ALGORITHMS FOR NUMERICAL INTEGRATION IN HIGH AND INFINITE DIMENSIONS: ANALYSIS, APPLICATIONS AND IMPLEMENTATION

Alexander D. Gilbert

Supervisors: A/Prof. Frances Y. Kuo and Prof. Ian H. Sloan

School of Mathematics and Statistics
UNSW Sydney

June 2018

Submitted in fulfillment of the requirements of the degree of
Doctor of Philosophy

PLEASE TYPE

THE UNIVERSITY OF NEW SOUTH WALES
Thesis/Dissertation Sheet

Surname or Family name: Gilbert

First name: Alexander Other name/s: Dominik

Abbreviation for degree as given in the University calendar: PhD

School: Mathematics and Statistics Faculty: Science

Title: Algorithms for Numerical Integration in High and Infinite Dimensions: Analysis, Applications and Implementation

Abstract 350 words maximum: (PLEASE TYPE)

Approximating high and infinite dimensional integrals numerically is in general a very difficult problem. However, it is also one that arises in several applications from statistics, finance and uncertainty quantification, thus motivating a real need for the development and analysis of efficient algorithms. The difficulty lies in the fact that in general high-dimensional problems suffer from the curse of dimensionality where the cost of an approximation rises exponentially with dimension. However, knowing certain properties of the integrands allows one to identify problems that do not suffer from the curse and for which efficient algorithms can be developed.

In this thesis we study numerical integration algorithms, specifically Quasi-Monte Carlo (QMC) quadrature rules and the Multivariate Decomposition Method (MDM), when bounds on the first mixed derivatives are known. The focus in this thesis is on analysis and development of algorithms, a new application for QMC methods from the field of uncertainty quantification and efficient strategies for implementing numerical integration algorithms.

The main results of this thesis are as follows. First, we present a full error analysis for the application of QMC methods to approximate the expectation of the smallest eigenvalue of an elliptic differential operator with coefficients that are parametrised by infinitely-many stochastic variables. Eigenvalue problems are used to model many physical situations in engineering and the natural sciences, and this problem is motivated by uncertainty quantification of such problems. It also represents a new application for QMC methods. Second, we provide explicit details and numerical results on how to efficiently implement the Multivariate Decomposition Method (MDM) for approximating infinite-dimensional integrals. The third contribution of this thesis is a new method of constructing optimal active sets for use in the MDM. Finally, we present two user-friendly Component-by-Component algorithms for constructing QMC lattice rules, which automatically choose good function space weight parameters. In all cases we present numerical results that display the advantages of the algorithms, and where appropriate substantiate our corresponding theoretical results.

Declaration relating to disposition of project thesis/dissertation

I hereby grant to the University of New South Wales or its agents the right to archive and to make available my thesis or dissertation in whole or in part in the University libraries in all forms of media, now or here after known, subject to the provisions of the Copyright Act 1968. I retain all property rights, such as patent rights. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

I also authorise University Microfilms to use the 350 word abstract of my thesis in Dissertation Abstracts International (this is applicable to doctoral theses only).

Witness Signature Date

The University recognises that there may be exceptional circumstances requiring restrictions on copying or conditions on use. Requests for restriction for a period of up to 2 years must be made in writing. Requests for a longer period of restriction may be considered in exceptional circumstances and require the approval of the Dean of Graduate Research.

FOR OFFICE USE ONLY Date of completion of requirements for Award:

ORIGINALITY STATEMENT

'I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.'

Signed ……..

Date ……..

Acknowledgements

The first, and most important, thank you goes to my supervisors Frances and Ian. Thank you both for your time, support and knowledge. I have learnt a great amount from both of you, not just about mathematics. You have given me many great opportunities, including to present my work all over the world and meet great new people, always going out of your way to introduce me. I did not expect any of this when I began my PhD three and a half odd years years ago. But mostly, I did not expect to become friends with my bosses. So thank you.

Secondly, I would like to thank my collaborators Ivan Graham, Rob Scheichl, Dirk Nuyens, and Greg Wasilkowski. It has truly been a pleasure working with you on our different projects, and I have learnt a tremendous amount from all of you.

Next are all of the friends who have hosted me for research visits over the years: Dirk in Leuven; Rob in Heidelberg; Rob and Ivan in Bath; Christian and my Austrian colleagues at JKU and RICAM in Linz; and Weichung at NTU in Taipei. Thank you all for supporting me, but also for making me feel welcome in your home institutions and cities.

The starting point for my journey into mathematics was a dull brick classroom at the end of a dull concrete hall, commanded by the fearsome Mrs. Blanton. Without whose "stern" teaching I would never have stuck with mathematics over all these years.

Finally, I would like to thank all of the friends I have made along the way. From my friends here at UNSW (Anna, Barton, Jeremy, Michael, Tim, Yoshihito... and someone else that I once knew) to all of the friends I have met all over the world at various conferences, you have all made my PhD time more enjoyable.

Cheers,

Alec.

June 6, 2018.

Contents

Chapter 1

Preliminary remarks

1.1 The story of this thesis

The unifying theme of this thesis is the following numerical integration problem: approximate an integral of the form

$$\int_{[0,1]^s} f(y_1, y_2, \ldots, y_s)\, \mathrm{d}y_1 \mathrm{d}y_2 \ldots \mathrm{d}y_s\,, \tag{1.1}$$

where the dimension s is very large—in the hundreds, thousands or often even infinite—and bounds on the first mixed derivatives of the integrand are known. We will look at different aspects of numerical integration, from a new application in Uncertainty Quantification where a problem of this form arises, to abstract implementation and algorithmic aspects relating to, in particular, Quasi-Monte Carlo quadrature rules and Multivariate Decomposition Methods.

Numerical approximation of integrals with high dimension is a difficult task that arises in numerous practical applications, from statistics [56, 78] to the pricing of financial derivatives [11, 34, 72] to quantifying the uncertainty in models from engineering and the natural sciences [38, 94]. The difficulty comes from the fact that for most traditional methods, such as the product of one-dimensional quadrature rules, the cost of an approximation increases exponentially with dimension, which means that very quickly computing even the simplest approximation is completely untenable. As an example, consider the product of one-dimensional trapezoidal rules. The simplest non-trivial approximation requires two points in each dimension, hence, in ten dimensions this product trapezoidal rule will evaluate the integrand at 2^{10} points. By the time we reach 100 dimensions such a rule requires 2^{100} function evaluations—which it may interest the reader to know is more than the current estimate of stars in the universe!

Having said all this, classical Monte Carlo methods, see e.g., [41], deserve an honourable mention. A Monte Carlo approximation of an integral is performed by taking the average of evaluations of the integrand at a number n, say, of randomly

distributed points. They are extremely robust, in that they work for any function that is square-integrable and the error does not explicitly depend on the dimension. The catch of this robustness is that the error decays very slowly, at a rate of only $1/\sqrt{n}$.

Nowadays, one of the state-of-the-art techniques for high-dimensional integration are *Quasi-Monte Carlo*, or QMC, methods, which for certain classes of integrands are able to achieve errors that converge significantly faster than Monte Carlo methods *and* that are independent of dimension, see, e.g., [18, 20]. Similarly to Monte Carlo, QMC methods approximate the integral by the average of n of function values, but now the points where the integrand is evaluated are chosen deterministically, with the explicit purpose of being well-distributed and exhibiting desirable approximation properties.

The classical study of QMC methods has roots in number theory from the mid-twentieth century, such as the works [40, 47, 51, 84], but during this period the interest was focussed on using number theoretic tools to construct well-distributed point sets, with little attention paid to the integrand or to the dimension of the problem. And because of the occurrence of pesky $(\log n)^s$ factors in the classical error bounds, the efficacy of classical QMC techniques for the approximation and error analysis of truly high-dimensional integrals was limited. The modern theory of QMC methods, however, takes a more rounded view, focussing not only on the point sets but also on identifying classes of functions for which QMC methods can achieve fast convergence and for which it is possible to alleviate the dependence of the error on the dimension.

A resurgence of interest in QMC rules occurred in the 1990's when they were used to efficiently approximate a 360-dimensional integral coming from a model for the value of a financial derivative, see [72]. Following on from this success, there was an interest in attempting to explain why the QMC rules had performed so well. One explanation was that although formally the integral depended on 360 dimensions, perhaps it was actually a low-dimensional problem masquerading as a high-dimensional problem, see, e.g., [11]. Shortly thereafter, a similar concept was formalised by Sloan & Woźniakowski [82], whose general idea was that to successfully tackle high-dimensional integration with QMC rules, not all of the variables can contribute equally to the integral. This concept was captured theoretically by introducing a function space that depends on a sequence of weight parameters, one for each variable, that model the relative importance of different variables. Most importantly, they identified conditions on the decay of the weights that allowed them to show that for integrands belonging to such a weighted function space it is

possible to construct QMC methods that perform well with no dependence of the error on dimension. Since then, there have been many advances in QMC methods based around analysis of the error in the setting of weighted function spaces. Major contributions include proving that error bounds for a class of specific QMC rules, called *lattice rules*, converge arbitrarily close to $1/n$ independently of dimension [55], and also the development of fast algorithms for the construction of such rules [69, 70]. Along with, more recently, the invention of *higher-order* QMC rules that, under additional assumptions on the higher-order derivatives of the integrand, can achieve convergence rates of $1/n^{\alpha}$ for $\alpha > 1$, also independent of dimension, see e.g., [14, 15, 19].

At the extreme of high-dimensionality, there has recently been significant interest in studying integration of functions that depend on infinitely-many variables, both from an approximation perspective but also concerning the theory of infinite-dimensional integration in its own right. For the successful numerical integration of infinite-variate functions, one must first somehow truncate the problem to finitely-many dimensions, because only then can quadrature rules be used. One method of truncation is to choose a large, but finite, number s and discard all of the variables beyond y_s. Another more sophisticated truncation algorithm is the Multivariate Decomposition Method, which approximates the infinite-dimensional integral by the sum of many integrals with small or moderate dimension, see, e.g., [31, 58, 63, 75, 76, 89, 90]. Both of these truncation methods will be discussed throughout this thesis.

Although the study of infinite-variate integration is of a deep theoretical interest, there are several applications where such an integral must be computed, which provides the practical motivation for studying the approximation of infinite-variate integrals. Notable examples come from the field of Uncertainty Quantification, where one is interested in modelling how a physical model behaves when the input parameters are uncertain. The prototypical example of such problems is the flow of fluid through a porous medium such as rock or sand, see e.g., [38]. In this problem, the quantities we are interested in (such as pressure at a point) are obtained from the output of a physical model for the porous flow (a diffusion equation) that takes as input the permeability at each point in the domain (as a coefficient in the diffusion equation). Motivated by the fact that it is not possible to know the permeability everywhere in the domain exactly, the permeability is modelled as a random field, which is assumed to be parametrised by an infinite number of stochastic parameters (e.g., by a Karhunen-Loève expansion). This stochastic dependence carries through the model to our quantities of interest, and so the goal now is to compute their

expected value with respect to the infinitely-many parameters, which is an infinite-dimensional integral. In Chapter 3 we will study a new application that fits this mould, and analyse the application of QMC methods to it.

Clearly, the study of high and infinite-dimensional numerical integration is of significant interest, both practically and theoretically. In terms of practical applications, key topics are the development, implementation and analysis of efficient algorithms. And the aim of this thesis is to tackle different aspects of these three topics.

1.2 Structure of this thesis

The structure of the remainder of this thesis is as follows. The next chapter introduces the relevant mathematical background on numerical integration, in a more technical manner than here. The topics discussed will be Quasi-Monte Carlo methods, Smolyak/sparse grid quadrature rules and the Multivariate Decomposition Method.

The first original contribution of this thesis, in Chapter 3, is the application of Quasi-Monte Carlo quadrature to an elliptic eigenvalue problem with coefficients that depend on countably-many stochastic parameters. The goal here is to approximate the expectation, with respect to the stochastic parameters, of the smallest eigenvalue. The physical motivation is the criticality problem for a nuclear reactor: in steady state the fission reaction can be modelled by an elliptic eigenvalue problem, and the smallest eigenvalue provides a measure of how close the reaction is to equilibrium—in terms of production/absorption of neutrons. The rates of absorption and production of neutrons are inputted into the model through the coefficients, which are parametrised by countably-many stochastic variables so as to model the uncertainty of the composition of materials inside the reactor, e.g., the control rods, reactor structure, fuel rods etc. The approximation is performed by truncating the stochastic dimension, discretising the spatial domain using Finite Element methods and approximating the expected value using QMC methods. A large portion of the work is devoted to proving bounds on the mixed derivatives in order to rigorously prove bounds on the approximation error.

Chapters 4 and 5 deal with computational and algorithmic aspects of the Multivariate Decomposition Method (MDM). Chapter 4 develops strategies for the efficient implementation of the MDM, and numerical results for the performance of our implementations are also presented. Then Chapter 5 introduces a new method for performing the truncation component of the MDM (known as constructing active sets) for a specific case of input parameters.

Finally, in Chapter 6 two user-friendly Component-by-Component algorithms for constructing QMC lattice rules are introduced. The purpose of both algorithms is to construct good lattice rules without the need for the user to specify the function space weights as inputs. Instead, the user supplies bounds on the mixed first derivatives and good choices of weights are calculated automatically inside the algorithm.

1.3 Original research contributions

The content of each chapter corresponds to one of the research projects that I have worked on throughout my PhD. All four projects have culminated in an original research article, two of which have been published already, one has been submitted and the other manuscript is in the very final stages of preparation (at the time of writing, it will be submitted within a month). Here I briefly outline the original contributions of this thesis. At the end of each chapter, in the Conclusion section, I will give further details on the context of the work from that chapter, explicitly stating what is original and what was my contribution. The original contributions are:

1. A full error analysis of the application of Quasi-Monte Carlo methods to a class of elliptic eigenvalue problems with stochastic coefficients. Including the derivation of explicit upper bounds on the derivatives of the smallest eigenvalue (and the corresponding eigenfunction) with respect to the stochastic parameters.
2. Details and strategies to efficiently implement the Multivariate Decomposition Method using Smolyak and Quasi-Monte Carlo quadrature rules. Including explicit pseudocodes and numerical results.
3. A new method of constructing optimal active sets for use in the truncation step of the Multivariate Decomposition Method.
4. Two new Component-by-Component constructions of lattices rules that automatically choose good function space weights.

1.4 List of original research articles

1. A. D. Gilbert, I. G. Graham, F. Y. Kuo, R. Scheichl, and I. H. Sloan. Analysis of Quasi-Monte Carlo methods for elliptic eigenvalue problems with stochastic coefficients. *In progress*, 2018.
2. A. D. Gilbert, F. Y. Kuo, D. Nuyens, and G. W. Wasilkowski. Efficient implementations of the Multivariate Decomposition Method for approximating infinite-variate integrals. *Submitted to SIAM J. Sci. Comp., arXiv:* `https://arxiv.org/pdf/1712.06782.pdf`, version December, 2017.

3. A. D. Gilbert and G. W. Wasilkowski. Small superposition dimension and active set construction for multivariate integration under modest error demand. *J. Complexity*, 42:94–109, 2017.
4. A. D. Gilbert, F. Y. Kuo, and I. H. Sloan. Hiding the weights — CBC black box algorithms with a guaranteed error bound. *Math. Comp. Simul.*, 143:202–214, 2018.

Chapter 2

Mathematical introduction

In this chapter we summarise the specific aspects of numerical integration that will form the foundation for the work presented throughout this thesis, including Quasi-Monte Carlo methods, Smolyak/sparse grid quadrature rules and the Multivariate Decomposition Method. We also introduce the notation that will be used throughout this thesis. To begin with we formalise the integration problem introduced in the previous chapter.

2.1 The integration problem

Previously, we discussed an important, but specific, high-dimensional integration problem. However, we are also interested in integrals over general product domains, with respect to different probability measures, or of infinite-variate functions, and in this section we formalise such concepts. To this end, consider the product domain $\mathcal{Y}^s$ where $\mathcal{Y} \subseteq \mathbb{R}$, and let $\boldsymbol{y} \coloneqq (y_1, y_2, \cdots, y_s) \in \mathcal{Y}^s$ denote the variables with respect to which integration will be taken. Also, let $\rho : \mathcal{Y} \to [0, \infty)$ be a probability density function and denote the s-dimensional product density by

$$\boldsymbol{\rho}_s(\boldsymbol{y}) \coloneqq \prod_{j=1}^{s} \rho(y_j)\,.$$

An s-dimensional integral with respect to this product density will be written as

$$\mathcal{I}_s(f) \coloneqq \int_{\mathcal{Y}^s} f(\boldsymbol{y})\boldsymbol{\rho}_s(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y}\,. \tag{2.1}$$

As the title suggests, for much of the work in this thesis we are interested in integration problems as the dimension tends to ∞. Consider now an infinite-variate function $f : \mathcal{Y}^{\mathbb{N}} \to \mathbb{R}$, where $\mathbb{N} \coloneqq \{1, 2, 3 \ldots\}$ denotes the natural numbers and the domain is a countable product of $\mathcal{Y}$ as defined by

$$\mathcal{Y}^{\mathbb{N}} \coloneqq \{\boldsymbol{y} = (y_1, y_2, \ldots) : y_j \in \mathcal{Y}, j \in \mathbb{N}\}\,.$$

Then the infinite-dimensional integral is defined to be

$$\mathcal{I}(f) \coloneqq \lim_{s\to\infty} \int_{\mathcal{Y}^s} f(y_1, y_2, \ldots, y_s, 0, 0, \ldots) \prod_{j=1}^{s} \rho(y_j)\, \mathrm{d}y_j \,, \tag{2.2}$$

provided of course, that this limit exists and $0 \in \mathcal{Y}$.

The theoretical study of infinite-dimensional integration is a deep and active topic of research, which, because our focus is on the practical aspects of approximation, is beyond the scope of this thesis. For further details on infinite-dimensional integration and tractability studies of such problems the reader should see, e.g., [35, 36, 46, 58, 63, 75].

A common thread of the problems dealt with in this thesis is not only numerical integration, but also the interaction between the regularity of the integrand and the integration problem (2.1) or (2.2). At different points we will either assume or prove that the mixed first derivatives of the integrand satisfy collections of bounds of the following form:

$$\left| \frac{\partial^{|\mathfrak{u}|}}{\partial \boldsymbol{y}_{\mathfrak{u}}} f(\boldsymbol{y}) \right| \leq B_{\mathfrak{u}} \,,$$

where, as appropriate, either $\mathfrak{u} \subseteq \{1, 2, \ldots, s\}$ or $\mathfrak{u} \subset \mathbb{N}$ with finite cardinality and each $0 < B_{\mathfrak{u}} < \infty$ will be known or computable. In the bounds above $\boldsymbol{y}_{\mathfrak{u}} = (y_j)_{j\in\mathfrak{u}}$ are the *active variables* and

$$\frac{\partial^{|\mathfrak{u}|}}{\partial \boldsymbol{y}_{\mathfrak{u}}} \coloneqq \prod_{j\in\mathfrak{u}} \frac{\partial}{\partial y_j}$$

is the first-order, mixed partial derivative with respect to $\boldsymbol{y}_{\mathfrak{u}}$.

2.2 An integration problem from Uncertainty Quantification

A common problem in Uncertainty Quantification (UQ) is to compute the expected value of a quantity of interest—which may represent a physical quantity such as pressure at a point or average displacement—with respect to countably-many stochastic parameters that model the uncertainty of a physical model. A prominent example is the model for the flow of fluid through a porous medium discussed in the previous chapter.

Formally, let the quantity of interest $f : \mathcal{Y}^{\mathbb{N}} \to \mathbb{R}$ depend on the stochastic parameters $\boldsymbol{y} = (y_j)_{j=1}^{\infty}$, where each $y_j \in \mathcal{Y}$ is distributed according to the probability density function $\rho : \mathcal{Y} \to [0, \infty)$. Then the expected value is an infinite-dimensional integral as defined as above in (2.2):

$$\mathbb{E}_{\boldsymbol{y}}[f] \coloneqq \int_{\mathcal{Y}^{\mathbb{N}}} f(\boldsymbol{y})\, \mathrm{d}\boldsymbol{y} \coloneqq \lim_{s\to\infty} \int_{\mathcal{Y}^s} f(y_1, y_2, \ldots, y_s, 0, 0, \ldots) \prod_{j=1}^{s} \left(\rho(y_j)\, \mathrm{d}y_j\right) . \tag{2.3}$$

In Chapter 3 we will study a new application from nuclear physics, which yields problems of the form (2.3).

2.3 Quasi-Monte Carlo methods

A Quasi-Monte Carlo (QMC) approximation of the integral (2.1)—when the domain is the unit cube, $\mathcal{Y}^s = [0,1]^s$, and the density is uniform, $\rho \equiv 1$—is an equal-weight quadrature rule

$$Q_{n,s}(\mathcal{P}_n)f \coloneqq \frac{1}{n}\sum_{k=0}^{n-1} f(\boldsymbol{t}^{(k)})\,, \tag{2.4}$$

where the quadrature points, $\boldsymbol{t}^{(k)} = (t_1^{(k)}, t_2^{(k)}, \ldots, t_s^{(k)})$, are chosen deterministically from $[0,1]^s$. The whole point set will be denoted by $\mathcal{P}_n \coloneqq \{\boldsymbol{t}^{(k)}\}_{k=0}^{n-1}$.

In this section we fix notation and briefly introduce aspects of QMC theory that are relevant to this thesis. The topics include randomly shifted lattice rules, weighted function spaces for error analysis, and the Component-by-Component construction. For a more comprehensive overview the reader is referred to the review papers [18, 61] or the book [20].

2.3.1 Lattice rules

A *rank-1 lattice rule* [67, 79] is a QMC rule for which the quadrature points are generated by scaled multiples of a single integer vector $\boldsymbol{z}$ called the *generating vector*:

$$\boldsymbol{t}^{(k)} = \left\{\frac{k\boldsymbol{z}}{n}\right\} \quad \text{for } k = 0, 1, \cdots, n-1\,,$$

where the braces denote that we take the fractional part of each component to ensure that each point belongs to $[0,1]^s$. Because the operation of taking the fractional component is periodic, $\{x\} = \{x+1\}$, each component z_i is restricted to the multiplicative group of integers modulo n, denoted by

$$\mathbb{U}_n \coloneqq \{z \in \mathbb{N} : z < n,\, \gcd(z,n) = 1\}\,,$$

and $\boldsymbol{z} \in \mathbb{U}_n^s = \mathbb{U}_n \times \cdots \times \mathbb{U}_n$. Restricting each z_i to be co-prime with n also ensures that every one-dimensional projection of the quadrature points defines a set of n distinct points in $[0,1]$. The number of positive integers less than and co-prime to n is given by the Euler totient function $\varphi(n) = |\mathbb{U}_n|$. So, for an n-point lattice rule in s dimensions there are $(\varphi(n))^s$ possible generating vectors, which suggests that when looking for good generating vectors a brute force search is infeasible even for modest n and s. Details on how to construct good generating vectors will be given

in Section 2.3.3. With a slight abuse of notation, when a QMC point set $\mathcal{P}_n$ is a lattice rule generated by $\boldsymbol{z}$ we will replace dependence on $\mathcal{P}_n$ by dependence on $\boldsymbol{z}$, instead of the cumbersome $\mathcal{P}_n(\boldsymbol{z})$. For example, a lattice rule QMC approximation is written $Q_{n,s}(\boldsymbol{z})f$.

It is also practically beneficial to incorporate randomness into the QMC approximation, and for lattice rules this is easily done by randomly shifting the points. Given some random shift $\boldsymbol{\Delta} \in [0,1)^s$, where each component Δ_i is independent and uniformly distributed on $[0,1)$, the $\boldsymbol{\Delta}$-shifted rank-1 lattice rule corresponding to a generating vector $\boldsymbol{z}$ has points

$$\boldsymbol{t}^{(k)} = \left\{ \frac{k\boldsymbol{z}}{n} + \boldsymbol{\Delta} \right\} \quad \text{for } k = 0, 1, \ldots, n-1\,.$$

The practical benefits of random shifting are that the final estimate of the integral is unbiased, performing several random shifts provides a practical estimate of the error, and also that the construction of good generating vectors is simplified in the case of randomly shifted lattice rules. The latter two points will be discussed further in the following two sections.

To illustrate the structure of lattice rules, the points for a two-dimensional lattice rule with 55 points generated by $\boldsymbol{z} = (1, 34)$ are plotted in Figure 2.1. The left axes plot an ordinary rank-1 lattice while the right axes plot the randomly shifted rule for the same generating vector.

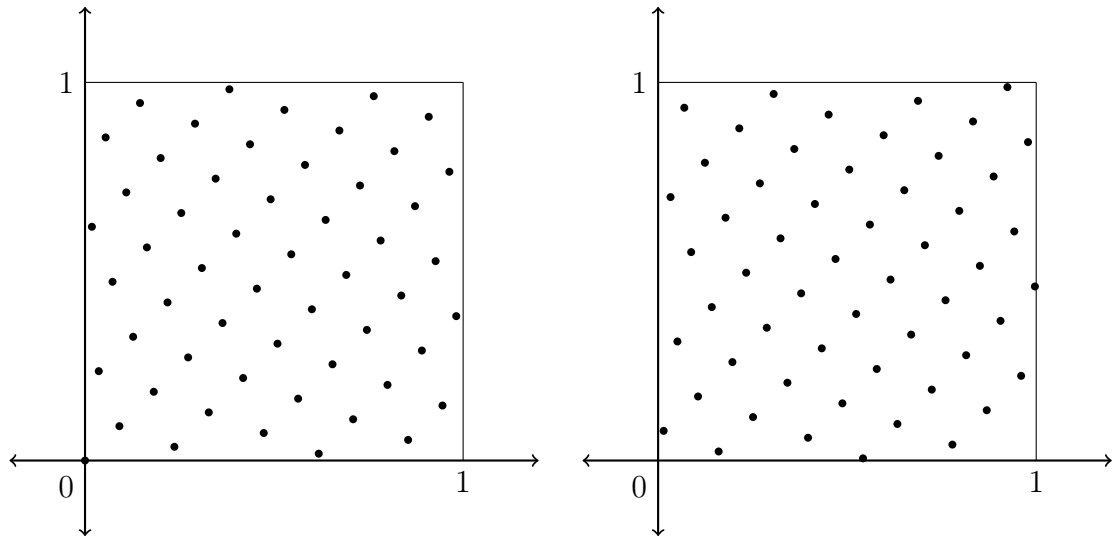


Figure 2.1: Quadrature points for a 2D lattice rule with $N = 55$, $\boldsymbol{z} = (1, 34)$: unshifted rule (left) and randomly shifted rule (right).

A key attribute of lattice rules is that given $\boldsymbol{z}$ they are incredibly simple to implement, but how does one choose a good generating vector? To answer this we

first require a measure of the quality of a QMC rule, which leads us to introduce some standard concepts of QMC error analysis.

2.3.2 Error analysis in weighted Sobolev spaces

The modern setting for the error analysis of a QMC approximation (2.4) with a point set $\mathcal{P}_n$, was introduced by Sloan & Woźniakowski [82] and assumes that the integrand f belongs to some s-variate weighted function space $\mathcal{W}_{s,\boldsymbol{\gamma}}$. In the original formulation each variable has an associated "weight" parameter $\gamma_i > 0$ whose size describes the importance of y_i. Weights of this kind $\boldsymbol{\gamma} = (\gamma_i)_{i=1}^{\infty}$ are incorporated into $\mathcal{W}_{s,\boldsymbol{\gamma}}$ through the weighted norm $\|\cdot\|_{s,\boldsymbol{\gamma}}$ and are nowadays referred to as *product weights*.

In general, given a collection of positive real numbers $\boldsymbol{\gamma} = \{\gamma_{\mathfrak{u}}\}_{\mathfrak{u}\subset\mathbb{N}}$, where $\mathfrak{u}$ denotes a finite subset of $\mathbb{N}$ and each $\gamma_{\mathfrak{u}}$ represents the importance of the collections of variables $\boldsymbol{y}_{\mathfrak{u}}$, let $\mathcal{W}_{s,\boldsymbol{\gamma}}$ be the s-dimensional weighted Sobolev space with unanchored norm

$$\|f\|_{s,\boldsymbol{\gamma}} = \left(\sum_{\mathfrak{u}\subseteq\{1:s\}} \frac{1}{\gamma_{\mathfrak{u}}} \int_{[0,1]^{|\mathfrak{u}|}} \left(\int_{[0,1]^{s-|\mathfrak{u}|}} \frac{\partial^{|\mathfrak{u}|}}{\partial \boldsymbol{y}_{\mathfrak{u}}} f(\boldsymbol{y}) \, \mathrm{d}\boldsymbol{y}_{-\mathfrak{u}} \right)^2 \mathrm{d}\boldsymbol{y}_{\mathfrak{u}} \right)^{\frac{1}{2}}, \tag{2.5}$$

where $\{1:s\}$ is shorthand for $\{1,2,\ldots,s\}$ and $\boldsymbol{y}_{-\mathfrak{u}} = (y_j)_{j\in\{1:s\}\setminus\mathfrak{u}}$ denotes *inactive variables*.

In practice it is difficult to work with general weights $\gamma_{\mathfrak{u}}$ and so often weights with some inherent structure are used. The three most common forms are:

1. *product weights* (see [82]) where for some sequence $1 \geq \gamma_1 \geq \gamma_2 \geq \cdots > 0$,

$$\gamma_{\mathfrak{u}} = \prod_{j\in\mathfrak{u}} \gamma_j \,; \tag{2.6}$$

2. *order dependent weights* (see [21]) where each weight depends only on the cardinality of the set:

$$\gamma_{\mathfrak{u}} = \Gamma_{|\mathfrak{u}|} \,, \tag{2.7}$$

   for a sequence of positive real numbers $\Gamma_0 \coloneqq 1, \Gamma_1, \Gamma_2, \ldots$; and

3. *product and order dependent (POD) weights*, which first appeared in [60] and are a hybrid of the previous two with

$$\gamma_{\mathfrak{u}} = \Gamma_{|\mathfrak{u}|} \prod_{j\in\mathfrak{u}} \gamma_j \,. \tag{2.8}$$

In the weighted function space $\mathcal{W}_{s,\boldsymbol{\gamma}}$, the *worst-case error* of $Q_{n,s}(\mathcal{P}_n)$ over the unit ball of $\mathcal{W}_{s,\boldsymbol{\gamma}}$ is defined by

$$e_{n,s,\boldsymbol{\gamma}}(\mathcal{P}_n) \coloneqq \sup_{\|f\|_{s,\boldsymbol{\gamma}} \leq 1} |\mathcal{I}_s f - Q_{n,s}(\mathcal{P}_n) f| \,, \tag{2.9}$$

from which it follows by linearity that the error of a QMC approximation satisfies

$$|\mathcal{I}_s f - Q_{n,s}(\mathcal{P}_n) f| \leq e_{n,s,\boldsymbol{\gamma}}(\mathcal{P}_n) \, \|f\|_{s,\boldsymbol{\gamma}} \,. \tag{2.10}$$

The above error bound on a QMC approximation is particularly useful because it easily splits into two relatively independent factors: the worst-case error, which measures the quality of the point set and is independent of the integrand; and the norm of the integrand, which is often considered fixed since the integrand is determined by the problem given. However, a subtle connection between the two quantities is that they both depend on the function space weights $\boldsymbol{\gamma}$, this will be investigated further in Chapter 6.

The benefits of random shifting outlined in the previous section are not limited to lattice point sets. Indeed, for $\mathcal{P}_n = \{\boldsymbol{t}^{(k)}\}_{k=0}^{n-1}$, given a uniformly distributed random shift $\boldsymbol{\Delta} \in [0,1)^s$ the randomly shifted point set $(\mathcal{P}_n; \boldsymbol{\Delta}) = \{\tilde{\boldsymbol{t}}_k\}_{k=0}^{n-1}$ is obtained by taking $\tilde{\boldsymbol{t}}_k = \left\{\boldsymbol{t}^{(k)} + \boldsymbol{\Delta}\right\}$, where the braces again ensure that the shifted point still belongs to $[0,1]^s$. We will write the shifted QMC approximation as $Q^{\mathrm{sh}}_{n,s}(\mathcal{P}_n; \boldsymbol{\Delta})$.

In this setting, the *shift-averaged worst-case error* is used as a measure of the quality of a point set. It is simply the worst-case error of the shifted point set, averaged in the root-mean-square sense over all possible shifts

$$e^{\mathrm{sh}}_{n,s,\boldsymbol{\gamma}}(\mathcal{P}_n) \coloneqq \sqrt{\mathbb{E}_{\boldsymbol{\Delta}}\left[e^2_{n,s,\boldsymbol{\gamma}}(\mathcal{P}_n; \cdot)\right]} = \sqrt{\int_{[0,1]^s} e^2_{n,s,\boldsymbol{\gamma}}(\mathcal{P}_n; \boldsymbol{\Delta}) \, \mathrm{d}\boldsymbol{\Delta}} \,. \tag{2.11}$$

It then follows from the error bound (2.10) that the root-mean-square error of a shifted QMC approximation (where the expected value is again taken with respect to the shift $\boldsymbol{\Delta}$) satisfies

$$\sqrt{\mathbb{E}_{\boldsymbol{\Delta}}\left[\left|\mathcal{I}_s f - Q^{\mathrm{sh}}_{n,s}\left(\mathcal{P}_n; \cdot\right) f\right|^2\right]} \leq e^{\mathrm{sh}}_{n,s,\boldsymbol{\gamma}}(\mathcal{P}_n) \, \|f\|_{s,\boldsymbol{\gamma}} \,. \tag{2.12}$$

As before, this error bound is useful because the right hand side splits into two factors, which respectively depend on the quadrature points and the integrand.

In practice, to estimate the quadrature error we perform a small number R of approximations that are generated by independent and identically distributed

random shifts, see e.g., [18]. Letting $\boldsymbol{\Delta}^{(r)}$ be the rth random shift, the corresponding QMC approximation is $Q^{\mathrm{sh}}_{n,s}(\mathcal{P}_n;\boldsymbol{\Delta}^{(r)})f$, and then the final estimate is taken to be the average of the R independent approximations

$$Q_{n,s,R}(\mathcal{P}_n)f \;\coloneqq\; \frac{1}{R}\sum_{r=1}^{R} Q^{\mathrm{sh}}_{n,s}(\mathcal{P}_n;\boldsymbol{\Delta}^{(r)})f\,. \tag{2.13}$$

Taking the sample variance over the R approximations gives an estimate of the mean-square error

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{\Delta}}\left[\left|\mathcal{I}_s f - Q^{\mathrm{sh}}_{n,s}(\mathcal{P}_n;\cdot)f\right|^2\right]\\ &\qquad\approx\; \frac{1}{R(R-1)}\sum_{r=1}^{R}\left|Q_{n,s,R}(\mathcal{P}_n)f - Q^{\mathrm{sh}}_{n,s}(\mathcal{P}_n;\boldsymbol{\Delta}^{(r)})f\right|^2.\end{aligned} \tag{2.14}$$

Typically, R is of the order 10-30.

The shift-averaged worst-case error of a randomly shifted lattice rule in the space $\mathcal{W}_{s,\boldsymbol{\gamma}}$ with general weights, is given explicitly by (see [18, Eq. (5.12)])

$$e^{\mathrm{sh}}_{n,s,\boldsymbol{\gamma}}(\boldsymbol{z}) \;=\; \sqrt{\sum_{\emptyset\neq\mathfrak{u}\subseteq\{1:s\}}\gamma_{\mathfrak{u}}\left(\frac{1}{n}\sum_{k=0}^{n-1}\prod_{j\in\mathfrak{u}}\mathscr{B}_2\left(\left\{\frac{kz_j}{n}\right\}\right)\right)}, \tag{2.15}$$

where $\mathscr{B}_2(x) = x^2 - x + \frac{1}{6}$ is the Bernoulli polynomial of degree 2. This provides us with a computable measure of the quality of a randomly shifted lattice rule.

2.3.3 The Component-by-Component construction

The Component-by-Component (CBC) construction, first invented by Korobov [52] and rediscovered in [80, 81], is an efficient method of constructing generating vectors that result in "good" lattice rules in the context of minimising the worst-case error. The CBC construction is a greedy algorithm that works through each component of the generating vector sequentially, choosing z_i to *minimise the shift-averaged worst-case error* (2.15) *in that dimension while all previous components remain fixed.*

Algorithm 2.1 The CBC algorithm

Given n, s and a sequence of weights $\boldsymbol{\gamma} = \{\gamma_{\mathfrak{u}}\}_{\mathfrak{u}\subseteq\{1:s\}}$.

1. Set z_1 to 1.
2. For $i = 2,\ldots,s$ choose $z_i \in \mathbb{U}_n$ so as to minimise $e^{\mathrm{sh}}_{n,i,\boldsymbol{\gamma}}(z_1,\ldots,z_{i-1},z_i)$ given that all of the previous components $z_1,\ldots,z_{i-1}$ remain fixed.

Setting z_1 to be 1 is done by convention, since in the first dimension every choice results in an equivalent quadrature rule. Using structured weights (product, order dependent or POD form) simplifies the formula for the shift-averaged worst-case error (2.15) even further, allowing the calculation of $e^{\mathrm{sh}}_{n,i,\boldsymbol{\gamma}}(z_1,\dots,z_{i-1},z_i)$ for all $z_i \in \mathbb{U}_n$ together to be performed as one matrix-vector product. In general a naive implementation of this algorithm costs $\mathcal{O}(s\,n^2)$ operations, however a fast construction performs the matrix-vector product using a fast Fourier transform (FFT), which reduces this to $\mathcal{O}(s\,n\log n)$ in the case of product weights and $\mathcal{O}(s\,n\log n + s^2 n)$ for order dependent or POD weights. For full details on the *Fast CBC construction* see [18, 69, 70].

A key result in the modern theory of QMC is that the shift-averaged worst-case error of a CBC generated lattice rule satisfies the following upper bound:

$$e^{\mathrm{sh}}_{n,s,\boldsymbol{\gamma}}(\boldsymbol{z}) \;\le\; \left(\frac{1}{\varphi(n)}\sum_{\emptyset\ne\mathfrak{u}\subseteq\{1:s\}}\gamma_{\mathfrak{u}}^{\eta}\left(\frac{2\zeta(2\eta)}{(2\pi^2)^{\eta}}\right)^{|\mathfrak{u}|}\right)^{\frac{1}{2\eta}} \quad \text{for all } \eta\in\left(\tfrac{1}{2},1\right], \tag{2.16}$$

where $\zeta(x)=\sum_{j=1}^{\infty}1/j^x$ for $x>1$ is the Riemann zeta function. This result was proved in [18] for general weights and in [13, 55] for product weights. In many cases, such as when n is prime or a prime power, $1/\varphi(n)=\mathcal{O}(1/n)$ and hence the upper bound (2.16) states that in the setting of weighted function spaces $\mathcal{W}_{s,\boldsymbol{\gamma}}$ the CBC algorithm generates lattice rules that achieve convergence in error that is arbitrarily close to $1/n$.

A crucial strength of QMC rules is that not only do they achieve good convergence rates but they are also well-suited to very high-dimensional problems; however, notice in (2.16) that the upper bound still depends on the dimension through the sum. The strategy to remove this dependence on dimension is to impose conditions on the weights—which equate to restrictions on the functions in $\mathcal{W}_{s,\boldsymbol{\gamma}}$—that ensure that the sum in (2.16) can be bounded independently of s. For example, in the case of product weights (2.6) a common condition is to assume that the weights are summable:

$$\sum_{j=1}^{\infty}\gamma_j \;<\; \infty\,.$$

2.4 Smolyak/sparse grid quadrature rules

Smolyak methods [83], also known more recently as sparse grid rules [10, 27], are efficient tensor product-type quadrature rules that have proven to be very effective in approximating moderate to high-dimensional integrals. The basis of Smolyak's

method is to take the tensor product of one-dimensional quadrature rules, but instead of taking the full tensor product—which is renowned for being extremely expensive as the dimension increases—a Smolyak approximation is a restricted sum of the tensor product of the differences of one-dimensional rules.

More precisely, let $\{U_i\}_{i\geq 0}$ be a sequence of one-dimensional quadrature rules that are designed to approximate integrals of the form

$$\int_{\mathcal{Y}} g(y)\rho(y)\mathrm{d}y\,,$$

with $U_0 \equiv 0$, the zero approximation. It is also commonly assumed that the rules $\{U_i\}_{i\geq 0}$ are ordered such that the number of quadrature points increases with i, which also generally means that the "precision" of the rule increases. A level $m > 0$ Smolyak approximation of the s-dimensional integral (2.1) is

$$\mathcal{Q}_{s,m}(f) \coloneqq \sum_{\substack{\boldsymbol{i}\in\mathbb{N}^s \\ |\boldsymbol{i}|\leq s+m-1}} \left(\bigotimes_{j=1}^{s} \left(U_{i_j} - U_{i_j-1}\right) \right)(f)\,, \tag{2.17}$$

where $\boldsymbol{i} = (i_1, i_2, \ldots, i_s)$ and $|\boldsymbol{i}| = \sum_{j=1}^{s} |i_j|$. The elements i_j of the vector $\boldsymbol{i}$ determine the precision in dimension j of each term in the sum, and the name "sparse grids" refers to the fact that the condition $|\boldsymbol{i}| \leq s + m - 1$ restricts how many dimensions can have higher precision difference rules simultaneously, see Figure 2.2. Most notably, the highest precision difference $(U_m - U_{m-1})$ can only ever be present in one dimension at a time.

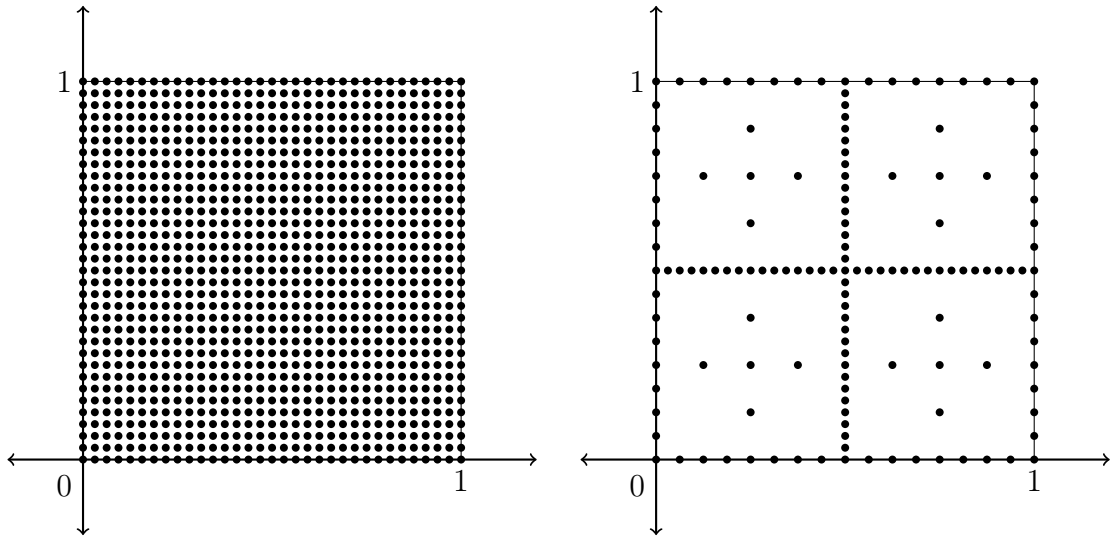


Figure 2.2: 2D quadrature points based on trapezoidal rules for level $m = 6$: full tensor product grid (left), and Smolyak's method (right).

A graphical comparison of the quadrature points used by a full tensor product approximation and a Smolyak approximation in two dimensions is given in Figure 2.2. The grids in each rule are generated by using trapezoidal rules on $\mathcal{Y} = [0, 1]$ as the one-dimensional rules and correspond to level $m = 6$. The full product grid on the left consists of 1089 points whereas the Smolyak grid uses only 145 points—this discrepancy only increases with dimension and level.

The tensor product formula (2.17) can also be written as an explicit weighted quadrature rule, which is easier to compute in practice. This quadrature formula depends on whether the one-dimensional rules are nested and will be presented in Chapter 4.

Also, note that Smolyak's method is not specific to numerical integration. The underlying product-of-differences structure is also very effective for function approximation, again see, e.g., [10].

This summary on Smolyak's method concludes with a typical error bound, which will hold for all integrands in the s-dimensional space of functions with bounded mixed derivatives up to order r:

$$\mathcal{W}_s^r(\mathcal{Y}^s) \coloneqq \left\{ f : \mathcal{Y}^s \to \mathbb{R} : \left\| \frac{\partial^{|\boldsymbol{q}|} f}{\partial y_1^{q_1} \partial y_2^{q_2} \cdots \partial y_s^{q_s}} \right\|_{L^\infty(\mathcal{Y}^s)} < \infty,\ \boldsymbol{q} \in \{0, 1, \ldots, r\} \right\} .$$

First, suppose that the number of points in the one-dimensional rules are of the order $n_i = \mathcal{O}(2^i)$, and that for $g \in \mathcal{C}^r(\mathcal{Y})$, the space of r times continuously differentiable functions on $\mathcal{Y}$, the rule U_i satisfies

$$\left| \int_{\mathcal{Y}} g(y)\rho(y)\,\mathrm{d}y - U_i(g) \right| = \mathcal{O}(n_i^{-r}) = \mathcal{O}(2^{-i\,r}) .$$

Then, it can be shown, see, e.g., [10, 91], that for $f \in \mathcal{W}_s^r(\mathcal{Y}^s)$ Smolyak's methods achieve an approximation error of

$$|\mathcal{I}_s(f) - \mathcal{Q}_{s,m}(f)| = \mathcal{O}\left(m^{(s-1)(r+1)} \cdot 2^{-m\,r}\right) . \tag{2.18}$$

2.5 The Multivariate Decomposition Method

The *Multivariate Decomposition Method* (MDM) is an algorithm for approximating the integral of an infinite-variate function, which works by approximating the infinite-dimensional integral by many low-dimensional integrals that can then be handled more easily by quadrature. The general idea of the MDM, see [31, 33, 58, 76, 89, 90] (as well as [63, 75] under the name of *Changing Dimension Algorithm*),

goes as follows. Assume that f admits a decomposition

$$f(\boldsymbol{y}) = \sum_{\mathfrak{u} \subset \mathbb{N}} f_{\mathfrak{u}}(\boldsymbol{y}_{\mathfrak{u}}), \tag{2.19}$$

where the sum is taken over all finite subsets of $\mathbb{N}$, and where each function $f_{\mathfrak{u}}$ depends only on the variables in $\boldsymbol{y}_{\mathfrak{u}} = (y_j)_{j \in \mathfrak{u}}$. With ρ a given probability density function on $\mathcal{Y}$ and $\rho_{\mathfrak{u}}(\boldsymbol{y}) \coloneqq \prod_{j \in \mathfrak{u}} \rho(y_j)$, we define the integral of f by

$$\mathcal{I}(f) \coloneqq \sum_{\mathfrak{u} \subset \mathbb{N}} \mathcal{I}_{\mathfrak{u}}(f_{\mathfrak{u}}), \qquad \text{with} \qquad \mathcal{I}_{\mathfrak{u}}(f_{\mathfrak{u}}) \coloneqq \int_{\mathcal{Y}^{|\mathfrak{u}|}} f_{\mathfrak{u}}(\boldsymbol{y}_{\mathfrak{u}}) \rho_{\mathfrak{u}}(\boldsymbol{y}_{\mathfrak{u}}) \, \mathrm{d}\boldsymbol{y}_{\mathfrak{u}}, \tag{2.20}$$

and let $\mathcal{I}_{\emptyset}(f_{\emptyset}) \coloneqq f_{\emptyset}$. Under certain conditions the integral above (2.20) coincides with the infinite-dimensional integral as defined in (2.3), see e.g., [58].

The MDM algorithm for approximating the integral (2.20) is

$$\mathcal{A}_{\varepsilon}(f) \coloneqq \sum_{\mathfrak{u} \in \mathcal{U}_{\varepsilon}} A_{\mathfrak{u}}(f_{\mathfrak{u}}), \tag{2.21}$$

where $\mathcal{U}_{\varepsilon}$ is the "active set"—a family of sets $\mathfrak{u} \subset \mathbb{N}$ that represent the terms $\mathcal{I}_{\mathfrak{u}}(f_{\mathfrak{u}})$ that contribute most to the integral—and for $\mathfrak{u} \neq \emptyset$, each $A_{\mathfrak{u}}$ is a $|\mathfrak{u}|$-dimensional quadrature rule using $n_{\mathfrak{u}}$ function evaluations. Since $f_{\emptyset}$ is constant we set $A_{\emptyset}(f_{\emptyset}) \coloneqq f_{\emptyset}$.

The error of the MDM algorithm satisfies the trivial bound

$$|\mathcal{I}(f) - \mathcal{A}_{\varepsilon}(f)| \leq \sum_{\mathfrak{u} \notin \mathcal{U}_{\varepsilon}} |\mathcal{I}_{\mathfrak{u}}(f_{\mathfrak{u}})| + \sum_{\mathfrak{u} \in \mathcal{U}_{\varepsilon}} |\mathcal{I}_{\mathfrak{u}}(f_{\mathfrak{u}}) - A_{\mathfrak{u}}(f_{\mathfrak{u}})|. \tag{2.22}$$

Given $\varepsilon > 0$, the strategy is to first choose an active set $\mathcal{U}_{\varepsilon}$ such that the first sum in (2.22) is at most $\varepsilon/2$, and then specify the quadrature rules such that the second sum in (2.22) is also at most $\varepsilon/2$, giving a total error of at most ε. With this in mind, the subscript ε is included to stress that, by necessity, the MDM algorithm $\mathcal{A}_{\varepsilon}$ and the active set $\mathcal{U}_{\varepsilon}$ depend on the error request ε.

We would need to impose additional conditions on the class of functions to ensure that the sum (2.19) is absolutely convergent, the integral (2.20) is well defined, and the quadrature rules in (2.21) converge appropriately to the corresponding integrals. The precise details will depend on the mathematical setting within which we choose to analyse the problem. We will outline some variants below, but a theoretical study of the MDM is not the purpose of this thesis.

2.5.1 The class of infinite-variate functions

The input parameters for the MDM, which determine the active set and the quadrature rules, depend on the theoretical setting within which we choose to perform the error analysis. There are two broad settings for the MDM in the literature: the weighted function class setting, from e.g., [33, 76, 89, 90], and the setting without weights, as in [58]. In the weighted setting, it is assumed that the integrand belongs to a function class equipped with a weighted norm and that the importance of subsets of variables are represented by the weights, which act as the algorithm inputs. Whereas in the setting without weights, instead of weights it is assumed that bounds on the norms of the decompositions functions $f_{\mathfrak{u}}$ are known and given as inputs. In this introductory chapter we follow the setting without weights, or "norm-bounds setting", and now give a brief summary of the key concepts that will be required in later chapters, for further details see [58]. The MDM in the context of the weighted setting will be introduced in Chapter 5.

The class of infinite-variate functions will be denoted $\mathcal{F}$ and we will outline its structure now. For each $\mathfrak{u} \subset \mathbb{N}$ let $F_{\mathfrak{u}}$ be a Banach space of functions of $|\mathfrak{u}|$ variables with norm denoted by $\|\cdot\|_{F_{\mathfrak{u}}}$, and assume that the functions in $F_{\mathfrak{u}}$ are integrable with respect to $\mathcal{I}_{\mathfrak{u}}$. We define $F_{\emptyset}$ to be the space of constant functions. Then $\mathcal{F}$ is defined to be the class of infinite-variate functions that admit a decomposition (2.19), with each decomposition function $f_{\mathfrak{u}}$ belonging to $F_{\mathfrak{u}}$. Further, assume that for each $\mathfrak{u}$ there exists some $0 < B_{\mathfrak{u}} < \infty$, which is either known or computable, such that

$$\|f_{\mathfrak{u}}\|_{F_{\mathfrak{u}}} \leq B_{\mathfrak{u}}\,, \tag{2.23}$$

and that the collection of bounds $\{B_{\mathfrak{u}}\}_{\mathfrak{u}\subset\mathbb{N}}$ is summable:

$$\sum_{\mathfrak{u}\subset\mathbb{N}} B_{\mathfrak{u}} < \infty\,. \tag{2.24}$$

In order to ensure that the integral (2.20) is well-defined we make further assumptions on $f \in \mathcal{F}$. Assume that each $\mathcal{I}_{\mathfrak{u}} : F_{\mathfrak{u}} \to \mathbb{R}$ is continuous with

$$C_{\mathfrak{u}} := \|\mathcal{I}_{\mathfrak{u}}\|_{F_{\mathfrak{u}}} := \sup_{g_{\mathfrak{u}}\in F_{\mathfrak{u}},\, \|g_{\mathfrak{u}}\|_{F_{\mathfrak{u}}}\leq 1} |\mathcal{I}_{\mathfrak{u}}(g_{\mathfrak{u}})| < \infty\,. \tag{2.25}$$

Finally, we assume that there exists $\alpha > 1$ such that

$$\sum_{\mathfrak{u}\subset\mathbb{N}} (C_{\mathfrak{u}} B_{\mathfrak{u}})^{1/\alpha} < \infty\,, \tag{2.26}$$

and define the maximum decay to be

$$\alpha_0 \coloneqq \operatorname{decay}\left(\{C_{\mathfrak{u}}B_{\mathfrak{u}}\}_{\mathfrak{u}\subset\mathbb{N}}\right) \coloneqq \sup\left\{\alpha : \sum_{\mathfrak{u}\subset\mathbb{N}}(C_{\mathfrak{u}}B_{\mathfrak{u}})^{1/\alpha} < \infty\right\} > 1\,. \tag{2.27}$$

Then (2.26) implies that for all $f \in \mathcal{F}$ the integral (2.20) is well-defined. Actually, the integral is still well-defined with $\alpha = 1$, however, the definition of the active set (see (2.33) below) requires $\alpha > 1$, again see [58].

The two collections of bounds $\{B_{\mathfrak{u}}\}_{\mathfrak{u}\subset\mathbb{N}}$ and $\{C_{\mathfrak{u}}\}_{\mathfrak{u}\subset\mathbb{N}}$ must be known because they are the inputs to the MDM, that is, they are required as inputs when we choose the active set and the quadrature rules.

2.5.2 The anchored decomposition

Rarely in practice is the integrand given in the form of a decomposition (2.19), however, there are known decomposition formulas that provide a way to compute the terms $f_{\mathfrak{u}}$, see [64] for a discussion. Perhaps the two most well-known decompositions are the ANOVA (Analysis of Variance) decomposition, see e.g., [85], and the anchored decomposition. The general idea of both decompositions for evaluating $f_{\mathfrak{u}}$ is to apply successive projections to f, where each projection eliminates the dependence on a variable y_j for $j \notin \mathfrak{u}$, see [64]. The ANOVA decomposition eliminates variables by integrating them out (and as such is not of much practical use in the context of MDM), whereas the anchored decomposition fixes a variable at a specific value in $\mathcal{Y}$. Thus, the anchored decomposition allows us to evaluate $f_{\mathfrak{u}}$ using only evaluations of the original integrand, and will be the decomposition of choice for the remainder of this thesis. The following formula from [64] allows us to explicitly compute the anchored decomposition term $f_{\mathfrak{u}}$ via

$$f_{\mathfrak{u}}(\boldsymbol{y}_{\mathfrak{u}}) \;=\; \sum_{\mathfrak{v}\subseteq\mathfrak{u}}(-1)^{|\mathfrak{u}|-|\mathfrak{v}|}f(\boldsymbol{y}_{\mathfrak{v}};\boldsymbol{a})\,, \tag{2.28}$$

where the "anchor" is $\boldsymbol{a} = (a_1, a_2, a_3, \ldots) \in \mathcal{Y}^{\mathbb{N}}$ and the jth element of the anchored point $(\boldsymbol{y}_{\mathfrak{v}};\boldsymbol{a})$ is given by

$$(\boldsymbol{y}_{\mathfrak{v}};\boldsymbol{a})_j \coloneqq \begin{cases} y_j & \text{if } j \in \mathfrak{v}\,, \\ a_j & \text{otherwise.} \end{cases}$$

Another practical consideration is how to choose the anchor so that evaluating f at an anchored point—which is still an infinite-dimensional vector—is actually possible. Often there is a natural choice of anchor, and in the section to follow we introduce an example integration problem where the choice $\boldsymbol{a} = \mathbf{0}$ is obvious.

2.5.3 An example integration problem

In this section we give an example of how to set up the framework for applying the MDM to numerically integrate the function (2.29) below. Namely, we explain how to choose the spaces $F_{\mathfrak{u}}$ and ultimately the input parameters $\{B_{\mathfrak{u}}\}_{\mathfrak{u}\subset\mathbb{N}}$, $\{C_{\mathfrak{u}}\}_{\mathfrak{u}\subset\mathbb{N}}$. Let $\mathcal{Y} = [-\frac{1}{2}, \frac{1}{2}]$, $\rho \equiv 1$ and consider the integrand $f : [-\frac{1}{2}, \frac{1}{2}]^{\mathbb{N}} \to \mathbb{R}$ given by

$$f(\boldsymbol{y}) \;=\; \frac{1}{1+\sum_{j\geq 1} y_j/j^{\beta}}\,, \tag{2.29}$$

with different parameters $\beta \geq 2$. This integrand is considered a prototype function for some PDE applications, see e.g., [60], and, for $\beta = 2$, has previously been considered in, e.g., [58, Example 5]. It will serve as the test integrand for our numerical results when we consider implementing the MDM in Chapter 4.

The decomposition terms of (2.29) will be computed using the anchored decomposition (2.28) with anchor $\mathbf{0} \coloneqq (0, 0, 0\ldots)$. By taking $\mathbf{0}$ as the anchor, the anchored variables simply drop out of the sum in the denominator and for $\mathfrak{v} \subset \mathbb{N}$ we can compute f at an anchored point $(\boldsymbol{y}_{\mathfrak{v}}; \mathbf{0})$ simply using

$$f(\boldsymbol{y}_{\mathfrak{v}}; \mathbf{0}) = \frac{1}{1+\sum_{j\in\mathfrak{v}} y_j/j^{\beta}}\,.$$

Let each $F_{\mathfrak{u}}$ be a reproducing kernel Hilbert space with norm

$$\|g_{\mathfrak{u}}\|_{F_{\mathfrak{u}}} \;\coloneqq\; \left(\int_{[-\frac{1}{2},\frac{1}{2}]^{|\mathfrak{u}|}} \left(\frac{\partial^{|\mathfrak{u}|}}{\partial \boldsymbol{y}_{\mathfrak{u}}} g_{\mathfrak{u}}(\boldsymbol{y}_{\mathfrak{u}})\right)^2 \mathrm{d}\boldsymbol{y}_{\mathfrak{u}}\right)^{\frac{1}{2}}, \tag{2.30}$$

which for $f_{\mathfrak{u}}$ given by the anchored decomposition (2.28) simplifies to

$$\|f_{\mathfrak{u}}\|_{F_{\mathfrak{u}}} \;\coloneqq\; \left(\int_{[-\frac{1}{2},\frac{1}{2}]^{|\mathfrak{u}|}} \left(\frac{\partial^{|\mathfrak{u}|}}{\partial \boldsymbol{y}_{\mathfrak{u}}} f(\boldsymbol{y}_{\mathfrak{u}}; \mathbf{0})\right)^2 \mathrm{d}\boldsymbol{y}_{\mathfrak{u}}\right)^{\frac{1}{2}}. \tag{2.31}$$

In $F_{\mathfrak{u}}$ we have that $\|\mathcal{I}_{\mathfrak{u}}\| = 12^{-|\mathfrak{u}|/2} =: C_{\mathfrak{u}}$, and for the integrand (2.29) the norm is bounded by (see [58, Subsection 5.4] for a derivation of the case $\beta = 2$)

$$\|f_{\mathfrak{u}}\|_{F_{\mathfrak{u}}} \;\leq\; \left(1 - \tfrac{1}{2}\zeta(\beta)\right)^{-(|\mathfrak{u}|+1)} |\mathfrak{u}|! \prod_{j\in\mathfrak{u}} j^{-\beta} \;=:\; B_{\mathfrak{u}}, \tag{2.32}$$

where $\zeta(x)$ is again the Riemann zeta function.

Clearly, the summability assumptions (2.24) and (2.26) are satisfied for all $\beta \geq 2$ and the maximum decay (2.27) is $\alpha_0 = \beta$.

2.5.4 Constructing the active set

The following criteria for constructing the active set $\mathcal{U}_\varepsilon$ from [58, 75] ensures that the truncation error (the first term in (2.22)) is less than $\varepsilon/2$ as required. Let the active set be given by

$$\mathcal{U}_\varepsilon = \{\mathfrak{u} : C_\mathfrak{u} B_\mathfrak{u} > T\}\,, \tag{2.33}$$

where for any $\alpha \in (1, \alpha_0)$ the threshold is

$$T \coloneqq \left(\frac{\varepsilon/2}{\sum_{\mathfrak{v}\subset\mathbb{N}} (C_\mathfrak{v} B_\mathfrak{v})^{1/\alpha}}\right)^{\frac{\alpha}{\alpha-1}}\,. \tag{2.34}$$

Then it was shown in [89] that

$$\sum_{\mathfrak{u}\notin\mathcal{U}_\varepsilon} |\mathcal{I}_\mathfrak{u}(f_\mathfrak{u})| \le \frac{\varepsilon}{2}$$

and the size of the active set is controlled by

$$|\mathcal{U}_\varepsilon| < \left(\frac{2}{\varepsilon}\right)^{\frac{1}{\alpha-1}} \left(\sum_{\mathfrak{u}\subset\mathbb{N}} (C_\mathfrak{u} B_\mathfrak{u})^{1/\alpha}\right)^{\frac{\alpha}{\alpha-1}} < \infty\,. \tag{2.35}$$

Note that for a given integration problem the choice of the active set is not unique.

2.5.5 Choosing the quadrature rules

The strategy for choosing the algorithms $A_\mathfrak{u}$ is to select the number of points $n_\mathfrak{u}$ that minimise the total cost of (2.21) while ensuring that the quadrature error (the second term in (2.22)) is bounded by $\varepsilon/2$. To that end we introduce the cost model from [58].

For finite $\mathfrak{u} \subset \mathbb{N}$ and arbitrary $\boldsymbol{y}_\mathfrak{u} \in \mathcal{Y}^{|\mathfrak{u}|}$, let the cost of evaluating $f_\mathfrak{u}(\boldsymbol{y}_\mathfrak{u})$ be given by $\pounds(|\mathfrak{u}|)$ where the function $\pounds : \mathbb{N}\cup\{0\} \to (0,\infty)$ is non-decreasing. Then the *information cost* of the algorithm $\mathcal{A}_\varepsilon$ is defined to be

$$\mathrm{cost}(\mathcal{A}_\varepsilon) \coloneqq \sum_{\mathfrak{u}\in\mathcal{U}_\varepsilon} n_\mathfrak{u} \pounds(|\mathfrak{u}|)\,. \tag{2.36}$$

In the case of the anchored decomposition, if the cost of evaluating $f(\boldsymbol{y}_\mathfrak{u}; \mathbf{0})$ is given by $\$(|\mathfrak{u}|)$ where $\$: \mathbb{N}\cup\{0\} \to (0,\infty)$ is a non-decreasing function, then based on the formula (2.28)

$$\pounds(k) = \sum_{\ell=0}^{k} \$(k) \binom{k}{\ell}\,.$$

We will also require knowledge of the error of the linear algorithms $\{A_{\mathfrak{u}}\}_{\mathfrak{u}\subset\mathbb{N}}$ in the spaces $\{F_{\mathfrak{u}}\}_{\mathfrak{u}\subset\mathbb{N}}$. For all $\mathfrak{u}\subset\mathbb{N}$, assume that there there exists a single $q>0$ and known real numbers $0<G_{\mathfrak{u},q}<\infty$ such that the algorithm $A_{\mathfrak{u},n}$ (the extra subscript denotes using n function evaluations) satisfies the worst-case error bound

$$\sup_{g_{\mathfrak{u}}\in F_{\mathfrak{u}},\|g_{\mathfrak{u}}\|_{F_{\mathfrak{u}}}\le 1}|\mathcal{I}_{\mathfrak{u}}(g_{\mathfrak{u}})-A_{\mathfrak{u},n}(g_{\mathfrak{u}})| \;\le\; \frac{G_{\mathfrak{u},q}}{(n+1)^q} \quad \text{for } n=0,1,2,\ldots\,. \tag{2.37}$$

It then follows that the total quadrature error is bounded above as follows

$$\left|\sum_{\mathfrak{u}\in\mathcal{U}_\varepsilon}\mathcal{I}_{\mathfrak{u}}(f_{\mathfrak{u}})-\mathcal{A}_\varepsilon(f)\right| \;\le\; \sum_{\mathfrak{u}\in\mathcal{U}_\varepsilon}\frac{G_{\mathfrak{u},q}B_{\mathfrak{u}}}{(n_{\mathfrak{u}}+1)^q}\,. \tag{2.38}$$

The number of points $n_{\mathfrak{u}}$ are now chosen so as to minimise the information cost (2.36) subject to the right hand side of the upper bound (2.38) equalling the error request $\varepsilon/2$. The real number solution to this constrained minimisation problem is obtained by Lagrange multipliers and is given by

$$h_{\mathfrak{u}} \;=\; \left(\frac{2}{\varepsilon}\sum_{\mathfrak{v}\in\mathcal{U}_\varepsilon}\pounds(|\mathfrak{v}|)^{q/(q+1)}(G_{\mathfrak{v},q}B_{\mathfrak{v}})^{1/(q+1)}\right)^{1/q}\left(\frac{G_{\mathfrak{u},q}B_{\mathfrak{u}}}{\pounds(|\mathfrak{u}|)}\right)^{1/(1+q)}\,. \tag{2.39}$$

Then letting each $A_{\mathfrak{u}}$ be a quadrature rule using $n_{\mathfrak{u}}=\lfloor h_{\mathfrak{u}}\rfloor$ points it follows that

$$\left|\sum_{\mathfrak{u}\in\mathcal{U}_\varepsilon}\mathcal{I}_{\mathfrak{u}}(f_{\mathfrak{u}})-\mathcal{A}_\varepsilon(f)\right| \;\le\; \frac{\varepsilon}{2}\,.$$

Chapter 3

Applying Quasi-Monte Carlo to a stochastic eigenvalue problem

3.1 Introduction

In this chapter we propose methods for solving random second-order elliptic eigenvalue problems (EVP) of the general form

$$-\nabla \cdot \big(a(\boldsymbol{x},\boldsymbol{y})\,\nabla u(\boldsymbol{x},\boldsymbol{y})\big) + b(\boldsymbol{x},\boldsymbol{y})\,u(\boldsymbol{x},\boldsymbol{y}) = \lambda(\boldsymbol{y})\,c(\boldsymbol{x})\,u(\boldsymbol{x},\boldsymbol{y}), \quad \text{for } \boldsymbol{x} \in D, \quad (3.1)$$

where the derivative operator ∇ is with respect to the *physical variable* $\boldsymbol{x}$ and where the *stochastic parameter*

$$\boldsymbol{y} \;=\; (y_j)_{j\in\mathbb{N}} \in U \coloneqq [-\tfrac{1}{2},\tfrac{1}{2}]^{\mathbb{N}} \quad (3.2)$$

is an infinite-dimensional vector of independently and identically distributed uniform random variables on $[-\frac{1}{2},\frac{1}{2}]$. For simplicity, the physical domain $D \subset \mathbb{R}^d$, for $d = 1, 2, 3$, is assumed to be a bounded convex domain with Lipschitz boundary. To guarantee well-posedness of the eigenvalue problem (3.1), we impose homogenous Dirichlet boundary conditions:

$$u(\boldsymbol{x},\boldsymbol{y}) = 0 \quad \text{for} \quad \boldsymbol{x} \in \partial D. \quad (3.3)$$

Under the initial assumption that $a(\cdot,\boldsymbol{y}), b(\cdot,\boldsymbol{y}), c \in L^\infty(D)$, together with

$$a(\boldsymbol{x},\boldsymbol{y}),\; c(\boldsymbol{x}) \geq a_{\min} \;>\; 0 \quad \text{for all } (\boldsymbol{x},\boldsymbol{y}) \in D \times U\,,$$

the eigenvalues in (3.1) are real and bounded from below (this is a simple extension of the results in [87, Sec. 3.2]), and the leftmost (or *dominant*) eigenvalue λ_1 is simple. Since the coefficients $a(\boldsymbol{x},\boldsymbol{y})$ and $b(\boldsymbol{x},\boldsymbol{y})$ depend on the stochastic parameters, the eigenvalues $\lambda(\boldsymbol{y})$ and corresponding eigenfunctions $u(\boldsymbol{x},\boldsymbol{y})$ will also be stochastic.

Problems of the form (3.1) appear in many areas of engineering and physics. Two prominent examples are in nuclear reactor physics [23, 49, 77, 87] and in photonics [22, 29, 54, 68]. Problems of a similar type also appear in quantum physics, in accoustic, electromagnetic or elastic wave propagation, and in structural mechanics, where there is a huge engineering literature on the topic (see e.g. [28, 74]).

In nuclear reactor physics, the eigenproblem (3.1) corresponds to the mono-energetic diffusion approximation of the neutron transport equation [23, 49, 77, 87]. The dominant eigenvalue λ_1 of (3.1) describes the criticality of the reactor, while the corresponding eigenfunction u_1 models the associated neutron flux. The coefficient functions a, b and c correspond, respectively, to the diffusion coefficient, the absorption cross section and the fission cross section of the various materials in the reactor. These coefficients can vary strongly in $\boldsymbol{x}$ and are subject to uncertainty in the composition of the constituent materials (e.g. liquid and vapour in the coolant), due to wear (e.g. "burnt" fuel) and due to geometric deviations from the original reactor design [3, 4, 92, 93].

In photonic band gap calculations in translationally invariant materials, e.g. in photonic crystal fibres (PCFs), two decoupled eigenproblems of the type (3.1) have to be solved (with periodic boundary conditions): the transverse magnetic (TM) and the transverse electric (TE) mode problem [22, 29, 54, 68]. Here, $b \equiv 0$ and we have either $a \equiv 1$ and $c = \kappa^2$ (TM mode problem) or $a = 1/\kappa^2$ and $c \equiv 1$ (TE mode problem), where $\kappa = \kappa(\boldsymbol{x}, \boldsymbol{y})$ is the refractive index of the PCF. The refractive index can be subject to uncertainty, due to heterogeneities or impurities in the material and due to geometric variations [24, 95].

The current chapter demonstrates the power of Quasi-Monte Carlo (QMC) methods for computing statistics of the eigenvalues of (3.1) with boundary conditions (3.3). Our analysis will be restricted to approximating the expected value of the dominant eigenvalue λ_1 and linear functionals of the corresponding eigenfunction, but the method is applicable much more generally, and in particular to the applications listed above. We assume that, for all $\boldsymbol{x} \in D$ and $\boldsymbol{y} \in U$, the stochastic coefficients admit series expansions of the following form:

$$a(\boldsymbol{x}, \boldsymbol{y}) \;=\; a_0(\boldsymbol{x}) + \sum_{j=1}^{\infty} y_j a_j(\boldsymbol{x}) \quad \text{and} \quad b(\boldsymbol{x}, \boldsymbol{y}) \;=\; b_0(\boldsymbol{x}) + \sum_{j=1}^{\infty} y_j b_j(\boldsymbol{x})\,. \tag{3.4}$$

Although the fields a and b are parametrised by the same infinite sequence of random variables $(y_j)_{j\geq 0}$, this setting for the coefficients allows complete flexibility with respect to the correlation between a and b. To model two fields that are not correlated with each other, it suffices to set $a_{2j-1} \equiv 0$ and $b_{2j} \equiv 0$, for all $j \geq 1$. On the

other hand, if there exists a $j \geq 1$ such that $a_j \not\equiv 0$ and $b_j \not\equiv 0$ then the two random fields will be correlated.

For a function $f : U \to \mathbb{R}$, its expected value with respect to the product uniform probability distribution is the infinite-dimensional integral as defined by:

$$\int_{[-\frac{1}{2},\frac{1}{2}]^{\mathbb{N}}} f(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} \coloneqq \lim_{s\to\infty} \int_{[-\frac{1}{2},\frac{1}{2}]^s} f\left(y_1,\ldots,y_s,0,\ldots\right)\mathrm{d}y_1\cdots\mathrm{d}y_s\,,$$

provided that the limit exists. Our quantity of interest is then

$$\mathbb{E}_{\boldsymbol{y}}\left[\lambda_1\right] = \int_{[-\frac{1}{2},\frac{1}{2}]^{\mathbb{N}}} \lambda_1(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y}\,. \tag{3.5}$$

Our strategy for approximating (3.5) is to first truncate the expansions in (3.4) to s parameters by setting $y_j = 0$, for $j > s$, thus reducing (3.5) to a finite-dimensional quadrature problem. Then, for each $\boldsymbol{y} \in [-\frac{1}{2},\frac{1}{2}]^s$, we approximate (3.1) using a finite element (FE) discretisation on a mesh $\mathscr{T}_h$ with mesh size h, to obtain a parametrised generalised matrix eigenproblem that can be solved iteratively (e.g. via an Arnoldi method or similar). The corresponding approximate dominant eigenvalue is denoted $\lambda_{1,s,h}(\boldsymbol{y})$. Now to approximate the integral in (3.5), we construct n suitable QMC quadrature points in the s-dimensional unit cube $[0,1]^s$ via a rank-1 lattice rule with generating vector $\boldsymbol{z} \in \mathbb{N}^s$ (cf. [18]). The entire pointset is shifted by a uniformly-distributed random shift $\boldsymbol{\Delta} \in [0,1]^s$ and then translated into the cube $[-\frac{1}{2},\frac{1}{2}]^s$. The final estimate of $\mathbb{E}_{\boldsymbol{y}}\left[\lambda_1\right]$ is then the (equal-weight) average of the approximate eigenvalues $\lambda_{1,s,h}$ at these n shifted QMC quadrature points and is denoted $Q^{\mathrm{sh}}_{n,s}(\boldsymbol{z},\boldsymbol{\Delta})\lambda_{1,s,h}$.

The error depends on h, s and n and to estimate it we make some further assumptions on the coefficients, which are all detailed in Assumption A3.1. In particular, to bound the FE error (w.r.t. h), we require some spatial regularity of $(a_j)_{j\geq 0}$, $(b_j)_{j\geq 0}$ and c. To bound the dimension-truncation error (w.r.t. s) and the quadrature error (w.r.t. n), we assume p-summability of the sequences $\left(\|a_j\|_{L^\infty(D)}\right)_{j\geq 0}$ and $\left(\|b_j\|_{L^\infty(D)}\right)_{j\geq 0}$, for some $p \in (0,1)$.

The main result we present in this chapter is that, under these assumptions, there exists a constant independent of h, s and n such that

$$\sqrt{\mathbb{E}_{\boldsymbol{\Delta}}\left[\left|\mathbb{E}_{\boldsymbol{y}}[\lambda_1] - Q^{\mathrm{sh}}_{n,s}(\boldsymbol{z},\boldsymbol{\Delta})\lambda_{1,s,h}\right|^2\right]} \leq C\left(h^2 + s^{-2(\frac{1}{p}-1)} + n^{-\alpha}\right), \tag{3.6}$$

where $\alpha = \min(1-\delta, 1/p - 1/2)$ for arbitrary $\delta \in (0, 1/2)$. This result, for which a full statement is given in Theorem 3.15 (along with a similar result for linear

functionals of the corresponding eigenfunction), summarises the individual contributions to the overall error from the three approximations, that is, discretisation (h), dimension-truncation (s) and quadrature (n). The errors in the three separate processes are established individually in Theorems 3.6, 3.12 and 3.13, respectively. To give a simple example of the power of estimate (3.6), if p is small enough, then the truncation error is negligible and the error is bounded by the optimal FE convergence rate h^2 plus a QMC convergence rate which is arbitrarily close to $1/n$. Importantly, the constant C does not depend on s.

A key result in obtaining (3.6) is Lemma 3.10, where we establish the regularity of λ_1 and u_1 with respect to $\boldsymbol{y}$. The bounds on the mixed partial derivatives $|\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} \lambda_1(\boldsymbol{y})|$ of λ_1 are of product and order-dependent (POD) form, as in the case of (linear) boundary value problems [60]. Here, $\boldsymbol{\nu}$ is a multi-index with finitely many non-zero entries $\nu_j \in \mathbb{N}$. The order dependence of the bounds in Lemma 3.10 is $(|\boldsymbol{\nu}|!)^{1+\epsilon}$, for ϵ arbitrarily close to zero, which is only slightly larger than in the bounds in [60] and still allows us to achieve the (nearly) optimal dimension-independent QMC convergence rates. The constants in the bounds depend on the gap between $\lambda_1(\boldsymbol{y})$ and the second smallest eigenvalue $\lambda_2(\boldsymbol{y})$. This is bounded away from zero uniformly in $\boldsymbol{y}$ under the assumptions which we shall make on a, b and c (cf. Assumption A3.1).

Although stochastic eigenproblems have been of interest in engineering for some time, the mathematical literature is less developed. A common method of tackling these problems is the *reduced basis method* [25, 48, 66, 73], whereby the full parametric solution (eigenvalue) is approximated in a low-dimensional subspace that is constructed as the span of the solution at specifically chosen parameter values. For the current work the most relevant paper is [1], where a sparse tensor approximation was used to estimate the expected value of the eigenvalue. A key result there is that simple eigenpairs are analytic with respect to the stochastic parameters, shown using the classical perturbation theory of Kato [50]. However, no bounds on the derivatives are given, which are required to theoretically justify the application of QMC rules. Here, we extend the results from [1] by proving explicit bounds on the derivatives, which in turn allows us to derive *a priori* error bounds. Alternatively, this chapter can also be viewed as extending the results on QMC methods for stochastic elliptic source problems [60] to eigenvalue problems, and we remark that because of the nonlinearity of eigenvalue problems this is not merely a trivial extension. Despite the increased difficulty of this nonlinearity, for all $p \in (0, 1)$, our error bound (3.6) achieves the same rates of convergence as the equivalent result for the PDE source problem in [60]. The only difference is that our result does not hold

for $p = 1$, in which case the result in [60] requires an additional assumption on the summability anyway.

The structure of this chapter is as follows. In Section 3.2, we provide some relevant background theory of elliptic eigenproblems and of randomised lattice rules and establish the FE error bound. Section 3.3 contains the parametric regularity analysis, which is then used in Section 3.4 to bound the quadrature and the truncation error. This chapter concludes with a brief numerical experiment in Section 3.5, which demonstrates the sharpness of the bounds.

3.2 Preliminary theory

In this section we present some preliminary theory on variational eigenvalue problems, FE discretisation and QMC methods. However, we will only very briefly touch on QMC theory since we have already provided a summary in Section 2.3. First we outline all of our assumptions on the coefficients, which, in particular, ensure that the problem (3.1) is well-posed.

Assumption A3.1.

1. *a and b are of the form* (3.4) *with $a_j,\ b_j, \in L^\infty(D)$ for all $j \geq 0$ and $c \in L^\infty(D)$.*
2. *There exists $a_{\min} > 0$ such that $a(\boldsymbol{x}, \boldsymbol{y}) \geq a_{\min}$, $b(\boldsymbol{x}, \boldsymbol{y}) \geq 0$ and $c(\boldsymbol{x}) \geq a_{\min}$ for all $\boldsymbol{x} \in D$, $\boldsymbol{y} \in U$.*
3. *For some $p \in (0, 1]$*

$$\sum_{j=1}^{\infty} \|a_j\|_{L^\infty(D)}^p < \infty \quad \text{and} \quad \sum_{j=1}^{\infty} \|b_j\|_{L^\infty(D)}^p < \infty\,.$$

4. *$a_j \in W^{1,\infty}(D)$ for $j \geq 0$ and*

$$\sum_{j=1}^{\infty} \|a_j\|_{W^{1,\infty}(D)} < \infty\,.$$

The assumption that $b(\boldsymbol{x}, \boldsymbol{y}) \geq 0$ is made without loss of generality because any EVP with $b < 0$, but satisfying the rest of Assumption A3.1, can be shifted to an equivalent problem with "new b" non-negative by adding a constant multiple σ of $c(\boldsymbol{x}) \cdot u(\boldsymbol{x}, \boldsymbol{y})$ to both sides of (3.1). To ensure non-negativity, the shifting factor σ is chosen so that $-b(\boldsymbol{x}, \boldsymbol{y}) \leq \sigma \cdot c(\boldsymbol{x})$ for all $\boldsymbol{x}$, $\boldsymbol{y}$. Such a σ exists due to Assumption A3.1.3. The eigenvalues of the original EVP are simply the eigenvalues of the shifted problem shifted by $-\sigma$, and the corresponding eigenspaces are unchanged. For specific details see, e.g., [29].

Throughout the chapter, when it is unambiguous we will drop the $\boldsymbol{x}$-dependence when referring to a function defined on D at a parameter value $\boldsymbol{y}$. For example, we will write the coefficients and eigenfunctions as $a(\boldsymbol{y}) \coloneqq a(\cdot, \boldsymbol{y})$, $b(\boldsymbol{y}) \coloneqq b(\cdot, \boldsymbol{y})$ and $u(\boldsymbol{y}) \coloneqq u(\cdot, \boldsymbol{y})$.

Assumption A3.1 (1, 2 and 3) implies that for all $\boldsymbol{y} \in U$ we have $a(\boldsymbol{y})$, $b(\boldsymbol{y}) \in L^\infty(D)$. Furthermore, by the triangle inequality, the L^∞-norms of these two coefficients can be bounded from above independently of $\boldsymbol{y}$: for all $\boldsymbol{x}, \boldsymbol{y}$

$$\|a(\boldsymbol{y})\|_{L^\infty(D)} \leq \|a_0\|_{L^\infty(D)} + \frac{1}{2}\sum_{j=1}^{\infty} \|a_j\|_{L^\infty(D)} ,$$

and similarly for $\|b(\boldsymbol{y})\|_{L^\infty(D)}$. For convenience we define a single upper bound for all three coefficients

$$a_{\max} \coloneqq \max\left\{\sup_{\boldsymbol{y}\in U} \|a(\boldsymbol{y})\|_{L^\infty(D)} , \sup_{\boldsymbol{y}\in U} \|b(\boldsymbol{y})\|_{L^\infty(D)} , \|c\|_{L^\infty(D)}\right\} . \tag{3.7}$$

In Assumption A3.1.4, $W^{1,\infty}(D)$ is the usual Sobolev space of functions with essentially bounded gradient on D, to which we attach the norm

$$\|v\|_{W^{1,\infty}(D)} \coloneqq \max\left\{\|v\|_{L^\infty(D)} , \|\nabla v\|_{L^\infty(D)}\right\} .$$

This assumption is needed to obtain the regularity result in Proposition 3.1.

3.2.1 Abstract theory for variational eigenproblems

To construct the variational formulation of the EVP (3.1), we introduce the test space $V \coloneqq H_0^1(D)$, the usual first-order Sobolev space of real-valued functions with vanishing boundary trace, its dual $V^* = H^{-1}(D)$, and equip V with the norm

$$\|v\|_V \coloneqq \|\nabla v\|_{L^2(D)} .$$

We identify $L^2(D)$ with its dual and note that we have the following compact embeddings $V \subset L^2(D) \subset V^*$. The $L^2(D)$ inner product is denoted by $\langle \cdot, \cdot \rangle$ and we use the same notation for the extension to the duality pairing on $V \times V^*$. Multiplying the eigenproblem (3.1) by $v \in V$ and integrating (by parts) over D, we obtain the variational formulation

$$\begin{aligned} &\int_D \big(a(\boldsymbol{x}, \boldsymbol{y})\nabla u(\boldsymbol{x}, \boldsymbol{y}) \cdot \nabla v(\boldsymbol{x}) + b(\boldsymbol{x}, \boldsymbol{y})u(\boldsymbol{x}, \boldsymbol{y})v(\boldsymbol{x})\big) \,\mathrm{d}\boldsymbol{x} \\ &\qquad = \lambda(\boldsymbol{y}) \int_D c(\boldsymbol{x})u(\boldsymbol{x}, \boldsymbol{y})v(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x} \quad \text{for all } v \in V . \end{aligned} \tag{3.8}$$

Correspondingly, for each $\boldsymbol{y}$ we define the symmetric bilinear forms $\mathcal{B}(\boldsymbol{y};\cdot,\cdot) : V \times V \to \mathbb{R}$ by

$$\mathcal{B}(\boldsymbol{y};w,v) \coloneqq \int_D \big(a(\boldsymbol{x},\boldsymbol{y})\nabla w(\boldsymbol{x})\cdot\nabla v(\boldsymbol{x}) + b(\boldsymbol{x},\boldsymbol{y})w(\boldsymbol{x})v(\boldsymbol{x})\big)\,\mathrm{d}\boldsymbol{x}\,, \tag{3.9}$$

and $\mathcal{D}(\cdot,\cdot) : V \times V \to \mathbb{R}$ by

$$\mathcal{D}(w,v) \coloneqq \int_D c(\boldsymbol{x})w(\boldsymbol{x})v(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\,, \tag{3.10}$$

which are both inner products on their respective domains. Again we will use the same notation for the corresponding duality pairings on $V \times V^*$. The induced norm for $\mathcal{D}$ is denoted

$$\|v\|_{\mathcal{D}} = \sqrt{\mathcal{D}(v,v)}$$

and is equivalent to the $L^2(D)$-norm:

$$\sqrt{a_{\min}}\,\|v\|_{L^2(D)} \le \|v\|_{\mathcal{D}} \le \sqrt{a_{\max}}\,\|v\|_{L^2(D)}\,, \quad \text{for } v \in L^2(D)\,. \tag{3.11}$$

For each $\boldsymbol{y} \in U$, the variational eigenproblem equivalent to (3.1) is then: Find $0 \neq u(\boldsymbol{y}) \in V$ and $\lambda(\boldsymbol{y}) \in \mathbb{R}$ such that

$$\begin{aligned} \mathcal{B}(\boldsymbol{y};u(\boldsymbol{y}),v) &= \lambda(\boldsymbol{y})\mathcal{D}(u(\boldsymbol{y}),v), \quad \text{for all } v \in V\,, \\ \|u(\boldsymbol{y})\|_{\mathcal{D}} &= 1\,. \end{aligned} \tag{3.12}$$

In the following, let $\boldsymbol{y} \in U$ be fixed. The bilinear form $\mathcal{B}(\boldsymbol{y};\cdot,\cdot)$ is coercive

$$\mathcal{B}(\boldsymbol{y};v,v) \ge a_{\min}\,\|v\|_V^2\,, \quad \text{for all } v \in V\,, \tag{3.13}$$

and bounded

$$\mathcal{B}(\boldsymbol{y};w,v) \le a_{\max}\left(1 + C_D^2\right)\|w\|_V\,\|v\|_V\,, \tag{3.14}$$

with both constants independent of $\boldsymbol{y}$. To establish (3.14) we have used the upper bound (3.7) and the Poincaré inequality:

$$\|v\|_{L^2(D)} \le C_D\,\|v\|_V\,, \quad \text{for } v \in V\,. \tag{3.15}$$

Throughout the chapter we shall repeatedly refer to the eigenvalues of the negative Laplacian on D, with boundary condition (3.3). These are strictly positive and,

counting multiplicities, we denote them by

$$0 < \chi_1 \leq \chi_2 \leq \cdots . \tag{3.16}$$

The optimal value of the Poincaré constant in (3.15) is $C_D = \chi_1^{-1/2}$.

The bilinear form $\mathcal{B}(\boldsymbol{y};\cdot,\cdot)$ in (3.9) is coercive and self-adjoint on V. Moreover $\mathcal{D}$ is self-adjoint on $L^2(D)$, and V is compactly embedded in $L^2(D)$. Thus, (3.12) admits countably-many eigenvalues $(\lambda_k(\boldsymbol{y}))_{k\in\mathbb{N}}$, which are all positive, have finite multiplicity and accumulate only at infinity. Counting multiplicities, we write them as

$$0 < \lambda_1(\boldsymbol{y}) \leq \lambda_2(\boldsymbol{y}) \leq \lambda_3(\boldsymbol{y}) \leq \cdots . \tag{3.17}$$

with $\lambda_k(\boldsymbol{y}) \to \infty$ as $k \to \infty$. For an eigenvalue $\lambda(\boldsymbol{y})$ of (3.12) we define its eigenspace to be

$$E(\boldsymbol{y}, \lambda(\boldsymbol{y})) \coloneqq \{u : u \text{ is an eigenfunction corresponding to } \lambda(\boldsymbol{y})\} , \tag{3.18}$$

and from these eigenspaces, we can choose a sequence of eigenfunctions $(u_k(\boldsymbol{y}))_{k\in\mathbb{N}}$ corresponding to $(\lambda_k(\boldsymbol{y}))_{k\in\mathbb{N}}$ that form an orthonormal basis in V with respect to $\mathcal{D}(\cdot,\cdot)$.

The *min-max principle* (e.g. [6, (2.8)]) gives a representation of the kth eigenvalue as a minimum over all subspaces $S_k \subset V$ of dimension k:

$$\lambda_k(\boldsymbol{y}) = \min_{\substack{S_k \subset V \\ \dim(S_k)=k}} \max_{0\neq v\in S_k} \frac{\mathcal{B}(\boldsymbol{y}; v, v)}{\mathcal{D}(v, v)} . \tag{3.19}$$

Combining the min-max representation with the fact that due to (3.7) both bilinear forms in (3.19) are bounded and since $\mathcal{B}(\boldsymbol{y};\cdot,\cdot)$ is coercive we can bound the kth eigenvalue above and below independently of $\boldsymbol{y}$. Indeed, we have

$$\lambda_k(\boldsymbol{y}) \geq \frac{a_{\min}}{a_{\max}} \min_{\substack{S_k \subset V \\ \dim(S_k)=k}} \max_{0\neq v\in S_k} \frac{\langle \nabla v, \nabla v\rangle}{\langle v, v\rangle} \quad \text{and}$$

$$\lambda_k(\boldsymbol{y}) \leq \frac{a_{\max}}{a_{\min}} \min_{\substack{S_k \subset V \\ \dim(S_k)=k}} \max_{0\neq v\in S_k} \frac{\langle \nabla v, \nabla v\rangle + \langle v, v\rangle}{\langle v, v\rangle} ,$$

but now the right hand side of both bounds contains the bilinear form corresponding to the negative Laplacian on D. Hence, using the min-max representation of the

kth Laplacian eigenvalue χ_k the bounds on $\lambda_k(\boldsymbol{y})$ can be equivalently written as

$$\underline{\lambda_k} \coloneqq \frac{a_{\min}}{a_{\max}}\chi_k \leq \lambda_k(\boldsymbol{y}) \leq \frac{a_{\max}}{a_{\min}}(\chi_k + 1) =: \overline{\lambda_k}\,. \tag{3.20}$$

Taking $v = u_k(\boldsymbol{y})$ as a test function in (3.12), we obtain

$$\lambda_k(\boldsymbol{y}) = \mathcal{B}\left(\boldsymbol{y}; u_k(\boldsymbol{y}), u_k(\boldsymbol{y})\right)\,. \tag{3.21}$$

Then, using coercivity (3.13) and then the upper bound (3.20) on $\lambda_k(\boldsymbol{y})$, we obtain

$$\|u_k(\boldsymbol{y})\|_V \leq \sqrt{\frac{\lambda_k(\boldsymbol{y})}{a_{\min}}} \leq \frac{\sqrt{a_{\max}(\chi_k+1)}}{a_{\min}} =: \overline{u_k}\,. \tag{3.22}$$

Of particular interest is the smallest (also referred to as the minimal, dominant or fundamental) eigenvalue $\lambda_1(\boldsymbol{y})$. It follows by the Krein-Rutman Theorem that for every $\boldsymbol{y}$ the fundamental eigenvalue $\lambda_1(\boldsymbol{y})$ is simple, see e.g. [44, Theorems 1.2.5 and 1.2.6]. The fact that $\lambda_1(\boldsymbol{y})$ is simple for all $\boldsymbol{y}$ along with the uniform bound (3.20) ensures that the integral (3.5) is well-defined.

In order to obtain an optimal order convergence estimate for the piecewise linear finite element approximations of the eigenvalue problem (3.12), we require H^2-regularity of the eigenfunctions. To this end, we introduce the space $Z \coloneqq V \cap H^2(D)$, which we equip with the norm

$$\|v\|_Z \coloneqq \left(\|v\|^2_{L^2(D)} + \|\Delta v\|^2_{L^2(D)}\right)^{\frac{1}{2}}\,. \tag{3.23}$$

Similarly, to achieve optimal order FE convergence for linear functionals $\mathcal{G} \in H^{-1+t}(D)$, with $t \in [0,1]$, of eigenfunctions we introduce the subspace $Z^t \subset V$ given by

$$Z^t \coloneqq \left\{v \in V : \Delta v \in H^{-1+t}(D)\right\}\,, \tag{3.24}$$

with norm

$$\|v\|_{Z^t} \coloneqq \left(\|v\|^2_{L^2(D)} + \|\Delta v\|^2_{H^{-1+t}(D)}\right)^{\frac{1}{2}}\,. \tag{3.25}$$

For $r \in [-1,2]$ the fractional-order Sobolev norm is given by

$$\|v\|_{H^r(D)} \coloneqq \left(\sum_{k=1}^{\infty} \chi_k^r \langle v, \phi_k\rangle_{H^r(D)\times H^{-r}(D)}\right)^{\frac{1}{2}}\,,$$

where $\langle\cdot,\cdot\rangle_{H^r(D)\times H^{-r}(D)}$ is the duality pairing on $H^r(D) \times H^{-r}(D)$, obtained by continuously extending the $L^2(D)$ inner product. Also, recall that $(\chi_k)_{k=1}^{\infty}$ are the

eigenvalues of the negative Laplacian (3.16), and denote the corresponding eigenfunctions by $(\phi_k)_{k=1}^{\infty}$, which are chosen so as to form an orthonormal basis in $L^2(D)$. Note that since D is convex, $Z^1 = Z = V \cap H^2(D)$ and the norm is given by $\|\cdot\|_{Z^1} = \|\cdot\|_Z$ as in (3.23).

The proposition below shows that under Assumption A3.1, in particular A3.1.4, the eigenfunctions belong to $H^2(D)$ with norm bounded in terms of the corresponding eigenvalue.

Proposition 3.1. *Let $\boldsymbol{y} \in U$, Assumption A3.1 hold and suppose $(\lambda(\boldsymbol{y}), u(\boldsymbol{y}))$ is an eigenpair of (3.12). Then $u(\boldsymbol{y}) \in Z = V \cap H^2(D)$ and there exists a constant $C > 0$ such that*

$$\|u(\boldsymbol{y})\|_Z \leq C\lambda(\boldsymbol{y}), \quad \text{for all} \quad \boldsymbol{y} \in U. \tag{3.26}$$

Proof. Since $u(\boldsymbol{y}) \in L^2(D)$ we can apply [60, Theorem 4.1] with $t = 1$ and

$$f = (\lambda(\boldsymbol{y})c - b(\boldsymbol{y}))u(\boldsymbol{y}) \in L^2(D),$$

to give $u(\boldsymbol{y}) \in V \cap H^2(D)$ with

$$\begin{aligned}\|u(\boldsymbol{y})\|_Z &\leq C\,\|(\lambda(\boldsymbol{y})c - b(\boldsymbol{y}))u(\boldsymbol{y})\|_{L^2(D)} \\ &\leq C\left(\lambda(\boldsymbol{y})\,\|c\|_{L^\infty(D)} + \|b(\boldsymbol{y})\|_{L^\infty(D)}\right)\|u(\boldsymbol{y})\|_{L^2(D)}\end{aligned}$$

Then, using (3.7), (3.11) and the normalisation in (3.12), we obtain

$$\begin{aligned}\|u(\boldsymbol{y})\|_Z &\leq C\,\frac{a_{\max}}{\sqrt{a_{\min}}}\,(\lambda(\boldsymbol{y}) + 1)\,\|u(\boldsymbol{y})\|_{\mathcal{D}} = C\frac{a_{\max}}{\sqrt{a_{\min}}}\,(\lambda(\boldsymbol{y}) + 1) \\ &\leq C\frac{a_{\max}}{\sqrt{a_{\min}}}\left(1 + \frac{1}{\lambda_1(\boldsymbol{y})}\right)\lambda(\boldsymbol{y}),\end{aligned}$$

and the result follows by the lower bound in (3.20). □

3.2.2 Bounding the spectral gap

Although the Krein-Rutman Theorem gives us that the smallest eigenvalue $\lambda_1(\boldsymbol{y})$ is simple for all $\boldsymbol{y}$, it turns out that in order to estimate the derivatives of $\lambda_1(\boldsymbol{y})$ in Section 3.3 we require that the *spectral gap* $\lambda_2(\boldsymbol{y}) - \lambda_1(\boldsymbol{y})$ is uniformly positive over all $\boldsymbol{y} \in U$. Here, under the summability assumption in Assumption A3.1, we prove the required uniform positivity. We remark that this proof provides a theoretical justification for an assumption made previously without proof in [1]. The first step is the following elementary lemma, which shows that subsets of ℓ^∞ that are majorised by an ℓ^q sequence (for some $0 \leq q < \infty$) are compact.

Lemma 3.2. *Let $\boldsymbol{\alpha} \in \ell^q$ for some $1 < q < \infty$. The set $U(\boldsymbol{\alpha}) \subset \ell^\infty$ given by*

$$U(\boldsymbol{\alpha}) \coloneqq \left\{ \boldsymbol{w} \in \ell^\infty : |w_j| \leq \frac{1}{2}|\alpha_j| \right\}$$

is a compact subset of ℓ^∞.

Proof. Since ℓ^∞ is a normed (and hence a metric) space, $U(\boldsymbol{\alpha})$ is compact if and only if it is sequentially compact. To show sequential compactness of $U(\boldsymbol{\alpha})$, take any sequence $\{\boldsymbol{y}^{(k)}\}_{k\geq 1} \subset U(\boldsymbol{\alpha})$ and, for each $k \geq 1$, write $\boldsymbol{y}^{(k)} = (y_j^{(k)})_{j\geq 1}$. Clearly, by definition of $U(\boldsymbol{\alpha})$, each $\boldsymbol{y}^{(k)} \in \ell^q$ and moreover,

$$\|\boldsymbol{y}^{(k)}\|_{\ell^q} \leq \|\boldsymbol{\alpha}\|_{\ell^q} < \infty \quad \text{for all } k \in \mathbb{N}\,.$$

So $\boldsymbol{y}^{(k)}$ is a bounded sequence in ℓ^q. Since $q < \infty$, ℓ^q is a reflexive Banach space, and so by [9, Theorem 3.18] $\{\boldsymbol{y}^{(k)}\}_{k\geq 1}$ has a subsequence that converges weakly to a limit in ℓ^q. We denote this limit by $\boldsymbol{y}^*$, and, with a slight abuse of notation, we denote the convergent subsequence again by $\{\boldsymbol{y}^{(k)}\}_{k\geq 1}$.

We now prove that $\boldsymbol{y}^* \in U(\boldsymbol{\alpha})$ and that the weak convergence is in fact strong, that is, we show $\boldsymbol{y}^{(k)} \to \boldsymbol{y}^*$ in ℓ^∞, as $k \to \infty$. For any $j \in \mathbb{N}$, consider the linear functional $f_j : \ell^q \to \mathbb{R}$ given by $f_j(\boldsymbol{w}) = w_j$, where w_j denotes the jth element of the sequence $\boldsymbol{w} = (w_j)_{j\geq 1} \in \ell^q$. Clearly, $f_j \in (\ell^q)'$ (the dual space) and using the weak convergence established above, it follows that

$$y_j^{(k)} = f_j(\boldsymbol{y}^{(k)}) \to f_j(\boldsymbol{y}^*) = y_j^* \quad \text{as } k \to \infty\,, \quad \text{for each fixed } \; j.$$

That is, we have componentwise convergence. Furthermore, since $|y_j^{(k)}| \leq |\alpha_j|$ it follows that $|y_j^*| \leq |\alpha_j|$ for each j, and hence $\boldsymbol{y}^* \in U(\boldsymbol{\alpha})$.

Now, for any $N \in \mathbb{N}$ we can write

$$\begin{aligned} \left\|\boldsymbol{y}^{(k)} - \boldsymbol{y}^*\right\|_{\ell^q}^q &= \sum_{j=1}^{N} |y_j^{(k)} - y_j^*|^q + \sum_{j=N+1}^{\infty} |y_j^{(k)} - y_j^*|^q \\ &\leq N \max_{j=1,2,\ldots,N} |y_j^{(k)} - y_j^*|^q + \sum_{j=N+1}^{\infty} |\alpha_j|^q\,. \end{aligned} \tag{3.27}$$

Letting $\varepsilon > 0$, since $\boldsymbol{\alpha} \in \ell^q$ we can choose $N \in \mathbb{N}$ such that

$$\sum_{j=N+1}^{\infty} |\alpha_j|^q \leq \frac{\varepsilon^q}{2}\,,$$

and since $\boldsymbol{y}^{(k)}$ converges componentwise we can choose $K \in \mathbb{N}$ such that

$$|y_j^{(k)} - y_j^*| \leq (2N)^{-1/q}\varepsilon \quad \text{for all } j = 1, 2, \ldots, N \text{ and } k \geq K\,.$$

Thus, by (3.27) we have that $\left\|\boldsymbol{y}^{(k)} - \boldsymbol{y}^*\right\|_{\ell^q}^q \leq \varepsilon^q$ for all $k \geq K$, and hence that $\left\|\boldsymbol{y}^{(k)} - \boldsymbol{y}^*\right\|_{\ell^q} \to 0$ as $k \to \infty$.

Because $\|\boldsymbol{w}\|_{\ell^\infty} \leq \|\boldsymbol{w}\|_{\ell^q}$ when $\boldsymbol{w} \in \ell^q$, and $1 < q < \infty$, this also implies that $\boldsymbol{y}^{(k)} \to \boldsymbol{y}^*$ in ℓ^∞, completing the proof. □

A key property following from the perturbation theory of Kato [50] is that the eigenvalues $\lambda_k(\boldsymbol{y})$ are continuous in $\boldsymbol{y}$, which for completeness is shown below in Proposition 3.3. First, we require some notation. Let $\Sigma(T)$ denote the spectrum of an operator T. For each $\boldsymbol{y} \in U$ consider the solution operator $T(\boldsymbol{y}) : V^* \to V$ given, for $f \in V^*$, by

$$\mathcal{B}(\boldsymbol{y}; T(\boldsymbol{y})f, v) = \mathcal{D}(f, v) \quad \text{for all } v \in V\,. \tag{3.28}$$

Due to the symmetry of both $\mathcal{B}(\boldsymbol{y}; \cdot, \cdot)$ and $\mathcal{D}(\cdot, \cdot)$, each operator $T(\boldsymbol{y})$ is self-adjoint with respect to $\mathcal{D}$. Since $\mathcal{B}(\boldsymbol{y}; \cdot, \cdot)$ is coercive (3.13) and bounded (3.14), by the Lax-Milgram Theorem, see, e.g., [9], for every $f \in V^*$ there exists a unique solution $T(\boldsymbol{y})f \in V$ to (3.28), which satisfies $\|T(\boldsymbol{y})f\|_V \leq \|f\|_{V^*}/a_{\min}$. Hence, each $T(\boldsymbol{y}) : V^* \to V$ is bounded and invertible.

We can also consider the operators $T(\boldsymbol{y}) : L^2(D) \to L^2(D)$, in which case due to the the compact embedding of V into $L^2(D)$, each $T(\boldsymbol{y}) : L^2(D) \to L^2(D)$ is compact. In this case, for $f \in L^2(D)$ the Lax-Milgram Theorem again gives a unique solution $T(\boldsymbol{y})f \in V$ with the bound

$$\|T(\boldsymbol{y})f\|_V \leq \frac{\sqrt{a_{\max}}}{a_{\min}} \|f\|_{L^2(D)}\,, \tag{3.29}$$

where we have also used the equivalence of norms (3.11).

From the spectral theory for compact, selfadjoint operators we know that each $T(\boldsymbol{y})$ admits countably-many eigenvalues, which are all finite, real, strictly positive and have finite multiplicity. Counting multiplicities, the eigenvalues of $T(\boldsymbol{y})$ are denoted (in decreasing order) by

$$\mu_1(\boldsymbol{y}) \geq \mu_2(\boldsymbol{y}) \geq \cdots > 0\,.$$

Comparing (3.12) with (3.28) we have that $\lambda(\boldsymbol{y})$ is an eigenvalue of (3.12) if and only if $\mu(\boldsymbol{y}) = 1/\lambda(\boldsymbol{y})$ is an eigenvalue of $T(\boldsymbol{y})$, and their eigenspaces coincide. Moreover,

because $\lambda_k(\boldsymbol{y})$ are enumerated in increasing order whereas $\mu_k(\boldsymbol{y})$ are decreasing we also have that $\mu_k(\boldsymbol{y}) = 1/\lambda_k(\boldsymbol{y})$.

Proposition 3.3. *Let Assumption A3.1 hold. Then the eigenvalues $\lambda_1, \lambda_2, \ldots$ are Lipschitz continuous in $\boldsymbol{y}$.*

Proof. We prove the result by establishing the continuity of the eigenvalues $\mu_k(\boldsymbol{y})$ of $T(\boldsymbol{y})$. Let $\boldsymbol{y}$, $\boldsymbol{y}' \in U$ and consider the operators $T(\boldsymbol{y}), T(\boldsymbol{y}') : L^2(D) \to L^2(D)$ as defined in (3.28). Since $T(\boldsymbol{y})$, $T(\boldsymbol{y}')$ are bounded and self-adjoint with respect to $\mathcal{D}$, it follows from [50, V, §4.3 and Theorem 4.10] that we have the following notion of continuity of $\mu(\cdot)$ in terms of $T(\cdot)$

$$\sup_{\mu \in \Sigma(T(\boldsymbol{y}))} \operatorname{dist}(\mu, \Sigma(T(\boldsymbol{y}'))) \leq \|T(\boldsymbol{y}) - T(\boldsymbol{y}')\|_{L^2(D) \to L^2(D)} \tag{3.30}$$

where for an operator $T : L^2(D) \to L^2(D)$ we define the norm

$$\|T\|_{L^2(D) \to L^2(D)} \coloneqq \sup_{f \in L^2(D),\, \|f\|_{L^2(D)} \leq 1} \|Tf\|_{L^2(D)} \,.$$

For an eigenvalue $\mu_k(\boldsymbol{y}) \in \Sigma(T(\boldsymbol{y}))$, (3.30) implies that there exists a $\mu_{k'}(\boldsymbol{y}') \in \Sigma(T(\boldsymbol{y}'))$ such that

$$|\mu_k(\boldsymbol{y}) - \mu_{k'}(\boldsymbol{y}')| \leq \|T(\boldsymbol{y}) - T(\boldsymbol{y}')\|_{L^2(D) \to L^2(D)} \,. \tag{3.31}$$

Note that this states that at $\boldsymbol{y}'$ there exists an eigenvalue of $T(\boldsymbol{y}')$ close to $\mu_k(\boldsymbol{y})$, but does not imply that at $\boldsymbol{y}'$ the kth eigenvalue of $T(\boldsymbol{y}')$ is close to $\mu_k(\boldsymbol{y})$, that is, in (3.31) k is not necessarily equal to k'. However, consider any $\mu_k(\boldsymbol{y})$ and let m denote its multiplicity. Since $m < \infty$ the collection $\mu_k(\boldsymbol{y}) = \mu_{k+1}(\boldsymbol{y}) = \cdots = \mu_{k+m-1}(\boldsymbol{y})$ is a *finite system* of eigenvalues in the sense of Kato. It then follows from the discussion in [50, IV, §3.5] that the eigenvalues in this system depend continuously on the operator with multiplicity preserved. This preservation of multiplicity is key, and in other words it states that for $T(\boldsymbol{y}')$ sufficiently close to $T(\boldsymbol{y})$ there are m eigenvalues $\mu_k(\boldsymbol{y}'), \mu_k(\boldsymbol{y}'), \ldots, \mu_{k+m-1}(\boldsymbol{y}') \in \Sigma(T(\boldsymbol{y}'))$, no longer neccesarily equal, that are close to $\mu_k(\boldsymbol{y})$.

This can be done for each eigenvalue $\mu_k(\boldsymbol{y}) \in \Sigma(T(\boldsymbol{y}))$, and a simple induction argument then shows that each μ_k is continuous in the following sense

$$|\mu_k(\boldsymbol{y}) - \mu_k(\boldsymbol{y}')| \leq \|T(\boldsymbol{y}) - T(\boldsymbol{y}')\|_{L^2(D) \to L^2(D)} \,.$$

It then follows from the relationship $\mu_k(\boldsymbol{y}) = 1/\lambda_k(\boldsymbol{y})$ along with the upper bound in (3.20) that we have a similar result for eigenvalues λ_k of (3.12):

$$|\lambda_k(\boldsymbol{y}) - \lambda_k(\boldsymbol{y}')| \leq \left(\frac{a_{\max}(\chi_k + 1)}{a_{\min}}\right)^2 \|T(\boldsymbol{y}) - T(\boldsymbol{y}')\|_{L^2(D)\to L^2(D)} . \tag{3.32}$$

All that remains is to bound the right hand of (3.32) by $C_{\text{Lip}} \|\boldsymbol{y} - \boldsymbol{y}'\|_{\ell^\infty}$, with $C_{\text{Lip}} > 0$ independent of $\boldsymbol{y}$ and $\boldsymbol{y}'$.

To this end, note that since the right hand side of (3.28) is independent of $\boldsymbol{y}$ we have

$$\mathcal{B}(\boldsymbol{y}; T(\boldsymbol{y})f, v) = \mathcal{B}(\boldsymbol{y}'; T(\boldsymbol{y}')f, v) \quad \text{for all } f \in L^2(D), v \in V .$$

Rearranging and then expanding this gives

$$\begin{aligned}\mathcal{B}\big(\boldsymbol{y}; [T(\boldsymbol{y}) - T(\boldsymbol{y}')]f, v\big) = \int_D \Big(&[a(\boldsymbol{x}, \boldsymbol{y}') - a(\boldsymbol{x}, \boldsymbol{y})] \nabla[T(\boldsymbol{y})f](\boldsymbol{x}) \cdot \nabla v(\boldsymbol{x}) \\ &+ [b(\boldsymbol{x}, \boldsymbol{y}') - b(\boldsymbol{x}, \boldsymbol{y})] \, [T(\boldsymbol{y})f](\boldsymbol{x}) \, v(\boldsymbol{x})\Big) \, \mathrm{d}\boldsymbol{x} .\end{aligned}$$

Letting $v = (T(\boldsymbol{y}) - T(\boldsymbol{y}'))f$, the left hand side can be bounded from below using the coercivity (3.13) of $\mathcal{B}(\boldsymbol{y}; \cdot, \cdot)$, and the right hand side can be bounded from above using the Cauchy-Schwartz inequality to give

$$\begin{aligned}a_{\min} \|(T(\boldsymbol{y}) - T(\boldsymbol{y}'))f\|_V^2 \leq \max\Big(&\|a(\boldsymbol{y}) - a(\boldsymbol{y}')\|_{L^\infty(D)}, \|b(\boldsymbol{y}) - b(\boldsymbol{y}')\|_{L^\infty(D)}\Big) \\ \cdot \Big(&\|T(\boldsymbol{y})f\|_V \|(T(\boldsymbol{y}) - T(\boldsymbol{y}'))f\|_V + \|T(\boldsymbol{y})f\|_{L^2(D)} \|(T(\boldsymbol{y}) - T(\boldsymbol{y}'))f\|_{L^2(D)}\Big) .\end{aligned}$$

Applying the Poincaré inequality (3.15) to both L^2-norm factors, dividing through by $a_{\min} \|(T(\boldsymbol{y}) - T(\boldsymbol{y}'))f\|_V$ and using the upper bound in (3.29) we have

$$\begin{aligned}&\|(T(\boldsymbol{y}) - T(\boldsymbol{y}'))f\|_V \\ &\leq \frac{\sqrt{a_{\max}}(1 + 1/\chi_1)}{a_{\min}^2} \|f\|_{L^2(D)} \max\Big(\|a(\boldsymbol{y}) - a(\boldsymbol{y}')\|_{L^\infty(D)}, \|b(\boldsymbol{y}) - b(\boldsymbol{y}')\|_{L^\infty(D)}\Big) .\end{aligned}$$

Applying the Poincaré inequality (3.15) to the left hand side now and taking the supremum over $f \in L^2(D)$ with $\|f\|_{L^2(D)} \leq 1$, in the operator norm we have

$$\begin{aligned}&\|T(\boldsymbol{y}) - T(\boldsymbol{y}')\|_{L^2(D)\to L^2(D)} \\ &\quad \leq \frac{\sqrt{a_{\max}\chi_1}(1 + 1/\chi_1)}{a_{\min}^2} \max\Big(\|a(\boldsymbol{y}) - a(\boldsymbol{y}')\|_{L^\infty(D)}, \|b(\boldsymbol{y}) - b(\boldsymbol{y}')\|_{L^\infty(D)}\Big) .\end{aligned}$$

Using this inequality as an upper bound for (3.32) we have that the eigenvalues inherit the continuity of the coefficients

$$|\lambda_k(\boldsymbol{y}) - \lambda_k(\boldsymbol{y}')| \;\leq\; C_k \max\left(\|a(\boldsymbol{y}) - a(\boldsymbol{y}')\|_{L^\infty(D)}\,,\; \|b(\boldsymbol{y}) - b(\boldsymbol{y}')\|_{L^\infty(D)}\right)\,, \quad (3.33)$$

where

$$C_k \;=\; \frac{a_{\max}^{5/2}\sqrt{\chi_1}(\chi_k+1)^2(1+1/\chi_1)}{a_{\min}^4} \;<\; \infty\,,$$

depends on k but is clearly independent of $\boldsymbol{y}$ and $\boldsymbol{y}'$.

Finally, to establish continuity in terms of $\boldsymbol{y}$, we expand the coefficients in (3.33) above and use the triangle inequality to give

$$\begin{aligned}|\lambda_k(\boldsymbol{y}) - \lambda_k(\boldsymbol{y}')| \;&\leq\; C_k \sum_{j=1}^{\infty} |y_j - y_j'| \max\left(\|a_j\|_{L^\infty(D)}\,,\; \|b_j\|_{L^\infty(D)}\right)\\ &\leq\; C_k\left(\sum_{j=1}^{\infty} \max\left(\|a_j)\|_{L^\infty(D)}\,,\; \|b_j\|_{L^\infty(D)}\right)\right)\|\boldsymbol{y} - \boldsymbol{y}'\|_{\ell^\infty}\,.\end{aligned}$$

By Assumption 3.1 the sum is finite, and hence the eigenvalue $\lambda_k(\boldsymbol{y})$ is Lipschitz in $\boldsymbol{y}$. □

Now that we have shown continuity of the eigenvalues and identified suitable compact subsets, we can show that the spectral gap is bounded uniformly away from 0.

Proposition 3.4. *Let Assumption A3.1 hold. Then there exists a $\delta > 0$, independent of $\boldsymbol{y}$, such that*

$$\lambda_2(\boldsymbol{y}) - \lambda_1(\boldsymbol{y}) \;\geq\; \delta\,. \quad (3.34)$$

Proof. The idea of the proof is to transfer some of the summability of the terms a_j, b_j in the coefficients onto the parameters by defining new parameters $\widetilde{y}_j = \alpha_j y_j$, and then rewriting $a(\boldsymbol{x}, \boldsymbol{y})$ as

$$a(\boldsymbol{x}, \boldsymbol{y}) \;=\; a_0(\boldsymbol{x}) \;+\; \sum_{j=1}^{\infty} \widetilde{y}_j \frac{a_j(\boldsymbol{x})}{\alpha_j}\,.$$

The sequence $\boldsymbol{\alpha} \in \ell^q$ is chosen to decay slowly enough so that $\sum_{j=1}^{\infty} \|a_j\|_{L^\infty(D)}/\alpha_j$ is still finite. We apply a similar reparametrisation procedure to $b(\boldsymbol{x}, \boldsymbol{y})$. Then using an intermediate result from the proof of Proposition 3.3 we can show that the eigenvalues of the "reparametrised" problem are continuous in the new parameter $\widetilde{\boldsymbol{y}}$,

which now ranges over the compact set $U(\boldsymbol{\alpha})$. The required bound on the spectral gap is obtained by using the equivalence of the eigenvalues of the original and reparametrised problems.

To give some details, note that there is no loss of generality in assuming $p > 1/2$. We consequently set $\varepsilon = 1 - p \in (0, 1/2)$ and consider the sequence $\boldsymbol{\alpha}$ defined by

$$\alpha_j \;=\; \|a_j\|^{\varepsilon}_{L^\infty(D)} + \|b_j\|^{\varepsilon}_{L^\infty(D)} + 1/j, \quad \text{for each} \;\; j = 1, 2, \ldots. \tag{3.35}$$

Setting $q = p/\varepsilon = p/(1-p) \in (1, \infty)$, using Assumption A3.1.3 and the triangle inequality, it is easy to see that $\boldsymbol{\alpha} \in \ell^q$. Moreoever, the inclusion of $1/j$ in (3.35) ensures that $\alpha_j \neq 0$, for all $j \geq 1$. Hence, from now on, for $\boldsymbol{w} = (w_j)_{j=1}^\infty \in \ell^\infty$, we can define the sequences $\boldsymbol{\alpha w} = (\alpha_j w_j)_{j=1}^\infty$ and $\boldsymbol{w}/\boldsymbol{\alpha} = (w_j/\alpha_j)_{j=1}^\infty$. Then, recalling the definition of $U(\boldsymbol{\alpha})$ in Lemma 3.2, it is easy to see that

$$\widetilde{\boldsymbol{y}} \in U(\boldsymbol{\alpha}) \text{ if and only if } \widetilde{\boldsymbol{y}}/\boldsymbol{\alpha} \in U \quad \text{and moreover} \quad \boldsymbol{y} \in U \text{ if and only if } \boldsymbol{\alpha y} \in U(\boldsymbol{\alpha})\,.$$

Now for $\boldsymbol{x} \in D$ and $\widetilde{\boldsymbol{y}} \in U(\boldsymbol{\alpha})$, we define

$$\widetilde{a}(\boldsymbol{x}, \widetilde{\boldsymbol{y}}) = a_0(\boldsymbol{x}) + \sum_{j=1}^{\infty} \widetilde{y}_j \frac{a_j(\boldsymbol{x})}{\alpha_j} \;\; \text{and} \;\; \widetilde{b}(\boldsymbol{x}, \widetilde{\boldsymbol{y}}) = b_0(\boldsymbol{x}) + \sum_{j=1}^{\infty} \widetilde{y}_j \frac{b_j(\boldsymbol{x})}{\alpha_j}\,,$$

from which it is easily seen that

$$\widetilde{a}(\boldsymbol{x}, \widetilde{\boldsymbol{y}}) = a(\boldsymbol{x}, \widetilde{\boldsymbol{y}}/\boldsymbol{\alpha}) \quad \text{and} \quad \widetilde{b}(\boldsymbol{x}, \widetilde{\boldsymbol{y}}) = b(\boldsymbol{x}, \widetilde{\boldsymbol{y}}/\boldsymbol{\alpha})\,. \tag{3.36}$$

Then we set

$$\widetilde{\mathcal{B}}(\widetilde{\boldsymbol{y}}; w, v) \;\coloneqq\; \int_D \Big(\widetilde{a}(\boldsymbol{x}, \widetilde{\boldsymbol{y}}) \nabla w(\boldsymbol{x}).\nabla v(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} + \widetilde{b}(\boldsymbol{x}, \widetilde{\boldsymbol{y}}) w(\boldsymbol{x}) v(\boldsymbol{x})\Big)\,\mathrm{d}\boldsymbol{x} \quad \text{for } w, v \in V\,,$$

and we consider the reparametrised eigenvalue problem find $\widetilde{\lambda}(\widetilde{\boldsymbol{y}}) \in \mathbb{R}$ and $0 \neq \widetilde{u}(\widetilde{\boldsymbol{y}}) \in V$ such that

$$\begin{aligned} \widetilde{\mathcal{B}}(\widetilde{\boldsymbol{y}}; \widetilde{u}(\widetilde{\boldsymbol{y}}), v) \;&=\; \widetilde{\lambda}_k(\widetilde{\boldsymbol{y}}) \mathcal{D}(\widetilde{u}(\widetilde{\boldsymbol{y}}), v) \quad \text{for all } v \in V\,, \\ \|\widetilde{u}(\widetilde{\boldsymbol{y}})\|_{\mathcal{D}} \;&=\; 1\,. \end{aligned} \tag{3.37}$$

Note that because we have equality between the original and reparametrised coefficients (3.36), for each $\boldsymbol{y} \in U$, and corresponding $\widetilde{\boldsymbol{y}} = \boldsymbol{\alpha y} \in U(\boldsymbol{\alpha})$, (3.12) and (3.37) are simply different ways of writing the same eigenvalue problem. In

particular, (3.36) implies that there is equality between eigenvalues $\lambda_k(\boldsymbol{y})$ of (3.12) and $\widetilde{\lambda}_k(\widetilde{\boldsymbol{y}})$ of the reparametrised eigenvalue problem (3.37)

$$\lambda_k(\boldsymbol{y}) \;=\; \widetilde{\lambda}_k(\widetilde{\boldsymbol{y}}) \quad \text{for } k \in \mathbb{N}\,, \tag{3.38}$$

and their eigenspaces coincide.

From the proof of Proposition 3.3 we know that the eigenvalues $\lambda_k(\boldsymbol{y})$ of (3.12) share the same continuity as the coefficients, see (3.33), which we now use to establish continuity of the reparametrised eigenvalues in $\widetilde{\boldsymbol{y}}$. For an eigenvalue $\widetilde{\lambda}_k(\widetilde{\boldsymbol{y}})$ of (3.37), using (3.38) in the inequality (3.33) we have

$$|\widetilde{\lambda}_k(\widetilde{\boldsymbol{y}}) - \widetilde{\lambda}_k(\widetilde{\boldsymbol{y}}')| \;\le\; C \max\left(\|a(\widetilde{\boldsymbol{y}}/\boldsymbol{\alpha}) - a(\widetilde{\boldsymbol{y}}/\boldsymbol{\alpha})\|_{L^\infty(D)}\,,\, \left\|b(\widetilde{\boldsymbol{y}}/\boldsymbol{\alpha}) - b(\widetilde{\boldsymbol{y}}'/\boldsymbol{\alpha})\right\|_{L^\infty(D)} \right),$$

which after expanding the coefficients and using the triangle inequality becomes

$$|\widetilde{\lambda}_k(\widetilde{\boldsymbol{y}}) - \widetilde{\lambda}_k(\widetilde{\boldsymbol{y}}')| \;\le\; \underbrace{C\left(\sum_{j=1}^{\infty} \frac{1}{\alpha_j} \max\left(\|a_j\|_{L^\infty(D)}\,,\; \|b_j\|_{L^\infty(D)} \right) \right)}_{\widetilde{C}_{\mathrm{Lip}}} \left\|\widetilde{\boldsymbol{y}} - \widetilde{\boldsymbol{y}}'\right\|_{\ell^\infty}\,,$$

where $\widetilde{C}_{\mathrm{Lip}}$ is clearly independent of $\widetilde{\boldsymbol{y}}$ and $\widetilde{\boldsymbol{y}}'$. Now by (3.35) together with Assumption A3.1, we have

$$\sum_{j=1}^{\infty} \frac{\|a_j\|_{L^\infty(D)}}{\alpha_j} \;\le\; \sum_{j=1}^{\infty} \|a_j\|_{L^\infty(D)}^{1-\varepsilon} \;=\; \sum_{j=1}^{\infty} \|a_j\|_{L^\infty(D)}^{p} <\; \infty\,,$$

with the analogous estimate for $\sum_{j=1}^{\infty} \|b_j\|_{L^\infty(D)}/\alpha_j$. Thus, $\widetilde{C}_{\mathrm{Lip}} < \infty$ and hence the reparametrised eigenvalues are continuous on $U(\boldsymbol{\alpha})$.

It immediately follows that the spectral gap $\widetilde{\lambda}_2(\widetilde{\boldsymbol{y}}) - \widetilde{\lambda}_1(\widetilde{\boldsymbol{y}})$ is also continuous on $U(\boldsymbol{\alpha})$, which by Lemma 3.2 is a compact subset of ℓ^∞. Therefore, the non-zero minimum is attained giving that the spectral gap $\widetilde{\lambda}_2(\widetilde{\boldsymbol{y}}) - \widetilde{\lambda}_1(\widetilde{\boldsymbol{y}})$ is uniformly positive. Finally, because there is equality between the original and reparametrised eigenvalues (3.38) the result holds for the original problem over all $\boldsymbol{y} \in U$. □

Remark 3.5. An explicit bound on the spectral gap can be obtained by assuming tighter restrictions on the coefficients. For example, if $a \equiv 1 \equiv c$ and b is weakly convex then [2] gives an explicit lower bound on the fundamental gap. Alternatively, using the upper and lower bounds on the eigenvalues (3.20), we can determine

restrictions on $a_{\min}$ and $a_{\max}$ such that the gap is bounded away from 0. Explicitly, if

$$\frac{a_{\min}}{a_{\max}} > \sqrt{\frac{\chi_1 + 1}{\chi_2}},$$

then $\lambda_2(\boldsymbol{y}) - \lambda_1(\boldsymbol{y}) \geq \underline{\lambda_2} - \overline{\lambda_1} > 0$.

3.2.3 Finite element discretisation

To approximate eigenpairs $(\lambda(\boldsymbol{y}), u(\boldsymbol{y}))$ we introduce a collection of finite element subspaces $V_h \subset V$ with dimension M_h, which are associated with a conforming triangulation $\mathscr{T}_h$ of the domain D and a basis $(\phi_{h,i})_{i=1}^{M_h}$. The parameter $h = \max\{\mathrm{diam}(\tau) : \tau \in \mathscr{T}_h\}$ is called the *mesh width*. The method works for very general spaces V_h, however to fully exploit higher rates of convergence stronger assumptions on the regularity of the coefficients and the domain would be required. As such, in the current chapter we restrict our attention to piecewise linear finite elements, that is, each V_h is the space of continuous functions that are linear on the elements of $\mathscr{T}_h$ and vanish on the boundary ∂D. It is well-known that the best approximation error for the space Z^t (as defined in (3.24) and (3.25)) by functions in V_h satisfies

$$\inf_{w_h \in V_h} \|v - w_h\|_V \leq C h^t \|v\|_{Z^t} \quad \text{for all } v \in Z^t. \tag{3.39}$$

For each $\boldsymbol{y} \in U$ the discrete eigenvalue problem is to find $\lambda_h(\boldsymbol{y}) \in \mathbb{R}$ and $u_h(\boldsymbol{y}) \in V_h$ satisfying

$$\begin{aligned} \mathcal{B}(\boldsymbol{y}; u_h(\boldsymbol{y}), v) &= \lambda_h(\boldsymbol{y}) \mathcal{D}(u_h(\boldsymbol{y}), v) \quad \text{for all } v \in V_h, \\ \|u_h(\boldsymbol{y})\|_{\mathcal{D}} &= 1. \end{aligned} \tag{3.40}$$

For each $\boldsymbol{y} \in U$, the discrete eigenvalue problem (3.40) admits M_h eigenvalues

$$0 < \lambda_{1,h}(\boldsymbol{y}) \leq \lambda_{2,h}(\boldsymbol{y}) \leq \cdots \leq \lambda_{M_h,h}(\boldsymbol{y}),$$

with corresponding eigenfunctions

$$u_{1,h}(\boldsymbol{y}),\ u_{2,h}(\boldsymbol{y}),\ \ldots,\ u_{M_h,h}(\boldsymbol{y}) \in V_h.$$

For each fixed k, the kth finite element eigenvalue $\lambda_{k,h}(\boldsymbol{y})$ converges from above to the kth eigenvalue of (3.12), that is, for each k,

$$\lambda_{k,h}(\boldsymbol{y}) \to \lambda_k(\boldsymbol{y}) \quad \text{with} \quad \lambda_{k,h}(\boldsymbol{y}) \geq \lambda_k(\boldsymbol{y})$$

and the corresponding FE eigenfunctions $(u_{k,h}(\boldsymbol{y}))_{k=1}^{M_h}$ satisfy

$$\mathrm{dist}_V(u_{k,h}(\boldsymbol{y}), E(\boldsymbol{y}, \lambda_k(\boldsymbol{y}))) \to 0$$

Where $\mathrm{dist}_V(v, \mathcal{P})$ is the distance of $v \in V$ from the subspace $\mathcal{P} \subset V$

$$\mathrm{dist}_V(v, \mathcal{P}) := \inf_{w \in \mathcal{P}} \|v - w\|_V .$$

The classical results on FE error estimates for eigenproblems are found in [5, 6, 7]; however, we cannot simply use these results verbatim since their constants depend in complex and often hidden ways on the eigenvalues and eigenvalue gaps. For us, this means that they depend on $\boldsymbol{y}$, and so care must be taken to ensure that all constants that occur can be bounded independently of $\boldsymbol{y}$. Theorem 3.6 below quantifies the FE convergence and in the proof we track the dependence of all constants on $\boldsymbol{y}$. The proof is rather long and technical, and as such is left for the appendix.

Theorem 3.6. *Let $\boldsymbol{y} \in U$ and suppose that Assumption A3.1 holds. Then for $h > 0$ sufficiently small*

$$|\lambda_1(\boldsymbol{y}) - \lambda_{1,h}(\boldsymbol{y})| \leq C_1 h^2 , \tag{3.41}$$

and $u_{1,h}(\boldsymbol{y}) \in E(\boldsymbol{y}, \lambda_{1,h}(\boldsymbol{y}))$ can be chosen such that

$$\|u_1(\boldsymbol{y}) - u_{1,h}(\boldsymbol{y})\|_V \leq C_2 h . \tag{3.42}$$

Moreover, for $\mathcal{G} \in H^{-1+t}$ with $t \in [0, 1]$

$$|\mathcal{G}(u_1(\boldsymbol{y})) - \mathcal{G}(u_{1,h}(\boldsymbol{y}))| \leq C_3 h^{1+t} , \tag{3.43}$$

and $C_1, C_2, C_3 > 0$ are independent of $\boldsymbol{y}$.

3.2.4 Quasi-Monte Carlo methods

In this section we briefly summarise the relevant aspects of QMC rules from Section 2.3, the main purpose being to fix notation. In this chapter we use randomly shifted lattice rules, where recall from Section 2.3.1 the points are constructed using a generating vector $\boldsymbol{z} \in \mathbb{N}^s$ and a uniformly distributed random shift $\boldsymbol{\Delta} \in [0, 1)^s$. Specifically, letting $\mathcal{Y} = [-\frac{1}{2}, \frac{1}{2}]$ and $\rho \equiv 1$ in the definition of the integral $\mathcal{I}_s f$ (2.1),

a shifted lattice rule approximates $\mathcal{I}_s f$ by an equal-weight quadrature rule

$$Q^{\mathrm{sh}}_{n,s}(\boldsymbol{z},\boldsymbol{\Delta})f \;=\; \frac{1}{n}\sum_{k=0}^{n-1} f\left(\left\{\frac{k\boldsymbol{z}}{n}+\boldsymbol{\Delta}\right\}-\tfrac{\mathbf{1}}{\mathbf{2}}\right), \tag{3.44}$$

where we have subtracted the vector $\frac{\mathbf{1}}{\mathbf{2}} \coloneqq (\frac{1}{2},\dots,\frac{1}{2})$ to map the points from $[0,1]^s$ to $[-\frac{1}{2},\frac{1}{2}]^s$. Also, for the remainder of this chapter, to ease notation we will not include the dependence on the generating vector $\boldsymbol{z}$ or the random shift $\boldsymbol{\Delta}$.

Following Section 2.3 we will use the CBC algorithm to construct the generating vector and perform the error analysis in the weighted Sobolev space $\mathcal{W}_{s,\boldsymbol{\gamma}}$ of functions with square-integrable mixed first derivatives. The space $\mathcal{W}_{s,\boldsymbol{\gamma}}$ is equipped with the unanchored norm (2.5) and has weights, which will be specified later, denoted by $\boldsymbol{\gamma} = \{\gamma_{\mathrm{u}}\}_{\mathrm{u}\subset\mathbb{N}}$. In this case, for $f \in \mathcal{W}_{s,\boldsymbol{\gamma}}$ the RMS error of a CBC-generated lattice rule approximation is bounded above by

$$\sqrt{\mathbb{E}_{\boldsymbol{\Delta}}\left(\left|\mathcal{I}_s f - Q^{\mathrm{sh}}_{n,s} f\right|^2\right)} \;\le\; \left(\frac{1}{\varphi(n)}\sum_{\emptyset\neq\mathrm{u}\subseteq\{1:s\}}\gamma_{\mathrm{u}}^{\eta}\left(\frac{2\zeta(2\eta)}{(2\pi^2)^{\eta}}\right)^{|\mathrm{u}|}\right)^{\frac{1}{2\eta}} \|f\|_{s,\boldsymbol{\gamma}} \tag{3.45}$$

for all $\eta \in (\frac{1}{2},1]$. This error bound was obtained by combining the upper bound (2.12) with the upper bound on the shift-averaged worst-case error (2.15).

Finally, to estimate the quadrature error for the numerical results in Section 3.5, we will perform R approximations generated by R independent random shifts.

3.3 Parametric regularity

In this section we examine the regularity of the minimal eigenpair $(\lambda_1(\boldsymbol{y}), u_1(\boldsymbol{y}))$ of the variational eigenproblem (3.12) with respect to the stochastic parameter $\boldsymbol{y}$. The results we obtain show that $\lambda_1(\boldsymbol{y})$ belongs to the weighted space $\mathcal{W}_{s,\boldsymbol{\gamma}}$ with norm defined in (2.5). This is required for the analysis of the QMC error in computing $\lambda_1(\boldsymbol{y})$. Also, to obtain an *a priori* bound on the QMC error we require a bound on the norm of $\lambda_1(\boldsymbol{y})$ in $\mathcal{W}_{s,\boldsymbol{\gamma}}$, hence we must bound its mixed first derivatives, see Lemma 3.10. There we use the bounds on the spectral gap obtained in Section 3.2.2, and present results not only for the first-order mixed derivatives but also for higher-order derivatives.

We begin with the following coercive-type estimate, which is required in order to bound the norm of the derivatives of the eigenfunction.

Lemma 3.7. *Let Assumption A3.1 hold, and for each $\boldsymbol{y} \in U$ and $\lambda \in \mathbb{R}$ define $\mathcal{B}^{\mathrm{sh}}_{\lambda}(\boldsymbol{y};\cdot,\cdot) : V \times V \to \mathbb{R}$ to be the shifted bilinear form given by*

$$\mathcal{B}^{\mathrm{sh}}_{\lambda}(\boldsymbol{y};u,v) \coloneqq \mathcal{B}(\boldsymbol{y};u,v) - \lambda \mathcal{D}(u,v)\,, \tag{3.46}$$

with $\mathcal{B}$ and $\mathcal{D}$ defined by (3.9) and (3.10), respectively. The $\lambda_1(\boldsymbol{y})$-shifted bilinear form restricted to the $\mathcal{D}$-orthogonal complement of the eigenspace corresponding to $\lambda_1(\boldsymbol{y})$, denoted by $E(\boldsymbol{y},\lambda_1(\boldsymbol{y}))^{\perp}$, is coercive. More precisely,

$$\mathcal{B}^{\mathrm{sh}}_{\lambda_1(\boldsymbol{y})}(\boldsymbol{y};v,v) \;\geq\; a_{\min}\left(1-\frac{\lambda_1(\boldsymbol{y})}{\lambda_2(\boldsymbol{y})}\right)\|v\|_V^2 \quad \text{for all } v \in E(\boldsymbol{y},\lambda_1(\boldsymbol{y}))^{\perp}\,. \tag{3.47}$$

Proof. Since the eigenfunctions $(u_k(\boldsymbol{y}))_{k\in\mathbb{N}}$ form a basis in V that is orthonormal with respect to the inner product $\mathcal{D}$, for $v \in E(\boldsymbol{y},\lambda_1(\boldsymbol{y}))^{\perp}$, letting $v_k(\boldsymbol{y}) \coloneqq \mathcal{D}(v,u_k(\boldsymbol{y}))u_k(\cdot,\boldsymbol{y})$ for $k = 1,2,\ldots$, we can write

$$v \;=\; \sum_{k=2}^{\infty} v_k(\boldsymbol{y})\,,$$

where we have used $v_1 = 0$ since $\mathcal{D}(v,u_1(\boldsymbol{y})) = 0$. Henceforth, we will suppress the dependence of the eigenvalues and v_k on $\boldsymbol{y}$. For $v \in E(\boldsymbol{y},\lambda_1)^{\perp}$ we have

$$\mathcal{B}^{\mathrm{sh}}_{\lambda_1}(\boldsymbol{y};v,v) \;=\; \mathcal{B}^{\mathrm{sh}}_{\lambda_1}\left(\boldsymbol{y};\sum_{k=2}^{\infty}v_k,\sum_{\ell=2}^{\infty}v_\ell\right) \;=\; \sum_{k,\ell=2}^{\infty}\left(\mathcal{B}(\boldsymbol{y};v_k,v_\ell)-\lambda_1\mathcal{D}(v_k,v_\ell)\right)\,.$$

Since all v_k are just scaled versions of u_k, they also satisfy the variational equation (3.8) so that $\mathcal{B}(\boldsymbol{y};v_k,v_\ell) = \lambda_k\mathcal{D}(v_k,v_\ell)$, and they are orthogonal with respect to $\mathcal{D}(\cdot,\cdot)$ implying that $\mathcal{B}(\boldsymbol{y};v_k,v_\ell) = 0$ for $k \neq \ell$. Thus we can reduce the above double sum to

$$\begin{aligned}
\mathcal{B}^{\mathrm{sh}}_{\lambda_1}(\boldsymbol{y};v,v) \;&=\; \sum_{k=2}^{\infty}\left(\mathcal{B}(\boldsymbol{y};v_k,v_k)-\frac{\lambda_1}{\lambda_k}\mathcal{B}(\boldsymbol{y};v_k,v_k)\right)\\
&\geq\; \left(1-\frac{\lambda_1}{\lambda_2}\right)\sum_{k=2}^{\infty}\mathcal{B}(\boldsymbol{y};v_k,v_k)\\
&=\; \left(1-\frac{\lambda_1}{\lambda_2}\right)\mathcal{B}(\boldsymbol{y};v,v) \;\geq\; a_{\min}\left(1-\frac{\lambda_1}{\lambda_2}\right)\|v\|_V^2\,.
\end{aligned}$$

□

Remark 3.8. A similar estimate holds for the shifted bilinear form on $V_h \times V_h$, provided h is sufficiently small such that the FE eigenvalue gap is uniformly bounded from below. Indeed, we can write

$$\lambda_{2,h} - \lambda_{1,h} \;=\; (\lambda_{2,h} - \lambda_1) - (\lambda_{1,h} - \lambda_1)\,,$$

and since the FE eigenvalues converge from above we can bound this from below by

$$\lambda_{2,h} - \lambda_{1,h} \;\geq\; \lambda_2 - \lambda_1 - |\lambda_1 - \lambda_{1,h}| \;\geq\; \delta - Ch^2\,.$$

with $C > 0$. The second inequality follows from Proposition 3.4 and Theorem 3.6, which give us $\lambda_2 - \lambda_1 \geq \delta > 0$ and $|\lambda_1 - \lambda_{1,h}| \leq Ch^2$, respectively. Thus, choosing h such that $Ch^2 < \delta$, or equivalently, taking $h < h_0$ with

$$h_0 \coloneqq \left(\frac{\delta}{C}\right)^{\frac{1}{2}}\,, \tag{3.48}$$

is a sufficient condition for $\lambda_{2,h} - \lambda_{1,h} > 0$, and then Lemma 3.7 can be rewritten for the FE eigenproblem.

We also require the following technical lemma to handle some combinatorial factors that arise when bounding the derivatives.

Lemma 3.9. *Let $\epsilon \in (0,1)$. For all $N \in \mathbb{N}$ the following holds:*

$$S_N(\epsilon) \coloneqq \sum_{k=1}^{N-1} \binom{N}{k}^{-\epsilon} \leq C_\epsilon \coloneqq \frac{2^{1-\epsilon}}{1-2^{-\epsilon}} \left(\frac{e^2}{\sqrt{2\pi}}\right)^{\epsilon}\,.$$

Proof. For $N = 1$ the sum is empty and so $S_1(\epsilon) = 0 < C_\epsilon$; thus for the remainder of the proof we assume $N \geq 2$. By the symmetry of the binomial coefficient the sum can be bounded by

$$S_N(\epsilon) \;\leq\; 2\sum_{k=1}^{\lfloor \frac{N}{2} \rfloor} \binom{N}{k}^{-\epsilon} \;\leq\; 2\left(\frac{e^2}{\sqrt{2\pi}}\right)^{\epsilon} \sum_{k=1}^{\lfloor \frac{N}{2} \rfloor} \left(\frac{k^{k+\frac{1}{2}}(N-k)^{N-k+\frac{1}{2}}}{N^{N+\frac{1}{2}}}\right)^{\epsilon}\,,$$

where we used the following bounds given by Stirling's formula

$$\sqrt{2\pi} N^{N+\frac{1}{2}} e^{-N} \;\leq\; N! \;\leq\; e N^{N+\frac{1}{2}} e^{-N}\,.$$

Since $k \le \frac{N}{2}$ we have the bound $k^{k+\frac{1}{2}} \le (\frac{N}{2})^k \sqrt{k}$, which gives

$$\begin{aligned} S_N(\epsilon) \;&\le\; 2\left(\frac{e^2}{\sqrt{2\pi}}\right)^{\epsilon} \sum_{k=1}^{\lfloor \frac{N}{2} \rfloor} \left(\frac{\left(\frac{N}{2}\right)^k \sqrt{k}(N-k)^{N-k+\frac{1}{2}}}{N^{N+\frac{1}{2}}} \right)^{\epsilon} \\ &=\; 2\left(\frac{e^2}{\sqrt{2\pi}}\right)^{\epsilon} \sum_{k=1}^{\lfloor \frac{N}{2} \rfloor} \frac{1}{2^{\epsilon k}} \left(\sqrt{k} \left(1 - \frac{k}{N}\right)^{N-k+\frac{1}{2}} \right)^{\epsilon} \;=\; 2\left(\frac{e^2}{\sqrt{2\pi}}\right)^{\epsilon} \sum_{k=1}^{\lfloor \frac{N}{2} \rfloor} \frac{[R_N(k/N)]^{\epsilon}}{(2^{\epsilon})^k} \,. \end{aligned}$$

where for $N = 2, 3, \ldots$ we define the functions $R_N : [0, 1/2] \to \mathbb{R}$ by

$$R_N(x) \;\coloneqq\; \sqrt{Nx}\,(1-x)^{N(1-x)+\frac{1}{2}} \;.$$

The next step is to prove, by induction, that for $N = 2, 3, \ldots$

$$R_N(x) \;\le\; 1 \quad \text{for all } x \in [0, \tfrac{1}{2}] \,. \tag{3.49}$$

For $N = 2$

$$R_2(x) \;=\; \sqrt{2x}\,(1-x)^{2(1-x)+\frac{1}{2}} \;\le\; \sqrt{2 \cdot \frac{1}{2}}(1-x)^{2(1-x)+\frac{1}{2}} \;\le\; 1 \,.$$

Then for $N \ge 2$, suppose $R_N(x) \le 1$ and consider R_{N+1}. For $x \in [1/(N+1), 1/2]$ we have

$$\begin{aligned} R_{N+1}(x) \;&=\; \sqrt{(N+1)x}\,(1-x)^{(N+1)(1-x)+\frac{1}{2}} \\ &=\; \sqrt{\frac{N+1}{N}}\sqrt{Nx}\,(1-x)^{1-x}(1-x)^{N(1-x)+\frac{1}{2}} \\ &=\; (1-x)^{1-x}\sqrt{\frac{N+1}{N}} R_N(x) \;\le\; (1-x)^{1-x}\sqrt{\frac{N+1}{N}} \,. \end{aligned}$$

To bound this from above we bound one x below by $1/(N+1)$ to give

$$R_{N+1}(x) \;\le\; \left(1 - \frac{1}{N+1}\right)^{1-x} \sqrt{\frac{N+1}{N}} \;=\; \left(\frac{N}{N+1}\right)^{\frac{1}{2}-x} \;\le\; 1 \,.$$

And for $x \in [0, 1/(N+1)]$

$$R_{N+1}(x) \;\le\; \sqrt{(N+1)\frac{1}{N+1}}(1-x)^{(N+1)(1-x)+\frac{1}{2}} \;=\; (1-x)^{(N+1)(1-x)+\frac{1}{2}} \;\le\; 1 \,.$$

Thus, for all $N = 2, 3, \ldots$ and $x \in [0, 1/2]$ we have $R_N(x) \le 1$.

Returning to the sum S_N, since

$$\frac{k}{N} \in [0, \tfrac{1}{2}] \quad \text{for all } k \le \frac{N}{2}\,, \quad \text{and} \quad R_N\left(\tfrac{k}{N}\right) \,=\, \sqrt{k}\left(1-\frac{k}{N}\right)^{N-k+\frac{1}{2}}$$

we have, by (3.49),

$$\begin{aligned} S_N(\epsilon) \,&\le\, 2\left(\frac{e^2}{\sqrt{2\pi}}\right)^{\epsilon} \sum_{k=1}^{\lfloor \frac{N}{2} \rfloor} \frac{[R_N(k/N)]^{\epsilon}}{(2^{\epsilon})^k} \,\le\, 2\left(\frac{e^2}{\sqrt{2\pi}}\right)^{\epsilon} \sum_{k=1}^{\lfloor \frac{N}{2} \rfloor} \frac{1}{(2^{\epsilon})^k} \\ &\le\, 2\left(\frac{e^2}{\sqrt{2\pi}}\right)^{\epsilon} \sum_{k=1}^{\infty} \frac{1}{(2^{\epsilon})^k} \,=\, \frac{2^{1-\epsilon}}{1-2^{-\epsilon}}\left(\frac{e^2}{\sqrt{2\pi}}\right)^{\epsilon} \,=:\, C_{\epsilon}\,, \end{aligned}$$

where we used the formula for the sum of a geometric series. □

Lemma 3.10 below gives the bounds on the derivatives of λ_1 and u_1 required for our QMC error analysis. We prove the bounds for higher order mixed derivatives, which will be written in multi-index notation. Let $\boldsymbol{\nu} = (\nu_j)_{j\in\mathbb{N}}$, with $\nu_j \in \mathbb{N} \cup \{0\}$, be a multi-index with only finitely many non-zero entries and $|\boldsymbol{\nu}| := \sum_{j\ge 1} \nu_j < \infty$. Let $\mathfrak{F}$ denote the set of all admissible multi-indexes. We will use $\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}}$ to denote the mixed partial derivative where the element ν_j is the order of the derivative with respect to y_j. Operations between multi-indices are handled component wise. Thus, for $\boldsymbol{m} = (m_j)_{j\in\mathbb{N}}, \boldsymbol{\nu} = (\nu_j)_{j\in\mathbb{N}}$ we use the following notation: $\boldsymbol{\nu} - \boldsymbol{m} := (\nu_j - m_j)_{j\in\mathbb{N}}$; $\boldsymbol{m} \le \boldsymbol{\nu}$ if $m_j \le \nu_j$ for all $j \in \mathbb{N}$; $\boldsymbol{m} < \boldsymbol{\nu}$ if $\boldsymbol{m} \le \boldsymbol{\nu}$ and $\boldsymbol{m} \ne \boldsymbol{\nu}$; and $\binom{\boldsymbol{\nu}}{\boldsymbol{m}} := \prod_{j\in\mathbb{N}} \binom{\nu_j}{m_j}$. For $j \in \mathbb{N}$, the jth unit multi-index is denoted by $\boldsymbol{e}_j$, that is, $\boldsymbol{e}_j$ is 1 in the jth position and 0 everywhere else.

Since the coefficients $a(\boldsymbol{x}, \boldsymbol{y})$ and $b(\boldsymbol{x}, \boldsymbol{y})$ in (3.4) are linear in the parameter $\boldsymbol{y}$, their derivatives are (suppressing the $\boldsymbol{x}$, $\boldsymbol{y}$ dependence below)

$$\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} a \,=\, \begin{cases} a & \text{if } \boldsymbol{\nu} = \boldsymbol{0}\,, \\ a_j & \text{if } \boldsymbol{\nu} = \boldsymbol{e}_j\,, \\ 0 & \text{otherwise,} \end{cases} \quad \text{and} \quad \partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} b \,=\, \begin{cases} b & \text{if } \boldsymbol{\nu} = \boldsymbol{0}\,, \\ b_j & \text{if } \boldsymbol{\nu} = \boldsymbol{e}_j\,, \\ 0 & \text{otherwise.} \end{cases} \tag{3.50}$$

Lemma 3.10. *Let $\epsilon \in (0,1)$, $\boldsymbol{\nu} \in \mathfrak{F}$ be a multi-index, and suppose that Assumption A3.1 holds. Then for all $\boldsymbol{y} \in U$ the derivative of the smallest eigenvalue of* (3.12) *is bounded by*

$$|\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} \lambda_1(\boldsymbol{y})| \,\le\, \overline{\lambda_1}\,(|\boldsymbol{\nu}|!)^{1+\epsilon}\,\boldsymbol{\beta}^{\boldsymbol{\nu}}\,, \tag{3.51}$$

and the norm of the derivative of the corresponding eigenfunction is similarly bounded by

$$\left\|\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} u_1(\boldsymbol{y})\right\|_V \leq \overline{u_1}\,(|\boldsymbol{\nu}|!)^{1+\epsilon}\,\boldsymbol{\beta}^{\boldsymbol{\nu}}\,, \tag{3.52}$$

where $\overline{\lambda_1}$ *and* $\overline{u_1}$ *are defined in* (3.20) *and* (3.22)*, respectively. The sequence* $\boldsymbol{\beta} = (\beta_j)_{j\in\mathbb{N}}$ *is defined by*

$$\beta_j \coloneqq C_{\boldsymbol{\beta}} \max\left(\|a_j\|_{L^\infty(D)}, \|b_j\|_{L^\infty(D)}\right), \tag{3.53}$$

with $C_{\boldsymbol{\beta}} > 0$ *independent of* $\boldsymbol{y}$ *given by*

$$C_{\boldsymbol{\beta}} \coloneqq \frac{1}{a_{\min} C_{\mathrm{gap}}}\left(1+\tfrac{1}{\chi_1}\right)\left(\frac{3a_{\max}^2}{2a_{\min}^2}\left(1+\tfrac{1}{\chi_1}\right)C_\epsilon + 1\right), \tag{3.54}$$

where χ_1 *is again the fundamental eigenvalue of the negative Laplacian,* C_ϵ *is as in Lemma 3.9, and*

$$C_{\mathrm{gap}} \coloneqq \inf_{\boldsymbol{y}\in U}\left(1-\tfrac{\lambda_1(\boldsymbol{y})}{\lambda_2(\boldsymbol{y})}\right). \tag{3.55}$$

Proof. First of all, since the spectral gap is uniformly bounded away from 0 (see Proposition 3.4), we have $0 < C_{\mathrm{gap}} < 1$ and the constant $C_{\boldsymbol{\beta}}$ is finite. Also, for $\boldsymbol{\nu} = \mathbf{0}$ the bounds (3.51) and (3.52) clearly hold by (3.20) and (3.22), respectively.

For $\boldsymbol{\nu} \neq \mathbf{0}$, we will prove the two bounds by induction on $|\boldsymbol{\nu}|$. To this end, we first obtain recursive bounds by differentiating the variational form (3.8) with respect to the stochastic parameters $\boldsymbol{y} \in U$. From [1] we know that simple eigenpairs of (3.12) are analytic in $\boldsymbol{y}$, so the partial derivatives $\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}}\lambda_1$, $\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} u_1$ exist and we further have $\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} u_1 \in V$. Hence, we can differentiate (3.8) with $\lambda = \lambda_1$ and $u = u_1$ using the Leibniz general product rule to obtain the following formula, which is true for all $v \in V$,

$$\begin{aligned}\sum_{\boldsymbol{m}\leq\boldsymbol{\nu}} \binom{\boldsymbol{\nu}}{\boldsymbol{m}} \Bigg(&- \left(\partial_{\boldsymbol{y}}^{\boldsymbol{m}}\lambda_1(\boldsymbol{y})\right)\int_D c(\boldsymbol{x})\left(\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}-\boldsymbol{m}} u_1(\boldsymbol{x},\boldsymbol{y})\right) v(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \\ &+ \int_D \left(\partial_{\boldsymbol{y}}^{\boldsymbol{m}} a(\boldsymbol{x},\boldsymbol{y})\right)\nabla\left(\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}-\boldsymbol{m}} u_1(\boldsymbol{x},\boldsymbol{y})\right)\cdot\nabla v(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \\ &+ \int_D \left(\partial_{\boldsymbol{y}}^{\boldsymbol{m}} b(\boldsymbol{x},\boldsymbol{y})\right)\left(\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}-\boldsymbol{m}} u_1\right) v(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\Bigg) = 0\,. \end{aligned} \tag{3.56}$$

Henceforth, we consider $\boldsymbol{y} \in U$ to be fixed and will suppress the dependence of $a(\boldsymbol{x}, \boldsymbol{y})$, $b(\boldsymbol{x}, \boldsymbol{y}), c(\boldsymbol{x}), \lambda_1(\boldsymbol{y})$, $u_1(\boldsymbol{x}, \boldsymbol{y})$, and their respective derivatives, on $\boldsymbol{x}$ and $\boldsymbol{y}$.

To obtain a bound on the derivatives of the eigenvalue, we take $v = u_1$ in (3.56). In this case, the $\boldsymbol{m} = \mathbf{0}$ term vanishes since (3.8) is satisfied for $\lambda = \lambda_1$ and $u = u_1$ with $\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} u_1 \in V$ as a test function. Separating out the $\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} \lambda_1$ term gives

$$\begin{aligned}(\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} \lambda_1)\mathcal{D}(u_1, u_1) \;=\; &- \sum_{\mathbf{0} \neq \boldsymbol{m} < \boldsymbol{\nu}} \binom{\boldsymbol{\nu}}{\boldsymbol{m}} (\partial_{\boldsymbol{y}}^{\boldsymbol{m}} \lambda_1) \int_D c\, (\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}-\boldsymbol{m}} u_1) u_1 \\ &+ \sum_{\mathbf{0} \neq \boldsymbol{m} \leq \boldsymbol{\nu}} \binom{\boldsymbol{\nu}}{\boldsymbol{m}} \bigg(\int_D (\partial_{\boldsymbol{y}}^{\boldsymbol{m}} a) \nabla (\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}-\boldsymbol{m}} u_1) \cdot \nabla u_1 + \int_D (\partial_{\boldsymbol{y}}^{\boldsymbol{m}} b)(\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}-\boldsymbol{m}} u_1) u_1 \bigg).\end{aligned}$$

The left-hand side can be simplified using $\|u_1\|_{\mathcal{D}} = 1$, and by (3.50) the terms involving higher-order derivatives of the coefficients are 0, so we have the recursive formula

$$\begin{aligned}\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} \lambda_1 \;= &- \sum_{\mathbf{0} \neq \boldsymbol{m} < \boldsymbol{\nu}} \binom{\boldsymbol{\nu}}{\boldsymbol{m}} (\partial_{\boldsymbol{y}}^{\boldsymbol{m}} \lambda_1) \int_D c \left(\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}-\boldsymbol{m}} u_1\right) u_1 \\ &+ \sum_{j=1}^{\infty} \nu_j \bigg(\int_D a_j \nabla \left(\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}-\boldsymbol{e}_j} u_1\right) \cdot \nabla u_1 \;+\; \int_D b_j \left(\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}-\boldsymbol{e}_j} u_1\right) u_1 \bigg).\end{aligned}$$

Taking the absolute value then applying the triangle and Cauchy-Schwarz inequalities gives the upper bound

$$\begin{aligned}|\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} \lambda_1| \;\leq\; &\sum_{\mathbf{0} \neq \boldsymbol{m} < \boldsymbol{\nu}} \binom{\boldsymbol{\nu}}{\boldsymbol{m}} |\partial_{\boldsymbol{y}}^{\boldsymbol{m}} \lambda_1| \left\|\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}-\boldsymbol{m}} u_1\right\|_{\mathcal{D}} \|u_1\|_{\mathcal{D}} \\ &+ \sum_{j=1}^{\infty} \nu_j \left(\|a_j\|_{L^\infty(D)} \left\|\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}-\boldsymbol{e}_j} u_1\right\|_V \|u_1\|_V + \|b_j\|_{L^\infty(D)} \left\|\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}-\boldsymbol{e}_j} u_1\right\|_{L^2(D)} \|u_1\|_{L^2(D)} \right).\end{aligned}$$

Then, by the equivalence of the norms $\|\cdot\|_{\mathcal{D}}$ and $\|\cdot\|_{L^2(D)}$ in (3.11), the Poincaré inequality (3.15), the upper bound (3.22) on $\|u_1\|_V$, and the normalisation of u_1, we have

$$\begin{aligned}|\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} \lambda_1| \;\leq\; &\sqrt{\frac{a_{\max}}{\chi_1}} \sum_{\mathbf{0} \neq \boldsymbol{m} < \boldsymbol{\nu}} \binom{\boldsymbol{\nu}}{\boldsymbol{m}} |\partial_{\boldsymbol{y}}^{\boldsymbol{m}} \lambda_1| \left\|\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}-\boldsymbol{m}} u_1\right\|_V \\ &+ \frac{\sqrt{a_{\max}(\chi_1 + 1)}}{a_{\min}} \sum_{j=1}^{\infty} \nu_j \left(\|a_j\|_{L^\infty(D)} + \tfrac{1}{\chi_1} \|b_j\|_{L^\infty(D)} \right) \left\|\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}-\boldsymbol{e}_j} u_1\right\|_V .\end{aligned}$$

Defining β_j as in (3.53) but leaving $C_{\boldsymbol{\beta}} > 0$ to be specified later, we obtain

$$|\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} \lambda_1| \le \sqrt{\frac{a_{\max}}{\chi_1}} \sum_{\mathbf{0} \ne \boldsymbol{m} < \boldsymbol{\nu}} \binom{\boldsymbol{\nu}}{\boldsymbol{m}} |\partial_{\boldsymbol{y}}^{\boldsymbol{m}} \lambda_1| \left\| \partial_{\boldsymbol{y}}^{\boldsymbol{\nu}-\boldsymbol{m}} u_1 \right\|_V + \frac{\sqrt{a_{\max}(\chi_1+1)}}{a_{\min}} \left(1 + \tfrac{1}{\chi_1}\right) \sum_{j=1}^{\infty} \nu_j \frac{\beta_j}{C_{\boldsymbol{\beta}}} \left\| \partial_{\boldsymbol{y}}^{\boldsymbol{\nu}-\boldsymbol{e}_j} u_1 \right\|_V . \tag{3.57}$$

Observe that this bound on the derivative of λ_1 depends on the lower order derivatives of both λ_1 and u_1.

To obtain a similar bound on the derivatives of the eigenfunction one might suppose that we let $v = \partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} u_1$ in (3.56), but this will not work. This is because in (3.56) the bilinear form acting on $\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} u_1$ (exactly $\mathcal{B}_{\lambda_1}^{\mathrm{sh}}$ from Lemma 3.7) is not coercive on the whole domain $V \times V$. The way around this is to note that $\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} u_1$ can be decomposed as a linear combination of the eigenfunctions and utilise the estimate in Lemma 3.7. We write $\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} u_1$ as

$$\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} u_1 = \sum_{k \in \mathbb{N}} \mathcal{D}(\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} u_1, u_k) u_k = \mathcal{D}(\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} u_1, u_1) u_1 + \widetilde{v}\,, \tag{3.58}$$

so that $\widetilde{v} \in E(\boldsymbol{y}, \lambda_1(\boldsymbol{y}))^{\perp}$ is the $\mathcal{D}$-orthogonal projection of $\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} u_1$ onto $E(\boldsymbol{y}, \lambda_1(\boldsymbol{y}))^{\perp}$. Applying the triangle inequality to this decomposition and then using (3.22) we can bound the norm by

$$\left\| \partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} u_1 \right\|_V \le \frac{\sqrt{a_{\max}(\chi_1+1)}}{a_{\min}} |\mathcal{D}(\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} u_1, u_1)| + \|\widetilde{v}\|_V . \tag{3.59}$$

Hence, it remains to bound $\mathcal{D}(\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} u_1, u_1)$ and $\|\widetilde{v}\|_V$. For the former, since $\mathcal{D}(u_1, u_1) = 1$ we have

$$0 = \partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} \mathcal{D}(u_1, u_1) = \sum_{\boldsymbol{m} \le \boldsymbol{\nu}} \binom{\boldsymbol{\nu}}{\boldsymbol{m}} \mathcal{D}(\partial_{\boldsymbol{y}}^{\boldsymbol{m}} u_1, \partial_{\boldsymbol{y}}^{\boldsymbol{\nu}-\boldsymbol{m}} u_1)\,.$$

By separating out the $\boldsymbol{m} = \mathbf{0}$ and $\boldsymbol{m} = \boldsymbol{\nu}$ terms, which are equal by symmetry, we obtain

$$\begin{aligned} |\mathcal{D}(\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} u_1, u_1)| &= \left| -\frac{1}{2} \sum_{\mathbf{0} \ne \boldsymbol{m} < \boldsymbol{\nu}} \binom{\boldsymbol{\nu}}{\boldsymbol{m}} \mathcal{D}(\partial_{\boldsymbol{y}}^{\boldsymbol{m}} u_1, \partial_{\boldsymbol{y}}^{\boldsymbol{\nu}-\boldsymbol{m}} u_1) \right| \\ &\le \frac{a_{\max}}{2\chi_1} \sum_{\mathbf{0} \ne \boldsymbol{m} < \boldsymbol{\nu}} \binom{\boldsymbol{\nu}}{\boldsymbol{m}} \left\| \partial_{\boldsymbol{y}}^{\boldsymbol{m}} u_1 \right\|_V \left\| \partial_{\boldsymbol{y}}^{\boldsymbol{\nu}-\boldsymbol{m}} u_1 \right\|_V , \end{aligned} \tag{3.60}$$

where we used the Cauchy-Schwarz inequality, the norm equivalence (3.11) and then the Poincaré inequality (3.15) to obtain V-norms.

For the V-norm of $\widetilde{v}$ we let $v = \widetilde{v}$ in (3.56) and separate out the $\boldsymbol{m} = \mathbf{0}$ term to give

$$\begin{aligned}\mathcal{B}\left(\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} u_1, \widetilde{v}\right) - \lambda_1 \mathcal{D}\left(\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} u_1, \widetilde{v}\right) \;=\; & \sum_{\mathbf{0} \neq \boldsymbol{m} < \boldsymbol{\nu}} \binom{\boldsymbol{\nu}}{\boldsymbol{m}} (\partial_{\boldsymbol{y}}^{\boldsymbol{m}} \lambda_1) \int_D c\, (\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}-\boldsymbol{m}} u_1) \widetilde{v} \\ & - \sum_{j=1}^{\infty} \nu_j \left(\int_D a_j \nabla(\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}-\boldsymbol{e}_j} u_1) \cdot \nabla \widetilde{v} + \int_D b_j (\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}-\boldsymbol{e}_j} u_1) \widetilde{v} \right), \end{aligned} \tag{3.61}$$

where the $\boldsymbol{m} = \boldsymbol{\nu}$ term vanishes from the right-hand side since $\widetilde{v}$ is orthogonal to u_1. Decomposing $\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} u_1$ as in (3.58) and then expanding, the left-hand side of (3.61) becomes

$$\begin{aligned} \text{LHS of (3.61)} \;&=\; \mathcal{B}\left(\mathcal{D}(\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} u_1, u_1) u_1 + \widetilde{v}, \widetilde{v}\right) - \lambda_1 \mathcal{D}\left(\mathcal{D}(\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} u_1, u_1) u_1 + \widetilde{v}, \widetilde{v}\right) \\ &=\; \mathcal{D}(\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} u_1, u_1)\big(\mathcal{B}(u_1, \widetilde{v}) - \lambda_1 \mathcal{D}(u_1, \widetilde{v})\big) + \mathcal{B}(\widetilde{v}, \widetilde{v}) - \lambda_1 \mathcal{D}(\widetilde{v}, \widetilde{v}) \\ &=\; \mathcal{B}(\widetilde{v}, \widetilde{v}) - \lambda_1 \mathcal{D}(\widetilde{v}, \widetilde{v}) \\ &\geq\; a_{\min} \left(1 - \tfrac{\lambda_1}{\lambda_2}\right) \|\widetilde{v}\|_V^2 \;\geq\; a_{\min} C_{\text{gap}} \|\widetilde{v}\|_V^2 \;, \end{aligned}$$

where the first term on the second line is 0 by (3.8) with $\widetilde{v}$ as a test function. The lower bound follows by the coercivity estimate (3.47) in Lemma 3.7, since $\widetilde{v} \in E(\boldsymbol{y}, \lambda_1(\boldsymbol{y}))^{\perp}$.

We can bound the right-hand side of (3.61) from above using a combination of the triangle, Cauchy-Schwarz, equivalence of norms (3.11) and Poincaré (3.15) inequalities to obtain

$$\begin{aligned} a_{\min} C_{\text{gap}} \|\widetilde{v}\|_V^2 \leq\; & \frac{a_{\max}}{\chi_1} \sum_{\mathbf{0} \neq \boldsymbol{m} < \boldsymbol{\nu}} \binom{\boldsymbol{\nu}}{\boldsymbol{m}} |\partial_{\boldsymbol{y}}^{\boldsymbol{m}} \lambda_1| \left\|\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}-\boldsymbol{m}} u_1\right\|_V \|\widetilde{v}\|_V \\ & + \sum_{j=1}^{\infty} \nu_j \left(\|a_j\|_{L^\infty(D)} + \tfrac{1}{\chi_1} \|b_j\|_{L^\infty(D)} \right) \|\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}-\boldsymbol{e}_j} u_1\|_V \|\widetilde{v}\|_V \;. \end{aligned}$$

Dividing through by $a_{\min} C_{\text{gap}} \|\widetilde{v}\|_V$ and using the definition of β_j in (3.53), again leaving $C_{\boldsymbol{\beta}} > 0$ to be specified later, we obtain

$$\begin{aligned} \|\widetilde{v}\|_V \;\leq\; \frac{1}{C_{\text{gap}}} \Bigg(& \frac{a_{\max}}{a_{\min}\, \chi_1} \sum_{\mathbf{0} \neq \boldsymbol{m} < \boldsymbol{\nu}} \binom{\boldsymbol{\nu}}{\boldsymbol{m}} |\partial_{\boldsymbol{y}}^{\boldsymbol{m}} \lambda_1| \left\|\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}-\boldsymbol{m}} u_1\right\|_V \\ & + \frac{1}{a_{\min}} \left(1 + \tfrac{1}{\chi_1}\right) \sum_{j=1}^{\infty} \nu_j \frac{\beta_j}{C_{\boldsymbol{\beta}}} \left\|\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}-\boldsymbol{e}_j} u_1\right\|_V \Bigg). \end{aligned} \tag{3.62}$$

Substituting the two bounds (3.60) and (3.62) into (3.59), the norm of the derivative of the eigenfunction is bounded above by

$$\begin{aligned}\left\|\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} u_1\right\|_V \leq & \frac{a_{\max}\sqrt{a_{\max}(\chi_1+1)}}{2\chi_1 a_{\min}} \sum_{\mathbf{0}\neq\boldsymbol{m}<\boldsymbol{\nu}} \binom{\boldsymbol{\nu}}{\boldsymbol{m}} \|\partial_{\boldsymbol{y}}^{\boldsymbol{m}} u_1\|_V \|\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}-\boldsymbol{m}} u_1\|_V \\ & + \frac{a_{\max}}{\chi_1 a_{\min} C_{\mathrm{gap}}} \sum_{\mathbf{0}\neq\boldsymbol{m}<\boldsymbol{\nu}} \binom{\boldsymbol{\nu}}{\boldsymbol{m}} |\partial_{\boldsymbol{y}}^{\boldsymbol{m}} \lambda_1| \left\|\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}-\boldsymbol{m}} u_1\right\|_V \\ & + \frac{1}{a_{\min} C_{\mathrm{gap}}} \left(1+\tfrac{1}{\chi_1}\right) \sum_{j=1}^{\infty} \nu_j \frac{\beta_j}{C_{\boldsymbol{\beta}}} \left\|\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}-\boldsymbol{e}_j} u_1\right\|_V . \end{aligned} \tag{3.63}$$

We are now ready to prove the bounds (3.51) and (3.52) by induction, but for the inductive step to work we require tighter constants than $\overline{\lambda_1}$ and $\overline{u_1}$, otherwise at each step they grow too large. Since the bounds (3.57) and (3.63) are true for all $\boldsymbol{\nu} \neq \mathbf{0}$, and these bounds do not involve the $\boldsymbol{\nu} = \mathbf{0}$ cases, we will use them to establish the base step of the induction $|\boldsymbol{\nu}| = 1$. Letting $\boldsymbol{\nu} = \boldsymbol{e}_i$ in (3.57) and (3.63) gives, with the aid of (3.22),

$$\begin{aligned} |\partial_{\boldsymbol{y}}^{\boldsymbol{e}_i} \lambda_1| &\leq \frac{a_{\max}(\chi_1+1)}{a_{\min}^2} \left(1+\tfrac{1}{\chi_1}\right) \frac{\beta_i}{C_{\boldsymbol{\beta}}}, \quad \text{and} \\ \left\|\partial_{\boldsymbol{y}}^{\boldsymbol{e}_i} u_1\right\|_V &\leq \frac{\sqrt{a_{\max}(\chi_1+1)}}{a_{\min}^2 C_{\mathrm{gap}}} \left(1+\tfrac{1}{\chi_1}\right) \frac{\beta_i}{C_{\boldsymbol{\beta}}}. \end{aligned}$$

Thus we proceed to prove that, for $\boldsymbol{\nu} \neq \mathbf{0}$,

$$|\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} \lambda_1(\boldsymbol{y})| \leq C_1 \left(|\boldsymbol{\nu}|!\right)^{1+\epsilon} \boldsymbol{\beta}^{\boldsymbol{\nu}}, \quad \text{and} \tag{3.64}$$

$$\left\|\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} u_1(\boldsymbol{y})\right\|_V \leq C_2 \left(|\boldsymbol{\nu}|!\right)^{1+\epsilon} \boldsymbol{\beta}^{\boldsymbol{\nu}}, \tag{3.65}$$

where

$$\begin{aligned} C_1 &:= \frac{a_{\max}(\chi_1+1)}{a_{\min}^2} \left(1+\tfrac{1}{\chi_1}\right) \frac{1}{C_{\boldsymbol{\beta}}}, \quad \text{and} \\ C_2 &:= \frac{\sqrt{a_{\max}(\chi_1+1)}}{a_{\min}^2 C_{\mathrm{gap}}} \left(1+\tfrac{1}{\chi_1}\right) \frac{1}{C_{\boldsymbol{\beta}}}, \end{aligned}$$

with $C_{\boldsymbol{\beta}} > 0$ still to be specified later to ensure that $C_1 \leq \overline{\lambda_1}$ and $C_2 \leq \overline{u_1}$.

For the inductive step for the eigenvalue derivative bound, suppose that $|\boldsymbol{\nu}| \geq 2$ and that the bounds (3.64) and (3.65) hold for all multi-indices of order $< |\boldsymbol{\nu}|$.

Substituting the induction assumptions into (3.57) gives

$$|\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}}\lambda_1| \leq \sqrt{\frac{a_{\max}}{\chi_1}} \sum_{\mathbf{0}\neq\boldsymbol{m}<\boldsymbol{\nu}} \binom{\boldsymbol{\nu}}{\boldsymbol{m}} C_1 \left(|\boldsymbol{m}|!\right)^{1+\epsilon} \boldsymbol{\beta}^{\boldsymbol{m}} \cdot C_2 \left(|\boldsymbol{\nu}-\boldsymbol{m}|!\right)^{1+\epsilon} \boldsymbol{\beta}^{\boldsymbol{\nu}-\boldsymbol{m}} + \frac{\sqrt{a_{\max}(\chi_1+1)}}{a_{\min}} \left(1+\tfrac{1}{\chi_1}\right) \sum_{j\in\mathbb{N}} \nu_j \frac{\beta_j}{C_{\boldsymbol{\beta}}} \cdot C_2 \left[(|\boldsymbol{\nu}|-1)!\right]^{1+\epsilon} \boldsymbol{\beta}^{\boldsymbol{\nu}-\boldsymbol{e}_j} .$$

Factoring out $C_1\boldsymbol{\beta}^{\boldsymbol{\nu}}$ this simplifies to

$$|\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}}\lambda_1| \leq C_1\boldsymbol{\beta}^{\boldsymbol{\nu}} \left(\sqrt{\frac{a_{\max}}{\chi_1}} C_2 \sum_{\mathbf{0}\neq\boldsymbol{m}<\boldsymbol{\nu}} \binom{\boldsymbol{\nu}}{\boldsymbol{m}} (|\boldsymbol{m}|!)^{1+\epsilon} (|\boldsymbol{\nu}-\boldsymbol{m}|!)^{1+\epsilon} + \frac{\sqrt{a_{\max}(\chi_1+1)}}{a_{\min}} \left(1+\tfrac{1}{\chi_1}\right) \frac{C_2}{C_1 C_{\boldsymbol{\beta}}} |\boldsymbol{\nu}| [(|\boldsymbol{\nu}|-1)!]^{1+\epsilon} \right).$$

Using the identity $\sum_{\boldsymbol{m}\leq\boldsymbol{\nu},|\boldsymbol{m}|=k} \binom{\boldsymbol{\nu}}{\boldsymbol{m}} = \binom{|\boldsymbol{\nu}|}{k}$ along with Lemma 3.9, we can bound the sum as follows

$$\begin{aligned} &\sum_{\mathbf{0}\neq\boldsymbol{m}<\boldsymbol{\nu}} \binom{\boldsymbol{\nu}}{\boldsymbol{m}} (|\boldsymbol{m}|!)^{1+\epsilon} (|\boldsymbol{\nu}-\boldsymbol{m}|!)^{1+\epsilon} \\ &= \sum_{k=1}^{|\boldsymbol{\nu}|-1} \sum_{\boldsymbol{m}\leq\boldsymbol{\nu},|\boldsymbol{m}|=k} \binom{\boldsymbol{\nu}}{\boldsymbol{m}} (k!)^{1+\epsilon} [(|\boldsymbol{\nu}|-k)!]^{1+\epsilon} \\ &= (|\boldsymbol{\nu}|!)^{1+\epsilon} \sum_{k=1}^{|\boldsymbol{\nu}|-1} \binom{|\boldsymbol{\nu}|}{k} \frac{(k!)^{1+\epsilon}[(|\boldsymbol{\nu}|-k)!]^{1+\epsilon}}{(|\boldsymbol{\nu}|!)^{1+\epsilon}} \\ &= (|\boldsymbol{\nu}|!)^{1+\epsilon} S_{|\boldsymbol{\nu}|}(\epsilon) \leq C_\epsilon (|\boldsymbol{\nu}|!)^{1+\epsilon}. \end{aligned} \tag{3.66}$$

Substituting this into the bound on $|\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}}\lambda_1|$ yields

$$|\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}}\lambda_1| \leq C_1 (|\boldsymbol{\nu}|!)^{1+\epsilon} \boldsymbol{\beta}^{\boldsymbol{\nu}} \left[\sqrt{\frac{a_{\max}}{\chi_1}} C_2 C_\epsilon + \frac{\sqrt{a_{\max}(\chi_1+1)}}{a_{\min}} \left(1+\tfrac{1}{\chi_1}\right) \frac{C_2}{C_1 C_{\boldsymbol{\beta}}} \right].$$

Substituting in the values for C_1 and C_2, the expression in between the square brackets simplifies to

$$\frac{1}{C_{\boldsymbol{\beta}}} \frac{1}{a_{\min} C_{\text{gap}}} \left(1+\tfrac{1}{\chi_1}\right) \left(\frac{a_{\max}}{a_{\min}} \sqrt{1+\tfrac{1}{\chi_1}}\, C_\epsilon + 1 \right), \tag{3.67}$$

and we will later specify $C_{\boldsymbol{\beta}}$ to ensure that this expression is bounded by 1, thus giving the required result (3.64).

For the eigenfunction derivative bounds, substituting the induction hypotheses (3.64) and (3.65) into (3.63) gives

$$\begin{aligned}\left\|\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} u_1\right\|_V \le\ & \frac{a_{\max}\sqrt{a_{\max}(\chi_1+1)}}{2\chi_1 a_{\min}} \sum_{\mathbf{0}\neq\boldsymbol{m}<\boldsymbol{\nu}} \binom{\boldsymbol{\nu}}{\boldsymbol{m}} C_2(|\boldsymbol{m}|!)^{1+\epsilon}\boldsymbol{\beta}^{\boldsymbol{m}} \cdot C_2(|\boldsymbol{\nu}-\boldsymbol{m}|!)^{1+\epsilon}\boldsymbol{\beta}^{\boldsymbol{\nu}-\boldsymbol{m}} \\ &+ \frac{a_{\max}}{\chi_1 a_{\min} C_{\mathrm{gap}}} \sum_{\mathbf{0}\neq\boldsymbol{m}<\boldsymbol{\nu}} \binom{\boldsymbol{\nu}}{\boldsymbol{m}} C_1(|\boldsymbol{m}|!)^{1+\epsilon}\boldsymbol{\beta}^{\boldsymbol{m}} \cdot C_2\left(|\boldsymbol{\nu}-\boldsymbol{m}|!\right)^{1+\epsilon}\boldsymbol{\beta}^{\boldsymbol{\nu}-\boldsymbol{m}} \\ &+ \frac{1}{a_{\min} C_{\mathrm{gap}}}\left(1+\tfrac{1}{\chi_1}\right)\sum_{j=1}^{\infty}\nu_j\,\frac{\beta_j}{C_{\boldsymbol{\beta}}}\,C_2[(|\boldsymbol{\nu}|-1)!]^{1+\epsilon}\boldsymbol{\beta}^{\boldsymbol{\nu}-\boldsymbol{e}_j}\,.\end{aligned}$$

Factoring out $C_2\boldsymbol{\beta}^{\boldsymbol{\nu}}$ and using (3.66) this becomes

$$\begin{aligned}\left\|\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} u_1\right\|_V \le\ & C_2\left(|\boldsymbol{\nu}|!\right)^{1+\epsilon}\boldsymbol{\beta}^{\boldsymbol{\nu}}\left[\frac{a_{\max}\sqrt{a_{\max}(\chi_1+1)}}{2\chi_1 a_{\min}} C_2 C_\epsilon\right. \\ &\left. + \frac{a_{\max}}{\chi_1 a_{\min} C_{\mathrm{gap}}} C_1 C_\epsilon + \frac{1}{a_{\min} C_{\mathrm{gap}}}\left(1+\tfrac{1}{\chi_1}\right)\frac{1}{C_{\boldsymbol{\beta}}}\right].\end{aligned}$$

Substituting in C_1 and C_2, the expression in between the square brackets simplifies to

$$\frac{1}{C_{\boldsymbol{\beta}}}\frac{1}{a_{\min} C_{\mathrm{gap}}}\left(1+\tfrac{1}{\chi_1}\right)\left(\frac{3a_{\max}^2}{2a_{\min}^2}\left(1+\tfrac{1}{\chi_1}\right)C_\epsilon + 1\right). \tag{3.68}$$

We now define $C_{\boldsymbol{\beta}}$ as in (3.54) so that the expression in (3.68) is exactly 1, thus proving the required bound for the eigenfunction (3.65), and ensuring also that the expression in (3.67) is bounded by 1 as required. This completes the induction proof for (3.64) and (3.65) for all $\boldsymbol{\nu} \neq \mathbf{0}$.

With this definition of $C_{\boldsymbol{\beta}}$ it can be verified that $C_1 \le \overline{\lambda_1}$ and $C_2 \le \overline{u_1}$ as required. Hence we have also proved (3.51) and (3.52) for all $\boldsymbol{\nu} \in \mathfrak{F}$. □

Remark 3.11. Since $V_h \subset V$, similar results hold for the FE approximations provided h is sufficiently small. In this case the constants are replaced by their FE counterparts $C_{\boldsymbol{\beta},h}, \overline{\lambda_{1,h}}, \overline{u_{1,h}}$.

3.4 Error analysis

Since we are only interested in the fundamental eigenpair, to aid in the notation we drop the subscript 1 and define $(\lambda, u) \coloneqq (\lambda_1, u_1)$. Henceforth, with a slight abuse of notation, we will use combinations of the subscripts s, h, n to denote, respectively, truncating the stochastic dimension to s variables, a FE approximation with mesh-width h and a lattice rule approximation with n points. Also, for the dimension truncation we will denote the truncated parameter vector by $\boldsymbol{y}_s \coloneqq (y_1, y_2, \ldots, y_s)$.

3.4.1 Dimension truncation error

To prove estimates on the strong and weak truncation error of the fundamental eigenpair we will make extensive use of Taylor series expansions in the variables $\boldsymbol{y}_{\{s+1,\dots\}} = (y_j)_{j\geq s+1}$ about $\mathbf{0}$.

Clearly, if the functions $(a_j)_{j\geq 1}, (b_j)_{j\geq 1}$ satisfy Assumption A3.1.3 then the sequence $\boldsymbol{\beta}$ is summable with the same p. Also, we will henceforth assume that $\boldsymbol{\beta}$ is ordered such that $\beta_1 \geq \beta_2 \geq \cdots$.

Theorem 3.12. *Let $s \in \mathbb{N}$. For all $\boldsymbol{y} \in U$ the strong truncation error of the minimal eigenpair is bounded by*

$$|\lambda(\boldsymbol{y}) - \lambda_s(\boldsymbol{y}_s)| \leq \frac{\overline{\lambda_1}}{2} \sum_{j\geq s+1} \beta_j \,, \tag{3.69}$$

$$\|u(\boldsymbol{y}) - u_s(\boldsymbol{y}_s)\|_V \leq \frac{\overline{u_1}}{2} \sum_{j\geq s+1} \beta_j \,. \tag{3.70}$$

The weak truncation error is bounded by

$$|\mathbb{E}_{\boldsymbol{y}}[\lambda - \lambda_s]| \leq 2\overline{\lambda_1} \left(\sum_{j\geq s+1} \beta_j \right)^2 , \tag{3.71}$$

and for $\mathcal{G} \in V^*$

$$|\mathbb{E}_{\boldsymbol{y}}[\mathcal{G}(u) - \mathcal{G}(u_s)]| \leq 2\overline{u_1}\, \|\mathcal{G}\|_{V^*} \left(\sum_{j\geq s+1} \beta_j \right)^2 . \tag{3.72}$$

Here $\overline{\lambda_1}$ and $\overline{u_1}$ are given in (3.20) *and* (3.22)*, respectively.*

Proof. We will prove the four bounds in order and by using different order Taylor series expansions with integral remainders. Since λ is analytic in $\boldsymbol{y}$, Taylor's Theorem allows us to expand λ as a zeroth order Taylor series in the variables $(y_j)_{j\geq s+1}$ about the point $\mathbf{0}$:

$$\lambda(\boldsymbol{y}) = \lambda(\boldsymbol{y}_s; \mathbf{0}) + \sum_{j\geq s+1} \int_0^1 \frac{\partial \lambda}{\partial y_j}(\boldsymbol{y}_s; t\boldsymbol{y}_{\{s+1,\dots\}}) y_j \,\mathrm{d}t \,.$$

Using the triangle inequality, that $|y_j| \leq \frac{1}{2}$ and the upper bound (3.51) we have

$$|\lambda(\boldsymbol{y}) - \lambda_s(\boldsymbol{y}_s)| \leq \sum_{j\geq s+1} \frac{\overline{\lambda_1}}{2} \beta_j \,.$$

The eigenfunction is also analytic so using a Taylor series expansion again the truncation error for the eigenfunction is

$$\begin{aligned}\|u(\boldsymbol{y}) - u_s(\cdot, \boldsymbol{y}_s)\|_V &= \left\| \sum_{j \geq s+1} \int_0^1 \frac{\partial u}{\partial y_j}(\boldsymbol{y}_s; t\boldsymbol{y}_{\{s+1,\ldots\}}) y_j \, \mathrm{d}t \right\|_V \\ &\leq \sum_{j \geq s+1} \frac{1}{2} \int_0^1 \left\| \frac{\partial u}{\partial y_j}(\boldsymbol{y}_s; t\boldsymbol{y}_{\{s+1,\ldots\}}) \right\|_V \mathrm{d}t \leq \sum_{j \geq s+1} \frac{1}{2} \overline{u_1} \, \beta_j \,,\end{aligned}$$

where we have used the upper bound (3.52).

For the weak error consider the first order Taylor series expansion of λ:

$$\begin{aligned}\lambda(\boldsymbol{y}) =& \lambda(\boldsymbol{y}_s; \boldsymbol{0}) + \sum_{j \geq s+1} \frac{\partial \lambda}{\partial y_j}(\boldsymbol{y}_s; \boldsymbol{0}) y_j \\ &+ \sum_{i,j \geq s+1} \frac{2}{(\boldsymbol{e}_i + \boldsymbol{e}_j)!} \int_0^1 (1-t) \frac{\partial^2 \lambda}{\partial y_i \partial y_j}(\boldsymbol{y}_s; t\boldsymbol{y}_{\{s+1,\ldots\}}) y_i y_j \, \mathrm{d}t \,.\end{aligned}$$

Taking the expected value with respect to $\boldsymbol{y}$, by linearity we obtain

$$\begin{aligned}\mathbb{E}_{\boldsymbol{y}}[\lambda - \lambda_s] =& \sum_{j \geq s+1} \mathbb{E}_{\boldsymbol{y}} \left[\frac{\partial \lambda}{\partial y_j}(\boldsymbol{y}_s; 0) y_j \right] \\ &+ \sum_{i,j \geq s+1} \mathbb{E}_{\boldsymbol{y}} \left[\frac{2}{(\boldsymbol{e}_i + \boldsymbol{e}_j)!} \int_0^1 (1-t) \frac{\partial^2 \lambda}{\partial y_i \partial y_j}(\boldsymbol{y}_s; t\boldsymbol{y}_{\{s+1,\ldots\}}) y_i y_j \mathrm{d}t \right] .\end{aligned}$$

Since each y_j are independent with mean zero, for $j \geq s+1$ we have

$$\mathbb{E}_{\boldsymbol{y}} \left[\frac{\partial \lambda}{\partial y_j}(\boldsymbol{y}_s; 0) y_j \right] = \mathbb{E}_{\boldsymbol{y}} \left[\frac{\partial \lambda}{\partial y_j}(\boldsymbol{y}_s; 0) \right] \mathbb{E}_{\boldsymbol{y}}[y_j] = 0 \,.$$

By (3.52), for all $\boldsymbol{y} \in U$ and $i, j \in \mathbb{N}$

$$\left| \frac{\partial^2 \lambda}{\partial y_i \partial y_j}(\boldsymbol{y}) y_i y_j \right| \leq \frac{1}{4} \overline{\lambda_1} \, 2^{1+\epsilon} \beta_i \beta_j \leq \overline{\lambda_1} \, \beta_i \beta_j \,,$$

since $\epsilon \leq 1$. From this and the triangle inequality it follows that the error is bounded by

$$|\mathbb{E}[\lambda - \lambda_s]| \leq 2 \sum_{i,j \geq s+1} \mathbb{E}_{\boldsymbol{y}} \left[\int_0^1 (1-t) \overline{\lambda_1} \, \beta_i \beta_j \, \mathrm{d}t \right] \leq 2 \overline{\lambda_1} \left(\sum_{j \geq s+1} \beta_j \right)^2 .$$

Similarly, the first order Taylor expansion for the eigenfunction is

$$\begin{aligned} u(\boldsymbol{y}) \; &= u(\boldsymbol{y}_s;\boldsymbol{0}) + \sum_{j\geq s+1} \frac{\partial u}{\partial y_j}(\boldsymbol{y}_s;\boldsymbol{0}) y_j \\ &\quad + \sum_{i,j\geq s+1} \frac{2}{(\boldsymbol{e}_i+\boldsymbol{e}_j)!} \int_0^1 (1-t) \frac{\partial^2 u}{\partial y_i \partial y_j}(\boldsymbol{y}_s; t\boldsymbol{y}_s) y_i y_j \,\mathrm{d}t \,. \end{aligned}$$

Taking the linear functional $\mathcal{G}$ and then the expected value of both sides

$$\begin{aligned} \mathbb{E}_{\boldsymbol{y}}\big[\mathcal{G}(u) - \mathcal{G}(u_s)\big] \; &= \sum_{j\geq s+1} \mathbb{E}_{\boldsymbol{y}}\left[\mathcal{G}\left(\frac{\partial u}{\partial y_j}(\boldsymbol{y}_s;\boldsymbol{0}) y_j\right)\right] \\ &\quad + \sum_{i,j\geq s+1} \mathbb{E}_{\boldsymbol{y}}\left[\mathcal{G}\left(\frac{2}{(\boldsymbol{e}_i+\boldsymbol{e}_j)!} \int_0^1 (1-t) \frac{\partial^2 u}{\partial y_i \partial y_j}(\boldsymbol{y}_s; t\boldsymbol{y}_s) y_i y_j \,\mathrm{d}t\right)\right] . \end{aligned}$$

For $j \geq s+1$, using the linearity of $\mathcal{G}$ and the independence of the variables $\boldsymbol{y}$

$$\mathbb{E}_{\boldsymbol{y}}\left[\mathcal{G}\left(\frac{\partial u}{\partial y_j}(\boldsymbol{y}_s;\boldsymbol{0}) y_j\right)\right] \;=\; \mathbb{E}_{\boldsymbol{y}}\left[\mathcal{G}\left(\frac{\partial u}{\partial y_j}(\boldsymbol{y}_s;\boldsymbol{0})\right)\right] \mathbb{E}_{\boldsymbol{y}}[y_j] \;=\; 0 \,.$$

For all $\boldsymbol{y} \in U$, $i, j \in \mathbb{N}$ using the upper bound (3.52)

$$\begin{aligned} \left|\mathcal{G}\left(\frac{\partial^2 u}{\partial y_i \partial y_j}(\boldsymbol{y}) y_i y_j\right)\right| \;&\leq\; \frac{1}{4} \left\|\mathcal{G}\right\|_{V^*} \left\|\frac{\partial^2 u}{\partial y_i \partial y_j}(\boldsymbol{y})\right\|_V \\ &\leq\; \frac{1}{4} \left\|\mathcal{G}\right\|_{V^*} \overline{u_1}\, 2^{1+\epsilon} \beta_i \beta_j \;\leq\; \left\|\mathcal{G}\right\|_{V^*} \overline{u_1}\, \beta_i \beta_j \,. \end{aligned}$$

Using these and the triangle inequality gives the last result

$$\begin{aligned} \left|\mathbb{E}_{\boldsymbol{y}}\left[\mathcal{G}(u) - \mathcal{G}(u_s)\right]\right| \;&\leq\; 2 \sum_{i,j\geq s+1} \mathbb{E}_{\boldsymbol{y}}\left[\int_0^1 (1-t) \left\|\mathcal{G}\right\|_{V^*} \overline{u_1}\, \beta_i \beta_j \,\mathrm{d}t\right] \\ &\leq\; 2 \left\|\mathcal{G}\right\|_{V^*} \overline{u_1} \left(\sum_{j\geq s+1} \beta_j\right)^2 . \end{aligned}$$

This completes the proof. □

Furthermore, in [60, Theorem 5.1] it was shown that under Assumption A3.1.3 the tail of the sum over β_j is bounded above by

$$\sum_{j\geq s+1} \beta_j \;\leq\; \min\left(\frac{p}{1-p}, 1\right) \left(\sum_{j=1}^{\infty} \beta_j^p\right)^{1/p} s^{-\frac{1}{p}+1} \,. \tag{3.73}$$

3.4.2 QMC error

Given the bounds in Lemma 3.10 on the mixed derivatives of the minimal eigenpair we obtain an upper bound on the root-mean-square error of the QMC approximation of the truncated problem.

Theorem 3.13. *Let $n \in \mathbb{N}$ be prime, $\mathcal{G} \in V^*$ and suppose that Assumption A3.1 holds. Then the root-mean-square errors of the CBC-generated randomly shifted lattice rule approximations of $\mathbb{E}_{\boldsymbol{y}}[\lambda_s]$ and $\mathbb{E}_{\boldsymbol{y}}[\mathcal{G}(u_s)]$ are bounded by*

$$\sqrt{\mathbb{E}_{\boldsymbol{\Delta}}\left[\left|\mathbb{E}_{\boldsymbol{y}}[\lambda_s] - Q^{\mathrm{sh}}_{n,s}\lambda_s\right|^2\right]} \le C_{1,\alpha} n^{-\alpha}, \quad \text{and} \tag{3.74}$$

$$\sqrt{\mathbb{E}_{\boldsymbol{\Delta}}\left[\left|\mathbb{E}_{\boldsymbol{y}}[\mathcal{G}(u_s)] - Q^{\mathrm{sh}}_{n,s}\mathcal{G}(u_s)\right|^2\right]} \le C_{2,\alpha} n^{-\alpha}, \tag{3.75}$$

where

$$\alpha = \begin{cases} 1-\delta, \text{ for arbitrary } \delta \in (0, \frac{1}{2}), & \text{if } p \in (0, \frac{2}{3}], \\ \frac{1}{p} - \frac{1}{2} & \text{if } p \in (\frac{2}{3}, 1), \end{cases} \tag{3.76}$$

and the constants $C_{1,\alpha}$ and $C_{2,\alpha}$ are independent of s.

Proof. Since the estimates from Lemma 3.10 are independent of $\boldsymbol{y}$ they can be used to bound the norm (squared) of λ_s in $\mathcal{W}_{s,\boldsymbol{\gamma}}$. By (3.51) we obtain

$$\|\lambda_s\|^2_{s,\boldsymbol{\gamma}} \le \overline{\lambda_1}^2 \sum_{\mathfrak{u}\subseteq\{1:s\}} \frac{\Lambda_{\mathfrak{u}}^2}{\gamma_{\mathfrak{u}}}, \quad \Lambda_{\mathfrak{u}} \coloneqq (|\mathfrak{u}|!)^{1+\epsilon} \prod_{j\in\mathfrak{u}} \beta_j,$$

with weights $\boldsymbol{\gamma}$ and $\epsilon \in (0,1)$ as yet unspecified. Then using (3.45) the mean-square error of the lattice rule approximation is bounded above

$$\mathbb{E}_{\boldsymbol{\Delta}}\left[\left|\mathbb{E}_{\boldsymbol{y}}[\lambda_s] - Q^{\mathrm{sh}}_{n,s}\lambda_s\right|^2\right] \le C_{s,\boldsymbol{\gamma},\eta}\, \varphi(n)^{-\frac{1}{\eta}}, \tag{3.77}$$

where

$$C_{s,\boldsymbol{\gamma},\eta} \coloneqq \overline{\lambda_1}^2 \left(\sum_{\emptyset\neq\mathfrak{u}\subseteq\{1:s\}} \gamma_{\mathfrak{u}}^{\eta}\, \rho(\eta)^{|\mathfrak{u}|}\right)^{\frac{1}{\eta}} \left(\sum_{\mathfrak{u}\subseteq\{1:s\}} \frac{\Lambda_{\mathfrak{u}}^2}{\gamma_{\mathfrak{u}}}\right)$$

We now choose the weight parameters such that $C_{s,\boldsymbol{\gamma},\eta}$ can be bounded independently of s. From [60, Lemma 6.2] the choice of weights that minimises $C_{s,\boldsymbol{\gamma},\eta}$

are

$$\gamma_{\mathfrak{u}}(\eta) \;=\; \left(\frac{\Lambda_{\mathfrak{u}}^2}{\rho(\eta)^{|\mathfrak{u}|}}\right)^{\frac{1}{1+\eta}} \tag{3.78}$$

which are of POD (product and order-dependent) form. With these weights it follows that $C_{s,\boldsymbol{\gamma},\eta} \le \overline{\lambda_1}^2\, S_{s,\eta}^{(1+\eta)/\eta}$, where

$$S_{s,\eta} \coloneqq \sum_{\mathfrak{u}\subseteq\{1:s\}} \left(\Lambda_{\mathfrak{u}}^{2\eta}\rho(\eta)^{|\mathfrak{u}|}\right)^{\frac{1}{1+\eta}}\,,$$

so we must show that the sum $S_{s,\eta}$ can be bounded independently of s. Let

$$q \coloneqq \frac{2\eta(1+\epsilon)}{1+\eta} \quad\text{and}\quad \alpha_j \coloneqq \left(\rho(\eta)(\beta_j)^{2\eta}\right)^{\frac{1}{1+\eta}} \quad\text{for all } j\in\mathbb{N}\,,$$

so that

$$S_{s,\eta} \;=\; \sum_{\ell=0}^{s}(\ell!)^q \sum_{\substack{\mathfrak{u}\subseteq\{1:s\}\\ |\mathfrak{u}|=\ell}} \prod_{j\in\mathfrak{u}} \alpha_j \;\le\; \sum_{\ell=0}^{s}(\ell!)^{q-1}\left(\sum_{j=1}^{s}\alpha_j\right)^{\ell} \;<\; \infty\,,$$

which holds by the ratio test provided that $q < 1$ and $\sum_{j=1}^{\infty}\alpha_j < \infty$. Under Assumption A3.1.3, we therefore require that

$$\frac{2\eta(1+\epsilon)}{1+\eta} < 1 \iff \epsilon < \frac{1-\eta}{2\eta} \quad\text{and}\quad \frac{2\eta}{1+\eta} \le p \iff \eta \ge \frac{p}{2-p}\,.$$

To balance these conditions with the requirement that $\eta\in(\frac{1}{2},1]$, we choose a different η depending on the decay rate p and then choose $\epsilon \coloneqq (1-\eta)/(4\eta)$. Note that $\eta = 1$ has to be excluded to ensure that $\epsilon > 0$.

For $p\in(0,\frac{2}{3}]$, we have $\frac{p}{2-p} \le \frac{1}{2}$ so there is no further restriction on η and we take $\eta \coloneqq \frac{1}{2(1-\delta)}$ for arbitrary $\delta\in(0,\frac{1}{2})$. However, for $p\in(\frac{2}{3},1)$ the value of η is restricted and we take it as small as possible, namely, $\eta \coloneqq \frac{p}{2-p}$. Substituting these choices of η into (3.77) and taking n to be prime (for simplicity) yields the result (3.74).

The error bound (3.75) follows in the same way, after observing that the norm of $\mathcal{G}(u_s)$ can be bounded using (3.52)

$$\begin{aligned}\|\mathcal{G}(u_s)\|_{s,\boldsymbol{\gamma}}^2 \;&\le\; \sum_{\mathfrak{u}\subseteq\{1:s\}} \frac{1}{\gamma_{\mathfrak{u}}} \int_{[0,1]^{|\mathfrak{u}|}} \left(\int_{[0,1]^{s-|\mathfrak{u}|}} \|\mathcal{G}\|_{V^*} \left\|\frac{\partial^{|\mathfrak{u}|}u_s}{\partial \boldsymbol{y}_{\mathfrak{u}}}(\cdot,\boldsymbol{y}_s)\right\|_V \mathrm{d}\boldsymbol{y}_{-\mathfrak{u}}\right)^2 \mathrm{d}\boldsymbol{y}_{\mathfrak{u}} \\ &\le\; \|\mathcal{G}\|_{V^*}^2\, \overline{u_1}^2 \sum_{\mathfrak{u}\subseteq\{1:s\}} \frac{\Lambda_{\mathfrak{u}}^2}{\gamma_{\mathfrak{u}}}\,.\end{aligned}$$

This completes the proof. □

Remark 3.14. The main ingredients in this proof are the bounds on the derivatives of the eigenvalue, which are needed to show that $\lambda_s \in \mathcal{W}_{s,\boldsymbol{\gamma}}$. As was stated in Remark 3.11, these bounds also hold for the derivatives of $\lambda_{s,h}$. Thus, for h sufficiently small (see (3.48)) the same error bound holds for the QMC error of the FE error approximation $\lambda_{s,h}$, but with the constants possibly depending on h.

3.4.3 Total error

Using the triangle inequality, the mean-square error of the combined truncation-FE-QMC approximation of the expected value of λ_1 can be bounded above by

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\Delta}}\left[\left|\mathbb{E}_{\boldsymbol{y}}[\lambda] - Q^{\mathrm{sh}}_{n,s}\lambda_{s,h}\right|^2\right] \le C\Big(&\mathbb{E}_{\boldsymbol{\Delta}}\left[\left|\mathbb{E}_{\boldsymbol{y}}[\lambda - \lambda_s]\right|^2\right] \\ &+ \mathbb{E}_{\boldsymbol{\Delta}}\left[\left|\mathbb{E}_{\boldsymbol{y}}[\lambda_s] - Q^{\mathrm{sh}}_{n,s}\lambda_s\right|^2\right] + \mathbb{E}_{\boldsymbol{\Delta}}\left[\left|Q^{\mathrm{sh}}_{n,s}(\lambda_s - \lambda_{s,h})\right|^2\right]\Big),\end{aligned} \tag{3.79}$$

for $C > 0$. Here we have conveniently split the total error into three separate errors: one each for the truncation, QMC and FE errors, respectively. Note that there are different ways of splitting the total error, but we have chosen the above technique because now the second term is the QMC error for the actual eigenvalue and not the FE approximation $\lambda_{s,h}$. This is important because it means that we do not need specific bounds on the parametric regularity of the FE eigenvalue.

The terms in the upper bound on the mean-square error (3.79) can be bounded using (3.71), (3.74) and (3.41), respectively, leading to the following theorem. A similar splitting argument using (3.70), (3.42) and (3.75) instead gives a bound on the error of the approximation for the corresponding eigenfunction.

Theorem 3.15. *Let Assumption A3.1 hold with $p \in (0,1)$, $h > 0$ be sufficiently small, $s \in \mathbb{N}$, prime $n \in \mathbb{N}$ and let $\boldsymbol{z} \in \mathbb{N}^s$ be a generating vector constructed using the CBC algorithm with weights given by* (3.78)*. Then the root-mean-square error—with respect to the random shift $\boldsymbol{\Delta} \in [0,1]^s$—of our truncation-FE-QMC approximation of the expectation of the minimal eigenvalue λ is bounded by*

$$\sqrt{\mathbb{E}_{\boldsymbol{\Delta}}\left[\left|\mathbb{E}_{\boldsymbol{y}}[\lambda] - Q^{\mathrm{sh}}_{n,s}\lambda_{s,h}\right|^2\right]} \le C_1\left(h^2 + s^{-2(\frac{1}{p}-1)} + n^{-\alpha}\right). \tag{3.80}$$

For any functional $\mathcal{G} \in H^{-1+t}(D)$ of the corresponding eigenfunction u, with $t \in [0,1]$, the truncation-FE-QMC approximation of its expected value is bounded by

$$\sqrt{\mathbb{E}_{\boldsymbol{\Delta}}\left[\left|\mathbb{E}_{\boldsymbol{y}}[\mathcal{G}(u)] - Q^{\mathrm{sh}}_{n,s}\mathcal{G}(u_{s,h})\right|^2\right]} \le C_2\left(h^{1+t} + s^{-2(\frac{1}{p}-1)} + n^{-\alpha}\right), \tag{3.81}$$

where

$$\alpha = \begin{cases} 1-\delta, \text{ for arbitrary } \delta \in (0, \frac{1}{2}), & \text{if } p \in (0, \frac{2}{3}], \\ \frac{1}{p} - \frac{1}{2} & \text{if } p \in (\frac{2}{3}, 1). \end{cases}$$

Comparing this with the corresponding result for the elliptic source problem [60, Theorem 8.1], observe that we obtain the exact same rates of convergence in s and n for all $p \in (0, 1)$. The only exception is that our results do not hold when $p = 1$, whereas [60, Theorem 8.1] presents a result when $p = 1$. However, for that case they do require an additional assumption. To compare the two FE convergence rates recall that the number of degrees of freedom in the FE grid is $M_h = \mathcal{O}(h^{-d})$. Letting $\mathcal{G} \in H^{-1+t}(D)$, note that the correct comparison is the case when the source term belongs to $L^2(D)$, so that in [60, Theorem 8.1] $\tau = 1 + t$. In this case the FE error convergence rate from [60, Theorem 8.1] for the linear functional $\mathcal{G}$ of the solution to the source problem is $M_h^{-\tau/d} = \mathcal{O}(h^{1+t})$, which is exactly the rate in (3.81) for eigenfunction.

3.5 Numerical results

Now we present numerical results on the performance of our truncation-FE-QMC algorithm in approximating the expected value of the smallest eigenvalue of an eigenvalue problem of the form (3.1). In our example, the stochastic coefficients a and b are composed of scaled trigonometric functions, with the purpose being to imitate the behaviour of a Karhunen-Loève expansion of a random field. We are interested in whether the quadrature component of the error matches the theoretical estimate (3.80) in Theorem 3.15, and so we will study different scalings of the basis functions a_j, b_j in the coefficients, which will correspond to different values of the decay parameter p.

As detailed in Section 2.3, to estimate the quadrature error, instead of performing a single randomly shifted approximation $Q_{n,s}^{\mathrm{sh}}\lambda_{s,h}$ we conduct a small number R of different approximations based on independently and identically distributed random shifts, then the final approximation is taken to be the average over the R independent approximations as in (2.13). In this way, we obtain an unbiased estimate of the integral (of $\lambda_{s,h}$) and the sample variance over the different shifts gives an estimate of the quadrature component of the mean-square error using (2.14).

The purpose of this chapter is to study the application of QMC methods to stochastic eigenvalue problems, thus in the numerical experiments below we fix the

truncation dimension at $s = 100$ and the FE meshwidth at $h = 1/256$ while varying the number of QMC points n. Also, we set the number of random shifts to be $R = 8$.

The smallest eigenvalue of each FE system is approximated using the `eigs` function in Matlab, which in turn runs the implicitly restarted Arnoldi method using the ARPACK library. We set the tolerance for the accuracy of this eigensolver to be 10^{-14}, so as to ensure that the numerical errors incurred by approximating the FE eigenvalues are negligible compared to the integration error.

In practice, we cannot compute the optimal function space weights $\gamma_{\mathfrak{u}}$ according to the formula (3.78) from the proof of Theorem 3.13, because (3.78) depends on the sequence $\boldsymbol{\beta}$, and $C_{\boldsymbol{\beta}}$ (3.54) contains factors that cannot be computed explicitly. As such, in our numerical experiments for $\mathfrak{u} \subset \mathbb{N}$ we set the function space weight $\gamma_{\mathfrak{u}}$ to be

$$\gamma_{\mathfrak{u}} \;=\; |\mathfrak{u}|! \prod_{j \in \mathfrak{u}} \left(\max \left\{ \|a_j\|_{L^\infty(D)} \,, \|b_j\|_{L^\infty(D)} \right\} \right)^{\eta} , \tag{3.82}$$

where $\eta = 4/3$ if $p \in (0, 2/3]$ and $\eta = 2 - p$ if $p \in (2/3, 1)$. Note that η in (3.82) is not the same as in (3.78).

For our numerical experiments we consider an eigenvalue problem (3.1) on the domain $D = (0,1)^2$ where the only non-trivial coefficient is $a(\boldsymbol{y})$ in the second-order term. Explicitly, the coefficient $a(\boldsymbol{y})$ is given as in (3.4) (with a_j defined below) but $b(\boldsymbol{y}) \equiv 0$ and $c \equiv 1$. For some decay $q \geq 4/3$, the basis functions for the coefficient $a(\boldsymbol{y})$ are defined to be

$$a_0 \equiv 2\,, \quad a_j(\boldsymbol{x}) \;=\; \frac{1}{1 + (\pi j)^q} \sin(j \pi x_1) \sin((j-1)\pi x_2)\,, \quad \text{for } \boldsymbol{x} = (x_1, x_2) \in (0,1)^2\,.$$

Clearly, for all $j \in \mathbb{N}$ we have that

$$\|a_j\|_{L^\infty(D)} \;=\; \frac{1}{1 + (\pi j)^q} \;<\; \frac{1}{j^q}\,,$$

and hence $\sum_{j=1}^{\infty} \|a_j\|_{L^\infty(D)} < \zeta(q)$. It follows that the coefficient is bounded above and below as required with

$$a_{\min} \;=\; 2 - \frac{\zeta(q)}{2} \quad \text{and} \quad a_{\max} \;=\; 2 + \frac{\zeta(q)}{2}\,.$$

Similarly, the parameter q determines the rate of decay of the norms of the basis functions, and in turn we can take p in Assumption A3.1.3 to satisfy $p \in (1/q, 1)$.

In our numerical experiments for this problem we consider $q = 4/3, 2, 3$ and the number of quadrature points given by $n = 251$, 503, 997, 199, 4001, 8009, 16001.

The RMS quadrature error is estimated by the square root of (2.14) and the rate of convergence is estimated using a least-squares fit of the RMS error estimates.

Regarding the convergence rates to expect, for $q = 4/3$ we have that the sequence $\boldsymbol{\beta}$ is p-summable for $p > 3/4$, whereas for the faster decays of $q = 2, 3$ we have that $\boldsymbol{\beta}$ is summable with exponent $p > 1/2$ and $p > 1/3$, respectively. Based on these restrictions on p, for each n we construct a generating vector by the CBC algorithm using weights given by (3.82) with $\eta = 2 - 1/q = 5/4$ for $q = 4/3$, and $\eta = 4/3$ for $q = 2, 3$. Hence from Theorem 3.15, for $q = 4/3$, because $p > 2/3$, we are in the regime where the convergence is limited and so expect a rate of around $-5/6 \approx -0.83$. However, for $q = 2, 3$ the rate is not restricted and we expect convergence close to $1/n$.

Figure 3.1 plots the estimated RMS quadrature error of our QMC approximation along with a Monte Carlo (MC) approximation for comparison, for $q = 4/3$ (left) and $q = 2$ (right). To ensure that the MC and QMC approximations use the same number of points, each MC approximation used $R \times n$ points. In Figure 3.1 the circle data points (triangles for MC) represent the estimated RMS errors, the dashed lines portray the expected convergence rates and the solid lines are a least-squares fit. Note that the axes of the two graphs are equal. Observe that the least-squares fit matches very closely to the expected rates. In fact, for the QMC error the least-squares computed rates are -0.826, -0.997 and -1.019 for $q = 4/3, 2, 3$, respectively, which are very close to the expected rates of $-5/6 \approx -0.83$, -1, and -1 from the theory. Also, as is expected for a problem of this smoothness QMC significantly outperforms the MC approximations, which decay at the anticipated rate of $1/\sqrt{n}$. For completeness, the computed values of the QMC RMS error estimates and the convergence rates for $q = 4/3, 2$ and 3 are given below in Table 3.1. We use the notation "e" to denote the base 10 exponent.

		RMS error estimate		
n	$R \times n$	$q = 4/3$ ($p \approx 3/4$)	$q = 2$ ($p \approx 1/2$)	$q = 3$ ($p \approx 1/3$)
251	2008	4.6 e−6	4.9 e−7	2.5 e−8
503	4024	1.8 e−6	2.3 e−7	1.0 e−8
997	7976	1.0 e−6	1.0 e−7	3.6 e−9
1999	15992	3.5 e−7	5.6 e−8	2.1 e−9
4001	32008	4.8 e−7	2.5 e−8	1.3 e−9
8009	64072	2.9 e−7	1.5 e−8	6.5 e−10
16001	128008	9.4 e−8	7.5 e−9	3.0 e−10
Estimated rate		−0.826	−0.997	−1.019

Table 3.1: QMC RMS error estimates for $q = 4/3, 2, 3$.

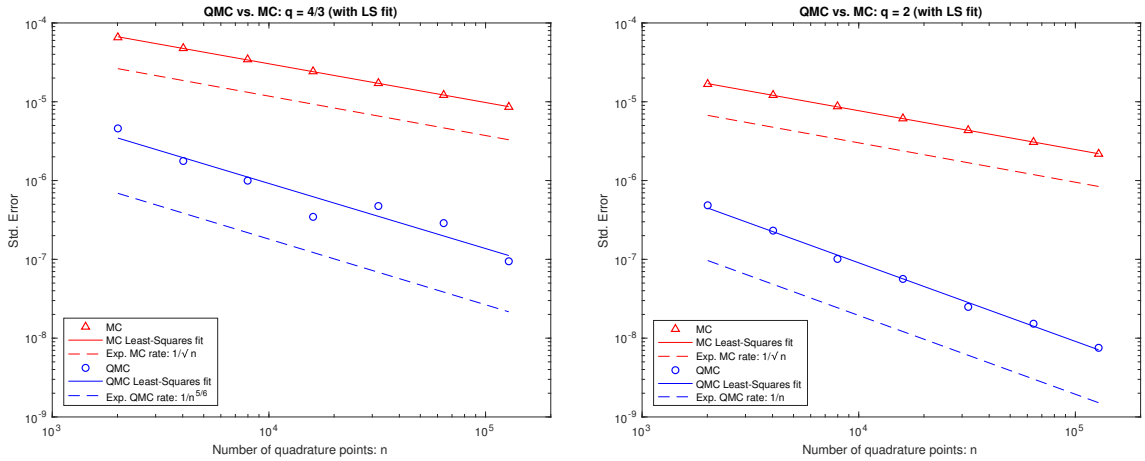


Figure 3.1: QMC and MC convergence for $q = 4/3$, $p \approx 3/4$ (left) and $q = 2$, $p \approx 1/2$ (right).

3.6 Conclusion

In this chapter we developed and analysed an algorithm that uses QMC rules to approximate the expectation of the eigenvalue of an elliptic eigenvalue problem with coefficients that depend on infinitely-many stochastic parameters. The algorithm, which can also be applied to linear functionals of the corresponding eigenfunction, performs an approximation by first truncating the stochastic domain to finitely-many parameters and discretising the spatial domain using FE methods. The expected value of the truncated-FE eigenvalue is then approximated using a randomly shifted lattice rule.

The main theoretical results presented were a rigorous analysis of the total approximation error and explicit bounds on the mixed derivatives (with respect to the stochastic parameters) of the smallest eigenvalue and the corresponding eigenfunction. Actually, the bounds on the derivatives were instrumental to analysing the error. In particular, they allowed us to bound the truncation error independently of the parameters, and then bound the QMC error independently of the truncation dimension. As with the source problem the convergence rate of the total error depends on the decay parameter p, and in almost all cases the convergence rate of the error was exactly the same as the corresponding rate in the error bounds for the source problem. The only exclusion was that our results do not hold when $p = 1$, in which case the result for the source problem also requires an additional assumption.

Finally, we applied our algorithm to an eigenvalue problem in two spatial dimensions for coefficients with different decays p, and presented results on the QMC component of the error. In all cases, the quadrature errors converged at the rate predicted by our theory, and decayed noticeably faster than a MC approximation.

Original research and my contribution

The work presented in this chapter was conducted jointly with Ivan Graham and Rob Scheichl (both at the University of Bath) along with my supervisors Frances Kuo and Ian Sloan.

Approximation of parametric, elliptic eigenvalue problems has previously been studied in [1] using sparse tensor products, and in that paper the authors proved that simple eigenvalues (along with the corresponding eigenfunctions) are analytic with respect to the stochastic parameters. The method of using truncation, finite elements and QMC to approximate the expected value has also been used previously for the source problem in, e.g., [60]. However, the application of QMC rules to stochastic eigenvalue problems is original, and so too is the error analysis. Loosely speaking, Section 3.2.2 and then everything from Section 3.3 onwards is original work. Auxiliary to our error analysis, we also provide an explicit proof that the spectral gap is bounded independently of $\boldsymbol{y}$ (see Section 3.2.2) prove explicit bounds on the derivatives of the smallest eigenvalue and eigenfunction (see Lemma 3.10).

Motivated by an interest in uncertainty quantification of the criticality problem for nuclear reactors, our colleagues Ivan Graham and Rob Scheichl had the idea to apply QMC to stochastic eigenvalue problems. After it was decided that we should study self-adjoint, elliptic eigenvalue problems, I took the lead responsibility for the error analysis, with corrections and input from Frances, Ian, Ivan and Rob. However, I should mention that much of the work on bounding the spectral gap was due to Ivan Graham. I also implemented the truncation-FE-QMC approximation algorithm and performed the numerical results.

3.A Proof of Theorem 3.6

In this appendix we follow [86] to bound the FE error, tracking the dependence of all constants on $\boldsymbol{y}$. We have opted for the classical min-max argument as opposed to the Babuška-Osborn theory [5, 6, 7] because it is more elementary and allows us to determine the influence of the constants. We begin with some preliminary definitions.

For $\boldsymbol{y} \in U$ and $h > 0$, define the orthogonal projection $P_h : V \to V_h$ of $u \in V$ with respect to the inner product $\mathcal{B}(\boldsymbol{y}; \cdot, \cdot)$ by

$$\mathcal{B}(\boldsymbol{y}; u - P_h u, v_h) \;=\; 0 \quad \text{for all } v_h \in V_h \,.$$

Although P_h depends on $\boldsymbol{y}$ through the bilinear form $\mathcal{B}(\boldsymbol{y}; \cdot, \cdot)$ we have suppressed this $\boldsymbol{y}$ dependence. Letting $\|\cdot\|_{\mathcal{B}(\boldsymbol{y})} = \sqrt{\mathcal{B}(\boldsymbol{y}; \cdot, \cdot)}$, due to $\mathcal{B}$-orthogonality the projection satisfies

$$\|u - P_h u\|_{\mathcal{B}(\boldsymbol{y})} \;=\; \inf_{v_h \in V_h} \|u - v_h\|_{\mathcal{B}(\boldsymbol{y})} \,.$$

Due to (3.13) and (3.14) the energy norm is equivalent to the V-norm:

$$\sqrt{a_{\min}}\, \|v\|_V \;\le\; \|v\|_{\mathcal{B}(\boldsymbol{y})} \;\le\; \sqrt{a_{\max}(1 + C_D^2)}\, \|v\|_V \quad \text{for all } v \in V. \tag{3.83}$$

Analogously to the min-max principle (3.19), when the k-dimensional subspaces S_k are restricted to V_h we have the min-max representation for the FE eigenvalues

$$\lambda_{k,h}(\boldsymbol{y}) \;=\; \min_{\substack{S_k \subset V_h \\ \dim(S_k) = k}} \max_{u_h \in S_k} \frac{\mathcal{B}(\boldsymbol{y}; u_h, u_h)}{\mathcal{D}(u_h, u_h)} \,. \tag{3.84}$$

The strategy of the proof is to first bound the difference between $u(\boldsymbol{y})$ and its projection $P_h u(\boldsymbol{y})$, which is fairly straightforward and follows from the FE results for elliptic source problems. The difficulty lies in the fact that the projections $P_h u(\boldsymbol{y})$ are not the same as the FE eigenfunctions $u_h(\boldsymbol{y})$. However, they are close. The next stage of the proof is to bound the eigenvalue and eigenfunction error in terms of the projection error. For the eigenvalue error a key ingredient is the classical min-max principle. Finally, combining the FE error bounds with the projection error bounds yields the required results.

Lemma 3.16. *Let $\boldsymbol{y} \in U$. The projection of $u_1(\boldsymbol{y}) \in E(\boldsymbol{y}, \lambda_1(\boldsymbol{y})) \subset V$ into V_h satisfies*

$$\|u_1(\boldsymbol{y}) - P_h u_1(\boldsymbol{y})\|_V \;\le\; Ch \,, \tag{3.85}$$

where $C > 0$ is independent of $\boldsymbol{y}$.

Proof. The projection $P_h u_1(\boldsymbol{y})$ can be equivalently viewed as the solution to the FE approximation of an elliptic source problem. Indeed, the variational eigenproblem (3.12) for the eigenpair $(\lambda_1(\boldsymbol{y}), u_1(\boldsymbol{y}))$ can be written as

$$\mathcal{B}(\boldsymbol{y}; u_1(\boldsymbol{y}), v) \;=\; \langle f(\boldsymbol{y}), v\rangle \quad \text{for all } v \in V\,,$$

where $f(\boldsymbol{y}) = \lambda_1(\boldsymbol{y}) c \cdot u_1(\boldsymbol{y})$ is now assumed fixed. The FE approximation problem is then to find $\widetilde{u}_h(\boldsymbol{y}) \in V_h$ such that

$$\mathcal{B}(\boldsymbol{y}; \widetilde{u}_h(\boldsymbol{y}), v_h) \;=\; \langle f(\boldsymbol{y}), v_h\rangle \quad \text{for all } v_h \in V_h\,,$$

for which, due to $\mathcal{B}$-orthogonality, the solution is exactly the projection of the eigenfunction: $\widetilde{u}_h(\boldsymbol{y}) = P_h u_1(\boldsymbol{y})$. This allows us to bound the projection error using the results from elliptic source problems. In particular, our differential operator fits the setting of affine parametric operator equations from [17]. Since $u_1(\boldsymbol{y}) \in Z$, it follows that $f(\boldsymbol{y}) \in L^2(D)$ for all $\boldsymbol{y}$. The spaces V_h satisfy the approximation property (3.39), thus by Theorem 2.4 in [17] we have

$$\|u_1(\boldsymbol{y}) - P_h u_1(\boldsymbol{y})\|_V \;\le\; C' \,\|f(\boldsymbol{y})\|_{L^2(D)}\, h\,, \tag{3.86}$$

with constant C' independent of $\boldsymbol{y}$ and h.

To bound $\|f(\boldsymbol{y})\|_{L^2(D)}$, we use the upper bound in (3.20), the bound (3.7) on c, and then the fact that $u_1(\boldsymbol{y})$ is normalised to give

$$\|f(\boldsymbol{y})\|_{L^2(D)} \;\le\; \lambda_1(\boldsymbol{y}) \sqrt{\|c\|_{L^\infty(D)}}\, \|u_1(\boldsymbol{y})\|_{L^2(D)} \;\le\; \frac{a_{\max}^{3/2}}{a_{\min}} (\chi_1 + 1)\,.$$

Substituting this into (3.86) we have our desired result with a constant independent of $\boldsymbol{y}$ and h. □

The lemmas that follow relate the eigenvalue error (Lemma 3.17) and the eigenfunction error (Lemma 3.19) to the projection errors that we have just bounded. Lemma 3.18 in the middle relates to the separation between the FE eigenvalues and $\lambda_1(\boldsymbol{y})$, it is used in the proof of Lemma 3.19.

Lemma 3.17. *Let $\boldsymbol{y} \in U$ and let $h > 0$ be sufficiently small. Then*

$$|\lambda_1(\boldsymbol{y}) - \lambda_{1,h}(\boldsymbol{y})| \;\le\; C\, \|u_1(\boldsymbol{y}) - P_h u_1(\boldsymbol{y})\|_V^2\,, \tag{3.87}$$

where $C > 0$ is independent of $\boldsymbol{y}$.

Proof. To prove the result we apply the min-max principle to $\lambda_{1,h}(\boldsymbol{y})$, where to obtain an upper bound, in the right hand side of (3.84) we take the particular subspace $S_{1,h}(\boldsymbol{y}) := P_h E(\boldsymbol{y}, \lambda_1(\boldsymbol{y}))$. The argument requires that $S_{1,h}(\boldsymbol{y})$ be a 1-dimensional subspace of V. Clearly, since $E(\boldsymbol{y}, \lambda_1(\boldsymbol{y}))$ is 1-dimensional it follows that $\dim(S_{1,h}(\boldsymbol{y})) \leq 1$. Suppose for a contradiction that $P_h u_1(\boldsymbol{y}) = 0$, then by the $\mathcal{B}$-orthogonality of P_h

$$1 \;=\; \|u_1(\boldsymbol{y})\|_{\mathcal{D}} = \frac{1}{\lambda_1(\boldsymbol{y})} \|u_1(\boldsymbol{y})\|_{\mathcal{B}(\boldsymbol{y})} \;=\; \frac{1}{\lambda(\boldsymbol{y})} \|u_1(\boldsymbol{y}) - P_h u_1(\boldsymbol{y})\|_{\mathcal{B}(\boldsymbol{y})} \,.$$

Then using the lower bound in (3.20) and the equivalence of norms (3.83) we have that

$$1 \;\leq\; \underbrace{\frac{a_{\max}^2(1+\chi_1)}{a_{\min}\chi_1}}_{C'} \|u_1(\boldsymbol{y}) - P_h u_1(\boldsymbol{y})\|_V^2 \;\leq\; C' C^2 h^2 \tag{3.88}$$

where in the last step we have used Lemma 3.16 with constant $C > 0$, which is independent of h and $\boldsymbol{y}$. For h sufficiently small this yields a contradiction, and so $\dim(S_{1,h}(\boldsymbol{y})) = 1$. Therefore, choosing $S_{1,h}(\boldsymbol{y})$ in the min-max principle (3.84) gives the inequality

$$\lambda_{1,h}(\boldsymbol{y}) \;\leq\; \max_{0 \neq v_h \in S_{1,h}(\boldsymbol{y})} \frac{\mathcal{B}(\boldsymbol{y}; v_h, v_h)}{\mathcal{D}(v_h, v_h)} \;=\; \frac{\mathcal{B}(\boldsymbol{y}; P_h u_1(\boldsymbol{y}), P_h u_1(\boldsymbol{y}))}{\mathcal{D}(P_h u_1(\boldsymbol{y}), P_h u_1(\boldsymbol{y}))} \,. \tag{3.89}$$

Using the fact that the norm of the projection is bounded by 1, the numerator is bounded by

$$\mathcal{B}(\boldsymbol{y}; P_h u_1(\boldsymbol{y}), P_h u_1(\boldsymbol{y})) \;\leq\; \mathcal{B}(\boldsymbol{y}; u_1(\boldsymbol{y}), u_1(\boldsymbol{y})) \;=\; \lambda_1(\boldsymbol{y}) \,, \tag{3.90}$$

where for the equality in the last step we have used the representation (3.21).

Expanding the denominator gives

$$\begin{aligned} &\mathcal{D}(P_h u_1(\boldsymbol{y}), P_h u_1(\boldsymbol{y})) \\ &\qquad = \|u_1(\boldsymbol{y})\|_{\mathcal{D}}^2 - 2\mathcal{D}(u_1(\boldsymbol{y}), u_1(\boldsymbol{y}) - P_h u_1(\boldsymbol{y})) + \|u_1(\boldsymbol{y}) - P_h u_1(\boldsymbol{y})\|_{\mathcal{D}}^2 \,. \end{aligned}$$

The first term on the right is 1 since $u_1(\boldsymbol{y})$ is normalised and the last term is positive, so we can bound $\mathcal{D}(P_h u_1(\boldsymbol{y}), P_h u_1(\boldsymbol{y}))$ by

$$\mathcal{D}(P_h u_1(\boldsymbol{y}), P_h u_1(\boldsymbol{y})) \;\geq\; 1 - 2\mathcal{D}(u_1(\boldsymbol{y}), u_1(\boldsymbol{y}) - P_h u_1(\boldsymbol{y})) \,.$$

Then using the fact that $u_1(\boldsymbol{y})$ is an eigenfunction satisfying (3.12) and $\mathcal{B}$-orthogonality of the projection P_h we have

$$\begin{aligned}\mathcal{D}(P_h u_1(\boldsymbol{y}), P_h u_1(\boldsymbol{y})) &\geq 1 - \frac{2}{\lambda_1(\boldsymbol{y})}\mathcal{B}(\boldsymbol{y}; u_1(\boldsymbol{y}) - P_h u_1(\boldsymbol{y}), u_1(\boldsymbol{y}) - P_h u_1(\boldsymbol{y})) \\ &\geq 1 - 2C' \left\| u_1(\boldsymbol{y}) - P_h u_1(\boldsymbol{y}) \right\|_V , \end{aligned} \tag{3.91}$$

with C' as in (3.88), where in the last step we have used the lower bound in (3.20) and the equivalence of norms (3.83).

For h sufficiently small the lower bound on the denominator in (3.91) is positive, and substituting it along within the upper bound on the numerator (3.90) into (3.89) then rearranging gives

$$|\lambda_1(\boldsymbol{y}) - \lambda_{1,h}(\boldsymbol{y})| \leq 2C' \lambda_{1,h}(\boldsymbol{y}) \left\| u_1(\boldsymbol{y}) - P_h u_1(\boldsymbol{y}) \right\|_V .$$

Now all that remains is to show that $\lambda_{1,h}(\boldsymbol{y})$ can be bounded from above independently of $\boldsymbol{y}$ and h. Analogously to (3.20), using the FE min-max representation (3.84) we have

$$\lambda_{1,h}(\boldsymbol{y}) \leq \frac{a_{\max}}{a_{\min}}(\chi_{1,h} + 1) ,$$

where $\chi_{1,h}$ the smallest eigenvalue corresponding to the negative Laplacian on D with boundary conditions (3.3), discretised in the FE space V_h. It is well-known, see, e.g. [8, Theorem 10.4], that in the current setting $\chi_{k,h} \leq \chi_1 + C''h^2$ with $C'' > 0$ independent of h. Thus for h sufficiently small there exists a constant such that $\lambda_{1,h}(\boldsymbol{y})$ can be bounded independent of $\boldsymbol{y}$ and h as required. □

Lemma 3.18. *Let $\boldsymbol{y} \in U$ and $h > 0$ be sufficiently small. Then for all $k = 2, 3, \ldots, M_h = \dim(V_h)$*

$$\frac{\lambda_1(\boldsymbol{y})}{\lambda_{k,h}(\boldsymbol{y}) - \lambda_1(\boldsymbol{y})} \leq \rho , \tag{3.92}$$

where $\rho > 0$ is independent of $\boldsymbol{y}$ and h.

Proof. We show that

$$\lambda_{2,h}(\boldsymbol{y}) - \lambda_1(\boldsymbol{y}) \geq \frac{1}{\rho}\lambda_1(\boldsymbol{y}) ,$$

which is equivalent since the left hand side of (3.92) attains its maximum when $k = 2$.

Since the eigenvalues converge from above, it follows from Proposition 3.4 and the upper bound on $\lambda_1(\boldsymbol{y})$ in (3.20) that

$$\lambda_{2,h}(\boldsymbol{y}) - \lambda_1(\boldsymbol{y}) \;\geq\; \lambda_2(\boldsymbol{y}) - \lambda_1(\boldsymbol{y}) \;\geq\; \delta \;\geq\; \frac{\delta a_{\min}}{a_{\max}(\chi_1 + 1)} \lambda_1(\boldsymbol{y}) \,/$$

□

Lemma 3.19. *Let $\boldsymbol{y} \in U$ and $h > 0$ be sufficiently small. Then*

$$\|u_1(\boldsymbol{y}) - u_{1,h}(\boldsymbol{y})\|_{\mathcal{D}} \leq C \, \|u_1(\boldsymbol{y}) - P_h u_1(\boldsymbol{y})\|_{\mathcal{D}} \,, \tag{3.93}$$

where C is independent of $\boldsymbol{y}$ and h.

Proof. The FE eigenfunctions form an orthonormal basis for V_h with respect to $\mathcal{D}(\cdot, \cdot)$, and so the projection of $u_1(\boldsymbol{y})$ can be written as

$$P_h u_1(\boldsymbol{y}) \;=\; \sum_{k=1}^{M_h} \alpha_{k,h}(\boldsymbol{y}) u_{k,h}(\boldsymbol{y}) \,,$$

where $\alpha_{k,h}(\boldsymbol{y}) \coloneqq \mathcal{D}\left(P_h u_1(\boldsymbol{y}), u_{k,h}(\boldsymbol{y})\right)$.

The key coefficient in this expansion is $\alpha_{1,h}(\boldsymbol{y})$—if we assume that $\alpha_{1,h}(\boldsymbol{y}) \geq 0$ (which we can always ensure by controlling the sign of $u_{1,h}(\boldsymbol{y})$), then the size of $\alpha_{1,h}(\boldsymbol{y})$ gives a measure of how close $P_h u_1(\boldsymbol{y})$ is to $u_{1,h}(\boldsymbol{y})$. As a first step towards (3.93), consider the difference

$$\begin{aligned} &\|u_1(\boldsymbol{y}) - \alpha_{1,h}(\boldsymbol{y}) u_{1,h}(\boldsymbol{y})\|_{\mathcal{D}} \\ &\qquad \leq\; \|u_1(\boldsymbol{y}) - P_h u_1(\boldsymbol{y})\|_{\mathcal{D}} + \|P_h u_1(\boldsymbol{y}) - \alpha_{1,h}(\boldsymbol{y}) u_{1,h}(\boldsymbol{y})\|_{\mathcal{D}} \,. \end{aligned} \tag{3.94}$$

The first term is exactly our target upper bound. The square of the second term can be written as

$$\|P_h u_1(\boldsymbol{y}) - \alpha_{1,h}(\boldsymbol{y}) u_{1,h}(\boldsymbol{y})\|_{\mathcal{D}}^2 \;=\; \sum_{k=2}^{M_h} \alpha_{k,h}(\boldsymbol{y})^2 \,.$$

By [86, Lemma 6.4] (or as is easily verified) we can replace $\alpha_{k,h}(\boldsymbol{y})$ by

$$\alpha_{k,h}(\boldsymbol{y}) \;=\; \frac{\lambda_1(\boldsymbol{y})}{\lambda_{k,h}(\boldsymbol{y}) - \lambda_1(\boldsymbol{y})} \mathcal{D}\left(u_1(\boldsymbol{y}) - P_h u_1(\boldsymbol{y}), u_{k,h}(\boldsymbol{y})\right) \,.$$

Then using Lemma 3.18 and letting Q_h denote the $\mathcal{D}$-orthogonal projection we have the upper bound

$$\begin{aligned}\|P_h u_1(\boldsymbol{y}) - \alpha_{1,h}(\boldsymbol{y}) u_{1,h}(\boldsymbol{y})\|_{\mathcal{D}}^2 &\leq \sum_{k=2}^{M_h} \rho^2 \mathcal{D}\left(u_1(\boldsymbol{y}) - P_h u_1(\boldsymbol{y}), u_{k,h}(\boldsymbol{y})\right)^2 \\ &= \rho^2 \sum_{k=2}^{M_h} \mathcal{D}\left(Q_h\left(u_1(\boldsymbol{y}) - P_h u_1(\boldsymbol{y})\right)\right), u_{k,h}(\boldsymbol{y}))^2 \\ &\leq \rho^2 \|u_1(\boldsymbol{y}) - P_h u_1(\boldsymbol{y})\|_{\mathcal{D}}^2 .\end{aligned}$$

Thus, our intermediate bound (3.94) can be written as

$$\|u_1(\boldsymbol{y}) - \alpha_{1,h}(\boldsymbol{y}) u_{1,h}(\boldsymbol{y})\|_{\mathcal{D}} \leq (1+\rho) \|u_1(\boldsymbol{y}) - P_h u_1(\boldsymbol{y})\|_{\mathcal{D}} . \tag{3.95}$$

The final step is to show that $\alpha_{1,h}(\boldsymbol{y})$ is close to 1 and that the FE eigenfunction error can be bounded in terms of our intermediate bound (3.95). To that end, by the reverse triangle inequality together with the fact that both $u_1(\boldsymbol{y})$ and $u_{1,h}(\boldsymbol{y})$ are normalised we have

$$\begin{aligned}\|u_1(\boldsymbol{y}) - \alpha_{1,h}(\boldsymbol{y}) u_{1,h}(\boldsymbol{y})\|_{\mathcal{D}} &\geq \left| \|u_1(\boldsymbol{y})\|_{\mathcal{D}} - \alpha_{1,h}(\boldsymbol{y}) \|u_{1,h}(\boldsymbol{y})\|_{\mathcal{D}} \right| \\ &\geq |1 - \alpha_{1,h}(\boldsymbol{y})| . \end{aligned} \tag{3.96}$$

Finally, by the triangle inequality and then using our intermediate bound (3.95) along with (3.96) we have

$$\begin{aligned}\|u_1(\boldsymbol{y}) - u_{1,h}(\boldsymbol{y})\|_{\mathcal{D}} &\leq \|u_1(\boldsymbol{y}) - \alpha_{1,h}(\boldsymbol{y}) u_1(\boldsymbol{y})\|_{\mathcal{D}} + |1 - \alpha_{1,h}(\boldsymbol{y})| \, \|u_{1,h}(\boldsymbol{y})\|_{\mathcal{D}} \\ &\leq 2(1+\rho) \|u_1(\boldsymbol{y}) - P_h u_1(\boldsymbol{y})\|_{\mathcal{D}} ,\end{aligned}$$

□

We now have all of the ingredients needed to prove our FE error bounds.

Proof of Theorem 3.6. The eigenvalue error (3.41) follows by the chain of inequalities (3.87) from Lemma 3.17 then (3.85) from Lemma 3.16. All of the constants involved are independent of $\boldsymbol{y}$ and h, so the final constant is also. Inside this proof we label this constant as C_1.

For the eigenfunction error, we use Lemma 3.1 from [6] to write

$$\|u_1(\boldsymbol{y}) - u_{1,h}(\boldsymbol{y})\|_{\mathcal{B}(\boldsymbol{y})}^2 = \lambda_{1,h}(\boldsymbol{y}) - \lambda_1(\boldsymbol{y}) + \lambda_1(\boldsymbol{y}) \|u_1(\boldsymbol{y}) - u_{1,h}(\boldsymbol{y})\|_{\mathcal{D}}^2 .$$

Using the equivalence of norms (3.83) on the left hand side, taking the absolute value then the triangle inequality on the right and using the upper bound (3.20) on $\lambda_1(\boldsymbol{y})$, this gives

$$\begin{aligned}\|u_1(\boldsymbol{y}) - u_{1,h}(\boldsymbol{y})\|_V^2 \;\leq\; & \frac{1}{a_{\min}}\big(|\lambda_{1,h}(\boldsymbol{y}) - \lambda_1(\boldsymbol{y})| \\ & + \frac{a_{\max}}{a_{\min}}(\chi_1 + 1)\,\|u_1(\boldsymbol{y}) - u_{1,h}(\boldsymbol{y})\|_{\mathcal{D}}^2\big) \\ \leq\; & C_2'\big(|\lambda_{1,h}(\boldsymbol{y}) - \lambda_1(\boldsymbol{y})| \;+\; \|u_1(\boldsymbol{y}) - u_{1,h}(\boldsymbol{y})\|_{\mathcal{D}}^2\big)\,,\end{aligned}$$

where we have combined the constants into C_2', independent of $\boldsymbol{y}$. The first term is exactly the eigenvalue error, which we have just shown is bounded above by $C_1 h^2$, and for the second term by Lemma 3.19 we have

$$\begin{aligned}\|u_1(\boldsymbol{y}) - u_{1,h}(\boldsymbol{y})\|_{\mathcal{D}} \;\leq\; & C_2''\,\|u_1(\boldsymbol{y}) - P_h u_1(\boldsymbol{y})\|_{\mathcal{D}} \\ \leq\; & C_2''\,\|u_1(\boldsymbol{y}) - P_h u_1(\boldsymbol{y})\|_V \;\leq\; C_2'' h\,,\end{aligned}$$

where for the second inequality we have used the equivalence of norms (3.11) and then the Poincaré inequality (3.15), and the last inequality follows by (3.85) from Lemma 3.16. At each step we have absorbed the constants, which are all independent of $\boldsymbol{y}$, into C_2''. Finally, combining everything the upper bound on the squared eigenfunction error is

$$\|u_1(\boldsymbol{y}) - u_{1,h}(\boldsymbol{y})\|_V^2 \;\leq\; C_2'\big(C_1 h^2 \;+\; C_2'' h^2\big)\,,$$

we have shown that C_2', C_1, C_2'' are all independent of $\boldsymbol{y}$ and h, so after taking the square root we have the result (3.42).

Having established the error in the V-norm, for the final error bound (3.43) we use the classical Aubin-Nitsche duality argument. Let $\mathcal{G} \in H^{-1+t}(D)$ and consider the dual problem: find $v_{\mathcal{G}}(\boldsymbol{y}) \in V$ such that

$$\mathcal{B}(\boldsymbol{y}; w, v_{\mathcal{G}}(\boldsymbol{y})) \;=\; \mathcal{G}(w) \quad \text{for all } w \in V\,. \tag{3.97}$$

By the symmetry of $\mathcal{B}(\boldsymbol{y}; \cdot, \cdot)$ and the standard theory for elliptic problems there exists a unique solution $v_{\mathcal{G}}(\boldsymbol{y})$, which is also in $H^{1+t}(D)$ and has norm bounded independently of $\boldsymbol{y}$: $\|v_{\mathcal{G}}(\boldsymbol{y})\|_V \leq C_3'\,\|\mathcal{G}\|_{V^*}$. It is a classical result that the constant is independent of h, and that it is also independent of $\boldsymbol{y}$ has been shown in [60]. Thus, using the equivalence of norms (3.11) and the best-approximation property

of $P_h v_{\mathcal{G}}(\boldsymbol{y})$ in the energy norm we have

$$\|v_{\mathcal{G}}(\boldsymbol{y}) - P_h v_{\mathcal{G}}(\boldsymbol{y})\|_V \leq \frac{a_{\max}(1+C_D^2)}{a_{\min}} \inf_{w_h \in V_h} \|v_{\mathcal{G}}(\boldsymbol{y}) - w_h\|_V \leq C_3'' h^t \,, \tag{3.98}$$

where in the last inequality we have used approximation property (3.39) and the upper bound on the Z^t-norm.

Letting $w = u_1(\boldsymbol{y}) - u_{1,h}(\boldsymbol{y})$ in (3.97), by $\mathcal{B}$-orthogonality of P_h and the boundedness of the bilinear form (3.14) we have

$$\begin{aligned}
|\mathcal{G}(u_1(\boldsymbol{y})) - \mathcal{G}(u_{1,h}(\boldsymbol{y}))| &= |\mathcal{B}(\boldsymbol{y}; u_1(\boldsymbol{y})) - u_{1,h}(\boldsymbol{y}), v_{\mathcal{G}}(\boldsymbol{y}))| \\
&= |\mathcal{B}(\boldsymbol{y}; u_1(\boldsymbol{y}) - u_{1,h}(\boldsymbol{y}), v_{\mathcal{G}}(\boldsymbol{y}) - P_h v_{\mathcal{G}}(\boldsymbol{y}))| \\
&\leq a_{\max}(1+C_D^2) \|u_1(\boldsymbol{y}) - u_{1,h}(\boldsymbol{y})\|_V \|v_{\mathcal{G}}(\boldsymbol{y}) - P_h v_{\mathcal{G}}(\boldsymbol{y}))\|_V \\
&\leq C_3 h^{1+t} \,.
\end{aligned}$$

In the last step we have used the upper bounds on the FE error for the eigenfunction (3.42) and the FE error in approximating $v_{\mathcal{G}}(\boldsymbol{y})$ (3.98). □

Chapter 4

Efficient implementations of the Multivariate Decomposition Method

The main focus of this chapter is on the implementation of the MDM algorithm

$$\mathcal{A}_\varepsilon(f) = \sum_{\mathfrak{u}\in\mathcal{U}_\varepsilon} A_\mathfrak{u}(f_\mathfrak{u}), \tag{4.1}$$

for approximating the integral of an ∞-variate function of the form

$$f(\boldsymbol{y}) = \sum_{\mathfrak{u}\subset\mathbb{N}} f_\mathfrak{u}(\boldsymbol{y}_\mathfrak{u}),$$

as outlined in Section 2.5. That is, given the definition of the active set $\mathcal{U}_\varepsilon$ and the choice of quadrature rules $A_\mathfrak{u}$, we develop computationally efficient strategies to evaluate (4.1) in certain scenarios by exploiting specific structures in the MDM algorithm and the quadrature rules of choice. Specifically,

- we assume a *product and order dependent* (POD) structure in the definition of the active set $\mathcal{U}_\varepsilon$;
- we utilise the *anchored decomposition* of functions; and
- we consider *Quasi-Monte Carlo methods* and *Smolyak's methods* as two alternatives for the quadrature rules $A_\mathfrak{u}$.

In Section 4.1 we explain the structure of our active set $\mathcal{U}_\varepsilon$ and provide an efficient strategy to construct it. Once the active set $\mathcal{U}_\varepsilon$ has been constructed, we need to evaluate the quadrature rules $A_\mathfrak{u}(f_\mathfrak{u})$ for each $\mathfrak{u}\in\mathcal{U}_\varepsilon$; this is formulated in Section 4.2. In this chapter we use the anchored decomposition [64] of f to compute the terms $f_\mathfrak{u}$ explicitly:

$$f_\mathfrak{u}(\boldsymbol{y}_\mathfrak{u}) = \sum_{\mathfrak{v}\subseteq\mathfrak{u}} (-1)^{|\mathfrak{u}|-|\mathfrak{v}|} f(\boldsymbol{y}_\mathfrak{v}; \mathbf{0}), \tag{4.2}$$

where $f(\boldsymbol{y}_\mathfrak{v}; \mathbf{0})$ indicates that we evaluate the function at $f(\boldsymbol{t})$ with components $t_j = y_j$ for $j \in \mathfrak{v}$ and $t_j = 0$ for $j \notin \mathfrak{v}$. Throughout this chapter, by a "naive"

implementation of the MDM algorithm, we mean an implementation that computes the sum in (4.1) term by term, with each $f_{\mathfrak{u}}$ evaluated using (4.2).

We consider only linear algorithms $A_{\mathfrak{u}}$ as the quadrature rules and our MDM algorithm can therefore be expressed as

$$\mathcal{A}_{\varepsilon}(f) = \sum_{\mathfrak{u}\in\mathcal{U}_{\varepsilon}} \sum_{\mathfrak{v}\subseteq\mathfrak{u}} (-1)^{|\mathfrak{u}|-|\mathfrak{v}|} A_{\mathfrak{u}}(f(\cdot_{\mathfrak{v}}; \mathbf{0})). \tag{4.3}$$

Notice inside the double sum in (4.3) that we would be applying a $|\mathfrak{u}|$-dimensional quadrature rule to a function which depends only on a subset $\mathfrak{v} \subseteq \mathfrak{u}$ of the variables. Moreover, the same evaluations of f could be repeated for different combinations of $\mathfrak{u}$ and $\mathfrak{v}$, while in practice the cost of evaluating f could be quite expensive. We will exploit structures in the quadrature rules to save on repeated evaluations in (4.3).

In Section 4.2.1 we first consider *Smolyak quadrature* to be used as the quadrature rules $A_{\mathfrak{u}}$ (see, e.g., [27, 83]). Then in Section 4.2.2 we consider instead an *extensible Quasi-Monte Carlo* (*QMC*) *sequence* to be used for the quadrature rules (see, e.g., [12, 18]). In both sections we explain how to regroup the terms by making use of the recursive structure and how to store some intermediate calculations for the specific quadrature rules to evaluate (4.3) efficiently.

Section 4.3 considers two different approaches to implement the Smolyak quadratures: the direct method and the combination technique. In Section 4.4 we consider a *randomised* Quasi-Monte Carlo sequence for the quadrature rules. This enables us to obtain an unbiased result and a practical estimate of the quadrature error for the MDM algorithm.

Each variant of our MDM algorithm involves three stages, as outlined in the pseudocodes; a summary is given as follows:

Pseudocodes 4.1 + 4.2A + 4.3A	Smolyak MDM – direct implementation
Pseudocodes 4.1 + 4.2A′ + 4.3A′	Smolyak MDM – combination technique
Pseudocodes 4.1 + 4.2B + 4.3B	Extensible QMC MDM
Pseudocodes 4.1 + 4.2B + 4.3B′	Extensible randomised QMC MDM

Section 4.5 is concerned with deriving a computable expression for estimating an infinite series that may appear in the definition of the active set. Finally in Section 4.6 we combine all ingredients and follow the mathematical setting of [58] (see Section 2.5) to construct the active set and choose the quadrature rules. We then apply the MDM algorithm to an example integrand that mimics the characteristics of the integrands arising from some parametrised PDE problems (see, e.g., [60] and Section 2.2).

4.1 Constructing the active set

Letting $w(\mathfrak{u})$ be a measure of the "significance" of the subset $\mathfrak{u}$, we assume that the mathematical analysis yields the definition of an active set of the general form

$$\mathcal{U}_\varepsilon \coloneqq \{\mathfrak{u} \subset \mathbb{N} : w(\mathfrak{u}) > T\}, \tag{4.4}$$

where T is a "threshold" parameter that depends on the overall error demand $\varepsilon > 0$ and possibly on all of $w(\mathfrak{u})$. For example, $w(\mathfrak{u})$ can be related to the weight parameters from a weighted function space setting (as in [33, 76, 89, 90], see also Section 2.3.2), or it can be related to the bounds on the norm of $f_\mathfrak{u}$ (as in the setting of [58] outlined in Section 2.5).

We assume $w(\emptyset) > T$ so that we always have $\emptyset \in \mathcal{U}_\varepsilon$. Furthermore, we assume specifically for $\mathfrak{u} \neq \emptyset$ that $w(\mathfrak{u})$ takes the *product and order dependent* (POD) form:

$$w(\mathfrak{u}) \coloneqq \Omega_{|\mathfrak{u}|} \prod_{j \in \mathfrak{u}} \omega_j, \tag{4.5}$$

where $\omega_1 \geq \omega_2 \geq \cdots$ is a non-increasing sequence of nonnegative real numbers controlling the "product aspect", and $\Omega_1, \Omega_2, \ldots$ is a second sequence of nonnegative real numbers controlling the "order dependent aspect", cf. (2.8). The only restriction on Ω_ℓ is that its growth is controlled by ω_ℓ, i.e., $\Omega_{\ell+1}\omega_{\ell+1} \leq \Omega_\ell$ for all $\ell \in \mathbb{N}$. This avoids pathological examples that may fool our stopping criteria.

With the active set defined by (4.4) and (4.5), we make a couple of obvious remarks:

1. If $\mathfrak{v} \in \mathcal{U}_\varepsilon$ then $\mathfrak{u} \in \mathcal{U}_\varepsilon$ for all sets $\mathfrak{u}$ satisfying $w(\mathfrak{u}) \geq w(\mathfrak{v})$.
2. If $\mathfrak{u} \notin \mathcal{U}_\varepsilon$ then $\mathfrak{v} \notin \mathcal{U}_\varepsilon$ for all sets $\mathfrak{v}$ satisfying $w(\mathfrak{u}) \geq w(\mathfrak{v})$.

We identify any finite non-empty set $\mathfrak{u} \subset \mathbb{N}$ with a vector containing the elements of $\mathfrak{u}$ in increasing order, i.e., if $|\mathfrak{u}| = \ell$ then

$$\mathfrak{u} \coloneqq (u_1, u_2, \ldots, u_\ell), \quad u_1 < u_2 < \cdots < u_\ell.$$

Then, due to our assumed POD structure in (4.5), we note that

3. $w(\mathfrak{u}) \geq w(\mathfrak{v})$ if $|\mathfrak{u}| = |\mathfrak{v}|$ and $u_i \leq v_i$ for all $i = 1, \ldots, \ell$.
4. $w(\{1, \ldots, \ell\}) \geq w(\{1, \ldots, \ell + 1\})$ for all $\ell \in \mathbb{N}$.
5. For any $\mathfrak{u} \in \mathcal{U}_\varepsilon$, a subset of $\mathfrak{u}$ need *not* be included in $\mathcal{U}_\varepsilon$.

Note that if the opposite of Item 5 were true, i.e., every subset of $\mathfrak{u} \in \mathcal{U}_\varepsilon$ also belongs to $\mathcal{U}_\varepsilon$, then the set $\mathcal{U}_\varepsilon$ is said to be "downward closed" in some papers; we do not impose this condition.

Combining the above, we deduce the following simple lemma.

Lemma 4.1. *Assume that the active set $\mathcal{U}_\varepsilon$ is defined by* (4.4) *and* (4.5).

- Superposition dimension: *Let $\sigma(\mathcal{U}_\varepsilon)$ be the largest possible value of ℓ for which $(1, 2, \ldots, \ell) \in \mathcal{U}_\varepsilon$, i.e., $w(\{1, 2, \ldots, \ell\}) > T$. Then for all $\mathfrak{u} \in \mathcal{U}_\varepsilon$ we have $|\mathfrak{u}| \leq \sigma(\mathcal{U}_\varepsilon)$.*
- Truncation dimension for sets of order ℓ: *For any $\ell = 1, \ldots, \sigma(\mathcal{U}_\varepsilon)$, let $\tau_\ell(\mathcal{U}_\varepsilon)$ be the largest possible value of $j \geq \ell$ for which $(1, 2, \ldots, \ell - 1, j) \in \mathcal{U}_\varepsilon$, that is, $w(\{1, 2, \ldots, \ell - 1, j\}) > T$. Then for all $\mathfrak{u} \in \mathcal{U}_\varepsilon$ with $|\mathfrak{u}| = \ell$, we have $u_\ell \leq \tau_\ell(\mathcal{U}_\varepsilon)$; and consequently, $i \leq u_i \leq \tau_\ell(\mathcal{U}_\varepsilon) - \ell + i$ for all $1 \leq i \leq \ell$.*
- Truncation dimension: *Let $\tau(\mathcal{U}_\varepsilon)$ be the largest possible value of j for which $j \in \mathfrak{u} \in \mathcal{U}_\varepsilon$, i.e., $\tau(\mathcal{U}_\varepsilon) = \max_{\mathfrak{u} \in \mathcal{U}_\varepsilon} \max_{j \in \mathfrak{u}} j$. Then $\tau(\mathcal{U}_\varepsilon) = \max_{1 \leq \ell \leq \sigma(\mathcal{U}_\varepsilon)} \tau_\ell(\mathcal{U}_\varepsilon)$.*

Proof. For the first point, suppose on the contrary that $\mathfrak{u} = (u_1, \ldots, u_{\sigma(\mathcal{U}_\varepsilon)+1}) \in \mathcal{U}_\varepsilon$. Then letting $\mathfrak{v} \coloneqq (1, \ldots, \sigma(\mathcal{U}_\varepsilon) + 1)$ we have $w(\mathfrak{v}) \geq w(\mathfrak{u}) > T$, which indicates that $\mathfrak{v} \in \mathcal{U}_\varepsilon$, contradicting the definition of $\sigma(\mathcal{U}_\varepsilon)$.

To demonstrate the second point, suppose on the contrary that $\mathfrak{u} = (u_1, \ldots, u_\ell) \in \mathcal{U}_\varepsilon$ with $u_\ell > \tau_\ell(\mathcal{U}_\varepsilon)$. Then we have $\mathfrak{v} = (1, \ldots, \ell - 1, u_\ell)$ with $w(\mathfrak{v}) \geq w(\mathfrak{u}) > T$, which indicates that $\mathfrak{v} \in \mathcal{U}_\varepsilon$, but this contradicts the definition of $\tau_\ell(\mathcal{U}_\varepsilon)$. The bound on u_i then follows easily.

The third point is straightforward. □

We construct the active set as outlined in Pseudocode 4.1. The algorithm adds the qualifying sets to the collection in the order of increasing cardinality. For each $\ell \geq 1$, starting from the set $(1, 2, \ldots, \ell)$, the algorithm incrementally generates and checks sets to be added to the collection. The algorithm terminates when it reaches a value of ℓ for which $(1, 2, \ldots, \ell) \notin \mathcal{U}_\varepsilon$, i.e., $w(\{1, 2, \ldots, \ell\}) \leq T$.

The assumptions on the structure of $w(\mathfrak{u})$ and properties 1.–5. above ensure that this stopping criteria is valid, and hence that Pseudocode 4.1 does indeed construct the active set (4.4). In particular, property 3. implies $w(\mathfrak{u}) \leq w(\{1, 2, \ldots, \ell\})$ for all sets with $|\mathfrak{u}| = \ell$, and then property 4. implies $w(\mathfrak{u}) \leq w(\{1, 2, \ldots, \ell\})$ for all sets with $|\mathfrak{u}| \geq \ell$. Thus, if $\{1, 2, \ldots, \ell\} \notin \mathcal{U}_\varepsilon$ then no set with cardinality ℓ or higher is in $\mathcal{U}_\varepsilon$.

We recommend storing the active set $\mathcal{U}_\varepsilon$ as an array of hash tables, with one table for each cardinality, since in the next section we will have to iterate over all subsets $\mathfrak{v} \subseteq \mathfrak{u} \in \mathcal{U}_\varepsilon$ and be able to update a table stored with each such $\mathfrak{v}$.

Remark 4.2. In the case where $w(\mathfrak{u})$ is of product form, i.e., $\Omega_\ell = 1$ for all $\ell \geq 0$, an alternative method for constructing active sets that are "optimal" will be detailed in Chapter 5.

Pseudocode 4.1 Constructing the active set

Add $\emptyset$ to $\mathcal{U}_\varepsilon$
for ℓ **from** 1 **to** $\ell_{\text{threshold}}$ **do** ▷ $\ell_{\text{threshold}}$ is a computational threshold
    $\mathfrak{u} \leftarrow (1, 2, \ldots, \ell)$
    $i \leftarrow \ell$ ▷ i is the index for the next increment
    while $i > 0$ **do**
        if $w(\mathfrak{u}) > T$ **then**
            $i \leftarrow \ell$ ▷ continue updating $\mathfrak{u}$ at last index
            Add $\mathfrak{u}$ to $\mathcal{U}_\varepsilon$ ▷ add $\mathfrak{u}$ to the active set
        else
            $i \leftarrow i - 1$ ▷ increment from lower index
        end if
        if $i = 0$ **then break while loop** ▷ move to next cardinality
        for j **from** i **to** ℓ **do** ▷ increment $\mathfrak{u}$ from u_i
            $u_j \leftarrow u_i + j - i + 1$
        end for
    end while
    break the outer loop if no sets of size ℓ found ▷ terminate
end for

4.2 Formulating the MDM algorithm

In this section we outline how to formulate the MDM algorithm (4.3) in a way that is specific to the quadrature rules used, so that the implementation can be as efficient as possible. We do this by exploiting the structure in the anchored decomposition (4.2), and also in the quadrature rules, which will be Smolyak's methods (also known as sparse grid methods) and Quasi-Monte Carlo rules.

Recall from (4.3) that the MDM algorithm using the anchored decomposition is given by

$$\mathcal{A}_\varepsilon(f) \;=\; \sum_{\mathfrak{u}\in\mathcal{U}_\varepsilon} \sum_{\mathfrak{v}\subseteq\mathfrak{u}} (-1)^{|\mathfrak{u}|-|\mathfrak{v}|} A_{\mathfrak{u}}(f(\cdot_{\mathfrak{v}}; \mathbf{0}))\,.$$

Clearly there will be subsets $\mathfrak{v}$ that will occur many times over, so implementing the MDM in this way could be severely inefficient, because it would evaluate the same functions $f(\cdot_{\mathfrak{v}}; \mathbf{0})$ at the same quadrature points over and over again. The goal of this section is to detail how to implement the quadrature approximations in such a way that each function $f(\cdot_{\mathfrak{v}}; \mathbf{0})$ is evaluated at each quadrature point *once only*.

The first step is to introduce the *extended active set*:

$$\mathcal{U}_\varepsilon^{\text{ext}} \coloneqq \{\mathfrak{v} \subset \mathbb{N} : \mathfrak{v} \subseteq \mathfrak{u} \text{ for } \mathfrak{u} \in \mathcal{U}_\varepsilon\},$$

that is, it includes all subsets of the sets in the active set. Then we can swap the sums above to give

$$\begin{aligned}\mathcal{A}_\varepsilon(f) &= \sum_{\mathfrak{v} \in \mathcal{U}_\varepsilon^{\text{ext}}} \sum_{\substack{\mathfrak{u} \in \mathcal{U}_\varepsilon \\ \mathfrak{u} \supseteq \mathfrak{v}}} (-1)^{|\mathfrak{u}|-|\mathfrak{v}|} A_\mathfrak{u}(f(\cdot_\mathfrak{v}; \mathbf{0})) \\ &= c_\emptyset\, f(\mathbf{0}) + \sum_{\emptyset \neq \mathfrak{v} \in \mathcal{U}_\varepsilon^{\text{ext}}} \sum_{\substack{\mathfrak{u} \in \mathcal{U}_\varepsilon \\ \mathfrak{u} \supseteq \mathfrak{v}}} (-1)^{|\mathfrak{u}|-|\mathfrak{v}|} A_\mathfrak{u}(f(\cdot_\mathfrak{v}; \mathbf{0})), \end{aligned} \tag{4.6}$$

where we separated out the $\mathfrak{v} = \emptyset$ terms, with

$$c_\emptyset \coloneqq \sum_{\mathfrak{u} \in \mathcal{U}_\varepsilon} (-1)^{|\mathfrak{u}|}.$$

After constructing the active set $\mathcal{U}_\varepsilon$, we go through it again to construct the extended active set $\mathcal{U}_\varepsilon^{\text{ext}}$, and at the same time store information regarding the superset structure of each element in $\mathcal{U}_\varepsilon^{\text{ext}}$. We would like to store just enough details so that for each $\mathfrak{v} \in \mathcal{U}_\varepsilon^{\text{ext}}$ we can compute the approximation $\sum_{\mathfrak{u} \in \mathcal{U}_\varepsilon \,:\, \mathfrak{u} \supseteq \mathfrak{v}} (-1)^{|\mathfrak{u}|-|\mathfrak{v}|} A_\mathfrak{u}(f(\cdot_\mathfrak{v}; \mathbf{0}))$ without the need to access the supersets of $\mathfrak{v}$. Specific details on how this is done will depend on the quadrature rule used.

4.2.1 Quadrature rules based on Smolyak's method

For a nonempty set $\mathfrak{u} \subset \mathbb{N}$ and integer $m \geq 1$, from (2.17) Smolyak's method applied to a function $g_\mathfrak{u}$ of the variables $\boldsymbol{y}_\mathfrak{u}$ takes the form

$$\mathcal{Q}_{\mathfrak{u},m}(g_\mathfrak{u}) \coloneqq \sum_{\substack{\boldsymbol{i}_\mathfrak{u} \in \mathbb{N}^{|\mathfrak{u}|} \\ |\boldsymbol{i}_\mathfrak{u}| \leq |\mathfrak{u}|+m-1}} \bigotimes_{j \in \mathfrak{u}} (U_{i_j} - U_{i_j-1})(g_\mathfrak{u}), \tag{4.7}$$

where recall that $|\boldsymbol{i}_\mathfrak{u}| \coloneqq \sum_{j \in \mathfrak{u}} i_j$, $m \geq 1$, and $\{U_i\}_{i \geq 1}$ is a sequence of one-dimensional quadrature rules, not necessarily nested, with $U_0 \coloneqq 0$ denoting the zero algorithm. Furthermore, we now assume that constant functions are integrated exactly, so that $U_i(1) = 1$ for $i \geq 1$—because ρ is a probability density.

For a nonempty subset $\mathfrak{v} \subseteq \mathfrak{u}$, suppose now that the function $g_{\mathfrak{v}}$ depends only on the variables $\boldsymbol{y}_{\mathfrak{v}}$. Then we have

$$\begin{aligned}
\mathcal{Q}_{\mathfrak{u},m}(g_{\mathfrak{v}}) \;&=\; \sum_{\substack{\boldsymbol{i}_{\mathfrak{u}} \in \mathbb{N}^{|\mathfrak{u}|} \\ |\boldsymbol{i}_{\mathfrak{u}}| \le |\mathfrak{u}|+m-1}} \Bigg(\bigotimes_{j \in \mathfrak{v}} (U_{i_j} - U_{i_j-1})(g_{\mathfrak{v}}) \Bigg) \Bigg(\bigotimes_{j \in \mathfrak{u} \setminus \mathfrak{v}} (U_{i_j} - U_{i_j-1})(1) \Bigg) \\
&=\; \sum_{\substack{\boldsymbol{i}_{\mathfrak{u}} \in \mathbb{N}^{|\mathfrak{u}|} \\ |\boldsymbol{i}_{\mathfrak{u}}| \le |\mathfrak{u}|+m-1,\, \boldsymbol{i}_{\mathfrak{u}\setminus\mathfrak{v}} = \mathbf{1}}} \bigotimes_{j \in \mathfrak{v}} (U_{i_j} - U_{i_j-1})(g_{\mathfrak{v}}) \;=\; \sum_{\substack{\boldsymbol{i}_{\mathfrak{v}} \in \mathbb{N}^{|\mathfrak{v}|} \\ |\boldsymbol{i}_{\mathfrak{v}}| \le |\mathfrak{v}|+m-1}} \bigotimes_{j \in \mathfrak{v}} (U_{i_j} - U_{i_j-1})(g_{\mathfrak{v}}) \\
&=\; \mathcal{Q}_{\mathfrak{v},m}(g_{\mathfrak{v}}). \qquad (4.8)
\end{aligned}$$

In the second equality above we used the assumption that the one-dimensional quadrature rules integrate the constant functions exactly and thus $(U_{i_j} - U_{i_j-1})(1)$ is 1 if $i_j = 1$ and is 0 otherwise. The above derivation (4.8) indicates how a Smolyak quadrature rule is projected down when it is applied to a lower dimensional function. This property is important in our efficient evaluation of (4.6).

In (4.6) we take

$$A_{\mathfrak{u}} \;\equiv\; \mathcal{Q}_{\mathfrak{u},m_{\mathfrak{u}}},$$

where the level $m_{\mathfrak{u}}$ has a direct relation to the number of quadrature points $n_{\mathfrak{u}}$, depending on the choice of the one-dimensional quadrature rules $\{U_i\}$.

We define

$$m_{\max} \;\coloneqq\; \max\{m_{\mathfrak{u}} : \emptyset \neq \mathfrak{u} \in \mathcal{U}_\varepsilon\}.$$

Since the value of $m_{\mathfrak{u}}$ is roughly the logarithm of $n_{\mathfrak{u}}$ and $|\mathfrak{u}| \le \sigma(\mathcal{U}_\varepsilon)$ (recall that $\sigma(\mathcal{U}_\varepsilon)$ is the superposition dimension of the active set as defined in Lemma 4.1), we expect $m_{\max}$ to be relatively small, e.g., $m_{\max} \approx 25$.

Using (4.8) we can rewrite (4.6) as follows (note the change from $\mathcal{Q}_{\mathfrak{u},m_{\mathfrak{u}}}$ to $\mathcal{Q}_{\mathfrak{v},m_{\mathfrak{u}}}$):

$$\begin{aligned}
\mathcal{A}_\varepsilon^{\mathrm{S}}(f) \;&=\; c_\emptyset\, f(\mathbf{0}) + \sum_{\emptyset \neq \mathfrak{v} \in \mathcal{U}_\varepsilon^{\mathrm{ext}}} \sum_{\substack{\mathfrak{u} \in \mathcal{U}_\varepsilon \\ \mathfrak{u} \supseteq \mathfrak{v}}} (-1)^{|\mathfrak{u}|-|\mathfrak{v}|}\, \mathcal{Q}_{\mathfrak{u},m_{\mathfrak{u}}}(f(\cdot_{\mathfrak{v}}; \mathbf{0})) \\
&=\; c_\emptyset\, f(\mathbf{0}) + \sum_{\emptyset \neq \mathfrak{v} \in \mathcal{U}_\varepsilon^{\mathrm{ext}}} \sum_{\substack{\mathfrak{u} \in \mathcal{U}_\varepsilon \\ \mathfrak{u} \supseteq \mathfrak{v}}} (-1)^{|\mathfrak{u}|-|\mathfrak{v}|}\, \mathcal{Q}_{\mathfrak{v},m_{\mathfrak{u}}}(f(\cdot_{\mathfrak{v}}; \mathbf{0})) \\
&=\; c_\emptyset\, f(\mathbf{0}) + \sum_{\emptyset \neq \mathfrak{v} \in \mathcal{U}_\varepsilon^{\mathrm{ext}}} \sum_{\substack{m=1 \\ c(\mathfrak{v},m) \neq 0}}^{m_{\max}} c(\mathfrak{v}, m)\, \mathcal{Q}_{\mathfrak{v},m}(f(\cdot_{\mathfrak{v}}; \mathbf{0}))\,, \qquad (4.9)
\end{aligned}$$

where for $\mathfrak{v} \neq \emptyset$ and $m = 1, \ldots, m_{\max}$ we define

$$c(\mathfrak{v}, m) \coloneqq \sum_{\substack{\mathfrak{u} \in \mathcal{U}_\varepsilon,\, \mathfrak{u} \supseteq \mathfrak{v} \\ m_\mathfrak{u} = m}} (-1)^{|\mathfrak{u}| - |\mathfrak{v}|}. \tag{4.10}$$

The values of $c(\mathfrak{v}, m)$ can be computed and stored while we construct the extended active set $\mathcal{U}_\varepsilon^{\text{ext}}$ as follows. We work through the sets in the active set in order of increasing cardinality. For each nonempty set $\mathfrak{u} \in \mathcal{U}_\varepsilon$, we calculate the required level $m_\mathfrak{u}$ then generate all nonempty subsets $\mathfrak{v} \subseteq \mathfrak{u}$, adding the missing subsets to $\mathcal{U}_\varepsilon^{\text{ext}}$ and updating$c(\mathfrak{v}, m)$ as we go. This procedure is given in Pseudocode 4.2A.

This formulation (4.9)–(4.10) allows us to compute the $\mathcal{Q}_{\mathfrak{v},m}(f(\cdot_\mathfrak{v}; \mathbf{0}))$ for different supersets $\mathfrak{u}$ with the same value of $m_\mathfrak{u}$ only once. If the Smolyak MDM algorithm is implemented in this way then there is no need to access the superset structure. Obviously, if $c(\mathfrak{v}, m) = 0$ then we do not perform the quadrature approximation.

Note that in practice calculating the number of Smolyak levels $m_\mathfrak{u}$ (or the number of QMC points in the next subsection) normally requires knowledge of the entire active set $\mathcal{U}_\varepsilon$, see (2.39) in Section 2.5.5, hence we compute them when constructing $\mathcal{U}_\varepsilon^{\text{ext}}$.

Note also that we do not need a separate data structure for $\mathcal{U}_\varepsilon^{\text{ext}}$: we can simply extend $\mathcal{U}_\varepsilon$ to $\mathcal{U}_\varepsilon^{\text{ext}}$ since Step 9 in Pseudocode 4.2A only adds subsets with lower cardinalities and would not interfere with Step 3 since we iterate in increasing cardinality. As we explained in the previous section, we store the active set $\mathcal{U}_\varepsilon$, and by extension the extended active set $\mathcal{U}_\varepsilon^{\text{ext}}$, as an array of hash tables to easily retrieve the $c(\mathfrak{v}, \cdot)$ table for each $\mathfrak{v}$.

A direct implementation of the MDM algorithm with Smolyak quadratures is given in Pseudocode 4.3A. The different formulas (4.17)–(4.18) or (4.19)–(4.20) for implementing the Smolyak quadrature, which depend on whether we have a non-nested or nested rule, will be discussed in Section 4.3.

Pseudocode 4.2A Constructing the extended active set for Smolyak

Initialise $\mathcal{U}_\varepsilon^{\mathrm{ext}} \leftarrow \mathcal{U}_\varepsilon$ ▷ start from the active set
Initialise $c_\emptyset \leftarrow 1$ ▷ for $\mathfrak{u} = \emptyset$
for $\emptyset \neq \mathfrak{u} \in \mathcal{U}_\varepsilon$ with $|\mathfrak{u}|$ from 1 to $\sigma(\mathcal{U}_\varepsilon)$ **do** ▷ traverse in increasing cardinality
    Calculate $m_\mathfrak{u}$ ▷ formula for $m_\mathfrak{u}$ is given from theory
    Update $c_\emptyset \leftarrow c_\emptyset + (-1)^{|\mathfrak{u}|}$
    Initialise $c(\mathfrak{u}, m) \leftarrow 0$ for m from 1 to $m_{\max}$
    for $\emptyset \neq \mathfrak{v} \subseteq \mathfrak{u}$ **do** ▷ generate nonempty subsets
        if $\mathfrak{v} \notin \mathcal{U}_\varepsilon^{\mathrm{ext}}$ **then** ▷ look up and add missing subset
            Add $\mathfrak{v}$ to $\mathcal{U}_\varepsilon^{\mathrm{ext}}$
            Initialise $c(\mathfrak{v}, m) \leftarrow 0$ for m from 1 to $m_{\max}$
        end if
        Update $c(\mathfrak{v}, m_\mathfrak{u}) \leftarrow c(\mathfrak{v}, m_\mathfrak{u}) + (-1)^{|\mathfrak{u}|-|\mathfrak{v}|}$ ▷ update relevant entry
    end for
end for

Pseudocode 4.3A Smolyak MDM

Initialise $\mathcal{A}_\varepsilon^{\mathrm{S}}(f) \leftarrow c_\emptyset \times f(\mathbf{0})$
for $\emptyset \neq \mathfrak{v} \in \mathcal{U}_\varepsilon^{\mathrm{ext}}$ **do**
    for m from 1 to $m_{\max}$ **do**
        if $c(\mathfrak{v}, m) \neq 0$ **then**
            Calculate $\mathcal{Q}_{\mathfrak{v},m}(f(\cdot_\mathfrak{v}; \mathbf{0}))$ using (4.17)–(4.18) or (4.19)–(4.20)
            Update $\mathcal{A}_\varepsilon^{\mathrm{S}}(f) \leftarrow \mathcal{A}_\varepsilon^{\mathrm{S}}(f) + c(\mathfrak{v}, m) \times \mathcal{Q}_{\mathfrak{v},m}(f(\cdot_\mathfrak{v}; \mathbf{0}))$
        end if
    end for
end for
return $\mathcal{A}_\varepsilon^{\mathrm{S}}(f)$

4.2.2 Quadrature rules based on Quasi-Monte Carlo methods

In this section we assume for simplicity that $\mathcal{Y} = [0,1]$ and $\rho \equiv 1$. Recall from Section 2.3 that an s-dimensional *Quasi-Monte Carlo* (QMC) rule with n points $\boldsymbol{t}^{(i)} = (t_1^{(i)}, t_2^{(i)}, \ldots, t_s^{(i)})$, $i = 0, \ldots, n-1$, approximates the integral of a function g by the equal-weight average

$$\int_{[0,1]^s} g(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} \;\approx\; \frac{1}{n}\sum_{i=0}^{n-1} g(\boldsymbol{t}^{(i)}). \tag{4.11}$$

In (4.1) each $A_{\mathfrak{u}}$ could be a different $|\mathfrak{u}|$-dimensional QMC rule with $n_{\mathfrak{u}}$ points, but in that case we would not be able to reuse any function evaluation.

Instead, we consider here an "*extensible Quasi-Monte Carlo sequence*". By "extensible" we mean that we can take just the initial dimensions of the initial points in the sequence. By "Quasi-Monte Carlo" we mean that a quadrature rule based on the first n points of this sequence has equal quadrature weights $1/n$. We choose to use a QMC rule with $\sigma(\mathcal{U}_\varepsilon)$ dimensions instead of $\tau(\mathcal{U}_\varepsilon)$, since $\tau(\mathcal{U}_\varepsilon)$ can be really large (e.g., 30000) while $\sigma(\mathcal{U}_\varepsilon)$ is rather small (e.g., 15), and QMC rules with fewer dimensions are of better quality.

Now we consider how to evaluate the terms in (4.6) where each $A_{\mathfrak{u}}$ is a QMC approximation (4.11); since QMC rules are designed to favour earlier dimensions they are not invariant under permutations of dimensions like Smolyak rules. Hence, applying the QMC rule $A_{\mathfrak{u}}$ to a function of the variables $\boldsymbol{y}_{\mathfrak{v}}$ only (with $\mathfrak{v} \subset \mathfrak{u}$) is more complicated because an equation like (4.8) does not hold in general for QMC rules. For any nonempty set $\mathfrak{u} \in \mathcal{U}_\varepsilon$ and nonempty subset $\mathfrak{v} \subseteq \mathfrak{u}$ we have

$$A_{\mathfrak{u}}(f(\cdot_{\mathfrak{v}}; \mathbf{0})) \;=\; \frac{1}{n_{\mathfrak{u}}} \sum_{i=0}^{n_{\mathfrak{u}}-1} f\left(\boldsymbol{t}^{(i)}_{\mathfrak{u}|\mathfrak{v}\to\mathfrak{v}}; \mathbf{0}\right), \tag{4.12}$$

where, loosely speaking, $\boldsymbol{t}_{\mathfrak{u}|\mathfrak{v}\to\mathfrak{v}}$ indicates that we map a point $\boldsymbol{t}$ to the variables $\boldsymbol{y}_{\mathfrak{u}}$ and then to $\boldsymbol{y}_{\mathfrak{v}}$, which is *not* the same as mapping $\boldsymbol{t}$ directly to $\boldsymbol{y}_{\mathfrak{v}}$. More explicitly, recalling that $\mathfrak{u} = (u_1, u_2, \ldots, u_{|\mathfrak{u}|})$ with ordered elements, the subscript $\mathfrak{u}|\mathfrak{v} \to \mathfrak{v}$ represents two steps. First, $\mathfrak{u}|\mathfrak{v}$ denotes that we take the first $|\mathfrak{u}|$-dimensions of the point $\boldsymbol{t}$ and apply them to the variables in $\boldsymbol{y}_{\mathfrak{u}}$, retaining only those components in $\mathfrak{v}$. Second, $\mathfrak{w} \to \mathfrak{v}$ denotes that we map the component indexed by w_i to the variable y_{v_i} for $i = 1, 2, \ldots, |\mathfrak{v}|$. Thus, for $j \in \mathfrak{v}$, the jth component of the mapped point $\boldsymbol{t}_{\mathfrak{u}|\mathfrak{v}\to\mathfrak{u}}$ is given explicitly by

$$(\boldsymbol{t}_{\mathfrak{u}|\mathfrak{v}\to\mathfrak{v}})_j \;=\; t_i \quad \text{with } u_i = j \in \mathfrak{v}\,,$$

where the notation $(\boldsymbol{t})_j$ denotes the jth entry of $\boldsymbol{t}$. The function f is then evaluated by anchoring all components outside of $\mathfrak{v}$ to zero; for $j \in \mathbb{N}$ the jth component of the mapped-then-anchored point is $(\boldsymbol{t}_{\mathfrak{u}|\mathfrak{u}\to\mathfrak{v}})_j = t_i$ if $u_i = j \in \mathfrak{v}$, and 0 otherwise.

As an example of the how the mapping works, take $\mathfrak{u} = (1, 5, 7)$ and $\mathfrak{v} = (1, 7)$. We get $\mathfrak{u}|\mathfrak{v} = (1, 3)$ since the set $\mathfrak{v}$ originates from the position $\mathfrak{w} = (1, 3)$ of its superset $\mathfrak{u}$. We assign the quadrature point $(t_1^{(i)}, t_2^{(i)}, t_3^{(i)})$ to the variables (y_1, y_5, y_7). Then the point $(t_1^{(i)}, t_3^{(i)})$ is assigned to the variables (y_1, y_7), and hence we evaluate $f(t_1^{(i)}, 0, 0, 0, 0, 0, t_3^{(i)}, 0, \ldots)$.

Substituting in (4.12) the algorithm (4.6) in this case is given by

$$\mathcal{A}_{\varepsilon}^{\mathrm{Q}}(f) \;=\; c_{\emptyset}\, f(\mathbf{0}) + \sum_{\emptyset\neq\mathfrak{v}\in\mathcal{U}_{\varepsilon}^{\mathrm{ext}}} \sum_{\substack{\mathfrak{u}\in\mathcal{U}_{\varepsilon}\\ \mathfrak{u}\supseteq\mathfrak{v}}} (-1)^{|\mathfrak{u}|-|\mathfrak{v}|} \left(\frac{1}{n_{\mathfrak{u}}} \sum_{i=0}^{n_{\mathfrak{u}}-1} f\left(\boldsymbol{t}_{\mathfrak{u}|\mathfrak{v}\to\mathfrak{v}}^{(i)};\mathbf{0}\right)\right). \tag{4.13}$$

Note that the same set $\mathfrak{v}=(1,7)$ can originate from the position $\mathfrak{w}=(1,3)$ of different supersets $\mathfrak{u}$: for example, $\mathfrak{u}=(1,6,7)$, $\mathfrak{u}=(1,4,7,13)$, and many others. We can make use of this repetition to save on computational cost.

Let $\mathcal{M}(\mathfrak{v})$ denote the set of all different positions that a nonempty set $\mathfrak{v}$ can originate from for all its supersets in the active set:

$$\mathcal{M}(\mathfrak{v}) \coloneqq \{\mathfrak{w}\subseteq\{1,\dots,\sigma(\mathcal{U}_{\varepsilon})\} \,:\, \mathfrak{w}\equiv\mathfrak{u}|\mathfrak{v} \text{ for some } \mathfrak{u}\in\mathcal{U}_{\varepsilon} \text{ with } \mathfrak{u}\supseteq\mathfrak{v}\}.$$

For simplicity and for convenience, we assume further that $n_{\mathfrak{u}}=2^{m_{\mathfrak{u}}}$, with $0\le m_{\mathfrak{u}}\le m_{\max}$ (e.g., with $m_{\max}\approx 25$). This allows us to rewrite each QMC approximation as a sum of blocks of points (recall that the QMC points are extensible)

$$\frac{1}{2^{m_{\mathfrak{u}}}}\sum_{i=0}^{2^{m_{\mathfrak{u}}}-1} f\left(\boldsymbol{t}_{\mathfrak{u}|\mathfrak{v}\to\mathfrak{v}}^{(i)};\mathbf{0}\right) \;=\; \frac{1}{2^{m_{\mathfrak{u}}}}\sum_{m=0}^{m_{\mathfrak{u}}}\sum_{i=\lfloor 2^{m-1}\rfloor}^{2^m-1} f\left(\boldsymbol{t}_{\mathfrak{u}|\mathfrak{v}\to\mathfrak{v}}^{(i)};\mathbf{0}\right),$$

where the floor function is used to specifically take care of the $m=0$ case. Substituting this into (4.13) and introducing a sum over $\mathcal{M}(\mathfrak{v})$, we have

$$\mathcal{A}_{\varepsilon}^{\mathrm{Q}}(f) \;=\; c_{\emptyset}\, f(\mathbf{0}) \;+\; \sum_{\emptyset\neq\mathfrak{v}\in\mathcal{U}_{\varepsilon}^{\mathrm{ext}}} \sum_{\mathfrak{w}\in\mathcal{M}(\mathfrak{v})} \sum_{\substack{\mathfrak{u}\in\mathcal{U}_{\varepsilon}\\ \mathfrak{u}\supseteq\mathfrak{v}\\ \mathfrak{u}|\mathfrak{v}\equiv\mathfrak{w}}} (-1)^{|\mathfrak{u}|-|\mathfrak{v}|} \frac{1}{2^{m_{\mathfrak{u}}}}\sum_{m=0}^{m_{\mathfrak{u}}}\sum_{i=\lfloor 2^{m-1}\rfloor}^{2^m-1} f\left(\boldsymbol{t}_{\mathfrak{w}\to\mathfrak{v}}^{(i)};\mathbf{0}\right),$$

where in the third sum we have added the restriction that $\mathfrak{u}|\mathfrak{v}$ is equivalent to the position $\mathfrak{w}$. Collecting the sums

$$S_{\mathfrak{v},\mathfrak{w},m}(f) \;\coloneqq\; \sum_{i=\lfloor 2^{m-1}\rfloor}^{2^m-1} f\left(\boldsymbol{t}_{\mathfrak{w}\to\mathfrak{v}}^{(i)};\mathbf{0}\right), \tag{4.14}$$

we can then rewrite the QMC MDM (4.13) as

$$\mathcal{A}_{\varepsilon}^{\mathrm{Q}}(f) \;=\; c_{\emptyset}\, f(\mathbf{0}) + \sum_{\emptyset\neq\mathfrak{v}\in\mathcal{U}_{\varepsilon}^{\mathrm{ext}}} \sum_{\mathfrak{w}\in\mathcal{M}(\mathfrak{v})} \sum_{\substack{m=0\\ c(\mathfrak{v},\mathfrak{w},m)\neq 0}}^{m_{\max}} c(\mathfrak{v},\mathfrak{w},m)\frac{S_{\mathfrak{v},\mathfrak{w},m}(f)}{2^{m_{\max}}}, \tag{4.15}$$

where for a nonempty set $\mathfrak{v}$, a position $\mathfrak{w} \in \mathcal{M}(\mathfrak{v})$, and $m = 0, \ldots, m_{\max}$ we define

$$c(\mathfrak{v}, \mathfrak{w}, m) := \sum_{\substack{\mathfrak{u} \in \mathcal{U},\, \mathfrak{u} \supseteq \mathfrak{v} \\ \mathfrak{u}|\mathfrak{v} \equiv \mathfrak{w},\, m_{\mathfrak{u}} \geq m}} (-1)^{|\mathfrak{u}|-|\mathfrak{v}|}\, 2^{m_{\max}-m_{\mathfrak{u}}}\,. \tag{4.16}$$

Note that we have chosen to multiply and divide by $2^{m_{\max}}$ to ensure that each $c(\mathfrak{v}, \mathfrak{w}, m)$ is integer valued.

We can compute and store a list of positions $\mathcal{M}(\mathfrak{v})$ and the values $c(\mathfrak{v}, \mathfrak{w}, m)$ when we construct $\mathcal{U}_\varepsilon^{\text{ext}}$ by extending the active set $\mathcal{U}_\varepsilon$, in a similar way to the Smolyak case in the previous subsection. This is presented in Pseudocode 4.2B. The new algorithm is more complicated due to the need to store the positions $\mathcal{M}(\mathfrak{v})$.

The MDM implementation using the formulation (4.14)–(4.16) does not require access to any subsets or supersets. For each nonempty $\mathfrak{v} \subseteq \mathcal{U}_\varepsilon^{\text{ext}}$, each position $\mathfrak{w} \in \mathcal{M}(\mathfrak{v})$ and for the different m, with $c(\mathfrak{v}, \mathfrak{w}, m) \neq 0$, the sums $S_{\mathfrak{v},\mathfrak{w},m}(f)$ are over disjoint sets of QMC points. In this way we will only evaluate each function $f(\cdot_{\mathfrak{w}\to\mathfrak{v}}; \mathbf{0})$ at each quadrature point once. An implementation of the MDM algorithm with QMC quadratures is given in Pseudocode 4.3B.

Pseudocode 4.2B Constructing the extended active set for QMC

Initialise $\mathcal{U}_\varepsilon^{\text{ext}} \leftarrow \mathcal{U}_\varepsilon$ ▷ start from the active set
Initialise $c_\emptyset \leftarrow 1$ ▷ for $\mathfrak{u} = \emptyset$
for $\emptyset \neq \mathfrak{u} \in \mathcal{U}_\varepsilon$ with $|\mathfrak{u}|$ from 1 to $\sigma(\mathcal{U}_\varepsilon)$ **do** ▷ traverse in increasing cardinality
    Calculate $m_\mathfrak{u}$ ▷ formula for $m_\mathfrak{u}$ is given from theory
    Update $c_\emptyset \leftarrow c_\emptyset + (-1)^{|\mathfrak{u}|}$
    Initialise $\mathcal{M}(\mathfrak{u}) \leftarrow \emptyset$
    for $\emptyset \neq \mathfrak{v} \subseteq \mathfrak{u}$ **do** ▷ generate nonempty subsets
        if $\mathfrak{v} \notin \mathcal{U}_\varepsilon^{\text{ext}}$ **then** ▷ look up and add missing subset
            Add $\mathfrak{v}$ to $\mathcal{U}_\varepsilon^{\text{ext}}$
            Initialise $\mathcal{M}(\mathfrak{v}) \leftarrow \emptyset$
        end if
        Set $\mathfrak{w} \leftarrow \mathfrak{u}|\mathfrak{v}$ ▷ identify the position where $\mathfrak{v}$ originates from $\mathfrak{u}$
        if $\mathfrak{w} \notin \mathcal{M}(\mathfrak{v})$ **then** ▷ look up and add missing position
            Add $\mathfrak{w}$ to $\mathcal{M}(\mathfrak{v})$
            Initialise $c(\mathfrak{v}, \mathfrak{w}, m) \leftarrow 0$ for m from 1 to $m_{\max}$
        end if
        for m from 0 to $m_\mathfrak{u}$ **do** ▷ update relevant entries
            Update $c(\mathfrak{v}, \mathfrak{w}, m) \leftarrow c(\mathfrak{v}, \mathfrak{w}, m) + (-1)^{|\mathfrak{u}|-|\mathfrak{v}|} \times 2^{m_{\max}-m_\mathfrak{u}}$
        end for
    end for
end for

Pseudocode 4.3B QMC MDM

Initialise $\mathcal{A}_\varepsilon^{\text{Q}}(f) \leftarrow c_\emptyset \times f(\mathbf{0})$
for $\emptyset \neq \mathfrak{v} \in \mathcal{U}_\varepsilon^{\text{ext}}$ **do**
    for $\mathfrak{w} \in \mathcal{M}(\mathfrak{v})$ **do**
        for m from 0 to $m_{\max}$ **do**
            if $c(\mathfrak{v}, \mathfrak{w}, m) \neq 0$ **then**
                Calculate $S_{\mathfrak{v},\mathfrak{w},m}(f)$ using (4.14)
                Update $\mathcal{A}_\varepsilon^{\text{Q}}(f) \leftarrow \mathcal{A}_\varepsilon^{\text{Q}}(f) + c(\mathfrak{v}, \mathfrak{w}, m) \times S_{\mathfrak{v},\mathfrak{w},m}(f)/2^{m_{\max}}$
            end if
        end for
    end for
end for
return $\mathcal{A}_\varepsilon^{\text{Q}}(f)$

4.3 Two implementations of Smolyak MDM

Here we compare two approaches to implement Smolyak quadrature in the context of MDM: the direct implementation and the combination technique.

4.3.1 Direct Smolyak implementation

From a practical point of view, it is more useful to write Smolyak's method as an explicit weighted quadrature rule as opposed to the tensor product form (4.7), see, e.g., [27]. We summarise this formulation below and provide the derivation in Appendix 4.A for completeness.

For each one-dimensional rule U_i, let n_i denote the number of quadrature points, $(w_{i,k})_{k=0}^{n_i-1}$ the quadrature weights, and $(t_{i,k})_{k=0}^{n_i-1}$ the quadrature nodes. Here for simplicity of notation we present the formula for an s-dimensional rule, with $s \geq 1$, which would need to be mapped to the set $\mathfrak{v}$ appropriately. The formula depends on whether the quadrature rules are nested, i.e., whether U_i includes all the quadrature points from U_{i-1}.

Non-nested case: For non-nested one-dimensional rules, Smolyak's method can be written explicitly as

$$\mathcal{Q}_{s,m}(g) \;=\; \sum_{\substack{\boldsymbol{i}\in\mathbb{N}^s \\ s\leq|\boldsymbol{i}|\leq s+m-1}} \sum_{k_1=0}^{n_{(i_1)}-1} \cdots \sum_{k_s=0}^{n_{(i_s)}-1} w_{\boldsymbol{i},\boldsymbol{k}}\, g(\boldsymbol{t}_{\boldsymbol{i},\boldsymbol{k}})\,, \tag{4.17}$$

where the quadrature point $\boldsymbol{t}_{\boldsymbol{i},\boldsymbol{k}} \in \mathcal{Y}^s$ has coordinates $(\boldsymbol{t}_{\boldsymbol{i},\boldsymbol{k}})_j = t_{i_j,k_j}$ for $j = 1, 2, \ldots, s$ and

$$w_{\boldsymbol{i},\boldsymbol{k}} \;=\; \sum_{\substack{\boldsymbol{p}\in\{0,1\}^s \\ s\leq|\boldsymbol{i}+\boldsymbol{p}|\leq s+m-1}} \prod_{j=1}^{s} \left((-1)^{p_j}\, w_{i_j,k_j}\right)\,. \tag{4.18}$$

Nested case: When $\{U_i\}$ are nested, we assume that the quadrature points and weights are ordered such that at level i the new points occur at the end of the point set, from index n_{i-1} onwards. That is, for all $i \in \mathbb{N}$ we have $t_{i,k} = t_{i+1,k}$ for $k = 0, 1, \ldots n_i - 1$. Then, to ensure that the function is only evaluated at each node once, (4.7) can be rewritten as

$$\mathcal{Q}_{s,m}(g) \;=\; \sum_{\substack{\boldsymbol{i}\in\mathbb{N}^s \\ s\leq|\boldsymbol{i}|\leq s+m-1}} \sum_{k_1=n_{(i_1-1)}}^{n_{(i_1)}-1} \cdots \sum_{k_s=n_{(i_s-1)}}^{n_{(i_s)}-1} w_{\boldsymbol{i},\boldsymbol{k}}\, g(\boldsymbol{t}_{\boldsymbol{i},\boldsymbol{k}})\,, \tag{4.19}$$

with weights

$$w_{\boldsymbol{i},\boldsymbol{k}} \;=\; \sum_{\substack{\boldsymbol{q}\in\mathbb{N}^s,\,\boldsymbol{q}\geq\boldsymbol{i}\\ s\leq|\boldsymbol{q}|\leq s+m-1}} \prod_{j=1}^{s} \left(w_{q_j,k_j} - w_{q_j-1,k_j}\right)\,, \tag{4.20}$$

where we set $w_{0,k} \equiv 0$ for all $k \geq 0$ and $w_{q,k} \equiv 0$ when $k \geq n_q$. In particular, when $q_j = i_j$ in (4.20) the weight that is subtracted is 0, that is, $w_{q_j-1,k_j} = w_{i_j-1,k_j} = 0$, since in (4.19) $k_j \geq n_{i_j-1}$.

4.3.2 Smolyak quadrature via the combination technique

The combination technique (it combines different straightforward tensor product rules, hence the name) provides an alternative formulation to (4.7) as follows (for a derivation see Appendix 4.B or, e.g., [39, 91])

$$\begin{aligned}
\mathcal{Q}_{s,m}(g) \;&=\; \sum_{\substack{\boldsymbol{i}\in\mathbb{N}^s\\ \max(m,s)\leq|\boldsymbol{i}|\leq s+m-1}} (-1)^{s+m-1-|\boldsymbol{i}|}\binom{s-1}{|\boldsymbol{i}|-m}\left(\bigotimes_{j=1}^{s} U_{i_j}\right)(g)\\
&=\; \sum_{r=\max(m-s+1,1)}^{m} (-1)^{m-r}\binom{s-1}{s+r-1-m} \sum_{\substack{\boldsymbol{i}\in\mathbb{N}^s\\ |\boldsymbol{i}|=s+r-1}} \left(\bigotimes_{j=1}^{s} U_{i_j}\right)(g)\\
&=\; \sum_{r=\max(m-s+1,1)}^{m} (-1)^{m-r}\binom{s-1}{m-r} \sum_{\substack{\boldsymbol{i}\in\mathbb{N}^s\\ |\boldsymbol{i}|=s+r-1}} \left(\bigotimes_{j=1}^{s} U_{i_j}\right)(g)\,,
\end{aligned} \tag{4.21}$$

where we have simplified using the symmetry of the binomial coefficient: $\binom{n}{k} = \binom{n}{n-k}$. We adopt the usual convention that $\binom{0}{0} \coloneqq 1$.

Using (4.21), we can now rewrite the MDM algorithm from the second equality in (4.9) as

$$\begin{aligned}
\mathcal{A}_{\varepsilon}^{\mathrm{C}}(f) \;&=\; c_{\emptyset}\, f(\mathbf{0}) + \sum_{\emptyset\neq\mathfrak{v}\in\mathcal{U}_{\varepsilon}^{\mathrm{ext}}} \sum_{\substack{\mathfrak{u}\in\mathcal{U}_{\varepsilon}\\ \mathfrak{u}\supseteq\mathfrak{v}}} (-1)^{|\mathfrak{u}|-|\mathfrak{v}|}\\
&\qquad\times \sum_{m=\max(m_{\mathfrak{u}}-|\mathfrak{v}|+1,1)}^{m_{\mathfrak{u}}} (-1)^{m_{\mathfrak{u}}-m}\binom{|\mathfrak{v}|-1}{m_{\mathfrak{u}}-m} \sum_{\substack{\boldsymbol{i}_{\mathfrak{v}}\in\mathbb{N}^{|\mathfrak{v}|}\\ |\boldsymbol{i}_{\mathfrak{v}}|=|\mathfrak{v}|+m-1}} \left(\bigotimes_{j\in\mathfrak{v}} U_{i_j}\right)(f(\cdot_{\mathfrak{v}};\mathbf{0}))\\
&=\; c_{\emptyset}\, f(\mathbf{0}) + \sum_{\emptyset\neq\mathfrak{v}\in\mathcal{U}_{\varepsilon}^{\mathrm{ext}}} \sum_{\substack{m=1\\ \widetilde{c}(\mathfrak{v},m)\neq 0}}^{m_{\max}} \widetilde{c}(\mathfrak{v},m)\,\widetilde{\mathcal{Q}}_{\mathfrak{v},m}(f(\cdot_{\mathfrak{v}};\mathbf{0}))\,,
\end{aligned} \tag{4.22}$$

where for a nonempty set $\mathfrak{v}$ and $m = 1, \ldots, m_{\max}$ we define

$$\widetilde{\mathcal{Q}}_{\mathfrak{v},m}(g_{\mathfrak{v}}) \coloneqq \sum_{\substack{\boldsymbol{i}_{\mathfrak{v}} \in \mathbb{N}^{|\mathfrak{v}|} \\ |\boldsymbol{i}_{\mathfrak{v}}| = |\mathfrak{v}| + m - 1}} \left(\bigotimes_{j \in \mathfrak{v}} U_{i_j} \right) (g_{\mathfrak{v}}) \,, \tag{4.23}$$

and

$$\widetilde{c}(\mathfrak{v}, m) \coloneqq \sum_{\substack{\mathfrak{u} \in \mathcal{U}_\varepsilon,\, \mathfrak{u} \supseteq \mathfrak{v} \\ m_{\mathfrak{u}} - |\mathfrak{v}| + 1 \le m \le m_{\mathfrak{u}}}} (-1)^{|\mathfrak{u}| - |\mathfrak{v}| + m_{\mathfrak{u}} - m} \binom{|\mathfrak{v}| - 1}{m_{\mathfrak{u}} - m} . \tag{4.24}$$

The formulation (4.22)–(4.24) is very similar to the formulation (4.9)–(4.10), and the computation of the values $\widetilde{c}(\mathfrak{v}, m)$ can also be done while constructing the extended active set. This is shown in Pseudocode 4.2A′, which works in a similar way to Pseudocode 4.2A. The key change is that Step 12 of Pseudocode 4.2A is replaced by Steps 12–14 of Pseudocode 4.2A′.

Pseudocode 4.2A′ Constructing the extended active set for Smolyak with combination technique

1: Initialise $\mathcal{U}_\varepsilon^{\text{ext}} \leftarrow \mathcal{U}_\varepsilon$ ▷ start from the active set
2: Initialise $c_\emptyset \leftarrow 1$ ▷ for $\mathfrak{u} = \emptyset$
3: **for** $\emptyset \neq \mathfrak{u} \in \mathcal{U}_\varepsilon$ with $|\mathfrak{u}|$ from 1 to $\sigma(\mathcal{U}_\varepsilon)$ **do** ▷ traverse in increasing cardinality
4: Calculate $m_{\mathfrak{u}}$ ▷ formula for $m_{\mathfrak{u}}$ is given from theory
5: Update $c_\emptyset \leftarrow c_\emptyset + (-1)^{|\mathfrak{u}|}$
6: Initialise $\widetilde{c}(\mathfrak{u}, m) \leftarrow 0$ for m from 1 to $m_{\max}$
7: **for** $\emptyset \neq \mathfrak{v} \subseteq \mathfrak{u}$ **do** ▷ generate nonempty subsets
8: **if** $\mathfrak{v} \notin \mathcal{U}_\varepsilon^{\text{ext}}$ **then** ▷ look up and add missing subset
9: Add $\mathfrak{v}$ to $\mathcal{U}_\varepsilon^{\text{ext}}$
10: Initialise $\widetilde{c}(\mathfrak{v}, m) \leftarrow 0$ for m from 1 to $m_{\max}$
11: **end if**
12: **for** m from $m_{\mathfrak{u}} - |\mathfrak{v}| + 1$ to $m_{\mathfrak{u}}$ **do** ▷ update relevant entries
13: Update $\widetilde{c}(\mathfrak{v}, m) \leftarrow \widetilde{c}(\mathfrak{v}, m) + (-1)^{|\mathfrak{u}| - |\mathfrak{v}| + m_{\mathfrak{u}} - m} \binom{|\mathfrak{v}| - 1}{m_{\mathfrak{u}} - m}$
14: **end for**
15: **end for**
16: **end for**

The quantity $\widetilde{\mathcal{Q}}_{\mathfrak{v},m}(g_{\mathfrak{v}})$ is a straightforward tensor product quadrature rule

$$\widetilde{\mathcal{Q}}_{s,m}(g) \coloneqq \sum_{\substack{\boldsymbol{i}\in\mathbb{N}^s \\ |\boldsymbol{i}|=s+m-1}} \sum_{k_1=0}^{n_{(i_1)}-1} \cdots \sum_{k_s=0}^{n_{(i_s)}-1} w_{\boldsymbol{i},\boldsymbol{k}}\, g(\boldsymbol{t}_{\boldsymbol{i},\boldsymbol{k}})\,, \tag{4.25}$$

with $w_{\boldsymbol{i},\boldsymbol{k}} = \prod_{j=1}^{s} w_{i_j,k_j}$ and $(\boldsymbol{t}_{\boldsymbol{i},\boldsymbol{k}})_j = t_{i_j,k_j}$ for $j = 1,\ldots,s$. So the implementation of the Smolyak MDM algorithm using the combination technique can be obtained analogously by modifying Pseudocode 4.3A, shown in Pseudocode 4.3A′. The essential changes are in Steps 5 and 6.

Pseudocode 4.3A′ Smolyak MDM with combination technique

1: Initialise $\mathcal{A}_\varepsilon^{\mathrm{C}}(f) \leftarrow c_\emptyset \times f(\mathbf{0})$
2: **for** $\emptyset \neq \mathfrak{v} \in \mathcal{U}_\varepsilon^{\mathrm{ext}}$ **do**
3: **for** m from 1 to $m_{\max}$ **do**
4: **if** $\widetilde{c}(\mathfrak{v},m) \neq 0$ **then**
5: Calculate $\widetilde{\mathcal{Q}}_{\mathfrak{v},m}(f(\cdot_{\mathfrak{v}};\mathbf{0}))$ using (4.25)
6: Update $\mathcal{A}_\varepsilon^{\mathrm{C}}(f) \leftarrow \mathcal{A}_\varepsilon^{\mathrm{C}}(f) + \widetilde{c}(\mathfrak{v},m) \times \widetilde{\mathcal{Q}}_{\mathfrak{v},m}(f(\cdot_{\mathfrak{v}};\mathbf{0}))$
7: **end if**
8: **end for**
9: **end for**
10: **return** $\mathcal{A}_\varepsilon^{\mathrm{C}}(f)$

4.3.3 Direct Smolyak vs. combination technique

Here we compare the computational cost between the direct Smolyak implementation $\mathcal{A}_\varepsilon^{\mathrm{S}}$ given by (4.9)–(4.10) and the combination technique implementation $\mathcal{A}_\varepsilon^{\mathrm{C}}$ given by (4.22)–(4.24). Throughout, we will use the notation $\mathrm{cost}(\cdot)$ to denote the whole cost, $\#(\cdot)$ to denote the number of function evaluations, and $\$(|\mathfrak{v}|)$ to denote the cost of evaluating the original integrand f at some anchored point $(\boldsymbol{y}_{\mathfrak{v}};\mathbf{0})$.

The cost of the direct Smolyak implementation is

$$\mathrm{cost}(\mathcal{A}_\varepsilon^{\mathrm{S}}) = \sum_{\emptyset\neq\mathfrak{v}\in\mathcal{U}_\varepsilon^{\mathrm{ext}}} \sum_{\substack{m=1 \\ c(\mathfrak{v},m)\neq 0}}^{m_{\max}} \mathrm{cost}\Big(\mathcal{Q}_{\mathfrak{v},m}(f(\cdot_{\mathfrak{v}};\mathbf{0}))\Big)\,.$$

Similarly for the combination technique the cost is

$$\mathrm{cost}(\mathcal{A}_\varepsilon^{\mathrm{C}}) = \sum_{\emptyset\neq\mathfrak{v}\in\mathcal{U}_\varepsilon^{\mathrm{ext}}} \sum_{\substack{m=1 \\ \widetilde{c}(\mathfrak{v},m)\neq 0}}^{m_{\max}} \mathrm{cost}\Big(\widetilde{\mathcal{Q}}_{\mathfrak{v},m}(f(\cdot_{\mathfrak{v}};\mathbf{0}))\Big)\,.$$

We expect that $\widetilde{c}(\mathfrak{v}, m)$ will be nonzero more often than $c(\mathfrak{v}, m)$ would be nonzero, so that the combination technique approach needs to compute more quadrature approximations. We need to account for both the cost to construct the quadrature weights and the cost of function evaluations.

In terms of the number of function evaluations, we have for the direct Smolyak implementation

$$\#(\mathcal{Q}_{s,m}) = \begin{cases} \displaystyle\sum_{|\boldsymbol{i}|\le s+m-1} \prod_{j=1}^{s} n_{i_j} & \text{if non-nested,} \\ \displaystyle\sum_{|\boldsymbol{i}|\le s+m-1} \prod_{j=1}^{s} (n_{i_j} - n_{i_j-1}) & \text{if nested.} \end{cases} \tag{4.26}$$

For the combination technique, we have

$$\begin{aligned} \#(\widetilde{\mathcal{Q}}_{s,m}) &= \sum_{|\boldsymbol{i}|=s+m-1} \prod_{j=1}^{s} n_{i_j} \\ &= \sum_{|\boldsymbol{i}|=s+m-1} \prod_{j=1}^{s} \left(\sum_{p_j=1}^{i_j} n_{p_j} - n_{p_j-1} \right) \\ &= \sum_{|\boldsymbol{i}|=s+m-1} \sum_{\boldsymbol{p}\in\mathbb{N}^s,\, \boldsymbol{p}\le\boldsymbol{i}} \prod_{j=1}^{s} (n_{p_j} - n_{p_j-1}) \\ &= \sum_{|\boldsymbol{p}|\le s+m-1} \binom{2s+m-|\boldsymbol{p}|-2}{s-1} \prod_{j=1}^{s} (n_{p_j} - n_{p_j-1}). \end{aligned}$$

So $\#(\mathcal{Q}_{s,m}^{\text{nested}}) \le \#(\widetilde{\mathcal{Q}}_{s,m}) \le \#(\mathcal{Q}_{s,m}^{\text{non-nested}})$.

For the direct Smolyak implementation we note that (4.17) (or (4.19) in the nested case) using the weights (4.18) (respectively (4.20)) is simply a grouping of all of the quadrature points, but the total collection of one-dimensional weights that need to be evaluated is the same as in (4.7); so the cost of computing the weights is $\sum_{|\boldsymbol{i}|\le s+m-1} s \prod_{j=1}^{s} (n_{i_j} + n_{i_j-1})$. On the other hand, the cost of computing the weights in (4.23) is clearly $\sum_{|\boldsymbol{i}|=|\mathfrak{v}|+m-1} |\mathfrak{v}| \prod_{j\in\mathfrak{v}} n_{i_j}$.

Thus for the direct Smolyak implementation we have

$$\text{cost}(\mathcal{Q}_{\mathfrak{v},m}) = \begin{cases} \displaystyle\sum_{|\boldsymbol{i}|\le|\mathfrak{v}|+m-1} \left(|\mathfrak{v}| \prod_{j\in\mathfrak{v}} (n_{i_j} + n_{i_j-1}) + \$(|\mathfrak{v}|) \prod_{j\in\mathfrak{v}} n_{i_j} \right) & \text{if non-nested,} \\ \displaystyle\sum_{|\boldsymbol{i}|\le|\mathfrak{v}|+m-1} \left(|\mathfrak{v}| \prod_{j\in\mathfrak{v}} (n_{i_j} + n_{i_j-1}) + \$(|\mathfrak{v}|) \prod_{j\in\mathfrak{v}} (n_{i_j} - n_{i_j-1}) \right) & \text{if nested,} \end{cases}$$

and for the combination technique

$$\operatorname{cost}(\widetilde{\mathcal{Q}}_{\mathfrak{v},m}) \;=\; \sum_{|\boldsymbol{i}|=|\mathfrak{v}|+m-1} \left(|\mathfrak{v}| \prod_{j\in\mathfrak{v}} n_{i_j} + \$(|\mathfrak{v}|) \prod_{j\in\mathfrak{v}} n_{i_j} \right) .$$

To summarise, the combination technique implementation is likely to require more quadrature approximations than the direct method (more nonzero $\widetilde{c}(\mathfrak{v},m)$ than $c(\mathfrak{v},m)$), and it uses more function evaluations than the direct method in the nested case, but the cost of computing the weights is cheaper. It is not immediately clear which method would be the overall winner.

4.4 MDM with randomised QMC

A randomly shifted version of the QMC approximation (4.11), see Section 2.3, takes the form

$$\frac{1}{n}\sum_{i=0}^{n-1} g\left(\{\boldsymbol{t}^{(i)}+\boldsymbol{\Delta}\}\right),$$

where $\boldsymbol{\Delta}\in[0,1)^s$ is the random shift with independent and uniformly distributed components $\Delta_j\in[0,1)$ for $j=1,\dots,s$, and the braces indicate that we take the fractional part of each component. Recall that a randomly shifted QMC method provides an unbiased estimate of the integral, and moreover, one can obtain a practical error estimate by using a number of independent random shifts.

In this section we outline how to implement randomised QMC (RQMC) versions of the QMC MDM from Section 4.2.2 by random shifting. One approach that comes to mind is to use a completely different set of independent shifts for each $A_{\mathfrak{u}}(f(\cdot_{\mathfrak{v}};\mathbf{0}))$ in (4.12). But with this approach none of the function evaluations can be reused. Another approach would be to use the same set of independent shifts for those $A_{\mathfrak{u}}(f(\cdot_{\mathfrak{v}};\mathbf{0}))$ with the same cardinality $|\mathfrak{u}|$, in a similar way to how the QMC method is applied to the MDM. But this approach also conflicts with our strategy to reuse function values based on the position where a subset $\mathfrak{v}$ originates from $\mathfrak{u}$. Furthermore, it is unclear in this case how to obtain a valid error estimate of the overall MDM algorithm because of the dependence between many shifts. So, instead of these two approaches, we will describe a third approach which allows for a valid error estimate and the complete reuse of function values as in the case of the non-randomised QMC MDM.

Our approach is to randomise the MDM algorithm itself, treating the integrand as a function of $\tau(\mathcal{U}_\varepsilon)$ variables, where we recall from Lemma 4.1 that $\tau(\mathcal{U}_\varepsilon)$ is the

truncation dimension, i.e., the largest index of the variables that appear in the active set. We need R independent random shifts $\boldsymbol{\Delta}^{(1)}, \dots, \boldsymbol{\Delta}^{(R)} \in [0,1)^{\tau(\mathcal{U}_\varepsilon)}$, and we take

$$\mathcal{A}_\varepsilon^{\mathrm{R}}(f) \;=\; \frac{1}{R} \sum_{r=1}^{R} \mathcal{A}_{\varepsilon,r}(f)\,, \tag{4.27}$$

where each $\mathcal{A}_{\varepsilon,r}(f)$ is a shifted QMC MDM algorithm analogous to (4.14)–(4.16),

$$\mathcal{A}_{\varepsilon,r}(f) \;\coloneqq\; c_\emptyset\, f(\mathbf{0}) + \sum_{\emptyset \neq \mathfrak{v} \in \mathcal{U}_\varepsilon^{\mathrm{ext}}} \sum_{\mathfrak{w} \in \mathcal{M}(\mathfrak{v})} \sum_{\substack{m=0 \\ c(\mathfrak{v},\mathfrak{w},m) \neq 0}}^{m_{\max}} c(\mathfrak{v},\mathfrak{w},m)\, \frac{\widetilde{S}^{(r)}_{\mathfrak{v},\mathfrak{w},m}(f)}{2^{m_{\max}}}\,, \tag{4.28}$$

but now with function values shifted by $\boldsymbol{\Delta}^{(r)}$ in

$$\widetilde{S}^{(r)}_{\mathfrak{v},\mathfrak{w},m}(f) \;\coloneqq\; \sum_{i=\lfloor 2^{m-1} \rfloor}^{2^m-1} f\left(\left\{\boldsymbol{t}^{(i)}_{\mathfrak{w}\to\mathfrak{v}} + \boldsymbol{\Delta}^{(r)}_{\mathfrak{v}}\right\}; \mathbf{0}\right). \tag{4.29}$$

Since $\mathcal{A}_{\varepsilon,1}(f), \dots, \mathcal{A}_{\varepsilon,R}(f)$ are independent random variables each with the same mean $\sum_{\mathfrak{u}\in\mathcal{U}_\varepsilon} \mathcal{I}_{\mathfrak{u}}(f_{\mathfrak{u}})$, their average $\mathcal{A}_\varepsilon^{\mathrm{R}}(f)$ also has the same mean. Moreover, the variance of $\mathcal{A}_\varepsilon^{\mathrm{R}}(f)$ is $1/R$ times the variance of each $\mathcal{A}_{\varepsilon,r}(f)$. The quadrature component of the root-mean-square error is given by $\sqrt{\mathbb{E}_{\boldsymbol{\Delta}}\,|\sum_{\mathfrak{u}\in\mathcal{U}_\varepsilon} \mathcal{I}_{\mathfrak{u}}(f_{\mathfrak{u}}) - \mathcal{A}_\varepsilon^{\mathrm{R}}(f)|^2}$ and can be estimated from these R sample values by

$$\sqrt{\frac{1}{R(R-1)} \sum_{r=1}^{R} \left(\mathcal{A}_\varepsilon^{\mathrm{R}}(f) - \mathcal{A}_{\varepsilon,r}(f)\right)^2} \;=\; \sqrt{\frac{1}{R(R-1)} \left(\sum_{r=1}^{R} \left(\mathcal{A}_{\varepsilon,r}(f)\right)^2 - R\left(\mathcal{A}_\varepsilon^{\mathrm{R}}(f)\right)^2\right)}.$$

The computation of $\mathcal{M}(\mathfrak{v})$, $c(\mathfrak{v},\mathfrak{w},m)$ and $\mathcal{U}_\varepsilon^{\mathrm{ext}}$ is exactly the same as in Pseudocode 4.2B. We only need to replace Pseudocode 4.3B for running the QMC MDM by Pseudocode 4.3B′, where we include also the error estimation.

Note that due to linearity in (4.27)–(4.28), we can also interpret $\mathcal{A}_\varepsilon^{\mathrm{R}}(f)$ as one MDM algorithm where each of the quadrature approximations $A_{\mathfrak{u}}$ is obtained by the average of R randomly shifted QMC rules. In this case we can also estimate the root-mean-square error corresponding to each $\mathfrak{u}$. However, note that the shifts between different sets $\mathfrak{u}$ are correlated and an estimate for the total variance cannot be obtained by simply summing up the individual variance estimates.

Pseudocode 4.3B′ RQMC MDM

Initialise $\mathcal{A}_{\varepsilon}^{\mathrm{R}}(f) \leftarrow 0$
Initialise $E \leftarrow 0$
for r from 1 to R **do** ▷ compute MDM for each shift
  Generate $\boldsymbol{\Delta}^{(r)}$ independently and uniformly from $[0,1)^{\tau^*}$
  Initialise $\mathcal{A}_{\varepsilon,r}(f) \leftarrow c_{\emptyset} \times f(\mathbf{0})$
  for $\emptyset \neq \mathfrak{v} \in \mathcal{U}_{\varepsilon}^{\mathrm{ext}}$ **do**
    for $\mathfrak{w} \in \mathcal{M}(\mathfrak{v})$ **do**
      for m from 0 to $m_{\max}$ **do**
        if $c(\mathfrak{v},\mathfrak{w},m) \neq 0$ **then**
          Calculate $\widetilde{S}^{(r)}_{\mathfrak{v},\mathfrak{w},m}(f)$ using (4.29)
          Update $\mathcal{A}_{\varepsilon,r}(f) \leftarrow \mathcal{A}_{\varepsilon,r}(f) + c(\mathfrak{v},\mathfrak{w},m) \times \widetilde{S}^{(r)}_{\mathfrak{v},\mathfrak{w},m}(f)/2^{m_{\max}}$
        end if
      end for
    end for
  end for
  Update $\mathcal{A}_{\varepsilon}^{\mathrm{R}}(f) \leftarrow \mathcal{A}_{\varepsilon}^{\mathrm{R}}(f) + \mathcal{A}_{\varepsilon,r}(f)$ ▷ sum up from different shifts
  Update $E \leftarrow E + (\mathcal{A}_{\varepsilon,r}(f))^2$ ▷ estimate error from different shifts
end for
Compute $\mathcal{A}_{\varepsilon}^{\mathrm{R}}(f) \leftarrow \mathcal{A}_{\varepsilon}^{\mathrm{R}}(f)/R$ ▷ final MDM approximation
Compute $E \leftarrow \sqrt{[E - R \times (\mathcal{A}_{\varepsilon}^{\mathrm{R}}(f))^2]/[R(R-1)]}$ ▷ final error estimate
return $\mathcal{A}_{\varepsilon}^{\mathrm{R}}(f)$ and E

4.5 Computing the threshold T

In the theoretical setting of the paper [58] outlined in Section 2.5, recall that the threshold parameter T to construct an active set (4.4) takes the form (see (2.34))

$$T = \left(\frac{\varepsilon/2}{\sum_{\mathfrak{u} \subset \mathbb{N}} [w(\mathfrak{u})]^{1/\alpha}} \right)^{\alpha/(\alpha-1)}, \tag{4.30}$$

where $\varepsilon > 0$ is a given error tolerance parameter, while $\alpha > 1$ is a free parameter with the constraint that $\sum_{\mathfrak{u} \subset \mathbb{N}} [w(\mathfrak{u})]^{1/\alpha} < \infty$.

In a practical implementation, we need to estimate the infinite sum in the denominator *from above*, to yield an underestimate of T, so that the required theoretical error tolerance ε is guaranteed, at the expense of enlarging the active set. Understandably, we should aim for a tight estimate of the infinite sum to avoid making the active set too large and thus the algorithm too expensive. Similarly, the free

parameter α should be chosen so as to make the threshold parameter T as large as possible.

Here we derive upper bounds on the infinite sum by assuming further structure in $w(\mathfrak{u})$, namely, that the POD form of $w(\mathfrak{u})$, see (4.5), is further parametrised for $\ell \geq 0$ and $j \geq 1$ by

$$\Omega_\ell \;=\; c_1(\ell!)^{b_1} \quad \text{and} \quad \omega_j = c_2 j^{-b_2}, \tag{4.31}$$

with $b_1 \geq 0$, $b_2 > 1$, $b_2 > b_1$, and $c_1, c_2 > 0$. Thus

$$\begin{aligned}\sum_{\mathfrak{u}\subset\mathbb{N}}[w(\mathfrak{u})]^{1/\alpha} \;&=\; \sum_{\ell=0}^{\infty}\sum_{|\mathfrak{u}|=\ell}\left(c_1(\ell!)^{b_1}\prod_{j\in\mathfrak{u}}(c_2 j^{-b_2})\right)^{1/\alpha}\\ &=\; c_1^{1/\alpha}\sum_{\ell=0}^{\infty}(\ell!)^a\, c^\ell \sum_{|\mathfrak{u}|=\ell}\prod_{j\in\mathfrak{u}} j^{-b},\end{aligned} \tag{4.32}$$

with

$$a \;=\; \frac{b_1}{\alpha}, \quad b \;=\; \frac{b_2}{\alpha}, \quad \text{and} \quad c \;=\; c_2^{1/\alpha}. \tag{4.33}$$

We know from [58, Lemma 10] that the infinite sum in (4.32) is finite if $b > 1$ and $b > a$. In turn this means that the free parameter $\alpha > 1$ in (4.30) should satisfy $\alpha < b_2$.

In the next two lemmas we obtain an upper bound on the infinite sum in (4.32) for somewhat general parameters a, b, c, without taking into account the constraints in how they relate to each other or how they relate to α. After the two lemmas we will discuss when and how the lemmas can be applied in our case.

Lemma 4.3. *For any $b > 1$ and $\ell \geq 1$ we have*

$$\sum_{|\mathfrak{u}|=\ell}\prod_{j\in\mathfrak{u}} j^{-b} \;\leq\; \frac{z^{\ell-1}}{(\ell-1)!}\left(1+\frac{z}{\ell}\right), \qquad z \;\coloneqq\; \frac{(2/3)^{b-1}}{b-1}. \tag{4.34}$$

Proof. By identifying $\mathfrak{u}$ with the vector $(u_1, u_2, \ldots, u_\ell)$ with ordered elements $u_1 < u_2 < \cdots < u_\ell$, we see that

$$\sum_{|\mathfrak{u}|=\ell}\prod_{j\in\mathfrak{u}} j^{-b} \;=\; \sum_{u_1=1}^{\infty} u_1^{-b} \sum_{u_2=u_1+1}^{\infty} u_2^{-b} \cdots \sum_{u_{\ell-1}=u_{\ell-2}+1}^{\infty} u_{\ell-1}^{-b} \sum_{u_\ell=u_{\ell-1}+1}^{\infty} u_\ell^{-b}. \tag{4.35}$$

For any $k \in \mathbb{N}$ and $p > 1$ we have

$$\sum_{j=k+1}^{\infty} j^{-p} \leq \int_{k+1/2}^{\infty} x^{-p}\,\mathrm{d}x \;=\; \frac{(k+1/2)^{-(p-1)}}{p-1} \;<\; \frac{k^{-(p-1)}}{p-1},$$

which holds since the mid-point rule underestimates this integral. Therefore, the very last sum in (4.35) is bounded by

$$\int_{u_{\ell-1}+1/2}^{\infty} x^{-b}\,\mathrm{d}x \;=\; \frac{(u_{\ell-1}+1/2)^{-(b-1)}}{b-1} \;<\; \frac{u_{\ell-1}^{-(b-1)}}{b-1},$$

and in turn the last two sums in (4.35) are bounded by

$$\frac{1}{b-1}\int_{u_{\ell-2}+1/2}^{\infty} x^{-(2b-1)}\,\mathrm{d}x \;=\; \frac{(u_{\ell-2}+1/2)^{-2(b-1)}}{2(b-1)^2} \;<\; \frac{u_{\ell-2}^{-2(b-1)}}{2(b-1)^2}.$$

Similarly, the last $\ell-1$ sums are bounded by

$$\frac{(u_1+1/2)^{-(\ell-1)(b-1)}}{(\ell-1)!\,(b-1)^{\ell-1}}.$$

The sum with respect to u_1 has to be treated differently since we do not want to integrate over $[1/2,\infty)$. For that purpose, we treat differently $u_1=1$ and $u_1\geq 2$, to obtain

$$\begin{aligned}\sum_{|\mathfrak{u}|=\ell}\prod_{j\in\mathfrak{u}} j^{-b} \;&\leq\; \frac{1}{(\ell-1)!\,(b-1)^{\ell-1}}\sum_{u_1=1}^{\infty} u_1^{-b}(u_1+1/2)^{-(\ell-1)(b-1)}\\ &<\; \frac{(2/3)^{(\ell-1)(b-1)}}{(\ell-1)!(b-1)^{\ell-1}} + \frac{1}{(\ell-1)!\,(b-1)^{\ell-1}}\sum_{u_1=2}^{\infty} u_1^{-\ell\,b+(\ell-1)}.\end{aligned}$$

Applying the integration estimate again to the sum over $u_1\geq 2$, we get that

$$\frac{1}{(\ell-1)!(b-1)^{\ell-1}}\sum_{u_1=2}^{\infty} u_1^{-\ell\,b+(\ell-1)} \leq \frac{1}{\ell!(b-1)^{\ell}}\left(\frac{2}{3}\right)^{\ell(b-1)}.$$

Combining the last two steps yields the estimate in (4.34). □

Lemma 4.4. *For every $a\in(0,1)$, $b>1$, $c>0$, $s\in\mathbb{N}$ and $t\in(0,1)$, we have*

$$\sum_{\ell=0}^{\infty}(\ell!)^a\,c^{\ell}\sum_{|\mathfrak{u}|=\ell}\prod_{j\in\mathfrak{u}} j^{-b} \;\leq\; 1+\sum_{\ell=1}^{s}(\ell!)^a\,c^{\ell}\,\frac{z^{\ell-1}}{(\ell-1)!}\left(1+\frac{z}{\ell}\right)+E_{s,t},$$

with $z\coloneqq(2/3)^{b-1}/(b-1)$ as in (4.34), and

$$\begin{aligned}E_{s,t} \;\coloneqq\; & c\left(1+\frac{z}{s+1}\right)\left[\frac{t^{s/a}}{1-t^{1/a}}\left(s+\frac{1}{1-t^{1/a}}\right)\right]^a\\ &\times\left[\exp\left(\left(\frac{c\,z}{t}\right)^{1/(1-a)}\right)\min\left(1,\left(\frac{c\,z}{t}\right)^{s/(1-a)}\right)\frac{1}{s!}\right]^{1-a}. \qquad (4.36)\end{aligned}$$

Proof. The result is obtained by applying Lemma 4.3 and then estimating the tail sum from $\ell = s+1$ by $E_{s,t}$. Indeed, we have

$$\begin{aligned}
&\sum_{\ell=d+1}^{\infty} (\ell!)^a \, c^\ell \, \frac{z^{\ell-1}}{(\ell-1)!} \left(1 + \frac{z}{\ell}\right) \\
&\leq c \left(1 + \frac{z}{s+1}\right) \sum_{\ell=s+1}^{\infty} \ell^a \, t^{\ell-1} \, \left(\frac{c\,z}{t}\right)^{\ell-1} \frac{1}{[(\ell-1)!]^{1-a}} \\
&\leq c \left(1 + \frac{z}{s+1}\right) \left(\sum_{\ell=s+1}^{\infty} \ell \, t^{(\ell-1)/a}\right)^a \left(\sum_{\ell=s+1}^{\infty} \frac{1}{(\ell-1)!} \, \left(\frac{c\,z}{t}\right)^{(\ell-1)/(1-a)}\right)^{1-a},
\end{aligned}$$

where the last step follows from Hölder's inequality with the conjugate pair $1/a$ and $1/(1-a)$.

Consider the equality

$$\sum_{\ell=s+1}^{\infty} \ell \, x^{\ell-1} \;=\; \frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{x^{s+1}}{1-x}\right) \;=\; \frac{x^s}{1-x}\left(d + \frac{1}{1-x}\right),$$

which after substituting in $x = t^{1/a}$ yields the third factor in (4.36).

We also have

$$\sum_{\ell=s+1}^{\infty} \frac{y^{(\ell-1)/(1-a)}}{(\ell-1)!} \;=\; \frac{y^{s/(1-a)}}{s!} \sum_{\ell=0}^{s-1} \frac{y^{\ell/(1-a)}}{\ell!} \;\leq\; \exp(y^{1/(1-a)}) \, \min\left(1, \frac{y^{s/(1-a)}}{s!}\right),$$

with last inequality due to Taylor's theorem. Substituting $y = c\,z/t$ yields the fourth factor in (4.36). □

The following corollary summarises the upper bound on the sum. It introduces two new free parameters: $s \in \mathbb{N}$ and $t \in (0,1)$.

Corollary 4.5. *Let $w(\mathfrak{u})$ have product and order dependent components satisfying (4.31) with $b_2 > 0$, also let $\alpha \geq 1$ and $\alpha \in (b_1, b_2)$. Then for every $s \in \mathbb{N}$ and every $t \in (0,1)$ we have*

$$\sum_{\mathfrak{u} \subset \mathbb{N}} [w(\mathfrak{u})]^{1/\alpha} \leq c_1^{1/\alpha} \left(1 + \sum_{\ell=1}^{s} (\ell!)^a \, c^\ell \, \frac{z^{\ell-1}}{(\ell-1)!} \left(1 + \frac{z}{\ell}\right) + E_{s,t}\right), \tag{4.37}$$

where a, b, c are specified in (4.33), and z and $E_{s,t}$ are as defined in Lemma 4.4.

Note that when $\alpha = 1$ the upper bound (4.37) on the sum is valid, even though in this case the threshold T in (4.30) is undefined.

Remark 4.6. If $b_1 = 0$ so that $w(\mathfrak{u})$ is of product form, then $a = 0$ in (4.33) and the sum can be bounded above using the same method as in [33, equation (7)], namely, for every $s \in \mathbb{N}$,

$$\sum_{\mathfrak{u} \subset \mathbb{N}} [w(\mathfrak{u})]^{1/\alpha} \leq c_1^{1/\alpha} \exp\left(\frac{c}{(b-1)(s+\frac{1}{2})^{b-1}}\right) \prod_{j=1}^{s} \left(1 + c j^{-b}\right) ,$$

where b, c are as given in (4.33), see also (5.16) later. In this case the size of the active set will be smaller than the general case because with $b_1 = 0$ each $w(\mathfrak{u})$ is smaller so the exact value of the sum $\sum_{\mathfrak{u} \subset \mathbb{N}} [w(\mathfrak{u})]^{1/\alpha}$ is smaller, and moreover our upper bound on the sum is tighter.

Note that the threshold parameter in the construction of optimal and quasi-optimal active sets in Chapter 5 also requires a good upper bound on the sum with $\alpha = 1$.

4.6 Numerical experiments

Until now we have ignored any theoretical details of the MDM and focussed purely on the implementation given the arbitrary input parameters ε, $\{\omega(\mathfrak{u})\}_{\mathfrak{u} \subset \mathbb{N}}$, T and $\{m_{\mathfrak{u}}\}_{\mathfrak{u} \in \mathcal{U}_\varepsilon}$. Below we give some brief details on how to specify these parameter values for a particular test integrand following the setting of [58], which was briefly summarised in Section 2.5. We compare between QMC MDM and Smolyak MDM, as well as demonstrate the speedup of our efficient implementations over the naive implementations.

4.6.1 Test integrand

We consider the integrand $f : [-\frac{1}{2}, \frac{1}{2}]^{\mathbb{N}} \to \mathbb{R}$ given by

$$f(\boldsymbol{y}) = \frac{1}{1 + \sum_{j \geq 1} y_j / j^{\beta}} , \tag{4.38}$$

with different parameters $\beta \geq 2$, which was studied in Section 2.5.3 and previously in [58, Example 5] (for the case $\beta = 2$). Although we do not know the exact value of the integral, we are able to calculate a good approximation as a reference value and use this reference value to calculate the total error of our MDM algorithms.

In the theoretical setting outlined in Section 2.5, it was assumed that each decomposition function $f_{\mathfrak{u}}$ belongs to a Banach space $F_{\mathfrak{u}}$ with known bounds on their norms, and in Section 2.5.3 we studied the example integrand (4.38) in the particular case of the anchored decomposition combined with the anchored norm (2.30). Recall that in this setting

$$\|\mathcal{I}_{\mathfrak{u}}\| = 12^{-|\mathfrak{u}|/2} =: C_{\mathfrak{u}}$$

and the norms of the decomposition functions are bounded by

$$\|f_{\mathfrak{u}}\|_{F_{\mathfrak{u}}} \leq \left(1 - \tfrac{1}{2}\zeta(\beta)\right)^{-(|\mathfrak{u}|+1)} |\mathfrak{u}|! \prod_{j\in\mathfrak{u}} j^{-\beta} =: B_{\mathfrak{u}},$$

where $\zeta(x) = \sum_{k=1}^{\infty} k^{-x}$ for $x > 1$ is the Riemann zeta function.

4.6.2 Active set construction

Following Section 2.5, the above description leads to an active set (4.4) with $w(\mathfrak{u}) := C_{\mathfrak{u}} B_{\mathfrak{u}}$, which is in POD form (4.31) with

$$c_1 = \frac{1}{1 - \frac{1}{2}\zeta(\beta)}, \quad c_2 = \frac{c_1}{\sqrt{12}}, \quad b_1 = 1, \quad b_2 = \beta,$$

and where the threshold T is given by (4.30) and can be estimated using (4.37). In our computations we fix $s = 1000$, $t = 0.5$, and maximise the threshold T for α over 100 equispaced points in $(1, \beta)$.

Table 4.1 presents details on the active set for the parameter values $\beta = 4, 3, 2.5$ and $\varepsilon = 10^{-1}, 10^{-2}, 10^{-3}$. The rows are labelled as follows: ε is the error request, T is the computed threshold for the active set, $\sigma(\mathcal{U}_\varepsilon)$ is the superposition dimension, $\tau(\mathcal{U}_\varepsilon)$ is the maximum truncation dimension, and the last ten rows display the number of sets of each size in the active set.

	$\beta = 4$			$\beta = 3$			$\beta = 2.5$	
ε	1e-1	1e-2	1e-3	1e-1	1e-2	1e-3	1e-1	1e-2
T	1.4e-4	2.8e-6	6.4e-8	4.0e-6	3.6e-8	3.8e-10	1.5e-8	4.9e-11
$\sigma(\mathcal{U}_\varepsilon)$	3	4	5	5	6	7	8	10
$\tau(\mathcal{U}_\varepsilon)$	10	28	72	86	418	1907	2528	24724
size 1	9	26	68	76	370	1686	2019	19750
2	12	48	159	195	1285	7327	10077	126882
3	5	28	132	202	1828	13117	21996	354377
4	0	4	36	80	1234	11907	26258	559155
5	0	0	1	10	361	5578	17874	536133
6	0	0	0	0	32	1145	6513	313623
7	0	0	0	0	0	69	1088	106877
8	0	0	0	0	0	0	47	18582
9	0	0	0	0	0	0	0	1210
10	0	0	0	0	0	0	0	8

Table 4.1: Results from the active set construction for various β and ε.

We see that although there are many sets to consider in MDM, even in the hardest case with $\beta = 2.5$ and $\varepsilon = 10^{-2}$ we only ever deal with integrals up to 10 dimensions, with the highest coordinate considered being 24724.

Below we will restrict ourselves to the case $\beta = 3$, with error request down to $\varepsilon = 10^{-6}$.

Actually, the above approach from [58] for prescribing the parameters $w(\mathfrak{u})$ and T overestimates the truncation error for our test integrand and makes the active set much larger than necessary. A tighter truncation error estimate can be done for this test integrand using a Taylor series argument (see e.g., [58, Remark 13]) and this should yield better input parameters $w(\mathfrak{u})$ and T to our efficient MDM algorithms. Analysis on the best strategy to prescribe the active set parameters for any given practical integrand falls outside the scope of this thesis.

4.6.3 QMC MDM

For the randomised QMC MDM we use an extensible rank-1 lattice rule with generating vector

$$\begin{aligned}\boldsymbol{z} = (&1, 756581, 694385, 178383, 437131, 945527, 62405, 1079809,\\ &991997, 750785, 187845, 1666795, 491701, 1092667, 1279469,\\ &817683, 1946073, 1946073, 1530387, 686611, \ldots),\end{aligned}$$

with $n = 2^m$ points for any $m = 0, \ldots, 25$. It is constructed by a CBC algorithm as outlined in [12], but the search criterion was appropriately modified to match the norm (2.30). Also, we apply the tent-transform (see [45]) $x \mapsto 1 - |2x - 1|$ to each component of the shifted quadrature points in $[0, 1]$, and then translate it to $[-\frac{1}{2}, \frac{1}{2}]$.

As in Section 2.5.5, we assume that the quadrature error for each decomposition term $f_{\mathfrak{u}}$ is bounded above by $G_{\mathfrak{u},q}\,(n_{\mathfrak{u}} + 1)^{-q}\,\|f_{\mathfrak{u}}\|_{F_{\mathfrak{u}}}$ (see (2.37)) with appropriate constants $G_{\mathfrak{u},q}$ and q. Then according to Section 2.5.5 we choose the number of points $n_{\mathfrak{u}} \geq h_{\mathfrak{u}}$, with

$$h_{\mathfrak{u}} \;=\; \left(\frac{2}{\varepsilon}\sum_{\mathfrak{v}\in\mathcal{U}} \pounds(|\mathfrak{v}|)^{q/(q+1)}\,(G_{\mathfrak{v},q}\,B_{\mathfrak{v}})^{1/(q+1)}\right)^{1/q} \left(\frac{G_{\mathfrak{u},q}\,B_{\mathfrak{u}}}{\pounds(|\mathfrak{u}|)}\right)^{1/(q+1)}, \tag{4.39}$$

where $\pounds(|\mathfrak{u}|)$ is the cost of evaluating the decomposition term $f_{\mathfrak{u}}$.

Although the above formula for $h_{\mathfrak{u}}$ is precisely (4.39), see also [58, Formula (28)], how we specify the other parameters here will deviate from [58]. Based on (4.2) we set $\pounds(|\mathfrak{u}|) = \max(2^{|\mathfrak{u}|}|\mathfrak{u}|, 1)$. The constant $G_{\mathfrak{u},q}$ arising from the theoretical error bound is a significant overestimate, so instead we set $G_{\mathfrak{u},q} = 1$, which the numerical experiments below suggest is a good choice in practice. Also, in the theoretical setting with the norm (2.30) involving mixed first order derivatives, we expect only up to first order convergence, leading to the choice $q = 1$. However, it is known from

[45] that randomly-shifted and then tent-transformed lattice rules can achieve nearly second order convergence if the integrand has mixed second order derivatives; thus we take $q = 2$ instead, without formally switching the setting. Again the numerical results below further motivate this choice. Finally, since we use base-2 extensible lattice rules we set $n_{\mathfrak{u}} = 2^{m_{\mathfrak{u}}}$ with $m_{\mathfrak{u}} = \max(\lceil \log_2(h_{\mathfrak{u}}) \rceil, 0)$.

In Figure 4.1 on the left we plot the error request ε against the estimated standard error obtained using 8 random shifts, which provides an estimate of the quadrature error only. The dashed line is the expected rate and is proportional to ε, i.e., it has slope -1. Notice that for $q = 1$ the standard error decays at a much faster rate than the error request ε, whereas for $q = 2$ the error request ε and the estimated standard error agree up to a constant factor. In short, setting the parameter to be $q = 1$ underestimates the convergence and forces the algorithm to do more work than is necessary, suggesting that $q = 2$ is the appropriate choice. For $q = 2$, is is possible to tune this constant offset factor by changing the value of $G_{\mathfrak{u},q}$ (which should be at least as large as $\|\mathcal{I}_{\mathfrak{u}}\| = 12^{-|\mathfrak{u}|/2}$ and indeed the theoretical bound yields $G_{\mathfrak{u},q} = g^{|\mathfrak{u}|}$ with some $g > 1$), but taking $G_{\mathfrak{u},q} = 1$ appears to give reasonable results in practice. Similar to the case with q, if it is too big then the quadrature rules will do too much work, while if it is too small they will underperform.

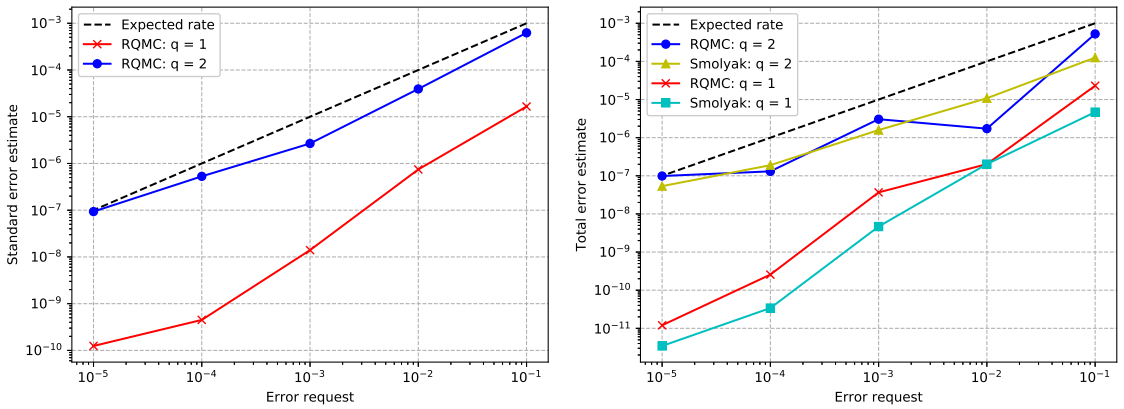


Figure 4.1: Error request ε against estimated standard error (left) and total error (right).

In Figure 4.1 on the right we plot the error request ε against the estimated total error (combining both the truncation error and the quadrature error) by comparing the results with a reference solution obtained using a higher precision QMC quadrature rule. The reference solution was computed using 2^{22} QMC points and 16 shifts in quad precision in 600 dimensions and resulted in a standard error of 8×10^{-13}. A similar computation in 800 dimensions with 2^{20} QMC points agreed up to 10 digits.

The two graphs in Figure 4.1 show the same trend, indicating that the truncation error is no worse than the quadrature error.

4.6.4 Smolyak MDM

For the Smolyak MDM we will use the composite trapezoidal rule as the one-dimensional rules, which we note are nested. As always, we set $U_0 \coloneqq 0$ to be the zero algorithm, and then following [27, Section 3] we take the first one-dimensional quadrature rule U_1 to be the single (mid-)point rule, i.e., $n_1 \coloneqq 1$ with point $t_{1,0} \coloneqq 0$ and weight $w_{1,0} \coloneqq 1$. This extra level ensures that the number of points does not grow too quickly. Then for each $i \geq 2$ we take the one-dimensional quadrature rule U_i to be the composite trapezoidal rule in $[-\frac{1}{2}, \frac{1}{2}]$ with $n_i \coloneqq 2^{i-1} + 1$ points at multiples of $1/2^{i-1}$, with the weights being $1/2^{i-1}$ for the interior points and $1/2^i$ at the two end points $\pm 1/2$.

To ensure that we iterate the points in a nested fashion, we label the points in the order of 0, $\pm 1/2$, $\pm 1/4$, $\pm 1/8$, $\pm 3/8, \ldots$ and so on. Explicitly, for $i \geq 2$ we can write

$$t_{i,k} \coloneqq \begin{cases} k/2^p - 1/2 & \text{if } k \text{ is odd, and } p \text{ is such that } 2^{p-1} < k < 2^p, \\ -t_{i,k-1} & \text{if } k \text{ is even,} \end{cases}$$

with the corresponding weights

$$w_{i,k} \coloneqq \begin{cases} 1/2^i & \text{if } k = 1, 2, \\ 1/2^{i-1} & \text{if } k = 0, 3, 4, \ldots, n_i - 1\,. \end{cases}$$

We choose the approximation levels $m_{\mathfrak{u}}$ of our quadrature rules in the direct Smolyak MDM implementation to be such that the number of function evaluations, see (4.26) with nested points, is at least $h_{\mathfrak{u}}$ given by the formula (4.39), with the same definitions of $\mathcal{L}(|\mathfrak{u}|) = \max(2^{|\mathfrak{u}|}|\mathfrak{u}|, 1)$, $G_{\mathfrak{u},q} = 1$ and $q = 2$. The justification for taking $q = 2$ here is that composite trapezoidal rules are known to give second order convergence for sufficiently smooth integrands in one dimension, and recall from (2.18) that this convergence is transferred, modulo log-factors, to the Smolyak rule for multivariate integrands with sufficient smoothness. We use the same values of $m_{\mathfrak{u}}$ for the combination technique variant even though the actual numbers of function evaluations are higher.

In particular, our chosen values of $m_{\mathfrak{u}}$ mean that the naive implementations of Smolyak MDM and QMC MDM would use roughly the same number of function evaluations for each $f_{\mathfrak{u}}$. However, the actual number of function evaluations for our

efficient implementations would be lower and hence achieve the savings we aim for; see the timings in the next subsection.

In Figure 4.1 on the right we also plotted the error request ε against the estimated total error of the direct Smolyak MDM implementation for $q = 1, 2$ compared with the same reference solution as for QMC MDM. Again, the faster than expected rate of convergence when $q = 1$ combined with the fact that the $q = 2$ case converges at the expected rate, provides empirical evidence that for this example $q = 2$ is the appropriate choice for the Smolyak implementation also.

4.6.5 Timing results

In Table 4.2 we present results on the run-time of our efficient MDM implementations compared with the naive implementations for error request ε from 10^{-1} down to 10^{-6}. We include results for QMC MDM with a single random shift, direct Smolyak MDM, and Smolyak MDM based on combination technique (CT-Smolyak). We report the total error with respect to the same reference solution in each case, as well as the run-time in seconds. The QMC results can vary between runs depending on the random shift, but to ensure consistency between the naive and efficient RQMC implementations the same seed was used for the pseudorandom number generator.

$\beta = 3$, reference value = 1.1011984577041

		efficient MDM		naive MDM		
ε	method	total error	time (s)	total error	time (s)	speedup
1e-01	QMC	2.24e-04	0.0022	2.24e-04	0.0043	2.0
	Smolyak	3.21e-05	0.0035	3.21e-05	0.0096	2.7
	CT-Smolyak	3.21e-05	0.0035	3.21e-05	0.010	2.9
1e-02	QMC	1.24e-05	0.035	1.24e-05	0.088	2.5
	Smolyak	9.32e-06	0.046	9.32e-06	0.17	3.6
	CT-Smolyak	9.32e-06	0.049	9.32e-06	0.19	3.9
1e-03	QMC	1.47e-06	0.47	1.47e-06	1.45	3.1
	Smolyak	4.94e-07	0.56	4.94e-07	2.61	4.6
	CT-Smolyak	4.94e-07	0.64	4.94e-07	3.17	5.0
1e-04	QMC	7.39e-08	5.14	7.39e-08	19.99	3.9
	Smolyak	1.22e-08	6.14	1.22e-08	36.69	6.0
	CT-Smolyak	1.22e-08	7.44	1.22e-08	47.07	6.3
1e-05	QMC	2.77e-09	51.02	2.76e-09	244.06	4.8
	Smolyak	1.55e-09	61.53	1.54e-09	463.77	7.5
	CT-Smolyak	1.54e-09	79.79	1.54e-09	622.25	7.9
1e-06	QMC	7.56e-10	467.60	3.38e-10	2758.02	5.9
	Smolyak	4.80e-10	570.26	5.40e-10	5244.56	9.2
	CT-Smolyak	1.22e-09	762.98	5.37e-10	7342.49	9.6

Table 4.2: Timing comparisons between efficient and naive MDM implementations.

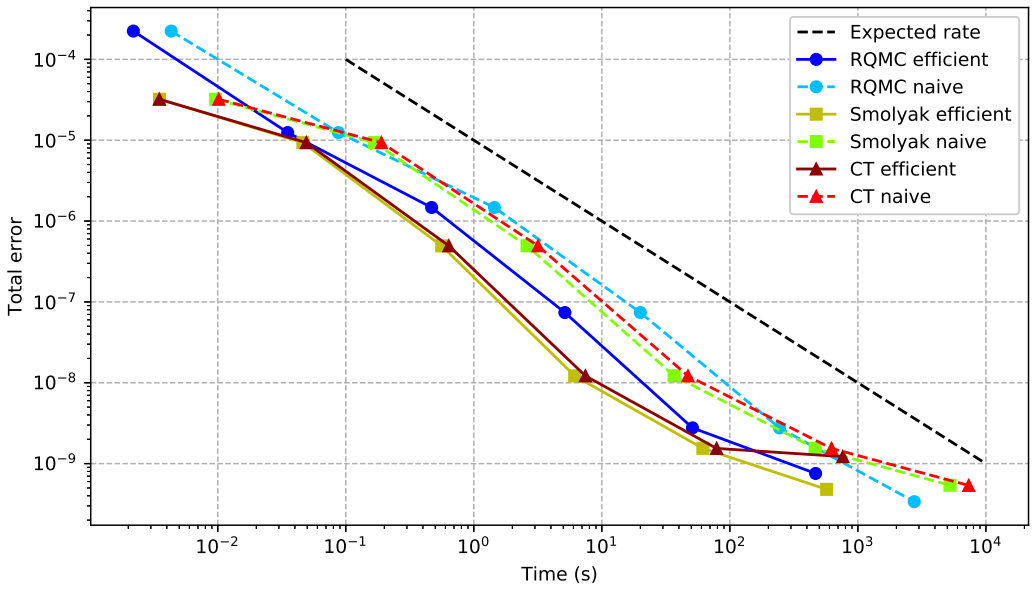


Figure 4.2: Estimated total error against time.

All calculations were done in x86 long double precision on a single node of the UNSW Katana cluster with an Intel Xeon X5675 3.07GHz CPU.

As ε decreases there is clearly an increasing speedup of the efficient implementations over the naive ones, for all three types of implementation. The reformulation of the MDM algorithm into this efficient formulation is the main result of this work.

We see more speedup in the case of Smolyak MDM compared with QMC MDM. This is as expected, because for QMC MDM there is extra work in managing the different positions that a nonempty set can originate from as a subset of another set in the active set: we need to cope with a more complicated data structure for (4.16) compared with (4.10) or (4.24), and we need more function evaluations. Additionally, we expect the QMC algorithms to be much more efficient when the truncation dimension goes up (i.e., when ε goes down), since the sizes of the Smolyak grids then increase much much faster than the powers of 2 of the QMC algorithms.

If we compare the direct Smolyak MDM with the combination technique Smolyak MDM then we notice that for our efficient reformulation the two methods have very similar running times, with a very minor advantage for the direct method, but with the naive formulation the direct method is the clear winner. Note that we can see the effect of rounding errors in the calculations for $\varepsilon = 10^{-5}$ and 10^{-6}, since the total errors should remain the same between the naive and efficient implementations of the same algorithm, while the direct Smolyak and combination technique should also have the same total errors.

In Figure 4.2 we plot the total error against time for the results in Table 4.2 to demonstrate the speedup of the efficient implementations. For each pair of efficient and naive results, we expect the data points to be at the same horizontal level (same total error) but with bigger and bigger gaps in time (the speedup factor increases) as the errors go down.

4.7 Conclusion

The MDM is a powerful algorithm for approximating integrals of ∞-variate functions, but care must be taken to ensure the implementation is efficient. In this chapter we have provided details and explicit pseudocodes explaining how to efficiently run all components of the algorithm: from constructing the active set to running the MDM for both randomised QMC rules and Smolyak quadrature rules. By reformulating the MDM we are able to save cost by reducing the amount of repeated function evaluations incurred because of the recursive structure of the anchored decomposition. We applied the MDM to an example integrand that possesses similar properties to those that arise in recent PDE problems with random coefficients. The numerical results clearly support the cost savings of the efficient reformulations.

Original research and my contribution

The work in this chapter corresponds to the paper [31], which was recently submitted to *SIAM Journal on Scientific Computing* and is available on arXiv: `https://arxiv.org/pdf/1712.06782.pdf`. It is joint work with Frances Kuo, Dirk Nuyens and Grzegorz Wasilkowski.

The MDM has existed in theory for some time, see [58, 63, 75, 76, 89, 90], however, the work in this chapter details the first implementations. Much of this project was dedicated to implementing the MDM in two separate pieces of software, Dirk Nuyens wrote an implementation in C++ and I wrote one in C. Furthermore, the goal of the project was to not only implement the MDM but to exploit structure in the algorithm to make the implementation as efficient as possible, which we achieve by the reformulations from Sections 4.2 – 4.4. These reformulations are a key contribution of this work and are also original.

Regarding my contributions, first of all, I wrote my own software implementing all of elements the MDM detailed in this chapter. Also, together with Frances and Dirk, I contributed to the development of the reformulations, including writing the different Pseudocodes for constructing the extended active set and running the MDM. The estimates in Section 4.5 were done by Grzegorz Wasilkowski. The numerical results in the paper were obtained using Dirk's code whereas the results in the current chapter are the output from my implementation.

4.A Calculating the Smolyak weights

For a sequence of one-dimensional quadrature rules $\{U_i\}_{i\geq 1}$, with $U_0 \coloneqq 0$, the s-dimensional Smolyak rule with level $m \geq 1$ is

$$\mathcal{Q}_{s,m} \;=\; \sum_{\substack{\boldsymbol{i}\in\mathbb{N}^s \\ |\boldsymbol{i}|\leq s+m-1}} \left(\bigotimes_{j=1}^{s} (U_{i_j} - U_{i_j-1}) \right). \tag{4.40}$$

To avoid repeated function evaluations at the same quadrature point, we write $\mathcal{Q}_{s,m}$ as an explicit sum over the unique quadrature points, and combine the weights for each point from different levels.

For $i \geq 1$, let the ith one-dimensional rule U_i have points $\mathcal{P}_i \coloneqq (t_{i,k})_{k=0}^{n_i-1}$ and weights $(w_{i,k})_{k=0}^{n_i-1}$:

$$U_i(g) = \sum_{k=0}^{n_i-1} w_{i,k}\, g(t_{i,k})\,.$$

Defining the weight functions

$$w_i(t) \;\coloneqq\; \begin{cases} w_{i,k} & \text{if } t = t_{i,k} \in \mathcal{P}_i\,, \\ 0 & \text{otherwise}\,, \end{cases} \tag{4.41}$$

we can write the one-dimensional difference rule as

$$\left(U_i - U_{i-1}\right)(g) \;=\; \sum_{t\in(\mathcal{P}_i\cup\mathcal{P}_{i-1})} \left(w_i(t) - w_{i-1}(t)\right) g(t)\,,$$

with $\mathcal{P}_0 \coloneqq \emptyset$, and write Smolyak's formula applied to a s-dimensional function g as

$$\mathcal{Q}_{s,m}(g) \;=\; \sum_{\substack{\boldsymbol{i}\in\mathbb{N}^s \\ |\boldsymbol{i}|\leq s+m-1}} \sum_{\boldsymbol{t}\in\otimes_{j=1}^{s}(\mathcal{P}_{i_j}\cup\mathcal{P}_{i_j-1})} \left(\prod_{j=1}^{s} (w_{i_j}(t_j) - w_{i_j-1}(t_j)) \right) g(\boldsymbol{t})\,. \tag{4.42}$$

We now rewrite (4.42) to sum over all distinct points in the disjoint "difference grids":

$$\mathcal{Q}_{s,m}(g) \;=\; \sum_{\substack{\boldsymbol{i}\in\mathbb{N}^s \\ |\boldsymbol{i}|\leq s+m-1}} \sum_{\boldsymbol{t}\in\otimes_{j=1}^{s}(\mathcal{P}_{i_j}\setminus\mathcal{P}_{i_j-1})} W_{s,m}(\boldsymbol{t})\, g(\boldsymbol{t})\,, \tag{4.43}$$

where the weight function $W_{s,m}(\boldsymbol{t})$ combines the weights for each distinct point $\boldsymbol{t}$ occurring in the sum in (4.42) as follows:

$$W_{s,m}(\boldsymbol{t}) \;=\; \sum_{\substack{\boldsymbol{q}\in\mathbb{N}^s \\ |\boldsymbol{q}|\le s+m-1}} \sum_{\substack{\boldsymbol{\tau}\in\otimes_{j=1}^{s}(\mathcal{P}_{q_j}\cup\mathcal{P}_{q_j-1}) \\ \boldsymbol{\tau}=\boldsymbol{t}}} \prod_{j=1}^{s}(w_{q_j}(\tau_j) - w_{q_j-1}(\tau_j))\,. \tag{4.44}$$

The above formula essentially came from (4.42) by replacing the labels $(\boldsymbol{i},\boldsymbol{t})$ with $(\boldsymbol{q},\boldsymbol{\tau})$ and then imposing the condition $\boldsymbol{\tau}=\boldsymbol{t}$. In the following, we derive explicit formulas for the weights depending on whether or not the one-dimensional quadrature rules are nested.

Nested case: Suppose first that the points sets $\mathcal{P}_i$ are nested such that

$$t_{i,k} \;=\; t_{i-1,k} \quad \text{for all } k = 0, 1, \dots\, n_{(i-1)} - 1 \text{ and } i \in \mathbb{N}\,,$$

and that the weights $(w_{i,k})_{k=0}^{n_i-1}$ are ordered accordingly. With this ordering,

$$\mathcal{P}_i \setminus \mathcal{P}_{i-1} \;=\; \{t_{i,k} : k = n_{(i-1)}, n_{(i-1)}+1, \dots, n_i - 1\}$$

and (4.43) can be written explicitly as s nested sums

$$\mathcal{Q}_{s,m}(g) = \sum_{\substack{\boldsymbol{i}\in\mathbb{N}^s \\ |\boldsymbol{i}|\le s+m-1}} \sum_{k_1=n_{(i_1-1)}}^{n_{(i_1)}-1} \cdots \sum_{k_s=n_{(i_s-1)}}^{n_{(i_s)}-1} W_{s,m}(\boldsymbol{t}_{\boldsymbol{i},\boldsymbol{k}})\, g(\boldsymbol{t}_{\boldsymbol{i},\boldsymbol{k}})\,,$$

where $\boldsymbol{k} = (k_1, \dots, k_s)$ and $\boldsymbol{t}_{\boldsymbol{i},\boldsymbol{k}} = (t_{i_1,k_1}, \dots, t_{i_s,k_s})$ belongs to the difference grid $\otimes_{j=1}^{s}(\mathcal{P}_{i_j} \setminus \mathcal{P}_{i_j-1})$.

To derive $W_{s,m}(\boldsymbol{t}_{\boldsymbol{i},\boldsymbol{k}})$, we note in the second sum in (4.44) that $\mathcal{P}_{q_j} \cup \mathcal{P}_{q_j-1} = \mathcal{P}_{q_j}$ since the points are nested, and we can restrict the first sum in (4.44) to $\boldsymbol{q} \ge \boldsymbol{i}$ because no component t_{i_j,k_j} can be in $\mathcal{P}_{q_j}$ for $q_j < i_j$. This gives

$$W_{s,m}(\boldsymbol{t}_{\boldsymbol{i},\boldsymbol{k}}) \;=\; \sum_{\substack{\boldsymbol{q}\in\mathbb{N}^s,\, \boldsymbol{q}\ge\boldsymbol{i} \\ |\boldsymbol{q}|\le s+m-1}} \sum_{\substack{\boldsymbol{\tau}\in\otimes_{j=1}^{s}\mathcal{P}_{q_j} \\ \boldsymbol{\tau}=\boldsymbol{t}_{\boldsymbol{i},\boldsymbol{k}}}} \prod_{j=1}^{s}(w_{q_j}(\tau_j) - w_{q_j-1}(\tau_j))\,.$$

Furthermore, since the points are nested we conclude that $\boldsymbol{t}_{\boldsymbol{i},\boldsymbol{k}}$ will occur exactly once in every grid $\otimes_{j=1}^{s}\mathcal{P}_{q_j}$ with $\boldsymbol{q} \ge \boldsymbol{i}$, and due to the ordering of the points it is given by $\boldsymbol{\tau} = \boldsymbol{t}_{\boldsymbol{q},\boldsymbol{k}} = \boldsymbol{t}_{\boldsymbol{i},\boldsymbol{k}}$. Hence, the second sum contains only a single point $\boldsymbol{t}_{\boldsymbol{q},\boldsymbol{k}}$

and it follows from (4.41) that

$$W_{s,m}(\boldsymbol{t}_{\boldsymbol{i},\boldsymbol{k}}) \;=\; \sum_{\substack{\boldsymbol{q}\in\mathbb{N}^s,\,\boldsymbol{q}\geq\boldsymbol{i}\\ |\boldsymbol{q}|\leq s+m-1}} \prod_{j=1}^{s} (w_{q_j,k_j} - w_{q_j-1,k_j})\,.$$

Non-nested case: When the points are not nested, (4.43) can similarly be expanded as s nested sums

$$\mathcal{Q}_{s,m}(g) \;=\; \sum_{\substack{\boldsymbol{i}\in\mathbb{N}^s\\ |\boldsymbol{i}|\leq s+m-1}} \sum_{k_1=0}^{n_{(i_1)}-1} \cdots \sum_{k_s=0}^{n_{(i_s)}-1} W_{s,m}(\boldsymbol{t}_{\boldsymbol{i},\boldsymbol{k}})\, g(\boldsymbol{t}_{\boldsymbol{i},\boldsymbol{k}})\,.$$

In this case, $\boldsymbol{t}_{\boldsymbol{i},\boldsymbol{k}}$ can occur in $\otimes_{j=1}^{s}(\mathcal{P}_{q_j}\cup\mathcal{P}_{q_j-1})$ only when $\boldsymbol{q}=\boldsymbol{i}$ or when $q_j = i_j+1$ for any index j (it only occurs in the next level when it is being subtracted), and moreover, for any $\boldsymbol{q}$ it can occur only once in $\otimes_{j=1}^{s}(\mathcal{P}_{q_j}\cup\mathcal{P}_{q_j-1})$. Thus we can simplify the formula (4.44) to

$$W_{s,m}(\boldsymbol{t}_{\boldsymbol{i},\boldsymbol{k}}) \;=\; \sum_{\substack{\boldsymbol{q}\in\mathbb{N}^s,\, q_j\in\{i_j,i_j+1\}\,\forall j\\ |\boldsymbol{q}|\leq s+m-1}} \prod_{j=1}^{s} (w_{q_j}(t_{i_j,k_j}) - w_{q_j-1}(t_{i_j,k_j}))\,,$$

which is equivalent to

$$W_{s,m}(\boldsymbol{t}_{\boldsymbol{i},\boldsymbol{k}}) \;=\; \sum_{\substack{\boldsymbol{p}\in\{0,1\}^s\\ |\boldsymbol{i}+\boldsymbol{p}|\leq s+m-1}} \prod_{j=1}^{s} (w_{i_j+p_j}(t_{i_j,k_j}) - w_{i_j+p_j-1}(t_{i_j,k_j}))\,.$$

Note that in the formula above only one of $w_{i_j+p_j}(t_{i_j,k_j})$ and $w_{i_j+p_j-1}(t_{i_j,k_j})$ is nonzero at a time. If $p_j = 0$ then $w_{i_j+p_j}(t_{i_j,k_j}) = w_{i_j}(t_{i_j,k_j}) = w_{i_j,k_j}$ and $w_{i_j+p_j-1}(t_{i_j,k_j}) = w_{i_j-1}(t_{i_j,k_j}) = 0$. If $p_j = 1$ then $w_{i_j+p_j}(t_{i_j,k_j}) = w_{i_j+1}(t_{i_j,k_j}) = 0$ and $w_{i_j+p_j-1}(t_{i_j,k_j}) = w_{i_j}(t_{i_j,k_j}) = w_{i_j,k_j}$. These yield

$$W_{s,m}(\boldsymbol{t}_{\boldsymbol{i},\boldsymbol{k}}) \;=\; \sum_{\substack{\boldsymbol{p}\in\{0,1\}^s\\ |\boldsymbol{i}+\boldsymbol{p}|\leq s+m-1}} \prod_{j=1}^{s} (-1)^{p_j} w_{i_j,k_j}\,.$$

4.B The combination technique formula

Here we give a derivation of the combination technique formula for Smolyak's method [39]. The derivation is the same as in [91], with only a change of how we index the vectors.

We begin by rewriting the tensor product of differences in (4.40) as

$$\bigotimes_{j=1}^{s}(U_{i_j} - U_{i_j-1}) \;=\; \sum_{\boldsymbol{q}\in\{0,1\}^s} (-1)^{|\boldsymbol{q}|} \bigotimes_{j=1}^{s} U_{i_j-q_j}\,.$$

Substituting this into (4.40) we see that each unique tensor product $\bigotimes_{j=1}^{s} U_{\ell_j}$ for $\boldsymbol{\ell} \in \mathbb{N}^s$, $|\boldsymbol{\ell}| \le m+s-1$,$\otimes_{j=1}^{s} U_{\ell_j}$ occurs with sign $(-1)^{|\boldsymbol{q}|}$ whenever $i_j = \ell_j + q_j$ for any $j = 1,\ldots,s$ with $\boldsymbol{q} \in \{0,1\}^s$ and $|\boldsymbol{i}| = |\boldsymbol{q}+\boldsymbol{\ell}| \le m+s-1$. It is more convenient to write the last condition as $|\boldsymbol{q}| \le m+s-1-|\boldsymbol{\ell}|$. The Smolyak operator (4.40) can then be written as a sum over the unique tensor products of one-dimensional operators instead of differences

$$\mathcal{Q}_{s,m} \;=\; \sum_{\substack{\boldsymbol{\ell}\in\mathbb{N}^s \\ |\boldsymbol{\ell}|\le m+s-1}} c_{s,m,\boldsymbol{\ell}} \bigotimes_{j=1}^{s} U_{\ell_j}\,, \qquad \text{with} \qquad c_{s,m,\boldsymbol{\ell}} \coloneqq \sum_{\substack{\boldsymbol{q}\in\{0,1\}^s \\ |\boldsymbol{q}|\le m+s-1-|\boldsymbol{\ell}|}} (-1)^{|\boldsymbol{q}|}\,.$$

Summing over all the possible values for $|\boldsymbol{q}|$ we have

$$c_{s,m,\boldsymbol{\ell}} \;=\; \sum_{k=0}^{\min(s,m+s-1-|\boldsymbol{\ell}|)} \sum_{\substack{\boldsymbol{q}\in\{0,1\}^s \\ |\boldsymbol{q}|=k}} (-1)^k \;=\; \sum_{k=0}^{\min(s,m+s-1-|\boldsymbol{\ell}|)} (-1)^k \binom{s}{k}\,.$$

Then using the identity

$$\sum_{k=0}^{n_1} (-1)^k \binom{n_2}{k} \;=\; \begin{cases} 0 & \text{if } n_1 \ge n_2\,, \\ (-1)^{n_1}\binom{n_2-1}{n_1} & \text{otherwise}\,, \end{cases}$$

we see that $c_{s,m,\boldsymbol{\ell}} = 0$ if $s \le m+s-1-|\boldsymbol{\ell}|$, or equivalently, $|\boldsymbol{\ell}| \le m-1$; otherwise for $|\boldsymbol{\ell}| \ge m$ we have

$$c_{s,m,\boldsymbol{\ell}} \;=\; (-1)^{m+s-1-|\boldsymbol{\ell}|}\binom{s-1}{m+s-1+|\boldsymbol{\ell}|} \;=\; (-1)^{m+s-1-|\boldsymbol{\ell}|}\binom{s-1}{|\boldsymbol{\ell}|-m}.$$

The formula for the combination technique is then given by

$$\mathcal{Q}_{s,m} \;=\; \sum_{\substack{\boldsymbol{i}\in\mathbb{N}^s \\ m\le|\boldsymbol{i}|\le m+s-1}} (-1)^{m+s-1-|\boldsymbol{i}|} \binom{s-1}{|\boldsymbol{i}|-m} \bigotimes_{j=1}^{s} U_{i_j}\,.$$

Chapter 5

A new method of constructing active sets for product weights

Again we consider approximating integrals with infinitely-many variables by the Multivariate Decomposition Method (MDM),

$$\mathcal{I}(f) \approx \mathcal{A}_\varepsilon(f) = \sum_{\mathfrak{u} \subset \mathcal{U}_\varepsilon} A_\mathfrak{u}(f_\mathfrak{u}),$$

and present here an alternate method of constructing active sets $\mathcal{U}_\varepsilon$ in the case where the inputs $w(\mathfrak{u})$ are in product form (2.6). The goal is to construct active sets that are smaller than those introduced in Sections 2.5.4 and 4.1, which for some threshold T were of the form

$$\mathcal{U}_\varepsilon = \{\mathfrak{u} \subset \mathbb{N} : w(\mathfrak{u}) > T\}.$$

In fact, we present two similar algorithms, one that produces active sets that are optimal, in a certain sense, and then a simplification of the first algorithm that produces quasi-optimal active sets.

In this chapter, we perform the error analysis in the weighted function space setting of, e.g., [33, 76, 89, 90] and hence the parameters $w(\mathfrak{u})$ are now related to the function space weights. However, the strategies outlined in this chapter can be easily adapted to the "norm bounds" setting of [58] and Section 2.5.1, or other more general spaces. Also, the pseudocodes detailing the two algorithms are presented so as to take arbitrary inputs.

5.1 Introduction

In this chapter, the functions $f : \mathcal{Y}^\mathbb{N} \to \mathbb{R}$ to be integrated belong to a weighted tensor product Banach space $\mathcal{F}_\boldsymbol{\gamma}$, which will be detailed fully in Section 5.2.1, and are assumed to admit a decomposition

$$f(\boldsymbol{y}) = \sum_{\mathfrak{u} \subset \mathbb{N}} f_\mathfrak{u}(\boldsymbol{y}_\mathfrak{u}). \tag{5.1}$$

As in Section 2.3.2, for each finite $\mathfrak{u} \subset \mathbb{N}$ the weight parameters $\gamma_{\mathfrak{u}}$ represents the importance of the subset of variables $\boldsymbol{y}_{\mathfrak{u}}$, and will appear in the norm in $\mathcal{F}_{\boldsymbol{\gamma}}$. In this chapter we assume the weights are of product form

$$\gamma_{\mathfrak{u}} \;=\; \prod_{j\in\mathfrak{u}} \gamma_j \,.$$

Again, the inputs to our algorithm are a collection of parameters $\{w(\mathfrak{u})\}_{\mathfrak{u}\subset\mathbb{N}}$, but since we are in the weighted function space setting their values will now be transformations of $\gamma_{\mathfrak{u}}$, see (5.10) later.

In Section 2.5 we saw that an essential component of the MDM is the construction of an *active set* $\mathcal{U}_\varepsilon$ of finite subsets $\mathfrak{u} \subset \mathbb{N}$ such that the integral of $\sum_{\mathfrak{u}\notin\mathcal{U}_\varepsilon} f_{\mathfrak{u}}$ can be neglected because its size is of the order ε. In other words, it is enough to approximate integrals of the partial sum

$$\sum_{\mathfrak{u}\in\mathcal{U}_\varepsilon} f_{\mathfrak{u}}.$$

For a fixed error request ε, the choice of active set is not unique (consider (4.4) and (2.34), which contained a free parameter), but the size and composition of $\mathcal{U}_\varepsilon$ directly affects the amount of work done by an MDM approximation. If an active set is larger than is required then the MDM algorithm must approximate more terms $\mathcal{I}_{\mathfrak{u}}(f_{\mathfrak{u}})$ than is necessary to achieve the given error. To reduce the work done by an MDM approximation we want active sets $\mathcal{U}_\varepsilon$ with small size and such that the largest cardinality among the elements $\mathfrak{u} \in \mathcal{U}_\varepsilon$, which recall is denoted

$$\sigma(\mathcal{U}_\varepsilon) \;\coloneqq\; \max\left\{|\mathfrak{u}| : \mathfrak{u} \in \mathcal{U}_\varepsilon\right\},$$

is also small. Active sets with small maximum cardinality are also desirable because low-dimensional integrals are computationally easier to approximate.

A specific construction of active sets, denoted here by $\mathcal{U}_\varepsilon^{\mathrm{PW}}$, was proposed in [75] (see also Section 2.5.4) and it was shown there that the largest cardinality among all $\mathfrak{u} \in \mathcal{U}_\varepsilon^{\mathrm{PW}}$ grows very slowly with decreasing ε,

$$\sigma\left(\mathcal{U}_\varepsilon^{\mathrm{PW}}\right) \;=\; \mathcal{O}\left(\frac{\ln(1/\varepsilon)}{\ln(\ln(1/\varepsilon))}\right) \quad \text{as } \varepsilon \;\to\; 0.$$

Moreover, the size $|\mathcal{U}_\varepsilon^{\mathrm{PW}}|$ grows polynomially in $1/\varepsilon$, however, the asymptotic constants were not investigated and, as we shall see, they could be very large. As such, in this chapter we consider constructing the smallest possible, or "optimal", active

sets denoted by $\mathcal{U}_\varepsilon^{\rm opt}$. The examples presented in Section 5.4 illustrate that the difference between the size of $\mathcal{U}_\varepsilon^{\rm opt}$ and $\mathcal{U}_\varepsilon^{\rm PW}$ can be very large.

The active sets $\mathcal{U}_\varepsilon^{\rm opt}$ presented in this chapter are optimal in the sense that if $\mathcal{U}_\varepsilon$ is an active set for the same error request as $\mathcal{U}_\varepsilon^{\rm opt}$ then $|\mathcal{U}_\varepsilon^{\rm opt}| \leq |\mathcal{U}_\varepsilon|$. The idea behind their construction is quite simple: choose the active set to consist of the subsets $\mathfrak{u} \subset \mathbb{N}$ that correspond to the largest parameters $w(\mathfrak{u})$. This strategy gives rise to two main computational difficulties: ordering the sequence $\{w(\mathfrak{u})\}_{\mathfrak{u}\subset\mathbb{N}}$ and, for a given error request, determining the number of sets to be included to ensure that $\mathcal{U}_\varepsilon^{\rm opt}$ is a valid active set. The algorithm can be simplified to give a construction of "quasi-optimal" active sets $\mathcal{U}_\varepsilon^{\rm q-opt}$, which uses an approximate ordering of $\{w(\mathfrak{u})\}_{\mathfrak{u}\subset\mathbb{N}}$ and leads to active sets with a slightly larger cardinality than $\mathcal{U}_\varepsilon^{\rm opt}$. Once the active set is constructed, the remaining steps for implementing the MDM algorithm will be the same as discussed in the previous chapter

Since, we are also interested in active sets with the smallest $\sigma(\mathcal{U}_\varepsilon)$, we introduce the following concept of *ε-superposition dimension* (or *superposition dimension* for short) defined by

$$\sigma_\varepsilon := \min\{\sigma(\mathcal{U}_\varepsilon) \ : \ \mathcal{U}_\varepsilon \text{ is an active set}\}.$$

The optimal active sets in our experiments have very small maximum cardinality, implying that the superposition dimension is also small.

Note that our concept of the superposition dimension depends on the integration problem as well as the error demand ε. Hence it is in the same spirit as the definition of *truncation dimension* introduced recently in [53]. They are different from the definitions in statistical literature, see, e.g., [11, 65, 71, 88], where superposition and truncation dimensions are defined based on ANOVA decompositions and without any relation to the integration problem or the error demand ε. Moreover, the dimensions from [11, 65, 88] depend on specific functions, whereas the dimensions in [53] and in this chapter are defined in the worst-case sense, i.e., are relevant to all functions from the space $\mathcal{F}_{\boldsymbol{\gamma}}$.

Although the algorithms for constructing $\mathcal{U}_\varepsilon^{\rm q-opt}$ and $\mathcal{U}_\varepsilon^{\rm opt}$ work for rather general problems and spaces, we applied them to the integration problem and for weighted spaces of functions with mixed first order partial derivatives bounded in L^p norms for $p \in [1, \infty]$. Such spaces have often been considered (mostly for $p = 2$) when analysing *Quasi-Monte Carlo methods*. The results on the size of active sets will depend both on the integrability parameter p and on how fast the weights converge

to zero. In the experiments, we considered

$$\gamma_j = \frac{c}{j^a} \quad \text{for } a = 2, 3, 4 \text{ and } c > 0.$$

The remainder of this chapter proceeds as follows. In Section 5.2 we summarise the mathematical setting of the paper. Then in Section 5.3 we outline the two new algorithms for constructing optimal and quasi-optimal active sets, along with explicit pseudocodes. Finally, numerical results giving the size of different active sets are given in Section 5.4.

5.2 Mathematical background

In this section we introduce the mathematical setting for our new constructions, namely, the weighted function class and the integration problem, then we conclude with a summary of the previous method for constructing active sets.

5.2.1 $\boldsymbol{\gamma}$-weighted spaces

To define the weighted function space we follow [43], and as previously in the setting without weights from Section 2.5.1 we begin by assuming that f admits a decomposition of the form (5.1), where each term $f_{\mathfrak{u}}$ depends only on the variables $\boldsymbol{y}_{\mathfrak{u}} = (y_j)_{j\in\mathfrak{u}}$ and belongs to a Banach space $F_{\mathfrak{u}}$ defined below. Then for a collection of positive weights $\boldsymbol{\gamma} = \{\gamma_{\mathfrak{u}}\}_{\mathfrak{u}\subset\mathbb{N}}$ and $p \in [1, \infty]$ we define $\mathcal{F}_{\boldsymbol{\gamma},p}$ to be the Banach space of functions defined on $\mathcal{Y}^{\mathbb{N}}$ with the norm

$$\|f\|_{\mathcal{F}_{\boldsymbol{\gamma},p}} = \left(\sum_{\mathfrak{u}\subset\mathbb{N}} \gamma_{\mathfrak{u}}^{-p} \, \|f_{\mathfrak{u}}\|_{F_{\mathfrak{u}}}^p\right)^{\frac{1}{p}},$$

for $p \in [1, \infty)$, and for $p = \infty$ the norm is

$$\|f\|_{\mathcal{F}_{\boldsymbol{\gamma},\infty}} = \sup_{\mathfrak{u}\subset\mathbb{N}} \frac{\|f_{\mathfrak{u}}\|_{F_{\mathfrak{u}}}}{\gamma_{\mathfrak{u}}}.$$

For the remainder of the chapter we assume that the weights $\gamma_{\mathfrak{u}}$ are of product form (2.6) with

$$\gamma_{\mathfrak{u}} = \prod_{j\in\mathfrak{u}} \frac{c}{j^a} \quad \text{for } a > 1 \text{ and } c > 0. \tag{5.2}$$

In general choosing the weights (in our case choosing a and c) for a specific integral or application is a difficult problem that we do not attempt to address in this chapter. We assume that the parameters a and c are given with the problem. The topic of how to choose the weights will be studied, albeit in a slightly different setting, in Chapter 6.

Comparing with the setting using bounds on the norm instead of weights, the function class $\mathcal{F}$ from Section 2.5.1 corresponds to the unit ball of $\mathcal{F}_{\boldsymbol{\gamma},p}$ in the specific case when $\gamma_{\mathfrak{u}} = B_{\mathfrak{u}}$ and $p = \infty$.

To completely specify $\mathcal{F}_{\boldsymbol{\gamma},p}$ we now define each $F_{\mathfrak{u}}$ to consist of functions with L^p-integrable mixed first derivatives. Let $\mathcal{Y} = [0,1]$, so that the integration domain is the (countably) infinite-dimensional unit cube $[0,1]^{\mathbb{N}}$, and consider the anchored decomposition with anchor $\mathbf{0}$ for evaluating $f_{\mathfrak{u}}$ (2.28). Then define each $F_{\mathfrak{u}}$ to be the space of functions $g_{\mathfrak{u}} : \mathcal{Y}^{|\mathfrak{u}|} \to \mathbb{R}$ that are absolutely continuous, anchored at $\mathbf{0}$, and equip $F_{\mathfrak{u}}$ with the norm

$$\|g_{\mathfrak{u}}\|_{F_{\mathfrak{u}}} := \left\| \frac{\partial^{|\mathfrak{u}|}}{\partial \boldsymbol{y}_{\mathfrak{u}}} g_{\mathfrak{u}}(\boldsymbol{y}_{\mathfrak{u}}) \right\|_{L^p(\mathcal{Y}^{|\mathfrak{u}|})}.$$

Of course, $F_{\emptyset}$ is the space of constant functions with the absolute value as its norm. For $f_{\mathfrak{u}}$ given by the anchored decomposition (2.28) the norm above simplifies to

$$\|f_{\mathfrak{u}}\|_{F_{\mathfrak{u}}} = \left\| \frac{\partial^{|\mathfrak{u}|}}{\partial \boldsymbol{y}_{\mathfrak{u}}} f(\cdot_{\mathfrak{u}}; \mathbf{0}) \right\|_{L^p(\mathcal{Y}^{|\mathfrak{u}|})},$$

so in this case $\mathcal{F}_{\boldsymbol{\gamma},p}$ is the Banach space of functions with norm, for $p \in [1,\infty)$,

$$\|f\|_{\mathcal{F}_{\boldsymbol{\gamma},p}} = \left(\sum_{\mathfrak{u} \subset \mathbb{N}} \gamma_{\mathfrak{u}}^{-p} \left\| \frac{\partial^{|\mathfrak{u}|}}{\partial \boldsymbol{y}_{\mathfrak{u}}} f(\cdot_{\mathfrak{u}}; \mathbf{0}) \right\|_{L^p(\mathcal{Y}^{|\mathfrak{u}|})}^p \right)^{\frac{1}{p}},$$

and for $p = \infty$

$$\|f\|_{\mathcal{F}_{\boldsymbol{\gamma},\infty}} = \sup_{\mathfrak{u} \subset \mathbb{N}} \frac{1}{\gamma_{\mathfrak{u}}} \left\| \frac{\partial^{|\mathfrak{u}|}}{\partial \boldsymbol{y}_{\mathfrak{u}}} f(\cdot_{\mathfrak{u}}; \mathbf{0}) \right\|_{L^\infty(\mathcal{Y}^{|\mathfrak{u}|})}.$$

Note that the space $\mathcal{F}_{\boldsymbol{\gamma},p}$ above is a special case of the function spaces studied in [90], and in particular it is shown in [90] that if

$$\sum_{\mathfrak{u} \subset \mathbb{N}} \gamma_{\mathfrak{u}} = \sum_{\mathfrak{u} \subset \mathbb{N}} \prod_{j \in \mathfrak{u}} \gamma_j < \infty$$

then $\sum_{\mathfrak{u} \subset \mathbb{N}} f_{\mathfrak{u}}(\boldsymbol{y}_{\mathfrak{u}})$ is absolutely convergent for all $\boldsymbol{y} \in \mathcal{Y}^{\mathbb{N}}$ and the function space $\mathcal{F}_{\boldsymbol{\gamma},p}$ is well-defined.

The space $\mathcal{F}_{\boldsymbol{\gamma},p}$ contains in particular the following class of functions.

Example 5.1. For a smooth function $g : \mathbb{R} \to \mathbb{R}$ and fast decaying numbers $a_1, a_2, \ldots$, consider

$$f(\boldsymbol{y}) = g\left(\sum_{j=1}^{\infty} y_j \, a_j \right) \quad \text{for } y_j \in \mathcal{Y}.$$

In particular, the test integrand (2.29) is of this form with $\mathcal{Y} = [-\frac{1}{2}, \frac{1}{2}]$. Then by the chain rule

$$\frac{\partial^{|\mathfrak{u}|}}{\partial \boldsymbol{y}_{\mathfrak{u}}} f^{(\mathfrak{u})}(\boldsymbol{y}_{\mathfrak{u}}; \mathbf{0}) \;=\; g^{(|\mathfrak{u}|)}\left(\sum_{j \in \mathfrak{u}} y_j \, a_j\right) \prod_{j \in \mathfrak{u}} a_j \,,$$

where $g^{(k)}$ denotes the kth derivative.

Hence, $f \in \mathcal{F}_{\boldsymbol{\gamma},p}$ if the derivatives of g and the coefficients a_j satisfy

$$\left(\sum_{\mathfrak{u} \subset \mathbb{N}} \frac{\prod_{j \in \mathfrak{u}} |a_j|^p}{\gamma_{\mathfrak{u}}^p} \int_{\mathcal{Y}^{|\mathfrak{u}|}} \left| g^{(|\mathfrak{u}|)}\left(\sum_{j \in \mathfrak{u}} y_j \, a_j\right) \right|^p \mathrm{d}\boldsymbol{y}_{\mathfrak{u}}\right)^{1/p} \;<\; \infty.$$

5.2.2 The integration problem

Consider again the integration functional $\mathcal{I} : \mathcal{F}_{\boldsymbol{\gamma},p} \to \mathbb{R}$ given by

$$\mathcal{I}(f) \;=\; \sum_{\mathfrak{u} \subset \mathbb{N}} \mathcal{I}_{\mathfrak{u}}(f_{\mathfrak{u}}) \,, \tag{5.3}$$

where the $|\mathfrak{u}|$-variate integrals $\mathcal{I}_{\mathfrak{u}} : F_{\mathfrak{u}} \to \mathbb{R}$ are given by

$$\mathcal{I}_{\mathfrak{u}}(g_{\mathfrak{u}}) \;=\; \int_{\mathcal{Y}^{|\mathfrak{u}|}} g_{\mathfrak{u}}(\boldsymbol{y}_{\mathfrak{u}}) \, \mathrm{d}\boldsymbol{y}_{\mathfrak{u}} \,,$$

and are continuous in $F_{\mathfrak{u}}$ with

$$\|\mathcal{I}_{\mathfrak{u}}\|_{F_{\mathfrak{u}}} \;=\; \frac{1}{(p^* + 1)^{|\mathfrak{u}|\,/p^*}} \,, \tag{5.4}$$

where p^* denotes the conjugate of p:

$$\frac{1}{p} + \frac{1}{p^*} \;=\; 1 \,.$$

The value for the norm of the integration functional (5.4) can be found in, e.g., [90], but for completeness we provide an explicit derivation. To begin with, consider $f : [0,1] \to \mathbb{R}$ belonging to $F_{\{1\}}$, then since f is anchored at 0 for $y \in [0,1]$ the integral of f can be written

$$\mathcal{I}_{\{1\}}(f) \;=\; \int_0^1 f(y) \, \mathrm{d}y \;=\; \int_0^1 \left(\int_0^1 f'(t) \, \mathbb{1}_{[0,y]}(t) \, \mathrm{d}t \right) \mathrm{d}y \;=\; \int_0^1 f'(t) \int_0^1 \mathbb{1}_{[0,y]}(t) \, \mathrm{d}y \, \mathrm{d}t \,,$$

where $\mathbb{1}_{\mathcal{S}}$ is the indicator function for a set $\mathcal{S}$. Evaluating the integral with respect to y and applying Hölder's inequality gives

$$\mathcal{I}_{\{1\}}(f) \;\leq\; \left(\int_0^1 f'(t)^p \,\mathrm{d}t\right)^{1/p} \left(\int_0^1 (1-t)^{p^*} \,\mathrm{d}t\right)^{1/p^*} \;=\; \|f\|_{F_{\{1\}}} \,\frac{1}{(p^*+1)^{1/p^*}}\,.$$

Finally, taking the supremum yields the one-dimensional version of (5.4):

$$\left\|\mathcal{I}_{\{1\}}\right\|_{F_{\{1\}}} \;=\; \frac{1}{(p^*+1)^{1/p^*}}\,.$$

The result for all finite $\mathfrak{u} \subset \mathbb{N}$ follows from the fact, see [42], that the norm of $\mathcal{I}_{\mathfrak{u}}$ is the product of the norm of the one-dimensional functionals:

$$\|\mathcal{I}_{\mathfrak{u}}\|_{F_{\mathfrak{u}}} \;=\; \prod_{j\in\mathfrak{u}} \left\|\mathcal{I}_{\{j\}}\right\|_{F_{\{j\}}} \;=\; \frac{1}{(p^*+1)^{|\mathfrak{u}|/p^*}}\,,$$

as required.

To ensure that the integral (5.3) is well-defined we assume that

$$\left(\sum_{\mathfrak{u}\subset\mathbb{N}} \gamma_{\mathfrak{u}}^{p^*}\,(p^*+1)^{-|\mathfrak{u}|}\right)^{1/p^*} \;<\; \infty\,. \tag{5.5}$$

Indeed, by Hölder's inequality, for $f \in \mathcal{F}_{\boldsymbol{\gamma},p}$ we have

$$|\mathcal{I}(f)| \;\leq\; \left(\sum_{\mathfrak{u}\subset\mathbb{N}} \gamma_{\mathfrak{u}}^{p^*}\,\|\mathcal{I}_{\mathfrak{u}}\|_{F_{\mathfrak{u}}}^{p^*}\right)^{1/p^*} \|f\|_{\mathcal{F}_{\boldsymbol{\gamma},p}}\,,$$

and in fact there exists an $f \in \mathcal{F}_{\boldsymbol{\gamma},p}$ that attains equality. It then follows by (5.4) that

$$\|\mathcal{I}\|_{\mathcal{F}_{\boldsymbol{\gamma},p}} \;=\; \left(\sum_{\mathfrak{u}\subset\mathbb{N}} \gamma_{\mathfrak{u}}^{p^*}\,(p^*+1)^{-|\mathfrak{u}|}\right)^{1/p^*}\,,$$

and so (5.5) is a necessary and sufficient condition for continuity of $\mathcal{I}$. For our case of product weights of the form (5.2), we have

$$\|\mathcal{I}\|_{\mathcal{F}_{\boldsymbol{\gamma},p}} \;=\; \left(\sum_{\mathfrak{u}\subset\mathbb{N}} \gamma_{\mathfrak{u}}^{p^*}\,(p^*+1)^{-|\mathfrak{u}|}\right)^{1/p^*} \;=\; \prod_{j=1}^{\infty}\left(1+\frac{c^{p^*}}{j^{a\,p^*}\,(p^*+1)}\right)^{1/p^*}\,,$$

which, leaving out the power $1/p^*$, can be bounded above and below by using

$$\sum_{j=1}^{\infty} \frac{c^{p^*}}{j^{a\,p^*}\,(p^*+1)} \;\leq\; \prod_{j=1}^{\infty}\left(1+\frac{c^{p^*}}{j^{a\,p^*}\,(p^*+1)}\right) \;\leq\; \exp\left(\sum_{j=1}^{\infty} \frac{c^{p^*}}{j^{a\,p^*}\,(p^*+1)}\right).$$

Both the upper and lower bounds are finite if and only if $\sum_{j=1}^{\infty} j^{-ap^*} < \infty$, hence in the present setting, (5.5) is equivalent to $a > 1/p^*$. For the remainder of the chapter it is assumed that $a > 1/p^*$, and thus the integral (5.3) is well-defined.

5.2.3 Constructing active sets for $p = 1$

A key component of the MDM, see Section 2.5, is the construction of *active sets* $\mathcal{U}_\varepsilon$, which are sets of finite subsets $\mathfrak{u} \subset \mathbb{N}$ such that when approximating $\mathcal{I}(f)$ it is enough to restrict the attention to functions

$$\sum_{\mathfrak{u} \in \mathcal{U}_\varepsilon} f_\mathfrak{u}.$$

Formally, in this chapter we wish to construct active sets such that the truncation component of the error of an MDM approximation can be controlled by

$$\left| \mathcal{I}\left(\sum_{\mathfrak{u} \notin \mathcal{U}_\varepsilon} f_\mathfrak{u} \right) \right| \leq \varepsilon \left\| \sum_{\mathfrak{u} \notin \mathcal{U}_\varepsilon} f_\mathfrak{u} \right\|_{\mathcal{F}_{\boldsymbol{\gamma},p}} \quad \text{for all } f \in \mathcal{F}_{\boldsymbol{\gamma},p}. \tag{5.6}$$

Any algorithm approximating $\sum_{\mathfrak{u} \in \mathcal{U}_\varepsilon} \mathcal{I}_\mathfrak{u}(f_\mathfrak{u})$ with the worst-case error on $\bigoplus_{\mathfrak{u} \in \mathcal{U}_\varepsilon} F_\mathfrak{u}$ bounded by ε has its worst-case error on the whole space $\mathcal{F}_{\boldsymbol{\gamma},p}$ bounded by

$$2^{1/p^*}\, \varepsilon.$$

The factor of $2^{1/p^*}$ is the result of applying Hölder's inequality, see, e.g., [53].

To obtain a more practical condition than (5.6), let $\mathcal{U}$ be an arbitrary set of finite subsets $\mathfrak{u} \subset \mathbb{N}$. Then the truncation error corresponding to $\mathcal{U}$ is bounded by

$$\begin{aligned} \left| \mathcal{I}\left(\sum_{\mathfrak{u} \notin \mathcal{U}} f_\mathfrak{u} \right) \right| &\leq \sum_{\mathfrak{u} \notin \mathcal{U}} \|f_\mathfrak{u}\|_{F_\mathfrak{u}} \, \|\mathcal{I}_\mathfrak{u}\|_{F_\mathfrak{u}} = \sum_{\mathfrak{u} \notin \mathcal{U}} \frac{\|f_\mathfrak{u}\|_{F_\mathfrak{u}}}{\gamma_\mathfrak{u}} \, \gamma_\mathfrak{u} \, \|\mathcal{I}_\mathfrak{u}\|_{F_\mathfrak{u}} \\ &\leq \left\| \sum_{\mathfrak{u} \notin \mathcal{U}} f_\mathfrak{u} \right\|_{\mathcal{F}_{\boldsymbol{\gamma},p}} \left(\sum_{\mathfrak{u} \notin \mathcal{U}} \gamma_\mathfrak{u}^{p^*} \, \|\mathcal{I}_\mathfrak{u}\|_{F_\mathfrak{u}}^{p^*} \right)^{1/p^*}. \end{aligned}$$

Hence, if $\mathcal{U}_\varepsilon$ is such that

$$\left(\sum_{\mathfrak{u} \notin \mathcal{U}_\varepsilon} \gamma_\mathfrak{u}^{p^*} \, \|\mathcal{I}_\mathfrak{u}\|_{F_\mathfrak{u}}^{p^*} \right)^{1/p^*} \leq \varepsilon\,, \tag{5.7}$$

then it is an active set satisfying (5.6). In fact, since Hölder's inequality is sharp (i.e., the equality is attained for some $f \in \mathcal{F}_{\boldsymbol{\gamma},p}$) (5.7) is equivalent to (5.6).

Clearly, there are many sets $\mathcal{U}_\varepsilon$ that meet the criteria (5.7)—any superset of $\mathcal{U}_\varepsilon$ also satisfies (5.7)—and the goal of this chapter is to construct the smallest possible active set.

A construction of active sets was first proposed in [75] and in this chapter such sets are denoted by $\mathcal{U}_\varepsilon^{\mathrm{PW}}$. Now, we recall the construction of $\mathcal{U}_\varepsilon^{\mathrm{PW}}$, first for $p = 1$, then for $p > 1$ in the following section.

For $p = 1$, the conjugate is $p^* = \infty$ and $\|\mathcal{I}_\mathfrak{u}\|_{F_\mathfrak{u}} = 1$ for all $\mathfrak{u}$. In this case, the condition (5.7) becomes

$$\left(\sum_{\mathfrak{u} \notin \mathcal{U}_\varepsilon} \gamma_\mathfrak{u}^{p^*} \, \|\mathcal{I}_\mathfrak{u}\|_{F_\mathfrak{u}}^{p^*} \right)^{1/p^*} = \sup_{\mathfrak{u} \notin \mathcal{U}_\varepsilon} \gamma_\mathfrak{u} \leq \varepsilon \,, \tag{5.8}$$

and hence if we define

$$\mathcal{U}_\varepsilon^{\mathrm{PW}} = \{\mathfrak{u} \subset \mathbb{N} : \ \gamma_\mathfrak{u} > \varepsilon\} \,, \tag{5.9}$$

then $\mathcal{U}_\varepsilon^{\mathrm{PW}}$ is an active set satisfying the criteria (5.7). For product weights of the form (5.2) this reduces to

$$\mathcal{U}_\varepsilon^{\mathrm{PW}} = \left\{ \mathfrak{u} \subset \mathbb{N} : \prod_{j \in \mathfrak{u}} \frac{c}{j^a} > \varepsilon \right\}.$$

Since in this case the truncation error is given by the supremum in (5.8), it is easy to see that $\mathcal{U}_\varepsilon^{\mathrm{PW}}$ is a subset of any $\mathcal{U}_\varepsilon$ satisfying (5.8). Hence, $\mathcal{U}_\varepsilon^{\mathrm{PW}}$ is the smallest active set that satisfies (5.6).

5.2.4 Constructing active sets for $p > 1$

For $p > 1$, the conjugate p^* is finite and the construction of $\mathcal{U}_\varepsilon$ is more complicated. To simplify the notation, let

$$w(\mathfrak{u}) = \frac{\gamma_\mathfrak{u}^{p^*}}{(p^*+1)^{|\mathfrak{u}|}} = \left(\frac{c^{p^*}}{p^*+1} \right)^{|\mathfrak{u}|} \prod_{j \in \mathfrak{u}} \frac{1}{j^{a\,p^*}}. \tag{5.10}$$

Although the parameters $w(\mathfrak{u})$ here play the same role as in Chapter 4, that of inputs into the MDM algorithm, because the theoretical settings differ between the two chapters the values they take here will be different from their values in Section 4.6.

For given ε and p, the active set is again given by including all $\mathfrak{u}$ with $w(\mathfrak{u})$ exceeding some threshold:

$$\mathcal{U}_\varepsilon^{\mathrm{PW}} = \left\{ \mathfrak{u} \subset \mathbb{N} : w(\mathfrak{u}) > T^{\mathrm{PW}}(\varepsilon) \right\},$$

but now the threshold is more complicated, and for $\alpha \in (1/(ap^*), 1)$ it is given by

$$T^{\mathrm{PW}}(\varepsilon) \coloneqq \left(\frac{\varepsilon^{p^*}}{\sum_{\mathfrak{u} \subset \mathbb{N}} \lceil w(\mathfrak{u}) \rceil^{\alpha}} \right)^{1/(1-\alpha)},$$

which is similar to (2.34) and (4.30). It was shown in [75] that such active sets satisfy (5.6). Note that the interval $(1/(ap^*), 1)$ is non-empty by the assumption that $a > 1/p^*$ introduced in Section 5.2.2. In our numerical experiments we approximated the sum of $[w(\mathfrak{u})]^{\alpha}$ for $\alpha = i/40$ $(40/(ap^*) < i \leq 39)$ and selected the value which resulted in the largest T. The approximations are calculated in a similar way as the computation of S_t explained later (see (5.16)).

5.3 Optimal active sets

Active sets that are defined by satisfying (5.7) are not unique, and the size of $\mathcal{U}_{\varepsilon}^{\mathrm{PW}}$ is dependent on the free parameter α. Also, although the sets $\mathcal{U}_{\varepsilon}^{\mathrm{PW}}$ contain a number of $\mathfrak{u}$'s with the largest $w(\mathfrak{u})$, the number of them could be much larger than is necessary. To this end we define the following notion of optimality of an active set.

Definition 5.2. An active set, denoted by $\mathcal{U}_{\varepsilon}^{\mathrm{opt}}$, is defined to be *optimal* if

$$|\mathcal{U}_{\varepsilon}^{\mathrm{opt}}| = \min\{|\mathcal{U}_{\varepsilon}| \ : \ \mathcal{U}_{\varepsilon} \text{ satisfies (5.6)}\}.$$

Also, define the *ε-superposition dimension* as the smallest $\sigma(\mathcal{U}_{\varepsilon})$ among all active sets:

$$\sigma_{\varepsilon} \coloneqq \min\left\{\sigma(\mathcal{U}_{\varepsilon}) \ : \ \mathcal{U}_{\varepsilon} \text{ satisfies (5.6)}\right\}.$$

The collection of those $\mathfrak{u}$ with the largest $w(\mathfrak{u})$ that are necessary for (5.6) would result in the optimal set $\mathcal{U}_{\varepsilon}^{\mathrm{opt}}$, and clearly the optimal set is always a subset of $\mathcal{U}_{\varepsilon}^{\mathrm{PW}}$. Hence, the property that the maximum cardinality behaves like

$$\sigma(\mathcal{U}_{\varepsilon}^{\mathrm{opt}}) = \mathcal{O}\left(\frac{\ln(1/\varepsilon)}{\ln(\ln(1/\varepsilon))}\right) \quad \text{as } \varepsilon \to 0,$$

is preserved.

More precisely, let $(\mathfrak{u}_j)_{j \in \mathbb{N}}$ be a sequence of all $\mathfrak{u} \subset \mathbb{N}$ ordered such that

$$w(\mathfrak{u}_1) \geq w(\mathfrak{u}_2) \geq w(\mathfrak{u}_3) \geq \cdots.$$

Then the optimal active set is given by

$$\mathcal{U}_\varepsilon^{\mathrm{opt}} \coloneqq \{\mathfrak{u}_1, \ldots, \mathfrak{u}_k\}$$

with $k = k(\varepsilon)$ such that

$$\|\mathcal{I}\|_{\mathcal{F}_{\boldsymbol{\gamma},p}}^{p^*} - \sum_{j=1}^{k} w(\mathfrak{u}_j) \;\leq\; \varepsilon^{p^*} \;<\; \|\mathcal{I}\|_{\mathcal{F}_{\boldsymbol{\gamma},p}}^{p^*} - \sum_{j=1}^{k-1} w(\mathfrak{u}_j). \tag{5.11}$$

Clearly, $\mathcal{U}_\varepsilon^{\mathrm{opt}}$ satisfies (5.7) and is an active set. In fact, since $w(\mathfrak{u}) > 0$ it is also immediately obvious that $\mathcal{U}_\varepsilon^{\mathrm{opt}}$ is the smallest set satisfying (5.7), and is by our definition *optimal*.

The problem with this approach is that we do not know *a priori* the number $k = k(\varepsilon)$ and ordering a large number of $w(\mathfrak{u})$ is prohibitively expensive. Therefore, the parameters $\{w(\mathfrak{u})\}_{\mathfrak{u}\subset\mathbb{N}}$ will be ordered online. Actually, we propose two ways of constructing active sets. The first and simpler one produces what we call, *quasi-optimal* sets $\mathcal{U}_\varepsilon^{\mathrm{q-opt}}$ because it uses an approximate ordering of $\{w(\mathfrak{u})\}_{\mathfrak{u}\subset\mathbb{N}}$ and so cannot be guaranteed to produce sets that are optimal. The second construction, uses the proper ordering of $\{w(\mathfrak{u})\}_{\mathfrak{u}\subset\mathbb{N}}$ and produces optimal sets $\mathcal{U}_\varepsilon^{\mathrm{opt}}$. However, the results in Section 5.4 suggest that often the difference between both sets is very small; sometimes these sets are equal.

To allow for the ordering of the parameters $w(\mathfrak{u})$, we assume that they satisfy the following properties. Let ℓ be a given cardinality and write $\mathfrak{u} = (u_1, \ldots, u_\ell)$, where $u_1 < \cdots < u_\ell$. The two properties are:

1. If

$$\mathfrak{u} = (u_1, \ldots, u_\ell) \text{ and } \mathfrak{v} = (v_1, \ldots, v_\ell) \text{ with } v_j \geq u_j \text{ for all } j, \tag{5.12}$$

   then $w(\mathfrak{u}) \geq w(\mathfrak{v})$.
2. There exists $L \in \mathbb{N}$ such that for all $\ell > L$,

$$w(u_1, \ldots, u_\ell) \geq w(u_1, \ldots, u_\ell, u_{\ell+1}). \tag{5.13}$$

Note that in the case of product weights of the form (5.2), clearly property (5.12) holds and for property (5.13) we can take $L = \lceil c^{1/a}(p^*+1)^{-1/(ap^*)} \rceil$.

In the following sections we detail the two new constructions more precisely, the main computational issue is how to order the parameters $\{w(\mathfrak{u})\}_{\mathfrak{u}\subset\mathbb{N}}$. To make the presentation as general as possible we consider the following abstract problem given

arbitrary parameters $\{w(\mathfrak{u})\}_{\mathfrak{u}\subset\mathbb{N}}$ and a threshold T. The condition (5.11) can be written abstractly as: find a set $\mathcal{U}_\varepsilon$ of finite $\mathfrak{u} \subset \mathbb{N}$ such that

$$\sum_{\mathfrak{u}\subset\mathcal{U}_\varepsilon} w(\mathfrak{u}) \geq T\,, \tag{5.14}$$

where in the setting of this chapter the threshold is

$$T \;=\; T^{\mathrm{opt}}(\varepsilon) \;\coloneqq\; \sum_{\mathfrak{u}\subset\mathbb{N}} w(\mathfrak{u}) - \varepsilon^{p^*}\,. \tag{5.15}$$

The general idea of our construction is to select sets $\mathfrak{u}$ with the largest $w(\mathfrak{u})$ and subtract $w(\mathfrak{u})$ from T, repeating until $T \leq 0$. Obviously, in general we cannot order the entire sequence of parameters, and so they are ordered in batches by partitioning $[0,\infty)$ into intervals Φ_j such that numbers in Φ_j are greater than those in Φ_{j+1}. The algorithm works through the intervals in order as follows. For each interval Φ_k, the algorithm searches for all of the sets $\mathfrak{u}$ with $w(\mathfrak{u}) \in \Phi_k$, sorts them in decreasing order, and then adds the corresponding $\mathfrak{u}$ to $\mathcal{U}_\varepsilon^{\mathrm{opt}}$ until $\mathcal{U}_\varepsilon^{\mathrm{opt}}$ satisfies (5.14).

In the construction of quasi-optimal active sets $\mathcal{U}_\varepsilon^{\mathrm{q-opt}}$ we do not sort the weights inside the intervals, instead we add the sets as they occur until the condition (5.14) is satisfied. Since the interval structure provides some sense of ordering, the parameters in this case can be thought of as being "approximately ordered". The construction of quasi-optimal sets is presented first because it is simpler.

5.3.1 A simplified construction of quasi-optimal active sets

Consider a partition of $[0,\infty)$ into countably-many intervals $\{\Phi_j\}_{j=1}^{\infty}$, which are ordered such that for any $j \in \mathbb{N}$ if $x \in \Phi_j$ and $y \in \Phi_{j+1}$ then $y < x$. For simplicity, we used $\Phi_1 = [10^{-1},\infty)$, and $\Phi_j = [10^{-j}, 10^{-j+1})$ for $j = 2,3,\ldots$ in our numerical experiments. We also associate with every interval a list $\mathcal{L}_j$ that contains $\mathfrak{u}$ for with $w(\mathfrak{u}) \in \Phi_j$.

To search through the sets in a systematic way, we must keep track of the current set $\mathfrak{u}$ and the index i that we are incrementing from, see Section 4.1 since the strategy is the same as was used there. The subroutine `increment_u` outlined below in Algorithm 5.1 details how to increment $\mathfrak{u}$ from index i.

A naive version of the algorithm for constructing $\mathcal{U}_\varepsilon^{\mathrm{q-opt}}$ is as follows. For each $j = 1,2,3,\ldots$ we work through the sets $\mathfrak{u}$ in order of increasing cardinality $\ell = 1,2,3,\ldots$, if $w(\mathfrak{u}) \in \Phi_j$ then we add $\mathfrak{u}$ to $\mathcal{U}_\varepsilon^{\mathrm{q-opt}}$ and subtract $w(\mathfrak{u})$ from T. The algorithm terminates when $T \leq 0$, in which case $\mathcal{U}_\varepsilon^{\mathrm{q-opt}}$ is an active set satisfying (5.14).

Pseudocode 5.1 Subroutine: `increment_u`

inputs: $\mathfrak{u}$, i

output: $\mathfrak{u}$

1: **if** $i = 0$ **then** return $\mathfrak{u}$ ▷ don't update $\mathfrak{u}$
2: $u_i \leftarrow u_i + 1$ ▷ updating u_i first
3: **for** $r = i+1, i+2, \ldots, |\mathfrak{u}|$ **do** ▷ incrementing $\mathfrak{u}$ from index $i+1$
4: $\quad u_r \leftarrow u_i + r - i$
5: **end for**
6: return $\mathfrak{u}$

However, in this case at each interval we would begin the search with all of the sets that correspond to the previous intervals, resulting in a large number of unnecessary visits to sets. In order to reduce the double-handling of sets between intervals we use the lists $\mathcal{L}_j$ to keep track of the sets that were visited in step $j-1$ but that belong to Φ_j. Then, at the jth step we begin with the sets $\mathfrak{u} \in \mathcal{L}_j$ that were found in the previous step.

The efficient version of the algorithm for constructing $\mathcal{U}_\varepsilon^{\mathrm{q-opt}}$ also works through intervals in the order $j = 1, 2, 3, \ldots$, but then proceeds as follows. For each $\mathfrak{u} \in \mathcal{L}_j$, if $w(\mathfrak{u}) \in \Phi_j$ then we add $\mathfrak{u}$ to $\mathcal{U}_\varepsilon^{\mathrm{q-opt}}$ and subtract $w(\mathfrak{u})$ from T. If $T \leq 0$ then $\mathcal{U}_\varepsilon^{\mathrm{q-opt}}$ satisfies (5.14) and we terminate the algorithm, otherwise we increment $\mathfrak{u}$. Whereas, if $w(\mathfrak{u}) \notin \Phi_j$ then we add $\mathfrak{u}$ to $\mathcal{L}_{j+1}$ and move to the next $\mathfrak{u} \in \mathcal{L}_j$.

Once we have searched through $\mathcal{L}_j$ we continue the search in increasing cardinality $\ell = \ell_{\mathrm{next}}, \ell_{\mathrm{next}} + 1, \ldots$, starting at $\ell_{\mathrm{next}} = \max_{\mathfrak{u} \in \mathcal{L}_j}(|\mathfrak{u}|) + 1$. For each $\mathfrak{u}$ with $|\mathfrak{u}| = \ell$, if $w(\mathfrak{u}) \in \Phi_j$ then we add $\mathfrak{u}$ to $\mathcal{U}_\varepsilon^{\mathrm{q-opt}}$, subtract $w(\mathfrak{u})$ from T, check for termination ($T \leq 0$) and increment $\mathfrak{u}$ from the final index. Otherwise, we add $\mathfrak{u}$ to $\mathcal{L}_{j+1}$ and increment $\mathfrak{u}$ from a lower index. When we can no longer increment $\mathfrak{u}$ (the incrementing index i is 0) we proceed to the next cardinality. We continue increasing the cardinality until we reach some ℓ such that $\ell > L$ and $\{1, 2, \ldots, \ell\} \notin \Phi_j$, in which case, because of the two properties (5.12)–(5.13), we know that there are no more sets with $w(\mathfrak{u}) \in \Phi_j$ and we proceed to the next interval.

Restarting the search at cardinality ℓ_{next} does not skip any sets because the way we increment $\mathfrak{u}$ remains the same throughout. Hence all of the sets for the current interval with cardinality $|\mathfrak{u}| < \ell_{\mathrm{next}}$ have been visited when incrementing from $\mathfrak{u} \in \mathcal{L}_j$.

The main procedure is outlined in Algorithm 5.2. In all of the algorithms $j_{\max}$ and $\ell_{\max}$ are computational thresholds denoting, respectively, the maximum number of intervals to be searched through and the maximum allowed cardinality of sets.

To make the presentation clearer Algorithm 5.2 is broken into two parts: First, we search starting from the sets found in the previous interval, which is handled by the subroutine `q-opt_search` in Algorithm 5.3. Then we continue searching through sets in order of increasing cardinality (line 9) starting where `q-opt_search` finished, at cardinality ℓ_{next}. The basic search structure is the same, however in `q-opt_search` each set we visit is checked to reduce multiple visits and ensure that the same set is not added to $\mathcal{U}_\varepsilon^{\text{q-opt}}$ more than once.

The notation

$$(\mathcal{U}_\varepsilon^{\text{q-opt}}, T, \ell_{\text{next}}, \mathcal{L}_{j+1}) \leftarrow \texttt{q-opt\_search}(\mathcal{U}_\varepsilon^{\text{q-opt}}, T, (w(\mathfrak{u}))_{\mathfrak{u}\subset\mathbb{N}}, \mathcal{L}_j, \mathcal{L}_{j+1}, \Phi_j)$$

denotes that we call `q-opt_search` with inputs $\mathcal{U}_\varepsilon^{\text{q-opt}}$, T, $(w(\mathfrak{u}))_{\mathfrak{u}\subset\mathbb{N}}$, $\mathcal{L}_j$, $\mathcal{L}_{j+1}$, Φ_j and then use the output to update $\mathcal{U}_\varepsilon^{\text{q-opt}}$, T, ℓ_{next} and $\mathcal{L}_{j+1}$.

5.3.2 Construction of optimal active sets

The construction of optimal active sets is very similar. The main difference is that in the jth step, we first create the list $\mathcal{L}_j$ and order its elements $\mathfrak{u}$ according to decreasing values of $w(\mathfrak{u})$, then only after this start subtracting the values $w(\mathfrak{u})$ from T. Again we terminate when $T \leq 0$ and the lists $\mathcal{L}_j$ will hold the sets visited in the previous interval.

In fact, if we do not care whether or not all of the sets are ordered but only that $\mathcal{U}_\varepsilon^{\text{opt}}$ consists of the sets with the largest weights, then we only need to sort the sets that come from the final interval. This is because at the previous intervals all of the sets will need to be added to $\mathcal{U}_\varepsilon^{\text{opt}}$, regardless of sorting. To do this in practice, for each interval we store the sum of all the weights corresponding to that interval. In Algorithm 5.4 we denote this by T_j. At the end of the jth step, we check whether Φ_j is the final interval, i.e., if $T - T_j \leq 0$, if so we sort the sets and add them one-by-one until the active set is complete. Otherwise we add all of the sets in $\mathcal{L}_j$ to $\mathcal{U}_\varepsilon^{\text{opt}}$, subtract T_j from T and go to the next interval. For completeness, the construction of optimal active sets is detailed separately below in Algorithm 5.4 and the subroutines `opt_search` in Algorithm 5.5, which handles the search from sets visited in the previous interval, and `opt_sort` in Algorithm 5.6, which handles the sorting component.

Pseudocode 5.2 Constructing the quasi-optimal active set

inputs: T, $\{w(\mathfrak{u})\}_{\mathfrak{u}\subset\mathbb{N}}$, $(\Phi_j)_{j=1}^{j_{\max}}$

output: $\mathcal{U}_\varepsilon^{\mathrm{q-opt}}$

$\mathcal{L}_j \leftarrow \emptyset$ for all $j = 1, 2, \ldots, j_{\max}$ ▷ initialising
$\mathcal{U}_\varepsilon^{\mathrm{q-opt}} \leftarrow \{\emptyset\}$
$T \leftarrow T - w(\emptyset)$ ▷ decrease T with each set added
if $T \leq 0$ **then** return $\mathcal{U}_\varepsilon^{\mathrm{q-opt}}$ ▷ quasi-optimal active set is complete
for $j = 1, 2, \ldots, j_{\max}$ **do** ▷ start at sets found previously
    $(\mathcal{U}_\varepsilon^{\mathrm{q-opt}}, T, \ell_{\mathrm{next}}, \mathcal{L}_{j+1}) \leftarrow$`q-opt_search`$(\mathcal{U}_\varepsilon^{\mathrm{q-opt}}, T, (w(\mathfrak{u}))_{\mathfrak{u}\subset\mathbb{N}}, \mathcal{L}_j, \mathcal{L}_{j+1}, \Phi_j)$
    if $T \leq 0$ **then** return $\mathcal{U}_\varepsilon^{\mathrm{q-opt}}$ ▷ quasi-optimal active set is complete
    for $\ell = \ell_{\mathrm{next}}, \ell_{\mathrm{next}} + 1, \ldots, \ell_{\max}$ **do** ▷ search through unvisited sets
        $\mathfrak{u} = \{1, 2, \ldots, \ell\}$
        $i \leftarrow \ell$ ▷ i keeps track of index to increment $\mathfrak{u}$ from
        if $w(\mathfrak{u}) \notin \Phi_j$ **and** $\ell > L$ **then** break ▷ no more $\mathfrak{u}$ with $w(\mathfrak{u}) \in \Phi_j$
        while $i > 0$ **do** ▷ when $i = 0$ there are no more $\mathfrak{u}$ of cardinality ℓ
            if $w(\mathfrak{u}) \in \Phi_j$ **then**
                add $\mathfrak{u}$ to $\mathcal{U}_\varepsilon^{\mathrm{q-opt}}$
                $T \leftarrow T - w(\mathfrak{u})$
                if $T \leq 0$ **then** return $\mathcal{U}_\varepsilon^{\mathrm{q-opt}}$ ▷ quasi-optimal set is complete
                $i \leftarrow \ell$ ▷ continue incrementing from last index
            else
                add $\mathfrak{u}$ to $\mathcal{L}_{j+1}$
                $i \leftarrow i - 1$ ▷ increment from lower index
            end if
            $\mathfrak{u} \leftarrow$`increment_u`$(\mathfrak{u}, i)$
        end while
    end for
end for

Pseudocode 5.3 Subroutine: `q-opt_search`

inputs: $\mathcal{U}_\varepsilon^{\mathrm{q-opt}}$, T, $(w(\mathfrak{u}))_{\mathfrak{u}\subset\mathbb{N}}$, $\mathcal{L}_j$, $\mathcal{L}_{j+1}$, Φ_j

outputs: $\mathcal{U}_\varepsilon^{\mathrm{q-opt}}$, ℓ_{next}, T, $\mathcal{L}_{j+1}$

$\ell_{\mathrm{next}} = 1$
for $\mathfrak{u} \in \mathcal{L}_j$ **do**
    $i \leftarrow |\mathfrak{u}|$
    while $i > 0$ **do**
        $\mathcal{L}_j \leftarrow \mathcal{L}_j \setminus \mathfrak{u}$ ▷ reducing the double-handling of sets
        if $w(\mathfrak{u}) \in \Phi_j$ **then**
            if $\mathfrak{u} \in \mathcal{U}_\varepsilon^{\mathrm{q-opt}}$ **then** break ▷ already visited future increments
            add $\mathfrak{u}$ to $\mathcal{U}_\varepsilon^{\mathrm{q-opt}}$
            $T \leftarrow T - w(\mathfrak{u})$
            if $T \leq 0$ **then** return $(\mathcal{U}_\varepsilon^{\mathrm{q-opt}}, \ell_{\mathrm{next}}, T, \mathcal{L}_{j+1})$
            $i = |\mathfrak{u}|$ ▷ continue incrementing from last index
        else
            add $\mathfrak{u}$ to $\mathcal{L}_{j+1}$ ▷ $\mathfrak{u}$ to be checked first in next interval
            $i \leftarrow i - 1$ ▷ increment from lower index
        end if
        $\mathfrak{u} \leftarrow$`increment_u`$(\mathfrak{u}, i)$
    end while
    $\ell_{\mathrm{next}} = |\mathfrak{u}| + 1$ ▷ main search will start at cardinality $|\mathfrak{u}| + 1$
end for
return $(\mathcal{U}_\varepsilon^{\mathrm{q-opt}}, \ell_{\mathrm{next}}, T, \mathcal{L}_{j+1})$

Pseudocode 5.4 Constructing the optimal active set

inputs: T, $\{w(\mathfrak{u})\}_{\mathfrak{u}\subset\mathbb{N}}$, $(\Phi_j)_{j=1}^{j_{\max}}$

output: $\mathcal{U}_\varepsilon^{\text{opt}}$

$T_j \leftarrow 0$, $\mathcal{L}_j \leftarrow \emptyset$ for all $j = 1, 2, \ldots, j_{\max}$ ▷ initialising
$\mathcal{U}_\varepsilon^{\text{opt}} \leftarrow \{\emptyset\}$
$T \leftarrow T - w(\emptyset)$ ▷ decrease T with each set added
if $T \leq 0$ **then** return $\mathcal{U}_\varepsilon^{\text{opt}}$ ▷ optimal active set is complete
for $j = 1, 2, \ldots, j_{\max}$ **do** ▷ start at sets found in previous step
    $(\ell_{\text{next}}, T_j, \mathcal{L}_j, \mathcal{L}_{j+1}) \leftarrow$`opt_search`$(\mathcal{U}_\varepsilon^{\text{opt}}, T_j, \{w(\mathfrak{u})\}_{\mathfrak{u}\subset\mathbb{N}}, \mathcal{L}_j, \mathcal{L}_{j+1}, \Phi_j)$
    for $\ell = \ell_{\text{next}}, \ell_{\text{next}} + 1, \ldots, \ell_{\max}$ **do** ▷ search through unvisited sets
        $\mathfrak{u} = \{1, 2, \ldots, \ell\}$
        $i \leftarrow \ell$ ▷ i keeps track of index to increment $\mathfrak{u}$ from
        if $w(\mathfrak{u}) \notin \Phi_j$ **and** $\ell > L$ **then** break ▷ no more $\mathfrak{u}$ with $w(\mathfrak{u}) \in \Phi_j$
        while $i > 0$ **do** ▷ when $i = 0$ there are no more $\mathfrak{u}$ of cardinality ℓ
            if $w(\mathfrak{u}) \in \Phi_j$ **then**
                add $\mathfrak{u}$ to $\mathcal{L}_j$
                $T_j \leftarrow T_j + w(\mathfrak{u})$
                $i \leftarrow \ell$ ▷ continue incrementing from last index
            else
                add $\mathfrak{u}$ to $\mathcal{L}_{j+1}$
                $i \leftarrow i - 1$ ▷ increment from lower index
            end if
            $\mathfrak{u} \leftarrow$`increment_u`$(\mathfrak{u}, i)$
        end while
    end for ▷ sort the current sets
    $(T, \mathcal{U}_\varepsilon^{\text{opt}}) \leftarrow$ `opt_sort` $(\mathcal{U}_\varepsilon^{\text{opt}}, T, T_j, \{w(\mathfrak{u})\}_{\mathfrak{u}\subset\mathbb{N}}, \mathcal{L}_j)$
    if $T \leq 0$ **then** return $\mathcal{U}_\varepsilon^{\text{opt}}$
end for

Pseudocode 5.5 Subroutine: `opt_search`

inputs: $\mathcal{U}_\varepsilon^{\text{opt}}$, T_j, $\{w(\mathfrak{u})\}_{\mathfrak{u}\subset\mathbb{N}}$, $\mathcal{L}_j$, $\mathcal{L}_{j+1}$, Φ_j

outputs: ℓ_{next}, T_j, $\mathcal{L}_j$, $\mathcal{L}_{j+1}$

$\ell_{\text{next}} = 1$
for $\mathfrak{u} \in \mathcal{L}_j$ **do**
    $i \leftarrow |\mathfrak{u}|$
    while $i > 0$ **do**
        $\mathcal{L}_j \leftarrow \mathcal{L}_j \setminus \mathfrak{u}$ ▷ reducing the double-handling of sets
        if $w(\mathfrak{u}) \in \Phi_j$ **then**
            if $\mathfrak{u} \in \mathcal{L}_j$ **then** break ▷ already visited $\mathfrak{u}$ and future increments
            add $\mathfrak{u}$ to $\mathcal{L}_j$, $T_j \leftarrow T_j + w(\mathfrak{u})$, $i \leftarrow |\mathfrak{u}|$
        else
            add $\mathfrak{u}$ to $\mathcal{L}_{j+1}$, $i \leftarrow i - 1$ ▷ $\mathfrak{u}$ to be checked first in next interval
        end if
        $\mathfrak{u} \leftarrow$`increment_u`$(\mathfrak{u}, i)$
    end while
    $\ell_{\text{next}} = |\mathfrak{u}| + 1$ ▷ main search will start at cardinality $|\mathfrak{u}| + 1$
end for
return $(\ell_{\text{next}}, T_j, \mathcal{L}_j, \mathcal{L}_{j+1})$

Pseudocode 5.6 Subroutine: `opt_sort`

inputs: $\mathcal{U}_\varepsilon^{\text{opt}}$, T, T_j, $\{w(\mathfrak{u})\}_{\mathfrak{u}\subset\mathbb{N}}$, $\mathcal{L}_j$

outputs: T, $\mathcal{U}_\varepsilon^{\text{opt}}$

if $T_j \geq T$ **then** ▷ first check if Φ_j was the last interval
    sort $\mathcal{L}_j$
    for $\mathfrak{u} \in \mathcal{L}_j$ **do** ▷ add sorted sets until active set is complete
        add $\mathfrak{u}$ to $\mathcal{U}_\varepsilon^{\text{opt}}$
        $T \leftarrow T - w(\mathfrak{u})$
        if $T \leq 0$ **then** break ▷ optimal active set is complete
    end for
else ▷ add all sets for the current interval and continue search
    add all $\mathfrak{u}$ to $\mathcal{U}_\varepsilon^{\text{opt}}$
    $T \leftarrow T - T_j$
end if
return T, $\mathcal{U}_\varepsilon^{\text{opt}}$

5.3.3 Computing the threshold T^{opt}

The algorithms for constructing active sets require the threshold $T^{\text{opt}}(\varepsilon)$ as input, and to compute $T^{\text{opt}}(\varepsilon)$ we must approximate

$$S := \sum_{\mathfrak{u}\subset\mathbb{N}} w(\mathfrak{u}) = \sum_{\mathfrak{u}\subset\mathbb{N}} \prod_{j\in\mathfrak{u}} \frac{(c/j^a)^{p^*}}{p^*+1} = \prod_{j=1}^{\infty}\left(1+\frac{(c/j^a)^{p^*}}{p^*+1}\right)$$

from above and with the relative error significantly smaller than ε^{p^*}.

This can be done as follows. For a large natural number t

$$\begin{aligned} S &= \exp\left(\ln\left(\prod_{j=t+1}^{\infty}\left(1+\frac{(c/j^a)^{p^*}}{p^*+1}\right)\right)\right)\prod_{j=1}^{t}\left(1+\frac{(c/j^a)^{p^*}}{p^*+1}\right) \\ &\le \exp\left(\frac{c^{p^*}}{p^*+1}\sum_{j=t+1}^{\infty} j^{-a\,p^*}\right)\prod_{j=1}^{t}\left(1+\frac{(c/j^a)^{p^*}}{p^*+1}\right) \\ &\le \exp\left(\frac{c^{p^*}}{p^*+1}\int_{t+1/2}^{\infty} x^{-a\,p^*}\mathrm{d}x\right)\prod_{j=1}^{t}\left(1+\frac{(c/j^a)^{p^*}}{p^*+1}\right) \\ &= \exp\left(\frac{c^{p^*}}{(p^*+1)\,(a\,p^*-1)\,(t+1/2)^{a\,p^*-1}}\right)\prod_{j=1}^{t}\left(1+\frac{(c/j^a)^{p^*}}{p^*+1}\right) =: S_t. \qquad (5.16) \end{aligned}$$

See also Remark 4.6. It is easy to see that the relative error between S and its approximation S_t is proportional to $1/t^{2a\,p^*-2}$ with the asymptotic constant

$$\frac{c^{p^*}}{(p^*+1)\,2^{ap^*-1}}\prod_{j=1}^{\infty}\left(1+\frac{(c/j^a)^{p^*}}{p^*+1}\right).$$

Although it only forms a small portion of this chapter, the accurate upper bound (5.16) is a crucial component of the two new constructions. Suppose we could not estimate the sum to within an accuracy of ε^{p^*}, then we would have that the input threshold is such that $T > S$ and both algorithms would never terminate.

5.4 Active set results

In this section we present results on the size of the active sets for parameters $p = 1, 2, \infty$, weights with decays of $a = 2, 3, 4$ and error requests $\varepsilon = 10^{-1}, 10^{-2}, 10^{-3}$. A collection of the active sets constructed above have been listed in full in the Appendix.

For $p = 1$ the active sets $\mathcal{U}_\varepsilon^{\text{PW}}$ are optimal and the results are given in Table 5.1.

For $p > 1$ the active sets are no longer optimal and the results for $p = 2, \infty$ are given in Tables 5.2 and 5.3, respectively. These results indicate that as p increases the active sets $\mathcal{U}_\varepsilon^{\text{PW}}$ are much larger than is necessary.

	$a=4$			$a=3$			$a=2$		
ε	10^{-1}	10^{-2}	10^{-3}	10^{-1}	10^{-2}	10^{-3}	10^{-1}	10^{-2}	10^{-3}
$\sigma(\mathcal{U}_\varepsilon^{\mathrm{PW}})$	1	2	2	2	2	3	2	3	4
$\lvert\mathcal{U}_\varepsilon^{\mathrm{PW}}\rvert$	2	6	10	4	8	22	6	22	114

Table 5.1: Active set results for $p = 1$ and $a = 4, 3, 2$, $\varepsilon = 10^{-1}, 10^{-2}, 10^{-3}$.

	$a=4$			$a=3$			$a=2$		
ε	10^{-1}	10^{-2}	10^{-3}	10^{-1}	10^{-2}	10^{-3}	10^{-1}	10^{-2}	10^{-3}
$\sigma(\mathcal{U}_\varepsilon^{\mathrm{opt}})$	1	2	2	1	2	3	2	3	4
$\lvert\mathcal{U}_\varepsilon^{\mathrm{opt}}\rvert$	2	4	9	2	7	24	4	30	255
$\lvert\mathcal{U}_\varepsilon^{\mathrm{q-opt}}\rvert$	2	4	9	2	7	26	6	32	261
$\lvert\mathcal{U}_\varepsilon^{\mathrm{PW}}\rvert$	3	8	20	5	18	70	15	158	1481

Table 5.2: Active set results for $p = 2$ and $a = 4, 3, 2$, $\varepsilon = 10^{-1}, 10^{-2}, 10^{-3}$.

	$a=4$			$a=3$			$a=2$		
ε	10^{-1}	10^{-2}	10^{-3}	10^{-1}	10^{-2}	10^{-3}	10^{-1}	10^{-2}	10^{-3}
$\sigma(\mathcal{U}_\varepsilon^{\mathrm{opt}})$	1	2	2	1	2	3	3	4	6
$\lvert\mathcal{U}_\varepsilon^{\mathrm{opt}}\rvert$	2	5	15	3	15	83	33	1346	45,446
$\lvert\mathcal{U}_\varepsilon^{\mathrm{q-opt}}\rvert$	2	5	15	3	15	92	38	1904	52,159
$\lvert\mathcal{U}_\varepsilon^{\mathrm{PW}}\rvert$	7	21	72	21	149	923	2358	120,935	

Table 5.3: Active set results for $p = \infty$ and $a = 4, 3, 2$, $\varepsilon = 10^{-1}, 10^{-2}, 10^{-3}$.

To observe how different choices of parameters a and c affect our construction, statistics on the resulting optimal active sets are given in Tables 5.4-5.7. Tables 5.4 and 5.5 give, respectively, the size and the superposition dimension of the optimal active set for $p = 2$ and an error request of 10^{-2}. For $p = \infty$, $\varepsilon = 10^{-2}$ the size and superposition dimension of the optimal active sets are given in Tables 5.6 and 5.7. The results for the quasi-optimal active set are again very similar and so have not been included here. As expected these results demonstrate that as the decay of the weights is slower or the weights become larger (a smaller and c larger) the problem becomes more difficult and the active sets are by necessity larger. However the superposition dimension remains relatively small, at most 6.

	a		
c	4	3	2
$\frac{1}{2}$	3	5	12
1	4	7	30
2	6	14	122

Table 5.4: $\lvert\mathcal{U}_\varepsilon^{\mathrm{opt}}(10^{-2})\rvert$ for $p = 2$ and different a, c.

	a		
c	4	3	2
$\frac{1}{2}$	1	2	2
1	2	2	3
2	2	3	4

Table 5.5: $\sigma(\mathcal{U}_\varepsilon^{\mathrm{opt}}(10^{-2}))$ for $p = 2$ and different a, c.

	a		
c	4	3	2
$\frac{1}{2}$	4	7	150
1	5	15	1346
2	8	43	31,013

Table 5.6: $|\mathcal{U}_\varepsilon^{\text{opt}}(10^{-2})|$ for $p = \infty$ and different a, c.

	a		
c	4	3	2
$\frac{1}{2}$	2	3	4
1	2	2	4
2	2	3	6

Table 5.7: $\sigma(\mathcal{U}_\varepsilon^{\text{opt}})$ for $\varepsilon = 10^{-2}$, $p = \infty$ and different a, c.

5.5 Conclusion

In this chapter we have introduced the notion of optimal active sets to be used in the MDM for infinite-variate integration, and presented an algorithm detailing their construction. By simplifying the sorting of the parameters, we also introduced a second computationally less intensive version of the algorithm that constructs quasi-optimal active sets. Our numerical results show that the quasi-optimal active sets are of a similar size to the optimal active sets; often the two sets are exactly the same. In all of our numerical results the optimal and quasi-optimal active sets are smaller than, and have superposition dimension less than or equal to, the active sets using the previous construction from [75].

Original research and my contribution

The work in this chapter was performed in collaboration with Grzegorz Wasilkowsi, and corresponds to a paper that was published in the *Journal of Complexity* in 2017 (see [33]).

The following topics have all been studied previously: the MDM algorithm, see [58, 63, 75, 76, 89, 90]; weighted function space setting, see [43]; and the first method of constructing active sets, which we denoted $\mathcal{U}_\varepsilon^{\text{PW}}$, see [75]. The original content of this chapter is the notion of optimal active sets and the two algorithms for constructing optimal and quasi-optimal active sets. Loosely speaking, everything in Sections 5.3 and 5.4 is new.

The ideas behind the algorithms for optimal and quasi-optimal active sets were due to Grzegorz Wasilkowski. I was responsible for implementing the two algorithms, formalising them into the pseudocodes and performing the numerical results. I also contributed to the structure of the two algorithms.

Chapter 6

Blackbox CBC algorithms for constructing lattice rules

The aim of the current chapter is to develop efficient and user-friendly methods for generating randomly shifted lattices rules to be used for the numerical computation of high-dimensional integrals of the form

$$\mathcal{I}_s(f) \coloneqq \int_{[0,1]^s} f(\boldsymbol{y}) \, \mathrm{d}\boldsymbol{y} \,. \tag{6.1}$$

This chapter introduces two new variants of the component-by-component (CBC) algorithm that, given bounds on the mixed first derivatives, choose not only the QMC points but also the function space weights, with a view to minimising the error of the QMC approximation. Although still playing an important role in our constructions, the weight parameters no longer need to be chosen by the practitioner because this choice is handled automatically inside the algorithms.

We begin with an informal discussion of the role that the weights play in both theory and practice, also considering previous strategies for choosing the weights.

6.1 How to choose the function space weights in practice?

Recall from Section 2.3 of Chapter 2 that the setting for the error analysis of a QMC approximation to (6.1) assumes that the integrand f belongs to the weighted function space $\mathcal{W}_{s,\boldsymbol{\gamma}}$, where in general for each $\mathfrak{u} \subset \{1:s\}$ the weight $\gamma_{\mathfrak{u}}$ represents the importance of the variables $\boldsymbol{y}_{\mathfrak{u}}$. For simplicity, in this introductory section we shall concentrate on product weights: $\gamma_{\mathfrak{u}} = \prod_{i \in \mathfrak{u}} \gamma_i$.

More precisely, in the weighted function space $\mathcal{W}_{s,\boldsymbol{\gamma}}$ the root-mean-square error of a randomly shifted QMC approximation satisfies

$$\sqrt{\mathbb{E}_{\boldsymbol{\Delta}} \left[\left| \mathcal{I}_s(f) - Q^{\mathrm{sh}}_{n,s}(\mathcal{P}_n; \cdot) f \right|^2 \right]} \le e^{\mathrm{sh}}_{n,s,\boldsymbol{\gamma}}(\mathcal{P}_n) \, \|f\|_{s,\boldsymbol{\gamma}} \,. \tag{6.2}$$

Even though the right hand side of this error bound conveniently separates into the product of two factors—one which depends only on the quadrature points and the

other which depends only on the integrand—a key aspect of the work in the current chapter is that both the worst-case error and the norm depend on the weights.

In this chapter the user is assumed to have information about the norm $\|f\|_{s,\boldsymbol{\gamma}}$ (as defined in (2.5)) in the form of estimates on the size of the mixed first derivatives of f. This information is given by the parameters B_ℓ and β_i in the following assumption.

Standing Assumption for Chapter 6: *For two sequences of positive real numbers $(B_\ell)_{\ell=1}^s$ and $(\beta_i)_{i=1}^s$, we assume that the mixed first derivatives of the integrand satisfy the following family of upper bounds, for each $\mathfrak{u} \subseteq \{1, 2, \ldots, s\}$:*

$$\int_{[0,1]^{|\mathfrak{u}|}} \left(\int_{[0,1]^{s-|\mathfrak{u}|}} \frac{\partial^{|\mathfrak{u}|}}{\partial \boldsymbol{y}_{\mathfrak{u}}} f(\boldsymbol{y}) \, \mathrm{d}\boldsymbol{y}_{-\mathfrak{u}} \right)^2 \mathrm{d}\boldsymbol{y}_{\mathfrak{u}} \le B_{|\mathfrak{u}|} \prod_{j \in \mathfrak{u}} \beta_j^2 \,. \tag{6.3}$$

Bounds of the form given in (6.3), together with explicit values of B_ℓ and β_i, have been found for certain PDE problems in several recent papers, including [60] and also the eigenvalue problem studied in Chapter 3.

The QMC rules used in this chapter are randomly shifted lattice rules, which will be generated by our new variants of the CBC algorithm. The original CBC construction works through each dimension $i = 1, 2, \ldots, s$ sequentially, choosing the ith component of the rule by minimising the worst-case error in that dimension while all previous components remain fixed. Because the worst-case error depends explicitly on the weights, the CBC algorithm requires the weights as inputs. Thus, the weights are not only useful for theory, but they are also a necessity for generating good lattice rules in practice.

The CBC construction also has the virtue that, as was shown in Section 2.3.3, the worst-case error of the resulting QMC approximation converges to zero at a rate that, depending on the weights, can be arbitrarily close to n^{-1}, with a constant that can be independent of s. From (6.2) and (2.16), the root-mean-square error of a CBC generated randomly shifted lattice rule approximation of $\mathcal{I}_s(f)$ (in the special case of prime n and product weights) is bounded above by

$$\left(\frac{1}{n-1} \prod_{i=1}^{s} \left(1 + \gamma_i^\eta \frac{2\zeta(2\eta)}{(2\pi^2)^\eta} \right) \right)^{\frac{1}{2\eta}} \|f\|_{s,\boldsymbol{\gamma}} \quad \text{for all } \eta \in \left(\tfrac{1}{2}, 1\right] , \tag{6.4}$$

where $\zeta(x) = \sum_{k=1}^\infty k^{-x}$ is the Riemann zeta function.

Until recently the choice of weights was generally *ad hoc* or assumed to be given, but in the paper [60] a new principle was used to determine weights for a particular problem: having estimated an upper bound on the norm of the integrand (which like the worst-case error depends on the weights), those authors chose weights that

minimise an upper bound on the error (6.4). (Note that [60] dealt with a specific problem of randomly shifted lattice rules applied to PDEs with random coefficients, however the strategy of that paper can easily be applied to other problems.) The result is a family of weight sequences indexed by the parameter η, where η affects the theoretical rate of convergence. The fact that η must be chosen by the user is a major drawback of the strategy in [60]. One option would be to take η as close to $\frac{1}{2}$ as possible to ensure a good convergence rate, however, because of the occurrence of the zeta function $\zeta(2\eta)$, the constant goes to ∞ as $\eta \to \frac{1}{2}$. A good rate of convergence does not help for a fixed value of n if the constant becomes too large. To obtain the best bound a delicate balance between the two factors in (6.4) is needed.

Another drawback of the method used in [60] is that the bound (6.4) is often a crude overestimate. The first algorithm we introduce in this chapter, the **double CBC (DCBC) algorithm**, counters this while at the same time removing the need to choose η by dealing with the exact shift-averaged worst-case error (see Section 2.3), rather than the upper bound given by the first factor in (6.4). For the case of product weights, at step i of the DCBC algorithm, after fixing the component of the lattice rule, the weight γ_i is chosen so as to minimise a bound on the error in the current dimension. An advantage of this method is that the choice of weight in each dimension adds virtually no extra computational cost to the algorithm.

The second algorithm we propose begins with the upper bound (6.4), and hence the family of weights indexed by η obtained following the strategy in [60]. To choose the "best" η, and in turn the "best" weights, an iteration of the CBC algorithm with respect to η is employed to minimise heuristically a bound on the approximation error. Because of this iteration process it is called the **iterated CBC (ICBC) algorithm**.

The philosophy of both algorithms is to concentrate on reducing the guaranteed error bounds. This paradigm represents a shift away from the usual focus on the best rate of convergence. It is particularly useful when dealing with problems where function evaluations are highly expensive, such as those where QMC methods have been applied to PDEs with random coefficients [60], for which there is often a practical limit on the number of quadrature points n.

So far we have discussed only product weights, however, both algorithms can be extended to cover a more general form of weights called "POD weights" (see Section 2.3).

The structure of this chapter is as follows. Details on our two new algorithms are given in Sections 6.2 and 6.3. Then in Section 6.4 we give tables for the guaranteed

error bounds resulting from the two algorithms, under various assumptions on the parameters B_ℓ and β_i in (6.3). The examples show that there are different situations where each of the algorithms outperforms the other, thus it is not possible to say that one algorithm is superior.

6.2 The double CBC algorithm

The algorithm introduced in this section (along with the algorithm in the next section) aims to choose weights so as to make the bound (6.2) on the root-mean-square error of the QMC approximation as small as possible. Hence we will require a bound on the norm of the integrand $f \in \mathcal{W}_{s,\boldsymbol{\gamma}}$ to be known and of the specific form given in the Assumption (6.3) in Section 6.1.

In this way the norm of f in $\mathcal{W}_{s,\boldsymbol{\gamma}}$, see (2.5), with some, as yet unspecified, weights $\boldsymbol{\gamma}$ will be bounded by

$$\|f\|_{s,\boldsymbol{\gamma}}^2 \le \sum_{\mathfrak{u}\subseteq\{1:s\}} \frac{1}{\gamma_{\mathfrak{u}}} B_{|\mathfrak{u}|} \prod_{j\in\mathfrak{u}} \beta_j^2 =: M_{s,\boldsymbol{\gamma}}\,, \tag{6.5}$$

and in turn from (6.2) the mean-square error of a lattice rule approximation will be bounded by

$$\mathbb{E}_{\boldsymbol{\Delta}}\left[\left|\mathcal{I}_s(f) - Q_{n,s}^{\mathrm{sh}}(\boldsymbol{z};\cdot)f\right|^2\right] \le \left(e_{n,s,\boldsymbol{\gamma}}^{\mathrm{sh}}(\boldsymbol{z})\right)^2 M_{s,\boldsymbol{\gamma}}\,. \tag{6.6}$$

The first new algorithm is named the **double CBC (DCBC) algorithm** since at each step two parameters are chosen: the component of the generating vector and the weight. We assume the weights are of product (2.6) or POD (2.8) form. In the case of POD weights we assume that the order dependent weight factors $(\Gamma_\ell)_{\ell=0}^s$ are given. Starting with the error bound (6.6), in each dimension, with all previous parameters remaining fixed, we choose the component of $\boldsymbol{z}$ to minimise $e_{n,s,\boldsymbol{\gamma}}^{\mathrm{sh}}$ and then the product component of the weight to minimise the entire bound.

6.2.1 The double CBC algorithm for product weights

In the case of product weights, the squared shift-averaged worst-case error (see (2.15)) of a lattice rule with generating vector $\boldsymbol{z}$ is

$$\begin{aligned}\left(e_{n,s,\boldsymbol{\gamma}}^{\mathrm{sh}}(z_1,\dots,z_s)\right)^2 &= -1 + \frac{1}{n}\sum_{k=0}^{n-1}\prod_{j=1}^{s}\left(1+\gamma_j\mathscr{B}_2\left(\left\{\frac{kz_j}{n}\right\}\right)\right)\\ &= \left(e_{n,s-1,\boldsymbol{\gamma}}^{\mathrm{sh}}(z_1,\dots,z_{s-1})\right)^2 + \gamma_s G_s(z_1,\dots,z_s)\,,\end{aligned}$$

in which the first term is independent of γ_s, and

$$G_s(z_1,\dots,z_s) := \frac{1}{n}\sum_{k=0}^{n-1}\mathscr{B}_2\left(\left\{\frac{kz_s}{n}\right\}\right)\prod_{j=1}^{s-1}\left(1+\gamma_j\mathscr{B}_2\left(\left\{\frac{kz_j}{n}\right\}\right)\right).$$

For product weights it is natural to assume that the bound on the norm is also of product form, that is, $B_\ell = 1$ for all $\ell = 1, 2, \dots, s$. It follows that this bound (6.5) can also be written recursively as

$$M_{s,\boldsymbol{\gamma}} = \sum_{\mathfrak{u}\subseteq\{1:s\}}\prod_{j\in\mathfrak{u}}\frac{\beta_j^2}{\gamma_j} = \prod_{j=1}^{s}\left(1+\frac{\beta_j^2}{\gamma_j}\right) = \left(1+\frac{\beta_s^2}{\gamma_s}\right)M_{s-1,\boldsymbol{\gamma}}\,. \tag{6.7}$$

In this situation, the bound (6.6) on the mean-square error can be written as

$$\left(\left(e^{\mathrm{sh}}_{n,s-1,\boldsymbol{\gamma}}(z_1,\dots,z_{s-1})\right)^2+\gamma_s G_s(z_1,\dots,z_s)\right)\left(1+\frac{\beta_s^2}{\gamma_s}\right)M_{s-1,\boldsymbol{\gamma}}\,. \tag{6.8}$$

Treating (6.8) as a function of γ_s and z_s, and noting that z_s is only present in G_s, at each step of the algorithm we can first choose z_s to minimise G_s and then choose γ_s to minimise the entire error bound. For future reference, the minimiser of expressions of this form is given by the following Lemma.

Lemma 6.1. *Suppose that a, b, c, d are positive real numbers. Then the function $h : (0,\infty) \to (0,\infty)$ given by $h(x) = (a+bx)(c+\frac{d}{x})$ is minimised by $x^* = \sqrt{\frac{ad}{bc}}$.*

Proof. The first two derivatives of h with respect to x are $h'(x) = bc - ad/x^2$ and $h''(x) = 2ad/x^3 > 0$ for $x > 0$, so h is convex. Solving $h'(x) = 0$ yields the formula for the stationary point x^*, which is the global minimum. □

Consequently, the choice of weight that minimises the bound on the mean-square error (6.8) is given by, with s replaced by i,

$$\gamma_i = \sqrt{\frac{\left(e^{\mathrm{sh}}_{n,i-1,\boldsymbol{\gamma}}(z_1,\dots,z_{i-1})\right)^2\beta_i^2}{G_i(z_1,\dots,z_i)}}\,. \tag{6.9}$$

Note that in the first dimension the upper bound on the mean-square error (6.8) becomes $G_1\,(\gamma_1+\beta_1^2)$, which attains its minimum when $\gamma_1 = 0$. Since 0 is not a sensible choice of weight our algorithm requires that γ_1 be given.

At each step of the algorithm, the process of choosing z_i to minimise G_i is the same as in the original CBC algorithm. Thus, the methods used in the fast CBC construction [18, 69, 70] can also be applied in this algorithm.

Algorithm 6.1 The double CBC algorithm for product weights

Given n, s, bounds of the form (6.3) with $B_\ell = 1$ for all ℓ, and the weight in the first dimension γ_1. Set z_1 to 1. Then for each $i = 2, \ldots, s$,

1. Choose $z_i \in \mathbb{U}_n$ to minimise $G_i(z_1, \ldots, z_{i-1}, z_i)$.
2. Set γ_i as in (6.9) and update the mean-square error bound (6.8).

6.2.2 The double CBC algorithm for POD weights

For weights of POD form (2.8), given a sequence of order dependent weight factors $(\Gamma_\ell)_{\ell=0}^s$ and a bound on the norm $M_{s,\boldsymbol{\gamma}}$ the algorithm chooses the product component of the weights γ_i in each dimension. Note that, for this case we no longer assume all $B_\ell = 1$. As before, the first step is to obtain a recursive formula for the bound on the norm of the integrand in each dimension. By splitting the sum in (6.5) according to whether or not s belongs to the set $\mathfrak{u}$, we have

$$\begin{aligned} M_{s,\boldsymbol{\gamma}} &= \sum_{\ell=0}^{s} \frac{B_\ell}{\Gamma_\ell} \sum_{\substack{\mathfrak{u}\subseteq\{1:s\}\\|\mathfrak{u}|=\ell}} \prod_{j\in\mathfrak{u}} \frac{\beta_j^2}{\gamma_j} = \sum_{\ell=0}^{s} \frac{B_\ell}{\Gamma_\ell} \left(\sum_{\substack{\mathfrak{u}\subseteq\{1:s-1\}\\|\mathfrak{u}|=\ell}} \prod_{j\in\mathfrak{u}} \frac{\beta_j^2}{\gamma_j} + \sum_{\substack{s\in\mathfrak{u}\subseteq\{1:s\}\\|\mathfrak{u}|=\ell}} \prod_{j\in\mathfrak{u}} \frac{\beta_j^2}{\gamma_j} \right) \\ &= M_{s-1,\boldsymbol{\gamma}} + \frac{\beta_s^2}{\gamma_s} \sum_{\ell=1}^{s} \underbrace{\frac{B_\ell}{\Gamma_\ell} \sum_{\substack{\mathfrak{u}\subseteq\{1:s-1\}\\|\mathfrak{u}|=\ell-1}} \prod_{j\in\mathfrak{u}} \frac{\beta_j^2}{\gamma_j}}_{H_{s-1,\ell-1}} = M_{s-1,\boldsymbol{\gamma}} + \frac{\beta_s^2}{\gamma_s} \sum_{\ell=0}^{s-1} H_{s-1,\ell}\,, \end{aligned} \tag{6.10}$$

where we have introduced the terms $H_{i,\ell}$ to simplify the notation. Applying a similar method of splitting the sum, a recursive formula for $H_{s,\ell}$ is obtained

$$\begin{aligned} H_{s,\ell} &\coloneqq \frac{B_{\ell+1}}{\Gamma_{\ell+1}} \sum_{\substack{\mathfrak{u}\subseteq\{1:s\}\\|\mathfrak{u}|=\ell}} \prod_{j\in\mathfrak{u}} \frac{\beta_j^2}{\gamma_j} = \frac{B_{\ell+1}}{\Gamma_{\ell+1}} \sum_{\substack{\mathfrak{u}\subseteq\{1:s-1\}\\|\mathfrak{u}|=\ell}} \prod_{j\in\mathfrak{u}} \frac{\beta_j^2}{\gamma_j} + \frac{B_{\ell+1}}{\Gamma_{\ell+1}} \sum_{\substack{s\in\mathfrak{u}\subseteq\{1:s\}\\|\mathfrak{u}|=\ell}} \prod_{j\in\mathfrak{u}} \frac{\beta_j^2}{\gamma_j} \\ &= H_{s-1,\ell} + \frac{\beta_s^2}{\gamma_s} \frac{B_{\ell+1}}{\Gamma_{\ell+1}} \sum_{\substack{\mathfrak{u}\subseteq\{1:s-1\}\\|\mathfrak{u}|=\ell-1}} \prod_{j\in\mathfrak{u}} \frac{\beta_j^2}{\gamma_j} \\ &= H_{s-1,\ell} + \frac{\beta_s^2}{\gamma_s} \frac{B_{\ell+1}}{B_\ell} \frac{\Gamma_\ell}{\Gamma_{\ell+1}} H_{s-1,\ell-1}\,, \end{aligned} \tag{6.11}$$

with $H_{i,0} = \frac{B_1}{\Gamma_1}$ for all $i = 1, 2, \ldots, s$ and $H_{i,\ell} = 0$ for all $\ell > i$.

It follows that for POD weights the upper bound (6.6) on the mean-square error of the QMC approximation can be written recursively as

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{\Delta}}\left[\left|\mathcal{I}_s(f)-Q^{\mathrm{sh}}_{n,s}(z_1,\dots,z_s;\cdot)f\right|^2\right] \\ &\quad\le\left(\left(e^{\mathrm{sh}}_{n,s-1,\boldsymbol{\gamma}}(z_1,\dots,z_s)\right)^2+\gamma_s G_s(z_1,\dots,z_s)\right)\left(M_{s-1,\boldsymbol{\gamma}}+\frac{\beta_s^2}{\gamma_s}\sum_{\ell=0}^{s-1}H_{s-1,\ell}\right),\end{aligned} \tag{6.12}$$

where $G_s(z_1,\dots,z_s)$ is now given by

$$\begin{aligned}&G_s(z_1,\dots,z_s)\\ &=\frac{1}{n}\sum_{k=0}^{n-1}\Bigg(\underbrace{\mathscr{B}_2\left(\left\{\frac{kz_s}{n}\right\}\right)}_{\Omega_n(z_s,k)}\sum_{\ell=1}^{s}\frac{\Gamma_\ell}{\Gamma_{\ell-1}}\underbrace{\sum_{\substack{\mathfrak{u}\subseteq\{1:s-1\}\\|\mathfrak{u}|=\ell-1}}\left(\prod_{i=1}^{\ell-1}\frac{\Gamma_i}{\Gamma_{i-1}}\right)\prod_{j\in\mathfrak{u}}\left(\gamma_j\mathscr{B}_2\left(\left\{\frac{kz_j}{n}\right\}\right)\right)}_{p_{s-1,\ell-1}(k)}\Bigg)\\ &=\frac{1}{n}\sum_{k=0}^{n-1}\Omega_n(z_s,k)\sum_{\ell=1}^{s}\frac{\Gamma_\ell}{\Gamma_{\ell-1}}p_{s-1,\ell-1}(k)\,.\end{aligned} \tag{6.13}$$

We have introduced the terms Ω_n, $p_{s-1,\ell-1}$ to simplify notation, and we have arranged to deal only with the ratios $\Gamma_\ell/\Gamma_{\ell-1}$ to improve numerical stability. Again by Lemma 6.1, the product component of the weight that minimises the bound on (6.12) is given by, with s replaced by i,

$$\gamma_i=\sqrt{\frac{\left(e^{\mathrm{sh}}_{n,i,\boldsymbol{\gamma}}(z_1,\dots,z_{i-1})\right)^2\beta_i^2\sum_{\ell=0}^{i-1}H_{i-1,\ell}}{M_{i-1,\boldsymbol{\gamma}}\,G_i(z_1,\dots,z_i)}}\,. \tag{6.14}$$

Calculating G_s for all $z_s\in\mathbb{U}_n$ by summing over all $\mathfrak{u}\subseteq\{1:s-1\}$ as in (6.13) would cost $\mathcal{O}(n^2 2^{s-1})$ operations and is infeasible for even moderate s. The cost can be reduced by storing Ω_n and constructing $p_{s-1,\ell-1}$ recursively. Letting $\boldsymbol{G}_s=[G_s(z_1,\dots,z_{s-1},z_s)]_{z_s\in\mathbb{U}_n}$, the calculation of G_s for all $z_s\in\mathbb{U}_n$ can be performed by the matrix-vector product

$$\boldsymbol{G}_s=\frac{1}{n}\boldsymbol{\Omega}_n\sum_{\ell=1}^{s}\frac{\Gamma_\ell}{\Gamma_{\ell-1}}\boldsymbol{p}_{s-1,\ell-1}\,, \tag{6.15}$$

where

$$\boldsymbol{\Omega}_n\coloneqq\left[\mathscr{B}_2\left(\left\{\frac{kz}{n}\right\}\right)\right]_{z\in\mathbb{U}_n,k=0,1,\dots,n-1}.$$

At each step the vectors $p_{s-1,\ell-1}$ can be constructed recursively as follows

$$\boldsymbol{p}_{s,\ell} \;=\; \boldsymbol{p}_{s-1,\ell} + \frac{\Gamma_\ell}{\Gamma_{\ell-1}} \gamma_s \boldsymbol{\Omega}_n(z_s,:) \,.*\, \boldsymbol{p}_{s-1,\ell-1}\,, \tag{6.16}$$

where $\boldsymbol{\Omega}_n(z_s,:)$ is the row corresponding the new component of the generating vector z_s, and $.*$ denotes component-wise multiplication, and with $\boldsymbol{p}_{s,0} = \mathbf{1}$, $\boldsymbol{p}_{s,\ell} = \mathbf{0}$ for all $\ell > s$. Note that (6.16) is obtained by splitting the sum according to whether or not $s \in \mathfrak{u}$, as in (6.10) and (6.11). Since the cost of updating $\boldsymbol{p}_{s,\ell}$ is $\mathcal{O}(s\,n)$ the total cost of calculating G_s in each dimension has been reduced to $\mathcal{O}(n^2 + sn)$ operations. Additionally, using the concepts from the Fast CBC algorithm [69, 70] this product can be performed more efficiently using the FFT, which would further reduce the cost to $\mathcal{O}(n \log n + sn)$. The total cost of the algorithm is now $\mathcal{O}(s\,n \log n + s^2 n)$ operations.

Algorithm 6.2 The double CBC algorithm for POD weights

Given n, s, bounds of the form (6.3), order dependent weight factors $\{\Gamma_\ell\}_{\ell=0}^{s}$, and the weight in the first dimension γ_1. Set $z_1 = 1$, $H_{0,0} = B_1/\Gamma_1$, $\boldsymbol{p}_{0,0} = \mathbf{1}$. Then for each $i = 2, \ldots, s$,

1. For $\ell = 0, \ldots, i-1$, update $H_{i-1,\ell}$ using (6.11) and $\boldsymbol{p}_{i-1,\ell}$ using (6.16).
2. Calculate $\boldsymbol{G}_i$ using (6.15) and FFT.
3. Choose $z_i \in \mathbb{U}_n$ to minimise $G_i(z_1, \ldots, z_{i-1}, z_i)$.
4. Set γ_i as in (6.14) and update the mean-square error bound (6.12).

We have so far neglected the question of how to choose the order dependent weight factors Γ_ℓ. Three possible choices are:

- Γ_ℓ given *a priori*, such as by the common choice $\Gamma_\ell = \ell!$.
- $\Gamma_\ell = \Gamma_\ell(\eta)$, that is, the order dependent weight factors of the weights $\gamma_{\mathfrak{u}}(\eta)$ from the formula (6.17) below. In this case we are still left with the predicament of how to choose η, a choice which this algorithm aimed to circumvent.
- $\Gamma_\ell = B_\ell$. Here the recursion for the bound on the norm (6.10) is the same as the product weight case (6.7), that is, the terms $H_{i,\ell}$ are no longer required. Further, since there is some inherent connection between the form of the bound on the norm and the weights this choice seems more natural than the other two.

6.3 The iterated CBC algorithm

Combining (6.6) with the upper bound on the shift-averaged worst-case error (2.16) we have that the mean-square error of a CBC constructed lattice rule approximation

is bounded above by

$$\mathbb{E}_{\boldsymbol{\Delta}}\left[\left|\mathcal{I}_s(f)-Q^{\mathrm{sh}}_{n,s}(\boldsymbol{z};\cdot)f\right|^2\right] \le \left(\frac{1}{\varphi(n)}\sum_{\emptyset\ne\mathfrak{u}\subseteq\{1:s\}}\gamma_{\mathfrak{u}}^{\eta}\left(\frac{2\zeta(2\eta)}{(2\pi^2)^{\eta}}\right)^{|\mathfrak{u}|}\right)^{\frac{1}{\eta}} \times\left(\sum_{\mathfrak{u}\subseteq\{1:s\}}\frac{1}{\gamma_{\mathfrak{u}}}B_{|\mathfrak{u}|}\prod_{j\in\mathfrak{u}}\beta_j^2\right) \quad \text{for all } \eta\in\left(\tfrac{1}{2},1\right].$$

From [60, Lemma 6.2], the weights that minimise this error bound for each $\eta \in \left(\frac{1}{2},1\right]$ are of POD form

$$\gamma_{\mathfrak{u}}(\eta) \;=\; \Gamma_{|\mathfrak{u}|}(\eta)\prod_{j\in\mathfrak{u}}\gamma_j(\eta) \;=\; \left(B_{|\mathfrak{u}|}\prod_{j\in\mathfrak{u}}\frac{(2\pi^2)^{\eta}\beta_j^2}{2\zeta(2\eta)}\right)^{\frac{1}{1+\eta}}. \tag{6.17}$$

For each $\eta \in \left(\frac{1}{2},1\right]$ the corresponding weights $\boldsymbol{\gamma}(\eta)=\{\gamma_{\mathfrak{u}}(\eta)\}_{\mathfrak{u}\subseteq\{1:s\}}$ can be taken as input into the CBC algorithm to construct a lattice rule generating vector that, through the weights, depends on η. We will label such a generating vector as $\boldsymbol{z}(\eta)$. Now that we know the weights $\boldsymbol{\gamma}(\eta)$ and generating vector $\boldsymbol{z}(\eta)$ explicitly, from (6.6), the mean-square error of the resulting QMC approximation is bounded by

$$\mathbb{E}_{\boldsymbol{\Delta}}\left[\left|\mathcal{I}_s(f)-Q^{\mathrm{sh}}_{s,n}(\boldsymbol{z};\cdot)f\right|^2\right] \le \left(e^{\mathrm{sh}}_{n,s,\boldsymbol{\gamma}(\eta)}(\boldsymbol{z}(\eta))\right)^2 M_{s,\boldsymbol{\gamma}(\eta)} =: E_{n,s,\boldsymbol{z}(\eta)}(\eta). \tag{6.18}$$

The goal of the **iterated CBC (ICBC) algorithm** is to carry out iterations of the original CBC algorithm to choose an η that minimises the right hand side of (6.18). However, since each component z_j is obtained by a minimisation over a set of integers and because this minimisation depends on the weights (and hence η), when treated as a function of η the shift-averaged worst-case error is discontinuous. Hence, we cannot guarantee that a minimum exists and as such our algorithm heuristically searches for a "good" value of η. As mentioned in the introduction, the choice of η is non-trivial since one needs to balance the size of the constant and the theoretical convergence rate.

Suppose that in the upper bound (6.18) the generating vector $\boldsymbol{z}$ remains fixed, then the upper bound, $E_{n,s,\boldsymbol{z}}(\eta)$, is a continuous function of the single variable η and can be minimised numerically.

The idea behind this algorithm is at each step of the iteration to use $E_{n,s,\boldsymbol{z}^{(k)}}(\eta)$ as an approximation to the right hand side of the upper bound (6.18). In this way the next iterate η_{k+1} is taken to be the minimiser of $E_{n,s,\boldsymbol{z}^{(k)}}(\eta)$, which can be found numerically using a quasi-Newton method.

Algorithm 6.3 The iterated CBC algorithm

Given n, s, bounds of the form (6.3), an initial $\eta_0 \in \left(\frac{1}{2}, 1\right]$, a tolerance τ and a maximum number of iterations $k_{\max}$. For $k = 0, 1, 2, \ldots, k_{\max}$:

1. Generate the weights $\gamma_{\mathfrak{u}}(\eta_k)$ using (6.17).
2. Construct the generating vector $\boldsymbol{z}^{(k)}$ from the original CBC algorithm with weights $\gamma_{\mathfrak{u}}(\eta_k)$.
3. If $\left|\dfrac{\mathrm{d}}{\mathrm{d}\eta} E_{n,s,\boldsymbol{z}^{(k)}}(\eta_k)\right| < \tau$ then end the algorithm.
4. Otherwise, choose η_{k+1} to be the minimiser of $E_{n,s,\boldsymbol{z}^{(k)}}(\eta)$, found numerically using a quasi-Newton algorithm.

Remark 6.2. For the quasi-Newton algorithm in Step 4 we require the derivative of $E_{n,s,\boldsymbol{z}}$, for fixed $\boldsymbol{z}$, with respect to η

$$\frac{\mathrm{d}E_{n,s,\boldsymbol{z}}}{\mathrm{d}\eta} = \left(\sum_{\emptyset\neq\mathfrak{u}\subseteq\{1:s\}} \gamma_{\mathfrak{u}}'(\eta)\left(\frac{1}{n}\sum_{k=0}^{n-1}\prod_{j\in\mathfrak{u}}\mathscr{B}_2\left(\left\{\frac{kz_j}{n}\right\}\right)\right)\right)\left(\sum_{\mathfrak{u}\subseteq\{1:s\}}\frac{B_{|\mathfrak{u}|}\prod_{j\in\mathfrak{u}}\beta_j^2}{\gamma_{\mathfrak{u}}(\eta)}\right)$$
$$-\left(\sum_{\emptyset\neq\mathfrak{u}\subseteq\{1:s\}} \gamma_{\mathfrak{u}}(\eta)\left(\frac{1}{n}\sum_{k=0}^{n-1}\prod_{j\in\mathfrak{u}}\mathscr{B}_2\left(\left\{\frac{kz_j}{n}\right\}\right)\right)\right)\left(\sum_{\mathfrak{u}\subseteq\{1:s\}}\frac{\gamma_{\mathfrak{u}}'(\eta)}{\gamma_{\mathfrak{u}}^2(\eta)}B_{|\mathfrak{u}|}\prod_{j\in\mathfrak{u}}\beta_j^2\right),$$

where the derivative of each weight with respect η is

$$\gamma_{\mathfrak{u}}'(\eta) \;=\; \gamma_{\mathfrak{u}}(\eta)\left(\frac{-\log\left(B_{|\mathfrak{u}|}\prod_{j\in\mathfrak{u}}\frac{\left(2\pi^2\right)^{\eta}\beta_j^2}{2\zeta(2\eta)}\right)}{(1+\eta)^2} + \frac{|\mathfrak{u}|\left(\log(2\pi^2) - \frac{2\zeta'(2\eta)}{\zeta(2\eta)}\right)}{(1+\eta)}\right).$$

6.4 Numerical results

For the numerical results we look at how each new method performs for different types of bounds (6.3), that is, for different sequences $\boldsymbol{\beta} \coloneqq (\beta_i)_{i=1}^s$ and $\boldsymbol{B} \coloneqq (B_\ell)_{\ell=1}^s$. As a figure of merit we will use the upper bound on the root-mean-square (RMS) error (cf. (6.6)), which we denote by

$$E_{n,s,\boldsymbol{\beta},\boldsymbol{B}}(\boldsymbol{\gamma},\boldsymbol{z}) \;\coloneqq\; e^{\mathrm{sh}}_{n,s,\boldsymbol{\gamma}}(\boldsymbol{z})\sqrt{M_{s,\boldsymbol{\gamma}}}\,.$$

Here we have specifically used this notation to indicate the dependence on the sequences $\boldsymbol{\beta}$, $\boldsymbol{B}$ but also to emphasise that this upper bound is primarily a function of $\boldsymbol{\gamma}$ and $\boldsymbol{z}$, the outputs of our algorithms. Note also that in Tables 6.2–6.8 the

results given for our algorithms are *guaranteed* error bounds (in the RMS sense) for integrands that satisfy the appropriate bounds.

In the examples let the maximum dimension be $s = 100$ and the number of points n be prime and ranging up to 32,003. Here we choose n to be prime because it makes the "fast" aspects of the implementation simpler, but note that n prime is not a requirement of either algorithm. Also, we use the notation "e" for the base-10 exponent.

6.4.1 Product weights

As a start, let $B_\ell = 1$ for all $\ell = 1, \ldots, s$, and consider the cases $\beta_i = i^{-2}$, $\beta_i = 0.5^i$ and $\beta_i = 0.8^i$. With $B_\ell = 1$ it is natural to restrict attention to product weights. The results for this case are given on the next page in Tables 6.2, 6.3, 6.4, respectively. These tables compare results for $E_{n,s,\boldsymbol{\beta},\boldsymbol{B}}$ from the DCBC, ICBC algorithms with the original CBC algorithm using common choices of product weights. The choices of common weights are $\gamma_i = i^{-1.1}$, $\gamma_i = i^{-2}$ and $\gamma_i(\eta)$ as in (6.17) with $\eta = 0.6, 1$. The row labelled "rate" gives the exponent (x) for a least-squares fit of the result $E_{n,s,\boldsymbol{\beta},\boldsymbol{B}}$ to a power law (n^{-x}). The bold font indicates entries with the lowest worst-case error bound.

Comparing results for the two new algorithms, note that for $\beta_i = i^{-2}$ (Table 6.2) the results of the DCBC algorithm are better, while for $\beta_i = 0.5^i$, 0.8^i (Tables 6.3, 6.4) the ICBC algorithm produces better bounds, and so it is not the case that one algorithm is always better than the other. However, in all cases the ICBC algorithm performs as well as or better than the original CBC algorithm with common choices of weights.

Table 6.1 gives the final value of η, denoted η^*, resulting from the ICBC algorithm for our three choices of $\boldsymbol{\beta}$, along with the resulting RMS error bound. Notice that, as expected the value of η^* found by the algorithm appears to approach 0.5 as n increases, albeit very slowly.

n	$\beta_i = i^{-2}$	$\beta_i = 0.5^i$	$\beta_i = 0.8^i$
251	0.672	0.616	0.756
499	0.668	0.615	0.744
997	0.661	0.610	0.735
1999	0.657	0.607	0.725
4001	0.652	0.604	0.715
7993	0.645	0.601	0.711
16001	0.642	0.597	0.700
32003	0.637	0.594	0.696

Table 6.1: Value of η^* from ICBC for product weights with $\beta_i = i^{-2}$, 0.5^i and 0.8^i.

	Variants		CBC with common weights			
n	**DCBC**	ICBC	$\gamma_i = i^{-1.1}$	$\gamma_i = i^{-2}$	$\gamma_i(\eta = 0.6)$	$\gamma_i(\eta = 1)$
251	**6.8e-3**	7.0e-3	3.5e-2	7.5e-3	8.2e-3	1.3e-2
499	**3.5e-3**	3.6e-3	2.1e-2	4.0e-3	4.2e-3	7.6e-3
997	**1.8e-3**	1.9e-3	1.3e-2	2.2e-3	2.2e-3	4.3e-3
1999	**9.7e-4**	1.0e-3	7.8e-3	1.2e-3	1.1e-3	2.4e-3
4001	**5.1e-4**	5.2e-4	4.8e-3	6.3e-4	5.8e-4	1.4e-3
7993	**2.7e-4**	2.7e-4	2.9e-3	3.4e-4	2.9e-4	7.8e-4
16001	**1.4e-4**	1.4e-4	1.8e-3	1.9e-4	1.5e-4	4.4e-4
32003	**7.4e-5**	7.5e-5	1.1e-3	1.0e-4	7.9e-5	2.5e-4
rate	**0.93**	0.93	0.71	0.88	0.95	0.82

Table 6.2: Results for the root-mean-square error bound $E_{n,s,\boldsymbol{\beta},\boldsymbol{B}}$ for $\beta_i = i^{-2}$: DCBC, ICBC and CBC results for common choices of weights.

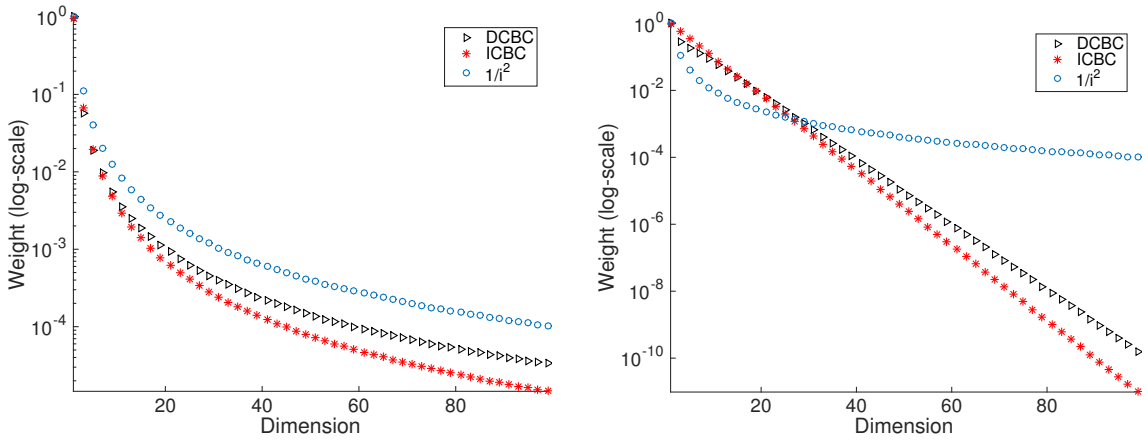
	Variants		CBC with common weights			
n	DCBC	**ICBC**	$\gamma_i = i^{-1.1}$	$\gamma_i = i^{-2}$	$\gamma_i(\eta = 0.6)$	$\gamma_i(\eta = 1)$
251	4.1e-3	**3.3e-3**	2.8e-2	5.5e-3	3.3e-3	6.7e-3
499	2.1e-3	**1.7e-3**	1.7e-2	2.9e-3	1.7e-3	3.6e-3
997	1.1e-3	**8.6e-4**	1.0e-2	1.6e-3	8.6e-4	2.0e-3
1999	5.6e-4	**4.4e-4**	6.2e-3	8.6e-4	4.4e-4	1.1e-3
4001	2.9e-4	**2.2e-4**	3.8e-3	4.6e-4	2.2e-4	5.8e-4
7993	1.5e-4	**1.1e-4**	2.3e-3	2.5e-4	1.1e-4	3.1e-4
16001	7.6e-5	**5.9e-5**	1.4e-3	1.4e-4	5.9e-5	1.7e-4
32003	3.9e-5	**3.0e-5**	8.7e-4	7.5e-5	3.0e-5	9.3e-5
rate	0.96	**0.96**	0.71	0.88	0.97	0.88

Table 6.3: Results for the root-mean-square error bound $E_{n,s,\boldsymbol{\beta},\boldsymbol{B}}$ for $\beta_i = 0.5^i$: DCBC, ICBC and CBC results for common choices of weights.

	Variants		CBC with common weights			
n	DCBC	**ICBC**	$\gamma_i = i^{-1.1}$	$\gamma_i = i^{-2}$	$\gamma_i(\eta = 0.6)$	$\gamma_i(\eta = 1)$
251	9.9e-2	**8.3e-2**	2.0e-1	2.8	1.6e-1	1.2e-1
499	5.7e-2	**5.0e-2**	1.2e-1	1.5	8.9e-2	7.2e-2
997	3.5e-2	**2.9e-2**	7.5e-2	8.2e-1	5.1e-2	4.5e-2
1999	2.1e-2	**1.7e-2**	4.6e-2	4.4e-1	2.8e-2	2.8e-2
4001	1.2e-2	**1.0e-2**	2.8e-2	2.4e-1	1.6e-2	1.8e-2
7993	7.3e-3	**5.9e-3**	1.7e-2	1.3e-1	9.1e-3	1.1e-2
16001	4.3e-3	**3.5e-3**	1.0e-2	7.1e-2	5.0e-3	6.7e-3
32003	2.5e-3	**2.0e-3**	6.4e-3	3.9e-2	2.9e-3	4.2e-3
rate	0.75	**0.75**	0.71	0.88	0.82	0.69

Table 6.4: Results for the root-mean-square error bound $E_{n,s,\boldsymbol{\beta},\boldsymbol{B}}$ for $\beta_i = 0.8^i$: DCBC, ICBC and CBC results for common choices of weights.

Figures 6.1a and 6.1b compare the weight γ_i in each dimension for the DCBC, ICBC algorithms, with $\gamma_i = i^{-2}$ as a reference, for $\beta_i = i^{-2}$ and $\beta_i = 0.5^i$, respectively. Here $n = 1999$.



(a) $\beta_i = i^{-2}$ (b) $\beta_i = 0.5^i$

Figure 6.1: Weight in each dimension found from DCBC and ICBC, with $\gamma_i = i^{-2}$ for comparison ($n = 1999$).

Figure 6.2 compares the RMS error bound in each dimension for the DCBC and ICBC algorithms, for $\beta_i = 0.8^i$ with $n = 1999$. Notice that due to the greedy nature of the DCBC algorithm, it performs better in the earlier dimensions, however, the ICBC algorithm produces weights with a better bound overall.

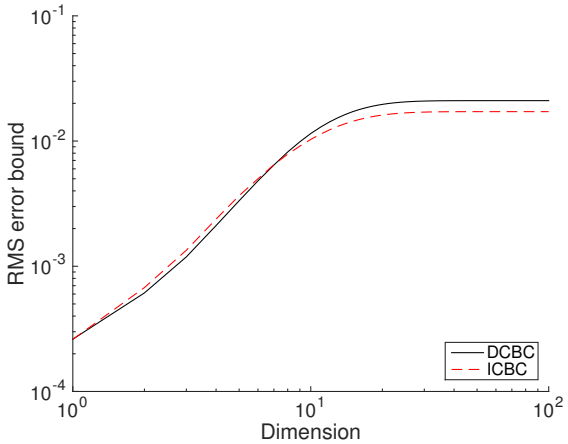


Figure 6.2: $E_{n,s,\boldsymbol{\beta},\boldsymbol{B}}$ in each dimension for DCBC and ICBC with $\beta_i = 0.8^i$ (loglog scale).

6.4.2 POD weights

Now we look at the performance of each algorithm for combinations of $B_\ell = \ell$ and $B_\ell = \ell!$ with $\beta_i = i^{-2}$ and $\beta_i = 0.5^i$, the results are given in Tables 6.5–6.8. Each table presents the RMS error bound $E_{n,s,\boldsymbol{\beta},\boldsymbol{B}}$ obtained from the DCBC algorithm for different choices of order dependent weights, and those obtained from the ICBC algorithm along with the output η^*. With regards to the strategy for choosing Γ_ℓ in the DCBC algorithm for POD weights, in three out of the four cases the best choice was to let $\Gamma_\ell = B_\ell$ (see Tables 6.5–6.7). However, in all cases the results are similar and indicate that the choice of Γ_ℓ does not greatly effect the final bound $E_{n,s,\boldsymbol{\beta},\boldsymbol{B}}$.

	DCBC		**ICBC**	
n	$E_{n,s,\boldsymbol{\beta},\boldsymbol{B}}$ $(\Gamma_\ell = B_\ell)$	$E_{n,s,\boldsymbol{\beta},\boldsymbol{B}}$ $(\Gamma_\ell = \ell!)$	$E_{n,s,\boldsymbol{\beta},\boldsymbol{B}}$	η^*
251	8.6e-3	8.5e-3	**8.7e-3**	0.680
499	4.6e-3	4.5e-3	**4.6e-3**	0.673
997	2.5e-3	2.5e-3	**2.5e-3**	0.666
1999	1.3e-3	1.3e-3	**1.3e-3**	0.659
4001	6.9e-4	7.0e-4	**6.8e-4**	0.655
7993	3.7e-4	3.7e-4	**3.6e-4**	0.650
16001	1.9e-4	2.0e-4	**1.9e-4**	0.645
32003	1.0e-4	1.1e-4	**1.0e-4**	0.640
rate	0.91	0.90	**0.92**	

Table 6.5: POD weight results with $\beta_i = i^{-2}$ and $B_\ell = \ell$: $E_{n,s,\boldsymbol{\beta},\boldsymbol{B}}$ from DCBC for different choices of Γ_ℓ, $E_{n,s,\boldsymbol{\beta},\boldsymbol{B}}$ from ICBC and η^* from ICBC.

	DCBC		ICBC	
n	$E_{n,s,\boldsymbol{\beta},\boldsymbol{B}}$ $(\Gamma_\ell = B_\ell)$	$E_{n,s,\boldsymbol{\beta},\boldsymbol{B}}$ $(\Gamma_\ell = \ell)$	$E_{n,s,\boldsymbol{\beta},\boldsymbol{B}}$	η^*
251	**9.2e-3**	1.1e-2	9.7e-3	0.692
499	**5.0e-3**	5.8e-3	5.1e-3	0.685
997	**2.7e-3**	3.2e-3	2.8e-3	0.679
1999	**1.5e-3**	1.7e-3	1.5e-3	0.673
4001	**7.9e-4**	9.6e-4	8.0e-4	0.667
7993	**4.2e-4**	5.2e-4	4.3e-4	0.661
16001	**2.3e-4**	2.8e-4	2.3e-4	0.656
32003	**1.2e-4**	1.6e-4	1.3e-4	0.651
rate	**0.89**	0.87	0.89	

Table 6.6: POD weight results with $\beta_i = i^{-2}$ and $B_\ell = \ell!$: $E_{n,s,\boldsymbol{\beta},\boldsymbol{B}}$ from DCBC for different choices of Γ_ℓ, $E_{n,s,\boldsymbol{\beta},\boldsymbol{B}}$ from ICBC and η^* from ICBC.

	DCBC		**ICBC**	
n	$E_{n,s,\boldsymbol{\beta},\boldsymbol{B}}$ $(\Gamma_\ell = B_\ell)$	$E_{n,s,\boldsymbol{\beta},\boldsymbol{B}}$ $(\Gamma_\ell = \ell!)$	$E_{n,s,\boldsymbol{\beta},\boldsymbol{B}}$	η^*
251	4.9e-3	5.0e-3	**3.8e-3**	0.619
499	2.5e-3	2.6e-3	**2.0e-3**	0.617
997	1.3e-3	1.4e-3	**1.0e-3**	0.612
1999	6.9e-4	7.2e-4	**5.3e-4**	0.608
4001	3.6e-4	3.8e-4	**2.7e-4**	0.605
7993	1.9e-4	2.0e-4	**1.4e-4**	0.602
16001	9.8e-5	1.0e-4	**7.2e-5**	0.597
32003	5.1e-5	5.3e-5	**3.7e-5**	0.595
rate	0.94	0.93	**0.95**	

Table 6.7: POD weight results with $\beta_i = 0.5^i$ and $B_\ell = \ell$: $E_{n,s,\boldsymbol{\beta},\boldsymbol{B}}$ from DCBC for different choices of Γ_ℓ, $E_{n,s,\boldsymbol{\beta},\boldsymbol{B}}$ from ICBC and η^* from ICBC.

	DCBC		**ICBC**	
n	$E_{n,s,\boldsymbol{\beta},\boldsymbol{B}}$ $(\Gamma_\ell = B_\ell)$	$E_{n,s,\boldsymbol{\beta},\boldsymbol{B}}$ $(\Gamma_\ell = \ell)$	$E_{n,s,\boldsymbol{\beta},\boldsymbol{B}}$	η^*
251	5.1e-3	5.1e-3	**4.0e-3**	0.625
499	2.6e-3	2.6e-3	**2.1e-3**	0.622
997	1.4e-3	1.4e-3	**1.1e-3**	0.618
1999	7.3e-4	7.3e-4	**5.6e-4**	0.614
4001	3.9e-4	3.8e-4	**2.9e-4**	0.608
7993	2.0e-4	2.0e-4	**1.5e-4**	0.604
16001	1.1e-4	1.0e-4	**7.9e-5**	0.602
32003	5.6e-5	5.5e-5	**4.1e-5**	0.599
rate	0.93	0.93	**0.95**	

Table 6.8: POD weight results with $\beta_i = 0.5^i$ and $B_\ell = \ell!$: $E_{n,s,\boldsymbol{\beta},\boldsymbol{B}}$ from DCBC for different choices of Γ_ℓ, $E_{n,s,\boldsymbol{\beta},\boldsymbol{B}}$ from ICBC and η^* from ICBC.

6.5 Conclusion

In this chapter we introduced two new CBC algorithms, the double CBC (DCBC) algorithm and the iterated CBC (ICBC) algorithm, which only require parameters specified by the problem to determine the point set for QMC integration, and provide guaranteed error bounds by also choosing "good" weight parameters. The numerical results show different examples where each algorithm performs better than the other. Both algorithms generally outperform the original CBC algorithm with common choices of weights. In all cases the entries $E_{n,s,\boldsymbol{\beta},\boldsymbol{B}}$ provide guaranteed upper bounds on the root-mean-square error for the randomly shifted integration rules, under the indicated assumptions on the bound parameters B_ℓ and β_i.

Original research and my contribution

The work in this chapter is based on joint work with my two supervisors Frances Kuo and Ian Sloan. The corresponding paper appeared in *Mathematics and Computers*

in Simulation in early 2018 (see [32]), and has been available online at `https://doi.org/10.1016/j.matcom.2016.06.005` since 2016.

The CBC algorithm and fast modifications have existed for some time, see, e.g., [52, 80, 81] and [69, 70], respectively. However, the idea to modify the CBC to also choose the weights and the two algorithms, the Double CBC and the Iterated CBC, are new.

For the first algorithm, Frances and Ian had the idea that the existing CBC construction could be modified to also choose the weight parameters in a greedy manner. I was tasked with exploring their idea, implementing the algorithm and performing numerical tests. During the tests we discovered that our new algorithm did not produce weights that were guaranteed to result in smaller error bounds than previous methods. From this I had the idea to choose the "best η" and then devised the Iterated CBC algorithm. Also, I implemented both algorithms and performed the numerical results.

Chapter 7

Concluding remarks

Although each of the previous chapters explored different topics in numerical integration, they were all concerned in some way or another with high-dimensional integrals of the form (2.1), or (2.2) when the dimension is infinite, in combination with how algorithms can utilise information about the derivatives of the integrand, both theoretically and practically. The main results of this thesis, which are also original, have been as follows.

In Chapter 3, we successfully applied Quasi-Monte Carlo (QMC) methods to a class of elliptic eigenvalue problems with coefficients that depend on infinitely-many stochastic parameters. The full algorithm approximated the expected value of the smallest eigenvalue (or a linear functional of the corresponding eigenfunction) by truncating the stochastic dimension, discretising the spatial domain using finite elements and then approximating the expected value (a high-dimensional integral) by a QMC lattice rule. The bulk of the theoretical analysis in this chapter was involved with proving explicit bounds on the mixed derivatives of the minimal eigenvalue and its eigenfunction. These bounds allowed us to perform a full error analysis, where we obtained results that in almost all cases give the same convergence rate as the results for the corresponding PDE source problem. Also, at the end of the chapter we presented numerical results on the QMC convergence of our approximations, which behave exactly as predicted by our theory.

Chapter 4 explored different ways to efficiently implement the Multivariate Decomposition Method (MDM) using Smolyak and QMC quadrature rules. Here we developed fast reformulations of the MDM by exploiting the structure in both the anchored decomposition and the quadrature rules used. We also provided explicit pseudocodes for constructing the active set and for running each of our fast, reformulated MDM approximations. Finally, we provided numerical results for applying the MDM to an example integrand, including estimates of the error, which behave as expected, and timing results comparing naive implementations with our fast reformulations, which clearly demonstrate the improved efficiency of our new reformulations. Although the main results of this chapter were the fast reformulations,

until this work the MDM was purely a theoretical algorithm, hence, even the first naive implementation is an achievement in itself.

Then in Chapter 5 we focussed on the truncation component of the MDM, introducing two new algorithms for constructing active sets in the specific case of inputs in product form. The first new algorithm produced *optimal* active sets—in the sense that for a given error request they have the smallest cardinality of any active set. The second algorithm was a simplification that constructed *quasi-optimal* active sets that were possibly larger. However, in all of the numerical results presented the difference between the size of the optimal and quasi-optimal active sets was minimal. Also, both the optimal and quasi-optimal active sets were smaller than the sets yielded by the previous construction. In this chapter the analysis was performed in the weighted setting, and the class of functions considered had mixed first derivatives bounded in L^p spaces. However, the presentation of both algorithms took generic inputs, and the general strategy for constructing optimal active sets can easily be adapted to the setting without weights, in which case the inputs would again involve bounds on the derivatives.

Finally, in Chapter 6 we introduced two algorithms for constructing lattice rule generating vectors to be used for numerical integration, which also automatically chose good function space weight parameters. There also the inputs to the algorithms were bounds on the mixed first derivatives. Although there were different situations where the two algorithms outperformed each other, in all of our numerical experiments the new algorithms yielded lattice rules with smaller worst-case errors than the original CBC algorithm with common choices of weights.

7.1 Future work & possible extensions

As discussed in Section 3.1 the study of QMC methods for stochastic eigenvalue problems can be thought of as an extension of the work on applying QMC to a PDE source problem with stochastic coefficients, see e.g., [60]. The stochastic PDE source problem has in recent years inspired a large amount of research, in particular on developing different QMC-based algorithms or on studying QMC for different classes of PDE problems, see [57] for a survey. Thus, there is a clear path forward with respect to future work on stochastic eigenvalue problems. Actually, there are several clear paths forward, including the following.

Possibly the easiest extension would be to look at using higher-order QMC rules to approximate the expected value. This should be relatively straightforward, because in Lemma 3.10 we also proved bounds on the higher-order derivatives that are required for the error analysis of high-order rules.

Another possibility would be to consider *lognormal* coefficients, as in e.g., [37], where each stochastic parameter is independently, normally distributed. However, handling such coefficients is not so straightforward, because although the analysis would follow the same basic structure, most of the results on the regularity of the eigenvalues (also eigenfunctions) would need to be reproved, and in this case dependence on the stochastic parameters is much harder to control because now each parameter belongs to $\mathbb{R}$. In particular, one would have to reprove that simple eigenvalues are analytic in $\boldsymbol{y}$ and proving that the spectral gap is bounded independently of $\boldsymbol{y}$ requires a new strategy.

Following from the success of multi-level algorithms for such PDE problems, see e.g., [62], one could apply a multi-level QMC algorithm to approximate the expected value. The difficulty here is that the error analysis for the eigenfunction requires bounds on the spatial and stochastic derivatives simultaneously.

Other extensions of the eigenvalue problem include studying Bayesian inversion, see e.g., [16], parametrising the coefficients using locally supported basis functions as in [26] or instead of parametrising the coefficients using a Karhunen-Loève-type series expansion one could approximate the random field directly using circulant embedding techniques such as in [38].

Of course, these extensions are not mutually exclusive, and often it is interesting to combine different avenues of study. For example, one could explore applying multi-level QMC to an eigenvalue problem with log-normal coefficients as in [59].

Also, although the work on the source problem provides a template for future research, as we saw in Chapter 3, for eigenvalue problems the corresponding analysis presents its own nuances and can be more difficult due to their inherent nonlinear nature. Clearly the application of QMC methods to stochastic eigenvalue problems is an interesting new topic, and is one that offers a multitude of directions for future study.

A possible extension of our work on the MDM is to study applying the algorithm to different types of integrands, especially given that we now have several efficient implementations. For example, it would be interesting to apply the MDM to an integrand that arose from a stochastic PDE problem such as those discussed above. It is already known that such integrands satisfy the required bounds on the derivatives and so fit into the setting from Section 4.6. However, one would still have to perform some analysis to combine the MDM with a method of approximating the PDE, such as finite elements.

7.2 A final word

Integrals of high and infinite-dimensional problems are, most of the time, very difficult to approximate numerically, because in general the problems themselves suffer from the curse of dimensionality. But motivated by several applications there is a real need to develop numerical integration algorithms that can efficiently handle such problems, even as the dimension tends to infinity. As we have seen throughout this thesis, the key to successful numerical integration in high dimensions is to exploit structure, such as smoothness or sparsity, of the integrand.

As with all areas of numerical analysis, when approximating high-dimensional integrals nothing comes free.

References

[1] R. Andreev and Ch. Schwab. Sparse tensor approximation of parametric eigenvalue problems. In I. G. Graham et al., editor, *Numerical Analysis of Multiscale Problems, Lecture Notes in Computational Science and Engineering*, pages 203–241. Springer–Verlag, Berlin–Heidelberg, Germany, 2012.

[2] B. Andrews and J. Clutterbuck. Proof of the fundamental gap conjecture. *J. Amer. Math. Soc.*, 24:899–916, 2011.

[3] M. N. Avramova and K. N. Ivanov. Verification, validation and uncertainty quantification in multi-physics modeling for nuclear reactor design and safety analysis. *Prog. Nucl. Energy*, 52(7):601–614, 2010.

[4] D. A. F. Ayres, M. D. Eaton, A. W. Hagues, and M. M. R. Williams. Uncertainty quantification in neutron transport with generalized polynomial chaos using the method of characteristics. *Ann. Nucl. Energy*, 45:14–28, 2012.

[5] I. Babuška and J. Osborn. Estimates for the errors in eigenvalue and eigenvector approximation by Galerkin methods, with particular attention to the case of multiple eigenvalues. *SIAM J. Numer. Anal.*, 24:1249–1276, 1987.

[6] I. Babuška and J. Osborn. Finite element-Galerkin approximation of eigenvalues and eigenvectors of selfadjoint problems. *Math. Comp.*, 52:275–297, 1989.

[7] I. Babuška and J. Osborn. Eigenvalue problems. In P. G. Ciarlet and J .L. Lions, editors, *Handbook of Numerical Analysis, Volume 2: Finite Element Methods (Part 1)*, pages 641–787. Elsevier Science, Amsterdam, The Netherlands, 1991.

[8] D. Boffi. Finite element approximation of eigenvalue problems. *Acta Numerica*, 19:1–120, 2010.

[9] H. Brezis. *Functional Analysis, Sobolev Spaces and Partial Differential Equations*. Universitext, Springer, New York, NY, USA, 2011.

[10] H.-J Bungartz and M. Griebel. Sparse grids. *Acta Numerica*, 13:147–269, 2004.

[11] R. E. Caflisch, W. Morokoff, and A. B. Owen. Valuation of mortgage backed securities using Brownian bridges to reduce the effective dimension. *J. Comp. Finance*, 1:27–46, 1997.

[12] R. Cools, F. Y. Kuo, and D. Nuyens. Constructing embedded lattice rules for multivariate integration. *SIAM J. Sci. Comp.*, 28:2162–2188, 2006.

[13] J. Dick. On the convergence rate of the component-by-component construction of good lattice rules. *J. Complexity*, 20:493–522, 2004.

[14] J. Dick. Explicit constructions of quasi-Monte Carlo rules for the numerical integration of high-dimensional periodic functions. *SIAM J. Numer. Anal.*, 45:2141–2176, 2007.

[15] J. Dick. Walsh spaces containing smooth functions and quasi-Monte Carlo rules of arbitrary high order. *SIAM J. Numer. Anal.*, 46:1519–1553, 2008.

[16] J. Dick, R. N. Gantner, Q. T. Le Gia, and Ch. Schwab. Higher order quasi-Monte Carlo integration for Bayesian estimation. *arXiv:* `https://arxiv.org/pdf/1602.07363.pdf`, version 2016.

[17] J. Dick, F. Y. Kuo, Q. T. Le Gia, D. Nuyens, and Ch. Schwab. Higher order QMC Petrov–Galerkin discretisation for affine parametric operator equations with random field inputs. *SIAM J. Numer. Anal.*, 52:2676–2702, 2014.

[18] J. Dick, F. Y. Kuo, and I. H. Sloan. High-dimensional integration: The quasi-Monte Carlo way. *Acta Numerica*, 22:133–288, 2013.

[19] J. Dick and F. Pillichshammer. Strong tractability of multivariate integration of arbitrary high order using digitally shifted polynomial lattice rules. *J. Complexity*, 23:436–453, 2007.

[20] J. Dick and F. Pillichshammer. *Digital Nets and Sequences: Discrepancy Theory and Quasi-Monte Carlo Integration.* Cambridge University Press, New York, NY, USA, 2010.

[21] J. Dick, I. H Sloan, X. Wang, and H. Woźniakowski. Good lattice rules in weighted Korobov spaces with general weights. *Numer. Math.*, 103:63–97, 2006.

[22] D. C. Dobson. An efficient method for band structure calculations in 2D photonic crystals. *J. Comput. Phys.*, 149(2):363–376, 1999.

[23] J. J. Duderstadt and L. J. Hamilton. *Nuclear Reactor Analysis.* John Wiley & Sons, Inc., New York London Sydney Toronto, 1976.

[24] V. Ehrlacher. *Some Mathematical Models in Quantum Chemistry and Uncertainty Quantification.* PhD thesis, CERMICS, Université Paris-Est, 2012.

[25] I. Fumagalli, A. Manzoni, N. Parolini, and M. Verani. Reduced basis approximation and a posteriori error estimates for parametrized elliptic eigenvalue problems. *ESAIM: M2AN*, 50:1857–1885, 2016.

[26] R. N. Gantner, L. Hermann, and Ch. Schwab. Quasi-Monte Carlo integration for affine-parametric, elliptic PDEs: local supports imply product weights. *ETH SAM report:* `https://www.sam.math.ethz.ch/sam_reports/reports_final/reports2016/2016-32.pdf`, version June 2016.

[27] T. Gerstner and M. Griebel. Numerical integration using sparse grids. *Numer. Alg.*, 18:209–232, 1998.

[28] D. Ghosh, R. G. Ghanem, and J. Red-Horse. Analysis of eigenvalues and modal interaction of stochastic systems. *AIAA Journal*, 43(10):2196–2201, 2005.

[29] S. Giani and I. G. Graham. Adaptive finite element methods for computing band gaps in photonic crystals. *Numer. Math.*, 121(1):31–64, 2012.

[30] A. D. Gilbert, I. G. Graham, F. Y. Kuo, R. Scheichl, and I. H. Sloan. Analysis of Quasi-Monte Carlo methods for elliptic eigenvalue problems with stochastic coefficients. *In progress*, 2018.

[31] A. D. Gilbert, F. Y. Kuo, D. Nuyens, and G. W. Wasilkowski. Efficient implementations of the Multivariate Decomposition Method for approximating infinite-variate integrals. *Submitted to SIAM J. Sci. Comp., arXiv:* `https://arxiv.org/pdf/1712.06782.pdf`, version December, 2017.

[32] A. D. Gilbert, F. Y. Kuo, and I. H. Sloan. Hiding the weights — CBC black box algorithms with a guaranteed error bound. *Math. Comp. Simul.*, 143:202–214, 2018.

[33] A. D. Gilbert and G. W. Wasilkowski. Small superposition dimension and active set construction for multivariate integration under modest error demand. *J. Complexity*, 42:94–109, 2017.

[34] P. Glasserman. *Monte Carlo Mothods in Financial Engineering*. Springer-Verlag, New York, NY, USA, 2004.

[35] M. Gnewuch. Infinite-dimensional integration on weighted Hilbert spaces. *Math. Comp.*, 81:2175–2205, 2012.

[36] M. Gnewuch, S. Mayer, and K. Ritter. On weighted Hilbert spaces and integration of functions of infinitely many variables. *J. Complexity*, 30:29–47, 2014.

[37] I. G. Graham, F. Y. Kuo, J.A. Nichols, R. Scheichl, Ch. Schwab, and I. H. Sloan. Quasi-Monte Carlo finite element methods for elliptic PDEs with lognormal random coefficients. *Numer. Math.*, 131:329–368, 2014.

[38] I. G. Graham, F. Y. Kuo, D. Nuyens, R. Scheichl, and I. H. Sloan. Quasi-Monte Carlo methods for elliptic PDEs with random coefficients and applications. *J. Comp. Phys.*, 230(10):3668–3694, 2011.

[39] M. Griebel, M. Schneider, and C. Zenger. A combination technique for the solution of sparse grid problems. In R. Bequwens and P. de Groen, editors, *Iterative Methods in Linear Algebra*, pages 263–281. Elsevier, North Holland, 1992.

[40] J. H. Halton. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numer. Math.*, 2:84–90, 1960.

[41] J. M. Hammersley and D. C. Handscomb. *Monte Carlo Methods*. Methuen & Co., London, UK, 1964.

[42] F. Hang and Y. Li. On applicability of the sparse grid method in the worst case setting. *Numer. Alg.*, 42:95–105, 2006.

[43] M. Hefter, K. Ritter, and G. W. Wasilkowski. On equivalence of weighted anchored and ANOVA spaces of functions with mixed smoothness of order one in L_1 or L_∞ norm. *J. Complexity*, 32:1–19, 2016.

[44] A. Henrot. *Extremum Problems for Eigenvalues of Elliptic Operators*. Birkhäuser Verlag, Basel, Switzerland, 2006.

[45] F. J. Hickernell. Obtaining $O(n^{-2+\epsilon})$ convergence for lattice quadrature rules. In K. T. Fang, F. J. Hickernell, and H. Niederreiter, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2000*, pages 274–289. Springer, Berlin–Heidelberg, Germany, 2002.

[46] F. J. Hickernell and X. Wang. The error bounds and tractability of quasi-Monte Carlo algorithms in infinite dimensions. *Math. Comp.*, 71:1641–1661, 2001.

[47] E. Hlawka. Funktionen von beschränkter Variation in der Theorie der Gleichverteilung. *Ann. Mat. Pura Appl.*, 54:325–333, 1991.

[48] T. Horger, Wohlmuth B., and T. Dickopf. Simultaneous reduced basis approximation of parameterized elliptic eigenvalue problems. *ESAIM: M2AN*, 51:443–465, 2017.

[49] E. Jamelota and P. Ciarlet Jr. Fast non-overlapping Schwarz domain decomposition methods for solving the neutron diffusion equation. *J. Comput. Phys.*, 241:445–463, 2013.

[50] T. Kato. *Perturbation Theory for Linear Operators*. Springer–Verlag, Berlin–Heidelberg, Germany, 1984.

[51] J. F. Koksma. Een algemeene stelling uit de theorie der gelijkmatige verdeeling modulo 1. *Mathematica B (Zutphen)*, 11:7–11, 1942/43.

[52] N. M. Korobov. Approximate evaluation of repeated integrals. *Dokl. Akad. Nauk SSSR*, 124:1207–1210, 1959.

[53] P. Kritzer, F. Pillichshammer, and G. W. Wasilkowski. Very low truncation dimension for high dimensional integration under modest error demand. *J. Complexity*, 35:63–85, 2016.

[54] P. Kuchment. The mathematics of photonic crystals. In *Mathematical Modeling in Optical Science*, pages 207–272. SIAM, Frontiers of Applied Mathematics, Philadelphia, PA, USA, 2001.

[55] F. Y. Kuo. Component-by-component constructions achieve the optimal rate of convergence for multivariate integration in weighted Korobov and Sobolev spaces. *J. Complexity*, 19:301–320, 2003.

[56] F. Y. Kuo, W. T. M. Dunsmuir, I. H. Sloan, M. P. Wand, and R. S. Womersley. Quasi-Monte Carlo for highly structured generalised response models. *Methodol. Comput. Appl. Probab.*, 10:239–275, 2008.

[57] F. Y. Kuo and D. Nuyens. Application of quasi-Monte Carlo methods to elliptic PDEs with random diffusion coefficients. *Found. Comp. Math.*, 16:1631–1696, 2016.

[58] F. Y. Kuo, D. Nuyens, L. Plaskota, I. H. Sloan, and G. W. Wasilkowski. Infinite-dimensional integration and the multivariate decomposition method. *J. Comput. Appl. Math.*, 326:217–234, 2017.

[59] F. Y. Kuo, R. Scheichl, Ch. Schwab, I. H. Sloan, and E. Ullmann. Multilevel quasi-Monte Carlo methods for lognormal diffusion problems. *Math. Comp.*, 86:2827–2860, 2017.

[60] F. Y. Kuo, Ch. Schwab, and I. H. Sloan. Quasi-Monte Carlo finite element methods for a class of elliptic partial differential equations with random coefficients. *SIAM J. Numer. Anal.*, 50(6):3351–3374, 2012.

[61] F. Y. Kuo, Ch. Schwab, and I. H. Sloan. Quasi-Monte Carlo methods for high-dimensional integration: the standard (weighted Hilbert space) setting and beyond. *ANZIAM Journal*, 53:1–37, 2012.

[62] F. Y. Kuo, Ch. Schwab, and I. H. Sloan. Multi-level quasi-Monte Carlo finite element methods for a class of elliptic PDEs with random coefficients. *Found. Comp. Math.*, 15:411–449, 2015.

[63] F. Y. Kuo, I. H. Sloan, G. W. Wasilkowski, and H. Woźniakowski. Liberating the dimension. *J. Complexity*, 26:422–454, 2010.

[64] F. Y. Kuo, I. H. Sloan, G. W. Wasilkowski, and H. Woźniakowski. On decompositions of multivariate functions. *Math. Comp.*, 79:953–966, 2010.

[65] R. Liu and A. Owen. Estimating mean dimensionality of analysis of variance decompositions. *J. Amer. Statist. Assoc.*, 101:712–721, 2006.

[66] L. Machiels, Y. Maday, I. B. Oliveira, A. T. Patera, and D. V. Rovas. Output bounds for reduced-basis approximations of symmetric positive definite eigenvalue problems. *C. R. Acad. Sci. Paris, Sér. I*, 331:153–158, 2000.

[67] H. Niederreiter. *Random number generation and quasi-Monte Carlo methods*. Regional Conference Series in Applied Mathematics. SIAM, 1992.

[68] R. Norton and R. Scheichl. Planewave expansion methods for photonic crystal fibres. *Appl. Numer. Math.*, 63:88–104, 2012.

[69] D. Nuyens and R. Cools. Fast algorithms for component-by-component construction of rank-1 lattice rules in shift-invariant reproducing kernel Hilbert spaces. *Math. Comp.*, 75(254):903–920, 2006.

[70] D. Nuyens and R. Cools. Fast component-by-component construction of rank-1 lattice rules with a non-prime number of points. *J. Complexity*, 22:4–28, 2006.

[71] A. B. Owen. Effective dimension for weighted function spaces. Technical report, Stanford University, 2012.

[72] S. Paskov and J. Traub. Faster valuation of financial derivatives. *J. Portfolio Management*, 22:113–120, 1995.

[73] G. S. H. Pau. Reduced-basis method for band structure calculations. *Phys. Rev. E*, 79(046704), 2007.

[74] C. L. Pettit. Uncertainty quantification in aeroelasticity: recent results and research challenges. *J. Aircraft*, 41(5):1217–1229, 2004.

[75] L. Plaskota and G. W. Wasilkowski. Tractability of infinite-dimensional integration in the worst case and randomized settings. *J. Complexity*, 27:505–518, 2011.

[76] L. Plaskota and G. W. Wasilkowski. Efficient algorithms for multivariate and ∞-variate integration with exponential weight. *Numer. Alg.*, 67:385–403, 2014.

[77] R. Scheichl. Parallel Solution of the Transient Multigroup Neutron Diffusion Equations with Multi-Grid and Preconditioned Krylov-Subspace Methods. Master's thesis, Schriften der Johannes Kepler Universität Linz, Vol. C21, 1997.

[78] V. Sinescu, F. Y. Kuo, and I. H. Sloan. On the choice of weights in a function space for quasi-Monte Carlo methods for a class of generalised response models in statistics. In J. Dick, F. Y. Kuo, G. W. Peters, and I. H. Sloan, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2012*, pages 631–647, Berlin–Heidelberg, Germany, 2013. Springer.

[79] I. H. Sloan and S. Joe. *Lattice Methods for Multiple Integration*. Oxford University Press, Oxford, UK, 1994.

[80] I. H. Sloan, F. Y. Kuo, and S. Joe. Constructing randomly shifted lattice rules in weighted Sobolev spaces. *SIAM J. Numer. Anal.*, 40(5):1650–1665, 2002.

[81] I. H. Sloan and A.V. Reztsov. Component-by-component construction of good lattice rules. *Math. Comp.*, 71(237):263–273, 2001.

[82] I. H. Sloan and H. Woźniakowski. When are quasi-Monte Carlo algorithms efficient for high dimensional integrals? *J. Complexity*, 14(1):1–33, 1998.

[83] S. A. Smolyak. Quadrature and interpolation formulas for tensor products of certain classes of functions. *Dokl. Acad. Nauk. SSSR*, 4:240–243, 1963.

[84] I. M. Sobol'. Distribution of points in a cube and approximate evaluation of integrals. *Ž. Vyčisl. Mat. i Mat. Fiz.*, 7:784–802, 1967.

[85] I. M. Sobol'. Sensitivity estimates for nonlinear mathematical models. *Matematicheskoe Modelirovanie*, 2:112–118, 1990.

[86] G. Strang and G. Fix. *An Analysis of the Finite Element Method*. Wellesley-Cambridge Press, Wellesley, MA, USA, 1973.

[87] E. L. Wachspress. *Iterative Solution of Elliptic Systems and Applications to the Neutron Diffusion Equations of reactor Physics*. Prentice-Hall, Inc., Englewood-Cliffs, NJ, USA, 1966.

[88] X. Wang and K.-T. Fang. Effective dimension and quasi-Monte Carlo integration. *J. Complexity*, 19:101–124, 2003.

[89] G. W. Wasilkowski. On tractability of linear tensor product problems for ∞-variate classes of functions. *J. Complexity*, 29:351–369, 2013.

[90] G. W. Wasilkowski. Tractability of approximation of ∞-variate functions with bounded mixed partial derivatives. *J. Complexity*, 30:325–346, 2014.

[91] G. W. Wasilkowski and H. Woźniakowski. Explicit cost bounds of algorithms for multivariate tensor product problems. *J. Complexity*, 11:1–56, 1995.

[92] M. M. R. Williams. A method for solving stochastic eigenvalue problems. *Appl. Math. Comput.*, 215(11):4729–4744, 2010.

[93] M. M. R. Williams. A method for solving stochastic eigenvalue problems II. *Appl. Math. Comput.*, 219(9):4729–4744, 2013.

[94] D. Zhang. *Stochastic Methods for Flow in Porous Media: Coping with Uncertainties*. Academic Press, San Diego, CA, USA, 2002.

[95] Z. Zhang, W. Chen, and X. Cheng. Sensitivity analysis and optimization of eigenmode localization in continuum systems. *Struct. Multidiscip. O.*, 52(2):305–317, 2015.