Chapter 3

Problem statement and main results

In this chapter, we first formally define the problem we aim to solve before stating our main results. This work concerns algorithms for computing approximation of Hilbert-valued functions from finitely many sample values. We define this concept formally in a moment. For now, though, we consider that an algorithm must take a finite input and produce a finite output. Hence, in order to discuss algorithms, we first need to define what these finite inputs and outputs are in our setting.

3.1 Samples

Let $f \in L^2_\varrho(\mathcal{U};\mathcal{V})$ be the function we seek to approximate. Throughout this work, we consider m *sample points* $\boldsymbol{y}_1,\dots,\boldsymbol{y}_m \in \mathcal{U}$ drawn randomly and independently according to the probability measure ϱ. Corresponding to each sample point, we consider the noisy *sample values*

$$d_i = f(\boldsymbol{y}_i) + n_i \in \mathcal{V}_h, \quad i = 1,\dots,m,$$

where $\boldsymbol{n} = (n_i)_{i=1}^m \in \mathcal{V}^m$ is an error term, referred to as the *sampling error*. Observe that the samples values d_i are assumed to be elements of the finite-dimensional space $\mathcal{V}_h$. This is a natural assumption to make. Indeed, in the context of parametric DEs, the value $f(\boldsymbol{y})$ (the solution of the DE with parameter value $\boldsymbol{y}$) is typically computed via a (finite element) discretization of the DE, thus yielding an element of $\mathcal{V}_h$, which is the corresponding discrete (finite element) space.

As a result of the assumption $d_i \in \mathcal{V}_h$, the error term n_i encompasses the error involved in approximating $f(\boldsymbol{y}_i) \in \mathcal{V}$ by an element of $\mathcal{V}_h$, e.g., the (finite element) discretization error in the context of a parametric DE. Note that we do not specify precisely how such an approximation is performed, nor how large an error this results in. In other words, we consider the computation that evaluates f at $\boldsymbol{y}_i$ as a black box. A particular case of interest is when the d_i are the orthogonal projections of the exact sample values $f(\boldsymbol{y}_i)$, i.e.,

$$d_i = \mathcal{P}_h(f(\boldsymbol{y}_i)), \quad i = 1,\dots,m.$$

However, we do not assume this in what follows, since in practice the numerical procedure that yields the d_i may not involve computing the projection $\mathcal{P}_h$. Our goal is to develop algorithms for which the error scales linearly in $\|\boldsymbol{n}\|_{2;\mathcal{V}}$, the norm of the noise, thus accounting for any black box mechanism for computing the samples.

Recall that we consider a basis $\{\varphi_k\}_{k=1}^K$ for $\mathcal{V}_h$. We assume that the computation that evaluates $f(\boldsymbol{y}_i)$ produces the coefficients of the sample values d_i in this basis (i.e., the finite element coefficients in the aforementioned example). Therefore, we now write the sample values as

$$d_i = f(\boldsymbol{y}_i) + n_i = \sum_{i=1}^{K} d_{ik}\varphi_k, \quad i = 1, \dots, m, \tag{3.1}$$

and consider the values $d_{ik} \in \mathbb{C}$ as the *data* we obtain by sampling f.

3.2 Problem statement

We now formally define the input and output of the algorithm. The *input* of the algorithm is the collection of sample points $(\boldsymbol{y}_i)_{i=1}^m$ and the array of mK values $(d_{i,k})_{i,k=1}^{m,K} \in \mathbb{C}^{m\times K}$ defined by (3.1). We next define the output. To this end, we first fix a multi-index set $\Lambda \subset \mathcal{F}$ of size $|\Lambda| = N$ for some $N \geq 1$. This set defines a polynomial space $\mathcal{P}_{\Lambda;\mathcal{V}_h}$, as in (2.9), within which we shall construct the resulting polynomial approximation. Hence, we consider an approximation of the form $\hat{f} \in \mathcal{P}_{\Lambda;\mathcal{V}_h}$ given by

$$\hat{f} : \boldsymbol{y} \mapsto \sum_{j=1}^{N} \left(\sum_{k=1}^{K} \hat{c}_{jk}\varphi_k \right) \Psi_{\boldsymbol{\nu}_j}(\boldsymbol{y}), \tag{3.2}$$

where $\hat{c}_{j,k} \in \mathbb{C}$ for $j \in [N]$, $k \in [K]$ and $\boldsymbol{\nu}_1, \dots, \boldsymbol{\nu}_N$ is some indexing of the multi-indices in Λ. In this way, we define formally the *output* of the algorithm as the array of coefficients $(\hat{c}_{jk})_{j,k=1}^{N,K} \in \mathbb{C}^{N\times K}$.

Finally, in order to define an algorithm we need one additional ingredient. Let

$$\boldsymbol{G} = (\langle \varphi_j, \varphi_k \rangle_{\mathcal{V}})_{j,k=1}^K \in \mathbb{C}^{K\times K} \tag{3.3}$$

denote the Gram matrix of the basis $\{\varphi_k\}_{k=1}^K \subset \mathcal{V}_h$. Note that $\boldsymbol{G}$ is self adjoint and positive definite. However, $\boldsymbol{G}$ is only equal to the identity when $\{\varphi_k\}_{k=1}^K$ is orthonormal. In what follows, we assume that it is possible to perform matrix-vector multiplications with $\boldsymbol{G}$. In other words, we have access to the function

$$\mathcal{T}_{\boldsymbol{G}} : \mathbb{C}^K \to \mathbb{C}^K, \quad \boldsymbol{x} \mapsto \boldsymbol{G}\boldsymbol{x}.$$

We also write $F(\boldsymbol{G})$ for the maximum number of arithmetic operations and comparisons required to evaluate $\mathcal{T}_{\boldsymbol{G}}(\boldsymbol{x})$ for arbitrary $\boldsymbol{x}$. Note that $F(\boldsymbol{G}) \leq K^2$ in general. However, this may be smaller when $\boldsymbol{G}$ is structured. For instance, in the case of a finite element discretization, this computation can often be performed in $\mathcal{O}(K)$ operations.

Definition 3.1 (Algorithm for polynomial approximation of Hilbert-valued functions). Let $\Lambda \subset \mathcal{F}$ of size $|\Lambda| = N$ be given, along with an indexing $\boldsymbol{\nu}_1, \ldots, \boldsymbol{\nu}_N$ of the multi-indices in Λ. An *algorithm for polynomial approximation of Hilbert-valued functions from sample values* is a mapping

$$\mathcal{A} : \mathcal{U}^m \times \mathbb{C}^{m\times K} \to \mathbb{C}^{N\times K}, \quad \left((\boldsymbol{y}_i)_{i=1}^m, (d_{i,k})_{i,k=1}^{m,K}\right) \mapsto (\hat{c}_{jk})_{j,k=1}^{N,K},$$

for which the evaluation of $\mathcal{A}((\boldsymbol{y}_i), (d_{i,k}))$ involves only finitely many arithmetic operations (including square roots), comparisons and evaluations of the matrix-vector multiplication function $\mathcal{T}_{\boldsymbol{G}}$. If (d_{ik}) is as in (3.1) for some $f \in L^2_\varrho(\mathcal{U}; \mathcal{V})$, then the resulting approximation $\hat{f}$ of f is given by (3.2), where $(\hat{c}_{jk}) = \mathcal{A}((\boldsymbol{y}_i), (d_{i,k}))$. The *computational cost* of an algorithm $\mathcal{A}$ is the maximum number of arithmetic operations and comparisons (including those used in the evaluation of $\mathcal{T}_{\boldsymbol{G}}$) used to compute the output from any input.

Remark 3.2. As formulated above, it is up to the user to choose a suitable multi-index set Λ. Fortunately, as we see in our main results below, this multi-index set is given simply and explicitly in terms of m and another parameter ϵ (a failure probability). In particular, no 'oracle' knowledge of the function being approximated is required. Thus, one can also make the stronger assertion in what follows in which the algorithm takes the same input, but outputs both the desired index set Λ and the polynomial coefficients. For ease of presentation, we shall not do this.

Remark 3.3. When $d = \infty$ each sample point $\boldsymbol{y}_i$ is an infinite sequence of real numbers. It is implicit in Definition 3.1 that the algorithm only accesses finitely many entries of this sequence. This does not cause any problems. As noted, the polynomial approximation is obtained in the index set Λ, which is a finite subset of $\mathcal{F}$. Hence, the multi-indices in Λ are nonzero only in their first n entries, for some finite n. Therefore, it is only necessary to access the first n entries of each sequence $\boldsymbol{y}_i$. More concretely, in our main results below, the polynomial approximation in infinite dimensions is obtained in a multi-index set $\Lambda = \Lambda_n^{\mathsf{HCI}}$ in which only the first n terms can be nonzero, where n is an integer given explicitly in terms of m and ϵ.

3.3 Main results

We now present the main results of this work. We reiterate at this stage that these results are formulated for Chebyshev and Legendre polynomials. See Chapter 11 for some further discussion on other polynomial systems.

As noted above, these results employ specific choices of the index set Λ in order to obtain the desired approximation rates. Specifically, in finite dimensions, we consider the *hyperbolic cross* index set

$$\Lambda = \Lambda^{\mathsf{HC}}_{n,d} = \left\{ \boldsymbol{\nu} = (\nu_k)_{k=1}^{d} \in \mathbb{N}_0^d : \prod_{k=1}^{d} (\nu_k + 1) \leq n \right\} \subset \mathbb{N}_0^d. \tag{3.4}$$

We term n the *order* of the hyperbolic cross. Note that it is common to consider (3.4) as the hyperbolic cross of order $n-1$. We use n here as it is slightly more convenient for this work. When defined this way, $\Lambda^{\mathsf{HC}}_{n,d}$ is in fact the union of all lower sets (see Definition 2.8) in d dimensions of size at most n (see, e.g., [8, Proposition 2.5]). Thus, this set is a natural choice for polynomial approximation.

In infinite dimensions, we define the following index set

$$\Lambda = \Lambda^{\mathsf{HCI}}_{n} = \left\{ \boldsymbol{\nu} = (\nu_k)_{k=1}^{\infty} \in \mathcal{F} : \prod_{j=1}^{n} (\nu_k + 1) \leq n,\ \nu_k = 0,\ k > n \right\} \subset \mathcal{F}.$$

Similarly, the union of all anchored sets (Definition 2.8) of size at most n in infinite dimensions is a subset of $\Lambda^{\mathsf{HCI}}_{n}$ (see, e.g., [8, Proposition 2.18]). Note that $\Lambda^{\mathsf{HCI}}_{n}$ is isomorphic to $\Lambda^{\mathsf{HC}}_{n,n}$ under the restriction map

$$\boldsymbol{\nu} = (\nu_k)_{k=1}^{\infty} \in \mathcal{F} \mapsto (\nu_k)_{k=1}^{n} \in \mathbb{N}_0^d.$$

For convenience, we now also define

$$N = \Theta(n,d) = \begin{cases} |\Lambda^{\mathsf{HC}}_{n,d}| & d < \infty, \\ |\Lambda^{\mathsf{HCI}}_{n}| = |\Lambda^{\mathsf{HC}}_{n,n}| & d = \infty, \end{cases} \tag{3.5}$$

as the cardinality of the index set employed. In general, the exact behaviour of $\Theta(n,d)$ is unknown. However, it admits a variety of different bounds. These are summarized as follows for $d < \infty$:

$$N = |\Lambda^{\mathsf{HC}}_{n,d}| \leq \min\left\{ 2n^3 4^d, \mathrm{e} n^{2+\log(d)/\log(2)}, \frac{n(\log(n) + d\log(2))^{d-1}}{(d-1)!} \right\}. \tag{3.6}$$

The bounds are based on [32, 89]. See also [8, Lemmas B.3–B.5].

Finally, we also define

$$\alpha = \begin{cases} 1 & \text{Legendre}, \\ \log(3)/\log(4) & \text{Chebyshev}, \end{cases} \tag{3.7}$$

and, given $m \geq 3$ and $\epsilon \in (0,1)$, $L = L(m,d,\epsilon)$ as

$$L = \begin{cases} \log(m) \cdot (\log(m) \cdot \min\{\log(m) + d, \log(\mathrm{e}d) \cdot \log(m)\} + \log(\epsilon^{-1})) & d < \infty, \\ \log(m) \cdot (\log^3(m) + \log(\epsilon^{-1})) & d = \infty. \end{cases} \tag{3.8}$$

3.3.1 Algebraic rates of convergence, finite dimensions

Theorem 3.4 (Existence of a mapping; algebraic case, finite dimensions). *Let $d \in \mathbb{N}$, $\{\Psi_{\boldsymbol{\nu}}\}_{\boldsymbol{\nu} \in \mathbb{N}_0^d} \subset L^2_{\varrho}(\mathcal{U})$ be either the orthonormal Chebyshev or Legendre basis and $\{\varphi_k\}_{k=1}^K$ be a basis for $\mathcal{V}_h$. Then for every $m \geq 3$, $0 < \epsilon < 1$ and $K \geq 1$, there is a mapping*

$$\mathcal{M} : \mathcal{U}^m \times \mathbb{C}^{m \times K} \to \mathbb{C}^{N \times K},$$

where $N = \Theta(n, d)$ is as in (3.5) with $n = \lceil m/L \rceil$ and $L = L(m, d, \epsilon)$ as in (3.8), with the following property. Let $f \in \mathcal{B}(\boldsymbol{\rho})$ for arbitrary $\boldsymbol{\rho} > \mathbf{1}$, draw $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m$ randomly and independently according to ϱ and let $(d_{ik})_{i,k=1}^{m,K} \in \mathbb{C}^{m \times K}$ be as in (3.1) for arbitrary noise terms $\boldsymbol{n} = (n_i)_{i=1}^m \in \mathcal{V}$. Let $(\hat{c}_{jk}) = \mathcal{M}((\boldsymbol{y}_i), (d_{ik}))$ and define the approximation $\hat{f}$ as in (3.2) based on the index set $\Lambda = \Lambda^{\mathsf{HC}}_{n,d}$. Then the following holds with probability at least $1 - \epsilon$. The error satisfies

$$\|f - \hat{f}\|_{L^2_{\varrho}(\mathcal{U};\mathcal{V})} \leq c_1 \cdot \zeta, \quad \|f - \hat{f}\|_{L^\infty(\mathcal{U};\mathcal{V})} \leq c_2 \cdot \sqrt{\frac{m}{L}} \cdot \zeta, \tag{3.9}$$

for any $0 < p \leq 1$, where

$$\zeta := C \cdot \left(\frac{m}{c_0 L}\right)^{1/2-1/p} + \frac{\|\boldsymbol{n}\|_{2;\mathcal{V}}}{\sqrt{m}} + \|f - \mathcal{P}_h(f)\|_{L^\infty(\mathcal{U};\mathcal{V})}, \tag{3.10}$$

$c_0, c_1, c_2 \geq 1$ are universal constants and $C = C(d, p, \boldsymbol{\rho})$ depends on d, p and $\boldsymbol{\rho}$ only.

We now make several remarks about this result. The same remarks apply (with obvious modifications) to all subsequent results as well. First, notice how the index set Λ in which the approximation is constructed is given completely explicitly in terms of m, d and ϵ. Thus, as claimed in Remark 3.2, no 'oracle' information about the function being approximated is required. Indeed, notice that the mapping described in this theorem is *universal* in the sense that its applies equally to *any* function $f \in \mathcal{B}(\boldsymbol{\rho})$ and *any* $\boldsymbol{\rho} > \mathbf{1}$.

A key aspect of this theorem is the factor ζ, defined in (3.10), which determines the error bounds (3.9). As claimed in Section 1.4, this incorporates three main key errors arising in the approximation process.

(i) *The approximation error.* This is the algebraically decaying term in ζ: namely, $E_{\mathsf{app}} = C \cdot (m/(c_0 L))^{1/2-1/p}$. It is completely equivalent to the best s-term approximation error bound in Theorem 2.4, except with s replaced by $m/(c_0 L)$.

(ii) *The sampling error.* This is the term $E_{\mathsf{samp}} = \|\boldsymbol{n}\|_{2;\mathcal{V}}/\sqrt{m}$, where $\boldsymbol{n} = (n_i)_{i=1}^m$ is as in (3.1). In other words, the effect of any errors in computing the sample values $f(\boldsymbol{y}_i)$ enters linearly in the overall error bound.

(iii) *The physical discretization error.* This is the term

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

It describes the effect of working in the finite-dimensional subspace $\mathcal{V}_h$, instead of the full space $\mathcal{V}$. Critically, it depends on the orthogonal projection (best approximation) $\mathcal{P}_h(f)$ of f from $\mathcal{V}_h$.

Notice that (i) also describes the *sample complexity* of the scheme. Indeed, Theorem 3.4 asserts that there is a polynomial approximation that can be obtained from m samples that attains the best s-term rate $s^{1/2-1/p}$, where $s = m/(c_0 L)$ scales like m up to the polylogarithmic factor L.

Theorem 3.4 asserts the existence of a mapping that takes samples values as its input and produces the coefficients of a polynomial approximation attaining a desired error bound as its output. The mapping, as we see later, arises as a minimizer of a certain weighted ℓ^1-minimization problem. Thus, it is not an algorithm in the sense of Definition 3.1. In the next two theorems we assert the existence of algorithms that attain the same error, plus additional algorithmic error terms.

Theorem 3.5 (Existence of an algorithm; algebraic case, finite dimensions). *Consider the setup of Theorem* 3.4*. Then, for every $t \geq 1$, there exists an algorithm*

$$\mathcal{A}_t : \mathcal{U}^m \times \mathbb{C}^{m\times K} \to \mathbb{C}^{N\times K},$$

in the sense of Definition 3.1 *such that the same property holds, except with* (3.9) *replaced by*

$$\|f - \hat{f}\|_{L^2_\varrho(\mathcal{U};\mathcal{V})} \leq c_1 \cdot \left(\zeta + \frac{1}{t}\right),$$

$$\|f - \hat{f}\|_{L^\infty(\mathcal{U};\mathcal{V})} \leq c_2 \cdot \sqrt{\frac{m}{L}} \cdot \left(\zeta + \frac{1}{t}\right),$$

where $c_1, c_2 \geq 1$ are as in (3.9) *and ζ is as in* (3.10)*. The computational cost of the algorithm is bounded by*

$$c_3 \cdot [m \cdot \Theta(n,d) \cdot d + t \cdot (m \cdot \Theta(n,d) \cdot K + (\Theta(n,d) + m) \cdot (F(\boldsymbol{G}) + K)) \cdot (\Theta(n,d))^\alpha], \tag{3.11}$$

where $n = \lceil m/L \rceil$ is as in Theorem 3.4*, $\Theta(n,d)$ is as in* (3.5)*, α is as in* (3.7) *and $c_3 > 0$ is a universal constant.*

The key element of this theorem is that the same error bound as in Theorem 3.4 is attained, up to an additional term. In particular, we have the three sources of errors (i)–(iii), plus the following:

(iv) *The algorithmic error.* This is the error $E_{\mathsf{alg}} = 1/t$ committed by the algorithm $\mathcal{A}_t$ in approximately computing the output of the mapping $\mathcal{M}$ in Theorem 3.4. It is given in terms of the parameter t, which also enters linearly into the computational cost estimate (3.11).

Unfortunately, the $1/t$ decay rate of the algorithmic error is slow. Hence, it may be computationally expensive to compute an approximation to within a desired error bound. Fortunately, as we now explain, it is possible to improve it to e^{-t} subject to an additional technical assumption.

Theorem 3.6 (Existence of an efficient algorithm; algebraic case, finite dimensions). *Consider the setup of Theorem 3.4. Then for every $t \geq 1$ and $\zeta' > 0$ there exists an algorithm*

$$\mathcal{A}_{t,\zeta'} : \mathcal{U}^m \times \mathbb{C}^{m \times K} \to \mathbb{C}^{N \times K},$$

in the sense of Definition 3.1 such that the same property holds whenever $\zeta' \geq \zeta$, except with (3.9) replaced by

$$\begin{aligned} \|f - \hat{f}\|_{L^2_\varrho(\mathcal{U};\mathcal{V})} &\leq c_1 \cdot (\zeta + \zeta' + \mathrm{e}^{-t}), \\ \|f - \hat{f}\|_{L^\infty(\mathcal{U};\mathcal{V})} &\leq c_2 \cdot \sqrt{\frac{m}{L}} \cdot (\zeta + \zeta' + \mathrm{e}^{-t}), \end{aligned} \tag{3.12}$$

where $c_1, c_2 \geq 1$ are as in (3.9) and ζ is as in (3.10). The computational cost of the algorithm is bounded by

$$c_3 \cdot [m \cdot \Theta(n,d) \cdot d + t \cdot (m \cdot \Theta(n,d) \cdot K + (\Theta(n,d) + m) \cdot (F(\boldsymbol{G}) + K)) \cdot (\Theta(n,d))^\alpha],$$

where $n = \lceil m/L \rceil$ is as in Theorem 3.4, $\Theta(n,d)$ is as in (3.5), α is as in (3.7) and $c_3 > 0$ is a universal constant.

We refer to this as an "efficient" algorithm, since the parameter t enters linearly in the computational cost but the algorithmic error scales like e^{-t}. The main limitation of this result is that the algorithm parameter ζ' needs to be an upper bound for the true error bound ζ in order for (3.12) to hold. This is a technical assumption for the proof, and does not appear necessary in practice. We demonstrate this phenomenon through numerical experiment in Chapter 5.

3.3.2 Algebraic rates of convergence, infinite dimensions

We now consider algebraic rates of convergence in the infinite-dimensional setting. The next three results should be compared against the corresponding best s-term approximation result, Theorem 2.5.

Theorem 3.7 (Existence of a mapping; algebraic case, infinite dimensions). *Let $d = \infty$, $\{\Psi_{\boldsymbol{\nu}}\}_{\boldsymbol{\nu} \in \mathbb{N}_0^d} \subset L^2_\varrho(\mathcal{U})$ be either the orthonormal Chebyshev or Legendre basis and $\{\varphi_k\}_{k=1}^K$ be a basis for $\mathcal{V}_h$. Then for every $m \geq 3$, $0 < \epsilon < 1$ and $K \geq 1$, there is a mapping*

$$\mathcal{M} : \mathcal{U}^m \times \mathbb{C}^{m \times K} \to \mathbb{C}^{N \times K},$$

where $N = \Theta(n, d)$ is as in (3.5) with $n = \lceil m/L \rceil$, where $L = L(m, d, \epsilon)$ is as in (3.8), with the following property. Let $\varepsilon > 0$, $0 < p < 1$ and $\boldsymbol{b} \in \ell^p(\mathbb{N})$, $\boldsymbol{b} > \boldsymbol{0}$, be monotonically nonincreasing. Let $f \in \mathcal{B}(\boldsymbol{b}, \varepsilon)$, draw $\boldsymbol{y}_1, \dots, \boldsymbol{y}_m$ randomly and independently according to ϱ and let $(d_{ik})_{i,k=1}^{m,K} \in \mathbb{C}^{m \times K}$ be as in (3.1) for arbitrary noise terms $\boldsymbol{n} = (n_i)_{i=1}^m \in \mathcal{V}$. Let $(\hat{c}_{jk}) = \mathcal{M}((\boldsymbol{y}_i), (d_{ik}))$ and define the approximation $\hat{f}$ as in (3.2) based on the index set $\Lambda = \Lambda_n^{\mathsf{HCI}}$. Then the following holds with probability at least $1 - \epsilon$. The error satisfies

$$\|f - \hat{f}\|_{L^2_\varrho(\mathcal{U};\mathcal{V})} \le c_1 \cdot \zeta, \quad \|f - \hat{f}\|_{L^\infty(\mathcal{U};\mathcal{V})} \le c_2 \cdot \sqrt{\frac{m}{L}} \cdot \zeta, \tag{3.13}$$

where

$$\zeta := C \cdot \left(\frac{m}{c_0 L}\right)^{1/2-1/p} + \frac{\|\boldsymbol{n}\|_{2;\mathcal{V}}}{\sqrt{m}} + \|f - \mathcal{P}_h(f)\|_{L^\infty(\mathcal{U};\mathcal{V})}, \tag{3.14}$$

$c_0, c_1, c_2 \ge 1$ are universal constants and $C = C(\boldsymbol{b}, \varepsilon, p)$ depends on $\boldsymbol{b}$, ε and p only.

Theorem 3.8 (Existence of an algorithm; algebraic case, infinite dimensions). *Consider the setup of Theorem 3.7. Then, for every $t \ge 1$, there exists an algorithm*

$$\mathcal{A}_t : \mathcal{U}^m \times \mathbb{C}^{m \times K} \to \mathbb{C}^{N \times K},$$

in the sense of Definition 3.1 such that the same property holds, except with (3.13) replaced by

$$\|f - \hat{f}\|_{L^2_\varrho(\mathcal{U};\mathcal{V})} \le c_1 \cdot \left(\zeta + \frac{1}{t}\right),$$
$$\|f - \hat{f}\|_{L^\infty(\mathcal{U};\mathcal{V})} \le c_2 \cdot \sqrt{\frac{m}{L}} \cdot \left(\zeta + \frac{1}{t}\right),$$

where $c_1, c_2 \ge 1$ are as in (3.13) and ζ is as in (3.14). The computational cost of the algorithm is bounded by

$$c_3 \cdot [m \cdot \Theta(n, \infty) \cdot n + t \cdot (m \cdot \Theta(n, \infty) \cdot K + (\Theta(n, \infty) + m) \cdot (F(\boldsymbol{G}) + K)) \cdot (\Theta(n, \infty))^\alpha],$$

where $n = \lceil m/L \rceil$ is as in Theorem 3.7, $\Theta(n, \infty)$ is as in (3.5), α is as in (3.7) and $c_3 > 0$ is a universal constant.

In finite dimensions, the computational cost estimate (3.11) is somewhat difficult to interpret, since its behaviour depends on the relative sizes of m and d. Fortunately, in infinite dimensions we can give a more informative assessment. Suppose, for simplicity, that K is fixed (for example, $K = 1$ in the case of a scalar-valued function approximation problem). Then the computational cost is bounded by

$$c \cdot m \cdot \Theta(n, \infty) \cdot n + c_K \cdot t \cdot m \cdot \Theta(n, \infty)^{\alpha+1},$$

where $c > 0$ is a universal constant $c_K > 0$ is a constant depending on K only. Recall from (3.5) that $\Theta(n, \infty) = |\Lambda_n^{\mathsf{HCI}}| = |\Lambda_{n,n}^{\mathsf{HC}}|$. Now, when $d = n$ and n is sufficiently large, the minimum in (3.6) is attained by the second term $\mathrm{e}n^{2+\log(n)/\log(2)}$. Substituting this into the above expression and recalling that $n = \lceil m/L \rceil$, where $L = L(m, \infty, \epsilon)$, we deduce that the computational cost is bounded by

$$c_K \cdot t \cdot m \cdot g(m)^{(\alpha+1)\log(4g(m))/\log(2)}, \quad g(m) := \left\lceil \frac{m}{\log(m) \cdot (\log^3(m) + \log(\epsilon^{-1}))} \right\rceil .$$

Since $m \geq 3$ by assumption, we have $\log(m) \geq 1$ and therefore $g(m) \leq m$. Hence, this admits the slightly looser upper bound

$$c_K \cdot t \cdot m^{1+(\alpha+1)\log(4m)/\log(2)}.$$

We deduce that the computational cost (for fixed K and t) is *subexponential* in m. Further, if we choose $t = m^{1/p-1/2}$ in accordance with the algebraically decaying term in (3.14), then we conclude the following: *it is possible to approximate a holomorphic function of infinitely many variables with error decaying algebraically fast in m via an algorithm whose computational cost is subexponential in m*. Whether this can be reduced to an algebraic cost is an open problem.

Theorem 3.9 (Existence of an efficient algorithm; algebraic case, infinite dimensions). *Consider the setup of Theorem 3.7. Then, for every $t \geq 1$ and $\zeta' > 0$ there exists an algorithm*

$$\mathcal{A}_{t,\zeta'} : \mathcal{U}^m \times \mathbb{C}^{m \times K} \to \mathbb{C}^{N \times K},$$

in the sense of Definition 3.1 such that the same property holds whenever $\zeta' \geq \zeta$, except with (3.13) replaced by

$$\|f - \hat{f}\|_{L^2_\varrho(\mathcal{U};\mathcal{V})} \leq c_1 \cdot (\zeta + \zeta' + \mathrm{e}^{-t}),$$

$$\|f - \hat{f}\|_{L^\infty(\mathcal{U};\mathcal{V})} \leq c_2 \cdot \sqrt{\frac{m}{L}} \cdot (\zeta + \zeta' + \mathrm{e}^{-t}),$$

where $c_1, c_2 \geq 1$ are as in (3.13) and $\zeta \leq \zeta'$ is as in (3.14). The computational cost of the algorithm is bounded by

$$c_3 \cdot [m \cdot \Theta(n,\infty) \cdot n + t \cdot (m \cdot \Theta(n,\infty) \cdot K + (\Theta(n,\infty) + m) \cdot (F(\boldsymbol{G}) + K)) \cdot (\Theta(n,\infty))^\alpha],$$

where $n = \lceil m/L \rceil$ is as in Theorem 3.7, $\Theta(n,\infty)$ is as in (3.5), α is as in (3.7) and $c_3 > 0$ is a universal constant.

3.3.3 Exponential rates of convergence, finite dimensions

Finally, we consider exponential rates of convergence in finite dimensions. The following results should be compared against Theorem 2.6.

Theorem 3.10 (Existence of a mapping; exponential case, finite dimensions). *Let $d \in \mathbb{N}$, $\{\Psi_{\boldsymbol{\nu}}\}_{\boldsymbol{\nu} \in \mathbb{N}_0^d} \subset L^2_\varrho(\mathcal{U})$ be either the orthonormal Chebyshev or Legendre basis and $\{\varphi_k\}_{k=1}^K$ be a basis for $\mathcal{V}_h$. Then for every $m \geq 3$, $0 < \epsilon < 1$ and $K \geq 1$, there is a mapping*

$$\mathcal{M} : \mathcal{U}^m \times \mathbb{C}^{m \times K} \to \mathbb{C}^{N \times K},$$

where $N = \Theta(n, d)$ is as in (3.5) with

$$n = \begin{cases} \lceil \sqrt{m/L} \rceil & \text{Legendre}, \\ \lceil m/(2^d L) \rceil & \text{Chebyshev}, \end{cases} \tag{3.15}$$

and L as in (3.8), with the following property. Draw $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m$ randomly and independently according to ϱ. Then, with probability at least $1 - \epsilon$, the following holds. Let $f \in \mathcal{B}(\boldsymbol{\rho})$ for arbitrary $\boldsymbol{\rho} > \mathbf{1}$, $(d_{ik})_{i,k=1}^{m,K} \in \mathbb{C}^{m \times K}$ be as in (3.1) for arbitrary noise terms $\boldsymbol{n} = (n_i)_{i=1}^m \in \mathcal{V}$, $(\hat{c}_{jk})_{j,k=1}^{N,K} = \mathcal{M}((\boldsymbol{y}_i)_{i=1}^m, (d_{ik})_{i,k=1}^{m,k})$ and define the approximation $\hat{f}$ as in (3.2) based on the index set $\Lambda = \Lambda^{\mathsf{HC}}_{n,d}$. Then the error satisfies

$$\|f - \hat{f}\|_{L^2_\varrho(\mathcal{U};\mathcal{V})} \leq c_1 \cdot \zeta, \quad \|f - \hat{f}\|_{L^\infty(\mathcal{U};\mathcal{V})} \leq c_2 \cdot \sqrt{\frac{m}{L}} \cdot \zeta, \tag{3.16}$$

for any

$$0 < \gamma < (d+1)^{-1} \left(d! \prod_{j=1}^{d} \log(\rho_j) \right)^{1/d},$$

where

$$\zeta := C \cdot \begin{cases} \exp\left(-\frac{\gamma}{2} \left(\frac{m}{c_0 L}\right)^{\frac{1}{d}}\right) & \text{Chebyshev} \\ \exp\left(-\gamma \left(\frac{m}{c_0 L}\right)^{\frac{1}{2d}}\right) & \text{Legendre} \end{cases} + \frac{\|\boldsymbol{n}\|_{2;\mathcal{V}}}{\sqrt{m}} + \|f - \mathcal{P}_h(f)\|_{L^\infty(\mathcal{U};\mathcal{V})}, \tag{3.17}$$

$c_0, c_1, c_2 \geq 1$ are universal constants and $C = C(d, \gamma, \boldsymbol{\rho})$ depends on d, γ and $\boldsymbol{\rho}$ only.

Theorem 3.11 (Existence of an algorithm; exponential case, finite dimensions). *Consider the setup of Theorem 3.10. Then, for every $t \geq 1$, there exists an algorithm*

$$\mathcal{A}_t : \mathcal{U}^m \times \mathbb{C}^{m \times K} \to \mathbb{C}^{N \times K},$$

in the sense of Definition 3.1 such that the same property holds, except with (3.16) replaced by

$$\|f - \hat{f}\|_{L^2_\varrho(\mathcal{U};\mathcal{V})} \leq c_1 \cdot \left(\zeta + \frac{1}{t}\right),$$

$$\|f - \hat{f}\|_{L^\infty(\mathcal{U};\mathcal{V})} \leq c_2 \cdot \sqrt{\frac{m}{L}} \cdot \left(\zeta + \frac{1}{t}\right),$$

where $c_1, c_2 \geq 1$ are as in (3.16) and ζ is as in (3.17). The computational cost of the algorithm is bounded by

$$c_3 \cdot [m \cdot \Theta(n,d) \cdot n + t \cdot (m \cdot \Theta(n,d) \cdot K + (\Theta(n,d) + m) \cdot (F(\boldsymbol{G}) + K)) \cdot (\Theta(n,d))^{\alpha}],$$

where n is as in (3.15), $\Theta(n,d)$ is as in (3.5), α is as in (3.7) and $c_3 > 0$ is a universal constant.

Theorem 3.12 (Existence of an efficient algorithm; exponential case, finite dimensions). *Consider the setup of Theorem 3.10. Suppose that there is a known upper bound $\zeta' \geq \zeta$, where ζ is as in (3.17). Then, for every $t \geq 1$ and $\zeta' > 0$ there exists an algorithm*

$$\mathcal{A}_{t,\zeta'} : \mathcal{U}^m \times \mathbb{C}^{m \times K} \to \mathbb{C}^{N \times K},$$

in the sense of Definition 3.1 for which the same property holds whenever $\zeta' \geq \zeta$, except with (3.16) replaced by

$$\|f - \hat{f}\|_{L^2_\varrho(\mathcal{U};\mathcal{V})} \leq c_1 \cdot (\zeta + \zeta' + \mathrm{e}^{-t}),$$
$$\|f - \hat{f}\|_{L^\infty(\mathcal{U};\mathcal{V})} \leq c_2 \cdot \sqrt{\frac{m}{L}}(\zeta + \zeta' + \mathrm{e}^{-t}),$$

where $c_1, c_2 \geq 1$ are as in (3.16) and ζ is as in (3.17). The computational cost of the algorithm is bounded by

$$c_3 \cdot [m \cdot \Theta(n,d) \cdot n + t \cdot (m \cdot \Theta(n,d) \cdot K + (\Theta(n,d) + m) \cdot (F(\boldsymbol{G}) + K)) \cdot (\Theta(n,d))^{\alpha}],$$

where n is as in (3.15), $\Theta(n,d)$ is as in (3.5), α is as in (3.7) and $c_3 > 0$ is a universal constant.

As before, suppose that K is fixed and, since we consider exponential rates, that d is also fixed. Then, using the third estimate in (3.6), we deduce that the computational cost of this algorithm is bounded by

$$c_{K,d} \cdot \left(m \cdot n^2 \cdot (\log(n))^{d-1} + t \cdot m \cdot \left(n \cdot (\log(n))^{d-1}\right)^{\alpha+1}\right).$$

Using the crude bound $n \leq m$, we obtain the bound

$$c_{K,d} \cdot \left(t \cdot m^{\alpha+2} (\log(m))^{(d-1)(\alpha+1)}\right).$$

Thus, for fixed t, the computational cost is polynomial in m as $m \to \infty$. In particular, with the efficient algorithm of Theorem 3.12 (subject to the caveat that an upper bound for the error is known) we deduce the following: *in fixed dimension d, it is possible to approximate a holomorphic function with error decaying exponentially fast in m via an algorithm whose computational cost is polynomial in m.* Whether the polynomial growth rate described above is sharp is an open problem.

Remark 3.13. There is a subtle difference between the algebraic and exponential results. The former are *nonuniform* in the sense that a single draw of the sample points $\boldsymbol{y}_1, \dots, \boldsymbol{y}_m$ is sufficient for recovery of a fixed function f with high probability up to the specified error bound. The latter are *uniform*, since a single draw of the sample points $\boldsymbol{y}_1, \dots, \boldsymbol{y}_m$ is sufficient for recovery of any function with high probability up to the specified error bound. The reason for this difference stems from bounding a *discrete* error term (8.10), which is a random variable depending on f and the sample points. In the algebraic case, in order to obtain the desired algebraic exponent $1/2 - 1/p$ we bound this term with high probability for each fixed f. See Step 4 of the proof of Theorem 8.2. This renders the ensuing result nonuniform. Conversely, in the exponential case (where the appearance of small algebraic factors is not a concern, since they can be absorbed into the exponentially decaying term) we bound this term with probability one for *any* f. See Step 4 of the proof of Theorem 8.4. Note that one could also derive uniform guarantees in the algebraic case by considering a fixed value of p and letting $\mathcal{M}$ and $\mathcal{A}$ depend on p, or by considering a restricted range $0 < p \leq p^* < 1$. Both strategies involve a larger value of n, with its size depending on p or p^*. See [8, Section 7.6.2] for further discussion.