Chapter 1

Introduction

A fundamental task in computational science and engineering involves accurately approximating a smooth target function from limited data. Such a task arises notably in the study of parametric models of physical processes. Here the variables represent the parameters in the system, e.g., material properties, forcing terms, or boundary information, and the parametric model is often represented as a (system of) Differential Equations (DEs) or Partial Differential Equations (PDEs) depending on these parameters. Important objectives involve understanding how the choice of such parameters affect the output(s) of the system and, in the stochastic setting, understanding how uncertainty in the parameter values propagates to its output – the latter being one of the key tasks in computational Uncertainty Quantification (UQ) [62, 91, 128, 131].

1.1 High-dimensional function approximation from limited samples

Abstractly, this task can be recast as that of approximating an unknown target function

$$f : \mathcal{U} \to \mathcal{V}, \quad \boldsymbol{y} \mapsto f(\boldsymbol{y}),$$

from *sample values* (or *snapshots*)

$$f(\boldsymbol{y}_1), \dots, f(\boldsymbol{y}_m). \tag{1.1}$$

Here, the input space $\mathcal{U}$ is typically a subset of $\mathbb{R}^d$ (in the finite-dimensional case) or $\mathbb{R}^{\mathbb{N}}$ (in the infinite-dimensional case). The output space $\mathcal{V}$ could either be a scalar field, a finite-dimensional vector space or an infinite-dimensional Banach or Hilbert space.

This problem is challenging in a number of ways. First, the dimension d is high, since modern parametric models typically involve many parameters. It may also be infinite, e.g., in the case of a random field represented via its Karhunen–Loève expansion. Therefore, care must be taken to design methods that scale well with dimension. In addition, the amount of samples m is often highly limited. For example, in the parametric DE setting, each evaluation of f involves an expensive computational simulation. The data (1.1) is also always corrupted by errors, due to noise in physical experiments or numerical error in solving a DE. And finally, since the output $f(\boldsymbol{y})$ is often the solution of DE parametrized by the vector $\boldsymbol{y}$, it may consequently take values in an infinite-dimensional Banach or Hilbert space. While it is commonplace to circumvent this issue in practice by considering scalar-valued *quantities of interest* (i.e., functions of the form $g(\boldsymbol{y}) = \mathcal{Q}(f(\boldsymbol{y}))$ for some known map $\mathcal{Q} : \mathcal{V} \to \mathbb{C}$),

approximating the full function f is both of theoretical interest and practical importance [52].

Remark 1.1. As a further consideration, we note that in many scenarios one may have substantial flexibility to choose the sample points $\boldsymbol{y}_1, \dots, \boldsymbol{y}_m \in \mathcal{U}$ in (1.1). However, in other scenarios they may be fixed, e.g., when dealing with legacy data. In this work, we consider Monte Carlo sampling – which may be considered either as a designed sampling strategy or a fixed one, depending on the setting. Here, the samples are drawn randomly and independently of an underlying probability measure on $\mathcal{U}$. This is very common in practice, in particular in UQ settings.

1.2 Smoothness and best s-term polynomial approximation

A key characteristic of parametric model problems is that the target function f is often smooth. There is now a large body of literature that has established that solution maps of a wide range of different parametric DEs are *holomorphic* (i.e., *analytic*) functions of their parameters. We mention in passing problems such as elliptic PDEs with affine and (certain) nonaffine parametric dependence, parabolic PDEs, PDEs over parametrized domains and shape uncertainty, parametric Initial Value Problems (IVPs), parametric hyperbolic problems and parametric control problems. Classical results in this area can be found in, e.g., [146]. For surveys of more recent results, we refer to [41] and [8] and references therein.

In tandem with the effort to establish holomorphic regularity of parametric DEs, there has also been a focus on applying polynomial methods, and in particular, *best s-term polynomial approximation* to construct finite approximations to such functions. In best s-term approximation, the function f is approximated by an s-term expansion corresponding to its largest s coefficients (measured in the $\mathcal{V}$-norm) with respect to a polynomial basis. Common choices include multivariate Taylor polynomials, tensor-product Legendre and Chebyshev polynomials on bounded hypercubes or tensor-product Hermite and Laguerre polynomials on $\mathbb{R}^d$ or $[0, \infty)^d$, respectively. Over the last fifteen years, there have been significant developments in the approximation theory of such techniques (see Section 1.6). Signature results have established *exponential* and *algebraic* convergence rates for the best s-term approximation. The former assert that the error decays at least exponentially fast in $s^{1/d}$ in finite dimensions for any holomorphic function. The latter assert that the error decays algebraically fast; specifically, like $s^{1/2-1/p}$ for some $0 < p < 1$. These algebraic rates also hold in infinite dimensions, thus establishing best s-term approximation as a (theoretical) means to approximate holomorphic functions of infinitely many variables. We review several such results in Section 2.6.

1.3 Computing sparse polynomial approximations

Unfortunately, the best s-term approximation cannot usually be computed from the samples (1.1). Indeed, constructing it in theory involves computing and then searching over infinitely many coefficients. Both tasks are generally impossible. Therefore, there has also been a focus on methods to compute accurate polynomial approximations from sample values.

One line of work focuses on *least-squares* methods, wherein a polynomial approximation (or sequence of approximations) is computed in a fixed polynomial subspace (or sequence of nested subspaces). See Section 1.6 for relevant references. Such methods are essentially optimal if a (sequence of) polynomial subspace that gives a quasi-best s-term approximation is known.

However, this information is generally unavailable in practice (although it may be for certain simple parametric DEs). It essentially equates to knowing the region of holomorphy of the underlying function, which is itself similar to knowing the order of importance of the parametric variables, and their relative strengths. To counter this, there are *adaptive* least-squares methods [33, 35, 41, 45, 63, 101, 102]. Here one strives to construct such subspaces adaptively using the given data (1.1), typically via a greedy procedure. However, these methods currently lack theoretical guarantees [35, 41].

To overcome this limitation, there has also been a substantial focus on methods inspired by *compressed sensing* [13, 61, 145]. See Section 1.6 once more for relevant references. These methods seek a polynomial approximation in a larger subspace, whose coefficients are defined as a minimizer of an ℓ^1- or weighted ℓ^1-minimization problem. A key component of this endeavour has been to determine the *sample complexity* of such schemes, i.e., quantifying how many (typically Monte Carlo) samples m are sufficient to obtain an approximation with a certain guaranteed error bound, involving a (weighted) best approximation error plus a truncation error. Yet, precise rates of approximation (i.e., algebraic or exponential in m) have typically not been derived for these schemes in previous works. Another key limitation of past work is that such methods are not algorithms per se. Indeed, they consider exact minimizers of nonlinear optimization problems, which cannot be computed exactly in finitely many arithmetic operations.

1.4 Problem and main contributions

Least-squares and compressed sensing techniques are commonly applied to compute polynomial approximations to parametric and stochastic DEs. However, as explained above, there is a key gap between theory and practice. The theory of the best s-term approximation asserts the existence of polynomial approximations that attain specific

algebraic or exponential rates of convergence for arbitrary holomorphic functions. Yet, it is currently unknown whether similar rates in terms of the number of samples m can be obtained via an algorithm that computes a polynomial approximation from the samples (1.1) in finitely many arithmetic operations. The purpose of this work is to close this gap.

We now describe the problem considered in this work. Let $\mathcal{U} = [-1, 1]^d$, where $d \in \mathbb{N}$ or $d = \infty$, and $\mathcal{V}$ be an arbitrary separable Hilbert space. Let ϱ be either the uniform or Chebyshev (arcsine) measure and consider the associated tensor-product Legendre or Chebyshev polynomials. Now let $f : \mathcal{U} \to \mathcal{V}$ be the unknown target function that we seek to approximate, draw m sample points $\boldsymbol{y}_1, \dots, \boldsymbol{y}_m$ i.i.d. from ϱ and let

$$d_i = f(\boldsymbol{y}_i) + n_i, \quad i = 1, \dots, m, \tag{1.2}$$

be m noisy samples of f. Then, informally stated, the problem we study in this work is the following: *devise algorithms that take* (1.2) *as input and compute the coefficients of a polynomial approximation* $\hat{f}$ *to* f *with guarantees on both the computational complexity and the error* $f - \hat{f}$. Note that the formal problem statement involves several technicalities (in particular, the definition of an algorithm), so we defer it to Section 3.2.

Our main contributions are on the existence of such algorithms (see Tables 4.2 and 4.3 and Algorithms 2 and 5). Let $\|\cdot\|_{L^2_\varrho(\mathcal{U};\mathcal{V})}$ be the Lebesgue–Bochner norm. Then, in all cases, we establish an error bound of the form

$$\|f - \hat{f}\|_{L^2_\varrho(\mathcal{U};\mathcal{V})} \lesssim E_{\mathsf{app}} + E_{\mathsf{samp}} + E_{\mathsf{disc}} + E_{\mathsf{alg}}, \tag{1.3}$$

with probability at least $1 - \epsilon$ with respect to the (Monte Carlo) draw of the sample points $\boldsymbol{y}_i$. This bound provides a complete accounting for the main sources of error in the problem.

- E_{app} is a *polynomial approximation error* term. Depending on the specific setup, it decays algebraically (Theorems 3.4–3.9) or exponentially (Theorems 3.10–3.12) with respect to m (up to several log terms). For instance, in the infinite-dimensional setting (Theorems 3.7–3.9), this term is given by

$$E_{\mathsf{app}} = C \cdot \left(\frac{m}{c_0 L}\right)^{1/2-1/p}, \quad L = \log(m) \cdot (\log^3(m) + \log(\epsilon^{-1})), \tag{1.4}$$

  where $c_0 \geq 1$ is a universal constant, C is a constant depending on (the region of holomorphy of) f only, $p \in (0, 1)$ is a parameter determined by the region of holomorphy of f and $0 < \epsilon < 1$ is the failure probability of (1.3). It is completely equivalent to the corresponding algebraic decay rate (Theorem 2.5) for the best s-term approximation error, except with s replaced by $m/(c_0 L)$.

- E_{samp} is the *sampling error*. It is equal to

$$E_{\text{samp}} = \sqrt{\frac{1}{m}\sum_{i=1}^{m} \|n_i\|_{\mathcal{V}}^2},$$

  i.e., the norm of the error in the samples (1.2). The presence of this term means that the algorithms are robust to errors in the samples.

- E_{disc} is the *physical discretization error*. This term accounts for the fact that an algorithm cannot work with (i.e., take as input, or perform computations in) $\mathcal{V}$ when it is an infinite-dimensional Hilbert space. The algorithms (see Tables 4.2 and 4.3 and Algorithms 2 and 5) therefore work in a finite-dimensional discretization space $\mathcal{V}_h \subseteq \mathcal{V}$. This is a standard step in parametric DEs, where discretization is often performed via techniques such as the Finite Element Method (FEM). In this case, $\mathcal{V}_h$ is a finite element space. The term E_{disc} quantifies the effect of this error. It is given by

$$E_{\text{disc}} = \|f - \mathcal{P}_h(f)\|_{L^\infty(\mathcal{U};\mathcal{V})},$$

  where $\mathcal{P}_h : \mathcal{V} \to \mathcal{V}_h$ is the orthogonal projection onto $\mathcal{V}$. In other words, the effect of working in $\mathcal{V}_h$ instead of $\mathcal{V}$ is determined by the error of the (pointwise) best approximation $\mathcal{P}_h(f)$ to f from $\mathcal{V}_h$. If $\mathcal{V}$ has finite dimension, we may set

$$\mathcal{V}_h = \mathcal{V},$$

  which implies that $E_{\text{disc}} = 0$ in this case.

- E_{alg} is the *algorithmic error*. It depends on the number of iterations t performed by the algorithm that computes the coefficients of the polynomial approximation $\hat{f}$. We construct one type of algorithm (see Table 4.2 and Algorithm 2) where this term is $\mathcal{O}(1/t)$ as $t \to \infty$. This decay is relatively slow, especially in the regime where E_{app} is exponentially small in m. However, we also present an *efficient* algorithm (Table 4.3 and Algorithm 5) for which this term decays exponentially fast in t (specifically, $\mathcal{O}(\mathrm{e}^{-t})$ as $t \to \infty$), subject to an additional theoretical constraint. This constraint is seemingly an artefact of the proof. Our numerical experiments suggest it is unnecessary in practice.

We also determine the computational cost of the algorithms in all cases. Here, we draw two main conclusions.

- In the infinite-dimensional case (Theorems 3.8–3.9), the computational cost is *subexponential* in m. Specifically, after t iterations of the algorithm, it is

$$\mathcal{O}\left(t \cdot m^{1+(\alpha+1)\log(4m)/\log(2)}\right), \quad m \to \infty,$$

  where $\alpha = 1$ (Legendre) or $\alpha = \log(3)/\log(4) \approx 0.79$ (Chebyshev).

- In the finite-dimensional, exponential setting (Theorems 3.11–3.12), the computational cost is *algebraic* in m for fixed d: namely,

$$\mathcal{O}\left(t \cdot m^{\alpha+2}(\log(m))^{(d-1)(\alpha+1)}\right), \quad m \to \infty,$$

  for the same values of α.

Note that these computational cost estimates also depend polynomially on the dimension of the discretization space $\mathcal{V}_h$.

1.5 Discussion and further contributions

This work bridges a gap between the best s-term polynomial approximation theory and algorithms for computing such approximations from sample values. In particular, it asserts algebraic and exponential rates with respect to the number of samples m that are highly similar to those of the best approximation. In other words, polynomial approximations of holomorphic functions can be achieved in a *sample efficient* manner. Furthermore, they can be computed in supexponential or algebraic computational cost.

Our main results assume holomorphy of the underlying function in order to attain these rates. However, they require no *a priori* knowledge of the region of holomorphy. As discussed, if such information is available, then least-squares methods can be used more straightforwardly to compute an approximation. The holomorphy assumption is made in order to have concrete algebraic and exponential rates. However, our algorithms exist independently of this smoothness assumption. It would be possible to also provide rates for other classes of functions, e.g., those possessing finite orders of (mixed) smoothness. We use holomorphy as our assumption due to its strong connections with the theory of parametric DEs.

Our algorithms and analysis are based on compressed sensing theory and involve computing approximate minimizers of certain weighted ℓ^1-minimization problems. Here we make several additional contributions.

(i) We provide precise error rates for polynomial approximation via compressed sensing. As noted, most prior work on compressed sensing involves quantifying the sample complexity to obtain a certain (weighted) best approximation error. We impose the holomorphy assumption to obtain specific algebraic and exponential rates.

(ii) Prior works consider polynomial approximations formed by exact minimizers of nonlinear optimization problems. We introduce novel, efficient algorithms to compute approximate minimizers in finite computational time (see also below).

(iii) While these algorithms are motivated by the desire to have full theoretical guarantees, they are also completely practical. We present a series of numerical experiments demonstrating their practical efficacy. In fact, our experiments show that these algorithms work even better than our theoretical results suggest.

(iv) Most prior works on compressed sensing (with the exception of [52]) focus on scalar-valued functions, e.g., quantities of interest of parametric DEs. We develop algorithms that work in the Hilbert-valued setting, and, crucially, provide error bounds that take into account physical discretization error (see above).

More precisely, we first formulate the approximation problem as the recovery of a finite, Hilbert-valued vector (i.e., an element of $\mathcal{V}^N$) via a so-called weighted, Square-Root LASSO (SR-LASSO) optimization problem. The use of the SR-LASSO, as opposed to the classical LASSO or various constrained formulations, is crucial to this work. It is *noise-blind*. Hence, it allows us to devise algorithms that do not require any a priori (and generally unavailable) estimates on the measurement error n_i in (1.2) or the truncation error with respect to the finite polynomial space in which the approximation is constructed.

To develop algorithms, we employ two key ideas. First, we use a powerful, general-purpose first-order optimization method for solving nonsmooth, convex optimization problems. Second, we use the technique of *restarts* to drastically accelerate its convergence. For the former, we employ the *primal-dual iteration* (also known as the Chambolle–Pock algorithm) [30,31]. We present error bounds for this method for solving the Hilbert-valued, weighted SR-LASSO, which decay like $\mathcal{O}(1/t)$, where t is the iteration number. Next, we use a novel restarting procedure, recently introduced in [47,48], to obtain faster, exponential decay of the form $\mathcal{O}(\mathrm{e}^{-t})$.

To the best of our knowledge, this is the first time either the primal-dual iteration or a restarting scheme has been applied to the problem of sparse polynomial approximation. Many existing works use blackbox solvers such as SPGL1 [143,144]. See [52] for a forward-backwards splitting technique in combination with Bregman iterations and fixed-point continuation and [142] for an approach based on Douglas–Rachford splitting. Besides its amenability to theoretical analysis, the primal-dual scheme is also particularly attractive because of its insensitivity to parameter choices and the possibility of performing acceleration via restarts.

As noted, polynomial-based methods have become popular tools for the practical approximation high-dimensional, holomorphic functions arising in problems in computational science and engineering. However, they are by no means the only method. Other popular techniques include Gaussian processes (also known as kriging) [128,131], radial basis methods [84,128], reduced-order methods [78,118] and,

recently, methods based on deep neural networks and deep learning [5, 6, 11, 49, 59, 60, 76, 92, 114, 115, 125]. Our goal in this work is to develop algorithms for constructing polynomial approximations that achieve the same rates as the theoretical benchmark provided by the best s-term polynomial approximation. An important consideration that we do not address in this work is *tractability* and the *information complexity* [111, 113] of these classes of functions and, in particular, whether polynomial-based methods constitute *optimal algorithms*. This question has been studied in the infinite-dimensional case in recent work [12]. Here, it is shown that the rate $m^{1/2-1/p}$ is a lower bound for the *(adaptive) m-width* for such classes, i.e., no combination of m (adaptive) linear samples and a (potentially nonlinear) reconstruction map can achieve an approximation error decaying faster than this rate. Notice that this rate is the same, up to constants and logarithmic factors, as (1.4). Unfortunately, this does not imply our algorithms are near optimal for this problem – and, moreover, that *standard information*, i.e., pointwise samples, constitutes near-optimal information – because our theoretical results in the infinite-dimensional case are *nonuniform*. See Remark 3.13 for further discussion on this point, and Chapter 11 for further comments on tractability.

1.6 Related work

The systematic study of best s-term polynomial approximation of high- or infinite-dimensional holomorphic functions began around 2010 with the works of [25, 42, 43, 75, 139]. For reviews, see [41] and [8, Chapter 3]. Note that many of these works assume the function is a solution of a parametric PDE, and therefore first demonstrate that such a function is holomorphic. However, other works avoid this step and use specific properties of the DE to obtain refined estimates. See, e.g., [19, 20] for results of this type. Other recent works such as [8, 27] also study the problem without assuming the function is a solution of a parametric PDE.

The study of least-squares method for constructing such approximations from sample points began in the early 2010s [34, 40, 100, 105]. There has since been significant research on this topic. Many subsequent works have pursued extensions, such as enhanced sampling strategies [65, 104, 107, 126, 135, 155, 156], near-optimal sampling strategies [9, 44, 71], optimal sampling strategies [21, 54, 56, 85, 93, 137], methods for general domains [14, 55, 103], optimal and adaptive methods [46, 101, 102] and multilevel strategies [70]. See [45, 67, 69] and [8, Chapter 5] for reviews.

Compressed sensing was introduced in the context of image and signal processing by modeling image and signals as sparse vectors [13, 29, 57, 61]. Its use in polynomial approximation started early in the last decade with the works of [26, 58, 99, 120, 149]. This has also led to substantial research. See [51, 52, 58, 99, 119, 151] and references therein for applications to parametric PDEs. Various extensions include refined sampling strategies [17, 53, 68, 72, 83, 94, 134], iterative methods and basis selection

techniques [16, 74, 142, 152, 152–154], nonconvex optimization methods [64, 140, 148, 150], sublinear-time algorithms [38, 39], gradient-enhaced minimization techniques [15, 66, 82, 117, 130, 133], methods for dealing with corrupted samples [3, 7, 80, 127] and multilevel and multifidelity strategies [28, 109]. For additional information and reviews, see [73, 86, 97, 98, 108] and [8, Chapter 7].

Our work combines and extends several key elements of this literature. First, weighted ℓ^1-minimization, which was developed in [1–4, 37, 116, 121, 151] and [8, Chapter 6–7]. Second, the notions of *lower* and *anchored* sets (see Section 2.7). These have been extensively studied in the best s-term polynomial approximation literature. Compressed sensing techniques aiming to exploit such structures were first considered in [2, 3, 37] and [8, Chapter 7]. Third, the extension of classical compressed sensing theory from vectors in $\mathbb{R}^N$ (or $\mathbb{C}^N$) to Hilbert-valued vectors in $\mathcal{V}^N$. This was first developed in [52]. In order to prove our main results, we also extend this work to the weighted setting.

See [30, 31, 31] for more on the primal-dual iteration and [122–124] for the general notion of restarts in continuous optimization. Note that there are also various non-optimization based techniques in the compressed sensing literature (see, e.g., [61]), including iterative thresholding and greedy methods. The latter are closely related to the adaptive least-squares methods discussed earlier [8, Section 6.2.5]. However, such techniques currently do not possess theoretical guarantees in the weighted setting.

There have been several previous attempts to connect compressed sensing theory for analyzing the sample complexity of polynomial approximations via (weighted) ℓ^1-minimization and best s-term polynomial approximation theory. In [119], the authors consider approximating scalar quantities of interest of solutions to affine parametric operator equations in Banach spaces. Assuming a certain weighted summability criterion, they first show holomorphy of the parametric solution map and then use a weighted ℓ^1-minimization procedure in combination with Chebyshev polynomials to derive algebraic rates of convergence, similar to (1.4). Our work is more general, since its starting point is a holomorphic function, not a solution of a parametric operator equation. We also consider Hilbert-valued functions, i.e., the whole solution map, not a scalar quantity of interest of it. Moreover, the work of [119] is based on exact minimizers of certain constrained, weighted ℓ^1-minimization problems, whereas we construct full algorithms. Recently, at the same time as writing this work, some similar results were presented in the book [8] written by two of the authors. However, these only consider the scalar-valued case and do not address algorithms, which is the main focus of this work.

1.7 Outline

The remainder of this work proceeds as follows. We commence in Chapter 2 by introducing preliminaries, including key notation and best s-term polynomial