Copyright Note:

Reproduced with permission from the Annual Review of Computer Science Vol. 3.
© 1988 by Annual Reviews Inc.

To be distributed only to participants of 4th ACM Symp. on Comp. Geo.

Note:
These are the uncorrected galley proofs

Ann. Rev. Comput. Sci. 1988. 3:85–119

87-42

ALGEBRAIC METHODS FOR GEOMETRIC REASONING

B. Buchberger

Research Institute for Symbolic Computation, University of Linz, A-4040 Linz, Austria

G. E. Collins

Computer Science Department, Ohio State University, Columbus, Ohio 43210

B. Kutzler

Research Institute for Symbolic Computation, University of Linz, A-4040 Linz, Austria

INTRODUCTION

In recent years, it has been widely recognized that geometric reasoning is one of the most promising areas of computer science. An extrapolation of market trends shows that worldwide the CAD/CAM/CAE industry, which is heavily based on geometric reasoning, expects an expansion from $5 billion in 1985 to $20 billion in 1990 and, hence, is one of the fastest growing industries. This is the practical reason for the challenge inherent in geometric reasoning. On the other hand, the algorithmic problems occurring in geometric reasoning have also an enormous scientific appeal. In the past few years, geometric reasoning problems have provoked a whole spectrum of new algorithmic techniques. Still, many of the more advanced geometric reasoning problems are far from a satisfactory solution, and the pertinent mathematical investigations have hardly scratched the surface. Thus, in the immediate future, we expect to see an enormous expansion of basic and applied research in geometric reasoning.

The present survey reviews recent *algebraic* approaches to the algo-

8756–7016/88/1115–0085$02.00

rithmic solutions of geometric-reasoning problems. Most of these approaches had their origin in computer algebra, the field that attempts to establish algorithms for algebraic problems, and so evolved independently of geometric reasoning. Only recently has it been observed that an intimate contact between the geometric reasoning community and the computer algebra community will be required if further progress in sophistication and a new level of automation are to be achieved in geometric reasoning systems. First reflections of this new trend can be found, for example, in the editorial of the *Journal of Symbolic Computation* (Buchberger et al 1985) and in the scope descriptions and programs of the 1986 International Workshop on Geometric Reasoning (Oxford) (Brady et al 1988), the 1987 SIAM Conference on Applied Geometry (Albany), the 1987 Workshop on Computational Issues in Geometry within the Summer Program on Robotics (IMA, Univ. Minneapolis), the 1988 ACM Symposium on Computational Geometry (Univ. Illinois, Urbana-Champaign), and the 1988 Workshop on Algorithm Aspects of Geometry and Algebra (MSI, Cornell Univ.).

There are at least three reasons why algebraic methods seem promising for the future of geometric reasoning. First, in advanced geometric reasoning applications, the consideration of *global* properties of (quite complicated) *nonlinear* objects is indispensable. These considerations are outside the scope of "traditional" computational geometry, which is almost exclusively concerned with *combinatorial* properties of *linear* (or very simple nonlinear) objects (see Preparata & Shamos 1985). Second, some advanced geometric problems need the *exact representation* of objects as opposed to *numerical* approximations. For example, tracing in the neighborhood of singularities needs algebraic considerations that go beyond numerical evaluation of points. Third, the automation of high-level geometric design processes—e.g. the conversion of parametric to implicit representations of geometric objects—needs *symbolic computation* involving parameters (variables, indeterminates). Algebra is the natural framework for this type of geometric computation.

In fact, (real and complex) algebraic geometry, by its very nature, should be the natural mathematical setting for geometric reasoning by algebraic techniques. However, strangely enough, mainstream algebraic geometry was totally nonconstructive in the last six decades, and most of the constructive techniques of the nineteenth-century algebraic geometry are available only in examples rather than in terms of general algorithms. For the immediate future, however, we envisage an evolutionary research impulse towards *algorithmic algebraic geometry*, turning the fascinating insights of nonconstructive algebraic geometry into algorithmic methods for geometric reasoning.

In this paper, we present algorithmic techniques from algebraic geometry that have evolved in the last few years off the beaten track of nonconstructive mainstream algebraic geometry. Three of these techniques (namely, the method of Characteristic Sets, the method of Gröbner Bases, and the method of Cylindrical Algebraic Decomposition) are universal—i.e. they provide algorithmic solutions to a whole class of (geometric) problems. These techniques are presented in the three main sections of the paper. Miscellaneous other algebraic techniques, which were devised recently for special geometric problems, are briefly summarized in the last section. The three universal techniques are ordered according to their first appearance in the literature. Incidentally, this order also corresponds to increasing generality.

The universal algebraic techniques reviewed in this paper are involved. We can therefore only sketch the ideas, give the main references, and point to some applications. We cannot provide details.

In the literature, the term "geometric reasoning" is used with many different connotations. Sometimes "geometric reasoning" is used in the restrictive sense of "geometric theorem proving" ("reasoning" = "proving"), sometimes it stands for "intelligent reasoning about geometric objects" (with some flavor of "artificial intelligence"), and sometimes it is more or less synonymous to "algorithmic problem solving for geometric objects." In this article, we adhere to the latter, most general view of "geometric reasoning." In this view, all types of algorithmic problems in mathematical models of geometric situations are in the scope of "geometric reasoning"—e.g. analysis of geometric objects, kinematics of robots, collision detection, determination of collision-free paths, fine motion planning, machining of specified geometric objects on specified machines, design of objects for performing specified functions, decision about the validity of certain classes of assertions about geometric situations ("geometric theorem proving"), transformation between different representations of geometric objects, etc. For a stimulating description of the emerging "science of geometric reasoning" see, for examle, Hopcroft & Krafft (1986).

THE METHOD OF CHARACTERISTIC SETS

Outline of the Method

Ritt (1950) introduced characteristic sets in the context of his work on differential algebra. Unfortunately, his work did not gain much attention until Wu (1984) revitalized this notion in his work on automated geometry theorem proving. Little literature exists on characteristic sets. Ritt's above-

mentioned book is hard to read and treats characteristic sets in the general context of differential polynomials. Wu's paper contains an introduction restricting the consideration to ordinary polynomials, as is sufficient for many interesting applications. In the sequel we exactly specify the notion "characteristic set of a given set of polynomials" and present the main algorithmic ideas required for their construction. First, we review some definitions.

Assuming that the indeterminates are linearly ordered according to their subscripts, $class(p)$ denotes the index of the highest indeterminate appearing in a polynomial p, and $initial(p)$ denotes the leading coefficient of p regarded as a (univariate) polynomial in its highest variable, say x_k, with coefficients in $K[x_1, \ldots, x_{k-1}]$, K a field.

Furthermore, a partial ordering $\prec$ on the polynomials is defined, such the $p \prec q$ (*p has lower rank than q*) iff either p is of lower class than q or p and q are of the same class, say k, and p has lower degree in x_k than q. In case neither p has lower rank than q nor q has lower rank than p, they are said to have the same rank, which is denoted by $p \sim q$.

A reduction relation is defined by the usual pseudo-division as follows: *q reduces to r* w.r.t. p iff r is the pseudo-remainder obtained from pseudo-dividing q by p, both regarded as polynomials in $x_{class(p)}$. [A detailed description of the pseudo-division algorithm can be found, for example, in Knuth (1969), pp. 368–69.] The remainder obtained by pseudo-dividing q by p will be denoted by $prem(q, p)$. Generally, q is said to be *reduced* w.r.t. p iff q has lower degree in $x_{class(p)}$ than p. Clearly, $prem(q, p)$ is reduced w.r.t. p.

We illustrate a reduction step by a short example: $q = x_2x_3 - x_2^2 + x_1$ has to be reduced w.r.t. $p = x_1x_2 + x_2 - x_1^2$. Since $class(p) = 2$, q and p have to be regarded as univariate polynomials in x_2—i.e. $q = -x_2^2 + x_3x_2 + x_1$, and $p = (x_1 + 1)x_2 - x_1^2$. Pseudo-dividing q by p requires multiplying q by an appropriate power s of $initial(p)$, and performing an ordinary division of $(initial(p))^s \cdot q$ by p. For our example this yields $(x_1 + 1)^2 \cdot q = (x_1x_3 + x_3 - x_2 - x_1^2) \cdot p + prem(q, p)$, where $prem(q, p) = x_1^3 x_3 + x_1^2 x_3 - x_1^4 + x_1^3 + 2x_1^2 + x_1$. [In contrast to the algorithm presented in Knuth's book, where the power of the initial, s is directly determined by the powers of the original polynomials and a priori enables ordinary division, for our purpose it suffices to choose the smallest s such that the ordinary division can be carried out. Algorithmically, this can be achieved by multiplying q by $initial(p)$ only "on demand" in the division algorithm.]

A finite set of polynomials $F = \{f_1, \ldots, f_k\}$ is called an *ascending set* iff $0 < class(f_1) < \cdots < class(f_k)$ and f_i is reduced w.r.t. f_i for each pair $j > i$ in case $k > 1$, and simply $f_1 \neq 0$ in case $k = 1$. The above partial ordering on polynomials is extended to ascending sets F, G by saying that $G = \{g_1, \ldots, g_m\}$ *has lower rank than* $F = \{f_1, \ldots, f_k\}$ $(G \prec F)$ iff there exists some $j \leq min(m, k)$, such that $g_j \prec f_j$, while $g_i \sim f_i$ for all $i < j$, or $m > k$ and $g_i \sim f_i$ for all $i \leq k$.

Reducing a polynomial q w.r.t. an ascending set $F = \{f_1, \ldots, f_k\}$ gives rise to the following representation of q:

$$(initial(f_1))^{s_1} \cdots (initial(f_k))^{s_k} \cdot q = h_1 \cdot f_1 + \cdots + h_{hk} \cdot f_k + prem(q, F)$$

where $prem(q, F)$ is given by $prem(\ldots prem(prem(q, f_k), f_{k-1}), \ldots f_1)$ and the h_i are polynomials. $Prem(q, F)$ is reduced w.r.t. f_i for all $i \leq k$ [we say, for short, $prem(q, F)$ is reduced w.r.t. F].

The kew property of ascending sets is that they are "triangularized"—i.e. each variable is "introduced" by exactly one polynomial. This special form is convenient for many purposes—for example, for computing the zeros of a set of polynomials. Therefore, the main interest lies in constructing, for a given set of polynomials F, an ascending set G, such that F and G have "almost" the same zeros. In detail, one requires that 1. G is an ascending set, 2. any zero of F is a zero of G, and 3. any zero of G that is not a zero of any of G's initials is a zero of F. Such a set G is called a *characteristic set* of F.

The following algorithm computes a *basic set* of a given set of polynomials F—i.e. a minimal ascending set, all of whoch elements belong to F.

```
BASIC_SET (in: F; out: B)
F' := F; B := ∅
do
   b := an element of F' with lowest rank
   F' := {f ∈ F' − {b} | f is reduced w.r.t b}
   B := B ∪ {b}
until F' = ∅
```

The basic set of F can be regarded as an approximation of a characteristic set of F, properties 1 and 2 of the above specification are clearly fulfilled. For 3 to be fulfilled, all elements of F have to be reducible to 0 w.r.t. G. Therefore, whenever one of these remainders is found to be nonzero, it is added to F.

```
COMPLETION (in: F, B; out: F')
F' := F
do for all f ∈ F
   r := prem(f, B)
if r ≠ 0 then F' := F' ∪ {r}
```

Clearly, the set of zeros is preserved by this algorithm. On the other hand, the basic set for the enlarged F will have lower rank than the previous

one and hence will allow more zero-reductions. In Ritt's algorithm for constructing characteristic sets, these two concepts of approximation and completion are applied iteratively:

```
CHARACTERISTIC_SET (in: F, out: G)
F1 := F; i := 1
do
  G := BASIC_SET (Fi)
  Fi+1 := COMPLETION (Fi, G)
  i := i+1
until Fi = Fi-1
```

It is important to note that the set of zeros might be slightly changed when computing a characteristic set for a given set F. Also, the final result depends on the ordering of the variables. We give a small example:

Consider $F = \{y^2-y, y^2-yx\}$. The set of zeros of F is shown in Figure 1 (*left*). For $x < y$, the above algorithm produces the characteristic set $G = \{(-x+1)y\}$, whose set of zeros is shown in Figure 1 (*right*). If the variables are ordered such that $y < x$, the resulting characteristic set is $G' = \{y^2-y, -yx+y\}$, whose zeros are identical with the zeros of F.

Applications to Geometric Reasoning

DECOMPOSITION OF ALGEBRAIC VARIETIES In the sequel, H will always denote a finite set of polynomials. The algebraic variety defined by H (i.e. the set of zeros of H), will be denoted by $|H|$. A set H is *decomposable* into $H_1, \ldots, H_t$, $t \geq 2$, iff $|H| = |H_1| \cup \cdots \cup |H_t|$. A decomposition is *uncontractible* iff none of the components can be omitted.

Given H, we want to find an uncontractible decomposition $H_1, \ldots, H_t$ of H such that none of the H_i can be decomposed any further. Ritt has given some methods for solving this problem. A description of his approach can also be found in Wu (1984) and Kutzler (1988).

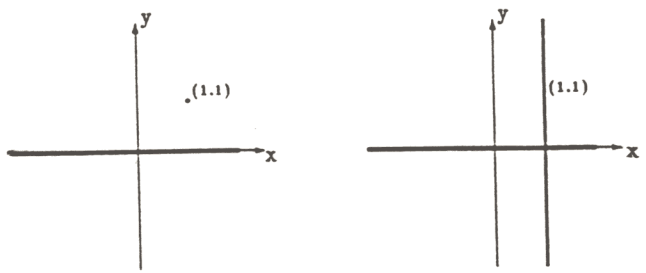

Figure 1 Set of zeros of $\{y^2-y, y^2-yx\}$ (*left*) and of $\{(-x+1)y\}$ (*right*).

Let an ascending set $F = \{f_1, \ldots, f_k\}$ be given and rename the indeterminates such that $y_i := x_{class(f_i)}$ for $1 \leq i \leq k$, and $u = u_1, \ldots, u_d$, $d = n-k$, are the remaining variables. Let $K_0 := K(u)$. F is called *irreducible*, iff $f_1^* := f_1$ cannot be factored in $K_0[y_1]$, and, for $2 \leq i \leq k$, $f_i^* := f_i[y_1 \leftarrow \eta_1, \ldots, y_{i-1} \leftarrow \eta_{i-1}]$ cannot be factored in $K_{i-1}[y_i]$, where η_{i-1} is a zero of f_{i-1}^* and $K_{i-1} := K_{i-2}(\eta_{i-1})$.

Let us now consider an irreducible ascending set F. The set of polynomials that have pseudo-remainder 0 w.r.t. F is an ideal and thus, by Hilbert's basis theorem, has a finite basis. Wu (1984) states that "there exists some mechanical procedure for computing such a finite basis" but that "the constructive proof is not a simple one." However, no reference to this procedure is given. Actually, using the Gröbner bases method (see next section), it is simple to compute such a basis (W. F. Schelter, personal communication). Let B_F denote such a finite basis that includes F itself. Clearly, F is a characteristic set of B_F and B_F is irreducible.

For an irreducible characteristic set $F = \{f_1, \ldots, f_k\}$ of H, B_F, $H \cup \{initial(f_1)\}, \ldots, H \cup \{initial(f_k)\}$ form a decomposition of H. Otherwise, there exists an index j and polynomials $g_1, \ldots, g_s$, such that $H \cup \{initial(f_1)\}, \ldots, H \cup \{initial(f_j)\}, H \cup \{g_1\}, \ldots, H \cup \{g_s\}$ form a decomposition of H. (Roughly, j indicates the polynomial in F, up to which the "partial" characteristic set is irreducible, and $g_1, \ldots, g_s$ are obtained from the prime factors of f_j^*.)

Hence a decomposition of H can be determined by computing a characteristic set of H, say F, and applying the above knowledge. (In case F contains only one polynomial of class 0—i.e. H is contradictory—the empty decomposition is returned.) Then the algorithm is recursively applied to the components $H \cup \{p\}$. (B_F is already irreducible by construction.) The final result is a decomposition into nondecomposable sets $B_{F_1}, \ldots, B_{F_t}$ with corresponding irreducible characteristic sets $F_1, \ldots, F_t$. By systematically applying a lemma which states that, for nondecomposable sets B_{F_i} and B_{F_j}, $B_{F_i} \subset B_{F_j}$ holds (i.e. B_{F_i} can be omitted in the decomposition) iff $prem(f, F_i) = 0$ for all $f \in B_{F_j}$, one can determine an uncontractible decomposition of H.

GEOMETRIC THEOREM PROVING The first mechanical proofs of geometric theorems were found by Gelernter (1959). His approach was axiomatic—i.e. he expressed the geometric theorems in a first-order geometric theory, which then was attacked by a resolution theorem prover. Only simple theorems can be proved by this method.

An alternative is to translate the geometric theorem into a corresponding algebraic theorem. The decision methods for real closed fields provide a general solution for this. But still, owing to the high complexity of these

methods, only rather simple theorems can be proved. More comments are given in the corresponding section.

Based on Ritt's work on characteristic sets, Wu (1978, 1984) developed a method for confirming geometric theorems of a special type. Roughly, his approach can be applied to geometric theorems whose algebraic translations are of the form

$$(\forall\alpha)\,(H(\alpha) = 0 \Rightarrow c(\alpha) = 0).$$

Here $H(\alpha) = 0$ stands for $h_1(\alpha) = 0 \wedge \cdots \wedge h_m(\alpha) = 0$, where the $h_i = 0$ are the polynomial equations corresponding to the hypotheses of the geometric theorem, and $c = 0$ corresponds to the conjecture. The α are vectors of (real) numbers.

We consider the following simple theorem: "Given a circle. Let AB and CD be two secants of equal length. Then the secants' midpoints have equal distance from the circle's center" (Figure 2). An algebraization is obtained by introducing variables for coordinates as shown in Figure 2 and expressing the hypotheses and the conjecture by the following polynomial equations:

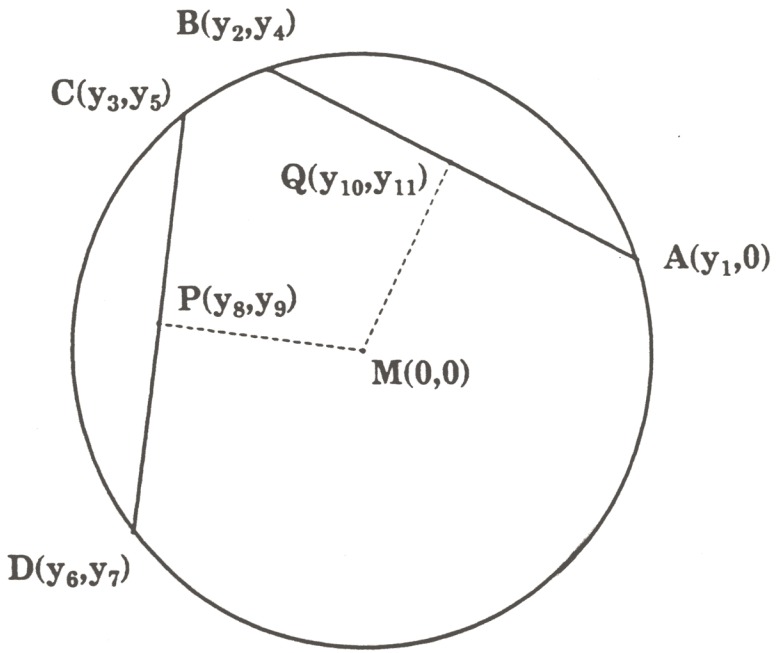

Figure 2 Example of a geometric theorem that can be confirmed by Wu's method.

$h_1 = y_2^2+y_4^2-y_1^2 = 0$ (B lies on the circle defined by M and A),

$h_2 = y_3^2+y_5^2-y_1^2 = 0$ (C lies on the circle defined by M and A),

$h_3 = y_6^2+y_7^2-y_1^2 = 0$ (D lies on the circle defined by M and A),

$h_4 = (y_3-y_6)^2+(y_5-y_7)^2-(y_2-y_1)^2-y_4^2 = 0$

(length of CD = length of AB),

$h_5 = (y_8-y_6)(y_5-y_7)-(y_9-y_7)(y_3-y_6) = 0$ (P lies on CD),

$h_6 = (y_8-y_6)^2+(y_9-y_7)^2-(y_3-y_8)^2-(y_5-y_9)^2 = 0$

(length of DP = length of PC),

$h_7 = (y_{10}-y_1)y_4-y_{11}(y_2-y_1) = 0$ (Q lies on AB),

$h_8 = (y_{10}-y_1)^2+y_{11}^2-(y_2-y_{10})^2-(y_4-y_{11})^2 = 0$

(length of AQ = length of QB),

$c = y_8^2+y_9^2-y_{10}^2-y_{11}^2 = 0$ (length of PM = length of QM).

In the above formulation, the conjecture polynomial does not vanish on *all* zeros of the hypotheses. The reason is that in case A and B happen to be identical, the above algebraization allows any point in the plane to be Q. Hence the theorem does not hold in this degenerate situation. One often has to exclude certain degenerate cases that can be described by one or more polynomial inequalities $d_1 \neq 0, \ldots, d_s \neq 0$, yielding

$$(\forall\alpha)\,(H(\alpha) = 0 \wedge d_1(\alpha) \neq 0 \wedge \cdots \wedge d_s(\alpha) \neq 0 \Rightarrow c(\alpha) = 0).$$

The traditional question in theorem proving is to decide the above formula for given $H, d_1, \ldots, d_s, c$. In contrast to that, Wu (1984) asked for finding "consistent" $d_1, \ldots, d_s$ for given H, c, such that the above formula holds. Since the nondegeneracy conditions heavily depend on the algebraization, this approach is more appropriate than the mere decision problem. However, there is an ongoing discussion whether Wu's formulation of the problem is adequate (see Chou & Yang 1987).

Normally, geometry is allied to real numbers. Hence all methods providing complete solutions for the above problem over algebraically closed fields (including characteristic sets and Gröbner bases) allow only confirmations of true geometric theorems.

Roughly, Wu's procedure for solving the problem of finding nondegeneracy conditions that have to be added to the hypotheses in order to be able to confirm the conjecture is based on the idea of decomposing, by Ritt's method, the variety defined by H into nondecomposable components $H_1, \ldots, H_r$. It can be shown that, for any H_i, the initials of a characteristic

set of H_i are suitable nondegeneracy conditions for c to be true on H_i, or no such nondegeneracy conditions exist. This can be decided by computing the pseudo-remainder of c w.r.t. a characteristic set of this component. In case the pseudo-remainder is 0, the conjecture holds on this component and the initials of the characteristic set are chosen as degeneracy polynomials. Otherwise the conjecture does not hold on this component and the polynomials that distinguish this component from the other components are chosen as degeneracy conditions. [These polynomials can be obtained easily by a tracing mechanism during the decomposition process. Details are given in Kutzler (1988).] If c holds on at least one of the components, all degeneracy polynomials collected give a solution for the above problem and hence the underlying geometric theorem is true subject to these nondegeneracy conditions. Otherwise no such nondegeneracy conditions exist.

This complete procedure for solving the above problem is not practical because the factorization over successive extension fields in Ritt's decomposition algorithm is extremely complex. All reported implementations are incomplete in one way or the other. Wu's implemented version uses some weaker notion of characteristic set (see Wu 1984) and does not involve decomposition. Chou (1985) developed a fast factorization algorithm that is only applicable for the quadratic case, however. Other implementations are Ko & Hussain (1985) and Kusche et al (1987).

Nevertheless, the implementations turn out to be a powerful practical tool for confirming true geometric theorems. The main reasons for this seem to be that, in many practical cases, the hypothesis polynomials are almost triangular (in case an "appropriate" ordering of the indeterminates is chosen) and that most theorems can be confirmed without involving factorization. One of the hardest theorems that has been successfully proved is Morley's Theorem—"For a triangle ABC the neighboring trisectors of the three angles of the triangle will intersect to form 27 triangles in all, of which 18 are equilateral"—as reported by Wu (1984). Chou (1986) provides the most extensive collection of geometric theorems that have been proved using Wu's method, and documents mechanical proofs of 360 theorems.

Applications of geometric theorem proving methods in computer vision are currently being investigated (see Swain & Mundy 1986)—e.g. for deducing facts about objects from properties observed in an image, and vice versa.

ALGEBRAIC SYSTEMS OF EQUATIONS Ritt's theory can also be used to compute the zeros of systems of algebraic equations, as has been reported by Wu (1986). No extensive investigation using existing implementations has been done so far, however. For examples of algebraic equations occurring in geometric reasoning, see the next section.

THE GRÖBNER BASES METHOD

Outline of the Method

The Gröbner bases method was introduced by Buchberger (1965, 1970). The method is applicable to a wide range of problems that involve finite sets F of multivariate polynomials. The core of the method is an algorithm (the Gröbner bases algorithm) for transforming an arbitrary set F of multivariate polynomials into a certain standard form G (the Gröbner basis form) that can be shown to possess many mathematically interesting and algorithmically useful properties. For solving algorithmic problems for input sets F, the Gröbner bases method suggests first transforming F into Gröbner basis form G and then solving the problem for input G, which is often easily possible using the fundamental properties of Gröbner bases. Buchberger (1985) provides the most comprehensive, easy-to-read tutorial on the Gröbner bases method, along with a complete bibliography. A more recent contribution (Buchberger 1988) contains a complete list of useful properties of Gröbner bases and provides more details for the examples of applications below.

We use the following notation: K denotes an arbitrary coefficient field; a, b, c, d are elements in coefficient fields; f, g, h, p, q are polynomials in $K[x_1, \ldots, x_n]$; s, t, u are power products—i.e. polynomials of the form $x_1^{i_1} \ldots x_n^{i_n}$; $C(f, u)$ is the coefficient at power product u in polynomial f; and F, G are finite sets of polynomials. We write $f \equiv_F g$ for "f is congruent to g modulo Ideal(F)," the ideal generated by F.

The basic notion of Gröbner bases theory is *polynomial reduction*. Roughly, f reduces to g modulo F iff g results from f by subtracting a suitable multiple $a \cdot u \cdot h$ of a polynomial $h \in F$ such that g is lower in an "admissible ordering" than f. The set of "admissible orderings" that can be used for this purpose can be characterized by two easy axioms. For all details, see Buchberger (1985). We need the following additional noration: $\succ$ (typed variable for admissible orderings); LP(f) (leading power product of f w.r.t. $\succ$); $f \rightarrow_F g$ (f reduces to g modulo F); $\rightarrow_F^*$ (reflexive-transitive closure of $\rightarrow_F$); $\leftrightarrow_F^*$ (reflexive-symmetric-transitive closure of $\rightarrow_F$); and f_F (f is in normal form modulo F—i.e. there does not exist any g such that $f \rightarrow_F g$). There exists an algorithm NF (the "normal form algorithm"), such that for all F, g: $g \rightarrow_F^* \mathrm{NF}(F, g)$, and $\underline{\mathrm{NF}(F, g)}_F$.

Note that, for fixed F, f, there may exist many different g such that $f \rightarrow_F^* g$ and $\underline{g}_F$—i.e. in general, "normal forms for polynomials f modulo F" are not unique. F is called a *Gröbner basis* (w.r.t. $\succ$) iff "normal forms modulo F are unique"—i.e. for all f, g_1, g_2: if $f \rightarrow_F^* g_1$, $f \rightarrow_F^* g_2$, $\underline{g_1}_F$, and $\underline{g_2}_F$, then $g_1 = g_2$. The main theorem of Gröbner bases theory gives an algorithmic characterization of Gröbner bases: F is a Gröbner basis iff,

for all $f, g \in F$, $\mathrm{NF}(F, \mathrm{SP}(f,g)) = 0$. Here $\mathrm{SP}(f,g)$ (the "*S-polynomial* of f and g") is defined by $\mathrm{SP}(f,g) := u \cdot f - v \cdot g$, where u, v are such that $u \cdot \mathrm{LP}(f) = v \cdot \mathrm{LP}(g) =$ least common multiple of $\mathrm{LP}(f)$ and $\mathrm{LP}(g)$. The proof of the main theorem is completely combinatorial and quite involved. Actually, the whole power of the Gröbner basis method is contained in this proof. The main theorem is the basis for the following algorithm GB (the Gröbner basis algorithm) that constructs, for a given set F, a Gröbner basis G such that $\mathrm{Ideal}(F) = \mathrm{Ideal}(G)$.

```
GB (in: F; out: G)
G := F
B := {{f, g} | f, g ∈ G, f ≠ g}
while B ≠ Ø do
  {f, g} := a pair in B
  B := B − {{f, g}}
  h := NF(G, SP(f, g))
  if h ≠ 0 then (B := B ∪ {{g, h} | g ∈ G}; G := G ∪ {h})
```

This algorithm is structurally simple. However, it is complex in terms of time and space consumed. Various theoretical and practical improvements of the rough form of the algorithm have enhanced the scope of applicability (see Buchberger 1985). The Gröbner bases algorithm GB is available in almost all major computer algebra systems, notably in the SAC2, SCRATCHPAD II, REDUCE, MAPLE, MACSYMA, and muMATH systems. The introduction of the Buchberger et al text (1982) contains the addresses of institutions from which these systems can be obtained. The implementations vary drastically in their efficiency, mostly because of the varying amount that has been taken into account. Also, computation time and space depend drastically on the admissible orderings used, on permutations of variables, on treating indeterminates as ring or field variables, on strategies for selecting pairs in the consideration of S-polynomials, and on many other factors. Thus, if one seriously considers solving problems of the type described in this paper one should try various systems and various orderings, strategies, etc.

Below we summarize those properties of Gröbner bases that are used in the subsequent applications. Most of these properties were already proven by Buchberger (1965, 1970). The property Elimination Ideals is due to Trinks (1978). Gröbner bases have many other interesting properties (see Buchberger 1988).

Ideal Membership
For all F, f: $f \in \mathrm{Ideal}(F)$ iff $\mathrm{NF}(\mathrm{GB}(F), f) = 0$.

Canonical Simplification
For all F, f, g: $f \equiv_F g$ iff $\mathrm{NF}(\mathrm{GB}(F), f) = \mathrm{NF}(\mathrm{GB}(F), g)$.

Radical Membership
For all F, f: f vanishes on all common zeros of F iff $1 \in \mathrm{GB}(F \cup \{y \cdot f - 1\})$, (where y is a new indeterminate).

Solvability of Polynomial Equations
For all F: F is solvable iff $1 \notin \mathrm{GB}(F)$.

Finite Solvability of Polynomial Equations
For all F: F has only finitely many solutions iff for all $1 \le i \le n$ there exists an $f \in \mathrm{GB}(F)$ such that $\mathrm{LP}(f)$ is a power of x_i.

Elimination Ideals, Solution of Polynomial Equations
Let $\succ$ be the lexical ordering defined by $x_1 \prec x_2 \prec \cdots \prec x_n$.
Then, for all F, $1 \le i \le n$: $\mathrm{GB}(F) \cap K[x_1, \ldots, x_i]$ is a Gröbner basis for the "i-th elimination ideal" generated by F—i.e. for $\mathrm{Ideal}(F) \cap K[x_1, \ldots, x_i]$.

Ideal Intersection
Let $\succ$ be the lexical ordering defined by $x_1 \prec x_2 \prec \cdots \prec x_n \prec y$, where y is a new variable. Then, for all F, G:
$\mathrm{GB}(\{y \cdot f \mid f \in F\} \cup \{(y-1) \cdot g \mid g \in G\}) \cap K[x_1, \ldots, x_n]$
is a Gröbner basis for $\mathrm{Ideal}(F) \cap \mathrm{Ideal}(G)$.

Applications to Geometric Reasoning

ALGEBRAIC SYSTEMS OF EQUATIONS Algebraic equations are ubiquitous in geometric reasoning. The Gröbner bases method yields a complete solution for this problem even in cases where the coefficients of the equations involve symbolic parameters. We give two examples: inverse robot kinematics and intersection of superellipsoids.

Inverse robot kinematics The problem of inverse robot kinematics is the problem of determining, for a given robot, the distances at the prismatic joints and the angles at the revolute joints that will result in a given position and orientation of the end-effector. Let us consider, for example, a robot having two revolute joints (two "degrees of freedom"). We introduce the following variables: l_1, l_2 for the lengths of the two robot arms; px, py, pz for the x-, y-, and z-coordinate of the position of the end-effector; ϕ, θ, ψ for the Euler angles of the orientation of the end effector; and δ_1, δ_2 for the angles describing rotation at the revolute joints. We introduce the sines and cosines of the angles occurring in the above description as separate

variables: s_1, c_1, s_2, c_2 for the sine and cosine of δ_1 and δ_2, and *sf*, *cf*, *st*, *ct*, *sp*, *cp* for the sine and cosine of ϕ, θ, and ψ. The interrelation of the physical entities described by the above variables is expressed in a system of equations consisting of 17 polynomials of maximal degree 3 in the above 15 variables (see Paul 1981). We apply the Gröbner bases method to the most general version of the problem where the geometrical variables l_1, l_2 and the position variables *px*, *pz* are considered as symbolic parameters.

For solving this problem we use the property Elimination Ideals of Gröbner bases. This property, read as an algorithm, tells us that we first have to compute the Gröbner bases of the set F of input polynomials. Since l_1, l_2, *px*, *pz* are to be treated as symbolic parameters, we work over $\mathbb{Q}(l_1, l_2, px, pz)[c_1, \ldots, sp]$. The resulting Gröbner basis has the following form:

$$c_1^2 + \frac{px^2}{pz^2 - 2\cdot l_1\cdot pz - l_2^2 + l_1^2} = 0,$$

$$c_2 + \frac{pz^2 - 2\cdot l_1\cdot pz - l_2^2 + l_1^2}{l_2}\cdot px\cdot c_1 = 0,$$

$$s_1^2 - \frac{pz^2 - 2\cdot l_1\cdot pz + px^2 - l_2^2 + l_1^2}{pz^2 - 2\cdot l_1\cdot pz - l_2^2 + l_1^2} = 0,$$

$$s_2 - \frac{pz - l_1}{l_2} = 0,$$

$$py + \frac{pz^2 - 2\cdot l_1\cdot pz - l_2^2 + l_1^2}{px}\cdot c_1\cdot s_1 = 0,$$

$$cf^2 - \frac{pz^2 - 2\cdot l_1\cdot pz + px^2 - l_2^2 + l_1^2}{pz^2 - 2\cdot l_1\cdot pz - l_2^2 + l_1^2} = 0,$$

$$ct = 0,$$

$$cp + \frac{pz^3 - 3\cdot l_1\cdot pz^2 - l_2^2\cdot pz + 3\cdot l_1^2\cdot pz + l_1\cdot l_2^2 - l_1^3}{l_2\cdot pz^2 - 2\cdot l_1\cdot l_2\cdot pz + l_2\cdot px^2 - l_2^3 + l_1^2\cdot l_2}\cdot s_1\cdot cf = 0,$$

$$sf + \frac{pz^2 - 2\cdot l_1\cdot pz - l_2^2 + l_1^2}{pz^2 - 2\cdot l_1\cdot pz + px^2 - l_2^2 + l_1^2}\cdot c_1\cdot s_1\cdot cf = 0,$$

$$st + \frac{pz^2 - 2\cdot l_1\cdot pz - l_2^2 + l_1^2}{pz^2 - 2\cdot l_1\cdot pz + px^2 - l_2^2 + l_1^2}\cdot s_1\cdot cf = 0,$$

$$sp + \frac{pz^4 - 4\cdot l_1\cdot pz^3 - 2\cdot l_2^2\cdot pz^2 + 6\cdot l_1^2\cdot pz^2 + 4\cdot l_1\cdot l_2^2\cdot pz - 4\cdot l_1^3\cdot pz + l_2^4 - 2\cdot l_1^2\cdot l_2^2 + l_1^4}{l_2\cdot px\cdot pz^2 - 2\cdot l_1\cdot l_2\cdot px\cdot pz + l_2\cdot px^3 - l_2^3\cdot px + l_1^2\cdot l_2\cdot px}\cdot c_1\cdot s_1\cdot cf = 0.$$

The above Gröbner basis has a remarkable structure:

1. The geometrical parameters l_1 and l_2 and the position parameters *px* and *pz* are still available as symbolic parameters in the polynomials of the Gröbner basis. Thus, the system is still "general." The Gröbner basis is in "closed form."
2. In accordance with the property Elimination Ideals, the system is "triangularized." In this example, this means that the first polynomial of the basis depends only on c_1, the second on c_1, c_2, the third on $c_1, c_2, s_1, \ldots$. After substitution of numerical values for the parameters l_1, l_2, px, pz, we can therefore numerically determine the possible values for c_1 from the first equation; then, for each of the values of c_1, determine the value of c_2 from the second equation; then, for each of the values of c_1, c_2, determine the value of s_1 from the third equation, etc.
3. The above method of numerical backward substitution based on the Gröbner basis, by the property Elimination Ideals, is guaranteed to yield *all* (real and complex) solutions of the system.
4. Again by Elimination Ideals, *no "extraneous"* solutions of the system are produced. (Other algebraic methods—e.g. the resultant method—may produce extraneous solutions.)

The above Gröbner basis was produced in 62 sec on an IBM 4341 using an implementation of the Gröbner bases method by R. Gebauer and H. Kredel in the SAC-2 computer algebra system. The computation time increases drastically when more complicated robot types are investigated. We are far from being able to treat the most general robot of six degrees of freedom. However, little research effort has been dedicated so far to this application of Gröbner bases. Using the special structure of the problem it may well be that more theoretical results can be derived that allow us to speed up drastically the general algorithm in this particular application.

Intersection of superellipsoids Superellipsoids (Barr 1981) are surfaces in 3-D space that have an implicit representation as the set of points (x, y, z) such that

$$\left(\left(\frac{x}{a}\right)^{2/\varepsilon_3} + \left(\frac{y}{b}\right)^{2/\varepsilon_2}\right)^{\varepsilon_3/\varepsilon_1} + \left(\frac{z}{c}\right)^{2/\varepsilon_1} - 1 = 0.$$

Superellipsoids are topologically equivalent to spheres. The exponents ε_1, $\varepsilon_2, \varepsilon_3$ open an enormous flexibility for adjusting the shape of superellipsoids in order to approximate real objects. Recently, superellipsoids have been

proposed for approximating parts of robots and obstacles in order to test for collision. The collision-detection problem of robots is thereby reduced to an intersection test for superellipsoids.

If ε is of the form $1/k$ (which is sufficiently general for practical purposes), this problem leads to an algebraic system of four polynomials in $\mathbb{Q}(a, b, c, A, B, C)[x, y, z, \lambda]$ of degree $2k$. For computing the Gröbner bases, we use the lexical ordering defined by $x \prec y \prec z \prec \lambda$. For $\varepsilon = 1$ (which is, actually, the ellipsoid case) we get the Gröbner basis

$$x^6 - p(x) = 0,\ y - q(x) = 0,\ z - r(x) = 0,\ \lambda - s(x) = 0.$$

Here, $p(x)$, $q(x)$, $r(x)$, $s(x)$ are univariate polynomials in x of degree 5 with coefficients that are rational expressions in the parameters a, b, c, A, B, C. The equation for λ is not interesting for the problem at hand and may be dropped. The printout of these rational expressions consumes approximately two pages. (Some simplification by extracting common subexpressions would be possible.) Again, the Gröbner basis has all the advantageous features described in the inverse kinematics application. Note in particuar that, in this Gröbner basis, the second, third, and fourth equations are linear in the variables y, z, λ, respectively. Therefore the Gröbner basis presents an explicit symbolic solution to the problem as soon as the solution value for x is numerically determined from the first equation, which is univariate in x.

The problem with this approach is, again, computation time. While the Gröbner basis computation for $\varepsilon = 1$ needs 15 minutes (on an IBM 4341 in the SAC-2 implementation of the Gröbner bases method), the computation already needs 19 hours for $\varepsilon = 1/2$. At the moment, this excludes practical applicability of the method. However, one should take into account that the source of complexity seems to be the extraneous extremal solutions that enter through the Lagrange factor method.

On a more general level, recently, Sakai & Aiba (1987) have built the Gröbner bases method into PROLOG as a general nonlinear constraint solver and showed applications to geometric theorem proving (see also below).

IMPLICITIZATION OF PARAMETRIC OBJECTS As has been pointed out repeatedly, the automatic transition between implicit and parametric representation of curves and surfaces is of fundamental importance in geometric modeling (see, for example, Sederberg et al 1984). Arnon & Sederberg (1984) show how Gröbner bases can be used for the general implicitization problem of $(n-1)$-dimensional hypersurfaces. Buchberger (1988), using the same approach, solves the following

General Implicitization Problem
Given: $p_1, \ldots, p_m \in K[x_1, \ldots, x_n]$.
Find: $f_1, \ldots, f_k \in K[y_1, \ldots, y_m]$,
such that for all $a_1, \ldots, a_m$:
$f_1(a_1, \ldots, a_m) = \cdots = f_k(a_1, \ldots, a_m) = 0$ iff
$a_1 = p_1(b_1, \ldots, b_n), \ldots, a_m = p_m(b_1, \ldots, b_n)$ for some $b_1, \ldots, b_n$.

Implicitization Algorithm
$\{f_1, \ldots, f_k\} := \mathrm{GB}(\{y_1 - p_1, \ldots, y_m - p_m\}) \cap K[y_1, \ldots, y_m]$,
where GB has to be computed using the lexical ordering determined by
$y_1 \prec \cdots \prec y_m \prec x_1 \prec \cdots \prec x_n$.

Let us consider, for example, the 3-D surface defined by the following parametric representation: $x = r \cdot t$, $y = r \cdot t^2$, $z = r^2$. Roughly, this surface has the shape of a ship hull whose keel is the y-axis and whose bow is the z-axis. Applying algorithm GB to $\{x - r \cdot t, y - r \cdot t^2, z - r^2\}$ with respect to the ordering $z \prec y \prec x \prec t \prec r$ yields the Gröbner basis $G = \{x^4 - y^2 \cdot z,\ t \cdot x - y,\ t \cdot y \cdot z - x^3,\ t^2 \cdot z - x^2,\ r \cdot y - x^2,\ r \cdot x - t \cdot z,\ r \cdot t - x,\ r^2 - z\}$. The polynomial depending only on x, y, z is an implicit equation for the surface. This example was computed in 4 sec on an IBM AT in Buchberger's own research implementation of the Gröbner bases method in the muMATH system. Other examples with more complicated coefficients and similar degree characteristics had computing times in the range of several seconds. We conjecture that the examples occurring in practice should be tractible by the method.

In fact, the above solution to the General Implicitization Problem provides also a solution to the

Inversion Problem for Parametric Representations
Given: $p_1, \ldots, p_m \in K[x_1, \ldots, x_n]$ and
a point $(a_1, \ldots, a_m)$ on the hypersurface
parametrically defined by $p_1, \ldots, p_m$.
Find: $\{(b_1, \ldots, b_n) | a_1 = p_1(b_1, \ldots, b_n), \ldots, a_m = p_m(b_1, \ldots, b_n)\}$.

For details see Buchberger (1988).

DETECTION OF SINGULARITIES In tracing implicitly given planar curves, numerical methods work well except when tracing curves through singular points (see Hoffmann 1987). C. Hoffmann (personal communication) has pointed out that Gröbner bases yield an immediate approach to detect all singular points of implicitly given planar curves. The singular points of a planar curve given by $f(x, y) = 0$ are exactly the points (a, b) that are

common zeros of f, f_x, and f_y. Hence, the problem of determining the set S of singular points of a planar curve f can be treated by the following algorithm:

$G :=$ GB($\{f, f_x, f_y\}$), where f_x, f_y are the partial derivatives of f w.r.t. x and y, respectively and GB has to be computed w.r.t. a lexical ordering of x, y.

$S :=$ set of common zeros of G determined by the successive substitution method.

Let us consider the planar curve given in Figure 3.

This curve has nine singular points. We detect them by applying GB to $\{f, f_x, f_y\}$, where $f := (x^2+y^2-1)\cdot((x-1)^2+y^2-1)\ ((x+1)^2+y^2-1)\ (x^2(y-1)^2-1)$. Application of GB, using the lexical ordering determined by $x \succ y$, yields $\{y^5\cdot p(y), x\cdot y\cdot p(y), x^2-y^4\cdot q(y)\}$, where $p(y) := y^4-\frac{3}{2}y^3-\frac{1}{4}y^2-\frac{9}{8}y-\frac{3}{8}$, and $q(y) := \frac{2558}{27}y^4-\frac{823}{9}y^3-\frac{38955}{54}y^2+\frac{823}{12}y+\frac{5}{4}$. One sees that, for any solution y of the first polynomial in the Gröbner basis, the second polynomial vanishes identically whereas the third equation yields at most two different values for x. Proceeding by the general substitution method for Gröbner bases, we obtain the following singular points: $(-1,1)$, $(1,1)$, $(-1/2,\sqrt{3}/2)$, $(1/2,\sqrt{3}/2)$, $(-\sqrt{3}/2,1/2)$, $(\sqrt{3}/2,1/2)$, $(0,0)$, $(-1/2,-\sqrt{3}/2)$, $(1/2,-\sqrt{3}/2)$. In accordance with the picture, we obtained five different values for y and, altogether, nine singular points. The computation took 78 sec in Buchberger's muMATH Gröbner bases package on an Apollo workstation emulation of an IBM AT.

GEOMETRIC THEOREM PROVING The first applications of the Gröbner bases method for automatically proving geometric theorems were given independently by Kutzler & Stifter (1986a) and Kapur (1986).

Kapur presented a complete solution for the finding problem posed by Wu. The main idea of his approach is a refutational one. Roughly, it is tested whether the conjecture lies in the radical defined by the hypotheses.

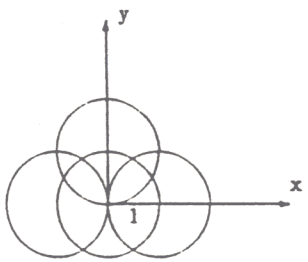

Figure 3 Planar curve with nine singular points.

This can be done by using the property Radical Membership of Gröbner bases. In case this is true, the theorem holds without any nondegeneracy condition. Otherwise, all elements of the resulting Gröbner basis not containing the new variable are candidates for the nondegeneracy condition. (It is easy to see that it always suffices to restrict the consideration to a single nondegeneracy condition.) If all of these finitely many candidates are contradictory to the hypotheses, then no "consistent" nondegeneracy condition exists. The complete method has been implemented by Kapur himself and Kusche et al (1987). Still, only confirmations of valid geometric theorems can be obtained even by this complete method, because the method treats variables as ranging over algebraically closed fields.

Kutzler & Stifter (1986a) gave a solution for a slightly different algebraic problem, which puts more restrictions on the nondegeneracy conditions to be found. Roughly, they compute the Gröbner basis of the hypothesis polynomials over some rational function field (with the "independent" variables adjoined as field variables) and compute the normal form of the conjecture w.r.t. this basis—i.e. perform an ideal membership test. In case the remainder is zero, the geometric theorem is confirmed, subject to certain denominators chosen as nondegeneracy conditions. (Independent variables describe points that can be chosen arbitrarily; dependent variables describe points that are constructed subject to conditions. This notion of independence coincides with the notion of algebraic independence in algebraic geometry and can be verified using Gröbner bases.) Kutzler & Stifter also gave an equivalent alternative, where the Gröbner basis is computed over the rationals (rather than over a rational function field), and "pseudo-reduction" is performed instead of (ordinary) reduction for the normal form computation. Their approach is powerful enough to confirm most theorems that can be confirmed by Kapur's method, but it is much faster. Implementations have been made by Kutzler & Stifter (1986a) and Kusche et al (1987).

The first method of Kutzler & Stifter was later also proposed by Chou & Schelter (1986). An approach similar to Kapur's is given by Carra & Gallo (1987). Another complete solution for Wu's problem, based on Gröbner bases, is given by Winkler (1987). Winkler's method uses syzygies and aims at computing a "simplest" nondegeneracy condition. Like Wu's method, Winkler's approach involves factorization over successive extension fields, and its practicality has not yet been tested.

PRIMARY DECOMPOSITION A polynomial ideal is "decomposable" iff it can be represented as the nontrivial intersection of two other polynomial ideals. Geometrically, this corresponds to a representation of the algebraic manifold (set of zeros) of the ideal as the nontrivial union of two algebraic

manifolds. It is well known in polynomial ideal theory that every polynomial ideal can be decomposed into finitely many ideals that can not be decomposed further ("irreducible components"), whose "multiplicities" are described by corresponding "primary ideals," and that this decomposition is essentially unique. This is the content of the famous Lasker-Noether decomposition theorem (see, for example, van der Waerden 1953). However, the proof of this theorem is nonconstructive—i.e. no general algorithmic method is provided that would find, for a polynomial ideal given by a finite basis F, the finite bases for its irreducible components and corresponding primary ideals.

Recently the problem of algorithmic primary decomposition has been completely solved using Gröbner bases. Still, the algorithm for the most general case is not yet implemented in a software system. Complete implementations may be expected in the very near future. A number of papers, of different generality and level of detail, contributed to the recent progress in this area: Kandri-Rody (1984), Lazard (1985), Gianni et al (1985), and Kredel (1987).

An exact formulation of the problem and a detailed description of the algorithms, which are quite involved, are beyond the scope of this paper. It is clear that automatic decomposition of algebraic manifolds (e.g. intersection curves of 3-D objects) should be of utmost importance for geometrical modeling where the global analysis of finitely represented objects, as opposed to a mere local numerical evaluation, is more and more desirable in advanced applications. All the algorithms invented for the solution of the primary decomposition problem heavily rely on the basic properties of Gröbner bases, notably on the properties Elimination Ideals, Ideal Membership, and properties derived from these properties as, for example, Intersection Ideal.

THE CYLINDRICAL ALGEBRAIC DECOMPOSITION METHOD

Introduction to the Method

Cylindrical algebraic decomposition (hereafter abbreviated CAD) was devised by Collins in 1974 as the key part of a new method for quantifier elimination for the elementary theory of real closed fields; the original and primary reference on this method is Collins (1975).

The ordered field of the real numbers is the most familiar example of a real closed field; see Chapter 9 of van der Waedren's text (1953) for a definition and alternative characterizations of a real closed field. The language of the elementary theory of real closed fields is constructed from constant symbols 0, 1, and -1; variables $x, y, z, \ldots$; operation symbols

$+$ and $\cdot$; relation symbols $=$ and $>$; and the usual Boolean connectives and quantifiers. The true sentences of the theory are those which are true whenever the constant symbols, operation symbols, and relation symbols are interpreted in the usual manner in any real field. Tarski (1951) presented a decision method for this theory—that is, an algorithm for deciding whether any sentence of the theory is true. He showed, moreover, that a sentence is true if and only if it is true in the field of real numbers. Still more important, he gave a quantifier elimination algorithm for the theory—that is, an algorithm that, given as input an arbitrary formula of the theory, produces as output an equivalent quantifier-free formula with the same free variables. If the input formula is a sentence then there are no free variables and one obtains a decision method as a corollary.

If, for example, the input formula is

$$(\exists x)(ax^2+bx+c=0)$$

then an equivalent output formula might be

$$(a \neq 0 \wedge b^2-4ac \geq 0) \vee (a=0 \wedge b \neq 0) \vee (a=0 \wedge b=0 \wedge c=0).$$

Many important mathematical problems can be expressed as formulas of this theory, and application of a quantifier elimination algorithm can then produce useful solutions. As just one nontrivial example, consider the problem of determining the best approximation of a polynomial of degree n by a polynomial of degree $n-2$ or less, relative to the max norm on $[-1, +1]$. Using variables as generic coefficients or parameters, it is easy, for any fixed n, to write a formula which states that $B(x)$, of degree $n-2$ or less, is the best approximation to $A(x)$, of degree n. Applying quantifier elimination to this formula will produce a quantifier-free formula in just the coefficients of A and B. The coefficients of B are algebraic functions of the coefficients of A, which will be given, in a strongly constructive sense, by this resulting formula.

Tarski's method is based on a generalized version of Sturm's theorem together with several ingenious devices. It proceeds by eliminating one innermost quantifier at a time. It is, however, quite impractical since it requires exponential maximum computing time even for formulas containing only a single quantified variable and no free variables.

Outline of the Method

The CAD method begins by putting the given formula into prenex form, so that all Boolean connectives are applied before the quantifiers. In fact, the input formula is written in the form

$$(Q_k x_k)(Q_{k+1} x_{k+1}) \cdots (Q_r x_r) F(x_1, \ldots, x_r)$$

where the $(Q_i x_i)$ are existential and universal quantifiers and F is a standard quantifier-free formula in the variables $x_1, \ldots, x_r$. This means that F is composed from standard atomic formulas, which are atomic formulas of the forms $A = 0$ and $A > 0$, where A is any polynomial with integral coefficients in the variables $x_1, \ldots, x_r$. Moreover, these polynomials are considered as polynomials in x_r with coefficients that are polynomials in $x_1, \ldots, x_{r-1}$; x_r is called the main variable. Recursively, the coefficients have x_{r-1} as main variable, and so on.

The next step of the CAD quantifier elimination method is to extract the set A of all of the different polynomials that occur in these standard atomic formulas.

Next a subalgorithm is applied that computes a CAD for the set A. A CAD for a set A of polynomials in r variables is defined, and computed, by induction on r. For $r = 1$ a CAD for A is a decomposition of the real line into a finite number of connected sets, called cells, in each of which each polynomial in A is sign-invariant. In this case we obtain a CAD for A by computing all of the real zeros of all the polynomials in A. These zeros, as one-point sets, and the open intervals, finite and infinite, between and outside these zeros are the cells of the CAD. The one-point sets are called sections of the CAD and the open intervals are called sectors. For $r > 1$, a "projection" of A, $A' = \text{proj}(A)$, is computed. A projection of A is a finite set, A', of polynomials in $x_1, \ldots, x_{r-1}$ such that from a CAD, D', for A' one can construct cylinders on the cells of D' such that each cylinder can be partitioned into a finite number of cells in each of which each polynomial in A is sign-invariant.

Figure 4 illustrates a CAD of $\mathbb{R}^2$ for a set of two bivariate polynomials, $A_1(x, y) = x^2 + y^2 - 3$, a circle, and $A_2(x, y) = xy - 1$, a hyperbola. In this example the projection consists of the three polynomials $A'_1(x) = 4(x^2 - 3)$, the discriminant of A_1, $A'_2(x) = x$, the leading coefficient of A_2, and $A'_3(x, y) = x^4 - 3x^2 + 1$, the resultant of A_1 and A_2. The seven real roots of these polynomials, 0, ± 0.62, ± 1.62 and ± 1.73, approximately, together with the eight open intervals between and beyond them, constitute a CAD, D', of $\mathbb{R}$ for the projection. The sections of the CAD, D, of $\mathbb{R}^2$ in each cylinder are the intersections of the graphs of A_1 and A_2 with the cylinder, and the sectors are the regions above, below, and between the sections. The total number of cells in D is 83. We give each cell of a CAD a tuple of indexes, numbering cylinders from left to right and numbering the cells in each cylinder from bottom to top. Thus the index tuples in the example are (1, 1), (1, 2), (1, 3), (2, 1), ..., (2, 5), (3, 1), ..., (3, 7), (4, 1), etc.

The minimal information needed about any CAD is the set of all index tuples and, with each index tuple, the sign of each polynomial in the associated cell, in the form of a sign tuple. For some applications, including

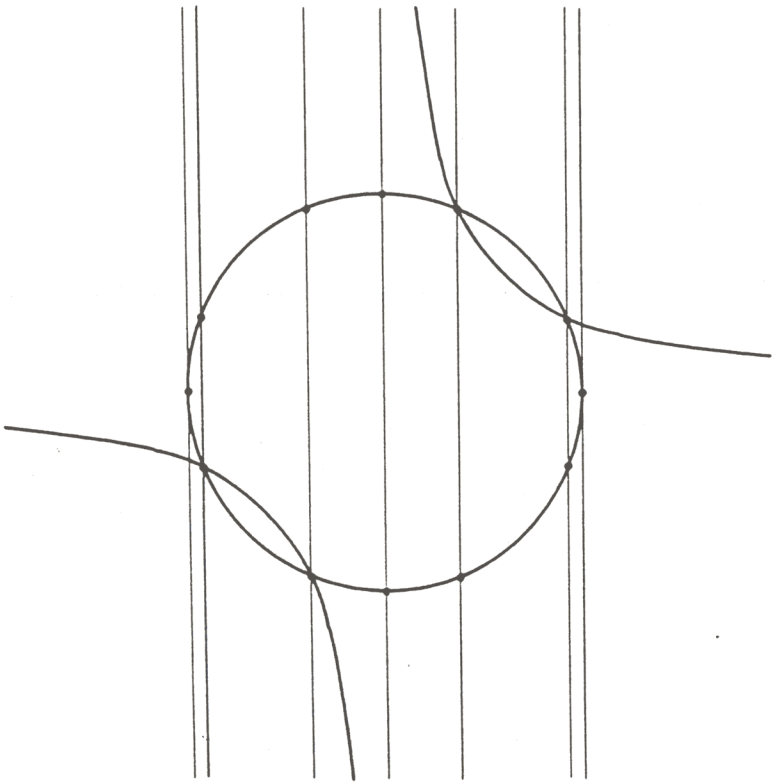

Figure 4 Cylindrical algebraic decomposition of $\mathbb{R}^2$ for $\{x^2 + y^2 - 3, xy - 1\}$.

elimination of quantifiers, this is the only information needed. However, in order to obtain the sign tuples we need to have sample points. A sample point for a cell is, as the name suggests, any point belonging to the cell.

As sample points for the sectors of our CAD of $\mathbb{R}$ we can choose rational numbers that are the endpoints of computed isolating intervals for the real roots of the projection. The sections are one-point sets and their sample points are their unique elements, generally irrational algebraic numbers. If we substitute the sample point of any cell of the CAD of $\mathbb{R}$ into each of the polynomials A_1 and A_2 we obtain two univariate polynomials. The real roots of these polynomials serve as the y-coordinates of sample points for the sectors in the cylinder based on our chosen cell of $\mathbb{R}$. Since these univariate polynomials have real algebraic numbers as coefficients, we must employ algorithms for performing exact arithmetic on real algebraic

numbers, and for determining the sign of any real algebraic number. In fact, one also needs an algorithm for computing the algebraic norm of a polynomial with algebraic number coefficients, and an algorithm for converting multiple algebraic extensions of the rational number field to simple extensions. See Loos (1982) for a description of the algebraic number algorithms used in the implementation of the algorithm in the SAC-2 computer algebra system (Collins & Loos 1980).

Now suppose that we have computed a CAD of $\mathbb{R}^2$ for $A = \{A_1, A_2\}$ in the above sense. How can we apply the information obtained to a quantifier elimination problem from which A_1 and A_2 might have been extracted? Suppose, for example, that our input formula was $(\exists y)(A_1(x,y) < 0 \wedge A_2(x,y) > 0)$. It is easy to see that any point x satisfies this formula if and only if all points in the same cell of the CAD of $\mathbb{R}$ do. Thus a cell of $\mathbb{R}$ satisfies the quantified formula if and only if there is some cell of the CAD of $\mathbb{R}^2$ that satisfies the formula $A_1(x,y) < 0 \wedge A_2(x,y) > 0$. But these cells are easily determined from the sign tuples of the CAD of $\mathbb{R}^2$. In our example these are just the two cells with indexes 5 and 11. Therefore the output of our quantifier elimination algorithm must be quantifier-free formula in x which is satisfied by just the points in these two cells. One such formula we can discover is simply $A'_3(x,y) < 0$.

There is an algorithmic method, given by Collins (1975), that produces a disjunctive formula, each disjunct being a defining formula for one cell. In the example, the disjunct produced for the first cell would be something like $-5/3 < x \wedge x > -1/2 \wedge A'_3(x) < 0$. Here $-5/3$ and $-1/2$ are endpoints of isolating intervals for roots of A'_3. For formulas in more than one variable, this method requires the use of a device called augmented projection, which involves the addition of some derivatives to the set of polynomials being projected. The augmented projection set properly includes the projection set, and therefore augmented projection produces a CAD that is a refinement of the ordinary CAD.

The essential property of a projection set, that it enables one to "lift" a CAD for the projection to a CAD of the next higher dimension space for the projected set, depends for its fulfillment only on the requirements that each cell be connected and be sign-invariant with respect to each polynomial in the projection. Thus, the cells being lifted need not themselves be cylindrical.

In 1979, Dennis Arnon made the very important observation that, as a result of this, the cells of a CAD for the projection could, in principle, be combined, or "clustered," before lifting, into larger cells that were still connected and sign-invariant. In effect this meant that all cells in a cluster could be lifted simultaneously, using a single sample point for the entire cluster. This is especially important since the real algebraic number arithmetic required for the lifting is time consuming, and the sample point for a cluster can often be chosen so as to have fewer irrational coordinates, or to have coordinates of lower algebraic degree. Notice that in our example there are 21 maximal clusters: 7 of dimension 2, 10 of dimension 1, and 4 of dimension 0.

The application of Arnon's insight necessitated the invention of an algorithm for deciding whether two cells of a CAD are adjacent, in the sense that their union is a connected set. There is indeed a decision method for this problem since it can itself be formulated in the elementary theory of real closed fields. Two cells are adjacent if and only if for every positive distance ε there exist points in the two cells which are less than ε apart. But this is impractical as an aid to producing a CAD since it involves the computation of another CAD of higher dimension.

Collins and Scott McCallum then collaborated with Arnon in devising adjacency algorithms for two- and three-dimensional spaces. These algorithms first appeared in Arnon (1981) and were subsequently published in Arnon et al (1984, 1988). The method for three-space consists essentially in projecting curves in three-space onto a coordinate plane using resultant calculation, followed by application of the two-space algorithm along with resolution of ambiguities resulting from the projection. One may hope that this inductive method can be extended to any dimension, but as of now this question remains unresolved. Prill (1986) has proposed an entirely different method for adjacency decision, but his method has not been implemented and so its practicality remains uncertain.

We have not discussed the projection operator, although we have observed that in our example it sufficed to include in the projection set the leading coefficients of the input polynomials, their discriminants, and the resultants of each pair of input polynomials. It was known to Collins as early as 1971 that these polynomials always sufficed for a projection of a set of bivariate polynomials, and he unsuccessfully attempted to extend this to more than two variables. Failing this he devised a projection method (Collins 1975) that necessitates the inclusion also of subresultants (which are the determinants of certain submatrixes of the Sylvester matrix, the determinant of which is the resultant). McCallum (1984), using ideas derived from some work of Zariski, succeeded in proving the sufficiency of a projection operator involving only leading coefficients, discriminants, and resultants, but under slightly changed induction hypotheses. In McCallum's theorem, sign-invariance is replaced by order-invariance and the requirement that the cell lifted be connected must now be supplemented by the requirement that the cell is a submanifold. The reduction in size of the projection sets when McCallum's method is used produces enormous benefits; however, there remain unresolved problems relating to the use of

clustering when it is used. A newer version of this result is provided by McCallum (1988).

Various new results on the cylindrical algebraic decomposition method and other decision methods for the theory of real closed fields, as well as a complete bibliography on these methods, may be found in the special issue of the *Journal of Symbolic Computation* (Arnon & Buchberger 1988).

Applications to Geometrical Reasoning

It is obvious that many geometrical problems can be translated into algebraic problems by the introduction of a rectilinear coordinate system, the standard technique of analytic geometry, and then solved by quantifier elimination. A necessary condition is, of course, that the geometric objects under consideration be algebraically describable—that is, correspond to the set of points satisfying a formula of the elementary theory of real closed fields. In algebraic geometry such a set of points is called a semi-algebraic set, and we may accordingly speak of semi-algebraic geometric objects, either planar (in two-space) or solid (in three-space).

ARRANGEMENTS PROBLEMS One class of geometric problems is the class of static arrangement problems. This class can be further subdivided into two subclasses of problems: decision problems and characterization problems. In a decision problem we might ask, for example, whether, given $n+1$ specified semi-algebraic objects, the first n of them can somehow be arranged to fit inside of the last one. (We are assuming here and in the following that all objects are bounded, although generalizations would certainly be possible.) We may here either allow or not allow these objects to be subjected to rotations for the purpose of achieving the desired arrangement. If the answer to a decision problem is yes then we may wish to know at least one possible arrangement. More generally, we may wish to have a characterization of all possible arrangements.

Let us consider some simple examples of each of these types of problem. As an example of a decision problem consider the question whether two given ellipses can be arranged to fit within a third ellipse, and suppose that rotation of the ellipses is not permitted. In particular, can two copies of the ellipse $x^2/4+y^2/1 = 1$ be placed inside of the ellipse $x^2/4+y^2/1 = r^2$ if $r = 3/2$? A little mathematical reasoning obviates quantifier elimination for this simple problem. The answer is "no"; in fact, the least value of r permitting an arrangement is 9/5. If we set $r = 19/10$, then there are infinitely many possible arrangements and we obtain a characterization, the solution of which is a quantifier-free formula in the coordinates of the translated small ellipses. A more difficult problem can be obtained by asking for the placement of four copies of the small ellipse inside a larger ellipse of the same eccentricity, with rotations permitted.

The thirteen-spheres problem is a somewhat different kind of arrangement-decision problem, which was debated by Newton and Gregory. It was known that one unit sphere could be surrounded by 12 other unit spheres with each of them touching the former. The question was whether 13 was also possible. Schuette & van der Waerden (1953) showed this to be impossible. The question could have been settled, in principle, by the decision method although the number of variables and polynomials entering into the formulation of this problem would make the computational requirements far too great for existing technology.

There is a famous unsolved problem on the maximal packing density of spheres. Let N_k be the maximum number of balls of radius 1 that can be packed in a cube with edge length k, so that the density of this packing is $(4\pi N_k)/3k^3$. The limit of this ratio as k approaches infinity is the unknown maximal packing density D. By partitioning the large cube into subcubes with edge length $2^{1/2}$ and centering balls at each vertex of each such subcube and at the center of each face of each subcube, Rees (1983, pp. 41–42) shows that $D \geq \pi/3(2)^{1/2}$. If D is larger than this then we can discover a denser packing by quantifier elimination, which enables us to compute N_k for each k and, if desired, to compute a possible arrangement of N_k balls in a k-cube. Of course, the required value of k is likely beyond current computational abilities.

GEOMETRIC MOTION PROBLEMS By adding a dimension of time, we arrive at geometric motion problems, which are of crucial interest for robot motion programming. There are two kinds of geometric motion problems: collision detection and path finding. In a collision detection problem we are given semi-algebraic descriptions of several objects and for each object a semi-algebraic function describing its displacement as a function of time. The collision detection problem is to decide whether at some time a collision will occur between two or more of the objects. To be more precise about the displacement description, consider as an example a collision detection problem in two-space. For each object we will have a quantifier-free formula $F(x, y)$ that describes the object before displacement (at time 0). Its displacement will be described as a function of time by means of a 2×2 rotation matrix $R(t)$ whose entries are semi-algebraic functions (equivalently, piecewise algebraic functions) of t, and a translation vector $T(t)$ whose components are semi-algebraic functions of t. The portion of space occupied by the object at time t is then given by the formula

$$F'(x', y', t) : (\exists x)(\exists y)(F(x, y) \wedge (x', y') = R(t)(x, y) + T(t)),$$

where of course the vector equation must be replaced by two scalar equations. In practice, of course, some of the objects are unmoving obstacles,

in which case one may just set $F'(x', y')$ equal to $F(x', y')$. Notice that we cannot instead describe the rotation by giving the angle of rotation as a function of time since the sine and cosine are not algebraic functions. The occurrence of a collision at time t is then expressed by asserting that some point of space belongs to two of the displaced objects. As a decision problem, collision detection merely asks whether there is a time t at which collision (overlapping) is occurring. More information is obtained by taking it as a quantifier elimination problem, obtaining a quantifier-free formula for the set of all times t at which collision is occurring. Within this formula we can find a description of the least element of the set, the algebraic number t_0, which is likely of most interest, and from t_0 we can easily determine which objects are coinciding.

In the simplest form of a path finding problem, we are given a placement of n objects, which are regarded as fixed obstacles, and for another object both an initial placement and a destination placement. The problem is to decide whether there exists a collision-free path (motion) taking this object (which we will call a vehicle) from its initial placement to its destination and, if so, to obtain a description of some such path. The solution to this problem is obtained by considering the set of all possible (collision-free) placements of the vehicle. In the space of all placements of the vehicle, there is a path of the vehicle from its origin to its destination if and only if the origin and the destination belong to the same connected component of the set of all possible placements. By cylindrical algebraic decomposition of the space together with cell adjacency computation we can compute the component containing the initial placement and ascertain whether it contains the destination, thereby solving the path decision problem. The applicability of cylindrical algebraic decomposition to path finding was first observed by Schwartz & Sharir (1983).

To find a path from origin to destination, we first find a chain of cells, $C_0, \ldots, C_n$, such that C_0 contains the origin, C_n contains the destination, and C_i is adjacent to C_{i+1} for $0 \leq i < n$. Whenever two cells of a CAD are adjacent, one of the two has higher dimension than the other. We can use this property to define a path through the sequence of cells $C_0, \ldots, C_n$. We first choose a point in each of the cells C_i that has higher dimension than either of its neighbors; if C_0 or C_n is such a cell, we choose the origin or the destination accordingly. Suppose p is a chosen point in a cell C and that D is an adjacent cell of lower dimension. We define a path from p in C to some point q in D. If C is a sector and D is an adjacent section in the same cylinder, it is clear how to do this; the path is a vertical straight line segment having q as endpoint. Suppose that C is a sector and that D belongs to an adjacent cylinder. If C is between two sectors, S and T, let $p = (x_1, \ldots, x_n)$. Then there are points $(x_1, \ldots, x_{n-1}, u)$ and $(x_1, \ldots, x_{n-1}, v)$ belonging to S and T, respectively. Let $x_n = a \cdot u + (1-a) \cdot v$, with $0 < a < 1$. Then we choose a path from p to D all of whose points are similarly situated, as defined by the ratio a, between S and T. If C is a section then D must belong to an adjacent cylinder, and the path from p to D must be contained in the section C. These three cases determine the x_n coordinate of a path from p to D as functions of the other $n-1$ coordinates, and the same three rules are applied inductively to define the other coordinates.

After applying this process as long as possible, we obtain points in all cells of the chain. In fact we obtain two points in cells having lower dimension than either neighbor, one point in all other cells. It remains only to define a path between the two points of the cells of locally low dimension, but this can be accomplished in a manner almost identical to the method described above for the case where one point is a limit point in an adjacent cell.

In a robotic environment one will often wish to move two or more vehicles simultaneously. This introduces no new difficulty; one now calculates in the space of all k-tuples of placements to move k vehicles simultaneously. The "path" that one then computes in the manner described above is really a k-tuple of paths, one for each vehicle. The speed at which this "k-path" is traversed is then arbitrary, but the speeds of the k vehicles must be synchronized as prescribed by the k-path.

Thus far we have assumed that all objects in our model are rigid. This assumption can be relaxed. If an object has several parts whose movements are partially independent, these parts can be modeled as separate objects with constraints on the relative placements of the parts. A variety of devices—for example, rotors, hinges, swivels and tethers—can all be modeled in this manner.

We note that in three-space the description of a rigid motion will require 7 variables: 4 for the sines and cosines of the two angles describing the rotation and 3 for the components of the translation. Thus for k vehicles, or for k vehicle parts if the vehicles are not rigid, one will need $7k$ free variables. In addition one will have 6 quantified variables in the statement that each pair of bodies is disjoint:

$(\forall x)(\forall y)(\forall z)(\forall u)(\forall v)(\forall w)$ (if (x, y, z) belongs to displaced body i and (u, v, w) belongs to displaced body j then $(x, y, z) \neq (u, v, w)$).

In this succint and imprecise form the free variables describing the displacements do not explicitly appear.

We have described a model for robot motion that is geometrically rich.

Certainly the artifacts of our industrial environment can be much more satisfactorily described as semi-algebraic objects than by polyhedra and other approximate methods amenable to combinatorial and numerical methods. In spite of this richness we also possess a mathematically precise and infallible algorithm for solving the pertinent motion problems within this model. Moreover, the computational cost of this algorithm is "feasible" in the weak sense that once we fix the number of objects (and hence bound the number of variables) the computational cost is bounded by a polynomial function of the algebraic complexity of the objects (the degrees and coefficient sizes of the polynomials describing them) (Collins 1975).

Nevertheless, in practical terms and relative to currently available computational power, the computational requirements are enormous. Still, the potential benefits and the prospects for future improvements in algorithms and increases in computational power are sufficient to warrant continued work towards utilization of this approach. To cite just a few possibilities and hopes, this is a computational problem rich in latent parallelism; consider, for example that in the CAD construction all cylinders can be processed independently. Also, it is very important that, as in many applications of CAD, it is not necessary to construct a CAD of the entire space, but only to construct those cells that satisfy certain conditions. For example, in the possible displacement space of a vehicle we only need to construct cells containing points satisfying $r^2+s2 = 1$. It is likely that the CAD algorithm can also be refined and improved in many other ways. Such improvements will be discovered only through the arduous process of many attempts to solve actual problems arising from simple models of this kind.

GEOMETRIC THEOREM PROVING All geometric sentences that can be translated, by using coordinates, into first-order formulae about real numbers can be decided by any decision method for real closed fields. Hence the class of problems that can be attacked is much larger than is the case for the methods based on characteristic sets or Gröbner bases. However, owing to the complexity of the general cylindrical algebraic decomposition method, only rather simple examples involving not more than 5–6 variables can be solved using the only available implementation in SAC-2.

Kutzler & Stifter (1986b) proposed a method for using CAD for finding nondegeneracy conditions. Roughly, only the dependent variables are quantified; the independent variables are considered free. Then the corresponding quantifier-free formula (in the independent variables) gives the necessary nondegeneracy condition.

CURVE ANALYSIS Numerical curve tracing is sufficient for regions of curves that do not include singular points. The exact topological structure of curves, however, is beyond the scope of numerical techniques. Arnon (1983) and McCallum (1987) use techniques from the CAD method to determine the exact topological structure of plane curves both globally and locally.

MISCELLANEOUS OTHER ALGEBRAIC METHODS

In this section, we point to recent papers that use various other algebraic algorithms for solving certain geometric reasoning problems.

A first field of investigation is *desingularization* of algebraic plane curves and space curves. This is an important problem in the context of curve tracing where numerical procedures work well except in the neighborhood of singular points. Hoffmann (1987) and Bajaj et al (1987) use facts from algebraic geometry to give an algorithmic solution to this problem. In the case of plane curves, the algorithm iteratively applies the theorem that any algebraic plane curve can be transformed, by a birational transformation, into a curve devoid of singularities. Before desingularizing a curve, the location of singularities may be detected by using Gröbner bases (see above). The singularities of space curves could be handled, in principle, in two ways. Either they are birationally mapped to a planar curve, which is always possible, or their representation as the intersection of two surfaces is handled directly. The theoretical questions involved in the second approach are not yet satisfactorily settled. Hoffmann (1987) gives the details of the first approach, which is based on finding the representation of a space curve as the intersection of a monoid and a cone.

A second geometric reasoning problem amenable to algebraic techniques is the problem of detecting and correcting *"improperly parametrized curves"*—i.e. (rational) parametric representations of curves where two or more parameter values correspond to the points on the curve. It is shown in Sederberg (1984) that this problem is intimately related to (a special form of) the problem of finding decompositions of polynomials—i.e. representations of polynomials f in the form $f(x) = g(h(x))$. The whole area of algorithms for polynomial decomposition has received considerable attention recently (see Barton & Zippel 1985; Kozen & Landau 1989).

We have already mentioned the *implicitization problem* in the section above on Gröbner bases. Sederberg et al (1984) give a solution to this problem based on resultant theory. Goldman et al (1984) systematize these results. It is not yet well understood how the approach based on Gröbner bases compares with the resultant approach. Sederberg et al (1984) also consider the inverse *problem of uniformization*—i.e. the problem of determining parametric representations for implicitly given curves or surfaces.

A classical result by Clebsch shows that uniformization is not always possible. (An algebraic curve has a parametric rational representation iff the curve has genus zero.) Sederberg et al (1984) give a uniformization method for second-degree curves and surfaces based on factorization. In a series of papers, Abhyankar & Bajaj (1987) address the uniformization problem and give solutions, based on results from algebraic geometry and simple algebraic algorithms, for degrees 2 and 3 curves and surfaces, and for arbitrary degree plane and space curves in the solvable cases.

Another advanced problem of geometric modeling for which algebraic techniques have been used is the automatic generation of *blending surfaces*—i.e. surfaces that smooth the intersection of two given surfaces. Hoffmann & Hopcroft (1986a) give a brief survey on existing methods for solving this problem and show that the potential method presented by Hoffmann & Hopcroft (1986b), under certain restrictions, is the method yielding blending surfaces of least degree. Roughly, for given algebraic surfaces $G = 0$ and $H = 0$ (G and H polynomials in the three variables, x, y, z), the potential method considers the spectrum of space curves obtained by intersecting the surfaces $G = s$ and $H = t$ (s and t real parameters). Parameters s and t are related by a curve equation $f(s, t) = 0$. The blending surface is then defined by $f(G, H) = 0$. It can be shown that blending surfaces of this kind can be made to satisfy certain additional conditions that correspond to the geometrical intuition about smooth blends if f is sufficiently general. (In fact, a projective variant of this method is necessary to obtain full generality.) For G, H of degree d, blending surfaces of degree $2d$ can be obtained. Warren (1986) provides an in-depth study of the algorithmic algebra behind blending surfaces.

Finally, Bajaj & Kim (1987) considered the problem of generating the boundary of *configuration space obstacles* using algebraic algorithms without going through Collins's general cylindrical algebraic decomposition method. This problem is of central importance for motion planning.

CONCLUSIONS

The examples presented in this paper may indicate that algebraic techniques can contribute to all areas of geometric reasoning. At present, the applicability of purely algebraic methods is severely limited by the universality and, therefore, inherent complexity of most of these methods. Still, some of the practical results are quite encouraging. Much more research will be needed to specialize and, thereby, speed up the general methods for practically relevant subclasses of the general problem class. Also, algebraic methods should be seen in the context of, and not opposed to, other methods. Often, algebraic methods may have their merit as

"preprocessors" for other methods. Preprocessing may be applied for problems with two (sets of) input parameters x and y: A slow algebraic method may be used once to transform the first parameter x into a "feasible" equivalent form x' before a fast numerical method is used for solving the problem for many instances of the second parameter y. For example, the algebraic transformation of a parametric representation P of a geometric object into an equivalent implicit representation P' by an algebraic method may be worthwhile before solving, for many points p, the point-inclusion problem "Is p in the object represented by P?"

Hence, in future software systems for geometric reasoning, numerical and combinatorial methods should be combined with new algebraic methods for broadening the scope of applicability and enhancing the level of sophistication of these systems. Also, we believe that an integrated study of numerical, combinatorial, logical, and algebraic algorithms for geometrical problems constitutes an emerging "science of geometric reasoning" that will be more than just a collection of results in the individual independent subareas.

ACKNOWLEDGMENT

This work has been supported partially by VOEST ALPINE AG, Linz.

Literature Cited

Abhyankar, S. S., Bajaj, C. 1987. Automatic parameterization of rational curves and surfaces. I. *Comput. Aided Design* 19: 11–14 (II, III, IV to appear)

Arnon, D. S. 1981. *Algorithms for the geometry of semi-algebraic sets*. PhD thesis. Univ. Wisconsin, Madison

Arnon, D. S. 1983. Topologically reliable display of algebraic curves. *Comput. Graphics* 17: 219–27

Arnon, D. S., Buchberger, B., eds. 1988. Algorithms in real algebraic geometry. *J. Symb. Comput.* 5 (Spec. iss. 1 and 2)

Arnon, D. S., Collins, G. E., McCallum, S. 1984. Cylindrical algebraic decomposition. II: an adjacency algorithm for the plane. *SIAM J. Comput.* 13: 865–77

Arnon, D. S., Collins, G. E., McCallum, S. 1988. An adjacency algorithm for cylindrical algebraic decomposition of three-dimensional space. *J. Symb. Comput.* 5:

Arnon, D. S., Sederberg, T. W. 1984. Implicit equation for a parametric surface by Gröbner bases. *Proc. 1984 MACSYMA User's Conf., Gen. Elect., Schenectady, New York*, ed. V. E. Golden, pp. 431–36

Bajaj, C., Hoffmann, C., Hopcroft, J., Lynch, R. 1987. Tracing surface intersections. Comput. Sci. Tech. Rep. 728, Purdue Univ.

Bajaj, C., Kim, M. S. 1987. Compliant motion planning with geometric models. *Proc. 3rd ACM Symp. Comput. Geom., Waterloo, Canada, June*

Barr, A. H. 1981. Superquadrics and angle-preserving transformations. *IEEE Comput. Graphics and Applic.* 1: 11–23

Barton, D. R., Zippel, R. 1985. Polynomial decomposition algorithms. *J. Symb. Comput.* 1: 159–68

Brady, M., Hopcroft, J., Mundy, J., eds. 1988. Proceedings of the 1986 International Workshop on Geometric Reasoning. *Artificial Intelligence J.*

Buchberger, B. 1965. *An algorithm for finding a basis for the residue class ring of a zero-dimensional ideal*. PhD thesis. Univ. Innsbruck, Austria (In German)

Buchberger, B. 1970. An algorithmic criterion for the solvability of algebraic systems of equations. *Aequationes Mathematicae* 4: 374–83 (In German)

Buchberger, B. 1985. Gröbner bases: an

algorithmic method in polynomial ideal theory. In *Multidimensional Systems Theory*, ed. N. K. Bose, 1: 184–232. Dordrecht: Reidel

Buchberger, B. 1988. Applications of Gröbner bases in non-linear computational geometry. *Proc. Workshop on Sci. Software, IMA, Minneapolis, March*

Buchberger, B., Collins, G. E., Loos, R., eds. 1982. *Computer Algebra: Symbolic and Algebraic Computation*. Vienna: Springer. 283 pp.

Buchberger, B., et al. 1985. Editorial. *J. Symb. Comput.* 1: 1–6

Carra, G., Gallo, G. 1987. A procedure to prove geometrical statements. *Proc. Int. Conf. Algebraic Algorithms and Error Correcting Codes 5, Menorca, Spain, June*

Chou, S. C. 1985. *Proving and discovering geometry theorems using Wu's method*. PhD thesis. Univ. Texas, Austin. 142 pp.

Chou, S. C. 1986. A collection of geometry theorems proved mechanically. *Inst. Comput. Sci. Tech. Rep. 1986–50*. Univ. Texas, Austin

Chou, S. C., Schelter, W. F. 1986. Proving geometry theorems with rewrite rules. *J. Autom. Reason.* 2: 253–73

Chou, S. C., Yang, J. G. 1987. On the algebraic formulation of geometry theorems. *Inst. Comput. Sci. Tech. Rep.* Univ. Texas, Austin

Collins, G. E. 1975. Quantifier elimination for the elementary theory of real closed fields by cylindrical algebraic decomposition. *Lect. Notes Comput. Sci.* 33: 134–83

Collins, G. E., Loos, R. 1980. The ALDES-SAC2 computer algebra system. Tech. Rep., Univ. Wisconsin, Madison

Gelernter, H. 1959. Realization of a geometry theorem-proving machine. *Proc. Int. Conf. Info. Process., Paris, 1959*

Gianni, P., Trager, B., Zacharias, G. 1985. Gröbner bases and primary decomposition of polynomial ideals. Tech. Rep., IBM T. J. Watson Res. Cent., Comput. Algebra Group, Yorktown Heights, New York

Goldman, R. N., Sederberg, T. W., Anderson, D. C. 1984. Vector elimination: a technique for the implicitization, inversion, and intersection of planar rational polynomial curves. *Comput. Aided Geom. Design* 1: 327–56

Hoffmann, C. 1987. Algebraic curves. *Proc. Workshop on Sci. Software, IMA, Minneapolis, March*

Hoffmann, C., Hopcroft, J. E. 1986a. Quadratic blending surfaces. *Comput. Aided Design* 18: 301–6

Hoffmann, C., Hopcroft, J. E. 1986b. The potential method for blending surfaces and corners. *Conf. Geom. Modelling SIAM, Philadelphia, USA, July*, ed. G. Farin

Hopcroft, J. E., Krafft, J. E. 1986. The challenge of robotics for computer science. In *Advances in Robotics, Vol. 1: Algorithmic and Geometric Aspects of Robotics*, ed. C. Yap, J. Schwartz. Hillsdale, NJ: L. Erlbaum Assoc.

Kandri-Rody, A. 1984. *Effective methods in the theory of polynomial ideals*. PhD thesis. Rensselaer Polytechnic Institute, Troy, New York. 111 pp.

Kapur, D. 1986. Geometry theorem proving using Hilbert's Nullstellensatz. *Proc. Symp. Symb. Algebraic Comput., Waterloo, Canada, July*, pp. 202–8

Knuth, D. E. 1969. *The Art of Computer Programming, Vol. 2: Seminumerical Algorithms*. Reading: Addison-Wesley. 624 pp.

Ko, H. P., Hussain, M. A. 1985. ALGE-Prover: an algebraic geometry theorem proving software. *Gen. Elect. Tech. Rep. 85CRD139*. 28 pp.

Kozen, D., Landau, S. 1989. Polynomial decomposition algorithms. *J. Symb. Comput.* In press

Kredel, H. 1987. Primary ideal decomposition. *Proc. Euro. Conf. Comput. Algebra, Leipzig, GDR, June*

Kusche, K., Kutzler, B., Mayr, H. 1987. Implementation of a geometry theorem proving package in SCRATCHPAD II. *Proc. Euro. Conf. Comput. Algebra, Leipzig, GDR, June*

Kutzler, B. 1988. *Algebraic methods for geometric theorem proving*. PhD thesis. Univ. Linz, Austria

Kutzler, B., Stifter, S. 1986a. Automated geometry theorem proving using Buchberger's algorithm. *Proc. Symp. Symb. Algebraic Comput., Waterloo, Canada, July*, pp. 209–14

Kutzler, B., Stifter, S. 1986b. New approaches to computerized proofs of geometry theorems. *Proc. Comput. Math., Stanford, July*

Lazard, D. 1985. Ideal bases and primary decomposition: case of two variables. *J. Symb. Comput.* 1: 261–70

Loos, R. 1982. Computing in algebraic extensions. In *Computer Algebra*, ed. B. Buchberger, G. E. Collins, R. Loos, 1: 173–88. Vienna: Springer. 283 pp.

McCallum, S. 1984. *An improved projection operator for cylindrical algebraic decomposition*. PhD thesis. Univ. Wisconsin, Madison

McCallum, S. 1987. On determination of the topology of a real algebraic curve near a point. *Res. Inst. Symb. Comput. Tech. Rep. 87-24.0*, Univ. Linz, Austria

McCallum, S. 1988. An improved projection operator for cylindrical algebraic decomposition. *J. Symb. Comput.* 5. In press

Paul, R. P. 1981. *Robot Manipulators: Mathematics, Programming, and Control*. Cambridge, Mass: MIT Press

Preparata, F. P., Shamos, M. I. 1985. *Computational Geometry*. New York: Springer. 390 pp.

Prill, D. 1986. On approximation and incidence in cylindrical algebraic decompositions. *SIAM J. Comput.* 15: 972–93

Rees, E. G. 1983. *Notes on Geometry*. Berlin: Springer. 109 pp.

Ritt, J. F. 1950. *Differential Algebra*. Colloq. Publ. Vol. 33. New York: Am. Math. Soc. 184 pp.

Sakai, K., Aiba, A. 1987. Introduction to CAL. *ICOT Tech. Rep.* Tokyo, Japan

Schuette, K., van der Waerden, B. L. 1953. Das Problem der dreizehn Kugeln. *Math. Ann.* 125: 325–34

Schwartz, J., Sharir, M. 1983. On the 'piano movers' problem. II. General techniques for computing topological properties of real algebraic manifolds. *Adv. Appl. Math.* 4: 298–351

Sederberg, T. W. 1984. Degenerate parametric curves. *Comput. Aided Design* 1: 301–7

Sederberg, T. W., Anderson, D. C., Goldman, R. N. 1984. Implicit representation of parametric curves and surfaces. *Comput. Vision, Graphics, Image Process.* 28: 72–84

Swain, M. J., Mundy, J. L. 1986. Experiments in using a geometry theorem prover to prove and develop theorems in computer vision. *Proc. IEEE Conf. Robot. Automat.* 1: 250–85

Tarski, A. 1951. *A Decision Method for Elementary Algebra and Geometry*. Berkeley: Univ. Calif. Press. 2nd ed.

Trinks, W. 1978. On B. Buchberger's method for solving systems of algebraic equations. *J. Number Theory* 10: 475–88 (In German)

van der Waerden, B. L. 1953. *Modern Algebra*, Vol. I, 2. New York: F. Ungar Co.

Warren, J. D. 1986. *On algebraic surfaces meeting with geometric continuity*. PhD thesis. Cornell Univ.

Winkler, F. 1987. A geometrical decision algorithm based on the Gröbner bases algorithm. *Res. Inst. Symbol. Comput. Tech. Rep. 87-40.0*. Univ. Linz, Austria

Wu, W. T. 1978. On the decision problem and the mechanization of theorem proving in elementary geometry. *Sci. Sinica* 21: 159–72; also *Contemp. Math.* 29: 213–34

Wu, W. T. 1984. Basic principles of mechanical theorem proving in elementary geometries. *J. Syst. Sci. Math. Sci.* 4: 207–35

Wu, W. T. 1986. On zeros of algebraic equations—an application of Ritt principle. *Kexue Tongbao* 31: 1–5