A General Approach to the Analysis of Controlled Perturbation Algorithms*

Kurt Mehlhorn† Ralf Osbild Michael Sagraloff†

December 2nd, 2010

Contents

Abstract

Controlled Perturbation (CP, for short) is an approach to obtaining efficient and robust implementations of a large class of geometric algorithms using the computational speed of multiple precision floating point arithmetic (compared to exact arithmetic), while bypassing the precision problems by perturbation. It also allows algorithms to be written without consideration of degenerate cases. CP replaces the input objects by a set of randomly perturbed (moved, scaled, stretched, etc.) objects and protects the evaluation of geometric predicates by guards. The execution is aborted if a guard indicates that the evaluation of a predicate with floating point arithmetic

*This work was partially supported German-Israeli Foundation (GIF) Grant 969/07.

†Max-Planck-Institut für Informatik, Campus E1.4, 66123 Saarbrücken, Germany.

may return an incorrect result. If the execution is aborted, the algorithm is rerun on a new perturbation and maybe a higher precision of the floating point arithmetic. If the algorithm runs to completion, it returns the correct output for the perturbed input.

The analysis of CP algorithms relates various parameters: the perturbation amount δ, the arithmetic precision L, the range of input values $[-M, M]$, and the number of input objects n. We present a general methodology for analyzing CP algorithms. It is powerful enough to analyze all geometric predicates that are formulated as signs of polynomials.

1 Introduction

Most algorithms of computational geometry are designed under two simplifying assumptions: the availability of a Real-RAM and non-degeneracy of the input. A Real-RAM computes with real numbers in the sense of mathematics. The notion of degeneracy depends on the problem; examples are collinear or co-circular points or three lines with a common intersection point. We call an algorithm designed under the two simplifying assumptions an *idealistic algorithm*. An idealistic algorithm A_I on input z halts with the correct result if z is non-degenerate and A_I is executed with exact real arithmetic. However, implementations have to deal with the precision problem (caused by the Real-RAM assumption) and the degeneracy problem (caused by the non-degeneracy assumption).

The *exact computation paradigm* [KLN91, JRZ91, FvW93, Yap97, MN94, MN99] addresses the precision problem. It proposes to implement a Real-RAM tuned to geometric computations. The degeneracy problem is addressed by reformulating the algorithms so that they can handle all inputs. This may require non-trivial changes. The approach is followed in systems such as LEDA [MN99] and CGAL [CGA].

Symbolic perturbation [EM90, ECS97, Sei98, Yap90] addresses the degeneracy problem. Instead of solving the problem on the given input z, one solves it on an input that is perturbed by infinitesimal amounts. The approach removes degeneracies; it requires exact arithmetic.

Halperin et al.[HS98, HR, HL04] proposed *controlled perturbation* (CP for short) as a solution for both problems. The idea is to perturb the input numerically and to control the effect of the perturbation (hence the name controlled perturbation). The hope is that the perturbed input is non-degenerate and can be handled with approximate arithmetic (see Section 2 for details). *CP algorithms compute approximate solutions in the following sense: they compute the exact output for a nearby input.* Halperin et. al. applied the idea to three problems (computing polyhedral arrangements, spherical arrangements, and arrangements of circles) and showed that CP variants of the respective idealistic algorithms can be made to work. Funke et al. [FKMS05, Kle04] extended their work and showed how to use CP for Delaunay triangulations and convex hulls in arbitrary dimensions. In the conference version of this paper [MOS06], we argued that CP is applicable to a wide class of geometric algorithms and outlined a general approach to analyzing CP algorithms. The approach requires nontrivial geometric reasoning for each geometric predicate. Caroli [Car07] applied the approach to geometric predicates required for the computation of circle arrangements and Voronoi diagrams of line segments. The analysis is quite lengthy, involved, and does not cover all predicates. In this paper, we considerably simplify the approach and turn the analysis of CP algorithms from an art to a craft. *In particular, we give an analysis of all predicates that can be realized by polynomial expressions*. Moreover, we resolve an issue that was left open by all previous papers: the analysis assumes that the perturbation is carried out in the space of real numbers, but implementations only work with floating point perturbations.

Controlled perturbation is not a panacea. It only applies if it is possible and permissible to perturb the input. If the exact result for the unperturbed input is needed, perturbation is not permissible. If the input consists of a numerical part and a combinatorial part and a consistency condition between the two, perturbing the numerical part and keeping it consistent with the combinatorial part might be impossible. There are positive examples where consistency can be maintained, e.g., a polygonal chain with vertex coordinates, and negative examples where consistency cannot be maintained, e.g., a polyhedron given by its incidence lattice and equations for the facets. Controlled perturbation is always possible if the input consists only of numerical values, e.g., point coordinates. It should also be noted that no perturbation scheme can remove a symbolic degeneracy, e.g., the three perpendicular bisectors of the edges of a triangle always meet in a common point. Perturbation may however help to discover redundant tests in a program.

This paper is structured as follows. In Section 2 we review the concept of controlled perturbation. In Section 3 we present a general methodology for analyzing CP algorithms (Subsection 3.3), show that it can handle all predicates defined as signs of polynomials (Subsection 3.4), discuss the issue that the analysis is carried out in real space but an implementation perturbs in the space of floating point numbers (Subsection 3.5), extend the analysis from predicates to algorithms (Subsection 3.6), and analyze the complexity of CP (Subsection 3.7). In Section 4, we compare the general methodology to an approach that uses more intensive geometric reasoning. We will see that the general methodology leads to similar results, but with slightly weaker constant factors. Section 5 suggests future work. Finally, in the Appendix (Section 6), we review the basics of floating point arithmetic and provide an error analysis for arithmetic expressions.

2 Controlled Perturbation

We review the concept of controlled perturbation; this section follows and also extends Funke et al. [FKMS05]. Geometric algorithms branch on geometric predicates, e.g., on the position of a point relative to a line or to a circle. Analytically, a geometric predicate is expressed as the sign of a real valued function f. Consider, for example, the *orientation predicate* for $d+1$ points $p_1,\dots,p_d$ and $q = p_{d+1}$ in $\mathbb{R}^d$: If $p_1,\dots,p_d$ define a hyperplane in $\mathbb{R}^d$, the predicate decides which of the associated halfspaces contains the query point q; the answer is given by the sign of a $(d+1)\times(d+1)$ determinant:

$$\mathit{orient}(p_1,\dots,p_d,q) := \operatorname{sign}\begin{vmatrix} p_{1,1} & \dots & p_{1,d} & 1 \\ \vdots & \dots & \vdots & \vdots \\ p_{d,1} & \dots & u_{d,d} & 1 \\ q_1 & \dots & q_d & 1 \end{vmatrix}. \tag{1}$$

The predicate evaluates to zero if and only if the $d+1$ points lie in a common hyperplane. This is considered a degeneracy. A perturbation of the points is likely to remove this degeneracy. Moreover, it may allow to determine the correct sign of the determinant by means of approximate arithmetic.

The value of the determinant above is the signed volume of the simplex spanned by the $d+1$ points. The sign is positive if the simplex has positive orientation and is negative otherwise. If the absolute value of the determinant is sufficiently large, approximate arithmetic determines the correct sign. Thus, in order to show that approximate arithmetic is able to determine the correct sign for a perturbed set of points, one only has to show that the volume of the simplex spanned by the perturbed points is sufficiently large. We show in our main theorem that a similar kind of reasoning is possible for all predicates that are formulated as signs of polynomials.

The evaluation of an arithmetic formula f in floating point arithmetic incurs round-off errors which may change the sign. If this stays undetected, the program may enter an illegal state and produce incorrect output or crash or loop; see [KMP$^+$08] for instructive examples. In order to protect against undesirable consequences of round-off errors, we postulate the availability of a *guard* G_f with the following *guard property: The guard G_f is a Boolean expression. If it evaluates to true when evaluated with floating point arithmetic, the floating point evaluation (fp-evaluation) of f yields the correct sign.* In this case, we also say that the evaluation of f is *fp-safe*. If G_f evaluates to false, we say that the guard failed.

Using guards we can transform an idealistic algorithm A_I into a *guarded algorithm A_g* in the following way: we protect every sign test by first testing the corresponding guard. If the guard fails, we abort A_g and return the message "unsuccessful computation". On the other hand, if the guarded algorithm A_g runs to completion, we return the message "successful computation". In a successful computation all branch decisions are made correctly and hence the combinatorial part of the output is correct. However, numerical values are only approximate. Also, the asymptotic running time of A_g on any input z will be at most the asymptotic running time of A_I on z; this assumes that the cost of evaluating a guard is of the same order as the cost of evaluating the corresponding expression.

We will use the 2d-orientation predicate for points $a = (a_x, a_y)$, $b = (b_x, b_y)$, $c = (c_x, c_y)$ in the plane as our running example; it is given by[1]

$$orient(a,b,c) = \operatorname{sign}(f) \quad \text{where} \quad f = (b_x - a_x) \cdot (c_y - a_y) - (b_y - a_y) \cdot (c_x - a_x).$$

By Theorem 10 in Section 6,

$$G_f \equiv \left(|\widetilde{f}| > 28 \odot M^2 \odot 2^{-L} \right)$$

has the guard property. Here $\widetilde{f}$ is the value of the expression f when evaluated with floating point arithmetic, $M \geq 1$ is a power of two[2] that bounds the absolute value of all arguments, L is the precision of the floating point system (see below), and $\oplus$, $\ominus$, and $\odot$ are the floating point implementations of $+$, $-$, and $\cdot$. Theorem 10 also exhibits a guard that fails less often, but is harder to compute. Alternatively, we can evaluate the defining expression with interval arithmetic and use the guard that zero is not contained in the result interval. For now we assume the existence of guards. In Section 6, we will show their existence and review the basics of floating point arithmetic. Floating point numbers are of the form

$$sign \cdot mantissa \cdot 2^{exponent}.$$

where the mantissa is an L-bit number; we refer to L as the precision of the floating point system. The error in a single floating point operation is proportional to 2^{-L}. Hardware floating point systems are available for $L = 26$ (IEEE single precision), $L = 52$ (IEEE double precision) and $L = 112$ (IEEE quadruple precision). Software floating point systems allow the user to choose L.

A δ-perturbation, $\delta \in \mathbb{R}^+$, of a real number r is a random number in the interval $[r - \delta, r + \delta]$. A δ-perturbation of a point $z \in \mathbb{R}^d$ is a point which results from δ-perturbations of z's coordinates. Alternatively, it could be a random point in the δ-sphere centered at z. We call δ the *perturbation amount* and the set of all possible δ-perturbations of a point z, denoted by $U_\delta(z)$, the *perturbation region*. In this paper we consider the entire input to an algorithm, which in fact is a set of geometric objects, as a real-valued higher-dimensional point $\overline{z}$ and assume that we may perturb all of its coordinates by up to δ. We come back to this assumption in Section 5.

The *controlled perturbation version A_{cp}* of an idealistic algorithm A_I works as follows: Let $\overline{z}$ be the input and let δ be a positive real. We first choose a δ-perturbation $z \in U_\delta(\overline{z})$ of $\overline{z}$ and then run

[1] An alternative formulation is $orient(a,b,c) = b_x c_y - b_x a_y - a_x c_y - b_y c_x + b_y a_x + a_y c_x$. For this formulation $G_f \equiv \left(|\widetilde{f}| > 30 \odot M^2 \odot 2^{-L} \right)$ has the guard property; see Section 6. In order to distinguish the formulations, we call the formulation of the footnote the "expanded" formulation and the formulation in the text, the "non-expanded" formulation.

[2] We restrict M to powers of two because this makes the computation of the bounds more efficient. We need M to be at least one, because the proofs of Theorems 11, 12, and 13 require that M^d is a nondecreasing function of d.

the guarded algorithm A_g on z. If A_g terminates successfully, we terminate A_{cp} as well and return the output of A_g together with the perturbed input z. If A_g aborts, however, we rerun A_g on a new perturbation z of $\overline{z}$. We may also adjust the CP parameters, i.e., increase the precision of the floating point arithmetic and/or the perturbation amount δ.

A controlled perturbation algorithm can be used without any analysis. Suppose we want to use it with a certain perturbation amount δ. We execute it with a certain precision L. If it does not succeed, we double L and repeat. It is easy to see that this strategy terminates for a wide class of geometric algorithms (Theorem 1). We give a quantitative relation (Theorem 6) between δ, L and characteristic quantities of the problem instance, e.g., the size of the instance and the largest coordinate, and analyse the complexity of the approach (Theorem 9).

3 A General Scheme for Analyzing Predicate Functions

Guards must be safe and should be effective, i.e., if a guard lets the computation continue, the approximate sign computation must be correct (safety), and guards should not stop the computation too often unnecessarily (effectiveness). It is usually difficult to analyze the conditions under which the floating point evaluation of a guard G_f returns true. For the purpose of the analysis and only for the purpose of the analysis, we therefore postulate the existence of a *bound predicate* $\mathcal{B}_f$ with the following property: *If $\mathcal{B}_f$ holds, G_f evaluates to true when evaluated with floating point arithmetic.* For a function f of k arguments, $\mathcal{B}_f \subseteq \mathbb{R}^k$ and G_f is a Boolean expression with k arguments. If $z = (z_1, \ldots, z_k) \in \mathcal{B}_f$, the floating point evaluation of G_f on z returns true.

In Section 6 we show how to define valid guards and bound predicates. It follows from Theorem 13 in Section 6 that if f is a polynomial, there is always a bound predicate $\mathcal{B}_f$ of the form

$$|f(z)| > K_f M^{\deg(f)} 2^{-L},$$

where $\deg(f)$ is the degree of the polynomial, K_f is a constant depending on the coefficients and the number of monomial terms, and M is the smallest power of two with

$$M \geq \max(1, \max\{(|x| : x \text{ is an argument of } f\}).$$

We define

$$EB_f(L) := K_f M^{\deg(f)} 2^{-L}.$$

as the right hand side of the bound predicate and frequently write EB_f instead of $EB_f(L)$. For the 2d-orientation predicate in the plane (in its non-expanded form), Theorem 13 in Section 6 yields

$$\mathcal{B}_f \equiv \left(|f(z)| > 56 M^2 2^{-L}\right)$$

as the the bound predicate corresponding to the guard given in the preceding section.[3]

We describe a methodology for analyzing predicate functions. We consider a geometric predicate defined as the sign of a function f of k variables defined on

$$A = [-M, M]^k.$$

[3] For the expanded version, Section 6 yields $\mathcal{B}_f \equiv \left(|f(z)| > 60 M^2 2^{-L}\right)$ as the bound predicate corresponding to the guard given in footnote 1.

Controlled perturbation replaces an input $\overline{z}$ by a random point in the cubic neighborhood $U_\delta(\overline{z})$. For simplicity[4], we assume that the input domain is such that $U_\delta(\overline{z}) \subseteq A$. We want to guarantee that for any $\overline{z}$, the bound predicate $\mathcal{B}_f$ holds for many arguments in the perturbation region $U_\delta(\overline{z})$. We use

$$S_\delta(\overline{z}) := U_\delta(\overline{z}) \cap \mathcal{B}_f = \left\{ z \in U_\delta(\overline{z}) : |f(z)| > EB_f(L) \right\}$$

for the part of the perturbation region where the bound predicate guarantees safety. Observe that this part depends on the choice of L as this choice influences EB_f. Also observe that $EB_f(L)$ can be made arbitrarily small. For the sake of simplicity, we suppress this dependency on L and also omit $\overline{z}$ most of the time. Then for a random choice of $z \in U_\delta$, the probability p_f of a successful evaluation of f at z satisfies[5]

$$p_f \geq \frac{\mu(S_\delta)}{\mu(U_\delta)} = \frac{\int_{x \in S_\delta} 1\, dx}{\int_{x \in U_\delta} 1\, dx}, \tag{2}$$

where μ denotes the Lebesgue measure. Our first theorem states that for any "reasonable" predicate function f, this ratio gets arbitrary close to 1 for sufficiently large L.

Theorem 1 *If f is upper continuous almost everywhere and has a zero set Z_f of measure zero, and if* $\lim_{L\to\infty} EB_f(L) = 0$ *then*

$$\lim_{L\to\infty} p_f = 1.$$

Proof: For any positive ε, let $A_\varepsilon := \{z \in U_\delta(\overline{z}) : |f(z)| \leq \varepsilon\}$ be the set of arguments whose function value is bounded by ε. Then $A_{\varepsilon_1} \subseteq A_{\varepsilon_2}$ whenever $\varepsilon_1 < \varepsilon_2$. If $z \in \cap_{\varepsilon>0} A_\varepsilon$ then $f(z) \leq \varepsilon$ for all positive ε and hence $f(z) = 0$. Thus $Z_f = \cap_{\varepsilon>0} A_\varepsilon$ and hence (A_ε is measurable since f is upper continuous almost everywhere) $\lim_{\varepsilon\to 0} \mu(A_\varepsilon) = \mu(Z_f) = 0$ by upper continuity of the Lebesgue measure. Hence $\mu(A_{EB_f(L)})$ tends towards zero as L goes to infinity. ∎

We remark, that the question, whether $\mu(Z_f) = 0$, may be non-trivial. For example, for three points u, v, and w in the plane, let

$$f(u,v,w) := sol(\ell_{uv}, \ell_{uw} \cap \ell_{vw}),$$

where *sol* (side of line) is the 2*d*-orientation function and ℓ_{uv}, ℓ_{uw}, and ℓ_{vw} are the three perpendicular bisectors. Since the three bisectors of a triangle intersect in a single point, $f \equiv 0$. Of course, no perturbation of the points will remove this degeneracy. Degeneracies that cannot be removed by perturbation are called *symbolic degeneracies*. Controlled perturbation may help to discover symbolic degeneracies. If a degeneracy does not go away by repeated perturbation, one may take this as an indication that the degeneracy is symbolic.

Theorem 1 establishes that CP works. However, it does not give a quantitative relation between the perturbation value δ, the precision L, and the success probability p_f of predicate evaluation. For quantitative estimates, we have to estimate the ratio of the two integrals in Formula (2). In Section 3.3 we introduce a general methodology for deriving such an estimate. We need some more notation.

[4] Alternatively, one may say that controlled perturbation replaces $\overline{z}$ by a random point in the neighborhood $U_\delta(\overline{z}) \cap A$. The volume of the neighborhood restricted to A is at least 2^{-k} times the volume of the full neighborhood. We leave it to the reader to check that all theorems in this paper stay true after a suitable change of constants. In some situations, one may want to consider only inputs with nonnegative coordinates. Then one would define $A = [0, M]$.

[5] We assume that for any $\delta \geq 0$ and any $\varepsilon \geq 0$, the set $\{z \in U_\delta(\overline{z}) : |f(z)| \leq \varepsilon\}$ is Lebesgue measurable.

3.1 Some Notation

Throughout the paper we deal with functions $f : \mathbb{R}^k \to \mathbb{R}$ in k variables $z_1, z_2, \dots, z_k$. The ‘coordinate’ projection $\pi_j : \mathbb{R}^k \to \mathbb{R}$ with $1 \le j \le k$ maps a k-dimensional point $z = (z_1, z_2, \dots, z_k)$ to its j-th coordinate $\pi_j(z) := z_j$. For any set $A \subseteq \mathbb{R}^k$, let $\pi_j(A) := \{\pi_j(a) : a \in A\}$ be the projection of A on its j-th coordinate.

The ‘prefix’ projection $\pi^{(j)} : \mathbb{R}^k \to \mathbb{R}^j$ with $1 \le j \le k$ maps a k-dimensional point $z = (z_1, z_2, \dots, z_k)$ to the tuple $(z_1, \dots, z_j)$ of its first j coordinates, i.e.,

$$\pi^{(j)}(z_1, z_2, \dots, z_k) = (z_1, \dots, z_j).$$

For any set $A \subseteq \mathbb{R}^k$, let $\pi^{(j)}(A) := \{\pi^{(j)}(a) : a \in A\}$.

In order to simplify notation, we use the following convention. For $z = (z_1, \dots, z_k) \in \mathbb{R}^k$, we use $y = (y_1, \dots, y_{k-1})$ to denote the projection of z on the first $k-1$ coordinates and x for the projection on the last coordinate. Then $y \in \mathbb{R}^{k-1}$, $x \in \mathbb{R}$, and $z = (y_1, y_2, \dots, y_{k-1}, x)$.

Frequently, we fix the first $k-1$ arguments of f and consider the function of the last argument obtained in this way. Suppose $f : \mathbb{R}^k \to \mathbb{R}$ and $y = (y_1, \dots, y_{k-1}) \in \mathbb{R}^{k-1}$. Then we define $f_y : \mathbb{R} \to \mathbb{R}$ by

$$f_y(x) = f(y_1, \dots, y_{k-1}, x).$$

A point y is a *degenerizer* if f_y is identically zero (i.e., $f_y(x) = 0$ for all $x \in \mathbb{R}$). We use $D_f \subseteq \mathbb{R}^{k-1}$ to denote the set of all degenerizers.

We use $Z_f \subseteq \mathbb{R}^k$ to denote the *zero set* of f (i.e., $Z_f = \{z \in \mathbb{R}^k : f(z) = 0\}$). A *critical set* for f is any superset of Z_f. We will use critical sets in the following context: Z_f and C_f are sets of measure zero and C_f has a “nicer structure” than Z_f and is therefore easier to handle.

For any point set $P \subset \mathbb{R}^k$ and $\delta > 0$ we define its closed δ-neighborhood by

$$U_\delta(P) := \{z \in \mathbb{R}^k : \exists p \in P \text{ with } |p_i - x_i| \le \delta \text{ for all } i\}.$$

3.2 A General Scheme: Intuition and Example

Let $A = [-M, +M]^k \subseteq \mathbb{R}^k$ and $f : A \to \mathbb{R}$. How can we estimate the volume of the region S_δ? Or equivalently, the volume of its complement. Let

$$R_\delta = \{z \in U_\delta(\overline{z}) : |f(z)| \le EB_f\}.$$

We call R_δ the *region of uncertainty*; see Figure 1(a) for an example. It is the region where the bound predicate does not guarantee fp-safety. We need to show that R_δ is small. Intuition tells us that f is small only close to its zero set. Since the zero set may be a complicated set, we consider it one variable at a time. This is akin to cylindrical algebraic decomposition [ACM84].

We postulate[6] the existence of a set $D \subset \mathbb{R}^{k-1}$ of measure zero that contains all degenerizers of f. Consider a fixed $y \in \mathbb{R}^{k-1}$ and assume $y \notin D$. Then f_y is not the constant zero function. We postulate the existence of a finite critical set C_y for f_y of cardinality at most N; N is a constant not depending on y. We also postulate the existence of a neighborhood U_y of C_y of volume $2N\gamma$, where γ is a suitable parameter, and a function g such that

$$g(y) \le \inf_{x \in \pi_k(A) \setminus U_y} |f(y, x)|.$$

We postulate[7] $g(y) > 0$ for $y \notin D$. For $y \in D$, we define $g(y) = 0$. What have we achieved?

[6] The occurrences of the word “postulate” in this paragraph define the applicability of our methodology.

[7] For $k = 1$, we postulate that g is a positive constant.

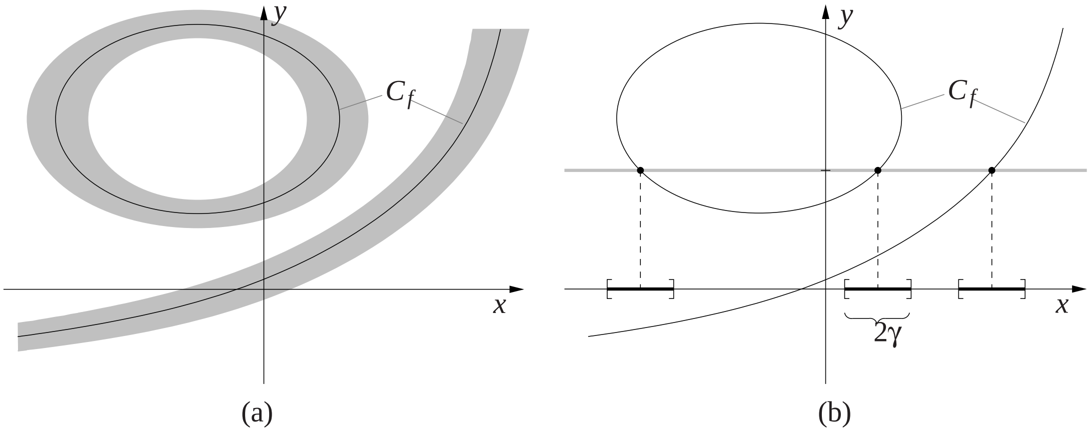

Figure 1: In (a), the critical set of f is indicated as a curve and the region of uncertainty is shown shaded. The region of uncertainty is located around the critical set. The horizontal axis corresponds to the last coordinate and the vertical axis corresponds to all other coordinates. In (b), a fixed value of y is indicated as a grey horizontal line. The x-axis shows the projection of the intersection of the line with the region of uncertainty.

1. The degenerizers are contained in a set D of measure zero.
2. For $y \notin D$, $g(y) > EB_f$ guarantees $f(y,x) > EB_f$ for x outside U_y. In other words, for a fraction[8]

   $$\frac{2\delta - 2N\gamma}{2\delta} = 1 - \frac{N\gamma}{\delta}$$

   of the $x \in U_\delta(\overline{z}_k)$, the evaluation of $f(y,x)$ is fp-safe.

Let $f_{k-1} = g$, $N_k = N$, and $\gamma_k = \gamma$. We now apply the same reasoning to f_{k-1}. This introduces N_{k-1} and γ_{k-1} and reduces f_{k-1} to f_{k-2}, a function of $k-2$ arguments. Continuing in this way, we arrive at a positive constant f_0. We choose L such that $f_0 > EB_f(L)$. For an random $z \in U_\delta(\overline{z})$ the bound predicate will then hold with probability

$$\prod_{1 \le i \le k} \left(1 - \frac{N_i \gamma_i}{\delta}\right).$$

Why is this the case? Consider a random $z \in U_\delta(\overline{z})$. If there is no j such that $\pi^{(j)}(z)$ belongs to the set D for f_{j+1}, we are always in case 2) and the probability bound holds. If some prefix $\pi^{(j)}(z)$ is in the set D for f_{j+1}, the prefix belongs to a set of measure zero and hence the probability bound stays valid. We next work through an example and then describe the general methodology in the next subsection.

We consider the 2d-orientation predicate and rename the point coordinates a_x, a_y, b_x, b_y, c_x, c_y as z_1 to z_6. The renaming helps to forget geometry. We obtain

$$\begin{aligned} f(z_1,\dots,z_6) &:= z_1z_4 + z_3z_6 + z_5z_2 - z_1z_6 - z_3z_2 - z_5z_4 \\ &= (z_3 - z_1)z_6 + z_1z_4 + z_5z_2 - z_3z_2 - z_5z_4. \end{aligned}$$

[8] Recall that we assume that $\overline{z}$ is such that $U_\delta(\overline{z}) \subseteq A$.

For fixed $y = (z_1, \dots, z_5)$, f_y is a polynomial of degree at most one in z_6. A point $y \in \mathbb{R}^5$ is a degenerizer if $z_1 = z_3$ and $z_1z_4 + z_5z_2 - z_3z_2 - z_5z_4 = 0$. We take $D = \{(z_1, \dots, z_5) : z_1 = z_3\}$. For $y \notin D$, f_y is a linear function in z_6 that is zero for

$$z_6 = -\frac{z_1z_4 + z_5z_4 - z_3z_2 - z_5z_4}{z_1 - z_3}.$$

Let C_y be the singleton set consisting of this point and let U_y be the γ_6-neighborhood of this point. We define

$$f_5(z_1, \dots, z_5) := |z_3 - z_1|\gamma_6 \leq \inf_{x \notin U_y} |f_6(z_1, \dots, z_5, x)|.$$

The next two reductions are trivial; for both steps we take $D = \emptyset$ and $C = \emptyset$ and $N = 0$ and set

$$f_3(z_1, z_2, z_3) = f_4(z_1, \dots, z_4) = f_5(z_1, \dots, z_5) = |z_3 - z_1|\gamma_6.$$

The function $z_3 \mapsto f_3(z_1, z_2, z_3)$ is different from the constant zero for all choices of (z_1, z_2), i.e., f_3 has no degenerizers. We choose $D = \emptyset$ for the reduction step from three arguments to two arguments. For fixed (z_1, z_2), $f_3(z_1, z_2, z_3)$ is zero for $z_3 = z_1$. Let $C_{(z_1,z_2)} = \{z_1\}$ and $U_{(z_1,z_2)}$ be the γ_3-neighborhood of this point. We can then define

$$f_2(z_1, z_2) = \gamma_3\gamma_6.$$

The next two reduction steps are again trivial. We take $D = \emptyset$, $C = \emptyset$ and $N = 0$ and set $f_0 = f_1(z_1) = \gamma_3\gamma_6$. We have now shown that

$$|f(z_1, \dots, z_6)| \geq \gamma_3\gamma_6$$

provided that

$$\left| z_6 - \frac{z_1z_4 + z_5z_4 - z_3z_2 - z_5z_4}{z_1 - z_3} \right| \geq \gamma_6 \quad \text{and} \quad |z_3 - z_1| \geq \gamma_3.$$

For any fixed $\bar{z} \in \mathbb{R}^6$, the probability that a random $z \in U_\delta(\bar{z})$ satisfies these conditions is at least

$$\left(1 - \frac{2\gamma_6}{2\delta}\right) \cdot \left(1 - \frac{2\gamma_3}{2\delta}\right).$$

Next observe that $(1 - \gamma_6/\delta)(1 - \gamma_3/\delta) \geq 1 - (\gamma_3 + \gamma_6)/\delta$. The right-hand side of the bound predicate is $EB_f = 56M^2 2^{-L}$. So in order to guarantee that the bound predicate holds with probability at least ρ, we only need to choose γ_6, γ_3 and L such that

$$EB_f(L) \leq \gamma_3\gamma_6 \quad \text{and} \quad \left(1 - \frac{\gamma_3 + \gamma_6}{\delta}\right) \geq \rho.$$

Setting $\gamma_3 = \gamma_6 = (1 - \rho)\delta/2$ yields the constraint

$$56M^2 2^{-L} \leq \left(\frac{(1-\rho)\delta}{2}\right)^2 \quad \text{or equivalently} \quad L \geq 7.807\ldots + 2\log\frac{M}{\delta} + 2\log\frac{1}{1-\rho}.$$

3.3 A General Scheme

We formally define the reduction process introduced informally in the preceeding section and prove a quantitative version of Theorem 1.

Definition 1 *Let $A \subseteq \mathbb{R}^k$, $B = \pi^{(k-1)}(A)$, and $f : A \to \mathbb{R}$. We call (f, D, C, N), where $N \in \mathbb{N}$ and $C = (C_y)_{y \in B}$ is a family of subsets of $\mathbb{R}$, an* admissible representation *of f if*

1. *$D \subseteq B$ is a set of measure zero that contains all degenerizers of f. We call the points $y \in B \setminus D$* regular.

2. *For each $y \in B$, C_y is a subset of $\mathbb{R}$ that contains the zero-set of f_y. If y is regular, C_y contains at most N elements.*

Every multivariate polynomial $f \in \mathbb{R}[y_1,\ldots,y_{k-1},x]$ of total degree $\deg(f) = n$ admits an admissible representation with $N = n$. We view f as a polynomial in the last variable x with coefficients $a_i(y) \in \mathbb{R}[y_1,\ldots,y_{k-1}]$. Let $n' \leq n$ be maximal such that $a_{n'}(y) \not\equiv 0$. The degenerizers of f are those y where all coefficients vanish simultaneously. We set D to the set of all y, where the leading coefficient $a_{n'}(y)$ vanishes. Then D contains all degenerizers and for all $y \notin D$ the polynomial f_y has exactly n' complex roots. We can now define C_y either as the set of all real roots of f_y or as the set of projections of all roots onto the real axis. In both cases, (f,D,C,n') constitutes an admissible representation of f. In Section 3.4 we will continue the investigation of polynomial predicate functions.

Definition 2 *Let $A \subseteq \mathbb{R}^k$, $B = \pi^{(k-1)}(A)$, $f : A \to \mathbb{R}$, N an integer, and $\gamma \in \mathbb{R}^+$. A function $g : B \to \mathbb{R}^+$ is an (N,γ)*-reduction of f *if there exists an admissible representation (f,D,C,N) of f such that, for each regular y, there exists a neighborhood $U = U_y$ of C_y of measure at most $2N\gamma$ such that*

$$x \notin U \implies g(y) \leq |f(y,x)|.$$

In the case $k = 1$, this amounts to the existence of a constant $c > 0$ with $c \leq |f(x)|$ for all $x \notin U$ and U a set of volume $2N\gamma$.

Many functions are reducible. We only have to set D to the set of degenerizers of f and C_y to the zero set of f_y for any y. If $N = \max\{|C_y| : y \notin D\}$ is finite and $g(y) := \inf_{x \notin U_\gamma(C_y)} |f(y,x)| > 0$ then (f,D,C,N) constitutes an admissible representation of f and g is an $(N,\gamma)-$reduction of f. We remark that our definition is more flexible. It allows us to define D as a proper superset of D_f and it allows us to define U_y and g in a more liberal way. We will put this added flexibility to good use in Section 3.4.

We are particularly interested in the case that the function g in Definition 2 is again reducible, say to h, and h is again reducible, ..., all the way down to a constant. This leads to the notion of *fully reducible*.

Definition 3 *Let $A \subseteq \mathbb{R}^k$, $B_j = \pi^{(j)}(A)$ and $f : A \to \mathbb{R}$. Then f is* fully reducible *to $f_0 \in \mathbb{R}^+$ if there are $N_k,\ldots,N_1 \in \mathbb{N}$, positive reals γ_k, ..., γ_1, and functions $f_j : B_j \to \mathbb{R}$ such that $f_k = f$ and f_{j-1} is an (N_j,γ_j)-reduction of f_j for all j, $k \geq j \geq 1$.*

We are now ready for a quantitative version of Theorem 1.

Theorem 2 *Let $\overline{z} \in A = [-M,M]^k \subseteq \mathbb{R}^k$, $f : A \to \mathbb{R}$, and $U_\delta(\overline{z}) \subseteq A$. Assume that f is fully reducible to $f_0 \in \mathbb{R}^+$ and let N_k to N_1 and γ_k to γ_1 be as in Definition 3. If $EB_f(L) \leq f_0$ (this can always be achieved by making L sufficiently large) then*

$$\mu(S_\delta(\overline{z})) \quad \geq \quad 2^k \prod_{1 \leq j \leq k} (\delta - \gamma_j N_j)\,.$$

The probability p_f of a successful predicate evaluation for a random point $z \in U_\delta(\overline{z})$ satisfies

$$p_f \geq \prod_{1 \leq j \leq k} \left(1 - \frac{\gamma_j N_j}{\delta}\right).$$

Proof: Let $B_j = [-M,+M]^j$. By Definition 3, there are functions $f_j : B_j \to \mathbb{R}$ with $f_k = f$ such that f_{j-1} is an (N_j,γ_j)-reduction of f_j for all j, $k \geq j \geq 1$.

We consider the first step of the reduction sequence. Let D and C be as in Definition 2. We will bound $\mu(S_\delta(\overline{z}))$ from below. Consider any $(y,x) \in U_\delta(\overline{z})$, $y \in \mathbb{R}^{k-1}$, $x \in \mathbb{R}$, such that y is regular. Then the cardinality of the critical set C_y is at most N_k and there is a neighborhood U_y of C_y of measure at most $2N_k\gamma_k$ such that $|f_{k-1}(y)| \leq |f_k(y,x)|$ for all $x \in \pi_k(U_\delta(\overline{z})) \setminus U_y$. Let $S_\delta = S_\delta(\overline{z})$, $U_\delta = U_\delta(\overline{z})$, $Y_\delta = \pi^{(k-1)}(U_\delta)$, and $X_\delta = \pi_k(U_\delta)$. Then $\mu(X_\delta \setminus U_y) \geq 2\delta - 2N_k\gamma_k$ and hence

$$\begin{aligned}
\mu(S_\delta) &= \int_{z\in S_\delta} 1\,dz = \int_{z\in U_\delta\,:\,|f(z)|>EB_f} 1\,dz \\
&= \int_{y\in Y_\delta} \left(\int_{x\in X_\delta\,:\,|f(y,x)|>EB_f} 1\,dx \right) dy \\
&\geq \int_{Y_\delta\setminus D} \left(\int_{x\in X_\delta\setminus U_y\,:\,|f(y,x)|>EB_f} 1\,dx \right) dy \\
&\geq \int_{Y_\delta\setminus D\,:\,|f_{k-1}(y)|>EB_f} \left(\int_{x\in X_\delta\setminus U_y\,:\,|f(y,x)|>EB_f} 1\,dy \right) dy \\
&\overset{(*)}{=} \int_{Y_\delta\setminus D\,:\,|f_{k-1}(y)|>EB_f} \left(\int_{x\in X_\delta\setminus U_y} 1\,dx \right) dy \\
&= \int_{Y_\delta\setminus D\,:\,|f_{k-1}(y)|>EB_f} \mu(X_\delta \setminus U_y)\,dy \\
&\geq \int_{Y_\delta\setminus D\,:\,|f_{k-1}(y)|>EB_f} (2\delta - 2\gamma_k N_k)\,dy \\
&\overset{(**)}{=} \int_{Y_\delta\,:\,|f_{k-1}(y)|>EB_f} (2\delta - 2\gamma_k N_k)\,dy \\
&= 2(\delta - \gamma_k N_k) \int_{Y_\delta\,:\,|f_{k-1}(y)|>EB_f} 1\,dy,
\end{aligned}$$

where equality $(*)$ holds because $|f(y,x)| \geq |f_{k-1}(y)|$ for all $y \notin D$ and $x \in X_\delta \setminus U_y$, and equality $(**)$ holds since D has measure 0. The integral $\int_{Y_\delta\,:\,|f_{k-1}(y)|>EB_f} 1\,dy$ in the last formula has the same form as the integral $\int_{U_\delta\,:\,|f(z)|>EB_f} 1\,dz$ in the first line, but for one smaller dimension. We can therefore continue in this way and establish the first claim.

For the second claim, we use Formula (2) and obtain

$$p_f = \frac{\mu(S_\delta)}{\mu(U_\delta)} \geq \frac{2^k \prod_{1\leq j\leq k} (\delta - \gamma_j N_j)}{(2\delta)^k} = \prod_{1\leq j\leq k} \left(1 - \frac{\gamma_j N_j}{\delta}\right) .$$

∎

We next specialize to an important subfamily of reducible functions for which the dependency of the f_j's on the γ_j's is explicitly expressed in terms of a factor $\gamma_j^{\alpha_j}$. This subfamily includes all multivariate polynomials, as we will show in the next subsection, and is particularly well suited to our approach.

Definition 4 (separable function) *Let $A \subseteq \mathbb{R}^k$ and $f : A \to \mathbb{R}$.*

(i) f is separable *if there exists a positive integer N, positive reals $\widetilde{\gamma}$ and α, and a function $h : B \to \mathbb{R}$, where $B = \pi^{(k-1)}(A)$, such that $\gamma^\alpha \cdot h$ is an (N,γ)-reduction of f for all $\gamma \leq \widetilde{\gamma}$, i.e., there exists an*

admissible representation (f,D,C,N) *of* f *such that, for each regular* y*, there exists a neighborhood* $U = U_y$ *of* C_y *of measure at most* $2N\gamma$ *such that*

$$x \notin U \implies \gamma^{\alpha} h(y) \leq |f(y,x)|.$$

In the case $k = 1$*, this amounts to the existence of a constant* $c > 0$ *with* $c\gamma^{\alpha} \leq |f(x)|$ *for all* $x \notin U$ *and* U *a set of volume* $2N\gamma$*.*

(ii) f *is* fully separable *if there exists a sequence of functions* $f_j : B_j \to \mathbb{R}$*, where* $B_j = \pi^{(j)}(A)$*,* $f_k = f$*,* $f_0 \in \mathbb{R}^+$*, and positive integers* N_j *and positive reals* $\widetilde{\gamma}_j$ *and* α_j *such that for all* j*,* $1 \leq j \leq k$*, and all* $\gamma_j \leq \widetilde{\gamma}_j$*, the function* $\gamma_j^{\alpha_j} \cdot f_{j-1}$ *is an* (N_j, γ_j)*-reduction of* f_j*.*

Assume now that f is fully separable with $\widetilde{\gamma}_j$'s, α_j's and N_j's as in Definition 4. Also assume that $\gamma_j \leq \widetilde{\gamma}_j$ for all j and $z = (z_1, \ldots, z_k) \in A$ is such that for all j, $y_{j-1} := \pi^{(j-1)}(z) \notin D_j$ and $z_j \notin U_{y_{j-1}}$, where $U_{y_{j-1}}$ has measure $2\gamma_j N_j$. Here D_j and $U_{y_{j-1}}$ are as in Definition 2. Then

$$|f(z)| = |f(z_1, \ldots, z_k)| \geq f_0 \cdot \gamma_1^{\alpha_1} \cdot \ldots \cdot \gamma_k^{\alpha_k}.$$

Thus, we obtain the following specialized version of Theorem 2 for fully separable functions.

Corollary 3 *Let* $A = [-M, +M]^k$*, let* $\overline{z} \in A$ *be such that* $U_\delta(\overline{z}) \subseteq A$*, and let* $f : A \to \mathbb{R}$ *be fully separable as in Definition 4. Assume further that* L *and* $\gamma_j \leq \widetilde{\gamma}_j$ *are such that*

$$EB_f(L) < f_0 \cdot \gamma_1^{\alpha_1} \cdot \ldots \cdot \gamma_k^{\alpha_k}.$$

Then the probability p_f *of a successful predicate evaluation for a random point* $z \in U_\delta(\overline{z})$ *satisfies*

$$p_f \geq \prod_{1 \leq j \leq k} \left(1 - \frac{\gamma_j N_j}{\delta}\right).$$

In the following section we will specialize the above result to multivariate polynomials. We will see that multivariate polynomials are fully separable and that the α_i's in Definition 4 can be chosen such that their sum is bounded by the total degree of the polynomial.

3.4 Polynomial Predicate Functions

We show that any nonzero polynomial is fully separable. We give explicit definitions for all quantities in Definition 4. We then show how to optimize the CP parameters. The reasoning is purely analytical and requires no geometric insight.

Let $f \in \mathbb{R}[z] := \mathbb{R}[z_1, \ldots, z_k]$ be a nonzero multivariate polynomial in k variables and total degree N. The infinity-norm $\|f\|_\infty$ of f is defined as the maximum of the absolute values of all its coefficients. The degree of f, considered as polynomial in z_i, $1 \leq i \leq k$, is denoted by $N_i = \deg_{z_i} f$. For the monomial basis of $\mathbb{R}[z_1, \ldots, z_k]$ we consider *lexicographic ordering with reversed significance*, denoted by $\succ$. Given two monomials $z^\alpha := z_1^{\alpha_1} \cdot \ldots \cdot z_k^{\alpha_k}$ and $z^\beta := z_1^{\beta_1} \cdot \ldots \cdot z_k^{\beta_k}$, we define

$$z^\alpha \succ z^\beta \Leftrightarrow \alpha_{k_0} > \beta_{k_0} \text{ with } k_0 := \max\{k' : \alpha_{k'} \neq \beta_{k'}\}.$$

With respect to this ordering, $\mathrm{lm}(f)$ denotes the *leading monomial term* of f and $\mathrm{lcf}(f)$ its corresponding coefficient. Given a vector $\alpha := (\alpha_1, \ldots, \alpha_k)$ of exponents, $f_{[\alpha]}$ denotes the reduction of f to the sum of all terms of f that contain x^α as a factor and $f^*_{[\alpha]} := f_{[\alpha]}/x^\alpha$. We remark that $f = \sum_\alpha f_\alpha z^\alpha$ and $\mathrm{lcf}(f) = f^*_{[\alpha^*]}$ for $\mathrm{lm}(f) = z^{\alpha^*}$.

Example. The monomial terms of the polynomial $f(z_1,z_2,z_3) := z_1^2 z_2 z_3^4 + 2z_1z_2z_3 + z_1z_2^5 - 4z_3^6 + z_1^7$ are ordered in the following way:

$$z_3^6 \succ z_1^2 z_2 z_3^4 \succ z_1z_2z_3 \succ z_1z_2^5 \succ z_1^7$$

and $\mathrm{lm}(f) = z_3^6$, and $\mathrm{lcf}(f) = -4$. For $\alpha := (1,1,1)$ we obtain $f_{[\alpha]} = z_1^2z_2z_3^4 + 2z_1z_2z_3$ and $f^*_{[\alpha]} = z_1z_3^3 + 2$.

Let us consider f as a univariate polynomial in $x = z_k$ with coefficients $a_i \in \mathbb{R}[y]$, where $y = (z_1,\ldots,z_{k-1})$, i.e.,

$$f(z) = a_{N_k}(y)x^{N_k} + \ldots + a_0(y) \in \mathbb{R}[y][x].$$

From our considerations in Section 3.3 we already know that there exists an admissible representation (f,D,C,N_k) of f with

$$D = \{y \in \mathbb{R}^{k-1} : a_{N_k}(y) = 0\}$$

the set of all y such that the leading coefficient $a_{N_k}(y)$ vanishes and

$$C_y = \Re\{z \in \mathbb{C} : f(y,z) = 0\} = \{a \in \mathbb{R} : \exists b \in \mathbb{R} \text{ with } f(y, a + \mathbf{i}\cdot b) = 0\}$$

the projection of all complex roots of f_y onto the real axis. D is an algebraic hypersurface in $\mathbb{R}^{k-1}$ and thus has measure 0. For each $y \notin D$, f_y is a univariate polynomial of degree N_k and hence C_y consists of at most N_k points. We next show that f is separable.

Lemma 1 *Let $f(z) = f(y,x) := a_{N_k}(y)x^{N_k} + \ldots + a_0(y) \in \mathbb{R}[y][x]$ be a multivariate polynomial and (f,D,C,N_k) an admissible representation of f as defined above. Then, for arbitrary $\gamma \geq 0$,*

$$g(y) := |a_{N_k}(y)| \cdot \left(\frac{N_k\gamma}{2e}\right)^{N_k}$$

is an (N_k,γ)-reduction of f. For $N_k = \gamma = 0$ we define $\gamma^{N_k} := 1$.

Proof: Let $\gamma \geq 0$ be fixed. According to the definition of an $(N_k,\gamma)-$reduction (see Definition 2), we have to exhibit for each y with $a_{N_k}(y) \neq 0$, a neighborhood U_y of C_y of volume at most $2N_k\gamma$ such that $g(y) \leq |f(y,x)|$ for all $x \notin U_y$.

We use the following result from [SY09]: Given a multiset $R := \{p_1,\ldots,p_n\}$ of not necessarily distinct points $p_i \in \mathbb{R}$, there exists a neighborhood $U(R)$ of R of volume $2n\gamma$ such that[9] for any $p \notin U(R)$ there is a reindexing of the points in P such that $|p - p_i| \geq \gamma \cdot \lfloor (i+1)/2 \rfloor$ for all i; the reindexing is by distance from p.

Now, for fixed $y \notin D$, let $r_1,\ldots,r_{N_k} \in \mathbb{C}$ denote the complex roots of $f_y(x)$ and $P := \{p_1,\ldots,p_{N_k}\}$ be the corresponding multiset of their projections onto the real axis. Then, by the preceding paragraph, there exists a neighborhood $U_y \subseteq \mathbb{R}$ of C_y of volume $2N_k\gamma$ such that for any $x \notin U_y$ we have

$$|x - r_i| \geq |x - \Re(r_i)| = |x - p_i| \geq \gamma \cdot \lfloor (i+1)/2 \rfloor .$$

[9] For completeness, we sketch the construction. We construct a set U_r of volume $n\gamma$ such that for any $x \notin U_r$ and any i, the cardinality of $\{j : p_j \in [x, x+i\gamma]\}$ is less than i. A symmetric construction gives a set U_ℓ such that for any $x \notin U_\ell$ and any i, the cardinality of $\{j : p_j \in [x - i\gamma, x]\}$ is less than i. Then $U_\ell \cup U_r$ is the desired set. Consider the x for which there is an i such that the cardinality of $\{j : p_j \in [x, x+i\gamma]\}$ is i or more. Let x_0 be the infimum of these x and let i_0 be such that $|\{j : p_j \in [x, x+i_0\gamma]\}| \geq i_0$. Add $(x_0, x_0 + i_0\gamma)$ to U_r. Delete the p_j in $[x_0, x_0 + i_0\gamma]$ and repeat the construction.

Hence

$$|f(y,x)| = |a_{N_k}(y) \cdot \prod_{1\le i\le N_k} (z-r_i)| \ge |a_{N_k}(y)| \cdot \prod_{1\le i\le N_k} \gamma \lfloor (i+1)/2 \rfloor ! \ge |g(y)|.$$

The last inequality requires justification. Let $n = N_k$. Then

$$\prod_{1\le i\le n} \lfloor (i+1)/2 \rfloor ! = \lfloor n/2 \rfloor ! \lceil n/2 \rceil !.$$

We show that the latter quantity is at least $(n/(2e))^n$. For even n this follows immediately from $\ell! \ge (\ell/e)^\ell$ for all integer ℓ. For odd n we have to work harder. The claim holds for $n=1$ and so we may assume $n \ge 3$. We use $\ell! \ge \sqrt{2\pi\ell}\,(\ell/e)^\ell$ (see [Knu73, Section 1.2.11.2, Equation (19)]) and estimate as follows:

$$\begin{aligned}
\frac{\lfloor n/2 \rfloor ! \lceil n/2 \rceil !}{(n/(2e))^n} &= \frac{((n-1)/2)!((n+1)/2)!(2e)^n}{n^n} \\
&\ge \frac{\sqrt{\pi(n-1)}((n-1)/(2e))^{(n-1)/2}\sqrt{\pi(n+1)}((n+1)/(2e))^{(n+1)/2}(2e)^n}{n^n} \\
&= \frac{\pi(n^2-1)^{n/2}(n+1)}{n^n} = \pi(n+1)\left(1-\frac{1}{n^2}\right)^{n/2} \ge \frac{\pi}{e}(n+1) \ge 1.
\end{aligned}$$

∎

The function g in the theorem above is a multivariate polynomial in one less variable. So we can apply the same reasoning to it and obtain a function g of one less variable. Continuing in this way, we show that f is fully separable.

Theorem 4 *Any nonzero multivariate polynomial is fully separable. More precisely, if $f \in \mathbb{R}[z_1,\dots,z_k]$ has leading monomial $\mathrm{lm}(f) = z^\alpha = z_1^{\alpha_1} \cdot \ldots \cdot z_k^{\alpha_k}$, we may take in Definition 4:*

$$\begin{aligned}
f_k &:= f, \\
f_i &:= \left| f^*_{[(0,\dots,0,\alpha_{i+1},\dots,\alpha_k)]} \cdot \prod_{j=i+1}^{k} \left(\frac{\alpha_j\gamma_j}{2e}\right)^{\alpha_j} \right|,
\end{aligned}$$

$\widetilde{\gamma}_i = \infty$ *and* $N_i = \alpha_i$.

Proof: According to the definition of $\mathrm{lm}(f)$ the polynomial $g_i := f^*_{[(0,\dots,0,\alpha_{i+1},\dots,\alpha_k)]}$ is an element of $\mathbb{R}[z_1,\dots,z_i]$. Considering g_i as a univariate polynomial in z_i with coefficients in $\mathbb{R}[z_1,\dots,z_{i-1}]$, that is, $g_i \in \mathbb{R}[z_1,\dots,z_{i-1}][z_i]$, it has degree α_i and its leading coefficient is given by $f^*_{[(0,\dots,0,\alpha_i,\dots,\alpha_k)]} \in \mathbb{R}[z_1,\dots,z_{i-1}]$. By Theorem 1,

$$f^*_{[(0,\dots,0,\alpha_i,\dots,\alpha_k)]}\left(\frac{\alpha_i\gamma_i}{2e}\right)^{\alpha_i}$$

is an $(\alpha_i,\gamma_i)-$reduction of g_i. Thus, our claim follows by induction over i. ∎

An application of Corollary 3 now gives the following bound on the probability p_f of a successful predicate evaluation.

Theorem 5 *Let f be a multivariate polynomial as in Theorem 4, $\overline{z} \in A = [-M,+M]^k$, $U_\delta(\overline{z}) \subseteq A$, and L be such that*

$$EB_f(L) < \operatorname{lcf}(f) \cdot \prod_{j=1}^{k} \left(\frac{\alpha_j \gamma_j}{2e}\right)^{\alpha_j}.$$

The probability p_f of a successful predicate evaluation for a random point $z \in U_\delta(\overline{z})$ satisfies

$$p_f \geq \prod_{1 \leq j \leq k} \left(1 - \frac{\gamma_j \alpha_j}{\delta}\right).$$

Example. We reconsider the orientation predicate from the beginning of the section. It is given by the polynomial

$$f(z_1,\ldots,z_6) := z_1z_4 + z_3z_6 + z_5z_2 - z_1z_6 - z_3z_2 - z_5z_4. \tag{3}$$

Its leading term is $\operatorname{lm}(f) = z_6z_3$ and its leading coefficient is $\operatorname{lcf}(f) = 1$. Now, for arbitrary $\gamma_1,\ldots,\gamma_6 \geq 0$, it follows that the probability p_f of a successful evaluation satisfies

$$p_f \geq \left(1 - \frac{\gamma_6}{\delta}\right) \cdot \left(1 - \frac{\gamma_3}{\delta}\right).$$

provided that $EB_f(L) < \gamma_6\gamma_3/(4e^2)$. Except for the factor $4e^2$, this bound is the same as the one obtained at the beginning of Section 3.3; the difference is that the bound now follows from a general result.

We next show how to minimize L subject to a constraint on p_f, for instance $p_f \geq \rho$. By Theorem 12, we can use $EB_f(L) = K_fM^N2^{-L}$ in the bound predicate, where $K_f = c_f(m_f + 2N)$, $c_f = \sum_\alpha \max(1,|f_\alpha|)$, and $m_f = |\{\alpha : f_\alpha \neq 0\}|$ is the number of monomial terms in $f = \sum_\alpha f_\alpha z^\alpha$. The leading monomial of f is z^{α^*}. Then, for arbitrary $\gamma_1,\ldots,\gamma_k \geq 0$, Theorem 4 tells us that

$$p_f \geq \prod_{1 \leq j \leq k} \left(1 - \frac{\gamma_j \alpha_j^*}{\delta}\right) \geq 1 - \sum_{1 \leq j \leq k} \frac{\gamma_j \alpha_j^*}{\delta}$$

provided L is such that

$$K_fM^N2^{-L} \leq |\operatorname{lcf}(f)| \cdot \prod_{j=1}^{k} \left(\frac{\alpha_j^* \gamma_j}{2e}\right)^{\alpha_j^*}. \tag{4}$$

For a fixed $\rho < 1$ we want to minimize L subject to the condition

$$h_1(\gamma_1,\ldots,\gamma_k) := 1 - \sum_{1 \leq j \leq k} \frac{\gamma_j \alpha_j^*}{\delta} - \rho \geq 0.$$

In an optimum solution, we have $h_1 = 0$; otherwise, we could increase a γ_j with $\alpha_j^* \neq 0$, which in turn would increase the right hand side of (4). We now use the method of Lagrange multipliers. Define

$$h_2(\gamma_1,\ldots,\gamma_k) := \log \prod_{j=1}^{k} \left(\frac{\alpha_j^* \gamma_j}{2e}\right)^{\alpha_j^*} = \sum_{j:\alpha_j^* \neq 0} \alpha_j^* \log \frac{\alpha_j^* \gamma_j}{2e}$$

We want to maximize h_2 subject to the constraint $h_1 = 0$. At a maximum, the gradients of h_1 and h_2 must be parallel and hence there must exist a Lagrange multiplier $\mu \in \mathbb{R}$ such that

$$\mu \cdot \frac{\alpha_j^*}{\delta} = \frac{\alpha_j^*}{\gamma_j} \quad \text{and hence} \quad \gamma_j = \frac{\delta}{\mu}$$

for all $j = 1,\dots,k$ with $\alpha_j^* \neq 0$. Replacing γ_j by $\frac{\delta}{\mu}$ in the condition $h_1(\gamma) = 0$, we obtain

$$\mu^{-1} = (1-\rho)\cdot\left(\sum_{1\le j\le k}\alpha_j^*\right)^{-1}.$$

Substituting the resulting value for the γ_j's into the right hand side of (4) and writing S for $\sum_{1\le j\le k}\alpha_j^* = \deg \operatorname{lm} f$, we obtain

$$|\operatorname{lcf}(f)|\cdot\prod_{j=1}^{k}\left(\frac{\alpha_j^*\gamma_j}{2e}\right)^{\alpha_j^*} = |\operatorname{lcf}(f)|\cdot\prod_{j=1}^{k}\left(\frac{\alpha_j^*\delta(1-\rho)}{2eS}\right)^{\alpha_j^*} \ge |\operatorname{lcf}(f)|\left(\frac{\delta(1-\rho}{2ek^*}\right)^{S},$$

where $k^* = |\{j : a_j^* \neq 0\}|$ is the number of variables in the leading monomial term. The last inequality uses the fact that $\prod_{1\le j\le k}(\alpha_j^*/S)^{\alpha_j^*}$ is minimized if $\alpha_j^* = S/k^*$ for all j with $a_j^* \neq 0$. The minimum is $(1/k^*)^S$. Thus (4) holds if L is such that

$$L \ge \log\left(K_f M^N\right) - \log|\lambda_{\alpha^*}| + \deg \operatorname{lm}(f)\cdot\log\frac{2ek^*}{\delta(1-\rho)}$$

or equivalently

$$L \ge \log\left(c_f(m_f+2N)\right) + N\log M - \log|\operatorname{lcf}(f)| + \deg \operatorname{lm}(f)\cdot\log\frac{2ek^*}{\delta(1-\rho)}. \tag{5}$$

We next simplify the right hand side at the expense of making it slightly larger. We use $k^* \le N$ and $\deg \operatorname{lm}(f) \le N$ and obtain the condition

$$L \ge \log\left(c_f(m_f+2N)\right) - \log|\operatorname{lcf}(f)| + N\left(3 + \log N + \log\frac{M}{\delta} + \log\frac{1}{1-\rho}\right). \tag{6}$$

Theorem 6 *Let $f = \sum_\alpha f_\alpha x^\alpha$ be a multivariate polynomial of total degree N, m_f monomial terms, $c_f = \sum_{\alpha:\, f_\alpha\neq 0}\max(1,|f_\alpha|)$, and k^* variables appearing in the lead monomial. If the variables are randomly perturbed by at most δ and after perturbation are bounded by M, the precision of the floating point system is L, and (6) or (5) holds, then the bound predicate holds with probability at least ρ.*

We next apply the general analysis to two examples. The first example is the 2d-orientation predicate and shows that the general analysis gives precision bounds comparable to those obtained by special purpose considerations. The second examples shows that the methodology can analyze fairly complex predicates; the underlying polynomial has 335 terms of degrees up to six; despite the complexity of the defining polynomial, the analysis is straightforward.

Example One: We consider the polynomial

$$f(z_1,\dots,z_6) := z_1z_4 + z_3z_6 + z_5z_2 - z_1z_6 - z_3z_2 - z_5z_4$$

underlying the 2d-orientation predicate and apply equation (5). The leading monomial term is $\operatorname{lm}(f) = z_3z_6$ with leading coefficient $\operatorname{lcf}(f) = 1$. Furthermore, $c_f = m_f = 6$, $N = 2$, $k = 6$, $\deg \operatorname{lm}(f) = 2$, and $k^* = 2$. Thus, if

$$\begin{aligned} L &\ge \log\left(c_f(m_f+2N)\right) - \log|\operatorname{lcf}(f)| + N\left(\log(4e) + \log\frac{M}{\delta} + \log\frac{1}{1-\rho}\right) \\ &= 12.79\ldots + 2\left(\log\frac{M}{\delta} + \log\frac{1}{1-\rho}\right), \end{aligned}$$

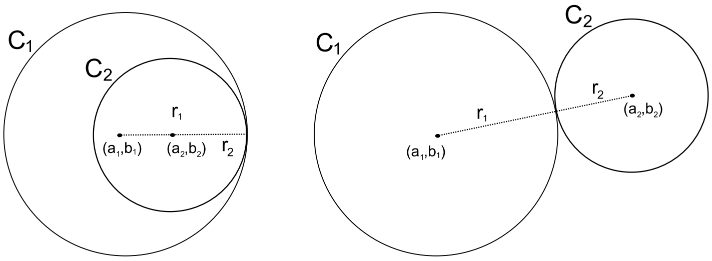

Figure 2: Given two circles C_i, $i = 1,2$, with midpoints (a_i, b_i) and radii $r_i = \sqrt{c_i}$, there are two degenerate situations of tangential intersection.

the probability of a successful predicate evaluation is at least ρ. Except for the constant additive factor this is the same bound as derived in the introductory discussion at the beginning of this section. The difference in the constant comes from two sources. First, the general theorem uses the bound predicate for the orientation predicate in expanded form. Second, the term $N\log(4e)$ comes from the estimate of the factorial in Lemma 1.

Example Two: The second example demonstrates the strength of the general approach. We study predicates that arise in the arrangement computation of circles in the plane. For the predicate to determine whether three circles have a common intersection point, the underlying polynomial is a multivariate polynomial in 9 variables with 335 monomials terms and total degree 6. Consider the following predicates:

1. Do circles
$$C_i := \{(x,y) \in \mathbb{R}^2 : q_i(x,y) := (x-a_i)^2 + (y-b_i)^2 - c_i = 0\},$$
$i = 1,2$ and $a_i, b_i \in \mathbb{R}$, $c_i \in \mathbb{R}_0^+$, intersect in exactly one, two or no points?

2. Do three circles C_1, C_2 and
$$C_3 := \{(x,y) \in \mathbb{R}^2 : q_3(x,y) := (x-a_3)^2 + (y-b_3)^2 - c_3 = 0\},$$
$a_3, b_3 \in \mathbb{R}$, $c_3 \in \mathbb{R}_0^+$, intersect in a common point and in which order do C_2 and C_3 intersect the circle C_1?

For two circles, there are two degenerate situations of tangential intersection; see Figure 2. W.l.o.g. assume $c_1 \geq c_2$. The distance $D := \sqrt{(a_1-a_2)^2 + (b_1-b_2)^2}$ of the centers is either $\sqrt{c_1} + \sqrt{c_2}$ or $\sqrt{c_1} - \sqrt{c_2}$. Hence, the following predicate function detects these situations:

$$\begin{aligned} f(a_1,a_2,b_1,b_2,c_1,c_2) &:= (D - \sqrt{c_1} - \sqrt{c_2}) \cdot (D - \sqrt{c_1} + \sqrt{c_2}) \\ &= (D - \sqrt{c_1})^2 - c_2 \\ &= D^2 + c_1 - c_2 - 2\sqrt{c_1(a_1-a_2)^2 + c_1(b_1-b_2)^2)} \end{aligned}$$

We remark that the circles intersect in exactly one point iff $f = 0$, do not intersect iff $f > 0$, and intersect in two distinct points iff $f < 0$. Since $D^2 + c_1 - c_2 \geq 0$ it follows that $f(a_1,a_2,b_1,b_2,c_1,c_2) = 0$ is equivalent to

$$g(a_1,a_2,b_1,b_2,c_1,c_2) := ((a_1-a_2)^2 + (b_1-b_2)^2 + c_1 - c_2)^2 - 4c_1((a_1-a_2)^2 + (b_1-b_2)^2) = 0.$$

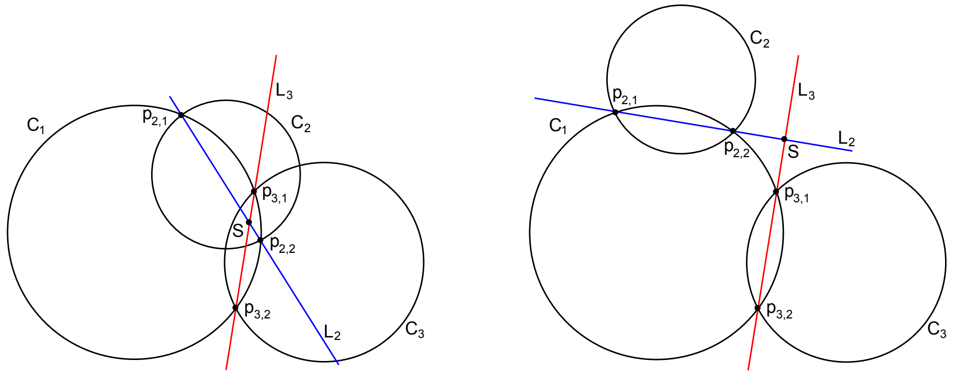

Figure 3: The location of the intersection point $S = L_2 \cap L_3$ with respect to C_1 determines whether the two pairs of points $\{p_{i,1}, p_{i,2}\}$, $i = 2,3$ are interleaving or not.

Furthermore, we have $g > 0$ iff C_1 and C_2 do not intersect and $g < 0$ iff the circles intersect in two distinct points. In terms of coordinates $(z_1,\dots,z_6) := (a_1,a_2,b_1,b_2,c_1,c_2)$ we obtain a multivariate polynomial of total degree $N = 4$ consisting of $m_g = 34$ monomial terms:

$$\begin{aligned} g(z_1,\dots,z_6) = & -4z_1z_2z_3^2 - 4z_1z_2z_4^2 + 8z_1z_2z_3z_4 + 4z_3z_4z_6 + 4z_1z_2z_5 + 4z_3z_4z_5 \\ & + 4z_1z_2z_6 - 4z_3z_4^3 - 4z_1^3z_2 + 6z_1^2z_2^2 + 2z_1^2z_3^2 + 2z_1^2z_4^2 - 2z_1^2z_5 - 2z_1^2z_6 \\ & - 4z_1z_2^3 + 2z_2^2z_3^2 + 2z_2^2z_4^2 - 2z_2^2z_5 - 2z_2^2z_6 - 4z_3^3z_4 + 6z_3^2z_4^2 - 2z_3^2z_5 - 2z_3^2z_6 \\ & - 4z_2^2z_3z_4 - 4z_1^2z_3z_4 + z_6^2 - 2z_5z_6 + z_5^2 - 2z_4^2z_6 - 2z_4^2z_5 + z_1^4 + z_2^4 + z_3^4 + z_4^4. \end{aligned}$$

We have $\mathrm{lm}(g) = z_6^2$, $\mathrm{lcf}(g) = 1$, $c_g = 100$, and $k^* = 2$. Hence it suffices to work with a precision

$$L \geq 22.06\ldots + 4\left(\log\frac{M}{\delta} + \log\frac{1}{1-\rho}\right)$$

to guarantee that the probability of a successful perturbation is larger than ρ.

Now let us find a predicate to answer the second question. If one of the circles C_2 or C_3 does not intersect C_1, there is nothing to do. Thus, we assume that each of them intersects C_1 in two points $\{p_{i,1}, p_{i,2}\} := C_i \cap C_1$, $i = 2,3$; the points may coincide. The difference

$$l_i(x,y) := (q_1 - q_i)(x,y) = 2(a_i - a_1)x + 2(b_i - b_1)y + a_1^2 - a_i^2 + b_1^2 - b_i^2 + c_i - c_1$$

of the two defining equations of C_1 and C_i is a linear equation in x and y and its vanishing set is the unique line L_i passing through the points $p_{i,1}$ and $p_{i,2}$. In the degenerate case $p_{i,1} = p_{i,2}$ the line L_i intersects C_1 tangentially at $p_{i,1}$. Then (see also Figure 3):

- $L_1 = L_2$ if and only if $\{p_{2,1}, p_{2,2}\} = \{p_{3,1}, p_{3,2}\}$.
- If $L_1 \neq L_2$ and $S := L_2 \cap L_3$ lies on C_1, then there exists exactly one common intersection point of C_1, C_2 and C_3, namely S.
- The pairs $\{p_{i,1}, p_{i,2}\}$, $i = 2,3$, of crossings with C_1 are interleaving if and only if S lies in the interior of C_1.

Hence, in order to get information about the order of the intersection points on C_1 we have to compute the lines L_i and their intersection $S = (x_0, y_0)$. Finally, we have to check the sign of $q_1(x_0, y_0)$. The coordinates x_0 and y_0 are obtained by solving the system $l_1 = l_2 = 0$ of linear equations; thus

$$x_0 = \frac{-a_1^2 b_3 + a_1^2 b_2 + \ldots - b_1^2 b_3 + c_3 b_2}{2(-a_2 b_1 - b_2 a_3 + a_2 b_3 - a_1 b_3 + b_1 a_3 + b_2 a_1)}$$

and

$$y_0 = -\frac{a_2 a_1^2 - a_2 b_3^2 + \ldots - c_2 a_3 - b_2^2 a_1}{2(-a_2 b_1 - b_2 a_3 + a_2 b_3 - a_1 b_3 + b_1 a_3 + b_2 a_1)},$$

where we omitted some of the terms in the numerators to preserve readability.[10] Plugging (x_0, y_0) into $q_1 = 0$, the defining equation of C_1, we obtain

$$q_1(x_0, y_0) = \frac{-2a_1 a_3^3 c_2 + 2c_1 b_3^3 b_1 + \ldots + 4a_1^3 a_2 b_3 b_2 - 6a_2^2 a_1^2 a_3^2}{4(-a_2 b_1 - b_2 a_3 + a_2 b_3 - a_1 b_3 + b_1 a_3 + b_2 a_1)^2}$$

with a numerator $h \in \mathbb{Z}[a_i, b_i, c_i]$ consisting of $m_h = 335$ monomial terms in the 9 variables a_i, b_i and c_i, $i = 1, 2, 3$. The sign of $q_1(x_0, y_0)$ is identical to the sign of h, as the denominator of $q_1(x_0, y_0)$ is always nonnegative. Rewriting h in terms of the variables $(z_1, \ldots, z_9) := (a_1, a_2, a_3, b_1, b_2, b_3, c_1, c_2, c_3)$ and considering our monomial ordering $\succ$ the leading monomial term of h is given by $z_5^2 z_9^2$ and the leading coefficient equals 1. Furthermore, its total degree equals 6 and $\|h\|_\infty = 8$. Thus $c_f \le 8m_f$. Now Theorem 6 implies that

$$\begin{aligned} L &\ge \log(8m_f(m_f + 2N)) + 6\log M + 4\left(\log(8e) + \log\frac{1}{\delta} + \log\frac{1}{1-\rho})\right) \\ &= 36.12\ldots + 6\log M + 4\left(\log\frac{1}{\delta} + \log\frac{1}{1-\rho}\right) \end{aligned}$$

guarantees that the sign of $q_1(x_0, y_0)$ can be evaluated successfully with probability larger than ρ.

3.5 Floating Point Perturbations

We address the issue that the analysis is carried out in real space but an actual implementation will choose perturbations in the set of floating point numbers. We performed the theoretical analysis in the real space $\mathbb{R}^k$; the perturbation of a point is a random point in the rectangular δ-neighborhood of the point. However, in an actual implementation the perturbed points have to belong to the discrete set $\mathbb{F}_L$ of floating point numbers of precision L. Previous papers remarked about this issue that for simplicity the analysis is carried out in the real space.

We have taken a different route here. Observe that our error analysis explicitly takes into account that real arguments are rounded to the nearest floating point number (Lines 1 and 3 in Table 1 and Theorem 12). Theorem 12 states that for any polynomial f of total degree N in k variables and any $(z_1, \ldots, z_k) \in [-M, M]^k$

$$|f(z_1, \ldots, z_k) - \widetilde{f}(fl(z_1), \ldots, fl(z_k))| \le K_f M^N 2^{-L}.$$

where $\widetilde{f}$ is the floating point version of f, i.e., all operations in f are replaced by their floating point counterpart, K_f is a suitable constant, and for any $x \in \mathbb{R}$, $fl(x)$ is a nearest (it is not important how ties are broken) floating point number (with mantissa length L).

[10] This and the following computations are performed with the Computer Algebra System Maple 12.

Theorem 7 *Let $\overline{z} \in [-M,+M]^k$ be such that $U := U_\delta(\overline{z}) \subseteq A$ and let $\mathbb{F}_L$ be the set of floating point numbers with mantissa length L. For any $u \in \mathbb{F}_L^k$, let p_u be the probability that $u = fl(z)$ for a random $z \in U$ (rounding is componentwise). Then Theorem 6 stays true if instead of choosing $z \in U$ uniformly at random, we choose $z \in \mathbb{F}_L^k$ according to the distribution $(p_u)_{u \in \mathbb{F}_L^k}$.*

Proof: The floating point evaluation of $f(z)$ is tantamount to computing $\widetilde{f}(u)$ since the first step in the evaluation is rounding z to $fl(z)$. ∎

How can we generate floating point numbers with the desired probabilities? Since coordinates are perturbed independently, we may restrict to a single coordinate. Let $\overline{z} \in [-M,M]$ be such that $U_\delta(\overline{z}) \subseteq A$. In order to reduce boundary effects, we select a $U \subseteq U_\delta(\overline{z})$ of width at least δ such that generating a random $z \in U$ is particularly simple; this will also give us a simple process for generating $fl(z)$. Reducing the size of the perturbation region by a factor of two does not change the character of our bounds; it only affects constant factors.

Let $e \in \mathbb{Z}$ be such that $2^{e-1} < \delta \le 2^e$. Then there is an integer W such that $z - \delta \le W \cdot 2^e < (W+1) \cdot 2^e \le z + \delta$. Let α be the longest common prefix of the binary representations of W and $W+1$, respectively. Then $\alpha 01^\ell$ and $\alpha 10^\ell$, where $\alpha \in \{0,1\}^*$ and $\ell \ge 0$, are the binary representations of W and $W+1$, respectively. We can choose the binary representation of a random real in the interval $U := [W, W+1] \cdot 2^e$ by first selecting either $\alpha 01^\ell$ or $\alpha 10^\ell$ with probability 1/2 each and then continuing random bit by random bit (or continuing in blocks of random bits). Continuing forever, we obtain the binary representation of a random real $z \in [W, W+1] \cdot 2^e$. In order to determine $fl(z)$, we do not have to continue forever, we can stop as soon as $fl(z)$ is determined. When is this the case? The binary representation of z is $\alpha(0|1)(1|0)^\ell \ldots \cdot 2^e$. When the number of bits following the leading one in this bitstring exceeds L, $fl(z)$ is known. Thus no more than L additional bits are needed except in one situation: There is no 1 in $\alpha(0|1)(1|0)^\ell$, i.e., α is empty and $\ell = 0$. Then we need to generate $L+1+r$ bits, where r is the number of leading zeros that we generate. The probability of generating r leading zeros is 2^{-r} and hence the expected number of bits to be generated in $L + O(1)$ in all situations. We summarize the discussion.

Lemma 2 *Let $\overline{z} \in [-M,M]$ be such that $U_\delta(\overline{z}) \subseteq A$. Then we can find a $U \subseteq U_\delta(\overline{z})$ of width at least δ such that $fl(z)$ for a random $z \in U$ can be generated in expected time $O(L)$.*

3.6 Analysis of a Complete Algorithm

We show how to extend the analysis of a single predicate to the analysis of a complete algorithm. Consider, for concreteness, an algorithm with input $\overline{z} \in \mathbb{R}^n$ that uses two geometric predicates. The predicates are implemented as the signs of polynomials f_1 and f_2, respectively. Our goal is to guarantee that the algorithm succeeds on a perturbation $z \in U_\delta(\overline{z})$ with probability at least $1/2$.

Let f_i be a polynomial of total degree N_i in k_i variables. Then there are no more than n^{k_i} argument tuples of k_i distinct arguments. If we guarantee that f_i fails on any specific k_i-tuple of arguments with probability at most $1/(4n^{k_i})$, the probability that f_i fails on some k_i-tuple of arguments is at most $1/4$ and hence the probability that either f_1 or f_2 fails on some argument is at most $1/2$. Thus the algorithm succeeds with probability at least 1/2.

Each of the two bounds on the error probability yields a lower bound on L. The larger of the bounds determines the value of L. Of course, the argument above extends to any number of predicates. Many algorithms in computational geometry use a small number of primitives of bounded arity and hence are covered by this argument, e.g., convex hulls, Delaunay triangulations, and Voronoi diagrams. We give a concrete example. The incremental Delaunay diagram algorithm uses the 2d-orientation and the 2d-side-of-circle predicate. There are at most n^3 invocations of the former predicate and at most n^4 invocations of the latter. Thus it suffices to guarantee that an orientation predicate fails with probability at most $1/(4n^3)$ and that a side-of-circle predicate fails with probability at most $1/(4n^4)$.

Theorem 8 *Let f_1 to f_r be multivariate polynomials such that each f_i is a nonzero polynomial of total degree at most N, has at most m monomial terms, $c_{f_i} \le c$, and $\mathrm{lcf}(f_i) \ge 1$. If an idealistic algorithm branches only on the signs of f_1 to f_r and the n inputs are randomly perturbed by at most δ and are bounded by M after perturbation (where M is an integral power of two), then the corresponding guarded algorithm fails with probability at most ε provided the precision L of the floating point system satisfies*

$$L \ge \log(c(m+2N)) + N\left(3 + \log N + \log\frac{M}{\delta} + \log r + N\log n + \log\frac{1}{\varepsilon}\right) \tag{7}$$

$$= \Omega(1) + N\left(\log\frac{M}{\delta} + N\log n + \log\frac{1}{\varepsilon}\right) \tag{8}$$

Proof: There are at most n^N distinct invocations for each of the f_i. Since each f_i is a nonzero polynomial we can apply Thm 6; we apply it with

$$\rho = 1 - \frac{\varepsilon}{rn^N}.$$

Then the probability that a fixed f_i fails on any specific k_i-tuple of inputs (k_i is the arity of f_i) is at most $\varepsilon/(rn^N)$ and hence the probability that some f_i fails on some k_i-tuple of distinct inputs is at most ε. We conclude that the guarded algorithm fails with probability at most ε.

Substituting the expression for ρ into equation (6) leads to condition (7). ▮

Some algorithms apply predicates to derived values, e.g., the plane-sweep algorithm for line segment intersection locates intersection points of input segments with respect to input segments. Usually, such predicates can be reformulated in terms of inputs[11] and then the analysis applies.

3.7 Efficiency of CP Algorithms

Controlled perturbation can be used without analysis. One starts with an idealistic algorithm, turns it into a guarded algorithm by guarding the evaluations of all predicates, and puts the guarded algorithm into a controlled perturbation loop as shown in Figure 4.

A predicate evaluation may be guarded in different ways. Suppose we branch on the sign of some expression E. We either perform an error analysis for E as described in Section 6 and use one of the guards derived there or we evaluate E with interval arithmetic and abort whenever the resulting interval contains zero.

The maximum allowable perturbation is usually dictated by the application. For example, if we design an object that is to be fabricated with a machine that has a tolerance of δ, we may allow a perturbation of up to δ. Or if the inputs are determined by physical measurements with error margin δ, we may allow perturbation of up to δ.

[11] Assuming that line segments are specified by their endpoints, the predicate would become a function of six input points.

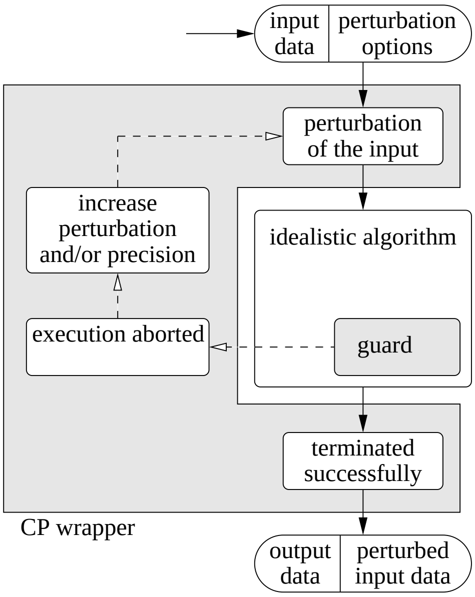

Figure 4: The control flow of the general CP template.

What is a suitable rule for increasing the precision? Let us assume that the cost of arithmetic with floating point numbers of precision L is $O(L^\alpha)$, where $1 \le \alpha \le 2$; $\alpha = 2$ corresponds to classical arithmetic and $\alpha = 1$ corresponds to fast arithmetic (ignoring logarithmic factors). Let us also assume that we have an algorithm that performs at most $T(n)$ steps on an input of size n. Then the cost of the algorithm on input size n and with precision L is $T(n)L^\alpha$. We also assume that for each fixed precision we do up to h iterations, and that after h unsuccessful iterations with the same precision, we increase the precision by a factor t. Let L_0 be the smallest value of L such that the probability of a successful execution is at least $1/2$. In order to bound the cost of the execution, we consider the executions with precision at most L_0 and the executions with precision more than L_0. The cost of the former executions is at most

$$T(n) \cdot \sum_{i \ge 0} h(L_0/t^i)^\alpha = O(T(n)L_0^\alpha).$$

The expected cost of the latter executions is at most

$$T(n) \cdot \sum_{i \ge 0} \sum_{1 \le j \le h} j(tL_0t^i)^\alpha 2^{-hi-(j-1)} = T(n)t^\alpha L_0^\alpha \cdot \sum_{i \ge 0} (t^\alpha 2^{-h})^i \sum_{1 \le j \le c} j2^{-(j-1)} = O(T(n)L_0^\alpha)$$

since the first such execution uses precision at most tL_0 and we proceed to precision tL_0t^i only if all preceding executions have failed. The last equality holds if $t^\alpha < 2^h$.

Theorem 9 *If at any fixed precision, up to h iterations are performed, and precision is increased by a factor of t after h unsuccessful iterations at a fixed precision, L_0 is the smallest value of L such that the probability of a successful execution on input size n is at least 1/2, the cost of arithmetic is $O(L^\alpha)$ with $1 \le \alpha \le 2$, and $t^\alpha < 2^h$, then the expected cost of the CP algorithm is*

$$O(T(n)L_0^\alpha).$$

4 Geometric Insight versus General Methodology

The analysis of the preceding section is basically analytical. It uses geometry only in a weak way, namely when the proof of Theorem 1 argues about the roots of a polynomial. However, the analysis does not exploit any specific geometric properties of the predicate. In particular, it does not give a geometric interpretation of the value of a predicate function. For the orientation function of $d+1$ points in $\mathbb{R}^d$ such an interpretation is available. The value of the predicate functions is $1/d!$ times the volume of the simplex spanned by the $d+1$ points. In this section, we give further examples of predicate functions whose value has a geometric interpretation. The geometric interpretation also yields a slightly improved analysis. The improvements are only in the constant factors. Constant factors are important in our context, because a few additional bits of precision may force a switch from native floating point arithmetic to software arithmetic. Note however, that the usage of CP discussed in the preceding section will automatically choose a large precision only if necessary.

Distinctness of Points: This example is a warm-up for the other examples. Our input is n points in the plane and we want to verify that they are distinct. We implement distinctness via the squared distance function, i.e.,

$$distinct(p,q) = \operatorname{sign}\left(dist(p,q)^2\right) = \operatorname{sign}\left((p_x - q_x)^2 + (p_y - q_y)^2\right).$$

This is a round-about way of implementing distinctness; simply comparing coordinates would be better as it incurs no round-off error.

The error bound of the polynomial $f = (p_x - q_x)^2 + (p_y - q_y)^2$ is $K_f M^2 2^{-L}$ for some constant K_f. The total degree and the degree of the lead monomial is two. So the general theorem yields the constraint.

$$L \geq \Omega(1) + 2\log M/\delta + 2\log\frac{1}{1-\rho}.$$

There are n^2 possible tests and hence we set $\rho = 1/(2n^2)$ as discussed in Section 3.6. So our constraint becomes

$$L \geq \Omega(1) + 2\log M/\delta + 4\log n.$$

A more geometric reasoning is as follows. We want that any two points have a minimum distance of at least γ, where $\gamma^2 = K_f M^2 2^{-L}$. We imagine that the points are perturbed one after the other. When the last point is perturbed, the previous points exclude a region of volume $n\pi\gamma^2$ of the region of perturbation, i.e., the probability that the perturbation does not guarantee distance γ from all preceding points is at most $n\pi\gamma^2/(4\delta^2)$ and hence the probability that the perturbation of some point does not guarantee this distance is at most $n^2\pi\gamma^2/(4\delta^2)$. Again, we require that the latter probability is at most $1/2$. The constraint on L becomes

$$L \geq \Omega(1) + 2\log M/\delta + 2\log n$$

and so the dependency on n is slightly less. Why is the dependency on n different?

Assume that the point p is fixed and q is still to be perturbed. Then an area of $\pi\gamma^2/(2\delta)^2$ is excluded from the perturbation region for q and hence the probability of failure is $\Theta(\gamma^2/\delta^2)$. In the general analysis, we consider one coordinate of q at a time. For each choice of the, say x-coordinate of q, we exclude an interval of length 2γ for the y-coordinate of q. Thus the probability of failure is $\Theta(\gamma/\delta)$. We need that the probability of failure is less $1/n^2$ and therefore the geometric reasoning of the previous paragraph leads to a better dependency on $\log n$.

Orientation Test in d-space The orientation test for $d+1$ points in $\mathbb{R}^d$ is realized as the sign of a determinant, see Section 2. The value of the determinant is $d!$ times the signed volume of the simplex spanned by the $d+1$ points. This volume may be considered as a distance to degeneracy. The volume of a simplex spanned by points p_1 to p_{d+1} is 1 over d times the $(d-1)$-dimensional volume of the base spanned by the points p_1 to p_d times the distance of p_{d+1} from the hyperplane spanned by p_1 to p_d. Continuing in this way, we obtain:

Lemma 3 *The determinant of (1) is equal to*

$$dist(p_1,p_2)\cdot dist(p_3,h(p_1,p_2))\cdot dist(p_4,h(p_1,p_2,p_3))\cdot\ldots\cdot dist(p_{d+1},h(p_1,\ldots,p_d)),$$

where $h(p_1,\ldots,p_k)$ is the affine space spanned by p_1 to p_k.

Consider now an algorithm that uses the 2d-orientation test and takes n points in the plane as its input. The error bound is again of the form $KM^2 2^{-L}$. The general methodology yields the constraint

$$L \geq \Omega(1) + 2\log M/\delta + 6\log n,$$

where 2 is the degree of the underlying polynomial and $6 = 2\cdot 3$; here 2 is the degree and the 3 reflects the fact that there are $\Theta(n^3)$ possible orientation test.

A more geometric reasoning is as follows. We want that any two points have a distance of at least γ_1 and that any point has a distance γ_2 from the line defined any other two points. If this holds, the orientation determinant has value at least $\gamma_1\gamma_2$. The condition on L is

$$\gamma_1\gamma_2 > KM^2 2^{-L}.$$

Again consider the perturbation of a single point. The $n-1$ other points exclude an area of at most $n\pi\gamma_1^2$ and the $\Theta(n^2)$ lines defined by the other points exclude an area of at most $n^2 2\sqrt{2}\delta 2\gamma_2$; the intersection of the line with the perturbation region has length at most $2\sqrt{2}\delta$ and there must be a margin of γ_2 on both sides of the line. Thus the probability that the perturbation of a point is bad is bounded by

$$C\cdot\frac{n\gamma_1^2 + n^2\delta\gamma_2}{\delta^2}$$

for some constant C. Again we need to require that n times this probability is at most $1/2$. With $\gamma_1 = \delta/(2n)$ and $\gamma_2 = \gamma_1^2/(n\delta)$ the probability constraint is satisfied and the condition on L becomes

$$L \geq \Omega(1) + 2\log M/\delta + 4\log n$$

and so the dependency on n is slightly less.

2d Side-of-Circle Test: We consider the side-of-circle test of four points in the plane. It tells the side of a query point with respect to an oriented circle defined by three points. We have three points $p_i = (z_i, y_i)$, $1 \leq i \leq 3$, and a query point $p = (x,y)$. Let us assume first that the three points are not collinear. Let R be the radius of the circle C defined by the first three points. The standard realization of the 2d side-of-circle test is via lifting the points to the paraboloid of revolution $z = x^2 + y^2$, i.e.,

$$soc(p_1,p_2,p_3,p) = \operatorname{sign} f_{soc}(p_1,p_2,p_3,p) \quad \text{where} \quad f_{soc}(p_1,p_2,p_3,p) = \begin{vmatrix} 1 & z_1 & y_1 & z_1^2+y_1^2 \\ 1 & z_2 & y_2 & z_2^2+y_2^2 \\ 1 & z_3 & y_3 & z_3^2+y_3^2 \\ 1 & x & y & x^2+y^2 \end{vmatrix}.$$

We next show how to interpret this formula in terms of the geometry in the plane. Let $c = (c_x, c_y)$ be an arbitrary point in the plane. Subtracting c_x from all entries in the second column, c_y from all entries in the third column, and adding $-2c_x \cdot$ second column $- 2c_y \cdot$ third column $+ (c_x^2 + c_y^2) \cdot$ first column to the last column does not change the value of the determinant. The entries in the last column become the squared distances of the points from c. We have thus shown that the value of the determinant is invariant under translations. We now specialize c to the center of the circle defined by p_1 to p_3. In this situation, we have

$$\begin{aligned}
|f_{soc}(p_1, p_2, p_3, p)| &= \left\| \begin{matrix} 1 & z_1 & y_1 & R^2 \\ 1 & z_2 & y_2 & R^2 \\ 1 & z_3 & y_3 & R^2 \\ 1 & x & y & x^2+y^2 \end{matrix} \right\| = \left\| \begin{matrix} 1 & z_1 & y_1 & 0 \\ 1 & z_2 & y_2 & 0 \\ 1 & z_3 & y_3 & 0 \\ 1 & x & y & x^2+y^2-R^2 \end{matrix} \right\| \\
&= |(x^2+y^2-R^2) \begin{vmatrix} 1 & z_1 & y_1 \\ 1 & z_2 & y_2 \\ 1 & z_3 & y_3 \end{vmatrix} | \\
&= |2\Delta(x^2+y^2-R^2)| \\
&= |2\Delta| \cdot |\sqrt{x^2+y^2} - R| \cdot (\sqrt{x^2+y^2} + R) \\
&\geq |2\Delta| \cdot R \cdot dist(p, C),
\end{aligned}$$

where Δ is the signed area of the triangle with vertices p_1 to p_3, C is the circle defined by these points, and $dist(p, C)$ is the distance of p from this circle. Let $a = dist(p_1, p_2)$, $b = dist(p_1, p_3)$, $c = dist(p_2, p_3)$, and let α be the angle at p_3 in the triangle (p_1, p_2, p_3). Then $2R = a/\sin\alpha$ and $|\Delta| = (1/2)bc\sin\alpha$ and hence $2R|\Delta| = 1/2 \cdot abc$. We obtain:

Lemma 4 *Let p_1, p_2, p_3 and p be four points in the plane. Then*

$$|f_{soc}(p_1, p_2, p_3, p)| \geq \frac{1}{2} dist(p_1, p_2) dist(p_1, p_3) dist(p_2, p_3) dist(C, p).$$

Proof: We have already argued the formula for non-collinear points p_1, p_2, and p_3. Continuity of the left and right side of the inequality extends the inequality to all situations. For collinear points p_1, p_2, and p_3, C is the line passing through these points. ■

Consider now an algorithm that uses the 2d-side-of-circle test and takes n points in the plane as its input. The error bound is of the form $KM^4 2^{-L}$. The general methodology yields the constraint

$$L \geq \Omega(1) + 4 \log M/\delta + 16 \log n,$$

where 4 is the degree of the underlying polynomial and $16 = 4 \cdot 4$; here one 4 is the degree and the other 4 reflects the fact that there are $\Theta(n^4)$ possible orientation test.

A more geometric reasoning is as follows. We want that any two points have a distance of at least γ_1 and that any point has a distance γ_2 from the circle defined by any other three points. If this holds, the side-of-circle determinant has value at least $\gamma_1^3 \gamma_2 / 2$. The condition on L is

$$\gamma_1^3 \gamma_2 / 2 > K_f M^2 2^{-L}.$$

Again consider the perturbation of a single point. The $n-1$ other points exclude an area of at most $n\pi\gamma_1^2$ and the $\Theta(n^3)$ circles defined by the other points exclude an area of $n^3 C \delta \gamma_2$. Thus the probability that the perturbation of a point is bad is bounded by

$$C \cdot \frac{n\gamma_1^2 + n^3\delta\gamma_2}{\delta^2}$$

for some constant C. Again we need to require that n times this probability is at most $1/2$. With $\gamma_1 = \Theta(\delta/n)$ and $\gamma_2 = \gamma_1^2/(n^2\delta)$, the probability constraint is satisfied and the condition on L becomes

$$L \geq \Omega(1) + 4\log M/\delta + 6\log n$$

and so the dependency on n is slightly less.

Improvements Coming from the Algorithm: Many algorithms in computational geometry are incremental. They obtain the solution for n points from a solution for $n-1$ points by making suitable additions and changes. An example is the incremental construction of Delaunay triangulations. Let D be the Delaunay triangulation for $n-1$ points and let p be an additional point. One first finds the triangle of the triangulation (we assume, for simplicity, that the new point is contained in the convex hull of the existing points) containing p, then splits this triangle into three triangles by connecting p to the corners of the triangle, and finally restores the Delaunay property. The point location step uses orientation tests and locates p with respect to edges of D. The update step uses side-of-circle tests and locates p with respect to the circumcircles of triangles in D. Thus in each update step at most $O(n)$ orientation- and side-of-circle tests are performed.

In this situation, the analysis of the side-of-circle predicate of the preceding section can be sharpened as follows. The perturbation of the n-th point has to avoid n circular regions of volume $\pi\gamma_1^2$ each and $O(n)$ annuli of area $C\delta\gamma_2$ each. Then the constraint for γ_1 and γ_2 becomes

$$C\frac{n\gamma_1^2 + n\delta\gamma_2}{\delta^2} \leq \frac{1}{2n}$$

and hence the constraint for L becomes

$$L \geq \Omega(1) + 4\log M/\delta + 5\log n;$$

this is slightly better than above. The paper [FKMS05] contains more examples of this kind.

5 Future Work

We have introduced a general methodology for analyzing CP algorithms and have shown that it is strong enough to handle all geometric predicates that can be expressed as the sign of a multivariate polynomial. A first challenge is to extend the analysis from polynomials to rational functions or expressions involving square roots. One can eliminate divisions and square roots by reformulation of the predicates as done in the concluding examples of Section 3.4. It would, however, be nice to handle them directly.

We view the input as a point in $\mathbb{R}^n$ and assume that all coordinates can be perturbed independently. Frequently, the input also has combinatorial structure, e.g, the input points are the vertices of a simple polygon. Then the perturbation must preserve the combinatorial structure. In some applications, it may suffice to perturb the polygon as a whole, e.g, by applying a rigid transformation to it. The second challenge is to make controlled applicable to problems whose input has combinatorial structure.

The error analysis given in the appendix (Section 6) assumes that expressions are evaluated by straight-line programs. However, more complex equations will be evaluated with a program involving branching and CP needs to be generalized to this situation; this is our third challenge. For example, we might compute the sign of the determinant of a $d \times d$ matrix A by computing an LU-decomposition $L'U'$ of the matrix and then determining the signs of the determinants of L' and U'. In [FKMS05] the bound predicate

$$\mathcal{B}_d \equiv \left(|\det A| > B_d := 1.01^2 \cdot 100 d^2 2^d M^d \varepsilon \right)$$

was derived for Gaussian elimination with partial pivoting and all entries of A bounded by M in absolute value.

So far, CP was only applied to fairly simple geometric problems. It would be interesting to apply is also to complex geometric objects, e.g., arrangements of algebraic curves; this is our fourth challenge.

6 Appendix: Floating Point Arithmetic and Error Analysis

This appendix is an abbreviated version of the notes for the lecture on floating point numbers and error analysis[12] within a course on Computational Geometry and Geometric Computing held by Eric Berberich, Kurt Mehlhorn, and Michael Sagraloff. All proofs can be found there. The lecture notes are based on the papers [MN94, Fun97, BFS01]; the treatment of square roots is novel.

Hardware floating point arithmetic is standardized in the IEEE floating point standard [IEE87]. A floating point number is specified by a sign s, a mantissa m, and an exponent e. The sign is $+1$ or -1. The mantissa consists of L bits $m_1, \ldots, m_L$, and e is an integer in the range $[e_{min}, e_{max}]$. The range of possible exponents contains zero and $e_{min} \leq -L-2$. The number represented by the triple (s, m, e) is as follows:

- If $e_{min} < e \leq e_{max}$, the number is $s \cdot (1 + \sum_{1 \leq i \leq L} m_i 2^{-i}) \cdot 2^e$. This is called a *normalized* number.
- If $e = e_{min}$ then the number is $s \cdot \sum_{1 \leq i \leq L} m_i 2^{-i} 2^{e_{min}+1}$. This is called a *subnormal* number. Observe that the exponent is $e_{min} + 1$. This is to guarantee that the distance of the largest subnormal number $(1 - 2^{-L}) 2^{e_{min}+1}$ and the smallest normalized number $1 \cdot 2^{e_{min}+1}$ is small.
- In addition, there are the special numbers $-\infty$ and $+\infty$ and a symbol NaN which stands for not-a-number. It is used as an error indicator, e.g., for the result of a division by zero.

Let $F = F(L, e_{min}, e_{max})$ be the set of real numbers (including $+\infty$ and $-\infty$) that can be represented as above.[13] A real number in F is called *representable*, a number in $\mathbb{R} \setminus F$ is called *non-representable*. The largest positive representable number (except for ∞) is $max_F = (2 - 2^{-L}) \cdot 2^{e_{max}}$, the smallest positive representable number is $min_F = 2^{-L} \cdot 2^{e_{min}+1} = 2^{-L+e_{min}+1}$, and the smallest positive normalized representable number is $mnorm_F = 1 \cdot 2^{e_{min}+1} = 2^{e_{min}+1}$.

F is a discrete subset of $\mathbb{R}$. For any real x, let $fl(x)$ be a floating point number closest[14] to x. By convention, if $x > max_F$, $fl(x) = \infty$, and if $x < -max_F$, $fl(x) = -\infty$. Arithmetic on floating point numbers is only approximate; it incurs roundoff error. It is important to distinguish between mathematical

[12] The full version can be found at `http://www.mpi-inf.mpg.de/departments/d1/teaching/ws09_10/CGGC/Notes/Numbers.pdf`

[13] Double precision floating point numbers are represented in 64 bits. One bit is used for the sign, 52 bits for the mantissa ($L = 52$) and 11 bits for the exponent. These 11 bits are interpreted as an integer $f \in [0 \ldots 2^{11} - 1] = [0 \ldots 2047]$. The exponent e equals $f - 1023$; $f = 2047$ is used for the special values and hence $e_{min} = -1023$ and $e_{max} = 1023$. The rules for $f = 2047$ are: If all m_i are zero and $f = 2047$ then the number is $+\infty$ or $-\infty$ depending on s. If $f = 2047$ and some m_i is nonzero, the triple represents NaN (= not a number).

[14] The IEEE-standard also specifies how to break ties. This is of no concern here.

E	condition	$\widetilde{E}$	m_E	ind_E	c_E	$\deg E$
a	constant in $\mathbb{R} \setminus \mathbb{F}$	$fl(a)$	$\max(mnorm_F, \lvert fl(a) \rvert)$	1	$\max(1, \lvert fl(a) \rvert)$	0
a	constant in $\mathbb{F}$	a	$\max(mnorm_F, \lvert a \rvert)$	0	$\max(1, \lvert a \rvert)$	0
x	var. ranging over $\mathbb{R}$	$fl(x)$	$\max(mnorm_F, \lvert fl(x) \rvert)$	1	1	1
x	var. ranging over $\mathbb{F}$	x	$\max(mnorm_F, \lvert x \rvert)$	0	1	1
$A+B$		$\widetilde{A} \oplus \widetilde{B}$	$m_A \oplus m_B$	$1+\max(ind_A, ind_B)$	$c_A + c_B$	$\max(\deg A, \deg B)$
$A-B$		$\widetilde{A} \ominus \widetilde{B}$	$m_A \oplus m_B$	$1+\max(ind_A, ind_B)$	$c_A + c_B$	$\max(\deg A, \deg B)$
$A \cdot B$		$\widetilde{A} \odot \widetilde{B}$	$\max(mnorm_F, m_A \odot m_B)$	$1 + ind_A + ind_B$	$c_A c_B$	$\deg A + \deg B$
$A^{1/2}$	$\widetilde{A} < \mathbf{u} m_A$	0	$2^{(t+1)/2} \sqrt{m_A}$	$2 + ind_A$	not defined	
$A^{1/2}$	$\widetilde{A} \geq \mathbf{u} m_A$	$\sqrt{\widetilde{A}}$	$\max(\sqrt{\widetilde{A}}, m_A \oslash \sqrt{\widetilde{A}})$	$2 + ind_A$	not defined	

Table 1: The recursive definitions of m_E, ind_E, c_E and $\deg E$. The first two columns specify the case distinction according to the syntactic structure of E, the third column contains the rule for computing $\widetilde{E}$, and the fourth to seventh columns contain the rules for computing m_E, ind_E, c_E and $\deg E$; $\oplus$, $\ominus$, and $\odot$ denote the floating point implementations of addition, subtraction, and multiplication, and $\sqrt{\ }$ denotes the floating point implementation of the square-root operation. Observe that $m_E = \infty$ if either $m_A = \infty$ or $m_B = \infty$.

operations and their floating point implementations. We use $\oplus$, $\ominus$, and $\odot$, for the floating point implementations of addition, subtraction, and multiplication, respectively. Only in this appendix, we use $^{1/2}$ for the square-root operation and $\sqrt{\ }$ for its floating point implementation. Generally, we use $\widetilde{\circ}$ for the floating point implementation of $\circ$. *The floating point implementations of the operations* $+$, $-$, $\cdot$, *and* $^{1/2}$ *yield the best possible result.* This is an axiom of floating point arithmetic. If $x, y \in F$ and $\circ \in \{+, -, \cdot\}$ then

$$x \widetilde{\circ} y = fl(x \circ y)$$

and

$$\sqrt{x} = fl(x^{1/2}).$$

We need bounds on the error in the floating point evaluation of simple arithmetic expressions. Any real constant or variable is an arithmetic expression and if A and B are arithmetic expression, then so are $A+B$, $A-B$, $A \cdot B$, and $A^{1/2}$. The latter assumes that the value of A is nonnegative. For an arithmetic expression E, let $\widetilde{E}$ be the result of evaluating E with floating point arithmetic. The quantity $\mathbf{u} = 2^{-L-1}$ is called *unit of roundoff*. Table 1 gives recursive definitions of quantities m_E, ind_E, c_E and $\deg E$; we bound $|E - \widetilde{E}|$ in terms of them. Intuitively, m_E is an upper bound on the absolute value of E, ind_E measures the complexity of the syntactic structure of E, $\deg E$ is the degree of E when interpreted as a polynomial, and c_E bounds the coefficient size when E is interpreted as a polynomial.

Theorem 10 *If* $ind_E \leq 2^{(L+1)/2} - 1$ *then*

$$|E - \widetilde{E}| \leq (ind_E + 1) \cdot \mathbf{u} \cdot m_E \leq (ind_E + 2) \odot \max(mnorm_F, m_E \odot \mathbf{u}) \leq (ind_E + 3) \cdot \max(mnorm_F, m_E \cdot \mathbf{u}),$$

where ind_E *and* m_E *are defined as in Table 1.*

For the 2d-orientation predicate

$$orient(a,b,c) = \operatorname{sign}((b_x - a_x) \cdot (c_y - a_y) - (b_y - a_y) \cdot (c_x - a_x))$$

for points $a = (a_x, a_y)$, $b = (b_x, b_y)$, $c = (c_x, c_y)$ in the plane we obtain $ind_E = 6$, and

$$m_E = \max(mnorm_F, (\hat{b_x} \oplus \hat{a_x}) \odot (\hat{c_y} \oplus \hat{a_y})) \oplus \max(mnorm_F, (\hat{b_y} \oplus \hat{a_y}) \odot (\hat{c_x} \oplus \hat{a_x})),$$

where $\hat{x} = \max(mnorm_F, |fl(x)|)$.

The error bound of Theorem 10 is only used for guards. For the analysis we use a simpler, but weaker bound. It applies to polynomial expressions, i.e., expressions using only constants, variables, additions, subtractions, and multiplications.

Theorem 11 *For a polynomial expression we have $m_E \leq c_E M^{\deg E}$, where m_E, c_E and $\deg E$ are defined as in Table 1 and M is the smallest power of two with*

$$M \geq \max(1, \max\{|x| : x \textit{ is a variable in } E\}).$$

This assumes that $c_E M^{\deg E}$ is representable.

We next specialize the theorem above to polynomial expressions that are sums of products, i.e., that correspond to the standard representation of polynomials. We consider polynomials in k variables z_1 to z_k. For $\alpha = (\alpha_1, \ldots, \alpha_k)$ let $z^\alpha = z_1^{\alpha_1} \cdots z_k^{\alpha_k}$. Any polynomial f in $\mathbb{R}[z_1, \ldots, z_k]$ can then be written as

$$f(z_1, \ldots, z_k) = \sum_\alpha f_a z^\alpha,$$

where f_α is the coefficient of the monomial term z^α. For simplicity assume that the coefficients are representable as floating point numbers. For a monomial term, $Z = f_\alpha z^\alpha$, we have $c_Z = \max(1, |f_\alpha|)$, $\deg Z = \deg(z^\alpha) = \sum_i \alpha_i$, and $ind_Z = 2 \deg Z$. For the entire polynomial, we have $c_f = \sum_\alpha \max(1, |f_\alpha|)$ and $\deg f$ equal to the total degree of f. The index depends on the order in which we add the monomial terms. If we sum serially, as in $((((t_1 + t_2) + t_3) + t_4) + t_5))$, the index is the number of monomial terms minus one plus the largest index of any monomial term. If we sum in the form of a binary tree as in $((t_1 + t_2) + ((t_3 + t_4) + t_5))$, the index is the logarithm of the number of monomial terms rounded upwards plus the largest index of any monomial term.

Theorem 12 *Let $f(z_1, \ldots, z_k) = \sum_\alpha f_a x^\alpha$ be a polynomial of total degree N. Let $c_f = \sum_\alpha \max(1, |f_\alpha|)$ and let $m_f = |\{\alpha : f_\alpha \neq 0\}|$ be the number of monomial terms in f. Let $M \geq 1$ be a power of two and let z_1 to z_k be real values with $|z_i| \leq M$ for all i. Then*

$$|f(z_1, \ldots, z_k) - \widetilde{f}(fl(z_1), \ldots, fl(z_k))| \leq c_f (m_f + 2N) M^N 2^{-L-1},$$

where $\widetilde{f}$ is the floating point version of f, i.e., all operations in f are replaced by their floating point counterpart.

Proof: We use Theorems 10 and 11. The index is largest if the monomial terms are summed serially. It is then equal to $m_f + 2N - 1$. Also $m_E \leq c_f M^N$. ▮

We apply Theorems 11 and 12 to the 2d-orientation predicate. Let $a = (a_x, a_y)$, $b = (b_x, b_y)$, $c = (c_x, c_y)$ be three points in the plane. Then

$$orient(a, b, c) = \operatorname{sign}((b_x - a_x) \cdot (c_y - a_y) - (b_y - a_y) \cdot (c_x - a_x)).$$

We already determined the index of this expression as 6. The c- and d-values are as follows. For any argument, both values are one, for $X = b_x - a_x$, we have $c_X = 2$ and $\deg X = 1$, for $X = (b_x -$

$a_x)\cdot(c_y-a_y)$, we have $c_X=4$ and $\deg X=2$, and finally for the entire expression we have $c_X=8$ and $\deg X=2$. We conclude that the roundoff error in evaluating $\mathit{orient}(a,b,c)$ with floating point arithmetic is at most

$$7\cdot\mathbf{u}\cdot 8\cdot M^2=56\cdot\mathbf{u}\cdot M^2=28M^2 2^{-L}.$$

where M is the smallest nonnegative power of two bounding all Cartesian coordinates. If we use the alternative formulation

$$\mathit{orient}(a,b,c)=b_xc_y-b_xa_y-a_xc_y-b_yc_x+b_ya_x+a_yc_x$$

we can apply Theorem 12 with $N=2$, $m_f=6$, and $c_f=6$. We obtain that the roundoff error is at most

$$6(6+4)M^2\cdot\mathbf{u}=60M^2\cdot\mathbf{u}=30M^2 2^{-L}.$$

We close this section with the definition of valid guards and bound predicates.

Theorem 13 *Let E be a polynomial expression. Then*

$$G_E\equiv\left(|\widetilde{E}|>(\mathit{ind}_E+2)\odot\max(\mathit{mnorm}_F,m_E\odot 2^{-L-1})\right),\quad \mathcal{B}_E\equiv\left(|E|>(\mathit{ind}_E+2)c_EM^{\deg E}2^{-L}\right) \tag{9}$$

and

$$G_E\equiv\left(|\widetilde{E}|>(\mathit{ind}_E+1)\cdot c_E\cdot M^{\deg E}2^{-L-1}\right),\quad \mathcal{B}_E\equiv\left(|E|>(\mathit{ind}_E+1)c_EM^{\deg E}2^{-L}\right) \tag{10}$$

define pairs of guard and bound predicate. Here $M\geq 1$ is a power of two no smaller than the absolute value of all arguments. This assumes that $c_EM^{\deg E}$ and $(\mathit{ind}_E+1)c_eM^{\deg E}2^{-L-1}$ are representable.

Proof: We first prove (9). Let $K=c_EM^{\deg E}\mathbf{u}$ and assume $|E|>2(\mathit{ind}_E+2)K$. By Theorem 10, $|\widetilde{E}-E|\leq(\mathit{ind}_E+2)\odot\max(\mathit{mnorm}_F,m_E\odot 2^{-L-1})$. Thus, if $|\widetilde{E}|$ is larger than the latter quantity, E and $\widetilde{E}$ have the same sign. Next observe that $\max(\mathit{mnorm}_F,m_E\mathbf{u})\leq K$ since $c_E\geq 1$, $M\geq 1$, $\deg E\geq 0$ and $e_{min}\leq -L-2$, and hence $c_EM^{\deg E}\mathbf{u}\geq\mathit{mnorm}_F$ and since $m_E\leq K$ by Theorem 11. Thus

$$\begin{aligned}|\widetilde{E}|&\geq|E|-|E-\widetilde{E}|>(2(\mathit{ind}_E+2)-(\mathit{ind}_E+1))K=(\mathit{ind}_E+3)K\\&\geq(\mathit{ind}_E+3)\max(\mathit{mnorm}_F,m_E\cdot\mathbf{u})\geq(\mathit{ind}_E+2)\odot\max(\mathit{mnorm}_F,m_E\odot\mathbf{u}),\end{aligned}$$

where the last inequality is part of Theorem 10.

We turn to (10). Let $K=c_EM^{\deg E}\mathbf{u}$ and $|E|>2(\mathit{ind}_E+1)K$. By Theorem 11, $m_E\leq c_EM^{\deg E}$. Thus $|\widetilde{E}-E|\leq(\mathit{ind}_E+1)c_EM^{\deg E}\mathbf{u}$. The latter is a floating point number by assumption and if $|\widetilde{E}|$ is larger than this quantity, E and $\widetilde{E}$ have the same sign. Finally,

$$|\widetilde{E}|\geq|E|-|E-\widetilde{E}|>(2(\mathit{ind}_E+1)-(\mathit{ind}_E+1))K=(\mathit{ind}_E+1)K.$$

■

For the orientation predicate (in expression form), $\mathit{orient}(a,b,c)=\operatorname{sign}((b_x-a_x)\cdot(c_y-a_y)-(b_y-a_y)\cdot(c_x-a_x))$, the second part of Theorem 13 yields the pair

$$G_E\equiv\left(|\widetilde{E}|>28\odot M^2\odot 2^{-L}\right)\qquad\qquad \mathcal{B}_E\equiv\left(|E|>56M^22^{-L}\right). \tag{11}$$

For the orientation predicate (in polynomial form), $\mathit{orient}(a,b,c)=b_xc_y-b_xa_y-a_xc_y-b_yc_x+b_ya_x+a_yc_x$, it yields the pair

$$G_E\equiv\left(|\widetilde{E}|>30\odot M^2\odot 2^{-L}\right)\qquad\qquad \mathcal{B}_E\equiv\left(|E|>60M^22^{-L}\right). \tag{12}$$

Acknowledgement: We want to thank the reviewers for their careful review and very helpful comments.

References

[ACM84] D. S. Arnon, G. E. Collins, and S. McCallum. Cylindrical algebraic decomposition I. *SIAM Journal of Computing*, 13(4):865–889, 1984.

[BFS01] Christoph Burnikel, Stefan Funke, and Michael Seel. Exact geometric computation using cascading. *Int. J. Comput. Geometry Appl.*, 11(3):245–266, 2001.

[Car07] M. Caroli. Evaluation of a generic method for analyzing controlled-perturbation algorithms. Master's thesis, Universität des Saarlandes, 2007.

[CGA] CGAL (Computational Geometry Algorithms Library). `www.cgal.org`.

[ECS97] I.Z. Emiris, J.F. Canny, and R. Seidel. Efficient perturbations for handling geometric degeneracies. *Algorithmica*, 19:219–242, 1997.

[EM90] H. Edelsbrunner and E.P. Mücke. Simulation of simplicity: A technique to cope with degenerate cases in geometric algorithms. *ACM Transactions on Graphics*, 9(1):66–104, January 1990.

[FKMS05] S. Funke, Ch. Klein, K. Mehlhorn, and S. Schmitt. Controlled Perturbation for Delaunay Triangulations. In *SODA*, pages 1047–1056, 2005.

[Fun97] Stefan Funke. Exact arithmetic using cascaded computation. Master's thesis, Universität des Saarlandes, 1997.

[FvW93] S. Fortune and C. van Wyk. Efficient exact integer arithmetic for computational geometry. In *7th ACM Conference on Computational Geometry*, pages 163–172, 1993.

[HL04] D. Halperin and E. Leiserowitz. Controlled perturbation for arrangements of circles. *International Journal of Computational Geometry and Applications*, 14(4):277–310, 2004. preliminary version in SoCG 2003.

[HR] D. Halperin and S. Raab. Controlled perturbation for arrangements of polyhedral surfaces with application to swept volumes. available from Halperin's home page; a preliminary version appeared in SoCG 1999, pages 163–172.

[HS98] D. Halperin and C. Shelton. A perturbation scheme for spherical arrangements with application to molecular modeling. *CGTA: Computational Geometry: Theory and Applications*, 10, 1998.

[IEE87] IEEE standard 754-1985 for binary floating-point arithmetic, 1987.

[JRZ91] M. Jünger, G. Reinelt, and D. Zepf. Computing correct Delaunay triangulations. *Computing*, 47:43–49, 1991.

[Kle04] Ch. Klein. Controlled perturbation for Voronoi diagrams. Master's thesis, Universität des Saarlandes, April 2004.

[KLN91] M. Karasick, D. Lieber, and L.R. Nackman. Efficient Delaunay triangulation using rational arithmetic. *ACM Transactions on Graphics*, 10(1):71–91, January 1991.

[KMP$^+$08] L. Kettner, K. Mehlhorn, S. Pion, S. Schirra, and C. Yap. Classroom Examples of Robustness Problems in Geometric Computations. *Computational Geometry: Theory and Applications (CGTA)*, 40:61–78, 2008. a preliminary version appeared in ESA 2004, LNCS 3221, pages 702 – 713.

[Knu73] D. E. Knuth. *The Art of Computer Programming (Volume I): Fundamental Algorithms*. Addison-Wesley, 1973.

[MN94] K. Mehlhorn and S. Näher. The implementation of geometric algorithms. In *Proceedings of the 13th IFIP World Computer Congress*, volume 1, pages 223–231. Elsevier Science B.V. North-Holland, Amsterdam, 1994. www.mpi-sb.mpg.de/~mehlhorn/ftp/ifip94.ps.

[MN99] K. Mehlhorn and S. Näher. *The LEDA Platform for Combinatorial and Geometric Computing*. Cambridge University Press, 1999.

[MOS06] K. Mehlhorn, R. Osbild, and M. Sagraloff. Reliable and Efficient Computational Geometry via Controlled Perturbation. In *ICALP*, volume 4051 of *LNCS*, pages 299–310, 2006.

[Sei98] R. Seidel. The nature and meaning of perturbations in geometric computing. *Discrete & Computational Geometry*, 19(1):1–17, 1998.

[SY09] M. Sagraloff and C.-K. Yap. An efficient and exact subdivision algorithm for isolating complex roots of a polynomial and its complexity analysis. available at http://www.mpi-inf.mpg.de/~msagralo/ceval.pdf, 2009.

[Yap90] C.-K. Yap. Geometric consistency theorem for a symbolic perturbation scheme. *J. Comput. Syst. Sci.*, 40(1):2–18, 1990.

[Yap97] C.-K. Yap. Towards exact geometric computation. *CGTA: Computational Geometry: Theory and Applications*, 7, 1997.