Complexity lower bounds for approximation algebraic computation trees

Felipe Cucker*
Department of Mathematics
City University of Hong Kong
83 Tat Chee Avenue, Kowloon
HONG KONG
e-mail: macucker@math.cityu.edu.hk

Dima Grigoriev*
IMR
Université de Rennes I
Campus de Beaulieu, Rennes 35042
FRANCE
e-mail: grigoriev@cs.uni-bonn.de

Abstract

We prove lower bounds for approximate computations of piecewise polynomial functions which, in particular, apply for round-off computations of such functions.

The goal of this paper is to prove lower bounds for approximated computations. As it is customary for lower bounds, we consider some form of algebraic tree as our computational model (cf. [Bürgisser, Clausen, and Shokrollahi 1996] or [Blum, Cucker, Shub, and Smale 1998] for algebraic trees). But, unlike the usual proofs of lower bounds, which deal with decision problems, we will consider computations of real functions. That is, we consider trees computing functions $f : \mathbb{R}^n \to \mathbb{R}$ and, also unlike the usual results on lower bounds, we will allow for approximate computations. To understand the nature of our results let us look first at an example.

Example 1 Given a compact polygon $P \subset \mathbb{R}^2$ consider the function $f : \mathbb{R}^2 \to \mathbb{R}$ defined by

$$f(c) = \max_{x \in P} \langle c, x \rangle^2.$$

Obviously, there is a partition of $\mathbb{R}^2$ into a finite number of regions V_i and for each such region there is a vertex v_i of P such that $f(c) = \langle c, v_i \rangle^2$ for all $c \in V_i$.

*This work was done while both authors were at MSRI. We would like to thank the institute for its support.

Let T be an algebraic computation tree computing f of Example 1. Then the number of leaves of T is at least the number of 2-dimensional regions V_i with pairwise different v_i. This follows from the fact that two different polynomials in $\mathbb{R}[x, y]$ can not coincide, as functions, on an open subset of $\mathbb{R}^2$. Therefore, since computation trees are binary, we have that the depth of T is at least the $\log_2$ of this number. This argument is independent of the fact that the input space is $\mathbb{R}^2$ (any $\mathbb{R}^n$ could be considered instead; just replace polygon by polyhedra and $\mathbb{R}[x, y]$ by $\mathbb{R}[x_1, \ldots, x_n]$). We intend to replicate it for approximate computations.

Now consider a round-off tree T, i.e. an algebraic tree whose computation nodes are affected by some kind of error and assume that the tree computes a δ-approximation of f, that is, the output $T(c)$ satisfies $|f(c) - T(c)| \leq \delta$ for all $c \in \mathbb{R}^2$.

If $\delta \neq 0$ a lower bound like the one above is no longer valid. To see why, consider a regular n-sided polygon inscribed in the unit circunference centered at the origin. For large n the polygon becomes "close" to the circumference and for n large enough $f(c)$ is δ-approximated by $\|c\|^2 = c_1^2 + c_2^2$. And this function can be computed with only three operations. So the $\log_2 n$ bound above is far to apply.

Thus, in order to obtain meaningfull lower bounds one needs to impose some condition on the parameter δ. We devote the next section to define the main concepts of the paper and to state our main theorem, where this condition is made explicit.

1 Piecewise Polynomial Functions and Round-off Computation Trees

In this paper will only deal with trees whose computation nodes perform additions, subtractions or multiplications.[1] It is immediate to prove that such a tree (with exact arithmetic) computes a very specific kind of functions, which we describe in the next definition.

Definition 1 A function $f : \mathbb{R}^n \to \mathbb{R}$ is called *piecewise polynomial* if there exists a finite partition $\mathbb{R}^n = \cup_i V_i$ of $\mathbb{R}^n$ into semi-algebraic sets V_i and for each i a polynomial $f_i \in \mathbb{R}[x_1, \ldots, x_n]$ such that $f_{|V_i} = f_i$.

Without loss of generality we will assume that if $i \neq j$ then $f_i \neq f_j$.

The function f of Example 1 is piecewise polynomial. Another example of this kind of function is provided by quantifier elimination in the theory of the reals. Such a procedure defines a piecewise polynomial function by associating, to each tuple of coefficients of an input formula, a vector of coefficients of an equivalent quantifier-free formula.

Apparently, computation of piecewise polynomial (or more generally, rational) functions was considered for the first time over the complex numbers rather than

[1] The extension of our results to the case of trees allowing divisions is an open problem.

over the reals, as in our case, by Strassen [1983] for the problem of computing GCDs of univariate polynomials.

Definition 2 Let T be an algebraic computation tree with input space $\mathbb{R}^n$ and output space $\mathbb{R}$. We say that T δ-approximates a function $f : \mathbb{R}^n \to \mathbb{R}$ if for every input $x \in \mathbb{R}^n$ the output $T(x)$ of T satisfies $|T(x) - f(x)| \leq \delta$. The number δ is the *approximation error*.

Remark 1
1) The approximation error in the definition above is absolute. It applies to problems where absolute errors are considered as in [Cucker and Smale 1997]. One may consider also *relative* approximation error by requiring $|T(x) - f(x)| \leq \delta|f(x)|$. All our results can be modified in a straightforward manner to hold for relative approximation errors as well.
2) Although our main motivation to deal with approximate computations is the consideration of round-off errors at the computation nodes of the tree, there are other possibilities as well. A worth noting one is the class of exact algorithms, i.e. algorithms in which no round-off errors occur, which are designed to produce an approximate solution instead of the exact one. This is a current practice to improve the efficiency over the known algorithms computing the exact solution of the problem.
3) To the best of our knowledge very little is know on lower bounds for approximate (or round-off) computations. A worth noting exception is a paper by Renegar [Renegar 1987] which gives lower bounds for approximating zeros of univariate polynomials.

We now describe the condition we will impose on δ in order to obtain lower bounds for the depth of approximate computations. This condition takes the form of a bound $\delta \leq \Gamma$ where Γ is a quantity depending only on the piecewise function f (rather than on the tree). We actually provide a family of conditions parameterized by a positive parameter τ.

Let $\tau > 0$. If f is piecewise polynomial we define

$$w(\tau) = \#\{i \mid V_i \text{ contains an } n\text{-dimensional cube of side } \tau\}$$

and

$$B_\tau = \min\{b \in \mathbb{R} \mid \text{ the cubes above are contained in } [-B, B]^n\}.$$

Denote by I_τ the set of indices i satisfiying the condition in the definition of $w(\tau)$ and let $d_\tau = \max_{i \in I_\tau} \text{degree}\,(f_i)$, $C_\tau = \min_{i \neq j} \|f_i - f_j\|_\infty$, where polynomials are identified with their vectors of coefficients. Define

$$\Gamma_\tau = \frac{C_\tau}{2} \left(2^{\frac{(D_\tau - 1)^2}{4} - 1} \left(\frac{\tau}{N_\tau} \right)^{\frac{D_\tau (D_\tau + 1)}{2}} \left(\frac{1}{B_\tau} \right)^{D_\tau} \right)^n$$

where $D_\tau = \max\{d_\tau, w(\tau)\}$ and $N_\tau = w(\tau)D_\tau n + 1$.

We can now state our main theorem.

Theorem 1 *If T δ-approximates f and*

$$\delta \leq \Gamma_\tau$$

then the depth k of T satisfies

$$k \geq \log_2 w(\tau).$$

In proving Theorem 1 the following lemma is essential.

Lemma 1 *Let $f \in \mathbb{R}[x_1, \ldots, x_n]$ with $\text{degree}_{x_i}(f) \leq d$ and $M = \|f\|_\infty$. Let $b_1, \ldots, b_n \in \mathbb{R}$, $|b_i| \leq B$, $N \in \mathbb{N}$, and consider the uniform grid $\overline{S}$ with mesh τ/N in the cube*

$$\prod_{i=1}^{n} [b_i - \tau, b_i].$$

Let $S \subseteq \overline{S}$ with $|S| = s$. If

$$s > s_n = N^n \left(1 - \left(1 - \frac{d}{N}\right)^n\right) = N^n - (N-d)^n$$

then there exists $x \in S$ such that

$$|f(x)| > \alpha = M \left(2^{\frac{(d-1)^2}{4}-1} \left(\frac{\tau}{N}\right)^{\frac{d(d+1)}{2}} \left(\frac{1}{B}\right)^d\right)^n.$$

PROOF. By induction on n.

BASE CASE, $n = 1$. In this case, $s_1 = d$, so assume there is a subset S_0 of $\overline{S}$ having $d+1$ points $w_0, \ldots, w_d$ in $\overline{S}$ such that $|f(w_i)| \leq \alpha$ for $i = 0, \ldots, d$. Then, interpolating f at these points we express each coefficient of f as a fraction

$$\frac{\sum_{x \in S_0} a_x f(x)}{\Delta}$$

where

$$\Delta = \prod_{\substack{w_i, w_j \in S_0 \\ 0 \leq j < i \leq d}} (w_i - w_j)$$

is the determinant of the Vandermonde matrix

$$V = \begin{pmatrix} 1 & w_0 & w_0^2 & \ldots & w_0^d \\ 1 & w_1 & w_1^2 & \ldots & w_1^d \\ \vdots & & & & \vdots \\ 1 & w_d & w_d^2 & \ldots & w_d^d \end{pmatrix}$$

and a_x are the determinants of suitable minors of V. The smallest possible value of $|\Delta|$ occurs when $w_0, \ldots, w_d$ are consecutive in S (i.e. $w_i - w_{i-1} = \tau/N$) and in this case we have

$$|\Delta| \geq \prod_{i=1}^{d} \left(\frac{i\tau}{N}\right)^{d-i+1} \geq \left(\frac{\tau}{N}\right)^{\frac{d(d+1)}{2}} \prod_{i=1}^{d} i!.$$

On the other hand, by Hadamard's inequality, $|a_x| \leq B^d d!$. Therefore, each coefficient of f is less, in absolute value, than

$$\frac{(d+1)B^d d!\alpha}{(\frac{\tau}{N})^{\frac{d(d+1)}{2}} \prod_{i=1}^{d} i!} \leq \frac{(d+1)B^d\alpha}{(\frac{\tau}{N})^{\frac{d(d+1)}{2}} 2^{\frac{(d-1)^2}{4}-1}}$$

the last inequality since $\prod_{i=1}^{d} i! \geq 2^{\frac{d^2}{4}-1}$ for all $d \geq 1$. But then

$$M < \frac{(d+1)B^d\alpha}{(\frac{\tau}{N})^{\frac{d(d+1)}{2}} 2^{\frac{(d-1)^2}{4}-1}}$$

which is in contradiction with the definition of α.

Induction step, $n \geq 2$. Write $f = \sum_I f_I X^I$ where $f_I \in \mathbb{R}[x_n]$ and X^I is a monomial in $x_1, \ldots, x_{n-1}$. Now, take f_{I_0} such that $\|f_{I_0}\|_\infty = M$. By the base of the induction, for all but at most d points x in the set

$$L = \{b_n - \tau, b_n - \tau + \frac{\tau}{N}, b_n - \tau + \frac{2\tau}{N}, \ldots, b_n - \tau + \frac{N\tau}{N} = b_n\}$$

we have

$$|f_{I_0}(x)| > \frac{M\left(2^{\frac{(d-1)^2}{4}-1}\frac{\tau}{N}\right)^{\frac{d(d-1)}{2}}}{B^d}. \tag{1}$$

Therefore, there are more than $s_n - dN^{n-1}$ points in S whose last coordinate satisfies (1). We conclude that there exists one such point $x^* \in L$ such that moreover

$$|S \cap \{x_n = x^*\}| \geq \frac{s_n - dN^{n-1}}{N-d} = s_{n-1}.$$

Now apply the inductive hypothesis to the polynomial $f_{|x_n=x^*} \in \mathbb{R}[x_1, \ldots, x_{n-1}]$ using that

$$\|f_{|x_n=x^*}\|_\infty > \frac{M\left(2^{\frac{(d-1)^2}{4}-1}\frac{\tau}{N}\right)^{\frac{d(d-1)}{2}}}{B^d}$$

and the conclusion follows. □

2 Proof of Theorem 1

Recall that I_τ is the set of indices i satisfiying the condition of the definition of $w(\tau)$, $D_\tau = \max\{w(\tau), d_\tau\}$, and $N_\tau = w(\tau)D_\tau n + 1$. Let $i \in I_\tau$ and consider the grid $\overline{S} \subset V_i$ as in Lemma 1. We say that a leaf η of T is *attached* to V_i if η is reached by at least $N_\tau^n / w(\tau)$ points of $\overline{S}$.

We claim that one leaf of T can not be attached to two different sets V_i. From this claim it follows that $k \geq \log_2 w(\tau)$. Indeed, if $k < \log_2 w(\tau)$ then $|\text{Leaves}(T)| < w(\tau)$ and, by the pigeonhole principle, there is a leaf of T attached to V_i. So, every V_i has a leaf attached to it. And, by hypothesis, each leaf of T is attached to at most one V_i. But then $|\text{Leaves}(T)| \geq w(\tau)$ and therefore, $k \geq \log_2 w(\tau)$.

To prove the claim, assume that there exist sets V_i and V_j, $i \neq j \in I_\tau$, such that a leaf η is attached to both of them. Let $C' = \|f_i - f_j\|_\infty$ and let $g \in \mathbb{R}[x_1, \ldots, x_n]$ be the output polynomial corresponding to η. Then either $\|f_i - g\|_\infty \geq C'/2$ or $\|f_j - g\|_\infty \geq C'/2$. We can assume, w.l.o.g., that the first inequality holds.

Let $S \subset \overline{S}$ be the set of points reaching the leaf η. Then,

$$|S| \geq \frac{N_\tau^n}{w(\tau)} > N_\tau^n \left(1 - \left(1 - \frac{D_\tau}{N_\tau}\right)^n\right)$$

by the choice of N_τ. Thus we can apply Lemma 1 to the polynomial $f = f_i - g$ with $M = C'/2$, $d = D_\tau$ and $B = B_\tau$ and we deduce that there is a point $x \in S$ such that

$$|f_i(x) - g(x)| > \frac{C'}{2}\left(2^{\frac{(D_\tau-1)^2}{4}-1}\left(\frac{\tau}{N_\tau}\right)^{\frac{D_\tau(D_\tau+1)}{2}}\left(\frac{1}{B_\tau}\right)^{D_\tau}\right)^n \geq \Gamma_\tau$$

since $C' \geq C_\tau$. But this, together with the hypothesis on Γ_τ, contradicts the fact that $|T(x) - f_i(x)| \leq \delta$. □

Remark 2 The lower bound in Theorem 1 is on the depth of T. A more involved issue is the consideration of the *topological complexity* of f (cf. [Smale 1987] for this concept, see also [Vassiliev 1992]), i.e. the number of leaves of T. This number is essentially the amount of branching necessary for solving the problem. In our discussion of Example 1 we saw that the topological complexity of f is at least the number of 2-dimensional regions V_i with pairwise different v_i which is at least $w(\tau)$ for each $\tau > 0$.

For the problem MAX, consisting of finding the largest coordinate of an input $x \in \mathbb{R}^n$ and for which the number of pieces is n, the question of the topological complexity is open (see [Grigoriev, Karpinski, and Yao 1998] for the discussion and the exponential lower bound for *ternary* rathen than the usual binary computation trees).

Implicit in the proof of Theorem 1 is the fact that, if $k = \log_2 w(\tau)$, then the topological complexity of T, $\text{TC}(T)$, satisfies $\text{TC}(T) \geq w(\tau)$. It is unclear to us

whether one can trade topological for arithmetical complexity, that is, whether one can reduce the topological complexity of an approximated computation at the expense of increasing the degree of the computed polynomials. We can prove, however, a trade-off between these complexities (and the approximation error δ). Let T be an algebraic computation tree and g_η the polynomial computed at leaf η. Define

$$d_T = \max_{\eta \text{ a leaf of } T} \text{degree}\,(g_\eta).$$

Note that $d_T \leq 2^k$ where k is the depth of T. Now define $D_{(\tau,T)} = \max\{d_\tau, d_T\}$ and

$$\Gamma_{(\tau,T)} = \frac{C_\tau}{2}\left(2^{\frac{(D_{(\tau,T)}-1)^2}{4}-1}\left(\frac{\tau}{N_\tau}\right)^{\frac{D_{(\tau,T)}(D_{(\tau,T)}+1)}{2}}\left(\frac{1}{B_\tau}\right)^{D_{(\tau,T)}}\right)^n.$$

The arguments of Theorem 1 yield the following.

Theorem 2 *If T δ-approximates f and*

$$\delta \leq \Gamma_{(\tau,T)}$$

then the topological complexity $\mathrm{TC}(T)$ *of T satisfies*

$$\mathrm{TC}(T) \geq w(\tau).$$

□

3 Extensions

Theorem 1 can be extended to some contexts where trees are endowed with additional capabilities. In this section we briefly discuss how this is carried out for two such capabilities: randomization and parallelism.

3.1 Randomized Trees

One can define randomized versions of approximation trees by allowing "coin tossing" and requiring the output to be a δ-approximation with high probability. More precisely, we consider trees with input space $\mathbb{R}^n \times \{0,1\}^m$ (for the arguments which follow the exact value of m is not important) and we fix a confidence degree σ satisfying $0 \leq \sigma \leq 1$. Then, such a tree δ-approximates f when, for each $x \in \mathbb{R}^n$ and for at least $\sigma 2^m$ points b in $\{0,1\}^m$, we have $|T(x,b) - f(x)| \leq \delta$.

Assume that this happens and let X be the union of the grids $\overline{S}$ associated to the sets V_i with $i \in I_\tau$. Then there exists a point $b^* \in \{0,1\}^m$ such that for at least $\sigma|X|$ points in X we have $|T(x,b) - f(x)| \leq \delta$. Fix the coin tossing b^* and call these points *good* (with respect to b^*).

Lemma 2 *At least* $\frac{\sigma}{2-\sigma}w(\tau)$ *sets* V_i *contain more than* $\frac{\sigma}{2}N_\tau^n$ *good points.*

PROOF. Let α be the number of sets V_i containing more than $\frac{\sigma}{2}N_\tau^n$ good points. Then

$$|\text{good points}| \leq \alpha N_\tau^n + (w(\tau) - \alpha)N_\tau^n \frac{\sigma}{2}$$

and since the number of good points is at least $\sigma N_\tau^n w(\tau)$ the result follows. □

To replicate the proof of Theorem 1 we now consider the deterministic tree resulting for replacing the coin tossing by the fixed point b^* and we modify the quantities appearing in the definition of Γ_τ to allow for the confidence σ. Thus, we define $I_{(\tau,\sigma)}$ to be the subset of I_τ with those indices i such that V_i satisfies Lemma 2. Then, one defines $d_{(\tau,\sigma)}, C_{(\tau,\sigma)}, D_{(\tau,\sigma)}$, $N_{(\tau,\sigma)}$ and $\Gamma_{(\tau,\sigma)}$ as in Section 1.

Notice that $d_{(\tau,\sigma)} \leq d_\tau$, $C_{(\tau,\sigma)} \geq C_\tau$, etc. and so $\Gamma_{(\tau,\sigma)} \geq \Gamma_\tau$.

Theorem 3 *If* T *is a randomized tree which* δ*-approximates* f *with confidence* σ *and*

$$\delta \leq \Gamma_{(\tau,\sigma)}$$

then the depth k *of* T *satisfies*

$$k \geq \log_2 \left(\frac{\sigma}{2} w(\tau) \right).$$

SKETCH OF PROOF. We say that η is attached to V_i if η is reached by at least $N^n_{(\tau,\sigma)}/w(\tau)$ *good* points in $\overline{S}$.

Again, we claim that a leaf can not be attached to two different sets V_i and from this claim it follows the theorem. Indeed, if

$$k < \log_2 \left(\frac{\sigma}{2} w(\tau) \right)$$

then $|\text{Leaves}(T)| < \sigma w(\tau)/2$ and, by the pigeonhole principle, there is a leaf of T attached to V_i. So, every V_i has a leaf attached to it. And, by hypothesis, each leaf of T is attached to at most one V_i. But then

$$|\text{Leaves}(T)| \geq \frac{\sigma}{2-\sigma} w(\tau) \geq \frac{\sigma}{2} w(\tau)$$

and therefore, $k \geq \log_2(\frac{\sigma}{2}w(\tau))$.

The claim is proved as in Theorem 1. □

Remark 3 When dealing with decision problems, the confidence degree σ is assumed to be greater than $1/2$ (or rather, the probability error $\varepsilon = 1 - \sigma$ is assumed to be smaller than $1/2$). This is due to the fact that an algorithm consisting on tossing a coin and answering `Yes` or `No` according with the outcome of that coin tossing (and independently of the input) is already a probabilistic algorithm of confidence

1/2. Theorem 3 shows that such a simple algorithm is not going to work in the non-decisional case.

We also mention that a complexity lower bound for a probabilistic tree deciding an arrangement of hyperplanes or a polyhedron was obtained in [Grigoriev 1998]. This bound is logarithmic in the number of faces.

3.2 Parallel Trees

Parallel computations can be modelled by a particular kind of trees. If p denotes the number of processors, at each computational node, the tree performs an arithmetic operation and stores its result in at most p coordinates of the state space. Also, at each branching node, the sign of at most p such coordinates is tested, giving thus rise to 2^p possible outcomes. An elementary computation yields an upper bound of 2^{pk} leaves for such a tree with depth k. Since in most parallel models the number of processors is bounded by 2^k this upper bound becomes 2^{k2^k}.

If the computations are performed exactly (without errors) it turns out that most of these leaves are irrelevant in the sense that there are no points in $\mathbb{R}^n$ reaching them. More precisely, Yao [1981] (see [Montaña and Pardo 1993]) shows that in this case, the number of leaves which are reached by some point in $\mathbb{R}^n$ is bounded by

$$2^{\mathcal{O}(k^2 n)}.$$

Notice that from this it follows the inequality

$$k = \Omega\left(\sqrt{\frac{\log|\text{Leaves}(T)|}{n}}\right).$$

We remark that an upper bound close to the latter lower one (for small dimensions) for the parallel complexity of deciding an arrangement of hyperplanes or a polyhedron (as in Remark 3) was given in [Grigoriev 1997].

An almost verbatim repetition of the proof of Theorem 1 yields the following.

Theorem 4 *If T is an exact parallel tree which δ-approximates f and*

$$\delta \leq \Gamma_\tau$$

then the depth k of T satisfies

$$k = \Omega\left(\sqrt{\frac{\log_2 w(\tau)}{n}}\right).$$

□

Remark 4 The requirement of exact arithmetic for T in Theorem 4 seems unavoidable if we want to use Yao's bound on the number of relevant leaves. To see why, consider a set of s lines in $\mathbb{R}^2$ given by linear polynomials $\ell_1, \ldots, \ell_s$ and assume that these lines pass through a common point ξ. Now consider a branch node which tests the signs of $\ell_1, \ldots, \ell_s$ at a point x. If $x = \xi$ and round-off errors are allowed when computing $\ell_i(\xi)$, $i = 1, \ldots, s$, we may get up to 2^s possible outcomes.

References

Blum, L., F. Cucker, M. Shub, and S. Smale (1998). *Complexity and Real Computation*. Springer-Verlag.

Bürgisser, P., M. Clausen, and A. Shokrollahi (1996). *Algebraic Complexity Theory*. Springer-Verlag.

Cucker, F. and S. Smale (1997). Complexity estimates depending on condition and round-off error. To appear in *Journal of the ACM*.

Grigoriev, D. (1997). Nearly sharp complexity bounds for multiprocessor algebraic computations. *Journal of Complexity 13*, 50–64.

Grigoriev, D. (1998). Randomized complexity lower bounds. In *30th annual ACM Symp. on the Theory of Computing*, pp. 219–223.

Grigoriev, D., M. Karpinski, and A. Yao (1998). An exponential lower bound on the size of algebraic decision trees for MAX problem. To appear in *Computational Complexity*.

Montaña, J. and L. Pardo (1993). Lower bounds for arithmetic networks. *Applicable Algebra in Engineering, Communication and Computing 4*, 1–24.

Renegar, J. (1987). On the worst case arithmetic complexity of approximating zeros of polynomials. *Journal of Complexity 3*, 90–113.

Smale, S. (1987). On the topology of algorithms I. *Journal of Complexity 3*, 81–89.

Strassen, V. (1983). The computational complexity of continued fractions. *SIAM Journal on Computing 12*, 1–27.

Vassiliev, V. (1992). *Complements of Discriminants of Smooth Maps: Topology and Applications*, Volume 98 of *Translations of Mathemathical Monographs*. American Mathematical Society.

Yao, A. (1981). On the parallel computation of the knapsack problem. In *13th annual ACM Symp. on the Theory of Computing*, pp. 123–127.