Johann Radon Institute for
Computational and Applied Mathematics
Austrian Academy of Sciences (ÖAW)

Adaptive approximation for multivariate linear problems with inputs lying in a cone

Y. Ding, F.J. Hickernell, P. Kritzer, S. Mak

RICAM-Report 2019-13

Adaptive Approximation for Multivariate Linear Problems with Inputs Lying in a Cone

Yuhan Ding, Fred J. Hickernell, Peter Kritzer, Simon Mak

March 25, 2019

Abstract

We study adaptive approximation algorithms for general multivariate linear problems where the sets of input functions are non-convex cones. While it is known that adaptive algorithms perform essentially no better than non-adaptive algorithms for convex input sets, the situation may be different for non-convex sets. A typical example considered here is function approximation based on series expansions. Given an error tolerance, we use series coefficients of the input to construct an approximate solution such that the error does not exceed this tolerance. We study the situation where we can bound the norm of the input based on a pilot sample, and the situation where we keep track of the decay rate of the series coefficients of the input. Moreover, we consider situations where it makes sense to infer coordinate and smoothness importance. Besides performing an error analysis, we also study the information cost of our algorithms and the computational complexity of our problems, and we identify conditions under which we can avoid a curse of dimensionality.

1 Introduction

In many situations, adaptive algorithms can be rigorously shown to perform *essentially no better* than non-adaptive algorithms. Yet, in practice adaptive algorithms are appreciated because they relieve the user from stipulating the computational effort required to achieve the desired accuracy. The key to resolving this seeming contradiction is to construct a theory based on assumptions that favor adaptive algorithms. We do that here.

Adaptive algorithms infer the necessary computational effort based on the function data sampled. Adaptive algorithms may perform better than non-adaptive algorithms if the set of input functions is *non-convex*. We construct adaptive algorithms for general multivariate linear problems where the input functions lie in non-convex cones. Our algorithms use a finite number of series coefficients of the input function to construct an approximate solution that satisfies an absolute error tolerance. We show our algorithms to be essentially optimal. We derive conditions under which the problem is tractable, i.e., the information cost of constructing the approximate solution does not increase exponentially with the dimension of the input function domain. In the remainder of this section we define the problem and essential notation. But first, we present a helpful example.

1.1 An Illustrative Example

Consider the case of approximating functions defined over $[-1,1]^d$, using a Chebyshev polynomial basis. The input function is denoted f, and the solution is $\mathrm{SOL}(f) = f$. In this case,

$$f = \sum_{\boldsymbol{k} \in \mathbb{N}_0^d} \widehat{f}(\boldsymbol{k}) u_{\boldsymbol{k}} =: \mathrm{SOL}(f), \qquad \boldsymbol{k} = (k_1, \ldots, k_d) \in \mathbb{N}_0^d,$$

$$u_{\boldsymbol{k}} := \prod_{\ell=1}^{d} \tilde{u}_{k_\ell}, \qquad \tilde{u}_k(x) := \cos(k \cos^{-1}(x)) \quad \forall k \in \mathbb{N}_0.$$

Approximating f well by a finite sum requires knowing which terms in the infinite series for f are more important. Let $\mathcal{F}$ denote a Hilbert space of input functions where the norm of $\mathcal{F}$ is a $\boldsymbol{\lambda}$-weighted norm of the series coefficients:

$$\|f\|_{\mathcal{F}} := \left\| \left(\frac{\widehat{f}(\boldsymbol{k})}{\lambda_{\boldsymbol{k}}} \right)_{\boldsymbol{k} \in \mathbb{N}_0^d} \right\|_2, \qquad \text{where } \boldsymbol{\lambda} = \left(\lambda_{\boldsymbol{k}}\right)_{\boldsymbol{k} \in \mathbb{N}_0^d}, \quad \lambda_{\boldsymbol{k}} := \prod_{\substack{\ell=1 \\ k_\ell > 0}}^{d} \frac{w_\ell}{k_\ell^r}, \quad r > 0.$$

The w_ℓ are non-negative *coordinate weights*, which embody the assumption that f may depend more strongly on coordinates with larger w_ℓ than those with smaller w_ℓ. The definition of the $\mathcal{F}$-norm implies that an input function must have series coefficients that decay quickly enough as the degree of the polynomial increases. Larger r implies smoother input functions.

The ordering of the weights,

$$\lambda_{\boldsymbol{k}_1} \geq \lambda_{\boldsymbol{k}_2} \geq \cdots > 0, \tag{1}$$

implies an ordering of the wavenumbers, $\boldsymbol{k}$. It is natural to approximate the solution using the first n series coefficients as follows:

$$\text{APP}(f, n) := \sum_{i=1}^{n} \widehat{f}(\boldsymbol{k}_i) u_{\boldsymbol{k}_i} \qquad \forall f \in \mathcal{F},\ n \in \mathbb{N}.$$

Here, we assume that it is possible to sample the series coefficients of the input function. This is a less restrictive assumption than being able to sample any linear functional, but it is more restrictive than only being able to sample function values. An important future problem is to extend the theory in this chapter to the case where the only function data available are function values.

The error of this approximation in terms of the norm on the output space, $\mathcal{G}$, can be expressed as

$$\|\text{SOL}(f) - \text{APP}(f, n)\|_{\mathcal{G}} = \left\| \left(\widehat{f}(\boldsymbol{k}_i)\right)_{i=n+1}^{\infty} \right\|_2,$$

$$\text{where} \qquad \left\| \sum_{\boldsymbol{k} \in \mathbb{N}_0^d} \widehat{g}(\boldsymbol{k}) u_{\boldsymbol{k}} \right\|_{\mathcal{G}} := \left\| \left(\widehat{g}(\boldsymbol{k})\right)_{\boldsymbol{k} \in \mathbb{N}_0^d} \right\|_2.$$

If one has a fixed data budget, n, then $\text{APP}(f, n)$ is the best answer.

However, our goal is an algorithm, $\text{ALG}(f, \varepsilon)$ that satisfies the error criterion

$$\|\text{SOL}(f) - \text{ALG}(f, \varepsilon)\|_{\mathcal{G}} \leq \varepsilon \qquad \forall \varepsilon > 0,\ f \in \mathcal{C}, \tag{2}$$

where ε is the error tolerance, and $\mathcal{C} \subset \mathcal{F}$ is the set of input functions for which ALG is successful. This algorithm contains a rule for choosing n—depending on f and ε—so that $\text{ALG}(f, \varepsilon) = \text{APP}(f, n)$. The objectives of this chapter are to

- construct such a rule,
- choose a set $\mathcal{C}$ of input functions for which the rule is valid,
- characterize the information cost of ALG,
- determine whether ALG has optimal information cost, and

- understand the dependence of this cost on the number of input variables, d, as well as the error tolerance, ε.

We return to this example in Section 1.6 to discuss the answers to some of these questions. We perform some numerical experiments for this example in Section 4.3.

1.2 General Linear Problem

Now, we define our problem more generally. A solution operator maps the input function to an output, $\mathrm{SOL} : \mathcal{F} \to \mathcal{G}$. As in the illustrative example above, the Banach spaces of inputs and outputs are defined by series expansions:

$$\mathcal{F} := \left\{ f = \sum_{\boldsymbol{k} \in \mathbb{K}} \widehat{f}(\boldsymbol{k}) u_{\boldsymbol{k}} : \|f\|_{\mathcal{F}} := \left\| \left(\frac{\widehat{f}(\boldsymbol{k})}{\lambda_{\boldsymbol{k}}} \right)_{\boldsymbol{k} \in \mathbb{K}} \right\|_{\rho} < \infty \right\}, \quad 1 \le \rho \le \infty,$$

$$\mathcal{G} := \left\{ g = \sum_{\boldsymbol{k} \in \mathbb{K}} \widehat{g}(\boldsymbol{k}) v_{\boldsymbol{k}} : \|g\|_{\mathcal{G}} := \left\| \left(\widehat{g}(\boldsymbol{k}) \right)_{\boldsymbol{k} \in \mathbb{K}} \right\|_{\tau} < \infty \right\}, \quad 1 \le \tau \le \rho.$$

Here, $\{u_{\boldsymbol{k}}\}_{\boldsymbol{k} \in \mathbb{K}}$ is a basis for the input Banach space $\mathcal{F}$, $\{v_{\boldsymbol{k}}\}_{\boldsymbol{k} \in \mathbb{K}}$ is a basis for the output Banach space $\mathcal{G}$, $\mathbb{K}$ is a countable index set, and $\boldsymbol{\lambda} = (\lambda_{\boldsymbol{k}})_{\boldsymbol{k} \in \mathbb{K}}$ is the sequence of weights. These bases are defined to match the solution operator:

$$\mathrm{SOL}(u_{\boldsymbol{k}}) = v_{\boldsymbol{k}} \qquad \forall \boldsymbol{k} \in \mathbb{K}. \tag{3}$$

The $\lambda_{\boldsymbol{k}}$ represent the importance of the series coefficients of the input function. The larger $\lambda_{\boldsymbol{k}}$ is, the more important $\widehat{f}(\boldsymbol{k})$ is.

Although this problem formulation is quite general in some aspects, condition (3) is somewhat restrictive. In principle, the choice of basis can be made via the singular value decomposition, but in practice, if the norms of $\mathcal{F}$ and $\mathcal{G}$ are specified without reference to their respective bases, it may be difficult to identify bases satisfying (3).

To facilitate our derivations below, we establish the following lemma via Hölder's inequality:

Lemma 1 *Let $\mathcal{K}$ be some proper or improper subset of the index set $\mathbb{K}$. Moreover, let ρ' be defined by the relation*

$$\frac{1}{\rho} + \frac{1}{\rho'} = \frac{1}{\tau}, \qquad \textit{i.e., } \rho' := \frac{\rho\tau}{\rho - \tau},$$

so $\tau \le \rho' \le \infty$. Let $\Lambda := \left\| \left(\lambda_{\boldsymbol{k}}\right)_{\boldsymbol{k} \in \mathcal{K}} \right\|_{\rho'}$ be the norm of a subset of the weights. Then the following are true for $f = \sum_{\boldsymbol{k} \in \mathcal{K}} \widehat{f}(\boldsymbol{k}) u_{\boldsymbol{k}}$:

$$\|\mathrm{SOL}(f)\|_{\mathcal{G}} = \left\| \left(\widehat{f}(\boldsymbol{k}) \right)_{\boldsymbol{k} \in \mathcal{K}} \right\|_{\tau} \le \|f\|_{\mathcal{F}} \, \Lambda, \tag{4}$$

$$\left| \widehat{f}(\boldsymbol{k}) \right| = \begin{cases} \dfrac{R \lambda_{\boldsymbol{k}}^{\rho'/\rho + 1}}{\Lambda^{\rho'/\rho}}, & \forall \boldsymbol{k} \in \mathcal{K}, \quad \textit{if } \rho' < \infty, \\ R \Lambda \delta_{\boldsymbol{k}, \boldsymbol{k}^*}, & \forall \boldsymbol{k} \in \mathcal{K},\ \boldsymbol{k}^* \in \mathcal{K} \textit{ satisfies } \lambda_{\boldsymbol{k}^*} = \Lambda, \quad \textit{if } \rho' = \infty, \end{cases}$$

$$\implies \quad \|f\|_{\mathcal{F}} = R \textit{ and } \|\mathrm{SOL}(f)\|_{\mathcal{G}} = R\Lambda. \tag{5}$$

Equality (5) *illustrates how inequality* (4) *may be made tight.*

Proof. We give the proof for $\rho' < \infty$. The proof for $\rho' = \infty$ follows similarly. The proof of inequality (4) proceeds by applying Hölder's inequality:

$$\begin{aligned}\|\mathrm{SOL}(f)\|_{\mathcal{G}} &= \left\|\sum_{\boldsymbol{k}\in\mathcal{K}} \widehat{f}(\boldsymbol{k}) v_{\boldsymbol{k}}\right\|_{\mathcal{G}} = \left\|\left(\widehat{f}(\boldsymbol{k})\right)_{\boldsymbol{k}\in\mathcal{K}}\right\|_{\tau} = \left[\sum_{\boldsymbol{k}\in\mathcal{K}} \left|\frac{\widehat{f}(\boldsymbol{k})}{\lambda_{\boldsymbol{k}}}\right|^{\tau} \lambda_{\boldsymbol{k}}^{\tau}\right]^{1/\tau} \\ &\le \left\|\left(\frac{\widehat{f}(\boldsymbol{k})}{\lambda_{\boldsymbol{k}}}\right)_{\boldsymbol{k}\in\mathcal{K}}\right\|_{\rho} \left\|\left(\lambda_{\boldsymbol{k}}\right)_{\boldsymbol{k}\in\mathcal{K}}\right\|_{\rho'} = \|f\|_{\mathcal{F}}\, \Lambda \qquad \text{since } \frac{1}{\rho} + \frac{1}{\rho'} = \frac{1}{\tau}.\end{aligned} \tag{6}$$

Substituting the formula for $\left|\widehat{f}(\boldsymbol{k})\right|$ in (5) into equation (6) and applying the relationship between ρ, ρ', and τ yields

$$\|\mathrm{SOL}(f)\|_{\mathcal{G}} = \frac{R\left\|\left(\lambda_{\boldsymbol{k}}^{\rho'/\rho+1}\right)_{\boldsymbol{k}\in\mathcal{K}}\right\|_{\tau}}{\Lambda^{\rho'/\rho}} = \frac{R\left\|\left(\lambda_{\boldsymbol{k}}\right)_{\boldsymbol{k}\in\mathcal{K}}\right\|_{\rho'}^{\rho'/\rho+1}}{\Lambda^{\rho'/\rho}} = R\Lambda.$$

Moreover,

$$\|f\|_{\mathcal{F}} = \left\|\left(\frac{\widehat{f}(\boldsymbol{k})}{\lambda_{\boldsymbol{k}}}\right)_{\boldsymbol{k}\in\mathcal{K}}\right\|_{\rho} = \frac{R\left\|\left(\lambda_{\boldsymbol{k}}^{\rho'/\rho}\right)_{\boldsymbol{k}\in\mathcal{K}}\right\|_{\rho}}{\Lambda^{\rho'/\rho}} = \frac{R\left\|\left(\lambda_{\boldsymbol{k}}\right)_{\boldsymbol{k}\in\mathcal{K}}\right\|_{\rho'}^{\rho'/\rho}}{\Lambda^{\rho'/\rho}} = R.$$

This completes the proof. □

Taking $\mathcal{K} = \mathbb{K}$ in the lemma above, the norm of the solution operator can be expressed in terms of the norm of $\boldsymbol{\lambda}$ as follows:

$$\|\mathrm{SOL}\|_{\mathcal{F}\to\mathcal{G}} = \sup_{\|f\|_{\mathcal{F}}\le 1} \|\mathrm{SOL}(f)\|_{\mathcal{G}} = \left\|\boldsymbol{\lambda}\right\|_{\rho'}. \tag{7}$$

We assume throughout this chapter that the weights are chosen to keep this norm is finite, namely,

$$\left\|\boldsymbol{\lambda}\right\|_{\rho'} < \infty. \tag{8}$$

As in Section 1.1, here in the general case the $\lambda_{\boldsymbol{k}}$ are assumed to have a known order as was specified in (1). We also assume that all $\lambda_{\boldsymbol{k}}$ are positive to avoid the trivial case where $\mathrm{SOL}(f)$ can be expressed exactly as a finite sum for all $f \in \mathcal{F}$.

1.3 An Approximation and an Algorithm

The optimal approximation based on n series coefficients of the input function is defined in terms of the series coefficients of the input function corresponding to the largest $\lambda_{\boldsymbol{k}}$ as follows:

$$\mathrm{APP} : \mathcal{F} \times \mathbb{N}_0 \to \mathcal{G}, \quad \mathrm{APP}(f,0) = 0, \quad \mathrm{APP}(f,n) := \sum_{i=1}^{n} \widehat{f}(\boldsymbol{k}_i) v_{\boldsymbol{k}_i} \ \forall n \in \mathbb{N}. \tag{9}$$

By the argument leading to (6) it follows that

$$\|\mathrm{SOL}(f) - \mathrm{APP}(f,n)\|_{\mathcal{G}} = \left\|\left(\widehat{f}(\boldsymbol{k}_i)\right)_{i=n+1}^{\infty}\right\|_{\tau}. \tag{10}$$

An upper bound on the approximation error follows from Lemma 1:

$$\|\mathrm{SOL}(f) - \mathrm{APP}(f,n)\|_{\mathcal{G}} \le \left\|\left(\frac{\widehat{f}(\boldsymbol{k}_i)}{\lambda_{\boldsymbol{k}_i}}\right)_{i=n+1}^{\infty}\right\|_{\rho} \left\|\left(\lambda_{\boldsymbol{k}_i}\right)_{i=n+1}^{\infty}\right\|_{\rho'}. \tag{11}$$

This leads to the following theorem.

Theorem 1 *Let $\mathcal{B}_R := \{f \in \mathcal{F} : \|f\|_{\mathcal{F}} \leq R\}$ denote the ball of radius R in the space of input functions. The error of the approximation defined in* (9) *is bounded tightly above as*

$$\sup_{f \in \mathcal{B}_R} \|\mathrm{SOL}(f) - \mathrm{APP}(f,n)\|_{\mathcal{G}} \leq R \left\| \left(\lambda_{\boldsymbol{k}_i}\right)_{i=n+1}^{\infty} \right\|_{\rho'}. \tag{12}$$

Moreover, the worst case error over $\mathcal{B}_R$ of $\mathrm{APP}'(\cdot,n)$, for any approximation based on n series coefficients of the input function, can be no smaller.

Proof. The proof of (12) follows immediately from (11) and Lemma 1. The optimality of APP follows by bounding the error of an arbitrary approximation, APP′, applied to functions that mimic the zero function.

Let $\mathrm{APP}'(0,n)$ depend on the series coefficients indexed by $\mathcal{J} = \{\boldsymbol{k}_1', \ldots, \boldsymbol{k}_n'\}$. Use Lemma 1 with $\mathcal{K} = \mathbb{K} \setminus \mathcal{J}$ to choose f to mimic the zero function, have norm R, and have as large a solution as possible, i.e.,

$$\begin{aligned} &\widehat{f}(\boldsymbol{k}_1') = \cdots = \widehat{f}(\boldsymbol{k}_n') = 0, \qquad \|f\|_{\mathcal{F}} = R, \\ &\|\mathrm{SOL}(f)\|_{\mathcal{G}} = R \left\| \left(\lambda_{\boldsymbol{k}}\right)_{\boldsymbol{k} \notin \mathcal{J}} \right\|_{\rho'} \qquad \text{by (5)}. \end{aligned} \tag{13}$$

Then $\mathrm{APP}'(\pm f, n) = \mathrm{APP}'(0,n)$ because f mimics the zero function, and

$$\begin{aligned} &\sup_{f \in \mathcal{B}_R} \|\mathrm{SOL}(f) - \mathrm{APP}(f,n)\|_{\mathcal{G}} \\ &\quad \geq \max_{\pm} \left\|\mathrm{SOL}(\pm f) - \mathrm{APP}'(\pm f, n)\right\|_{\mathcal{G}} = \max_{\pm} \left\|\mathrm{SOL}(\pm f) - \mathrm{APP}'(0,n)\right\|_{\mathcal{G}} \\ &\quad \geq \frac{1}{2}\left[\left\|\mathrm{SOL}(f) - \mathrm{APP}'(0,n)\right\|_{\mathcal{G}} + \left\|-\mathrm{SOL}(f) - \mathrm{APP}'(0,n)\right\|_{\mathcal{G}}\right] \\ &\quad \geq \|\mathrm{SOL}(f)\|_{\mathcal{G}} = R \left\| \left(\lambda_{\boldsymbol{k}}\right)_{\boldsymbol{k} \notin \mathcal{J}} \right\|_{\rho'} \qquad \text{by (13)}. \end{aligned}$$

The ordering of the $\lambda_{\boldsymbol{k}}$ implies that $\left\| \left(\lambda_{\boldsymbol{k}}\right)_{\boldsymbol{k} \notin \mathcal{J}} \right\|_{\rho'}$ for arbitrary $\mathcal{J}$ can be no smaller than the case $\mathcal{J} = \{\boldsymbol{k}_1, \ldots, \boldsymbol{k}_n\}$. This completes the proof. □

While approximation APP is a key piece of the puzzle, our ultimate goal is an algorithm, $\mathrm{ALG} : \mathcal{C} \times [0,\infty)$, satisfying the absolute error criterion (2). The non-adaptive Algorithm 1 satisfies this error criterion for $\mathcal{C} = \mathcal{B}_R$.

Algorithm 1 Non-Adaptive ALG for a Ball of Input Functions

Parameters: the Banach spaces $\mathcal{F}$ and $\mathcal{G}$, including the weights $\boldsymbol{\lambda}$; the ball radius, R; APP satisfying (12)
Input: a black-box function, f; an absolute error tolerance, $\varepsilon > 0$
Ensure: Error criterion (2) for $\mathcal{C} = \mathcal{B}_R$
  Choose $n^* = \min\left\{n \in \mathbb{N}_0 : \left\| \left(\lambda_{\boldsymbol{k}_i}\right)_{i=n+1}^{\infty} \right\|_{\rho'} \leq \varepsilon/R\right\}$
  Return $\mathrm{ALG}(f,\varepsilon) = \mathrm{APP}(f, n^*)$

After defining the information cost of an algorithm and the problem complexity in the next subsection, we demonstrate that this non-adaptive algorithm is optimal when the set of inputs is chosen to be $\mathcal{C} = \mathcal{B}_R$. However, typically one cannot bound the norm of the input function a priori, so Algorithm 1 is impractical.

The key difficulty is that error bound (12) depends on the norm of the input function. In contrast, we will construct error bounds for $\mathrm{APP}(f,n)$ that only depend on function data. These

will lead to *adaptive* algorithms ALG satisfying error criterion (2). For such algorithms, the set of allowable input functions, $\mathcal{C}$, will be a *cone*, not a ball.

Note that algorithms satisfying error criterion (2) cannot exist for $\mathcal{C} = \mathcal{F}$. Any algorithm must require a finite sample size, even if it is huge. Then, there must exist some $f \in \mathcal{F}$ that looks exactly like the zero function to the algorithm but for which $\|\mathrm{SOL}(f)\|_{\mathcal{G}}$ is arbitrarily large. Thus, algorithms satisfying the error criterion exist only for some strict subset of $\mathcal{F}$. Choosing that subset well is both an art and a science.

1.4 Information Cost and Problem Complexity

The information cost of $\mathrm{ALG}(f, \varepsilon)$ is denoted $\mathrm{COST}(\mathrm{ALG}, f, \varepsilon)$ and defined as the number of function data—in our situation, series coefficients—required by $\mathrm{ALG}(f, \varepsilon)$. For adaptive algorithms this cost varies with the input function f. We also define the information cost of the algorithm in general, recognizing that it will tend to depend on $\|f\|_{\mathcal{F}}$:

$$\mathrm{COST}(\mathrm{ALG}, \mathcal{C}, \varepsilon, R) := \max_{f \in \mathcal{C} \cap \mathcal{B}_R} \mathrm{COST}(\mathrm{ALG}, f, \varepsilon).$$

Note that while the cost depends on $\|f\|_{\mathcal{F}}$, $\mathrm{ALG}(f, \varepsilon)$ has no knowledge of f beyond the fact that it lies in $\mathcal{C}$. It is common for $\mathrm{COST}(\mathrm{ALG}, \mathcal{C}, \varepsilon, R)$ to be $\mathcal{O}(\varepsilon^{-p})$, or perhaps asymptotically $c \log(1 + \varepsilon^{-1})$.

Let $\mathcal{A}(\mathcal{C})$ denote the set of all possible algorithms that may be constructed using series coefficients and that *satisfy error criterion* (2). We define the *computational complexity* of a problem as the information cost of the best algorithm:

$$\mathrm{COMP}(\mathcal{A}(\mathcal{C}), \varepsilon, R) := \min_{\mathrm{ALG} \in \mathcal{A}(\mathcal{C})} \mathrm{COST}(\mathrm{ALG}, \mathcal{C}, \varepsilon, R).$$

These definitions follow the information-based complexity literature [12, 11]. We define an algorithm to be *essentially optimal* if there exist some fixed positive ω, $\varepsilon_{\max}$, and $R_{\min}$ for which

$$\mathrm{COST}(\mathrm{ALG}, \mathcal{C}, \varepsilon, R) \leq \mathrm{COMP}(\mathcal{A}(\mathcal{C}), \omega\varepsilon, R) \qquad \forall\, \varepsilon \in (0, \varepsilon_{\max}],\ R \in [R_{\min}, \infty). \quad (14)$$

If the complexity of the problem is $\mathcal{O}(\varepsilon^{-p})$, the cost of an essentially optimal algorithm is also $\mathcal{O}(\varepsilon^{-p})$. If the complexity of the problem is asymptotically $c \log(1 + \varepsilon^{-1})$, then the cost of an essentially optimal algorithm is also asymptotically $c \log(1 + \varepsilon^{-1})$. We will show that our adaptive algorithms presented in Sections 2 and 3 are essentially optimal.

Theorem 2 *The non-adaptive Algorithm 1 has an information cost for the set of input functions $\mathcal{C} = \mathcal{B}_R$ that is given by*

$$\mathrm{COST}(\mathrm{ALG}, \mathcal{B}_R, \varepsilon, R') = \min \left\{ n \in \mathbb{N}_0 : \left\| \left(\lambda_{\boldsymbol{k}_i}\right)_{i=n+1}^{\infty} \right\|_{\rho'} \leq \varepsilon / R \right\}.$$

This algorithm is essentially optimal for the set of input functions $\mathcal{B}_R$, namely,

$$\mathrm{COST}(\mathrm{ALG}, \mathcal{B}_R, \varepsilon, R') \leq \mathrm{COMP}(\mathcal{A}(\mathcal{B}_R), \omega\varepsilon, R') \qquad \forall\, \varepsilon \in (0, \varepsilon_{\max}],\ R \in [R_{\min}, \infty),$$

where $\varepsilon_{\max}$ and $R_{\min}$ are arbitrary and fixed, and $\omega = R_{\min}/R$.

Proof. Fix positive $\varepsilon_{\max}$, $R_{\min}$, R, and ω as defined above. For $0 < \varepsilon \leq \varepsilon_{\max}$ and $R_{\min} \leq R' \leq R$, the information cost of non-adaptive Algorithm 1 follows from its definition. Let

$$n^*(\varepsilon, R) := \mathrm{COST}(\mathrm{ALG}, \mathcal{B}_R, \varepsilon, R').$$

Construct an input function $f \in \mathcal{B}_{R'}$ as in the proof of Theorem 1 with $\mathcal{J} = \{\boldsymbol{k}_1, \ldots, \boldsymbol{k}_{n^*(\omega\varepsilon, R')}\}$. By the argument in the proof of Theorem 1, any algorithm in $\mathcal{A}(\mathcal{B}_{R'})$ that can approximate $\mathrm{SOL}(f)$ with an error no greater than $\omega\varepsilon$ must use at least $n^*(\omega\varepsilon, R')$ series coefficients. Thus,

$$\begin{aligned}
\mathrm{COST}(\mathrm{ALG}, \mathcal{B}_R, \varepsilon, R') &= n^*(\varepsilon, R) = n^*(\varepsilon R'/R, R') \\
&\leq n^*(\omega\varepsilon, R') \qquad \text{since } R'/R \geq \omega \\
&\leq \mathrm{COMP}(\mathcal{A}(\mathcal{B}_{R'}), \omega\varepsilon, R') \leq \mathrm{COMP}(\mathcal{A}(\mathcal{B}_R), \omega\varepsilon, R').
\end{aligned}$$

Thus, Algorithm 1 is essentially optimal. □

For Algorithm 1, the information cost, $\mathrm{COST}(\mathrm{ALG}, \mathcal{B}_R, \varepsilon, R)$, depends on the decay rate of the tail norm of the $\lambda_{\boldsymbol{k}_i}$. This decay may be algebraic or exponential and also determines the problem complexity, $\mathrm{COMP}(\mathcal{A}(\mathcal{B}_R), \varepsilon, R)$, as a function of the error tolerance, ε.

This theorem illustrates how an essentially optimal algorithm for solving a problem for a ball of input functions, $\mathcal{C} = \mathcal{B}_R$, can be non-adaptive. However, as alluded to above, we claim that it is impractical to know a priori which ball your input function lies in. On the other hand, in the situations described below where $\mathcal{C}$ is a cone, we will show that $\mathcal{A}(\mathcal{C})$ actually contains only adaptive algorithms via the lemma below. The proof of this lemma follows directly from the definition of non-adaptivity.

Lemma 2 *For a given set of input functions, $\mathcal{C}$, if $\mathcal{A}(\mathcal{C})$ contains any non-adaptive algorithms, then for every $\varepsilon > 0$,*

$$\mathrm{COMP}(\mathcal{A}(\mathcal{C}), \varepsilon) := \sup_{R>0} \mathrm{COMP}(\mathcal{A}(\mathcal{C}), \varepsilon, R) < \infty.$$

1.5 Tractability

Besides understanding the dependence of $\mathrm{COMP}(\mathcal{A}(\mathcal{C}), \varepsilon, R)$ on ε, we also want to understand how $\mathrm{COMP}(\mathcal{A}(\mathcal{C}), \varepsilon, R)$ depends on the dimension of the domain of the input function. Suppose that $f : \Omega^d \to \mathbb{R}$, for some $\Omega \subseteq \mathbb{R}$, and let $\mathcal{F}_d$ denote the dependence of the input space on the dimension d. The set of functions for which our algorithms succeed, $\mathcal{C}_d$, depends on the dimension, too. Also, SOL, APP, COST, and COMP depend implicitly on dimension, and this dependence is sometimes indicated explicitly by the subscript d.

Different dependencies of $\mathrm{COMP}(\mathcal{A}(\mathcal{C}_d), \varepsilon, R)$ on the dimension d and the error tolerance ε are formalized as different notions of tractability. Since the complexity is defined in terms of the best available algorithm, tractability is a property that is inherent to the problem, not to a particular algorithm. We define the following notions of tractability (for further information on tractability we refer to the trilogy [8], [9], [10]). Note that in contrast to these references we explicitly include the dependence on R in our definitions. This dependence is natural for cones $\mathcal{C}$ and might be different if $\mathcal{C}$ is not a cone.

- We say that the adaptive approximation problem is strongly polynomially tractable if and only if there are non-negative C, p, $\varepsilon_{\max}$, and $R_{\min}$ such that

$$\mathrm{COMP}(\mathcal{A}(\mathcal{C}_d), \varepsilon, R) \leq C\, R^p\, \varepsilon^{-p} \qquad \forall d \in \mathbb{N},\ \varepsilon \in (0, \varepsilon_{\max}],\ R \in [R_{\min}, \infty).$$

The infimum of p satisfying the bound above is denoted by p^* and is called the exponent of strong polynomial tractability.

- We say that the problem is polynomially tractable if and only if there are non-negative $C, p,\ q,\ \varepsilon_{\max}$, and $R_{\min}$ such that

$$\mathrm{COMP}(\mathcal{A}(\mathcal{C}_d), \varepsilon, R) \le C\, d^{\,q}\, R^p\, \varepsilon^{-p} \qquad \forall d \in \mathbb{N},\ \varepsilon \in (0, \varepsilon_{\max}],\ R \in [R_{\min}, \infty).$$

- We say that the problem is weakly tractable iff

$$\lim_{d+R\,\varepsilon^{-1}\to\infty} \frac{\log\ \mathrm{COMP}(\mathcal{A}(\mathcal{C}_d), \varepsilon, R)}{d + R\,\varepsilon^{-1}} \;=\; 0.$$

Necessary and sufficient conditions on these tractability notions will be studied for different types of algorithms in Sections 2.2 and 3.3.

We remark that, for the sake of brevity, we focus here on tractability notions that are summarized as algebraic tractability in the recent literature (see, e.g., [6]). Theoretically, one could also study exponential tractability, where one would essentially replace ε^{-1} by $\log(1+\varepsilon^{-1})$ in the previous tractability notions. A more detailed study of tractability will be done in a future paper.

1.6 The Illustrative Example Revisited

The example in Section 1.1 chooses $\rho = \tau = 2$ and $\rho' = \infty$. Thus, we obtain by Theorem 2:

$$\mathrm{COMP}(\mathcal{A}(\mathcal{B}_R), \varepsilon, R) = \mathrm{COST}(\mathrm{ALG}, \mathcal{B}_R, \varepsilon, R) = \min\{n \in \mathbb{N}_0 : \lambda_{\boldsymbol{k}_{n+1}} \le \varepsilon/R\}.$$

Using the non-increasing ordering of the $\lambda_{\boldsymbol{k}_i}$, we employ a standard technique for bounding the $n+1^{\text{st}}$ largest $\lambda_{\boldsymbol{k}}$ in terms of the sum of the p^{th} power of all the $\lambda_{\boldsymbol{k}}$. For $0 < 1/r < p$,

$$\begin{aligned}
(n+1)\lambda_{\boldsymbol{k}_{n+1}}^p &\le \sum_{i=1}^{n+1} \lambda_{\boldsymbol{k}_i}^p \le \sum_{\boldsymbol{k}\in\mathbb{N}_0^d} \lambda_{\boldsymbol{k}}^p = \prod_{\ell=1}^{d} \left[1 + w_\ell^p \sum_{k=1}^{\infty} \frac{1}{k^{pr}}\right] \\
&= \prod_{\ell=1}^{d} \left[1 + w_\ell^p \zeta(pr)\right] = \exp\left(\sum_{\ell=1}^{d} \log\bigl(1 + w_\ell^p \zeta(pr)\bigr)\right) \\
&\le \exp\left(\zeta(pr) \sum_{\ell=1}^{\infty} w_\ell^p\right) \qquad \text{since } \log(1+x) \le x\ \forall x \ge 0.
\end{aligned}$$

Hence, substituting the above upper bound on $\lambda_{\boldsymbol{k}_{n+1}}$ into the formula for the complexity of the problem, we obtain an upper bound on the complexity:

$$\begin{aligned}
&\mathrm{COMP}(\mathcal{A}(\mathcal{B}_R), \varepsilon, R) \\
&\quad\le \min\left\{n \in \mathbb{N}_0 : \frac{1}{n+1} \exp\left(\zeta(pr) \sum_{\ell=1}^{\infty} w_\ell^p\right) \le \left(\frac{\varepsilon}{R}\right)^p\right\} \\
&\quad= \left\lceil \left(\frac{R}{\varepsilon}\right)^p \exp\left(\zeta(pr) \sum_{\ell=1}^{\infty} w_\ell^p\right)\right\rceil - 1.
\end{aligned}$$

If $p^\dagger$ is the infimum of the p for which $\sum_{\ell=1}^{\infty} w_\ell^p$ is finite, and $p^\dagger$ is finite, then we obtain strong polynomial tractability and an exponent of strong tractability that is $p^* = \max(1/r, p^\dagger)$. On the other hand, if the coordinate weights are all unity, $w_1 = w_2 = \cdots = 1$, then there are 2^d different $\lambda_{\boldsymbol{k}}$ with a value of 1, and so $\mathrm{COMP}(\mathcal{A}(\mathcal{B}_R), \varepsilon, R) \ge 2^d$, and the problem is not tractable.

1.7 What Comes Next

In the following section we define a cone of input functions, $\mathcal{C}$, in (16) whose norms can be bounded above in terms of the series coefficients obtained from a pilot sample. Adaptive Algorithm 2 is shown to be optimal for this $\mathcal{C}$. We also identify necessary and sufficient conditions for tractability.

Section 3 considers the situation where function data is relatively inexpensive, and we track the decay rate of the series coefficients. Adaptive Algorithm 3 is shown to be optimal in this situation.

Section 4 considers the case where the most suitable weights $\boldsymbol{\lambda}$ are not known a priori and are instead inferred from function data. Adaptive Algorithm 4 combines this inference step with Algorithm 2 to construct an approximation that satisfies the error criterion.

2 Bounding the Norm of the Input Function Based on a Pilot Sample

2.1 The Cone and the Optimal Algorithm

The premise of an adaptive algorithm is that the finite information we observe about the input function tells us something about what is not observed. Let n_1 denote the number of pilot observations, based on the set of wavenumbers

$$\mathcal{K}_1 := \{\boldsymbol{k}_1, \ldots, \boldsymbol{k}_{n_1}\}, \tag{15}$$

where the $\boldsymbol{k}_i$ are defined by the ordering of the $\lambda_{\boldsymbol{k}}$ in (1). Let A be some constant inflation factor greater than one. The cone of functions whose norm can be bounded well in terms of a pilot sample, $\{\widehat{f}(\boldsymbol{k}_1), \ldots, \widehat{f}(\boldsymbol{k}_{n_1})\}$, is given by

$$\mathcal{C} = \left\{ f \in \mathcal{F} : \|f\|_{\mathcal{F}} \le A \left\| \left(\frac{\widehat{f}(\boldsymbol{k})}{\lambda_{\boldsymbol{k}}} \right)_{\boldsymbol{k} \in \mathcal{K}_1} \right\|_{\rho} \right\}. \tag{16}$$

Referring to error bound (11), we see that the error of $\mathrm{APP}(f, n)$ depends on the series coefficients not sampled. The definition of $\mathcal{C}$ allows us to bound these as follows:

$$\begin{aligned} \left\| \left(\frac{\widehat{f}(\boldsymbol{k}_i)}{\lambda_{\boldsymbol{k}_i}} \right)_{i=n+1}^{\infty} \right\|_{\rho} &= \left[\|f\|_{\mathcal{F}}^{\rho} - \left\| \left(\frac{\widehat{f}(\boldsymbol{k}_i)}{\lambda_{\boldsymbol{k}_i}} \right)_{i=1}^{n} \right\|_{\rho}^{\rho} \right]^{1/\rho} \qquad \forall f \in \mathcal{F} \\ &\le \left[A^{\rho} \left\| \left(\frac{\widehat{f}(\boldsymbol{k})}{\lambda_{\boldsymbol{k}}} \right)_{\boldsymbol{k} \in \mathcal{K}_1} \right\|_{\rho}^{\rho} - \left\| \left(\frac{\widehat{f}(\boldsymbol{k}_i)}{\lambda_{\boldsymbol{k}_i}} \right)_{i=1}^{n} \right\|_{\rho}^{\rho} \right]^{1/\rho} \qquad \forall f \in \mathcal{C}. \end{aligned}$$

This inequality together with error bound (11) implies the data-based error bound

$$\|\mathrm{SOL}(f) - \mathrm{APP}(f, n)\|_{\mathcal{G}} \le \mathrm{ERR}\big(\big(\widehat{f}(\boldsymbol{k}_i)\big)_{i=1}^{n}, n\big) \qquad \forall f \in \mathcal{C}, \tag{17a}$$

where

$$\begin{aligned} &\mathrm{ERR}\big(\big(\widehat{f}(\boldsymbol{k}_i)\big)_{i=1}^{n}, n\big) \\ &\qquad := \left[A^{\rho} \left\| \left(\frac{\widehat{f}(\boldsymbol{k})}{\lambda_{\boldsymbol{k}}} \right)_{\boldsymbol{k} \in \mathcal{K}_1} \right\|_{\rho}^{\rho} - \left\| \left(\frac{\widehat{f}(\boldsymbol{k}_i)}{\lambda_{\boldsymbol{k}_i}} \right)_{i=1}^{n} \right\|_{\rho}^{\rho} \right]^{1/\rho} \big\| (\lambda_{\boldsymbol{k}_i})_{i=n+1}^{\infty} \big\|_{\rho'}, \end{aligned}$$

Algorithm 2 ALG Based on a Pilot Sample

Parameters: the Banach spaces $\mathcal{F}$ and $\mathcal{G}$, including the weights $\boldsymbol{\lambda}$; an initial sample size, $n_1 \in \mathbb{N}$; an inflation factor, $A > 1$; APP satisfying (11)
Input: a black-box function, f; an absolute error tolerance, $\varepsilon > 0$
Ensure: Error criterion (2) for the cone defined in (16)
Let $n \leftarrow n_1 - 1$
repeat
Let $n \leftarrow n + 1$
Compute $\mathrm{ERR}\bigl(\bigl(\widehat{f}(\boldsymbol{k}_i)\bigr)_{i=1}^n, n\bigr)$ as defined in (17)
until $\mathrm{ERR}\bigl(\bigl(\widehat{f}(\boldsymbol{k}_i)\bigr)_{i=1}^n, n\bigr) \le \varepsilon$
Return $\mathrm{ALG}(f, \varepsilon) = \mathrm{APP}(f, n)$

$$n \ge n_1. \quad (17b)$$

This error bound decays as $\bigl\|\bigl(f(\boldsymbol{k}_i)/\lambda_{\boldsymbol{k}_i}\bigr)_{i=1}^n\bigr\|_\rho$ increases and as the tail norm of the $\lambda_{\boldsymbol{k}_i}$ decreases. This data-driven error bound underlies Algorithm 2, which is successful for $\mathcal{C}$ defined in (16):

Theorem 3 *Algorithm 2 yields an answer satisfying absolute error criterion* (2)*, i.e.,* $\mathrm{ALG} \in \mathcal{A}(\mathcal{C})$ *for* $\mathcal{C}$ *defined in* (16)*. The information cost is*

$$\mathrm{COST}(\mathrm{ALG}, \mathcal{C}, \varepsilon, R) = \min\Bigl\{n \ge n_1 : \bigl\|\bigl(\lambda_{\boldsymbol{k}_i}\bigr)_{i=n+1}^\infty\bigr\|_{\rho'} \le \varepsilon/[(A^\rho - 1)^{1/\rho} R]\Bigr\}. \quad (18)$$

There exist positive $\varepsilon_{\max}$ *and* $R_{\min}$ *for which the computational complexity has the lower bound*

$$\mathrm{COMP}(\mathcal{A}(\mathcal{C}), \varepsilon, R) \ge \min\Bigl\{n \ge n_1 : \bigl\|\bigl(\lambda_{\boldsymbol{k}_i}\bigr)_{i=n+1}^\infty\bigr\|_{\rho'} \le 2\varepsilon/[(1 - 1/A)R]\Bigr\} \quad \forall \varepsilon \in (0, \varepsilon_{\max}],\ R \in [R_{\min}, \infty). \quad (19)$$

Algorithm 2 is essentially optimal. Moreover, $\mathcal{A}(\mathcal{C})$ *contains only adaptive algorithms.*

Proof. The upper bound on the computational cost of this algorithm is obtained by noting that

$$\begin{aligned}
&\mathrm{COST}(\mathrm{ALG}, \mathcal{C}, \varepsilon, R)\\
&= \max_{f \in \mathcal{C} \cap \mathcal{B}_R} \min\Bigl\{n \ge n_1 : \mathrm{ERR}\bigl(\bigl(\widehat{f}(\boldsymbol{k}_i)\bigr)_{i=1}^n, n\bigr) \le \varepsilon\Bigr\}\\
&\le \max_{f \in \mathcal{C} \cap \mathcal{B}_R} \min\left\{n \ge n_1 : (A^\rho - 1)^{1/\rho} \left\|\left(\frac{\widehat{f}(\boldsymbol{k})}{\lambda_{\boldsymbol{k}}}\right)_{\boldsymbol{k} \in \mathcal{K}_1}\right\|_\rho \bigl\|\bigl(\lambda_{\boldsymbol{k}_i}\bigr)_{i=n+1}^\infty\bigr\|_{\rho'} \le \varepsilon\right\}\\
&\le \min\Bigl\{n \ge n_1 : (A^\rho - 1)^{1/\rho} R \bigl\|\bigl(\lambda_{\boldsymbol{k}_i}\bigr)_{i=n+1}^\infty\bigr\|_{\rho'} \le \varepsilon\Bigr\},
\end{aligned}$$

since $\bigl\|\bigl(\widehat{f}(\boldsymbol{k})/\lambda_{\boldsymbol{k}}\bigr)_{\boldsymbol{k} \in \mathcal{K}_1}\bigr\|_\rho \le \bigl\|\bigl(\widehat{f}(\boldsymbol{k}_i)/\lambda_{\boldsymbol{k}_i}\bigr)_{i=1}^n\bigr\|_\rho \le \|f\|_{\mathcal{F}} \le R$ for all $f \in \mathcal{B}_R$, $n \ge n_1$. Moreover, this inequality is tight for some $f \in \mathcal{C} \cap \mathcal{B}_R$, namely, those certain f for which $\widehat{f}(\boldsymbol{k}_i) = 0$ for $i > n_1$. This completes the proof of (18).

To prove the lower complexity bound, choose $\varepsilon_{\max}$ and $R_{\min}$ such that

$$\bigl\|\bigl(\lambda_{\boldsymbol{k}_i}\bigr)_{i=n_1+1}^\infty\bigr\|_{\rho'} > 2\varepsilon_{\max}/[(1 - 1/A)R_{\min}].$$

Let ALG' be any algorithm that satisfies the error criterion, (2), for this choice of $\mathcal{C}$ in (16). Fix $R \in [R_{\min}, \infty)$ and $\varepsilon \in (0, \varepsilon_{\max}]$ arbitrarily. Two fooling functions will be constructed of the form $f_\pm = f_1 \pm f_2$.

The input function f_1 is defined via its series coefficients as in Lemma 1, having nonzero coefficients only for $\boldsymbol{k} \in \mathcal{K}_1$:

$$\left|\widehat{f}_1(\boldsymbol{k})\right| = \begin{cases} \dfrac{R(1+1/A)\lambda_{\boldsymbol{k}}^{\rho'/\rho+1}}{2\left\|\left(\lambda_{\boldsymbol{k}}\right)_{\boldsymbol{k}\in\mathcal{K}_1}\right\|_{\rho'}^{\rho'/\rho}}, & \boldsymbol{k} \in \mathcal{K}_1, \\ 0, & \boldsymbol{k} \notin \mathcal{K}_1, \end{cases} \qquad \|f_1\|_{\mathcal{F}} = \frac{R(1+1/A)}{2}.$$

Suppose that $\mathrm{ALG}'(f_1, \varepsilon)$ samples the series coefficients $\widehat{f}_1(\boldsymbol{k})$ for $\boldsymbol{k} \in \mathcal{J}$, and let n denote the cardinality of $\mathcal{J}$.

Now, construct the input function f_2, having zero coefficients for $\boldsymbol{k} \in \mathcal{J}$ and also as in Lemma 1:

$$\left|\widehat{f}_2(\boldsymbol{k})\right| = \begin{cases} \dfrac{R(1-1/A)\lambda_{\boldsymbol{k}}^{\rho'/\rho+1}}{2\left\|\left(\lambda_{\boldsymbol{k}}\right)_{\boldsymbol{k}\notin\mathcal{J}}\right\|_{\rho'}^{\rho'/\rho}}, & \boldsymbol{k} \notin \mathcal{J}, \\ 0, & \boldsymbol{k} \in \mathcal{J}, \end{cases} \qquad \|f_2\|_{\mathcal{F}} = \frac{R(1-1/A)}{2},$$

$$\|\mathrm{SOL}(f_2)\|_{\mathcal{G}} = \frac{R(1-1/A)}{2}\left\|\left(\lambda_{\boldsymbol{k}}\right)_{\boldsymbol{k}\notin\mathcal{J}}\right\|_{\rho'}. \tag{20}$$

Let $f_{\pm} = f_1 \pm f_2$. By the definitions above, it follows that

$$\begin{aligned} \|f_{\pm}\|_{\mathcal{F}} = \|f_1 \pm f_2\|_{\mathcal{F}} &\le \|f_1\|_{\mathcal{F}} + \|f_2\|_{\mathcal{F}} = R, \\ \left\|\left(\frac{\widehat{f}_{\pm}(\boldsymbol{k}_i)}{\lambda_{\boldsymbol{k}_i}}\right)_{i=1}^{n_1}\right\|_{\rho} &= \left\|\left(\frac{\widehat{f}_1(\boldsymbol{k}_i) \pm \widehat{f}_2(\boldsymbol{k}_i)}{\lambda_{\boldsymbol{k}_i}}\right)_{i=1}^{n_1}\right\|_{\rho} \\ &\ge \left\|\left(\frac{\widehat{f}_1(\boldsymbol{k}_i)}{\lambda_{\boldsymbol{k}_i}}\right)_{i=1}^{n_1}\right\|_{\rho} - \left\|\left(\frac{\widehat{f}_2(\boldsymbol{k}_i)}{\lambda_{\boldsymbol{k}_i}}\right)_{i=1}^{n_1}\right\|_{\rho} \\ &\ge \|f_1\|_{\mathcal{F}} - \|f_2\|_{\mathcal{F}} = \frac{R}{A} \ge \frac{\|f_{\pm}\|_{\mathcal{F}}}{A}. \end{aligned}$$

Therefore, $f_{\pm} \in \mathcal{C} \cap \mathcal{B}_R$. Moreover, since the series coefficients for $f_{\pm}$ are the same for $\boldsymbol{k} \in \mathcal{J}$, it follows that $\mathrm{ALG}'(f_+, \varepsilon) = \mathrm{ALG}'(f_-, \varepsilon)$. Thus, $\mathrm{SOL}(f_+)$ must be quite similar to $\mathrm{SOL}(f_-)$.

Using an argument like that in the proof of Theorem 1, it follows that

$$\begin{aligned} \varepsilon &\ge \max_{\pm} \left\|\mathrm{SOL}(f_{\pm}) - \mathrm{ALG}'(f_{\pm}, \varepsilon)\right\|_{\mathcal{G}} = \max_{\pm} \left\|\mathrm{SOL}(f_{\pm}) - \mathrm{ALG}'(f_{+}, \varepsilon)\right\|_{\mathcal{G}} \\ &\ge \frac{1}{2}\left[\left\|\mathrm{SOL}(f_+) - \mathrm{ALG}'(f_+, \varepsilon)\right\|_{\mathcal{G}} + \left\|\mathrm{SOL}(f_-) - \mathrm{ALG}'(f_+, \varepsilon)\right\|_{\mathcal{G}}\right] \\ &\ge \frac{1}{2}\|\mathrm{SOL}(f_+ - f_-)\|_{\mathcal{G}} = \|\mathrm{SOL}(f_2)\|_{\mathcal{G}} = \frac{R(1-1/A)}{2}\left\|\left(\lambda_{\boldsymbol{k}}\right)_{\boldsymbol{k}\notin\mathcal{J}}\right\|_{\rho'} \qquad \text{by (20)} \\ &\ge \frac{R(1-1/A)}{2}\left\|\left(\lambda_{\boldsymbol{k}_i}\right)_{i=n+1}^{\infty}\right\|_{\rho'}, \end{aligned}$$

by the ordering of the $\boldsymbol{k}$ in (1). By the choice of $R_{\min}$ and $\varepsilon_{\max}$ above, it follows that $n > n_1$. This inequality then implies lower complexity bound (19). Because $\lim_{R\to\infty} \mathrm{COMP}(\mathcal{A}(\mathcal{C}), \varepsilon, R) = \infty$ it follows from Lemma 2 that $\mathcal{A}(\mathcal{C})$ contains only adaptive algorithms.

The essential optimality of Algorithm 2 follows by observing that

$$\mathrm{COST}(\mathrm{ALG}, \mathcal{C}, \varepsilon, R) \le \mathrm{COMP}(\mathcal{A}(\mathcal{C}), \omega\varepsilon, R) \qquad \text{for } \omega = \frac{1-1/A}{2(A^{\rho}-1)^{1/\rho}}.$$

This satisfies definition (14). □

The above derivation assumes that $A > 1$. If $A = 1$, then our cone consists of functions whose series coefficients vanish for wavenumbers outside $\mathcal{K}_1$. The exact solution can be constructed using only the pilot sample. Our algorithm is then non-adaptive, but succeeds for input functions in the cone $\mathcal{C}$, which is an unbounded set.

We may not be able to guarantee that a particular f of interest lies in our cone, $\mathcal{C}$, but we may derive necessary conditions for f to lie in $\mathcal{C}$. The following proposition follows from the definition of $\mathcal{C}$ in (16) and the fact that the term on the left below underestimates $\|f\|_{\mathcal{F}}$.

Proposition 1 *If $f \in \mathcal{C}$, then*

$$\left\| \left(\frac{\widehat{f}(\boldsymbol{k}_i)}{\lambda_{\boldsymbol{k}_i}} \right)_{\boldsymbol{k}_i=1}^{n} \right\|_{\rho} \leq A \left\| \left(\frac{\widehat{f}(\boldsymbol{k})}{\lambda_{\boldsymbol{k}}} \right)_{\boldsymbol{k} \in \mathcal{K}_1} \right\|_{\rho} \qquad \forall n \in \mathbb{N}. \tag{21}$$

If condition (21) is violated in practice, then $f \notin \mathcal{C}$, and Algorithm 2 may output an incorrect answer. The remedy is to make $\mathcal{C}$ more inclusive by increasing the inflation factor, A, and/or the pilot sample size, n_1.

2.2 Tractability

In this section, we write $\mathcal{C}_d$ instead of $\mathcal{C}$, to stress the dependence on d, and for the same reason we write $\lambda_{d,\boldsymbol{k}_i}$ instead of $\lambda_{\boldsymbol{k}_i}$. Recall that we assume that $\lambda_{d,\boldsymbol{k}_1} \geq \lambda_{d,\boldsymbol{k}_2} \geq \cdots > 0$. Let

$$n(\delta, d) := \min \left\{ n \geq 0 : \left\| (\lambda_{d,\boldsymbol{k}_i})_{i=n+1}^{\infty} \right\|_{\rho'} \leq \delta \right\} \qquad \forall \delta > 0.$$

From Equations (18) and (19), we obtain that

$$\mathrm{COMP}(\mathcal{A}(\mathcal{C}_d), \omega_{\mathrm{lo}}\varepsilon, R) \leq n(\varepsilon/R, d) \leq \mathrm{COMP}(\mathcal{A}(\mathcal{C}_d), \omega_{\mathrm{hi}}\varepsilon, R) \qquad \forall \varepsilon \in (0, \varepsilon_{\max}],\ R \in [R_{\min}, \infty),$$

where the positive constants ω_{lo} and ω_{hi} depend on A, but not depend on d, ε, or R. From the equation above, it is clear that tractability depends on the behavior of $n(\varepsilon/R, d)$ as R/ε and d tend to infinity. We would like to study under which conditions we obtain the various tractability notions defined in Section 1.5.

To this end, we distinguish two cases, depending on whether ρ' is infinite or not. This distinction is useful because it allows us to relate the computational complexity of the algorithms considered in this chapter to the computational complexity of linear problems on certain function spaces considered in the classical literature on information-based complexity, as for example [8]. The case $\rho' = \infty$ corresponds to the worst-case setting, where one studies the worst performance of an algorithm over the unit ball of a space. The results in Theorem 4 below are indeed very similar to the results for the worst-case setting over balls of suitable function spaces. The case $\rho < \infty$ corresponds to the so-called average-case setting, where one considers the average performance over a function space equipped with a suitable measure. For both of these settings there exist tractability results that we will make use of here.

CASE 1: $\rho' = \infty$: If $\rho' = \infty$, we have, due to the monotonicity of the $\lambda_{d,\boldsymbol{k}_i}$,

$$n(\varepsilon/R, d) = \min \left\{ n \geq 0 \colon \lambda_{d,\boldsymbol{k}_{n+1}} \leq \varepsilon/R \right\}.$$

We then have the following theorem.

Theorem 4 *Using the same notation as above, the following statements hold for the case $\rho' = \infty$.*

1. *We have strong polynomial tractability if and only if there exist $\eta > 0$ and $i_0 \in \mathbb{N}$ such that*

$$\sup_{d\in\mathbb{N}} \sum_{i=i_0}^{\infty} \lambda_{d,\boldsymbol{k}_i}^{\eta} < \infty. \tag{22}$$

   Furthermore, the exponent of strong polynomial tractability is then equal to the infimum of those $\eta > 0$ for which (22) holds.

2. *We have polynomial tractability if and only if there exist $\eta_1, \eta_2 \geq 0$ and $\eta_3, K > 0$ such that*

$$\sup_{d\in\mathbb{N}} d^{-\eta_1} \sum_{i=\lceil Kd^{\eta_2}\rceil}^{\infty} \lambda_{d,\boldsymbol{k}_i}^{\eta_3} < \infty.$$

3. *We have weak tractability if and only if*

$$\sup_{d\in\mathbb{N}} \exp(-cd) \sum_{i=1}^{\infty} \exp\left(-c\left(\frac{1}{\lambda_{d,\boldsymbol{k}_i}}\right)\right) < \infty \quad \textit{for all} \quad c > 0. \tag{23}$$

Proof. Letting $\widetilde{\varepsilon} := \sqrt{\varepsilon/R}$, we see that $n(\varepsilon/R, d) = \min\left\{n \geq 0\colon \lambda_{d,\boldsymbol{k}_{n+1}} \leq \widetilde{\varepsilon}^2\right\}$. The latter expression is well studied in the context of tractability of linear problems in the worst-case setting defined on unit balls of certain spaces, and if and only if conditions on the $\lambda_{d,\boldsymbol{k}_i}$ for various tractability notions are known. These conditions can be found in [8, Chapter 5] for (strong) polynomial tractability and [13] for weak tractability.

Since, in this chapter, we consider $\min\left\{n \geq 0\colon \lambda_{d,\boldsymbol{k}_{n+1}} \leq \varepsilon/R\right\}$, and in [8] and [13] ε/R is replaced by the square of the error tolerance, there are slight differences between the results here and those in the aforementioned references; to be more precise, the exponent of strong polynomial tractability is η here, whereas it is 2η in [8], and $1/\lambda_{d,\boldsymbol{k}_i}$ in (23) corresponds to $1/\sqrt{\lambda_{d,\boldsymbol{k}_i}}$ in [13]. □

CASE 2: $\rho' < \infty$: In this case, letting $\widetilde{\varepsilon} := (\varepsilon/R)^{\rho'/2}$ and $\widetilde{\lambda}_{d,i} := \lambda_{d,\boldsymbol{k}_i}^{\rho'}$, we have

$$\begin{aligned} n(\varepsilon/R, d) &= \min\left\{n \geq 0\colon \sum_{i=n+1}^{\infty} \lambda_{d,\boldsymbol{k}_i}^{\rho'} \leq (\varepsilon/R)^{\rho'}\right\} \\ &= \min\left\{n \geq 0\colon \sum_{i=n+1}^{\infty} \widetilde{\lambda}_{d,i} \leq \widetilde{\varepsilon}^2\right\}. \end{aligned} \tag{24}$$

However, the latter expression corresponds exactly to the average-case tractability (with respect to the parameters $\widetilde{\lambda}_{d,i}$ and $\widetilde{\varepsilon}$) defined on certain spaces as studied in, e.g., [8]. This leads us to the following theorem.

Theorem 5 *Using the same notation as above, the following statements hold for the case $\rho' < \infty$.*

1. *We have strong polynomial tractability if and only if there exist $\eta \in (0, 1)$ and $i_0 \in \mathbb{N}$ such that*

$$\sup_{d\in\mathbb{N}} \sum_{i=i_0}^{\infty} \lambda_{d,\boldsymbol{k}_i}^{\rho'\eta} < \infty. \tag{25}$$

   Furthermore, the exponent of strong polynomial tractability is then

$$\inf\left\{\rho'\eta/(1-\eta)\colon \eta \textit{ satisfies } (25)\right\}.$$

2. *We have polynomial tractability if and only if there exist* $\eta_1, \eta_2 \geq 0$ *and* $\eta_3 \in (0,1), K > 0$ *such that*

$$\sup_{d\in\mathbb{N}} d^{-\eta_1} \sum_{i=\lceil K d^{\eta_2}\rceil}^{\infty} \lambda_{d,\boldsymbol{k}_i}^{\rho'\,\eta_3} < \infty.$$

3. *Let* $t_{d,i} := \sum_{k=i}^{\infty} \lambda_{d,\boldsymbol{k}_i}$. *We have weak tractability if and only if*

$$\lim_{i\to\infty} t_{d,i}\,(\log i)^2 = 0 \quad \textit{for all } d,$$

and there exists a function $f : [0, 1/2) \to \{1, 2, 3, \ldots\}$ *such that*

$$\sup_{\beta\in(0,1/2]} \beta^{-2} \sup_{d\geq f(\beta)} \sup_{i\geq\lceil\exp(d\sqrt{b})\rceil+1} \lim_{i\to\infty} t_{d,i}\,(\log i)^2 < \infty.$$

Proof. The proof of the theorem is similar to that of Theorem 4, using (24). □

Remark 1 *Results for further tractability notions, such as quasi-polynomial tractability or* (s,t)*-weak tractability, can be shown using similar arguments as above and results from [6], [9], [13], and the papers cited therein.*

To be more concrete, we consider the situation where the $\lambda_{\boldsymbol{k}}$ are specified in terms of positive *coordinate weights*, $w_1, \ldots, w_d$, and positive *smoothness weights*, $s_1, s_2, \ldots$:

$$\lambda_{d,\boldsymbol{k}} := \prod_{\substack{\ell=1\\ k_\ell>0}}^{d} w_\ell s_{k_\ell}, \qquad \boldsymbol{k} \in \mathbb{N}_0^d,\ d \in \mathbb{N}. \tag{26}$$

This is a generalization of the example in Section 1.1, where $s_k = k^{-r}$. This form of the $\lambda_{d,\boldsymbol{k}}$ is considered in greater detail in Section 4. The same argument as in Section 1.6 implies that the sum of the $\lambda_{d,\boldsymbol{k}_i}^{\eta}$ is bounded above as

$$\sum_{i=1}^{\infty} \lambda_{d,\boldsymbol{k}_i}^{\eta} \leq \exp\left(\sum_{k=1}^{\infty} s_k^{\eta} \sum_{\ell=1}^{d} w_\ell^{\eta}\right) \leq \exp\left(\sum_{k=1}^{\infty} s_k^{\eta} \sum_{\ell=1}^{\infty} w_\ell^{\eta}\right).$$

Moreover, it also follows that for any fixed positive integer i_0, the sum of the $\lambda_{d,\boldsymbol{k}_i}^{\eta}$ is bounded below as

$$\sup_{d\in\mathbb{N}} \sum_{i=i_0}^{\infty} \lambda_{d,\boldsymbol{k}_i}^{\eta} \geq w_1^{\eta} \sum_{k=i_0}^{\infty} s_{\kappa_i}^{\eta}$$

considering only $\boldsymbol{k}_i$ of the form $(k, 0, 0, \ldots, 0)$, and ordering $\boldsymbol{s}$ so that $s_{\kappa_1} \geq s_{\kappa_2} \geq \cdots$,

$$\sup_{d\in\mathbb{N}} \sum_{i=i_0}^{\infty} \lambda_{d,\boldsymbol{k}_i}^{\eta} \geq s_1^{\eta} \sum_{i=i_0}^{\infty} w_{\ell_i}^{\eta}$$

considering only $\boldsymbol{k}_i$ of the form $(0, \ldots, 0, 1, 0, \ldots, 0)$, where the non-zero component is at the ℓ-th position with $i_0 \leq \ell \leq d$, and ordering $\boldsymbol{w}$ so that $w_{\ell_1} \geq w_{\ell_2} \geq \cdots$.

Thus, we have necessary and sufficient conditions for strong tractability.

Corollary 1 *For the $\lambda_{d,\boldsymbol{k}}$ of the form* (26) *we have strong polynomial tractability if and only if there exists $\eta > 0$ such that*

$$\sum_{k=1}^{\infty} s_k^{\eta} < \infty \text{ and } \sum_{\ell=1}^{\infty} w_\ell^{\eta} < \infty.$$

Remark 2 *Note that in the setting of this example, the term $\sum_{i=1}^{\infty} \lambda_{d,\boldsymbol{k}_i}^{\eta}$ will usually depend exponentially on d unless the coordinate weights decay to zero fast enough with increasing ℓ. Hence, we can only hope for tractability under the presence of decaying w_ℓ. For further details on weighted approximation problems and tractability, we refer to [8].*

3 Tracking the Decay Rate of the Series Coefficients of the Input Function

From error bound (10) it follows that the faster the $\widehat{f}(\boldsymbol{k}_i)$ decay, the faster $\mathrm{APP}(f, n)$ converges to the solution. Unfortunately, adaptive Algorithm 2 does not adapt to the decay rate of the $\widehat{f}(\boldsymbol{k}_i)$ as $i \to \infty$. It simply bounds $\|f\|_{\mathcal{F}}$ based on a pilot sample. The algorithm presented in this section tracks the rate of decay of the $\widehat{f}(\boldsymbol{k}_i)$ and terminates sooner if the $\widehat{f}(\boldsymbol{k}_i)$ decay more quickly. Similar algorithms for quasi-Monte Carlo integration are developed in [3], [5], and [4].

There is an implicit assumption in this section that function data are cheap and we can afford a large sample size. A large sample size is required to do meaningful tracking of the decay of the series coefficients. The previous section and the next section are more suited to the case when function data are expensive and the final sample size must be modest.

Let $(n_j)_{j\geq 0}$ be a strictly increasing sequence of non-negative integers. This sequence may increase geometrically or algebraically. Define the sets of wavenumbers analogously to (15),

$$n_{-1} = 0, \qquad \mathcal{K}_j := \{\boldsymbol{k}_{n_{j-1}+1}, \ldots, \boldsymbol{k}_{n_j}\} \quad \text{for } j \in \mathbb{N}_0.$$

If $n_0 = 0$, then $\mathcal{K}_0$ is empty. For any $f \in \mathcal{F}$, define the norms of subsets of series coefficients:

$$\sigma_j(f) := \left\| \left(\frac{\widehat{f}(\boldsymbol{k})}{\lambda_{\boldsymbol{k}}} \right)_{\boldsymbol{k}\in\mathcal{K}_j} \right\|_{\rho} \qquad \text{for } j \in \mathbb{N}. \tag{27}$$

Thus, $\|f\|_{\mathcal{F}} = \left\| \left(\sigma_j(f)\right)_{j\in\mathbb{N}_0} \right\|_{\rho}$.

For this section, we define the cone of input functions by

$$\mathcal{C} := \left\{ f \in \mathcal{F} \colon \sigma_{j+r}(f) \leq ab^r \sigma_j(f) \ \forall j, r \in \mathbb{N} \right\}. \tag{28}$$

Here, a and b are positive reals with $b < 1 < a$. The constant a is an inflation factor, and the constant b defines the general rate of decay of the $\sigma_j(f)$ for $f \in \mathcal{C}$. Because ab^r may be greater than one, we do not require the series coefficients of the solution, $\mathrm{SOL}(f)$, to decay monotonically. However, we expect their partial sums to decay steadily. The series coefficients for wavenumbers $\boldsymbol{k} \in \mathcal{K}_0$ do not affect the definition of $\mathcal{C}$ and may behave erratically. Lemma 1 implies that

$$\left\| \left(\widehat{f}(\boldsymbol{k})\right)_{\boldsymbol{k}\in\mathcal{K}_j} \right\|_{\tau} \leq \sigma_j(f)\Lambda_j, \qquad \text{where } \Lambda_j := \left\| \left(\lambda_{\boldsymbol{k}}\right)_{\boldsymbol{k}\in\mathcal{K}_j} \right\|_{\rho'}. \tag{29}$$

From (7) and (8) it follows that the norm of the solution operator is

$$\|\mathrm{SOL}\|_{\mathcal{F}\to\mathcal{G}} = \left\| \left(\Lambda_j\right)_{j\in\mathbb{N}_0} \right\|_{\rho'} < \infty \tag{30}$$

If f belongs to the $\mathcal{C}$ defined in (28) and $n_0 = 0$, then

$$\begin{aligned}\|f\|_{\mathcal{F}} &= \left\| \left(\left\| \left(\frac{\widehat{f}(\boldsymbol{k})}{\lambda_{\boldsymbol{k}}} \right)_{\boldsymbol{k}\in\mathcal{K}_j} \right\|_\rho \right)_{j\in\mathbb{N}} \right\|_\rho = \left\| \left(\sigma_j(f)\right)_{j\in\mathbb{N}} \right\|_\rho \\ &\le \left\| \left(\sigma_1(f), ab\sigma_1(f), ab^2\sigma_1(f), \ldots\right) \right\|_\rho \\ &= \left(1 + \frac{a^\rho b^\rho}{1-b^\rho}\right)^{1/\rho} \left\| \left(\frac{\widehat{f}(\boldsymbol{k})}{\lambda_{\boldsymbol{k}}} \right)_{\boldsymbol{k}\in\mathcal{K}_1} \right\|_\rho .\end{aligned}$$

Comparing this inequality to the definition of $\mathcal{C}$ in the previous section, it can be seen that $\mathcal{C}$ defined in (28) is a subset of $\mathcal{C}$ defined in (16) if we choose $A = \left(1 + \frac{a^\rho b^\rho}{1-b^\rho}\right)^{1/\rho}$ in (16).

From the expression for the error in (10) and the definition of the cone in (28), we can now derive a data-driven error bound for all $f \in \mathcal{C}$ and $j \in \mathbb{N}$:

$$\begin{aligned}&\|\mathrm{SOL}(f) - \mathrm{APP}(f, n_j)\|_{\mathcal{G}} \\ &\quad= \left\| \left(\widehat{f}(\boldsymbol{k}_i) \right)_{i=n_j+1}^{\infty} \right\|_\tau = \left\| \left(\left\| \left(\widehat{f}(\boldsymbol{k}) \right)_{\boldsymbol{k}\in\mathcal{K}_l} \right\|_\tau \right)_{l=j+1}^{\infty} \right\|_\tau \\ &\quad\le \left\| \left(\sigma_l(f)\Lambda_l\right)_{l=j+1}^{\infty} \right\|_\tau \qquad \text{by (29)} \\ &\quad= \left\| \left(\sigma_{j+r}(f)\Lambda_{j+r}\right)_{r=1}^{\infty} \right\|_\tau \\ &\quad\le a\sigma_j(f) \left\| \left(b^r\Lambda_{j+r}\right)_{r=1}^{\infty} \right\|_\tau =: \mathrm{ERR}\left(\left(\widehat{f}(\boldsymbol{k}_i) \right)_{i=1}^{n_j}, n_j \right) \qquad \text{by (28).} \end{aligned} \tag{31}$$

This upper bound depends only on the function data and the parameters defining $\mathcal{C}$. The error vanishes as $j \to \infty$ because $\sigma_j(f) \le ab^{j-1}\sigma_1(f) \to 0$ and $\Lambda_j \to 0$. Moreover, the error bound for $\mathrm{APP}(f, n_j)$ depends on $\sigma_j(f)$, whose rate of decay need not be postulated in advance.

These assumptions accommodate both the cases where the approximation converges algebraically and exponentially. To illustrate the algebraic case, suppose that $\widehat{f}(\boldsymbol{k}_i)/\lambda_{\boldsymbol{k}_i} = \mathcal{O}(i^{-r_\Delta})$ for some positive $r_\Delta > 1/\rho$. For this algebraic case one would normally define $\mathcal{C}$ in terms of an exponentially increasing sequence, $(n_j)_{j\ge0}$, e.g., $n_j = n_0 2^j$, which implies that

$$\begin{aligned}\sigma_j(f) &= \left[\sum_{i=n_0 2^{j-1}+1}^{n_0 2^j} \left| \frac{\widehat{f}(\boldsymbol{k}_i)}{\lambda_{\boldsymbol{k}_i}} \right|^\rho \right]^{1/\rho} = \mathcal{O}\left(\left[\sum_{i=n_0 2^{j-1}+1}^{n_0 2^j} i^{-\rho r_\Delta} \right]^{1/\rho} \right) \\ &= \mathcal{O}\left(2^{-j(r_\Delta - 1/\rho)}\right).\end{aligned}$$

Reasonable functions would satisfy

$$C_{\mathrm{lo}} 2^{-j(r_\Delta - 1/\rho)} \le \sigma_j(f) \le C_{\mathrm{up}} 2^{-j(r_\Delta - 1/\rho)}$$

for some constants C_{lo} and C_{up}. Choosing $a \ge C_{\mathrm{up}}/C_{\mathrm{lo}}$ and $b \ge 2^{-(r_\Delta - 1/\rho)}$ causes the cone $\mathcal{C}$ to include such functions. Note that only the ratio of C_{up} to C_{lo} need be assumed to determine a, and choosing b larger than necessary does not affect the order of the decay of the error bound.

To illustrate the exponential case, suppose that $\widehat{f}(\boldsymbol{k}_i)/\lambda_{\boldsymbol{k}_i} = \mathcal{O}(\mathrm{e}^{-r_\Delta i})$. For this exponential case one would normally define $\mathcal{C}$ in terms of an arithmetic sequence, $(n_j)_{j\ge0}$, e.g., $n_j = n_0 + js$, where s is a positive integer. This implies that

$$\sigma_j(f) = \left[\sum_{i=n_0+js-s+1}^{n_0+js} \left| \frac{\widehat{f}(\boldsymbol{k}_i)}{\lambda_{\boldsymbol{k}_i}} \right|^\rho \right]^{1/\rho} = \mathcal{O}\left(\left[\sum_{i=n_0+js-s+1}^{n_0+js} \mathrm{e}^{-\rho r_\Delta i} \right]^{1/\rho} \right)$$

$$= \mathcal{O}\left(\mathrm{e}^{-jr\Delta s}\right).$$

Analogous to the algebraic case, reasonable functions would satisfy $C_{\mathrm{lo}}\mathrm{e}^{-jr\Delta s} \leq \sigma_j(f) \leq C_{\mathrm{up}}\mathrm{e}^{-jr\Delta s}$ for some constants C_{lo} and C_{up}. Choosing $a \geq C_{\mathrm{up}}/C_{\mathrm{lo}}$ and $b \geq \mathrm{e}^{-r\Delta s}$ causes the cone $\mathcal{C}$ to include such functions. Again, only the ratio of C_{up} to C_{lo} need be assumed to determine a, and choosing b larger than necessary does not affect the order of the decay of the error bound.

3.1 The Adaptive Algorithm and Its Computational Cost

The data-driven error bound in (31) forms the basis for an adaptive Algorithm 3, which solves our problem for input functions in the cone $\mathcal{C}$ defined in (28). The following theorem establishes its viability and computational cost. In deriving upper bounds on the computational cost and lower bounds on the complexity, we may sacrifice tightness for simplicity.

Algorithm 3 Adaptive ALG for a Cone of Input Functions Tracking the Series Coefficient Decay Rate

Parameters: the Banach spaces $\mathcal{F}$ and $\mathcal{G}$, including the weights $\boldsymbol{\lambda}$; a strictly increasing sequence of non-negative integers, $(n_j)_{j\geq 0}$; an inflation factor, a; the general decay rate, b; APP satisfying (10)
Input: a black-box function, f; an absolute error tolerance, $\varepsilon > 0$
Ensure: Error criterion (2) for the cone defined in (28)
  Let $j \leftarrow 0$
  repeat
    Let $j \leftarrow j + 1$
    Compute $\mathrm{ERR}\big(\big(\widehat{f}(\boldsymbol{k}_i)\big)_{i=1}^{n_j}, n_j\big)$ as defined in (31)
  until $\mathrm{ERR}\big(\big(\widehat{f}(\boldsymbol{k}_i)\big)_{i=1}^{n_j}, n_j\big) \leq \varepsilon$
  Return $\mathrm{ALG}(f,\varepsilon) = \mathrm{APP}(f, n_j)$

Theorem 6 *Algorithm 3 yields an answer satisfying absolute error criterion* (2)*, i.e.,* $\mathrm{ALG} \in \mathcal{A}(\mathcal{C})$ *for* $\mathcal{C}$ *defined in* (28)*. The information cost is* $\mathrm{COST}(\mathrm{ALG}, f, \varepsilon) = n_{j^*}$*, where* j^* *is defined implicitly as*

$$j^* = \min\left\{j \in \mathbb{N} : \mathrm{ERR}\big(\big(\widehat{f}(\boldsymbol{k}_i)\big)_{i=1}^{n_j}, n_j\big) \leq \varepsilon\right\}. \tag{32}$$

Moreover, $\mathrm{COST}(\mathrm{ALG}, \mathcal{C}, \varepsilon, R) \leq n_{j^\dagger}$*, where* $j^\dagger$ *is defined as follows:*

$$j^\dagger = \min\left\{j \in \mathbb{N} : \left\|\big(b^{j+r}\Lambda_{j+r}\big)_{r=1}^{\infty}\right\|_\tau \leq \frac{b\varepsilon}{Ra^2}\left(\frac{1-b^{j\rho}}{1-b^\rho}\right)^{1/\rho}\right\}. \tag{33}$$

Proof. The value of j^* in (32) follows directly from the error criterion. The success of the algorithm follows from the error bound in (31).

For the remainder of the proof consider R and ε to be fixed. For any $f \in \mathcal{C} \cap \mathcal{B}_R$ and for any $j^\dagger$ defined as in (33), it follows that

$$\begin{aligned} R \geq \|f\|_{\mathcal{F}} &= \left\|\left(\frac{\widehat{f}(\boldsymbol{k})}{\lambda_{\boldsymbol{k}}}\right)_{\boldsymbol{k}\in\mathbb{K}}\right\|_\rho \geq \left\|(\sigma_j(f))_{j=1}^{j^\dagger}\right\|_\rho \qquad \text{by (27)} \\ &\geq \left\|\Big(a^{-1}b^{1-j^\dagger}\sigma_{j^\dagger}(f), \ldots, a^{-1}b^{-1}\sigma_{j^\dagger}(f), \sigma_{j^\dagger}(f)\Big)\right\|_\rho \qquad \text{by (28)} \end{aligned}$$

$$
\begin{aligned}
&\geq \frac{\sigma_{j^\dagger}(f)}{a} \left\| \left(b^{1-j^\dagger}, \ldots, b^{-1}, 1\right) \right\|_\rho = \frac{b\sigma_{j^\dagger}(f)}{a} \left(\frac{b^{-j^\dagger \rho} - 1}{1 - b^\rho} \right)^{1/\rho} \\
&\geq \sigma_{j^\dagger}(f) \, \frac{Ra}{\varepsilon} \left\| \left(b^r \Lambda_{j^\dagger + r}\right)_{r=1}^{\infty} \right\|_\tau \qquad \text{by the definition of } j^\dagger \text{ in (33)} \\
&= \frac{R}{\varepsilon} \, \mathrm{ERR}\left(\left(\widehat{f}(\boldsymbol{k}_i)\right)_{i=1}^{n_{j^\dagger}}, n_{j^\dagger} \right).
\end{aligned}
$$

From this last inequality, it follows that $j^\dagger \geq j^*$. □

Although Algorithm 3 tracks the decay rate of the $\widehat{f}(\boldsymbol{k}_i)$, the information cost bound and complexity bound in the theorem above do not reflect different decay rates of the $\widehat{f}(\boldsymbol{k}_i)$. That is a subject for future investigation.

3.2 Essential Optimality of the Algorithm

To establish the essential optimality of Algorithm 3 requires some additional, reasonable assumptions on the sequences $(n_j)_{j \in \mathbb{N}_0}$ and $\left(\sigma_j(f)\right)_{j \in \mathbb{N}_0}$. Recall from (30) that $\left(\Lambda_j\right)_{j \in \mathbb{N}_0}$ has a finite ρ' norm. We require that the Λ_j must decay steadily with j:

$$
\alpha^{-1} \beta^r \Lambda_j \leq \Lambda_{j+r} \leq \alpha \gamma^r \Lambda_j \quad \forall j, r \in \mathbb{N}_0, \qquad \text{for some } \beta, \gamma < 1 \leq \alpha. \tag{34}
$$

We also assume that the ratio of the largest to smallest $\lambda_{\boldsymbol{k}}$ in a group is bounded above:

$$
\sup_{j \in \mathbb{N}} \frac{\lambda_{\boldsymbol{k}_{n_{j-1}+1}}}{\lambda_{\boldsymbol{k}_{n_j}}} \leq S_1 < \infty. \tag{35}
$$

For the illustrative choices of $(n_j)_{j \in \mathbb{N}_0}$ and $\left(\boldsymbol{k}_i\right)_{i \in \mathbb{N}}$ preceding Section 3.1 this assumption holds. Let $\mathrm{card}(\cdot)$ denote the cardinality of a set. We assume that if $\mathcal{J}$ is an arbitrary set of wavenumbers with $\mathrm{card}(\mathcal{J}) \leq n_j$, then there exists some $l \leq n_{j+1}$ for which $\mathcal{K}_l \setminus \mathcal{J}$ retains some significant fraction of the original $\mathcal{K}_l$ elements:

$$
\inf_{j \in \mathbb{N}} \min_{\mathcal{J} \subset \mathbb{K} \,:\, \mathrm{card}(\mathcal{J}) \leq n_j} \max_{0 \leq l \leq j+1} \frac{\mathrm{card}(\mathcal{K}_l \setminus \mathcal{J})}{\mathrm{card}(\mathcal{K}_l)} \geq S_2 > 0. \tag{36}
$$

Again, for the illustrative choices of $(n_j)_{j \in \mathbb{N}_0}$ and $\left(\boldsymbol{k}_i\right)_{i \in \mathbb{N}}$ preceding Section 3.1 this assumption holds.

The following theorem establishes a lower bound on the complexity of our problem for input functions in $\mathcal{C}$. The theorem after that shows that the cost of our algorithm as given in Theorem 6 is essentially no worse than this lower bound.

Theorem 7 *A lower bound on the complexity of the linear problem is*

$$
\mathrm{COMP}(\mathcal{A}(\mathcal{C}), \varepsilon, R) > n_{j^\ddagger},
$$

where

$$
j^\ddagger = \max \left\{ j \in \mathbb{N} : b^{j+1} \Lambda_{j+1} > \frac{2a\alpha\varepsilon}{R(a-1)(1-b^\rho)^{1/\rho}} \left[1 + \left(\frac{1}{S_2} - 1 \right) S_1^\rho \right]^{1/\rho} \right\}.
$$

Proof. As in the proof of Theorem 3 we consider fixed and arbitrary R and ε. We proceed by carefully constructing the test input functions, f_1 and $f_\pm = f_1 \pm f_2$, lying in $\mathcal{C} \cap \mathcal{B}_R$, which yield

the same approximate solution but different true solutions. This leads to a lower bound on $\mathrm{COMP}(\mathcal{A}(\mathcal{C}), \varepsilon, R)$. The proof is provided for $\rho' < \infty$. The proof for $\rho' = \infty$ is similar.

The first test function $f_1 \in \mathcal{C}$ is defined in terms of its series coefficients—inspired by Lemma 1—as

$$f_1 = f_{10} + f_{11} + \cdots, \qquad \widehat{f}_{1j}(\boldsymbol{k}) := \begin{cases} \dfrac{c_1 b^j \lambda_{\boldsymbol{k}}^{\rho'/\rho+1}}{\Lambda_j^{\rho'/\rho}}, & \boldsymbol{k} \in \mathcal{K}_j, \\ 0, & \boldsymbol{k} \notin \mathcal{K}_j, \end{cases}$$

$$c_1 := \frac{R(a+1)(1-b^\rho)^{1/\rho}}{2a}.$$

It can be verified that the test function lies both in $\mathcal{B}_R$ and in $\mathcal{C}$:

$$\sigma_j(f_1) = \left\| \left(\frac{\widehat{f}_{1j}(\boldsymbol{k})}{\lambda_{\boldsymbol{k}}} \right)_{\boldsymbol{k} \in \mathcal{K}_l} \right\|_\rho = c_1 b^j, \qquad j \in \mathbb{N}_0,$$

$$\|f_1\|_{\mathcal{F}} = \left\| \big(\sigma_j(f)\big)_{j \in \mathbb{N}_0} \right\|_\rho = \frac{c_1}{(1-b^\rho)^{1/\rho}} = \frac{R(a+1)}{2a} \le R,$$

$$\sigma_{j+r}(f_1) = b^r \sigma_j(f_1) \le a b^r \sigma_j(f_1), \qquad j, r \in \mathbb{N}_0.$$

Now let ALG' be an arbitrary algorithm in $\mathcal{A}(\mathcal{C})$, and suppose that $\mathrm{ALG}'(f_1, \varepsilon)$ samples $f_1(\boldsymbol{k})$ for $\boldsymbol{k} \in \mathcal{J}$. Let $\widetilde{\mathcal{K}}_j = \mathcal{K}_j \setminus \mathcal{J}$ for all non-negative integers j. Construct the function f_2, having zero coefficients for $\boldsymbol{k} \in \mathcal{J}$, but otherwise looking like f_1:

$$f_2 = f_{20} + f_{21} + \cdots, \qquad \widehat{f}_{2j}(\boldsymbol{k}) := \begin{cases} \dfrac{c_2 b^j \lambda_{\boldsymbol{k}}^{\rho'/\rho+1}}{\widetilde{\Lambda}_j^{\rho'/\rho}}, & \boldsymbol{k} \in \widetilde{\mathcal{K}}_j, \\ 0, & \text{otherwise}, \end{cases}$$

$$c_2 := \frac{R(a-1)(1-b^\rho)^{1/\rho}}{2a}, \qquad \widetilde{\Lambda}_j := \left\| \big(\lambda_{\boldsymbol{k}}\big)_{\boldsymbol{k} \in \widetilde{\mathcal{K}}_j} \right\|_{\rho'} \le \Lambda_j,$$

$$\sigma_j(f_2) = \left\| \left(\frac{\widehat{f}_{2j}(\boldsymbol{k})}{\lambda_{\boldsymbol{k}}} \right)_{\boldsymbol{k} \in \widetilde{\mathcal{K}}_j} \right\|_\rho = \begin{cases} c_2 b^j, & \widetilde{\mathcal{K}}_j \ne \emptyset, \\ 0, & \widetilde{\mathcal{K}}_j = \emptyset, \end{cases} \qquad j \in \mathbb{N}_0,$$

$$\|f_2\|_{\mathcal{F}} = \left\| \big(\sigma_j(f_2)\big)_{j \in \mathbb{N}_0} \right\|_\rho \le \frac{c_2}{(1-b^\rho)^{1/\rho}} = \frac{R(a-1)}{2a} \le R,$$

$$\|\mathrm{SOL}(f_{2j})\|_{\mathcal{G}} = \sigma_j(f_{2j}) \widetilde{\Lambda}_j = c_2 b^j \widetilde{\Lambda}_j, \qquad j \in \mathbb{N}_0,$$

$$\|\mathrm{SOL}(f_2)\|_{\mathcal{G}} = \left\| \big(c_2 b^j \widetilde{\Lambda}_j\big)_{j \in \mathbb{N}_0} \right\|_\tau = \frac{R(a-1)(1-b^\rho)^{1/\rho}}{2a} \left\| \big(b^j \widetilde{\Lambda}_j\big)_{j \in \mathbb{N}_0} \right\|_\tau. \tag{37}$$

Furthermore, define $f_\pm = f_1 \pm f_2$. It can be verified that $f_\pm$ also lie both in $\mathcal{B}_R$ and in $\mathcal{C}$:

$$\|f_\pm\|_{\mathcal{F}} \le \|f_1\|_{\mathcal{F}} + \|f_2\|_{\mathcal{F}} \le \frac{c_1 + c_2}{(1-b^\rho)^{1/\rho}} = R,$$

$$\sigma_j(f_\pm) \ge \sigma_j(f_1) - \sigma_j(f_2) \ge (c_1 - c_2)\, b^j = \frac{R(1-b^\rho)^{1/\rho} b^j}{a}, \qquad j \in \mathbb{N}_0,$$

$$\begin{aligned} \sigma_{j+r}(f_\pm) &\le \sigma_{j+r}(f_1) + \sigma_{j+r}(f_2) \le (c_1 + c_2) b^{j+r} \\ &= R(1-b^\rho)^{1/\rho} b^{j+r} \le a b^r \sigma_j(f_\pm), \qquad j \in \mathbb{N}_0. \end{aligned}$$

Since $\widehat{f}_2(\boldsymbol{k}) = 0$ for $\boldsymbol{k} \in \mathcal{J}$, it follows that $\mathrm{ALG}'(f_\pm, \varepsilon) = \mathrm{ALG}'(f_1, \varepsilon)$. But, even though the two test functions $f_\pm$ lead to the same approximate solution, they have different true solutions. In particular,

$$\varepsilon \ge \max\left\{ \left\| \mathrm{SOL}(f_+) - \mathrm{ALG}'(f_+, \varepsilon) \right\|_{\mathcal{G}}, \left\| \mathrm{SOL}(f_-) - \mathrm{ALG}'(f_-, \varepsilon) \right\|_{\mathcal{G}} \right\}$$

$$
\begin{aligned}
&\geq \frac{1}{2}\left[\left\|\mathrm{SOL}(f_+) - \mathrm{ALG}(f_1,\varepsilon)\right\|_{\mathcal{G}} + \left\|\mathrm{SOL}(f_-) - \mathrm{ALG}'(f_1,\varepsilon)\right\|_{\mathcal{G}}\right] \\
&\qquad \text{since } \mathrm{ALG}'(f_\pm,\varepsilon) = \mathrm{ALG}'(f_1,\varepsilon) \\
&\geq \frac{1}{2}\left\|\mathrm{SOL}(f_+) - \mathrm{SOL}(f_-)\right\|_{\mathcal{G}} \quad \text{by the triangle inequality} \\
&\geq \frac{1}{2}\left\|\mathrm{SOL}(f_+ - f_-)\right\|_{\mathcal{G}} \quad \text{since SOL is linear} \\
&= \left\|\mathrm{SOL}(f_2)\right\|_{\mathcal{G}} = \frac{R(a-1)(1-b^\rho)^{1/\rho}}{2a}\left\|\left(b^j\widetilde{\Lambda}_j\right)_{j\in\mathbb{N}_0}\right\|_\tau \qquad \text{by (37).}
\end{aligned} \tag{38}
$$

Suppose that $\mathrm{card}(\mathcal{J}) = \mathrm{COST}(\mathrm{ALG}', f_\pm, \varepsilon) \leq n_{j^\star}$. Then by condition (36), there exists an $l^\star \leq j^\star + 1$ where $\mathrm{card}(\widetilde{\mathcal{K}}_{l^\star}) \geq S_2\,\mathrm{card}(\mathcal{K}_{l^\star})$. This implies a lower bound on $\widetilde{\Lambda}_{l^\star}$. Let $m = n_{l^\star} - n_{l^\star-1} = \mathrm{card}(\mathcal{K}_{l^\star})$. Then, $m_{\mathrm{in}} = \lceil S_2 m \rceil \geq S_2 m$ is a lower bound on $\mathrm{card}(\widetilde{\mathcal{K}}_{l^\star})$, and $m_{\mathrm{out}} = m - m_{\mathrm{in}} \leq (1-S_2)m$ is an upper bound on $\mathrm{card}(\mathcal{K}_{l^\star} \setminus \widetilde{\mathcal{K}}_{l^\star})$. Moreover,

$$
\begin{aligned}
\Lambda_{l^\star}^\rho &= \sum_{i\in\mathcal{K}_{l^\star}} \lambda_{\boldsymbol{k}_i}^\rho = \widetilde{\Lambda}_{l^\star}^\rho + \sum_{i\in\mathcal{K}_{l^\star}\setminus\widetilde{\mathcal{K}}_{l^\star}} \lambda_{\boldsymbol{k}_i}^\rho \\
&\leq \widetilde{\Lambda}_{l^\star}^\rho + m_{\mathrm{out}}\lambda_{\boldsymbol{k}_{n_{l^\star-1}+1}}^\rho \qquad \text{by the ordering of the } \lambda_{\boldsymbol{k}_i} \\
&\leq \widetilde{\Lambda}_{l^\star}^\rho + m_{\mathrm{out}}S_1^\rho\lambda_{\boldsymbol{k}_{n_{l^\star}}}^\rho \qquad \text{by (35)} \\
&\leq \widetilde{\Lambda}_{l^\star}^\rho + \frac{m_{\mathrm{out}}}{m_{\mathrm{in}}}S_1^\rho\widetilde{\Lambda}_{l^\star}^\rho \qquad \text{by the definition of } \widetilde{\Lambda}_{l^\star} \\
&\leq \left[1 + \left(\frac{1}{S_2} - 1\right)S_1^\rho\right]\widetilde{\Lambda}_{l^\star}^\rho \qquad \text{by the bounds on } m_{\mathrm{in}} \text{ and } m_{\mathrm{out}} \\
&\leq \left[1 + \left(\frac{1}{S_2} - 1\right)S_1^\rho\right] b^{-\rho l^\star}\left\|\left(b^j\widetilde{\Lambda}_j\right)_{j\in\mathbb{N}_0}\right\|_\tau^\rho .
\end{aligned}
$$

Returning to (38), the above inequality implies that

$$
\varepsilon \geq \frac{R(a-1)(1-b^\rho)^{1/\rho}}{2a}\left[1 + \left(\frac{1}{S_2} - 1\right)S_1^\rho\right]^{-1/\rho} b^{l^\star}\Lambda_{l^\star}.
$$

Since $l^\star \leq j^\star + 1$ it follows that $b^{l^\star} \geq b^{j^\star+1}$ and from condition (34) it follows that $\Lambda_{l^\star} \geq \Lambda_{j^\star+1}/\alpha$. Thus,

$$
\varepsilon \geq \frac{R(a-1)(1-b^\rho)^{1/\rho}}{2a\alpha}\left[1 + \left(\frac{1}{S_2} - 1\right)S_1^\rho\right]^{-1/\rho} b^{j^\star+1}\Lambda_{j^\star+1}.
$$

If any algorithm satisfies the error tolerance ε for all input functions in $\mathcal{C}\cap\mathcal{B}_R$ and has information cost no greater than $n_{j^\star}$, then $j^\star$ must satisfy the above inequality. By contrast, if the above inequality is violated for any $j^\star$, then the information cost of the successful algorithm must be greater than $n_{j^\star}$. This completes the proof. □

Theorem 8 *Adaptive Algorithm 3 is essentially optimal for the cone of input functions defined in* (28).

Proof. Let $j^\dagger(\varepsilon)$ be defined as in (33), with the ε dependence made explicit. Choose $\varepsilon_{\max}$ and $R_{\min}$ in (14) such that $j^\dagger(\varepsilon) \geq 2$. This definition implies that

$$
b^{j^\dagger(\varepsilon)}\Lambda_{j^\dagger(\varepsilon)} = \frac{[1-(\gamma b)^\tau]^{1/\tau}}{\alpha}\left\|\left(b^{j^\dagger(\varepsilon)-1+r}\alpha\gamma^{r-1}\Lambda_{j^\dagger(\varepsilon)}\right)_{r=1}^\infty\right\|_\tau
$$

$$
\begin{aligned}
&\geq \frac{[1-(\gamma b)^\tau]^{1/\tau}}{\alpha} \left\| \left(b^{j^\dagger(\varepsilon)-1+r}\Lambda_{j^\dagger(\varepsilon)-1+r}\right)_{r=1}^{\infty} \right\|_\tau \qquad \text{by (34)} \\
&> \frac{b[1-(\gamma b)^\tau]^{1/\tau}\varepsilon}{Ra^2\alpha} \left(\frac{1-b^{\rho(j^\dagger(\varepsilon)-1)}}{1-b^\rho}\right)^{1/\rho} \qquad \text{by (33)} \\
&\geq \frac{b[1-(\gamma b)^\tau]^{1/\tau}\varepsilon}{Ra^2\alpha} \qquad \text{since } j^\dagger(\varepsilon) \geq 2 \\
&= \frac{\alpha^2}{(b\beta)^2} \times \frac{2a\alpha\omega\varepsilon}{R(a-1)(1-b^\rho)^{1/\rho}} \left[1+\left(\frac{1}{S_2}-1\right)S_1^\rho\right]^{1/\rho}
\end{aligned}
$$

where

$$
\omega = \frac{(a-1)b^3\beta^2(1-b^\rho)^{1/\rho}[1-(\gamma b)^\tau]^{1/\tau}}{2a^3\alpha^4} \left[1+\left(\frac{1}{S_2}-1\right)S_1^\rho\right]^{-1/\rho}.
$$

Making the ε dependence explicit in the definition of $j^\ddagger(\varepsilon)$ in Theorem 7 it follows from the above inequality that

$$
b^{j^\dagger(\varepsilon)}\Lambda_{j^\dagger(\varepsilon)} > \frac{\alpha^2}{(b\beta)^2} b^{j^\ddagger(\omega\varepsilon)+2}\Lambda_{j^\ddagger(\omega\varepsilon)+2} \geq \alpha b^{j^\ddagger(\omega\varepsilon)}\Lambda_{j^\ddagger(\omega\varepsilon)} \qquad \text{by (34)}.
$$

If $j^\dagger(\varepsilon) \geq j^\ddagger(\omega\varepsilon)$, then (34) implies that

$$
b^{j^\dagger(\varepsilon)}\Lambda_{j^\dagger(\varepsilon)} \leq \alpha(\gamma b)^{j^\dagger(\varepsilon)-j^\ddagger(\omega\varepsilon)} b^{j^\ddagger(\omega\varepsilon)}\Lambda_{j^\ddagger(\omega\varepsilon)} \leq \alpha b^{j^\ddagger(\omega\varepsilon)}\Lambda_{j^\ddagger(\omega\varepsilon)}.
$$

But, this contradicts the above inequality. Thus, $j^\dagger(\varepsilon) < j^\ddagger(\omega\varepsilon)$, and so

$$
\mathrm{COST}(\mathrm{ALG}, \mathcal{C}, \varepsilon, R) \leq n_{j^\dagger(\varepsilon)} < n_{j^\ddagger(\omega\varepsilon)} < \mathrm{COMP}(\mathcal{A}(\mathcal{C}), \omega\varepsilon, R).
$$

Thus, Algorithm 3 is essentially optimal. □

3.3 Tractability

We again would like to study tractability. As it turns out, by using the relation between the cones defined in (16) and (28), respectively, we easily obtain sufficient conditions for tractability.

Theorem 9 *The respective conditions presented in Theorem 4 for the case where $\rho' = \infty$ and in Theorem 5 for the case where $\rho' < \infty$ are sufficient for strong polynomial, polynomial, and weak tractability of the approximation problem defined on cones as in* (28)*.*

Proof. As pointed out above, $\mathcal{C}$ defined in (28) is a subset of $\mathcal{C}$ defined in (16), by choosing $A = \left(1+\frac{a^\rho b^\rho}{1-b^\rho}\right)^{1/\rho}$ in (16). This means that the approximation problem on $\mathcal{C}$ defined in (28) is essentially (i.e., up to constants depending on A, a, b and ρ) no harder than the same problem on $\mathcal{C}$ defined in (16). This, however, implies that all sufficient conditions in Theorem 4 are also sufficient in the case considered in Theorem 9. □

Theorem 9 yields sufficient conditions for the tractability notions considered here. A general result for necessary conditions seems to be more difficult to obtain and is left open for future research.

4 Inferring Coordinate and Smoothness Importance

In Sections 2 and 3, the weights $\boldsymbol{\lambda} = (\lambda_{\boldsymbol{k}})_{\boldsymbol{k}\in\mathbb{K}}$, which appear in the definition of the cone of inputs, $\mathcal{C}$, are taken as given and fixed. One may assume the form suggested in (26), which defines $\boldsymbol{\lambda}$ in terms of coordinate weights and smoothness weights. However, practically speaking it may be difficult to know a priori the values of these weights. This section explores a situation where the initial data collected for the input function data can be used to learn $\boldsymbol{\lambda}$, inferring which input variables in f may be more important and the smoothness of the function.

The motivation for this section is situations where the relative importance of the d input variables of the function is not known from physical considerations. We also envision situations where the cost of function data is large, e.g., the result of an expensive computer simulation. Thus, we are not concerned with the cost of the algorithm beyond the information cost, which we hope to limit to $\mathcal{O}(d)$.

4.1 Product, Order and Smoothness Dependent (POSD) Weights

The $u_{\boldsymbol{k}}$ and the $\lambda_{\boldsymbol{k}}$ in this section are defined as

$$u_{\boldsymbol{k}} = \prod_{\ell=1}^{d} \tilde{u}_{k_\ell}, \quad \lambda_{\boldsymbol{k}} = \Gamma_{\|\boldsymbol{k}\|_0} \prod_{\substack{\ell=1 \\ k_\ell>0}}^{d} w_\ell s_{k_\ell}, \quad \Gamma_0 = s_1 = 1, \quad \boldsymbol{k} \in \mathbb{N}_0^d, \tag{39}$$

where $\boldsymbol{w} = (w_\ell)_{l=1}^d$ is the vector of coordinate weights, $\boldsymbol{s} = (s_k)_{k=1}^\infty$ is the vector of smoothness weights, $\boldsymbol{\Gamma} = (\Gamma_m)_{m=1}^d$ is the vector of *order weights*, and $\|\boldsymbol{k}\|_0$ denotes the number of nonzero elements of $\boldsymbol{k}$. The intuition behind these weights is as follows:

- Coordinate weights quantify the importance for the d input variables in f.
- Smoothness weights quantify the importance of the $\tilde{u}_k$. E.g., if the $\tilde{u}_k$ are polynomials of degree k as in Section 1.1, then the faster the s_k decay, the smoother f is.
- Order weights quantify the importance of effects with different orders; $\boldsymbol{k}$ having one nonzero element corresponds to a first-order or main effect, $\boldsymbol{k}$ having two nonzero elements corresponds to a second-order (interaction) effect. (e.g., first-order, second-order).

This parametrization is motivated by several guiding principles from the experimental design literature [14], which are briefly described below. In statistical parlance, the terms $\widehat{f}(\boldsymbol{k})u_{\boldsymbol{k}}$ are effects.

- *Effect sparsity* assumes that only a small number of inputs in f are important. In (39), this sparsity means that only a small number of product weights $\boldsymbol{w}$ are large. This principle arises in the sufficient condition for strong tractability in Corollary 1.
- *Effect heredity* assumes that lower-order effects are more important than higher-order effects. E.g., $\lambda_{(1,0,0,\ldots,0)}$ should be larger than $\lambda_{(1,1,0,\ldots,0)}$. In (39), this heredity can be enforced by assuming that the order weights Γ_m decrease with m.
- *Effect hierarchy* assumes that an effect is active *only* when all its component effects are active. For example, $\lambda_{(1,1,0,\ldots,0)} > 0$ only when $\lambda_{(1,0,0,\ldots,0)}$ and $\lambda_{(0,1,0,\ldots,0)}$ are both nonzero. This hierarchy is implicitly enforced by the product structure of the weights in (39).
- *Effect smoothness* assumes that lower-degree effects are more important than higher-degree effects. For example, when the $(\tilde{u}_k)_{k\in\mathbb{N}_0}$ are polynomials, this means that linear effects are more important than quadratic effects, which are in turn more significant than cubic effects, and so on. Effect smoothness can be imposed by assuming $\boldsymbol{s}$ to be a decreasing sequence.

The $\lambda_{\boldsymbol{k}}$ defined in (39) are called product, order and smoothness dependent (POSD) weights. From a quasi-Monte Carlo (QMC) perspective, the POSD weights in (39) generalize upon the product-and-order dependent (POD) weights in [7], which were introduced for analyzing QMC methods in partial differential equations with random coefficients. The latter POD weights can be recovered by ignoring the smoothness weights.

Our POSD weights differ from the smoothness-driven product-and-order dependent (SPOD) weights in [2], which were recently used to analyze higher-order QMC methods for stochastic partial differential equations. These SPOD weights take the form:

$$\gamma_{\mathfrak{u}} = \sum_{\boldsymbol{k}\in\{1,\dots,\alpha\}^{|\mathfrak{u}|}} \|\boldsymbol{k}\|_1! \prod_{\ell\in\mathfrak{u}} \left(2^{\delta(k_\ell,\alpha)} w_\ell^{k_\ell}\right), \quad \|\boldsymbol{k}\|_1 = \sum_{l=1}^{d} k_\ell, \quad \mathfrak{u}\subseteq\{1,\dots,d\},$$

where $\delta(k_\ell,\alpha)$ is 1 if $k_\ell = \alpha$ and 0 otherwise. Intuitively, the SPOD weights quantify the importance of each *subspace* (indexed by $\mathfrak{u}$), under a common smoothness structure among subspaces (for further details on SPOD weights, we refer the reader to [2]). In contrast, the proposed POSD weights in (39) instead quantify the importance of each *Fourier series coefficient* $\widehat{f}(\boldsymbol{k})$ (indexed by $\boldsymbol{k}$), under a common smoothness structure among coefficients.

4.2 Inferring POSD Weights from an Initial Sample

Let $\mathcal{C}_{\boldsymbol{\lambda}}$ denote the cone of inputs defined in (16) by POSD weights $\boldsymbol{\lambda} = \big(\lambda_{\boldsymbol{k}}\big)_{\boldsymbol{k}\in\mathbb{N}_0^d}$. As mentioned above, our goal here is to infer $\boldsymbol{\lambda}$ from input function data. We start with an initial set of wavenumbers:

$$\bar{\mathcal{K}} = \{(0,\dots,0,k,0,\dots,0) : k = 0,\dots,k_{\max}\}. \tag{40}$$

The approximation to f based on sampling the series coefficients for these wavenumbers is

$$f_{\text{app}} = \sum_{\boldsymbol{k}\in\bar{\mathcal{K}}} \widehat{f}(\boldsymbol{k}) u_{\boldsymbol{k}}.$$

We choose the $\mathcal{C}_{\boldsymbol{\lambda}}$ that best fits f by selecting $\boldsymbol{\lambda}$ to make the norm of f_{app} small:

$$\bar{\boldsymbol{\lambda}} = \boldsymbol{\lambda}(\bar{\boldsymbol{w}}, \bar{\boldsymbol{s}}, \boldsymbol{\Gamma}),$$
$$\text{where } (\bar{\boldsymbol{w}}, \bar{\boldsymbol{s}}) = \min\left\{ \underset{(\boldsymbol{w},\boldsymbol{s})\in\mathcal{W}\times\mathcal{S}}{\operatorname{argmin}} \left\| \left(\frac{\widehat{f}(\boldsymbol{k})}{\lambda_{\boldsymbol{k}}(\boldsymbol{w},\boldsymbol{s},\boldsymbol{\Gamma})} \right)_{\boldsymbol{k}\in\bar{\mathcal{K}}} \right\|_\rho \right\}. \tag{41}$$

Here, $\mathcal{W}$ is a candidate set for coordinate weights, e.g., $\mathcal{W} = [0, w^*]^d$, and $\mathcal{S}$ is a candidate set for the smoothness weights, e.g., $\mathcal{S} = \{(1/k^r)_{k=1}^{\infty} : r > 0\}$. The inner minimization finds the $(\boldsymbol{w},\boldsymbol{s})$ that minimizes the approximate norm of the input function. This minimizer may be non-unique, so the outer minimization chooses the smallest such $(\boldsymbol{w},\boldsymbol{s})$. Making the coordinate and smoothness weights as small as possible helps enforce the principles of effect sparsity. The optimum, $(\bar{\boldsymbol{w}}, \bar{\boldsymbol{s}})$, then defines the *data-inferred* POSD $\boldsymbol{\lambda}$, denoted $\bar{\boldsymbol{\lambda}}$.

The candidate sets $\mathcal{W}$ and $\mathcal{S}$ should be constructed such that the coordinate and smoothness weights have a priori upper bounds. Otherwise the inner minimization would choose huge values for $\boldsymbol{w}$ and $\boldsymbol{s}$ to maximize the $\lambda_{\boldsymbol{k}}(\boldsymbol{w},\boldsymbol{s},\boldsymbol{\Gamma})$ and minimize the norm of f_{app}. The cardinality of the initial set of wavenumbers is $dk_{\max}+1$. There is a trade-off between keeping $k_{\max}$ small enough to reducing cost and making $k_{\max}$ large enough to ensuring robustness.

For simplicity, we assume that order weights, $\boldsymbol{\Gamma}$, are fixed a priori. If desired, they too could be inferred as the next step. However, since we want to limit the size of the initial sample to $\mathcal{O}(d)$ we must sample judiciously the higher order interactions.

The optimization in (41) is nontrivial to solve numerically. In practice, we iteratively optimize over $\boldsymbol{w}$ and then $\boldsymbol{s}$ until convergence is reached. At each step of the iteration $\|(\hat{f}(\boldsymbol{k})/\lambda_{\boldsymbol{k}})_{\boldsymbol{k}\in\bar{\mathcal{K}}}\|_\rho$ decreases.

Algorithm 4 combines the construction of data-inferred POSD weights, $\bar{\boldsymbol{\lambda}}$, with Algorithm 2 of Section 2. This algorithm succeeds for input functions in the cone

$$\bar{\mathcal{C}} := \left\{ f \in \mathcal{F} : f \in \mathcal{C}_{\bar{\boldsymbol{\lambda}}} \text{ for } \bar{\boldsymbol{\lambda}} \text{ defined in (41)} \right\}. \tag{42}$$

The reason that $\bar{\mathcal{C}}$ is a cone is that the data-inferred $\bar{\boldsymbol{\lambda}}$ for the input function f is exactly the same as for the input function cf, where c is any constant.

Algorithm 4 Adaptive ALG Based on Data-Inferred POSD Weights

Parameters: the bases $\{u_{\boldsymbol{k}}\}_{\boldsymbol{k}\in\mathbb{N}_0^d}$ and $\{v_{\boldsymbol{k}}\}_{\boldsymbol{k}\in\mathbb{N}_0^d}$; candidate sets $\mathcal{W}$ and $\mathcal{S}$; maximum smoothness degree, $k_{\max}$; order weights, $\boldsymbol{\Gamma}$; an inflation factor, $A > 1$; APP satisfying (11)
Input: a black-box function, f; an absolute error tolerance, $\varepsilon > 0$
Ensure: Error criterion (2) for the cone defined in (42)
Define the initial set of wavenumbers $\bar{\mathcal{K}}$ defined in (40)
Evaluate initial sample $\left\{\widehat{f}(\boldsymbol{k})\right\}_{\boldsymbol{k}\in\bar{\mathcal{K}}}$
Compute data-driven POSD weights, $\bar{\boldsymbol{\lambda}}$ according to (41)
Using these weights, $\bar{\boldsymbol{\lambda}}$, perform Algorithm 2 to obtain $\mathrm{ALG}(f,\varepsilon)$
Return $\mathrm{ALG}(f,\varepsilon)$

4.3 Numerical Examples

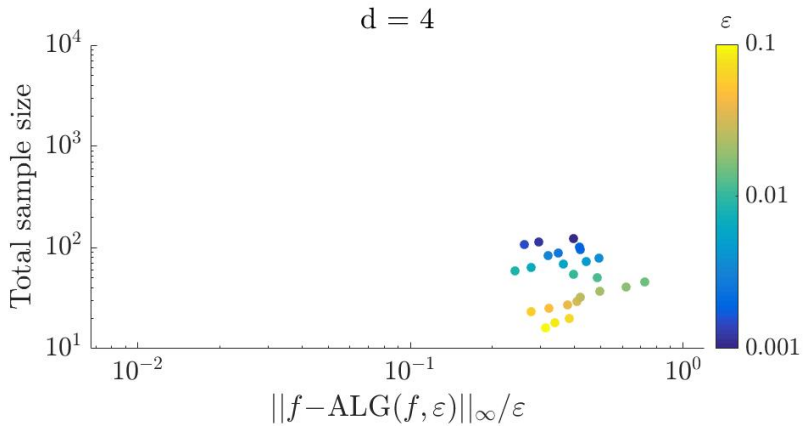
We now investigate the numerical performance of this adaptive algorithm using data-inferred POSD weights. For simplicity, only the case of $\rho = \infty$ and $\rho' = \tau = 1$ is considered in the following examples. Here, the basis functions $(u_{\boldsymbol{k}})_{\boldsymbol{k}\in\mathbb{N}_0^d}$ are Chebyshev polynomials in Section 1.1, and the solution operator is $\mathrm{SOL}(f) = f$ (i.e., function approximation). We note that $\|f - \mathrm{ALG}(f,\varepsilon)\|_\infty \le \|f - \mathrm{ALG}(f,\varepsilon)\|_{\mathcal{G}}$, so our error criterion (2) implies that $\|f - \mathrm{ALG}(f,\varepsilon)\|_\infty \le \varepsilon$.

The simulation set-up is as follows. The Fourier coefficients for input function f, $\{\hat{f}(\boldsymbol{k})\}_{\boldsymbol{k}\in\mathbb{N}_0^d}$, are randomly sampled as:

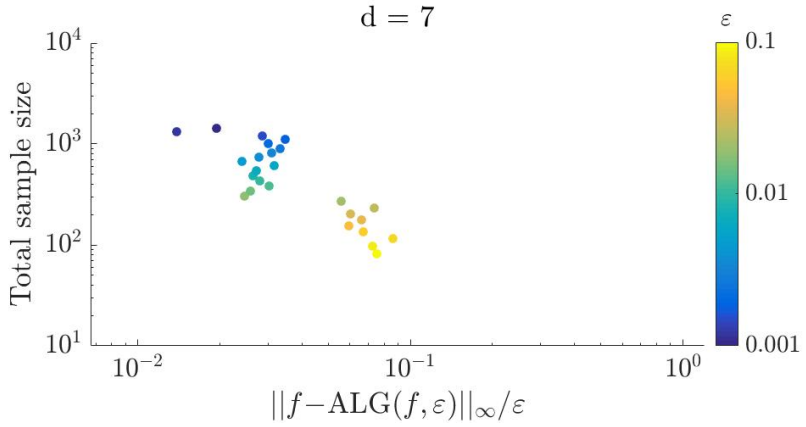
$$\hat{f}(\boldsymbol{k}) = Z_{\boldsymbol{k}}\,\Gamma^{\mathrm{tr}}_{\|\boldsymbol{k}\|_0} \prod_{\substack{\ell=1\\ k_\ell>0}}^{d} w^{\mathrm{tr}}_\ell s^{\mathrm{tr}}_{k_\ell}, \quad Z_{\boldsymbol{k}} \overset{i.i.d.}{\sim} \mathrm{Unif}[-1,1], \quad \boldsymbol{k}\in\mathbb{N}_0^d.$$

Here, $(w^{\mathrm{tr}}_\ell)_{\ell=1}^d = (1/L^2(\ell))_{l=1}^d$, $(\Gamma^{\mathrm{tr}}_k)_{k=1}^\infty \equiv 1$ and $(s^{\mathrm{tr}}_j)_{j=1}^{k_{\max}} = (1/j^4)_{j=1}^4$ are the true coordinate, order, and smoothness weights, and $Z_{\boldsymbol{k}}$ randomly sets the magnitude and sign of each coefficient. Moreover, $\big(L(\ell)\big)_{\ell=1}^d$ is a random permutation of $1,\ldots,d$ to ensure that the order of input variables does not necessarily reflect their order of importance. We also set $\boldsymbol{\Gamma} = \boldsymbol{\Gamma}^{\mathrm{tr}}$ in Algorithm 4 and use an inflation factor of $A = 1.1$.

Figures 1 (a) and (b) display the total required sample size from Algorithm 4, as a function of the error to tolerance ratio, $\|f - \mathrm{ALG}(f,\varepsilon)\|_\infty/\varepsilon$, in $d = 4$ and $d = 7$ dimensions, respectively. Each data point corresponds to a different error tolerance ε. A ratio $\|f - \mathrm{ALG}(f,\varepsilon)\|_\infty/\varepsilon$ close to, but not exceeding, one is desired, since this shows that our adaptive algorithm is successful. For $d = 4$, $\|f - \mathrm{ALG}(f,\varepsilon)\|_\infty/\varepsilon$ fluctuates around 0.4 for all choices of ε; for $d = 7$, this ratio begins at ≈ 0.1 for $\varepsilon = 0.1$, then decreases to ≈ 0.014 for $\varepsilon = 0.001$. This shows that our adaptive approximation algorithm works reasonably well. It appears slightly more effective in lower dimensions than in higher dimensions. A likely reason is that the underlying POSD structure can be more easily learned from a small pilot sample in lower dimensions than in higher dimensions.



(a) f is a $d = 4$-dim. function with random Fourier coefficients.



(b) f is a $d = 7$-dim. function with random Fourier coefficients.

Figure 1: Total required sample size as a function of error ratio $\|f - \mathrm{ALG}(f, \varepsilon)\|_\infty/\varepsilon$, with points colored by the absolute error tolerance level ε.

Acknowledgement

F. J. Hickernell, P. Kritzer, and S. Mak gratefully acknowledge the support of the Statistical and Applied Mathematical Sciences Institute year-long "Program on Quasi-Monte Carlo and High-Dimensional Sampling Methods for Applied Mathematics" through NSF-DMS-1638521. F. J. Hickernell also acknowledges the support of NSF-DMS-152268. F. J. Hickernell and P. Kritzer thank the RICAM Special Semester Program 2018 for support. P. Kritzer gratefully acknowledges support by the Austrian Science Fund (FWF) Project F5506-N26, which is part of the Special Research Program "Quasi-Monte Carlo Methods: Theory and Applications".

References

[1] R. Cools and D. Nuyens, editors. *Monte Carlo and Quasi-Monte Carlo Methods: MCQMC, Leuven, Belgium, April 2014*, volume 163 of *Springer Proceedings in Mathematics and Statistics*. Springer-Verlag, Berlin, 2016.

[2] Josef Dick, Frances Y Kuo, Quoc T Le Gia, Dirk Nuyens, and Christoph Schwab. Higher order QMC Petrov–Galerkin discretization for affine parametric operator equations with random field inputs. *SIAM Journal on Numerical Analysis*, 52(6):2676–2702, 2014.

[3] F. J. Hickernell and Ll. A. Jiménez Rugama. Reliable adaptive cubature using digital sequences. In Cools and Nuyens [1], pages 367–383. arXiv:1410.8615 [math.NA].

[4] F. J. Hickernell, Ll. A. Jiménez Rugama, and D. Li. Adaptive quasi-Monte Carlo methods for cubature. In J. Dick, F. Y. Kuo, and H. Woźniakowski, editors, *Contemporary Computational Mathematics — a celebration of the 80th birthday of Ian Sloan*, pages 597–619. Springer-Verlag, 2018.

[5] Ll. A. Jiménez Rugama and F. J. Hickernell. Adaptive multidimensional integration based on rank-1 lattices. In Cools and Nuyens [1], pages 407–422. arXiv:1411.1966.

[6] P. Kritzer and H. Woźniakowski. Simple characterizations of exponential tractability for linear multivariate problems. *J. Complexity*, 51:110–128, 2019.

[7] F. Y. Kuo, C. 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.*, 50:3351–3374, 2012.

[8] E. Novak and H. Woźniakowski. *Tractability of Multivariate Problems Volume I: Linear Information*. Number 6 in EMS Tracts in Mathematics. European Mathematical Society, Zürich, 2008.

[9] E. Novak and H. Woźniakowski. *Tractability of Multivariate Problems Volume II: Standard Information for Functionals*. Number 12 in EMS Tracts in Mathematics. European Mathematical Society, Zürich, 2010.

[10] E. Novak and H. Woźniakowski. *Tractability of Multivariate Problems Volume III: Standard Information for Operators*. Number 18 in EMS Tracts in Mathematics. European Mathematical Society, Zürich, 2012.

[11] J. F. Traub, G. W. Wasilkowski, and H. Woźniakowski. *Information-Based Complexity*. Academic Press, Boston, 1988.

[12] J. F. Traub and A. G. Werschulz. *Complexity and Information.* Cambridge University Press, Cambridge, 1998.

[13] A. G. Werschulz and H. Woźniakowski. A new characterization of (s,t)-weak tractability. *J. Complexity*, 38:68–79, 2017.

[14] C.F. Jeff Wu and Michael S. Hamada. *Experiments: Planning, Analysis, and Optimization.* John Wiley & Sons, 2009.

Authors' addresses:

Yuhan Ding
Department of Mathematics
Misericordia University
301 Lake Street, Dallas, PA 18704 USA

Fred J. Hickernell
Department of Applied Mathematics
Illinois Institute of Technology
RE 220, 10 W. 32nd Street, Chicago, IL 60616 USA

Peter Kritzer
Johann Radon Institute for Computational and Applied Mathematics (RICAM)
Austrian Academy of Sciences
Altenbergerstr. 69, 4040 Linz, Austria

Simon Mak
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology
755 Ferst Drive, Atlanta, GA 30332 USA