Electronic Colloquium on Computational Complexity, Report No. 36 (2003)

# Polynomial Equation Elimination via Tarski Algebra

B. Litow [§]

April 28, 2003

**Abstract**

The elimination problem is classical: implicitly express one of the variables occurring in a finite system of polynomial equations as an algebraic function of a designated subset of the remaining variables. Solutions to this problem by resultants, or more comprehensively by use of Gröbner basis methods are available. In this paper we show under an assumption that a very direct solution can be carried out using Tarski algebra. The Tarski algebra approach has two advantages over other, more involved methods. First, it allows for the direct determination of the possibility of eliminating variables in terms of deciding a single sentence. Second, assuming that a deep result of Grigoriev can be extended from sentences to formulas of Tarski algebra, the algorithm we present is in EXPTIME, while other methods are so far only known to have a doubly exponential worst case running time.

## 1 The elimination problem

In this paper we work over $\mathbb{R}$, the field of real numbers. All instances of O-notation indicate absolute, positive constants.

Let $\mathbf{x}$ and $\mathbf{y}$ denote the variables $x_1, \ldots, x_r$, and $y_1, \ldots, y_s$, respectively. We also let $\mathbf{y}'$ denote $y_2, \ldots, y_s$. Given a finite set $\Omega$ of equations $P_1 = 0, \ldots, P_s = 0$, where $P_i$ is an integer coefficient polynomial in $\mathbf{x}$ and $\mathbf{y}$, the elimination problem involves two determinations. Before proceeding to specify these, note that satisfaction of $\Omega$ at $\mathbf{x}, \mathbf{y}$ is equivalent to $P_1^2(\mathbf{x}, \mathbf{y} + \cdots + P_s^2(\mathbf{x}, \mathbf{y}) = 0$. We denote this by $\Omega(\mathbf{x}, \mathbf{y}) = 0$. Observe that an essential property of $\mathbf{R}$ is being used here.

In describing the elimination questions we work with neighborhoods of 0. This restriction amounts to assuming that $\Omega(0, \ldots, 0) = 0$. This is not a serious restriction, e.g., a neighborhood of $\mathbf{a}, \mathbf{b} \in \mathbb{R}^r \times \mathbf{R}^s$ simply requires working with $(x_1 - a_1)^2 + \cdots + (y_s - b_s)^2$.

The formula $\mathcal{N}(u_1, \ldots, u_p, v)$ denotes

$$v > 0 \wedge u_1^2 + \cdots + u_p^2 < v \ .$$

[§]School of Information Technology, James Cook University, Townsville, Qld. 4811, Australia bruce@cs.jcu.edu.au



ISSN 1433-8092

Let $\mathbf{z} = z_1, \ldots, z_r$. The formula $\mathcal{E}(\mathbf{x}, \mathbf{y}, \mathbf{z}, u)$ denotes

$$\mathcal{N}(\mathbf{x}, u) \wedge \mathcal{N}(\mathbf{y}, u) \wedge \mathcal{N}(\mathbf{z}, u) \wedge \Omega(\mathbf{x}, \mathbf{y}) = 0 \wedge \Omega(\mathbf{x}, \mathbf{z}) = 0 \Rightarrow (y_1 - z_1)^2 + \cdots + (y_s - z_s)^2 = 0 \ .$$

**The elimination questions.**

1. Decide whether there exist open balls of radius $u$ about 0, such that if $\mathbf{x}, \mathbf{y}$ are in their respective balls, and $\Omega$ is satisfied, then $\mathbf{y}$ is uniquely determined by $\mathbf{x}$. This question can be answered by deciding the sentence

$$\exists u \ \forall \mathbf{x}, \mathbf{y}, \mathbf{z} \mathcal{E}(\mathbf{x}, \mathbf{y}, \mathbf{z}, u) \ . \tag{1}$$

2. Assuming Eq. 1, compute an $r+1$-variate, integer coefficient polynomial $A$, and $0 < \epsilon$ such that for $\mathcal{N}(\mathbf{x}, y_1, \epsilon)$, $A(\mathbf{x}, y_1) = 0$ implicitly defines $y_1$ as a function of $\mathbf{x}$.

It is classical that a sufficient condition ensuring Eq. 1 is that $\mathrm{Det}J \neq 0$, where $J$ is the $s \times s$ matrix such that

$$J_{i,j} = \frac{\partial P_i}{\partial y_j}(0, \ldots, 0) \ .$$

See [9]. This sufficient condition can be tested in $\mathbf{P}$. Conventional results about polynomials can be used to compute a neighborhood size. However, Eq. 1 expresses the necessary and sufficient condition for elimination in a neighborhood of 0. Note that if Eq. 1, then in some neighborhood of 0, $y_1, \ldots, y_s$ are all functions of $\mathbf{x}$. We will show that this implies the existence of the polynomial $A$, and that such a polynomial can be computed.

Both resultants and Gröbner basis methods can be used to solve the elimination problem. A resultant-based approach , with an application to context-free grammar generating series is presented in [5]. An alternative, Gröbner basis method is described in [8]. This paper also points out that zero-dimensional elimination ideals can obstruct the resultant-based approach. For a general discussion of elimination theory, see [2]. In any case, these methods have doubly exponential worst case running time.

## 2 Tarski algebra

The approach of this paper is based on the first order theory of $\mathbb{R}$, which will be referred to as Tarski algebra, TA, in honor of Alfred Tarski who was the first to exhibit a quantifier elimination decision method for the theory. See [10]. We sketch only as much of TA as is needed to develop a solution to the elimination problem. More detailed treatment can be found in [7]. Related material is in [3, 6].

TA is first order logic with equality to which are adjoined the finitely many standard axioms for $\mathbb{R}$, which involve the 0-ary function symbols $0, 1$, and the 2-ary function symbols $+, \times$, with their standard interpretations. Numerals are terms built up in obvious fashion from $0, 1, +, \times$. Note that integers up to $2^n$ can be expressed by terms of length $O(n^2)$, at worst by using binary notation where $2^k$ is expressed by the term

$$\overbrace{(1+1) \times \cdots \times (1+1)}^{k} \ .$$

The terms of TA are in fact polynomials. It is easy to see that the length of the usual sparse notation for an integer coefficient polynomial is polynomially related to the length of its corresponding TA term. It is now straightforward to define the length $|F|$ of a TA formula $F$ to be the sum of the lengths of all of its terms, and all logical symbols. Since all formulas are TA formulas, the modifier TA will usually be dropped.

An atomic formula has the form $P \lhd 0$, where $P$ is a term, and $\lhd \in \{=, \leq, \geq, <, >\}$. In fact, by Theorem 2, we can restrict ourselves to $\lhd \in \{=, >\}$. It is worth noting that our arguments do not depend on excluding occurrences of $<$, etc.

If formula $F$ has the free variables $z_1, \ldots, z_h$, then the set of tuples $(a_1, \ldots, a_h) \in \mathbb{R}^h$ such that $F(a_1, \ldots, a_h)$ is called its extension. Two formulas in the same free variables are equivalent if their extensions coincide.

Collins established the next result, showing that it is the number of distinct variables, rather than simply formula length that most sensitively influences time complexity of TA quantifier elimination. See[1].

**Theorem 1 (Collins)** *A quantifier-free formula $F'$, equivalent to a given prenex formula $F$ can be computed in time $|F|^{2^{O(r)}}$ time, where $r$ is the number of distinct variables.*

Note the separation of the number of variables from the overall formula size in terms of effect on the worst case time. This separation is refined further by the next result.

A TA formula is said to be a G-formula if it is in prenex, and all of its atomic formulas have the form $P \geq 0$, where, of course $P$ is a term. The next result is due to Grigoriev. The result is stated in terms of sentences, but it is a plausible conjecture that it extends to formula generally. See [4]. In stating this theorem, $\exists_i$ $(\forall_j)$ abbreviates a list of existentially (universally) quantified variables, and $P$ is a quantifier-free formula. Our statement of this result combines the time bound with Lemma 13 of the paper.

**Theorem 2 (Grigoriev)** *A G-sentence $F = \exists_1 \forall_2 \cdots \exists_a P$ can be converted into an equivalent quantifier-free sentence*

$$\bigvee_{m_1} \bigwedge_{m_2} \cdots \bigvee_{m_a} P(w_{m_1,\ldots,m_a}) \ ,$$

*where the $w_{m_1,\ldots,m_a}$ are lists of rationals substituting for the variables. in time $|F|^{r^{O(a)}}$, where $r$ is the number of distinct variables, and $a$ is the number of quantifier alternations.*

We make the assumption that Theorem 2 extends to formulas. In this case. $P(w_{m_1,\ldots,m_a})$ may contain free variables.

Observe that the atomic formula $P = 0$ can be expressed as the G-formula $P \geq 0 \wedge \forall x \ x \times x - P \geq 0$, where $x$ does not occur in $P$. However, it is not clear that any prenex formula can be converted into an equivalent G-formula without increasing

the quantifier alternation by an unbounded amount. In our application, it will be easy to express everything necessary in terms of G-formulas and maintain an $O(1)$ bound on quantifier alternation.

The next lemma enables us to go from the assertion that a neighborhood exists to actually computing a suitable radius for them. Notation is drawn from Theorem 2

**Lemma 1** *Let $F$ be a G-formula in a single free variable such that*

$$\exists u \forall v \; u > 0 \wedge v * v < u \Rightarrow F(v) \; ,$$

*i.e., $F(v)$ in some neighborhood of 0, then a positive rational $\alpha$ can be computed in $|F|^{r^{O(a)}}$ time such that $\forall v \; v \cdot v < \alpha \Rightarrow F(v)$.*

**Proof :** By Theorem 2, $F$ is equivalent to a quantifier-free formula $F'$ in a single free variable. $F'$ can be computed in time $|F|^{r^{O(a)}}$. This implies of course that $|F'| < |F|^{r^{O(a)}}$. Now, $F'$ is equivalent to a DNF expression $X = \bigvee_i \bigwedge_j X_{i,j}$, where each $X_{i,j}$ is an atomic formula of one of the two forms

$$P \begin{array}{l} > 0 \\ = 0 \end{array} \; ,$$

and each $P$ occurs in $F'$, so $|P| < |F'|$. Note that we do not actually construct the DNF expression, we use it in our argument.

Consider a clause $X_i = \bigwedge_j X_{i,j}$, and its constituent atomic formulas. Observe that the extension of atomic formula $P > 0$ is an open set, while that of $P = 0$ is a finite set of reals. Both sets could be empty. The extension of a clause, then is either an open set, or a finite set of reals. By assumption, at least one clause extension must be a nonempty open set, and the finite union over all these nonempty open sets is a subset of the extension of $X$. It follows from the standard topology of $\mathbf{R}$ that at least one clause extension must contain a neighborhood of 0. Again, by topology, the extension of at least one $X_{i,j}$ must contain a neighborhood of 0. The sentence $\exists u \forall v \; u > 0 \wedge v * v < u \Rightarrow X_{i,j}(v)$ expresses this fact, and so a suitable $X_{i,j}$ of form $P > 0$ can be determined in $|F|^{r^{O(a)}}$ time.

Let $P > 0$ be the selected formula. It is clear that the required $\alpha$ can be computed in PTIME from $P$. □

## 3 A Tarski algebra solution

We can now prove, using notation from Theorem 2,

**Theorem 3** *If Theorem 2 extends to formulas, the elimination problem can be solved in EXPTIME.*

**Proof :** First, note that $\mathcal{E}(\mathbf{x}, \mathbf{y}, \mathbf{z}, u)$, and associated formulas that are inputs to the decision method have $O(1)$ alternations of quantifiers, and $O(1)$ atomic formulas. This means that they can be converted to G-formula and retain $O(1)$ alternations. From this

point on, we set $a = O(1)$.

That Eq. 1 can be decided in the indicated times follows directly from Theorem 2. Assuming Eq. 1, by Theorem2 and Lemma 1, a rational $\delta$ can be computed such that

$$\forall \mathbf{x}, \mathbf{y}, \mathbf{z} \mathcal{E}(\mathbf{x}, \mathbf{y}, \mathbf{z}, \delta) ,$$

and $\delta$ requires $n^{r^{O(a)}}$ bits.

By the Theorem 2, and the upper bound on the bit size of $\delta$,

$$\forall \mathbf{y}', \mathbf{z} \mathcal{E}(\mathbf{x}, \mathbf{y}, \mathbf{z}, \delta)$$

can be converted into an equivalent quantifier-free formula $\hat{\mathcal{E}}(\mathbf{x}, y_1)$ in time

$$(n^{r^{O(a)}})^{r^{O(a)}} = n^{r^{O(a)}} .$$

Now we use the form of $\hat{\mathcal{E}}(\mathbf{x}, y_1)$ provided by Theorem 2. An equivalent DNF formula $X = \bigvee_i \bigwedge_j X_{i,j}$ can clearly be constructed by brute force in time polynomial in the number of clauses, which is, directly from the time bound, $m_1 \cdots m_a < n^{r^{O(a)}}$. Each $X_{i,j}$ is an atomic formula of form $Q(\alpha_2, \ldots, \alpha_s, \beta_1, \ldots, \beta_r) \geq 0$, where $\alpha_2, \ldots, \alpha_s$ are rationals substituted for $y_2, \ldots, y_s$, and $\beta_1, \ldots, \beta_r$ are rationals substituted for $z_1, \ldots, z_r$, and $Q$ occurred in the quantifier-free formula $\hat{\mathcal{E}}(\mathbf{x}, y_1)$.

Each clause $X_i = \bigwedge_j X_{i,j}$ has no more than $a$ conjuncts, so picturing a clause

$$X_i = (Q_1 \geq 0) \wedge \cdots \wedge (Q_b \geq 0) ,$$

where $b \leq a$ we can expand $X_i$ as

$$\bigvee_g (\bigwedge_h (Q_h > 0) \wedge \bigwedge_f (Q_f = 0)) .$$

Note that in some cases there may only be strict inequality conjuncts, or equality conjuncts. This expansion results in a new DNF expression equivalent to $X$ consisting of at most $2^a$ times as many clauses as in $X$.

Consider a clause $\bigwedge_h (Q_h > 0) \wedge \bigwedge_f (Q_f = 0)$. We use the fact in some neighborhood of 0, say $\mathcal{N}$, if $(a_1, \ldots, a_r, b), (a_1, \ldots, a_r, c) \in \mathbf{N}$, then $\hat{\mathcal{E}}(\mathbf{a}, b)$ and $\hat{\mathcal{E}}(\mathbf{a}, c)$ imply that $b = c$. Thus the functional dependence of $y_1$ on $\mathbf{x}$ is enforced solely by the subclause $\bigwedge_f (Q_f = 0)$. This follows from the fact that for fixed $\mathbf{x}$, the set of $y_1$ such that $Q > 0$ is open, and a finite intersection of open sets is again open. The subclause of the equality formuals is equivalent to $\sum_f Q_f^2 = 0$. For each such atomic formula $A$, decide whether first

$$\exists v \forall \mathbf{x} \exists y_1 \mathcal{N}(y_1, v) \wedge \mathcal{N}(\mathbf{x}, v) \Rightarrow A(\mathbf{x}, y_1) ,$$

that is, $A$ is not vacuous, and second

$$\exists v \forall w \mathcal{N}(\mathbf{x}, v) \wedge \mathcal{N}(y_1, v) \wedge A(\mathbf{x}, y_1) \wedge A(\mathbf{x}, w) \Rightarrow y_1 = w .$$

These formulas can be recast as G-formula and retain $O(1)$ alternations. Thus, the time to find an $A$ that passes both tests (and we have shown one must exist) is still bounded above by $n^{r^{O(a)}}$. Now $a = O(1)$, and $n^{r^{O(1)}} = 2^{r^{O(1)} \cdot \log n} = 2^{n^{O(1)}}$, which is EXPTIME. $\square$

# 4 Conclusion

Based on a conjecture concerning Theorem 2, an EXPTIME algorithm for the elimination problem has been given. In addition, Tarski algebra provides a framework for dealing with complex algebraic problems with conceptual clarity, since TA captures a substantial fragment of our intuition about $\mathbb{R}$.

There is the question of whether Theorem 2 does extend to formulas, in the indicated form. Due to the quite involved nature of Grigoriev's argument in [4], the author is unable to convince himself that this extension holds. However, Lemma 10 of that paper suggests that the conjecture is at least plausible, since it is given in terms of fromula equivalence. That elimination is in EXPTIME on this assumption may provide sufficient motivation to an expert in algorithmic algebra to verify (or disprove) the conjecture.

# References


[1] G. Collins. Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In *Automata theory and formal languages*, pages 134–183. springer, 1975.

[2] D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties and Algorithms.* Springer, 1992.

[3] J.H. Davenport, Y. Siret, and E. Tournier. *Computer Algebra.* Academic Press, 1988.

[4] D.Yu. Grigoriev. Complexity of deciding Tarski algebra. *J. Symb. Comp.*, 5:65–108, 1988.

[5] W. Kuich and A. Salomaa. *Semirings,Automata,Languages.* Springer-Verlag, 1986.

[6] M. Mignotte. *Mathematics for Computer Algebra.* Springer, 1992.

[7] B. Mishra. *Algorithmic Algebra.* Springer, 1993.

[8] A. Panholzer. Gröbner bases and the defining polynomial of a context-free grammar generating fucntion. theoretical informatics dept. Tech. U. Vienna, 2002.

[9] M. Spivak. *Calculus on Manifolds.* Addison-Wesley, 1965.

[10] A. Tarski. A decision method for elementary algebra and geometry. Technical report, Rand Corp., 1948.




ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'