Discrete & Computational Geometry (2022) 68:664–708
https://doi.org/10.1007/s00454-022-00403-x

On the Complexity of the Plantinga–Vegter Algorithm

Felipe Cucker[1] · **Alperen A. Ergür**[2] · **Josué Tonelli-Cueto**[3]

Received: 14 April 2020 / Revised: 11 October 2021 / Accepted: 26 January 2022 /
Published online: 29 August 2022

Abstract

We introduce tools from numerical analysis and high dimensional probability for precision control and complexity analysis of subdivision-based algorithms in computational geometry. We combine these tools with the continuous amortization framework from exact computation. We use these tools on a well-known example from the subdivision family: the adaptive subdivision algorithm due to Plantinga and Vegter. The only existing complexity estimate on this rather fast algorithm was an exponential worst-case upper bound for its interval arithmetic version. We go beyond the worst-case by considering both average and smoothed analysis, and prove polynomial time complexity estimates for both interval arithmetic and finite-precision versions of the Plantinga–Vegter algorithm.

Keywords Plantinga–Vegter algorithm · Subdivision methods · Complexity

Mathematics Subject Classification 65D18 · 68W40

Editor in Charge: Kenneth Clarkson

Felipe Cucker, Alperen A. Ergür, and Josué Tonelli-Cueto contributed equally to this work.

Felipe Cucker
macucker@cityu.edu.hk

Alperen A. Ergür
alperen.ergur@utsa.edu

Josué Tonelli-Cueto
josue.tonelli.cueto@bizkaia.eu

1 Department of Mathematics, City University of Hong Kong, Hong Kong, China

2 Mathematics Department, University of Texas at San Antonio, One UTSA Circle, San Antonio, TX 78249, USA

3 OURAGAN team, Inria Paris & IMJ-PRG, Sorbonne Université, Paris, France

Introduction

Subdivision based algorithms are ubiquitous in computational geometry. These algorithms have the advantage of simplicity, and often have good practical performance. The two main challenges related to subdivision based algorithms are the control of precision (or a termination criterion), and complexity analysis. As late as summer 2019, complexity analysis aspect of subdivision based geometric algorithms was considered to be "largely open" [40]. In this paper, we contribute to both of the main challenges by introducing a hybrid toolbox that combines condition numbers, high dimensional probability theory, and continuous amortization framework introduced by Burr et al. [12]. To keep our writing focused, and the length of the article finite, we only showcase the toolbox on a well-known member of this large family; the algorithm of Plantinga and Vegter.

Plantinga–Vegter (PV) algorithm is an adaptive subdivison algorithm for meshing curves and surfaces [30]. The algorithm admits an implicit equation of a curve or a surface and outputs an isotopic piecewise linear approximation with controlled Hausdorff distance. The initial paper of Plantinga and Vegter contained no complexity analysis and not even a formal setting fixing either the kind of functions implicitly defining the considered curves and surfaces or the arithmetic used. However, concrete implementations in the paper indicated the efficiency of the algorithm. The algorithm is now widely considered to be very efficient.

The first complexity analysis of the PV algorithm was published thirteen years later by Burr et al. [10] (cf. [11]). The paper of Burr et al. focused on the subdivision procedure of the Plantinga–Vegter algorithm and only analyzed the complexity for polynomials with integer coefficients. The paper provides bounds that are exponential both in the degree d of the input polynomial and in its logarithmic height τ. The discrepancy between the exponential complexity estimate and the practical efficiency of the PV algorithm was marked by the following comment at the end of the paper:

> Even though our bounds are optimal, in practice, these are quite pessimistic [...]

The authors further observe that, following from their Proposition 5.2 (see Theorem 5.2 below) an instance-based analysis of the algorithm (i.e., one yielding a cost that depends on the input at hand) could be derived from the evaluation of a certain integral. And they conclude their paper by writing

> Since the complexity of the algorithm can be exponential in the inputs [size], the integral must be described in terms of additional geometric and intrinsic parameters.

In this paper, we make progress towards these aims by going beyond the worst-case analysis and by using condition numbers. We believe condition numbers are a perfect fit for the latter aim as they provide a geometric and arguably intrinsic parameter.

We analyze the complexity of the PV algorithm in two different versions corresponding, roughly speaking, to its arithmetic complexity and its (arguably more realistic) bit complexity. Our analysis deals with the subdivision routine of the PV algorithm for curves and surfaces as the special cases for $n = 2$ and $n = 3$, but we aim for estimates that hold for any n. We perform both average and smoothed analysis

for the two versions of the PV algorithm, so we provide four different complexity analyses.

The average analysis framework is well known. The smoothed analysis framework might, in contrast, require a bit of an explanation. Suppose we endow the space of n-variate degree d polynomials with a norm $\|\cdot\|$, and a probability measure μ (with as few assumptions as possible on μ). Suppose g is a random polynomial distributed with respect to μ. Then we consider an arbitrary polynomial f, and we fix a tolerance parameter $\sigma > 0$. We consider $q = f + \sigma\|f\|g$ as random perturbation of f with tolerance σ, and conduct average analysis of the PV algorithm for q. This type of estimate could a priori depend on the arbitrary polynomial f. We aim for a uniform estimate that provides an upper bound for any f, and depends only on σ, n, and d. This uniform upper bound will be the smoothed analysis of the PV algorithm. It turns out that this random perturbation idea was already considered in the computational geometry literature in an experimental fashion, and there were aims for building a theoretical framework (see [23, Sect. 4]).

Our main results, Theorems 2.6 and 2.7, provide the four promised estimates on the complexity of the PV algorithm for any number of variables n. For the special case of the plane curves, the average and smoothed analysis of the arithmetic complexity of the PV algorithm are respectively $\mathcal{O}(d^7)$ and $\mathcal{O}(d^7(1+1/\sigma)^3)$. The average and smoothed analysis of the bit complexity are just slightly worse: $\mathcal{O}(d^7\log^2 d)$ and $\mathcal{O}(d^7(1+1/\sigma)^3\log^2 d)$, respectively. These bounds are in marked contrast with the $\mathcal{O}(2^{\tau d^4\log d})$ *worst-case* complexity bound in [10].

For a clear presentation of our contribution and related complexity considerations we need to make a few remarks:

(1) The use of floating-point arithmetic generates numerical errors which accumulate during the computation. An important remark is that, despite this accumulation of errors, our algorithm returns a correct output, a subdivision with the properties we want. It is, in this sense, a *certified* algorithm. At the heart of this remark is the fact that a sufficiently small perturbation of a correct subdivision is still a correct subdivision for a generic (i.e., non-singular) input. Condition numbers allow us to estimate how large this perturbation may be. Then, the fact that we can estimate these condition numbers, we control the precision of the operations' round-off, and we know how these operations are sequenced further allows us to ensure that the subdivision we constructed is close enough to the one we would have done in an error-free context and both yield polygons with the same isotopy type. Needless to say, for input data outside the set satisfying the generic property above our reasoning does not hold. The set of such inputs, referred to as *ill-posed* in numerical analysis, has measure zero. Condition numbers relate to ill-posedness in the sense that the closer a data is to the set of ill-posed inputs the larger becomes its condition number. It is these facts that allows one to establish average and smoothed analysis by means of probabilistic estimates on the condition numbers. This general scheme was proposed in [34]. A more detailed discussion of these issues is in [2, Sect. 9.5]. A relatively early case of a fully studied variable-precision algorithm is in [19]. An account of the use of floating-point arithmetic in computational geometry is given in [23].

(2) Most of the probabilistic analyses for cost measures or condition numbers use the Gaussian measure. This choice is mainly for technical convenience. For the analysis of condition numbers, this goes back to Goldstine and von Neumann [25] and, more recently, resulted in simple bounds for a large class of condition numbers [4, 5, 20, 29]. In the last few years, however, the search for more robust complexity analysis resulted in estimates that hold for a (quite) general family of measures. The family of *sub-Gaussian* measures which includes all compactly supported random variables provides a good testing ground. An analysis of a condition number for these distributions occupies [21, 22]. It is for this class of distributions (sub-Gaussians with an anti-concentration property) that our results, both average and smoothed, are proved.

(3) The subdivision procedure we analyze can be considered at three levels of generality: the *abstract*, in which we only take into account the number of iterations of the subdivision procedure; the *interval*, in which we take also into account the number of arithmetic operations; and the *effective*, in which we take into account not only the number of arithmetic operations, but also the precision that they need, obtaining a realistic estimation of the bit-cost of the algorithm. This division follows a trend for analysing subdivision algorithms initiated by Xu and Yap [39] (cf. [40]). Our condition-based analysis can be applied at each of these three levels, hopefully showing the usefulness of the approach. Whereas this paper focuses on a particular subdivision procedure we believe that the techniques in this paper can be readily applied to other subdivision based algorithms in computational geometry. We note, however, that the complexity analysis in this paper would have been impossible without the *continuous amortization* technique developed in the exact numerical context [8, 12]. In this regard, we hope to trigger a fruitful exchange of ideas between the different approaches to continuous computation and improve our (seemingly preliminary) understanding of the complexity of subdivision algorithms in computational geometry.

Outline

The rest of the paper is structured as follows: We start with a section that contains notation. We beg readers' pardon for this inconvenient start; this seemed the simplest way for getting things clear. Then in Sect. 1 we discuss the Plantinga–Vegter algorithm and the n-dimensional generalization of its subdivision method in the abstract, the interval arithmetic, and the effective versions. Section 2 introduces our randomness model and contains main complexity estimates of this paper. In Sect. 3, we present a geometric framework (read Hilbert space structure) to deal with homogeneous polynomials. In Sect. 4, we introduce the condition number κ_{aff}—both local, i.e., at a point x, and global—along with its main properties. In Sect. 5, we present the existing results on the complexity of Plantinga–Vegter algorithm from [10], and we relate these results to the local condition number. In Sect. 6, we carry out the finite-precision analysis deriving the corresponding bounds for bit-cost. Finally, in Sect. 7, we derive average and smoothed complexity bounds under (quite) general randomness assumptions.

Notation

Throughout the paper, we will assume some familiarity with the basics of differential geometry. For a smooth map $f\colon \mathbb{R}^m \to \mathbb{R}$, $\mathrm{D}_x f\colon \mathrm{T}_x\mathbb{R}^m \cong \mathbb{R}^m \to \mathrm{T}_x\mathbb{R} \cong \mathbb{R}$ denotes the tangent map of f at $x \in \mathbb{R}^m$. We will write $\partial f\colon \mathbb{R}^m \to \mathbb{R}^m$, $x \mapsto \partial f(x)$, when we see it as a smooth function of x. When we want to see ∂f as a vector of formal derivatives, we will write $\partial f(X)$ where X represents formal variables. For general smooth maps between smooth manifolds $F\colon \mathcal{M} \to \mathcal{N}$, we will just write $\mathrm{D}_x F\colon \mathrm{T}_x\mathcal{M} \to \mathrm{T}_{F(x)}\mathcal{N}$ as the tangent map.

In what follows, $\mathcal{P}_{n,d}$ will denote the set of real polynomials in the n variables $X_1, \ldots, X_n$ with degree at most d, $\mathcal{H}_{n,d}$ the set of homogeneous real polynomials in the $n+1$ variables $X_0, X_1, \ldots, X_n$ of degree d, and $\|\cdot\|$ and $\langle \cdot, \cdot \rangle$ will denote the standard norm and inner product in $\mathbb{R}^m$ as well as the Weyl norm and inner product in $\mathcal{P}^m_{n,d}$ and $\mathcal{H}^m_{n,d}$. Given a polynomial $f \in \mathcal{P}_{n,d}$, $f^{\mathsf{h}} \in \mathcal{H}_{n,d}$ will be its homogenization and ∂f the polynomial map given by its partial derivatives. We will denote by the Cyrillic character Ю, 'yu', the central projection (3.1) that maps $\mathbb{R}^n$ into $\mathbb{S}^n$. For details see Sect. 3. Additionally, $V_{\mathbb{R}}(f)$ and $V_{\mathbb{C}}(f)$ will be, respectively, the real and complex zero sets of f.

For a set $S \subset \mathbb{R}^n$, we will denote by $\square S$ the set of n-boxes of the form $x + I^n$, where I is an interval, that are contained in S and, for a given box $B \in \square\mathbb{R}^n$, $m(B)$ will be its middle point, $w(B)$ its width, and $\operatorname{vol} B = w(B)^n$ its volume.

Regarding probabilistic conventions, we will denote the probability of an event by $\mathbb{P}$, random variables by $\mathfrak{x}, \mathfrak{y}, \ldots$ and random polynomials by $\mathfrak{f}, \mathfrak{g}, \mathfrak{q}, \ldots$. The expression $\mathbb{E}_{\mathfrak{x}\in K} g(\mathfrak{x})$ will denote the expectation of $g(\mathfrak{x})$ when $\mathfrak{x}$ is sampled uniformly from the set K and $\mathbb{E}_{\mathfrak{y}} g(\mathfrak{y})$ the expectation of $g(\mathfrak{y})$ with respect to a previously specified probability distribution of $\mathfrak{y}$.

Regarding complexity parameters, n will be the number of variables, d the degree bound, and $N = \binom{n+d}{n}$ the dimension of $\mathcal{P}_{n,d}$. Finally, ln will denote the natural logarithm and log the logarithm in base 2.

1 The Plantinga–Vegter (Subdivision) Algorithm

Given a real smooth hypersurface in $\mathbb{R}^n$ described implicitly by a map $f\colon \mathbb{R}^n \to \mathbb{R}$ and a region $[-a, a]^n$, the Plantinga–Vegter Algorithm constructs a piecewise-linear approximation of the intersection of its zero set $V_{\mathbb{R}}(f)$ with $[-a, a]^n$ isotopic to this intersection inside $[-a, a]^n$. The Plantinga–Vegter algorithm (see Fig. 1 for an illustration[1]) is divided in two phases:

(1) Subdivision phase: In this phase, the Plantinga–Vegter algorithm subdivides $[-a, a]^n$ into smaller and smaller boxes until all the boxes satisfy a certain condition (see (1.1)).
(2) Post-processing phase: In this phase, the Plantinga–Vegter algorithm uses the obtained subdivision to produce a piecewise-linear approximation of the given hypersurface.

[1] This figure is taken from [37, Fig. 5§1].

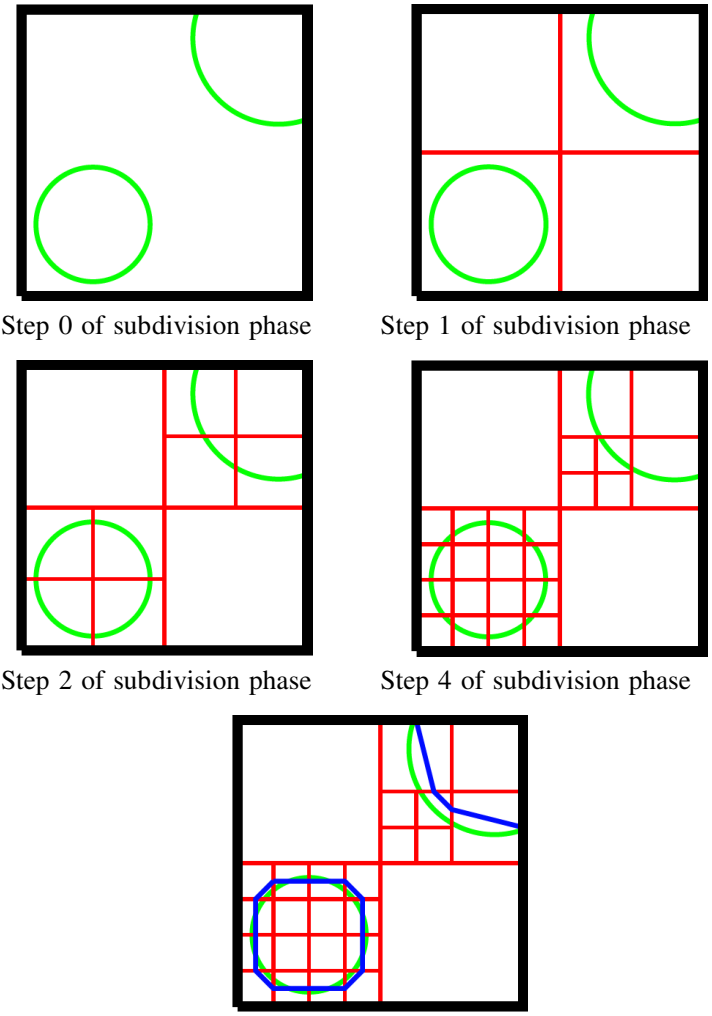

Fig. 1 Plantinga–Vegter applied to $f = X^4 - 6X^3 + 2X^2Y^2 - 6X^2Y - 34\,X^2 - 6XY^2 - 320\,XY + 376\,X + Y^4 - 6Y^3 - 34\,Y^2 + 376\,Y + 3128$ in $[-10, 10]^2$

We will focus on the subdivision phase of the Plantinga–Vegter algorithm. We do this because the complexity of subdivision-based algorithms is usually dominated by the complexity of the subdivision phase. This follows the guidelines of the first complexity analysis given by Burr et al. [10] (cf. [11]).

We note that it would be interesting to incorporate the complexity of the post-processing phase of the algorithm to our estimates in this paper: either the original one by Plantinga–Vegter [30], for $n \leq 3$, or the generalization to higher dimensions

by Galehouse [24], for arbitrary n. We also don't cover existing extensions of the Plantinga–Vegter algorithm to singular curves [9].

From now on, when we say Plantinga–Vegter algorithm we are referring to the Plantinga–Vegter subdivision phase and, following [10], we restrict to the case in which $f : \mathbb{R}^n \to \mathbb{R}$ is a polynomial. We now describe this algorithm at three levels: abstract, interval and effective.

1.1 Abstract Level: Algorithm PV-ABSTRACT

The Plantinga–Vegter algorithm subdivides $[-a, a]^n$ until a certain regularity condition is satisfied in each of the boxes B of the subdivision. Let $h, \tilde{h} : \mathbb{R}^n \to (0, \infty)$ be some fixed positive maps, conveniently chosen (see (1.3) and Remark 1.2 below). Then this regularity condition is

$$C_f(B): \quad \text{either } 0 \notin (hf)(B) \text{ or } 0 \notin \langle (\tilde{h}\partial f)(B), (\tilde{h}\partial f)(B) \rangle. \tag{1.1}$$

Here $f(B)$ stands for the set of values of f on the box B. Note that this condition is satisfied when either B does not contain any zero of f or no pair of gradient vectors of f are orthogonal in B.

In its abstract form, the Plantinga–Vegter algorithm is described in Algorithm 1 below. The StandardSubdivision procedure in the description refers to taking a box B and subdividing it into 2^n boxes of equal size.

Algorithm 1: PV-Abstract

Input : $f : \mathbb{R}^n \to \mathbb{R}$ with interval approximations $\square[hf]$ and $\square[\tilde{h}\nabla f]$
$a \in (0, \infty)$
Precondition : $V_{\mathbb{R}}(f)$ is smooth inside $[-a, a]^n$

```
S̃ ← {[−a, a]^n}
S ← ∅
repeat
    Take B in S̃
    S̃ ← S̃ \ {B}
    if C_f(B) true then
        S ← S ∪ {B}
    else
        S̃ ← S̃ ∪ StandardSubdivision(B)
until S̃ = ∅
return S
```

Output : Subdivision $\mathcal{S} \subseteq \square[-a, a]^n$ of $[-a, a]^n$
Postcondition : For all $B \in \mathcal{S}$, $C_f(B)$ is true

1.2 Interval Level: Algorithm PV-INTERVAL

To check condition $C_f(B)$, we use interval approximations allowing us to certify whether or not 0 is in the image of B under a certain map. Recall that an *interval approximation* [31] of a function $F\colon \mathbb{R}^m \to \mathbb{R}^{m'}$ is a map

$$\square[F]\colon \square\mathbb{R}^m \to \square\mathbb{R}^{m'} \tag{1.2}$$

such that for all $B \in \square\mathbb{R}^m$,

$$F(B) \subseteq \square[F](B). \tag{1.2}$$

A natural choice for the interval approximation of a C^1-function $F\colon \mathbb{R}^m \to \mathbb{R}^{m'}$ is its *standard interval approximation*

$$\square\mathbb{R}^m \ni B \mapsto \square_{\mathrm{std}}[F](B) := F(m(B)) + \sqrt{m}\left(\sup_{x\in B} \|\mathrm{D}_x F\|\right)\left[-\frac{w(B)}{2}, \frac{w(B)}{2}\right]^{m'}$$

where $\mathrm{D}_x F$ is the tangent map of F at x and $\|\mathrm{D}_x F\|$ its operator norm. Note that to construct this one in practice, we need to be able to evaluate F and to compute efficiently upper bounds for $\sup_{x\in B} \|\mathrm{D}_x F\|$. In our case, this is possible due to the fact that we are working with polynomials.

Let $f \in \mathcal{P}_{n,d}$. We will consider

$$h(x) = \frac{1}{\|f\|(1+\|x\|^2)^{(d-1)/2}} \quad \text{and} \quad \tilde{h}(x) = \frac{1}{d\|f\|(1+\|x\|^2)^{d/2-1}} \tag{1.3}$$

along with the maps

$$\widehat{f}\colon x \mapsto h(x)\, f(x) = \frac{f(x)}{\|f\|(1+\|x\|^2)^{(d-1)/2}} \tag{1.4}$$

and

$$\widehat{\partial f}\colon x \mapsto \tilde{h}(x)\partial f(x) = \frac{\partial f(x)}{d\|f\|(1+\|x\|^2)^{d/2-1}} \tag{1.5}$$

where $\|f\|$ is the Weyl norm of f (which we recall in Definition 3.2). In Sect. 3.3 we will prove the following property of $\widehat{f}$ and $\widehat{\partial f}$.

Proposition 1.1 *Let $f \in \mathcal{P}_{n,d}$. Then*

$$\square[hf]\colon B \mapsto \widehat{f}(m(B)) + \left(1+\sqrt{d}\right)\sqrt{n}\left[-\frac{w(B)}{2}, \frac{w(B)}{2}\right] \tag{1.6}$$

is an interval approximation of hf, and

$$\square[\tilde{h}\partial f]\colon B \mapsto \widehat{\partial f}(m(B)) + \left(1+\sqrt{d-1}\right)\sqrt{n}\left[-\frac{w(B)}{2}, \frac{w(B)}{2}\right]^n$$

is an interval approximation of $\tilde{h}\partial f$.

Remark 1.2 A natural question at this point is why we are using interval approximations for hf and $\tilde{h}\partial f$ instead of for f and ∂f. We work with hf and $\tilde{h}f$ for the sake of simplicity. We prefer to work with the simpler interval approximations for hf and $\tilde{h}\partial f$ (shown in Proposition 1.1) than with possibly more complex ones for f and ∂f.

We now note that checking the condition "$0 \notin \langle B, B\rangle$" for a box B can be reduced to checking

$$\sqrt{\frac{n}{2}}\, w(B) \leq \|m(B)\|.$$

To do the latter we will use Lemma 3.6 (which we also prove in Sect. 3.3). Together with the interval approximations in Proposition 1.1, we derive a condition $C_f^{\square}$, implying $C_f(B)$ and easy to check.

Theorem 1.3 *Let $B \in \square\mathbb{R}^n$. If the condition*

$$C_f^{\square}(B) := |\widehat{f}(m(B))| > 2\sqrt{dn}\, w(B) \quad \text{or} \quad \|\widehat{\partial f}(m(B))\| > 2\sqrt{2d}\, n w(B).$$

is satisfied, then $C_f(B)$ is true.

Theorem 1.3 is the basis of the interval version of Algorithm 2.

Algorithm 2: PV-INTERVAL

Input : $f \in \mathcal{P}_{n,d}$
$a \in (0, \infty)$
Precondition : $V_{\mathbb{R}}(f)$ is smooth inside $[-a, a]^n$

$\tilde{\mathcal{S}} \leftarrow \{[-a, a]^n\}$
$\mathcal{S} \leftarrow \varnothing$
repeat
  Take B in $\tilde{\mathcal{S}}$
  $\tilde{\mathcal{S}} \leftarrow \tilde{\mathcal{S}} \setminus \{B\}$
  if $|\widehat{f}(m(B))| > (1 + \sqrt{d})\sqrt{n}\, w(B)$ **then**
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{B\}$
  else if $\|\widehat{\partial f}(m(B))\| > \sqrt{2}(1 + \sqrt{d-1})nw(B)$ **then**
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{B\}$
  else
    $\tilde{\mathcal{S}} \leftarrow \tilde{\mathcal{S}} \cup$ STANDARDSUBDIVISION(B)
until $\tilde{\mathcal{S}} = \varnothing$
return $\mathcal{S}$

Output : Subdivision $\mathcal{S} \subseteq \square[-a, a]^n$ of $[-a, a]^n$
Postcondition : For all $B \in \mathcal{S}$, $C_f(B)$ is true

Remark 1.4 There are other alternatives for interval approximations and our framework has the flexibility to incorporate these alternatives. For instance, the interval approximations in [10], which we will refer to as BGT, are based on the Taylor expansion at the midpoint. In the interlude at the end of Sect. 5, we will show that our complexity analysis also applies to this interval approximation.

Remark 1.5 We have described Algorithm 2 without any reference to interval approximations. Such references have been replaced by explicit conditions on $|\widehat{f}(m(B))|$ and $\|\widehat{\partial f}(m(B))\|$.

1.3 Effective Level: Algorithm PV-EFFECTIVE

For the effective version (Algorithm 3), we will use floating-point numbers (cf. [2, Sect. O.3.1] or [26, Sect. 1.2]). We do this, instead of using fixed-point or big rationals, because the use of floating-point is computationally cheap, both in time and space. We want to emphasize, however, that our use of floating-point numbers does not compromise the correctness of the algorithm (cf. Corollary 6.4). A floating-point number has the form

$$\pm 0.a_1a_2\dots a_{\mathbf{m}}\, 2^e,$$

where $a_1, \dots, a_m \in \{0, 1\}$ and $e \in \mathbb{Z}$. In general, the *number of significant digits*, $\mathbf{m}$, is fixed during the computation of arithmetic expressions, but it can be updated at different iterations of an algorithm if an increase in precision is needed.

We note that every real number $x \in \mathbb{R}$ has a floating-point approximation $r_{\mathbf{m}}(x)$ with $\mathbf{m}$ digits, such that

$$r_{\mathbf{m}}(x) = x(1+\delta)$$

for some $\delta \in (-2^{-(\mathbf{m}-1)}, 2^{-(\mathbf{m}-1)})$. Moreover, given two floating-point numbers x and y with $\mathbf{m}$ significant digits, we can easily compute

$$r_{\mathbf{m}}(x+y),\quad r_{\mathbf{m}}(x-y),\quad r_{\mathbf{m}}(xy),\quad r_{\mathbf{m}}(x/y),\quad \text{and}\quad r_{\mathbf{m}}(\sqrt{x})$$

in $\mathcal{O}(\mathbf{m}^2)$ bit-operations. Comparisons between floating-point numbers can also be made using this amount of bit-operations.

Remark 1.6 In the above estimation we are ignoring the complexity of adding the exponents or operating with them. In general the size of e is of the order of $|\log|x||$, and so the bit-size of e is of the order of $|\log\log|x||$. This means that, unless the numbers we deal with are enormous, one should not worry about the bit-size of e for cost estimates.

Finite-precision analyses do not rely on the precise form of floating-point numbers but just in some general properties which we now summarize. There is a subset $\mathbb{F} \subset \mathbb{R}$ of *floating-point numbers* (which we assume contains 0), a *rounding map* $r\colon \mathbb{R} \to \mathbb{F}$, and a *round-off unit* $\mathbf{u} \in (0, 1)$ satisfying the following conditions:

- For any $x \in \mathbb{F}$, $r(x) = x$. In particular, $r(0) = 0$.
- For any $x \in \mathbb{R}$, $r(x) = x(1+\delta)$ with $|\delta| \le \mathbf{u}$.

Moreover, for $\circ \in \{+, -, \times, /\}$, there are approximate versions

$$\tilde{\circ}\colon \mathbb{F} \times \mathbb{F} \to \mathbb{F}$$

such that for all $x, y \in \mathbb{F}$,

$$x \,\tilde{\circ}\, y = (x \circ y)(1+\delta) \tag{1.6}$$

for some δ such that $|\delta| < \mathbf{u}$. We also assume that there is

$$\widetilde{\sqrt{}} : \mathbb{F} \to \mathbb{F}$$

such that for all $x \in \mathbb{F}$ with $x \geq 0$,

$$\widetilde{\sqrt{x}} = \sqrt{x}(1+\delta)$$

for some δ such that $|\delta| < \mathbf{u}$. Each of these operations and comparisons between numbers in $\mathbb{F}$ can be done with cost $\mathcal{O}(\log^2(1/\mathbf{u}))$. For the floating-point numbers we described above we have $\mathbf{u} = 2^{-(\mathbf{m}-1)}$.

Once the way we deal with finite precision is clear, we introduce the efficient version of the Plantinga–Vegter algorithm (Algorithm 3 below). We note that the algorithm updates the number of significant digits, $\mathbf{m} := |\log \mathbf{u}| + 1$, depending on the width of the box that is being considered, being able, if necessary, to read the coefficients of f with this updated precision.

Algorithm 3: PV- EFFECTIVE

Input : $f \in \mathcal{P}_{n,d}$
$a \in [1, \infty)$
Precondition : $V_{\mathbb{R}}(f)$ is smooth inside $[-a, a]^n$

$\mathbf{m}_0 \leftarrow 7 + \lceil \log \sqrt{dn} \rceil$
$\tilde{\mathcal{S}} \leftarrow \{[-a, a]^n\}$
$\mathcal{S} \leftarrow \varnothing$
repeat
  Take B in $\tilde{\mathcal{S}}$
  $\tilde{\mathcal{S}} \leftarrow \tilde{\mathcal{S}} \setminus \{B\}$
  $\mathbf{m}_B \leftarrow \mathbf{m}_0 + \lceil \max\{\log a, \log(a/w(B))\} \rceil$
  Switch to floating-point numbers with $\mathbf{m}_B$ significant digits
  if $|\widehat{f}(m(B))| > 4\sqrt{dn}\, w(B)$ **then**
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{B\}$
  else if $\|\widehat{\partial f}(m(B))\| > 6\sqrt{d}\, n w(B)$ **then**
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{B\}$
  else
    $\tilde{\mathcal{S}} \leftarrow \tilde{\mathcal{S}} \cup$ STANDARDSUBDIVISION(B)
until $\tilde{\mathcal{S}} = \varnothing$
return $\mathcal{S}$

Output : Subdivision $\mathcal{S} \subseteq \square[-a, a]^n$ of $[-a, a]^n$
Postcondition : For all $B \in \mathcal{S}$, $C_f(B)$ is true

Remark 1.7 As in the case of Algorithm 2, we could rewrite $|\widehat{f}(m(B))| > 4\sqrt{dn}\, w(B)$ and $\|\widehat{\partial f}(m(B))\| > 6\sqrt{d}\, n w(B)$ in Algorithm 3 by $0 \notin \tilde{\square}[hf]$ and $0 \notin \tilde{\square}[\|\tilde{h}\partial f\|]$,

respectively, for some effective interval approximations $\tilde{\square}$ (in the sense of [40]). Our writing of the algorithm, however, is led by the wish to explicitly describe the interval approximations we use, as noted in Remark 1.5.

2 Main Results

In this section, we outline without proofs the main results of this paper. In the first part, we describe our randomness assumptions for polynomials. In the second one, we give precise statements for our bounds on the average and smoothed complexity of phase I of the Plantinga–Vegter Algorithm with infinite precision. In the last part, we state similar results in the context of finite-precision arithmetic.

2.1 Randomness Model

Most of the literature on random multivariate polynomials considers polynomials with Gaussian independent coefficients and relies on techniques that are only useful for Gaussian measures. We will instead consider a general family of measures relying on robust techniques coming from geometric functional analysis. Let us recall some basic definitions.

(P1) A random variable $\mathfrak{x} \in \mathbb{R}$ is called *centered* if $\mathbb{E}\mathfrak{x} = 0$.
(P2) A random variable $\mathfrak{x} \in \mathbb{R}$ is called *subgaussian* if there exists a K such that for all $p \geq 1$,

$$(\mathbb{E}|\mathfrak{x}|^p)^{1/p} \leq K\sqrt{p}.$$

The smallest such K is called the Ψ_2-*norm* of $\mathfrak{x}$.
(P3) A random variable $\mathfrak{x} \in \mathbb{R}$ satisfies the *anti-concentration property with constant* ρ if

$$\max\{\mathbb{P}(|\mathfrak{x} - u| \leq \varepsilon) \mid u \in \mathbb{R}\} \leq \rho\varepsilon.$$

The sub-Gaussian property (P2) has other equivalent formulations. We refer the interested reader to [38]. We note that the anti-concentration property (P3) is equivalent to having a density (with respect to the Lebesgue measure) bounded by $\rho/2$.

Definition 2.1 A *dobro random polynomial* $\mathfrak{f} \in \mathcal{H}_{n,d}$ with parameters K and ρ is a polynomial

$$\mathfrak{f} := \sum_{|\alpha|=d} \binom{d}{\alpha}^{1/2} \mathfrak{c}_\alpha X^\alpha \tag{2.1}$$

such that the $\mathfrak{c}_\alpha$ are independent centered sub-Gaussian random variables with Ψ_2-norm at most K and anti-concentration property with constant ρ. A *dobro random polynomial* $\mathfrak{f} \in \mathcal{P}_{n,d}$ is a polynomial f such that its homogenization $\mathfrak{f}^{\mathsf{h}}$ is so.

Remark 2.2 The word "dobro" appears in several Slavic languages and it means good. The word "dobra" in Turkish means straight and honest, and the word has similar connotations in Greek.

Some classes of dobro random polynomials of interest are the following three.

(N) A *KSS random polynomial* is a dobro random polynomial such that each $\mathfrak{c}_\alpha$ in (2.1) is Gaussian with unit variance. For this model we have $K\rho = 1/\sqrt{2\pi}$.
(U) A *Weyl random polynomial* is a dobro random polynomial such that each $\mathfrak{c}_\alpha$ in (2.1) has uniform distribution in $[-1, 1]$. For this model we have $K\rho \leq 1$.
(E) For $\ell \geq 2$, an *ℓ-random polynomial* is a dobro random polynomial whose coefficients are independent identically distributed random variables with density function

$$t \mapsto \frac{1}{2\Gamma\,(1+1/\ell)}\, e^{-|t|^\ell}.$$

We have in this case that $\rho \leq 1$ and $K \leq 6/5$.

Remark 2.3 The relevant complexity parameter for a dobro random polynomial $\mathfrak{f} \in \mathcal{P}_{n,d}$ with constants K and ρ is the product $K\rho$. This is so because this product is invariant under scalings of $\mathfrak{f}$ and condition numbers will be scale-invariant. Note that, for $t > 0$, $t\mathfrak{f}$ is still dobro, but with constants tK and ρ/t.

Remark 2.4 If we are interested in integer polynomials, dobro random polynomials may seem inadequate. One may be inclined to consider random polynomials $\mathfrak{f} \in \mathcal{P}_{n,d}$ such that $\mathfrak{c}_\alpha$ is a random integer in the interval $[-2^\tau, 2^\tau]$, i.e., $\mathfrak{c}_\alpha$ is a random integer of bit-size at most τ. As $\tau \to \infty$ and after we normalize the coefficients dividing by 2^τ, this random model converges to that of Weyl random polynomials. Yet, in order to have a more satisfactory understanding of random integer polynomials, one has to consider random variables without a continuous density function. The techniques we employed in this note, coming originally from geometric functional analysis, have already been used to analyze condition numbers of random matrices with such discrete distributions [32, 38].

Remark 2.5 Even though there is a widespread agreement that average-case analysis is a better picture of performance in practice than worst-case analysis, it is not itself without contention. The most common objection to average-case analysis is that its underlying probability distribution may not be an accurate reflection of "real life." In particular, that it may result in bounds that are too "optimistic." An alternate form of analysis, called *smoothed analysis*, was introduced by Spielmann and Teng with the goal of overcoming this objection. The basic idea is to replace "behavior at a random data" by "behavior at a random small perturbation of arbitrary data." We won't attempt to describe the rationale of this setting. This can be read in [35, 36] or in [2, Sect. 2.2.7]. But as our development allows to include smoothed-analysis results without a substantial additional effort, we do so in parts (S) of Theorems 2.6, 2.7, and 5.9.

2.2 Complexity at the Interval and Effective Levels

The following two theorems give bounds for, respectively, the average and smoothed complexity of Algorithms 2 and 3. In both of them, the 'big $\mathcal{O}$' notation is not asymptotic. It refers to the existence of a multiplicative constant, which we do not specify, and holds for all values of a, K, ρ, d, and n.

Theorem 2.6 (complexity of Algorithm 2):

(A) *Let $\mathfrak{f} \in \mathcal{P}_{n,d}$ be a dobro random polynomial with parameters K and ρ. The expected number of boxes in the final subdivision $\mathcal{S}$ of Algorithm 2 on input $(\mathfrak{f}, a)$ is at most*

$$d^n\, N^{(n+1)/2} \max\{1, a^n\}\, 2^{12n\log n+8}(K\rho)^{n+1}$$

and the expected number of arithmetic operations is at most

$$\mathcal{O}\big(d^{n+1}\, N^{(n+3)/2} \max\{1, a^n\}\, 2^{12n\log n+8}(K\rho)^{n+1}\big).$$

(S) *Let $f \in \mathcal{P}_{n,d}$, $\sigma > 0$, and $\mathfrak{g} \in \mathcal{P}_{n,d}$ a dobro random polynomial with parameters $K \geq 1$ and ρ. Then the expected number of n-cubes of the final subdivision $\mathcal{S}$ of Algorithm 2 on input $(\mathfrak{q}_\sigma, a)$ where $\mathfrak{q}_\sigma = f + \sigma\|f\|\mathfrak{g}$ is at most*

$$d^n\, N^{(n+1)/2} \max\{1, a^n\}\, 2^{12n\log n+8}(K\rho)^{n+1}\left(1+\frac{1}{\sigma}\right)^{n+1}$$

and the expected number of arithmetic operations is at most

$$\mathcal{O}\left(d^{n+1}\, N^{(n+3)/2} \max\{1, a^n\}\, 2^{12n\log n+8}(K\rho)^{n+1}\left(1+\frac{1}{\sigma}\right)^{n+1}\right).$$

Theorem 2.7 (complexity of Algorithm 3):

(A) *Let $\mathfrak{f} \in \mathcal{P}_{n,d}$ be a dobro random polynomial with parameters K and ρ. The expected number of boxes in the final subdivision $\mathcal{S}$ of Algorithm 3 on input $(\mathfrak{f}, a)$ is at most*

$$d^n\, N^{(n+1)/2}\, a^n 2^{15n\log n+12}(K\rho)^{n+1}$$

and the expected number of arithmetic operations is at most

$$\mathcal{O}\big(d^{n+1}\, N^{(n+3)/2}\, a^n 2^{15n\log n+12}(K\rho)^{n+1}\big).$$

Moreover, the expected bit-cost of Algorithm 3 on input $(\mathfrak{f}, a)$ is at most

$$\mathcal{O}\big(d^{n+1}\, N^{(n+3)/2}\, a^n\, 2^{15n\log n+12}\log^2(dna)(K\rho)^{n+1}\big),$$

under the assumptions that floating-point arithmetic is done using standard arithmetic and that the cost of operating with the exponents is negligible.

(S) *Let $f \in \mathcal{P}_{n,d}$, $\sigma > 0$, and $\mathfrak{g} \in \mathcal{P}_{n,d}$ a dobro random polynomial with parameters $K \geq 1$ and ρ. Then the expected number of n-cubes of the final subdivision $\mathcal{S}$ of Algorithm 3 on input $(\mathfrak{q}_\sigma, a)$ where $\mathfrak{q}_\sigma = f + \sigma \|f\| \mathfrak{g}$ is at most*

$$d^n\, N^{(n+1)/2}\, a^n\, 2^{15n\log n+12} (K\rho)^{n+1} \left(1+\frac{1}{\sigma}\right)^{n+1}$$

and the expected number of arithmetic operations is at most

$$\mathcal{O}\left(d^{n+1}\, N^{(n+3)/2}\, a^n 2^{15n\log n+12} (K\rho)^{n+1} \left(1+\frac{1}{\sigma}\right)^{n+1}\right).$$

Moreover, the expected bit-cost of Algorithm 3 on input $(\mathfrak{q}_\sigma, a)$ is at most

$$\mathcal{O}\left(d^{n+1}\, N^{(n+3)/2}\, a^n\, 2^{15n\log n+12} \log^2(dna) (K\rho)^{n+1} \left(1+\frac{1}{\sigma}\right)^{n+1}\right),$$

under the assumptions that floating-point arithmetic is done using standard arithmetic and that the cost of operating with the exponents is negligible.

Fix a dimension n, a box $[-a, a]^n$ and a dobro distribution (and with it, the parameters ρ and K). If d is let to vary, $N = \binom{n+d}{n} \leq e^n(1 + d/n)^n$. Hence the bounds of Theorems 2.6 and 2.7 are of the order $d^{(n^2+5n)/2}$. The complexity estimate in [10, Thm. 4.3] reads as follows:

$$2^{\mathcal{O}(d^{n+1}(n\tau + nd\log(nd))n\log a)}$$

with τ being the largest bit-size of the coefficients of f. One can see that the average analysis estimates (and the smoothed analysis, for a fixed σ) are exponentially smaller than this worst-case estimate. This seems to relate better with the efficiency in practice of the Plantinga–Vegter algorithm.

We note, however, that the bound in [10] and our bounds cannot be directly compared. Not only because the former is worst-case and the latter average-case (or smoothed) but because of the different underlying settings: the bound in [10] applies to integer data, ours to real data. Nevertheless, the bounds for the effective version Algorithm 3 apply to the real data under finite precision and provides estimates for the bit complexity.

3 Geometric Framework

There is an extensive literature on norms of polynomials and their relation to norms of gradients in $\mathcal{H}_{n,d}$. The PV algorithm, however, works in the affine space with non-homogenous polynomials. We first establish basic definitions and inequalities that

allow us to translate existing results into the setting of the PV algorithm. After the transfer is completed, we continue with establishing interval approximations.

3.1 Weyl Norm

We first introduce the Weyl inner product on $\mathcal{H}_{n,d}$.

Definition 3.1 The *Weyl inner product* on $\mathcal{H}_{n,d}$ is given by

$$\langle f, g\rangle := \sum_{\alpha} \binom{d}{\alpha}^{-1} f_\alpha g_\alpha$$

for $f = \sum_\alpha f_\alpha X^\alpha$, $g = \sum_\alpha g_\alpha X^\alpha \in \mathcal{H}_{n,d}$; and the *Weyl inner product* on $\mathcal{H}^q_{n,d}$ is given by

$$\langle \mathbf{f}, \mathbf{g}\rangle := \sum_{i=1}^{q} \langle f_i, g_i\rangle$$

for $\mathbf{f} = (f_i), \mathbf{g} = (g_i) \in \mathcal{H}^q_{n,d}$.

To extend this inner product to $\mathcal{P}_{n,d}$, we use the homogeneization map

$$\begin{aligned} {}^{\mathsf{h}}\colon \mathcal{P}_{n,d} &\to \mathcal{H}_{n,d},\\ f &\mapsto f^{\mathsf{h}} := f(X_1/X_0, \ldots, X_n/X_0)X_0^d \end{aligned}$$

and its componentwise extension ${}^{\mathsf{h}}\colon \mathcal{P}^q_{n,d} \to \mathcal{H}^q_{n,d}$.

Definition 3.2 The *Weyl inner product* on $\mathcal{P}^q_{n,d}$ is given by

$$\langle \mathbf{f}, \mathbf{g}\rangle := \langle \mathbf{f}^{\mathsf{h}}, \mathbf{g}^{\mathsf{h}}\rangle$$

for $\mathbf{f}, \mathbf{g} \in \mathcal{P}^q_{n,d}$.

For both $\mathcal{H}^q_{n,d}$ and $\mathcal{P}^q_{n,d}$ the Weyl norm is the norm induced by the Weyl inner product. Note that for $F \in \mathcal{H}^q_{n,d}$, we have that $\partial F(X) \in \mathcal{H}^{q(n+1)}_{n,d-1}$ and so we can talk about the Weyl norm of $\partial F(X)$. Recall that we write explicitly the vector X of indeterminates to indicate that we are working with $\partial F(X)$ as a vector of formal derivatives of F. The following proposition comes in handy.

Proposition 3.3 *Let* $\mathbf{f} \in \mathcal{H}^q_{n,d}$ *and* $y \in \mathbb{S}^n$. *Then,* (1) $\|\mathbf{f}(y)\| \le \|\mathbf{f}\|$, (2) $\left\|\mathrm{D}_y\mathbf{f}_{|\mathrm{T}_y\mathbb{S}^n}\right\| \le \sqrt{d}\|\mathbf{f}\|$, *and* (3) $\|\partial\mathbf{f}(X)\| \le d\|\mathbf{f}\|$.

Proof (1) is [2, Lem. 16.6], (2) is the Exclusion Lemma [2, Lem. 19.22], and (3) can be shown by a direct computation, arguing as in the proof of [2, Lem. 16.46]. Alternatively, one can also see [37, $1^{\S 1}$] for a direct account of the proofs. □

3.2 Central Projection and Homogeneization

Let $\mathrm{IO}\colon \mathbb{R}^n \to \mathbb{S}^n$ be the map given by

$$\mathrm{IO}\colon x \mapsto \frac{1}{\sqrt{1+\|x\|^2}}\begin{pmatrix}1\\ x\end{pmatrix}. \tag{3.1}$$

One can see that IO is the map induced by the central projection of $\{1\}\times\mathbb{R}^n$ onto the sphere $\mathbb{S}^n$ and that this map induces a diffeomorphism between $\mathbb{R}^n$ and the upper half of $\mathbb{S}^n$.

Given $\mathbf{f}\in\mathcal{P}^q_{n,d}$, we observe that

$$\mathbf{f}^{\mathsf{h}}(\mathrm{IO}(x)) = \frac{\mathbf{f}(x)}{(1+\|x\|^2)^{d/2}}, \tag{3.2}$$

and so, by the chain rule,

$$\mathrm{D}_{\mathrm{IO}(x)}\mathbf{f}^{\mathsf{h}}\mathrm{D}_x\mathrm{IO} = \frac{\mathrm{D}_x\mathbf{f}}{(1+\|x\|^2)^{d/2}} - \frac{d\cdot\mathbf{f}(x)\,x^{\mathrm{T}}}{(1+\|x\|^2)^{d/2+1}} \tag{3.3}$$

where $\mathrm{D}_y\mathbf{f}^{\mathsf{h}}\colon \mathrm{T}_y\mathbb{R}^{n+1}\cong\mathbb{R}^{n+1}\to \mathrm{T}_{\mathbf{f}^{\mathsf{h}}(y)\mathbb{R}^q}\cong\mathbb{R}^q$, $\mathrm{D}_x\mathbf{f}\colon \mathrm{T}_x\mathbb{R}^n\cong\mathbb{R}^n\to\mathrm{T}_{\mathbf{f}(x)}\mathbb{R}^q\cong\mathbb{R}^q$, and $\mathrm{D}_x\mathrm{IO}\colon \mathrm{T}_x\mathbb{R}^n\to\mathrm{T}_{\mathrm{IO}(x)}\mathbb{S}^n=\mathrm{IO}(x)^{\perp}$ are respectively the tangent maps of $\mathbf{f}^{\mathsf{h}}$, $\mathbf{f}$, and IO.

It is important to note that IO deforms the metric. For each $x\in\mathbb{R}^n$, we can see that the singular values of $\mathrm{D}_x\mathrm{IO}$ are

$$\sigma_1(\mathrm{D}_x\mathrm{IO}) = \ldots = \sigma_{n-1}(\mathrm{D}_x\mathrm{IO}) = \frac{1}{\sqrt{1+\|x\|^2}},\quad \sigma_n(\mathrm{D}_x\mathrm{IO}) = \frac{1}{1+\|x\|^2},$$

and so, in particular,

$$\|\mathrm{D}_x\mathrm{IO}\| = \frac{1}{\sqrt{1+\|x\|^2}}. \tag{3.4}$$

With the above, we next prove a version of Proposition 3.3 for $\mathcal{P}^q_{n,d}$.

Proposition 3.4 *Let $\mathbf{f}\in\mathcal{P}^q_{n,d}$ be a polynomial map. Then the map*

$$\mathbf{F}\colon x\mapsto \frac{\mathbf{f}(x)}{\|\mathbf{f}\|(1+\|x\|^2)^{(d-1)/2}}$$

is $(1+\sqrt{d})$-Lipschitz and, for all x, $\|\mathbf{F}(x)\|\le\sqrt{1+\|x\|^2}$.

Proof For the Lipschitz property, it is enough to bound the norm of the derivative of the map by $1+\sqrt{d}$. Due to (3.2),

$$\mathbf{F}(x) = \sqrt{1+\|x\|^2}\,\frac{\mathbf{f}^{\mathsf{h}}(\mathrm{IO}(x))}{\|\mathbf{f}\|} \tag{3.5}$$

and so, by the chain rule,

$$\mathrm{D}_x\mathbf{F} = \frac{\mathbf{f}^{\mathsf{h}}(\mathrm{IO}(x))}{\|\mathbf{f}\|} \cdot \frac{x^{\mathrm{T}}}{\sqrt{1+\|x\|^2}} + \sqrt{1+\|x\|^2}\,\frac{\mathrm{D}_{\mathrm{IO}(x)}\mathbf{f}\,\mathrm{D}_x\mathrm{IO}}{\|\mathbf{f}\|}.$$

Now, by the triangle inequality,

$$\|\mathrm{D}_x\mathbf{F}\| \le \frac{\|\mathbf{f}^{\mathsf{h}}(\mathrm{IO}(x))\|}{\|\mathbf{f}\|} \cdot \frac{\|x\|}{\sqrt{1+\|x\|^2}} + \sqrt{1+\|x\|^2}\,\frac{\|\mathrm{D}_{\mathrm{IO}(x)}\mathbf{f}\,\mathrm{D}_x\mathrm{IO}\|}{\|\mathbf{f}\|}.$$

On the one hand,

$$\frac{\|\mathbf{f}^{\mathsf{h}}(\mathrm{IO}(x))\|}{\|\mathbf{f}\|} \le 1,$$

by Proposition 3.3 (1). On the other hand,

$$\begin{aligned}\left\|\mathrm{D}_{\mathrm{IO}(x)}\mathbf{f}\,\mathrm{D}_x\mathrm{IO}\right\| &= \left\|\mathrm{D}_{\mathrm{IO}(x)}\mathbf{f}_{|\mathrm{T}_{\mathrm{IO}(x)}\mathbb{S}^n}\,\mathrm{D}_x\mathrm{IO}\right\| \\ &\le \left\|\mathrm{D}_{\mathrm{IO}(x)}\mathbf{f}_{|\mathrm{T}_{\mathrm{IO}(x)}\mathbb{S}^n}\right\| \cdot \|\mathrm{D}_x\mathrm{IO}\| \le \frac{\sqrt{d}\,\|\mathbf{f}\|}{\sqrt{1+\|x\|^2}},\end{aligned}$$

by Proposition 3.3 (2) and (3.4). Hence

$$\|\mathrm{D}_x\mathbf{F}\| \le \frac{\|x\|}{\sqrt{1+\|x\|^2}} + \sqrt{d} \le 1 + \sqrt{d}$$

as we wanted to show. The claim about $\|\mathbf{F}(x)\|$ follows from Proposition 3.3 (1) applied to the expression (3.5) for $\mathbf{F}$. □

3.3 Interval Approximations

Recall that our interval approximations, given in Proposition 1.1, rely on the functions $\widehat{f}$ and $\widehat{\partial f}$ given, respectively, in (1.4) and (1.5). The following lemma will give us the justification of our interval approximations, and with it a proof of Proposition 1.1.

Lemma 3.5 *Let $f \in \mathcal{P}_{n,d}$. Then:*

(i) *The map $\widehat{f}$ given in (1.4) is $(1+\sqrt{d})$-Lipschitz and for all $x \in \mathbb{R}^n$, it satisfies $|\widehat{f}(x)| \le \sqrt{1+\|x\|^2}$.*
(ii) *The map $\widehat{\partial f}$ given in (1.5) is $(1+\sqrt{d-1})$-Lipschitz and for all $x \in \mathbb{R}^n$, it satisfies $\|\widehat{\partial f}(x)\| \le \sqrt{1+\|x\|^2}$.*

Proof of Proposition 1.1 It is a straightforward consequence of the Lipschitz properties in Lemma 3.5. □

Proof of Lemma 3.5 (i) Apply Proposition 3.4 with $\mathbf{f} = f$, then $\widehat{f} = \mathbf{F}$ and both claims follow. (ii) Apply Proposition 3.4 with $\mathbf{f} = f$, then $\widehat{\partial f} = \|\partial f(X)\| \, \mathbf{F}/(d\|f\|)$ and the claims follow since $\|\partial f(X)\|/(d\|f\|) \leq 1$ by Proposition 3.3 (3). □

Once we have shown that our interval approximations are so, we show Theorem 1.3 which reduces the interval condition $C_f(B)$ to the condition $C_f^{\square}(B)$ at a point.

Lemma 3.6 *Let $x \in \mathbb{R}^n$ and $s \in [0, 1/\sqrt{2}]$. Then for all $v, w \in B(x, s\,\|x\|)$, we have $\langle v, w\rangle > \|v\|\|w\|(1 - 2s^2) \geq 0$.*

Proof of Theorem 1.3 By the standard ℓ_2-ℓ_∞ inequality—which states that $\|x\| \leq \sqrt{n}\|x\|_\infty$ for $x \in \mathbb{R}^n$—interval approximations of Proposition 1.1 satisfy that for all $B \in \square\mathbb{R}^n$

$$\operatorname{dist}\left((hf)(m(B)), \square[hf](B)\right) \leq \left(1 + \sqrt{d}\right)\frac{\sqrt{n}\, w(B)}{2} \quad \text{and} \tag{3.6}$$

$$\operatorname{dist}\left((\tilde{h}\partial f)(m(B)), \square[\tilde{h}\partial f](B)\right) \leq \left(1 + \sqrt{d-1}\right)\frac{n w(B)}{2} \tag{3.7}$$

where dist is the usual Euclidean distance.

When the inequality on $\widehat{f}(m(B))$ in $C_f^{\square}(B)$ is satisfied, then (3.6) guarantees that $0 \notin \square[hf](B)$. Similarly, when the inequality on $\widehat{\partial f}(m(B))$ in $C_f^{\square}(B)$ is satisfied, then (3.7) and Lemma 3.6 (with $s = 1/\sqrt{2}$) guarantee that $0 \notin \left\langle \square[\tilde{h}\partial f](B), \square[\tilde{h}\partial f](B)\right\rangle$. Hence $C_f^{\square}(B)$ implies $C_f(B)$. □

Proof of Lemma 3.6 Let $s = \cos\theta$, so that $\theta \in [0, \pi/4]$, $c = \sqrt{1 - s^2}$ and $K_c := \{u \in \mathbb{R}^n \mid \langle x, u\rangle \geq \|x\|\,\|u\|\,c\}$ the convex cone of those vectors u whose angle $\widehat{xu}$ with x, is at most θ.

Given $v, w \in K_c$, we have, by the triangle inequality, that $\angle(vw) \leq \angle(vx) + \angle(xw) \leq 2\theta \leq \pi/2$ (here $\angle$ denotes angle). Thus

$$\cos\angle(vw) \geq \cos\left(\angle(vx) + \angle(xw)\right) \geq \cos 2\theta = 1 - 2s^2 \geq 0.$$

And so, it is enough to show that $B(x, s\|x\|) \subseteq K_c$ or, equivalently, to show that $\operatorname{dist}(x, \partial K_c) \leq s\|x\|$.

Now, $\operatorname{dist}(x, \partial K_c) = \min\{\|x - u\| \mid u \in K_c,\ \langle x, u\rangle = \|x\|\,\|u\|\,c\}$ and this minimum equals the distance of x to a line having an angle θ with x, which is $\|x\|\,s$. □

4 Condition Number

As other numerical algorithms in computational geometry, the Plantinga–Vegter algorithm has a cost which significantly varies with inputs of the same size, even if the coefficients are rational and inputs have the same bit-size. One wants to explain this variation in terms of geometric properties of the input. Condition numbers allow for such an explanation.

Definition 4.1 [3, 13, 18] Given $f \in \mathcal{H}_{n,d}$, $f \neq 0$, the *local condition number* of f at $y \in \mathbb{S}^n$ is

$$\kappa(f, y) := \frac{\|f\|}{\sqrt{f(y)^2 + \|\mathrm{D}_y f_{|\mathrm{T}_y\mathbb{S}^n}\|^2/d}}.$$

Given $f \in \mathcal{P}_{n,d}$, the *local affine condition number* of f at $x \in \mathbb{R}^n$ is

$$\kappa_{\mathrm{aff}}(f, x) := \kappa(f^{\mathsf{h}}, \mathrm{IO}(x)).$$

4.1 What Does κ_{aff} Measure?

The nearer the hypersurface $V_{\mathbb{R}}(f)$ is to having a singularity at $x \in \mathbb{R}^n$, the smaller are the boxes drawn by the Plantinga–Vegter algorithm around x. Instead of controlling how near x is of being a singularity of f, we perform a Copernican turn and we control instead how near f is of having a singularity at x. This is precisely what $\kappa_{\mathrm{aff}}(f, x)$ does.

Theorem 4.2 (condition number theorem) *Let $x \in \mathbb{R}^n$ and*

$$\Sigma_x := \{g \in \mathcal{P}_{n,d} \mid g(x) = 0,\ \mathrm{D}_x g = 0\} \tag{4.1}$$

be the set of polynomials in $\mathcal{P}_{n,d}$ that have a singularity at x. Then for every $f \in \mathcal{P}_{n,d}$,

$$\frac{\|f\|}{\kappa_{\mathrm{aff}}(f, x)} = \mathrm{dist}(f, \Sigma_x),$$

where dist *is the distance induced by the Weyl norm on $\mathcal{P}_{n,d}$.*

Proof This is a reformulation of [3, Thm. 4.4] (cf. [2, Prop. 19.6]). □

Theorem 4.2 provides a geometric interpretation of the local condition number, and a corresponding "intrinsic" complexity parameter as desired by Burr et al. [10, 11]. The next result is an essential tool for our probabilistic analyses. Note that, in the case under consideration, Σ_x is a linear subspace of codimension $n + 1$ inside $\mathcal{P}_{n,d}$.

Corollary 4.3 *Let $x \in \mathbb{R}^n$ and let $\mathrm{R}_x : \mathcal{P}_{n,d} \to \Sigma_x^{\perp}$ be the orthogonal projection onto the orthogonal complement of the linear subspace Σ_x. Then*

$$\kappa_{\mathrm{aff}}(f, x) = \frac{\|f\|}{\|\mathrm{R}_x f\|}.$$

Proof We have that $\mathrm{dist}(f, \Sigma_x) = \|\mathrm{R}_x f\|$ since Σ_x is a linear subspace. Hence Theorem 4.2 finishes the proof. □

4.2 Regularity Inequality

After doing our Copernican turn, we can control how near is $f \in \mathcal{P}_{n,d}$ of having a singularity at $x \in \mathbb{R}^n$. The regularity inequality [6, Prop. 3.6] (cf. [37, Prop. $1^{\S 2}3$]) allows us to recover how near is x of being a singularity of f. More precisely, the regularity inequality gives lower bounds for the value of the function or its derivative in terms of the condition number.

Proposition 4.4 (regularity inequality) *Let $f \in \mathcal{P}_{n,d}$ and $x \in \mathbb{R}^n$. Then either*

$$|\widehat{f}(x)| > \frac{1}{2\sqrt{2d}\,\kappa_{\mathrm{aff}}(f,x)} \quad \text{or} \quad \|\widehat{\partial f}(x)\| > \frac{1}{2\sqrt{2d}\,\kappa_{\mathrm{aff}}(f,x)}.$$

Proof Without loss of generality assume that $\|f\| = 1$. Let $y := \mathrm{IO}(x)$, $g := f^{\mathsf{h}}$, and assume that the first inequality does not hold. Then, by (3.2),

$$|g(y)| \leq \frac{1}{2\sqrt{2d}\,\kappa(g,y)\sqrt{1+\|x\|^2}}.$$

Now,

$$\frac{1}{\sqrt{2}\,\kappa(g,y)} \leq \max\left\{|g(y)|, \frac{\|\partial_y g|_{\mathrm{T}_y\mathbb{S}^n}\|}{\sqrt{d}}\right\} = \frac{\|\partial_y g|_{\mathrm{T}_y\mathbb{S}^n}\|}{\sqrt{d}},$$

since $|g(y)| < 1/(\sqrt{2}\,\kappa(g,y))$. Thus, by (3.3) and (3.4), we get

$$\frac{1}{\sqrt{2}\,\kappa(g,y)} \leq \left\| \frac{\mathrm{D}_x f}{(1+\|x\|^2)^{d/2}} - \frac{df(x)\,x^T}{(1+\|x\|^2)^{d/2+1}} \right\| \left(\frac{1+\|x\|^2}{\sqrt{d}}\right).$$

We divide by $\sqrt{d}$ and use the triangle inequality to obtain

$$\begin{aligned}\frac{1}{\sqrt{2d}\,\kappa(g,y)} &\leq \frac{\|\mathrm{D}_x f\|}{d\,(1+\|x\|^2)^{d/2-1}} + \frac{|f(x)|}{(1+\|x\|^2)^{(d-1)/2}} \cdot \frac{\|x\|}{\sqrt{1+\|x\|^2}} \\ &= \|\widehat{\partial f}(x)\| + |\widehat{f}(x)|\,\frac{\|x\|}{\sqrt{1+\|x\|^2}}.\end{aligned}$$

By our assumption and $\|x\| < \sqrt{1+\|x\|^2}$, the above inequality implies

$$\frac{1}{\sqrt{2d}\,\kappa(g,y)} < \|\widehat{\partial f}(x)\| + \frac{1}{2\sqrt{2d}\,\kappa_{\mathrm{aff}}(f,x)},$$

from where the desired inequality follows. □

5 Complexity Analysis of the Interval Version

We analyze the complexity of Algorithm 2 in terms of the number of arithmetic operations the algorithm performs. This task reduces to estimating the number of boxes in the final subdivision produced by the algorithm. At the interval level, this is so, because each iteration of the algorithm takes the same number of arithmetic operations and the number of iterations is bounded by twice the number of final cubes. This was the underlying strategy in [10].

5.1 Local Size Bound Framework

The original analysis in [10] was based on the notion of local size bound.

Definition 5.1 A *local size bound* for $C\colon \square\mathbb{R}^n \to \{\texttt{True}, \texttt{False}\}$ is a function $b\colon \mathbb{R}^n \to [0,\infty)$ such that for all $x \in \mathbb{R}^n$,

$$b(x) \leq \inf\{\mathrm{vol}(B) \mid x \in B \in \square\mathbb{R}^n \text{ and } C(B) = \texttt{False}\}.$$

The idea behind the local size bound is that it gives us the size from which every box containing x satisfies C. In our case, we will apply this to the condition $C_f^{\square}$ introduced in Theorem 1.3. The following result, based on the notion of *continuous amortization* developed by Burr et al. [8, 12] is proven in [10, Prop. 5.2].

Theorem 5.2 *The number of boxes in the final subdivision $\mathcal{S}$ returned by Algorithm 2 on input (f, a) is at most*

$$\max\left\{1, \int_{[-a,a]^n} \frac{2^n}{b(x)}\,\mathrm{d}x\right\},$$

where b is a local size bound for $C_f^{\square}$ (of Theorem 1.3). Moreover, the bound is finite if and only if the algorithm terminates.

To effectively use Theorem 5.2 we need explicit constructions for the local size bound.

5.2 Condition-Based Local Size Bound and Complexity

The following result expresses a local size bound for $C_f^{\square}$ in terms of the local condition number $\kappa_{\mathrm{aff}}(f, x)$.

Theorem 5.3 *The map*

$$x \mapsto \frac{1}{(2^{5/2} d n\, \kappa_{\mathrm{aff}}(f, x))^n}$$

is a local size bound for $C_f^{\square}$ (of Theorem 1.3).

Proof Let $x \in \mathbb{R}^n$. Since $x \in B$, $\|x - m(B)\| \leq \sqrt{n}\, w(B)/2$. Hence, by Lemma 3.5 and the regularity inequality (Proposition 4.4), either

$$|\widehat{f}(m(B))| \geq \frac{1}{2\sqrt{2d}\,\kappa_{\mathrm{aff}}(f,x)} - (1+\sqrt{d})\frac{\sqrt{n}\,w(B)}{2} \quad \text{or}$$
$$\|\widehat{\partial f}(m(B))\| \geq \frac{1}{2\sqrt{2d}\,\kappa_{\mathrm{aff}}(f,x)} - (1+\sqrt{d-1})\frac{\sqrt{n}\,w(B)}{2}.$$

This means that $C_f^{\square}(B)$ is true if either

$$2\sqrt{2d}\,(1+\sqrt{d})\sqrt{n}\,\kappa_{\mathrm{aff}}(f,x)\,w(B) < 1 \quad \text{or}$$
$$2\sqrt{2d}\,(1+\sqrt{d-1})\,n\kappa_{\mathrm{aff}}(f,x)\,w(B) < 1.$$

Hence we get that $C_f^{\square}(B)$ is true when both conditions are satisfied and the inequality $1+\sqrt{d} \leq 2\sqrt{d}$ finishes the proof. □

Using the results above, we get the following theorem exhibiting a condition-based complexity analysis of Algorithm 1.

Theorem 5.4 *The number of boxes in the final subdivision $\mathcal{S}$ of Algorithm 2 on input (f,a) is at most*

$$d^n \max\{1,a^n\}\, 2^{n\log n + 9n/2}\, \mathbb{E}_{\mathfrak{x}\in[-a,a]^n}(\kappa_{\mathrm{aff}}(f,\mathfrak{x})^n).$$

The number of arithmetic operations performed by Algorithm 2 on input (f,a) is at most

$$\mathcal{O}\big(d^{n+1}\max\{1,a^n\}\, 2^{n\log n + 9n/2}\, N\, \mathbb{E}_{\mathfrak{x}\in[-a,a]^n}(\kappa_{\mathrm{aff}}(f,\mathfrak{x})^n)\big).$$

Proof The first statement follows from Theorems 5.2 and 5.3 combined with the fact that $\int_{[-a,a]^n} \kappa_{\mathrm{aff}}(f,x)^n\, \mathrm{d}x$ equals $(2a)^n\, \mathbb{E}_{\mathfrak{x}\in[-a,a]^n}(\kappa_{\mathrm{aff}}(f,\mathfrak{x})^n)$. The latter follows from the fact that one performs $\mathcal{O}(dN)$ arithmetic operations to test $C_f^{\square}$ and that the number of boxes that the algorithm generates is at most two times the number of final boxes. □

The above condition-based complexity estimate will become the main tool to prove Theorem 2.6 in Sect. 7 where we will study the quantity $\mathbb{E}_{\mathfrak{x}\in[-a,a]^n}(\kappa_{\mathrm{aff}}(f,\mathfrak{x})^n)$ for random $\mathfrak{f}$.

In the literature on numerical algorithms in real algebraic geometry [3, 6, 7, 15–18], it is customary the use the following global condition number

$$\kappa_{\mathrm{aff}}(f) := \max_{x\in[-a,a]^n} \kappa_{\mathrm{aff}}(f,x).$$

The quantity $\mathbb{E}_{\mathfrak{x}\in[-a,a]^n}(\kappa_{\mathrm{aff}}(f,\mathfrak{x})^n)$ in Theorem 5.4 is an average quantity, whereas the condition number $\kappa_{\mathrm{aff}}(f)$ is a global supremum. The average quantity has finite

expectation (over ƒ), whereas the global supremum does not admit a bounded first moment. This shows that a condition-based precision control combined with adaptive complexity techniques such as continuous amortization may lead to substantial improvements in computational real algebraic geometry.

5.3 Interlude: Complexity of the Interval Version of [10]

In [10], Burr et al. gave an interval version of Algorithm 1 different from Algorithm 2 based in the BGT interval approximation which relies on Taylor series. We provide a condition-based and probabilistic complexity analysis of this algorithm, although only for the interval version, on which we only bound the number of cubes and not the number of arithmetic operations. We recall that Burr et al. [10] showed that

$$\mathcal{C}(f,x) := \min\left\{\frac{2^{n-1}d/\ln(1+2^{2-2n})+\sqrt{n}/2}{\operatorname{dist}(x,V_{\mathbb{C}}(f))}, \frac{2^{2n}(d-1)/\ln(1+2^{2-4n})+\sqrt{n/2}}{\operatorname{dist}((x,x),V_{\mathbb{C}}(g_f))}\right\},$$

where g_f is the polynomial $\langle \mathrm{D}f(X), \partial f(Y)\rangle$, is a local size bound for the condition that their interval version of Algorithm 1 checks.

Theorem 5.5 [10] *The map*

$$x \mapsto \frac{1}{\mathcal{C}(f,x)^n}$$

is a local size bound function for the condition that the BGT interval version of Algorithm 1 checks.

Looking at the definition of $\mathcal{C}(f,x)$ in [10] one can see that $1/\mathcal{C}$ measures how near is x of being a singular zero of f. This is similar to $1/\kappa_{\mathrm{aff}}$ which, by Theorem 4.2, measures how near is f of having x as a singular zero. The following result relates these two quantities.

Theorem 5.6 *Let $d > 1$ and $f \in \mathcal{P}_{n,d}$. Then, for all $x \in \mathbb{R}^n$,*

$$\mathcal{C}(f,x) \leq 2^{3n} d^2 \kappa_{\mathrm{aff}}(f,x).$$

Proof Note that Lemma 3.5 holds over the complex numbers as well. Due to this and the fact that $V_{\mathbb{C}}(f) = V_{\mathbb{C}}(\widehat{f})$, we have that

$$|\widehat{f}(x)| \leq (1+\sqrt{d})\operatorname{dist}(x, V_{\mathbb{C}}(f)).$$

Now, if $\sqrt{2}(1+\sqrt{d-1})\operatorname{dist}((y_1,y_2),(x,x)) < \|\widehat{\partial f}(x)\|$, then $\sqrt{2}(1+\sqrt{d-1})\|y_i - x\| < \|\widehat{\partial f}(x)\|$. Thus, by Lemma 3.5, $\sqrt{2}\,\|\widehat{\partial f}(y_i) - \widehat{\partial f}(x)\| < \|\widehat{\partial f}(x)\|$ and so, by Lemma 3.6, $0 \notin \langle \widehat{\partial f}(y_1), \widehat{\partial f}(y_2)\rangle$. Hence

$$\|\widehat{\partial f}(x)\| \leq \sqrt{2}(1+\sqrt{d-1})\operatorname{dist}(x, V_{\mathbb{C}}(g_f)).$$

The bound now follows from Proposition 4.4, together with $2^{3(n-1)}d + \sqrt{n} \leq 2^{3n-2}d$ and

$$\min\left\{\frac{2^{n-1}d}{\ln(1+2^{2-2n})} + \frac{\sqrt{n}}{2}, \frac{2^{2n}(d-1)}{\ln(1+2^{2-4n})} + \sqrt{\frac{n}{2}}\right\} \leq 2^{3n-4}d + \frac{\sqrt{n}}{2}.$$

The latter follows from

$$\frac{1}{\ln(1+2^{2-2n})} \leq 2^{2n-3} \quad \text{and} \quad \frac{1}{\ln(1+2^{2-4n})} \leq 2^{4n-3},$$

which are deduced from first-order approximations of the natural logarithm. □

Theorems 5.5 and 5.6 combine to give an analog of Theorem 5.3 for the BGT interval version of Algorithm 1. Also, [10, Thm. 5.1] provides an analog of Theorem 5.2 in this setting. We can therefore proceed to derive the following result, a BGT version of Theorem 5.4, in the same manner that the latter is derived from Theorems 5.2 and 5.3.

Corollary 5.7 *The number of boxes in the final subdivision $\mathcal{S}$ of the BGT interval version of Algorithm 1 on input (f, a) is at most*

$$d^{2n} \max\{1, a^n\}\, 2^{3n^2+2n}\, \mathbb{E}_{x\in[-a,a]^n}(\kappa_{\mathrm{aff}}(f,x)^n).$$

Remark 5.8 The main difference between $C(f,x)$ and $\kappa(f,x)$ is that $C(f,x)$ is a non-linear quantity and is hard to compute and to analyze, while the local condition number $\kappa(f,x)$—as indicated in Corollary 4.3—is a linear quantity, easier to compute and analyze.

We finish this interlude giving a form of Theorem 2.6 for the BGT version of Algorithm 1 (which, obviously, deals only with number of boxes, not with number of arithmetic operations). It is proved as Theorem 2.6 (see Sects. 7.2 and 7.3) with Corollary 5.7 taking the role of Theorem 5.2.

Theorem 5.9 (A) *Let $\mathfrak{f} \in \mathcal{P}_{n,d}$ be a dobro random polynomial with parameters K and ρ. The expected number of boxes in the final subdivision $\mathcal{S}$ of the BGT interval version of Algorithm 1 on input $(\mathfrak{f}, a)$ is at most*

$$d^{n^2}\, N^{(n+1)/2} \max\{1, a^n\}\, 2^{3n^2+n\log n+7n+15/2}(K\rho)^{n+1}.$$

(S) *Let $f \in \mathcal{P}_{n,d}$, $\sigma > 0$, and $\mathfrak{g} \in \mathcal{P}_{n,d}$ a dobro random polynomial with parameters $K \geq 1$ and ρ. Then the expected number of boxes of the final subdivision $\mathcal{F}$ of the BGT interval version of Algorithm 1 on input $(\mathfrak{q}_\sigma, a)$ where $\mathfrak{q}_\sigma = f + \sigma\|f\|\mathfrak{g}$ is at most*

$$d^{n^2}\, N^{(n+1)/2} \max\{1, a^n\}\, 2^{3n^2+n\log n+7n+15/2}(K\rho)^{n+1}\left(1+\frac{1}{\sigma}\right)^{n+1}.$$

6 Error and Complexity Analysis of the Effective Version

We next work on the framework of floating-point numbers introduced in § 1.3. For an arithmetic expression ϕ and a point $x \in \mathbb{R}$, we will denote by $\mathtt{fl}(\phi(x)) \in \mathbb{F}$ the value obtained when evaluating ϕ at $r(x) \in \mathbb{F}$ using floating-point finite precision. In general, our objective is to show that for such expressions ϕ in our algorithm we have, for some other expression $\psi(x)$ and some $k \geq 1$ satisfying $k\mathbf{u} < 1$,

$$\mathtt{fl}(\phi(x)) = \phi(x) + \psi(x)\,\theta_k$$

where θ_k is any number $\delta \in \mathbb{R}$ satisfying

$$|\delta| \leq \frac{k\mathbf{u}}{1 - k\mathbf{u}}.$$

This is the general strategy in [26, Chap. 3].

6.1 Finite-Precision Computations

We study the errors due to finite-precision in Algorithm 3 and show its correctness. In all what follows, we use *numerical algorithm* to refer to an algorithm meant to be implemented with finite precision and analyzed in terms of error accumulation. This is common terminology.

The following two propositions bound the forward error in the computation of $|\widehat{f}(x)|$ and $\|\widehat{\partial f}(x)\|$. Because their proofs are a variation of well-known results (e.g. [15, Thm. 6.10]) and are more tedious than enlightening, we defer them to an appendix.

Proposition 6.1 *There is a numerical algorithm which, with input $f \in \mathcal{P}_{n,d}$ and $x \in \mathbb{R}^n$, computes $|\widehat{f}(x)|$. This algorithm performs $\mathcal{O}(dN)$ arithmetic operations, and, on input $x \in \mathbb{F}^n$ and $f \in \mathcal{P}_{n,d} \cap \mathbb{F}[X_1, \ldots, X_n]$, the computed value $\mathtt{fl}(|\widehat{f}(x)|)$ satisfies*

$$\mathtt{fl}(|\widehat{f}(x)|) = |\widehat{f}(x)| + \sqrt{1 + \|x\|}\,\theta_{32\,d\log(n+1)}.$$

In particular, if the round-off unit satisfies

$$\mathbf{u} \leq \frac{1}{64\,d\log(n+1)},$$

then for $x \in [-a, a]^n \cap \mathbb{F}^n$,

$$\left|\mathtt{fl}(|\widehat{f}(x)|) - |\widehat{f}(x)|\right| \leq 64\sqrt{2}\,d\sqrt{n+1}\,\log(n+1)\max\{1, a\}\,\mathbf{u}.$$

The above remains true for arbitrary f and x if we apply the algorithm to $r(f)$ and $r(x)$.

Proposition 6.2 *There is a numerical algorithm which, with input $f \in \mathcal{P}_{n,d}$ and $x \in \mathbb{R}^n$, computes $\|\widehat{\partial f}(x)\|$. It performs $\mathcal{O}(dN)$ arithmetic operations, and, on input $x \in \mathbb{F}^n$ and $f \in \mathcal{P}_{n,d} \cap \mathbb{F}[X_1, \ldots, X_n]$, the computed value $\|\widehat{\partial f}(x)\|$ satisfies*

$$\mathtt{fl}(\|\widehat{\partial f}(x)\|) = \|\widehat{\partial f}(x)\| + \sqrt{1 + \|x\|}\,\theta_{32\,d\log(n+1)}.$$

In particular, if the round-off unit satisfies

$$\mathbf{u} \leq \frac{1}{64\,d\log(n+1)},$$

then for $x \in [-a, a]^n \cap \mathbb{F}^n$,

$$\left|\mathtt{fl}(\|\widehat{\partial f}(x)\|) - \|\widehat{\partial f}(x)\|\right| \leq 64\sqrt{2}\,d\sqrt{n+1}\log(n+1)\max\{1, a\}\,\mathbf{u}.$$

The above remains true for arbitrary f and x if we apply the algorithm to $r(f)$ and $r(x)$.

We can now show the correctness of Algorithm 3. We will denote by $\mathtt{fl}(B)$ the rounding $r(B)$ of a box B given by

$$m(\mathtt{fl}(B)) = m(B)(1+\theta_1) \quad \text{and} \quad w(\mathtt{fl}(B)) = w(B)(1+\theta_1).$$

Similarly, we will write $\mathtt{fl}(f)$ to denote the rounding $r(f)$ of f. The next theorem shows that if the round-off unit is sufficiently small, then a floating-point version of condition $C_f^{\square}(B)$ is good enough to check $C_f(B)$.

Theorem 6.3 *Let $B \in \square[-a, a]^n$. If*

$$C_f^{\mathrm{FP}} := \begin{cases} \mathtt{fl}\big(|\widehat{\mathtt{fl}(f)}(m(\mathtt{fl}(B)))|\big) > \mathtt{fl}\big(4\sqrt{d}\sqrt{n+1}\,w(\mathtt{fl}(B))\big) & \text{or} \\ \big(\|\widehat{\partial\mathtt{fl}(f)}(m(\mathtt{fl}(B)))\|\big) > \mathtt{fl}\big(6\sqrt{d}\,(n+1)\,w(\mathtt{fl}(B))\big) \end{cases}$$

and

$$\mathbf{u} \leq \frac{1}{128\sqrt{dn}} \cdot \frac{\min\{1, w(B)\}}{\max\{1, a\}},$$

then $C_f^{\square}(B)$ holds and, hence, so does $C_f(B)$.

Corollary 6.4 *Algorithm 3 is correct.*

Proof of Theorem 6.3 Note that the conditions of Propositions 6.1 and 6.2 are satisfied. Therefore, using our hypothesis on the magnitude of $\mathbf{u}$, we have

$$|\widehat{f}(m(B))| > \mathtt{fl}\big(|\widehat{\mathtt{fl}(f)}(m(\mathtt{fl}(B)))|\big) - \sqrt{d}\log(n+1)\min\{1, w(B)\}, \tag{6.1}$$

$$\|\widehat{\partial f}(m(B))\| > \mathtt{fl}\big(\|\widehat{\partial\mathtt{fl}(f)}(m(\mathtt{fl}(B)))\|\big) - \sqrt{d}\,\log(n+1)\min\{1, w(B)\}. \tag{6.2}$$

By error analysis (Proposition A.1), we have that

$$\texttt{fl}\big(4\sqrt{d}\sqrt{n+1}\,w(\texttt{fl}(B))\big) = 4\sqrt{d}\sqrt{n+1}\,w(B)(1+\theta_8) \quad \text{and} \tag{6.3}$$

$$\texttt{fl}\big(4\sqrt{d}\,(n+1)\,w(\texttt{fl}(B))\big) = 6\sqrt{d}\,(n+1)\,w(B)(1+\theta_8). \tag{6.4}$$

Hence, again by the bound on $\mathbf{u}$, from (6.3) we get

$$\texttt{fl}\big(4\sqrt{d}\sqrt{n+1}\,w(\texttt{fl}(B))\big) > 4\sqrt{d}\sqrt{n+1}\,w(B)\left(1-\frac{1}{8\sqrt{dn}}\cdot\frac{\min\{1,w(B)\}}{\max\{1,a\}}\right) \tag{6.5}$$

and from (6.4)

$$\texttt{fl}\big(4\sqrt{d}(n+1)\,w(\texttt{fl}(B))\big) > 6\sqrt{d}(n+1)\,w(B)\left(1-\frac{1}{8\sqrt{dn}}\cdot\frac{\min\{1,w(B)\}}{\max\{1,a\}}\right). \tag{6.6}$$

Now, combining (6.1) and (6.5), we get

$$\begin{aligned}|\widehat{f}(m(B))| &> 2\sqrt{d}\sqrt{n+1}\,w(B) \\ &+ 2\sqrt{d}\sqrt{n+1}\,w(B)\left(1-\frac{1}{4\sqrt{dn}}\cdot\frac{\min\{1,w(B)\}}{\max\{1,a\}}-\frac{\log(n+1)}{2\sqrt{n+1}}\min\left\{1,\frac{1}{w(B)}\right\}\right)\end{aligned} \tag{6.7}$$

and, combining (6.2) and (6.6),

$$\begin{aligned}\|\widehat{\partial f}(m(B))\| &> 3\sqrt{d}\,(n+1)\,w(B) \\ &+ 3\sqrt{d}\,(n+1)\,w(B)\left(1-\frac{1}{6\sqrt{dn}}\cdot\frac{\min\{1,w(B)\}}{\max\{1,a\}}-\frac{\log(n+1)}{2(n+1)}\min\left\{1,\frac{1}{w(B)}\right\}\right).\end{aligned} \tag{6.8}$$

Now, the term between parentheses in the right-hand side of (6.7) is positive since

$$\begin{aligned}&\frac{1}{4\sqrt{dn}}\cdot\frac{\min\{1,w(B)\}}{\max\{1,a\}}+\frac{\log(n+1)}{2\sqrt{n+1}}\min\left\{1,\frac{1}{w(B)}\right\} \\ &\qquad\le \frac{1}{4\sqrt{dn}}+\frac{\log(n+1)}{2\sqrt{n+1}} \le \frac{1}{4}+\frac{1}{2}<1,\end{aligned}$$

and so is the one in the right-hand side of (6.8) since

$$\begin{aligned}&\frac{1}{6\sqrt{dn}}\cdot\frac{\min\{1,w(B)\}}{\max\{1,a\}}+\frac{\log(n+1)}{2(n+1)}\min\left\{1,\frac{1}{w(B)}\right\} \\ &\qquad\le \frac{1}{6\sqrt{dn}}+\frac{1}{2\sqrt{n+1}} \le \frac{1}{6}+\frac{1}{2\sqrt{2}}<1.\end{aligned}$$

Therefore our claim holds. □

6.2 Complexity of Algorithm 3

We now prove the analogous of Theorem 5.3 in the finite-precision setting. To do so we have to slightly modify the sense of the term 'local size bound' to take finite precision into account.

Definition 6.5 A *local size bound* for C_f^{FP} is a function $b_f^{\mathrm{FP}}\colon \mathbb{R}^n \to [0,\infty)$ such that for all $x \in \mathbb{R}^n$,

$$b_f^{\mathrm{FP}}(x) \leq \inf\left\{ \mathrm{vol}(B) \;\middle|\; \begin{array}{c} x \in B \in \square\mathbb{R}^n,\ C_f^{\mathrm{FP}}(B) = \texttt{False} \\ \text{with } \mathbf{u} \leq \dfrac{1}{128\sqrt{dn}} \cdot \dfrac{\min\{1, w(B)\}}{\max\{1,a\}} \end{array} \right\}.$$

The modification takes into account that the condition C_f^{FP} is checked with sufficiently large precision, as indicated by Theorem 6.3. The theorem below gives us the local size bound for finite precision.

Theorem 6.6 *The map*

$$x \mapsto \frac{1}{(2^6 dn\kappa_{\mathrm{aff}}(f,x))^n}$$

is a local size bound for C_f^{FP} *(of Theorem 6.3).*

Proof The proof is similar to the one of Theorem 6.3. For now on, let $B \in \square\mathbb{R}^n$ be such that $x \in B$. By Propositions 6.1 and 6.2, and the bound on $\mathbf{u}$, we have that

$$\begin{aligned} \texttt{fl}\big(|\widehat{\texttt{fl}(f)}(m(\texttt{fl}(B)))|\big) &> |\widehat{f}(m(B))| - \sqrt{d}\log(n+1)\min\{1, w(B)\} \quad \text{and} \\ \texttt{fl}\big(\|\widehat{\partial\texttt{fl}(f)}(m(\texttt{fl}(B)))\|\big) &> \|\widehat{\partial f}(m(B))\| - \sqrt{d}\log(n+1)\min\{1, w(B)\}. \end{aligned}$$

By error analysis (Proposition A.1),

$$\begin{aligned} 4\sqrt{d}\sqrt{n+1}\, w(B)\left(1 + \frac{1}{8\sqrt{dn}} \cdot \frac{\min\{1, w(B)\}}{\max\{1,a\}}\right) &> \texttt{fl}\big(4\sqrt{d}\sqrt{n+1}\, w(\texttt{fl}(B))\big), \\ 6\sqrt{d}\,(n+1)\, w(B)\left(1 + \frac{1}{8\sqrt{dn}} \cdot \frac{\min\{1, w(B)\}}{\max\{1,a\}}\right) &> \texttt{fl}\big(4\sqrt{d}\,(n+1)\, w(\texttt{fl}(B))\big). \end{aligned}$$

By the regularity inequality (Proposition 4.4) and Lemma 3.5, we know that either

$$\begin{aligned}
&\mathtt{fl}\Big(|\widehat{\mathtt{fl}(f)}(m(\mathtt{fl}(B)))|\Big)\\
&\quad > \frac{1}{2\sqrt{2d}\kappa_{\mathrm{aff}}(f,x)} - \frac{(1+\sqrt{d})\sqrt{n}}{2}\, w(B) - \sqrt{d}\,\log(n+1)\min\{1, w(B)\}\\
&\quad > \frac{1}{2\sqrt{2d}\kappa_{\mathrm{aff}}(f,x)} - 2\sqrt{dn}\, w(B) \qquad \text{or}\\
&\mathtt{fl}\Big(\|\widehat{\partial\mathtt{fl}(f)}(m(\mathtt{fl}(B)))\|\Big)\\
&\quad > \frac{1}{2\sqrt{2d}\kappa_{\mathrm{aff}}(f,x)} - \frac{(1+\sqrt{d-1})\sqrt{n}}{2}\, w(B) - \sqrt{d}\,\log(n+1)\min\{1, w(B)\}\\
&\quad > \frac{1}{2\sqrt{2d}\kappa_{\mathrm{aff}}(f,x)} - 2\sqrt{dn}\, w(B).
\end{aligned}$$

Hence $C_f^{\mathrm{FP}}(B)$ holds as long as

$$\begin{aligned}
&\frac{1}{2\sqrt{2d}\kappa_{\mathrm{aff}}(f,x)} - 2\sqrt{dn}\, w(B)\\
&\quad > 6\sqrt{d}\,(n+1)\, w(B)\left(1 + \frac{1}{8\sqrt{dn}} \cdot \frac{\min\{1, w(B)\}}{\max\{1, a\}}\right),
\end{aligned}$$

which is implied by

$$2^6\, d\,(n+1)\,\kappa_{\mathrm{aff}}(f,x)\, w(B) < 1.$$

This means that $C_f^{\mathrm{FP}}(B)$ is true when $\mathrm{vol}(B) < 1/(2^6 dn\kappa_{\mathrm{aff}}(f,x))^n$, which is what we wanted to show. □

Using continuous amortization [8, 12] (we use the statement in [11, Thm. 5]), we obtain the following condition-based complexity analysis of Algorithm 3.

Theorem 6.7 *The number of boxes in the final subdivision $\mathcal{S}$ of Algorithm 3 on input (f, a) is at most*

$$d^n a^n 2^{n\log n + 8n}\, \mathbb{E}_{\mathfrak{x}\in[-a,a]^n}(\kappa_{\mathrm{aff}}(f,\mathfrak{x})^n).$$

The number of arithmetic operations performed by Algorithm 3 on input (f, a) is at most

$$\mathcal{O}\Big(d^{n+1}\, a^n\, 2^{n\log n + 8n}\, N\, \mathbb{E}_{\mathfrak{x}\in[-a,a]^n}(\kappa_{\mathrm{aff}}(f,\mathfrak{x})^n)\Big).$$

Furthermore, the bit-cost of Algorithm 3 on input (f, a) is at most

$$\mathcal{O}\Big(d^{n+1}\, a^n\, 2^{n\log n + 8n}\, N \log^2(dna)\, \mathbb{E}_{\mathfrak{x}\in[-a,a]^n}(\kappa_{\mathrm{aff}}(f,\mathfrak{x})^n \log^2\kappa_{\mathrm{aff}}(f,x))\Big)$$

under the assumptions that floating-point arithmetic is done using standard arithmetic and that the cost of operating with the exponents is negligible.

Proof The first two claims follow from Theorems 6.6 and 5.2. For the third claim, we recall the following variant of Theorem 5.2 that can be found in [11, Thm. 5]. Let $\mathcal{S}$ be the final subdivision output by Algorithm 3 and $h\colon (0,\infty) \to (0,\infty)$ a continuous map. Then

$$\sum_{B\in\mathcal{S}} h(w(B)) \leq \max\left\{h(2a), \int_{[-a,a]^n} \frac{2^n}{b_f^{\mathrm{FP}}(x)}\, h\left(\frac{b_f^{\mathrm{FP}}(x)^{1/n}}{2}\right) \mathrm{d}x\right\}.$$

Applying Theorem 6.6, we get that $\sum_{B\in\mathcal{S}} h(w(B))$ is bounded by

$$\max\left\{h(2a), 2^{n\log n+7n} d^n \int_{[-a,a]^n} \kappa_{\mathrm{aff}}(f,x)^n\, h(2^5 dn\kappa_{\mathrm{aff}}(f,x))\, \mathrm{d}x\right\}.$$

Now, we note that testing C_f^{FP} at each of the boxes along the way takes at most $\mathcal{O}(dN)$ arithmetic operations and that the number of boxes that the algorithm deals with is at most twice the number of final boxes. Because of this, the bit-cost of the algorithm (ignoring the cost of operating with exponents) in floating-point arithmetic is

$$\mathcal{O}\left(dN \sum_{B\in\mathcal{S}} \mathbf{m}_B^2\right).$$

This is so, because each arithmetic operation takes $\mathcal{O}(\mathbf{m}^2)$ bit-time and $\mathbf{m}_B$ is the largest precision needed to test C_f^{FP} in any box that is an ancestor of B. Hence, by Theorem 6.3 and the relation of $\mathbf{m}_B$ to $\mathbf{u}$, taking

$$h(w(B)) = \mathcal{O}\left(\max\left\{\log^2 2^9\sqrt{dn}a, \log^2 2^9\sqrt{dn}\frac{a}{w(B)}\right\}\right)$$

gives the final bound. □

The above condition-based complexity estimate will become the complexity estimates in Theorem 2.7 in the coming Sect. 7.

7 Probabilistic Analyses

In this section, we prove Theorems 2.6 and 2.7 stated in Sect. 2 using Theorems 5.3 and 6.6 and their corollaries respectively.

7.1 Some Useful Tools

The main tools we are going to use are a tail bound on the norm of a random vector and a small ball type estimate to ensure norm of a random projection is not too small.

Following [37, $5^{\S 1}$], we will give explicit constants avoiding the use of undefined absolute constants. This will require us to sketch some proofs.

Theorem 7.1 *Let $\mathfrak{x} \in \mathbb{R}^N$ be a random vector where each component $\mathfrak{x}_i$ is centered and sub-Gaussian with Ψ_2-norm K. Then for all $t \geq 5K\sqrt{N}$,*

$$\mathbb{P}\left(\|\mathfrak{x}\| \geq t\right) \leq \exp\left(-\frac{t^2}{(5K)^2}\right).$$

Sketch of proof We follow the ideas in [38, Thm. 2.6.3]. Note that $\|\mathfrak{x}\| \geq t$ is equivalent to $e^{s^2\|\mathfrak{x}\|^2} \geq e^{s^2t^2}$. By Markov's inequality and independence,

$$\mathbb{P}\left(\|\mathfrak{x}\| \geq t\right) \leq e^{-s^2t^2}\mathbb{E}e^{s^2\|\mathfrak{x}\|^2} = \prod_{i=1}^{N}\mathbb{E}e^{s^2\mathfrak{x}_i^2}.$$

By assumption, for each i,

$$\mathbb{E}e^{s^{2l}\mathfrak{x}_i^2} = \sum_{l=0}^{\infty}\frac{s^{2l}\mathbb{E}\mathfrak{x}_i^{2l}}{l!} \leq \sum_{l=1}^{\infty}\frac{s^{2l}K^{2l}(2l)^l}{l!} \leq \sum_{l=0}^{\infty}(2eK^2s^l)^l,$$

since $l! \geq (l/e)^l$. Thus, taking $s^2 = 1/(4eK^2)$, we get

$$\mathbb{P}\left(\|\mathfrak{x}\| \geq t\right) = 2^N e^{-t^2/(4eK^2)}.$$

The claim is now trivial assuming $t \geq \sqrt{8e\ln 2}\,K\sqrt{N}$. □

Theorem 7.2 [33, Cor. 1.4] *Let $\mathfrak{x} \in \mathbb{R}^N$ be a random vector where each component $\mathfrak{x}_i$ has the anti-concentration property with constant ρ and $P : \mathbb{R}^N \to \mathbb{R}^N$ an orthogonal projection onto a k-dimensional linear subspace of $\mathbb{R}^N$. Then for all $\varepsilon > 0$,*

$$\mathbb{P}\left(\|P\mathfrak{x}\| \leq \sqrt{k}\varepsilon\right) \leq (3\rho\varepsilon)^k.$$

Sketch of proof Note that by assumption, each $\mathfrak{x}_i$ has probability density (with respect to the Lebesgue measure) bounded by $\rho/2$. Then, by [28, Thm. 1.1], $P\mathfrak{x}$ has probability density (with respect to the Lebesgue measure) bounded by $(\rho/\sqrt{2})^k$. Thus

$$\mathbb{P}\left(\|P\mathfrak{x}\| \leq \sqrt{k}\varepsilon\right) \leq \omega_k\left(\frac{\sqrt{k}\rho}{\sqrt{2}}\right)^k,$$

where ω_k is the volume of the k-dimensional Euclidean ball. Now, $\omega_k k^{k/2} \leq (2e)^{k/2}\pi^{k/2}$, from where the claim follows. □

7.2 Average Complexity Analysis

The following theorem is the main technical result from which the average complexity bound will follow.

Theorem 7.3 *Let $\mathfrak{f} \in \mathcal{P}_{n,d}$ be a dobro random polynomial with parameters K and ρ. For all $x \in \mathbb{R}^n$ and $t \geq e$,*

$$\mathbb{P}\left(\kappa_{\mathrm{aff}}(\mathfrak{f}, x) \geq t\right) \leq 2\left(\frac{N}{n+1}\right)^{(n+1)/2} (15\,K\rho)^{n+1} \frac{(\ln t)^{(n+1)/2}}{t^{n+1}}.$$

Remark 7.4 By [21, (1)], we have $K\rho \geq 1/4$ for a dobro random polynomial $\mathfrak{f}$ with parameters K and ρ. This fact will be used without mention in the bounds below.

Proof of Theorem 7.3 By Corollary 4.3, we have that $\kappa_{\mathrm{aff}}(\mathfrak{f}, x) = \|\mathfrak{f}\|/\|\mathrm{R}_x\mathfrak{f}\|$ with R_x an orthogonal projection onto the $(n+1)$-dimensional linear subspace $\Sigma_x^{\perp}$. By the union bound, for all $u, t > 0$,

$$\mathbb{P}\left(\kappa_{\mathrm{aff}}(\mathfrak{f}, x) \geq t\right) \leq \mathbb{P}\left(\|\mathfrak{f}\| \geq u\right) + \mathbb{P}\left(\|\mathrm{R}_x\mathfrak{f}\| \leq u/t\right). \tag{7.1}$$

We apply now Theorems 7.1 to the first term and 7.2 to the second. Thus for $u > 5K\sqrt{N}$ and $t > 0$,

$$\mathbb{P}\left(\kappa_{\mathrm{aff}}(\mathfrak{f}, x) \geq t\right) \leq \exp\frac{-u^2}{(5K)^2} + \left(\frac{3u\rho}{t\sqrt{n+1}}\right)^{n+1}.$$

We set $u = 5K\sqrt{N \ln t}$, so we get

$$\mathbb{P}\left(\kappa_{\mathrm{aff}}(\mathfrak{f}, x) \geq t\right) \leq t^{-N} + \left(\frac{15\,K\rho\sqrt{N}}{\sqrt{n+1}}\right)^{n+1} \frac{(\ln t)^{(n+1)/2}}{t^{n+1}}$$

for $t \geq e$. The inequality $n+1 \leq N$ and Remark 7.4 finish the proof. □

Theorem 7.3 immediately gives probabilistic bounds for $\mathbb{E}_{\mathfrak{x}\in[-a,a]^n}(\kappa_{\mathrm{aff}}(\mathfrak{f}, \mathfrak{x})^n)$ and $\mathbb{E}_{\mathfrak{x}\in[-a,a]^n}(\kappa_{\mathrm{aff}}(\mathfrak{f}, \mathfrak{x})^n \log^2 \kappa_{\mathrm{aff}}(\mathfrak{f}, \mathfrak{x}))$ for a random $\mathfrak{f}$. The two corollaries below, together with Theorems 5.3 and 6.6, give us the proof of the part (A) of Theorems 2.6 and 2.7.

Theorem 7.5 *Let $\mathfrak{f} \in \mathcal{P}_{n,d}$ be a dobro random polynomial with parameters K and ρ and $\alpha \in [1, n+1)$. Then*

$$\mathbb{E}_{\mathfrak{f}}\mathbb{E}_{\mathfrak{x}\in[-a,a]^n}(\kappa_{\mathrm{aff}}(\mathfrak{f}, \mathfrak{x})^{\alpha}) \leq 4\,\frac{\alpha\sqrt{n+1}}{n+1-\alpha}\left(\frac{N}{n+1-\alpha}\right)^{(n+1)2} (25\,K\rho)^{n+1}.$$

Corollary 7.6 *Let $\mathfrak{f} \in \mathcal{P}_{n,d}$ be a dobro random polynomial with parameters K and ρ. Then*

$$\mathbb{E}_{\mathfrak{f}}\mathbb{E}_{\mathfrak{x}\in[-a,a]^n}(\kappa_{\mathrm{aff}}(\mathfrak{f}, \mathfrak{x})^n) \leq N^{(n+1)/2} 2^{5n+(3/2)\log n+15/2} (K\rho)^{n+1}.$$

Corollary 7.7 *Let $\mathfrak{f} \in \mathcal{P}_{n,d}$ be a dobro random polynomial with parameters K and ρ. Then*

$$\mathbb{E}_{\mathfrak{f}}\mathbb{E}_{\mathfrak{x}\in[-a,a]^n}\left(\kappa_{\mathrm{aff}}(\mathfrak{f},\mathfrak{x})^n \log^2\kappa_{\mathrm{aff}}(\mathfrak{f},\mathfrak{x})\right) \leq N^{(n+1)/2}\, 2^{6n+(3/2)\log n+12}(K\rho)^{n+1}.$$

Proof of Theorem 7.5 By the Fubini–Tonelli theorem,

$$\mathbb{E}_{\mathfrak{f}}\mathbb{E}_{\mathfrak{x}\in[-a,a]^n}(\kappa_{\mathrm{aff}}(\mathfrak{f},\mathfrak{x})^\alpha) = \mathbb{E}_{\mathfrak{x}\in[-a,a]^\alpha}\mathbb{E}_{\mathfrak{f}}(\kappa_{\mathrm{aff}}(\mathfrak{f},\mathfrak{x})^n),$$

so it is enough to have a uniform bound for

$$\mathbb{E}_{\mathfrak{f}}(\kappa_{\mathrm{aff}}(\mathfrak{f},x)^\alpha) = \int_1^\infty \mathbb{P}\left(\kappa_{\mathrm{aff}}(\mathfrak{f},x)^\alpha \geq t\right)\mathrm{d}t.$$

Now, by Theorem 7.3, this is bounded by

$$e^\alpha + 2\left(\frac{N}{\alpha(n+1)}\right)^{(n+1)/2}(15\,K\rho)^{n+1}\int_1^\infty \frac{(\ln t)^{(n+1)/2}}{t^{(n+1)/\alpha}}\,\mathrm{d}t.$$

After the change of variables $t = e^{\alpha s/(n+1-\alpha)}$ the bound becomes

$$\begin{aligned} &e^\alpha + 2\,\frac{\alpha}{n+1-\alpha}\left(\frac{N}{(n+1-\alpha)(n+1)}\right)^{(n+1)/2}(15\,K\rho)^{n+1}\int_1^\infty s^{(n+1)/2}e^{-s}\,\mathrm{d}s \\ &= e^\alpha + 2\,\frac{\alpha}{n+1-\alpha}\left(\frac{N}{(n+1-\alpha)(n+1)}\right)^{(n+1)/2}\Gamma\left(\frac{n+3}{2}\right)(15\,K\rho)^{n+1}, \end{aligned}$$

where Γ is Euler's Gamma function. We note that $e^\alpha \leq e^{n+1}$ and that, by the Stirling estimates,

$$\Gamma\left(\frac{n+3}{2}\right) \leq \sqrt{2\pi}\left(\frac{n+3}{2e}\right)^{(n+2)/2} \leq \sqrt{2\pi}\left(\frac{n+1}{e}\right)^{(n+2)/2}.$$

Combining all these inequalities, we obtain the desired upper bound. □

Proof of Corollary 7.6 We take $\alpha = n$ in Theorem 7.5. □

Proof of Corollary 7.7 Recall that $\log^2 y \leq 5\sqrt{y}$ for $y \geq 1$. Hence

$$\mathbb{E}_{\mathfrak{f}}\mathbb{E}_{\mathfrak{x}\in[-a,a]^n}(\kappa_{\mathrm{aff}}(\mathfrak{f},\mathfrak{x})^n \log^2\,\kappa_{\mathrm{aff}}(\mathfrak{f},\mathfrak{x})) \leq 2^{5/2}\mathbb{E}_{\mathfrak{f}}\mathbb{E}_{\mathfrak{x}\in[-a,a]^n}(\kappa_{\mathrm{aff}}(\mathfrak{f},\mathfrak{x})^{n+1/2})$$

and the claim follows using Theorem 7.5 with $\alpha = n + 1/2$. □

We can finally prove the average complexity bounds in our main theorems.

Proof of Theorem 2.6 (A) The expected number of boxes we want to bound is bounded by the expectation of the estimate for this quantity in Theorem 5.4 with respect to a dobro random $\mathfrak{f} \in \mathcal{P}_{n,d}$, that is,

$$d^n \max\{1, a^n\}\, 2^{n \log n + 9n/2}\, \mathbb{E}_{\mathfrak{f} \in \mathcal{P}_{n,d}} \mathbb{E}_{\mathfrak{x} \in [-a,a]^n} (\kappa_{\mathrm{aff}}(\mathfrak{f}, \mathfrak{x})^n).$$

A bound for the inner double expectation is in Corollary 7.6. The bound for the expected number of operations is similarly derived. □

Proof of Theorem 2.7 (A) Similar to the proof above but using Corollaries 7.6 and 7.7 to get upper bounds for the two expectations (arithmetic cost and, also now, bit-cost). □

7.3 Smoothed Complexity Analysis

The tools used for our average complexity analysis yield also a smoothed complexity analysis (see [35] or [2, Sect. 2.2.7]). We provide this analysis following the lines of [22].

The main idea of smoothed complexity is to have a complexity measure interpolating between worst-case complexity and average-case complexity. More precisely, we are interested in the maximum—over $f \in \mathcal{P}_{n,d}$—of the average cost of the algorithm when the input polynomial has the form

$$\mathfrak{q}_\sigma := f + \sigma \|f\|\, \mathfrak{g} \tag{7.2}$$

with $\mathfrak{g} \in \mathcal{P}_{n,d}$ a dobro random polynomials with parameters $K \geq 1$ and ρ, and $\sigma \in (0, \infty)$. Notice that the perturbation $\sigma \|f\|\, \mathfrak{g}$ of f is proportional to both σ and $\|f\|$. The following lemma shows how Theorems 7.1 and 7.2 apply to this class of random polynomials.

Lemma 7.8 *Let $\mathfrak{q}_\sigma$ be as in (7.2). Then for $t > 1 + \sigma\sqrt{N}$*

$$\mathbb{P}(\|\mathfrak{q}_\sigma\| \geq t\|f\|) \leq \exp \frac{-(t-1)^2}{(\sigma 5K)^2}$$

and, for every $x \in \mathbb{R}^n$,

$$\mathbb{P}(\|\mathrm{R}_x \mathfrak{q}_\sigma\| \leq \varepsilon) \leq \left(\frac{3\rho\varepsilon}{\sigma\, \|f\| \sqrt{n+1}}\right)^{n+1},$$

where R_x is as in Corollary 4.3.

Proof By the triangle inequality we have $\mathbb{P}(\|\mathfrak{q}_\sigma\| \geq t\|f\|) \leq \mathbb{P}(\|\mathfrak{g}\| \geq (t-1)/\sigma)$. Then we apply Theorem 7.1 which finishes the proof of the first claim. The second claim is a direct consequence of Theorem 7.2. □

As in the average case, this leads to a tail bound.

Theorem 7.9 *Let $\mathfrak{q}_\sigma$ be as in (7.2) and $x \in \mathbb{R}^n$. Then for $\sigma > 0$ and $t \geq e$,*

$$\mathbb{P}\left(\kappa_{\mathrm{aff}}(\mathfrak{q}_\sigma, x) \geq t\right) \leq 2\left(\frac{N}{n+1}\right)^{(n+1)/2}(15\,K\rho)^{n+1}\frac{(\ln t)^{(n+1)/2}}{t^{n+1}}\left(1+\frac{1}{\sigma}\right)^{n+1}.$$

Proof We proceed as in the proof of Theorem 7.3, but with Lemma 7.8 using $u = \|f\|(\sigma 5K\sqrt{N \ln t} + 1)$. This gives the desired bound arguing as in that proof after noticing that

$$u \leq \|f\|(1+\sigma)5K\sqrt{N \ln t},$$

which holds since $5K\sqrt{N \ln t} \geq 1$. □

As in the average case, Theorem 7.9 yields probabilistic bounds for both $\mathbb{E}_{\mathfrak{x}\in[-a,a]^n}(\kappa_{\mathrm{aff}}(\mathfrak{f}, \mathfrak{x})^n)$ and $\mathbb{E}_{\mathfrak{x}\in[-a,a]^n}(\kappa_{\mathrm{aff}}(\mathfrak{f}, \mathfrak{x})^n \log^2 \kappa_{\mathrm{aff}}(\mathfrak{f}, \mathfrak{x}))$ for random $\mathfrak{f}$. The two corollaries below, together with Theorems 5.3 and 6.6, give us the proof of the part (S) of Theorems 2.6 and 2.7.

Theorem 7.10 *Let $\mathfrak{q}_\sigma$ be as in (7.2) and $\alpha \in [1, n+1)$. Then for all $f \in \mathcal{P}_{n,d}$ and all $\sigma > 0$,*

$$\mathbb{E}_{\mathfrak{q}_\sigma}\mathbb{E}_{\mathfrak{x}\in[-a,a]^n}(\kappa_{\mathrm{aff}}(\mathfrak{q}_\sigma, \mathfrak{x})^\alpha) \leq 4\,\frac{\alpha\sqrt{n+1}}{n+1-\alpha}\left(\frac{N}{n+1-\alpha}\right)^{(n+1)/2}(25\,K\rho)^{n+1}\left(1+\frac{1}{\sigma}\right)^{n+1}.$$

Proof The proof is as that of Theorem 7.5, but using Theorem 7.9 instead of Theorem 7.3. □

Corollary 7.11 *Let $\mathfrak{q}_\sigma$ be as in (7.2). Then for all $f \in \mathcal{P}_{n,d}$ and all $\sigma > 0$,*

$$\mathbb{E}_{\mathfrak{q}_\sigma}\mathbb{E}_{\mathfrak{x}\in[-a,a]^n}(\kappa_{\mathrm{aff}}(\mathfrak{q}_\sigma, \mathfrak{x})^n) \leq N^{(n+1)/2}\,2^{5n+(3/2)\log n+15/2}(K\rho)^{n+1}\left(1+\frac{1}{\sigma}\right)^{n+1}.$$

Corollary 7.12 *Let $\mathfrak{q}_\sigma$ be as in (7.2). Then for all $f \in \mathcal{P}_{n,d}$ and all $\sigma > 0$,*

$$\mathbb{E}_{\mathfrak{q}_\sigma}\mathbb{E}_{\mathfrak{x}\in[-a,a]^n}(\kappa_{\mathrm{aff}}(\mathfrak{q}_\sigma, \mathfrak{x})^n \log^2 \kappa_{\mathrm{aff}}(\mathfrak{q}_\sigma, \mathfrak{x})) \leq N^{(n+1)/2}\,2^{6n+(3/2)\log n+12}(K\rho)^{n+1}\left(1+\frac{1}{\sigma}\right)^{n+1}.$$

Proof of Corollaries 7.11 and 7.12 We do as in the proof of Corollaries 7.6 and 7.7 but using Theorem 7.10 instead of Theorem 7.5. □

We conclude showing how the smoothed complexity estimates follow.

Proof of Theorem 2.6 (S) The proof is the same as that of Theorem 2.6 (A), but using Corollary 7.11 instead of Corollary 7.6. □

Proof of Theorem 2.7 (S) The proof is the same as that of Theorem 2.6 (A), but using Corollaries 7.11 and 7.12 instead of Corollaries 7.6 and 7.7. □

Acknowledgements We cordially thank Michael Burr and Elias Tsigaridas for useful discussions. We also thank the two anonymous reviewers for their very detailed feedback that greatly helped us to improve this paper. Additionally, J. T.-C. is grateful to Evgenia Lagoda for moral support and Gato Suchen for useful suggestions for this paper.

Declarations

Funding This work was supported by the Einstein Fundation Berlin. F.C. was partially supported by a GRF grant from the Research Grants Council of the Hong Kong SAR (project number CityU 11302418). A.E. is supported by US National Science Foundation grant CCF 2110075. J. T.-C. was by a postdoctoral fellowship of the 2020 "Interaction" program of the *Fondation Sciences Mathématiques de Paris*, and partially supported by ANR JCJC GALOP (ANR-17-CE40-0009), the PGMO grant ALMA, and the PHC GRAPE.

Sources An extended abstract containing some of the results was presented at ISSAC'19 [14]. Some preliminary versions of the results in Sect. 7 was included in the doctoral thesis of Tonelli-Cueto [37].

Appendix A Proofs of Propositions 6.1 and 6.2

We proceed by introducing a new error symbol which will make our manipulations easier, then we recall some fundamental numerical algorithms for computing inner product and monomials and we apply them to the computed quantities during the execution of Algorithm 3.

A.1 The Arithmetic of Error Accumulation

To ease the technique of [26, Chap. 3], we will use the symbol θ_k allowing any real number $k \geq 1$ in the subindex. Note that this does not affect any of the results. As the symbol θ_k might be difficult to parse, let us explain in more detail how it works. Let ϕ be some arithmetic expression. Whenever we write an expression of the form

$$\mathtt{fl}(\phi(x)) = \tilde{\phi}\,(x, \theta_{t_1}, \ldots, \theta_{t_\ell}) \tag{A1}$$

for some arithmetic expression $\tilde{\phi}$ and for some real numbers $t_1, \ldots, t_\ell \geq 1$, we will mean that, as long as $\max\{t_1, \ldots, t_\ell\}\,\mathbf{u} < 1/2$, we have

$$\mathtt{fl}(\phi(x)) = \tilde{\phi}\,(x, \tau_1, \ldots, \tau_\ell)$$

for some

$$\tau_1 \in \left[-\frac{t_1\mathbf{u}}{1-t_1\mathbf{u}}, \frac{t_1\mathbf{u}}{1-t_1\mathbf{u}}\right], \ldots, \tau_\ell \in \left[-\frac{t_\ell\mathbf{u}}{1-t_\ell\mathbf{u}}, \frac{t_\ell\mathbf{u}}{1-t_\ell\mathbf{u}}\right].$$

We note that in this notation we are allowing more freedom as we do not require $t_1, \ldots, t_\ell$ to be integers. Furthermore, and this will make it computationally as useful as Landau notation, we introduce the following additional, asymmetric, notation.

Assume $\max\{t_1,\dots,t_\ell,t'_1,\dots,t'_{\ell'}\}\mathbf{u} < 1/2$ and $x \in \mathbb{R}$. We write

$$\tilde{\phi}(x,\theta_{t_1},\dots,\theta_{t_\ell}) = \tilde{\phi}'\big(x,\theta_{t'_1},\dots,\theta_{t'_{\ell'}}\big) \tag{A2}$$

to mean that for every

$$\tau_1 \in \left[-\frac{t_1\mathbf{u}}{1-t_1\mathbf{u}},\frac{t_1\mathbf{u}}{1-t_1\mathbf{u}}\right],\dots,\tau_\ell \in \left[-\frac{t_\ell\mathbf{u}}{1-t_\ell\mathbf{u}},\frac{t_\ell\mathbf{u}}{1-t_\ell\mathbf{u}}\right],$$

there exist

$$\tau'_1 \in \left[-\frac{t'_1\mathbf{u}}{1-t'_1\mathbf{u}},\frac{t'_1\mathbf{u}}{1-t'_1\mathbf{u}}\right],\dots,\tau'_{\ell'} \in \left[-\frac{t'_{\ell'}\mathbf{u}}{1-t'_{\ell'}\mathbf{u}},\frac{t'_{\ell'}\mathbf{u}}{1-t'_{\ell'}\mathbf{u}}\right]$$

—of course, depending on $\tau_1,\dots,\tau_\ell$—such that

$$\tilde{\phi}(x,\tau_1,\dots,\tau_\ell) = \tilde{\phi}'(x,\tau'_1,\dots,\tau'_{\ell'}).$$

This is consistent with notation (A1) in the sense that if both (A1) and (A2) hold then $\mathtt{fl}(\phi(x)) = \tilde{\phi}'(x,\theta_{t_1},\dots,\theta_{t_\ell})$. This will allow us to mechanically perform the finite precision analysis using the following rules.

Proposition A.1 *For all $s, s' \geq 1$, the following holds for the error symbol:*

(E1) *If $s \leq s'$, $\theta_s = \theta_{s'}$.*
(E2) *$\theta_s + \theta_{s'} + \theta_s\theta_{s'} = \theta_{s+s'}$. In particular, $\theta_s + \theta_{s'} = \theta_{s+s'}$ and $(1+\theta_s)(1+\theta_{s'}) = 1+\theta_{s+s'}$.*
(E3) *$(1+\theta_s)^{-1} = 1+\theta_{2s}$.*
(E4) *$\sqrt{1+\theta_s} = 1+\theta_s$.*
(E5) *For all $t \in \mathbb{R}$, $t\theta_s = |t|\theta_s = \theta_{\max\{1,|t|\}s}$.*
(E6) *For all $t, t' \in \mathbb{R}$, $t\theta_s + t'\theta_{s'} = (|t|+|t'|)\theta_{\max\{s,y\}}$.*
(E7) *For all $t, t' \in (0,\infty)$, if $t < t'$, then $t\theta_s = t'\theta_s$.*
(E8) *$|1+\theta_s| = 1+\theta_s$.*

Proof This follows from [26, Lem. 3.1 and 3.3] □

The definition and properties of θ follow the lines of classical error analysis, as e.g., in [26, Chap. 3]. Our presentation may differ in minor details which we have chosen for our own convenience. In all what follows, the round-off unit $\mathbf{u}$ is always sufficiently small, so that the inequalities $t\mathbf{u} < 1/2$ hold true for the values of t at hand. As is customary in finite-precision analyses, we will not explicitly point to these bounds.

A.2 Basic Finite Precision Algorithms

The following two propositions show the nice properties of the numerical computations that underlie the Algorithm 3. Their statements refer to three aspects: (1) the number

of arithmetic operations performed, (2) error estimates for a given input, and (3) error estimates for approximate inputs. From these bounds we can obtain bit-complexity estimates, as floating-point operations take $\mathcal{O}(|\log \mathbf{u}|^2)$-time (this being non-tight, one can obtain better bounds using fast multiplication algorithms).

An algorithm computing inner products with sharper error bounds was recently analyzed in [1] (see also [27] for a survey on another family of recent improvements in this respect). For our purposes, however, the simpler Proposition A.2 is sufficient.

Proposition A.2 *There is a numerical algorithm which, with input $x, y \in \mathbb{R}^m$, computes $\langle x, y\rangle$. This algorithm satisfies the following:*

(i) *It performs $\mathcal{O}(m)$ arithmetic operations.*
(ii) *On input $x, y \in \mathbb{F}^m$, the computed value $\mathtt{fl}(\langle x, y\rangle)$ satisfies*

$$\mathtt{fl}(\langle x, y\rangle) = \langle x, y\rangle + \langle |x|, |y|\rangle\, \theta_{\log m+2},$$

where $|x| = (|x_1|, \ldots, |x_n|)$.
(iii) *Assume $\tilde{x}, \tilde{y} \in \mathbb{F}^m$ and $x, y \in \mathbb{R}^m$ are such that, for all i,*

$$\tilde{x}_i = x_i + t_i\theta_\epsilon \quad \text{and} \quad \tilde{y}_i = y_i + t'_i\theta_{\epsilon'}$$

for some $t, t' \in [0,\infty)^m$ and $\epsilon, \epsilon' \geq 1$. Then the computed value $\mathtt{fl}(\langle \tilde{x}, \tilde{y}\rangle)$ satisfies

$$\mathtt{fl}(\langle \tilde{x}, \tilde{y}\rangle) = \langle x, y\rangle + \max\{\langle |x|, |y|\rangle, \langle |t|, |y|\rangle, \langle |x|, |t'|\rangle, \langle |t|, |t'|\rangle\}\, \theta_{\log m+\epsilon+\epsilon'+2}.$$

Proposition A.3 *There is a numerical algorithm which, with input $x \in \mathbb{R}^m$, computes $\|x\|$. This algorithm satisfies the following:*

(i) *It performs $\mathcal{O}(m)$ arithmetic operations.*
(ii) *On input $x \in \mathbb{F}^m$, the computed value $\mathtt{fl}(\|x\|)$ satisfies*

$$\mathtt{fl}(\|x\|) = \|x\|(1 + \theta_{\log m+3}).$$

(iii) *Assume $\tilde{x} \in \mathbb{F}^m$ and $x \in \mathbb{R}^m$ are such that, for all i,*

$$\tilde{x}_i = x_i + t_i\theta_\epsilon$$

for some $t \in [0,\infty)^m$ and $\epsilon \geq 1$. Then the computed value $\mathtt{fl}(\|\tilde{x}\|)$ satisfies

$$\mathtt{fl}(\|\tilde{x}\|) = \|x\| + \max\{\|x\|, \|t\|\}\theta_{\log m+\epsilon+3}.$$

Proposition A.4 *There is a numerical algorithm which, with input $x \in \mathbb{R}^n$ and $\alpha \in \mathbb{N}$, computes x^α. This algorithm satisfies the following:*

(i) *It performs $\mathcal{O}(\log|\alpha|)$ arithmetic operations.*

(ii) *On input $x \in \mathbb{F}^n$, the computed value $\mathtt{fl}(x^\alpha)$ satisfies*

$$\mathtt{fl}(x^\alpha) = \begin{cases} x^\alpha\big(1+\theta_{|\alpha|-1}\big), & \text{if } |\alpha| > 1, \\ x^\alpha, & \text{otherwise.} \end{cases}$$

(iii) *Assume that $\tilde{x} \in \mathbb{F}^n$ and $x \in \mathbb{R}^n$ are such that, for all i,*

$$\tilde{x}_i = x_i(1+\theta_\epsilon)$$

for some $t \in [0,\infty)^m$ and $\epsilon \geq 1$. Then the computed value $\mathtt{fl}(\tilde{x}^\alpha)$ satisfies

$$\mathtt{fl}(\tilde{x}^\alpha) = \begin{cases} x^\alpha\big(1+\theta_{|\alpha|(1+\epsilon)-1}\big), & \text{if } \alpha \neq 0, \\ 1, & \text{otherwise.} \end{cases}$$

Proof of Proposition A.2 The algorithm will first perform all the products $x_i\, y_i$ and them perform their sum by recursively dividing the sum into

$$\sum_{i\in I} x_i\, y_i + \sum_{i\in I^{\complement}} x_i\, y_i,$$

where I and its complement, $I^{\complement}$ have size almost equal, differing in at most one.

(i) We initially perform m products and then $m-1$ additions. Note that the latter is independent of how we achieve the final sum, we sum as we do to minimize the error.

(ii) We will prove using induction the stronger claim that for the above algorithm

$$\mathtt{fl}\left(\langle x, y\rangle\right) = \langle x, y\rangle + \langle |x|, |y|\rangle\, \theta_{\lceil \log m\rceil+1},$$

where $\lceil x\rceil$ is the minimum integer bigger or equal than x. Note that the claim is true for $m=1$ and $m=2$. By the recursive nature of the algorithm, we have that

$$\begin{aligned}
\mathtt{fl}\left(\sum_{i=1}^{m} x_i\, y_i\right) &= \mathtt{fl}\left(\sum_{i\in I} x_i\, y_i\right) \widetilde{+}\ \mathtt{fl}\left(\sum_{i\in I^{\complement}} x_i\, y_i\right) \\
&= \left\{\sum_{i\in I} x_i\, y_i + \left(\sum_{i\in I} |x_i||y_i|\right)\theta_{\lceil \log |I|\rceil+1}\right. \\
&\quad \left. + \sum_{i\in I^{\complement}} x_i\, y_i + \left(\sum_{i\in I^{\complement}} |x_i||y_i|\right)\theta_{\lceil \log (n-|I|)\rceil+1}\right\}(1+\theta_1) \quad \text{(induction)} \\
&= \left\{\sum_{i=1}^{n} x_i\, y_i + \left(\sum_{i=1}^{n} |x_i||y_i|\right)\theta_{\log \max\{|I|, n-|I|\}+1}\right\}(1+\theta_1) \\
&= \big(\langle x, y\rangle + \langle |x|, |y|\rangle\, \theta_{\lceil \log \max\{|I|, n-|I|\}\rceil+1}\big)(1+\theta_1). \qquad \text{(E6)}
\end{aligned}$$

Now, when $|I|$ and $n - |I|$ differ in at most one, we have that

$$\left\lceil \log \max\{|I|, n-|I|\} \right\rceil + 1 \leq \lceil \log n \rceil.$$

Thus

$$\begin{aligned}
&= \big(\langle x, y\rangle + \langle |x|, |y|\rangle\, \theta_{\lceil \log n\rceil}\big)(1+\theta_1)\\
&= \langle x, y\rangle + \langle x, y\rangle\, \theta_1 + \langle |x|, |y|\rangle\big(\theta_{\lceil \log n\rceil} + \theta_{\lceil \log n\rceil}\theta_1\big)\\
&= \langle x, y\rangle + \langle |x|, |y|\rangle\theta_1 + \langle |x|, |y|\rangle\big(\theta_{\lceil \log n\rceil} + \theta_{\lceil \log n\rceil}\theta_1\big)\langle x, y\rangle \leq \langle |x|, |y|\rangle\\
&= \langle x, y\rangle + \langle |x|, |y|\rangle\big(\theta_{\lceil \log n\rceil} + \theta_1 + \theta_{\lceil \log n\rceil}\theta_1\big) \qquad \text{(E1)}\\
&= \langle x, y\rangle + \langle |x|, |y|\rangle\, \theta_{\lceil \log n\rceil+1}. \qquad \text{(E2)}
\end{aligned}$$

(iii) Note that

$$\begin{aligned}
\langle \tilde{x}, \tilde{y}\rangle &= \langle x, y\rangle + \langle (t_i\theta_\epsilon), y\rangle + \langle x, (t_i'\theta_{\epsilon'})\rangle + \langle (t_i\theta_\epsilon), (t_i'\theta_{\epsilon'})\rangle\\
&= \langle x, y\rangle + \langle |t|, |y|\rangle\theta_\epsilon + \langle |x|, |t'|\rangle\theta_{\epsilon'} + \langle |t|, |t'|\rangle\, \theta_\epsilon\theta_{\epsilon'} \qquad \text{(E6)}\\
&= \langle x, y\rangle + \max\{\langle |t|, |y|\rangle, \langle |x|, |t'|\rangle, \langle |t|, |t'|\rangle\}\, (\theta_\epsilon + \theta_{\epsilon'} + \theta_\epsilon\theta_{\epsilon'}) \qquad \text{(E7)}\\
&= \langle x, y\rangle + \max\{\langle |t|, |y|\rangle, \langle |x|, |t'|\rangle, \langle |t|, |t'|\rangle\}\, \theta_{\epsilon+\epsilon'}. \qquad \text{(E2)}
\end{aligned}$$

An analogous statement holds for $\langle |\tilde{x}|, |\tilde{y}|\rangle$. Now, combining this and (A.2), we get that

$$\begin{aligned}
\mathtt{fl}(\langle \tilde{x}, \tilde{y}\rangle) &= \langle \tilde{x}, \tilde{y}\rangle + \langle |\tilde{x}|, |\tilde{y}|\rangle\, \theta_{\log m+2}\\
&= \langle x, y\rangle + \max\{\langle |t|, |y|\rangle, \langle |x|, |t'|\rangle, \langle |t|, |t'|\rangle\}\, \theta_{\epsilon+\epsilon'}\\
&\quad + \big(\langle |x|, |y|\rangle + \max\{\langle |t|, |y|\rangle, \langle |x|, |t'|\rangle, \langle |t|, |t'|\rangle\}\, \theta_{\epsilon+\epsilon'}\big)\, \theta_{\log m+2}\\
&= \langle x, y\rangle + \max\{\langle |x|, |y|\rangle, \langle |t|, |y|\rangle, \langle |x|, |t'|\rangle, \langle |t|, |t'|\rangle\} \qquad \text{(E7)}\\
&\qquad\qquad \cdot\, (\theta_{\epsilon+\epsilon'} + \theta_{\log m+2} + \theta_{\epsilon+\epsilon'}\theta_{\log m+2})\\
&= \langle x, y\rangle + \max\{\langle |x|, |y|\rangle, \langle |t|, |y|\rangle, \langle |x|, |t'|\rangle, \langle |t|, |t'|\rangle\}\, \theta_{\log m+\epsilon+\epsilon'+2}. \qquad \square
\end{aligned}$$

Proof of Proposition A.3 The proof is analogous to that of Proposition A.2. □

Proof of Proposition A.4 The proof is analogous to that of Proposition A.2, but we have to take into account that errors accumulate additively since in each multiplication the errors of the computed quantities are added by (E2). □

A.3 The Final Proofs

The following lemma is useful.

Lemma A.5 *There is a numerical algorithm which, with input $f \in \mathcal{P}_{n,d}$, computes the Weyl norm $\|f\|$ of f. This algorithm performs $\mathcal{O}(N)$ arithmetic operations, and, on input $f \in \mathcal{P}_{n,d} \cap \mathbb{F}[X_1, \ldots, X_n]$, the computed value $\mathtt{fl}(\|f\|)$ satisfies*

$$\mathtt{fl}(\|f\|) = \|f\|(1 + \theta_{\log N+8}).$$

Moreover, for general $f \in \mathcal{P}_{n,d}$,

$$\mathtt{fl}(\|r(f)\|) = \|f\|(1+\theta_{\log N+9}).$$

Proof To compute the Weyl norm, we first compute the vector

$$\left(\binom{d}{\alpha}^{-1/2} f_\alpha\right)$$

and then its norm. To compute the vector, we take the floating point approximation of $\binom{d}{\alpha}$, we compute its square root and we divide f_α by the computed square root. Hence

$$\begin{aligned}\mathtt{fl}\left(\binom{d}{\alpha}^{-1/2} f_\alpha\right) &= \binom{d}{\alpha}^{-1/2} f_\alpha \frac{1+\theta_1}{\sqrt{1+\theta_1}(1+\theta_1)} \\ &= \binom{d}{\alpha}^{-1/2} f_\alpha(1+\theta_5) \qquad \text{(Proposition A.1)}\end{aligned}$$

Now, the lemma follows from Proposition A.3. □

We can now give the proofs of Propositions 6.1 and 6.2.

Proof of Proposition 6.1 We first compute $f(x)$ as $\langle (f_\alpha), (x^\alpha)\rangle$, where the x^α are computed one by one, and then divide the result by the computed $\|f\|\|(1,x)\|^{d-1}$ to obtain $\widehat{f}(x)$. By Propositions A.2 and A.4 and (E7), we have that

$$\mathtt{fl}\,(f(x)) = f(x) + \|f\|\,\|(1,x)\|^d \theta_{\log N+d+1},$$

since $\langle (|f_\alpha|), (|x^\alpha|)\rangle = g(|x|)$, where $g = \sum_\alpha |f_\alpha| X^\alpha$, is bounded by $\|f\|\,\|(1,x)\|^d$, by Lemma 3.5. Also, by Proposition A.3, Lemma A.5, and (E2), we have that

$$\mathtt{fl}\,(\|f\|\,\|(1,x)\|^{d-1}) = \|f\|\,\|(1,x)\|^{d-1}\big(1+\theta_{\log N+d\log(n+1)+4d+2}\big).$$

Now, $N \le (n+1)^d$. Thus we have that

$$\begin{aligned}\mathtt{fl}\,(f(x)) &= f(x) + \|f\|\,\|(1,x)\|^d \theta_{3d\log(n+1)} \qquad \text{and} \\ \mathtt{fl}\,(\|f\|\,\|(1,x)\|^{d-1}) &= \|f\|\,\|(1,x)\|^{d-1}\big(1+\theta_{8d\log(n+1)}\big).\end{aligned}$$

Although, doing this we are not obtaining tight bounds, we have to recall that the number of digits is proportional to the logarithm of what is inside θ. To finish, we only

have to do the division. Thus

$$\begin{aligned}
\mathtt{fl}(\widehat{f}(x)) &= \frac{\mathtt{fl}(f(x))}{\mathtt{fl}(\|f\|\,\|(1,x)\|^{d-1})}\,(1+\theta_1)\\
&= \frac{f(x)+\|f\|\|(1,x)\|^d\theta_{3d\log(n+1)}}{\|f\|\,\|(1,x)\|^{d-1}(1+\theta_{7d\log(n+1)})}\,(1+\theta_1)\\
&= \frac{\widehat{f}(x)+\|(1,x)\|\,\theta_{3d\log(n+1)}}{1+\theta_{8d\log(n+1)}}\,(1+\theta_1)\\
&= (\widehat{f}(x)+\|(1,x)\|\,\theta_{3d\log(n+1)}\big(1+\theta_{16d\log(n+1)+1}\big)\\
&= \widehat{f}(x)+\widehat{f}(x)\,\theta_{10d\log(n+1)+1}\\
&\quad +\|(1,x)\|\big(\theta_{3d\log(n+1)}+\theta_{3d\log(n+1)}\theta_{14d\log(n+1)+1}\big)\\
&= \widehat{f}(x)+\|(1,x)\|\big(\theta_{16d\log(n+1)+1}\\
&\quad +\theta_{3d\log(n+1)}+\theta_{3d\log(n+1)}\theta_{16d\log(n+1)+1}\big)\\
&= \widehat{f}(x)+\|(1,x)\|\,\theta_{19d\log(n+1)+1}\\
&= \widehat{f}(x)+\|(1,x)\|\,\theta_{20d\log(n+1)}
\end{aligned}$$

where the first equality follows from the way we compute $\widehat{f}(x)$, the second one from the above identities, the fourth one from (E3) and (E2), the sixth one from Lemma 3.5 and (E7), the eighth one from (E2), and the last one from (E1). The result for $r(f)$ and $r(x)$ follows similarly. □

Proof of Proposition 6.2 We compute each $\partial_j f(x)$ as we computed $f(x)$. After that, we compute $\|\partial f(x)\|, d\|f\|\|(1,x)\|^{d-2}$ and their quotient. By Propositions A.2 and A.4, and (E7), we have that

$$\mathtt{fl}(\partial_j f(x)) = \partial_j f(x) + \partial_j g(|x|)\,\theta_{\log N+d+1},$$

where $g = \sum_\alpha |f_\alpha|\, X^\alpha$. Now, by Proposition A.3, we have that

$$\mathtt{fl}(\|\partial f(x)\|) = \|\partial f(x)\| + \max\{\|\partial f(x)\|, \partial g(|x|)\|\}\,\theta_{\log N+\log n+d+4}.$$

However, by Lemma 3.5, $\|\partial f(x)\|$ and $\|\partial g(|x|)\|$ are bounded by $d\|f\|\,\|(1,x)\|^{d-1}$. Thus, by (E7),

$$\mathtt{fl}\,(\|\partial f(x)\|) = \|\partial f(x)\| + d\|f\|\,\|(1,x)\|^{d-1}\theta_{\log N+\log n+d+4}.$$

Again, by Proposition A.3, Lemma A.5, and (E2), we have that

$$\mathtt{fl}\,(d\|f\|\,\|(1,x)\|^{d-2}) = d\|f\|\,\|(1,x)\|^{d-2}\big(1+\theta_{\log N+d\log(n+1)+4d+2}\big).$$

Now, as $N \leq (n+1)^d$, we have

$$\texttt{fl}\left(\|\partial f(x)\|\right) = \|\partial f(x)\| + d\|f\|\,\|(1,x)\|^{d-1}\theta_{7d\log(n+1)} \qquad \text{and}$$
$$\texttt{fl}\left(d\|f\|\,\|(1,x)\|^{d-2}\right) = d\|f\|\,\|(1,x)\|^{d-2}\big(1+\theta_{8d\log(n+1)}\big).$$

Now, arguing as in Proposition 6.1, the desired statement follows.

References

1. Blanchard, P., Higham, N.J., Mary, T.: A class of fast and accurate summation algorithms. SIAM J. Sci. Comput. **42**(3), A1541–A1557 (2020)
2. Bürgisser, P., Cucker, F.: Condition. Grundlehren der Mathematischen Wissenschaften, vol. 349. Springer, Heidelberg (2013)
3. Bürgisser, P., Cucker, F., Lairez, P.: Computing the homology of basic semialgebraic sets in weak exponential time. J. ACM **66**(1), # 5 (2019)
4. Bürgisser, P., Cucker, F., Lotz, M.: Smoothed analysis of complex conic condition numbers. J. Math. Pures Appl. **86**(4), 293–309 (2006)
5. Bürgisser, P., Cucker, F., Lotz, M.: The probability that a slightly perturbed numerical analysis problem is difficult. Math. Comput. **77**(263), 1559–1583 (2008)
6. Bürgisser, P., Cucker, F., Tonelli-Cueto, J.: Computing the homology of semialgebraic sets. I: Lax formulas. Found. Comput. Math. **20**(1), 71–118 (2020)
7. Bürgisser, P., Cucker, F., Tonelli-Cueto, J.: Computing the homology of semialgebraic sets. II: General formulas. Found. Comput. Math. **21**(5), 1279–1316 (2021)
8. Burr, M.A.: Continuous amortization and extensions: with applications to bisection-based root isolation. J. Symbol. Comput. **77**, 78–126 (2016)
9. Burr, M., Choi, S.W., Galehouse, B., Yap, Ch.K.: Complete subdivision algorithms. II: Isotopic meshing of singular algebraic curves. J. Symbol. Comput. **47**(2), 131–152 (2012)
10. Burr, M.A., Gao, S., Tsigaridas, E.: The complexity of an adaptive subdivision method for approximating real curves. In: 42nd ACM International Symposium on Symbolic and Algebraic Computation (Kaiserslautern 2017), pp. 61–68. ACM, New York (2017)
11. Burr, M., Gao, S., Tsigaridas, E.: The complexity of subdivision for diameter-distance tests. J. Symbol. Comput. **101**, 1–27 (2020)
12. Burr, M., Krahmer, F., Yap, Ch.: Continuous amortization: a non-probabilistic adaptive analysis technique. In: Electronic Colloquium on Computational Complexity, # 136 (2009). https://eccc.weizmann.ac.il/report/2009/136/
13. Cucker, F.: Approximate zeros and condition numbers. J. Complex. **15**(2), 214–226 (1999)
14. Cucker, F., Ergür, A.A., Tonelli-Cueto, J.: Plantinga–Vegter algorithm takes average polynomial time. In: 44th ACM International Symposium on Symbolic and Algebraic Computation (Beijing 2019), pp. 114–121. ACM, New York (2019)
15. Cucker, F., Krick, T., Malajovich, G., Wschebor, M.: A numerical algorithm for zero counting. I. Complexity and accuracy. J. Complex. **24**(5–6), 582–605 (2008)
16. Cucker, F., Krick, T., Malajovich, G., Wschebor, M.: A numerical algorithm for zero counting. II. Distance to ill-posedness and smoothed analysis. J. Fixed Point Theory Appl. **6**(2), 285–294 (2009)
17. Cucker, F., Krick, T., Malajovich, G., Wschebor, M.: A numerical algorithm for zero counting. III: Randomization and condition. Adv. Appl. Math. **48**(1), 215–248 (2012)
18. Cucker, F., Krick, T., Shub, M.: Computing the homology of real projective sets. Found. Comput. Math. **18**(4), 929–970 (2018)
19. Cucker, F., Peña, J.: A primal-dual algorithm for solving polyhedral conic systems with a finite-precision machine. SIAM J. Optim. **12**(2), 522–554 (2002)
20. Demmel, J.W.: The probability that a numerical analysis problem is difficult. Math. Comput. **50**(182), 449–480 (1988)
21. Ergür, A.A., Paouris, G., Rojas, J.M.: Probabilistic condition number estimates for real polynomial systems I: a broader family of distributions. Found. Comput. Math. **19**(1), 131–157 (2019)

22. Ergür, A.A., Paouris, G., Rojas, J.M.: Smoothed analysis for the condition number of structured real polynomial systems. Math. Comput. **90**(331), 2161–2184 (2021)
23. Funke, S.: Of what use is floating-point arithmetic in computational geometry? In: Efficient Algorithms. Lecture Notes in Comput. Sci., vol. 5760, pp. 341–354. Springer, Berlin (2009)
24. Galehouse, B.T.: Topologically Accurate Meshing Using Domain Subdivision Techniques. PhD thesis, New York University (2009)
25. Goldstine, H.H., von Neumann, J.: Numerical inverting of matrices of high order. II. Proc. Am. Math. Soc. **2**, 188–202 (1951)
26. Higham, N.J.: Accuracy and Stability of Numerical Algorithms. SIAM, Philadelphia (1996)
27. Jeannerod, C.-P.: Exploiting structure in floating-point arithmetic. In: Mathematical Aspects of Computer and Information Sciences (Berlin 2015). Lecture Notes in Comput. Sci., vol. 9582, pp. 25–34. Springer, Cham (2016)
28. Livshyts, G., Paouris, G., Pivovarov, P.: On sharp bounds for marginal densities of product measures. Israel J. Math. **216**(2), 877–889 (2016)
29. Lotz, M.: On the volume of tubular neighborhoods of real algebraic varieties. Proc. Am. Math. Soc. **143**(5), 1875–1889 (2015)
30. Plantinga, S., Vegter, G.: Isotopic approximation of implicit curves and surfaces. In: 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing (Nice 2004), pp. 245–254. ACM, New York (2004)
31. Ratschek, H., Rokne, J.: Computer Methods for the Range of Functions. Ellis Horwood Series: Mathematics and its Applications. Halsted Press, New York (1984)
32. Rudelson, M., Vershynin, R.: The Littlewood–Offord problem and invertibility of random matrices. Adv. Math. **218**(2), 600–633 (2008)
33. Rudelson, M., Vershynin, R.: Small ball probabilities for linear images of high-dimensional distributions. Int. Math. Res. Not. IMRN **2015**(19), 9594–9617 (2015)
34. Smale, S.: Complexity theory and numerical analysis. In: Acta Numer., vol. 6, pp. 523–551. Cambridge University Press, Cambridge (1997)
35. Spielman, D.A., Teng, S.-H.: Smoothed analysis of algorithms. In: International Congress of Mathematicians (Beijing 2002), vol. 1, pp. 597–606. Higher Ed. Press, Beijing (2002)
36. Spielman, D.A., Teng, S.-H.: Smoothed analysis: an attempt to explain the behavior of algorithms in practice. Commun. ACM **52**(10), 77–84 (2009)
37. Tonelli-Cueto, J.: Condition and Homology in Semialgebraic Geometry. PhD thesis, Technische Universität Berlin (2019). https://doi.org/10.14279/depositonce-9453
38. Vershynin, R.: High-Dimensional Probability. Cambridge Series in Statistical and Probabilistic Mathematics, vol. 47. Cambridge University Press, Cambridge (2018)
39. Xu, J., Yap, Ch.: Effective subdivision algorithm for isolating zeros of real systems of equations, with complexity analysis. In: 44th ACM International Symposium on Symbolic and Algebraic Computation (Beijing 2019), pp. 355–362. ACM, New York (2019)
40. Yap, Ch.: Towards soft exact computation (invited talk). In: Computer Algebra in Scientific Computing (Moscow 2019). Lecture Notes in Comput. Sci., vol. 11661, pp. 12–36. Springer, Cham (2019)