On the query complexity of real functionals

Hugo Férée, Mathieu Hoyrup
LORIA - Université de Lorraine & Inria
Nancy, France
Email: {hugo.feree,mathieu.hoyrup}@loria.fr

Walid Gomaa
Egypt-Japan University of Science and Technology
(on leave from Alexandria University)
Alexandria, Egypt
Email: walid.gomaa@ejust.edu.eg

***Abstract*—Recently Kawamura and Cook developed a framework to define the computational complexity of operators arising in analysis. Our goal is to understand the effects of complexity restrictions on the analytical properties of the operator. We focus on the case of norms over C[0,1] and introduce the notion of dependence of a norm on a point and relate it to the query complexity of the norm. We show that the dependence of almost every point is of the order of the query complexity of the norm. A norm with small complexity depends on a few points but, as compensation, highly depends on them. We characterize the functionals that are computable using one oracle call only and discuss the uniformity of that characterization.**

***Index Terms*—Computable analysis, computational complexity, oracle Turing machine, polynomial time computable functional, norm.**

I. Introduction

An approach to computable analysis is the so-called *Type-Two Theory of Effectivity (TTE)* which enables one to extend computability theory from discrete spaces to many continuous spaces arising in mathematical analysis [1], [2]. On the other side, computational complexity theory over continuous spaces is still in its infancy. A theory applicable to the space of real numbers has been developed by Ko and Friedman [3], [4] and has given many results. However, this theory is not readily extendible to “larger” spaces such as the space $\mathcal{C}[0,1]$ of continuous real functions defined over the unit interval, and a more general, abstract theory is still lacking. First approaches have been developed by Weihrauch [5] on metric spaces, and by Schröder [6] who argues that in order to express computational complexity in terms of first-order time functions (as in the discrete setting), one must restrict to σ-compact spaces. Recently Kawamura and Cook [7] developed a framework applicable to the space $\mathcal{C}[0,1]$ (which is not σ-compact), using higher-order complexity theory and in particular second-order polynomials. In particular their theory enables them to prove uniform versions of older results about the complexity of solving differential equations, as well as new results [8], [9].

Our general goal is to study the complexity of operators defined over $\mathcal{C}[0,1]$ and particularly to understand the implication that complexity restrictions have on the analytical properties of the operator. Looking for connections between computation and analysis is an old and fruitful field of investigation. The most famous example is the fact that on many sorts of topological spaces, a computable function must be continuous and, further, the continuous functions are exactly the functions that are computable relative to some oracle. Topology is always hidden behind computability notions, which explains why higher-order recursion theory and computable analysis are intimately related to descriptive set theory. Such a correspondence between computation and topology also comes up in complexity theory: bounds on the resources available during the computation are reflected in analytical constraints over the functions to be computed, confining them to live in a smaller space. Illustrations of this principle appear in several places. Townsend [10] characterized relativized polynomial classes of type-2 relations by means of topological notions: for instance if A is an alphabet then a subset of $(A^*)^{A^*}$ is in Σ_1^P relative to some oracle (written $\underline{\Sigma}_1^P$ in his paper) if and only if it is a “polynomially open set”, in a certain sense. In analysis, a real function $f : [0,1] \to \mathbb{R}$ is polynomial-time computable relative to an oracle if and only if it has a polynomial modulus of uniform continuity [4].

This paper is a first study along these lines of the complexity theory over $\mathcal{C}[0,1]$ recently developed by Kawamura and Cook, in which such correspondences are not known to date. Some typical questions are: what are the topological implications of limiting the resources of a machine computing a functional? what is the class of functionals that are computable in polynomial time relative to an oracle? Observe that a bound on a resource such as time imposes two conditions on the machine operation: it restricts its internal computation time as well as the information queried to the oracle. We mostly concentrate on the second constraint, expressed in terms of *query complexity*.

The potential limitations imposed by resource bounds on the computation of functionals over $\mathcal{C}[0,1]$ come from the representation of input functions $f \in \mathcal{C}[0,1]$ which does not give a global view on f but local information only: the whole function is not approximated, for example, by piecewise linear functions, but rather the oracle evaluates the function on demand at queried points, in addition to giving a modulus of continuity of f to the machine. The penny-pinching character of the oracle describing the input is due to the huge amount of information a function contains (one can see [11] for a quantitative analysis of this fact). As a result, little can be known about f in polynomial time, and classical operators such as taking the supremum or the integral of a function are not polynomial-time computable because a machine needs exponential time to evaluate its input on the whole interval.

In this paper, we do not consider general functionals but we focus on the simpler case of norms over $\mathcal{C}[0,1]$. The general problem is: what are the analytical effects of bounding the computational resources to compute a norm? As explained above, bounding the allowed number of queries to the oracle prevents the machine to evaluate its input $f \in \mathcal{C}[0,1]$ at too many points, hence a norm with low query complexity should "depend" on a small set. We formalize this idea by introducing two notions: the quantitative notion of *dependence of a norm on a point* and the qualitative notion of *relevance of a point w.r.t. a norm*. Intuitively, a norm $\|.\|$ has high dependence on a point if changing f around that point results in a big change in the value $\|f\|$. We then show how the query complexity of the norm impacts on these properties: a norm with low complexity depends on a small set but, as compensation, the dependence on that set is very high.

We also investigate the extreme case when only one oracle call is allowed to the machine computing a functional and obtain a characterization. Surprisingly, the argument to obtain such a characterization is much more involved than expected. It contains subtleties that make it non-uniform in terms of complexity.

The paper is organized as follows. In Section II we present the background on complexity in analysis needed for our results. In Section III we formalize the notions of *dependence* of a norm on a point and of *relevant* points w.r.t. a norm. We then show how they reflect the query complexity of the norm. In Section IV we characterize the class of functionals that are computable by an oracle Turing machine making only one query to the oracle. In Section V we conclude the paper with open questions to be investigated in the future.

II. Definitions and Preliminary Results

A. Notations and basic definitions

Σ denotes the alphabet $\{0,1\}$. The length of a finite word u over Σ is denoted by $|u|$.

Let $A \subseteq [0,1]$ and $\beta \in [0,1]$. The distance of β to A is $d(\beta, A) = \inf\{|\alpha - \beta| : \alpha \in A\}$. Given $r > 0$, the neighborhood of A of radius r is $N(A,r) = \{\beta \in [0,1] : d(\beta, A) < r\}$. If $\alpha \in [0,1]$, we simply write $N(\alpha, r) = N(\{\alpha\}, r) = \{\beta \in [0,1] : |\alpha - \beta| < r\}$. We will also consider the closed neighborhood $\overline{N}(A,r) = \{\beta : d(\beta, A) \leq r\}$.

We assume the space $\mathcal{C}[0,1]$ of continuous functions from $[0,1]$ to $\mathbb{R}$ with the usual structure of real vector space. The uniform norm is defined by $\|f\|_\infty = \max_{x\in[0,1]} |f(x)|$. The L^1-norm is defined by $\|f\|_1 = \int_0^1 |f(x)| \,\mathrm{d}x$.

If $f \in \mathcal{C}[0,1]$ then the ***support*** of f is $\mathrm{Supp}(f) = \{x \in [0,1] : f(x) \neq 0\}$. Observe that $\mathrm{Supp}(f)$ is an open set. We say that f is ***supported*** on a set $A \subseteq [0,1]$ if $\mathrm{Supp}(f) \subseteq A$. $h_{\alpha,r}$ denotes the maximal 1-Lipschitz function supported on $N(\alpha, r)$.

$\mathrm{Lip}_1 \subseteq \mathcal{C}[0,1]$ denotes the set of 1-Lipschitz functions from $[0,1]$ to $\mathbb{R}$.

A norm F over $\mathcal{C}[0,1]$ is ***weaker*** than a norm G if there is a constant c such that $F(f) \leq c \cdot G(f)$ for all $f \in \mathcal{C}[0,1]$, or equivalently the functional $F : \mathcal{C}[0,1] \to \mathbb{R}$ is continuous w.r.t. the topology of the norm G.

B. Polynomial time computable functionals

We briefly recall the formalism of [7].

1) Oracle Turing machine: An oracle Turing machine $\mathcal{M}$ taking as input a finite string $u \in \Sigma^*$ and consulting an oracle given by a function $\varphi : \Sigma^* \to \Sigma^*$ is denoted $\mathcal{M}^\varphi(u)$.

A function $\varphi : \Sigma^* \to \Sigma^*$ is ***regular*** if $|u| \leq |v|$ implies $|\varphi(u)| \leq |\varphi(v)|$ for all $u, v \in \Sigma^*$. The ***size*** of a regular function φ is the function $|\varphi| : \mathbb{N} \to \mathbb{N}$ defined by $|\varphi|(n) = |\varphi(0^n)|$ (as φ is regular, 0^n can be replaced by any word of length n).

The pairing of two regular functions $\varphi, \psi : \Sigma^* \to \Sigma^*$ is the regular function $\langle \varphi, \psi \rangle$ defined by $\langle \varphi, \psi \rangle(0u) = \varphi(u)10^{|\psi(u)|}$ and $\langle \varphi, \psi \rangle(1u) = \psi(u)10^{|\varphi(u)|}$.

2) Polynomial time oracle Turing machine: ***Second-order polynomials*** are defined inductively in the following way: every positive integer is a second-order polynomial, every first-order variable n is a second-order polynomial, if P and Q are second-order polynomials then so are $P+Q$, PQ and $X(P)$ where X is a second-order variable.

An oracle Turing machine $\mathcal{M}$ ***runs in polynomial time*** if there is a second-order polynomial $P(n, X)$ such that for any regular function φ and input $u \in \Sigma^*$, $\mathcal{M}^\varphi(u)$ halts in at most $P(|u|, |\varphi|)$ steps.

3) Representation of $\mathcal{C}[0,1]$: For $n \in \mathbb{N}$, let $\mathbb{D}_n = \{\frac{p}{2^n} : p \in \mathbb{N}, 0 \leq p \leq 2^n\}$ and $\mathbb{D} = \bigcup_n \mathbb{D}_n$ be the set of dyadic rational numbers in the interval $[0,1]$. Every string $u \in \Sigma^*$ represents a dyadic rational d_u whose binary expansion is $0.u$.

A ***modulus of continuity*** of a function $f \in \mathcal{C}[0,1]$ is a function $\mu : \mathbb{N} \to \mathbb{N}$ such that if $|x-y| \leq 2^{-\mu(n)}$ then $|f(x) - f(y)| \leq 2^{-n}$.

An ***approximation function*** of f is a function $f_\mathbb{D} : \mathbb{D} \times \mathbb{N} \to \mathbb{D}$ such that $|f_\mathbb{D}(d,n) - f(d)| \leq 2^{-n}$. We can assume w.l.o.g. that $f_\mathbb{D}(d,n) \in \mathbb{D}_n$.

We represent μ by the function $\overline{\mu}(u) = 0^{\mu(|u|)}$. We represent $f_\mathbb{D}$ by the function $\overline{f_\mathbb{D}} : \Sigma^* \times \Sigma^* \to \Sigma^*$ defined by $\overline{f_\mathbb{D}}(u,v) = w$ such that $d_w = f_\mathbb{D}(d_u, |v|)$.

If μ is a modulus of continuity of f and $f_\mathbb{D}$ an approximation function of f then $\langle \overline{\mu}, \overline{f_\mathbb{D}} \rangle$ is a ***representation*** of f.

A functional $F : \mathcal{C}[0,1] \to \mathbb{R}$ is computable if there is an oracle Turing machine $\mathcal{M}$ such that for any $f \in \mathcal{C}[0,1]$, any representation φ of f and any $n \in \mathbb{N}$ written in unary notation, $\mathcal{M}^\varphi(n)$ halts and outputs a dyadic number d such that $|F(f) - d| \leq 2^{-n}$.

Definition II.1. A functional $F : \mathcal{C}[0,1] \to \mathbb{R}$ is polynomial-time computable if it is computable by an oracle Turing machine that runs in polynomial time.

Example II.1. Assume $(q_n)_{n\in\mathbb{N}}$ is a polynomial-time computable enumeration of the dyadic rational numbers in the interval $[0,1]$, i.e. there is a polynomial-time computable function $\psi : \Sigma^* \to \Sigma^*$ such that $q_n = d_{\psi(0^n)}$. Define the functional

$$F_0(f) = \sum_{n\in\mathbb{N}} \frac{|f(q_n)|}{2^n}.$$

F_0 is a norm over $\mathcal{C}[0,1]$. It can be easily verified that F_0 can be computed by a machine with computational time bounded by a second-order polynomial in terms of the size of (a representation of) f and the precision parameter.

A ***relativized*** oracle Turing machine is an oracle Turing machine that has access to an auxiliary oracle $A \in \{0,1\}^\mathbb{N}$ and queries $A(n)$ by writing the binary expansion of n on an extra query tape. The representation of functions in $\mathcal{C}[0,1]$ is natural in the precise sense that it is *admissible*: a functional $F : \mathcal{C}[0,1] \to \mathbb{R}$ is computable by a relativized oracle Turing machine if and only if it is continuous w.r.t. the topology of the uniform norm (see [1] for precise results on admissibility of representations). In particular, a norm is computable by relativized oracle Turing machines if and only if it is weaker than the uniform norm.

III. Norms

Ko [12] introduced the class $\mathsf{NP}_\mathbb{R}$ of NP real numbers and showed that it coincides with the class of maximum values of polynomial time computable functions over $[0,1]$. The separation problem $\mathsf{P}_\mathbb{R} = \mathsf{NP}_\mathbb{R}$ lies between the problems $\mathsf{P} = \mathsf{NP}$ and $\mathsf{EXP} = \mathsf{NEXP}$. Friedman [13] obtained similar results for the integral values of polynomial time computable functions. These results show that separating complexity classes of real numbers is as difficult as in the case of sets of strings. However the situation is different for complexity classes of functionals: Ko and Friedman [3] proved that the functional mapping $f \in \mathcal{C}[0,1]$ to $\max\{f(x) : x \in [0,1]\}$ is not polynomial-time computable (in a certain sense that is weaker than the one of Cook and Kawamura used here, Definition II.1), while there is a way to express the fact that it is an NP functional. Similarly, the functional mapping $f \in \mathcal{C}[0,1]$ to $\int_0^1 f(x)\,\mathrm{d}x$ can be proved not to be polynomial-time computable.

The main reason why lower bounds are much easier to achieve in the case of functionals lies in the fact that time restrictions not only bound the internal computation time of a machine but also *limit its access to the input* – it contrasts with the classical setting where time restrictions usually do not prevent the machine to access its input entirely. Hence a machine running in polynomial time does not have time enough to evaluate the input function on a large set so it can hardly distinguish between some very different functions. It suggests that if a machine computes a norm, bounding its computation time must have implications on the topology induced by the norm, which raises the following question: what are the topologies induced by polynomial-time computable norms?

We already know that they cannot be equivalent to the topologies of the uniform norm and the L^1 norm. One can prove that (i) a polynomial-time computable norm is incomparable with the L^1 norm, (ii) it cannot be complete, (iii) the notion of convergence it induces is incomparable with convergence in probability. Hence polynomial-time computable norms live in a reduced space, outlined by these properties. Our goal is to circumscribe more accurately complexity classes of norms by having a finer look into their analytical properties. The subsequent notions of *dependence* of a norm on a point and of *relevance* of a point w.r.t. a norm will make it possible.

A. *Dependence of a norm on a point*

Let F be a norm over $\mathcal{C}[0,1]$ and $\alpha \in [0,1]$. Intuitively, one would say that the norm of a function depends on its value at α if modifying it at α *only* changes its norm. But two problems appear: first we consider continuous functions so modifying a function f at α is not possible without modifying f also around α; second, if $f = 0$ then modifying f anywhere will automatically change $F(f) = 0$ to some positive value, as F is a norm. To get around these issues, the solution consists in defining a quantitative dependence notion that relates the size of the neighborhood of α on which f is modified to the alteration of the value $F(f)$. As F is a norm, it has a certain homogeneity that allows us to focus on the function $f = 0$ only.

From now on we assume that F is a norm over $\mathcal{C}[0,1]$ that is weaker than the uniform norm.

Definition III.1. Let F be a norm on $\mathcal{C}[0,1]$ that is weaker than the uniform norm and $\alpha \in [0,1]$. The ***dependence*** of F on $\alpha \in [0,1]$ is the function $\mathrm{d}_{F,\alpha} : \mathbb{N} \to \mathbb{R}$ defined by

$$\begin{aligned}\mathrm{d}_{F,\alpha}(n) &= \sup\{l : \exists f \in \mathrm{Lip}_1, \mathrm{Supp}(f) \subseteq N(\alpha, 1/l) \\ &\qquad \text{and } F(f) > 2^{-n}\} \\ &= \inf\{l : \forall f \in \mathrm{Lip}_1, \mathrm{Supp}(f) \subseteq N(\alpha, 1/l) \\ &\qquad \text{implies } F(f) \leq 2^{-n}\}.\end{aligned}$$

Observe that the first set is downward closed, so the two definitions are equivalent.

For every α and n, $\mathrm{d}_{F,\alpha}(n) \geq 1$. Indeed, if $l < 1$ then every function is supported on $N(\alpha, 1/l) = [0,1]$. For every α, $\mathrm{d}_{F,\alpha}$ is nondecreasing, unbounded and $\mathrm{d}_{F,\alpha}(n) \leq 2c \cdot 2^n$ if $F \leq c \cdot \|.\|_\infty$.

One easily checks that the set $\{l : \forall f \in \mathrm{Lip}_1, \mathrm{Supp}(f) \subseteq N(\alpha, 1/l) \text{ implies } F(f) \leq 2^{-n}\}$ is closed, so the infimum is a minimum.

We recall that $h_{\alpha,r}$ is the maximal 1-Lipschitz function supported on $N(\alpha, r)$. When the norm is monotonic, i.e. $|f| \leq |g|$ implies $F(f) \leq F(g)$,

$$\mathrm{d}_{F,\alpha}(n) = \sup\left\{l : F(h_{\alpha,1/l}) > 2^{-n}\right\}.$$

Let us illustrate Definition III.1 on a few examples.

Example III.1. Let F be the uniform norm. For all $\alpha \in [0,1]$ and $n \in \mathbb{N}$, $2^n \leq \mathrm{d}_{F,\alpha}(n) \leq 2^{n+1}$. The dependence of a point w.r.t. to the uniform norm is maximal.

Example III.2. Let F be the L^1 norm. For all $\alpha \in [0,1]$ and $n \in \mathbb{N}$, $2^{\frac{n-1}{2}} \leq \mathrm{d}_{F,\alpha}(n) \leq 2^{\frac{n+1}{2}}$. While exponential, the dependence of a point w.r.t. the L^1 norm is smaller than for the uniform norm.

Example III.3. Let $Q = \{q_0, q_1, \dots\} \subseteq [0,1]$ be a countable dense set and $F(f) = \sum_i 2^{-i}|f(q_i)|$. For all i and $n \geq i$, $\mathrm{d}_{F,q_i}(n) \geq 2^{n-i}$: intuitively, the norm of a function depends much on its value on Q. Moreover, it does not depend much on points that are "far away" from the points q_i and the dependence on almost every point (in the sense of Lebesgue measure) is bounded by a polynomial. Indeed, let α be such that $|\alpha - q_i| \geq \epsilon/i^2$ for all i: one has $\mathrm{d}_{F,\alpha}(n) \leq n^2/\epsilon$ for all n. To show that, let $f \in \mathrm{Lip}_1$ with $\mathrm{Supp}(f) \subseteq N(\alpha, \epsilon/n^2)$. f is null on $\{q_0, \dots, q_n\}$ so $F(f) = \sum_{i>n} 2^{-i}|f(q_i)| \leq 2^{-n}$ as $\|f\|_\infty \leq 1$. Observe that the set of such α has measure $\geq 1 - \epsilon\pi^2/3$ which can be made arbitrarily close to 1. So for almost all α, $\mathrm{d}_{F,\alpha}$ is bounded by a polynomial (which depends on α).

Example III.4. Given $a, b \in [0,1]$ such that $0 < b - a < 1/2$, we construct a compact set $K_{a,b}$ in the following way. It is a variation of the third-middle Cantor set. To each $w \in \{0,1\}^*$ is associated a closed interval I_w such that $I_{w0} \cup I_{w1} \subseteq I_w$. $K_{a,b}$ is defined as the intersection over $n \in \mathbb{N}$ of $\bigcup_{w \in \{0,1\}^n} I_w$. If ϵ is the empty word then let $I_\epsilon = [a,b]$. Given I_w we define I_{w0} and I_{w1}. I_{w0} and I_w have the same left endpoint, I_{w1} and I_w have the same right endpoint and $|I_w0| = |I_{w1}| = |I_w|^2$. One has $|I_w| = (b-a)^{2^{|w|}} < 2^{-2^{|w|}}$.

Given p, let n be such that $2^{n-1} < p \leq 2^n$. Let $Q_n = \{q_1, \dots, q_{2^n}\}$ be the left endpoints of the intervals I_w for $w \in \{0,1\}^n$. As $|I_w| < 2^{-2^n} \leq 2^{-p}$, Q_n is 2^{-p} close to $K_{a,b}$ in the Hausdorff metric. The size of Q_n is $2^n < 2p$ which is polynomial in p.

Now we define a measure $\mu_{a,b}$ supported on $K_{a,b}$. $K_{a,b}$ is homeomorphic to the Cantor space: a point $x \in K_{a,b}$ is mapped to the infinite sequence $s_0 s_1 \dots$ such that $x \in I_{s_0 \dots s_n}$ for all n. $\mu_{a,b}$ is defined as the image of the uniform measure over the Cantor space under this mapping. For instance one has $\mu_{a,b}(I_w) = 2^{-|w|}$.

The point is that the functional $\mathcal{C}[0,1] \to \mathbb{R}$ which maps f to $\int |f| \, \mathrm{d}\mu_{a,b}$ is polynomial-time computable. Indeed, to evaluate the integral at precision 2^{-p}, let n be such that $2^{n-1} < \mu_f(p+1) \leq 2^n$, evaluate f at Q_n with precision 2^{-p-1} and output the average of the values.

Now we define a measure μ supported on $[0,1]$ and such that the $L^1(\mu)$-norm is polynomial-time computable. Consider the set of dyadic intervals $[k/2^n, k + 1/2^n]$ with $n \geq 2$ and $0 \leq k < 2^n$. Let $[a_i, b_i]$ be the enumeration of this set derived from the enumeration of pairs (k, n) in lexicographic order. Let $\mu = \sum_i 2^{-i} \mu_{a_i, b_i}$ and F be the $L^1(\mu)$ norm.

Claim III.1. If $\alpha \in \bigcup_i K_{a_i,b_i}$ then $\mathrm{d}_{F,\alpha}(n) \geq c\frac{2^n}{n}$ for some c and all n.

Proof: Let $\alpha \in K_{a_i,b_i}$. We prove that $\mathrm{d}_{F,\alpha}(n) \geq c\frac{2^{n-i-2}}{n}$ for some c and all n.

Let w be such that $\alpha \in I_w$. As $h_{\alpha,1/l}$ is 1-Lipschitz, $h_{\alpha,1/l} \geq \frac{1}{2l}$ on I_w if $|I_w| \leq \frac{1}{2l}$ so $\int h_{\alpha,1/l}\,\mathrm{d}\mu_i \geq \frac{2^{-|w|}}{2l}$. As $|I_w| < 2^{-2^{|w|}}$, $|I_w| \leq \frac{1}{2l}$ is satisfied as soon as $|w| \geq \log\log(2l)$. Let then $|w| = \lceil \log\log(2l) \rceil$: $\int h_{\alpha,1/l}\,\mathrm{d}\mu_i > \frac{1}{4l\log(2l)}$. If $l = \frac{2^{n-i-2}}{n}$ then $\int h_{\alpha,1/l}\,\mathrm{d}\mu_i > 2^{i-n}$ so $F(h_{\alpha,1/l}) > 2^{-n}$, so $\mathrm{d}_{F,\alpha}(n) \geq \frac{2^{n-i-2}}{n}$. The proof actually works only if n is sufficiently large so that $l > 1/2$, so one get the result for all n by choosing a suitable constant c. ■

Proposition III.1. *For each $n \in \mathbb{N}$, the function $\alpha \mapsto \mathrm{d}_{F,\alpha}(n)$ is continuous. Moreover one has for all α, β,*

$$|\mathrm{d}_{F,\alpha}(n) - \mathrm{d}_{F,\beta}(n)| \leq |\alpha - \beta| \mathrm{d}_{F,\alpha}(n) \mathrm{d}_{F,\beta}(n).$$

Proof: Let $l < \mathrm{d}_{F,\alpha}(n)$, $f \in \mathrm{Lip}_1$ with $\mathrm{Supp}(f) \subseteq N(\alpha, 1/l)$ and $F(f) > 2^{-n}$. $\mathrm{Supp}(f) \subseteq N(\beta, 1/l + |\alpha - \beta|)$ so $\mathrm{d}_{F,\beta}(n) \geq l/(1 + l|\alpha - \beta|)$. As it is true for every $l < \mathrm{d}_{F,\alpha}(n)$, $\mathrm{d}_{F,\beta}(n) \geq \mathrm{d}_{F,\alpha}(n)/(1 + \mathrm{d}_{F,\alpha}(n)|\alpha - \beta|)$ so $\mathrm{d}_{F,\alpha}(n) - \mathrm{d}_{F,\beta}(n) \leq |\alpha - \beta| \mathrm{d}_{F,\alpha}(n)\mathrm{d}_{F,\beta}(n)$. Exchanging α and β gives the result. As $\mathrm{d}_{F,\alpha}(n)$ is bounded by $c \cdot 2^n$ for some c, it implies that $\alpha \mapsto \mathrm{d}_{F,\alpha}(n)$ is continuous. ■

Proposition III.2. *If a norm F is weaker than a norm G then there exists k such that for all α and n, $\mathrm{d}_{F,\alpha}(n) \leq \mathrm{d}_{G,\alpha}(n + k)$.*

Proof: Straightforward from the definition, using k such that $F \leq 2^k G$. ■

However, non equivalent norms may not be distinguished by their dependence functions. If F is a norm then $F(f) + |f(0) - f(1)|$ and $F(f) + |f(0)| + |f(1)|$ are generally non-equivalent but have exactly the same dependence functions.

Definition III.2. The ***maximal dependence*** of a norm F is the function $\mathrm{D}_F : \mathbb{N} \to \mathbb{R}$ defined by

$$\mathrm{D}_F(n) = \max_{\alpha \in [0,1]} \mathrm{d}_{F,\alpha}(n).$$

For every norm F that is weaker than the uniform norm, there exists a constant c such that $\mathrm{D}_F(n) \leq c \cdot 2^n$. The next result gives a lower bound, which is optimal as it is reached by the L^1 norm.

Proposition III.3. *For every norm F there exists $c > 0$ such that $\mathrm{D}_F(n) \geq c \cdot 2^{\frac{n}{2}}$ for all n.*

Proof: Let $\mathbb{N} \in \mathbb{N} \setminus \{0\}$. On easily checks that the sum $\sum_{i=0}^{N} h_{i/N,1/N}$ equals the constant function $\frac{1}{N}$. By triangular inequality, $\frac{F(1)}{N} \leq \sum_i F(h_{i/N,1/N}) \leq (N + 1)\max_i F(h_{i/N,1/N})$ so there exists $i \in \{0, \dots, N\}$ such that $F(h_{i/N,1/N}) \geq \frac{F(1)}{N(N+1)}$. Let $0 < c < \sqrt{F(1)}/4$ and $N = \lceil c2^{\frac{n}{2}} \rceil$. One has $\frac{F(1)}{N(N+1)} > \frac{F(1)}{(N+1)^2} > 2^{-n}$ if n is sufficiently large, so $\mathrm{D}_F(n) \geq \mathrm{d}_{F,\frac{i}{N}}(n) \geq N \geq c2^{\frac{n}{2}}$. Changing c one can obtain the inequality for all n. ■

As we will see later (Proposition III.4), there exists a point whose dependence is at least $c \cdot 2^{\frac{n}{2}}$.

Each point of high dependence has an influence on the value of the norm, but does not usually determine that value. However, the next theorem shows that the whole set of points of high dependence taken together determine the value of the norm up to some precision. Let

$$\mathrm{R}_{n,l} = \{\alpha : \mathrm{d}_{F,\alpha}(n) \geq l\}.$$

As $\alpha \mapsto \mathrm{d}_{F,\alpha}(n)$ is continuous, $\mathrm{R}_{n,l}$ is a closed set.

Theorem III.1. *Let $l \leq \mathrm{D}_F(n)$. If $f \in \mathrm{Lip}_1$ is null on $\mathrm{R}_{n,l}$ then $F(f) \leq l^2 2^{-n+6}$.*

Proof idea: Decompose f as a sum of small functions supported on intervals of length at most $2/l$. Each small function is supported on a small neighborhood of radius $1/l$ of some point α satisfying $\mathrm{d}_{F,\alpha}(n) \leq l$, so the norm of each small function is at most 2^{-n}. The number of small functions is quadratic in l, which gives the result. ■

This result gives a strategy to evaluate the norm of a function. Indeed, let $\epsilon > 0$ and $l = 2^{\frac{n-7}{2}}\sqrt{\epsilon}$. Theorem III.1 implies that if $f, g \in \mathrm{Lip}_1$ coincide on $\mathrm{R}_{n,l}$ then $|F(f) - F(g)| \leq F(f - g) \leq \epsilon$ (applying the theorem to $\frac{f-g}{2} \in \mathrm{Lip}_1$) so in order to know the norm of f up to ϵ, it is sufficient to evaluate f on $\mathrm{R}_{n,l}$.

B. Relevant points

Intuitively, the norm of a function f depends on the values of f on points of high dependence, i.e. points α whose function $\mathrm{d}_{F,\alpha}$ is large. Several questions arise: at which points a machine computing a norm should evaluate its input function? Can we separate the points into two classes, the points that are relevant to compute the norm and the points that are not, according to the growth of their dependence function? To answer the second question, we need to find a threshold. The example of the L^1 norm shows that one cannot hope in general to have points whose dependence function grows faster than $2^{\frac{n}{2}}$, so the threshold should be at most of the order of $2^{\frac{n}{2}}$. Proposition III.3 suggests (but does not imply) that points whose dependence is at least $2^{\frac{n}{2}}$ might always exist. It is indeed the case as Proposition III.4 below shows. We can then choose $2^{\frac{n}{2}}$ as a threshold that separates $[0, 1]$ into the two classes of *relevant* and *irrelevant* points. The interest of these notions will be demonstrated by Theorem III.2, III.3 and III.4.

Definition III.3. Let F be a norm over $\mathcal{C}[0,1]$. A point α is ***relevant*** w.r.t. F if there exists $c > 0$ such that $\mathrm{d}_{F,\alpha}(n) \geq c \cdot 2^{\frac{n}{2}}$ for all n.

First observe that the set $\mathscr{R}$ of relevant points is a countable union of growing compact sets,

$$\mathscr{R}_k = \{\alpha : \forall n, \mathrm{d}_{F,\alpha}(n) \geq 2^{\frac{n}{2}-k}\} = \bigcap_n \mathrm{R}_{n,2^{\frac{n}{2}-k}}$$

$$\mathscr{R} = \bigcup_k \mathscr{R}_k.$$

As $\mathrm{d}_{F,\alpha}(n) \geq 1$ for all α and n, $\mathscr{R}_k = \{\alpha : \forall n > 2k, \mathrm{d}_{F,\alpha}(n) \geq 2^{\frac{n}{2}-k}\}$.

Proposition III.3 can be strengthened: relevant points always exist and are dense, which fits with the intuition that a norm should "look everywhere" to separate different functions.

Proposition III.4. *Let F be a norm. The set of relevant points is dense.*

Proof: Let $p \in \mathbb{N} \setminus \{0\}$ and $a \in [2^{-p}, 1-2^{-p}]$. We construct a relevant point $\alpha \in \overline{N}(a, 2^{-p})$. Let $c \in \mathbb{N}$ be such that $F(h_{a,2^{-p}}) > 2^{-c}$. α will be the limit of a sequence α_n defined by induction on n, satisfying $F(h_{\alpha_n,2^{-n-p}}) > 2^{-2n-c}$. Start with $\alpha_0 = a$.

Given α_n, we decompose $h_{\alpha_n,2^{-n-p}}$ as a sum of four functions (see Figure 1). Let $\beta_1 = \beta_2 = \alpha_n$, $\beta_3 = \alpha_n - 2^{-n-p-1}$ and $\beta_4 = \alpha_n + 2^{-n-p-1}$. $h_{\alpha_n,2^{-n-p}} = \sum_{i=1}^4 h_{\beta_i,2^{-n-p-1}}$ so by triangular inequality there exists $\alpha_{n+1} \in \{\beta_1,\beta_2,\beta_3,\beta_4\}$ such that $F(h_{\alpha_{n+1},2^{-n-p-1}}) \geq F(h_{\alpha_n,2^{-n-p}})/4 > 2^{-2n-2-c}$.

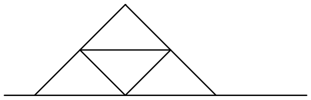


Fig. 1: $h_{\alpha_n,2^{-n-p}} = \sum_{i=1}^4 h_{\beta_i,2^{-n-p-1}}$

As $|\alpha_n - \alpha_{n+1}| \leq 2^{-n-p-1}$, α_n is a Cauchy sequence, let α be its limit. One has $|\alpha - \alpha_n| \leq 2^{-n-p}$ so $h_{\alpha_n,2^{-n-p}}$ is supported on $N(\alpha, 2^{-n-p+1})$. As $F(h_{\alpha_n,2^{-n-p}}) > 2^{-2n-c}$, $\mathrm{d}_{F,\alpha}(2n+c) \geq 2^{n+p-1}$. Let $k \geq \frac{c+3}{2} - p$. For every $n > 2k > c$, $\mathrm{d}_{F,\alpha}(n) \geq 2^{\frac{n}{2}-k}$. ■

Let us illustrate the notion of relevant point on a few examples.

Example III.5. Let F be the uniform norm. Every point is relevant, and $\mathscr{R} = \mathscr{R}_0 = [0,1]$.

Example III.6. Let F be the L^1 norm. Every point is relevant, and $\mathscr{R} = \mathscr{R}_1 = [0,1]$.

Example III.7. Let $F(f) = \sum_i 2^{-i}|f(q_i)|$. Every q_i is relevant and $\mathscr{R}_i$ contains $\{q_0, \dots, q_{2i}\}$. Whether $\mathscr{R}$ contains only the numbers q_i depends on the way they are distributed in the unit interval:

1) let us consider the *canonical enumeration of the dyadic rationals*, defined in the following way: for $i = 2^n + k$ with $0 \leq k < 2^n$, let $q_i = (2k+1)2^{-n}$. The important feature of this enumeration is that dyadic rationals are far from each other, in terms of their indices: if $i < j$ then $|q_i - q_j| \geq \frac{1}{j}$. Indeed, let n be such that $2^n \leq j < 2^{n+1}$: $|q_i - q_j| \geq 2^{-n} \geq \frac{1}{j}$.

   Lemma III.1. *If $(q_i)_{i\in\mathbb{N}}$ is the canonical enumeration of the dyadic rationals then $\mathscr{R} = \mathbb{D}$. Moreover if $\alpha \notin \mathbb{D}$ then $\mathrm{d}_{F,\alpha}(n) \leq 2(n+1)$ for infinitely many n.*

   Proof: Let $\alpha \notin \mathbb{D}$ and $n_0 \in \mathbb{N}$. Assume $\mathrm{d}_{F,\alpha}(n) > 2(n+1)$ for all $n \geq n_0$.
   First, $F(h_{\alpha,\frac{1}{2(n_0+1)}}) > 2^{-n_0}$ so there exists $i \leq n_0$ such that $|\alpha - q_i| < \frac{1}{2(n_0+1)}$ otherwise the first $n_0 + 1$ terms in the sum defining F are null and the norm is at most 2^{-n_0}. Take $i \leq n_0$ minimizing $|\alpha - q_i|$. As $\alpha \neq q_i$ there exists $n > n_0$ such that $|\alpha - q_i| \geq \frac{1}{2(n+1)}$. Take n minimal. Again, $F(h_{\alpha,\frac{1}{2(n+1)}}) > 2^{-n}$ so there exists $j \leq n$ such that $|\alpha - q_j| < \frac{1}{2(n+1)}$. As $|\alpha - q_j| < |\alpha - q_i|$, $j > n_0 \geq i$. Now, $|q_i - q_j| \leq |\alpha - q_i| + |\alpha - q_j| < \frac{1}{2n} + \frac{1}{2(n+1)} < \frac{1}{n} \leq \frac{1}{j}$, which gives a contradiction. ■

   More generally, and by the same argument, if f is non-increasing and $i < j$ implies $|q_i - q_j| \geq f(j)$ then if $\alpha \notin \mathbb{D}$, $\mathrm{d}_{F,\alpha}(n) \leq \frac{f(n+1)}{2}$ for infinitely many n. In particular if $f(n) = o(2^{\frac{n}{2}})$ then $\mathscr{R} = \mathbb{D}$.

2) we now consider the case when the sequence $(q_i)_{i\in\mathbb{N}}$ accumulates quickly at a point $\alpha \notin \mathbb{D}$, in which case α may be relevant. For instance, if $|q_{2i} - \alpha| < 2^{-2i}$ then one easily checks that $\mathrm{d}_{F,\alpha}(n) \geq 2^{\frac{n}{2}-2}$ for all n, so α is relevant.

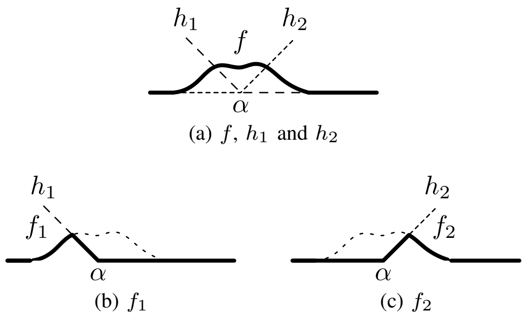
The terminology is justified by the next result: the value $F(f)$ up to some precision (decreasing to 0 as k grows) only depends on the values of f on $\mathscr{R}_k$, so the points of $\mathscr{R}_k$ are relevant to evaluate the norm of a function.

Theorem III.2. *Let F be a norm that is weaker than the uniform norm. There exists a constant c such that if $f \in \mathrm{Lip}_1$ is null on $\mathscr{R}_{2k}$ then $F(f) \leq c \cdot 2^{-k}$.*

We first need a few lemmas.

Lemma III.2. *For every $\alpha \in [0,1]$ and $n \in \mathbb{N}$ there* *exists β such that $|\alpha-\beta| \leq 1/(2\mathrm{d}_{F,\alpha}(n))$ and $\mathrm{d}_{F,\beta}(n+2) \geq 2\mathrm{d}_{F,\alpha}(n)$.*

Proof: Let $l < \mathrm{d}_{F,\alpha}(n)$ and $f \in \mathrm{Lip}_1$ be supported on $N(\alpha, 1/l)$ and $F(f) > 2^{-n}$. Let h_1, h_2 be the maximal 1-Lipschitz functions supported on $[0,\alpha]$ and $[\alpha, 1]$ respectively. Let $f_1 = \min(f, h_1)$, $f_2 = \min(f, h_2)$ and $f_3 = f_4 = (f - f_1 - f_2)/2$. All f_i are 1-Lipschitz and are supported on $N(\beta_i, 1/(2l))$ for some $\beta_i \in \overline{N}(\alpha, 1/(2l))$. As $f = f_1 + f_2 + f_3 + f_4$ there exists i such that $F(f_i) > 2^{-n-2}$, which implies $d_{\beta_i}(n+2) > 2l$.



(a) f, h_1 and h_2

(b) f_1

(c) f_2

Fig. 2: $f = f_1 + f_2 + f_3 + f_4$

To each $l < \mathrm{d}_{F,\alpha}(n)$ is associated some β_l. By compactness of $[0,1]$, there exists an accumulation point β when l tends to $\mathrm{d}_{F,\alpha}(n)$. By continuity, β satisfies the conditions. ■

Lemma III.3. *For every $\alpha \in [0,1]$ and every $n \in \mathbb{N}$ there exists β such that $|\alpha - \beta| \leq 1/\mathrm{d}_{F,\alpha}(n)$ and $\mathrm{d}_{F,\beta}(n+p) \geq 2^{\frac{p-3}{2}}\mathrm{d}_{F,\alpha}(n)$ for all $p \in \mathbb{N}$.*

Proof: We iteratively apply Lemma III.2. Let $\beta_0 = \alpha$. Given β_p, let β_{p+1} be obtained by applying Lemma III.2 to β_p and $n + 2p$. One has $\mathrm{d}_{F,\beta_p}(n+2p) \geq 2^p \mathrm{d}_{F,\alpha}(n)$ and $|\beta_p - \beta_{p+1}| \leq 2^{-p-1}/\mathrm{d}_{F,\alpha}(n)$. β_p is a Cauchy sequence, let β be its limit. As $|\beta - \beta_p| \leq 2^{-p}/\mathrm{d}_{F,\alpha}(n)$, a function supported on $N(\beta_p, 2^{-p}/\mathrm{d}_{F,\alpha}(n))$ is supported on $N(\beta, 2^{-p+1}/\mathrm{d}_{F,\alpha}(n))$ so $\mathrm{d}_{F,\beta}(n+2p) \geq 2^{p-1}\mathrm{d}_{F,\alpha}(n)$. As $\mathrm{d}_{F,\beta}(n+2p+1) \geq \mathrm{d}_{F,\beta}(n+2p) \geq 2^{p-1}\mathrm{d}_{F,\alpha}(n)$ we get the result. ■

We will need the following refinement of Theorem III.1.

Lemma III.4. *There exists a constant c such that for all ϵ and all $f \in \mathrm{Lip}_1$, if $|f| \leq \epsilon$ on $R_{n,l}$ then $F(f) \leq c\epsilon + l^2 2^{-n+6}$.*

Proof: Let c be such that $F \leq c\|.\|_\infty$. We decompose f into a sum $f = g + h$ of 1-Lipschitz functions such that $\|g\|_\infty \leq \epsilon$ and $h = 0$ on $\mathrm{R}_{n,l}$. g and h are defined as $g(x) = f(x)$ if $|f(x)| \leq \epsilon$, $g(x) = -\epsilon$ if $f(x) \leq -\epsilon$ and $g(x) = \epsilon$ if $f(x) \geq \epsilon$, and $h = f - g$.

One easily checks that g and h are 1-Lipschitz. As $f = g + h$, $F(f) \leq F(g) + F(h)$ by triangular inequality. $F(g) \leq c\epsilon$ and $F(h) \leq l^2 2^{-n+6}$ by Theorem III.1. ■

Proof of Theorem III.2: Lemma III.3 gives

$$\mathrm{R}_{2k,l} \subseteq N\Big(\bigcap_{n \geq 2k} \mathrm{R}_{n, 2^{\frac{n-3}{2}-k} l}, 1/l\Big).$$

Assume $l \geq 2^{\frac{3}{2}}$. The set $\bigcap_{n \geq 2k} \mathrm{R}_{n, 2^{\frac{n-3}{2}-k} l}$ is contained in $\mathscr{R}_k$. Indeed, if $n \geq 2k$ then $l 2^{\frac{n-3}{2}-k} \geq 2^{\frac{n}{2}-k}$.

As a result, if $f \in \mathrm{Lip}_1$ is null on $\mathscr{R}_k$ then $|f| \leq 1/l$ on $\mathrm{R}_{2k,l}$, so by Lemma III.4 $F(f) \leq \frac{c}{l} + l^2 2^{-2k+6}$ for some c. Now take $l = 2^{\frac{k}{2}}$: $F(f) \leq c 2^{-\frac{k}{2}} + 2^{-k+6} \leq c' 2^{-\frac{k}{2}}$ for some c' that only depends on c (strictly speaking, the result is proved for $k \geq 3$ as we need $l \geq 2^{\frac{3}{2}}$. However, the result can be obtained for all k by changing the constant c'). ■

The result can easily be generalized from the case of 1-Lipschitz functions to any $f \in \mathcal{C}[0,1]$.

Corollary III.1. *Let F be a norm that is weaker than the uniform norm. There exists $c \in \mathbb{N}$ such that if $f \in \mathcal{C}[0,1]$ is null on $\mathscr{R}_{2\mu_f(p)}$ then $F(f) \leq c \cdot 2^{-p}$, where μ_f is a modulus of continuity of f.*

Proof: Let c_0 be such that $F \leq c_0\|.\|_\infty$. Let $L = 2^{\mu_f(p)-p+1}$. Let $k = \mu_f(p)$. There exists an L-Lipschitz function g which is null on $\mathscr{R}_{2k}$ and such that $\|f - g\|_\infty \leq 2^{-p}$. Indeed, let $p_i = i2^{-\mu_f(p)}$ for $i = 0$ to $2^{\mu_f(p)}$. Let $g(p_i) = 0$ if $d(p_i, \mathscr{R}_{2k}) < 2^{-\mu_f(p)}$, $g(p_i) = f(p_i)$ otherwise and extend g to a linear function between two consecutive points. One easily checks that g satisfies the required condtions.

First, $F(f - g) \leq c_0 2^{-p}$. Applying Theorem III.2 to g/L gives $F(g) \leq c_1 L 2^{-\mu_f(p)} \leq c_1 2^{-p}$ for some constant c_1, so $F(f) \leq c_0 2^{-p} + c_1 2^{-p}$. ■

C. Complexity of norms

We now show how the complexity of a norm has an influence on the shape of the norm, which can be measured by the way it depends on the points and by the size of the set of relevant points. A complexity restriction on a norm has two different effects: it bounds the internal computation time of the machine, reducing its computational power, and it bounds the number and size of the queries to the oracle representing the input, which reduces its knowledge of the input. We focus on the second type of restriction only, measured by the number of queries submitted to the oracle. To capture exactly the effects of this second type of restriction, we allow an extra oracle to a machine, giving it arbitrary computational power.

1) Query complexity:

Definition III.4. A ***bounding class*** is a class $\mathcal{T}$ of functions $t : \mathbb{N} \to \mathbb{R}$ satisfying the following conditions:

- $\mathcal{T}$ contains the constant function 1,
- $\mathcal{T}$ is closed downwards: if $t' \leq t$ and $t \in \mathcal{T}$ then $t' \in \mathcal{T}$,
- $\mathcal{T}$ is stable under multiplication by a polynomial.

In particular $\mathcal{T}$ contains all the (first-order) polynomials. The class POLY of functions that are bounded by polynomials is a bounding class. We say that a bounding class $\mathcal{T}$ is ***sub-exponential*** if for every $t \in \mathcal{T}$ and $\epsilon > 0$, $t = o(2^{\epsilon n})$. The class POLY is an example of a sub-exponential bounding class.

Here we represent the null function $z \in \mathcal{C}[0,1]$ using the modulus $\mu_z(p) = p$ and the approximation function $z_{\mathbb{D}}(q,p) = 0$. Given an oracle Turing machine $\mathcal{M}$ and $n \in \mathbb{N}$, run $\mathcal{M}$ on input n and oracle $\langle \overline{\mu_z}, \overline{z_{\mathbb{D}}} \rangle$. q is an ***oracle query*** if the machine eventually asks the oracle for the value of $z_{\mathbb{D}}(q,p)$ for some p. Let Q_n be the set of oracle queries.

Definition III.5. Let $\mathcal{T}$ be a bounding class. A norm over $\mathcal{C}[0,1]$ has ***query complexity*** in $\mathcal{T}$ if it is computable by a relativized oracle Turing machine with an extra oracle $A \in \{0,1\}^{\mathbb{N}}$ for which the function $n \mapsto |Q_n|$ belongs to $\mathcal{T}$.

A bound on the time complexity always induces a bound on the query complexity: if a norm $\|.\|$ is computable by a machine that on oracle φ and input n, halts in time $t(|\varphi|, n)$, then $\|.\|$ has query complexity $t(\mathrm{id}, n)$. In particular, every polynomial-time computable norm has polynomial query complexity.

2) Relating query complexity, dependence and relevant points:

we are now able to relate the query complexity of a norm to the way it depends on points. The results are based on the following simple observation: if a norm F depends on a point then a machine computing F must query the oracle around that point.

Lemma III.5. *The following equivalent statements hold for all α, n:*

$$R_{n,l} \subseteq \overline{N}(Q_{n+1}, 1/l)$$
$$\alpha \in \overline{N}(Q_{n+1}, 1/\mathrm{d}_{F,\alpha}(n))$$
$$\mathrm{d}_{F,\alpha}(n) \leq 1/d(\alpha, Q_{n+1}).$$

Proof: Let $l = 1/d(\alpha, Q_{n+1})$. If $f \in \mathrm{Lip}_1$ is supported on $N(\alpha, 1/l)$ then $f = 0$ on Q_{n+1}. One can choose a representation of f satisfying $f_{\mathbb{D}}(q,p) = 0$ for $q \in Q_{n+1}$. The machine cannot distinguish f from 0 so it must output a rational number r such that $|r| = |r - F(0)| < 2^{-n+1}$. As $|r - F(f)| < 2^{-n+1}$, we conclude $F(f) < 2^{-n}$. As a result, $\mathrm{d}_{F,\alpha}(n) \leq l$. ■

The notions of dependence and relevant points enable us to express formally the intuition that a polynomial-time computable norm cannot depend on a large set of points, as a machine computing it in polynomial-time only has little time to evaluate its input. It is more generally true of any norm that has low query complexity.

Theorem III.3. *Let F be a norm that is weaker than the uniform norm and $\mathcal{T}$ a bounding class. If F has query complexity in $\mathcal{T}$ then for almost every α, $\mathrm{d}_{F,\alpha} \in \mathcal{T}$.*

In particular if $\mathcal{T}$ is sub-exponential then the set of relevant points has Lebesgue measure 0.

Proof: Let $t \in \mathcal{T}$ be such that $|Q_n| \leq t(n)$ for all n. Let $U_i = \bigcup_n N(Q_n, \frac{1}{2t(n)(n+i+1)^2})$ and $A_i = [0,1] \setminus U_i$. One has

$$\mu(U_i) \leq \sum_n \frac{1}{(n+i+1)^2} \leq \frac{1}{i}.$$

so $\mu(A_i) \geq 1 - \frac{1}{i}$. Let $\alpha \in A_i$ and $n \in \mathbb{N}$: as $\alpha \notin N(Q_n, \frac{1}{2t(n)(n+i+1)^2})$, $\mathrm{d}_{F,\alpha}(n) \leq 1/d(\alpha, Q_{n+1}) \leq 2t(n+1)(n+i+2)^2$ by Lemma III.5. The function $n \mapsto 2t(n+1)(n+i+2)^2$ belongs to $\mathcal{T}$. ■

Moreover,

Theorem III.4. *Let F be a norm that is weaker than the uniform norm and $\mathcal{T}$ a sub-exponential bounding class. If F has query complexity in $\mathcal{T}$ then the set of relevant points has Hausdorff dimension* 0.

Proof: We slightly refine the preceding proof. For $s > 0$, we replace the sets U_i by $V_i^s = \bigcup_n N\left(Q_n, (t(n)(n+i)^2)^{-1/s}\right)$.

Again, $\bigcap_i V_i^s$ contains all the relevant points as $(t(n)(n+i)^2)^{1/s} = o(2^{\frac{n}{2}})$, and its dimension is $\leq s$. As it is true for any $s > 0$, the set of relevant points has Hausdorff dimension 0. ■

In particular, if F is polynomial-time computable then most of the points are irrelevant. In other words, F depends on a small set. As we show now, it is balanced by the fact that it highly depends on some points. We know from Proposition III.4 that there exist points whose dependence function is at least of the order of $2^{\frac{n}{2}}$ and the example of the L^1 norm shows that the coefficient $\frac{1}{2}$ cannot be increased in general. However for polynomial-time computable norms, the coefficient can be taken arbitrarily close to 1. First, one easily improves Proposition III.5.

Proposition III.5. *Let F be a norm that is weaker than the uniform norm and $\mathcal{T}$ a bounding class. If F has* *query complexity in $\mathcal{T}$ then $\lambda n.\frac{2^n}{\mathrm{D}_F(n)} \in \mathcal{T}$.*

Proof: Let $t \in \mathcal{T}$ be such that $|Q_n| \leq t(n)$ for all n. Let $l < F(1)2^n/(3t(n))$. Let $g = \frac{1}{t(n)}\sum_{q\in Q_n} h_{q,1/l}$. $g \in \mathrm{Lip}_1$ and $g = \lambda x.1/(lt(n))$ on Q_n so $F(1/(lt(n)) - g) \leq 2^{-n+1}$, hence $F(g) \geq F(1)/(lt(n)) - 2^{-n+1}$. As a result there exists $q \in Q_n$ such that $F(h_{q,1/l}) \geq F(1)/(lt(n)) - 2^{-n+1} > 2^{-n}$, so $\mathrm{D}_F(n) \geq \mathrm{d}_{F,q}(n) \geq l$. As it is true for any $l < F(1)2^n/(3t(n))$, $\mathrm{D}_F(n) \geq F(1)2^n/(3t(n))$. ■

For instance if F has polynomial query complexity then $D_F(n) \geq \frac{2^n}{P(n)}$ for some polynomial n. We raise the question whether this bound is reached: is there some α such that $\lambda n.\frac{2^n}{\mathrm{d}_{F,\alpha}(n)} \in \mathcal{T}$? We leave the question open, but in the case when $\mathcal{T}$ is the class of functions that are bounded by polynomials, we are able to prove that $\mathrm{d}_{F,\alpha}$ can be larger than $2^{(1-\epsilon)n}$ for any $\epsilon > 0$. More precisely,

Theorem III.5. *Let F be a polynomial-time computable norm. There exist $\alpha \in [0,1]$ and $c > 0$ such that $\mathrm{d}_{F,\alpha}(n) \geq 2^{n-c\sqrt{n}\log n}$ for all sufficiently large n. The set of such α is even dense.*

Proof idea: The idea is to start from some triangular function $h_{\alpha_0,l}$ and to decompose it as a sum of many smaller triangular functions. As most of them will be far away from the query sets of the machine computing the norm, their norms will be very small. As the sum of the norms of all the small functions is bounded below by the norm of the initial function, one of the few functions, $h_{\alpha_1,l'}$ that are close to the query set must have a large norm. Applying the same argument to the smaller function and iterating to infinity produces a sequence α_i converging to some α which will satisfy the conclusion of the theorem. ■

We do not know whether this result can be improved: is the bound provided by Proposition III.5 reached? In other words, is there α such that $\mathrm{d}_{F,\alpha}(n) \geq \frac{2^n}{P(n)}$ for some polynomial P and all n? Adapting the proof of Theorem III.5 we are only able to prove the existence of α such that $\mathrm{d}_{F,\alpha}(n) \geq \frac{2^n}{P(n)}$ for some polynomial P and *infinitely many* n.

We end this section by a characterization of the norms that have polynomial query complexity.

A family of compact sets $K_k \subseteq [0,1]$ can be ***polynomially covered*** if there exists a polynomial $P(n,k)$ such that for each n,k there exists a set $A_{n,k}$ of cardinality bounded by $P(n,k)$ that $K_k \subseteq N(A_{n,k}, 2^{-n})$.

Proposition III.6. *A norm has polynomial query complexity if and only if for every k, $\mathscr{R}_k$ can be polynomially covered, uniformly in k.*

Proof sketch: If a norm has polynomial query complexity then by Lemma III.5, $\mathscr{R}_k \subseteq \mathrm{R}_{2(n+k),2^n} \subseteq \overline{N}(Q_{2(n+k)+1}, 2^{-n})$. We then use the assumption that $|Q_{2(n+k)+1}|$ is polynomial in n,k.

Conversely, assume that $\mathscr{R}_k \subseteq N(A_{n,k}, 2^{-n})$ for some set $A_{n,k}$ of cardinality bounded by $P(n,k)$ for some polynomial P. We can assume w.l.o.g. that the points of $A_{n,k}$ belong to $\mathbb{D}_n$. The additional oracle provides two types of information: given n,k, it provides the set $A_{n,k}$ and given n,k,p and a list of $|A_{n,k}|$ values, it provides a 2^{-p}-approximation of $\|f\|$ where f is the piecewise linear function with the corresponding values on $A_{n,k}$. We now describe the machine computing the norm. On input p, the machine asks for $\mu_f(p)$, then asks the auxiliary oracle for $A_{\mu_f(p),2\mu_f(p)}$, evaluates f on the latter set at precision 2^{-p} and then using the returned values, asks the auxiliary oracle for the norm of the corresponding piecewise linear function and outputs that value. Corollary III.1 tells us that the output value is within $c2^{-p}$ of the value norm of f, for some constant c. Taking c into account, the machine can be adjusted to compute the norm. It is routine to check that the oracle can be coded as an element of $\{0,1\}^{\mathbb{N}}$ and that the queries of the machines are polynomial in p and the size of the representation of f. ■

IV. One oracle access

In this section we investigate the extreme case of a functional whose "query complexity[1]" is bounded by 1, i.e. a functional $F : \mathcal{C}[0,1] \to \mathbb{R}$ that is computable by a machine making at most one oracle call on each precision input n. Technically, the representation of f contains two types of information: the modulus of continuity of f and the values of f on the dyadic rationals. Here we separate the representation of f into two oracles and restrict the machine to perform one query to the approximation oracle.

We are able to characterize exactly this class of functionals. While the result looks natural, the proof is more delicate than expected: it is what makes the result interesting. As the last result (Proposition IV.1) shows, the argument hides subtleties that make it non-uniform.

Observe that it is trivial to obtain a characterization of functionals $F : (\mathbb{N} \to \mathbb{N}) \to (\mathbb{N} \to \mathbb{N})$ that are computable by an oracle Turing machine such that to compute $F(f)(n)$, does only one oracle call to f. Obviously, the functional F can be expressed as $F(f)(n) = \varphi(f(\psi(n))$ for some computable functions $\varphi, \psi : \mathbb{N} \to \mathbb{N}$. Indeed, let $\varphi(m) = F(\lambda x.m)$ and $\psi(n)$ be the question asked by the machine to the oracle

[1] here we make informal use of this expression as the query complexity is only defined for norms.

on input n. If F is assumed to be polynomial-time computable, then so are φ and ψ. The argument is much more elaborate on the real numbers.

Theorem IV.1. *For a functional $F : \mathcal{C}[0,1] \to \mathbb{R}$ the following are equivalent:*

1) *F is computable by a (polynomial time) oracle Turing machine that does at most one query to the approximation oracle,*
2) *there exist a (polynomial time) computable real number $\alpha \in [0,1]$ and a (polynomial time) computable real function $\phi : \mathbb{R} \to \mathbb{R}$ which is uniformly continuous and whose modulus of uniform continuity is bounded by a computable function (a polynomial) such that $F(f) = \phi(f(\alpha))$ for all $f \in \mathcal{C}[0,1]$.*

Observe that ϕ is uniquely determined by $\phi(v) = F(\lambda x.v)$ where $\lambda x.v$ is the constant function with value v. If F is not constant then α can be proved to be unique.

One could imagine a kind of BSS model of oracle computation for functionals, where the machine is allowed to ask its oracle for the value $f(\alpha)$, giving α in one step, and getting the value $f(\alpha)$ in one step as well (see [14] for the usual BSS model). In such a model, it is obvious that a functional F computed by a machine making one oracle call should be of the form $F(f) = \phi(f(\alpha))$. Theorem IV.1 tells us that this is also true for oracle machines working at finite precision. Note however that the two models would not have the same computation power: the uniform norm would not be computable in such a model, as the machine should evaluate its input function on an infinite set (which, at finite precision, can be approximated by evaluating the function on a finite set).

A. Uniformity

Our question is now: can α and ϕ be efficiently computed from F? As for ϕ, the answer is positive: it can be easily recovered as $\phi(v) = F(\lambda x.v)$. However the proof of Theorem IV.1 is not fully uniform as to compute α, one first has to find input functions witnessing that F is not constant. These objects can be effectively found, but not necessarily efficiently. The next result shows that this problem cannot be got around.

Proposition IV.1. *There exist α_k and ϕ_k such that α_k is not computable in polynomial time in k and ϕ_k is not constant but $F_k(f) := \phi_k(f(\alpha_k))$ is computable in time polynomial in k.*

As F_k is not constant, the decomposition (α_k, ϕ_k) of F_k is unique.

Proof: Let $A \subseteq \mathbb{N}$ be such that the problem $k \in A$ is decidable in time 2^k but not in polynomial time. Let $\alpha_k = 1$ if $k \in A$, $\alpha_k = 0$ if $k \notin A$. Let $\phi_k(x) = 2^{-2^k} \cdot x$. Let $F_k(f) = \phi_k(f(\alpha_k))$. First, α_k is not polynomially computable in k.

Of course, each F_k is computable in polynomial time separately (for k is fixed and hence constant). Moreover, we prove that F_k is polynomial time computable, uniformly in k. In other words, there is an oracle Turing machine $\mathcal{M}$ such that on oracle s representing $f \in \mathcal{C}[0,1]$, $\mathcal{M}^s(k,n)$ halts in time bounded above by a polynomial $P(|s|,k,n)$ and outputs a rational r such that $|F_k(f) - r| \leq 2^{-n}$.

Intuitively, if f does not vary much then it can be evaluated at 0 so computing α_k is not necessary; if f varies much then its modulus is large as well as the size of the representation of f, which gives enough computation time for evaluating α_k.

Let μ_f and $f_\mathbb{D}$ be a modulus and an approximation function of f. Given inputs (n,k), the machine queries $\mu_f(0)$ to the oracle and next:

- if $n \leq 2^k - \mu_f(0) - 2$, then query $f_\mathbb{D}(0,0)$ and output an approximation of $2^{-2^k} f_\mathbb{D}(0,0)$ with precision 2^{-n-1}.
- if $n \geq 2^k - \mu_f(0) - 1$ then decide $k \in A$, compute α_k accordingly, query $f_\mathbb{D}(\alpha_k, n)$ and output $2^{-2^k} f_\mathbb{D}(\alpha_k, n)$.

First observe that $|f(\alpha_k) - f(0)| \leq |f(0) - f(1)| \leq 2^{\mu_f(0)}$. If $n \leq 2^k - \mu_f(0) - 2$, then $|\mathcal{M}^f(k,n) - F(f)| = |2^{-2^k} f_\mathbb{D}(0,0) - 2^{-2^k} f(\alpha_k)| = 2^{-2^k} |f_\mathbb{D}(0,0) - f(\alpha_k)| \leq 2^{-2^k}(|f_\mathbb{D}(0,0) - f(0)| + |f(0) - f(\alpha_k)|) \leq 2^{-2^k}(1 + 2^{\mu_f(0)}) \leq 2^{-2^k + \mu_f(0) + 1} \leq 2^{-n-1}$. So the output value is a 2^{-n}-approximation of $F_k(f)$. The computations are done in polynomial time.

If $n \geq 2^k - \mu_f(0) - 1$ then the computation of α_k runs in time $\leq 2^k \leq n + \mu_f(0) + 1$ which is polynomial in n and $\mu_f(0)$. And it can be easily verified that $2^{-2^k} f_\mathbb{D}(\alpha_k, n)$ is a 2^{-n}-approximation of $F_k(f)$. ■

V. Summary and Open Questions

We have introduced the dependence function $\mathrm{d}_{F,\alpha}$ of a norm F on a point α (Definition III.1) and the notion of a relevant point (Definition III.3).

The norm of a function is determined by the values of the function on the points of high dependence (Theorem III.1). The set of relevant points is a growing union of compact sets $\mathscr{R} = \bigcup_k \mathscr{R}_k$. $\mathscr{R}$ is always dense (Proposition III.4) and the norm of a function at precision 2^{-k} is determined by the value of the function on $\mathscr{R}_{2\mu_f(k)}$ (Theorem III.2). Hence a machine computing the norm can evaluate its input function at the relevant points only. Moreover, the machine *has to* evaluate its input around the relevant point: each relevant point must be close to some oracle query (Lemma III.5).

We have shown the effects of query complexity restrictions on the norm, measured by the dependence function and the set of relevant points. In particular if a norm is computable in polynomial time then its set of relevant points has Hausdorff dimension 0 (Theorem III.4), almost every point has a polynomial dependence function (Theorem III.3) and there exist points of very high dependence (Theorem III.5). We also get a characterization of norms with polynomial query complexity (Proposition III.6).

We characterize the functionals $F : \mathcal{C}[0,1] \to \mathbb{R}$ that are computable by a Turing machine allowed to make at most one oracle query on each input (Theorem IV.1). We show that the characterization is not fully uniform in terms of complexity (Proposition IV.1).

A. Open questions

Is it possible to obtain a nice characterization of the functionals $F : \mathcal{C}[0,1] \to \mathbb{R}$ that are polynomial-time computable relative to some oracle, i.e. to extend Proposition III.6 to the general case? More generally is it possible to extend our analysis from norms to general functionals over $\mathcal{C}[0,1]$? The dependence of a functional on a point should be local, i.e. depend on the argument $f \in \mathcal{C}[0,1]$ of the functional: the functional $F(f) = f(f(0))$ intuitively depends on 0 and $f(0)$.

Our analysis is relevant when considering deterministic time complexity classes. What about the nondeterministic case? The uniform norm is nondeterministically polynomial-time computable, contrary to the L^1 norm, but they have the same sets of relevant points. What about space complexity?

Acknowledgments

The authors wish to thank Emmanuel Hainry, Emmanuel Jeandel and Romain Péchoux for fruitful discussions on the subject.

References

[1] K. Weihrauch, *Computable Analysis: An Introduction*. Berlin: Springer, 2000.

[2] V. Brattka, P. Hertling, and K. Weihrauch, "A tutorial on computable analysis," in *New Computational Paradigms*, S. B. Cooper, B. Löwe, and A. Sorbi, Eds. Springer New York, 2008, pp. 425–491.

[3] K.-I. Ko and H. Friedman, "Computational complexity of real functions," *Theor. Comput. Sci.*, vol. 20, pp. 323–352, 1982.

[4] K.-I. Ko, *Complexity Theory of Real Functions*. Birkhäuser, 1991.

[5] K. Weihrauch, "Computational complexity on computable metric spaces," *Math. Log. Q.*, vol. 49, no. 1, pp. 3–21, 2003.

[6] M. Schröder, "Spaces allowing type-2 complexity theory revisited," *Math. Log. Q.*, vol. 50, no. 4-5, pp. 443–459, 2004.

[7] A. Kawamura and S. Cook, "Complexity Theory for Operators in Analysis," in *Proc. of the 42nd ACM Symposium on Theory of Computing (STOC 2010)*, 2010, pp. 495–502.

[8] A. Kawamura, "Lipschitz continuous ordinary differential equations are polynomial-space complete," *Computational Complexity*, vol. 19, no. 2, pp. 305–332, 2010.

[9] A. Kawamura, H. Ota, C. Rösnick, and M. Ziegler, "Computational complexity of smooth differential equations," in *MFCS*, ser. Lecture Notes in Computer Science, B. Rovan, V. Sassone, and P. Widmayer, Eds., vol. 7464. Springer, 2012, pp. 578–589.

[10] M. Townsend, "Complexity for type-2 relations," *Notre Dame Journal of Formal Logic*, vol. 31, no. 2, pp. 241–262, 1990.

[11] A. Farjudian, "On the Kolmogorov complexity of continuous real functions," in *CiE*, ser. Lecture Notes in Computer Science, vol. 6735. Springer, 2011, pp. 81–91.

[12] K.-I. Ko, "The maximum value problem and NP real numbers," *J. Comput. Syst. Sci.*, vol. 24, no. 1, pp. 15–35, 1982.

[13] H. Friedman, "The computational complexity of maximization and integration," *Advances in Math.*, vol. 53, no. 1, pp. 80–98, 1984.

[14] L. Blum, F. Cucker, M. Shub, and S. Smale, *Complexity and real computation*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1998.

Appendix A
Proof of Theorem III.1

The proof intuitively works as follows: if f is null on $\mathrm{R}_{n,l}$ then it is supported on points of low dependence. f can then be decomposed as a sum of functions supported on small neighborhoods of points of low dependence. The norm of each small function being small, the norm of their sum will also be small.

We actually prove that if f is moreover nonnegative then $F(f) \le l^2 2^{-n+5}$. It gives the result for general f by decomposing $f = f^+ - f^-$ where f^+ and f^- are nonnegative.

Lemma A.1. *Let $\alpha \in [0,1]$ and $g \in \mathrm{Lip}_1$. If $\mathrm{Supp}(g)$ is disjoint from $R_{n,l}$ and contained in $N(\alpha, 1/l)$ then $F(g) \le 2^{-n+1}$.*

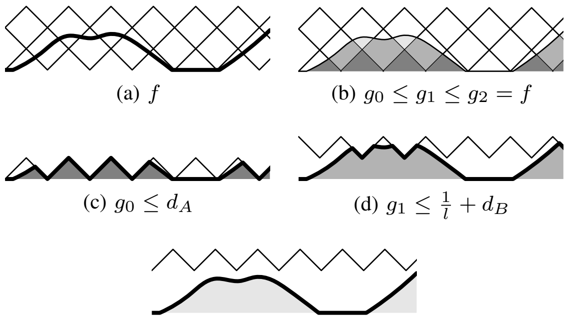
Proof: If $\mathrm{d}_{F,\alpha}(n) \le l$ then $F(g) \le 2^{-n}$. Otherwise, $g(\alpha) = 0$ and we decompose g into the sum of two 1-Lipschitz functions g_0, g_1 supported on $(\alpha - 1/l, \alpha)$ and $(\alpha, \alpha + 1/l)$ respectively. For each $i \in \{0,1\}$, either $g_i = 0$ or there exists $\beta \in \mathrm{Supp}(g_i)$. In the latter case, g_i is supported on $N(\beta, 1/l)$ and $\mathrm{d}_{F,\beta}(n) < l$ so $F(g_i) \le 2^{-n}$. As a result, $F(g) \le F(g_0) + F(g_1) \le 2^{-n+1}$. ■

Proof of Theorem III.1: If $l \le 1$ then $\mathrm{R}_{n,l} = [0,1]$ so $F(f) = 0$. We assume now that $l > 1$. Let $A = \{\frac{2k}{l} : k \in \mathbb{N}, k \le \frac{l}{2}\}$ and $B = \{\frac{2k+1}{l} : k \in \mathbb{N}, k \le \frac{l-1}{2}\}$. Let $d_A(x) = d(x, A)$ and $d_B(x) = d(x, B)$. Observe that $d_A + d_B = \frac{1}{l}$.

Let $f \in \mathrm{Lip}_1$ be nonnegative. Define for $n, i \ge 0$ the following functions, depicted in Figure 3:

$$\begin{aligned} g_{2n} &= \min\left(\tfrac{2n}{l} + d_A, f\right), & f_0 &= g_0, \\ g_{2n+1} &= \min\left(\tfrac{2n+1}{l} + d_B, f\right), & f_{i+1} &= g_{i+1} - g_i. \end{aligned}$$

For all $i \ge 0$, g_i is 1-Lipschitz, so f_0 is 1-Lipschitz and f_{i+1} is 2-Lipschitz. One has $g_{2n} \le g_{2n+1} \le g_{2n} + 2d_B$ and $g_{2n+1} \le g_{2n+2} \le g_{2n+1} + 2d_A$ so $0 \le f_{2n+1} \le 2d_B$ and $0 \le f_{2n+2} \le 2d_A$. In particular, $f_{2n+1} = 0$ on B and $f_{2n} = 0$ on A.


(a) f

(b) $g_0 \le g_1 \le g_2 = f$

(c) $g_0 \le d_A$

(d) $g_1 \le \frac{1}{l} + d_B$

(e) $f = g_2 \le \frac{2}{l} + d_A$

Fig. 3: The functions g_i

Assume that $\mathrm{Supp}(f)$ is disjoint from $\mathrm{R}_{n,l}$. As $l \le \mathrm{D}_F(n)$, $\mathrm{R}_{n,l} \neq \emptyset$ so f vanishes at some point. As f is 1-Lipschitz, $\|f\|_\infty \le 1$ so $f_k = 0$ for all $k \ge l+1$: indeed, $g_k = g_{k-1} = f$.

For each i, $\mathrm{Supp}(f_i) \subseteq \mathrm{Supp}(f)$ is also disjoint from $\mathrm{R}_{n,l}$. As f_i is null on A or B, $\frac{f_i}{2}$ is the sum of at most $\frac{l+3}{2}$ functions g satisfying the conditions of Lemma A.1 so $F(f_i) \le (l+3)2^{-n+1}$. As $f = f_0 + \ldots + f_k$ with $k = \lceil l+1 \rceil - 1 < l+1$, $F(f) \le (l+2)(l+3)2^{-n+1} \le l^2 2^{-n+5}$ as $l \ge 1$. ■

Appendix B
Proof of Theorem III.5

Lemma B.1. *Let $k, p, n \in \mathbb{N}$ and $\alpha \in [2^{-k}, 1 - 2^{-k}]$. There exists $\beta \in N(\alpha, 2^{-k})$ such that*

$$F(h_{\beta,2^{-p-k}}) \ge \frac{1}{P(n)}(2^{-p-1}F(h_{\alpha,2^{-k}}) - 2^{p-n+1}).$$

Proof: We decompose $h_{\alpha,2^{-k}}$ as a sum of 2^{2p} functions $h_{\beta_i,2^{-p-k}}$, $i = 1\ldots, 2^{2p}$ where β_i ranges over the set $B = \{\alpha \pm j2^{-p-k} : 0 \le j < 2^p\}$ of cardinality $2^{p+1} - 1$ (see Figure 4). Each β is the center of at most 2^p functions, so

$$F(h_{\alpha,2^{-k}}) \le 2^p \sum_{\beta \in B} F(h_{\beta,2^{-p-k}}).$$

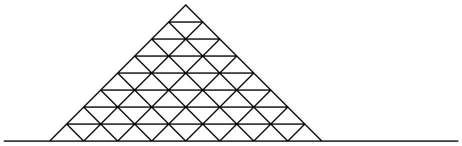


Fig. 4: $h_{\alpha,2^{-k}} = \sum_{i=1}^{2^{2p}} h_{\beta_i,2^{-p-k}}$. Here, $p = 3$.

We split B into a disjoint union $B_0 \cup B_1$: $\beta \in B_0$ iff $d(\beta, Q_n) < 2^{-p-k}$. First, $|B_0| \le 2|Q_n|$. Indeed, to each $\beta \in B_0$ one can associate some $q \in Q_n$ such that $|\beta - q| < 2^{-p-k}$. This mapping is two-to-one so $|B_0| \le 2|Q_n|$.

Let $M = \max_{\beta \in B_0} F(h_\beta, 2^{-p-k})$. Observe that $M = 0$ if $B_0 = \emptyset$. If $\beta \in B_1$ then $F(h_{\beta,2^{-p-k}}) \le 2^{-n+1}$, so

$$\begin{aligned} F(h_{\alpha,2^{-k}}) &\le 2^p \sum_{\beta \in B_0} F(h_{\beta,2^{-p-k}}) + 2^p \sum_{\beta \in B_1} F(h_{\beta,2^{-p-k}}) \\ &\le 2^p |B_0| M + 2^{p-n+1}|B_1| \\ &\le 2^{p+1}P(n)M + 2^{2p-n+2}. \end{aligned}$$

As a result,

$$M \ge \frac{1}{P(n)}(2^{-p-1}F(h_{\alpha,2^{-k}}) - 2^{p-n+1}).$$

If the term on the right-hand side is positive then M is positive so $B_0 \neq \emptyset$ and $M = F(h_{\beta,2^{-p-k}})$ for some $\beta \in B_0$. If the right-hand side is nonpositive then every β satisfies the required condition. ■

Claim B.1. There exist k_0 and d such that for all $k \geq k_0$ and $p = \lceil d\sqrt{k}\log(k)\rceil$,

$$\frac{2^{-p-2-k-\sqrt{k}}}{P(2p+4+k+\sqrt{k})} > 2^{-(p+k)-\sqrt{p+k}}.$$

Proof: We look for p satisfying $\sqrt{p+k} - \sqrt{k} > 2 + \log P(2p+4+k+\sqrt{k})$. Let d_0 be the degree of P and $d > 2d_0$. Let $p = \lceil \sqrt{k}\log k\rceil \in o(k)$.

$\sqrt{p+k} - \sqrt{k} \sim \frac{p}{2\sqrt{k}} \geq d\log k/2$ and $2 + \log P(2p+4+k+\sqrt{k}) \sim d_0 \log k$ so the inequality is satisfied for all sufficiently lage k. ■

Lemma B.2. *If $F(h_{\alpha,2^{-k}}) > 2^{-k-\sqrt{k}}$ then there exists $\beta \in N(\alpha, 2^{-k})$ such that $F(h_{\beta,2^{-p-k}}) > 2^{-p-k-\sqrt{p+k}}$ where $p = \lceil \sqrt{k}\log k\rceil$.*

Proof: Let $p \in \mathbb{N}$ and $n = \lceil 2p+3+k+\sqrt{k}\rceil$. Applying Lemma B.1 gives $\beta \in N(\alpha, 2^{-k})$ such that

$$F(h_{\beta,2^{-p-k}}) \geq \frac{2^{-p-2-k-\sqrt{k}}}{P(2p+4+k+\sqrt{k})}.$$

When p tends to infinity, the right-hand side eventually exceeds $2^{-(p+k)-\sqrt{p+k}}$. Actually, it happens early.

As a result, $F(h_{\beta,2^{-p-k}}) > 2^{-(p+k)-\sqrt{p+k}}$ for $p = \lceil d\sqrt{k}\log k\rceil$, if $k \geq k_0$. ■

Proof of Theorem III.5: Observe that if the result holds for a norm $F' = NF$ with $N \in \mathbb{N}$ then the result holds for F.

Lemma B.2 provides d and k_0. We define by induction α_i, p_i, k_i. We start from k_0, $\alpha_0 \in [2^{-k_0}, 1-2^{-k_0}]$ and we assume that $F(h_{\alpha_0,2^{-k_0}}) > 2^{-\lambda k_0}$, multiplying F by a constant if necessary. Applying Lemma B.2 to α_i and k_i gives β. Let $p_i = \lceil d\sqrt{k_i}\log(k_i)\rceil$, $\alpha_{i+1} = \beta$ and $k_{i+1} = k_i + p_i$. One has $F(h_{\alpha_i,2^{-k_i}} > 2^{-k_i-\sqrt{k_i}}$ and $|\alpha_i - \alpha_{i+1}| < 2^{-k_i}$ for all i.

As k_i is increasing, α_i is a Cauchy sequence so it converges to some α. One has $|\alpha_i - \alpha| \leq 2^{-k_i+1}$.

Let $n_i = \lceil k_i + \sqrt{k_i}\rceil$ for some i. $h_{\alpha_i,2^{-k_i}}$ is supported on $N(\alpha, 2^{-k_i+2})$ and $F(h_{\alpha_i,2^{-k_i}}) > 2^{-k_i-\sqrt{k_i}} \geq 2^{-n_i}$ so $\mathrm{d}_{F,\alpha}(n_i) \geq 2^{k_i-2}$.

One has $n_{i+1} - k_i \sim k_{i+1} - k_i \sim d\sqrt{k_i}\log k_i$, so if i is sufficiently large then $n_{i+1} - k_i \leq (d+1)\sqrt{k_i}\log k_i \leq (d+1)\sqrt{n_i}\log(n_i)$.

Now given $n \in \mathbb{N}$, let i be such that $n_i \leq n < n_{i+1}$. If i is sufficiently large then $k_i \geq n - (d+1)\sqrt{n}\log n$ so $\mathrm{d}_{F,\alpha}(n) \geq \mathrm{d}_{F,\alpha}(n_i) \geq 2^{k_i-2} \geq 2^{n-(d+1)\sqrt{n}\log n - 2}$.

As a result, for all sufficiently large n, $\mathrm{d}_{F,\alpha}(n) \geq 2^{n-(d+2)\sqrt{n}\log n}$. ■

Appendix C
Proof of Theorem IV.1

Notation C.1. Assume a function $m\colon \mathbb{N} \to \mathbb{N}$. Let $\mathcal{C}_m[0,1]$ denote the set of continuous functions defined over the interval $[0,1]$ having a modulus of continuity m.

We prove $1 \Rightarrow 2$, the other direction is straightforward. Let $\mathcal{M}$ be an oracle Turing machine computing F and such that on each input n the machine does at most one oracle call to the approximation oracle. We have two cases.

Case 1. Assume F is constant. That is $F(f) = a$ for some $a \in \mathbb{R}$. Let $\phi(y) = a$ and $\alpha = 0$ for instance. Obviously, F is (polynomial-time) computable if and only if ϕ is (polynomial-time) computable.

Case 2. Assume F is not constant. We first define $\phi(y) = F(g_y)$ where g_y is the constant function with value y.

Let f_0, g_0 and n_0 be such that $|F(f_0) - F(g_0)| > 2^{-n_0+1}$. Let m be a common modulus of continuity for f_0 and g_0. We can assume w.l.o.g. that $m(p) \geq p$ and m is nondecreasing, replacing $m(p)$ with $\max(m(0), m(1), \ldots, m(p), p)$ if necessary. Let $M(p) = m(p+1)$. $M \geq m$ is also a modulus for f_0 and g_0.

When the machine computing F is run on oracle (M, ψ) and input $n \geq n_0$ for some approximation function ψ, the machine must consult ψ. Indeed, otherwise it cannot distinguish between f_0 and g_0, which implies $|F(f_0) - F(g_0)| \leq 2^{-n+1} \leq 2^{-n_0+1}$ contradicting the assumption. For every $n \geq n_0$, let (q_n, p_n) be the query submitted by $\mathcal{M}^{(M,\psi)}(n)$ to ψ ((q_n, p_n) does not depend on ψ, as the machine is deterministic does not have consulted it yet).

Claim C.1. ϕ is uniformly continuous and $n \mapsto p_{n+1}$ is a modulus of uniform continuity for ϕ.

Proof: Assume $|x-y| \leq 2^{-p_{n+1}}$. Let r be a rational number such that $|x-r| < 2^{-p_{n+1}}$ and $|y-r| < 2^{-p_{n+1}}$. When evaluating $F(g_x)$ and $F(g_y)$ at precision 2^{-n-1}, the oracle can answer r to the query (q_{n+1}, p_{n+1}). The deterministic machine will produce the same output, so $|F(g_x) - F(g_y)| \leq 2^{-n}$, hence $|\phi(x) - \phi(y)| \leq 2^{-n}$. ■

Observe that the modulus of uniform continuity of ϕ is computable. In case the machine runs in polynomial time, the modulus is bounded by a polynomial.

We now define α.

Claim C.2. Assume α is an accumulation point of the sequence $(q_n)_{n\in\mathbb{N}}$. Then

1) Let $f, g \in \mathcal{C}_M[0,1]$: if $f = g$ on a neighborhood of α then $F(f) = F(g)$.

2) Let $f, g \in \mathcal{C}_m[0,1]$: if $f(\alpha) = g(\alpha)$ then $F(f) = F(g)$.

Proof: 1) Assume $f = g$ on a neighborhood U of α where $f, g \in \mathcal{C}_M[0,1]$. Both functions f and g have representations ψ_f and ψ_g such that $\psi_f(q,p) = \psi_g(q,p)$ for every $q \in U \cap \mathbb{Q}$ and $p \in \mathbb{N}$. Since α is an accumulation point of $(q_n)_{n\in\mathbb{N}}$, there exists an infinite set $E \subseteq \mathbb{N}$ such that for all $k \in E$ we have $q_k \in U$. Given the representations ψ_f and ψ_g, the machine cannot distinguish between f and g for input precisions $k \in E$, that is $\mathcal{M}^{M,\psi_f}(k) = \mathcal{M}^{M,\psi_g}(k) = r_k$ for every $k \in E$. We have $|r_k - F(f)| \le 2^{-k}$ and $|r_k - F(g)| \le 2^{-k}$, so $|F(f) - F(g)| \le 2^{-k+1}$. Given that E is infinite we have the desired result $F(f) = F(g)$. This proves the first part of the claim.

2) Now assume $f(\alpha) = g(\alpha)$ with $f, g \in \mathcal{C}_m[0,1]$. There exists a function g_n such that: (1) g_n coincides with f on $N(\alpha, 2^{-n})$, (2) $\|g - g_n\|_\infty \le 2^{-n+1}$ and (3) M is a modulus for g_n (g_n cannot have modulus m in general, this is why we need to consider M). g_n can be constructed as follows.

Let $\beta_n = \max(\alpha - 2^{-n}, 0)$ and $\gamma_n = \min(\alpha + 2^{-n}, 1)$. Let $\delta_1 = f(\beta_n) - g(\beta_n)$ and $\delta_2 = f(\gamma_n) - g(\gamma_n)$. We have $|\delta_1| \le |f(\beta_n) - f(\alpha)| + |f(\alpha) - g(\alpha)| + |g(\alpha) - g(\beta_n)| \le |\beta_n - \alpha| + 0 + |\beta_n - \alpha| \le 2^{-n+1}$. Similarly, $|\delta_2| \le 2^{-n+1}$.

Now define $g_n \colon [0,1] \to \mathbb{R}$ as follows. For $\beta_n \le x \le \gamma_n$: $g_n(x) = f(x)$. For $x \le \beta_n$: $g_n(x) = g(x) + \delta_1$. For $x \ge \gamma_n$: $g_n(x) = g(x) + \delta_2$. It can be easily verified that g_n satisfies the required properties. We have $f = g_n$ on a neighborhood of α, so from Part (1) of the Claim we have $F(f) = F(g_n)$. As F is continuous and g_n converge to g in the uniform norm, $F(g_n)$ converge to $F(g)$ so $F(f) = F(g)$. This completes the proof of Claim C.2. ■

Claim C.3. If $(q_n)_{n\in\mathbb{N}}$ has more than one accumulation point, then F is constant on $\mathcal{C}_m[0,1]$.

Proof: Assume that $(q_n)_{n\in\mathbb{N}}$ has two different accumulation points α and β. Let $f, g \in \mathcal{C}_m[0,1]$ be arbitrary. We will show that $F(f) = F(g)$, hence F is constant. We consider two cases. First, assume $|g(\beta) - f(\alpha)| \le |\beta - \alpha|$. Let h be the linear function such that $h(\alpha) = f(\alpha)$ and $h(\beta) = g(\beta)$. By the current assumption, $h \in \mathrm{Lip}_1[0,1] \subseteq \mathcal{C}_m[0,1]$ as $m(p) \ge p$ so by Claim C.2 we have $F(f) = F(h) = F(g)$.

Next consider the case $|g(\beta) - f(\alpha)| > |\beta - \alpha|$. Without loss of generality assume that $f(\alpha) > g(\beta)$. One can find $h_1, \ldots, h_{2k+2} \in \mathrm{Lip}_1[0,1]$ such that, letting $h_0 = g$ and $h_{2k+2} = f$, $h_{2i+1}(\beta) = h_{2i}(\beta)$ and $h_{2i+2}(\alpha) = h_{2i+1}(\alpha)$, $i = 0, \ldots, k$. From Claim C.2, $F(g) = F(h_1) = \ldots = F(h_{2k+2}) = F(f)$. ■

As F is not constant by assumption, the sequence $(q_n)_{n\in\mathbb{N}}$ has a unique accumulation point α. Recall the function $\phi(y) = F(g_y)$ where g_y is the constant function with value y. By Claim C.2, $F(f) = \phi(f(\alpha))$ for all $f \in \mathcal{C}_m[0,1]$, as f coincides with $g_{f(\alpha)}$ at α.

Claim C.4. ϕ is uniformly continuous and $n \mapsto p_{n+1}$ is a modulus of uniform continuity for ϕ.

Proof: Assume $|x - y| \le 2^{-p_{n+1}}$. Let r be a rational number such that $|x - r| < 2^{-p_{n+1}}$ and $|y - r| < 2^{-p_{n+1}}$. When evaluating $F(g_x)$ and $F(g_y)$ at precision 2^{-n-1}, the oracle can answer r to the query (q_{n+1}, p_{n+1}). The deterministic machine will produce the same output, so $|F(g_x) - F(g_y)| \le 2^{-n}$, hence $|\phi(x) - \phi(y)| \le 2^{-n}$. ■

Observe that the modulus of uniform continuity of ϕ is computable. In case the machine runs in polynomial time, the modulus is bounded by a polynomial.

In order to prove that α is (polytime) computable, we first show that the speed of convergence of $(q_n)_{n\in\mathbb{N}}$ to α is also a modulus of continuity for ϕ.

Claim C.5. The function $n \mapsto -\log_2|\alpha - q_{n+1}| - 1$ is a modulus of uniform continuity for ϕ. In other words, $|x - y| \le 2|\alpha - q_{n+1}|$ implies $|\phi(x) - \phi(y)| \le 2^{-n}$.

Proof: Let $n \in \mathbb{N}$. Assume $x, y \in \mathbb{R}$ such that $|x - y| \le 2^{-(-\log_2|\alpha - q_{n+1}| - 1)} = 2|\alpha - q_{n+1}|$. Then let f, g be the affine functions satisfying $f(\alpha) = x$, $g(\alpha) = y$, and $f(q_{n+1}) = g(q_{n+1}) = \frac{x+y}{2}$. As $|x - y| \le 2|\alpha - q_{n+1}|$, $f, g \in \mathrm{Lip}_1[0,1] \subseteq \mathcal{C}_m[0,1]$. Then f and g have representations ψ_f and ψ_g that give the same answer for the query (q_{n+1}, p_{n+1}). Hence, $\mathcal{M}^{M,\psi_f}(n+1) = \mathcal{M}^{M,\psi_g}(n+1)$. Thus we have $|F(f) - F(g)| \le 2^{-n}$. Therefore, $|\phi(x) - \phi(y)| = |\phi(f(\alpha)) - \phi(g(\alpha))| = |F(f) - F(g)| \le 2^{-n}$. This completes the proof of the Claim. ■

As ϕ is not constant, its modulus cannot be sub-linear, so q_n must converge quickly to α.

Claim C.6. There exists $k \in \mathbb{N}$ such that for all $n \in \mathbb{N}$ the following holds: $|\alpha - q_n| \le 2^{k-n}$.

Proof: By assumption F is not constant, so ϕ is not constant. As we show now, it implies that its modulus must be at least linear, i.e. $m(n) \ge n - k$ for some k and all sufficiently large n.

Let $a < b$ be such that $\phi(a) \ne \phi(b)$. Let $k \in \mathbb{N}$ be such that $|\phi(a) - \phi(b)| \ge 2^{-k}(b-a)$. Let m be a modulus of uniform continuity of ϕ. Let n be such that $|\phi(a) - \phi(b)| > 2^{-n}$: one must have $b - a \ge 2^{-m(n)}$. We divide the interval $[a,b]$ into intervals of length $\le 2^{-m(n)}$. We can take $p = \lfloor (b-a)2^{m(n)} \rfloor \ge 1$ subintervals. As ϕ does not vary more than 2^{-n} on each subinterval, a repeated use of the triangular inequality gives $|\phi(a) - \phi(b)| \le p2^{-n} \le (b-a)2^{m(n)-n}$ so $m(n) \ge n - k$.

As $m(n) = -\log_2 |\alpha - q_{n+1}| - 1$ is a modulus of ϕ, one has $|\alpha - q_{n+1}| \leq 2^{k-n-1}$. This completes the proof of the Claim. ■

To summarize, for every $f \in \mathcal{C}_m[0,1]$, $F(f) = \phi(f(\alpha))$. Observe that the construction of α may depend on m. We show that it does not.

Let m' be a nondecreasing function satisfying $m'(p) \geq p$ and such that F is not constant on $\mathcal{C}_{m'}[0,1]$. The argument developed so far associates to m' a real number $\alpha' \in [0,1]$ such that $F(f) = \phi(f(\alpha'))$ for all $f \in \mathcal{C}_{m'}[0,1]$.

Claim C.7. $\alpha' = \alpha$.

Proof: Assume $\alpha' \neq \alpha$. As ϕ is not constant there exist a, b such that $\phi(a) \neq \phi(b)$. a and b can be chosen arbitrarily close to each other: we can assume without loss of generality that $|a - b| \leq |\alpha - \alpha'|$. Let f be the affine function satisfying $f(\alpha) = a$ and $f(\alpha') = b$. Then f is in $\mathrm{Lip}_1[0,1] \subseteq \mathcal{C}_m[0,1] \cap \mathcal{C}_{m'}[0,1]$. As a result, $\phi(f(\alpha)) = F(f) = \phi(f(\alpha'))$ which implies $\phi(a) = \phi(b)$: we get a contradiction. ■

As a result, for every $f \in \mathcal{C}[0,1]$, $F(f) = \phi(f(\alpha))$. Indeed, f has a modulus $m' \geq m$ so $F(f) = \phi(f(\alpha')) = \phi(f(\alpha))$.

Now, to compute α, we need a modulus m such that F is not constant on $\mathcal{C}_m[0,1]$. The point is that m can be assumed to be $m(n) = n + k$ for some k. Indeed, as F is not constant and the Lipschitz functions are dense in $\mathcal{C}[0,1]$, there exist two such Lipschitz functions f_0 and g_0 such that $F(f_0) \neq F(g_0)$. If 2^k bounds the Lipschitz constants of f_0 and g_0 then $m(n) = n + k$ is a common modulus for f_0 and g_0 so F is not constant on $\mathcal{C}_m[0,1]$. From this m, the sequence q_n can be computed (in polynomial time) so α can be computed (in polynomial time) thanks to Claim C.6.

This completes the proof of the theorem.