8-2017

Efficient Methods for Multidimensional Global Polynomial Approximation with Applications to Random PDEs

Peter A. Jantsch
University of Tennessee, Knoxville, pjantsch@vols.utk.edu

To the Graduate Council:

I am submitting herewith a dissertation written by Peter A. Jantsch entitled "Efficient Methods for Multidimensional Global Polynomial Approximation with Applications to Random PDEs." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Mathematics.

Clayton G. Webster, Major Professor

We have read this dissertation and recommend its acceptance:

Steven Wise, Ohannes Karakashian, Xiaobing Feng, Jack Dongarra

Accepted for the Council:
Dixie L. Thompson

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Efficient Methods for Multidimensional Global Polynomial Approximation with Applications to Random PDEs

A Dissertation Presented for the
Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

Peter A. Jantsch

August 2017

Acknowledgments

First of all, thanks to the doctoral committee, Profs. Dongarra, Feng, Karakashian and Wise, for being so generous with their time. You each are excellent professors, educators, and researchers who shaped my knowledge of, and passion for mathematics.

I am especially grateful to my advisor, Clayton Webster, for his patience and guidance in the accomplishment of this work. Over the past several years working toward this degree, I've been greatly helped by his insight and enthusiasm for new mathematical ideas. And of course, thanks for all the reminders that my life hasn't peaked (at least not yet).

Thanks also to all my collaborators on this work. I must especially acknowledge Aretha Teckentrup and Max Gunzburger for their work on multilevel collocation methods, and Diego Galindo and Guannan Zhang for their collaboration on the accelerated collocation algorithm. Thanks also to Miroslav Stoyanov, Nick Dexter and Hoang Tran for many helpful conversations, discussions, and collaborations.

Thanks to the Department of Mathematics at UT, and especially to Dr. Webster, for financial support during my tenure in Tennessee. Many thanks as well to the wonderful Pam Armentrout. She is an invaluable source of advice and support to every mathematics graduate student.

To all my math-loving friends, especially Brian Allen, Eddie Tu and Michael Kelly; thanks for being there with advice, help and friendship. To my non-math-loving friends, thank you for being able to talk about things other than math! Your friendships helped me through many challenges, and you've all meant so much to me during my time in Knoxville.

Finally, to my family for loving and supporting me in my pursuit of this degree—thank you! This dissertation is especially dedicated to my parents, Steve and Darla Jantsch. I could not have made it this far without you, and I love you both dearly.

Soli Deo gloria

Abstract

In this work, we consider several ways to overcome the challenges associated with polynomial approximation and integration of smooth functions depending on a large number of inputs. We are motivated by the problem of forward uncertainty quantification (UQ), whereby inputs to mathematical models are considered as random variables. With limited resources, finding more efficient and accurate ways to approximate the multidimensional solution to the UQ problem is of crucial importance, due to the "curse of dimensionality" and the cost of solving the underlying deterministic problem.

The first way we overcome the complexity issue is by exploiting the structure of the approximation schemes used to solve the random partial differential equations (PDE), thereby significantly reducing the overall cost of the approximation. We do this first using multilevel approximations in the physical variables, and second by exploiting the hierarchy of nested sparse grids in the random parameter space. With these algorithmic advances, we provably decrease the complexity of collocation methods for solving random PDE problems.

The second major theme in this work is the choice of efficient points for multidimensional interpolation and interpolatory quadrature. A major consideration in interpolation in multiple dimensions is the balance between stability, i.e., the Lebesgue constant of the interpolant, and the granularity of the approximation, e.g., the ability to choose an arbitrary number of interpolation points or to adaptively refine the grid. For these reasons, the Leja points are a popular choice for approximation on both bounded and unbounded domains. Mirroring the best-known results for interpolation on compact domains, we show that Leja points, defined for weighted interpolation on $\mathbb{R}$, have a Lebesgue constant which grows subexponentially in the number of interpolation nodes. Regarding multidimensional quadratures, we show how certain new rules, generated from conformal mappings of classical interpolatory rules, can be used to increase the efficiency in approximating multidimensional integrals. Specifically, we show that the convergence rate for the novel mapped sparse grid interpolatory quadratures is improved by a factor that is exponential in the dimension of the underlying integral.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Mathematical modeling is an important tool for decision making in a diverse array of scientific and engineering fields, as well as manufacturing, economic forecasting, public policy, and many others. The solution of a mathematical model can be viewed as a mapping from input data—e.g., coefficients, forcing terms, initial and boundary conditions, domain geometry—to an output of interest. In practice, the input data may be affected by a large amount of uncertainty due to intrinsic variability or the difficulty in accurately characterizing the physical system. In order to correctly predict the behavior of the system, it is especially pertinent to understand and propagate the effect of the input uncertainty to the output of the simulation, i.e., to the solution of the mathematical model. Such uncertainties can be included in the mathematical model by adopting a probabilistic setting. Given statistical information about the input variables, the goal then is to understand statistics of the solution, e.g. mean and variance, or statistics of some functional of the solutions, e.g. outflow across a boundary. This is called the forward uncertainty quantification (UQ) problem, and these desired outputs are known as quantities of interest (QoI).

One of the important models of forward uncertainty quantification is partial differential equations (PDEs) with random input data. Assuming the random input may be parameterized by some finite dimensional random vector, $\boldsymbol{y} \in \mathbb{R}^N$, the goal in this setting is to find the solution u, which for almost every $\boldsymbol{y}$ satisfies the problem

$$\mathscr{D}(a(\boldsymbol{y}))[u] = f(\boldsymbol{y}) \text{ in } D, \tag{1.1}$$

subject to suitable boundary conditions, where $\mathscr{D}$ is a (physical) differential operator on the domain D, and a and f are random fields (see Chapter 2). Numerical solution of PDEs with random inputs is a large and active research area; see, e.g., the overview [48] and the references cited therein.

From a computational point of view, the major challenges to solving these problems stem from the dimensionality of the input data, and the complexity of the underlying physical model. First, the data could depend strongly on a large number of variables, introducing the "curse of dimensionality." In the general setting described above, this so-called "curse" refers to an exponential relationship between the computational effort required to numerically find the solution $u(\boldsymbol{y})$ and the dimension N of the input parameter. This means that to construct an accurate polynomial approximation, one must take a number of samples, M, which grows quite rapidly with respect to N.

The second major issue is related to the first: each of the samples required by interpolation based methods for random PDEs—and all sampling methods in general, in which the stochastic and deterministic degrees of freedom are uncoupled—may be extremely expensive to compute. As mentioned above, to construct a fully-discrete solution, these sample evaluations require the numerical solution of the underlying deterministic, physical PDE model, which may be nonlinear, time dependent, stiff, or otherwise computationally intensive. This computational effort is multiplied by the possibly large number of samples necessary to construct an accurate interpolant, which may quickly exhaust available computational resources. Furthermore, for many situations it is not totally understood *which* sample points are best to use in multidimensional domains.

Among the myriad approaches to approximating the random dependence of the solution map $u(\boldsymbol{y})$, corresponding to (1.1), the Monte Carlo (MC) method is perhaps the simplest (see, e.g., [33]). This method involves random sampling of the input vector of random variables (also referred to as the *stochastic parameter space*) and the numerical solution of the deterministic PDE at each of the sample points. In addition to the benefits of simple implementation and a natural decoupling of the stochastic and spatial degrees of freedom, MC methods feature a convergence rate that is independent of the dimension of the stochastic space, making it particularly attractive for high-dimensional problems. However, the convergence, $\mathcal{O}(M^{-1/2})$ where M is the number of samples, is in general very slow. Especially in the case the stochastic space is only of moderate dimension and the solution of the PDE or a functional of interest is smooth with respect to the random parameters, better convergence rates can be achieved using more sophisticated methods. Other ensemble-based methods, including quasi-MC and importance sampling (see [69, 53, 88] and the references therein), have been devised to increase convergence rates, e.g., proportional to $M^{-1}\log(M)^{r(N)}$, however, the function $r(N) > 0$ increases with dimension N. Moreover, since both MC and quasi-MC are quadrature techniques for QoIs, neither have the ability to simultaneously provide an approximation to the solution map $\boldsymbol{y} \mapsto u(\boldsymbol{y})$, required by a large class of applications.

In the last decade, two global polynomial approaches have been proposed that often feature fast convergence rates: *intrusive* stochastic Galerkin (SG) methods, based on predefined orthogonal polynomials [41, 99], or best M-term and quasi-optimal approaches [18, 23, 94, 6]; and *non-intrusive* stochastic collocation (SC) methods, based on (sparse) global Lagrange interpolating polynomials [2, 71, 70], orthogonal polynomial basis expansion [29], or even local hierarchical basis functions [47, 64]. These methods converge rapidly when the PDE solution $u(\boldsymbol{y})$ is highly regular with respect to $\boldsymbol{y}$, a property evident in a wide class of high-dimensional applications.

Stochastic Galerkin methods based on global polynomials [41, 99] seek an approximation to the solution map $u(\boldsymbol{y})$ through projection into a given multidimensional polynomial space. The drawback to this method is that this projection is done simultaneously with the Galerkin projection of the physical problem, leading to large linear systems which couple the physical and random degrees of freedom. Though they feature spectral rates of convergence, the computational effort required to solve the Stochastic Galerkin systems is generally only feasible in for simpler problems (1.1). The work [27] shows that in terms of computational work versus error, Galerkin methods in general fall behind non-intrusive interpolation methods in all but the simplest cases.

Stochastic collocation (SC) methods [2, 71, 70] are similar to MC methods in the sense that they involve only the independent solution of a sequence of deterministic PDEs at given sample points in the stochastic space. However, rather than approximating QoIs through random sample averages, SC methods attempt to reconstruct the coefficients to a (global) polynomial approximation to the function $u(\boldsymbol{y})$ only through these point values. This reconstruction is commonly based on (sparse) Lagrange interpolation [2, 71, 70], discrete L^2 projections [65, 66], or compressed sensing [19, 79, 80, 34]. For problems where the solution is a smooth function of the random input variables and the dimension of the stochastic space is moderate, SC methods based on global polynomials have been shown to converge much faster than MC methods [2, 71, 70].

With this motivation in mind, this work considers the problem of efficient approximation of multi-dimensional functions and integrals by global polynomial methods. Our contributions to this effort may be divided into roughly two main avenues of thought: Part I looks at how to exploit the structure of fully discrete stochastic collocation solutions to drastically—and provably—reduce the computational complexity of solving random PDEs, and thus mitigate the curse of dimensionality. In Part II, we explore the problem of choosing "good" points for both multidimensional interpolation and interpolatory quadrature. Here we take a step back from the random PDE setting, and consider just the problem of multidimensional approximation, noting that the analysis easily applies to collocation methods for solving random PDEs in the interpolation case, and in the quadrature case the analogy is to quadrature approximation of multidimensional integral QoIs. "Good points" in the interpolation setting means that the points have a Lebesgue constant that grows at a reasonable rate, and hence can be used to construct an accurate approximation with few samples. In this respect, we prove that the Leja sequence is a promising point set for interpolation. In the quadrature setting, we show that we can improve the convergence rates for multidimensional quadratures by using conformal mappings to transform classical interpolatory quadrature rules.

1.1 Complexity of Stochastic Collocation Methods

As described above, this work focuses on methods of multidimensional interpolation. In our case, the dependence of the solution u of the random PDE (1.1) on the multidimensional parameter $\boldsymbol{y} \in \mathbb{R}^N$ is approximated via a global polynomial interpolation scheme based on evaluations of u. We can justify the choice of global polynomials in the situation where the parameter dependence of u is very smooth, as is the case in the parametric PDEs considered in this work. Note the the regularity requirements could be relaxed when the constructions use local polynomial bases such as wavelets, splines, etc; see [47, 38]. A typical multidimensional polynomial interpolant used in this work can be written

$$u(\boldsymbol{y}) \approx \mathcal{I}_M[u](\boldsymbol{y}) = \sum_{j=1}^{M} \boldsymbol{c}_j \Psi_j(\boldsymbol{y}), \tag{1.2}$$

where the $\{\Psi_j\}_{j=1}^M$ are a global polynomial basis, and the M coefficients $\{\boldsymbol{c}_j\}_{j=1}^M$ are determined by the evaluation of u at certain sample points, i.e., $\{u(\boldsymbol{y}_j)\}_{j=1}^M$. Here we note that in the case of random PDEs, the "evaluations" $\{u(\boldsymbol{y}_j)\}_{j=1}^M$, and hence the coefficients $\{\boldsymbol{c}_j\}_{j=1}^M$, are actually functions from the solution space of the deterministic problem, e.g., $u(\boldsymbol{y}) \in H_0^1(D)$ for almost every $\boldsymbol{y}$. Moreover, as solutions to PDEs, in practice these sample evaluations are only computed approximately, and depending on the underlying model, may be quite expensive to approximate.

1.1.1 Multilevel Methods for Stochastic Problems

In Chapter 4, we introduce a multilevel stochastic collocation (MLSC) approach for reducing the computational cost incurred by standard, i.e., single level, SC methods. Drawing inspiration from multigrid solvers for linear equations, the main idea behind multilevel methods is to utilize a hierarchical sequence of spatial approximations to the underlying PDE model that are then combined with a related sequence of stochastic discretizations, i.e., the interpolant (1.2) for several different values of M, in such a way as to minimize computational cost.

Starting with the pioneering works [52] in the field of integral equations and [42] in the field of computational finance, the multilevel approach has been successfully applied to many applications of MC methods; see, e.g., [5, 15, 26, 44, 43, 54, 67]. The MLSC method proposed in this chapter is similar to the construction found in [8], where the authors propose to adapt the resolution of the spatial and stochastic discretizations to reduce the total degrees of freedom. In contrast, our construction provides the flexibility of optimizing the interpolation operators used at each level of discretization to minimize computational cost. Our method is also similar to the multilevel quadrature approximations of moments of the solution studied in [50, 51], which consider quasi-MC, polynomial chaos and collocation schemes. However, our focus is on the analysis of the computational complexity of the multilevel interpolation algorithms which also includes results for functionals of the solution. In particular, we prove new interpolation error bounds on functionals of the solution that are needed for the analysis of the MLSC methods.

Our major contribution to the area of multilevel methods, described in Chapter 4, is to provide a rigorous convergence and computational cost analysis of the novel multilevel stochastic collocation method in the case of elliptic equations, demonstrating its advantages compared to standard single-level stochastic collocation approximations (1.2), as well as multilevel MC methods. We also provide numerical results which corroborate the theory, and discuss practical implementation issues.

1.1.2 Acceleration of Stochastic Collocation Methods

The dominant cost in applying any non-intrusive approach such as (1.2) lies in the solution of the underlying linear/nonlinear PDEs (1.1) for a large set of values of $\boldsymbol{y}$. In practice, solutions to the deterministic PDEs are often computed using iterative solvers, e.g., conjugate gradient (CG) methods for symmetric positive-definite linear systems, generalized minimal residual method (GMRES) for non-symmetric linear systems [81], and fixed-point iteration methods [78] for nonlinear PDEs. Several methods for improving the performance of iterative solvers have been proposed, especially subspace and preconditioner methods for iterative Krylov solvers. A strategy utilizing shared search directions for solving a collection of linear systems based on the CG method is proposed in [13]. In [74], a technique called *Krylov recycling* was introduced to solve sets of linear systems sequentially, based on ideas adapted from restarted and truncated GMRES (see [83] and the references therein). We refer to [56, 40, 77, 30, 45] for applications of improved Krylov solvers and preconditioners in SG approximation.

On the other hand, for a general iterative method, improved initial approximations can also significantly reduce the number of iterations required to reach a prescribed accuracy. A sequential orthogonal expansion is utilized in [40, 75], such that a low resolution solution provides an initial guess for the solution of the system with an enriched basis. However, at each step, all the expansion coefficients must be explicitly recomputed, resulting in increased costs. In [45], an extension of a mean-based preconditioner is applied to each linear system in the SC approach, wherein the solution of the j-th system is given as the initial vector for the $(j+1)$-th system. This approach, as well as the Krylov recycling method, imposes a full ordering of the linear systems that appear in the SC approximation, rather than the loose "level-by-level" ordering we adopt.

In Chapter 5, we propose to accelerate, i.e., *to improve the computational efficiency*, of non-intrusive approximations, focusing on SC approaches that construct a sequence of multi-dimensional Lagrange interpolants in a hierarchical sequence of polynomial spaces. As opposed to the multilevel methods described above, which reduce the overall computational burden by taking advantage of a hierarchical spatial approximation, our approach exploits the structure of the SC interpolant to accelerate the solution of the underlying ensemble of deterministic solutions. Specifically, we predict the solution of the parametrized PDE at each collocation point using a previously assembled lower fidelity interpolant constructed on a subset of the high fidelity collocation grid. We then use this prediction to provide deterministic (linear/nonlinear) iterative solvers with initial approximations which continue to improve as the algorithm progresses through the levels of the interpolant. As a particular application, we pose this acceleration technique in the context of hierarchical SC methods that employ sparse tensor products of globally defined Lagrange polynomials [71, 70], on nested one-dimensional Clenshaw-Curtis abscissas. However, the same idea can be extended to other non-intrusive collocation approaches including orthogonal polynomials [99], as well as piecewise wavelet polynomials expansions [11, 47].

The major result of Chapter 5 is to prove that this accelerated collocation algorithm provides a reduction in computational complexity versus methods employing a naive iterative solver approach. We also apply a similar technique to provide good preconditioners at a reduced cost. Numerical examples for random PDEs with both linear and nonlinear underlying physical problem are given to support the theory.

1.2 Efficient point sets for Multidimensional Interpolation and Interpolatory Quadrature

The second line of thought, considered in Part II, explores the problem of choosing good points for multidimensional interpolation and interpolatory quadrature. As mentioned above, rather than the random PDE setting above, in Chapters 6 and 7 we only consider the problem of multidimensional approximation, noting that the analysis easily applies to collocation methods for solving random PDEs, or for computing multidimensional integrals for QoIs in the quadrature case.

1.2.1 Polynomials and Potential Theory

In the approximation of higher dimensional interpolation problems we may be willing to sacrifice some stability, i.e., allow a larger growth-rate for the Lebesgue constant, in exchange for more flexibility in choosing the number of multi-dimensional interpolation points. This flexibility is lacking in the standard Smolyak sparse grids based on Gauss–Legendre and Clenshaw–Curtis abscissa, where, especially in higher dimensions, the size of the set of multidimensional interpolation points grows very rapidly as the fidelity of the sparse grid approximation is increased. A popular choice in recent years are the Leja points [68]. For the compact domain $[-1,1] \subset \mathbb{R}$, these are defined recursively: given a point y_0, for $n = 1, 2, \ldots,$ define the next Leja point as

$$y_n = \arg\max_{y\in[-1,1]} \prod_{j=0}^{n-1} |y - y_j| \,. \tag{1.3}$$

There is still some ambiguity in this definition, since the maximum may be attained at several points. For our purposes, we may choose any maximizer y_n without affecting the analysis.

In addition, by introducing an appropriate weight function $w : \mathbb{R} \to [0,1]$, we may also define the Leja sequence for weighted interpolation on the real line. Given a point x_0, for $n \geq 1$ we recursively define:

$$y_n = \arg\max_{y \in \mathbb{R}} \left\{ w(y) \prod_{j=0}^{n-1} |y - y_j| \right\}. \tag{1.4}$$

As above, any maximizer is suitable, so we are not worried about the ambiguity in this definition. We make specific assumptions on the class of weight functions in §6.2, but mention that this class includes the commonly encountered Gaussian density, $w(y) = e^{-y^2}$.

The works [37, 68] show that a contracted version of the weighted Leja sequence (1.4) is asymptotically Fekete. Specifically, this means that we first multiply the weighted Leja sequence by a contraction factor, i.e.,

$$y_{n,j} := n^{-1/\alpha} y_j, \quad j = 0, \ldots, n, \tag{1.5}$$

for some appropriate real number $\alpha = \alpha(w) > 1$, depending on the weight w. Then the discrete point-mass measures μ_n giving weight $1/(n+1)$ to each of the first $n+1$ *contracted* Leja points, i.e.,

$$\mu_n := \frac{1}{n+1} \sum_{j=0}^{n} \delta_{\{y_{n,j}\}}, \tag{1.6}$$

converge weak*, as $n \to \infty$, to an equilibrium measure on a compact subset of $\mathbb{R}$. In other words, the Leja points asymptotically distribute similar to Fekete points, which are known to be a "good" set of points for interpolation (see §6.2.1 for a precise, potential theoretic explanation).

In fact, the asymptotically Fekete property is a necessary (but not sufficient) property for a set of points to have a subexponentially growing Lebesgue constant, and motivates our study of the weighted Leja sequence for Lagrange interpolation. Our contribution, given in Chapter 6, is to show that for a general class of weight functions w, the Lebesgue constant for Lagrange interpolation on the weighted Leja sequence (1.4) grows subexponentially in n. This result mirrors the best known results for the standard Leja points, and gives some theoretical justification for the use of weighted Leja points for interpolation in unbounded domains.

1.2.2 Sparse Quadrature Rules with Conformal Mappings

Standard interpolatory quadrature methods, such as Gauss–Legendre and Clenshaw–Curtis, tend to have points which cluster near the endpoints of the domain. As seen in the well-known interpolation example of Runge, this can mitigate the spurious effects of the growth of the polynomial basis functions at the boundary. However, this clustering can be problematic and inefficient in some situations. Gauss–Legendre and Clenshaw–Curtis grids, with n quadrature points on $[-1,1]$, are spaced asymptotically as $\frac{n}{\pi\sqrt{1-y^2}}$ [60]. Hence these clustered grids may have a factor of $\pi/2$ fewer points near the middle of the domain, compared with a uniform grid. This may have unintended negative effects in certain situations, and the issue is compounded when considering integrals over high-dimensional domains.

For numerical integration of an analytic function in one dimension, the convergence of quadrature approximations based on orthogonal polynomial interpolants depends crucially on the size of the region of analyticity, which we denote by Σ. More specifically, they depend on ρ, the parameter yielding the largest Bernstein ellipse, which is defined as

$$E_\rho := \{z \in \mathbb{C} : z + \frac{1}{z} \leq \rho\}, \tag{1.7}$$

contained in region of analyticity Σ [96]. This gives some intuition as to why the most stable quadrature rules place more nodes toward the boundary of the domain $[-1,1]$; since the boundary of E_ρ is close to $\{\pm 1\}$, the analyticity requirement is weaker near the endpoints of the domain. More specifically, to be analytic in E_ρ, the radius of the Taylor series of f at $\{\pm 1\}$ is only required to be $\rho - 1/\rho$, while the radius of the Taylor series centered at 0 is required to be at least $\rho + 1/\rho$.

On the other hand, the appearance of the Bernstein ellipse in the analysis is not tied fundamentally to the integrand, but only to the choice of polynomials as basis functions [49]. Thus, we may consider other types of quadrature rules which still take advantage of the analyticity of the integrand. Using non-polynomial functions as a basis for the rule may improve the convergence rate of the approximation. Much research has gone into investigating ways to find the optimal quadrature rule for a function analytic in Σ, and to overcome the aforementioned "$\pi/2$-effect", including end-point correction methods [1, 63, 57], non-polynomial based approximation [9, 7, 16, 84, 98], and the transformation methods [31, 46, 49, 58, 59, 76, 85] which map a given set of quadrature points to a less clustered set. In this chapter, we consider the transformation approach, based on the concept of conformal mappings in the complex plane. Many such transformations have been considered in the literature, but we consider here the transformations from [49], which offer the following benefits: (1) practical and implementable maps; and (2) simple concepts leading to theorems which may precisely quantify their benefits in mitigating the effect of the endpoint clustering.

Our contribution to this line of research, given in Chapter 7, is to implement and analyze the application of the transformed rules to sparse grid quadratures in the high-dimensional setting. For high-dimensional integration over the cube $[-1,1]^d$, the endpoint clustering means that a simple tensor product quadrature rule may use $(\pi/2)^d$ too many points. On the other hand, we show that for sparse Smolyak quadrature rules based on tensorization of transformed one-dimensional quadrature, this effect may be mitigated to some degree. We provide an analysis of the sparse grid mapped method to show that the improvement in the convergence rate to a d-dimensional integral is $(\pi/2)^{1/\xi(d)}$, where $\xi(d)^{-1} \geq d$.

Part I

Complexity of Stochastic Collocation Methods

Chapter 2

Problem Setting

In this chapter, we describe in more detail the uncertainty quantification problem (1.1) of PDEs with coefficients modeled as random coefficients. We make the necessary assumptions and definitions so that the linear/nonlinear problem is well defined and has an appropriate weak form, and discuss the spatial discretization to the underlying physical problem.

2.1 Random/Parameterized PDEs

Let $D \subset \mathbb{R}^d, d = 1, 2, 3$, be a bounded domain, and $(\Omega, \mathcal{F}, \mathbb{P})$ denote a complete probability space, where Ω is the sample space, $\mathcal{F} \subseteq 2^\Omega$ is a σ-algebra, and $\mathbb{P}$ is the associated probability measure. Define $\mathscr{D}(a)$ as a differential operator that depends on a random field $a(x, \omega)$ with $(x, \omega) \in D \times \Omega$. The forcing term $f = f(x, \omega)$ can be assumed to be a random field in an analogous way. Then we make the previous stochastic parameterized boundary value problem precise: find a stochastic function $u : \overline{D} \times \Omega \to \mathbb{R}$, such that it holds $\mathbb{P}$-a.e. in Ω

$$\mathscr{D}(a)[u] = f \quad \text{in } D, \tag{2.1}$$

subject to suitable (possibly parameterized) boundary conditions.

In many applications, the source of uncertainty can be approximated with only a finite number of uncorrelated, or even independent, random variables. For instance, a and f in (2.1) may have a piecewise representation, or have spatial variation that can be modeled as a correlated random field, making them amenable to approximation by a Karhunen-Loève (KL) expansion [62]. In practice, one has to truncate such expansions according to the desired accuracy of the simulation. As such, we make the following assumption regarding the random input data a and f (cf [48, 71]).

Assumption A1. (Finite dimensional noise) *The random fields a and f have the form:*

$$a(x,\omega) = a(x,\boldsymbol{y}(\omega)) \ \ \textit{and} \ \ f(x,\omega) = f(x,\boldsymbol{y}(\omega)) \ \textit{on} \ \ D \times \Omega,$$

where $\boldsymbol{y}(\omega) = [y_1(\omega),\dots,y_N(\omega)] : \Omega \to \mathbb{R}^N$ is a vector of uncorrelated random variables.

Assumption A1 is naturally satisfied by random fields that only depend on a finite set of parameters, e.g.,

$$a(\mathbf{x},\omega) = a(\mathbf{x},\boldsymbol{y}(\omega)) = a_0 + \sum_{n=1}^{N} y_n(\omega) a_n(\mathbf{x}), \quad \{a_n\}_{n=0}^N \subset L^2(D),$$

where $\boldsymbol{y}(\omega)$ is a vector of independent random variables. If this is not the case, approximations of a that satisfy Assumption A1 can be obtained by appropriately truncating a spectral expansion such as the Karhunen-Loève expansion [23, 41]. This introduces an additional error; see [71] for a discussion of the effect of this error on the convergence of stochastic collocation methods and [35, 14] for bounds on the truncation error. As an alternative to truncating infinite expansions, one can also consider using dimension-adaptive sparse grids as interpolation operators. For more details on this type of approximation, we refer the reader to [17, 39].

Another setting having a finite number of random variables occurs when the coefficient a and the forcing function f depends on a finite number of *independent* scalar random physical parameters, e.g., diffusivities, reaction rates, porosities, elastic moduli, etc. In this case, each of the N parameters would have its own PDF $\varrho_n(y_n)$, $n = 1,\dots,N$, so that the joint PDF is now given by $\varrho(\boldsymbol{y}) = \prod_{n=1}^N \varrho_n(y_n)$. The algorithms discussed in part I apply equally well to this setting. In the past several years, there has been much research on PDEs which depend on a countably-infinite dimensional parameter [17, 18, 23]. These works are able to show that for many random PDE problems, the solution map is sufficiently smooth so as to have a best-M term polynomial expansion which converges with dimension-independent convergence rates. This analysis provides a rigorous theoretical justification for the use of global polynomial reconstructions methods, and relies on complex analyticity assumptions on u similar to those we consider below. On the other hand, practical algorithms for constructing solutions to countably infinite problems are not well developed; see [94].

Now Assumption A1 and the Doob-Dynkin lemma [73] guarantee that $a(x, \boldsymbol{y}(\omega))$ and $f(x, \boldsymbol{y}(\omega))$ are Borel-measurable functions of the random vector $\boldsymbol{y} : \Omega \to \mathbb{R}^N$. In our setting, we let $\Gamma_n = y_n(\Omega) \subset \mathbb{R}$ be the image of the random variable y_n, and set $\Gamma = \prod_{n=1}^N \Gamma_n$, for $N \in \mathbb{N}_+$. If the distribution measure of $\boldsymbol{y}(\omega)$ is absolutely continuous with respect to the Lebesgue measure, there exists a joint probability density function of $\boldsymbol{y}(\omega)$ denoted by $\varrho(\boldsymbol{y}) : \Gamma \to \mathbb{R}_+$, with $\varrho(\boldsymbol{y}) = \prod_{n=1}^N \varrho_n(y_n)$. Therefore, based on Assumption A1, the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ is mapped to $(\Gamma, \mathcal{B}(\Gamma), \varrho(\boldsymbol{y})d\boldsymbol{y})$, where $\mathcal{B}(\Gamma)$ is the Borel σ-algebra on Γ and $\varrho(\boldsymbol{y})d\boldsymbol{y}$ is a probability measure on $\mathcal{B}(\Gamma)$. Assuming the solution u of (2.1) is σ-measurable with respect to a and f, the Doob-Dynkin lemma guarantees that $u(x, \omega)$ can also be characterized by the same random vector $\boldsymbol{y}$, i.e., $u(x, \omega) = u(x, y_1(\omega), \ldots, y_N(\omega))$.

Let $W(D)$ be a Banach space, and in addition to Assumption A1, assume the random input data are chosen so that the stochastic system (2.1) is well-posed and has a unique solution u in the weighted Bochner spaces $L^q_\varrho(\Gamma; W(D))$, which for $1 \leq q \leq \infty$ are defined by

$$L^q_\varrho(\Gamma; X(D)) = \Big\{ v : \Gamma \to W(D) \mid v \text{ is strongly meas. and } \|v\|_{L^q_\varrho(\Gamma; W(D))} < \infty \Big\}$$

with corresponding norm $\| \cdot \|_{L^q_\varrho(\Gamma; W(D))}$ given by

$$\|v\|^q_{L^q_\varrho(\Gamma; W(D))} = \int_\Gamma \|v(\cdot, \boldsymbol{y})\|^q_{W(D)} \varrho(\boldsymbol{y}) \mathrm{d}\boldsymbol{y}.$$

Note that the above integral will be replaced by the ϱ-essential supremum when $q = \infty$. In this setting, the solution space consists of Banach-space valued functions that have finite q-th order moments. Two example problems posed in this setting are given as follows.

Example 2.1. (Linear elliptic problem). Find $u : \overline{D} \times \Gamma \to \mathbb{R}$ such that ϱ-a.e.

$$\begin{cases} -\nabla \cdot (a(x,\boldsymbol{y})\nabla u(x,\boldsymbol{y})) &= f(x,\boldsymbol{y}) \quad \text{in } D \times \Gamma, \\ u(x,\boldsymbol{y}) &= 0 \qquad\quad\ \ \text{on } \partial D \times \Gamma, \end{cases} \tag{2.2}$$

where the well-posedness of (2.2) is guaranteed in $L^2_\varrho(\Gamma; H^1_0(D))$ with $a(x,\boldsymbol{y})$ uniformly elliptic, i.e., for ϱ-a.e. $\boldsymbol{y} \in \Gamma$,

$$a_{\min} \leq \|a(x,\boldsymbol{y})\|_{L^\infty(D)} \leq a_{\max} \ \text{ with } \ a_{\min}, a_{\max} \in (0,\infty), \tag{2.3}$$

and $f(x,\boldsymbol{y})$ square integrable, i.e., $\int_D \int_\Gamma f^2(x,\boldsymbol{y})\, d\varrho(\boldsymbol{y}) dx < +\infty$. We note that well-posedness can also be established in a stochastic sense; c.f. [14]. We also remark that the uniform ellipticity can be relaxed in certain situations, e.g., in groundwater flow problems where Γ is an unbounded domain [2, 15, 93].

Example 2.2. (Nonlinear elliptic problem). For $k \in \mathbb{N}$, find $u : \overline{D} \times \Gamma \to \mathbb{R}$ such that ϱ-a.e.

$$\begin{cases} -\nabla \cdot (a(x,\boldsymbol{y})\nabla u(x,\boldsymbol{y})) + u(x,\boldsymbol{y})|u(x,\boldsymbol{y})|^k &= f(x,\boldsymbol{y}) \quad \text{in } D, \\ u(x,\boldsymbol{y}) &= 0 \qquad\quad\ \ \text{on } \partial D. \end{cases} \tag{2.4}$$

The well-posedness of (2.4) is guaranteed in $L^2_\varrho(\Gamma; W(D))$ with a, f as in Example 2.1 and $W(D) = H^1_0(D) \cap L^{k+2}(D)$ [71].

2.1.1 Spatial Approximation

In what follows, we treat the solution to (2.1) as a parameterized function $u(x,\boldsymbol{y})$ of the N-dimensional random variables $\boldsymbol{y} \in \Gamma$. Moreover, since the solution u can be viewed as a mapping $u : \Gamma \to W(D)$, for convenience we may omit the dependence on $x \in D$ and write $u(\boldsymbol{y})$ to emphasize the dependence of u on $\boldsymbol{y}$. This leads to a general weak formulation [48] of the PDE in (2.1),

$$\int_D \left(\sum_{\nu \in \Lambda_1 \cup \Lambda_2} S_\nu(u(\boldsymbol{y}); \boldsymbol{y})\, T_\nu(v) \right) dx = \int_D f(\boldsymbol{y})\, v\, dx, \ \ \forall v \in W(D), \ \varrho\text{-a.e. in } \Gamma. \tag{2.5}$$

Here $T_\nu, \nu \in \Lambda_1 \cup \Lambda_2$ are linear operators independent of $\boldsymbol{y}$, while the operators S_ν are given to be linear for $\nu \in \Lambda_1$, and nonlinear for $\nu \in \Lambda_2$. Thus, the stochastic parameterized boundary-value problem (2.1) has been converted into a deterministic parametric problem (2.5).

Let $\{\varphi_i\}_{i=1}^{M_h}$ be a finite element (FE) basis of the space $W_h(D) \subset W(D)$. A general SC approach requires an approximate solution $u_h(\cdot, \boldsymbol{y}) \in W_h(D)$

$$u_h(x, \boldsymbol{y}_{L,j}) = \sum_{i=1}^{M_h} c_{L,j,i}\, \varphi_i(x), \quad j = 1, \ldots, M_L. \tag{2.6}$$

at a set of points $\{\boldsymbol{y}_{L,j}\}_{j=1}^{M_L} \subset \Gamma$. The vector $\boldsymbol{c}_{L,j} := (c_{L,j,1}, \ldots, c_{L,j,M_h})^\top$ solves

$$\begin{aligned}\sum_{i=1}^{M_h} c_{L,j,i} \int_D \sum_{\nu \in \Lambda_1} S_\nu\left(\varphi_i; \boldsymbol{y}_{L,j}\right) T_\nu(\varphi_{i'})\, dx \qquad\qquad\qquad\qquad & \qquad (2.7)\\ = \textstyle\int_D f(\boldsymbol{y}_{L,j})\varphi_{i'} - \sum_{\nu \in \Lambda_2} S_\nu \left(\sum_{i=1}^{M_h} c_{L,j,i}\, \varphi_i; \boldsymbol{y}_{L,j}\right) T_\nu(\varphi_{i'})\; dx, \;\; i' = 1, \ldots, M_h,&\end{aligned}$$

for $j = 1, \ldots, M_L$, with S_ν and T_ν defined as above. Note that for u_h, (2.7) is equivalent to (2.5) with the nonlinear operators subtracted on the right hand side. When $\Lambda_2 = \emptyset$, the PDE is linear, and a standard FE discretization leads to a linear system of equations. We consider only the linear form in Chapter 4, while in Chapter 5, we consider both linear and nonlinear equations. Because each chapter relies on specific assumptions about the spatial discretization used, we delay discussion of convergence rates to the individual chapters.

Chapter 3

Sparse Grid Interpolation

The algorithms described later in Part I will apply to a broad class of multidimensional approximation methods. Recall that we have defined a general polynomial interpolant in (1.2). In this chapter, however, we discuss a specific version of such an interpolant, namely sparse grid collocation based on globally defined Lagrange polynomials. This interpolant will satisfy the specific assumptions we make for general interpolation algorithms in the following chapters. Furthermore, we will analyze in detail the application of the multilevel and acceleration methods of Chapters 4 and 5, respectively, to global sparse grid interpolation.

3.1 Sparse Grid Construction

The construction of the interpolant in the N-dimensional space $\Gamma = \prod_{n=1}^{N} \Gamma_n$ is based on sequences of one-dimensional Lagrange interpolation operators $\{\mathscr{U}_n^{p(l)}\}_{l \in \mathbb{N}} : C^0(\Gamma_n) \to \mathbb{P}_{p(l)-1}(\Gamma_n)$, where $\mathbb{P}_p(\Gamma_n)$ denotes the space of polynomials of degree p on Γ_n. In particular, for each $n = 1, \ldots, N$, let $l \in \mathbb{N}_+$ denote the one-dimensional level of approximation and let $\{y_{n,j}^{(l)}\}_{j=1}^{p(l)} \subset \Gamma_n$ denote a sequence of one-dimensional interpolation points in Γ_n. Here, $p(l) : \mathbb{N}_+ \to \mathbb{N}_+$ is such that $p(1) = 1$ and $p(l) < p(l+1)$ for $l = 2, 3, \ldots$, so that $p(l)$ strictly increases with l and defines the total number of collocation points at level l. For a univariate function $v \in C^0(\Gamma_n)$, we define $\mathscr{U}_n^{p(l)}$ by

$$\mathscr{U}_n^{p(l)}[v](y_n) = \sum_{j=1}^{p(l)} v\big(y_{n,j}^{(l)}\big)\varphi_{n,j}^{(l)}(y_n) \quad \text{for } l_n = 1, 2, \ldots, \tag{3.1}$$

where $\varphi_{n,j}^{(l)} \in \mathbb{P}_{p(l)-1}(\Gamma_n)$, $j = 1, \ldots, p(l)$, are Lagrange fundamental polynomials of degree $p(l) - 1$, which are completely determined by the property $\varphi_{n,j}^{(l)}(y_{n,i}^{(l)}) = \delta_{i,j}$.

Using the convention that $\mathscr{U}_n^{p(0)} = 0$, we introduce the difference operator given by

$$\Delta_n^{p(l)} = \mathscr{U}_n^{p(l)} - \mathscr{U}_n^{p(l-1)}. \tag{3.2}$$

For the multivariate case, we let $\boldsymbol{l} = (l_1, \ldots, l_N) \in \mathbb{N}^N$ denote a multi-index and $L \in \mathbb{N}_+$ denote the total level of the sparse grid approximation. We also let $g(\mathbf{l}) : \mathbb{N}_+^N \to \mathbb{N}_+$ be a strictly increasing function, defining a mapping between the multi-index $\boldsymbol{l}$ and the sparse grid level L. Now, from (3.2), the *L-th level generalized sparse-grid approximation* of $v \in C^0(\Gamma)$ is given by

$$\begin{aligned} \mathcal{A}_L^{p,g}[v](\boldsymbol{y}) &= \sum_{g(\mathbf{l}) \le L} \left(\Delta^{p(l_1)} \otimes \cdots \otimes \Delta^{p(l_N)}\right)[v](\boldsymbol{y}) \\ &= \sum_{g(\mathbf{l}) \le L} \sum_{\mathbf{i} \in \{0,1\}^N} (-1)^{|\mathbf{i}|} \left(\mathscr{U}^{p(l_1 - i_1)} \otimes \cdots \otimes \mathscr{U}^{p(l_N - i_N)}\right)[v](\boldsymbol{y}), \end{aligned} \tag{3.3}$$

where $\mathbf{i} = (i_1, \ldots, i_N)$ is a multi-index with $i_n \in \{0, 1\}$, $|\mathbf{i}| = i_1 + \cdots + i_N$.

This approximation lives in the tensor product polynomial space $\mathcal{P}_{\Lambda_L^{p,g}}$ given by

$$\mathbb{P}_{\Lambda_L^{p,g}} = \text{span}\left\{ \prod_{n=1}^N y_n^{l_n} \;\middle|\; \mathbf{l} \in \Lambda_L^{p,g} \right\},$$

where the multi-index set is defined as follows

$$\Lambda_L^{p,g} = \left\{ \mathbf{l} \in \mathbb{N}^N \;\middle|\; g(\mathbf{p}^\dagger(\mathbf{l} + \mathbf{1})) \le L \right\}.$$

Here $\mathbf{p}^\dagger(\mathbf{l}) = (p^\dagger(l_1), \ldots, p^\dagger(l_N))$, and $p^\dagger(l) := \min\{w \in \mathbb{N}_+ : p(w) \ge l\}$ is the left inverse of p (see [3, 48]). The approximation (3.3) requires the independent evaluation of v on a deterministic set of *distinct collocation points* given by

$$\mathcal{H}_L^{p,g} = \bigcup_{L-N+1 \le g(\boldsymbol{l}) \le L} \prod_{1 \le n \le N} \left\{ y_{n,j}^{l_n} \right\}_{j=1}^{p(l_n)}$$

having cardinality M_L. Some examples of functions $p(l)$ and $g(\boldsymbol{l})$ and the corresponding polynomial approximation spaces are given in Table 3.1. In the last example in the table $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_N) \in \mathbb{R}_+^N$ is a vector of weights reflecting the anisotropy of the system, i.e., the relative importance of each dimension [70]; we then define $\alpha_{min} := \min\limits_{n=1,\dots,N} \alpha_n$. The corresponding anisotropic versions of the other approximations and corresponding polynomial subspaces can be analogously constructed.

Table 3.1: The functions $p : \mathbb{N}_+ \to \mathbb{N}_+$ and $g : \mathbb{N}_+^N \to \mathbb{N}$ and the corresponding multiindex subspaces.

Multiindex Space	$p(l)$	$g(\boldsymbol{l})$
Tensor product	$p(l) = l$	$\max\limits_{1 \le n \le N} (l_n - 1)$
Total degree	$p(l) = l$	$\sum_{n=1}^N (l_n - 1)$
Hyperbolic cross	$p(l) = l$	$\prod_{n=1}^N (l_n - 1)$
Sparse Smolyak	$p(l) = 2^{l-1} + 1,\ l > 1$	$\sum_{n=1}^N (l_n - 1)$
Anisotropic Sparse Smolyak	$p(l) = 2^{l-1} + 1,\ l > 1$	$\sum_{n=1}^N \frac{\alpha_n}{\alpha_{min}} (l_n - 1),\ \boldsymbol{\alpha} \in \mathbb{R}_+^N$

For Smolyak multiindex spaces, the most popular choice of points are the sparse grids based on the one-dimensional Clenshaw-Curtis abscissas [21] which are the extrema of Chebyshev polynomials, including the end-point extrema. For level l, and in the particular case $\Gamma_n = [-1, 1]$ and $p(l) > 1$, the resulting points are given by

$$y_{n,j}^l = -\cos\left(\frac{\pi(j-1)}{p(l)-1}\right) \quad \text{for } j = 1, \dots, p(l). \tag{3.4}$$

In particular, in the Sparse Smolyak construction from Table 3.1, the choice

$$p(1) = 1,\ p(l) = 2^{l-1} + 1 \text{ for } \ l > 1, \ \text{ and } \ g(\mathbf{l}) = \sum_{n=1}^{N} (l_n - 1). \tag{3.5}$$

results in a *nested* family of one-dimensional abscissas, i.e., $\left\{y_{n,j}^{l}\right\}_{j=1}^{p(l)} \subset \left\{y_{n,j}^{l+1}\right\}_{j=1}^{p(l+1)}$. Here, the sparse grids are also nested, i.e.,

$$\mathcal{H}_L^{p,g} \subset \mathcal{H}_{L+1}^{p,g}.$$

This corresponds to the most widely used sparse-grid approximation, as first described in [86]. This is the typical choice we will make in the following chapters, however much of the analysis does not depend strongly on this choice of m and g, and we could use other functions, e.g., anisotropic approximations. We remark also that other nested families of sparse grids can be constructed from, e.g., the Leja points [25], Gauss-Patterson [95], Newton-Cotes, etc.

Remark 3.1. *In general, the growth rate $p(l)$ can be chosen as any increasing function on $\mathbb{N}$. However, for non-nested point families, such as standard Gaussian abscissas, the approximation* (3.3) *is no longer guaranteed to be an interpolant, but the analysis of the approximation error remains similar to the analysis presented here (see [71] for more details).*

3.2 Lagrange Interpolating Formulation

When the multidimensional point sets are nested, the approximation $\mathcal{A}_L^{p,g}[v]$ is a Lagrange interpolating polynomial [71], and thus (3.3) can be rewritten as a linear combination of Lagrange basis functions,

$$\begin{aligned} \mathcal{A}_L^{p,g}[v](\boldsymbol{y}) &= \sum_{j=1}^{M_L} v(\boldsymbol{y}_{L,j}) \Psi_{L,j}(\boldsymbol{y}) \\ &= \sum_{j=1}^{M_L} v(\boldsymbol{y}_{L,j}) \underbrace{\sum_{\mathbf{l} \in \mathcal{J}(L,j)} \sum_{\mathbf{i} \in \{0,1\}^N} (-1)^{|\mathbf{i}|} \prod_{n=1}^{N} \psi_{k_n(j)}^{l_n - i_n}(y_n)}_{\Psi_{L,j}(\boldsymbol{y})}, \end{aligned} \tag{3.6}$$

where the index set $\mathscr{J}(L,j)$ is defined by

$$\mathscr{J}(L,j) = \left\{ \mathbf{l} \in \mathbb{N}_+^N \;\middle|\; g(\mathbf{l}) \le L \text{ and } \boldsymbol{y}_{L,j} \in \bigotimes_{n=1}^{N} \vartheta^{p(l_n - i_n)} \text{ with } \mathbf{i} \in \{0,1\}^N \right\},$$

and $\vartheta^{p(l_n)} = \{y_{n,1}^{l_n}, \dots, y_{n,p(l_n)}^{l_n}\} \subset \Gamma_n$.

For a given L and j, this represents the subset of multi-indices corresponding to the tensor-product operators $\mathscr{U}^{p(l_1-i_1)} \otimes \cdots \otimes \mathscr{U}^{p(l_N-i_N)}$ in (3.3) with the supporting point $\boldsymbol{y}_{L,j}$. Then for each $\mathbf{l} \in \mathscr{J}(L,j)$ and $\mathbf{i} \in \{0,1\}^N$, the function $\prod_{n=1}^N \psi_{k_n(j)}^{l_n - i_n}(y_n)$ with $k_n(j) \in \{1, \dots, p(l_n - i_n)\}, n = 1, \dots, N$, represents the unique Lagrange basis function for the operator $\mathscr{U}^{p(l_1-i_1)} \otimes \cdots \otimes \mathscr{U}^{p(l_N-i_N)}$ corresponding to $\boldsymbol{y}_{L,j}$. Therefore, the functions $\{\Psi_{L,j}\}_{j=1}^{M_L}$ are given by a linear combination of tensorized Lagrange polynomials satisfying the "delta property", i.e., $\Psi_{L,j'}(\boldsymbol{y}_{L,j}) = \delta_{jj'}$ for $j, j' = 1, \dots, M_L$. We require an interpolant of this form for our analysis in Chapter 5; see (5.1).

3.3 Convergence of Sparse Grid Collocation

In this section, we examine the convergence of the sparse grid interpolation methods described above. We will give two lemmas, the first regarding convergence in terms of the number of points, M_L, and the second in terms of the sparse grid level L.

We first need some understanding of the regularity of the solution $u : \Gamma \to H_0^1(D)$ to the parameterized elliptic PDE described in Example 2.1. As such, we require the additional assumption on the regularity of the coefficient a:

Assumption A2. *Assume that $a : \Gamma \to L^\infty(D)$ has a holomorphic complex continuation $a^* : \mathbb{C}^N \to L^\infty(D)$.*

Next, we use assumption A2 to show that the approximate PDE solutions u_{h_k} are analytic in a region $\Sigma(\boldsymbol{\rho}) \subset \mathbb{C}^N$. For $\boldsymbol{\rho} = (\rho_1, \dots, \rho_N) \in (1, \infty)^N$, this region will have the form

$$\Sigma(\boldsymbol{\rho}) = \prod_{1 \le n \le N} \Sigma(n; \rho_n) \subset \mathbb{C}^N, \tag{3.7}$$

where $\Sigma(n;\rho_n)$ denotes the region bounded by the Bernstein ellipse,

$$\Sigma(n;\rho_n) = \left\{ \frac{1}{2}\left(z_n + z_n^{-1}\right) : z_n \in \mathbb{C}, |z_n| = \rho_n \right\}.$$

The set $\Sigma(\boldsymbol{\rho}) \subset \mathbb{C}^N$ is the product of ellipses in the complex plane, with foci $z_n = \pm 1$, which are the endpoints of the domain $\Gamma_n, n = 1, \ldots, N$. Such ellipses are common in proving convergence results for global interpolation schemes. Chapter 7 contains a more thorough examination of these ellipses in global polynomial approximations.

The following result on the analyticity of the solution u is proved in [23, Theorem 1.2] and [94, Lemma 3.3 and Theorem 2.5].

Lemma 3.1.1. (Analyticity of the PDE solution u) *Under the assumption A2, there exists $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_N) \in (1,\infty)^N$ such that the complex extension of u to the polyellipse $\Sigma(\boldsymbol{\rho})$, $u^* : \Sigma(\boldsymbol{\rho}) \to H_0^1(D)$ is well-defined and analytic in an open region containing $\Sigma(\boldsymbol{\rho})$.*

In §4.4, we will also show that Assumption A2 leads to analyticity of certain functionals of the solution. Note that with less regularity in the solution, we might use local basis functions such as wavelets or splines to construct the interpolant [47, 64].

For a function v which admits an analytic extension in a polyellipse, convergence with respect to the total number of collocation points for the tensor product, sparse isotropic, and anisotropic Smolyak approximations (see Table 3.1), using both Clenshaw–Curtis and Gaussian nodes, was analyzed in [2, 71, 70]. We restate the result here.

Theorem 3.2. *Let W denote a general Banach space and let $v \in C^0(\Gamma; W)$ admit an analytic extension in the complex polyellipse $\Sigma(\boldsymbol{\rho})$. Then, with $r = \min_{1\le n\le N} \rho_n$, there exist constants $C(N)$ and $\mu(r,N)$, depending on N, such that*

$$\|v - \mathcal{A}_L^{p,g} v\|_{L^2_\varrho(\Gamma;W)} \le C(N)\, M_L^{-\mu(r,N)}\, \zeta(v),$$

where M_L is the number of points used by $\mathcal{A}_L^{p,g}$ and

$$\zeta(v) \equiv \max_{\boldsymbol{z}\in\Sigma(\boldsymbol{\rho})} \|v(\boldsymbol{z})\|_W. \tag{3.8}$$

Note that this estimate is not the best possible asymptotic estimate. Yet, it is satisfactory for the MLSC examined in Chapter 4, which combines several different levels L_k into the approximation. Some of the levels in the multilevel construction may be too small for the asymptotic theory to apply, and so we take a safer estimate.

Remark 3.3. Dimension-dependent convergence rate. *The asymptotic rate of convergence $\mu = \mu(r, N)$ in general deteriorates with growing dimension N of the stochastic space. For example, we have $\mu = r/N$ in the tensor product case, and for Smolyak sparse grids this is improved to $\mu = r/\log(N)$. The use of sparse grid SC methods is hence only of interest for dimensions N for which $\mu \geq 1/2$ so that the error still converges faster than the corresponding Monte Carlo sampling error. The multilevel approximation presented in Chapter 4 suffers from the same deterioration of convergence rate, and roughly speaking, the MLSC method can improve on the multilevel Monte Carlo method only when standard SC performs better that standard Monte Carlo; see [22, Theorem 4.1].*

Remark 3.4. Anisotropic sparse grid approximations. *To define anisotropic Smolyak approximations, we introduce a weight vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N)$ into the definition of g to reflect the relative importance of each dimension when selecting points, e.g., the anisotropic sparse Smolyak space uses $p(l) = 2^{l-1} + 1$, $l > 1$ and $g(\boldsymbol{l}) = \sum_{n=1}^{N} \frac{\alpha_n}{\alpha_{min}}(l_n - 1)$. The weight α_n is related to the size of the largest Bernstein ellipse Σ on which the map $u : \Gamma_n \to C^0(\prod_{j \neq n} \Gamma_n, W)$ can be analytically extended. These weights can be computed either* a priori *or* a posteriori*; see [70, section 2.2]. For an* isotropic grid*, all the components of the weight vector $\boldsymbol{\alpha}$ are the same so that one has to take the worst case scenario, i.e., choose the components of $\boldsymbol{\alpha}$ to all equal to the minimum α_{min}.*

In § 5.3.1, we will also require estimates on the convergence in terms of the sparse grid level $L \in \mathbb{N}^{+}$. Again, this can be given by a restatement of a result from [71, 70]. According to Lemma 3.1.1, Assumption A2 implies that u is analytic in a polyellipse $\Sigma(\boldsymbol{\rho})$ given by (3.7). Then the usual sparse grid convergence theory from [71, 70] gives:

Lemma 3.4.1. *Let u satisfy Assumption A2. For $L \in \mathbb{N}^+$, the interpolation error $u - \mathcal{A}_L^{p,g}[u]$ of the sparse grid SC method using Clenshaw-Curtis abscissas can be bounded as*

$$\|u - \mathcal{A}_L^{p,g}[u]\|_{L^\infty(\Gamma; H_0^1(D))} \leq C_{\mathrm{sc}} \mathrm{e}^{-rN2^{L/N}},$$

where, for a constant $0 < \delta < 1$, the rate $r = (1-\delta)\min_{1 \leq n \leq N} \log \rho_n$, and the constant $C_{\mathrm{sc}} > 0$ depends on N, u, and δ.

Chapter 4

Multilevel Stochastic Collocation Methods

Some content of the following chapter first appeared (see [92]) in the SIAM/ASA Journal of Uncertainty Quantification in 2015, published by the Society for Industrial and Applied Mathematics (SIAM) and the American Statistical Association (ASA). Copyright by SIAM and ASA. Unauthorized reproduction is prohibited. The author completed this work in collaboration with Max Gunzburger, Aretha Teckentrup, and Clayton G. Webster. Some notation has been slightly edited to maintain consistency with other chapters in this manuscript, and much of the introductory material has been altered

In this chapter, we analyze a multilevel version of the stochastic collocation method that, as is the case for multilevel Monte Carlo (MLMC) methods, uses hierarchies of spatial approximations to reduce the overall computational complexity. In addition, our proposed approach utilizes, for approximation in stochastic space, a sequence of multi-dimensional interpolants of increasing fidelity which can then be used for approximating statistics of the solution as well as for building high-order surrogates featuring faster convergence rates. A rigorous convergence and computational cost analysis of the new multilevel stochastic collocation method is provided in the case of elliptic equations, demonstrating its advantages compared to standard single-level stochastic collocation approximations as well as MLMC methods.

The outline of the chapter is as follows. In §4.1, we introduce some further assumptions on the parametrization of the random inputs that are used to transform the original stochastic problem into a deterministic parametric version, and necessary assumptions about the regularity of the solution of the PDE, which are in addition to the assumptions made in Chapter 2. A description of the spatial and stochastic approximations as well as the formulation of the MLSC method follows in §4.2. In §4.3, we provide a general convergence and complexity analysis for the MLSC method. As an example of a specific single level SC approach satisfying our interpolation assumptions, in §4.4 we analyze the ML method using generalized sparse grid stochastic collocation approach based on global Lagrange interpolation introduced in §3.1. In §4.5, we provide numerical results that illustrate the theoretical results and complexity estimates and also explore issues related to the implementation of the MLSC method.

4.1 Further Assumptions

In this chapter, we will work only in basic setting of a linear random PDE (2.2), which was introduced in Example 2.1. In addition to Assumption A1, we make the following assumptions on a. We note that some of what is stated in the following has already been assumed in §2.1, but we restate it here to make the setting more precise.

Assumption A3. (Boundedness) *The image* $\Gamma_n := y_n(\Omega)$ *of* y_n *is bounded for all* $n \in \{1, \dots, N\}$ *and, with* $\Gamma = \prod_{n=1}^{N} \Gamma_n$*, the random variables* $\boldsymbol{y}$ *have a joint probability density function* $\varrho(\boldsymbol{y}) = \prod_{n=1}^{N} \widetilde{\varrho}(y_n) \in L^\infty(\Gamma)$*, where* $\widetilde{\varrho}(\cdot) : [-1,1] \to \mathbb{R}$ *denotes the one-dimensional PDF corresponding to the probability space of the random fields. Without loss of generality, we assume that* $\Gamma = [-1,1]^N$*.*

Assumption A4. (Existence and uniqueness) *The coefficient* $a(\mathbf{x}, \boldsymbol{y})$ *is uniformly bounded and coercive, i.e., there exists* $a_{min} > 0$ *and* $a_{max} < \infty$ *such that for* ϱ*-almost every* $\boldsymbol{y}$*,*

$$a_{min} \leq a(\mathbf{x}, \boldsymbol{y}) \leq a_{max} \ \ \forall \, \mathbf{x} \in \overline{D}$$

and $f \in H^{-1}(D)$ is independent of $\boldsymbol{y}$, so that the problem (2.2) admits a unique solution $u \in L^2_\varrho(\Gamma; H^1_0(D))$ with realizations in $H^1_0(D)$, i.e., $u(\cdot, \boldsymbol{y}) \in H^1_0(D)$ ϱ-almost everywhere.

Assumption A3 can be weakened to include the case of unbounded random variables such as Gaussian variables. See [2] for an analysis of the interpolation error and note that, with only minor modifications, the multilevel stochastic collocation method introduced in this chapter also applies to unbounded random variables. We also consider the problem of interpolation on unbounded domains in Chapter 6. Furthermore, Assumption A4 can be weakened to include coefficients a that are not uniformly coercive; see [15, 93]. Finally, we remark that the multilevel stochastic collocation method proposed in this chapter is not specific to the model problem given in Example 2.1; it can be applied also to higher-order PDEs and other types of boundary conditions.

4.2 Hierarchical multilevel stochastic collocation methods

We begin by recalling that standard stochastic collocation (SC) methods generally build an approximation of the solution u by evaluating a spatial approximation $u_h(\cdot, \boldsymbol{y}) \in V_h$ at a given set of points $\{\boldsymbol{y}_m\}_{m=1}^M$ in Γ, where $V_h \subset H^1_0(D)$ is a finite-dimensional subspace. In other words, we compute $\{u_h(\cdot, \boldsymbol{y}_m)\}_{m=1}^M$. Then, given a basis $\{\Psi_m(\boldsymbol{y})\}_{m=1}^M$ for the space $\mathcal{P}_M = \operatorname{span}\{\Psi_m(\boldsymbol{y})\}_{m=1}^M \subset L^2_\varrho(\Gamma)$, we use those samples to construct the fully discrete approximation given by the interpolant

$$u^{(\mathrm{SL})}_{M,h}(\mathbf{x}, \boldsymbol{y}) = \mathcal{I}_M[u_h](\mathbf{x}, \boldsymbol{y}) = \sum_{m=1}^{M} c_m(\mathbf{x})\, \Psi_m(\boldsymbol{y}), \tag{4.1}$$

where the coefficients $c_m(\mathbf{x})$ are fully determined by the semi-discrete solutions at the collocation points, $u_h(\mathbf{x}, \boldsymbol{y}_m)$ for $m = 1, \ldots, M$. In (4.1), we label the standard SC approximation by ‘SL’ to indicate that that approximation is constructed using a single set of points $\{\boldsymbol{y}_m\}_{m=1}^M$ in stochastic space, in contrast to the multilevel approximations considered below that use a hierarchy of point sets; thus, in this chapter we refer to (4.1) as a *single* *level* approximation. A wide range of choices for the interpolation points $\{\boldsymbol{y}_m\}_{m=1}^M$ and basis functions $\{\Psi_m(\boldsymbol{y})\}_{m=1}^M$ are possible. A particular example of the approximation (4.1), namely global Lagrange interpolation on generalized sparse grids, was given in Chapter 3, and will be analyzed in §4.4

Convergence of the SC approximation (4.1) is often assessed in the natural $L^2_\varrho(\Gamma; H^1_0(D))$-norm, and the goal is to determine a bound on the error $\|u - \mathcal{I}_M[u_h]\|_{L^2_\varrho(\Gamma;H^1_0(D))}$. To obtain a good approximation with SC methods, it is necessary in general to use accurate spatial approximations u_h and a large number M of collocation points. To determine the coefficients $c_m(\mathbf{x})$ of the interpolant (4.1), the method requires the computation of $u_h(\cdot, \boldsymbol{y}_m)$ for $m = 1, \ldots, M$ so that, in practice, the cost can grow quickly with increasing N. Therefore, to reduce the overall cost, we consider a multilevel version of SC methods that combines different levels of fidelity of both the spatial and parameter approximations.

4.2.1 Hierarchical spatial approximations

For spatial approximation, we use a hierarchical family of finite element discretizations [10, 20]. As discussed in [50], the formulation of the multilevel method does not depend on the specific spatial discretization scheme used and the results readily hold for other choices. For $k \in \mathbb{N}_0$, define a hierarchy of nested finite element spaces

$$V_{h_0} \subset V_{h_1} \subset \cdots \subset V_{h_k} \subset \cdots \subset H^1_0(D),$$

where each V_{h_k} consists of continuous, piecewise polynomial functions on a shape regular triangulation ρ_{h_k} of D having maximum mesh spacing parameter h_k. Note that k merely serves to index the given spaces; the approximation properties of the space V_{h_k} is governed by h_k. For simplicity, we assume that the triangulations $\{\rho_{h_k}\}_{k\in\mathbb{N}_0}$ are generated by iterative uniform subdivisions of the initial triangulation ρ_0; this implies that $h_k = \eta^{-k} h_0$ for some $\eta \in \mathbb{N}$, $\eta > 1$ and that indeed the corresponding finite element spaces are nested.

Remark 4.1. *For simplicity, we have assumed that the finite element family of spaces is nested, and in fact, are constructed by a series of uniform subdivisions of a parent mesh with mesh size h_0. Neither of these assumptions are necessary for our algorithms or conclusions* *to hold, provided $\eta_1 \leq h_k/h_{k+1} \leq \eta_2$ for some $0 < \eta_1 < \eta_2 < \infty$ and all $k \in \mathbb{N}_0$; in such cases, the finite element spaces are not necessarily nested.*

We also let $u_{h_k}(\cdot, \boldsymbol{y})$ denote the Galerkin projection of $u(\cdot, \boldsymbol{y})$ onto V_{h_k}, i.e., $u_{h_k} \in V_{h_k}$ denotes the finite element approximation. Note that $u_{h_k}(\cdot, \boldsymbol{y})$ is still a function on the stochastic parameter space Γ. We assume the following approximation property of the finite element spaces $\{V_{h_k}\}_{k\in\mathbb{N}_0}$:

Assumption A5. *There exist positive constants α and C_s, independent of h_k, such that for all $k \in \mathbb{N}_0$,*

$$\|u - u_{h_k}\|_{L^2_\varrho(\Gamma;H^1_0(D))} \leq C_s\, h_k^\alpha.$$

In general, the rate α depends on the (spatial) regularity of u, which in turn depends on the regularity of a and f as well as on the geometry of the domain D. For example, if a, f, and D are sufficiently regular so that $u \in L^2_\varrho(\Gamma; H^2(D))$, Assumption A5 holds with $\alpha = 1$ and C_s dependent only on a and $\|u\|_{L^2_\varrho(\Gamma;H^2(D))}$. For additional examples and detailed analyses of finite element errors, see [93].

4.2.2 Stochastic interpolation

For stochastic approximation, we use interpolation over Γ, where we assume $u \in C^0(\Gamma; H^1_0(D))$. The specific choice of interpolation scheme is not crucial at this juncture. We begin by letting $\{\mathcal{I}_{M_k}\}_{k=0}^\infty$ denote a sequence of interpolation operators $\mathcal{I}_{M_k} : C^0(\Gamma) \to L^2_\varrho(\Gamma)$ using M_k points. We assume the following:

Assumption A6. *There exist positive constants C_I, C_ζ, and β, and a Banach space $\Lambda(\Gamma; H^1_0(D)) \subset L^2_\varrho(\Gamma; H^1_0(D))$ containing the finite element approximations $\{u_{h_k}\}_{k\in\mathbb{N}_0}$ such that for all $v \in \Lambda(\Gamma; H^1_0(D))$ and all $k \in \mathbb{N}_0$*

$$\|v - \mathcal{I}_{M_k} v\|_{L^2_\varrho(\Gamma;H^1_0(D))} \leq C_I\, \sigma(M_k)\, \zeta(v),$$

for some decreasing sequence $\{\sigma_k\}_{k\in\mathbb{N}_0}$*, with* $\sigma_k = \sigma(M_k)$*, and operator* $\zeta : \Lambda(\Gamma; H_0^1(D)) \to \mathbb{R}$ *that admits the estimates*

$$\zeta(u_{h_k}) \leq C_\zeta\, h_0^\beta \quad \text{and} \quad \zeta(u_{h_{k+1}} - u_{h_k}) \leq C_\zeta\, h_{k+1}^\beta.$$

Remark 4.2. *As in the previous section, k is merely an index; we use the same index for the hierarchies of spatial and stochastic approximations because, in the multilevel SC method we introduce below, these two hierarchies are closely connected.*

Remark 4.3. *σ_k determines the approximation properties of the interpolant. Moreover, we allow non-unique interpolation operators in the sequence, i.e., it is possible that, for any $k = 0, \ldots, \infty$, $M_{k+1} = M_k$ and therefore $\mathcal{I}_{M_{k+1}} = \mathcal{I}_{M_k}$ and $\sigma_{k+1} = \sigma_k$. Thus, although the spatial approximation improves with increasing k, i.e., $h_{k+1} < h_k$, we allow for the parameter space approximation for the index $k+1$ remaining the same as that for k.*

In §4.4, Assumption A6 is shown to hold, with $\sigma_k = M_k^{-\mu}$, for global Lagrange interpolation using generalized sparse grids. The bounds on the function ζ in Assumption A6 are shown to be the key to balancing spatial and stochastic discretizations through the multilevel formulation. Crucially, we make use of the fact that the interpolation error is proportional to the size of the function being interpolated, measured in an appropriate norm. In the case of the model problem (2.2), this norm is usually related to the (spatial) $H_0^1(D)$-norm. The bounds in Assumption A6 then arise from the fact that for any $k \in N_0$, $\|u_{h_k}\|_{H_0^1(D)}$ is bounded by a constant, independent of k, whereas $\|u_{h_k} - u_{h_{k-1}}\|_{H_0^1(D)}$ decays with h_k^β for some $\beta > 0$. We usually have $\beta = \alpha$, where α is as in Assumption A5. Note that we have chosen to scale the bound on $\zeta(u_{h_k})$ by h_0^β to simplify calculations. Because h_0 is a constant, this does not affect the nature of the assumption.

4.2.3 Formulation of the multilevel method

As in the previous sections, denote by $\{u_{h_k}\}_{k\in\mathbb{N}_0}$ and $\{\mathcal{I}_{M_k}\}_{k\in\mathbb{N}_0}$ sequences of spatial approximations and interpolation operators in parameter space, respectively. Then, for any

$K \in \mathbb{N}$, the formulation of the multilevel method begins with the simple telescoping identity

$$u_{h_K} = \sum_{k=0}^{K} (u_{h_k} - u_{h_{k-1}}), \tag{4.2}$$

where, for simplicity, we set $u_{h_{-1}} := 0$.

It follows from Assumption A6 that as $k \to \infty$, less accurate interpolation operators are needed in order to estimate $u_{h_k} - u_{h_{k-1}}$ to achieve a required accuracy. We therefore define our multilevel interpolation approximation as

$$u_K^{(\mathrm{ML})} := \sum_{k=0}^{K} \mathcal{I}_{M_{K-k}}[u_{h_k} - u_{h_{k-1}}] = \sum_{k=0}^{K} \left(u^{(\mathrm{SL})}_{M_{K-k},h_k} - u^{(\mathrm{SL})}_{M_{K-k},h_{k-1}} \right). \tag{4.3}$$

Rather than simply interpolating u_{h_K}, this approximation uses different levels of interpolation on each difference $u_{h_k} - u_{h_{k-1}}$ of finite element approximations. To preserve convergence, the estimator uses the most accurate interpolation operator $\mathcal{I}_{M_K}$ on the coarsest spatial approximation u_{h_0} and the least accurate interpolation operator $\mathcal{I}_{M_0}$ on the finest spatial approximation $u_{h_K} - u_{h_{K-1}}$. Note that in (4.3) a single index k is used to select appropriate spatial and stochastic approximations and thus these approximations are indeed closely related.

4.3 Analysis of the multilevel approximation

This section is devoted to proving the convergence of the multilevel approximation defined in §4.2.3 and analyzing its computational complexity. We first prove, in §4.3.1, a general error bound, whereas in Sections 4.3.2 and 4.3.3 we prove a bound on the computational complexity in the particular case of an algebraic decay of the interpolation errors.

4.3.1 Convergence analysis

We consider the convergence of the multilevel approximation $u_K^{(\mathrm{ML})}$ to the true solution u in the natural norm $\| \cdot \|_{L^2_\varrho(\Gamma;H^1_0(D))}$.

First, we use the triangle inequality to split the error into the sum of a spatial discretization error and a stochastic interpolation error, i.e.,

$$\|u - u_K^{(\mathrm{ML})}\|_{L^2_\varrho(\Gamma;H^1_0(D))} \leq \underbrace{\|u - u_{h_K}\|}_{(I)}{}_{L^2_\varrho(\Gamma;H^1_0(D))} + \underbrace{\|u_{h_K} - u_K^{(\mathrm{ML})}\|}_{(II)}{}_{L^2_\varrho(\Gamma;H^1_0(D))}. \tag{4.4}$$

The aim is to prove that with the interpolation operators $\{\mathcal{I}_{M_k}\}_{k=0}^K$ chosen appropriately, the stochastic interpolation error (II) of the multilevel approximation converges at the same rate as the spatial discretization error (I), hence resulting in a convergence result for the total error.

For the spatial discretization error (I), it follows immediately from Assumption A5 that

$$(I) \leq C_s h_K^\alpha.$$

From (4.2) and Assumption A6, we estimate the stochastic interpolation error using the triangle inequality:

$$\begin{aligned}(II) &= \Big\| \sum_{k=0}^K (u_{h_k} - u_{h_{k-1}}) - \mathcal{I}_{M_{K-k}}(u_{h_k} - u_{h_{k-1}}) \Big\|_{L^2_\varrho(\Gamma;H^1_0(D))} \\ &\leq \sum_{k=0}^K \big\| (u_{h_k} - u_{h_{k-1}}) - \mathcal{I}_{M_{K-k}}(u_{h_k} - u_{h_{k-1}}) \big\|_{L^2_\varrho(\Gamma;H^1_0(D))} \\ &\leq \sum_{k=0}^K C_I\, C_\zeta\, \sigma_{K-k}\, h_k^\beta.\end{aligned}$$

To obtain an error of the same size as (I), we choose interpolation operators such that

$$\sigma_{K-k} \leq C_s \left((K+1)\, C_I\, C_\zeta \right)^{-1} h_K^\alpha\, h_k^{-\beta}. \tag{4.5}$$

Continuing from above,

$$(II) \leq \sum_{k=0}^K C_s \left((K+1)\, C_I\, C_\zeta \right)^{-1} h_K^\alpha\, h_k^{-\beta} C_I\, C_\zeta\, h_k^\beta = C_s h_K^\alpha,$$

as required. It follows that with σ_k as in (4.5)

$$\|u - u_K^{(\mathrm{ML})}\|_{L^2_\varrho(\Gamma;H^1_0(D))} \leq 2\, C_s\, h_K^\alpha.$$

4.3.2 Cost analysis

We now proceed to analyze the computational cost of the MLSC method. We consider the *ε-cost of the estimator*, denoted here by $C_\varepsilon^{\mathrm{ML}}$, which is the computational cost required to achieve a desired accuracy ε. In order to quantify this cost, we use the convergence rates of the spatial discretization error and, for the stochastic interpolation error, the rates given by assumptions A5 and A6. In particular, we will assume that A6 holds with $\sigma_k = M_k^{-\mu}$ for some $\mu > 0$.

Remark 4.4. *The choice $\sigma_k = M_k^{-\mu}$ best reflects approximations based on SC methods that employ sparse grids. In particular, as mentioned in §4.2.2, algebraic decay holds for the generalized sparse grid interpolation operators considered in Chapter 3; see Theorem 3.2. For other possible choices in the context of quadrature, see [50].*

In general, the MLSC method involves solving, for each k, the deterministic PDE for each of the M_k sample points from Γ; in fact, according to (4.3), two solves are needed, one for each of two spatial grid levels. Thus, we also require a bound on the cost, which we denote by C_k, of computing $u_{h_k} - u_{h_{k-1}}$ at a sample point. We assume:

Assumption A7. *There exist positive constants γ and C_c, independent of h_k, such that $C_k \leq C_c\, h_k^{-\gamma}$ for all $k \in \mathbb{N}_0$.*

If an optimal linear solver is used to solve the finite element equations for u_{h_k}, this assumption holds with $\gamma \approx d$ (see, e.g., [10]), where d is the spatial dimension. Note that the constant C_c will in general depend on the refinement ratio η described in §4.2.1.

We quantify the total computational cost of the MLSC approximation (4.3) using the metric

$$C^{(\mathrm{ML})} = \sum_{k=0}^{K} M_{K-k}\, C_k. \tag{4.6}$$

We now have the following result for the ε-cost of the MLSC method required to achieve an accuracy $\|u - u_K^{(\mathrm{ML})}\|_{L^2_\varrho(\Gamma;H^1_0(D))} \leq \varepsilon$. In the analysis, we define the relations $a \lesssim b$ and $a \eqsim b$ to indicate that $a \leq Cb$ (resp. $a = Cb$) for some constant C independent the mesh width h, the number of interpolation points M and the accuracy ε.

Theorem 4.5. *Suppose assumptions A5–A7 hold with $\sigma_k = M_k^{-\mu}$, and assume that $\alpha \geq \min(\beta, \mu\gamma)$. Then, for any $\varepsilon < \exp[-1]$, there exists an integer K, and a sequence $\{M_k\}_{k=0}^K$, such that*

$$\|u - u_K^{(\mathrm{ML})}\|_{L^2_\varrho(\Gamma;H^1_0(D))} \leq \varepsilon$$

and

$$C_\varepsilon^{(\mathrm{ML})} \lesssim \begin{cases} \varepsilon^{-\frac{1}{\mu}}, & \text{if } \beta > \mu\gamma \\ \varepsilon^{-\frac{1}{\mu}} |\log \varepsilon|^{1+\frac{1}{\mu}} & \text{if } \beta = \mu\gamma \\ \varepsilon^{-\frac{1}{\mu} - \frac{\gamma\mu-\beta}{\alpha\mu}} & \text{if } \beta < \mu\gamma. \end{cases} \tag{4.7}$$

Proof. As in (4.4), we consider separately the two error contributions (I) and (II). To achieve the desired accuracy, it is sufficient to bound both error contributions by $\frac{\varepsilon}{2}$. Without loss of generality, for the remainder of this proof we assume $h_0 = 1$. If this is not the case, we simply need to rescale the constants C_s, C_ζ, and C_c.

First, we choose K large enough so that $(I) \leq \frac{\varepsilon}{2}$. By Assumption A5, it is sufficient to require $C_s h_K^\alpha \leq \frac{\varepsilon}{2}$. Because the hierarchy of meshes $\{h_k\}_{k\in\mathbb{N}_0}$ is obtained by uniform refinement, $h_k = \eta^{-k} h_0 = \eta^{-k}$, and we have

$$h_K \leq \big(\frac{\varepsilon}{2C_s}\big)^{1/\alpha} \quad \text{if} \quad K = \left\lceil \frac{1}{\alpha} \log_\eta \big(\frac{2C_s}{\varepsilon}\big) \right\rceil. \tag{4.8}$$

This fixes the total number of levels K.

In order to obtain the multilevel estimator with the smallest computational cost, we now determine the $\{M_k\}_{k=0}^K$ so that the computational cost (4.6) is minimized, subject to the requirement $(II) \leq \frac{\varepsilon}{2}$. Treating the M_k as continuous variables, we use the Lagrange multiplier method. To begin, we form the Lagrange function, using assumptions A5-A7.

$$\mathcal{L}(M_0,\dots,M_K,\lambda) = \sum_{k=0}^{K} M_{K-k}\,\eta^{k\gamma} + \lambda\left(\sum_{k=0}^{K} C_I\,C_\zeta\,M_{K-k}^{-\mu}\,\eta^{-k\beta} - \varepsilon/2\right).$$

To find a relative extremum, we require $\nabla\mathcal{L}=0$, leading to the $K+2$ conditions

$$\frac{\partial\mathcal{L}}{\partial M_{K-k}} = \eta^{k\gamma} - \lambda C_I\,C_\zeta\mu M_{K-k}^{-(\mu+1)}\eta^{-k\beta} = 0, \quad k=0,\dots,K, \tag{4.9}$$

$$\frac{\partial\mathcal{L}}{\partial\lambda} = \sum_{k=0}^{K} C_I\,C_\zeta\,M_{K-k}^{-\mu}\,\eta^{-k\beta} - \varepsilon/2 = 0. \tag{4.10}$$

Solving the first $K+1$ equations (4.9) for M_{K-k} yields

$$M_{K-k} = (C_I\,C_\zeta\mu\lambda)^{1/(\mu+1)}\eta^{\frac{-k(\beta+\gamma)}{\mu+1}}, \quad k=0,\dots,K. \tag{4.11}$$

Now, substitute (4.11) into (4.10), and solve for λ to obtain

$$\lambda = (2^{\mu+1}C_IC_\zeta)^{1/\mu}\mu^{-1}\varepsilon^{-(\mu+1)/\mu}\mathcal{S}(\eta,K)^{(\mu+1)/\mu},$$

where

$$\mathcal{S}(\eta,K) = \sum_{k=0}^{K}\eta^{-k(\frac{\beta-\gamma\mu}{\mu+1})}.$$

Inserting this into (4.11) results in the optimal choice

$$M_{K-k} = \big(2\,C_I\,C_\zeta\,\mathcal{S}(\eta,K)\big)^{1/\mu}\,\varepsilon^{-1/\mu}\,\eta^{-\frac{k(\beta+\gamma)}{\mu+1}}. \tag{4.12}$$

Because M_{K-k} given by (4.12) is, in general, not an integer, we choose

$$M_{K-k} = \left\lceil (2\,C_I\,C_\zeta\,\mathcal{S}(\eta,K))^{1/\mu}\,\varepsilon^{-1/\mu}\,\eta^{-\frac{k(\beta+\gamma)}{\mu+1}}\right\rceil. \tag{4.13}$$

Note that this choice determines the sequence $\{M_k\}_{k=0}^K$ and consequently $\{\mathcal{I}_{M_k}\}_{k=0}^K$. Also note that, in practice, this choice may not be possible for all interpolation schemes; see Remark 4.6.

With the number of samples M_{K-k} fixed, we now examine the complexity of the multilevel approximation. Since $\lceil x \rceil < x+1$, for any $x \in \mathbb{R}$, we have

$$\begin{aligned} C_\varepsilon^{(\mathrm{ML})} &= \sum_{k=0}^{K} M_{K-k} C_k \eqsim \sum_{k=0}^{K} M_{K-k}\, \eta^{k\gamma} \\ &\lesssim \sum_{k=0}^{K} \left(\frac{\varepsilon}{\mathcal{S}(\eta, K)} \right)^{-\frac{1}{\mu}} \eta^{-k\frac{\beta+\gamma}{\mu+1}}\, \eta^{k\gamma} + \sum_{k=0}^{K} \eta^{k\gamma} \\ &\eqsim \varepsilon^{-\frac{1}{\mu}} \mathcal{S}(\eta, K)^{\frac{1}{\mu}} \sum_{k=0}^{K} \eta^{-k\frac{\beta+\gamma-\gamma(\mu+1)}{\mu+1}} + \sum_{k=0}^{K} \eta^{k\gamma} \\ &\eqsim \varepsilon^{-\frac{1}{\mu}} \mathcal{S}(\eta, K)^{\frac{1}{\mu}} \sum_{k=0}^{K} \eta^{-k\frac{\beta-\gamma\mu}{\mu+1}} + \sum_{k=0}^{K} \eta^{k\gamma} \\ &\eqsim \varepsilon^{-\frac{1}{\mu}} \mathcal{S}(\eta, K)^{1+\frac{1}{\mu}} + \sum_{k=0}^{K} \eta^{k\gamma}. \end{aligned} \tag{4.14}$$

To bound the cost in terms of ε, first note that because $K < \frac{1}{\alpha}\log_\eta(2C_s/\varepsilon) + 1$ by (4.8), we have

$$\sum_{k=0}^{K} \eta^{k\gamma} \le \frac{\eta^{\gamma K}}{1-\eta^{-\gamma}} \le \frac{\eta^{\gamma}(2C_s)^{\gamma/\alpha}}{1-\eta^{-\gamma}} \varepsilon^{-\gamma/\alpha}. \tag{4.15}$$

Next, we need to consider different values of β and μ. When $\beta > \gamma\mu$, $\mathcal{S}(\eta, K)$ is a geometric sum that converges to a limit independent of K. Because $\alpha \ge \gamma\mu$ implies that $\varepsilon^{-\gamma/\alpha} \le \varepsilon^{-\frac{1}{\mu}}$ for $\varepsilon < \exp[-1]$, we have $C_\varepsilon^{(\mathrm{ML})} \lesssim \varepsilon^{-\frac{1}{\mu}}$ in this case.

When $\beta = \gamma\mu$, we find that $\mathcal{S}(\eta, K) = K+1$, and so, using (4.8) and $\alpha \ge \mu\gamma$,

$$C_\varepsilon^{(\mathrm{ML})} \lesssim \varepsilon^{-\frac{1}{\mu}} (K+1)^{1+\frac{1}{\mu}} + \varepsilon^{-\frac{\gamma}{\alpha}} \eqsim \varepsilon^{-\frac{1}{\mu}} |\log \varepsilon|^{1+\frac{1}{\mu}}.$$

For the final case of $\beta < \gamma\mu$, we reverse the index in the sum $\mathcal{S}(\eta, K)$ to obtain a geometric sequence

$$\mathcal{S}(\eta, K) = \sum_{k=0}^{K} \eta^{(k-K)\frac{\beta-\gamma\mu}{\mu+1}} = \eta^{-K\frac{\beta-\gamma\mu}{\mu+1}} \sum_{k=0}^{K} \eta^{-k(\frac{\gamma\mu-\beta}{\mu+1})} \lesssim \varepsilon^{\frac{\beta-\gamma\mu}{\alpha(\mu+1)}}.$$

Because $\alpha \ge \beta$, this gives

$$C_\varepsilon^{(\mathrm{ML})} \lesssim \varepsilon^{-\frac{1}{\mu}} \varepsilon^{\frac{\beta-\gamma\mu}{\alpha(\mu+1)}(1+\frac{1}{\mu})} + \varepsilon^{-\frac{\gamma}{\alpha}} \eqsim \varepsilon^{-\frac{1}{\mu}-\frac{\gamma\mu-\beta}{\alpha\mu}}.$$

This completes the proof. □

Remark 4.6. Error and quadrature level. *In this section, we characterized the convergence of the interpolation errors in terms of the number of interpolation points* M_k*. Yet when computing interpolants based on sparse grid techniques (see Chapter 3), an arbitrary number of points will not in general have an associated sparse grid. Thus, choosing an interpolant using the optimal number of points according to* (4.13) *may not be possible in practice. However, in light of estimates such as [71, Lemma 3.9], it is not unreasonable to make the assumption that given any number of points* M*, there exists an interpolant using* $\widetilde{M}$ *points, with*

$$M \leq \widetilde{M} \leq CM^{\delta} \tag{4.16}$$

for some $\delta \geq 1$*. We can think of* δ *as measuring the inefficiency of our sparse grids in representing higher-dimensional polynomial spaces. Using* (4.16)*, one can proceed as in Theorem 4.5 to derive a bound on the* ε*-cost of the resulting multilevel approximation.*

Another possibility would be to solve a discrete, constrained minimization problem to find optimal interpolation levels, relying on convergence results for the interpolation error in terms of the interpolation level rather than number of points; see [70, Theorem 3.4]. However, our cost metric relies on precise knowledge of the number of points, making theoretical comparison difficult.

Remark 4.7. Cancellations and computational cost. *The cost estimate* (4.6) *takes into consideration the cost of all the terms in the multilevel estimator* (4.3)*. However, when the same interpolation operator is used on two consecutive levels, terms in the multilevel approximation cancel and need in fact not be computed. For example, if* $\mathcal{I}_{M_{K-k}} = \mathcal{I}_{M_{K-k-1}}$*, then*

$$\mathcal{I}_{M_{K-k}}(u_{h_k} - u_{h_{k-1}}) + \mathcal{I}_{M_{K-k-1}}(u_{h_{k+1}} - u_{h_k}) = \mathcal{I}_{M_{K-k}}(u_{h_{k+1}} - u_{h_{k-1}})$$

so that the computation of the interpolants of u_{h_k} *is not necessary. Especially in the context of sparse grid interpolation, in practice we choose the same interpolation grid for several consecutive levels, leading to a significant reduction in the actual computational cost compared to that estimated in Theorem 4.5. The effect of these cancellations is clearly visible in some of the numerical experiments of §4.5.*

Comparison to single level collocation methods

Under the same assumptions as in Theorem 4.5, for any $M_{sl} \in \mathbb{N}_0$ and h_{sl}, the error in the standard single-level SC approximation (4.1) can be bounded by

$$\|u - u^{(\mathrm{SL})}_{M_{sl},h_{sl}}\|_{L^2_\varrho(\Gamma;H^1_0(D))} \leq C_s\, h^{\alpha}_{sl} + C_I\, \zeta(u_h)\, M_{sl}^{-\mu}.$$

To make both contributions equal to $\varepsilon/2$, it suffices to choose $h_{sl} \eqsim \varepsilon^{1/\alpha}$ and $M_{sl} \eqsim \varepsilon^{-1/\mu}$. This choice determines M_{sl} and hence $\mathcal{I}_{M_{sl}}$. The computational cost to achieve a total error of ε is then bounded by

$$C^{(\mathrm{SL})}_{\varepsilon} \eqsim h_{sl}^{-\gamma} M_{sl} \eqsim \varepsilon^{-\frac{1}{\mu}-\frac{\gamma}{\alpha}}.$$

A comparison with the bounds on computational complexity proved in Theorem 4.5 shows clearly the superiority of the multilevel method.

In the case $\beta > \gamma\mu$, the convergence rate of the finite element correction errors is comparatively larger than the convergence rate of the interpolant when multiplied by the cost factor γ. From (4.14), this indicates that the cost $M_{K-k}C_k$ is largest at the coarsest level $k = 0$, and hence most of the computational effort of the multilevel approximation is expended computing $\mathcal{I}_{M_K}(u_{h_0})$. The savings in cost compared to single level SC hence correspond to the difference in cost between obtaining samples u_{h_0} on the coarse grid h_0 and obtaining samples u_{h_K} on the fine grid $h_{sl} = h_K$ used by the single-level method. This gives a saving of $(h_{sl}/h_0)^\gamma \eqsim \varepsilon^{\gamma/\alpha}$.

The case $\beta = \mu\gamma$ corresponds to the computational effort being spread evenly across the levels, and, up to a log factor, the savings in cost are again of order $\varepsilon^{\gamma/\alpha}$.

In contrast, when $\beta < \gamma\mu$, i.e., when the interpolation error is converging quickly compared to the finite element approximations, the computational cost of computing one sample of u_{h_k} grows comparatively quickly with respect to k, and most of the computational effort of the multilevel approximation is on the finest level $k = K$. The benefits compared to single level SC hence corresponds approximately to the difference between M_K and M_{sl}. This gives a savings of $M_K/M_{sl} \eqsim (h^{\beta}_K)^{1/\mu} \eqsim \varepsilon^{\beta/\alpha\mu}$.

4.3.3 Multilevel approximation of functionals

In applications, it is often of interest to bound the error in the expected value of a functional ψ of the solution u, where $\psi : H_0^1(D) \to \mathbb{R}$. Similar to (4.1), the SC approximation of $\psi(u)$ is given by

$$\psi_{k,h}^{(\mathrm{SL})}[u] = \mathcal{I}_{M_k}\left[\psi(u_h)\right] \tag{4.17}$$

and, similar to (4.3), the multilevel interpolation approximation of $\psi(u)$ is given by

$$\psi_K^{(\mathrm{ML})}[u] := \sum_{k=0}^{K} \mathcal{I}_{M_{K-k}}\big(\psi(u_{h_k}) - \psi(u_{h_{k-1}})\big), \tag{4.18}$$

where, as before, we set $u_{h_{-1}} := 0$ and we also assume, without loss of generality, that $\psi(0) = 0$. Note that in the particular case of linear functionals ψ, we in fact have

$$\psi_{k,h}^{(\mathrm{SL})}[u] = \psi(u_{k,h}^{(\mathrm{SL})}) \quad \text{and} \quad \psi_K^{(\mathrm{ML})}[u] = \psi(u_K^{(\mathrm{ML})}).$$

Analogous to Theorem 4.5, we have the following result about the ε-cost for the error $\big|\mathbb{E}\big[\psi(u) - \psi_K^{(\mathrm{ML})}[u]\big]\big|$ in the expected value of the multilevel approximation of functionals.

Proposition 4.8. *Suppose there exist positive constants $\alpha, \beta, \mu, \gamma, C_s, C_I, C_\zeta, C_c$, with $\alpha \geq \min(\beta, \mu\gamma)$, and an operator $\zeta : \Lambda(\Gamma; \mathbb{R}) \to \mathbb{R}$, for a Banach space $\Lambda(\Gamma; \mathbb{R}) \subset L_\varrho^2(\Gamma; \mathbb{R})$ containing the finite element approximations $\{\psi(u_{h_k})\}_{k \in \mathbb{N}_0}$, such that for all $k \in \mathbb{N}_0$ we have*

F1. $|\mathbb{E}[\psi(u) - \psi(u_{h_k})]| \leq C_s\, h_k^\alpha$

F2. $\big|\mathbb{E}\big[\psi(u_{h_k}) - \psi(u_{h_{k-1}}) - \mathcal{I}_{M_{K-k}}(\psi(u_{h_k}) - \psi(u_{h_{k-1}}))\big]\big| \leq C_I\, M_{K-k}^{-\mu}\, \zeta(\psi(u_{h_k}) - \psi(u_{h_{k-1}}))$

F3. $\zeta(\psi(u_{h_k}) - \psi(u_{h_{k-1}})) \leq C_\zeta\, h_k^\beta$

F4. $C_k = C_c\, h_k^{-\gamma}$.

Then, for any $\varepsilon < \exp[-1]$, there exists an integer K and a sequence $\{M_k\}_{k=0}^K$ such that

$$\big|\mathbb{E}\big[\psi(u) - \psi_K^{(\mathrm{ML})}(u)\big]\big| \leq \varepsilon,$$

with computational cost $C_\varepsilon^{(\mathrm{ML})}$ bounded as in Theorem 4.5.

The Assumptions F1–F4 are essentially the same as the Assumptions A5–A7 of Theorem 4.5, with perhaps different values for the constants C_s, C_I, C_ζ, and C_c. Certainly, bounded linear functionals have this inheritance property. In §4.4, we give some examples of nonlinear functionals that also have this property.

4.4 Multilevel approximation using generalized sparse grid interpolants

In this section, we use a specific example of a single level SC method that will be used to construct the interpolation operators in our MLSC approach. As such, recall the definition of the multi-dimensional (including sparse grid) interpolation from Chapter 3, which is defined in (3.3).

$$\mathcal{A}_L^{p,g}[v] = \sum_{g(\boldsymbol{l}) \leq L} \bigotimes_{n=1}^{N} \Delta_n^{p(l_n)}[v].$$

For the specific MLSC method in this section, the general interpolation operators introduced in §4.2.2 are chosen as $\mathcal{I}_{M_k} = \mathcal{A}_{L_k}^{p,g}$ with $M_k := M_{L_k}$. However, we have already noted in Remark 4.6 that an arbitrary number of points will not in general have an associated sparse grid, and in practice a rounding strategy has to be applied to choose the interpolation operator on each level. For examples of rounding strategies, see the numerical examples in §4.5. Note that although in theory this rounding may change the computational complexity of the MLSC estimators, our numerical investigations confirm that the complexities proved in Theorem 4.5 are a good fit in practice.

Remark 4.9. *Note that the sparse grid construction also contains a second notion of* levels. *The levels in the sparse grid case should not be confused with the levels used previously in the multilevel algorithm. For the latter, 'levels' refer to members of hierarchies of spatial and stochastic approximations, both of which were indexed by k. In this section, 'levels' refer to a sequence, indexed by l, of stochastic polynomial spaces and corresponding point sets used to construct a specific sparse grid interpolant. The result of this construction, i.e., of using* *the levels indexed by l, is the interpolants used in the previous sections that were indexed by k.*

The goal of the section is to verify the the assumptions of our multilevel collocation scheme for the generalized global sparse grid operator $\mathcal{I}_{M_k} = \mathcal{A}^{p,g}_{L_k}$. The convergence of the global sparse grid operators applied to the the approximate solutions u_{h_k}, and the functionals $\psi(u_{h_k})$, depends on some analytic regularity of the PDE with respect to the parameterization.

Recalling assumption A2 and the definition of the Bernstein polyellipse (3.7), we have used Lemma 3.1.1 to show that the approximate PDE solutions u_{h_k} are analytic in the region $\Sigma(\boldsymbol{\rho}) \subset \mathbb{C}^N$, for $\boldsymbol{\rho} = (\rho_1, \dots, \rho_N) \in (1, \infty)^N$. We have seen in Theorem 3.2 that under these assumptions, there exist constants $C(N)$ and $\mu(r, N)$, depending on N and $r = \min_{1 \le n \le N} \rho_n$, such that

$$\|v - \mathcal{I}_{M_k} v\|_{L^2_\varrho(\Gamma; H^1_0(D))} \le C(N)\, M_k^{-\mu(r,N)}\, \zeta(v),$$

where

$$\zeta(v) \equiv \max_{\boldsymbol{z} \in \Sigma(\boldsymbol{\rho})} \|v(\boldsymbol{z})\|_{H^1_0(D)}.$$

We thus verify the convergence assumptions A6 and those given in F2 and F3 by showing that the bounds on the interpolation error above apply to the approximate solutions u_{h_k} and the functionals $\psi(u_{h_k})$, for $k \in \mathbb{N}_0$, Define the Banach space $\Lambda(\Gamma; H^1_0(D))$ consisting of all functions $v \in C^0(\Gamma; H^1_0(D))$ such that v admits an analytic extension in the region $\Sigma(\boldsymbol{\rho})$. It follows from Lemma 3.1.1 that, under appropriate assumptions on a, we have $u \in \Lambda(\Gamma; H^1_0(D))$. Because the dependence on $\boldsymbol{y}$ is unchanged in the approximate solution u_{h_k}, it also follows that $u_{h_k} \in \Lambda(\Gamma; H^1_0(D))$ for all $k \in \mathbb{N}_0$, and hence also $u_{h_k} - u_{h_{k-1}} \in \Lambda(\Gamma; H^1_0(D))$ for all $k \in \mathbb{N}$.

Similar to Assumption A5, it follows from standard finite element theory [10, 20] that with ζ as in (3.8), $\zeta(u_{h_k})$ can be bounded by a constant independent of k, whereas $\zeta(u_{h_k} - u_{h_{k-1}})$ can be bounded by a constant multiple of h_k^α for some $\alpha > 0$. In general, the constants appearing in these estimates will depend on norms of a and f as well as on the mesh refinement parameter η. We can hence conclude that with $\mathcal{I}_{M_k} = \mathcal{A}^{p,g}_{L_k}$, Assumption A6 is satisfied for the interpolation schemes considered in Theorem 3.2. Therefore, for the numerical examples presented in §4.5, we utilize the sparse grid stochastic collocation as the interpolation scheme.

Now we verify the analyticity assumption in Theorem 3.2 also for the functionals $\psi(u)$. Because Lemma 3.1.1 already gives an analyticity result for u, we use the following result, which can be found in [97], about the composition of two functions on general normed vector spaces.

Theorem 4.10. *Let X_1, X_2, and X_3 denote normed vector spaces and let $\theta : X_1 \to X_2$ and $\nu : X_2 \to X_3$ be given. Suppose that θ is analytic on X_1, ν is analytic on X_2 and $\theta(X_1) \subseteq X_2$. Then the composition $\nu \circ \theta : X_1 \to X_3$ is analytic on X_1.*

Hence, if we can show that ψ is an analytic function of u, we can conclude that $\psi(u)$ is analytic on $\Sigma(\boldsymbol{\rho})$. To this end, we need the notion of analyticity for functions defined on general normed vector spaces, which we will now briefly recall.

Given normed vector spaces X_1 and X_2 and an infinitely Frèchet differentiable function $\theta : X_1 \to X_2$, we can define a Taylor series expansion of θ at the point ξ in the following way [12]:

$$T_{\theta,\xi}(x) = \sum_{j=0}^{\infty} \frac{1}{j!} d^j\theta(\xi)(x-\xi)^j, \tag{4.19}$$

where $x, \xi \in X_1$, the notation $(x-\xi)^j$ denoting the j-tuple $(x-\xi, \dots, x-\xi)$ and $d^j\theta(\xi)$ denoting the j-linear operator corresponding to the j-th Frèchet differential $D^j\theta(\xi)$. The function θ is then said to be *analytic* in a set $Z \subset X_1$ if, for every $z \in Z$, $T_{\theta,z}(x) = \theta(x)$ for all x in a neighbourhood $N_r(z) = \{x \in Z : \|x-z\|_{X_1} < r\}$, for some $r > 0$. The following result now immediately follows from Theorem 4.10.

Lemma 4.10.1. *Let the assumptions of Lemma 3.1.1 be satisfied. Suppose ψ, viewed as a mapping from $H_0^1(D)$ to $\mathbb{R}$, is analytic in the set $\Sigma(u) \subset H_0^1(D)$, and $u(\boldsymbol{z}; x) \in \Sigma(u)$ for all $\boldsymbol{z} \in \Sigma(\boldsymbol{\rho})$. Then, $\psi \circ u$, viewed as a mapping from Γ to $\mathbb{R}$, admits an analytic extension to the set $\Sigma(\boldsymbol{\rho})$.*

Together with Theorem 3.2, now with $W = \mathbb{R}$, it then follows from Lemma 4.10.1 that assumptions F2 and F3 in Proposition 4.8 are satisfied for the interpolation schemes considered in this section, provided the functional ψ is an analytic function of u. Note that the function ζ in Theorem 3.2 acts on $\psi(u)$ instead of u in this case, leading to optimal convergence rates in h of the stochastic interpolation error.

To finish the analysis, we give some examples of functionals that satisfy the assumptions of Lemma 4.10.1. We in particular make use of the following result on Taylor expansions [12].

Lemma 4.10.2. *Let $\theta : X_1 \to X_2$, for normed vector spaces X_1 and X_2, and let $Z \subset X_1$. If $\|d^j\theta(z)\| \leq C^j j!$ for all $z \in Z$ and some $C < \infty$, where $\|\cdot\|$ denotes the usual operator norm, then θ is analytic on Z. In particular, θ is analytic on Z if $\|d^j f(z)\| = 0$ for all $z \in Z$ and all $j \geq j^*$, for some $j^* \in \mathbb{N}$.*

Example 4.11. (*Bounded linear functionals*) In this case, for any $v, w \in H_0^1(D)$, we have

$$d\psi(v)(w) = \psi(w) \qquad \text{and} \qquad d^j\psi(v) \equiv 0 \quad \forall\, j \geq 2,$$

which implies that ψ is analytic on all of (complex-valued) $H_0^1(D)$. Examples of bounded linear functionals include point evaluations of the solution u in one spatial dimension and local averages of the solution u in some subdomain $D^* \subset D$, computed as $\frac{1}{|D^*|}\int_{D^*} u\mathrm{d}x$, in any spatial dimension.

Example 4.12. (*Higher order moments of bounded linear functionals*) As a generalization of the above example, consider the functional $\psi(v) = \Psi(v)^q$, for some bounded linear functional Ψ on $H_0^1(D)$ and some $q \in \mathbb{N}$. For any $v \in H_0^1(D)$, the differentials of ψ are

$$\begin{aligned} &d^j\psi(v)(w_1, \ldots, w_j) = \Psi(v)^{q-j} \prod_{i=1}^{j} (q - i + 1)\, \Psi(w_i), && 1 \leq j \leq q, \\ &d^j\psi(v) \equiv 0, && j \geq q + 1, \end{aligned}$$

from which it follows that ψ is analytic on all of $H_0^1(D)$.

Example 4.13. (*Spatial L^2-norm*) Consider the functional $\psi(v) = \int_D v^2 \mathrm{d}x = \|v\|^2_{L^2(D)}$. For any $v \in H^1_0(D)$, the differentials of ψ are

$$\begin{aligned}
d\psi(v)(w_1) &= \lim_{\delta\to 0} \frac{\int_D (v+\delta w_1)^2 - \int_D v^2}{\delta} = \lim_{\delta\to 0} \frac{\int_D \delta v w_1 + \int_D \delta^2 w_1^2}{\delta} = 2\int_D v w_1, \\
d^2\psi(v)(w_1, w_2) &= \lim_{\delta\to 0} \frac{2\int_D (v+\delta w_2) w_1 - 2\int_D v w_1}{\delta} = 2\int_D w_2 w_1, \\
d^j\psi(v) &\equiv 0 \ \ \forall j \geq 2,
\end{aligned}$$

which implies that ψ is analytic on the entire space $H^1_0(D)$. For the functional $\psi(v) = \|v\|_{L^2(D)}$, we use Theorem 4.10 and the analyticity of the square root function on $(0, \infty)$ to conclude that ψ is analytic on any subset $\Sigma(u) \subseteq H^1_0(D)$ not containing 0.

The analysis in this example can easily be extended to the functionals $\|v\|_{H^1_0(D)}$ and $\|v\|^2_{H^1_0(D)}$.

4.5 Numerical Examples

The aim of this section is to demonstrate numerically the significant reductions in computational cost possible with the use of the MLSC approach. As an example, consider the following boundary value problem on either $D = (0, 1)$ or $D = (0, 1)^2$:

$$\begin{cases} -\nabla \cdot (a(\boldsymbol{y}, \mathbf{x}) \nabla u(\boldsymbol{y}, \mathbf{x})) &= 1 \qquad \text{for } \mathbf{x} \in D \\ u(\boldsymbol{y}, \mathbf{x}) &= 0 \qquad \text{for } \mathbf{x} \in \partial D. \end{cases} \tag{4.20}$$

The coefficient a takes the form

$$a(\boldsymbol{y}, \mathbf{x}) = 0.5 + \exp\left[\sum_{n=1}^{N} \sqrt{\lambda_n} b_n(\mathbf{x}) y_n\right], \tag{4.21}$$

where $\{y_n\}_{n\in\mathbb{N}}$ is a sequence of independent, uniformly distributed random variables on [-1,1] and $\{\lambda_n\}_{n\in\mathbb{N}}$ and $\{b_n\}_{n\in\mathbb{N}}$ are the eigenvalues and eigenfunctions of the covariance operator with kernel function $C(x, x') = \exp[-\|\mathbf{x}-\mathbf{x}'\|_1]$. Explicit expressions for $\{\lambda_n\}_{n\in\mathbb{N}}$ and $\{b_n\}_{n\in\mathbb{N}}$

are computable [41]. In the case $D = (0,1)$, we have

$$\lambda_n^{\text{1D}} = \frac{2}{w_n^2+1} \quad \text{and} \quad b_n^{\text{1D}}(\mathbf{x}) = A_n(\sin(w_n\mathbf{x}) + w_n\cos(w_n\mathbf{x})) \quad \text{for all } n \in \mathbb{N},$$

where $\{w_n\}_{n\in\mathbb{N}}$ are the (real) solutions of the transcendental equation

$$\tan(w) = \frac{2\,w}{w^2-1}$$

and the constant A_n is chosen so that $\|b_n\|_{L^2(0,1)} = 1$. In two spatial dimensions, with $D = (0,1)^2$, the eigenpairs can be expressed as

$$\lambda_n^{\text{2D}} = \lambda_{i_n}^{\text{1D}}\,\lambda_{j_n}^{\text{1D}} \quad \text{and} \quad b_n^{\text{2D}} = b_{i_n}^{\text{1D}}\,b_{j_n}^{\text{1D}}$$

for some $i_n, j_n \in \mathbb{N}$. In both one and two spatial dimensions, the eigenvalues λ_n decay quadratically with respect to n [14].

Let $a^*(\boldsymbol{z},\mathbf{x}) = 0.5 + \exp\left[\sum_{n=1}^N \sqrt{\lambda_n} b_n(\mathbf{x}) z_n\right]$ be the complex extension of a. Given a multiindex $\nu \in \mathbb{N}_0^N$, it is easy to see that the mixed partial derivatives of a^* satisfy

$$\partial_\nu a^*(\boldsymbol{z},\mathbf{x}) := \frac{\partial^{|\nu|}a}{\partial^{\nu_1}z_1\ldots\partial^{\nu_N}z_N}(\boldsymbol{z},\mathbf{x}) = a(\boldsymbol{z},\mathbf{x})\prod_{n=1}^N(\sqrt{\lambda_n}b_n(\mathbf{x}))^{\nu_n}.$$

Thus, given $\boldsymbol{z} \in \mathbb{C}^N$, the power series

$$a^*(\boldsymbol{z}',\mathbf{x}) = \sum_{\nu\in\mathbb{N}_0^N}\frac{\partial_\nu a^*(\boldsymbol{z},\mathbf{x})}{\nu!}\prod_{n=1}^N(z_n'-z_n)^{\nu_n}$$

converges for all $\boldsymbol{z}' \in \mathbb{C}^N$ such that $|z_n' - z_n| < \frac{1}{\sqrt{\lambda_n}\|b_n(\mathbf{x})\|_{L^\infty(D)}}$, $n = 1,\ldots,N$, and thus $a(\boldsymbol{z},\mathbf{x})$ satisfies Assumption A2.

For spatial discretization, we use continuous, piecewise-linear finite elements on uniform triangulations of D, starting with a mesh width of $h = 1/2$. As interpolation operators, we choose the (isotropic) sparse grid interpolation operator (4.4), using p and g given by the classic Smolyak approximation in Table 3.1, based on Clenshaw-Curtis abscissas; see Chapter 3.

The goal of the computations is to estimate the error in the expected value of a functional ψ of the solution of (4.20). For fair comparisons, all values of ε reported are relative accuracies, i.e., we have scaled the errors by the value of $\mathbb{E}[\psi(u)]$ itself. We consider two different settings: in §4.5.1, we consider problem (4.20) in two spatial dimensions with $N = 10$ random variables whereas, in Sections 4.5.2 and 4.5.3, we work in one spatial dimension with $N = 20$ random variables. Because the exact solution u is unavailable, the error in the expected value of $\psi(u)$ has to be estimated. In Sections 4.5.1 and 4.5.2, we compute the error with respect to an "overkilled" reference solution obtained using a fine mesh spacing h^* and high interpolation level L^*. However, because this is generally not feasible in practice, we show in §4.5.3 how the error can be estimated when the exact solution is not available and one cannot compute using a fine spatial mesh and high stochastic interpolation level. The cost of the multilevel estimators is computed as discussed in §4.3.2 and Remark 4.7, with $\gamma = d$, i.e., by assuming the availability of an optimal linear solver. For non-optimal linear solvers for which $\gamma > d$, the savings possible with the multilevel approach will be even greater than demonstrated below.

4.5.1 $\mathbf{d = 2, N = 10}$

As the quantity of interest, we choose the average value of u in a neighborhood of the midpoint $(1/2, 1/2)$, computed as $\psi(u) = \frac{1}{|D^*|}\int_{D^*} u(\mathbf{x})\mathrm{d}\mathbf{x}$, where D^* denotes the union of the six elements adjacent to the node located at $(1/2, 1/2)$ of the uniform triangular mesh with mesh size $h = 1/256$.

We start by confirming, in Figure 4.1, the assumptions of Proposition 4.8. The reference values are computed with spatial mesh width $h^* = 1/256$ and stochastic interpolation level $L^* = 5$.

The top-left plot of Figure 4.1 shows the convergence of the finite element error in the expected value of $\psi(u)$, and confirms that assumption F1 of Proposition 4.8 holds with $\alpha = 2$.

The top-right plot of Figure 4.1 shows the absolute value of the interpolation error in the quantities $\psi(u_h)$ and $\psi(u_h) - \psi(u_{2h})$ for a fixed interpolation level $l = 1$, i.e. for fixed M_l, as a function of h. We see that the interpolation error in $\psi(u_h)$ is bounded by a constant

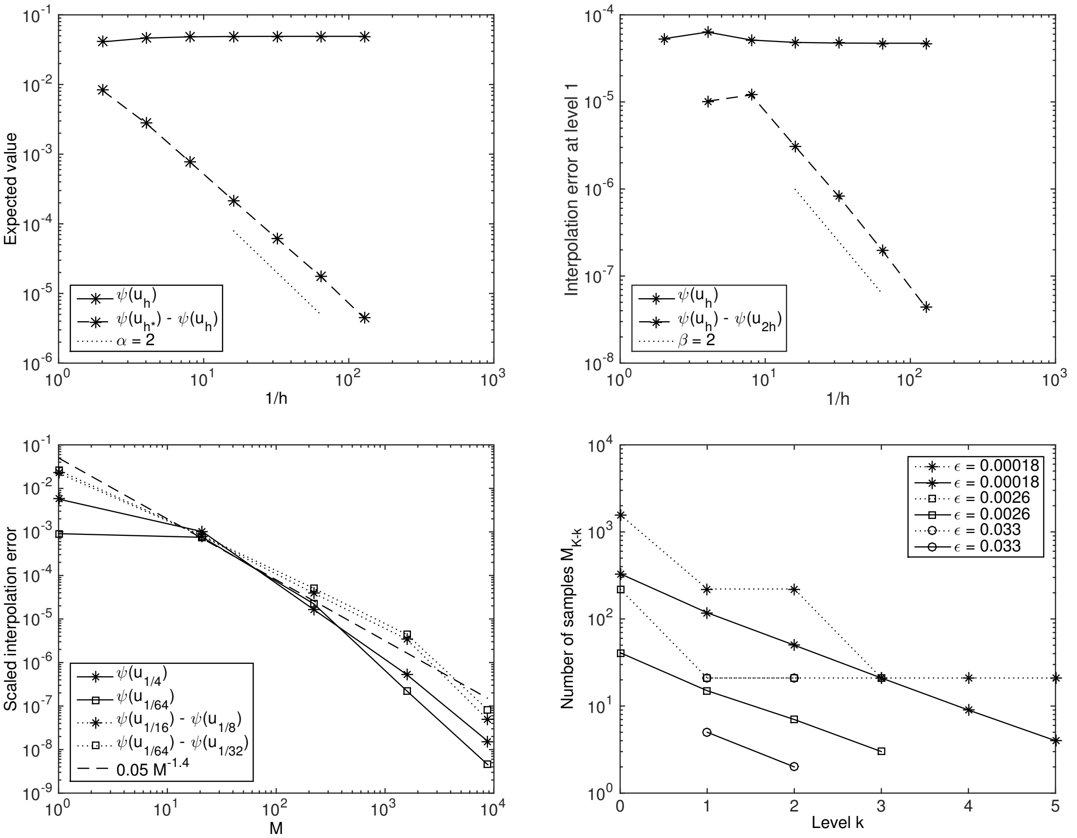

Figure 4.1: $D = (0,1)^2$, $N = 10$. Top left: $\mathbb{E}[\mathcal{I}_{M_5}\psi(u_h)]$ and $\mathbb{E}[\mathcal{I}_{M_5}(\psi(u_{1/256}) - \psi(u_h))]$ versus $1/h$ (assumption F1). Top right: $|\mathbb{E}[(\mathcal{I}_{M_5} - \mathcal{I}_{M_l})\psi(u_h)]|$ and $|\mathbb{E}[(\mathcal{I}_{M_5} - \mathcal{I}_{M_l})(\psi(u_h) - \psi(u_{2h}))]|$ versus $1/h$ (assumption F3). Bottom left: $|\mathbb{E}[(\mathcal{I}_{M_5} - \mathcal{I}_{M_l})\psi(u_h)]/h_0^2|$ and $|\mathbb{E}[(\mathcal{I}_{M_5} - \mathcal{I}_{M_l})(\psi(u_h) - \psi(u_{2h}))]/h^2|$ versus M_l, for various h (assumption F2). Bottom right: number of samples M_{K-k} versus k.

independent of h, whereas the interpolation error in $\psi(u_h) - \psi(u_{2h})$ decays quadratically in h. This confirms assumption F3 with $\beta = 2$.

The bottom-left plot of Figure 4.1 shows the interpolation error in $\psi(u_h)$ scaled by h_0^2 and the interpolation error in $\psi(u_h) - \psi(u_{2h})$ scaled by h^2 for several values of h. According to assumptions F2 and F3, these plots should all result in a straight line $CM^{-\mu}$, where $C = C_I C_\zeta$. The best fit which has $C = 0.05$ and $\mu = 1.4$ is added for comparison.

The bottom-right plot of Figure 4.1 shows the number of samples M_k computed using the formula (4.12), with $C = 0.05$ and $\mu = 1.4$, for several values of ε. The finest level K was determined using the estimates on the finite element error from the top-left plot. Solid lines correspond to numbers rounded up to the nearest integer, as is done in (4.13), whereas dotted lines correspond to the number of samples rounded up to the next level of the sparse grid. As stated in Remark 4.7, when the same number of points are used for consecutive levels, cancellations occur leading to savings in cost.

In Figure 4.2, we study the cost of the standard and multilevel collocation methods to achieve a given total accuracy ε. In both plots, the data labeled 'SC' and 'MLSC' denote standard and multilevel stochastic collocation, respectively. For data labeled 'formula', the number of samples was determined by the formula (4.12) with $C = 0.05$ and $\mu = 1.4$, rounded up to the next sparse grid level (the dotted lines in the bottom right plot of Figure 4.1). For data labeled 'best', the number of samples was chosen by trial and error so as to achieve a total accuracy ε for the smallest computational cost. For all methods, we chose $h_0 = 1/4$.

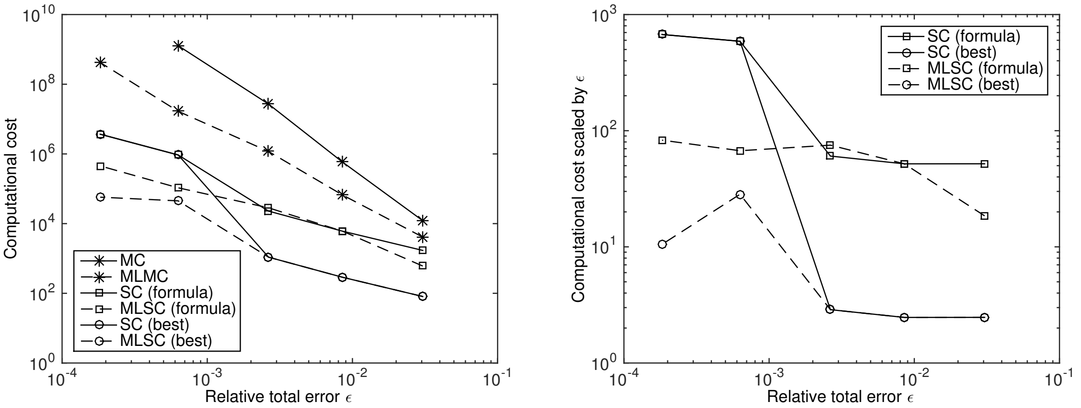

Figure 4.2: $D = (0, 1)^2$, $N = 10$. Left: computational cost versus relative error ε. Right: computational cost scaled by $\varepsilon^{-1.36}$ versus relative error ε.

In the left plot of Figure 4.2, we simply plot the computational cost of the different estimators against ε. For comparison, we have also added corresponding results for Monte Carlo (MC) and multilevel Monte Carlo (MLMC) estimators. In both the 'formula' and the 'best' case, the multilevel collocation method outperforms standard SC. Both collocation-based methods outperform both Monte Carlo approaches.

In the right plot in Figure 4.2, we compare the observed computational cost with that predicted by Proposition 4.8 for the standard and multilevel collocation methods. In our computations, we observed $\alpha \approx 2$, $\beta \approx 2$, and $\mu \approx 1.4$, which with $\gamma = 2$ gives computational costs of ε^{-1} and $\varepsilon^{-1.72}$ for the multilevel and standard SC method, respectively. We therefore plot the computational cost scaled by ε^{1}. We see that both multilevel methods indeed seem to grow approximately like ε^{-1}, with the 'formula' case growing slightly faster for large value of ε and the 'best' case growing slightly faster for small values of ε. The costs for both standard collocation methods grow a lot faster with ε.

Figure 4.3 provides results for a different quantity of interest, $\psi(u) = \|u\|_{L^2(D)}$. The left plot corresponds to the bottom-left plot in Figure 4.1 and again confirms that the interpolation error in $\psi(u_h) - \psi(u_{2h})$ scales with h^2. The right plot corresponds to the left plot of Figure 4.2, where we plot the computational cost of the different estimators against ε. We see that all collocation-based methods outperform the Monte Carlo approaches. In both the 'formula' and the 'best' case, the multilevel collocation method again outperforms standard SC.

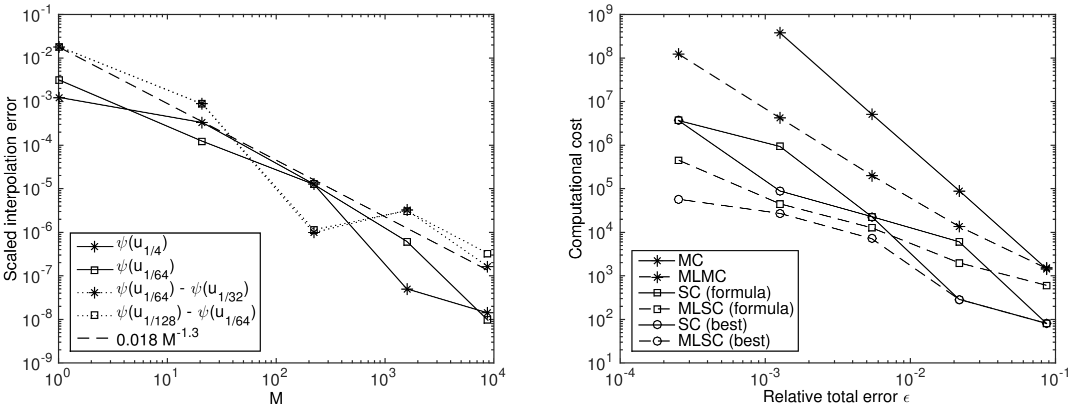

Figure 4.3: $D = (0,1)^2$, $N = 10$. Left: $\mathbb{E}[\mathcal{I}_5\psi(u_h) - \mathcal{I}_{M_k}\psi(u_h)]/h_0^2$ and $[\mathcal{I}_5(\psi(u_h) - \psi(u_{2h})) - \mathcal{I}_{M_k}(\psi(u_h) - \psi(u_{2h})]/h^2$ versus M_k, for various h. Right: computational cost versus relative error ε.

Remark 4.14. *Before considering the second model problem, let us briefly comment on the differences between the 'best' and the 'formula' multilevel methods. The 'formula' multilevel collocation method performs sub-optimally mainly for two reasons. First, it always rounds up the number of samples M_k to the nearest sparse grid level, which may be substantially higher than the number of samples actually required. Secondly, it does not take into account sign changes in the interpolation error, which in practice can lead to significant reductions in the interpolation error of the multilevel method. For both of these reasons, the interpolation error is often a lot smaller than the required $\varepsilon/2$, leading to sub-optimal performance. This* *issue is partly addressed in §4.5.3, where we consider not always rounding up, but rounding the number of samples either up or down to the nearest sparse grid level.*

4.5.2 $\mathbf{d = 1, N = 20}$

We now repeat the numerical tests done in the previous section for the case $D = (0, 1)$ and $N = 20$. For the quantity of interest, we choose the expected value of the solution u evaluated at $x = \frac{3}{4}$. The reference values are computed using the mesh width $h^* = 1/1024$ and interpolation level $L^* = 5$.

We again start by confirming, in Figure 4.4, the assumptions of Proposition 4.8. The four plots of that figure convey the same information as do the corresponding plots in Figure 4.1 and again confirm assumptions F1, F2, and F3 of that theorem with $\alpha = 2$ and $\beta = 2$ and, in the bottom-left plot, the best line fit $C = C_I C_\zeta$ with $C = 0.005$ and $\mu = 0.8$.

Figure 4.5 conveys the same information and uses the same labeling as does Figure 4.2. Again, for both the 'formula' and 'best' cases, the multilevel collocation method eventually outperforms standard SC and both collocation-based methods also outperform the Monte Carlo approaches. Based on the values $\alpha \approx 2$, $\beta \approx 2$, and $\mu \approx 0.8$, Proposition 4.8 now predicts the computational costs of $\varepsilon^{-1.25}$ and $\varepsilon^{-1.75}$ for the multilevel and the standard collocation methods, respectively. The right-plot in Figure 4.5 indicates that the 'formula' multilevel collocation method indeed seems to grow like $\varepsilon^{-1.25}$ whereas the 'best' multilevel method actually seems to grow slower for small values of ε. This is likely due to the different signs of the interpolation errors in the multilevel estimator. Also, again, the costs for both standard collocation methods grow a lot faster with ε.

4.5.3 Practical implementation

In Sections 4.5.1 and 4.5.2, the accuracy of the computed estimates was assessed by comparison to a reference solution. Of course, in practice, a fine-grid, high-level reference solution is not available. Therefore, in this section, we describe how to implement the MLSC method without having recourse to a reference solution. We suggest the following practical strategy that is similar to the one proposed in [42]. In order to determine the

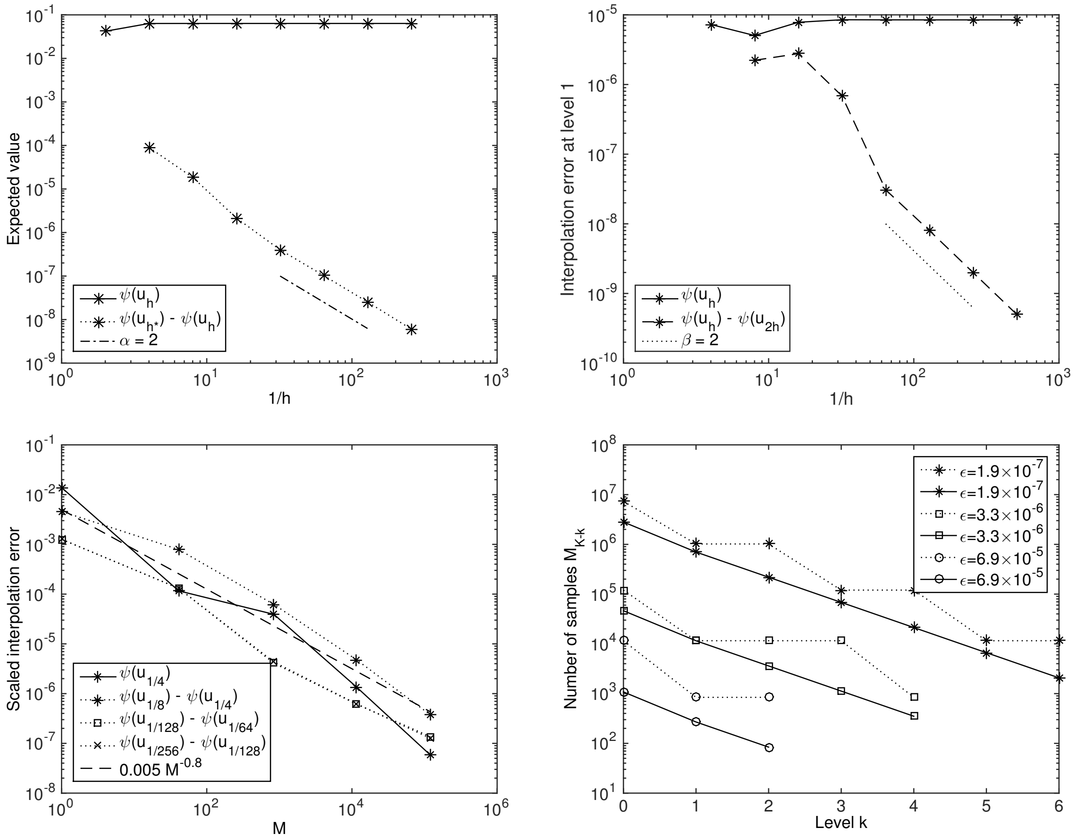

Figure 4.4: $D = (0,1)$, $N = 20$. Top left: $\mathbb{E}[\mathcal{I}_{M_5}\psi(u_h)]$ and $\mathbb{E}[\mathcal{I}_{M_5}(\psi(u_{1/1024}) - \psi(u_h))]$ versus $1/h$ (assumption F1). Top right: $|\mathbb{E}[(\mathcal{I}_{M_5} - \mathcal{I}_{M_l})\psi(u_h)]|$ and $|\mathbb{E}[(\mathcal{I}_{M_5} - \mathcal{I}_{M_l})(\psi(u_h) - \psi(u_{2h}))]|$ versus $1/h$ (assumption F3). Bottom left: $|\mathbb{E}[(\mathcal{I}_{M_5} - \mathcal{I}_{M_l})\psi(u_h)]/h_0^2|$ and $|\mathbb{E}[(\mathcal{I}_{M_5} - \mathcal{I}_{M_l})(\psi(u_h) - \psi(u_{2h}))]/h^2|$ versus M_l, for various h (assumption F2). Bottom right: number of samples M_{K-k} versus k.

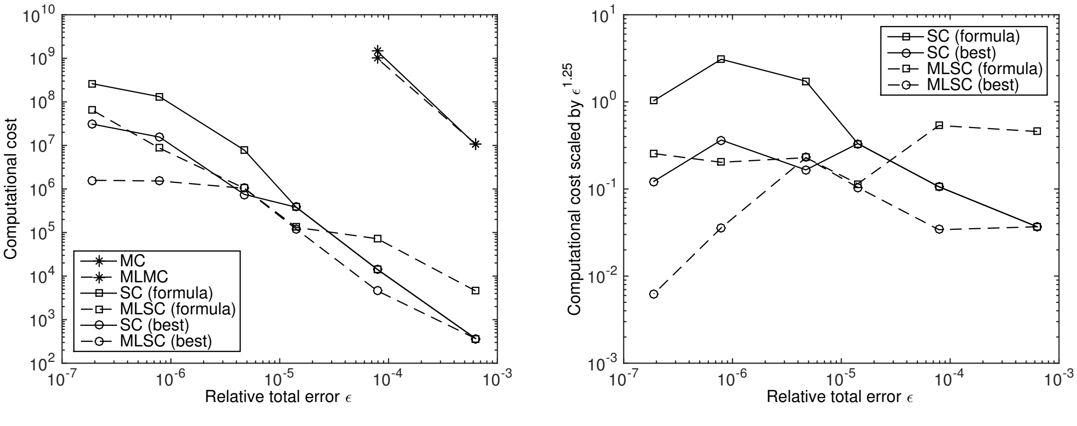

Figure 4.5: $D = (0,1)$, $N = 20$. Left: computational cost versus relative error ε. Right: computational cost scaled by $\varepsilon^{-1.25}$ versus relative error ε.

number of levels we need, we assume that equality holds on assumption F1, i.e. we assume $\mathbb{E}[\psi(u) - \psi(u_{h_k})] = C_s h_k^\alpha$, and use the equality

$$\begin{aligned}\mathbb{E}[\psi(u_{h_k}) - \psi(u_{h_{k-1}})] &= \mathbb{E}[\psi(u) - \psi(u_{h_{k-1}})] - \mathbb{E}[\psi(u) - \psi(u_{h_k})] \\ &= C_s h_{k-1}^\alpha - C_s h_k^\alpha \\ &= (\eta^\alpha - 1)\, \mathbb{E}[(\psi(u) - \psi(u_{h_k}))],\end{aligned}$$

where we recall that $\eta = h_{k-1}/h_k$. Hence, the condition $\mathbb{E}[\psi(u) - \psi(u_{h_k})] \leq \varepsilon/2$ is equivalent to the condition $\mathbb{E}[\psi(u_{h_k}) - \psi(u_{h_{k-1}})] \leq (\eta^\alpha - 1)\varepsilon/2$. We then have the following algorithm.

1. Estimate the constants α, β, μ, and $C = C_I\, C_\zeta$.

2. Start with $K = 1$.

3. Calculate the optimal number of samples $M_k, k = 0, \ldots, K$, according to the formula (4.12), and round to the nearest sparse grid level.

4. Test for convergence by checking if there holds

$$\mathbb{E}[\psi(u_{h_k}) - \psi(u_{h_{k-1}})] \leq (\eta^\alpha - 1)\, \varepsilon/2.$$

5. If not converged, set $K = K + 1$ and return to step 3.

Note that in this procedure, steps 3 and 4 ensure that the interpolation error and the spatial discretization error are each less than the required tolerance $\varepsilon/2$, respectively.

The estimation of the constants α, β, μ, and C in step 1 can be done relatively cheaply from computations done using mesh widths h_0, h_1, and h_2 and interpolation levels $k = 0, 1, 2$. It is of course also possible to iterate over step 1, in the same manner as we iterate over steps 3 and 4, and to continuously update our estimates of these constants as we increase the number of levels in our multilevel estimator. This approach would eliminate some of the problems related to possible pre-asymptotic effects. It is also possible to use the idea behind the continuation MLMC (CMLMC) method in [24] and use a Bayesian approach to estimating the constants.

We test the algorithm using the the model problem from §4.5.2. For the results provided below, we estimated the convergence rate α from the level 1 interpolants $\mathcal{I}_{M_1}$ of $\psi(u_0)$, $\psi(u_1)$, and $\psi(u_2)$, resulting in $\alpha \approx 2.1$. In light of the results in Chapter 3, we assumed $\beta = \alpha$. We then used the first three interpolation levels of $\psi(u_0)$ and $\psi(u_1) - \psi(u_0)$ to obtain the estimates $C \approx 0.01$ and $\mu \approx 0.8$. Note that the value of μ is the same as in §4.5.2 whereas the value of the constant C is slightly larger. This is due to the fact that, for the large values of h used to estimate this constant, the function $\zeta(\psi(u_h) - \psi(u_{2h}))$ has probably not yet settled into its asymptotic quadratic decay.

As mentioned in §4.5.1, always rounding the number of samples resulting from formula (4.12) up to the next sparse grid level may lead to a substantial increase in the computational cost and hence a sub-optimal performance of the multilevel method. In practice, one might therefore consider not always rounding up, but instead rounding either up or down. As long as we do not round down more frequently than we round up, or at least not much more often, this approach should still result in an interpolation error below the required tolerance $\varepsilon/2$.

Table 4.1 shows the number of samples M_{K-k} resulting from the implementation described in this section for the model problem with $d = 1$ and $N = 20$ from §4.5.2. For each value of ε, the first row, denoted by 'formula', corresponds to the numbers M_{K-k} resulting from formula (4.12) rounded up to the nearest integer. The second row, denoted 'up', are the numbers in the first row rounded up to the next corresponding sparse grid level. For the final row, denoted 'up/down', the rounding of the number of samples was done in the following way: first, all numbers were rounded either up or down to the nearest corresponding sparse grid level. If this resulted in more numbers being rounded down than up, we chose the number that was rounded down by the largest amount and then instead rounded this number up. This procedure was continued iteratively. The same was done when more numbers were rounded up than down.

To confirm that the adaptive procedure still achieves the required tolerance on the total error, we have, for Table 4.2, computed the stochastic interpolation and finite element errors (with respect to a reference solution) and the computational cost of the multilevel approximations from Table 4.1. For comparison, we have added the results for the multilevel

Table 4.1: $D = (0, 1)$, $N = 20$. Number of samples M_{K-k} computed using formula (4.12) and various rounding schemes.

ε	level	0	1	2	3	4
6.3e-4	formula	191	48	15		
	up	841	841	41		
	up/down	841	41	41		
7.9e-5	formula	3002	747	233	73	
	up	11561	841	841	841	
	up/down	841	841	841	41	
1.4e-5	formula	27940	6949	2169	677	212
	up	120401	11561	11561	841	841
	up/down	11561	11561	841	841	841
4.7e-6	formula	110310	27433	8562	2672	834
	up	120401	120401	11561	11561	841
	up/down	120401	11561	11561	11561	841

method which was manually found to give a total error less than ε at minimal cost, which was already computed in §4.5.2 assuming a reference solutions was available. Note that for large values of ε, the adaptive procedure described in this section overestimated the finite element error, leading to a larger number of levels K compared to that found in §4.5.2. It is clear from Table 4.2 that not only does the alternative rounding procedure yield the required bound on the error, it also significantly reduces the computational cost of the multilevel method, bringing it close to what was manually found to be the minimal cost possible.

4.6 Remarks

Computing solutions of stochastic partial differential equations using stochastic collocation methods can become prohibitively expensive as the dimension of the random parameter space increases. Drawing inspiration from recent work in multilevel Monte Carlo methods, this work proposed a multilevel stochastic collocation method, based on a hierarchy of spatial and stochastic approximations. A detailed computational cost analysis showed, in all cases,

Table 4.2: $D = (0, 1)$, $N = 20$. Stochastic interpolation and spatial errors (with respect to the reference solution) and computational cost of various multilevel methods.

ε		Interpolation error	Spatial error	Cost
6.3e-4	up	6.7e-5	3.4e-5	8266
	up/down	2.8e-4	3.4e-5	4902
	best	8.0e-5	2.9e-4	369
7.9e-5	up	2.2e-5	6.3e-6	85558
	up/down	3.0e-5	6.3e-6	15650
	best	2.4e-5	3.4e-5	4591
1.4e-5	up	2.7e-6	1.6e-6	853207
	up/down	8.3e-6	1.6e-6	158714
	best	3.9e-6	6.3e-6	119699
4.7e-6	up	7.3e-8	1.6e-6	1519787
	up/down	1.2e-6	1.6e-6	1038183
	best	1.2e-6	1.6e-6	1038183

a sufficient improvement in costs compared to single-level methods. Furthermore, this work provided a framework for the analysis of a multilevel version of any method for SPDEs in which the spatial and stochastic degrees of freedom are decoupled.

The numerical results practically demonstrated this significant decrease in complexity versus single level methods for each of the problems considered. Likewise, the results for the model problem showed multilevel SC to be superior to multilevel MC even up to $N = 20$ dimensions.

One of the largest obstacles to the practicality of stochastic collocation methods is the huge growth in the number of points between grid levels. In the multilevel case, this can lead to a large amount of computational inefficiency. Certain simple rounding schemes were proposed to mitigate this effect, and proved to be extremely effective for the problems considered. Similarly, since most of our example problems involved computation of a reference solution for the estimation of the necessary constants, a more practical multilevel stochastic collocation algorithm that dispensed with the need for a reference solution was proposed and tested.

It is clear that for any sampling method for SPDEs, whether Monte Carlo or stochastic collocation, multilevel methods are to be preferred over single-level methods for improved efficiency. Especially in the case of stochastic collocation methods, multilevel approaches enable one to further delay the curse of dimensionality, tempering the explosion of computational effort that results when the stochastic dimension increases. Though Monte Carlo methods are often preferable for problems involving a large stochastic dimension, multilevel approaches greatly improve the effectiveness of stochastic collocation methods versus Monte Carlo methods.

Chapter 5

Accelerating Stochastic Collocation Methods

Some content of this chapter first appeared in the paper [36], in the SIAM/ASA Journal of Uncertainty Quantification, published by the Society for Industrial and Applied Mathematics (SIAM) and the American Statistical Association (ASA). Copyright by SIAM and ASA. Unauthorized reproduction is prohibited. The work was accomplished in collaboration with Diego Galindo, Clayton Webster, and Guannan Zhang. It has been slightly edited to maintain consistency with other chapters in this manuscript, and much of the introductory material has been moved to Chapters 1.

In this chapter, we propose another general acceleration technique for decreasing the computational complexity of stochastic collocation methods to solve PDEs with random input data. Specifically, we predict the solution of the parametrized PDE at each collocation point using a previously assembled lower fidelity interpolant, and use this prediction to provide deterministic (linear/nonlinear) iterative solvers with initial approximations which continue to improve as the algorithm progresses through the levels of the interpolant. With nested collocation points, these coarse predictions can be assembled as a sub-step in the construction of the high-fidelity interpolant. As a concrete example, we develop our approach in the context of stochastic collocation approaches employing sparse tensor products of globally defined Lagrange polynomials on nested one-dimensional Clenshaw-Curtis abscissas, providing a rigorous computational complexity analysis of the resulting fully discrete sparse grid stochastic collocation approximation, with and without acceleration, and demonstrating the effectiveness of our proposed algorithm.

We begin in §5.1 by recalling the class of parameterized PDEs under consideration, as well as the construction of the fully discrete solution. In §5.2 we give our acceleration technique in the context of general SC methods for the approximation of both linear and nonlinear stochastic parameterized elliptic PDEs using iterative solvers. In §5.3, we provide a rigorous computational complexity analysis of our approach, in the specific context of the sparse grid SC approximations defined in §3.1. Finally, in §5.4 we provide several numerical examples, including both moderately large-dimensional linear and nonlinear parametrized PDEs, illustrating the theoretical results and the improved efficiency of this technique.

5.1 Fully-discrete collocation approximation

Recall the stochastic parameterized boundary value problem from (2.1), given in weak form by (2.5). The acceleration technique proposed in §5.2 and the sparse-grid SC method discussed in §5.3 will be based on spatial approximation of the solution given by (2.6).

For $L \in \mathbb{N}_+$, let $\mathcal{I}_L$ be a general interpolation operator that utilizes M_L collocation points, denoted $\mathcal{H}_L = \{\boldsymbol{y}_{L,j}\}_{j=1}^{M_L}$. Moreover, assume that we have a family of interpolation operators $\{\mathcal{I}_L\}_{L\in\mathbb{N}_+}$, which approximates the solution $u_h(x,\cdot)$ in the polynomial spaces $\mathcal{P}_1(\Gamma) \subset \ldots \subset \mathcal{P}_L(\Gamma) \subset \mathcal{P}_{L+1}(\Gamma) \subset \ldots \subset L^2_\varrho(\Gamma)$, of increasing fidelity, defined on sets of sample points $\mathcal{H}_L \subset \Gamma$. Assume further that the fully discrete solution $u_{h,L} \in V_h(D) \otimes \mathcal{P}_L(\Gamma)$ has Lagrange interpolating form

$$u_{h,L}(x,\boldsymbol{y}) := \mathcal{I}_L[u_h](x,\boldsymbol{y}) = \sum_{j=1}^{M_L} \left(\sum_{i=1}^{M_h} c_{L,j,i}\varphi_i(x) \right) \Psi_{L,j}(\boldsymbol{y}), \tag{5.1}$$

where $\{\Psi_{L,j}\}_{j=1}^{M_L}$ is a basis for $\mathcal{P}_L(\Gamma)$. The approximation (5.1) can be constructed by solving for $u_h(x,\boldsymbol{y}_{L,j})$ *independently* at each sample point $\boldsymbol{y}_{L,j} \in \mathcal{H}_L$. In §5.3, we construct a specific example of an interpolation scheme satisfying (5.1), namely global sparse grid collocation.

5.2 Accelerating stochastic collocation methods

We next introduce our acceleration scheme for both linear and nonlinear elliptic PDEs. For each $L \in \mathbb{N}_+$, the bulk of the computational cost in constructing (5.1) goes into solving the M_L systems of equations (2.7) corresponding to $\boldsymbol{y}_{L,j}$, $j = 1, \ldots, M_L$. In this chapter, we consider iterative solvers for the system in (2.7), and propose an acceleration scheme to reduce the total number of iterations necessary to solve the collection of systems over the set of sample parameters. We remark that here the word 'acceleration' does not indicate that the convergence properties of the iterative solver are improved, but rather that the overall computational work required by the SC method is reduced.

Denoting by $\widetilde{u}_h$ the output of the selected iterative solver for the system (2.7), for $\boldsymbol{y}_{L,j} \in \mathcal{H}_L$ the semi-discrete solution $u_h(x, \boldsymbol{y}_{L,j})$ is approximated by

$$u_h(x, \boldsymbol{y}_{L,j}) = \sum_{i=1}^{M_h} c_{L,j,i}\, \varphi_i(x) \approx \widetilde{u}_h(x, \boldsymbol{y}_{L,j}) = \sum_{i=1}^{M_h} \widetilde{c}_{L,j,i}\, \varphi_i(x),$$

where we define $\widetilde{\boldsymbol{c}}_{L,j} = (\widetilde{c}_{L,j,1}, \ldots, \widetilde{c}_{L,j,M_h})^\top$. Therefore the final SC approximation is given by a perturbation of (5.1), i.e.,

$$\widetilde{u}_{h,L}(x, \boldsymbol{y}) := \sum_{j=1}^{M_L} \left(\sum_{i=1}^{M_h} \widetilde{c}_{L,j,i}\, \varphi_i(x) \right) \Psi_{L,j}(\boldsymbol{y}). \tag{5.2}$$

To start the iterative solver for the system (2.7), it is common to use a zero initial guess, denoted by $\boldsymbol{c}_{L,j}^{(0)} = (0, \ldots, 0)^\top$. However, we can better predict the solution at level L using lower level approximations: Assume that we first obtain $\widetilde{u}_{h,L-1}(x, \boldsymbol{y})$ by collocating solutions to (2.7) over $\mathcal{H}_{L-1}$. Then at level L, for each new point $\boldsymbol{y}_{L,j} \in \mathcal{H}_L \setminus \mathcal{H}_{L-1}$, the initial guess $\boldsymbol{c}_{L,j}^{(0)}$ can be given by interpolating the solutions from level $L-1$, i.e.,

$$\boldsymbol{c}_{L,j}^{(0)} := \Big(\widetilde{u}_{h,L-1}(x_1, \boldsymbol{y}_{L,j}), \ldots, \widetilde{u}_{h,L-1}(x_{M_h}, \boldsymbol{y}_{L,j}) \Big)^\top = \sum_{j'=1}^{M_{L-1}} \widetilde{\boldsymbol{c}}_{L-1,j'} \Psi_{L-1,j'}(\boldsymbol{y}_{L,j}). \tag{5.3}$$

For a convergent interpolation scheme, we expect the necessary number of iterations to compute $\widetilde{\boldsymbol{c}}_{L,j}$ to become smaller as the level L increases to an overall maximum level, denoted

$L_{\max}$. As such, the construction of the desired solution $\widetilde{u}_{h,L_{\max}}$ is accelerated through the intermediate solutions $\{\widetilde{u}_{h,L}\}_{L=1}^{L_{\max}-1}$. This approach reduces computational cost by improving initial guesses, but does not depend on the specific solver used. Thus, our scheme may always be combined with faster solvers or better preconditioners. In Algorithm 1, we outline the acceleration procedure described above, using a general nonlinear iterative method for the solution of (2.7). The update function $\mathscr{S}$ in line 12 depends on the chosen iterative method, and is defined later for two specific examples.

Algorithm 1: *The accelerated SC algorithm*

Goal: Compute $\widetilde{u}_{h,L_{\max}}(x,\boldsymbol{y}) := \sum_{j=1}^{M_{L_{\max}}} \left(\sum_{i=1}^{M_h} \widetilde{c}_{L_{\max},j,i}\, \varphi_i(x) \right) \Psi_{L_{\max},j}(\boldsymbol{y})$

```
 1: Define $M_0 = 1$ and $\widetilde{\boldsymbol{c}}_{0,1} = (0,\ldots,0)^\top$
 2: for $L = 1,\ldots,L_{\max}$ do
 3:     for $\boldsymbol{y}_{L,j} \in \mathcal{H}_L \setminus \left( \bigcup_{l=1}^{L-1} \mathcal{H}_l \right)$ do
 4:         Compute the initial guess according to (5.3):
 5:         $\boldsymbol{c}_{L,j}^{(0)} = \sum_{j'=1}^{M_{L-1}} \widetilde{\boldsymbol{c}}_{L-1,j'}\, \Psi_{L-1,j'}(\boldsymbol{y}_{L,j})$
 6:         Initialize: $k = 1$
 7:         repeat
 8:             Compute residual $\boldsymbol{r}_{L,j}^{(k)} = (r_{L,j,1}^{(k)},\ldots,r_{L,j,M_h}^{(k)})^\top$:
 9:             for $i = 1,\ldots,M_h$ do
10:                 $r_{L,j,i}^{(k)} = \int_D f(\boldsymbol{y}_{L,j})\, \varphi_i \;-\; \sum_{\nu \in \Lambda_1 \cup \Lambda_2} S_\nu \left( \sum_{i'=1}^{M_h} c_{L,j,i'}^{(k)}\, \varphi_{i'}(x), \boldsymbol{y}_{L,j} \right) T_\nu(\varphi_i)\, dx$
11:             end for
12:             Update the solution: $\boldsymbol{c}_{L,j}^{(k+1)} = \boldsymbol{c}_{L,j}^{(k)} + \mathscr{S}(\boldsymbol{r}_{L,j}^{(1)},\ldots,\boldsymbol{r}_{L,j}^{(k)})$
13:             $k = k+1$
14:         until $\|\boldsymbol{c}_{L,j}^{(k)} - \boldsymbol{c}_{L,j}^{(k-1)}\| < \tau$
15:         $\widetilde{\boldsymbol{c}}_{L,j} = \boldsymbol{c}_{L,j}^{(k)}$
16:     end for
17: end for
```

The efficiency of the proposed algorithm depends crucially on the number of times the iterative solver is used, i.e., how many sample points are in the set $\Delta\mathcal{H}_L = \mathcal{H}_L \setminus \left(\bigcup_{l=1}^{L-1} \mathcal{H}_l \right)$ for each level L. In fact, if the sample points are not nested, it could be the case that $\Delta\mathcal{H}_L = \mathcal{H}_L$, and the algorithm may be very inefficient. Hence, in the following sections we will assume:

Assumption A8. *Assume that the point sets $\mathcal{H}_L, L = 1, \dots, L_{\max}$ are nested, i.e.,*

$$\mathcal{H}_1 \subset \mathcal{H}_2 \subset \dots \subset \mathcal{H}_{L_{\max}} \subset \Gamma.$$

Then $\Delta\mathcal{H}_L = \mathcal{H}_L \setminus \mathcal{H}_{L-1}$, and we can construct the intermediate solutions $\{\widetilde{u}_{h,L}\}_{L=1}^{L_{\max}-1}$ using a subset of the information needed to approximate $\widetilde{u}_{h,L_{\max}}$.

In §3.1 we constructed a specific interpolant using a point set which satisfies Assumption A8. Next, we give several examples using Algorithm 1, looking at iterative solvers for both nonlinear and linear elliptic PDEs.

Example 5.1. Consider the weak form of the nonlinear elliptic PDE in Example 2.2, letting $S_1(v; \boldsymbol{y}) = a(x, \boldsymbol{y})\nabla v$, $T_1(v) = \nabla v$, $S_2(v, \boldsymbol{y}) = v(x, \boldsymbol{y})|v(x, \boldsymbol{y})|^s$, and $T_2(v) = v$; this implies $\Lambda_1 = \{1\}$, $\Lambda_2 = \{2\}$. Define the matrix $\boldsymbol{A}_{L,j} = \boldsymbol{A}(\boldsymbol{y}_{L,j})$, $j = 1, \dots, M_L$ by

$$[\boldsymbol{A}_{L,j}]_{i,i'} = \int_D a(\boldsymbol{y}_{L,j})\nabla\varphi_{i'}\nabla\varphi_i \, dx, \quad \text{for } \; i, i' = 1, \dots, M_h. \tag{5.4}$$

Then using the fixed point iterative method in Algorithm 1, for the update step we define

$$\mathscr{S}(\boldsymbol{r}_{L,j}^{(1)}, \dots, \boldsymbol{r}_{L,j}^{(k)}) = \boldsymbol{A}_{L,j}^{-1}\boldsymbol{r}_{L,j}^{(k)}.$$

With $u_{h,L}^{(k)}(x, \boldsymbol{y}_{L,j}) = \sum_{i=1}^{M_h} c_{L,j,i}^{(k)} \varphi_i(x)$, this update is equivalent to solving the linear system

$$\int_D a(\boldsymbol{y}_{L,j})\nabla u_{h,L}^{(k+1)} \; \nabla v \; dx = \int_D \left[f(\boldsymbol{y}_{L,j}) - u_{h,L}^{(k)}(\boldsymbol{y}_{L,j})|u_{h,L}^{(k)}(\boldsymbol{y}_{L,j})|^s \right] v \, dx \;\; \forall v \in V_h(D),$$

to update $u_h^{(k)}$ to $u_h^{(k+1)}$ at the $(k+1)$-th iteration. Note that each iteration of the solver in Algorithm 1 requires the solution of this system, which is not aided by our algorithm.

Example 5.2. As a special case of the example above, consider the weak form of the linear elliptic problem in Example 2.1 with $\Lambda_1 = \{1\}$, $\Lambda_2 = \emptyset$, $S_1(v; \boldsymbol{y}) = a\nabla v$ and $T_1(v) = \nabla v$ in (2.7). Due to the linearity, at each collocation point the solution $u_h(x, \boldsymbol{y}_{L,j}) = \sum_{i=1}^{M_h} c_{L,j,i}\varphi_i(x)$ can be approximated by solving the following linear system

$$\boldsymbol{A}_{L,j}\boldsymbol{c}_{L,j} = \boldsymbol{f}_{L,j}, \tag{5.5}$$

with $\boldsymbol{A}_{L,j} = \boldsymbol{A}(\boldsymbol{y}_{L,j})$, $j = 1, \ldots, M_L$ as in (5.4), and $(\boldsymbol{f}_{L,j})_i = \int_D f(x, \boldsymbol{y}_{L,j})\varphi_i(x)dx$ for $i = 1, \ldots, M_h$. Under our assumptions on the coefficient a, the linear system (5.5) is symmetric positive definite, and we use the CG method [81] to find its solution. For $k \in \mathbb{N}^+$, by defining

$$\boldsymbol{p}_{L,j}^{(k)} = \boldsymbol{r}_{L,j}^{(k)} - \sum_{k'<k} \frac{\boldsymbol{p}_{L,j}^{(k')\top} \boldsymbol{A}_{L,j} \boldsymbol{r}_{L,j}^{(k)}}{\boldsymbol{p}_{L,j}^{(k')\top} \boldsymbol{A}_{L,j} \boldsymbol{p}_{L,j}^{(k')}} \boldsymbol{p}_{L,j}^{(k')},$$

we get the update function

$$\mathscr{S}(\boldsymbol{r}_{L,j}^{(1)}, \ldots, \boldsymbol{r}_{L,j}^{(k)}) = \frac{\boldsymbol{p}_{L,j}^{(k)\top} \boldsymbol{r}_{L,j}^{(k)}}{\boldsymbol{p}_{L,j}^{(k)\top} \boldsymbol{A}_{L,j} \boldsymbol{p}_{L,j}^{(k)}} \boldsymbol{p}_{L,j}^{(k)}.$$

Recall the following well-known error estimate for CG:

$$\left\| \boldsymbol{c}_{L,j} - \boldsymbol{c}_{L,j}^{(k)} \right\|_{\boldsymbol{A}_{L,j}} \leq 2 \left(\frac{\sqrt{\kappa_{L,j}} - 1}{\sqrt{\kappa_{L,j}} + 1} \right)^k \left\| \boldsymbol{c}_{L,j} - \boldsymbol{c}_{L,j}^{(0)} \right\|_{\boldsymbol{A}_{L,j}}, \tag{5.6}$$

where $\kappa_{L,j} = \kappa(\boldsymbol{y}_{L,j})$ denotes the condition number of $\boldsymbol{A}_{L,j}$, $\boldsymbol{c}_{L,j}^{(0)}$ is the vector of initial guess and $\boldsymbol{c}_{L,j}^{(k)}$ is the output of the k-th iteration of the CG solver. As opposed to Example 5.1, for this example Algorithm 1 provides initial guesses for the solution of the linear system (5.5).

To evaluate the efficiency of the accelerated SC method, we define cost metrics for standard and accelerated SC approximations. In general, the computational cost in floating point operations (flops) is the combined total number of iterations to solve (2.7) for each of the $M_{L_{\max}}$ sample points—denoted by K_{zero} and K_{acc} for the standard and accelerated SC methods, respectively—multiplied by the cost of performing one iteration, denoted $\mathcal{C}_{\text{iter}}$. Let $\mathcal{C}_{\text{int}}$ be the additional cost of interpolation incurred by using the accelerated initial vectors (5.3). Then we define the respective cost metrics for the two different cases.

$$\text{Standard SC cost:} \quad \mathcal{C}_{\text{zero}} = \mathcal{C}_{\text{iter}}\, K_{\text{zero}}, \tag{5.7}$$

$$\text{Accelerated SC cost:} \quad \mathcal{C}_{\text{acc}} = \mathcal{C}_{\text{iter}}\, K_{\text{acc}} + \mathcal{C}_{\text{int}}. \tag{5.8}$$

In Example 5.2, the discretization of the linear PDE leads to M_L sparse systems of equations of size $M_h \times M_h$. When solving these systems with a CG solver, K_{zero} and K_{acc}

are the sum of solver iterations contributed from each sample system. In this case, the cost of one iteration is just the cost of one matrix vector product, i.e., $\mathcal{C}_{\text{iter}} = C_D M_h$, where C_D depends on the domain D and the type of finite element basis.

Remark 5.3. (Interpolation costs). *Many adaptive interpolation schemes already require evaluation of the intermediate interpolation operators as in (5.3), e.g., to compute residual error estimators. Thus, these methods will incur the interpolation cost $\mathcal{C}_{\text{int}}$ even in the zero initial vector case. Furthermore, for most nonlinear problems the deterministic solver is expensive, so reducing the number of iterations is the most important element in reducing the cost. In each of these settings, we can define the cost metrics as simply K_{zero} and K_{acc}.*

Remark 5.4. (Hierarchical preconditioner construction). *When solving linear systems using iterative methods, convergence properties can be improved by considering the condition number of the system. As with initial vectors, an interpolation algorithm can be used to construct good, cheap preconditioners. We consider preconditioner algorithms where an explicit preconditioner matrix, or its inverse, is constructed. In this case, for some low collocation level L_{PC}, we construct a strong preconditioner, $P_{L_{\text{PC}},j} := P(\boldsymbol{y}_{L_{\text{PC}},j})$, for each individual iterative solver, $j = 1, \ldots, M_{L_{\text{PC}}}$. Then, these lower level preconditioners are interpolated for the subsequent levels. More specifically, for $L > L_{\text{PC}}$, and $\boldsymbol{y}_{L,j} \in \mathcal{H}_L \setminus \mathcal{H}_{L_{\text{PC}}}$, we use the preconditioner*

$$\widetilde{P}_{L,j} := \widetilde{P}(\boldsymbol{y}_{L,j}) = \sum_{j'=1}^{M_{L_{\text{PC}}}} P_{L_{\text{PC}},j'} \; \Psi_{L_{\text{PC}},j'}(\boldsymbol{y}_{L,j}). \tag{5.9}$$

Numerical illustrations of this approach are given in §5.4.

5.3 Applications to sparse grid stochastic collocation

In this section, we provide a specific example of an interpolation scheme satisfying the assumptions described in §5.2, a generalized sparse grid SC approach. We briefly review the construction of sparse grid interpolants, and rigorously analyze the approximation errors and the complexities of both the standard and accelerated SC approaches, in order to demonstrate the improved efficiency of the proposed acceleration technique when applied to iterative linear solvers. Note that the analysis in §5.3.1 and §5.3.2 are conducted in the setting of using Clenshaw-Curtis sparse grid, thus we assume the independence of all the random variables $\{y_n, n = 1, \dots, N\}$ in this section.

In what follows, we use the sparse grid operators described in §3.1. Specifically, we define the operator $\mathcal{I}_L := \mathcal{A}_L^{m,g}$, where we make the specific choices

$$p(1) = 1,\, p(l) = 2^{l-1} + 1 \text{ for } \ l > 1, \ \text{ and } \ g(\mathbf{l}) = \sum_{n=1}^{N} (l_n - 1). \tag{5.10}$$

For the remainder of the chapter, we will also assume that $\mathcal{I}_L$ uses the Clenshaw-Curtis sparse grid based on (3.4). Our analysis does not depend strongly on this choice of p and g, and we could use other functions, e.g., anisotropic approximations. With p, g fixed, we then write $\mathcal{H}_L = \mathcal{H}_L^{p,g}$.

Finally, to construct the fully-discrete approximation in the space $V_h(D) \otimes \mathcal{P}_{\Lambda_L^{p,g}}(\Gamma)$ we apply the Lagrange interpolating form of operator $\mathcal{I}_L[\cdot]$, given by (3.6), to $u_h(x, \boldsymbol{y})$ in (2.6) to obtain:

$$u_{h,L}(x, \boldsymbol{y}) = \mathcal{I}_L[u_h](x, \boldsymbol{y}) = \sum_{j=1}^{M_L} \left(\sum_{i=1}^{M_h} c_{L,j,i} \varphi_i(x) \right) \Psi_{L,j}(\boldsymbol{y}). \tag{5.11}$$

Due to the delta property of the basis function $\Psi_{L,j}(\boldsymbol{y})$, the interpolation matrix for $\mathcal{I}_L[u_h]$ is a diagonal matrix, and thus the coefficient vectors $\boldsymbol{c}_{L,j} = (c_{L,j,1}, \dots, c_{L,j,M_h})$ for $j = 1, \dots, M_L$ can be computed by *independently* solving M_L systems of type (2.7).

5.3.1 Error estimates for fixed L

In what follows, we focus on the linear elliptic problem described in Examples 2.1 and 5.2, and present a detailed convergence and complexity analysis of a fully discrete SC approximation, denoted $\widetilde{u}_{h,L}$, for any fixed level, $1 \leq L \leq L_{\max}$. This analysis provides the basis for analyzing the computational complexity of our acceleration method constructed over the sequence of levels $1 \leq L \leq L_{\max}$. As specified above we consider only the isotropic Smolyak version of SC interpolant given by (3.3), defined on Clenshaw-Curtis abscissas. However, our analysis can be extended without any essential difficulty to anisotropic SC methods and some more complicated underlying PDEs.

The differential operator corresponding to the parameterized elliptic PDE (2.2) admits a weak form that is a symmetric, uniformly coercive and continuous bilinear operator on $H_0^1(D)$; i.e., there exist $\alpha, \beta > 0$, depending on $a_{\min}$ and $a_{\max}$ but independent of $\boldsymbol{y}$, such that for every $v, w \in H_0^1(D)$,

$$\left| \int_D a(\boldsymbol{y}) \nabla v \, \nabla w \, dx \right| \leq \alpha \, \|v\|_{H_0^1(D)} \, \|w\|_{H_0^1(D)} \quad \text{and} \quad \beta \, \|v\|_{H_0^1(D)}^2 \leq \int_D a(\boldsymbol{y}) |\nabla v|^2 \, dx.$$

In this case, the bilinear form induces a norm, $\|v\|^2 = \int_D a(\boldsymbol{y}) |\nabla v|^2 \, dx$, which for functions $v(x) = \sum_{i=1}^{M_h} c_i \phi_i(x) \in V_h(D)$, with $\boldsymbol{c} = (c_1, \ldots, c_{M_h})$, coincides with the discrete norm $\|\boldsymbol{c}\|_{\boldsymbol{A}(\boldsymbol{y})}$, where the matrix $A(\boldsymbol{y})$ is defined in (5.4). Thus we have

$$\text{Continuity:} \quad \|\boldsymbol{c}\|_{\boldsymbol{A}(\boldsymbol{y})} = \|v\| \leq \sqrt{\alpha} \, \|v\|_{H_0^1(D)}, \text{ and,} \tag{5.12a}$$

$$\text{Ellipticity:} \quad \sqrt{\beta} \, \|v\|_{H_0^1(D)} \leq \|v\| = \|\boldsymbol{c}\|_{\boldsymbol{A}(\boldsymbol{y})}. \tag{5.12b}$$

In order to investigate the complexity of $\widetilde{u}_{h.L}, L \in \mathbb{N}_+$, we first need to derive sufficient conditions for the error $\|u - \widetilde{u}_{h,L}\|_{L_\varrho^2}$ to achieve a tolerance of $\varepsilon > 0$, where $L_\varrho^2 := L_\varrho^2(\Gamma; H_0^1(D))$. Using the triangle inequality, the total error can be split into three parts, i.e.,

$$\|u - \widetilde{u}_{h,L}\|_{L_\varrho^2} \leq \underbrace{\|u - u_h\|}_{e_1}{}_{L_\varrho^2} + \underbrace{\|u_h - u_{h,L}\|}_{e_2}{}_{L_\varrho^2} + \underbrace{\|u_{h,L} - \widetilde{u}_{h,L}\|}_{e_3}{}_{L_\varrho^2}. \tag{5.13}$$

The contributions of e_1 and e_2 correspond to the FE and SC errors, respectively, and have been previously examined [71]. The error e_3 contributed by the linear solver is often omitted from the analysis in the literature, and in practice can be controlled by setting a tight tolerance on the iterative solver. However, the analysis presented here is focused on providing cost estimates for the iterative solver and requires careful consideration of this term. First, we recall error estimates for e_1 and e_2, given from [71].

Lemma 5.4.1. *Let $\mathcal{T}_h$ be a uniform finite element mesh over $D \subset \mathbb{R}^d, d = 1, 2, 3$, with $M_h = \mathcal{O}(1/h^d)$ grid points. For the elliptic PDE in Example 2.1, when $u(x, \boldsymbol{y}) \in L_\varrho^2(\Gamma; H_0^1(D) \cap$*

$H^{s+1}(D)), s \in \mathbb{N}_+$, *the error of the finite element approximation* u_h *is bounded by*

$$\|u - u_h\|_{L^2_\varrho} \leq C_{\text{fem}}\, h^s, \tag{5.14}$$

where the constant C_{fem} *is independent of* h *and* $\boldsymbol{y}$*.*

For e_2, recall the error estimate of Theorem 3.4.1, which states a convergence rate in terms of the level $L \in \mathbb{N}^+$:

$$\|u - \mathcal{I}_L[u]\|_{L^\infty(\Gamma;H^1_0(D))} \leq C_{sc}\mathrm{e}^{-rN2^{L/N}}, \tag{5.15}$$

where, for a constant $0 < \delta < 1$, the rate $r = (1-\delta)\min_{1\leq n\leq N}\log\rho_n$, and the constant $C_{sc} > 0$ depends on N, u, and δ.

Note that we have already assumed $\Gamma = [-1,1]^N$ with the uniform measure, so the essential supremum above is taken with respect to Lebesgue measure. We remark also that the projection of u into the finite element subspace, denoted u_h, also satisfies Assumption A2 with the same region of analyticity, and therefore the application of the interpolant, $\mathcal{I}_L$, to the semidiscete solution u_h will converge as in (5.15).

We now consider the global solver error e_3 in (5.13), which is the error incurred by approximating the solution to (5.5) at each sample point. The difference $u_{h,L} - \widetilde{u}_{h,L}$ can be written as an interpolant of the solver error, i.e., $u_{h,L} - \widetilde{u}_{h,L} = \mathcal{I}_L[u_h - \widetilde{u}_h]$, which represents the solver error amplified by the interpolation operator. Define the Lebesgue constant of the operator $\mathcal{I}_L[\cdot]$ by $C_L = \max_{\boldsymbol{y}\in\Gamma}\sum_{j=1}^{M_L}|\Psi_{L,j}(\boldsymbol{y})|$ where $\Psi_{L,j}$ is given in (3.6). For $\mathcal{I}_L[\cdot]$ in (3.6), we have

$$\|u_{h,L} - \widetilde{u}_{h,L}\|_{L^\infty(\Gamma;H^1_0(D))} \leq C_L \max_{j=1,\ldots,M_L} \|u_h(\boldsymbol{y}_{L,j}) - \widetilde{u}_h(\boldsymbol{y}_{L,j})\|_{H^1_0(D)}\,.$$

Thus, from the ellipticity condition in (5.12b),

$$\begin{aligned}
e_3 &\leq C_L \max_{j=1,\ldots,M_L} \|u_h(\boldsymbol{y}_{L,j}) - \widetilde{u}_h(\boldsymbol{y}_{L,j})\|_{H_0^1(D)} \\
&\leq C_L \frac{1}{\sqrt{\beta}} \max_{j=1,\ldots,M_L} \|\boldsymbol{c}_{L,j} - \widetilde{\boldsymbol{c}}_{L,j}\|_{\boldsymbol{A}(\boldsymbol{y}_{L,j})} \\
&\leq \frac{\tau}{\sqrt{\beta}} C_L,
\end{aligned}$$

where τ is defined to be the tolerance of the linear solver. Note that the expression $u_h - \widetilde{u}_h$ is only defined at collocation points. The solver error for each fixed $\boldsymbol{y}_{L,j} \in \mathcal{H}_L$ is controlled by the CG convergence estimate (5.6). We now provide an upper bound of the Lebesgue constant C_L in the following lemma.

Lemma 5.4.2. *The Lebesgue constant for the isotropic sparse-grid interpolation operator* $\mathcal{I}_L[\cdot]$ (3.6)*, using the Clenshaw-Curtis rule on* $\Gamma = \prod_{n=1}^{N} \Gamma_n = [-1,1]^N$ *is bounded by*

$$C_L \leq [(L+1)(L+2)]^N, \tag{5.16}$$

where L*,* N *are the interpolation level and dimension of the parameter space, respectively.*

Proof. For $n = 1,\ldots,N$, define λ_{l_n} to be the Lebesgue constant of the one-dimensional operator $\mathscr{U}^{p(l_n)}$. For Lagrange interpolants based on Clenshaw-Curtis abscissas, we have that $\lambda_{l_n} \leq \frac{2}{\pi}\log\left(p\left(l_n\right)-1\right)+1$ for $l_n \geq 2$ [28]. Combining this with the growth rate m given by (5.10), it is easy to obtain that $\lambda_{l_n} \leq 2l_n - 1$ for $l_n \geq 2$.

For $v \in C^0(\Gamma_n)$, the difference operator $\Delta^{p(l_n)}$ for $l_n = 1$ satisfies

$$\left\|\Delta^{p(1)}[v]\right\|_{L^\infty(\Gamma_n)} = \left\|\mathscr{U}^{p(1)}[v]\right\|_{L^\infty(\Gamma_n)} \leq \lambda_1 \max_{y_n \in \vartheta^1} |v(y_n)|.$$

For $l_n \geq 2$, the triangle inequality yields

$$\begin{aligned}
\left\|\Delta^{p(l_n)}[v]\right\|_{L^\infty(\Gamma_n)} &= \left\|\mathscr{U}^{p(l_n)}[v] - \mathscr{U}^{p(l_n-1)}[v]\right\|_{L^\infty(\Gamma_n)} \\
&\leq (\lambda_{l_n} + \lambda_{l_n-1}) \max_{y_n \in \vartheta^{l_n}} |v(y_n)|.
\end{aligned}$$

Finally, for $v \in C^0(\Gamma)$, we bound the norm of the interpolant $\mathcal{I}_L[v]$ by

$$\begin{aligned}
\|\mathcal{I}_L[v]\|_{L^\infty(\Gamma)} &= \left\| \sum_{g(\mathbf{l})\leq L} \Delta^{p(l_1)} \otimes \cdots \otimes \Delta^{p(l_N)}[v] \right\|_{L^\infty(\Gamma)} \\
&\leq \left(2^N \sum_{g(\mathbf{l})\leq L} \prod_{n=1}^{N} l_n \right) \max_{j=1,\ldots,M_L} |v(\boldsymbol{y}_{L,j})| \\
&\leq 2^N \left(\sum_{l=1}^{L+1} l \right)^N \max_{j=1,\ldots,M_L} |v(\boldsymbol{y}_{L,j})| \\
&= [(L+1)(L+2)]^N \max_{j=1,\ldots,M_L} |v(\boldsymbol{y}_{L,j})|,
\end{aligned}$$

which gives the desired estimate. □

5.3.2 Complexity analysis

Now we analyze the cost of constructing $\widetilde{u}_{h,L_{\max}}$, $L_{\max} \in \mathbb{N}_+$, with the prescribed accuracy ε. Here we assume $\varepsilon > 0$ is sufficiently small, and study the asymptotic growth of the total costs (5.8) for the construction of $\widetilde{u}_{h,L_{\max}}$ by the accelerated algorithm described in §5.2. For comparison, we will also analyze the cost (5.7) associated with the standard SC method, where iterative solvers for the sequence of solutions to the linear systems (5.5) are seeded with the zero vector as an initial guess. According to the error estimates discussed in §5.3.1, a sufficient condition to ensure $\|u - \widetilde{u}_{h,L_{\max}}\|_{L^2_\varrho} \leq \varepsilon$ is that

$$\|e_1\|_{L^2_\varrho} \leq C_{\text{fem}} h^s \leq \frac{\varepsilon}{3}, \tag{5.17a}$$

$$\|e_2\|_{L^2_\varrho} \leq \|e_2\|_{L^\infty_\varrho} \leq C_{\text{sc}} \, \mathrm{e}^{-rN2^{L_{\max}/N}} \leq \frac{\varepsilon}{3}, \tag{5.17b}$$

$$\|e_3\|_{L^2_\varrho} \leq \|e_3\|_{L^\infty_\varrho} \leq (L_{\max}+2)^{2N} \frac{\tau}{\sqrt{\beta}} \leq \frac{\varepsilon}{3}. \tag{5.17c}$$

In §5.2 we defined K_{zero} and K_{acc} as the total number of solver iterations used by the standard and accelerated SC methods, respectively, to solve (5.5) at each sample point. Now let $K_{\text{zero}}(\varepsilon)$ and $K_{\text{acc}}(\varepsilon)$ represent the minimum values of K_{zero} and K_{acc}, respectively, needed to satisfy the inequalities (5.17). Here we aim to estimate upper bounds of $K_{\text{zero}}(\varepsilon)$

and $K_{\mathrm{acc}}(\varepsilon)$. Note that, for fixed dimension N, level $L_{\max}$, and mesh size h, the total number of iterations is determined by the inequality (5.17c). Thus, the estimation of $K_{\mathrm{zero}}(\varepsilon)$ and $K_{\mathrm{acc}}(\varepsilon)$ has two steps: (i) Given N and ε, estimate the maximum possible h to satisfy (5.17a) and the minimum $L_{\max}$ that achieves (5.17b); (ii) Substitute the obtained values into (5.17c) to estimate upper bounds on $K_{\mathrm{zero}}(\varepsilon)$ and $K_{\mathrm{acc}}(\varepsilon)$ according to the CG error estimate (5.6). For (i), we have the following lemma, that follows immediately from Lemmas 5.4.1 and 3.4.1.

Lemma 5.4.3. *Given the assumptions of Lemmas 5.4.1 and 3.4.1, the error bounds* (5.17a) *and* (5.17b) *can be achieved by choosing the mesh size h and the level $L_{\max}$ according to*

$$h(\varepsilon) = \left(\frac{\varepsilon}{3C_{\mathrm{fem}}}\right)^{1/s} \quad \textit{and} \quad L_{\max}(\varepsilon) = \left\lceil \frac{N}{\log 2} \log\left(\frac{1}{rN}\log\left(\frac{3C_{\mathrm{sc}}}{\varepsilon}\right)\right)\right\rceil. \tag{5.18}$$

For convenience, we treat the integer quantities $K_{\mathrm{zero}}(\varepsilon)$, $K_{\mathrm{acc}}(\varepsilon)$, and $L_{\max}(\varepsilon)$ as positive real numbers in the rest of this section. Now, based on the estimate in Lemma 5.4.2 for the Lebesgue constant $C_{L_{\max}}$, we state the following lemma related to the choice of an appropriate tolerance $\tau(\varepsilon)$ to satisfy the error bounds (5.17c).

Lemma 5.4.4. *Let $\varepsilon > 0$. Given the assumptions of Lemmas 5.4.1 and 3.4.1, a sufficient condition to ensure $e_3 < \varepsilon/3$ is that*

$$\tau(\varepsilon) = \frac{\sqrt{\beta}\,\varepsilon}{3(L_{\max}(\varepsilon)+2)^{2N}}. \tag{5.19}$$

Moreover, it holds

$$\frac{1}{\sqrt{\beta}}(L+2)^{2N}\tau(\varepsilon) \leq C_{\mathrm{sc}}\,\mathrm{e}^{-rN2^{L/N}} \quad \textit{for} \;\; L = 0,\dots,L_{\max}(\varepsilon)-1,$$

where $L_{\max}(\varepsilon)$ is the minimum level given in (5.18).

Proof. (5.19) is an immediate result of (5.17c). For $L = 0,\dots,L_{\max}(\varepsilon)-1$, we have

$$\frac{1}{\sqrt{\beta}}(L+2)^{2N}\tau(\varepsilon) \leq \frac{1}{\sqrt{\beta}}(L_{\max}(\varepsilon)+2)^{2N}\tau(\varepsilon) \leq \frac{\varepsilon}{3} \leq C_{\mathrm{sg}}\,\mathrm{e}^{-rN2^{(L_{\max}(\varepsilon)-1)/N}} \leq C_{\mathrm{sg}}\,\mathrm{e}^{-rN2^{L/N}},$$

which completes the proof. □

Using the selected $h := h(\varepsilon)$, $L_{\max} := L_{\max}(\varepsilon)$, and $\tau := \tau(\varepsilon)$, we now estimate the upper bounds on the number of CG iterations needed to solve a linear system at a point $\boldsymbol{y}_{L_{\max},j} \in \mathcal{H}_{L_{\max}}$. To proceed, define

$$k_{\text{zero}} := \max_{\boldsymbol{y}_{L_{\max},j} \in \mathcal{H}_{L_{\max}}} k_{L_{\max},j}, \quad \text{and} \quad k_{\text{acc}}^{L} := \max_{\boldsymbol{y}_{L,j} \in \Delta\mathcal{H}_L} k_{L,j} \quad \text{for} \quad L = 1, \dots, L_{\max},$$

where $k_{L,j}$ is the number of CG iterations required to achieve $\|\boldsymbol{c}_{L,j} - \boldsymbol{c}_{L,j}^{(k_{L,j})}\|_{\boldsymbol{A}_{L,j}} \leq \tau(\varepsilon)$, which, in general, depends on the choice of initial vector. Note that, in the case $\boldsymbol{c}_{L,j}^{(0)} = (0, \dots, 0)^\top$, there is no improvement in the iteration count as the level L increases, so k_{zero} does not depend on L. Now we give the following estimates on k_{zero} and $\{k_{\text{acc}}^{L}\}_{L=1}^{L_{\max}}$.

Lemma 5.4.5. *Under the conditions of Lemmas 5.4.1 and 3.4.1, for any $\boldsymbol{y}_{L_{\max},j} \in \mathcal{H}_{L_{\max}}$, if the CG method with zero initial vector is used to solve (5.5) to tolerance $\tau > 0$, then k_{zero} can be bounded by*

$$k_{\text{zero}} \leq \log\left(\frac{2\sqrt{\alpha}\, \|u_h\|_{L^\infty(\Gamma;H_0^1(D))}}{\tau}\right) \Big/ \log\left(\frac{\sqrt{\overline{\kappa}}+1}{\sqrt{\overline{\kappa}}-1}\right). \tag{5.20}$$

Here $\overline{\kappa} = \sup_{\boldsymbol{y}\in\Gamma} \kappa(\boldsymbol{y})$, with $\kappa(\boldsymbol{y})$ the condition number of the matrix $\boldsymbol{A}(\boldsymbol{y})$ corresponding to (2.7). Alternatively, if the initial vector is given by the acceleration method as in (5.3), then, for $L = 1, \dots, L_{\max}$, k_{acc}^{L} can be bounded by

$$k_{\text{acc}}^{L} \leq \log\left(\frac{4\sqrt{\alpha}C_{\text{sc}}\, \mathrm{e}^{-rN2^{(L-1)/N}}}{\tau}\right) \Big/ \log\left(\frac{\sqrt{\overline{\kappa}}+1}{\sqrt{\overline{\kappa}}-1}\right). \tag{5.21}$$

Proof. Let $\boldsymbol{y}_{L,j}$ be an arbitrary point in $\mathcal{H}_L, 1 \leq L \leq L_{\max}$. Given an initial guess $\boldsymbol{c}_{L,j}^{(0)}$, the minimum number of CG iterations needed to achieve tolerance $\tau > 0$ can be obtained immediately from (5.6), that is,

$$k_{L,j} = \left\lceil \log\left(\frac{2\|\boldsymbol{c}_{L,j} - \boldsymbol{c}_{L,j}^{(0)}\|_{\boldsymbol{A}_{L,j}}}{\tau}\right) \Big/ \log\left(\frac{\sqrt{\kappa_{L,j}}+1}{\sqrt{\kappa_{L,j}}-1}\right) \right\rceil, \tag{5.22}$$

where $\boldsymbol{A}_{L,j} = \boldsymbol{A}(\boldsymbol{y}_{L,j})$ is the FE system matrix corresponding to parameter $\boldsymbol{y}_{L,j}$, and $\kappa_{L,j} = \kappa(\boldsymbol{y}_{L,j})$ is the condition number of $\boldsymbol{A}_{L,j}$ (See Example 5.2). In the case that $\boldsymbol{c}_{L,j}^{(0)} = (0, \dots, 0)^\top$, the estimate in (5.20) can be obtained from (5.12a), i.e.,

$$\left\|\boldsymbol{c}_{L,j}-\boldsymbol{c}_{L,j}^{(0)}\right\|_{\boldsymbol{A}_{L,j}}=\|\boldsymbol{c}_{L,j}\|_{\boldsymbol{A}_{L,j}}\leq\sqrt{\alpha}\,\|u_h\|_{L^\infty(\Gamma;H_0^1(D))}\,.$$

Alternatively, when using $\widetilde{u}_{h,L-1}$ for $L=1,\dots L_{\max}$ to provide initial vectors for the CG solver (based on (5.3)), for $\boldsymbol{y}_{L,j}\in\Delta\mathcal{H}_L$ we use Lemma 5.4.4 and (5.12a) to get the following estimate:

$$\begin{aligned}\left\|\boldsymbol{c}_{L,j}-\boldsymbol{c}_{L,j}^{(0)}\right\|_{\boldsymbol{A}_{L,j}}&\leq\sqrt{\alpha}\left(\|u_h-u_{h,L-1}\|_{L^\infty(\Gamma;H_0^1(D))}+\|u_{h,L-1}-\widetilde{u}_{h,L-1}\|_{L^\infty(\Gamma;H_0^1(D))}\right)\\&\leq\sqrt{\alpha}\bigg(C_{\mathrm{sc}}\,\mathrm{e}^{-rN2^{(L-1)/N}}+\frac{1}{\sqrt{\beta}}(L+1)^{2N}\tau\bigg)\\&\leq 2\sqrt{\alpha}C_{\mathrm{sc}}\,\mathrm{e}^{-rN2^{(L-1)/N}}.\end{aligned}\tag{5.23}$$

With (5.22), this leads directly to the estimate in (5.21). □

Remark 5.5. (Acceleration over sparse grid levels). *We can combine* (5.23) *with the CG error estimate* (5.6) *to see that*

$$\left\|\boldsymbol{c}_{L,j}-\boldsymbol{c}_{L,j}^{(k)}\right\|_{\boldsymbol{A}_{L,j}}\leq 4\sqrt{\alpha}\,C_{sc}\left(\frac{\sqrt{\kappa_{L,j}}-1}{\sqrt{\kappa_{L,j}}+1}\right)^k\mathrm{e}^{-rN2^{(L-1)/N}}.$$

From this, we clearly see that the the necessary number of iterations needed to reach a given tolerance is not fixed, but rather continues to improve as the algorithm moves through the levels L of the SC interpolant. This improvement is also not affected by reducing the size of the spatial mesh. Furthermore, we see that our algorithm does not preclude preconditioning, which improves the convergence rate of the solver by reducing the condition number $\kappa_{L,j}$.

In the accelerated case, the sparse-grid interpolant $\mathcal{I}_{L_{\max}}[u_h]$ must be constructed in the following fashion: before solving the system (5.5) corresponding to a sample point $\boldsymbol{y}_{L,j}\in\Delta\mathcal{H}_L$, we must first solve the systems for all sample points in $\mathcal{H}_{L-1}$. With a total number $\Delta M_L=\#(\Delta\mathcal{H}_L)$ of new linear systems at level L, the total number of CG iterations for the newly added points at level L can be bounded by $\Delta M_L k_{\mathrm{zero}}$ and $\Delta M_L k_{\mathrm{acc}}^L$, for the standard and the accelerated cases, respectively. Then since $M_{L_{\max}}=\sum_{L=1}^{L_{\max}}\Delta M_L$, we find that the total number of iterations for the standard and accelerated schemes can be bounded as

$$K_{\text{zero}}(\varepsilon) \leq M_{L_{\max}}\, k_{\text{zero}}, \quad \text{and} \quad K_{\text{acc}}(\varepsilon) \leq \sum_{L=1}^{L_{\max}} \Delta M_L\, k_{\text{acc}}^L.$$

This leads to the following estimates.

Theorem 5.6. *Given Assumption A2, and the conditions of Lemmas 5.4.1 and 3.4.1, for $\varepsilon > 0$, the minimum total number of CG iterations $K_{\text{zero}}(\varepsilon)$ to achieve $\|u - \widetilde{u}_{h,L_{\max}}\|_{L^2_\varrho} < \varepsilon$, using zero initial vectors is bounded by*

$$\begin{aligned} K_{\text{zero}}(\varepsilon) \leq C_1 &\left[\log\left(\frac{3C_{\text{sc}}}{\varepsilon}\right)\right]^N \left[C_2 + \frac{1}{\log 2}\log\log\left(\frac{3C_{\text{sc}}}{\varepsilon}\right)\right]^{N-1} \\ &\times \frac{1}{\log\left(\frac{\sqrt{\overline{\kappa}}+1}{\sqrt{\overline{\kappa}}-1}\right)}\left\{\log\left(\frac{C_3}{\varepsilon}\right) + C_4 + 2N\log\log\left[\frac{1}{rN}\log\left(\frac{3C_{\text{sc}}}{\varepsilon}\right)\right]\right\}, \end{aligned} \tag{5.24}$$

where $\overline{\kappa}$ is as defined in Lemma 5.4.5, and the constants C_1, C_2, C_3 and C_4 are defined by

$$\begin{aligned} C_1 &= \left(\frac{\mathrm{e}}{\log 2}\right)^{N-1}\left(\frac{2}{rN}\right)^N, \quad C_2 = 1 + \frac{1}{\log 2}\log\left(\frac{1}{rN}\right), \\ C_3 &= 6\sqrt{\frac{\alpha}{\beta}}\,\|u_h\|_{L^\infty(\Gamma;H_0^1(D))}, \quad C_4 = 2N\log\left(\frac{2N}{\log 2}\right). \end{aligned} \tag{5.25}$$

Proof. To achieve the prescribed error, we balance the three error sources that contribute to the total error (5.13). To control e_1 and e_2, set $h = h(\varepsilon)$ and $L_{\max} = L_{\max}(\varepsilon)$ according to Lemma 5.4.3. For the solver error e_3, we choose the solver tolerance $\tau = \tau(\varepsilon)$ according to Lemma 5.4.4. As above, the total number of iterations $K_{\text{zero}}(\varepsilon)$ can be bounded by

$$K_{\text{zero}}(\varepsilon) \leq M_{L_{\max}}\, k_{\text{zero}}. \tag{5.26}$$

From Lemma 5.4.4 and 5.4.5, we have

$$
\begin{aligned}
k_{\mathrm{zero}} &\leq \log\left(\frac{2\sqrt{\alpha}\,\|u_h\|_{L^\infty(\Gamma;H_0^1(D))}}{\tau}\right) \Big/ \log\left(\frac{\sqrt{\overline{\kappa}}+1}{\sqrt{\overline{\kappa}}-1}\right) \\
&\leq \log\left(\frac{6\sqrt{\alpha}\,\|u_h\|_{L^\infty(\Gamma;H_0^1(D))}\,(L_{\max}+2)^{2N}}{\sqrt{\beta}\varepsilon}\right) \Big/ \log\left(\frac{\sqrt{\overline{\kappa}}+1}{\sqrt{\overline{\kappa}}-1}\right) \qquad (5.27) \\
&\leq \left\{\log\left(\frac{C_3}{\varepsilon}\right) + C_4 + 2N\log\log\left(\frac{1}{rN}\log\left(\frac{3C_{\mathrm{sc}}}{\varepsilon}\right)\right)\right\} \Big/ \log\left(\frac{\sqrt{\overline{\kappa}}+1}{\sqrt{\overline{\kappa}}-1}\right).
\end{aligned}
$$

In addition, following [71, Lemma 3.9], we bound the number of interpolation points:

$$
\begin{aligned}
M_{L_{\max}} &\leq \sum_{L=1}^{L_{\max}} 2^L \binom{N-1+L}{N-1} \qquad (5.28) \\
&\leq \sum_{L=1}^{L_{\max}} 2^L \left(1+\frac{L}{N-1}\right)^{N-1} \mathrm{e}^{N-1} \\
&\leq \mathrm{e}^{N-1} 2^{L_{\max}+1}\left(1+\frac{L_{\max}}{N-1}\right)^{N-1} \qquad (5.29) \\
&\leq 2\mathrm{e}^{N-1}\left\{\log\left(\frac{3C_{\mathrm{sc}}}{\varepsilon}\right)\right\}^N \left\{C_2 + \frac{1}{\log 2}\log\log\left(\frac{3C_{\mathrm{sc}}}{\varepsilon}\right)\right\}^{N-1},
\end{aligned}
$$

where in the last line we have used (5.18) to replace $L_{\max}$. Substituting (5.27) and (5.28) into (5.26) concludes the proof. □

Theorem 5.7. *Given Assumption A2, and the conditions of Lemmas 5.4.1 and 3.4.1, for $\varepsilon > 0$, the minimum total number of CG iterations $K_{\mathrm{acc}}(\varepsilon)$, to achieve $\|u - \widetilde{u}_{h,L_{\max}}\|_{L^2_\varrho} < \varepsilon$, in Algorithm 1, is bounded by*

$$
\begin{aligned}
K_{\mathrm{acc}}(\varepsilon) \leq C_1 &\left[\log\left(\frac{3C_{\mathrm{sc}}}{\varepsilon}\right)\right]^N \left[C_2 + \frac{1}{\log 2}\log\log\left(\frac{3C_{\mathrm{sc}}}{\varepsilon}\right)\right]^{N-1} \\
&\times \frac{1}{\log\left(\frac{\sqrt{\overline{\kappa}}+1}{\sqrt{\overline{\kappa}}-1}\right)}\left\{C_5 + 2\left(2^{\frac{1}{N}}-1\right)\log\left(\frac{3C_{\mathrm{sc}}}{\varepsilon}\right) + 2N\log\log\left[\frac{1}{rN}\log\left(\frac{3C_{\mathrm{sc}}}{\varepsilon}\right)\right]\right\},
\end{aligned}
\tag{5.30}
$$

where $\overline{\kappa} = \sup_{\boldsymbol{y}\in\Gamma}(\kappa(\boldsymbol{y}))$, C_1 and C_2 are defined as in (5.25), and C_5 is defined by

$$
C_5 = 2N\log\left(\frac{2N}{\log 2}\right) + \log\left(4\sqrt{\frac{\alpha}{\beta}}\right).
$$

Proof. To achieve the prescribed error, we again choose $h = h(\varepsilon)$, $L_{\max} = L_{\max}(\varepsilon)$ and $\tau = \tau(\varepsilon)$ as in Lemmas 5.4.3 and 5.4.4. Then, $K_{\text{acc}}(\varepsilon)$ can be bounded by

$$K_{\text{acc}}(\varepsilon) = \sum_{L=1}^{L_{\max}} \sum_{\boldsymbol{y}_{L,j} \in \Delta\mathcal{H}_L} k_{L,j} \leq \sum_{L=1}^{L_{\max}} \Delta M_L \, k_{\text{acc}}^L.$$

From Lemma 5.4.4 and 5.4.5, for $L = 1, \ldots, L_{\max}$, we have

$$\begin{aligned}
k_{\text{acc}}^L &\leq \log\left(\frac{4\sqrt{\alpha} C_{\text{sc}} \, \mathrm{e}^{-rN2^{(L-1)/N}}}{\tau}\right) \Bigg/ \log\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right) \\
&\leq \frac{1}{\log\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)} \log\left(\frac{12\sqrt{\alpha} C_{\text{sc}} C_{L_{\max}} \mathrm{e}^{-rN2^{(L-1)/N}}}{\sqrt{\beta}\varepsilon}\right) \\
&= \frac{1}{\log\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)} \log\left[\left(\frac{3C_{\text{sc}} \mathrm{e}^{-rN2^{L/N}}}{\varepsilon}\right) 4\sqrt{\frac{\alpha}{\beta}} C_{L_{\max}} \mathrm{e}^{rN2^{L/N} - rN2^{(L-1)/N}}\right] \\
&\leq \frac{1}{\log\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)} \left[\log\left(4\sqrt{\frac{\alpha}{\beta}} C_{L_{\max}}\right) + rN\left(2^{L/N} - 2^{(L-1)/N}\right)\right].
\end{aligned}$$

Hence,

$$K_{\text{acc}}(\varepsilon) \leq M_{L_{\max}} \frac{\log\left(4\sqrt{\alpha/\beta} C_{L_{\max}}\right)}{\log\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)} + \frac{rN}{\log\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)} \underbrace{\sum_{L=1}^{L_{\max}} \Delta M_L \left(2^{L_{\max}/N} - 2^{(L-1)/N}\right)}_{S},$$

where S can be bounded using results from geometric sums, i.e.,

$$\begin{aligned}
S &\leq \sum_{L=1}^{L_{\max}} 2^L \binom{N-1+L}{N-1} \left(2^{L_{\max}/N} - 2^{(L-1)/N}\right) \\
&\leq \mathrm{e}^{N-1} \left(1 + \frac{L_{\max}}{N-1}\right)^{N-1} \sum_{L=1}^{L_{\max}} \left(2^{L_{\max}/N} - 2^{(L-1)/N}\right) 2^L \\
&= \mathrm{e}^{N-1} \left(1 + \frac{L_{\max}}{N-1}\right)^{N-1} \left\{\left(1 - \frac{1}{2^{1+1/N}}\right) 2^{L_{\max}+1} 2^{L_{\max}/N} + \frac{2}{2^{1+1/N}-1} - 2^{1+L_{\max}/N}\right\} \\
&\leq \mathrm{e}^{N-1} \left(1 + \frac{L_{\max}}{N-1}\right)^{N-1} \left(2^{1/N} - 1\right) 2^{L_{\max}+2} 2^{L_{\max}/N}.
\end{aligned}$$

Combining the last two inequalities, along with (5.28), we get

$$
\begin{aligned}
K_{\text{acc}}(\varepsilon) \leq \mathrm{e}^{N-1}\left(1+\frac{L_{\max}}{N-1}\right)^{N-1} & 2^{L_{\max}+1} \\
& \times \frac{1}{\log\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)} \log\left(4\sqrt{\frac{\alpha}{\beta}}\right) + 2N\log\left(L_{\max}+2\right) + 2rN\left(2^{1/N}-1\right)2^{L_{\max}/N}.
\end{aligned}
$$

Substituting (5.18) for $L_{\max}$ concludes the proof. □

In the case of the accelerated SC method, an interpolant $\mathcal{I}_{L-1}[\widetilde{u}_h]$, defined by (3.6) and (5.2), must be evaluated for each of the ΔM_L collocation points in $\Delta\mathcal{H}_L$. Each interpolant evaluation costs $2M_{L-1}-1$ operations, i.e., additions and multiplications, and must be evaluated for each of the M_h components of the FE coefficient vector. Then the interpolation cost on each level is $M_h\Delta M_L(2M_{L-1}-1)$ for $L = 1,\ldots,L_{\max}(\varepsilon)$. Now we give an estimate of the total interpolation cost $\mathcal{C}_{\text{int}}(\varepsilon)$ for our algorithm to achieve the prescribed accuracy ε.

Theorem 5.8. *Given Assumption A2 and the conditions of Lemma 5.4.1, for sufficiently small $\varepsilon > 0$, the total cost of interpolation when using the sparse grid interpolation method in (5.3) is bounded by*

$$
\mathcal{C}_{\text{int}}(\varepsilon) \leq M_h C_8 \left(\log\left(\frac{3C_{\text{sc}}}{\varepsilon}\right)\right)^{2N}\left\{C_2 + \frac{1}{\log 2}\log\log\left(\frac{3C_{\text{sc}}}{\varepsilon}\right)\right\}^{2(N-1)},
$$

where C_2 are defined as in Theorem 5.6, and $C_8 = 64\mathrm{e}^{-2}\left(\mathrm{e}/rN\right)^{2N}$.

Proof. The total interpolation cost is bounded by

$$
\begin{aligned}
\mathcal{C}_{\text{int}}(\varepsilon) &\leq 2M_h \sum_{L=2}^{L_{\max}(\varepsilon)} \Delta M_L M_{L-1} \\
&\leq 2M_h \sum_{L=2}^{L_{\max}(\varepsilon)} 2^L \binom{N-1+L}{N-1} \sum_{l=1}^{L} 2^l \binom{N-1+l}{N-1} \\
&\leq 4M_h \left\{\binom{N-1+L_{\max}(\varepsilon)}{N-1}\right\}^2 4^{L_{\max}(\varepsilon)+1} \\
&\leq 16M_h \mathrm{e}^{2(N-1)} 4^{L_{\max}(\varepsilon)}\left(1+\frac{L_{\max}(\varepsilon)}{N-1}\right)^{2(N-1)}. \qquad (5.31)
\end{aligned}
$$

Substituting the definition of $L_{\max}(\varepsilon)$ from Lemma 5.4.3 into (5.31) concludes the proof. $\square$

Based on Theorems 5.6, 5.7 and 5.8, we finally discuss the savings of the accelerated SC method proposed in §5.2. By comparing the estimates of $K_{\text{zero}}(\varepsilon)$ and $K_{\text{acc}}(\varepsilon)$, we see that the acceleration technique reduces $\log(C_3/\varepsilon)$ in (5.24) to $2\left(2^{1/N}-1\right)\log\left(3C_{\text{sc}}/\varepsilon\right)$ in (5.30). On the other hand, when taking into account the cost of interpolation $\mathcal{C}_{\text{int}}$, we must consider the cost $\mathcal{C}_{\text{iter}}$ of performing each iteration.

In the case of CG solvers, $\mathcal{C}_{\text{iter}}$ is the cost of one matrix-vector multiplication, and will be determined by the size of the unknown vector, M_h, and the sparsity of the mass matrix $\boldsymbol{A}(\boldsymbol{y})$. Thus $\mathcal{C}_{\text{iter}}$ is proportional to the size of the finite element vector, i.e., $\mathcal{C}_{\text{iter}} = C_D M_h$, where C_D depends on the dimension d of the physical domain and choice of finite element basis. For example, without the use of a preconditioner, we can assume that the condition numbers of the matrices $\boldsymbol{A}(\boldsymbol{y})$, for $\boldsymbol{y} \in \Gamma$, satisfy $\overline{\kappa} := \sup_{\boldsymbol{y}\in\Gamma} \kappa(\boldsymbol{y}) \leq (C_\kappa/h)^2$, where the constant $C_\kappa > 0$ is independent of $\boldsymbol{y} \in \Gamma$ [4]. Then we can specify the contribution of the condition number in Theorems 5.6 and 5.7; using $\log(x) \geq (x-1)/x$ and Lemmas 5.4.1 and 5.4.3, we estimate

$$\frac{1}{\log\left(\frac{\sqrt{\overline{\kappa}}+1}{\sqrt{\overline{\kappa}}-1}\right)} \leq \frac{\sqrt{\overline{\kappa}}+1}{2} \leq C_\kappa \left(\frac{3C_{\text{fem}}}{\varepsilon}\right)^{1/s}.$$

Now as $\varepsilon \to 0$, the asymptotic iterative solver costs, $\mathcal{C}_{\text{zero}} = C_D M_h K_{\text{zero}}$ are of the order $M_h \left(\frac{1}{\varepsilon}\right)^{1/s} \left\{\log\left(\frac{1}{\varepsilon}\right)\right\}^{N+1} \left\{\log\log\left(\frac{1}{\varepsilon}\right)\right\}^{N-1}$, while in the accelerated case, the estimate for $C_D M_h K_{\text{acc}}$, is of the same order with respect to ε, but with an improvement from the factor $\left(2^{1/N}-1\right)$ in the constant. For the accelerated method, the additional interpolation costs $\mathcal{C}_{\text{int}}$ are of order $M_h \left\{\log\left(\frac{1}{\varepsilon}\right)\right\}^{2N} \left\{\log\log\left(\frac{1}{\varepsilon}\right)\right\}^{2(N-1)}$, which is negligible compared to the iterative solver complexity. It is clear that, asymptotically, the accelerated method leads to a net reduction in computational cost. We remark that for many adaptive interpolation methods, the addition of new points already involves evaluation of the current (coarse) interpolant. In this case, the cost of interpolation can be ignored, and the accelerated method should be used.

5.4 Numerical examples

The goal of this section is to demonstrate the reduction in computational cost of SC methods using the proposed acceleration technique. In Example 5.1, we first use the accelerated SC method to solve a stochastic parameterized elliptic PDE with one spatial dimension, and compute the overall cost and iteration savings gained by acceleration. Example 5.2 considers a similar problem and looks at the number of CG iterations versus the collocation error, also demonstrating the effect of varying parameter dimension N on the convergence of the individual systems. In addition, as described in Remark 5.4, we extend our acceleration technique to interpolated preconditioners, which also exhibit the improvements of the method. Finally, Example 5.3 applies the accelerated method to iterative solvers for nonlinear parametrized PDEs.

The analysis in section 5.3.1 had two components: (i) estimates for the reduction in total solver iterations using acceleration, and (ii) interpolation costs. The interpolation costs can be computed exactly for the non-adaptive methods we consider, and in Example 5.1, we balance all error contributions and examine the total cost, including both solver iterations and interpolation construction. In Examples 5.2 and 5.3 we focus only on the number of iterations of the CG solver.

Example 5.1

We consider the following elliptic stochastic parameterized PDE

$$\begin{cases} -\nabla \cdot \left(a\left(x, \boldsymbol{y}\right) \nabla u\left(x, \boldsymbol{y}\right)\right) &= 10 \quad \text{in } D \times \Gamma, \\ u(x, \boldsymbol{y}) &= 0 \quad \text{on } \partial D \times \Gamma, \end{cases} \tag{5.32}$$

where $D = [0, 1]$, $\boldsymbol{y} = (y_1, y_2, y_3, y_4)^\top$, $\Gamma_n = [-1, 1], n = 1, \ldots, 4$, and a is given by:

$$\log\left(a\left(x, \boldsymbol{y}\right) - 1\right) = \mathrm{e}^{-1/8}\left(y_1 \cos \pi x + y_2 \sin \pi x + y_3 \cos 2\pi x + y_4 \sin 2\pi x\right). \tag{5.33}$$

The $\{y_i\}_{i=1}^4$ are i.i.d. uniform random variables in $[-1,1]$. In the one-dimensional physical domain, an FE discretization using linear elements yields tridiagonal, symmetric positive-definite systems. While this type of system could be solved efficiently by direct methods, nevertheless we use CG solvers to demonstrate the convergence properties of the acceleration method.

Table 5.1 compares the standard and the accelerated SC methods, where the error for each approximate solution, $\widetilde{u}_{h,L_{\max}}$, is computed against a highly refined approximate reference solution $\widetilde{u}_{h^*,L^*}$ with $h^* = 2^{-14}, L^* = 10$. In Figure 5.1 we plot the savings of the accelerated SC method, computed according to the cost metrics (5.7) and (5.8). Since the constants C_{fem} and C_{sc} in Lemma 5.4.3 are not known *a priori*, to balance the error contributions in (5.17), we use a set of 100 realizations of $u_h(x, \boldsymbol{y})$, obtained with very small mesh size and CG tolerance, as reference solutions; and then tune h, τ, L using those realizations, until the interpolant achieves the desired overall error ε in the L^2_ϱ norm. Especially for the larger systems, i.e., those with a large number of spatial degrees of freedom, significant savings are achieved. The percent savings in the number of iterations versus the cost of interpolation are calculated according to

$$\frac{\mathcal{C}_{\text{zero}} - \mathcal{C}_{\text{acc}}}{\mathcal{C}_{\text{zero}}} = \frac{M_h C_D (K_{\text{zero}} - K_{\text{acc}}) - \mathcal{C}_{\text{int}}}{M_h C_D K_{\text{zero}}},$$

where $C_D = 5$, since the matrices are tridiagonal.

Table 5.1: Comparison in computational cost between the standard and the accelerated SC methods for solving (5.32)–(5.33).

Tot. Err	FE DoFs	SC Pts	CG tol	K_{zero}	K_{acc}	Savings
1×10^{-2}	255	137	1×10^{-3}	28,259	21,123	19.4 %
5×10^{-3}	511	401	5×10^{-3}	173,671	83,884	42.4%
1×10^{-3}	2,047	1,105	1×10^{-4}	2,001,905	626,215	62.3%
5×10^{-4}	4,095	2,929	5×10^{-5}	10,878,352	1,842,703	74.5%
1×10^{-4}	16,383	7,537	1×10^{-5}	114,570,175	12,345,968	75.1%

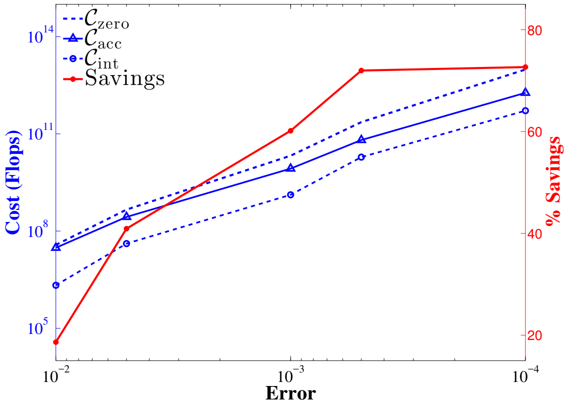

Figure 5.1: Cost (left axis) and percent savings (right axis) of the accelerated SC method versus the standard SC method for (5.32)–(5.33). Costs are computed using (5.7) and (5.8).

Example 5.2

We consider the following stochastic parameterized linear elliptic problem

$$\begin{cases} -\nabla \cdot (a\,(x, \boldsymbol{y})\, \nabla u\,(x, \boldsymbol{y})) = \cos(x_1)\sin(x_2) & \text{in } \; D \times \Gamma, \\ u(x, \boldsymbol{y}) = 0 & \text{on } \; \partial D \times \Gamma, \end{cases} \tag{5.34}$$

where $D = [0,1] \times [0,1]$, $\Gamma_n = [-\sqrt{3}, \sqrt{3}], n = 1, \ldots, N$, and $x = (x_1, x_2)$ is the spatial variable. The random variables $\{y_n\}_{n=1}^N$ are i.i.d. and are each uniformly distributed in $[-\sqrt{3}, \sqrt{3}]$, with zero mean and unit variance, i.e., $\mathbb{E}[y_n] = 0$, and $\mathbb{E}[y_n y_m] = \delta_{nm}$, for $n, m \in \mathbb{N}_+$. The coefficient a represents the N-term truncation of an expansion of a random field with stationary covariance function, given by $\text{Cov}\left[\log\left(a - 0.5\right)\right](x_1, x_1') = \exp\left(-(x_1 - x_1')^2/R_c^2\right)$, where $x_1, x_1' \in [0,1]$, and R_c is the correlation length. Then, we have

$$\log(a(x, \boldsymbol{y}) - 0.5) = 1 + y_1 \left(\sqrt{\pi} R_c/2\right)^{1/2} + \sum_{n=2}^{N} \zeta_n \varphi_n(x) y_n, \tag{5.35}$$

where ζ_n and $\varphi_n(x)$ are the eigenvalues and eigenfunctions associated with the covariance function; see [71] for more details on this example and the explicit calculation of the eigenvalues and eigenfunctions. Here we will consider two correlation lengths, namely $R_c = 1/2$, and $R_c = 1/64$. where Figure 5.2 shows the corresponding decay of eigenvalues.

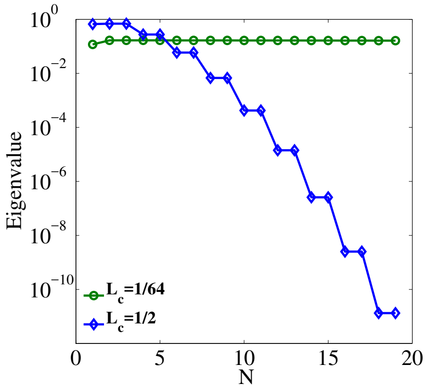

Figure 5.2: First 19 eigenvalues for (5.35) for correlation length $R_c = 1/64, 1/2$.

For the spatial discretization, we use a finite element approximation on a regular triangular mesh with linear finite elements and 4225 degrees of freedom. The CG method is used for the linear solver with diagonal preconditioners and a tolerance of 10^{-14}.

First, in Table 5.2, we report the error and total iteration count of both the standard case, using zero initial vectors, and accelerated SC construction, computed with several dimensions N and $R_c = 1/64$. The error is measured using the expectation of the approximate solutions, $\|\mathbb{E}[u_{h,L_{\max}}] - \mathbb{E}[u_{h,L^*}]\|_{L^2(D)}$, for $L_{\max} = 1, \ldots, 7$, where the "exact" solution $\mathbb{E}[u_{h,L^*}]$ is computed using $L^* = 8$. We compare these errors against the cumulative total number of iterations, K_{zero} and K_{acc}, needed to construct $\mathbb{E}[u_{h,L_{\max}}]$.

An alternative approach to accelerating SC methods is found in [45]. For a particular SC level $L_{\max}$, this method orders the collocation points lexicographically, with each dimension ordered according to the decay of the eigenvalues associated with (5.35). We also implemented a similar method without the sequential ordering; for a given level L, at each new collocation point in $\Delta\mathcal{H}_L$ the solution at the nearest collocation point from lower levels is given as an initial guess to accelerate the CG solver. We refer to this method as the "nearest neighbor" approach. Figure 5.3 shows the average number of iterations needed to solve the linear system (5.5), where the average is taken over the new points at level L, i.e., $\Delta\mathcal{H}_L$, for $L = 1, \ldots, 7$. We compare our interpolated acceleration algorithm, the nearest neighbor approach, and standard SC method without acceleration, for $N = 3$ and $N = 11$, using $R_c = 1/64$. The interpolated initial vector provided by the acceleration algorithm

Table 5.2: Iteration counts and savings of the accelerated SC method for solving (5.34)–(5.35) with correlation length $R_c = 1/64$, and parameter dimensions $N = 3, 5, 7, 9$, and 11.

	Error	SC Pts	K_{zero}	K_{acc}	Savings in K
N=3	3.83e-8	25	6,780	5,991	11.6%
	9.57e-10	69	18,893	14,628	22.6%
	9.86e-12	177	48,691	27,765	43.0%
N=5	5.28e-07	61	17058	15095	11.6%
	1.03e-08	241	67,955	53,992	20.6%
	1.44e-10	801	226,597	150,241	33.7%
N=7	2.43e-08	589	168,237	136,072	19.1%
	6.63e-10	2,465	706,049	500,718	29.1%
	1.94e-11	9,017	2,585,970	1,496,391	42.1%
N=9	1.68e-07	1,177	338,428	277,583	18.0%
	7.83e-09	6,001	1,729,337	1,273,895	26.3%
	8.86e-11	26,017	7,505,343	4,719,820	37.1%
N=11	2.59e-07	2,069	596,368	495,705	16.9%
	2.43e-08	12,497	3,608,185	2,736,615	24.2%
	1.95e-09	63,097	18,231,420	12,139,658	33.4%

yields a reduction in the average number of iterations at each level, which increases with L. Figure 5.3 also shows the effect of using the nearest neighbor solution as the initial vector, which provides some improvement over the standard case using zero initial vectors, but the savings do not match those of our approach.

The left plot of Figure 5.4 shows the total iteration savings achieved by the acceleration algorithm with different maximum collocation levels $L_{\max} = 1, \ldots, 6$. The savings are measured as the percentage reduction in the cumulative iteration count up to level $L_{\max}$, relative to standard case using zero initial vectors, i.e., $(K_{\text{zero}} - K_{\text{zero}})/K_{\text{zero}}$. Here we also see the effect of random parametric dimension on the convergence of SC methods: as N increases, our algorithm provides less accurate initial guesses for a given maximum SC level $L_{\max}$. This can also be seen by comparing the left and right plots of Figure 5.3, which show how the average number of iterations at a given SC level L changes from $N = 3$ to $N = 11$.

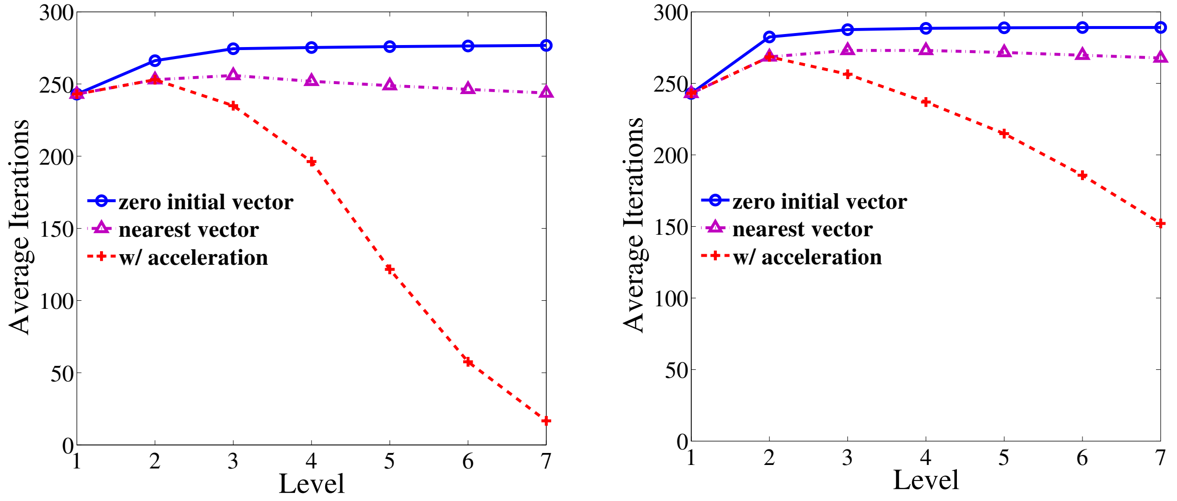

Figure 5.3: Comparison of the average CG iterations per level for solving problem (5.34)–(5.35) with dimensions $N = 3$ (left) and $N = 11$ (right), and correlation length $R_c = 1/64$.

On the other hand, the right plot of Figure 5.4 shows the same total iteration savings now plotted versus error. As above, the error is measured as $\|\mathbb{E}[u_{h,L_{\max}}] - \mathbb{E}[u_{h,L^*}]\|_{L^2(D)}$, with $L^* = 7$. These results are in agreement with the theoretical asymptotic estimates from Theorem 5.7, which predict an increased savings vs error for larger dimensions.

For two different correlation lengths $R_c = 1/2$ and $R_c = 1/64$, Figure 5.5 plots the convergence of the error in $\mathbb{E}[u_{h,L}]$ versus the total number of CG iterations for $N = 3$ and $N = 11$. The larger correlation length, $R_c = 1/2$, results in slower convergence of the SC interpolant than for $R_c = 1/64$, but note that the accelerated method reduces the total iteration count in both cases.

On the other hand, we can employ *anisotropic* methods to increase the efficiency of SC in the case of larger correlation lengths [70]. Anisotropic SC methods will place more points in directions corresponding to large eigenvalues of (5.35), and the importance of each dimension is encoded in a weight vector (see (5.10)). Figure 5.6 plots the average number of iterations for problem (5.34)–(5.35) with a relatively large correlation length $R_c = 1/2$, and $N = 11$. Here we employ the weights given by an *a posteriori* selection described in [70], i.e., the weight vector $\boldsymbol{\alpha} \in \mathbb{R}^N$, with $\alpha_1 = 0.85, \alpha_2 = \alpha_3 = 0.8, \alpha_4 = \alpha_5 = 1.0, \alpha_6 = \alpha_7 = 1.6, \alpha_8 = \alpha_9 = 2.6, \alpha_{10} = \alpha_{11} = 3.7$. The acceleration method decreases the average number of iterations

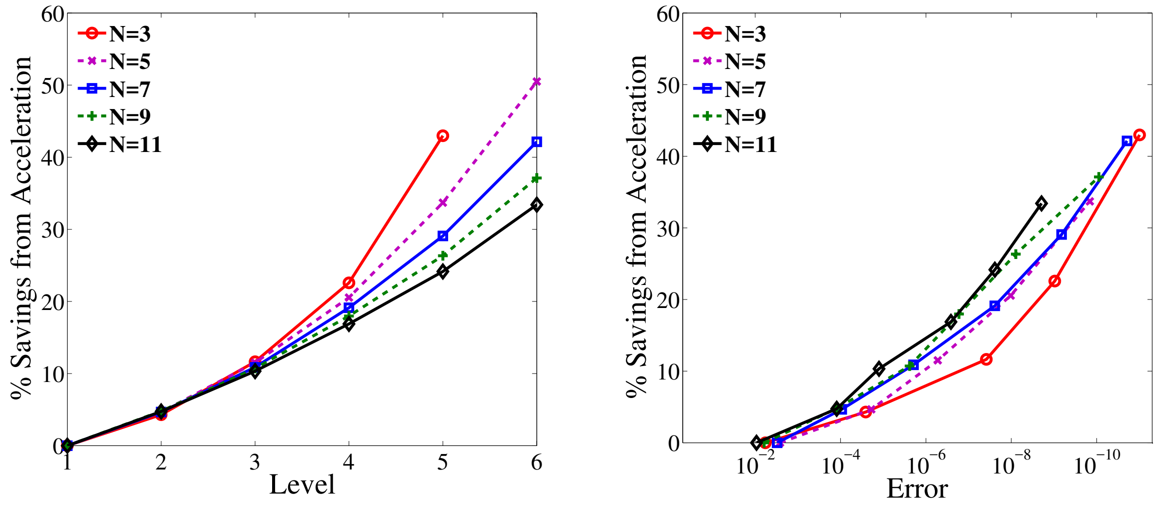

Figure 5.4: Percentage cumulative reduction in CG iterations vs level (left) and error (right) for solving (5.34)–(5.35) using our accelerated approach, with $N = 3, 5, 7, 9, 11$, and for correlation length $R_c = 1/64$.

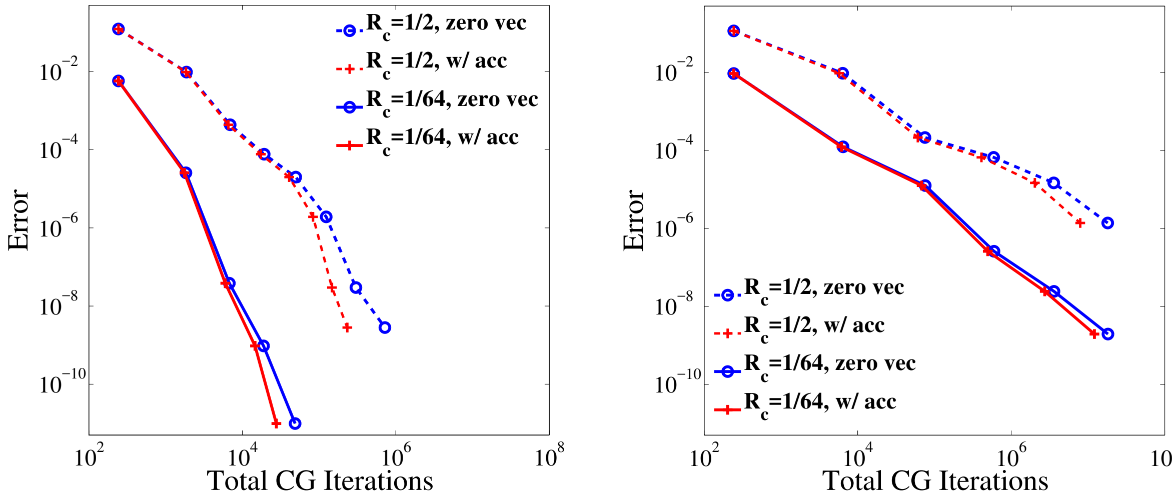

Figure 5.5: The convergence of the SC approximation for solving (5.34)–(5.35), using CG, with and without acceleration, for correlation lengths $R_c = 1/64, 1/2$, and dimensions $N = 3$ (left), and $N = 11$ (right).

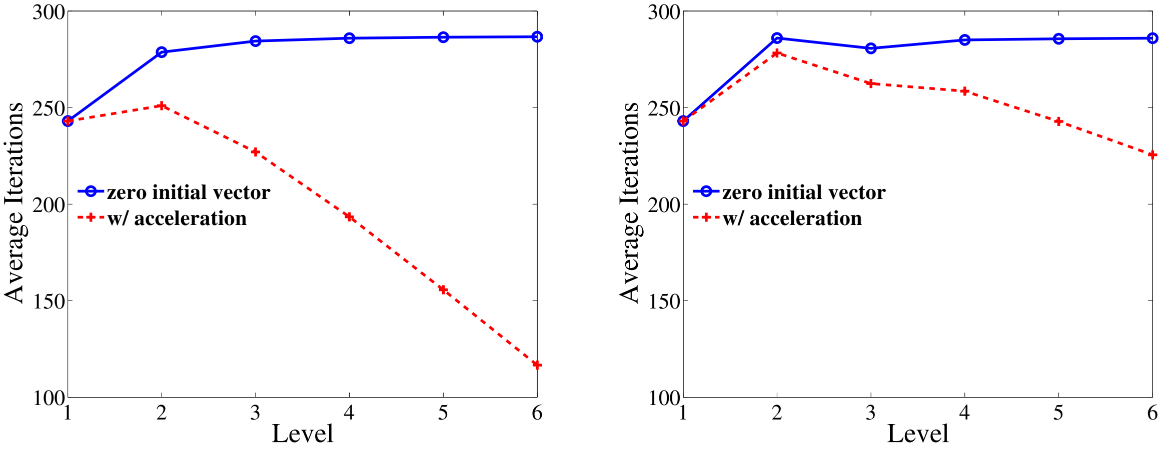

Figure 5.6: Average CG iterations per level for solving problem (5.34)–(5.35) for $N = 11$ and with correlation length $R_c = 1/2$, using an isotropic SC (left) and anisotropic SC (right). The inefficiencies from using an isotropic grid are partially offset by increased gains from acceleration.

needed to solve the linear system, but the effect is not as pronounced as in the case of an isotropic SC method. This occurs because the isotropic method places far too many points in relatively unimportant directions, thus the dependence of $u(\boldsymbol{y})$ on a certain component y_n of $\boldsymbol{y}$ may be well approximated at very low levels. Anisotropic methods exhibit better convergence with respect to $M_{L_{\max}}$ (and lower interpolation costs) versus isotropic methods, yet we see here that the acceleration algorithm helps to somewhat offset the inefficiency of isotropic methods for anisotropic problems.

In the preceding results we have used a simple diagonal preconditioner strategy. As described in Remark 5.4, we can also construct efficient preconditioners with our acceleration scheme. Table 5.3 shows the effectiveness of different preconditioning strategies for solving equations (5.34)–(5.35), with $N = 7$ and $R_c = 1/64$, where we compare the average number of iterations needed to solve (5.5) at each new point $\boldsymbol{y}_{L,j} \in \Delta\mathcal{H}_L$ at a given level L. Here we compute an incomplete Cholesky preconditioner for each linear system on the levels $L = 1, \ldots, L_{\mathrm{PC}}$, for $L_{\mathrm{PC}} = 1, 2$, and 3, and use these to provide an "accelerated" preconditioner (5.9) for the systems on the remaining levels $L_{\mathrm{PC}} + 1, \ldots, L_{\max}$. We compare this against the cases where a simple diagonal preconditioner and an incomplete Cholesky preconditioner are used. The three-level accelerated preconditioner reduces the average number of iterations to

Table 5.3: Average iteration counts for the standard (top), and the accelerated (bottom) SC method using six preconditioner schemes to solve (5.34)–(5.35) with $N = 7$, and $R_c = 1/64$.

CG iterations for standard SC						
Level	No PC	Diag PC	Inc. Chol.	$L_{\mathrm{PC}} = 1$	$L_{\mathrm{PC}} = 2$	$L_{\mathrm{PC}} = 3$
1	243	243	55	55	–	–
2	311.8	278.4	54.7	60.7	54.7	–
3	332.3	284.9	54.6	63.5	54.9	54.6
4	341.0	286.1	54.6	65.2	55.3	54.6
5	345.8	286.7	54.6	66.2	55.5	54.6
6	348.4	286.9	54.6	66.7	55.6	54.6

CG iterations for accelerated SC						
Level	No PC	Diag PC	Inc. Chol.	$L_{\mathrm{PC}} = 1$	$L_{\mathrm{PC}} = 2$	$L_{\mathrm{PC}} = 3$
1	243	243	55	55	–	–
2	299.3	264.6	52.9	58.4	52.9	–
3	295.8	251.3	49.1	57.1	49.4	49.1
4	270.8	225.8	43.7	52.3	44.2	43.7
5	237.0	194.3	37.3	45.8	38.0	37.3
6	186.1	151.9	28.9	36.0	29.5	28.9

within a decimal point of the incomplete Cholesky preconditioner, and the cost of computing the low-level preconditioners and interpolating is relatively cheap in comparison.

Example 5.3

The preceding experiments demonstrate the benefits of using acceleration to reduce the overall number of iterations of an individual linear solvers. In the case of a nonlinear PDE, the possibilities for savings can be even greater than the linear cases above, since convergence of a nonlinear solver may be slow or even unattainable from a poor initial vector. In this example, we consider the problem

$$\begin{cases} -\nabla \cdot \left(a\left(x,\boldsymbol{y}\right)\nabla u\left(x,\boldsymbol{y}\right)\right) + F[u](x,\boldsymbol{y}) = x & \text{in } D \times \Gamma, \\ u(0,\boldsymbol{y}) = 0 & \text{in } \Gamma, \\ u'(1,\boldsymbol{y}) = 1 & \text{in } \Gamma, \end{cases}$$

where a is given by (5.33), $D = [0,1]$, $\Gamma_n = [-1,1]$, $n = 1,\ldots,4$, and $F[u]$ is some nonlinear function of u. In what follows, we consider the nonlinear functions $F[u] = u^5$, and $F[u] = uu'$.

Nonlinear problems are typically solved with the use of iterative methods such as Picard iterations or Newton's method. We implement a combination of these methods that begins with Picard iterations, then utilizes Newton's method once the relative errors are small. For spatial discretization, we use piecewise linear finite elements on $[0,1]$ with a mesh size of $h = 1/500$, and solved the resulting systems at each iteration using exact methods. We remark that the solution of the linear systems is *not* accelerated by our algorithm, but we only decrease the total number of Picard and Newton iterations. The stopping criterion for the solver is a relative tolerance of 10^{-8} in the l^2 norm.

Results for these experiments are given in Figure 5.7. For each SC level, $L = 1,\ldots,8$, we plot the average number of nonlinear iterations, where the average is taken over the set of points which are new to level L, namely $\Delta\mathcal{H}_L$. Finally, we show the total computational time in Table 5.4, for different maximum levels of collocation approximation, measured on a workstation with 1.7GHz dual core processors and 8 GB of RAM. We note that in Table 5.4, the size of the finite element system is fixed. Thus, as we move to higher levels of

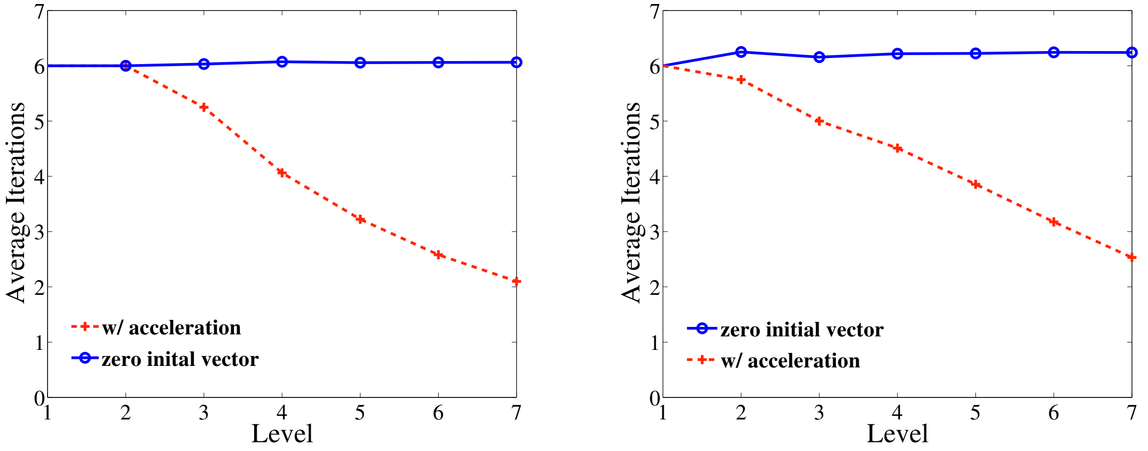

Figure 5.7: Average number of nonlinear iterations per level for solving problem (5.36) with $F[u] = uu'$ (left) and $F[u] = u^5$ (right).

collocation, the approximation in parameter space becomes relatively more expensive to compute compared to the solving the finite element systems. This is why the savings begin to decrease after level 5, even though Figure 5.7 shows dramatic savings in iterations for higher levels. Furthermore, the reason for the negative savings for the SC approximation with $L = 2$ is that the interpolant is not yet accurate enough to overcome the additional cost of the acceleration.

5.5 Remarks

In this chapter, we proposed and analyzed an acceleration method for construction of sparse interpolation-based approximate solutions to PDEs with random input parameters. The acceleration method exploits the sequence of increasingly accurate approximate solutions to provide increasingly good initial guesses for the underlying deterministic iterative solvers. We have developed this method using a global Lagrange polynomial basis but the method can easily be extended to other non-intrusive methods.

While our method takes advantage of the natural structure provided by hierarchical SC methods, we do not take advantage of any hierarchy in the spatial approximation. Our method may be used in combination with multilevel methods [92] to accelerate the

Table 5.4: Computational time in seconds for computing solution to problem 5.36 using the accelerated method ("acc") and the standard method ("zero").

SC Level	2	3	4	5	6
$F[u] = u^5$, acc	.03018	.113832	.2746	.7039	2.33314
$F[u] = u^5$, zero	.025976	.119256	.339678	.949184	2.61958
% Savings	-16.2	4.5	19.2	25.8	10.9
$F[u] = uu'$, acc	.027754	.089082	.22706	.629451	2.05741
$F[u] = uu'$, zero	.026527	.090435	.273355	.895027	2.4008
% Savings	-4.6	1.5	16.9	29.7	14.3

construction of SC interpolants, and reuse information from level to level. The combination of the acceleration scheme with multilevel methods will be the subject of future work.

We rigorously studied error estimates in the special the case of linear elliptic PDEs with random inputs, providing complexity estimates for the proposed method. Several numerical examples confirm the expected performance. While the analysis of §5.3.1 applies to linear stochastic parameterized PDEs, the acceleration method may be even more well suited to nonlinear problems, as convergence rates may be improved, based on the choice of a good initial guess for nonlinear iterative solvers. A final numerical example demonstrates the advantage of our approach to nonlinear problems. A more rigorous study of acceleration for nonlinear solvers and extension to time dependent problems may provide interesting opportunities in the future.

Part II

Efficient Point Sets for Multidimensional Interpolation and Quadrature

Chapter 6

Lebesgue Constants for Leja Sequences on Unbounded Domains

The following content has been submitted for publication, and is available in [55]. The author acknowledges colloboration from Guannan Zhang on this project.

The Lebesgue constant for a countable set of nodes provides a measure of how well the interpolant of a function at the given points compares to best polynomial approximation of the function. We are especially interested in how this constant grows with the number of interpolation nodes, i.e., the corresponding degree of the interpolating polynomial, in an unbounded domain. Due to a simple recursive formulation, the Leja points show promise as a foundation for multi-dimensional approximation methods such as sparse grid collocation [68]. As such, in this chapter we analyze the Lebesgue constant for a sequence of weighted Leja points on the real axis. Leveraging results from weighted potential theory [82], and orthogonal polynomials with exponential weights [61], we show that the Lebesgue constant for the weighted Leja points grows subexponentially with the number of interpolation nodes.

The rest of the chapter is organized as follows. In §6.1, we introduce the concept of weighted Lagrange interpolation of a function on the real line, and in Theorem 6.1 state our main result that describes the growth of the Lebesgue constant for weighted Leja points. To prove our new theorem, we use results from potential theory, which we introduce in §6.2. Specifically, we exploit the relationship between discrete potentials and polynomials with zeros at the Leja points, and the fact that the measures μ_n converge weak* to the appropriate equilibrium measure of the Fekete points. While potential theory gives us almost the whole result, we also require some explicit estimates on the spacing of the weighted Leja points, which are given in §6.3. The completion of the proof of our main theorem describing the growth of the Lebesgue constant for weighted Leja points is given in §6.4, followed by concluding remarks.

6.1 Lagrange Interpolation and Leja Points

In this section we recall in more detail the problem of weighted Lagrange interpolation of a function on the real line. We also discuss the Lebesgue constant for a set of interpolation points, and show how it relates to the best approximation error. Finally, in §6.1.1 we describe our main contribution, which involves a theoretical estimate of the growth of the Lebesgue constant of the weighted Leja sequence versus of the number of interpolation points. More specifically, in Theorem 6.1 we prove that the Lebesgue constant of the weighted Leja points grows subexponentially.

To make the setting precise, assume we are given a continuous function f on $\mathbb{R}$ that we would like to interpolate. In other words, we have a set of $n+1$ points, $\{x_k\}_{k=0}^n \subset \mathbb{R}$, and the values $\{f(x_k)\}_{k=0}^n$ at each of those points. Lagrange interpolation constructs a polynomial $\mathcal{I}_n[f]$, of degree n, that matches f at every interpolation point, i.e.,

$$\mathcal{I}_n[f](x_k) = f(x_k), \quad k = 0, \ldots, n.$$

The fundamental Lagrange basis functions for $\{x_k\}_{k=0}^n$ are defined as:

$$l_{n,k}(x) = \prod_{\substack{j=0 \\ j\neq k}}^{n} \frac{(x - x_j)}{(x_k - x_j)}, \qquad k = 0, \ldots, n. \tag{6.1}$$

These functions satisfy $l_{n,k}(x_j) = \delta_{j,k}$ for all $j, k = 0, \ldots, n$. The unique Lagrange interpolant of degree n for f is then given by

$$\mathcal{I}_n[f](x) = \sum_{k=0}^{n} f(x_k) l_{n,k}(x). \tag{6.2}$$

Given an appropriate weight function $w : \mathbb{R} \to [0, 1]$, to estimate the w-weighted approximation error for this interpolation scheme, we define $\mathbb{P}_n = \operatorname{span}\{x^j\}_{j=0}^n$ to be the space of polynomials of degree at most n over $\mathbb{R}$, and let p_n be an arbitrary element of $\mathbb{P}_n$. Then the error in the norm of $L^\infty(\mathbb{R})$, with $\|\cdot\|_\infty := \|\cdot\|_{L^\infty(\mathbb{R})}$, is given by

$$\begin{aligned} \|w\,(f - \mathcal{I}_n[f])\,\|_\infty &\leq \|w\,(f - p_n)\,\|_\infty + \|w\,\mathcal{I}_n[p_n - f]\|_\infty \\ &\leq \|w\,(f - p_n)\,\|_\infty \,(1 + \mathbb{L}_n)\,, \end{aligned} \tag{6.3}$$

where the quantity

$$\mathbb{L}_n := \sup_{x \in \mathbb{R}} \left\{ \sum_{k=0}^{n} \frac{w(x)|l_{n,k}(x)|}{w(x_k)} \right\} \tag{6.4}$$

is called the Lebesgue constant. In contrast to the case of unweighted Lagrange interpolation on a bounded domain, here the Lebesgue constant explicitly involves the weight function w.

In the inequality (6.3), we may take the infimum over all $p_n \in \mathbb{P}_n$, to see that the Lebesgue constant relates the error in interpolation to the best approximation error by a polynomial in $\mathbb{P}_n$:

$$\|w\,(f - \mathcal{I}_n[f])\,\|_\infty \leq (1 + \mathbb{L}_n) \inf_{p_n \in \mathbb{P}_n} \|w\,(f - p_n)\,\|_\infty. \tag{6.5}$$

Thus, we see that the problem of constructing a stable and accurate Lagrange interpolant consists in the construction of a set of interpolation points for which $\mathbb{L}_n$ does not grow too quickly.

6.1.1 Our contribution

In this work we prove the following result:

Theorem 6.1. *Let $\alpha > 1$ and assume $w : \mathbb{R} \to [0,1]$ is a weight function of the following form*

$$w(x) = \exp(-Q(x)), \quad \textit{with} \quad Q(x) = |x|^\alpha, \quad x \in \mathbb{R}. \tag{6.6}$$

Then the Lebesgue constant for the weighted Leja sequence (1.4), defined on $\mathbb{R}$, grows subexponentially with respect to the number of interpolation points n , i.e.,

$$\lim_{n\to\infty} (\mathbb{L}_n)^{1/n} = \lim_{n\to\infty} \left(\sup_{x\in\mathbb{R}} \left\{ \sum_{k=0}^{n} \left| \frac{w(x) \prod_{\substack{j=0\\ j\neq k}}^{n} (x - x_j)}{w(x_k) \prod_{\substack{j=0\\ j\neq k}}^{n} (x_k - x_j)} \right| \right\} \right)^{1/n} = 1.$$

The rest of this chapter is devoted to the proof of Theorem 6.1. Similar to the case of unweighted Leja points [90, 91], in §6.2, we explore the connection between polynomials and weighted potentials, and show how classical weighted potential theory can be used to understand the asymptotic behavior (with respect to n) of an n^{th} degree polynomial with roots at the contracted Leja points. While these techniques give us most of the result, the final part of the proof requires an explicit estimate on the spacing of the weighted Leja nodes, which is developed in §6.3. Finally, in §6.4, we combine the spacing result and weighted potential theory to complete the proof of Theorem 6.1.

6.2 Weighted Potential Theory

In this section, we state some necesary definitions and results from weighted potential theory, which will be the main tools we use to prove Theorem 6.1. For more details, we refer the interested reader to [82]. The class of weights used in this chapter, defined in (6.6), are a subset of the well-studied *Freud weights* [61]. From (6.6), note first that we may extend Q to be a function on $\mathbb{C}$, and that w has the following properties:

1. The extended weight function $w : \mathbb{C} \to [0,1]$ is continuous in $\mathbb{C}$.

2. The set $\Sigma_0 := \{x \in \mathbb{R} \,|\, w(x) > 0\}$ has positive capacity, i.e.,

$$\mathrm{cap}(\Sigma_0) = \sup\{\mathrm{cap}(K) : K \subseteq \Sigma_0, K \text{ compact}\} > 0,$$

where

$$\operatorname{cap}(K) = \exp\left(\inf\left\{\int_K\int_K \log|x-t|\,d\mu(x)d\mu(t) : \mu \in \mathcal{M}(K)\right\}\right).$$

3. The limit $|x|w(x) \to 0$ as $|x| \to \infty$, $x \in \mathbb{R}$.

In the language of weighted potential theory, these properties imply that w is *admissible.*

Furthermore, we also define the Mhaskar-Rhamanov-Saff number $a_n = a_n(w)$, as the unique solution to the equation (see [82, Corollary IV.1.13]):

$$n = \frac{1}{\pi}\int_{-a_n}^{a_n} \frac{xQ'(x)}{\sqrt{a_n^2 - x^2}}\,dx. \tag{6.7}$$

This number a_n has a few special properties which we use in the following analysis. First, the weighted sup-norm of an n^{th} degree polynomial on $\mathbb{R}$ is realized on the compact set $[-a_n, a_n]$, i.e., for all $p_n \in \mathbb{P}_n$,

$$\|p_n w\|_\infty = \sup_{|x|\le a_n} |p_n(x)|w(x), \tag{6.8}$$

and $|p_n(x)|w(x) < \|p_n w\|_\infty$ for $|x| > a_n$ [82]. Second, from [61, p. 27], $a_n \to \infty$ at approximately the rate $n^{1/\alpha}$, i.e.,

$$a_n \sim n^{1/\alpha}. \tag{6.9}$$

Here, and in what follows, for two sequences a_n, b_n, we write $a_n \sim b_n$ if and only if there exist constants $C_1, C_2 > 0$, independent of n, such that $C_1 \le \frac{a_n}{b_n} \le C_2$.

Let $\mathcal{M}(\mathbb{R})$ be the collection of all positive unit Borel measures μ with $\operatorname{Supp}(\mu) \subseteq \mathbb{R}$. For $\mu \in \mathcal{M}(\mathbb{R})$ and $x, t \in \mathbb{R}$, define the weighted energy integral

$$\begin{aligned} I_w(\mu) &= \int\int \log\left(|x-t|\,w(x)w(t)\right)^{-1}\,d\mu(x)d\mu(t) \\ &= \int\int \log\frac{1}{|x-t|}\,d\mu(x)d\mu(t) + 2\int Q\,d\mu. \end{aligned}$$

We also define the logarithmic potential by

$$U^{\mu}(x) := \int \log \frac{1}{|x-t|}\, d\mu(t). \tag{6.10}$$

The goal of weighted potential theory is to find and analyze the measure $\mu \in \mathcal{M}(\mathbb{R})$ that minimizes the weighted energy integral $I_w(\mu)$. The following theorem may be found in general form in [82, Theorem I.1.3], and is presented here for the specific case (6.6) of a continuous, admissible weight w on $\mathbb{R}$.

Theorem 6.2. *Let w be a continuous, admissible weight function on $\mathbb{R} \subset \mathbb{C}$, and define*

$$V_w := \inf \left\{ I_w(\mu) \,\middle|\, \mu \in \mathcal{M}(\mathbb{R}) \right\}. \tag{6.11}$$

Then we have the following properties:

- *The quantity V_w is finite.*
- *There exists a unique measure $\mu_w \in \mathcal{M}(\mathbb{R})$ such that*

  $$I_w(\mu_w) = V_w,$$

  and the equilibrium measure μ_w has finite logarithmic energy, i.e.,

  $$-\infty < \int\int \log \frac{1}{|x-t|}\, d\mu_w(t) d\mu_w(x) = \int U^{\mu_w}(x)\, d\mu_w(x) < \infty.$$

- *Let F_w be the modified Robin constant for w, given by*

  $$F_w := V_w - \int Q\, d\mu_w. \tag{6.12}$$

  The logarithmic potential U^{μ_w} is continuous for $z \in \mathbb{C}$ and, moreover, for every $x \in$ Supp$(\mu_w) \subset \mathbb{R}$,

  $$U^{\mu_w}(x) + Q(x) = F_w. \tag{6.13}$$

Proof. The first two statements are quoted directly from, and proved in, [82, Theorem I.1.3]. To prove the third statement, we note that $\mathbb{C} \setminus \mathbb{R}$ has exactly two connected components, namely $\{\mathrm{Im}(z) > 0\}$ and $\{\mathrm{Im}(z) < 0\}$, and that of course every point in $\mathrm{Supp}(\mu_w) \subset \{\mathrm{Im}(z) = 0\}$ is a boundary point for both of these sets. Thus, by [82, Theorem I.5.1], U^{μ_w} is continuous on $\mathrm{Supp}(\mu_w)$. Hence, from [82, Theorem I.4.4], U^{μ_w} is continuous on all of $\mathbb{C}$, and (6.13) holds for every $x \in \mathrm{Supp}(\mu_w) \subset \mathbb{R}$. □

6.2.1 Weighted Fekete Points

In this section we describe the connection between Leja points and the weighted equilibrium measure μ_w. For $n \geq 0$, let $\mathcal{T}_n$ denote a general set of points in $\mathbb{R}$ with cardinality $|\mathcal{T}_n| = n+1$, and let w be an admissible weight on $\mathbb{R}$. We say a set of $n+1$ points is (weighted-)Fekete if it maximizes the quantity:

$$\mathcal{F}_n = \underset{|\mathcal{T}_n|=n+1}{\arg\max} \left(\prod_{\substack{t,s\in\mathcal{T}_n \\ t\neq s}} |t-s| w(t) w(s) \right)^{\frac{2}{(n+1)(n+2)}} . \tag{6.14}$$

It is known that the Lebesgue constant for a set of Fekete points $\mathcal{F}_n$ satisfies

$$\mathbb{L}(\mathcal{F}_n) := \sup_{x\in\mathbb{R}} \sum_{s\in\mathcal{F}_n} \left| \frac{w(x) \prod_{t\neq s}(x-t)}{w(s) \prod_{t\neq s}(s-t)} \right| \leq n+1.$$

Furthermore, we also know that for a sequence of Fekete point sets, $\{\mathcal{F}_n\}_{n\geq 1}$,

$$\lim_{n\to\infty} \left(\prod_{\substack{t,s\in\mathcal{F}_n \\ t\neq s}} |t-s| w(t) w(s) \right)^{\frac{2}{(n+1)(n+2)}} = \exp(-V_w),$$

where V_w, as defined in (6.11), is the weighted logarithmic capacity for $\mathbb{R}$ with respect to w. For interpolation schemes, we are also interested in arrays of points with similar asymptotic properties to Fekete points in the limit as $n \to \infty$, since this is a necessary condition for a sequence of points to have a well-behaved Lebesgue constant. Thus, we make the following definition:

Definition 6.3. *A sequence of point sets $\{\mathcal{T}_n\}_{n\geq 1}$, with $|\mathcal{T}_n| = n$, $n \geq 1$, is called asymptotically (weighted) Fekete if*

$$\lim_{n\to\infty} \left(\prod_{\substack{t,s\in\mathcal{T}_n \\ t\neq s}} |t-s|w(t)w(s) \right)^{\frac{2}{(n+1)(n+2)}} = \exp(-V_w).$$

Note that a sequence of interpolation points may be *asymptotically* Fekete but not Fekete, i.e., without satisfying (6.14) for any $n \in \mathbb{N}$. The following lemma, first proved in [37] in a more general setting than the one considered here, and later in [68], indicates that the contracted Leja sequence distributes asymptotically like the Fekete points.

Lemma 6.3.1. *The contracted Leja sequence, defined by* (1.4) *and* (1.5) *is asymptotically Fekete.*

Next we define the discrete point-mass measure associated with the points $\mathcal{T}_n$ as

$$\nu_{\mathcal{T}_n} = \frac{1}{n+1} \sum_{t\in\mathcal{T}_n} \delta_{\{t\}},$$

where $\delta_{\{t\}}$ is the standard Dirac delta function for the point $t \in \mathcal{T}_n$. If a sequence of measures $\{\nu_{\mathcal{T}_n}\}_{n\geq 0}$ corresponds to an asymptotically Fekete sequence of interpolation nodes, the next lemma tells us that they converge to a particular measure; see [37, Theorem 2.3], and [68, Theorem 3.1].

Lemma 6.3.2. *Let μ_w be the equilibrium measure for $\mathbb{R}$ with respect to w (see Theorem 6.2), and let $\{\mathcal{T}_n\}_{n\geq 0}$ be an asymptotically Fekete sequence of point sets with corresponding discrete measures $\{\nu_{\mathcal{T}_n}\}_{n\geq 0}$. Then we have*

$$\lim_{n\to\infty} \nu_{\mathcal{T}_n} = \lim_{n\to\infty} \frac{1}{n+1} \sum_{t\in\mathcal{T}_n} \delta_{\{t\}} = \mu_w,$$

where equality is understood in the weak sense. In particular, for the measures μ_n, defined by* (1.6)*, corresponding to the contracted Leja sequence,*

$$\lim_{n\to\infty} \mu_n = \mu_w.$$

6.2.2 Potentials and Polynomials

Taken together, the previous two lemmas tell us that the discrete point-mass measures associated with the contracted Leja sequence converge weak* to the weighted equilibrium measure for $\mathbb{R}$ corresponding to the weight w given in (6.6). This fact enables us to make a key connection between potential theory and Leja points, and provides the basis for the proof of Theorem 6.1.

With $\{x_{n,j}\}_{j=0}^{n}$ as in (1.5), define $P_{n,k}$ to be the polynomial with roots at each of the n contracted Leja points $x_{n,j}, j = 0, \dots, k-1, k+1, \dots, n$, i.e.,

$$P_{n,k}(x) = \prod_{\substack{j=0\\ j\neq k}}^{n} (x - x_{n,j}),$$

and let $\mu_{n,k}$ be the measure which assigns mass $\frac{1}{n}$ to each of the roots of $P_{n,k}$, i.e.,

$$\mu_{n,k} = \frac{1}{n} \sum_{\substack{j=0\\ j\neq k}}^{n} \delta_{\{x_{n,j}\}}. \tag{6.15}$$

Then, taking the logarithm of $|P_{n,k}^{1/n} w|$, we convert the polynomial into a discrete logarithmic potential with respect to the measure $\mu_{n,k}$, i.e.,

$$\begin{aligned}\log |P_{n,k}(x) w(x)^n|^{1/n} &= \frac{1}{n} \sum_{\substack{j=0\\ j\neq k}}^{n} \log |x - x_{n,j}| - Q(x)\\ &= -U^{\mu_{n,k}}(x) - Q(x).\end{aligned}$$

By Lemma 6.3.1, the weighted Leja sequence is asymptotically Fekete, and therefore we have $\mu_{n,k} \to \mu_w$ in the weak* sense. This connections allows us to exploit potential theory to understand the asymptotic behavior of weighted polynomials. In particular, by considering polynomials with roots at the contracted Leja points (1.5), we explicitly explore this asymptotic behavior in the following two lemmas, which will be an essential part of the proof of Theorem 6.1.

Lemma 6.3.3. *Given $\varepsilon > 0$, there exists an $N \in \mathbb{N}$ such that, for $n > N$ and $0 \leq k \leq n$,*

$$\left| \|P_{n,k} w^n\|_\infty^{1/n} - \exp\left(-F_w\right) \right| < \varepsilon.$$

Proof. First, [82, Theorem I.3.6] implies that for all n and $0 \leq k \leq n$,

$$\|P_{n,k} w^n\|_\infty \geq \exp(-nF_w).$$

This yields $\liminf_{n\to\infty} \|P_{n,k} w^n\|_\infty^{1/n} \geq \exp(-F_w)$ independently of k. In the remainder of the proof, we seek to show that
$\limsup_{n\to\infty} \|P_{n,k} w^n\|_\infty^{1/n} \leq \exp(-F_w)$.

Now let $\varepsilon > 0$ be given. We will seek to show that there exists an N such that for $n > N$, and for all $0 \leq k \leq n$,

$$\sup_{x\in\mathbb{R}} \left\{ \frac{1}{n} \log |P_{n,k}(x)| - Q(x) \right\} \leq -F_w + \epsilon.$$

Define $K_w := \mathrm{Supp}(\mu_w)$. Because of (6.33), we know that for our weight function

$$\sup_{x\in\mathbb{R}} \left\{ \frac{1}{n} \log |P_{n,k}(x)| - Q(x) \right\} = \sup_{x\in K_w} \left\{ \frac{1}{n} \log |P_{n,k}(x)| - Q(x) \right\},$$

and from (6.13), we have the relation

$$U^{\mu_w}(x) + Q(x) = F_w, \quad \forall x \in K_w \subset \mathbb{R}.$$

Hence we can write

$$\sup_{x\in K_w} \left\{ \frac{1}{n} \log |P_{n,k}(x)| - Q(x) \right\} = -F_w + \sup_{x\in K_w} \left\{ -U^{\mu_{n,k}}(x) + U^{\mu_w}(x) \right\}. \tag{6.16}$$

Let $\delta > 0$, to be chosen later. We rewrite the arguments of the supremum on the right-hand side of (6.16) as integrals and divide them each into two parts:

$$\begin{aligned} -U^{\mu_{n,k}}(x) + U^{\mu_w}(x) = &\int_{|x-t|\geq\delta} \log|x-t|\, d\mu_{n,k}(t) - \int_{|x-t|\geq\delta} \log|x-t|\, d\mu_w(t) \\ &+ \int_{|x-t|<\delta} \log|x-t|\, d\mu_{n,k}(t) - \int_{|x-t|<\delta} \log|x-t|\, d\mu_w(t). \end{aligned}$$

First, for $\delta < 1$, clearly

$$\int_{|x-t|<\delta} \log|x-t|\, d\mu_{n,k}(t) = \sum_{\substack{j\neq k \\ |x-x_{n,j}|<\delta}} \log|x-x_{n,j}| \leq 0. \tag{6.17}$$

To deal with the other pieces, first define the function $\chi_\delta(t;x)$ to be the indicator function for the set $K_w \setminus B(x,\delta)$, where $B(x,\delta)$ is the ball of radius δ about x. We claim that for fixed $\delta > 0$, the function

$$g(x) := \int_{B(x,\delta)} \log|x-t|\, d\mu_w(t),$$

is continuous. To see this, let $f_\delta(t;x) := \chi_\delta(t;x)\log|x-t|$. Then,

$$g(x) := \int_{B(x,\delta)} \log|x-t|\, d\mu_w(t) = U^{\mu_w}(x) - \int_{K_w} f_\delta(t;x) d\mu_w(t).$$

The first function on the right-hand side is continuous by Theorem 6.2. To see that the latter is continuous, let $\{y_n\}_{n=1}^\infty \subset K_w$ be a sequence converging to x. Then as $y_n \to x$, $f_\delta(t;y_n)$ converges to $f_\delta(t;x)$, and $|f_\delta(t;y_n)| \leq \max\{\log(\operatorname{diam} K_w), \log\frac{1}{\delta}\}$. Hence, by the bounded convergence theorem, $g(y_n) \to g(x)$.

Since the support of the measure μ_w is compact, we know the function $\log|x-t|$ is uniformly bounded above for $x,t \in K_w$. As $\delta \to 0$, $f_\delta(t;x)$ is a decreasing sequence of integrable functions, which converge pointwise almost everywhere to $\log|x-t|$. Hence by the monotone convergence theorem, as $\delta \to 0$,

$$\int_{K_w} f_\delta(t;x)\, d\mu_w(t) \to \int_{K_w} \log|x-t|\, d\mu_w(t).$$

Hence, for any x, there exists a $1 > \delta_x > 0$ such that

$$-\int_{|x-t|<\delta_x} \log|x-t|\,d\mu_w(t) = \int_{K_w} f_{\delta_x}(t;x)\,d\mu_w(t) - \int_{K_w} \log|x-t|\,d\mu_w(t) \leq \varepsilon/4. \tag{6.18}$$

Furthermore, by the continuity argument in the previous paragraph, we can choose an $r_x < \delta_x$ so that for any $y \in K_w$ with $|y-x| < r_x$,

$$\left| \int_{|y-t|<\delta_x} \log|y-t|\,d\mu_w(t) - \int_{|x-t|<\delta_x} \log|x-t|\,d\mu_w(t) \right| \leq \varepsilon/4. \tag{6.19}$$

Again by compactness, we can cover K_w by some finite set $\{B(y_i, r_{y_i})\}_{i=1}^{M}$. Moreover, there exists a $\delta > 0$ such that for any $x \in K_w$, $B(x,\delta) \subset B(y_i, r_{y_i})$ for some $i = 1, \ldots, M$. This will be the chosen δ. Indeed, from (6.18) and (6.19), and by $\delta < r_{y_i} < \delta_{y_i}$,

$$\begin{aligned} -\int_{|x-t|<\delta} \log|x-t|\,d\mu_w(t) &\leq -\int_{|x-t|<\delta_{y_i}} \log|x-t|\,d\mu_w(t) \\ &\leq -\int_{|y_i-t|<\delta_{y_i}} \log|y_i-t|\,d\mu_w(t) + \varepsilon/4 \\ &\leq \varepsilon/2. \end{aligned} \tag{6.20}$$

Finally we deal with the remaining integrals in (6.16). For any x, the function $\log|x-t|$ is continuous on the set $|x-t| > \delta$. The fact that $\mu_{n,k} \to \mu_w$ weak* implies by definition that there exists an $N_1 \in \mathbb{N}$, such that if $n > N_1$,

$$\int_{|x-t|\geq\delta} \log|x-t|\,d\mu_{n,k}(t) \leq \int_{|x-t|\geq\delta} \log|x-t|\,d\mu_w(t) + \varepsilon/4.$$

Moreover, for any non-negative integers $k_1 \neq k_2$, we find for some $C > 0$,

$$\begin{aligned}\Big| \int_{|x-t|\geq\delta} \log|x-t|\, d\mu_{n,k_1}(t) - \int_{|x-t|\geq\delta} \log|x-t|\, d\mu_{n,k_2}(t) \Big| \\ = \left| \frac{1}{n} \sum_{\substack{j\neq k_1 \\ |x-x_{n,j}|\geq\delta}} \log|x-x_{n,j}| - \frac{1}{n} \sum_{\substack{j\neq k_2 \\ |x-x_{n,j}|\geq\delta}} \log|x-x_{n,j}| \right| \\ \leq \frac{1}{n} \left| \log\left(\frac{\operatorname{diam}(K_w)}{\delta} \right) \right|.\end{aligned}$$

The right-hand side is small as $n \to \infty$, so we can choose $N_2 > N_1$ such that for $n > N_2$, and $0 \leq k_1, k_2 \leq n$,

$$\left| \int_{|x-t|\geq\delta} \log|x-t|\, d\mu_{n,k_1}(t) - \int_{|x-t|\geq\delta} \log|x-t|\, d\mu_{n,k_2}(t) \right| < \varepsilon/4.$$

This implies that for $n > N_2$ and $0 \leq k \leq n$,

$$\int_{|x-t|\geq\delta} \log|x-t|\, d\mu_{n,k}(t) \leq \int_{|x-t|\geq\delta} \log|x-t|\, d\mu_w(t) + \varepsilon/2. \tag{6.21}$$

Furthermore, by compactness of K_w, using standard arguments we can also choose $N > \max\{N_1, N_2\}$ to be independent of x. See [Taylor, Lemma 2.4.12].

Combining (6.17), (6.20), and (6.21) with (6.16) yields the desired result. □

Lemma 6.3.4. *For all $\varepsilon > 0$, there exist $\delta > 0$ and $N \in \mathbb{N}$, such that for $n > N$, and $0 \leq k \leq n$,*

$$\left| \left(w(x_{n,k})^n \prod_{|x_{n,k}-x_{n,j}|\geq\delta} |x_{n,k}-x_{n,j}| \right)^{1/n} - \exp(-F_w) \right| < \varepsilon.$$

Proof. Let $\varepsilon > 0$ be given, and $K_w = \operatorname{Supp}(\mu_w)$ as above. To prove the lemma, it will be enough to show that

$$\left| \log \left(w(x_{n,k})^n \prod_{|x_{n,k}-x_{n,j}|\geq\delta} |x_{n,k}-x_{n,j}| \right)^{1/n} - (-F_w) \right| < \varepsilon.$$

First, notice that

$$\log\left(\prod_{|x_{n,k}-x_{n,j}|\geq\delta}|x_{n,k}-x_{n,j}|\right)^{1/n}=\int_{|t-x_{n,k}|\geq\delta}\log|t-x_{n,k}|\;d\mu_{n,k}(t),$$

and of course

$$\log\left(w(x_{n,k})^n\right)^{1/n}=-Q(x_{n,k}).$$

Furthermore, we have already seen from (6.13) that

$$U^{\mu_w}(x)+Q(x)=F_w,\quad \forall x\in K_w\subset\mathbb{R}. \tag{6.22}$$

Thus, we estimate

$$\begin{aligned}&\left|\log\left(w(x_{n,k})^n\prod_{|x_{n,k}-x_{n,j}|\geq\delta}|x_{n,k}-x_{n,j}|\right)^{1/n}-(-F_w)\right|\\&\quad=\left|\log\left(w(x_{n,k})^n\prod_{|x_{n,k}-x_{n,j}|\geq\delta}|x_{n,k}-x_{n,j}|\right)^{1/n}+U^{\mu_w}(x_{n,k})+Q(x_{n,k})\right|\\&\quad\leq\underbrace{\left|\int_{|t-x_{n,k}|\geq\delta}\log|t-x_{n,k}|\;d\mu_{n,k}(t)-\int_{|t-x_{n,k}|\geq\delta}\log|t-x_{n,k}|\;d\mu_w(t)\right|}_{A}\\&\qquad\qquad+\underbrace{\left|\int_{|t-x_{n,k}|<\delta}\log|t-x_{n,k}|\;d\mu_w(t)\right|}_{B}+\left|-Q(x_{n,k})+Q(x_{n,k})\right|.\end{aligned}$$

The last term is equal to zero, so it is left to show that there exists a $\delta>0$ and $N\in\mathbb{N}$ independent of n and k such that $A<\varepsilon/2$ and $B<\varepsilon/2$. The proof for the quantity A is shown in the proof of Theorem 6.3.3, and the proof for B follows essentially from the proof of [90, Theorem 2.4.6], so we forgo the details here. □

6.3 Spacing of the weighted Leja points

The goal of this section is to state and prove a result regarding the spacing of the contracted Leja sequence. This will be crucial to the final step in the proof of Theorem 6.1.

Theorem 6.4. *Let w and $\alpha > 1$ be as in (6.6), and let $n \in \mathbb{N}$, with $0 \leq i, j \leq n$. Then, for some constant $C > 0$, independent of n, the contracted Leja sequence (1.5) satisfies the spacing property*

$$C|x_{n,i} - x_{n,j}| \geq n^{-1}. \tag{6.23}$$

To prove Theorem 6.4, the main spacing result for the contracted Leja sequence, we use a weighted version of the classical Markov-Bernstein inequalities, which relate norms of polynomials to norms of their derivatives. First, for a_n and Q as defined in (6.7) and (6.6), respectively, define the function

$$\varphi_n(t) = \frac{|t - a_{2n}||t + a_{2n}|}{n\sqrt{(|t + a_n| - a_n\zeta_n)(|t - a_n| + a_n\zeta_n)}}, \tag{6.24}$$

where

$$\zeta_n = (\alpha n)^{-2/3}.$$

Remark 6.5. *The function φ_n plays the same role as the function*

$$\phi_n(t) = \frac{1}{n\sqrt{1 - t^2}},$$

for the Markov-Bernstein inequalities for unweighted polynomials on $[-1, 1]$.

Proof of Theorem 6.4. Let φ be as in (6.24). The main fact we need for this proof is a Bernstein-type inequality for weighted polynomials, which can be found, for instance, in [61, Theorem 10.1]: for any polynomial p_n of degree $n \geq 1$, there exists some C, independent of p_n and n, such that

$$|(p_n(t)w(t))'| \leq \frac{C}{\varphi_n(t)}\|p_n w\|_\infty, \quad t \in \mathbb{R}. \tag{6.25}$$

From [61, Theorem 5.4(b)], we estimate that

$$\sup_{t\in[-a_n,a_n]}\left|\frac{1}{\varphi_n(t)}\right| \sim \sqrt{\alpha}\frac{n}{a_n}.$$

Hence, for any polynomial p_n of degree n, and $t \in \mathbb{R}$,

$$|(p_n(t)w(t))'| \leq C\frac{n}{a_n}\|p_n w\|_\infty. \tag{6.26}$$

In particular, this holds for the polynomial P_n defined by

$$P_n(t) := \prod_{j=0}^{n-1}(t - x_j). \tag{6.27}$$

Given $0 \leq j < n$, by the mean value theorem, there exists a point t between x_j and x_n such that

$$\frac{|P_n(x_j)w(x_j) - P_n(x_n)w(x_n)|}{|x_n - x_j|} = |(P_n(t)w(t))'|\,.$$

Notice that for $0 \leq j < n$, $P_n(x_j) = 0$ by definition. Then from (6.26),

$$\frac{|P_n(x_n)w(x_n)|}{|x_n - x_j|} \leq \frac{Cn}{a_n}\,|P_n(x_n)w(x_n)|\,,$$

which implies

$$C|x_n - x_j| \geq \frac{a_n}{n}.$$

Using the fact $a_n \sim n^{1/\alpha}$ from (6.9), we get

$$C|x_n - x_j| \geq \frac{a_n}{n} \sim n^{1/\alpha - 1}. \tag{6.28}$$

Let $n \geq 1$, and $j < n$, such that $x_{n,j}, x_{n,n} \geq 0$. Then using (6.28), along with (1.5), we calculate

$$C|x_{n,n} - x_{n,j}| = Cn^{-1/\alpha}|x_n - x_j| \geq n^{-1/\alpha}n^{1/\alpha-1} = n^{-1}.$$

Now let $i, j \leq n$, and assume without loss of generality that $i < j$. The above calculation shows that

$$2C|x_{n,i} - x_{n,j}| \geq j^{-1} \geq n^{-1}.$$

which, up to constants independent of n, is the desired result. □

6.4 Proof of Theorem 6.1

In this section, we prove our main theorem concerning the growth of the Lebesgue constant of the weighted Leja sequence. Similar to the proof in the unweighted case given in [90, 91], we separate the proof of the theorem into several smaller components.

To begin, we first show that the Lebesgue constant of the weighted Leja sequence on the real line is equal to a weighted Lebesgue constant of the contracted Leja sequence (1.5) on a fixed compact set. To do this, we first use the fact from (6.8) that supremum a w-weighted, n^{th} degree polynomial is realized in the compact set $[-a_n, a_n]$. Then, we exploit the specific form (6.6) of our weight function to show that

$$Q(n^{1/\alpha}x) = nQ(x), \tag{6.29}$$

which in turn implies that

$$w(x) = w(n^{-1/\alpha}x)^n. \tag{6.30}$$

Finally, let $0 < c < \infty$ be the smallest constant such that

$$\sup_n n^{-1/\alpha} a_n \leq c. \tag{6.31}$$

Note that $c < \infty$ by (6.9). Now defining $K := [-c, c]$, this means that

$$y \in [-a_n, a_n] \implies x := n^{-1/\alpha} y \in K. \tag{6.32}$$

Furthermore, for any $n = 1, 2, \ldots$, let $q_n \in \mathbb{P}_n$. Define $\widetilde{q}_n(x) \in \mathbb{P}_n$ to be the unique polynomial such that

$$q_n(x) = n^{-n/\alpha} \widetilde{q}_n(n^{1/\alpha} x)$$

Then we calculate

$$\begin{aligned}
\sup_{x \in \mathbb{R}} w(x)^n \left| q_n(x) \right| &= n^{-n/\alpha} \sup_{x \in \mathbb{R}} w(n^{1/\alpha} x) \left| \widetilde{q}_n(n^{1/\alpha} x) \right| \\
&= n^{-n/\alpha} \sup_{y \in \mathbb{R}} w(y) \left| \widetilde{q}_n(y) \right| \\
&= n^{-n/\alpha} \sup_{y \in [-a_n, a_n]} w(y) \left| \widetilde{q}_n(y) \right| \\
&= \sup_{y \in [-a_n, a_n]} w(n^{-1/\alpha} y)^n \left| q_n(n^{-1/\alpha} y) \right| \\
&\leq \sup_{x \in K} w(x)^n \left| q_n(x) \right| \\
&\leq \sup_{x \in \mathbb{R}} w(x)^n \left| q_n(x) \right|.
\end{aligned}$$

From this string of inequalities we have that for any $n \geq 1$, and $q_n \in \mathbb{P}_n$,

$$\sup_{x \in \mathbb{R}} w(x)^n \left| q_n(x) \right| = \sup_{x \in K} w(x)^n \left| q_n(x) \right|.$$

Then using [82, Corollary III.2.6], we know that $\operatorname{supp}(\mu_w) =: K_w \subseteq K$, and

$$\sup_{x \in K} w(x)^n \left| q_n(x) \right| = \sup_{x \in K_w} w(x)^n \left| q_n(x) \right|. \tag{6.33}$$

Now from the definition (6.8), along with (6.29)–(6.33), we calculate

$$\begin{aligned}
\mathbb{L}_n &= \sup_{x\in\mathbb{R}} \left\{ \sum_{k=0}^{n} \left| \frac{w(x)}{w(x_k)} \left(\prod_{\substack{j=0\\ j\neq k}}^{n} \frac{x-x_j}{x_k-x_j} \right) \right| \right\} \\
&= \sup_{x\in[-a_n,a_n]} \left\{ \sum_{k=0}^{n} \left| \frac{w(x)}{w(x_k)} \left(\prod_{\substack{j=0\\ j\neq k}}^{n} \frac{x-x_j}{x_k-x_j} \right) \right| \right\} \\
&= \sup_{x\in[-a_n,a_n]} \left\{ \sum_{k=0}^{n} \left| \frac{w(n^{-1/\alpha}x)^n}{w(n^{-1/\alpha}x_k)^n} \left(\prod_{\substack{j=0\\ j\neq k}}^{n} \frac{n^{-1/\alpha}(x-x_j)}{n^{-1/\alpha}(x_k-x_j)} \right) \right| \right\} \\
&\le \sup_{y\in K} \left\{ \sum_{k=0}^{n} \left| \frac{w(y)^n}{w(x_{n,k})^n} \left(\prod_{\substack{j=0\\ j\neq k}}^{n} \frac{y-x_{n,j}}{x_{n,k}-x_{n,j}} \right) \right| \right\} \\
&\le n \left\{ \max_{k=0,\dots,n} \left(\frac{\sup_{y\in K_w} \left| w(y)^n \prod_{\substack{j=0\\ j\neq k}}^{n} (y-x_{n,j}) \right|}{w(x_{n,k})^n \prod_{\substack{j=0\\ j\neq k}}^{n} |x_{n,k}-x_{n,j}|} \right) \right\}.
\end{aligned}$$

Thus, to show that this Lebesgue constant grows at a subexponential rate, the above calculation indicates that we only need to show that

$$\lim_{n\to\infty} \left\{ n \left(\max_{k=0,\dots,n} \frac{\sup_{y\in K_w} |w(y)^n \prod_{\substack{j=0\\ j\neq k}}^{n} (y-x_{n,j})|}{w(x_{n,k})^n \prod_{\substack{j=0\\ j\neq k}}^{n} |x_{n,k}-x_{n,j}|} \right) \right\}^{1/n} = 1. \tag{6.34}$$

Of course, $n^{1/n} \to 1$ as $n\to\infty$, so to prove (6.34), it is enough to show that, uniformly in k, the numerator and denominator both converge to $\exp(-F_w)$, i.e.,

$$\lim_{n\to\infty} \sup_{y\in K_w} \left(|w(y)^n \prod_{\substack{j=0\\ j\neq k}}^{n} (y-x_{n,j})| \right)^{1/n} = \exp(-F_w), \tag{6.35}$$

and

$$\lim_{n\to\infty}\left\{w(x_{n,k})^n\prod_{\substack{j=0\\j\neq k}}^{n}|x_{n,k}-x_{n,j}|\right\}^{1/n}=\exp(-F_w), \tag{6.36}$$

with both limits independent of $k = 0, \dots, n$. Recall that F_w was defined explicitly in (6.12), and is called the Robin constant with respect to the weight w.

Let $\delta > 0$, and $k = 0, \dots, n$. To prove (6.36), we split the product into two parts:

$$\begin{aligned}&\prod_{\substack{j=0\\j\neq k}}^{n}|x_{n,k}-x_{n,j}|w(x_{n,k})\\&\qquad=\underbrace{\left(w(x_{n,k})^n\prod_{|x_{n,k}-x_{n,j}|\geq\delta}|x_{n,k}-x_{n,j}|\right)}_{A_1(n,k,\delta)}\underbrace{\left(\prod_{|x_{n,k}-x_{n,j}|<\delta}|x_{n,k}-x_{n,j}|\right)}_{A_2(k,n,\delta)}.\end{aligned}$$

Then we seek to show that as $n \to \infty$ and $\delta \to 0$,

$$A_1(n,k,\delta)^{1/n}\to\exp(-F_w), \tag{6.37}$$

and

$$A_2(n,k,\delta)^{1/n}\to 1, \tag{6.38}$$

and that convergence of the limits is independent of $k = 0, \dots, n$.

We have reduced the proof to essentially a problem in weighted potential theory. The convergence of the limits (6.35) and (6.37) follow directly from Lemmas 6.3.3 and 6.3.4, respectively, which are proven in the appendix. Thus, we have left to show statement (6.38), which requires a more direct approach. We explicitly use the spacing of the contracted Leja sequence from Theorem 6.4, and find that the remainder of the estimate involving $A_2(n,k,\delta)$ follows from this spacing lemma.

By assuming $\delta < 1$, it is clear that the product $A_2(n,k,\delta)$ is always less than one. Therefore, the following theorem will complete the proof of Theorem 6.1.

Lemma 6.5.1. *Given $\varepsilon > 0$, there exists $\delta > 0$, $N \in \mathbb{N}$ such that for $n > N$, and $0 \leq k \leq n$,*

$$\left(\prod_{|x_{n,k}-x_{n,j}|<\delta} |x_{n,k} - x_{n,j}| \right)^{1/n} > 1 - \varepsilon.$$

Proof. We first split the product into two components:

$$\prod_{|x_{n,k}-x_{n,j}|<\delta} |x_{n,k} - x_{n,j}| = \prod_{x_{n,j} \in X_1(k,\delta)} |x_{n,k} - x_{n,j}| \times \prod_{x_{n,j} \in X_2(k,\delta)} |x_{n,k} - x_{n,j}|.$$

where

$$X_1(k,\delta) := \left\{ x_{n,j} \;\middle|\; j \leq n,\, x_{n,k} - \delta < x_{n,j} \leq x_{n,k} \right\},$$
$$X_2(k,\delta) := \left\{ x_{n,j} \;\middle|\; j \leq n,\, x_{n,k} \leq x_{n,j} < \; x_{n,k} + \delta \right\}.$$

At least one of these sets may be empty, and in that case we simply set the corresponding product equal to one. Now, let m_1, m_2 be the cardinality of the sets $X_1(k,\delta)$ and $X_2(k,\delta)$, resp., and label these points in the following way

$$x_{n,k} - \delta \leq x_{n,i_{m_1}} \leq \ldots \leq x_{n,i_1} < x_{n,k} < x_{n,j_1} < \ldots < x_{n,j_{m_2}} < x_{n,k} + \delta.$$

Then from Theorem 6.4, we can show that for any $1 \leq s \leq m_1$,

$$|x_{n,k} - x_{n,i_s}| = |x_{n,k} - x_{n,i_1}| + \ldots + |x_{n,i_{s-1}} - x_{n,i_s}| \geq \frac{s}{Cn}. \tag{6.39}$$

Similarly, for $1 \leq t \leq m_2$,

$$|x_{n,k} - x_{n,j_t}| = |x_{n,k} - x_{n,j_1}| + \ldots + |x_{n,j_{t-1}} - x_{n,j_t}| \geq \frac{t}{Cn}. \tag{6.40}$$

Now, using (6.39) and Sterling's approximation, we see that

$$\begin{aligned}\left(\prod_{x_{n,j}\in X_1(k,\delta)}|x_{n,k}-x_{n,j}|\right)^{1/n} &= \left(\prod_{s=1}^{m_1}|x_{n,k}-x_{n,i_s}|\right)^{1/n} \geq \left(\prod_{s=1}^{m_1}\frac{s}{Cn}\right)^{1/n}\\ &= \left(\frac{m_1!^{\frac{1}{m_1}}}{Cn}\right)^{m_1/n} \geq \left(\frac{m_1}{Cn}\right)^{m_1/n}. \end{aligned} \tag{6.41}$$

Similarly, we can show that

$$\left(\prod_{x_{n,j}\in X_2(k,\delta)}|x_{n,k}-x_{n,j}|\right)^{1/n} \geq \left(\frac{m_2}{Cn}\right)^{m_2/n}. \tag{6.42}$$

As $\tau \to 0^+$, the function $(\frac{\tau}{C})^{2\tau} \to 1$. Thus, we let $\tau < \min\{C, \frac{1}{e}\}$ be small enough so that

$$1-\varepsilon < \left(\frac{\tau}{C}\right)^{2\tau} < 1.$$

Let m be the number of Leja points within the interval $\{t \in \mathbb{R} : |x_{n,k}-t| < \delta\}$. According to [90, Theorem 2.4.5], for our chosen $\tau > 0$, we can choose $N \in \mathbb{N}$, and $\delta_0 > 0$ such that if $n > N$, and $\delta < \delta_0$,

$$\max\left\{\frac{m_1}{n}, \frac{m_2}{n}\right\} \leq \frac{m}{n} = \int_{|t-x_{n,k}|<\delta} d\mu_{n,k}(t) < \tau.$$

We know $f(x) = x^x$ is a decreasing function on $(0, \frac{1}{e})$, and hence from (6.41) and (6.42), this implies that

$$\begin{aligned}\left(\prod_{x_{n,j}\in X_1(k,\delta)}|x_{n,k}-x_{n,j}|\right)^{1/n} & \left(\prod_{x_{n,j}\in X_2(k,\delta)}|x_{n,k}-x_{n,j}|\right)^{1/n}\\ &\geq \left(\frac{m_1}{Cn}\right)^{m_1/n}\left(\frac{m_2}{Cn}\right)^{m_2/n}\\ &\geq \left(\frac{\tau}{C}\right)^{2\tau} > 1-\varepsilon,\end{aligned}$$

which is the desired result for $X_1(k,\delta)$ and $X_2(k,\delta)$. This completes the proof. □

6.5 Remarks

In this chapter, we considered the properties of Leja points for weighted Lagrange interpolation on an unbounded domain. Due to their nested structure, simple recursive formulation, and generally stable behavior, Leja points show promise for high-dimensional interpolation methods. Our contribution to this area was to prove that the Lebesgue constant for the weighted Leja sequence grows subexponentially with respect to the number of interpolation nodes. Furthermore, we proved a theorem regarding the separation of the weighted Leja points.

Of course, a subexponential rate encompasses a wide range of growth, potentially much bigger than the optimal Lebesgue constant $\mathcal{O}(\log n)$. On the other hand, our experience with Leja points indicates that the Lebesgue constant grows linearly, i.e., $\mathcal{O}(n)$, with respect to the number of nodes. Our proof relies on potential theory, which gives only asymptotic estimates of growth. We expect that a more explicit estimate of the Lebesgue constant would require different techniques, and this is the subject of future work.

Chapter 7

Sparse grid quadrature based on conformal mappings

This work has been submitted to the Proceedings of the 3rd Sparse Grids and Applications Workshop, held in October 2016 in Miami, FL.

For functions which are complex analytic in a certain domain containing a compact interval $I \subset \mathbb{R}$, this chapter looks at how we may find better points by transforming classical interpolation sequences under a conformal mappings [49, 58]. We demonstrate the extension of these quadrature approximations, built from conformal mapping of interpolatory rules, to sparse grid quadrature in the multidimensional setting. In one dimension, computation of an integral involving an analytic function using these transformed quadrature rules can improve the convergence rate by factor approaching $\pi/2$ versus classical interpolatory quadrature [49]. This work shows that this $\pi/2$ improvement increases exponentially with the dimension of the underlying integral problem.

The outline of the chapter is as follows. First, we introduce the one-dimensional transformed quadrature rules in §7.1, and in §7.1.2 describe how to use them in the construction of sparse grid quadrature rules for integration of multidimensional functions. In §7.2, we provide a brief analysis of the corresponding mapped method to show that the improvement in the convergence rate to a d-dimensional integral is $(\pi/2)^{1/\xi(d)}$, where $\xi(d)^{-1} \geq d$, and provide numerical tests for the sparse grid transformed quadrature rules in

§7.3. We conclude this chapter with some remarks on the benefits and limitations of the method in §7.4.

7.1 Transformed Quadrature Rules

In this section, we introduce one-dimensional transformed quadrature rules, based on the conformal mappings described in [49], applied to classical polynomial interpolation based rules. These rules will be used as a foundation for sparse tensor product quadrature rules for computing high-dimensional integrals, introduced in later sections.

To begin, suppose we want to integrate a given function f over the domain $[-1, 1]$, and assume this function admits an analytic extension in a region $[-1, 1] \subset \Sigma \subset \mathbb{C}$. Given a set of points $\{x_j\}_{j=1}^n$, an interpolatory quadrature rule is defined from the Lagrange interpolant of f, which is the unique degree $n-1$ polynomial matching f at each of the abcissas x_j, i.e.,

$$L_n[f](x) = \sum_{j=1}^n f(x_j) l_j^n(x), \quad \text{where} \quad l_j^n(x) = \prod_{\substack{i=1 \\ i \neq j}}^n \frac{x - x_i}{x_j - x_i}.$$

The quadrature approximation of the integral of f, denoted $Q_n[f]$, is then defined by

$$\int_{-1}^1 f(x)\, dx \approx \int_{-1}^1 L_n[f](x)\, dx = \sum_{j=1}^n c_j f(x_j) =: Q_n[f], \tag{7.1}$$

with weights given explicitly as

$$c_j = \int_{-1}^1 l_j^n(x)\, dx. \tag{7.2}$$

Now, according to the Cauchy integral theorem, since f has an analytic extension, we can evaluate the integral along any (complex) path contained in Σ with endpoints $\{\pm 1\}$. Next, let g be a conformal mapping satisfying the conditions:

$$g(\pm 1) = \pm 1, \text{ and } g([-1, 1]) \subset \Sigma. \tag{7.3}$$

According to the argument above, the integral can be rewritten as the path integral from -1 to 1, with the path parameterized by the map g, i.e.,

$$\int_{-1}^{1} f(x)\,dx = \int_{-1}^{1} f(g(s))g'(s)\,ds.$$

Applying our original quadrature rule to the latter integral,

$$\int_{-1}^{1} f(g(s))g'(s)\,ds \approx \sum_{j=1}^{n} \underbrace{c_j g'(x_j)}_{:=\tilde{c}_j} f(\underbrace{g(x_j)}_{:=\tilde{x}_j}) =: \widetilde{Q}_n[f], \tag{7.4}$$

we obtain a new quadrature rule with transformed weights $\{\tilde{c}_j\}_{j=1}^n$ and points $\{\tilde{x}_j\}_{j=1}^n$.

Equation (7.4) provides the motivation for the choice of the specific conformal mapping g. Specifically, the Taylor series for f, centered at points $x \in [-1,1]$ which are close to the boundary, may have a radius which extends beyond the largest Bernstein ellipse in which f is analytic. We may then hope to find a g such that a Bernstein ellipse is conformally mapped onto the whole region where f is analytic, where classical convergence theory yields the convergence rate for $f \circ g$. In addition to (7.3), it is especially advantageous to have g map $[-1,1]$ onto itself, i.e.,

$$g([-1,1]) = [-1,1]. \tag{7.5}$$

In this case, the transformed weights and points remain real-valued, and we avoid evaluations of f with complex inputs.

We now turn our attention to several specific conformal mappings which satisfy the conditions (7.3), along with the extra condition (7.5). For more details on the derivation of the maps, see [49]. The first mapping we consider applies to functions which admit an analytic extension at every point on real line; in other words, functions which have only complex singularities. In this case, the natural transformations to consider are ones that conformally map a Bernstein ellipse (3.7) to a strip about the real line. Specifically, we define a map which takes the Bernstein ellipse with shape parameter ρ to the complex strip with half-width $\frac{2}{\pi}(\rho - 1)$, as shown in Figure 7.1. First, given a value for ρ, we define the parameter

$$m^{1/4} = 2\sum_{j=1}^{\infty} \rho^{-4(j-\frac{1}{2})^2} \Bigg/ \left(1 + 2\sum_{j=1}^{\infty} \rho^{-4j^2}\right),$$

and $K = K(m)$ to the be the elliptic parameter corresponding to m; see [32]. Now we define the mapping

$$g_1(z) = \tanh^{-1}\left(m^{1/4}\mathrm{sn}\left(\frac{2K}{\pi \sin^{-1}(z)|m}\right)\right) / \tanh\left(m^{1/4}\right). \tag{7.6}$$

We'll refer to this map as the "*strip map*" in the following.

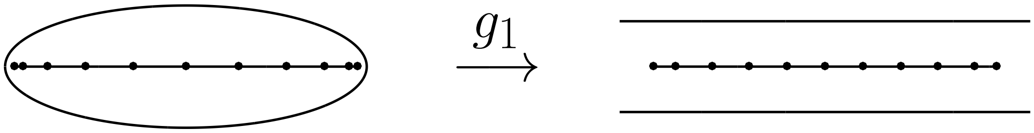

Figure 7.1: The mapping (7.6) takes the Bernstein ellipse $E_{1.4}$ (left) to a strip of half-width $2(1.4-1)/\pi \approx .255$.

According to (7.4), we also need to know the derivative of g_1, given by

$$g_1'(z) = \frac{2Km^{1/4}}{\pi\sqrt{1-z^2}} \frac{\mathrm{cn}(\omega|m)\mathrm{dn}(\omega|m)}{(1 - m^{1/2}\mathrm{sn}(\omega|m))} / \tanh\left(m^{1/4}\right). \tag{7.7}$$

with $\omega = 2K\sin^{-1}(z)/\pi$. For our applications, we also require the values of g_1' at the endpoints of the interval, which are given by

$$g_1'(\pm 1) = 4K^2 m^{1/4}\left(1 + m^{1/2}\right)/\pi^2 \tanh\left(m^{1/4}\right).$$

Another way to change the endpoint clustering, and transform the quadrature rule under a conformal map, is to use an appropriately normalized truncation of the power series for $\sin^{-1}(z)$. The map $\frac{2}{\pi}\sin^{-1}(z)$ perfectly eliminates the clustering of the Gauss–Legendre and Clenshaw–Curtis points, but since it has singularities at ±1, it is useless for our purposes. On the other hand, by considering a *truncation* of the power series

$$\sin^{-1}(z) = \sum_{k=1}^{\infty} \frac{\Gamma(k+1/2)}{\Gamma(1/2)} \frac{z^{2k+1}}{(2k+1)k!},$$

we define a more desirable mapping. To this end, for $M \geq 1$, we define

$$g_2(z) = c(M) \sum_{k=1}^{M} \frac{\Gamma(k+1/2)}{\Gamma(1/2)} \frac{z^{2k+1}}{(2k+1)k!}, \tag{7.8}$$

with an appropriately chosen constant $c(M) < 1$ so that $g_2(\pm 1) = \pm 1$. This mapping is much easier to implement than the previous mapping. We will call this map the "*pill map*", since it maps the Bernstein ellipse to a pill-shaped region about $[-1, 1]$ with flatter sides. In Figure 7.2, we plot the image of the ellipse E_ρ with $\rho = 1.4$, under the mapping (7.8) with $M = 4$. The region on the right has almost flat sides, with width a little bigger than $\frac{2}{\pi}(1.4 - 1) \approx .255$.

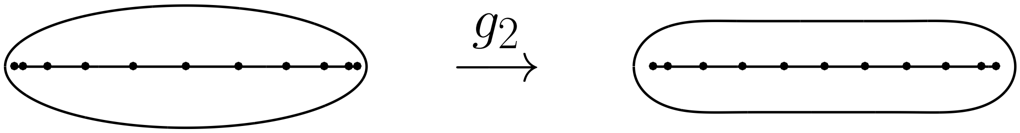

Figure 7.2: The mapping (7.8), with $M = 4$, takes the Bernstein ellipse $E_{1.4}$ (left) to a pill-shaped region with sides of length $\approx .255$.

7.1.1 Standard One-dimensional Quadrature Rules

Here we give a brief summary of some standard interpolatory-type quadrature rules, to which we will apply the mappings of the previous section. Only the nodes are discussed here, as the weights for each method will be defined according to (7.2). For an overview of the theory of interpolatory quadrature, see [96, Ch. 19].

The first quadrature rule is based on the extrema of the Gauss–Chebychev polynomials. For a given number of points n, these are given by:

$$x_j = \cos\left(\frac{(j-1)\pi}{n}\right), \quad 1 \leq j \leq n. \tag{7.9}$$

If we choose the number of nodes to be $n = 2^{i-1} + 1, i > 1$, then they form a nested sequence known as the Clenshaw–Curtis nodes.

Another set of points of interest are the well-known Gaussian abscissa, which are the roots of orthogonal polynomials with respect to a given measure. Here we consider the sequence of Gauss–Legendre nodes, which consists of the roots of the sequence of polynomials orthogonal to the uniform measure on $[-1, 1]$, i.e., the n roots of the polynomials

$$P_n(x) = \frac{d^n}{dx^n}\left[(x^2-1)^n\right], \quad n \geq 0. \tag{7.10}$$

With the introduction of a weight into the integral from (7.1), other families of orthogonal polynomials can be used. The main advantage of Gauss points is their high degree of accuracy, i.e., the one-dimensional quadrature rules built from n Gauss points integrate exactly polynomials of degree $2n-1$. However, Gauss–Legendre points do not form a nested sequence, which may lead to inefficiency in the high-dimensional quadrature setting. In fact, without nestedness of the one-dimensional sequence, the sparse grid rule described in the following section may not even be interpolatory. We also remark that nested quadrature sequences based on the roots of orthogonal polynomials, the so-called Gauss–Patterson points, are also available, but we do not consider these types of rules herein.

The final set of nodes we consider are known as the Leja sequence. Leja points satisfy a recursive definition, that is, given a point $x_1 \in [-1, 1]$, for $n \geq 2$ define

$$x_n = \operatorname*{arg\,min}_{x\in[-1,1]} \prod_{j=1}^{n-1} |x - x_j|, \tag{7.11}$$

where we typically take $x_1 = 0$. Of course, there may be several minimizers to (7.11), so for computational purposes, we simply choose the minimizer closest to the left endpoint. The Leja sequence is typically better suited for high-dimensional interpolation versus quadrature. In the interpolation setting, Leja sequences are known to have good properties for approximation in high-dimensions [68], and there has been much research related to the stability properties of such nodes when used for Lagrange interpolation [91, 55]. In the quadrature setting, the lack of symmetry can sometimes lead to null weights assigned to certain nodes. On the other hand, they have the added benefits of being a nested sequence that grows one point at a time, and have asymptotic distribution which is that same as that of Gauss and Clenshaw–Curtis nodes.

7.1.2 Sparse Quadrature for High Dimensional Integrals

For the numerical approximation of high-dimensional integrals over product domains, it is natural to consider simple tensor products of one-dimensional quadrature rules. Unfortunately, these rules suffer from the curse of dimensionality, as the number of points required to accurately compute the integral grows exponentially with the underlying dimension of the integral; i.e., a rule using n points in each dimension requires n^d points. For certain smooth integrands, we can mitigate this effect by considering sparse combinations of tensor products of these one-dimensional rules, i.e., sparse grid quadrature. It is known that sparse grid rules can asymptotically achieve approximately the same order of accuracy as full tensor product quadrature, but use only a fraction of the number quadrature nodes [38, 72, 71].

Rather than the one-dimensional integral from before, we let $d > 1$ be the dimension and define $\Gamma := [-1,1]^d$. In addition, by letting $\boldsymbol{x} = (x_1, \ldots, x_d)$ be an arbitrary element of Γ, we consider the problem of approximating the integral

$$I^d[f] = \int_\Gamma f(\boldsymbol{x})\, d\boldsymbol{x}, \tag{7.12}$$

using transformed quadrature rules.

We review the construction of sparse grid quadratures here, noting that it is the same as the construction detailed in Section 3.1. To define the sparse grid rules, we first denote by $\{I_{p(l)}\}_{l\geq 1}$ a sequence of given one-dimensional quadrature operators using $p(l)$ points. Here $I_{p(l)}$ may be a standard interpolatory quadrature $Q_{p(l)}$ from (7.1), or its conformally transformed version $\widetilde{Q}_{p(l)}$ from (7.4). With $I_0 := 0$, define the difference operator

$$\Delta_l := I_{p(l)} - I_{p(l-1)}.$$

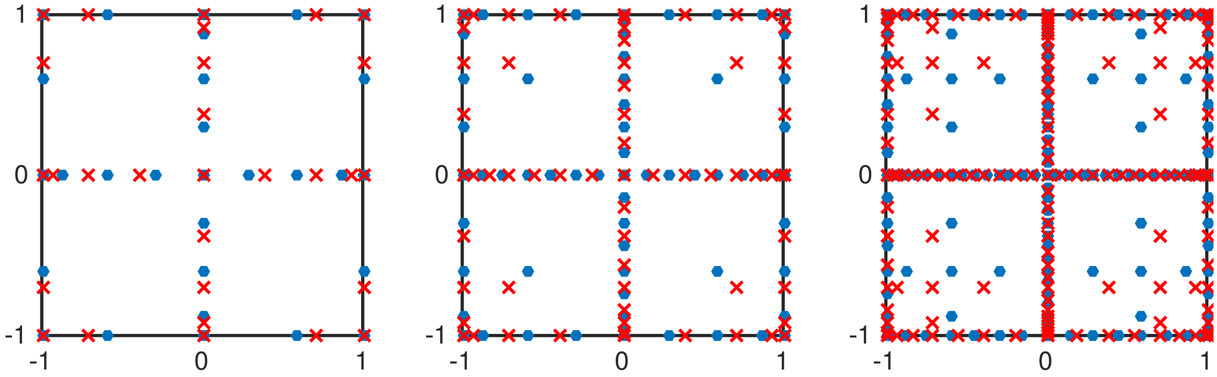

Figure 7.3: Location of the two-dimensional transformed sparse grid nodes (blue dot) using an underlying Clenshaw–Curtis rule, compared to standard Clenshaw–Curtis sparse grids (red x).

Then given a set of multiindices $\Lambda_w \subset \mathbb{N}_0^d$, we define the sparse grid quadrature operator to be

$$I_{N_w}[f] = \sum_{\boldsymbol{l}\in\Lambda_w} \bigotimes_{i=1}^{d} \Delta_{p(l_i)}[f] = \sum_{\boldsymbol{l}\in\Lambda_w} \bigotimes_{i=1}^{d} \left(I_{p(l_i)} - I_{p(l_i-1)}\right)[f], \tag{7.13}$$

where we refer to the natural number w as the *level* of the sparse grid rule, and N_w is the total number of points in Γ used by the sparse grid. The choice of multiindex set Λ_w may vary based on the problem at hand. It may be anisotropic, i.e., dimension dependent, or if appropriate error indicators are defined, it may even be chosen adaptively. Some typical choices are given in Table 3.1, but for simplicity, we consider only standard isotropic Sparse Smolyak grids. For more information on anisotropic rules, see [70].

The effect of the conformal mapping on the placement of the nodes used by the sparse quadrature rule (7.13) is similar to the one-dimensional case. In Figure 7.3, we have plotted the nodes of a two-dimensional Clenshaw–Curtis sparse grid with the transformation map (7.8), using $\rho = 1.4$, versus a traditional Clenshaw–Curtis sparse grid. Note how the clustering of the nodes toward the outer boundary of the cube is diminished.

7.2 Comparison of the transformed sparse grid quadrature method

In this section we investigate the potential improvement in convergence for computation of high-dimensional integrals using the sparse grid quadrature method based on TQ rules. The different mappings (7.6) and (7.8), since they have different properties, will be considered separately. Furthermore, the focus of this section will be on the transformation of Gauss–Legendre rules, though we remark that starting from a one-dimensional convergence result such as the following theorem, the rest of the analysis is similar for the Clenshaw–Curtis case. We begin by quoting the following one-dimensional result stated from [49], establishing the convergence of the transformed Gauss–Legendre rule for an analytic integrand.

Theorem 7.1. *For some $\rho > 1$, let f be analytic and uniformly bounded by $A > 0$ in a region $\Omega_\rho \supset [-1, 1]$. Given a conformal map $g : \Sigma_\rho \to \Omega$ satisfying (7.3), for $n \geq 1$ the transformed Gauss–Legendre quadrature rule (7.4) has the error bound*

$$\left| I[f] - \widetilde{Q}_n[f] \right| \leq \frac{64A\gamma}{15(1-\rho^{-2})}\, \rho^{-2n}, \tag{7.14}$$

where $\gamma = \sup_{s \in \Sigma_\rho} g'(s)$.

Now taking a specific region of analyticity and a given conformal map, Theorem 7.1 may be used to fully quantify the benefit of the transformation method. We start by considering functions analytic in the strip S_ε of half width ε about the real line, and the Gauss–Legendre rule transformed under the map (7.6).

Theorem 7.2 ([49, Theorem 3.1])**.** *Let f be analytic and uniformly bounded by $A > 0$ in a strip S_ε of half width ε about the real line, and g_1 the conformal map (7.6) mapping $E_{1+\frac{\pi}{2}\varepsilon} \to S_\varepsilon$. Then for $n \geq 1$, and any $\tilde{\varepsilon} < \varepsilon$, the transformed Gauss–Legendre quadrature rule has the error bound*

$$\left| I[f] - \widetilde{Q}_n[f] \right| \leq \frac{64A\gamma}{15(1-(1+\pi/2\tilde{\varepsilon})^{-2})} \left(1 + \frac{\pi}{2}\tilde{\varepsilon}\right)^{-2n}, \tag{7.15}$$

where $\gamma = \sup_{s \in S_{\tilde{\varepsilon}}} g_1'(s)$.

Note that we must take $\tilde{\varepsilon} < \varepsilon$, since otherwise the value of γ is infinite for the strip mapping g_1. However, we do not lose much, and this theorem shows that we can achieve savings of almost a factor of $\pi/2$ for functions analytic in a strip S_ε.

For the mapping (7.8), the results are somewhat more complicated, due to the fact that the properties of the map depend crucially on the chosen degree M of the truncation, and for a given M we may not be able to realize the full factor of $\pi/2$. From a practical standpoint, this is not much worse than the case of the strip mapping (7.6), since full information about the analyticity of the integrand may not be available, and hence it may be difficult to tune the parameter of the mapping to the integral at hand. Thus, what we have in the case of the map (7.8) is a more precise result with all the parameters specified. The following result from [49] will apply to functions which are analytic in the ε-neighborhood of $[-1, 1]$, denoted U_ε. Then we have the following theorem.

Theorem 7.3 ([49, Theorem 6.1])**.** *Let $\varepsilon \leq .8$, and let f be analytic and uniformly bounded by $A > 0$ in a ε-neighborhood U_ε of $[-1, 1]$. Let g_2 be the conformal map (7.8), truncated at degree $M = 4$. Then for $n \geq 1$, the transformed Gauss–Legendre quadrature rule has the error bound*

$$\left| I[f] - \widetilde{Q}_n[f] \right| \leq \frac{64A\gamma}{15(1 - (1 + 1.3\varepsilon)^{-2})} (1 + 1.3\varepsilon)^{-2n}, \tag{7.16}$$

where $\gamma = \sup_{s \in S_\varepsilon} g_2'(s)$.

From the one dimensional results of Theorem 7.2 and Theorem 7.3, for the maps (7.6) and (7.8), resp., we are able to fully quantify the benefits of the TQ rules applied to sparse grid quadrature in high dimensions. The following theorems give the convergence rate for a sparse grid quadrature approximation of an analytic integrand based on the Gauss–Legendre points. Recall that we are considering only isotropic sparse Smolyak constructions, according to the last row in Table 3.1.

Corollary 7.3.1. *Let f be analytic in $\prod_{i=1}^d S_\varepsilon$ for some $\varepsilon > 0$, and let g_1 be the conformal mapping (7.6). Then for any $\tilde{\varepsilon} < \varepsilon$, the sparse quadrature (7.13) built from transformed Gauss–Legendre quadrature rules satisfies the following error bound in terms of the number* *of quadrature nodes:*

$$|I^d[f] - I_{N_w}[f]| \le C(\tilde{\varepsilon}, f, \gamma, d)\, \left(1 + \frac{\pi}{2}\tilde{\varepsilon}\right)^{-\frac{2d}{2^{1/d}} N_w^{\xi(d)}}. \tag{7.17}$$

with

$$\xi(d) = \frac{\log(2)}{d(\zeta + \log(d))}. \tag{7.18}$$

Corollary 7.3.2. *For some $0 < \varepsilon \le .8$, let f be analytic in $\prod_{i=1}^d U_\varepsilon$, and let g_2 be the conformal mapping (7.8) truncated at degree $M = 4$. Then the sparse quadrature (7.13) built from transformed Gauss–Legendre quadrature rules satisfies the following error bound in terms of the number of quadrature nodes:*

$$|I^d[f] - I_{N_w}[f]| \le C(\varepsilon, f, \gamma, d)\, (1 + 1.3\varepsilon)^{-\frac{2d}{2^{1/d}} N_w^{\xi(d)}}, \tag{7.19}$$

with ξ as in (7.18).

The proofs of the preceding results are omitted. We mention that from the one dimensional results of Theorem 7.2 and Theorem 7.3, they follow from well-known sparse grid analysis techniques and estimates on the number of quadrature nodes; see, e.g., [71]. We remark again that it is not necessary to use the same ε in each dimension, but we make that choice for clarity of presentation. As mentioned in §7.1.2, in the case that the integrand f has dimension-dependent smoothness, *anisotropic* sparse grid methods are available.

We now make a few remarks on the improvements of Corollary 7.3.1 and Corollary 7.3.2 over sparse grids based on traditional interpolatory quadrature methods. First, note that for functions $f \in C(\Gamma)$ which admit an analytic extension in either $\prod_{i=1}^d S_\varepsilon$ or $\prod_{i=1}^d U_\varepsilon$, the largest (isotropic) polyellipse in which f is analytic has the shape parameter $\rho = 1+\varepsilon$. Hence, the convergence rate of typical sparse grid Gauss–Legendre quadrature, using N abscissa, is

$$|I[f] - \mathcal{I}_N[f]| = \mathcal{O}\left((1+\varepsilon)^{-\frac{2d}{2^{1/d}} N^{\xi(d)}}\right). \tag{7.20}$$

Thus, the improvement in convergence rate is multiplied exponentially in the sparse grid case, i.e., in the case of Corollary 7.3.1, the number of points required to reach a certain tolerance is reduced by a factor approaching $(\pi/2)^{\xi(d)^{-1}}$, with ξ as in (7.18). In other words, let N_{SGTQ} and N_{SG} be the necessary number of points for the right-hand sides of (7.17) and (7.20), respectively, to be less than a given tolerance. Then, we may calculate the limit

$$\lim_{\varepsilon \to 0} \frac{N_{SG}}{N_{SGTQ}} = \left(\frac{\pi}{2}\right)^{\xi(d)^{-1}}. \tag{7.21}$$

The constants are ignored in the calculation, though the transformed quadrature may have slightly improved constant versus the standard case. We also note that $\xi(d)^{-1} \geq d$, so the improvement is exponential in the dimension. In the case of the sparse grid quadrature approximation transformed by (7.8), we use (7.19), so the factor is $1.3^{\xi(d)^{-1}}$ rather than $(\pi/2)^{\xi(d)^{-1}}$. As mentioned in the work [49], the factor of 1.3 is still much less than $\pi/2 \approx 1.57$, but for small ε and large truncation parameter M this can improved to 3/2; see [49, Theorem 6.2].

7.3 Numerical Tests of the Sparse Grid Transformed Quadrature Rules

In this section we test the sparse grid transformed quadrature rules on a number of multidimensional integrals, and compare the performance versus standard rules. The transformed rules we consider are based on the conformal mapping of Gauss–Legendre, Clenshaw–Curtis, and Leja quadrature nodes, which are describe in §7.1.1. We transform these rules using both of the conformal mappings (7.6) and (7.8), using the Matlab code provided in [49] to generate the one-dimensional quadrature sequences. The Tasmanian sparse grid toolkit [87, 89] is used for the implementation of the full sparse grid quadrature rule.

7.3.1 Comparison of Maps

For the first test, we compare the sparse grid methods with the transformed quadratures to traditional quadrature approximations for computing the integral of three test functions over the cube $[-1,1]^3$ in three dimensions. In each case, we compare the different maps (7.6)

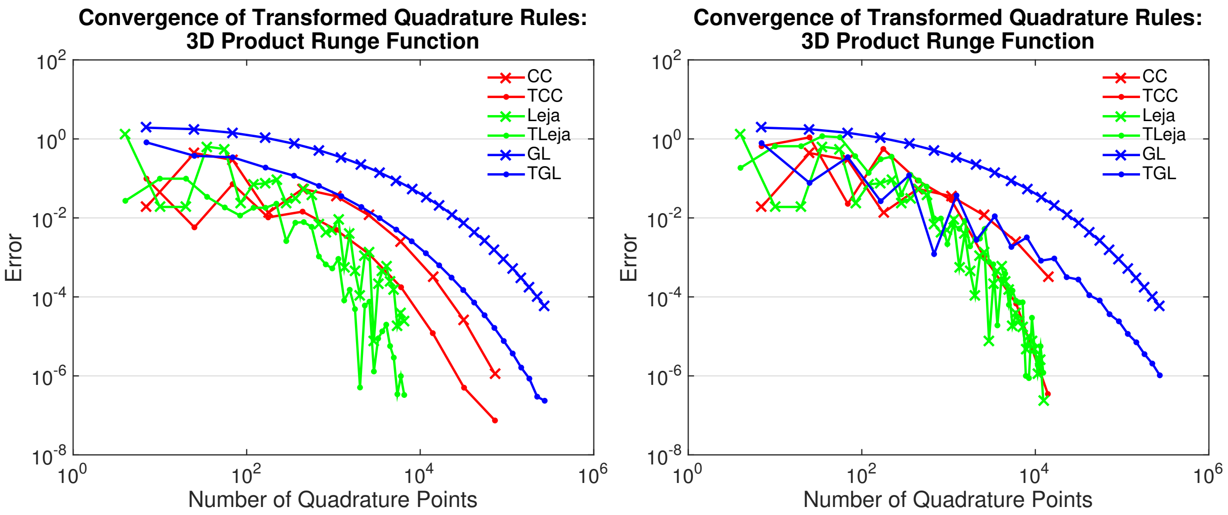

Figure 7.4: Comparison of sparse grid quadrature rules for computing the integral of (7.22) over the cube $[-1,1]^3$, using the conformal maps (7.6) (left), and (7.8) (right).

and (7.8) for the generation of the transformed one-dimensional quadrature from the Clenshaw–Curtis, Gauss–Legendre, and Leja rules. The chosen mapping parameters are $\rho = 1.4$ with (7.6) and truncation parameter $M = 4$ for (7.8).

In Figure 7.4, we plot the results for approximating the integral over $[-1,1]^3$ of the function

$$f(x,y,z) = \frac{1}{(1+5x^2)(1+5y^2)(1+5z^2)}. \tag{7.22}$$

This function has complex singularities at points $\boldsymbol{z} \in \mathbb{C}^3$ where at least one coordinate $z_j = \frac{1}{\sqrt{5}}\mathrm{i}$, and is hence analytic in the complex hyper-strip $\prod_{i=1}^{3} S_{1/\sqrt{5}}$. As expected, the quadrature generated according to the mapping (7.6) performs the best here, though the chosen parameter $\rho = 1.4$ is somewhat less than the optimal, since the value $\frac{2}{\pi}(1.4-1) \approx .255 < 1/\sqrt{5}$. Regardless, the transformed sparse grid approximations again perform better than their classical counterparts, gaining up to two orders of magnitude in the error for Clenshaw–Curtis and Gauss rules. Note that on the right-hand plot, the transformation (7.8) does not work well with the Leja rule. The results for the standard quadrature are repeated in each plot for ease of comparison.

Figure 7.5 again shows the results for approximating the integral of the function

$$f(x,y,z) = \exp^{-10(x^2+y^2+z^2)}, \tag{7.23}$$

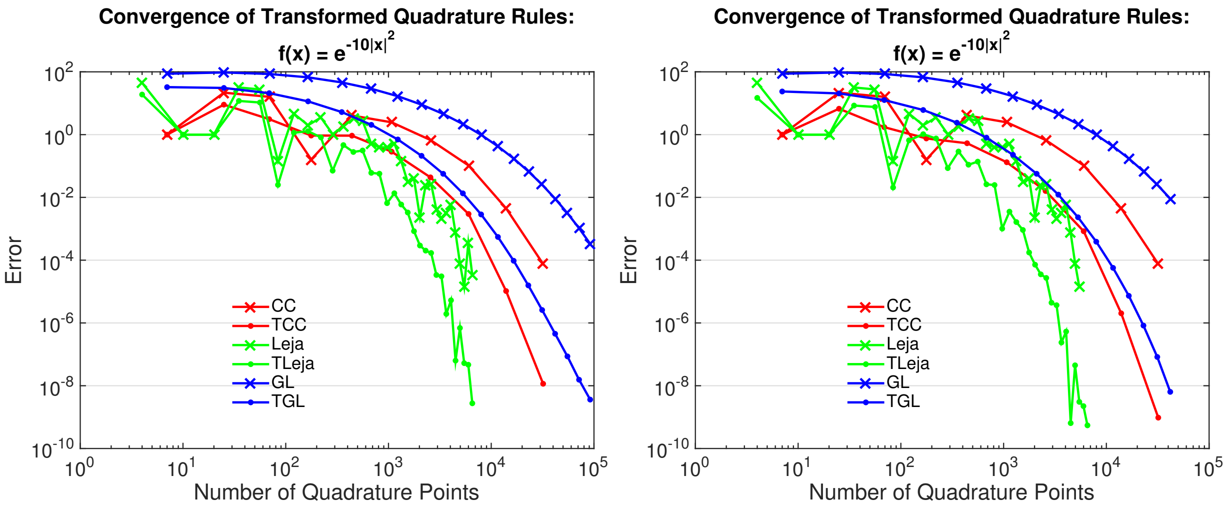

Figure 7.5: Comparison of sparse grid quadrature rules for computing the integral of (7.23) over the cube $[-1,1]^3$, using the conformal maps (7.6) (left), and (7.8) (right).

over the cube $[-1,1]^3$. This function is entire, but grows rapidly in the complex hyperplane away from $[-1,1]^3$. The left-hand plot shows the performance of the sparse grid transformed quadratures using the transformation (7.6), while the right-hand plot uses (7.8). In each case, the sparse quadrature approximations using mapped rules outperform traditional sparse grid quadrature, and there is only a slight difference in the performance of the transformed rules corresponding to the different mappings.

Finally, in Figure 7.6, we plot results for approximating the integral of the function

$$f(x,y,z) = \cos(1 + x^2 + y^2 + z^2), \tag{7.24}$$

over the cube, $[-1,1]^3$. This function is entire and does not grow too quickly away from the unit cube in the complex hyperplane $\mathbb{C}^3$. On the other hand, by fixing the parameters in the conformal mapping, we are restricting the analyticity of the composition $f \circ g$, and hence restricting the convergence rate of the transformed sparse grid rules. In other words, the conformal mapping technique cannot take full advantage of the analyticity of this function. Thus, we see that the rules based on holomorphic mappings are inferior for computing the integral of this function, though the transformed Leja rule using (7.6) is competitive, at least up to the computed level.

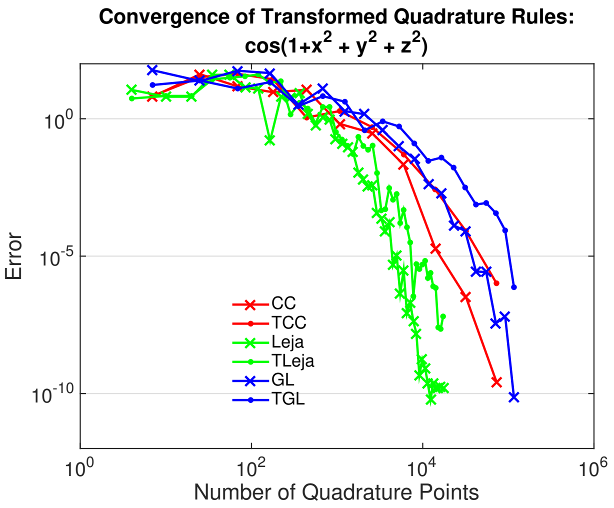

Figure 7.6: Comparison of sparse grid quadrature rules for computing the integral of (7.24) over the cube $[-1,1]^3$, using the conformal maps (7.6) (left), and (7.8) (right).

7.3.2 Effect of Dimension

Next we investigate the effect of increasing the dimension d of the integral problem, and see whether the holomorphic transformation idea indeed decreases the computational cost with growing dimension. The test integral for this experiment is

$$\int_{[-1,1]^d} \prod_{i=1}^{d} \left(\frac{1}{1+5x_i^2} \right) d\boldsymbol{x}. \tag{7.25}$$

In Table 7.1 we compare the number of points used to estimate the integral (7.25) in $d = 2, 4, 6$ dimensions, up to the given error tolerance. We use both the Clenshaw–Curtis and the Leja rules, with their corresponding transformed versions. Here we implement only the map (7.6) with $\rho = 1.7$, which maps the ellipse (3.7) to a strip of half-width $\frac{1}{\pi}(1.7-1) \approx 1/\sqrt{5}$. This integral has simple product structure, so we compare the computed sparse grid approximation to the "true" integral value computed to high precision. As expected, the sparse grid rules using transformed quadrature need far fewer points to compute the value of the integral up to a given tolerance, as compared with standard sparse grid rules. As the dimension increases, because of the doubling rule p from Table 3.1, the number of points grows rapidly from one level to the next. Thus, a certain grid may vastly undershoot or overshoot the optimal number of points needed to achieve a certain error. Furthermore, it

Table 7.1: Comparison of the number of points used by a given sparse grid quadrature rule to approximate the integral (7.25) to the given tolerance

Dimension	Tol	CC	TCC	Ratio	Leja	TLeja	Ratio
2	10^{-7}	1537	705	2.18	666	435	1.53
4	10^{-5}	1507329	271617	5.55	73815	20475	3.61
6	10^{-2}	6436865	127105	50.64	593775	12376	47.98

may be the case that the convergence has not yet reached the asymptotic regime for such a large tolerance 10^{-2}, and so we claim from Table 7.1 that the transformed sparse grid rules may work well even before the convergence is governed by the asymptotic theory.

7.4 Remarks

In this chapter, we have demonstrated the application of the transformed quadrature rules of [49] to isotropic sparse grid quadrature in high dimensions, and showed that in certain situations we are able to speed up convergence of a transformed sparse approximation by a factor approaching $(\pi/2)^{\xi(d)^{-1}}$, where $\xi(d)^{-1} \approx d \log d$. We applied the rules to several test integrals, and experimented with different conformal mappings g, and found that the sparse grid quadratures with conformally mapped rules outperformed the standard sparse grid rules based on one-dimensional interpolatory quadrature by a significant amount for several example integrands. The convergence is shown to be improved even when the mappings are not tuned to the specific integrand at hand, and even before convergence enters the asymptotic regime.

Bibliography

[1] Alpert, B. K. (1999). Hybrid Gauss-trapezoidal quadrature rules. *SIAM Journal on Scientific Computing*, 20(5):1551–1584. 11

[2] Babuška, I., Nobile, F., and Tempone, R. (2007). A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, 45(3):1005–1034. 3, 4, 16, 23, 28

[3] Bäck, J., Nobile, F., Tamellini, L., and Tempone, R. (2011). Stochastic spectral Galerkin and collocation methods for PDEs with random coefficients: A numerical comparison. In Hesthaven, J. S. and Rønquist, E. M., editors, *Spectral and High Order Methods for Partial Differential Equations*, volume 76 of *Lecture Notes in Computational Science and Engineering*. Springer Berlin Heidelberg, Berlin, Heidelberg. 19

[4] Bank, R. E. and Scott, L. R. (1989). On the conditioning of finite element equations with highly refined meshes. *SIAM Journal on Numerical Analysis*, 26(6):1383–1394. 77

[5] Barth, A., Schwab, C., and Zollinger, N. (2011). Multi-level Monte Carlo finite element method for elliptic PDEs with stochastic coefficients. *Numer. Math.*, 119(1):123–161. 5

[6] Beck, J., Nobile, F., Tamellini, L., and Tempone, R. (2014). Convergence of quasi-optimal stochastic Galerkin methods for a class of PDEs with random coefficients. *Computers and Mathematics with Applications*, 67(4):732–751. 3

[7] Beylkin, G. and Sandberg, K. (2005). Wave propagation using bases for bandlimited functions. *Wave Motion*, 41(3):263–291. 11

[8] Bieri, M. (2011). A sparse composite collocation finite element method for elliptic SPDEs. *SIAM Journal on Numerical Analysis*, 49(6):2277–2301. 5

[9] Boyd, J. P. (2004). Prolate spheroidal wavefunctions as an alternative to Chebyshev and Legendre polynomials for spectral element and pseudospectral algorithms. *Journal of Computational Physics*, 199(2):688–716. 11

[10] Brenner, S. and Scott, L. (2008). *The Mathematical Theory of Finite Element Methods*, volume 15 of *Texts in Applied Mathematics*. Springer, third edition. 29, 34, 42

[11] Burkardt, J., Gunzburger, M., Webster, C. G., and Zhang, G. (2014). A hyper-spherical sparse grid approach for high-dimensional discontinuity detection. *SIAM Journal on Numerical Analysis*. To appear. 7

[12] Chae, S. B. (1985). *Holomorphy and Calculus in Normed Spaces*, volume 92 of *Monographs and textbooks in pure and applied mathematics*. 43, 44

[13] Chan, T. F. and Ng, M. K. (1999). Galerkin projection methods for solving multiple linear systems. *SIAM Journal on Scientific Computing*, 21(3):836–850. 6

[14] Charrier, J. (2012). Strong and weak error estimates for elliptic partial differential equations with random coefficients. *SIAM Journal on Numerical Analysis*, 50(1):216–246. 14, 16, 46

[15] Charrier, J., Scheichl, R., and Teckentrup, A. (2013). Finite element error analysis of elliptic PDEs with random coefficients and its application to multilevel Monte Carlo methods. *SIAM J. Numer. Anal.*, 51(1):322–352. 5, 16, 28

[16] Chen, Q.-Y., Gottlieb, D., and Hesthaven, J. S. (2005). Spectral methods based on prolate spheroidal wave functions for hyperbolic PDEs. *SIAM Journal on Numerical Analysis*, 43(5):1912–1933. 11

[17] Chkifa, A., Cohen, A., and Schwab, C. (2014). High-dimensional adaptive sparse polynomial interpolation and applications to parametric PDEs. *Foundations of Computational Mathematics*, 14(4):601–633. 14

[18] Chkifa, A., Cohen, A., and Schwab, C. (2015). Breaking the curse of dimensionality in sparse polynomial approximation of parametric PDEs. *Journal de Mathématiques Pures et Appliquées*, 103(2):400–428. 3, 14

[19] Chkifa, A., Dexter, N., Tran, H., and Webster, C. G. (2016). Polynomial approximation via compressed sensing of high-dimensional functions on lower sets. *arXiv preprint arXiv:1602.05823*. 4

[20] Ciarlet, P. G. (1978). *The Finite Element Method for Elliptic Problems*. North–Holland. 29, 42

[21] Clenshaw, C. W. and Curtis, A. R. (1960). A method for numerical integration on an automatic computer. *Numerische Mathematik*, 2(1):197–205. 20

[22] Cliffe, K., Giles, M., Scheichl, R., and Teckentrup, A. (2011). Multilevel monte carlo methods and applications to elliptic PDEs with random coefficients. *Comput. Vis. Sci.*, 14(1):3–15. 24

[23] Cohen, A., DeVore, R., and Schwab, C. (2011). Analytic regularity and polynomial approximation of parametric and stochastic elliptic PDEs. *Analysis and Applications*, 9(01):11–47. 3, 14, 23

[24] Collier, N., Haji-Ali, A.-L., Nobile, F., von Schwerin, E., and Tempone, R. (2015). A continuation multilevel monte carlo algorithm. *BIT Numerical Mathematics*, 55(2):399–432. 53

[25] De Marchi, S. (2004). On Leja sequences: some results and applications. *Applied mathematics and computation*, 152(3):621–647. 21

[26] Dereich, S. and Heidenreich, F. (2011). A multilevel Monte Carlo algorithm for Lévy-driven stochastic differential equations. *Stoch. Proc. Appl.*, 121(7):1565–1587. 5

[27] Dexter, N. C., Webster, C. G., and Zhang, G. (2016). Explicit cost bounds of stochastic galerkin approximations for parameterized pdes with random coefficients. *Computers & Mathematics with Applications*, 71(11):2231–2256. 3

[28] Dzjadyk, V. K. and Ivanov, V. V. (1983). On asymptotics and estimates for the uniform norms of the Lagrange interpolation polynomials corresponding to the Chebyshev nodal points. *Analysis Mathematica*, 9(2):85–97. 68

[29] Eldred, M. and Burkardt, J. (2009). Comparison of non-intrusive polynomial chaos and stochastic collocation methods for uncertainty quantification. In *47th AIAA Aerospace*

Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, page 976. 3

[30] Ernst, O. G. and Ullmann, E. (2010). Stochastic Galerkin matrices. *SIAM Journal on Matrix Analysis and Applications*, 31(4):1848–1872. 6

[31] Favati, P., Lotti, G., and Romani, F. (1993). Bounds on the error of Fejér and Clenshaw–Curtis type quadrature for analytic functions. *Applied mathematics letters*, 6(6):3–8. 11

[32] Fettis, H. E. (1969). Note on the computation of Jacobi's nome and its inverse. *Computing*, 4(3):202–206. 117

[33] Fishman, G. (1996). *Monte Carlo: concepts, algorithms, and applications*. Springer. 2

[34] Foucart, S. and Rauhut, H. (2013). *A Mathematical Introduction to Compressive Sensing*. Applied and Numerical Harmonic Analysis. Birkhäuser. 4

[35] Frauenfelder, P., Schwab, C., and Todor, R. (2005). Finite elements for elliptic problems with stochastic coefficients. *Comput. Methods Appl. Mech. Engrg.*, 194(2-5):205–228. 14

[36] Galindo, D., Jantsch, P., Webster, C. G., and Zhang, G. (2016). Accelerating stochastic collocation methods for partial differential equations with random input data. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1):1111–1137. 58

[37] Garcıa, A. L. (2010). Greedy energy points with external fields. *Recent Trends in Orthogonal Polynomials and Approximation Theory, Contemporary Mathematics*, 507:189–207. 9, 98

[38] Gerstner, T. and Griebel, M. (1998). Numerical integration using sparse grids. *Numerical algorithms*, 18(3-4):209. 5, 120

[39] Gerstner, T. and Griebel, M. (2003). Dimension–adaptive tensor–product quadrature. *Computing*, 71(1):65–87. 14

[40] Ghanem, R. G. and Kruger, R. M. (1996). Numerical solution of spectral stochastic finite element systems. *Computer Methods in Applied Mechanics and Engineering*, 129(3):289–303. 6, 7

[41] Ghanem, R. G. and Spanos, P. D. (1991). *Stochastic finite elements: a spectral approach*. Springer, New York. 3, 14, 46

[42] Giles, M. (2008). Multilevel monte carlo path simulation. *Oper. Res.*, 56(3):607–617. 5, 51

[43] Giles, M. B. and Reisinger, C. (2012). Stochastic finite differences and multilevel Monte Carlo for a class of SPDEs in finance. *SIFIN*, 3(1):572–592. 5

[44] Gittelson, C., Könnö, J., Schwab, C., and Stenberg, R. (2013). The multilevel Monte Carlo finite element method for a stochastic Brinkman problem. *Numer. Math.*, 125:347–386. 5

[45] Gordon, A. D. and Powell, C. E. (2012). On solving stochastic collocation systems with algebraic multigrid. *IMA Journal of Numerical Analysis*, 32(3):1051–1070. 6, 7, 81

[46] Götz, M. (2001). Optimal quadrature for analytic functions. *Journal of computational and applied mathematics*, 137(1):123–133. 11

[47] Gunzburger, M. D., Webster, C. G., and Zhang, G. (2014a). An adaptive wavelet stochastic collocation method for irregular solutions of partial differential equations with random input data. In *Sparse Grids and Applications-Munich 2012*, pages 137–170. Springer. 3, 5, 7, 23

[48] Gunzburger, M. D., Webster, C. G., and Zhang, G. (2014b). Stochastic finite element methods for partial differential equations with random input data. *Acta Numerica*, 23:521–650. 2, 14, 16, 19

[49] Hale, N. and Trefethen, L. N. (2008). New quadrature formulas from conformal maps. *SIAM Journal on Numerical Analysis*, 46(2):930–948. 10, 11, 114, 115, 116, 122, 123, 125, 129

[50] Harbrecht, H., Peters, M., and Siebenmorgen, M. (2013). *On Multilevel Quadrature for Elliptic Stochastic Partial Differential Equations*, pages 161–179. Springer Berlin Heidelberg, Berlin, Heidelberg. 6, 29, 34

[51] Harbrecht, H., Peters, M., and Siebenmorgen, M. (2016). Multilevel Accelerated Quadrature for PDEs with Log-Normally Distributed Diffusion Coefficient. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1):520–551. 6

[52] Heinrich, S. (2001). Multilevel Monte Carlo methods. In *Large-scale scientific computing*, pages 58–67. Springer. 5

[53] Helton, J. and Davis, F. (2003). Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems. *Reliability Engineering and System Safety*, 81:23–69. 3

[54] Hoel, H., von Schwerin, E., Szepessy, A., and Tempone, R. (2012). Adaptive multilevel Monte Carlo simulation. In Engquist, B., Runborg, O., and Tsai, Y.-H. R., editors, *Numerical Analysis of Multiscale Computations*, volume 82 of *Lect. Notes Comp. Sci.*, pages 217–234. Springer. 5

[55] Jantsch, P., Webster, C. G., and Zhang, G. (2016). On the Lebesgue constant of weighted Leja points for Lagrange interpolation on unbounded domains. *arXiv preprint arXiv:1606.07093*. 91, 119

[56] Jin, C., Cai, X.-C., and Li, C. (2007). Parallel domain decomposition methods for stochastic elliptic equations. *SIAM Journal on Scientific Computing*, 29(5):2096–2114. 6

[57] Kapur, S. and Rokhlin, V. (1997). High-order corrected trapezoidal quadrature rules for singular functions. *SIAM Journal on Numerical Analysis*, 34(4):1331–1356. 11

[58] Kosloff, D. and Tal-Ezer, H. (1989). Modified Chebyshev pseudospectral method with $O(N^{-1})$ time step restriction. 11, 114

[59] Kowalski, M., Werschulz, A. G., and Woźniakowski, H. (1985). Is Gauss quadrature optimal for analytic functions? *Numerische Mathematik*, 47(1):89–98. 11

[60] Krylov, V. I. (2006). *Approximate calculation of integrals*. Dover. 10

[61] Levin, A. L. and Lubinsky, D. S. (2001). *Orthogonal polynomials for exponential weights*. CMS books in mathematics. Springer Science & Business Media. 91, 94, 95, 105, 106

[62] Loève, M. (1977). *Probability theory. I*, volume 45 of *Graduate Texts in Mathematics*. Springer-Verlag. 14

[63] Ma, J., Rokhlin, V., and Wandzura, S. (1996). Generalized Gaussian quadrature rules for systems of arbitrary functions. *SIAM Journal on Numerical Analysis*, 33(3):971–996. 11

[64] Ma, X. and Zabaras, N. (2009). An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations. *J. Comput. Phys.*, 228(8):3084–3113. 3, 23

[65] Migliorati, G., Nobile, F., von Schwerin, E., and Tempone, R. (2013). Approximation of quantities of interest in stochastic PDEs by the random discrete l^2 projection on polynomial spaces. *SIAM Journal on Scientific Computing*, 35(3):A1440–A1460. 4

[66] Migliorati, G., Nobile, F., von Schwerin, E., and Tempone, R. (2014). Analysis of discrete l^2 projection on polynomial spaces with random evaluations. *Foundations of Computational Mathematics*, 14(3):419–456. 4

[67] Mishra, S. and Schwab, C. (2012). Sparse tensor multi-level Monte Carlo finite volume methods for hyperbolic conservation laws with random initial data. *Mathematics of Computation*, 81(280):1979–2018. 5

[68] Narayan, A. and Jakeman, J. D. (2014). Adaptive Leja sparse grid constructions for stochastic collocation and high-dimensional approximation. *SIAM Journal on Scientific Computing*, 36(6):A2952–A2983. 8, 9, 91, 98, 119

[69] Niederreiter, H. (1992). *Random number generation and quasi-Monte Carlo methods*, volume 63 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA. 3

[70] Nobile, F., Tempone, R., and Webster, C. G. (2008a). An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, 46(5):2411–2442. 3, 4, 7, 20, 23, 24, 38, 83, 121

[71] Nobile, F., Tempone, R., and Webster, C. G. (2008b). A sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, 46(5):2309–2345. 3, 4, 7, 14, 16, 21, 23, 24, 38, 66, 74, 80, 120, 124

[72] Novak, E. and Ritter, K. (1996). High dimensional integration of smooth functions over cubes. *Numerische Mathematik*, 75(1):79–97. 120

[73] Øksendal, B. (2003). *Stochastic differential equations*. Springer. 15

[74] Parks, M. L., De Sturler, E., Mackey, G., Johnson, D. D., and Maiti, S. (2006). Recycling Krylov subspaces for sequences of linear systems. *SIAM Journal on Scientific Computing*, 28(5):1651–1674. 6

[75] Pellissetti, M. F. and Ghanem, R. G. (2000). Iterative solution of systems of linear equations arising in the context of stochastic finite elements. *Advances in Engineering Software*, 31(8):607–616. 7

[76] Petras, K. (1998). Gaussian versus optimal integration of analytic functions. *Constructive approximation*, 14(2):231–245. 11

[77] Powell, C. E. and Elman, H. C. (2009). Block-diagonal preconditioning for spectral stochastic finite-element systems. *IMA Journal of Numerical Analysis*, 29(2):350–375. 6

[78] Quarteroni, A., Sacco, R., and Saleri, F. (2000). *Numerical Mathematics*. Texts in applied mathematics. Springer. 6

[79] Rauhut, H. and Schwab, C. (2017). Compressive sensing Petrov-Galerkin approximation of high-dimensional parametric operator equations. *Mathematics of Computation*, 86(304):661–700. 4

[80] Rauhut, H. and Ward, R. (2016). Interpolation via weighted ℓ_1-minimization. *Applied and Computational Harmonic Analysis*, 40(2):321–351. 4

[81] Saad, Y. (2003). *Iterative methods for sparse linear systems*. SIAM. 6, 63

[82] Saff, E. B. and Totik, V. (1997). *Logarithmic Potentials with External Fields*, volume 316. Springer Science & Business Media. 91, 94, 95, 96, 97, 100, 108

[83] Simoncini, V. and Szyld, D. B. (2007). Recent computational developments in Krylov subspace methods for linear systems. *Numerical Linear Algebra with Applications*, 14(1):1–59. 6

[84] Slepian, D. and Pollak, H. O. (1961). Prolate spheroidal wave functions, Fourier analysis and uncertainty–I. *Bell Labs Technical Journal*, 40(1):43–63. 11

[85] Slevinsky, R. M. and Olver, S. (2015). On the use of conformal maps for the acceleration of convergence of the trapezoidal rule and sinc numerical methods. *SIAM Journal on Scientific Computing*, 37(2):A676–A700. 11

[86] Smolyak, S. (1963). Quadrature and interpolation formulas for tensor products of certain classes of functions. *Dokl. Akad. Nauk SSSR*, 4:240–243. 21

[87] Stoyanov, M. (2017). User manual: Tasmanian sparse grids v4.0. Technical report. 125

[88] Stoyanov, M. and Webster, C. G. (2014). A gradient-based sampling approach for stochastic dimension reduction for partial differential equations with random input data. *International Journal for Uncertainty Quantification*. To appear. 3

[89] Stoyanov, M. K. and Webster, C. G. (2016). A dynamically adaptive sparse grids method for quasi-optimal interpolation of multidimensional functions. *Computers & Mathematics with Applications*, 71(11):2449–2465. 125

[90] Taylor, R. (2008). *Lagrange interpolation on Leja points*. PhD thesis. 94, 104, 107, 112

[91] Taylor, R. and Totik, V. (2010). Lebesgue constants for Leja points. *IMA Journal of Numerical Analysis*, 30(2):462–486. 94, 107, 119

[92] Teckentrup, A. L., Jantsch, P., Webster, C. G., and Gunzburger, M. (2015). A multilevel stochastic collocation method for partial differential equations with random input data. *SIAM/ASA Journal on Uncertainty Quantification*, 3(1):1046–1074. 26, 88

[93] Teckentrup, A. L., Scheichl, R., Giles, M. B., and Ullmann, E. (2013). Further analysis of multilevel Monte Carlo methods for elliptic PDEs with random coefficients. *Numer. Math.*, 3(125):569–600. 16, 28, 30

[94] Tran, H., Webster, C. G., and Zhang, G. (2014). Analysis of quasi-optimal polynomial approximations for parameterized PDEs with deterministic and stochastic coefficients. Technical Report ORNL/TM-2014/468, Oak Ridge National Laboratory. Submitted to Numerishe Mathematik. 3, 15, 23

[95] Trefethen, L. N. (2008). Is Gauss quadrature better than Clenshaw-Curtis? *SIAM review*, 50(1):67–87. 21

[96] Trefethen, L. N. (2013). *Approximation theory and approximation practice*. SIAM. 10, 118

[97] Whittlesey, E. (1965). Analytic functions in Banach spaces. *Proc. Amer. Math. Soc*, 16(5):1077–1083. 43

[98] Xiao, H., Rokhlin, V., and Yarvin, N. (2001). Prolate spheroidal wavefunctions, quadrature and interpolation. *Inverse problems*, 17(4):805. 11

[99] Xiu, D. and Karniadakis, G. E. (2002). The Wiener–Askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing*, 24(2):619–644. 3, 7

Vita

Peter Jantsch, son of Steven and Darla Jantsch, is a native of Beaver Falls, Pennsylvania. He graduated from Beaver County Christian School in 2007, and from there earned a Bachelor of Science in Mathematics from Grove City College, graduating *summa cum laude* in 2011. After college, Peter spent the next six years at the University of Tennessee, Knoxville. There he earned a Master of Science in Mathematics in 2013, after which he achieved a Doctor of Philosophy degree in Mathematics in the summer of 2017. He thesis work was advised by Prof. Clayton G. Webster in the Department of Mathematics at UTK. In 2017, Peter was awarded the National Science Foundation Mathematical Sciences Postdoctoral Research Fellowship. He will use this fellowship to begin three years of postdoctoral work in the Department of Mathematics at Texas A&M University, where he will be advised by Prof. Ronald Devore.