> *You might object that it would be reasonable enough for me to try to expound the differential calculus, or the theory of numbers, to you, because the view that I might find something of interest to say to you about such subjects is not* prima facie *absurd; but that geometry is, after all, the business of geometers, and that I know, and you know, and I know that you know, that I am not one; and that it is useless for me to try to tell you what geometry is, because I simply do not know.*
>
> — G.H.Hardy, in "What is Geometry?"
>
> 1925 Presidential Address to the Mathematical Association

Lecture 6
Exact Geometric Computation

We characterize the notion of "geometric computation" by pointing out some common features. Along the way, we give one answer to the age-old question: what is geometry? This analysis will lead to a general prescription for attacking nonrobustness in geometric computations.

Throughout the history of mathematics, different answers to this age-old question has been given. From intuitive popular accounts of geometry [4] to quests for the scope and place of geometry [1, 6, 1] The geometric vein is evident is almost every branch of mathematics. Although we begin with geometry, we are soon led to algebra, and throughout the development of geometry, we see a strong interplay between geometry and algebra. Geometry is highly intuitive, certainly much more than algebra. On the other hand, a purely geometric approach cannot progress without a proper algebraic foundation. This is the impulse behind René Descartes' program to algebraize geometry. Once we introduce the Cartesian plane, the intuitive relationship between points and lines can now be reduced to algebraic relations that are amenable to automatic proofs and calculations. Further abstractions are possible: in Felix Klein's Erlangen Program, the essence of geometry is reduced to invariant groups. Modern algebraic geometry has taken this abstraction to yet another level; the progression in these abstractions are described under different "epochs" by Jean Dieudonné's [2].

There is a way to formalize the intuitive geometry without algebra, going back to Euclid. This culminated in Hilbert's axiomatic or formal mathematics. Hilbert also used geometry as the launching point for this work. The formal (logical) approach and the algebraic viewpoints of geometry is given a new synthesis in Alfred Tarski's view of "elementary geometry and algebra". But Tarski's profound insight is the view that real elementary geometry is first order theory of semialgebraic sets. Moreover, there is a decision procedure for this theory based on a generalization of Sturm theory. In this lecture, we will review this aspect of Tarski's work and subsequent development by Collins.

We should mention another aspect of geometry: the combinatorial vein which is associated with the name of Paul Erdös. This connection is important from the computational viewpoint: the complexity of geometric algorithms are often intimately related to combinatorial bounds. The topological connection is another mode of abstraction (again, algebraization will play a central role).

To all these insights about the nature of geometry, we now add the computational perspective.

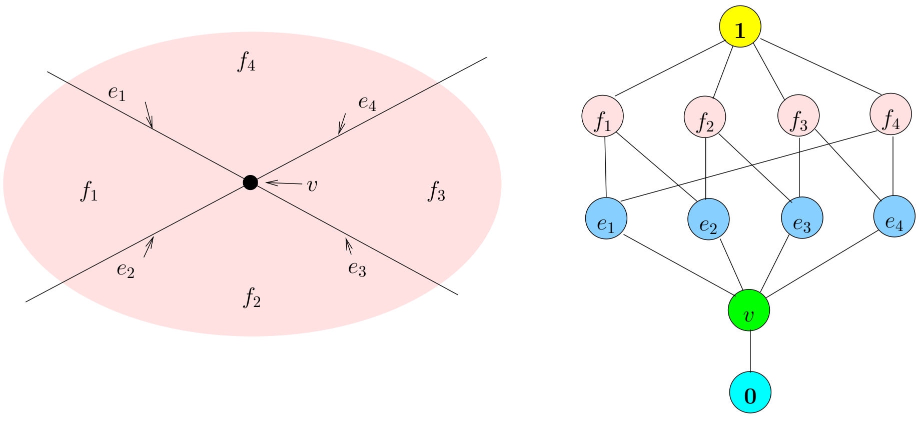
§1. What is Geometry?

When is a computation said to be "geometric"? One answer is "when it involves geometric objects". But what are geometric objects? We begin with the uncontroversial remark that notions such as points, lines, hyperplanes, polytopes, triangulated surfaces are geometric objects *par excellence*. We further observe that the set of all points, the set of all lines, and other basic geometric objects show a common feature: each set constitute a continuum of parametrized geometric objects of the same type. Our notation in the introduction, $\mathsf{Point}(x,y)$ and $\mathsf{Line}(a,b,c)$, hints are this property. In general, we can view geometric objects as $OBJ(x_1,\ldots,x_n)$ where $x_1,\ldots,x_n$ are numerical parameters. We might say this is Descartes' general insight, that geometric objects of a fixed type constitute a parametric space that can be investigated algebraically.

Next, there is a sense of space in geometry. Geometric objects are located in space, and hence they can be in in "spatial relationship" with each other. Thus, $\mathsf{Point}(x,y)$ and $\mathsf{Line}(a,b,c)$ each corresponds to a suitable subset of $\mathbb{R}^2$. It is this embedding in a common substrate that allows spatial relationships to be defined. Examples of spatial relationships include inside/outside, disjoint/intersecting, touching (or incidence), neighborhood (or adjacency) sidedness, co-incidence or other syzyzetic relations. For instance:

- A point can be inside a polytope.
- A line can lie on a hyperplane or be parallel to it, or intersect it transversally.
- Three points may be collinear.
- In dimensions 3 or more, four points may co-coplanar; and when co-planar, they may also be co-circular.
- Two faces of a polytope may be adjacent to each other.

Such relationships are numerically defined, but more importantly, they are[1] **discrete**. A collection of geometric objects will define an implicit collection of such relationships; typically, these relationships can be represented by combinatorial structures such as graphs with parametric labels on its vertices and edges (see [9]). For instance, a collection of points define a convex polytope (i.e., its convex hull). This polytope has facets of various dimensions which are in incidence (or adjacency) relationships. The **incidence graph** [3] is then used to represent this relationship among the facets.



(a) Arrangement of 2 lines

(b) Incidence Graph

Figure 1: Incidence Graph

Figure 1(a) shows a pair of lines in the plane, and the partition of the plane by these two lines into subsets of dimension 0 (the point v), dimension 1 (the line segments e_i's) and dimension 2 (the faces f_i's). These subsets are related in Figure 1(b) by an incidence graph.

More generally, let us define this concept for a collection H of n hyperplanes in d-dimensions: this collection partitions space into pairwise disjoint regions called **faces**, where each face f has a dimension $\dim(f)$ between 0 and d. Two faces f,g are **incident** (on each other) provided (1) $|\dim(f)-\dim(g)|=1$ and (2) either $f \subseteq \overline{g}$ or $g \subseteq \overline{f}$. Here, $\overline{g}$ denotes closure of a set g under the usual topology. The incidence graph $I(H)$ of H comprises a node $n(f)$ for each face f, and an edge from $n(f)$ to $n(g)$ provided $\dim(f)=\dim(g)+1$ and f,g are incident. It is convenient to introduce two **improper faces** denoted $\mathbf{0}$ and $\mathbf{1}$ where $\dim(\mathbf{0})=-1$ and $\dim(\mathbf{1})=d+1$. Moreover, every 0-dimensional face is incident on $\mathbf{0}$ and every d-dimensioan face is incident on $\mathbf{1}$.

Thus, in general, we are dealing with collection of geometric objects in some special relations, which we might term a "geometric complex" for lack of a better term. We continue to write $OBJ(x_1,\ldots,x_n)$ for such

[1] Hoffmann [5] calls them "symbolic".

an object. What is important to realize is that OBJ contains combinatorial data (like graphs) as well as numerical parameters $x_1, \ldots, x_n$. Moreover, the number of numerical parameters for a given class of objects need not be fixed or bounded. This suggests the mnemonic,

$$GEOMETRIC = NUMERIC + COMBINATORIAL \tag{1}$$

But the mere presense of numbers and combinatorial structures in a computation alone does not qualify a computation to be called gemetric. This can be explained by way of two non-examples.
(a) Computing the determinant of a matrix M is non-geometric, even though the matrix structure is a combinatorial structure (albeit very regular one) and there are numerical entries.
(b) Computing the shortest path in a graph with edge weights (say, using Djikstra's algorithm) is likewise non-geometric, even though it has a combinatorial structure (the graph) and numerical values (the weights).

What is implicit in the formula (1) is that there are certain **consistency constraints** that must hold between the numeric and the combinatorial parts. In the extreme case, the numeric part completely determines the combinatorial part. In these two non-examples of geometric computation, there is no relationship between the combinatorial and its numerical parts.

¶1. Remark. Euclidean geometry is the default "geometry" in our discussions. Robustness considerations for other geometries studied in mathematics such as hyperbolic geometry, projective geometry, etc, should have similar considerations as those in the Euclidean case.

§2. What Exact Geometric Computation?

The general outline is this:

- First we prescribe the Exact Geometric Computation (EGC) solution.
- Then we show how EGC is possible if we have computable zero bounds.
- Finally we outline an implementation technique called precision-driven evaluation.

¶2. The EGC Prescription. Having analyzed the concept of "geometry", let us now see how it shows up during a computation. For our purposes, we may view a computation as a possibly infinite rooted tree T, with bounded branching at each step. Each non-terminal node of T will be classified as a **construction step** if it has exactly one child, otherwise it is a **branching step**. An example of a construction step might be

```
$x \leftarrow x + 1$;
$x \leftarrow f(y, z)$;
```

As example of a branching step would be

```
if $z \geq 0$ then goto $L$;
⋮
$L : \cdots$
```

where L is a label. A **test value** is any quantity z that appears in a branching step. In modern computer languages, such "go-to" statements are packaged into case statements, while loops, and so on. The program itself is a finite set of instructions. But when the all the loops and recursions a program is "unrolled" into all possible computation paths, we obtain an infinite branching tree T. Although in principle, binary branching suffices, it is most natural to consider three-way branching for geometric computation because many geometric predicates are naturally 3-valued. E.g. is a point in/out/on a triangle? Is the area of a triangle 0/positive/negative?

Many geometric computations may be classified into one of the following categories: (1) constructing geometric complexes, (2) deriving geometric relationship among geometric complexes, and (3) searching in geometric complexes. Regardless of the category, our understand of geometry implies that the geometric relationships of a particular input instance are encoded by branching choices during a computation.

Although T is infinite, a correct program must halt for any input instance. Halting computations correspond to a path from the root to a leaf. Suppose our computation is to output a geometric object $OBJ(y_1, \ldots, y_k)$ with parameters $y_1, \ldots, y_k$. If the input has parameters $\mathbf{x} = (x_1, \ldots, x_n)$ then each $y_i = y_i(\mathbf{x})$ is a function of the input numbers $x_1, \ldots, x_n$. Thus, all inputs $\mathbf{x}$ that end up at a specific leaf will yield the same parametrized object $OBJ(y_1(\mathbf{x}), \ldots, y_k(\mathbf{x}))$. Although the parameters may vary with $\mathbf{x}$, the combinatorial structure of the output is invariant at each leaf. We come to this general conclusion: *if ensure that every branch is correct, we guarantee the correct combinatorial structure.*

Here then is the prescription of the **Exact Geometric Computation Approach** (EGC): *it is to ensure that all branches for a computation path are correct.*

¶3. The Zero Problem. This seems almost trivial, but what are the issues? We must not forget that the computed values $y_i = y_i(\mathbf{x})$ are all approximate. The EGC prescription does not explicitly tell us how accurately to compute the y_i's. But it does tell us that the test values $z = z(\mathbf{x})$ that appear in our branching steps must be approximated to enough accuracy to determine its sign! Thus, we say that the central problem of EGC is the **sign problem**, to determine the sign of a numerical constant z. Of course, this would be trivial if z as given in an explicit form, say as machine number. Instead, z is given as an expression $z(\mathbf{x})$ in terms of the input parameters. Typically, we may reduce the sign problem to the simpler **zero problem**: to decide if $z(\mathbf{x}) = 0$. If z turns out to be non-zero, then assuming the ability to approximate z to any precision, we can eventually decide whether $z > 0$ or $z < 0$. The zero problem is a very deep question in mathematics, but in particular, it is intimately connected to questions in transcendental number theory.

For our purposes, we can pose the zero problem as follows. Fix a set Ω of real operators and let $Expr(\Omega)$ denote the set of expressions over Ω. A typical example is

$$\Omega_1 = \{\pm, \times, \div\} \cup \mathbb{Z}.$$

Note that the constants $a \in \mathbb{Z}$ are considered 0-ary operators. Thus each expression $e \in Expr(\Omega_1)$ denotes a rational number $\mathtt{val}(e) \in \mathbb{Q}$, if defined. The reason that $\mathtt{val}(e)$ may be undefined is because $\div$ is a partial operator ($a \div b$ is undefined if $b = 0$). These expressions can be viewed as rooted trees, but in general, we view them as directed acyclic graphs (DAG's). In general, there is a natural evaluation function

$$\mathtt{val} : Expr(\Omega) \to \mathbb{R}$$

which is a partial function if Ω contains any partial function such as $\div$.

So the question before us is this: given an expression $e \in Expr(\Omega)$, to decide if $\mathtt{val}(e)$ is defined and if so, return its sign. If we can solve this problem for an algorithm whose numerical operators belong to Ω, then we carry out the EGC prescription for this algorithm.

¶4. Precision-Driven Evaluation So, to achieve EGC, we need to determine the sign of each test value. A simple solution is to determine the sign of *all* numerical values, regardless of whether it is a test value or not. But wastefulness aside, we run into a more basic difficulty: what should be the precision to which each numerical value z be approximated? We cannot tell in advance – it depends on subsequent use of z in the evaluation of another value that depends on z!

To discuss errors/precision, it is useful to introduce this notation: let $z, \widetilde{z}, p \in \mathbb{R}$. An expression of the form "$z\langle p\rangle$" is a shorthand for "$z(1 \pm 2^{-p})$". Therefore

$$\widetilde{z} = z\langle p\rangle \tag{2}$$

means that $|\widetilde{z} - z| \le |z|2^{-p}$, i.e., $\widetilde{z}$ is a p**-bit relative approximation** of z. Similarly, "$z[p]$" is a shorthand for "$z \pm 2^{-p}$". Therefore

$$\widetilde{z} = z[p] \tag{3}$$

means that $\widetilde{z}$ is a p**-bit absolute approximation** of z.

In (2) and (3), should p be regarded as the precision or error in $\widetilde{z}$ as an approximation to z? We will make this distinction: if p is given *a priori* (before $\widetilde{z}$ as computed), then we call it **precision**. But if p is computed *a posteriori* (from z and $\widetilde{z}$), then we call it **error**. E.g., we might say all our computations are carried out to "100 bits of relative precision", and or we might say that a particular approximate value $\widetilde{z}$ has "100 bits of absolute error".

A basic observation (Exercise) is this:

Lemma 1. $\widetilde{z} = z\langle 1\rangle$ *iff* $\mathtt{sign}(z) = \mathtt{sign}(\widetilde{z})$.

Thus, computing z to relative one-bit amounts to determining the sign of z. We propose a general solution [10] that is akin to what is often called lazy evaluation in the computing milieu. We store the extire expression $z(\mathbf{x})$ for each numerical quantity z as a function of the input parameters $\mathbf{x}$. When the sign of z is actually needed, we will evaluate the expression for z to determine the sign. But how is this evaluation carried out?

We will regard each node u in an expression as representing an (exact) numerical value $\mathtt{val}(u)$, and also storing an approximate numerical values $\widetilde{val}(u)$. (It is convenient to abuse notation and use u and $\mathtt{val}(u)$ interchangeably.) In case u is an internal node, it also represents an operation $op(u)$ and it will have a number of children equal to the arity of $op(u)$. E.g., if $op(u)$ is a binary operation, it will have two children. We illustrate this in Figure 2 for a node x In Figure 2(a), we show a multiplication node x (whose value is denoted x, by our abuse of terminology). This node depends two other nodes y and z, and so we have the relation $x = yz$.

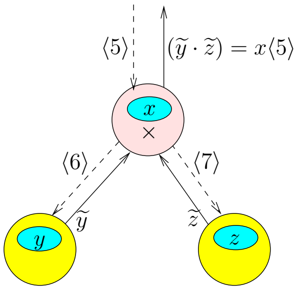


Figure 2: Precision and error propagation at a node

Suppose we want to compute x to p-bits of relative precision. If we require $p+1$-bits of relative precision from y and $p+2$-bits of relative precision from z, then it is easy to see that an exact multiplication of the approximate values will yield the desired result for x. In other words:

$$x = yz, \quad \widetilde{y} = y\langle n+1\rangle, \quad \widetilde{z} = z\langle n+2\rangle. \Rightarrow .\widetilde{y}\cdot\widetilde{z} = x\langle n\rangle. \tag{4}$$

Giving a similar rule for multiplication with absolute error is slightly more involved. On the other hand, we can give a similar rule for addition to absolute precision:

$$x = y+z, \quad \widetilde{y} = y[n+1], \quad \widetilde{z} = z[n+1]. \Rightarrow .\widetilde{y}+\widetilde{z} = x[n]. \tag{5}$$

Giving a similar rule for addition with relative error is slightly more involved. We refer to [8] for this and other rules for the common real operators

$$\pm, \times, \div, \sqrt{\cdot}, \exp, \log\,.$$

Suppose E_z is the expression for z. For simplicity, we imagine E_z to be a tree rooted at z. The leaves of E_z contain the input parameters $\mathbf{x} = (x_1, \ldots, x_n)$. At each node y of E_z, we store an arbitrary precision (bigfloat) approximation $\widetilde{y}$ for y.

§3. Algebraic Background

We assume some familiarity with the following tower of algebraic structures:

$$\mathbb{N} \subseteq \mathbb{Z} \subseteq \mathbb{Q} \subseteq \mathbb{A} \subseteq \mathbb{R} \subseteq \mathbb{C}.$$

The prime example of a ring is $\mathbb{Z}$ (integers), and the rational numbers $\mathbb{Q}$ is our prime example of a field. So we are mainly interested in computation over suitable subsets of the real numbers $\mathbb{R}$, but ocassionally the complex numbers $\mathbb{C}$. Also, we write $\mathbb{A}$ for the set of **real algebraic numbers**. The set of all algebraic numbers is therefore given by $\mathbb{A}[\mathbf{i}]$.

We will try to state our results as concretely as possible. But student less familiar with abstract algebra may you may substitute these prime examples whenever we say "ring" or "field". Much of this material is summarized from my book [7].

Let K be a field. If X is a variable, let $K[X]$ denote the polynomial ring with coefficients in R. A typical polynomial $A = A(X)$ has the form

$$A(X) := \sum_{i=0}^{m} a_i X^i, \qquad a_m \neq 0$$

where each non-zero a_i is called a **coefficient** of A. Here, a_m is called the **leading coefficient** and m is called the **degree** of $A(X)$. These are denoted $\mathtt{lead}(A) = a_m$ and $\deg(A) = m$ respectively. In case $A(X) = 0$, we define[2] $\mathtt{lead}(0) = 0$ and $\deg(0) = -\infty$. If $\mathtt{lead}A = 1$ then A is said to be **monic**. If $\alpha \in K$ and $A(\alpha) = 0$, we say α is a **root** of $A(X)$. There are several ways to measure the size of polynomials that takes into account its coefficients. For any number $p \geq 1$, define the p**-norm**,

$$\|A(X)\|_p := \sqrt[p]{\sum_{i=0}^{m} |a_i|^p}.$$

We can also define $\|A(X)\|_\infty = \max_{i=0}^{m} |a_i|$, The case $p = 2$ and $p = \infty$ (respectively) are also called the (Euclidean) length and height of A.

A pair of polynomials $p, q \in K[X]$ defines a **rational function** in X which is usually denoted p/q. Moreover, if p'/q' is another rational function, then p/q and p'/q' are declared to be equal if and only if $pq' = p'q$. The set of rational functions is denoted $K(X)$.

In case $K = \mathbb{C}$, we have the Fundamental Theorem of Algebra which says that every $p(X) \in \mathbb{C}[X]$ has exactly $\deg(p)$ complex roots. A number $\alpha \in \mathbb{C}$ is said to be **algebraic** if it is the root of a polynomial $p \in \mathbb{Z}[X]$. If further, p is monic, then α ia called an **algebraic integer**. This generalizes our usual concept of integers.

¶5. UFD. Let R be a ring, $a, b \in R$. A domain is a ring in which $ab = 0$ implies $a = 0$ or $b = 0$. We say a **divides** b if there is a c such that $ac = b$. If $ab = 0$ and $b \neq 0$ then we call a a zero-divisor. Thus a domain is a ring whose only zero-divisor is 0. A **unit** a is an element that divides 1. Alternatively, the equation $ax = 1$ has a solution x. E.g., $\mathbb{Z}$ is a domain in which the only units are ± 1 In a field, all non-zero elements are units. A **unique factorization domain** (UFD) is a domain in which every element a can be written as a product $a = \prod_{i=1}^{k} p_i$ of irreducible elements p_i; this product is unique up to units in the sense that if $a = \prod_{i=j}^{\ell} q_j$ then $k = \ell$ and by suitable reordering, each $p_i = u_i q_i$ were u_i are units. The Fundamental Theorem of Arithmetic tells us that $\mathbb{Z}$ is a UFD. A basic result of Gauss says that if R is a UFD if and only if $R[X]$ is a UFD.

In a UFD, we can define the concept of a GCD of a set of elements. In the ring $K[X]$ over a field K, Euclid's algorithm can be extended to compute the GCD of two polynomials.

¶6. Resultants. A basic tool in algebraic number theory is the concept of resultants. In the following, assume D is a UFD. Given $p, q \in D[X]$, we define their resultant $\mathtt{res}(p, q) \in D$ to be the determinant of the

[2] Sometimes, $\deg(0)$ is defined as $+\infty$ instead of $-\infty$. Both definitions have advantages, but neither seems superior: for instance, exactly one of the following statements is true in either definition: (1) "the division algorithm for A, B produces Q, R such that $A = BQ + R$ with $\deg R < \deg B$". (2) "Every complex polynomial of degree m has exactly m roots". Of course, in practice, we just have to add the qualification that A is non-zero. If we define $\deg(0) = +\infty$, then the condition "$\deg A \leq d$" can be used to exclude the case $A = 0$, but then the condition "$\deg A > \deg B$" may produce some surprises. We have opted to avoid surprises.

Sylvester matrix $S(p,q)$ of p,q. If $m = \deg p$ and $n = \deg q$, then $S(p,q)$ is a $m+n$ square matrix whose first n rows are filled with the coefficients of

$$X^{n-1}p, X^{n-2}p, \ldots, Xp, p$$

and the remaining m rows are filled with the coefficients of

$$X^{m-1}q, X^{m-2}q, \ldots, Xq, q.$$

If $p = \sum_{i=0}^m p_i X^i$ and $q = \sum_{j=0}^n q_i X^j$ then

$$S(p,q) \quad = \quad \begin{array}{cc} \underbrace{\left[\begin{array}{cccccccc} p_m & p_{m-1} & \cdots & & p_0 & & & \\ & p_m & p_{m-1} & \cdots & & p_0 & & \\ & & \ddots & & & & \ddots & \\ & & & p_m & p_{m-1} & \cdots & & p_0 \\ \hline q_n & q_{n-1} & \cdots & q_1 & q_0 & & & \\ & q_n & q_{n-1} & \cdots & q_1 & q_0 & & \\ & & \ddots & & & & \ddots & \\ & & & q_n & q_{n-1} & \cdots & & q_0 \end{array}\right]}_{\begin{array}{cccccccc} X^{m+n-1} & X^{m+n-2} & \cdots & X^n & X^{n-1} & \cdots & \cdots & X^0 \end{array}} & \begin{array}{c} X^{n-1}p \\ X^{n-2}p \\ \vdots \\ p \\ \hline X^{m-1}q \\ X^{m-2}q \\ \vdots \\ q \\ \\ \end{array} \end{array}$$

Note that $X^i p$ is a polynomial of degree $m+i$ and its coefficients fill the ith row of $S(p,q)$. The jth column contains the coefficients of X^{m+n-j}, and thus the leading coefficient of $X^i p$ will be in the $(i, n-i)$th entry of $S(p,q)$. The first n rows will therefore contain the coefficients of p but each is a right-shift of the previous row. The last m row is similarly filles with right-shifted coefficients of q.

We now prove a basic lemma:

Lemma 2. *$\mathtt{GCD}(p,q)$ is a constant iff $\mathtt{res}(p,q) \neq 0$.*

Proof. Suppose $\mathtt{res}(p,q) = 0$. This means $\det(S) = 0$ where $S = S(p,q)$. So a non-trivial linear combination of the rows of S vanishes, i.e.,

$$w \cdot S = 0 \tag{6}$$

where

$$w = (u_{n-1}, u_{n-2}, \ldots, u_0, v_{m-1}, v_{m-2}, \ldots, v_0)$$

is a nonzero row vector of length $m+n$. If

$$x = (X^{m+n-1}, X^{m+n-2}, \ldots, X, 1)^T$$

is a column vector of length $m+n$, then (6) is equivalent to

$$w \cdot S \cdot x = 0.$$

This latter equation can be re-written as the polynomial equation

$$U(X)p(X) + V(X)q(X) = 0 \tag{7}$$

where $U(X) = \sum_{i=0}^{n-1} u_i X^i$ and $V(X) = \sum_{j=0}^{m-1} u_j X^j$. Note that U and V has degree at most $n-1$ and $m-1$, respectively, and since w is nonzero, we have $VU \neq 0$. Since D is a UFD, (7) implies that $p(X)$ divides $V(X)q(X)$. Thus,

$$\mathtt{GCD}(p(X), V(X))\mathtt{GCD}(p(X), q(X)) = p(X)$$

$$\deg(\texttt{GCD}(p(X), V(X))) + \deg(\texttt{GCD}(p(X), q(X))) = m = \deg(p(X)).$$

But $\deg(\texttt{GCD}(p(X), V(X))) \leq \deg(V(X)) \leq m-1$. Hence $\deg(\texttt{GCD}(p(X), q(X))) \geq 1$. This proves our claim that $\texttt{GCD}(p, q)$ is non-constant.

Conversely, suppose $\texttt{GCD}(p, q) = g(X)$ is non-constant. Then, letting $U(X) := q(X)/g(X)$ and $V(X) := -p(X)/g(X)$, we see that (7) holds. This implies (6) holds for some non-zero w. This shows S is singular, or $\texttt{res}(p, q) = \det(S) = 0$. **Q.E.D.**

The concept of a resultant can be generalized to the the notion of subresultants. We state some basic properties of resultants: let A, B be polynomials with $\deg A = m, \deg B = n$, $\texttt{lead}A = a, \texttt{lead}B = b$. Also let the roots of A and B be $\alpha_1, \ldots, \alpha_m$ and $\beta_1, \ldots, \beta_n$ (resp.).

Lemma 3.
(i) $\texttt{res}((X-\alpha)p, q) = q(\alpha)\texttt{res}(p, q)$.
(ii) $\texttt{res}(A, B) = a^n \prod_{i=1}^m B(\alpha_i)$.
(iii) $\texttt{res}(A, B) = (-1)^{mn} b^m \prod_{j=1}^n A(\beta_j)$.
(iv) $\texttt{res}(A, B) = a^n b^m \prod_{i=1}^m \prod_{j=1}^n (\alpha_i - \beta_j)$.

It is easy to show (ii)-(iv) from (i). The proof of (i) is somewhat involved, but can be achieved by a direct computation. Also, we have

Lemma 4. *Let $A(\alpha) = 0, B(\beta) = 0$.*
(i) If $\alpha \neq 0$ then $1/\alpha$ is a root of $X^m A(1/X)$.
(ii) $\beta \pm \alpha$ is a root of $\texttt{res}_Y(A(Y), B(X \mp Y))$.
(ii) $\alpha\beta$ is a root of $\texttt{res}_Y(A(Y), Y^n B(X/Y))$.

It follows from the above that the set of algebraic numbers forms a field. Moreover, the algebraic integers forms a ring.

¶7. Root Bounds. There is a very large classical literature on root bounds. Here, we content ourselves with a simple one.

Suppose $p(X) = \sum_{i=0}^m a_i X^i$, and $\alpha \neq 0$ is a root of $p(X)$. Then we have the following bound of Cauchy:

$$\frac{|a_0|}{|a_0| + H_0} < |\alpha| < 1 + \frac{H_m}{|a_m|}$$

where $H_0 = \max\{|a_1|, |a_2|, \ldots, |a_m|\}$ and $H_m = \max\{|a_0|, |a_1|, \ldots, |a_{m-1}|\}$.

Let us first prove the upper bound. If $|\alpha| \leq 1$, the result is true. Otherwise, we have

$$|a_m| \cdot |\alpha|^m \leq H_m \sum_{i=0}^{m-1} |\alpha|^i$$

which leads to the stated bound. For the lower bound, consider the polynomial $p(1/X)X^m$ instead.

We also have root separation bounds: this is slightly more involved to prove, and requires the concept of discriminants. The **discriminant** of a polynomial $A(X)$ is defined to be $\text{disc}(A) := (-1)^{\binom{m}{2}}(1/a)\texttt{res}(A, A')$ where $a = \texttt{lead}A$ and A' is the derivative of A. For instance if $A(X) = aX^2 + bX + C$ then $\text{disc}(A) = b^2 - 4ac$. Two important properties of the discriminant are

- $\text{disc}(A) \in D$
- $$\text{disc}(A) = a^{2m-2} \prod_{1 \leq i < j \leq m} (\alpha_i - \alpha_j)^2$$

  where $\alpha_1, \ldots, \alpha_m$ are all the roots of $A(X)$ in the algebraic closure $\overline{D}$ of the domain D.

It follows from the second property that disc(A) vanishes iff A has multiple roots.

The following bound is from Mahler:

Theorem 5. *Let α, α' be distinct roots of $A(X) \in \mathbb{C}[X]$. Then*

$$|\alpha - \alpha'| > \sqrt{|\mathrm{disc}(A)|}\|A\|_2^{-m+1} m^{-(m+2)/2}(\sqrt{3}.$$

Because of the presense of $\mathrm{disc}(A)$ in the right hand side, this bound is trivial if A has multiple roots. To recover a useful bound, we can replace A by its square-free part (Exercise). There is a generalization due to Davenport which gives a lower bound on a product of differences of roots of A.

¶8. Sturm Theory. Suppose $A, B \subseteq \mathbb{R}[X]$ and $\deg A > \deg B \geq 0$. We define a **generalized Sturm sequence** for (A, B) to be a sequence

$$\overline{A} = (A_0, A_1, \ldots, A_h)$$

where $A_0 = A, A_1 = B$ and for $i = 1, \ldots, h-1$,

$$\beta_i A_{i+1} = \alpha_i A_{i-1} + Q_i A_i$$

where $\alpha_i, \beta_i \in \mathbb{R}$, $Q_i \in \mathbb{R}[X]$ and $\alpha_i \beta_i < 0$, and finally

$$A_h | A_{h-1}.$$

If $B = A'$ (the derivative of A) then we simply call $\overline{A}$ a Sturm sequence for A.

The "standard construction" of such a generalized Sturm sequence is where

$$A_{i+1} = -(A_{i-1} \mod A_i)$$

where $A \mod B = R$ means that there exists $Q \in \mathbb{R}[X]$ such that $A = QB + R$ where $\deg R < \deg B$. By high school division of polynomials, it is seen that Q and R is uniquely determined by these conditions. Moreover, in case $A, B \in \mathbb{Z}[X]$, the standard construction yields a sequence of polynomials where each $A_i \in \mathbb{Q}[X]$. As noted in [7], there are better methods than this standard construction in which the A_i's are computed as integer polynomials. In the following, it is convenient to assume some construction method so that, once A, B are given, the rest of the sequence $\overline{A}$ is determined.

If $\overline{a} = (a_0, a_1, \ldots, a_h)$ is a sequence of real numbers, we define the **sign variation** $\mathtt{Var}(\overline{a})$ of $\overline{a}$ to be the number of sign variations (i.e., transitions from + to − or + to −) after we omit all 0 values from the sequence. E.g. $\mathtt{Var}(2, 0, 3.5, -1.2, 0, -10, 0, 0, 3) = 2$. If $\overline{A}$ is a sequence of real polynomials $(A_0(X), \ldots, A_h(X))$ and $a \in \mathbb{R}$ then define

$$\mathtt{Var}_{\overline{A}}(a) := \mathtt{Var}((A_0(a), \ldots, A_h(a))).$$

If $\overline{A}$ is a generalized Sturm sequence for A, B then we also write $\mathtt{Var}_{A,B}(a)$ for $\mathtt{Var}_{\overline{A}}(a)$. This notation is justified as it is easily shown (Exercise) that the sign variation does not depend on particular choice of $\overline{A}$. Moreover, when $B = A'$, we simply write $\mathtt{Var}_A(a)$ instead of $\mathtt{Var}_{A,A'}(a)$.

Theorem 6 (Sturm). *Let $A(X) \in \mathbb{R}[X]$ have positive degree and $A'(X)$ denotes its derivative. If $a < b \in \mathbb{R}$ such that $A(a)A(b) \neq 0$ then the number of distinct real roots of $A(X)$ in the interval $[a, b]$ is equal to*

$$\mathtt{Var}_A(a) - \mathtt{Var}_A(b).$$

Proof. Let $(A_0, A_1, \ldots, A_h)$ be the Sturm sequence of A. For $a \leq c \leq b$, define $v_i(c) := \mathtt{Var}(A_{i-1}(c), A_i(c), A_{i+1}(c))$ for $i = 0, \ldots, h$ (assume $A_{-1}(c) = A_{h+1}(c) = 0$). We initially assume that $C = \mathtt{GCD}(A, A')$ is constant (so A, A') has no common zero).

- For $i = 1, \ldots, h$, if $A_{i-1}(c) = A_i(c) = 0$ then $A_{i-2}(c) = A_{i+1}(c) = 0$. Thus, if there are two consecutive zeros in $(A_0(c), \ldots, A_h(c))$ then the entire sequence is 0.
- So $A_h(c) \neq 0$ (otherwise, $A_{h-1}(c) = 0$ by definition of h, and hence $A_i(c) = 0$ for all i; in particular c is common zero of A, A', contradiction)
- Call c a special value if there is some $i \in \{0, 1, \ldots, h\}$ such that $A_i(c) = 0$. There are finitely many special values. Moreover, $\mathtt{val}_A(c)$ is constant as c varies between two consecutive special values. In other words, $\mathtt{val}_A(c)$ can only change when c passes through a specail value. Thus, we next try to understand how $\mathtt{val}_A(c)$ changes at special values.

- For any c, there exist a subset $I \subseteq \{0, 1, \ldots, h\}$ such that

$$\mathtt{Var}_A(c) = \sum_{i \in I} v_i(c). \tag{8}$$

For instance, if $\mathtt{val}_A(c) = (A_0(c), A_1(c), \ldots, A_6(c)) = (2, -1, 0, 1, -2, 3, -4)$ then some possible choices of I are $\{0, 4, 6\}$ or $\{1, 3, 5\}$. Moreover, we may choose I such that $i \in I$ implies $A_{i-1}(c)A_{i+1}(c) \neq 0$ unless $i = 0$ or $i = h$. In the preceding example, the choice $I = \{0, 4, 6\}$ has this property, but not $I = \{1, 3, 5\}$.

- CLAIM: For all $i \in I$, we have (1) $v_i(c^-) - v_i(c^+) = 1$ if $i = A_0(c) = 0$, and (2) $v_i(c^-) - v_i(c^+) = 0$ otherwise.

- To see (1), we consider two cases: either A_0 is increasing at c or decreasing at c. If A_0 is increasing at c, then $A'(c) > 0$ and $v_0(c)$ increases by 1 as we pass through c. If A_0 is decreasing at c, we again conclude that $v_0(c)$ increases by 1 through c.

- To see (2), we consider two cases: If $A_i(c) = 0$, then $i > 0$ (because of (1)) and $i < h$ (because $A_h(c) \neq 0$). Then $A_{i-1}(c)A_{i+1}(c) \neq 0$. Then, regardless of the values of $A_i(c^-)$ and $A_i(c^+)$, the value $v_i(c^-) - v_i(c^+)$ is 0. If $A_i(c) \neq 0$, then by our assumption that $A_{i-1}(c)A_{i+1}(c) \neq 0$ unless $i = 0$ or $i = h$, we conclude that $v_i(c^-) - v_i(c^+) = 0$ again.

- Our claim therefore implies that $\mathtt{val}_A(c)$ is constant incrementing each time that c passes through a real zero of A. Thus $\mathtt{Var}_A(a) - \mathtt{val}_A(c)$ equals the number of real zeros of A in $[a, b]$.

Finally, suppose $\deg(A_h) > 0$. The sequence $(A_0/A_h, A_1/A_h, \ldots, A_{h-1}/A_h, 1)$ has the same properties as what we proved in (i). Moreover, the sign variation of this modified sequence at any c, that is not a zero of A_h, is equal to $\mathtt{Var}_A(c)$ **Q.E.D.**

See [7, Theorem 7.3, p. 194] for a very general form of this basic theory of counting sign changes.

Exercises

Exercise 3.1: Prove that $\mathbb{A}[\mathbf{i}]$ is the set of all algebraic numbers. ♢

Exercise 3.2: Let $\overline{A}$ be a generalized Sturm sequence for A, B. Show that the value $\mathtt{Var}_{\overline{A}}(a)$ does not depend on the choice of $\overline{A}$. ♢

Exercise 3.3: Let $A(X) = X^4 - 8X^3 + 2X^2 - 14$ and $B(X) = X^3 + X^3 - 7X^2 + X - 1$.
(i) Compute the standard Sturm sequence for A, B.
(ii) Compute the standard Sturm sequence for A.
(iii) Determine the number of real roots of A. ♢

End Exercises

§4. Exact Numerical Algebraic Number Computation

We now demonstrate that the set $\mathbb{A}$ of real algebraic numbers is a suitable domain for computation. More precisely, our goals in this section are:

- To provide a representation for elements of $\mathbb{A}$.

- To show that the operations $\pm, \times, \div, \sqrt{\cdot}$ and comparisons on $\mathbb{A}$ can be carried out on such representations.

¶9. Isolating Interval Approach. There are several known methods in computer algebrac for achieving the above goals. A standard representation real algebraic numbers is called the **isolating interval** representation. If $\alpha \in \mathbb{A}$, then such a representation of α is a pair $(A(X), I)$ where $A(X) \in \mathbb{Z}[X]$ and I is an isolating interval of $A(X)$ containing α. Moreover, if $w(I) > 0$ then α lies in the interior of I. We will write

$$\alpha \simeq (A(X), I) \tag{9}$$

in this case. Here, "isolating interval" means that I contains a unique real root of $A(X)$. Usually, we also require that $A(X)$ be square-free. The **square-free part** of A is defined as follows:

$$sqfr(A) = \frac{A}{\mathtt{GCD}(A, A')}$$

where A' is the derivative of A. If $sqfr(A) = A$, we say A is **square-free**.

Note that we can refine the interval I in (9) very easily: evaluate $A(mid(I))$. If this is zero, we can replace I be $[mid(I), mid(I)]$. Otherwise, if $A(mid(I))A(\underline{I}) < 0$, we replace I by $[\underline{I}, mid(I)]$; otherwise, replace I by $[\overline{I}, mid(I)]$. This process can be repeated as often as we like to make $w(I)$ as small as we wish.

Most computer algebra books will assume that the standard representation in the power basis, i.e., we have a list of the coefficients of $A(X)$. Therefore, when we want to compute a representation of

$$\gamma = \alpha + \beta$$

where $\beta \simeq (B(X), J)$. By the resultant results of the previous section, we know that $C(X) = \mathtt{res}_Y(A(Y), B(X-Y))$ contains $\alpha+\beta$ as a root. We can replace $C(X)$ by its square-free part if desired. Finally, we would hope that $I + J$ is an isolating interval of $C(X)$. This can be checked using Sturm sequence of $C(X)$. If so, we can output

$$\gamma \simeq (C(X), I + J)$$

If $I + J$ is not a isolating, we half the width of I and J using the refinement step above, and check again if $I + J$ is an isolating interval of $C(X)$. We can repeat this until $I + J$ is an isolating interval. We can similarly perform the other arithmetic operations. We leave it as an exercise to compute $\sqrt{\alpha}$.

Comparisons between α and β can be decided at once if I and J are disjoint. Otherwise, we can repeat unte the root separation bound, and declare $\alpha = \beta$.

¶10. Expression Approach. We now discuss a somewhat unconventional representation: we let $A(X)$ be an expression. In this case, the operations $\pm, \times, \div$ becomes trivial. E.g., to perform the operation

$$x \leftarrow y + z$$

we simply construct a new node for x and make the two children of this node to be the expressions for y and z.

So the main issue is how to form comparisons. This, in turn, can be reduced to checking if a number is zero. We use the method of **constructive zero bounds**: suppose there is a systematic method to attach root bound $B(u) > 0$ to each node u in an expression e with the property that if $\mathtt{val}(u) \neq 0$ then

$$|\mathtt{val}(u)| \geq B(u).$$

The construction of B is constructive in the following sense: we can maintain with each node u a fixed set of numerical parameters, $q_1(u), q_2(u), \ldots, q_k(u)$ such that (1),

$$B(u) = \beta(q_1(u), \ldots, q_k(u))$$

where β is a computable function, and for each node u, we can compute its parameters $q_i(u)$ from the set of parameters at its children. In a later section, we will give such a constructive zero bound.

Now, we can use our precision-driven evaluation mechanism on the expression e to approximate e to absolute p-bits for $p = 1, 2, 4, 8, \ldots$ until such p where $|\widetilde{e}| > 2^{-p}$ or $p = B(e)$. If $\widetilde{e}| > 2^{-p}$, then the sign of e is the sign of $\widetilde{e}$. Otherwise we conclude that $e = 0$.

¶11. EVAL The **root isolation problem** is this: given a polynomial $f(X)$, to compute an isolating interval for each of the real zeros of $f(X)$. A traditional approach is to use the Sturm sequence of $f(X)$, as we have seen in the previous section. But this turns out that more efficient methods are possible. In recent years, the Descartes method (based on the Descartes Rule of Signs) has been popular. We now describe an even simpler approach, based on interval evaluation. Like the Descartes Method, we require $f(X)$ to be square-free.

In fact, f need not be a polynomial. Let $f : I_0 \to \mathbb{R}$ be any real function on an interval $I_0 \subseteq \mathbb{R}$ that satisfies:

- The derivative f' is defined on I_0.
- f has finitely many zeros in I_0, and $f(x) = 0$ implies $f'(x) \neq 0$.
- We can compute the box functions $\square f(I)$ and $\square f'(I)$ where $I \subseteq I_0$ are dyadic intervals.
- For any dyadic number $x \in I_0$, we can determine the sign of $f(x)$.

Any such f is said to belong to $PV(I_0)$ (or to PV if $I_0 = \mathbb{R}$). Then we have the following extremely easy algorithm:

```
EVAL($f, I_0$):
    Input: $f \in PV(I_0)$ where $I_0$ is a finite dyadic interval
            whose endpoints are not roots of $f$.
    Output: A list $L$ of isolating intervals for each real root of $f$ in $I_0$
    ----------------------------------------------------------------
    Initialize two lists, $Q \leftarrow \{I_0\}$ and $L \leftarrow \emptyset$
    while $Q$ is nonempty
        Remove $I$ from $Q$
        if $0 \notin \square f(I)$, discard $I$
        elif $0 \notin \square f'(I)$
            if $(f(\overline{I})f(\underline{I}) < 0)$, $L.append(I)$
        else
            Insert $I_0 = [\underline{I}, mid(I)]$ and $I_1 = [mid(I), \overline{I}]$ into $Q$
            if $f(mid(I)) = 0$, $L.append([mid(I), mid(I)])$
        return($L$)
```

§5. Cylindrical Algebraic Decomposition

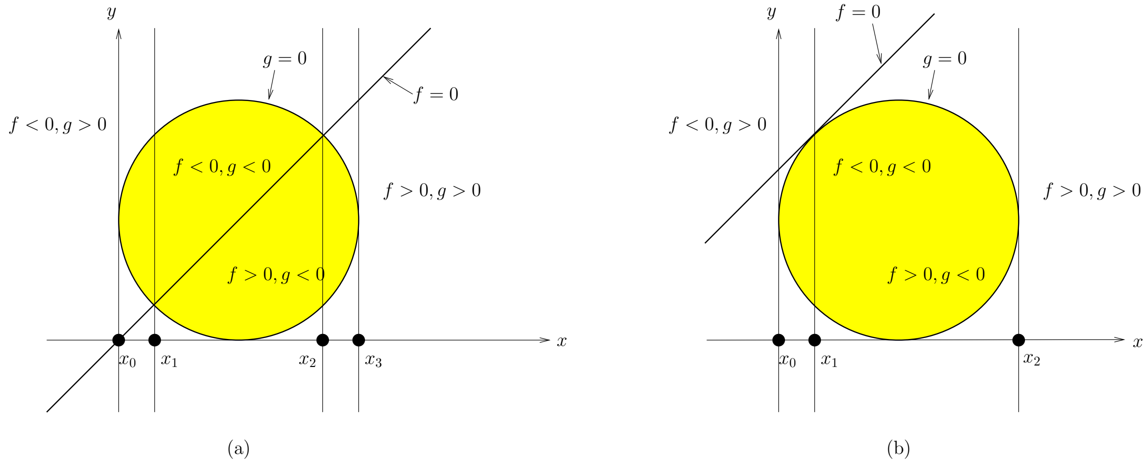
Elementary Geometry usually associated with the study of geometric figures in 2 or 3 dimensions and their relationships, as taught in high school. This tradition goes back to the Greeks as well as many ancient civilizations. It is Descartes' major innovation to consider geometric objects as parametric objects. This allows us to reduce geometric questions to numerical computation.

To see the tremendous power of this approach, consider what the alternative might be. Indeed, historically there is another development of geometry, alongside with the algebraization: this is based on the axiomatic approach. We could not do any geometric computation in the usual understanding of geometric computation. Instead, the computational focus in this parallel development is theorem proving. Here the idea is to encode theorems as statements in first order logic, and to deduce them from the axioms. This turned out to be notoriously difficult – only fairly trivial theorems could be proved by first order logic provers. Subsequent clarification by Wu Wen-Tsun indicated why – most elementary theorems are true only “generically”, not universally. Moreover, the algebraization approach was hugely successful. Nevertheless, there is an interesting convergence of the logical approach with the algebraization approach in the work of Alfred Tarski. According to Tarski, elementary geometry can be encoded in the first order theory of real closed fields. This turned out to be extremely fruitful: it led to a decision procedure for this language. But in the hands of G.E.Collins, this developed a more geometric viewpoint leading to a procedure to decompose Euclidean space into what is now known as a Cylindrical Algebraic Decomposition (CAD). In some sense, this is the superalgorithm which allows us to solve all questions of elementary geometry. Our goal in this section is to give the basic elements of computing CAD.

¶12. Sign-Invariant Decomposition. Consider the following a planar geometric figure consisting of a line $f = 0$ and a circle $g = 0$ where

$$f = X - Y, \qquad g = (X-1)^2 + (Y-1)^2 - 1.$$

This is illustrated in Figure 3(a).



(a)

(b)

Figure 3: Sign-invariant regions of a line $f = 0$ and a circle $g = 0$.

The plane $\mathbb{R}^2$ is partitioned by these equations into connected subsets of dimensions, with the property that the sign of $\Sigma = \{f, g\}$ is invariant in each subset. We call such regions a **sign-invariant region**. Consider the following four sign conditions:

$$\{f < 0, g < 0\}, \quad \{f < 0, g > 0\}, \quad \{f > 0, g < 0\}, \quad \{f > 0, g > 0\}.$$

Each of them determine a unique 2-dimensional sign-invariant region. Next consider the following sign conditions:

$$\{f = 0, g < 0\}, \quad \{f = 0, g > 0\}, \quad \{f > 0, g = 0\}, \quad \{f > 0, g = 0\}.$$

The first condition again determines a unique sign-invariant region. However, because of the equality condition, any sign regions is necessarily 1-dimensional. But the second condition $\{f = 0, g > 0\}$ determines two sign-invariant regions. The third and fourth conditions again determine a unique sign-invariant region. Finally, there are two 0-dimensional sign-invariant regions corresponds to sign condition

$$\{f = 0, g = 0\}$$

They correspond to the two intersection points of the line and the circle.

In general, if $F \subseteq \mathbb{R}[X_1, \ldots, X_n] = \mathbb{R}[\mathbf{X}]$ is a set of polynomials, we say a set $R \subseteq \mathbb{R}^n$ is F**-invariant** if each polynomial in F has a definite sign throughout R. The partition of $\mathbb{R}^n$ into the collection of maximal connected subsets that are F-invariant is called a F**-invariant decomposition** of $\mathbb{R}^n$.

Thus, our example shows that the $\{f, g\}$-invariant decomposition of $\mathbb{R}^2$ has 11 sign-invariant regions.

¶13. Cylindrical Cell Decomposition. All the sign-invariant regions in the preceding example has an additional property: they are each homeomorphic to $\mathbb{R}^d$ if the region is d-dimensional. Such regions will be called **cells** or d**-cells**. In general, sign-invariant regions need not be cells, as we can easily imagine regions that are non-simply connected for $d \geq 2$. We will next subdivide these sign-invariant into smaller sets which will always be cells.

First, we consider certain special values of x,

$$x_0 < x_1 < x_2 < x_3$$

where each x_i is the x-projection of the intersection of the two curves $f = 0, g = 0$, or the x-extremal points of the curve $g = 0$. Note that $f = 0$ has no x-extremal point. We can define a point p to be x-extremal for a general curve $f = 0$ if $f(p) = f_Y(p) = 0$ where $f_Y = \frac{\partial f}{\partial Y}$.

We can say that these special values determine a **cell decomposition** of the x-axis (i.e., $\mathbb{R}$) where the cells are

$$CAD(\emptyset) : \{x < x_0\}, \{x_0\}, \{x_0 < x < x_1\}, \{x_1\}, \{x_1 < x < x_2\}, \{x_2\}, \{x_2 < x < x_3\}, \{x_3\}, \{x_3 < x\}. \tag{10}$$

Note that these cells are ordered in a natural way, and their dimensions alternate between 1- and 0-dimension. This pattern will be repeated.

Next, for each cell C' of (10), we consider the **cylinder** $C' \times \mathbb{R}$. We now look at the intersection of the F-invariant regions with this cylinder. This will decompose $C' \times \mathbb{R}$ into a sequence of cells. If case C' is a 0-cell, we obtain another sequence analoguous of (10). In case C' is a 1-cell, we obtain a sequence

$$\begin{aligned} CAD(C') : \quad & \{y < g_0(x)\}, \{g_0(x) = 0\}, \{g_0(x) < y < g_1(x)\}, \{g_1(x) = 0\}, \{g_1(x) < y < g_2(x)\}, \\ & \{g_2(x) = 0\}, \{g_2(x) < y < g_3(x)\}, \{g_3(x) = 0\}, \{g_3(x) < y\}. \end{aligned} \tag{11}$$

where we write "$\{g_i(x) < y < g_{i+1}(x)\}$" as shorthand for the set $\{(x, y) : x \in C', g_i(x) < y < g_{i+1}(x)\}$. Here, each $g_i : C' \to \mathbb{R}$ is an implicit function locally determined by either $f = 0$ or $g = 0$ over the cell C'. Note that the dimension of the cells in (11) alternate between 2- and 1-dimensions.

The union of all the cells of in $CAD(C')$ as C' range over $CAD(\emptyset)$ will constitute our desired decomposition of $\mathbb{R}^2$. We will call this set of cells the **F-cylindrical decomposition** or a **cylindrical algebraic decomposition** (CAD) of $\mathbb{R}^2$.

¶14. Tarski's Theorem. To understand the significance of the cylindrical nature of a CAD, we return to the logical roots of this definition. Consider the following first order sentence in the theory of closed real fields:

$$\phi_1 : \quad (\forall X)(\exists Y)[(X > Y) \wedge ((X-1)^2 + (Y-1)^2 > 1)]. \tag{12}$$

How can we check if this sentence is true? It is not that easy without some geometric insights! But suppose we rewrite it in the following form:

$$\phi_1 : \quad (\forall X)(\exists Y)[(f(X,Y) > 0) \wedge (g(X,Y) > 0)]$$

where f, g are the polynomials in our example of Figure 3(a). We will call

$$M(X,Y) : (f(X,Y) > 0) \wedge (g(X,Y) > 0)$$

the **matrix** of this form of ϕ_1. Then we begin to see this geometrically: it sufficies to search among the cells of the CAD for $\{f, g\}$. Indeed, searching for all x amounts to scanning each cell C' of $CAD(\emptyset)$ of (10). For each C', we want to know if there is a cell of $CAD(C')$ for which $f > 0$ and $g > 0$. In each case, we can see that there is such a cell. Hence ϕ_1 is a valid.

To make the search more explicit, we write it as a doubly nested loop:

```
$(\forall X)(\exists Y)[M(X,Y)]$:
    for $C' \in CAD(\emptyset)$
        $Found \leftarrow$ false
        for $C \in CAD(C')$
            if ($M(C)$)
                $Found \leftarrow$ true; $Break$;
        if (not $Found$), return(false)
return(true)
```

In the inner for-loop, we write "$M(C)$" for the if-clause. This needs to be explained: recall that $M(X,Y)$ is $(f(X,Y) > 0) \wedge (g(X,Y) > 0)$. Then $M(C)$ is interpreted as the clause "$(f(C) > 0) \wedge (g(C) > 0)$". More precisely, we can pick *any* $(x_0, y_0) \in C$ and evaluate "$(f(x_0, y_0) > 0) \wedge (g(x_0, y_0) > 0)$". This outcome of this predicate evaluation does not depend of the particular choice of (x_0, y_0) since $M(X, Y)$ is sign-invariant on C.

Next, consider the following sentences:

$$\begin{aligned} \phi_2 : &\quad (\forall X)(\exists Y)[(f > 0) \wedge (g < 0)], \\ \phi_3 : &\quad (\exists X)(\forall Y)[(f > 0) \wedge (g > 0)], \\ \phi_4 : &\quad (\exists X)(\forall Y)[(f > 0) \wedge (g < 0)]. \end{aligned}$$

For ϕ_2, our search in $CAD(C')$ fails when $C' = \{x < x_0\}$. Therefore ϕ_2 is invalid. The previous doubly nested loop can be used, except that we replace the if clause by $(f(C) > 0) \wedge (g(C) < 0)$. For ϕ_3 and ϕ_4, we use a somewhat different doubly nested loop:

```
$(\exists X)(\forall Y)[M(X,Y)]$:
    for $C' \in CAD(\emptyset)$
        $Found \leftarrow$ true
        for $C \in CAD(C')$
            if not $M(C)$
                $Found \leftarrow$ false; $Break$;
        if($Found$) return(true);
```

A point $\alpha \in C$ is called a **sample point** of C. We see that in decision problem for sentences, the availability of a sample point $(x_0, y_0) \in C$ for each cell is extremely useful: the abstract predicate $M(C)$ can be replaced by the explicit predicate $M(x_0, y_0)$. We shall see in our algorithms that such sample points can be recursively constructed.

In general, for any sentence ϕ, we can put it into the prenex form:

$$(Q_1X_1)(Q_2X_2)\cdots(Q_nX_n)[M(X_1, \ldots, X_n)] \tag{13}$$

where all the quantifiers $Q_1, \ldots, Q_n$ appear as a prefix, followed by the matrix $M(X_1, \ldots, X_n)$. This matrix is a Boolean combination of atomic formulas of the form

$$f_i(X_1, \ldots, X_n) \circ 0, \qquad i = 1, \ldots, m$$

where f_i is a polynomial and $\circ \in \{<, >, =, \neq, \geq, \leq\}$. Let $F = \{f_1, \ldots, f_m\}$ be all these polynomials. Then we can compute a F-cylindrical decomposition of $\mathbb{R}^n$. The truth of ϕ can be reduced to a n-nested loop to search the cells of the decomposition. Thus, being able to compute CAD's will allow us to decide any sentence of this theory. This is Tarski's fundamental decidability result.

¶15. Resultants as Projection. We want to interpret Lemma 2 as saying that the resultant is a kind of projection operator.

In general, if p, q are multivariate polynomials, we write $\texttt{res}_Y(p, q)$ for the resultant in which we treat p, q as univariate polynomials in one of the variables Y.

In the following two lemmas, let $p, q \in \mathbb{Z}[X, Y]$, and

$$r(X) = \texttt{res}_Y(p, q) \in \mathbb{Z}[X].$$

Moreover, let $\texttt{lead}_Y(p), \texttt{lead}_Y(q) \in \mathbb{Z}[X]$ denote the leading coefficients of p, q, viewed as polynomials in Y.

Before we apply Lemma 2, we prove a helper lemma:

Lemma 7. *Set*

$$p_0(Y) := p(\alpha_0, Y), \qquad q_0(Y) := q(\alpha_0, Y)$$

for some $\alpha_0 \in \mathbb{C}$. *If* $\texttt{lead}_Y(p)(\alpha_0) \neq 0$ *or* $\texttt{lead}_Y(q)(\alpha_0) \neq 0$, *then*

$$r(\alpha_0) = \texttt{lead}_Y(p)(\alpha_0)^m \texttt{res}(p_0, q_0) \tag{14}$$

for some $m \geq 0$.

Proof. Wlog, assume $\mathtt{lead}_Y(p)(\alpha_0) \neq 0$. Thus $\deg(p_0) = \deg_Y(p)$. However, $\deg(q_0) = \deg_Y(q) - m$ for some $m \geq 0$. Clearly, $r(X) = \det S(X)$ where $S(X)$ is the Sylvester matrix of p, q. So $r(\alpha_0) = \det S(\alpha_0)$. But $\mathtt{res}(p_0, q_0) = \det S_0$ for another Sylvester matrix. Moreover, S_0 is obtained from $S(\alpha_0)$ by deleting the first m rows and first m columns of $S(\alpha_0)$. From the shape of the Sylvester matrices, our claim follows. **Q.E.D.**

Lemma 8 (Projection Lemma). *et $\alpha_0 \in \mathbb{C}$ and $r(X)$ be non-vanishing. Then the following two statements are equivalent:*
(i) $r(\alpha_0) = 0$.
(ii) Either
(ii-a) there exists $\beta_0 \in \mathbb{C}$ such that $p(\alpha_0, \beta_0) = q(\alpha_0, \beta_0) = 0$, or
(ii-b) $\mathtt{lead}_Y(p)(\alpha_0) = \mathtt{lead}_Y(q)(\alpha_0) = 0$.

Proof. (i)⇒(ii): Suppose $r(\alpha_0) = 0$. We may assume that (ii-b) does not hold, i.e., $\mathtt{lead}_Y(p)(\alpha_0) \neq 0$ or $\mathtt{lead}_Y(q)(\alpha_0) \neq 0$. Otherwise, by Lemma 7, we see that $r(\alpha_0) = 0$ iff $\mathtt{res}(p_0, q_0) = 0$. Hence $\mathtt{res}(p_0, q_0) = 0$. By Lemma 2, we further conclude that $\mathtt{GCD}(p_0, q_0) = g(Y)$ for some polynomial $g(Y)$ with positive degree. Thus, there exists $\beta_0 \in \mathbb{C}$ such that $g(\beta_0) = 0$. This implies $p_0(\beta_0) = q_0(\beta_0) = 0$. This implies condition (ii).

(ii)⇒(i): Conversely, suppose (ii) holds. As in the proof of Lemma 7, let $S(X)$ be the Sylvester matrix of p, q such that $r(X) = \det S(X)$. If $\mathtt{lead}_Y(p)(\alpha_0) = \mathtt{lead}_Y(q)(\alpha_0) = 0$, then it is clear that $r(\alpha_0) = \det S(\alpha_0) = 0$ since the first column of $S(\alpha_0)$ is entirely zero. Hence, we may assume from (ii) that there exists $\beta_0 \in \mathbb{C}$ such that $p_0(\beta_0) = q_0(\beta_0)$. This implies $\mathtt{GCD}(p_0, q_0)$ is non-constant. By Lemma 2, this means $\mathtt{res}(p_0, q_0) = 0$. Moreover, (14) implies that $r(\alpha_0) = 0$, as desired. **Q.E.D.**

The following corollary shows why this is called a projection lemma:

Corollary 9. *Let $p, q \in \mathbb{Z}[X, Y]$. If $\mathtt{res}_Y(p, q)$ does not vanish, then the set $\mathtt{Zero}(res_Y(p, q)$ contains the x-projections of $\mathtt{Zero}(p, q)$.*

Proof. Suppose $p(\alpha_0, \beta_0) = q(\alpha_0, \beta_0) = 0$. Then $\mathtt{GCD}(p_0, q_0)$ has positive degree, and hence $\mathtt{res}_Y(p_0, q_0) = 0$ **Q.E.D.**

Next assume $\mathtt{res}_Y(p, p_Y)$ is non-vanishing. According to our corollary, $\mathtt{Zero}(res_Y(p, p_Y)$ contains the projections of all the x-extremal points of the curve $p = 0$.

REMARK: An important remark is that even if α_0 is real, this lemma does not guarantee that β_0 would be real. For each real zero α_0 of $\mathtt{res}_Y(p, q)$, we want to "lift" α_0 to the *real* point (α_0, β_0) that satisfy $p = q = 0$. To do this lifting process, we look at each real zero α_0 of $\mathtt{res}_Y(p, q)$. However, many of these α_0 do not lift to any real points! This can fail for two reasons: either it lift to a complex point (α_0, β_0) or this α_0 satisfy condition (ii-b).

¶16. Language of CAD. We now introduce some useful terminology, in order to give a general description of CAD's.

Let $C \subseteq \mathbb{R}^{n-1}$ be any set. Then the **cylinder** over C is the set $C \times \mathbb{R} \subseteq \mathbb{R}^n$. By a **stack** over C we mean a decomposition of $C \times \mathbb{R}$ into an odd number of cells,

$$C_1 < C_2 < \cdots < C_{2m+1}$$

where for each $i = 1, \ldots, m$:

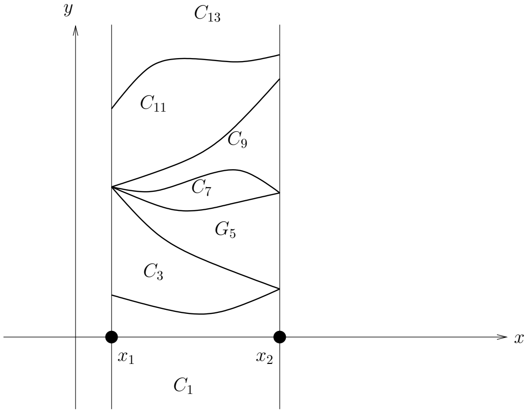
- C_{2i} is called a **section** and is the graph of a real function g_i on C,

$$g_i : C \to \mathbb{R}.$$

- C_{2i-1} is called a **sector** and is given by

$$\{(p, y) \in \mathbb{R}^n : p \in C, g_{i-1}(p) < y < g_{i+1}(p)\}$$

For uniformity in notation, we may assume $g_0 = -\infty$ and $g_{2m+2} = \infty$ in this definition.



(b)

Figure 4: A Stack over C

Such a stack is illustrated in Figure 4.

We now define a **cylindrical decomposition** (CD) to be any finite partition D of $\mathbb{R}^n$ into a collection of cells with the following properties. If $n = 1$, any finite partition of $\mathbb{R}$ into points and open intervals is a CD. Recursively, for $n \geq 2$, there exists a CD D' of $\mathbb{R}^{n-1}$ such that:

- For each $C \in D$, there exists a $C' \in D'$ such that C projects onto C', i.e., $\pi_n(C) = C'$.
- For each $C' \in D'$, the set of $C \in D$ that projects onto C' forms a stack over C'.

If $\pi_n(C) = C'$, we say C is a **child** of C'. We call D' a **projection** of D; conversely D is called an **extension** of D'.

¶17. Projection Operator in the plane. Let $F \subseteq \mathbb{Z}[X_1, \ldots, X_n]$, and D be a CD of $\mathbb{R}^n$.

If each cell C of D is F-invariant, we call D a **cylindrical decomposition** for F. How shall we compute D? If $n = 1$, this is easy (recall that we know how to isolate zeros of polynomials and represent them). Assume $n \geq 2$. The strategy is that we first construct another set $PROJ(F) \subseteq \mathbb{Z}[X_1, \ldots, X_{n-1}]$ such that a cylindrical decomposition D' for $PROJ(F)$ can be extended into a cylindrical decomposition D for F.

The property that makes this possible is the notion of "delineability". We say a function $f : \mathbb{R}^n \to \mathbb{R}$ is **delineable** over a set $S \subseteq \mathbb{R}^{n-1}$ if the set of real zeros of f, restricted to the cylinder $S \times \mathbb{R}$ define the graphs of a finite number of functions of the form

$$g_i : S \to \mathbb{R} \quad (i = 1, \ldots, k)$$

with the following properties:
(i) For $p \in S$,

$$g_1(p) < g_2(p) < \cdots < g_k(p).$$

(ii) For each $i = 1, \ldots, k$, there is a positive integer m_i such that $g_i(p)$ is a root of the polynomial $f(p, X_n)$ of multiplicity m_i.

The graphs of the functions g_i has the form $\{(p, g_i(p)) : p \in S\}$ and are called ***f*-sections**. These f-sections partition the cylinder $S \times \mathbb{R}$ into sets of the form

$$\{(p, z) : g_i(p) < z < g_{i+1}(p)\}$$

for $i = 0, \ldots, k$. Here, we assume $g_0(p) = -\infty$ and $g_{k+1}(p) = +\infty$. We call these the ***f*-sectors**. Thus, the cylinder $S \times \mathbb{R}$ is partitioned into k sections and $k + 1$ sectors. If S is a $(n - 1)$-cell, then each section is an $(n - 1)$-cell, and each sector is an n-cell.

The **order** of a real analytic function $f : \mathbb{R}^n \to \mathbb{R}$ at a point $p \in \mathbb{R}$ is the least k such that some k-th partial derivative of f does not vanish at p. If there is no such k, then the order is ∞. We say f is **order invariant** on a set $S \subseteq \mathbb{R}^n$ if the order of f is constant for all $p \in S$.

Observe that the f-sections and f-sectors are automatically f-sign invariant. We have the following theorem from McCallum:

Theorem 10 (McCallum). *Let $n \geq 2$ and $\overline{X} = (X_1, \ldots, X_{n-1})$. Let $f(\overline{X}, X_n)$ be a polynomial in $\mathbb{R}[\overline{X}, X_n]$ of positive X_n degree. Let $D(\overline{X})$ be the discriminant of $f(\overline{X}, X_n)$ and $D(\overline{X})$ is non-zero. If f is degree-invariant and non-vanishing on $S \subseteq \mathbb{R}^{n-1}$, and D is order-invariant on S, then f is delineable on S and order-invariant on each f-section on S.*

The use of order-invariance in this theorem is somewhat incompatible with the usual emphasis on sign-invariance. Nevertheless, this theorem also shows that order-invariance can be propagated from sign-invariance. Based on this theorem, we may define the following: let F be a **squarefree basis** in $\mathbb{Z}[X_1, \ldots, X_n]$. By squarefree basis, we mean that the polynomials have positive degree in X_n, are primitive, squarefree and pairwise relatively prime. The **projection** of F is the set $PROJ(F) \subseteq \mathbb{Z}[X_1, \ldots, X_{n-1}]$ formed by the union of the following three sets:
(i) $coeff(F)$ is the set of all non-zero coefficients of $f \in F$,
(ii) $disc(F)$ is the set of all non-zero discriminants of $f \in F$,
(iii) $res(F)$ is the set of all non-zero $\texttt{res}_{X_n}(f, g)$ for $f, g \in F$.

¶18. Additional Issues in CAD. This area has remained an active research area even today (2009). One of the most pressing issue is to improve the complexity of CAD. The number of cells is easily seen to be bounded by a double exponential in n, the number of variables. This double exponential complexity is inherent [Davenport and Heintz, 1987]. Although the worst case complexity of CAD construction has improved dramatically over the years, this is still a bottleneck for widespread applications. There are several ways to improve the complexity. For instance, it is clear that the prenex form is in some sense the worst way to decide sentences – most natural sentences have local structures that can be decoupled and solved separately and combined in an effective way. A simple observation is that CAD size is double exponential, not in n (number of variables), but in the number of alternations of quantifiers in the prenex sentence. This means that, for instance, those with no alternation of quantifiers can be solved much faster (e.g., in polynomial space). Another direction is the development of numerical techniques for CAD [Hong, Collins-McCullum, etc]. The variables in a sentence are not completely independent – for instance, if we discuss the geometry of n points in Euclidean d-space, there are nd variables. This fact can be exploited. In 1983, Schwartz and Sharir applied CAD techniques to the problem of robot motion planning. This introduced an additional issue in CAD construction: the need to determine adjacencies between cells. Note that in this application, the sign-invariant regions are primary, and there is no need to have cylindrical decomposition. This can reduce the number of regions to single exponential.

Exercises

Exercise 5.1: Consider the sentence ϕ in the prenex form of (13). Describe the n-nested loop to evaluate the truth of ϕ given a CAD for the polynomials in the matrix of ϕ. ◊

Exercise 5.2: (Ellipse Problems) Consider the ellipse

$$E : \frac{(X-c)^2}{a^2} + \frac{(Y-d)^2}{b^2} = 1.$$

What are the conditions on a, b, c, d so that E is contained in the unit circle $X^2 + Y^2 = 1$. This can be formulated as the sentence

$$(\forall X)(\forall Y)[\frac{(X-c)^2}{a^2} + \frac{(Y-d)^2}{b^2} \leq 1. \Rightarrow .X^2 + Y^2 \leq 1].$$

Describe how we can use CAD to solve this problem.
NOTE: Lazard (1987) succeeded in solving this with the help of MACSYMA. This problem could not be solved by the available CAD software at that time. ◊

Exercise 5.3: In many applications, not all the real variables in a first order sentence are independent. The most important situation is when the variables are naturally grouped into k-tuples. For instance, let $k = 2$, and our sentence is about points in the plane. How can we take advantage of this in constructing CAD? ♢

End Exercises

§6. APPENDIX: Subresultant Theory

In order to understand Collin's theory of CAD, we will delve deeper into Sturm sequences. Indeed, Tarski's original decision procedure was viewed as a generalization of Sturm sequences. For computational efficiency, we also need to delve into the theory of subresultants. For a more complete treatment, see [7].

A ring R is an **ordered ring** if it contains a subset $P \subseteq R$ with the property that P is closed under addition and multiplication, and for all $x \in R$, exactly one of the following three conditions hold:

$$x = 0, \quad x \in P, \quad -x \in P.$$

Such a set P is called a **positive set** of R. Then R is totally ordered by the relation $x < y$ iff $y - x \in P$.

Throughout the following development, assume D is a UFD that is also ordered. For instance, $D = R[X_1, \ldots, X_n]$ where $R \subseteq \mathbb{R}$ is a UFD. In this case, the positive set $P \subseteq D$ can be taken to be those polynomials whose leading coefficient is a positive number in D.

Why is efficiency a problem for computing the standard Sturm sequences in $D[X]$? The reason is that such sequences assumes that D is a field. If D is not a field, we must replace D by its quotient field, $Q(D)$. In order words, when we compute the remainder of polynomials, $A, B \in D[X]$, the result, $\mathtt{rem}(A, B)$ will in general be an element of $Q(D)[X]$, not of $D[X]$. E.g, from $D = \mathbb{Z}$ we must go to $Q(D) = \mathbb{Q}$. This turns out to be very inefficient [7]. We then proceed as follows:

Suppose $A, B \in D[X]$. If $\deg A \geq \deg B$, let us define the **pseudo-remainder** of A, B to be the remainder of $b^{\delta+1}A$ divided by B, where $b = \mathtt{lead}(B)$ and $\delta = \deg A - \deg B$, i.e.,

$$\mathtt{prem}(A, B) := \mathtt{rem}(b^{\delta+1}A, B).$$

It is not hard to see, by looking at the process of long division of polynomials, to see that $\mathtt{prem}(A, B) \in D[X]$. If $\deg A < \deg B$, then $\mathtt{prem}(A, B) := A$.

Given $A, B \in D[X]$, we define a **polynomial remainder sequence** (PRS) of A, B to be a sequence of polynomials

$$(A_0, A_1, \ldots, A_h)$$

where $A_0 = A, A_1 = B$ and for $i \geq 1$,

$$\beta_i A_{i+1} = \alpha_i A_{i-1} + Q_i A_i \tag{15}$$

for some $\beta_i, \alpha_i \in D$ and $Q_i \in D[X]$ and $\deg A_{i-1} < \deg A_i$. Moreover, the termination of the PRS at A_h is determined by the condition that $A_{h+1} = 0$ where A_{h+1} it is defined by (15) from A_{h-1} and A_h. This h is defined because the degree of the A_i is strictly decreasing.

There are many ways to form the PRS of A, B. One way is to use pseudo remainders is to choose (15) to be:

$$A_{i+1} = \mathtt{prem}(A_{i-1}, A_i). \tag{16}$$

This is called the **Pseudo PRS** of A, B. For $D = \mathbb{Z}$, the pseudo PRS will contain generally exponentially large coefficients, and this is not practical. So, instead, we can replace (16) by

$$A_{i+1} = \mathtt{prim}(\mathtt{prem}(A_{i-1}, A_i)). \tag{17}$$

Here, $\mathtt{prim}(A)$ is the **primitive part** of A, defined to be $A/\mathtt{cont}(A)$ where $\mathtt{cont}(A)$ is the GCD of all the coefficients of A. This produces a PRS that is optimal in terms of coefficient sizes. However, computing $\mathtt{prim}(A)$ is quite expensive.

Let $(\beta_1, \ldots, \beta_{h-1})$ where each $\beta_i \in D$. We say that a PRS $(A_0, \ldots, A_h)$ for A, B is **based on** $(\beta_1, \ldots, \beta_{h-1})$ if for $i \geq 1$,

$$A_{i+1} = \frac{\mathtt{prem}(A_{i-1}, A_i)}{\beta_i}. \tag{18}$$

We will describe an algorithm for computing a PRS based on a suitable sequence of β_i's in which the β_i's are easy to compute and the coefficients of the PRS remains polynomially bounded in terms of $\deg A_0$. See [7] for proofs.

Here is the algorithm:

```
SubPRS Algorithm
Input $A, B \in D[X]$, $\deg(A) \geq \deg(B) > 0$
Output PRS $(A_0, \ldots, A_h)$ for $A, B$
▷ INITIALIZATION
        $A_0 \leftarrow A$;    $A_1 \leftarrow B$
        $a_0 \leftarrow \mathrm{lc}(A_0)$;    $a_1 \leftarrow \mathrm{lc}(A_1)$
        $d_1 \leftarrow \deg(A_0) - \deg(A_1)$
        $\psi_0 \leftarrow 1$;    $\psi_1 \leftarrow a_1^{d_1}$
        $\beta_1 \leftarrow (-1)^{d_1+1}$
▷ LOOP
    for ($i = 1$;    $true$;    $i$++)
        $A_{i+1} \leftarrow \frac{\mathtt{prem}(A_{i-1}, A_i)}{\beta_i}$        ◁ Exact Division
        $a_{i+1} \leftarrow \mathrm{lc}(A_{i+1})$
        if ($a_{i+1} = 0$)
            $h \leftarrow i$;  break        ◁ Exit For-Loop
        $d_{i+1} \leftarrow \deg(A_i) - \deg(A_{i+1})$
        $\psi_{i+1} \leftarrow \psi_i \left(\frac{a_{i+1}}{\psi_i}\right)^{d_{i+1}}$
        $\beta_{i+1} \leftarrow (-1)^{d_{i+1}+1} (\psi_i)^{d_i+1} a_i$
```

References

[1] S.-S. Chern. What is Geometry? *Amer. Math. Monthly*, 97(8):679–686, 1990.

[2] J. Dieudonné. *History of Algebraic Geometry.* Wadsworth Advanced Books & Software, Monterey, CA, 1985. Trans. from French by Judith D. Sally.

[3] H. Edelsbrunner. *Algorithms in Combinatorial Geometry.* Springer-Verlag, 1987.

[4] D. Hilbert and S. Cohn-Vossen. *Geometry and the Imagination.* Chelsea, 2nd edition edition, 1952.

[5] C. M. Hoffmann. *Geometric and Solid Modeling: an Introduction.* Morgan Kaufmann Publishers, Inc., San Mateo, California 94403, 1989.

[6] A. Tarski. What is Elementary Geometry? In L. Brouwer, E. Beth, and A. Heyting, editors, *Studies in Logic and the Foundations of Mathematics – The Axiomatic Method with Special Reference to Geometry and Physics*, pages 16–29. North-Holland Publishing Company, Amsterdam, 1959.

[7] C. K. Yap. *Fundamental Problems of Algorithmic Algebra.* Oxford University Press, 2000.

[8] C. K. Yap. On guaranteed accuracy computation. In F. Chen and D. Wang, editors, *Geometric Computation*, chapter 12, pages 322–373. World Scientific Publishing Co., Singapore, 2004.

[9] C. K. Yap. Robust geometric computation. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 41, pages 927–952. Chapman & Hall/CRC, Boca Raton, FL, 2nd edition, 2004.

[10] C. K. Yap and T. Dubé. The exact computation paradigm. In D.-Z. Du and F. K. Hwang, editors, *Computing in Euclidean Geometry*, pages 452–492. World Scientific Press, Singapore, 2nd edition, 1995.