp. 57–80

[Gas] **Gasperson**, R.: *We are thinking about you, Haiku by Robert Gasperson*, https://hubpages.com/politics/haiku-for-2011-japanese-tsunami-charity, Accessed: 2021-04-01

[Gel06a] **Gelman**, A.: *Prior distributions for variance parameters in hierarchical models*, Bayesian analysis 1.3, 2006, pp. 515–534

[Gel06b] **Gelman**, A.; **Hill**, J.: *Data analysis using regression and multilevel/hierarchical models*, Cambridge university press, 2006

[Gel13] **Gelman**, A. et al.: *Bayesian data analysis*, CRC press, 2013

[Ger03] **Gerstner**, T.; **Griebel**, M.: *Dimension–adaptive tensor–product quadrature*, Computing 71.1, 2003, pp. 65–87

[Ger98] **Gerstner**, T.; **Griebel**, M.: *Numerical integration using sparse grids*, Numerical Algorithms 18.3–4, 1998, pp. 209–232, doi:10.1023/A:1019129717644

[Gla12] **Glazyrin**, A.: *Lower bounds for the simplexity of the n-cube*, Discrete Mathematics 312.24, 2012, pp. 3656–3662

[Gla16] **Glaser**, P.; **Schick**, M.; **Petridis**, K.; **Heuveline**, V.: *Comparison between a polynomial chaos surrogate model and Markov chain Monte Carlo for inverse uncertainty quantification based on an electric drive test bench*, ECCOMAS Congress, vol. 2016, 2016

[Gla17] **Glaser**, P.; **Kosmas**, P.; **Schick**, M.; **Heuveline**, V.: *Modeling of a likelihood function based on a global sensitivity analysis*, PAMM 17.1, 2017, pp. 719–720

[Gre18] **Grey**, Z. J.; **Constantine**, P. G.: *Active subspaces of airfoil shape parameterizations*, AIAA Journal 56.5, 2018, pp. 2003–2017

[Gri05] **Griebel**, M.: *Sparse grids and related approximation schemes for higher dimensional problems*, Citeseer, 2005

[Gri90] **Griebel**, M.; **Schneider**, M.; **Zenger**, C.: *A combination technique for the solution of sparse grid problems*, 1990

[Had02] **Hadamard**, J.: *Sur les problèmes aux dérivées partielles et leur signification physique*, Princeton university bulletin, 1902, pp. 49–52

[Han20] **Hanel**, R.; **Thurner**, S.: *Boosting test-efficiency by pooled testing strategies for SARS-CoV-2*, arXiv preprint arXiv:2003.09944, 2020

[Har83] **Harper**, W. V.; **Gupta**, S. K.: *Sensitivity/uncertainty analysis of a borehole scenario comparing latin hypercube sampling and deterministic sensitivity approaches*, tech. rep., Battelle Memorial Inst., 1983

[Has18] **Haszeldine**, R. S.; **Flude**, S.; **Johnson**, G.; **Scott**, V.: *Negative emissions technologies and carbon capture and storage to achieve the Paris agreement commitments*, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 376.2119, 2018, p. 20160447

[Has70] **Hastings**, W. K.: *Monte Carlo sampling methods using Markov chains and their applications*, 1970

[Hin83] **Hindmarsh**, A. C.: *ODEPACK, a systematized collection of ODE solvers*, Scientific computing, 1983, pp. 55–64

[Hno20] *Publicly available code for Hokkaido-Nansei-Oki tsunami simulation*, `https://github.com/SC-SGS/tsunami-uq`, Accessed on 2021-07-23

[Hof14] **Hoffman**, M. D.; **Gelman**, A.: *The no-u-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo.* J. Mach. Learn. Res. 15.1, 2014, pp. 1593–1623

[Hog20] **Hogan**, C. A.; **Sahoo**, M. K.; **Pinsky**, B. A.: *Sample pooling as a strategy to detect community transmission of sars-cov-2*, JAMA 323.19, 2020, pp. 1967–1969, `doi:10.1001/jama.2020.5445`

[Höl01] **Höllig**, K.; **Reif**, U.; **Wipper**, J.: *Weighted extended B-spline approximation of dirichlet problems*, SIAM J. Numer. Anal. 39.2, 2001, pp. 442–462

[Höl03] **Höllig**, K.: *Finite element methods with B-splines*, SIAM, 2003

[Höl13] **Höllig**, K.; **Hörner**, J.: *Approximation and Modeling with B-Splines*, Philadelphia: SIAM, 2013

[Hug05] **Hughes**, T. J.; **Cottrell**, J. A.; **Bazilevs**, Y.: *Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement*, Computer methods in applied mechanics and engineering 194.39-41, 2005, pp. 4135–4195

[Hug96] **Hughes**, R. B.; **Anderson**, M. R.: *Simplexity of the cube*, Discrete Mathematics 158.1-3, 1996, pp. 99–150

[IPC05] **IPCC**: *Special report on carbon dioxide capture and storage*, Cambridge University Press, 2005

[IPC11] **IPCC**: *Special report on renewable energy sources and climate change mitigation*, Cambridge University Press, 2011

[Jef15] **Jefferson**, J. L.; **Gilbert**, J. M.; **Constantine**, P. G.; **Maxwell**, R. M.: *Active subspaces for sensitivity analysis and dimension reduction of an integrated hydrologic model*, Computers & Geosciences 83, 2015, pp. 127–138

[JHU] *COVID-19 dashboard by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University (JHU)*, `https://coronavirus.jhu.edu/map.html`, Accessed: 2021-04-01

[Joh19] **John**, D. N.; **Schick**, M.; **Heuveline**, V.: *Learning model discrepancy of an electric motor with Bayesian inference*, Multidisciplinary Digital Publishing Institute Proceedings, vol. 33, 1, 2019, p. 11

[Joh21a] **John**, D.: *Uncertainty Quantification for an electric motor inverse problem tackling the model discrepancy challenge*, PhD thesis, Ruprecht-Karls-Universität Heidelberg, 2021

[Joh21b] **John**, D.; **Stohrer**, L.; **Schillings**, C.; **Schick**, M.; **Vincent**, H.: *Hierarchical surrogate-based Approximate Bayesian Computation for an electric motor test bench*, Journal of Computational Physics, 2021, to be published

[Köp18] **Köppel**, M. et al.: *Comparison of data-driven uncertainty quantification methods for a carbon dioxide storage benchmark scenario*, Computational Geosciences, 2018, doi:10.1007/s10596-018-9785-x

[KöpDa] **Köppel**, M. et al.: *Datasets and executables of data-driven uncertainty quantification benchmark in carbon dioxide storage*, doi:10.5281/zenodo.933827

[Kur05] **Kurganov**, A.; **Petrova**, G.: *Central-upwind schemes on triangular grids for hyperbolic systems of conservation laws*, Numerical Methods for PDE 21.3, 2005, pp. 536–552

[Lab20] **LabCorp**: *Accelerated Emergency Use Authorization (EUA) Summary COVID-19 RT-PCR Test*, tech. rep., Laboratory Corporation of America, 2020

[Mar17] **Martin**, F.: *WEB-spline Approximation and Collocation for Singular and Time-dependent Problems*, Shaker Verlag, 2017

[Mar19] **Margossian**, C. C.: *A review of automatic differentiation and its efficient implementation*, Wiley interdisciplinary reviews: data mining and knowledge discovery 9.4, 2019, e1305

[Mat01] **Matsuyama**, M.; **Tanaka**, H.: *An experimental study of the highest run-up height in the 1993 Hokkaido Nansei-Oki earthquake tsunami*, National Tsunami Hazard Mitigation Program Review and International Tsunami Symposium (ITS), 2001, pp. 879–889

[McC20] **McClellan**, M.; **Gottlieb**, S.; **Mostashari**, F.; **Rivers**, C.; **Silvis**, L.: *A national COVID-19 surveillance system: achieving containment*, Margolis Center for Health Policy. https://healthpolicy. duke. edu, 2020

[McK] **McKie**, R.: *Carbon capture is vital to meeting climate goals, scientists tell green critics*, https://www.theguardian.com/environment/2021/jan/16/carbon-capture-vital-meeting-climate-goals-scientists-cut-emissions, 16 Jan 2021, Accessed: 2021-04-01, The Guardian

[Met53] **Metropolis**, N.; **Rosenbluth**, A. W.; **Rosenbluth**, M. N.; **Teller**, A. H.; **Teller**, E.: *Equation of state calculations by fast computing machines*, The journal of chemical physics 21.6, 1953, pp. 1087–1092

[Met87] **Metropolis**, N.: *The beginning of the Monte Carlo method*, Los Alamos Science 15, 1987, pp. 125–130

[Naj16] **Najm**, H.; **Chowdhary**, K.: *Inference given summary statistics*, Handbook of Uncertainty Quantification, Cham: Springer International Publishing, 2016, pp. 1–35, isbn:978-3-319-11259-6, doi:10.1007/978-3-319-11259-6_68-1, https://doi.org/10.1007/978-3-319-11259-6_68-1

[Nar20] **Narayanan**, K. et al.: *Pooling RT-PCR or NGS samples has the potential to cost-effectively generate estimates of COVID-19 prevalence in resource limited environments*, medRxiv, 2020, doi:10.1101/2020.04.03.20051995

[Nea11] **Neal**, R. M. et al.: *MCMC using Hamiltonian dynamics*, Handbook of Markov chain Monte Carlo 2.11, 2011, p. 2

[Nej20] *Rapid response to an outbreak in Qingdao, China*, https://www.nejm.org/doi/full/10.1056/NEJMc2032361, Accessed: 2021-04-01

[Nie05] **Nielsen**, O.; **Roberts**, S.; **Gray**, D.; **McPherson**, A.; **Hitchman**, A., et al.: *Hydrodymamic modelling of coastal inundation*, 2005, pp. 518–523

[Nob08] **Nobile**, F.; **Tempone**, R.; **Webster**, C. G.: *A sparse grid stochastic collocation method for partial differential equations with random input data*, SIAM Journal on Numerical Analysis 46.5, 2008, pp. 2309–2345

[Noc06] **Nocedal**, J.; **Wright**, S.: *Numerical optimization*, Springer Science & Business Media, 2006

[Nor11] **Nordbotten**, J. M.; **Dahle**, H. K.: *Impact of the capillary fringe in vertically integrated models for CO2 storage*, Water Resources Research 47.2, 2011

[Nor20] **Noriega**, R.; **Samore**, M.: *Increasing testing throughput and case detection with a pooled-sample Bayesian approach in the context of COVID-19*, bioRxiv, 2020

[Nör24] **Nörlund**, N. E.: *Vorlesungen über Differenzenrechnung*, Grundlehren der mathematischen Wissenschaften 13, Springer, 1924, doi:10.1007/978-3-642-50824-0

[Ola11] **Oladyshkin**, S.; **Class**, H.; **Helmig**, R.; **Nowak**, W.: *An integrative approach to robust design and probabilistic risk assessment for CO2 storage in geological formations*, Computational Geosciences 15.3, 2011, pp. 565–577

[Ola12] **Oladyshkin**, S.; **Nowak**, W.: *Data-driven uncertainty quantification using the arbitrary polynomial chaos expansion*, Reliability Engineering & System Safety 106, 2012, pp. 179–190

[Pan97] **Pannell**, D. J.: *Sensitivity analysis of normative economic models: theoretical framework and practical strategies*, Agricultural economics 16.2, 1997, pp. 139–152

[Pas16] *Python Active Subspace Utility Library (PASUL)*, https://github.com/paulcon/active_subspaces, [Online; Accessed: 2021-04-01], software provided under MIT license, 2016

[Peh14] **Peherstorfer**, B.; **Pflüger**, D.; **Bungartz**, H.-J.: *Density estimation with adaptive sparse grids for large data sets*, Proceedings of the 2014 SIAM International Conference on Data Mining, ed. by **Zaki**, M. et al., SIAM, 2014, pp. 443–451, doi:10.1137/1.9781611973440.51

[Peh15] **Peherstorfer**, B.; **Kowitz**, C.; **Pflüger**, D.; **Bungartz**, H.-J.: *Selected recent applications of sparse grids*, Numerical Mathematics: Theory, Methods and Applications 8.1, 2015, pp. 47–77, doi:10.4208/nmtma.2015.w05si

[Pet20a] **Peto**, J.: *COVID-19 mass testing facilities could end the epidemic rapidly*, BMJ 368, 2020, doi:10.1136/bmj.m1163

[Pet20b] **Peto**, J. et al.: *Universal weekly testing as the UK COVID-19 lockdown exit strategy.* Lancet (London, England), 2020

[Pfa16] **Pfander**, D.; **Heinecke**, A.; **Pflüger**, D.: *A new subspace-based algorithm for efficient spatially adaptive sparse grid regression, classification and multi-evaluation*, Sparse Grids and Applications-Stuttgart 2014, Springer, 2016, pp. 221–246

[Pfa19] **Pfander**, D.; **Daiss**, G.; **Pflüger**, D.: *Heterogeneous distributed big data clustering on sparse grids*, Algorithms 12.3, 2019, p. 60

[Pfl10] **Pflüger**, D.: *Spatially Adaptive Sparse Grids for High-Dimensional Problems*, München: Verlag Dr. Hut, 2010

[Pfl20] **Pflüger**, D.; **Wolff**, T. de; **Rehme**, M. F.; **Heuer**, J.; **Bittner**, M.-I.: "Evaluation of pool-based testing approaches to enable population-wide screening for Covid-19," U.S. pat., Patent Pending, U.S. Application No. 63/033,501, 2020

[Pie12] **Piegl**, L.; **Tiller**, W.: *The NURBS book*, Springer Science & Business Media, 2012

[Pil04] **Pilcher**, C. D. et al.: *Frequent detection of acute primary HIV infection in men in Malawi*, Aids 18.3, 2004, pp. 517–524

[Pil05] **Pilcher**, C. D. et al.: *Detection of acute infections during HIV testing in North Carolina*, New England Journal of Medicine 352.18, 2005, pp. 1873–1883

[Pol12] **Polson**, N. G.; **Scott**, J. G., et al.: *On the half-cauchy prior for a global scale parameter*, Bayesian Analysis 7.4, 2012, pp. 887–902

[Qua10] **Quarteroni**, A.; **Sacco**, R.; **Saleri**, F.: *Numerical mathematics*, vol. 37, Springer Science & Business Media, 2010

[Ram88] **Ramsay**, J. O. et al.: *Monotone regression splines in action*, Statistical science 3.4, 1988, pp. 425–441

[Rob15] **Roberts**, S.; **Nielsen**, O.; **Gray**, D.; **Sexton**, J.; **Davies**, G.: *ANUGA User Manual*, Geoscience Australia, 2015

[Roc04] **Rochelle**, C. A.; **Czernichowski-Lauriol**, I.; **Milodowski**, A.: *The impact of chemical reactions on CO2 storage in geological formations: a brief review*, Geological Society, London, Special Publications 233.1, 2004, pp. 87–106

[Rom20] **Romer**, P.; **Shah**, R.: *Testing is our way out*, Wall Street Journal 3, 2020

[Ros52] **Rosenblatt**, M.: *Remarks on a multivariate transformation*, The annals of mathematical statistics 23.3, 1952, pp. 470–472

[Ros95] **Rosenthal**, J. S.: *Minorization conditions and convergence rates for Markov chain Monte Carlo*, Journal of the American Statistical Association 90.430, 1995, pp. 558–566

[Run01] **Runge**, C.: *Über empirische Funktionen und die Interpolation zwischen äquidistanten Ordinaten*, Zeitschrift für Mathematik und Physik 46.224-243, 1901, p. 20

[Rus10] **Russi**, T. M.: *Uncertainty quantification with experimental data and complex system models*, PhD thesis, UC Berkeley, 2010

[Sal04] **Saltelli**, A.; **Tarantola**, S.; **Campolongo**, F.; **Ratto**, M.: *Sensitivity analysis in practice: a guide to assessing scientific models*, vol. 1, Wiley Online Library, 2004

[Sal08] **Saltelli**, A. et al.: *Global sensitivity analysis: the primer*, John Wiley & Sons, 2008

[Sal16] **Salvatier**, J.; **Wiecki**, T. V.; **Fonnesbeck**, C.: *Probabilistic programming in Python using PyMC3*, PeerJ Computer Science 2, 2016, e55

[Sch18] **Schober**, P.: *Solving dynamic portfolio choice models in discrete time using spatially adaptive sparse grids*, Sparse Grids and Applications – Miami 2016, ed. by **Garcke**, J.; **Pflüger**, D.; **Webster**, C. G.; **Zhang**, G., Lecture Notes in Computational Science and Engineering 123, Springer, 2018, pp. 135–173, doi:10.1007/978-3-319-75426-0_7

[Sch19] **Schnick**, C.: *Vergleich von Dimensionsreduktionsmethoden für Surrogate auf Dünnen Gittern*, Bachelor's Thesis superv. by Dirk Pflüger and Michael Rehme, University of Stuttgart, 2019

[Sch46] **Schoenberg**, I. J.: *Contributions to the problem of approximation of equidistant data by analytic functions*, Q. Appl. Math. 4, 1946, 45–99 and 112–141

[Sch53] **Schoenberg**, I. J.; **Whitney**, A.: *On pólya frequence functions. III. the positivity of translation determinants with an application to the interpolation problem by spline curves*, Trans. Am. Math. Soc. 74.2, 1953, pp. 246–259

[Sch97] **Schaller**, R. R.: *Moore's law: past, present and future*, IEEE Spectrum 34.6, 1997, pp. 52–59

[Seo15] **Seo**, D. H.; **Whang**, D. H.; **Song**, E. Y.; **Han**, K. S.: *Occult hepatitis B virus infection and blood transfusion*, World journal of hepatology 7.3, 2015, p. 600

[SGpp] *SG^{++} sparse grid framework*, `https://github.com/SGpp/SGpp`, [Online; Accessed: 2021-04-01]

[Sha20] **Shani-Narkiss**, H.; **Gilday**, O. D.; **Yayon**, N.; **Landau**, I. D.: *Efficient and practical sample pooling high-throughput PCR diagnosis of COVID-19*, medRxiv, 2020

[Sib20] **Sibiri**, H.; **Zankawah**, S.; **Prah**, D.: *Coronavirus diseases 2019 (COVID-19) response: highlights of Ghanas scientific and technological innovativeness and breakthroughs*, 2020

[Sic11] **Sickel**, W.; **Ullrich**, T.: *Spline interpolation on sparse grids*, Applicable Analysis 90.3–4, 2011, pp. 337–383, `doi:10.1080/00036811.2010.495336`

[Sil12] **Silver**, N.: *The signal and the noise: why so many predictions fail - but some don't*, Penguin Publishing Group, 2012, `isbn:9781101595954`

[Sin20] **Sinnott-Armstrong**, N.; **Klein**, D.; **Hickey**, B.: *Evaluation of group testing for SARS-CoV-2 RNA*, medRxiv, 2020

[Smo63] **Smolyak**, S. A.: *Quadrature and interpolation formulas for tensor products of certain classes of functions*, Doklady Akademii Nauk, vol. 148, Russian Academy of Sciences, 1963, pp. 1042–1045

[Sob59] **Sobel**, M.; **Groll**, P. A.: *Group testing to eliminate efficiently all defectives in a binomial sample*, Bell System Technical Journal 38.5, 1959, pp. 1179–1252

[Sta18] **Stan Development Team**: *Stan modeling language users guide and reference manual*, 2018, Version 2.24, `http://mc-stan.org`

[Ste66] **Stein**, P.: *A note on the volume of a simplex*, The American Mathematical Monthly 73.3, 1966, pp. 299–301

[Sto18] **Stohrer**, L.: *Uncertainty Quantification for virtual prototypes*, MA thesis, University of Mannheim, 2018

[Str94] **Strøm**, K.: *On convolutions of B-splines*, Journal of Computational and Applied Mathematics 55.1, 1994, pp. 1–29

[Sul15] **Sullivan**, T. J.: *Introduction to uncertainty quantification*, vol. 63, Springer, 2015

[Syn07] **Synolakis**, C.: *Standards, criteria, and procedures for NOAA evaluation of tsunami numerical models*, `https://nctr.pmel.noaa.gov/benchmark/`, 2007, Accessed: 2021-04-01

[Täu20] **Täufer**, M.: *Rapid, large-scale, and effective detection of COVID-19 via non-adaptive testing*, BioRxiv, 2020

[The16] **Theano Development Team**: *Theano: A Python framework for fast computation of mathematical expressions*, arXiv e-prints abs/1605.02688, 2016, http://arxiv.org/abs/1605.02688

[Tik77] **Tikhonov**, A. N.; **Arsenin**, V. Y.: *Solutions of ill-posed problems*, New York, 1977, pp. 1–30

[Tol18] **Toliyat**, H. A.; **Kliman**, G. B.: *Handbook of electric motors*, vol. 120, CRC press, 2018

[Val16] **Valentin**, J.; **Pflüger**, D.: *Hierarchical gradient-based optimization with B-splines on sparse grids*, Sparse Grids and Applications-Stuttgart 2014, Springer, 2016, pp. 315–336

[Val18] **Valentin**, J.; **Sprenger**, M.; **Pflüger**, D.; **Röhrle**, O.: *Gradient-based optimization with B-splines on sparse grids for solving forward-dynamics simulations of three-dimensional, continuum-mechanical musculoskeletal system models*, International journal for numerical methods in biomedical engineering 34.5, 2018, e2965

[Val19] **Valentin**, J.: *B-splines on Sparse Grids. Algorithms and Application to Higher-Dimensional Optimization*, PhD thesis, University of Stuttgart, IPVS, 2019

[ValTex] **Valentin**, J.: *Unofficial Latex template for PhD theses of the University of Stuttgart*, https://github.com/valentjn/uni-stuttgart-phdthesis-template, Accessed: 2021-04-01

[Vir20] **Virtanen**, P. et al.: *SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python*, Nature Methods 17, 2020, pp. 261–272, doi:10.1038/s41592-019-0686-2

[Wen04] **Wendland**, H.: *Scattered data approximation*, vol. 17, Cambridge university press, 2004

[Wen95] **Wendland**, H.: *Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree*, Advances in computational Mathematics 4.1, 1995, pp. 389–396

[Wid08] **Widmer**, G.; **Hiptmair**, R.; **Schwab**, C.: *Sparse adaptive finite elements for radiative transfer*, Journal of Computational Physics 227.12, 2008, pp. 6071–6105

[Wie38] **Wiener**, N.: *The homogeneous chaos*, American Journal of Mathematics 60.4, 1938, pp. 897–936

[Wol87] **Wold**, S.; **Esbensen**, K.; **Geladi**, P.: *Principal component analysis*, Chemometrics and intelligent laboratory systems 2.1-3, 1987, pp. 37–52

[Xio19] **Xiong**, W.; **Lu**, H.; **Ding**, J.: *Determination of varying group sizes for pooling procedure*, Computational and mathematical methods in medicine, 2019

[Xiu02] **Xiu**, D.; **Karniadakis**, G. E.: *The Wiener–Askey polynomial chaos for stochastic differential equations*, SIAM journal on scientific computing 24.2, 2002, pp. 619–644

[Yel20] **Yelin**, I. et al.: *Evaluation of COVID-19 RT-qPCR test in multi-sample pools*, Clinical Infectious Diseases, 2020, doi:10.1093/cid/ciaa531

[Zen91] **Zenger**, C.: *Sparse grids*, Parallel Algorithms for Partial Differential Equations: Proceedings of the Sixth GAMM-Seminar, ed. by **Hackbusch**, W., Notes on Numerical Fluid Mechanics 31, Vieweg, 1991, pp. 241–251, isbn:978-3-528-07631-3