Real Quantifier Elimination in Practice

Andreas Dolzmann, Thomas Sturm, and Volker Weispfenning

Universität Passau, D-94030 Passau, Germany
{dolzmann,sturm,weispfen}@fmi.uni-passau.de

MIP-9720

17 December 1997

Abstract. We give a survey of three implemented real quantifier elimination methods: partial cylindrical algebraic decomposition, virtual substitution of test terms, and a combination of Gröbner basis computations with multivariate real root counting. We examine the scope of these implementations for applications in various fields of science, engineering, and economics.

1 Introduction

The basic motivation of real quantifier elimination is to "eliminate" unwanted variables from an algebraic description of some situation. The unwanted variables may represent unknown real quantities, e.g. quantities that cannot be measured or determined directly in the given model. Consider the following simple example: We want to test the solvability of the equation $ax^2+b=0$ where a and b are considered real parameters. Then we find that this equation has a real solution if and only if $ab<0$ or $b=0$.

Thus even if we start with a single equation and eliminate a single variable, we cannot avoid the use of order inequalities in the elimination result. Moreover, we have to allow logical connectives like "and" and "or." So it makes sense to formulate the real elimination from the outset as the problem of eliminating variables from a Boolean combination of polynomial equations and inequalities. In fact, the proper formulation of the problem requires first-order logic.

Nevertheless, the heart of real quantifier elimination belongs to algebra, more precisely to real algebra. The central problem is to count the number of real solutions of a system of polynomial equations and inequalities with parametric coefficients as a function of these coefficients. In other words, the coefficients may take on arbitrary real values, and the number of solutions should be expressed uniformly by conditions on these parametric coefficients.

The past history of real quantifier elimination dates back to the 17th century. In 1637, Descartes established his *rule of signs*, cf. [Des73]:

> *Let f be a squarefree univariate real polynomial. Denote by s the number of sign changes in the coefficients of f ignoring zeroes, and by n the number of positive real roots of f. Then $n \leq s$, generally $s-n$ is even, and if all roots of f are real, then $n=s$.*

Budan (1807), cf. [BdB07], and Fourier (1831), cf. [Fou31], showed that Descartes' rule of signs is actually a special case of a more general theorem:

> *Let f be as above, let D be the finite sequence of higher non-zero derivatives of f, and let $a<b$ be real numbers such that $f(a) \neq 0$ and $f(b) \neq 0$. Denote by s the difference between the number of sign changes in the sequence $D(a)$ and the number of sign changes in the sequence $D(b)$, and let n be the number of real roots of f on $[a,b]$. Then $n \leq s$, $s-n$ is even, and if all roots of f are real, then $n=s$.*

The first decisive exact result on real root counting is the famous theorem of Sturm (1835), cf. [Stu35]:

> *Let f, a, b, n be as above, and let S be the finite sequence of polynomials obtained from f and its derivative f' by successive division with negative remainder. Denote by s the difference between the number of sign changes in the sequence $S(a)$ and the number of sign changes in the sequence $S(b)$. Then $n=s$.*

Sturm's crucial idea was to change Budan's and Fourier's sequence D of higher non-zero derivatives to the *Sturm sequence* S described above in order to get the exact number of roots.

In 1853 Sylvester observed that Sturm's theorem can easily be extended to cover an additional inequality as a side condition, cf. [Syl53]:

> *Let f, a, b be as above, and let g be another univariate real polynomial that is relatively prime to f. Denote by T the finite sequence of polynomials obtained from f and $f'g$ by successive division with negative remainder. Denote by s the difference of the number of sign changes in the sequence $T(a)$ and the number of sign changes in the sequence $T(b)$. Let n_+ and n_- be the number of real roots of f in $[a, b]$ at which g is positive or negative, respectively. Then $n_+ - n_- = s$.*

The sequence T is usually refered to as *Sturm–Sylvester sequence*. From the total number of real roots of f in $[a, b]$ obtained by Sturm's theorem and the difference of numbers obtained by Sturm–Sylvester one can easily obtain the number of real roots of f in the interval $[a, b]$, where g is positive or negative, respectively. By a little combinatorial trick, this procedure can be extended to count the number of real roots of f, where finitely many other real univariate polynomials have fixed signs.

Such a combinatorial extension of the Sturm–Sylvester theorem is the core of the first real quantifier elimination procedure found by Tarski in the 1930's, which remained unpublished until 1948, cf. [Tar48]. Tarski's procedure was very inefficient, more precisely it was not elementary recursive.

In 1975 Collins introduced a new method called *cylindrical algebraic decomposition* (CAD), cf. [Col75], which is worst-case doubly exponential in the number of variables. This was the first real quantifier elimination procedure which has been implemented.

For a long time the use of quantifier elimination in application problems outside pure mathematics has been fairly limited due to the practical complexity of the implemented methods. Not until a few years ago, some of these methods have been able to solve problems of interesting size in science, engineering, and also in economics, namely in operations research. Though the enormous increase in computational power plays a certain role, it was mainly theoretical work that contributed to this development.

On one hand CAD has gone through numerous improvements, cf. [McC88], [Hon90,Hon92], resulting in *partial* CAD, cf. [CH91], implemented in Hong's QEPCAD program. On the other hand it had been shown that real quantifier elimination is inherently hard for some problem classes, cf. [DH88,Wei88]. Thus the attention turned to special procedures for restricted problem classes, where the elimination procedures can be tuned to the structure of the problem. The focus was on considering formulas in which the occurrence of quantified variables is restricted to low degrees, cf. [Wei88,LW93,Hon93a,Hon93b,GV93,Wei94b,Wei97a]. This was initiated by the third author in 1988. In his *virtual substitution* method the number of parameters plays a minor role for the complexity. The worst-case complexity of the method is doubly exponential only in the number of the *quantifier blocks* of the input formula. This makes the method attractive for problems containing many parameters. It is implemented in the REDUCE package REDLOG, cf. [DS96,DS97a] by the first and the second author. The version of the method currently implemented is incomplete in that it can fail for input problems violating certain degree restrictions wrt. the quantified variables. In principle the method can be extended to arbitrary degrees, cf. [Wei97a].

In 1993 the third author introduced a new complete elimination procedure based on comprehensive Gröbner bases in combination with multivariate real root counting, cf. [Wei93] and [Wei92,BW94,PRS93]. The focus of the method is on problems containing many equations. It is, however, complete, i.e., there is no restriction on the possible input problems. This procedure has been implemented in the package QERRC by the first author within the computer algebra system MAS, cf. [Dol94].

In recent years there have been impressive results on asymptotically fast real elimination algorithms, cf. [Ren92,BPR94]. Implementation of these methods is still at a very early stage. So the question to what extent these methods are of practical relevance cannot be answered yet.

In this note, we will examine the applicability of automatic real quantifier elimination using the three available packages QEPCAD, REDLOG, and QERRC. We wish to emphasize that we do not consider this to be a competition between the packages. In particular, the majority of the examples will be computed with REDLOG since our current research focuses on this package and the methods implemented there.

The plan of our paper is as follows: Section 2 summarizes the logical foundations necessary for understanding the discussed elimination methods. Section 3 focusses on the mathematical background of the packages under consideration. Section 4 discusses how to encode problems from science, engineering, and economics in such a way that they can be solved by elimination methods. It also includes some automatic elimination examples with timings and references to further examples. In Section 5, we will summarize and evaluate our results.

2 A Formal Framework

In order to give a formal framework for real quantifier elimination, we introduce first-order logic on top of polynomial equations and inequalities.

We consider multivariate polynomials $f(u, x)$ with rational coefficients, where $u = (u_1 \ldots, u_m)$ and $x = (x_1, \ldots, x_n)$. We call u *parameters* and we call x *main variables*. Equations will be expressions of the form $f = 0$, inequalities are of the form $f \geq 0$, $f > 0$, or $f \neq 0$. Equations and inequalities are called *atomic formulas*. *Quantifier-free formulas* are Boolean combinations of atomic formulas by the logical operators "$\wedge$," "$\vee$," and "$\neg$." *Existential formulas* are of the form $\exists x_1 \ldots \exists x_n \psi(u, x)$, where ψ is a quantifier-free formula. Similarly, *universal formulas* are of the form $\forall x_1 \ldots \forall x_n \psi(u, x)$. A general *first-order formula* has several alternating blocks of existential and universal quantifiers in front of a quantifier-free formula.

The real *quantifier elimination problem* can be phrased as follows: Given a formula φ, find a quantifier-free formula φ' such that both φ and φ' are equivalent in the domain of the real numbers. A procedure computing such a φ' from φ is called a real *quantifier elimination procedure.*

Quantifier elimination for an existential formula $\varphi(u) \equiv \exists x_1 \ldots \exists x_n \psi(u, x)$ has a straightforward geometric interpretation: Let

$$M = \{ (u, x) \in \mathbb{R}^{m+n} \mid \psi(u, x) \},$$

and let $M' = \{ u \in \mathbb{R}^m \mid \varphi(u) \}$. Then M' is the projection of M along the coordinate axes of the existentially quantified variables x onto the parameter space. Quantifier elimination yields a quantifier-free description of this projection.

Sets defined by first-order formulas are called *definable sets.* Sets defined by quantifier-free formulas are called *semi-algebraic sets.* Obviously, every semi-algebraic set is definable. The existence of a quantifier elimination procedure implies that, vice versa, every definable set is already semi-algebraic.

We have already indicated in the Introduction that quantifier elimination for existential formulas provides a test that determines the solvability of a parametric system of equations in dependence on the parameters. The procedure gives, however, no information on possible *solutions* of the input system. This point of view gives rise to the following generalization of quantifier elimination: Given an existential formula $\varphi \equiv \exists x_1 \ldots x_n \psi(u, x)$, we wish to compute a set

$$\Phi' = \{ \big(\varphi'_k(u), \alpha_k(u)\big) \mid k \in K \}, \quad K \text{ finite},$$

which has the following properties:

1. The φ'_k are quantifier-free formulas. The α_k provide terms $t_1(u)$, …, $t_n(u)$ corresponding to the eliminated variables x_1, …, x_n.
2. Define φ' as $\bigvee_{k \in K} \varphi'_k$. Then φ' is equivalent to φ in the reals. In other words, φ' is obtained from φ by quantifier elimination.
3. Fix real values for the parameters u: if φ and hence φ' holds, then there is some φ'_k which holds. The corresponding *answer* α_k is a sample point, which when *virtually substituted* for x satisfies ψ.

For the notion of virtual substitution cf. Section 3.2. An algorithm mapping φ to Φ' as described above is called an *extended* quantifier elimination procedure, or a quantifier elimination *with answer.*

3 The Implemented Real Elimination Methods

3.1 Cylindrical Algebraic Decomposition

Cylindrical algebraic decomposition (CAD), cf. [Col75,ACM84a], is the oldest and most elaborate implemented real quantifier elimination method. It was developed by Collins and his students starting in 1974. During the last 10 years particularly Hong made very significant theoretical contributions that improved the performance of the method dramatically, cf. [Hon90,Hon92], resulting in *partial* CAD, cf. [CH91]. Hong has implemented partial CAD in his program QEPCAD based on the computer algebra C-library SACLIB. QEPCAD is not officially published but available from Hong on request.

We sketch the basic ideas of CAD: Suppose we are given an input formula

$$\varphi(u_1, \dots, u_m) \equiv \mathsf{Q}_1 x_1 \dots \mathsf{Q}_n x_n \psi(u_1, \dots, u_m, x_1, \dots, x_n), \quad \mathsf{Q}_i \in \{\exists, \forall\}.$$

Let F be the set of all polynomials occurring in ψ as left hand sides of atomic formulas. Call $C \subseteq \mathbb{R}^{m+n}$ *sign invariant* for F, if every polynomial in F has constant sign on all points in C. Then $\psi(c)$ is either "true" or "false" for all $c \in C$.

Suppose we have a finite sequence $\Pi_1, \dots, \Pi_{m+n}$ with the following properties:

1. Each Π_i is a finite partition of $\mathbb{R}^i$ into connected semi-algebraic cells. For $1 \leq j \leq n$ each Π_{m+j} is labeled with Q_j.
2. Π_{i-1} consists for $1 < i \leq m+n$ exactly of the projections of all cells in Π_i along the coordinate of the i-th variable in $(u_1, \dots, u_m, x_1, \dots, x_n)$. For each cell C in Π_{i-1} we can determine the preimage $S(C) \subseteq \Pi_i$ under the projection.
3. For each cell C in Π_m we know a quantifier-free formula $\delta_C(u_1, \dots, u_m)$ describing this cell.
4. Each cell C in Π_{m+n} is sign invariant for F. Moreover for each cell C in Π_{m+n}, we are given a *test point* t_C in such a form that we can determine the sign of $f(t_C)$ for each $f \in F$ and thus evaluate $\psi(t_C)$.

A quantifier-free equivalent for φ is obtained as the disjunction of all δ_C for which C in Π_m is *valid* in the following recursively defined sense:

1. For $m \leq i < m+n$, we have that Π_{i+1} is labeled:
   (a) If Π_{i+1} is labeled "$\exists$," then C in Π_i is valid if at least one $C' \in S(C)$ is valid.
   (b) If Π_{i+1} is labeled "$\forall$," then C in Π_i is valid if all $C' \in S(C)$ are valid.
2. A cell C in Π_{m+n} is valid if $\psi(t_C)$ is "true."

We now have to clarify how to obtain such a sequence $\Pi_1, \dots, \Pi_{m+n}$, the quantifier-free formulas δ_C, and the test points t_C. This happens in two phases, the *projection phase* and the *construction phase*.

In the projection phase, one determines from $F \subseteq \mathbb{R}[u_1, \dots, u_m, x_1, \dots, x_n]$ a new finite set

$$F' \subseteq \mathbb{R}[u_1, \dots, u_m, x_1, \dots, x_{n-1}]$$

such that the following condition is satisfied: Consider $a, b \in \mathbb{R}^{m+n-1}$ such that for all $f' \in F'$ the signs of both $f'(a), f'(b) \in \mathbb{R}$ are equal. Then for all $f \in F$ the corresponding univariate polynomials $f(a, x_n), f(b, x_n) \in \mathbb{R}[x_n]$ both have the same number of different real and complex roots.

This has the following important consequence: Let C be a connected subset of $\mathbb{R}^{m+n-1}$ that is sign invariant for F'. For each $f \in F$ consider the functions $\varrho_k : C \to \mathbb{R}$ assigning to $a \in C$ the k-th real root of $f(a, x_n) \in \mathbb{R}[x_n]$. Then all these ϱ_k are continuous. Moreover, the graphs of the various ϱ_k do not intersect. In other words, the order of the real roots does not change as they continuously change their position on the real line. In the CAD framework they are called *delineated*. The step from F to F' is called a *projection*.

Iterative application of such projections leads to a finite sequence F_{m+n}, ..., F_1, where

$$F_{m+n} := F, \qquad F_i := F'_{i+1} \quad \text{for} \quad 1 \leq i < m+n.$$

Technically, each F_i will contain certain coefficients, discriminants, resultants, and subresultant coefficients obtained from the polynomials in F_{i+1} and their higher derivatives, regarded as univariate polynomials in their last variable, which is the $(i+1)$-st one in $(u_1, \dots, u_m, x_1, \dots, x_n)$. The final set F_1 contains univariate polynomials in u_1.

The construction phase starts with the construction of a partition $\Pi_1 \subseteq \mathbb{R}$ of the real line into finitely many intervals that are sign-invariant for F_1. Simultaneously one obtains both a test point and a quantifier-free description for each interval.

For $1 < i \leq m+n$ the cell partitions $\Pi_i \subseteq \mathbb{R}^i$ are constructed recursively: Above each connected cell C in Π_{i-1} the roots of all polynomials in F_i regarded as univariate polynomials in their last variable are delineated. They thus cut the *cylinder* above C into finitely many connected semi-algebraic cells. We take Π_i to consist of all these cells arising from cylinders above the cells of Π_{i-1}. Notice that we know which cells $S(C)$ in Π_i origin from which cell C in Π_{i-1}. The required test points in the new cells are obtained by lifting the test point in the base cell C in Π_{i-1} into the various segments of the cylinder. A quantifier-free description of each cell in Π_i is constructed together with the cell from the polynomials in F_i.

Hong made the important observation that not every cell in this construction is actually necessary for eliminating quantifiers in a given input formula φ. Consider, e.g., the case that φ is of the form

$$\exists x_1 \forall x_2 \big(f(u_1, x_1) < 0 \lor g(u_1, x_1, x_2) = 0\big).$$

Then it is superfluous to construct the cells in the cylinder above a cell C of Π_2 where $f(u_1, x_1) < 0$, since we know that $f < 0 \lor g = 0$ will be "true" for every point in this cylinder. This observation generalizes to *partial* CAD, which systematically exploits the logical structure of the input formula. This can cut down enormously the number of cells to be constructed.

Notice that CAD performs a cell decomposition of the $m+n$ dimensional space of both parameters and quantified variables. Thus the number of parameters is as relevant for the complexity as the number of quantified variables.

3.2 Virtual Substitution of Parametric Test Points

The virtual substitution method dates back to a theoretical paper by the third author in 1988, cf. [Wei88]. During the last five years a lot of theoretical work has been done to improve the method, cf. [LW93,Wei94b,DSW96,DS97b,Wei97a]. After promising experimental implementations by Burhenne in 1990, cf. [Bur90], and by the second author in 1992, the method was efficiently reimplemented within the REDUCE package REDLOG by the first and the second author. REDLOG is in fact a *computer logic system* providing not only quantifier elimination but a sophisticated working environment for first-order logic over various languages and theories, cf. [DS97a]. There are also interfaces to QEPCAD and QERRC available such that these packages can be called from REDLOG and the results are available to be further processed by REDLOG. The REDLOG source code and documentation are freely available on the WWW.[1]

The applicability of the method in the form described here is restricted to formulas in which the quantified variables occur at most quadratically. Moreover quantifiers are eliminated one by one, and the elimination of one quantifier can increase the degree of other quantified variables. On the other hand there are various heuristic methods built in for decreasing the degrees during elimination. One obvious example for such methods is polynomial factorization.

For eliminating the quantifiers from an input formula

$$\varphi(u_1, \dots, u_m) \equiv \mathsf{Q}_1 x_1 \dots \mathsf{Q}_n x_n \psi(u_1, \dots, u_m, x_1, \dots, x_n), \quad \mathsf{Q}_i \in \{\exists, \forall\}$$

the elimination starts with the innermost quantifier regarding the other quantified variables within ψ as extra parameters. Universal quantifiers are handled by means of the equivalence $\forall x \psi \longleftrightarrow \neg\exists x \neg\psi$. We may thus restrict our attention to a formula of the form

$$\varphi^*(u_1, \dots, u_k) \equiv \exists x \psi^*(u_1, \dots, u_k, x),$$

where the $u_{m+1}, \dots, u_k$ are actually x_i quantified from further outside.

We fix real values $a_1, \dots, a_k$ for the parameters $u_1, \dots, u_k$. Then all polynomials occurring in ψ^* become linear or quadratic univariate polynomials in x with real coefficients. So the set

$$M = \{\, b \in \mathbb{R} \mid \psi^*(a_1, \dots, a_k, b) \,\}$$

of all real values b of x satisfying ψ^* is a finite union of closed, open, and half-open intervals on the real line. The endpoints of these intervals are among $\pm\infty$ together with the real zeros of the linear and quadratic polynomials occurring in ψ^*. Candidate terms $\alpha_1, \dots, \alpha_r$ for the zeros can be computed uniformly in $u_1, \dots, u_k$ by the solution formulas for linear and quadratic equations.

If all inequalities in ψ^* are weak, then all the intervals constituting M will, into each direction, be either unbounded or closed. In the latter case, such an interval will contain its real endpoint. Thus M is non-empty if and only if the

[1] `http://www.fmi.uni-passau.de/~redlog/`

substitution of $\pm\infty$ or of one of the candidate solutions α_j for x satisfies ψ^*. The substitution of $\pm\infty$ into a polynomial equation or inequality is evaluated in the obvious sense. The substitution of expressions in $u_1, \ldots, u_k$ of the form $(a + b\sqrt{c})/d$ among the α_j can be rewritten in such a way that all denominators involving the u_i and all square-root expressions are removed from the result, cf. [Wei97a]. By disjunctively substituting all candidates into ψ^* we obtain a quantifier-free formula equivalent to $\exists x \psi^*$.

If ψ^* contains also strict inequalities, we need to add to our candidates for points in M expressions of the form $\alpha \pm \varepsilon$, where α is candidate solution for some left-hand side polynomial occurring in a strict inequality. The symbol ε stands for a positive infinitesimal number. Again the substitution of these expressions into a polynomial equation or inequality can be rewritten in such a form that there occur neither denominators involving any of the u_i, nor any square root expressions, nor the symbol ε in the result, cf. [Wei97a]. Again this yields a quantifier-free formula equivalent to $\exists x \psi^*$. For practical applications this method, of course, has to be refined by a careful selection of a smaller number of candidate solutions and by a combination with powerful simplification techniques for quantifier-free formulas, cf. [DS97b] for details.

Recall that the well-known solution formula for quadratic equations

$$ax^2 + bx + c = 0$$

requires $a \neq 0$. In our situation a is a term in $u_1, \ldots, u_k$, so $a \neq 0$ can in general not be decided uniformly but depends on the interpretation of the u_i. Thus a quadratic polynomial $ax^2 + bx + c$ does not only deliver two square-root expressions α_1 and α_2 as candidate solutions but also $\alpha_3 = -c/b$, which in turn requires $b \neq 0$. Let t_1, t_2, and t_3 be the candidate points for M obtained from α_1, α_2, and α_3, respectively, by possibly adding or subtracting ε. With the substitution of all the t_i into ψ^*, it is necessary to add the conditions on the non-vanishing of a and b. Formally, we obtain

$$\bigl(a \neq 0 \wedge \Delta \geq 0 \wedge (\psi^*[x/t_1] \vee \psi^*[x/t_2])\bigr) \vee \bigl(a = 0 \wedge b \neq 0 \wedge \psi^*[x/t_3]\bigr),$$

where Δ denotes the discriminant of the equation $ax^2 + bx + c = 0$. If, however, a is a rational constant, then the case distinction is superfluous. In particular, if a is non-zero, the second case can be dropped.

Suppose we have eliminated an existential quantifier. Then we have in general obtained a disjunction $\psi_1' \vee \cdots \vee \psi_r'$. If the next quantifier to be eliminated is also an existential one, then we make use of the equivalence

$$\exists x_{n-1}(\psi_1' \vee \cdots \vee \psi_r') \longleftrightarrow \exists x_{n-1}(\psi_1') \vee \cdots \vee \exists x_{n-1}(\psi_r')$$

eliminating all $\exists x_{n-1}(\psi_j')$ independently. As a consequence, no candidate solutions obtained from, say, ψ_1' are substituted into the other ψ_j. This decreases the complexity class of our procedure for single quantifier blocks from doubly exponential to singly exponential in the number of quantifiers, cf. [Wei88].

Dramatic improvements of the general procedure sketched up to now can be obtained by reducing the number of test candidates for M depending on the structure of the formula ψ^*, cf. [LW93,Wei97a]. One simple instance for such an improvement is the following natural extension of *Gauss elimination*: Suppose ψ^* is of the form

$$bx + c = 0 \wedge \psi^{**},$$

where at least one of the coefficient terms b, c is a rational non-zero constant. Then we know that under any interpretation of the u_i the equation is *non-trivial*, i.e. different from $0 = 0$. Hence the only test candidate required is $-c/b$ substituted, of course, with the condition $b \neq 0$. No additional test candidates arising from equations or inequalities in the remainder ψ^{**} of ψ^* need be considered. This idea can easily be extended to a quadratic equation instead of a linear one, taking into account again the discriminant.

Ideas very similar to our extended Gauss elimination have independently been considered by Hong within the CAD framework, cf. [Hon93b].

An extended quantifier elimination can be straightforwardly derived from this method by not constructing a disjunction at the end. Instead all the quantifier-free substitution results are kept separately together with the candidate terms yielding them.

The notion of *virtual substitution* refers to both adding conditions and resolving non-standard subterms such as surds or infinitesimals. The complexity of this method depends on the number of quantified variables and, even more, on the number of quantifier changes. In theory, parameters play a minor role for the complexity. They turn in fact out to be very cheap in practice, too.

3.3 Gröbner Bases plus Multivariate Real Root Counting

The basis for this quantifier elimination method is a theorem on root counting similar to those discussed in the introduction. This beautiful and far-reaching extension of a univariate theorem by Hermite (1853), cf. [Her05], was found independently by Becker and Wörmann, cf. [BW94], and Pedersen, Roy, and Szpirglas, cf. [PRS93].

> *Let $I \subseteq \mathbb{R}[x_1, \ldots, x_n]$ be a zero-dimensional ideal. For $g \in \mathbb{R}[x_1, \ldots, x_n]$ consider the symmetric quadratic form $Q_g = \big(\mathrm{trace}(m_{gb_ib_j})\big)_{1 \leq i,j \leq d}$ on the linear $\mathbb{R}$-space $S = \mathbb{R}[x_1, \ldots, x_n]/I$, where $\{b_1, \ldots, b_d\} \subseteq S$ is a basis, and the $m_h : S \to S$ are linear maps defined by $m_h(f + I) = (hf) + I$. Let s be the signature of Q_g, and denote by n_+ and n_- the number of real roots of I at which g is positive or negative, respectively. Then $n_+ - n_- = s$.*

The use of a Gröbner basis of the ideal I allows to obtain a basis of S, and to perform arithmetic there, cf. [Buc65], thus obtaining the matrix Q_g.

Similar to the Sturm–Sylvester theorem, this approach can be extended to obtain the exact number of roots under a side condition, and can be moreover extended to several side conditions:

Let F, $\{g_1,\dots,g_k\} \subseteq \mathbb{R}[x_1,\dots,x_n]$ be finite, and assume that $I = \mathrm{Id}(F)$ is zero-dimensional. Denote by N the number of real roots $a \in \mathbb{R}^n$ of F for which $g_i > 0$ for $1 \le i \le k$. Define

$$\Gamma(\{g_1,\dots,g_k\}) = \{\, g_1^{e_1}\cdots g_k^{e_k} \mid (e_1,\dots,e_k) \in \{1,2\}^k \,\}.$$

Then defining Q_g as above we have $2^k N = \sum_{\gamma\in\Gamma(\{g_1,\dots,g_k\})} \mathrm{sig}(Q_\gamma)$.

For real quantifier elimination, this root counting has to be further extended to multivariate polynomials with parametric coefficients in such a way that it will remain correct for *every* real specialization of the parameters including specializations to zero. This task has been carried out by the third author using comprehensive Gröbner bases, cf. [Wei93]. It has been implemented by the first author within the package QERRC of the computer algebra system MAS, cf. [Dol94] and also [Sch91,Lip93].

We explain how to eliminate the quantifiers from an arbitrary input formula

$$\varphi(u_1,\dots,u_m) \equiv \mathsf{Q}_1 x_1 \dots \mathsf{Q}_n x_n \psi(u_1,\dots,u_m,x_1,\dots,x_n), \quad \mathsf{Q}_i \in \{\exists,\forall\}.$$

Using CNF/DNF computations, eliminating the quantifiers blockwise starting with the innermost block, and using the equivalence $\exists x\psi \longleftrightarrow \neg\forall x\neg\psi$, the problem can be reduced to quantifier elimination for the existential formula

$$\varphi(u)^* \equiv \exists x_1 \dots \exists x_n \Big(\bigwedge_{f\in F} f(u,x) = 0 \wedge \bigwedge_{g\in G} g(u,x) > 0 \Big).$$

If φ does not contain any equation at all, it is possible to construct redundant equations from the inequalities $g > 0$. We may thus assume that $F \neq \emptyset$. As a first step, we compute a *Gröbner system* of F, cf. [Wei92], i.e., a system of the form

$$\{ (\psi_1(u), F_1(u,x)), \dots, (\psi_r(u), F_r(u,x)) \}$$

such that whenever $\psi_i(a)$ holds for $a \in \mathbb{R}^m$, then $F_i(a,x)$ is a Gröbner basis of $\{f_1(a,x),\dots,f_k(a,x)\} \subseteq \mathbb{R}[x]$. On the formula level this corresponds to passing from φ^* to the equivalent formula

$$\varphi^{**}(u) \equiv \bigvee_{i=1}^{r} \Big(\psi_i(u) \wedge \exists x_1 \dots \exists x_n \Big(\bigwedge_{f\in F_i} f(u,x) = 0 \wedge \bigwedge_{g\in G} g(u,x) > 0 \Big)\Big).$$

We further restrict our attention to quantifier elimination for a single disjunctive branch

$$\varphi_i^{**}(u) \equiv \psi_i(u) \wedge \exists x_1 \dots \exists x_n \Big(\bigwedge_{f\in F_i} f(u,x) = 0 \wedge \bigwedge_{g\in G} g(u,x) > 0 \Big)$$

of φ^{**}. Notice that the condition ψ_i ensures that F_i is a Gröbner basis of $\mathrm{Id}(F)$ if φ_i^{**} holds, and that in this case we know the head terms of the polynomials in F_i, which are uniformly determined. We thus can uniformly determine the dimension of $\mathrm{Id}(F_i)$.

Let us for the moment assume that this dimension is zero. Then we can determine for $\gamma \in \Gamma(G)$ the matrices Q_γ of the above theorem on root counting. In contrast to the situation of the theorem, however, the matrix elements are not real numbers but polynomials in $u_1, \ldots, u_m$.

For a univariate polynomial p denote by $\tau(p)$ the number of positive real roots of p minus the number of negative real roots of p counting multiplicities. For quadratic forms Q it is well-known that $\mathrm{sig}(Q) = \tau(\chi_Q)$. Moreover, we have that

$$\sum_{\gamma \in \Gamma(G_i)} \tau(\chi_{Q_\gamma}) = \tau\Big(\prod_{\gamma \in \Gamma(G_i)} \chi_{Q_\gamma}\Big)$$

reducing the computation of the sum of signatures to that of $\tau(\prod \chi_{Q_\gamma})$ for a single univariate polynomial $\prod \chi_{Q_\gamma}$ which has only real roots. We have to construct a quantifier-free formula stating that $\tau(\prod \chi_{Q_\gamma}) \neq 0$. This can be done by applying Descartes' rule of signs discussed in the introduction to all possible combinations of signs of the coefficients of $\prod \chi_{Q_\gamma}$.

It remains to be clarified how to proceed in the case that $\mathrm{Id}(G_i)$ has a dimension greater than zero: We then compute, uniformly in $u_1, \ldots, u_m$, a maximally independent set $Y \subseteq \{x_1, \ldots, x_n\}$. Since the elimination ideal of $\mathrm{Id}(G_i)$ wrt. $\{x_1, \ldots, x_n\} \setminus Y$ is zero-dimensional, we can eliminate all quantifiers $\exists x_i$ for $x_i \notin Y$ considering all $x_j \in Y$ as additional parameters.

In the procedure above, it can happen that already $Y = \{x_1, \ldots, x_n\}$. That is, there is no *non-trivial* equation under the condition ψ_i. It is, however, always possible to replace φ_i^{**} by an equivalent formula

$$\varphi_i^{***}(u) \equiv \bigvee_j \Big(\psi_i(u) \land \exists x_1 \ldots \exists x_n \Big(a_j(u,x) = 0 \land \bigwedge_{g \in G} g(u,x) > 0\Big)\Big)$$

such that restarting the whole elimination procedure on φ_i^{***}, we can eliminate at least one quantifier in the next step.

Introducing several sophisticated modifications, the first author has improved the practical complexity of this method considerably. First, it is not necessary to consider all permutations of signs of coefficients of $\prod \chi_{Q_\gamma}$. Furthermore, inequalities $g \neq 0$ can be treated much more efficiently than by coding them as $g > 0 \lor -g > 0$ or $g^2 > 0$, cf. [Dol94].

Like CAD, this approach is a general real quantifier elimination procedure. In practice a successful application requires that the system of equations in the input is *globally* zero-dimensional i.e. zero-dimensional for all real parameter values. Moreover there should be at most one order side condition. This class of inputs is nevertheless mathematically quite interesting and comprises e.g. most implicitization problems for parametric real varieties. Similar to the virtual substitution method, the number of parameters plays a minor role compared to that of the quantified variables.

4 Applying Quantifier Elimination

All example computations mentioned in this section have been performed on a SUN Ultra 1 Model 140 workstation using 32 MB of memory.

4.1 Constraint Solving

Constraint solving classically involves a system

$$f_1(x_1, \dots, x_n) \geq 0, \quad \dots, \quad f_k(x_1, \dots, x_n) \geq 0$$

of parameter-free non-strict polynomial inequalities. One wishes to test whether such a system is *feasible*, i.e., whether it has a solution. Moreover one often wishes to determine at least one solution if there exists one. Most classical methods, e.g. the *simplex method*, cf. [Dan51], or the *Fourier–Motzkin method*, cf. [Mot36], require the variables $x_1, \dots, x_n$ to occur only linearly.

In this setting, quantifier elimination applied to the formula

$$\exists x_1 \dots \exists x_n \big(f_1(x_1, \dots, x_n) \geq 0 \wedge \dots \wedge f_k(x_1, \dots, x_n) \geq 0\big)$$

will yield "true" if and only if the system is feasible. Extended quantifier elimination will in addition yield a sample solution.

Moreover quantifier elimination can be straightforwardly applied to solve the following generalizations of the problem:

- The constraints are arbitrary Boolean combinations instead of simply conjunctions.
- The coefficients of the x_i are polynomials $a_i(u_1, \dots, u_m)$ in the parameters.
- The constraints include strict inequalities such as "$>$" and "$\neq$."
- The x_i occur with arbitrary degree. There are also arbitrary products of different x_i and x_j. Here, one possibly has to obey restrictions imposed by the quantifier elimination method used.

Notice that with strict inequalities the sample solutions provided by REDLOG will in general involve non-standard terms containing "ε." This can in principle be overcome for the price of higher complexity, cf. [LW93]. With higher degrees, solutions can in general not be determined as closed terms but in some algebraic form.

Example 1. Determine necessary and sufficient conditions for the feasibility of

$$x_1^2 + x_2^2 \leq u_1, \quad x_1^2 > u_2.$$

Applying QEPCAD to $\exists x_1 \exists x_2 (x_1^2 + x_2^2 \leq u_1 \wedge x_1^2 > u_2)$ yields $u_1 - u_2 > 0 \wedge u_1 \geq 0$ after 1 s. REDLOG obtains the same result after 10 ms. QERRC could not compute a result within 1 minute. Extended quantifier elimination in REDLOG yields also within 10 ms in addition the sample point $(x_1, x_2) = (\sqrt{u_1}, 0)$.

4.2 Optimization

Optimization differs from constraint solving in that a sample solution point is required for which an objective function $F(x_1, \ldots, x_n)$ is minimal. The simplex algorithm, cf. [Dan51], is designed for this task with a parameter-free linear function. With the Fourier–Motzkin, cf. [Mot36], method linear objective functions are introduced via an additional constraint $z \geq F(x_1, \ldots, x_n)$, where z is a new variable.

This coding of objective functions by constraints can be used also with quantifier elimination methods. For instance, one can do *hyperbolic optimization*, i.e., optimization wrt. an objective function $F(u, x) = f(u, x)/g(u, x)$ for polynomials f and g and a system of polynomial constraints $\psi(u, x)$ by eliminating

$$\exists x_1 \ldots \exists x_n \Big(\psi \wedge \big((g > 0 \wedge f \leq zg) \vee (g < 0 \wedge f \geq zg)\big)\Big).$$

Example 2. We consider the following example taken from [EPS87]:

$$F = -\frac{x_1 + 3x_2 + 2}{2x_1 + x_2 + 1} \quad \text{subject to} \quad x_1 + x_2 \leq 2 \wedge x_1 \leq 1 \wedge x_1 \geq 0 \wedge x_2 \geq 0.$$

QEPCAD obtains after 1 s the elimination result $3z + 8 \geq 0$ stating that $-8/3$ is the minimal value of F. REDLOG obtains after 70 ms the less simplified though equivalent result

$$\begin{aligned}&3z^2 + 2z - 16 \leq 0 \vee 2z + 3 \geq 0 \vee \\ &\quad (10z^2 + 13z + 4 \geq 0 \wedge 2z^2 + 5z + 2 \leq 0 \wedge 2z^2 + 3z + 1 \geq 0 \wedge z + 3 \neq 0).\end{aligned}$$

QERRC could not determine a minimum within 1 minute. Extended quantifier elimination in REDLOG yields also within 70 ms in addition sample solution points in the form of fractions involving z.

See [Wei94a,Wei97b] for further information on optimization by quantifier elimination.

4.3 Scheduling

The classical scheduling problem is as follows: *Tasks* 1, …, n have to be performed on machines 1, …, m. Each task i is assigned to a machine m_i and requires a processing time p_i. Moreover, there is a partial ordering "$\prec$" defined on the tasks. The problem is to determine starting times t_i for each task i such that none of the ordering constraints on the tasks is violated, no machine is occupied by more than one task at the same time, and the last task is finished at the earliest time possible.

This classical setting can be restated as a quantifier elimination problem as follows:

$$\exists t_1 \dots \exists t_n \Big(\bigwedge_{i=1}^{n} t_i \geq 0 \wedge$$
$$\bigwedge_{i \prec j} t_i + p_i \leq t_j \wedge \bigwedge_{\substack{1 \leq i < j \leq n \\ m_i = m_j}} (t_i + p_i \leq t_j \vee t_j + p_j \leq t_i) \wedge \bigwedge_{i=1}^{n} z \geq t_i + p_i \Big).$$

The last conjunction of constraints involving z corresponds to the treatment of objective functions discussed with optimization.

Quantifier elimination allows to tackle the following more general types of scheduling problems:

- For each job there is given an interval of time within which it has to be performed.
- Certain machines are allowed to process several tasks in parallel.
- Certain jobs require more than one machine in parallel.
- The cost function to be minimized is an arbitrary piecewise rational function of starting times of the tasks.
- The extreme freedom in the form of constraints allows to do *hierarchical* scheduling: In two elimination steps, a second objective function is optimized under the assumption of an optimal solution wrt. a first objective function.

This includes e.g. *job shop* problems on several machines, problems requiring several units of different resources arising, e.g., in the construction of buildings, and transportation problems involving penalties for earliness or tardiness of certain tasks. Here the constraint formula will usually involve disjunctions resulting from resource restrictions.

Table 1. A simple scheduling problem

job	1	2	3	4	5	6	7	8	9	10	11
p_i	40	20	15	20	20	20	10	15	15	20	15
m_i	4	2	1	1	2	4	3	2	1	3	4

$1 \prec 2 \prec 3$
$4 \prec 5 \prec 6 \prec 7$
$8 \prec 9 \prec 10 \prec 11$

Example 3. We consider the simple scheduling problem figured in Table 1. Both QEPCAD and QERRC fail on the input formula, which has 11 quantified variables and one parameter z. It contains 42 atomic formulas. Extended quantifier elimination in REDLOG provides after 3.7 s the elimination result $z - 75 \geq 0$ stating that the last task will have finished after 75 units of time. It also provides appropriate starting times $t_1, \dots, t_{11}$ for the jobs. Figure 1 shows this result as a Gantt chart.

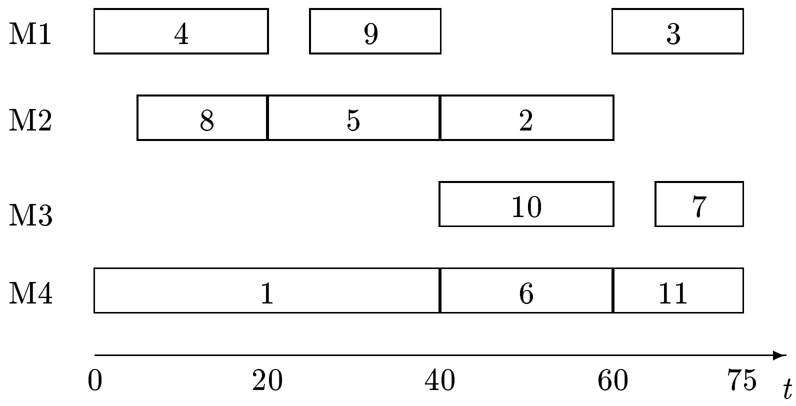

Fig. 1. A solution to the problem in Table 1

4.4 Simulation, Sizing, and Diagnosis

We consider simulation, design, and error diagnosis of electrical, mechanical, or hydraulic networks.

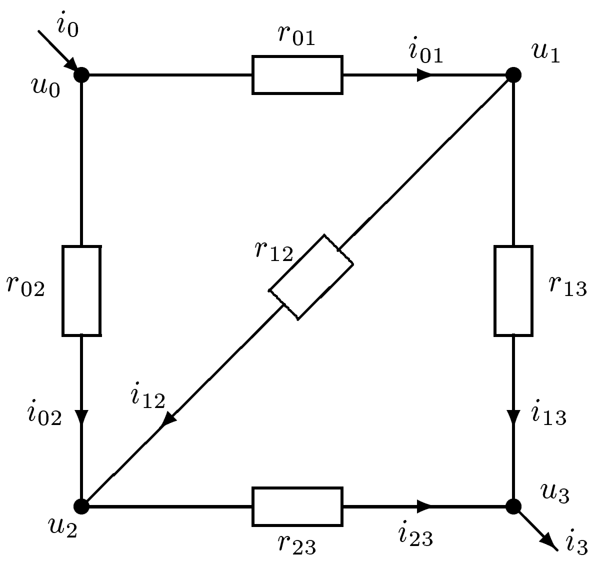

Fig. 2. A simple example network consisting of resistors

Figure 2 shows a very simple electrical sample network, cf. [Wei97b]. It consists of resistances r_{jk} subject to wattage restrictions p_{jk}. For fixed resistances r_{jk} and input voltage $u_3 - u_0$, all amperages i_{jk} and voltages u_k are uniquely determined by physical laws. A description of this circuit as a formula is obtained as follows: By Ohm's law we have

$$\omega \equiv \bigwedge_{(j,k)\in N} (u_k - u_j) = r_{jk} i_{jk}, \quad N = \{(0,1),(0,2),(1,2),(1,3),(2,3)\}.$$

Next, Kirchhoff's laws state that

$$\kappa \equiv -i_0 + i_{01} + i_{02} = 0 \wedge -i_{01} + i_{12} + i_{13} = 0 \wedge$$
$$-i_{02} - i_{12} + i_{23} = 0 \wedge -i_{13} - i_{23} + i_3 = 0.$$

Furthermore, we write down the wattage restrictions for the resistors:

$$\rho \equiv \bigwedge_{(j,k)\in N} (u_k - u_j) i_{jk} \le p_{jk}.$$

This yields an algebraic translation $\nu = \omega \wedge \kappa \wedge \rho$. Finally, we may normalize $u_0 = 0$, and we may certainly assume that $i_3 = i_0$. Since our network is intended to be a 12 V circuit, we also set $u_3 = 12$ yielding ν_0.

The easiest thing to do now is *simulation* of the circuit, i.e., we plug in values for the r_{jk}, and then eliminate all the amperages and voltages. This will yield either "true" or "false." In the latter case some wattage restriction is violated. Extended quantifier elimination would yield in addition to "true" sample values for the amperages and voltages, which are in fact the uniquely determined values assumed. More sophisticated applications of quantifier elimination in this area include, cf. [Stu97]:

Sizing In the example one can determine resistances obeying certain constraints on the amperages and voltages.

Error diagnosis Imagine a concrete network in which all resistances, amperages, and voltages are known. Suppose r_{12} is actually an ammeter for monitoring the network. If the amperage i_{12} changes, one determines from its new value which part of the circuit is damaged in which way.

Example 4. In ν_0 described above we plug in 0.25 W resistors with the following resistances (in Ω): $r_{01} = 1000$, $r_{02} = 100$, $r_{12} = 300$, $r_{13} = 47000$, $r_{23} = 200$. QEPCAD fails on this input due to lack of memory. QERRC also fails to compute a result within 1 minute. REDLOG obtains within 130 ms the elimination result "false." Changing r_{23} to 400, extended quantifier elimination in REDLOG yields after 70 ms "true" together with the exact amperages and voltages.

4.5 Control Theory and Stability

Hong, Liska, and Steinberg have shown that certain stability problems for ODE's and PDE's can be rephrased as real quantifier elimination problems. While many of these problems are still out of the range of present day elimination methods, some non-trivial and interesting instances of such problems have been solved by QEPCAD and REDLOG, cf. [HLS97].

Abdallah, Dorato, and Yang have also worked in this area. They have solved an open problem in control theory using QEPCAD in combination with REDLOG, cf. [ADY+96,DYA97].

Jirstrand has examined stationary, stability, and following of a polynomially parameterized curve for dynamical systems using QEPCAD, cf. [Jir97].

4.6 Real Implicitization

For $n < m$ let $f : \mathbb{R}^n \to \mathbb{R}^m$ be a rational map with component functions p_i/q, where p_i, $q \in \mathbb{R}[x_1, \ldots, x_n]$ for $1 \leq i \leq m$. The image $f(\mathbb{R}^n)$ of f is a definable and hence semi-algebraic subset of $\mathbb{R}^m$ described by the formula

$$\exists x_1 \ldots \exists x_n \big(q(x_1, \ldots, x_n) \neq 0 \wedge \bigwedge_{i=1}^{m} p_i(x_1, \ldots, x_n) = u_i q(x_1, \ldots, x_n)\big).$$

Our aim is to obtain a quantifier-free description of $f(\mathbb{R}^n)$ in the variables u_1, …, u_m, preferably a single equation, which would provide an implicit definition of f.

Example 5. Descartes' folium $d : \mathbb{R} \to \mathbb{R}^2$ is given by component functions $3x_1/(1 + x_1^3)$ and $3x_1^2/(1 + x_1^3)$ for u_1 and u_2, respectively, cf. [CLO92]. For obtaining an implicit form we apply quantifier elimination to

$$\exists x_1 \big(1 + x_1^3 \neq 0 \wedge 3x_1 = u_1(1 + x_1^3) \wedge 3x_1^2 = u_2(1 + x_1^3)\big).$$

QEPCAD obtains after 1 s the result $u_1^3 - 3u_1u_2 + u_2^3 = 0$. QERRC obtains after 1.6 s an elimination result with 7 atomic formulas. This can be automatically simplified to the QEPCAD result using Gröbner basis methods built into REDLOG, cf. [DS97b]. REDLOG fails on this example due to a violation of the degree restrictions.

Example 6. The *Enneper surface* $e : \mathbb{R}^2 \to \mathbb{R}^3$ is given by the component functions $3x_1 + 3x_1x_2^2 - x_1^3$, $3x_2 + 3x_1^2x_2 - x_2^3$, and $3x_1^2 - 3x_2^2$ for u_1, u_2, and u_3, respectively, cf. [CLO92]. From the corresponding formula

$$\exists x_1 \exists x_2 (3x_1 + 3x_1x_2^2 - x_1^3 = u_1 \wedge 3x_2 + 3x_1^2x_2 - x_2^3 = u_2 \wedge 3x_1^2 - 3x_2^2 = u_3),$$

REDLOG can eliminate the innermost quantifier $\exists x_2$ within 200 ms. Both QEPCAD and QERRC cannot do so. Also, none of the programs can eliminate $\exists x_1$. From the REDLOG elimination result only QERRC can eliminate $\exists x_1$. This takes 13 minutes. Only QEPCAD in combination with REDLOG is able to prove the equivalence of this result to a known complex implicitization.

4.7 Automatic Theorem Proving

Calculus

Example 7. Colmerauer has proved the following theorem using PROLOG III, cf. [Col90]: *The infinite sequence of real numbers defined by $x_{i+2} := |x_{i+1}| - x_i$, where x_1 and x_2 are arbitrary, is always periodic, and the period is 9.* Both QEPCAD and QERRC fail to prove the theorem due to lack of memory. REDLOG succeeds within 4 s.

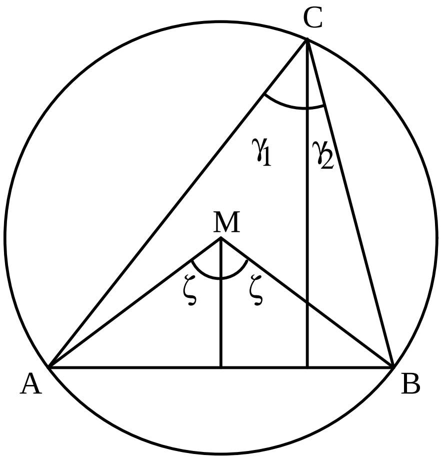

Fig. 3. The angle at circumference is half the angle at center

Geometry

Example 8. Let M be the center of the circumcircle of a triangle ABC. Then $\angle ACB = \angle AMB/2$ (see Figure 3). Choosing coordinates $A = (-a, 0)$, $B = (a, 0)$, $C = (x_0, y_0)$, and $M = (0, b)$ and encoding angles into tangents, an algebraic translation of this problem reads as follows:

$$\forall x \forall t_1 \forall t_2 \forall t \forall b \big(c^2 = a^2 + b^2 \wedge c^2 = x_0^2 + (y_0 - b)^2 \wedge \\ y_0 t_1 = a + x_0 \wedge y_0 t_2 = a - x_0 \wedge (1 - t_1 t_2) t = t_1 + t_2 \longrightarrow bt = a \big).$$

Both QEPCAD and QERRC fail on this input. REDLOG yields after 60 ms the quantifier-free equivalent $a \neq 0 \vee x_0 \neq 0 \vee y_0 \neq 0$ containing non-degeneracy conditions for the triangles.

A variant of the virtual substitution method developed by the authors has been successfully applied to numerous examples in geometric theorem proving, cf. [DSW96].

4.8 Computer Aided Design, Computer Vision, and Solid Modeling

Solids correspond to connected regular closed semi-algebraic sets in 3-space, which we describe by quantifier-free formulas.

Projections, Lighting, and Shading of Solids Projections are either parallel or central projections of a solid onto the surface of another solid in 3-space. Quantifier elimination can also be used for computing the shaded and lighted parts of a solid or the boundary between these parts. All these computations can be performed wrt. a parametric direction of the light, cf. [SW97a].

Example 9. We consider *a parametric quadric* in space of the form $\{ (u, v, w) \in \mathbb{R}^3 \mid au^2 + bv^2 + cw^2 \leq 1 \}$ and a light ray (k, l, m). Depending on the signs of the parameters a, b, c, this quadric is an ellipsoid, a cylinder, or a hyperboloid. Describing the boundary between the lighted and the shaded part of the quadric by

$$\begin{aligned} &ax^2 + by^2 + cz^2 = 1 \land \neg \exists u \exists v \exists w \exists t (t \neq 0 \land \\ &\quad au^2 + bv^2 + cw^2 \leq 1 \land u + tk = x \land v + tl = y \land w + tm = z) \end{aligned}$$

REDLOG obtains within 8.5 s the quantifier-free equivalent

$$ak^2 + bl^2 + cm^2 > 0 \land akx + bly + cmz = 0 \land ax^2 + by^2 + cz^2 - 1 = 0.$$

The converse computation to projections is to reconstruct a solid from its image. REDLOG has been used for reconstructing a rectangular solid from its parallel projection within 15.6 s eliminating 15 existential quantifiers and 2 universal quantifiers in 3 blocks, cf. [SW97a].

Offsets, Rounding, and Blending Solid modeling, cf. [Hof89,BKOS97], is concerned with the mathematical manipulation of solids. Typical operations on solids are modified set operations, offset operations, e.g. shrinking and expanding, or rounding and blending of solids. The latter is not part of standard computer aided design systems, since in general the range of valid solids there is not closed under the operations of rounding and blending, cf. [Ros85,RR86]. Representing solids by quantifier-free formulas might be a promising approach, cf. [SW97b] for the explicit computation of these operations.

4.9 Collision Detection and Path Finding

Suppose quantifier-free formulas $\alpha(x, y, z)$ and $\beta(x, y, z)$ describe objects A and B in 3-space at time $t = 0$, and (k, l, m), (p, q, r) are velocity vectors for the motion of A and B, respectively. Then a collision situation of A and B is characterized by both objects having a point in common. This is described by the formula $\psi \equiv \exists t \big(t > 0 \land \varphi(t)\big)$, where

$$\varphi(t) \equiv \exists x \exists y \exists z \big(\alpha(x - kt, y - lt, z - mt) \land \beta(x - pt, y - qt, z - rt)\big).$$

If we eliminate the four quantifiers from ψ, we get either "false," i.e., A and B will never collide, or "true" which means that a collision will take place. In the latter case, extended quantifier elimination will yield, in addition, a common point and the corresponding time. In general, this does not describe the first collision of A and B. In order to obtain time and location of the first contact of the two objects, we have to apply extended quantifier elimination to the formula $\exists t\big(t > 0 \wedge \varphi(t) \wedge \forall t'\big(0 < t' < t \longrightarrow \neg\varphi(t')\big)\big)$.

For the automatic solution of collision problems using partial CAD and REDLOG cf. [CH91] and [SW97a], respectively.

The explicit computation of a path for a robot avoiding collision with given obstacles is much harder than collision detection. One possible approach is to follow the *Voronoi diagram*, cf. [BKOS97], of the obstacles. For the computation of Voronoi diagrams in REDLOG cf. [SW97a]. Another approach uses the computation of a semi-algebraic cell decomposition of configuration space together with a computation of adjacent cells, cf. [ACM84b,McC95]. In general, one wishes not only to avoid contact to the obstacles but to keep a certain safety distance. Here one would use offset computations discussed above.

5 Conclusions

We have given a survey of three implemented quantifier elimination methods sketching their advantages and drawbacks for certain types of problems. Testing their applicability to various problems in science, engineering, and economics, it has turned out that none of them is definitely superior to the others. It can, in contrast, be necessary to combine all three of them in order to solve a problem. Exploring the rules governing the applicability of the various methods and designing a scheduling procedure automating the generation of subproblems and their distribution to the available procedures is certainly an interesting topic for further research in this area. The availability of a uniform interface to all implementations in REDLOG is a reasonable first step into this direction. Altogether quantifier elimination is now at a stage where it can solve significant application problems.

References

[ACM84a] Dennis S. Arnon, George E. Collins, and Scott McCallum. Cylindrical algebraic decomposition I: The basic algorithm. *SIAM Journal on Computing*, 13(4):865–877, November 1984.

[ACM84b] Dennis S. Arnon, George E. Collins, and Scott McCallum. Cylindrical algebraic decomposition II: An adjacency algorithm for the plane. *SIAM Journal on Computing*, 13(4):878–889, November 1984.

[ADY+96] Chaouki T. Abdallah, Peter Dorato, Wei Yang, Richard Liska, and Stanly Steinberg. Applications of quantifier elimination theory to control system design. In *Proceedings of the 4th IEEE Mediterranean Symposium on Control and Automation*, pages 340–345. IEEE, 1996.

[BdB07] Ferdinand François Désiré Budan de Boislaurent. Nouvelle méthode pour la résolution des équations numériques d'un degré quelconque, 1807. 2nd edition, Paris 1822.

[BKOS97] Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf. *Computational Geometry, Algorithms and Applications.* Springer, Berlin, Heidelberg, New York, 1997.

[BPR94] Saugata Basu, Richard Pollack, and Marie-Françoise Roy. On the combinatorial and algebraic complexity of quantifier elimination. In Shafi Goldwasser, editor, *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pages 632–641, Los Alamitos, CA, USA, November 1994. IEEE Computer Society Press.

[Buc65] Bruno Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal.* Doctoral dissertation, Mathematical Institute, University of Innsbruck, Innsbruck, Austria, 1965.

[Bur90] Klaus-Dieter Burhenne. Implementierung eines Algorithmus zur Quantorenelimination für lineare reelle Probleme. Diploma thesis, Universität Passau, D-94030 Passau, Germany, December 1990.

[BW94] Eberhard Becker and Thorsten Wörmann. On the trace formula for quadratic forms. In William B. Jacob, Tsit-Yuen Lam, and Robert O. Robson, editors, *Recent Advances in Real Algebraic Geometry and Quadratic Forms*, volume 155 of *Contemporary Mathematics*, pages 271–291, Providence, Rhode Island, 1994. American Mathematical Society, American Mathematical Society. Proceedings of the RAGSQUAD Year, Berkeley, 1990–1991.

[CH91] George E. Collins and Hoon Hong. Partial cylindrical algebraic decomposition for quantifier elimination. *Journal of Symbolic Computation*, 12(3):299–328, September 1991.

[CLO92] David Cox, John Little, and Donald O'Shea. *Ideals, Varieties and Algorithms.* Undergraduate Texts in Mathematics. Springer-Verlag, New York, Berlin, Heidelberg, 1992.

[Col75] George E. Collins. Quantifier elimination for the elementary theory of real closed fields by cylindrical algebraic decomposition. In H. Brakhage, editor, *Automata Theory and Formal Languages. 2nd GI Conference*, volume 33 of *Lecture Notes in Computer Science*, pages 134–183, Berlin, Heidelberg, New York, May 1975. Gesellschaft für Informatik, Springer-Verlag.

[Col90] Alain Colmerauer. Prolog III. *Communications of the ACM*, 33(7):70–90, July 1990.

[Dan51] George B. Dantzig. Maximization of a linear function of variables subject to linear inequalities. In T. C. Koopmans, editor, *Activity Analysis of Production and Allocation*, pages 339–347. John Wiley & Sons, New York, 1951.

[Des73] René Descartes. La géometrie—livre III. In Charles Adam and Paul Tannery, editors, *Discours de la Méthode & Essais*, volume 6 of *Œuvres des Descartes*, pages 442–485. Librairie Philosophique J. Vrin, Paris, new edition, 1973.

[DH88] James H. Davenport and Joos Heintz. Real quantifier elimination is doubly exponential. *Journal of Symbolic Computation*, 5(1–2):29–35, February–April 1988.

[Dol94] Andreas Dolzmann. Reelle Quantorenelimination durch parametrisches Zählen von Nullstellen. Diploma thesis, Universität Passau, D-94030 Passau, Germany, November 1994.

[DS96] Andreas Dolzmann and Thomas Sturm. Redlog user manual. Technical Report MIP-9616, FMI, Universität Passau, D-94030 Passau, Germany, October 1996. Edition 1.0 for Version 1.0.

[DS97a] Andreas Dolzmann and Thomas Sturm. Redlog: Computer algebra meets computer logic. *ACM SIGSAM Bulletin*, 31(2):2–9, June 1997.

[DS97b] Andreas Dolzmann and Thomas Sturm. Simplification of quantifier-free formulae over ordered fields. *Journal of Symbolic Computation*, 24(2):209–231, August 1997.

[DSW96] Andreas Dolzmann, Thomas Sturm, and Volker Weispfenning. A new approach for automatic theorem proving in real geometry. Technical Report MIP-9611, FMI, Universität Passau, D-94030 Passau, Germany, May 1996. To appear in the Journal of Automated Reasoning.

[DYA97] Peter Dorato, Wei Yang, and Chaouki Abdallah. Robust multi-objective feedback design by quantifier elimination. *Journal of Symbolic Computation*, 24(2):153–159, August 1997. Special issue on applications of quantifier elimination.

[EPS87] Horst A. Eiselt, Giorgio Pederzoli, and Carl-Louis Sandblom. *Continuous Optimization Models*. De Gruyter, Berlin, 1987.

[Fou31] Jean Baptiste Joseph Fourier. *Analyse des équations déterminées*. F. Didot, Paris, 1831.

[GV93] Laureano González-Vega. A combinatorial algorithm solving some quantifier elimination problems. Technical Report 11-1993, University of Cantabria, Departemento de Mathematicas, Estadistica y Computación, Facultad de Ciencias, Avenida de Los Castros, 39071 Santander, Spain, November 1993.

[Her05] Charles Hermite. Remarques sur le théorème de M. Sturm. In Emile Picard, editor, *Œuvres des Charles Hermite*, volume 1, pages 284–287. Gauthier-Villars, Paris, 1905.

[HLS97] Hoon Hong, Richard Liska, and Stanly Steinberg. Testing stability by quantifier elimination. *Journal of Symbolic Computation*, 24(2):161–187, August 1997. Special issue on applications of quantifier elimination.

[Hof89] Christoph M. Hoffmann. *Geometric and Solid Modeling*. Computer Graphics and Geometric Modeling. Morgan Kaufmann, San Mateo, California, 1989.

[Hon90] Hoon Hong. An improvement of the projection operator in cylindrical algebraic decomposition. In Shunro Watanabe and Morio Nagata, editors, *Proceedings of the International Symposium on Symbolic and Algebraic Computation (ISSAC 90)*, pages 261–264, New York, August 1990. ACM, ACM Press.

[Hon92] Hoon Hong. Simple solution formula construction in cylindrical algebraic decomposition based quantifier elimination. In Paul S. Wang, editor, *Proceedings of the International Symposium on Symbolic and Algebraic Computation (ISSAC 92)*, pages 177–188, Baltimore, MD, July 1992. ACM, ACM Press.

[Hon93a] Hoon Hong. Quantifier elimination for formulas constrained by quadratic equations. In Manuel Bronstein, editor, *Proceedings of the International Symposium on Symbolic and Algebraic Computation (ISSAC 93)*, pages 264–274, New York, July 1993. ACM, ACM Press.

[Hon93b] Hoon Hong. Quantifier elimination for formulas constrained by quadratic equations via slope resultants. *THE Computer Journal*, 36(5):440–449, 1993. Special issue on computational quantifier elimination.

[Jir97] Mats Jirtsrand. Nonlinear control system design by quantifier elimination. *Journal of Symbolic Computation*, 24(2):137–152, August 1997. Special issue on applications of quantifier elimination.

[Lip93] Frank Lippold. Implementierung eines Verfahrens zum Zählen reeller Nullstellen multivariater Polynome. Diploma thesis, Universität Passau, D-94030 Passau, Germany, September 1993.

[LW93] Rüdiger Loos and Volker Weispfenning. Applying linear quantifier elimination. *The Computer Journal*, 36(5):450–462, 1993. Special issue on computational quantifier elimination.

[McC88] Scott McCallum. An improved projection operation for cylindrical algebraic decomposition of three-dimensional space. *Journal of Symbolic Computation*, 5(1–2):141–161, February–April 1988.

[McC95] Scott McCallum. Partial solution to path finding problems using the CAD method. Electronic Proceedings of the IMACS ACA 1995 on `http://math.unm.edu/ACA/1995.html`, 1995.

[Mot36] Theodore S. Motzkin. *Beiträge zur Theorie der linearen Ungleichungen.* Doctoral dissertation, Universität Zürich, 1936.

[PRS93] P. Pedersen, M.-F. Roy, and A. Szpirglas. Counting real zeroes in the multivariate case. In F. Eysette and A. Galigo, editors, *Computational Algebraic Geometry*, pages 203–224, Boston, 1993. MEGA '92.

[Ren92] James Renegar. On the computational complexity and geometry of the first-order theory of the reals. *Journal of Symbolic Computation*, 13(3):255–352, March 1992. Part I–III.

[Ros85] Jaroslaw R. Rossignac. *Blending and Offsetting Solid Models.* Ph.D. thesis, Department of Electrical Engineering, College of Engineering and Applied Science, University of Rochester, Rochester, New York 14627, July 1985.

[RR86] Jaroslaw R. Rossignac and Aristides A. G. Requicha. Offsetting operations in solid modelling. *Computer Aided Geometric Design*, 3(2):129–148, August 1986.

[Sch91] Elke Schönfeld. Parametrische Gröbnerbasen im Computeralgebrasystem ALDES/SAC-2. Diploma thesis, Universität Passau, D-94030 Passau, Germany, May 1991.

[Stu35] Jacques Charles François Sturm. Mémoire sur la résolution des équations numériques. In *Mémoires présentés par divers Savants étrangers à l'Académie royale des sciences, section Sc. math. phys.*, volume 6, pages 273–318, 1835.

[Stu97] Thomas Sturm. Reasoning over networks by symbolic methods. Technical Report MIP-9719, FMI, Universität Passau, D-94030 Passau, Germany, December 1997.

[SW97a] Thomas Sturm and Volker Weispfenning. Computational geometry problems in Redlog. Technical Report MIP-9708, FMI, Universität Passau, D-94030 Passau, Germany, April 1997.

[SW97b] Thomas Sturm and Volker Weispfenning. Rounding and blending of solids by a real elimination method. In Achim Sydow, editor, *Proceedings of the 15th IMACS World Congress on Scientific Computation, Modelling, and Applied Mathematics (IMACS 97)*, volume 2, pages 727–732, Berlin, August 1997. IMACS, Wissenschaft & Technik Verlag.

[Syl53] James Joseph Sylvester. On a theory of the syzegetic relations of two rational integral functions, comprising an application to the theory of Sturm's functions, and that of the greatest algebraical common measure. *Philosophical Transactions of the Royal Society London*, 143:407–548, 1853.

[Tar48] Alfred Tarski. A decision method for elementary algebra and geometry. Technical report, RAND, Santa Monica, CA, 1948.

[Wei88] Volker Weispfenning. The complexity of linear problems in fields. *Journal of Symbolic Computation*, 5(1–2):3–27, February–April 1988.

[Wei92] Volker Weispfenning. Comprehensive Gröbner bases. *Journal of Symbolic Computation*, 14:1–29, July 1992.

[Wei93] Volker Weispfenning. A new approach to quantifier elimination for real algebra. Technical Report MIP-9305, FMI, Universität Passau, D-94030 Passau, Germany, July 1993. To appear in the proceedings of the Collins Symposium 1993.

[Wei94a] Volker Weispfenning. Parametric linear and quadratic optimization by elimination. Technical Report MIP-9404, FMI, Universität Passau, D-94030 Passau, Germany, April 1994.

[Wei94b] Volker Weispfenning. Quantifier elimination for real algebra—the cubic case. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation in Oxford*, pages 258–263, New York, July 1994. ACM Press.

[Wei97a] Volker Weispfenning. Quantifier elimination for real algebra—the quadratic case and beyond. *Applicable Algebra in Engineering Communication and Computing*, 8(2):85–101, February 1997.

[Wei97b] Volker Weispfenning. Simulation and optimization by quantifier elimination. *Journal of Symbolic Computation*, 24(2):189–208, August 1997. Special issue on applications of quantifier elimination.