Comparing real algebraic numbers of small degree

Ioannis Z. Emiris and Elias P. Tsigaridas

Department of Informatics and Telecommunications
National Kapodistrian University of Athens, Greece
{emiris,et}@di.uoa.gr

Abstract. We study polynomials of degree up to 4 over the rationals or a computable real subfield. Our motivation comes from the need to evaluate predicates in nonlinear computational geometry efficiently and exactly. We show a new method to compare real algebraic numbers by precomputing generalized Sturm sequences, thus avoiding iterative methods; the method, moreover handles all degenerate cases. Our first contribution is the determination of rational isolating points, as functions of the coefficients, between any pair of real roots. Our second contribution is to exploit invariants and Bezoutian subexpressions in writing the sequences, in order to reduce bit complexity. The degree of the tested quantities in the input coefficients is optimal for degree up to 3, and for degree 4 in certain cases. Our methods readily apply to real solving of pairs of quadratic equations, and to sign determination of polynomials over algebraic numbers of degree up to 4. Our third contribution is an implementation in a new module of library SYNAPS v2.1. It improves significantly upon the efficiency of certain publicly available implementations: Rioboo's approach on AXIOM, the package of Guibas-Karavelas-Russel [11], and CORE v1.6, MAPLE v9, and SYNAPS v2.0. Some existing limited tests had shown that it is faster than commercial library LEDA v4.5 for quadratic algebraic numbers.

1 Introduction

Our motivation comes from computer-aided geometric design and nonlinear computational geometry, where predicates rely on real algebraic numbers of small degree. These are crucial in software libraries such as ESOLID [15], EXACUS(eg. [12], [2]), and CGAL(eg. [8]). Predicates must be decided exactly in all cases, including degeneracies. We focus on real algebraic numbers of degree up to 4 and polynomials in one variable of arbitrary degree or in 2 variables of degree ≤ 2. Efficiency is critical because comparisons on such numbers lie in the inner loop of most algorithms, including those for computing the arrangement of algebraic curves, arcs or surfaces, the Voronoi diagrams of curved objects, eg. [2, 7, 12, 14] and kinetic data-structures [11].

Our work is also a special-purpose quantifier elimination method for one or two variables and for parametric polynomial equalities and inequalities of low degree. Our approach extends [16, 22] because our rational isolating points eliminate the need of multiple sign evaluations in determining the sign of a univariate polynomial over an algebraic number of degree ≤ 4. We also extend the existing approaches so as to solve some simple bivariate problems (sec. 7).

Our method is based on pre-computed generalized Static sequences; in other words, we implement straight-line programs for each comparison Finding isolating points of low algebraic degree (and rational for polynomials of degree ≤ 4) is a problem of independent interest. It provides starting points for iterative algorithms and has direct applications, e.g. [7]. Our Sturm-based algorithms rely on isolating points in order to avoid iterative methods (which depend on separation bounds) and the explosion of the algebraic degree of the tested quantities. In order to reduce the computational effort, we factorize the various quantities by the use of invariants and/or by the elements of the Bezoutian matrix; for our implementation, this is done in an automated way.

We have implemented a package of algebraic numbers as part of the SYNAPS 2.1 library [6] and show that it compares favorably with other software. Our code can also exist as a stand-alone `C++` software package. We call our implementation S^3 which stands for *Static Sturm Sequences* (or *Salmon-Sturm-Sylvester*).

The following section overviews some of the most relevant existing work. Next, we formalize Sturm sequences. Sect. 4 studies discrimination systems and its connection to the invariants of the polynomial. Sect. 5 obtains rational isolating points for degree ≤ 4. Sect. 6 bounds complexity and sect. 7 applies our tools to sign determination and real solving and indicates some of our techniques for automatic code generation. Sect. 8 illustrates our implementation with experimental results. We conclude with future work.

2 Previous work and contribution

Although the roots of rational polynomials of degree up to 4 can be expressed explicitly with radicals, the computation of the real roots requires square and cubic roots of complex numbers. Even if only the smallest (or largest) root is needed, one has to compute all real roots (cf [13]). Another critical issue is that there is no formula that provides isolating rational points between the real roots of polynomials: this problem is solved in this paper for degree ≤ 4.

In quantifier elimination, an effort was made to optimize low level, operations, eg. [16, 22]. However, by that approach, there is a need for multiple Sturm sequences. By our approach, we need to evaluate only one Sturm sequence in order to decide the sign of a polynomial over a cubic or quartic algebraic number.

Our discrimination system for the quartic is the same as that in [23], but is derived differently and corrects a small error in [23].

Rioboo implemented in AXIOM an arithmetic of real algebraic numbers of arbitrary degree with coefficients from a real closed field [17]. The extension he proposed for the sign evaluation is essentially based upon theorem 2.

Iterative methods based on the approach of Descartes / Uspensky seem to be the fastest means of isolating real roots, in general (cf. [18]). Such methods

sec	rfc	g	pg	rma	$rnma$
S^3	137	2	5	10	34
CORE v1.6	–	10	17	20	195
LEDA v4.5	374	5	11	19	104

Table 1. Computing the arrangement of 300 circular arcs with: random full circles (rfc), circles centered on the crosspoints and the centers of the cells of a square grid of cell size 10^4 (g), the same perturbed (pg), random monotone arcs (rma), random non-monotone arcs (rnma). All the geometry code was done in CGAL and so the first column shows the arithmetic used.

are implemented in SYNAPS. An iterative method using Sturm sequences, has been implemented in [11]. Both methods are tested in sec. 8.

LEDA and CORE[1] evaluate expression trees built recursively from integer operations and $\sqrt[k]{}$, and rely on separation bounds. LEDA treats arbitrary algebraic numbers, by the *diamond operator*, based on Descartes/Uspensky iteration. But it faces efficiency problems ([20]) in computing isolating intervals for degree 3 and 4, since Newton's iteration cannot always be applied with interval coefficients. CORE recently provided for dealing with algebraic numbers using Sturm sequences. Currently this operator cannot handle multiple roots.

Precomputed quantities for the comparison of quadratic algebraic numbers were used in [5], derived from the u-resultant and Descartes' rule of sign. In [14], the same problem was solved with static Sturm sequences, thus improving upon the runtime by up to 10%. In generalizing these methods to higher degree, it is not obvious how to determine the (invariant) quantities to be tested in order to minimize the bit complexity. Another major issue is the isolating points as well as the need of several Sturm sequences.

The contribution of this paper starts with quadratic numbers, for which we reformulated the existing method in order to make it generalizable to higher degree [10]. The efficiency of our implementation for quadratic numbers is illustrated in [8]; we copy table 1 from this paper. For algebraic numbers of degree 3 and 4, preliminary results are in [10, 9], where more details can be found, which cannot fit here for reasons of space. Our novelty is to use a *single* Sturm sequence for degree up to 4 (see cor. 4), first by considering the discrimination system of the given polynomial for purposes of root classification and in order to derive a square-free polynomial defining the algebraic number, second by deriving *rational* isolating points (theorems 6 and 10) and finally by reducing the computational effort through factoring of the tested quantities using invariants and elements of the Bezoutian matrix. Our implementation computes these quantities in an automatic way.

[1] http://www.algorithmic-solutions.com/enleda.htm, http://www.cs.nyu.edu/exact/core

3 Sturm Sequences

Sturm sequences is a well known and useful tool for isolating the roots of any polynomial (cf [24], [1]). Additionally, the reader can refer to [14] where Sturm sequences are used for comparing algebraic numbers of degree 2, or to [5] where the comparison of such numbers was done by the resultant. In the sequel $\mathbf{D}$ is a ring, $\mathbf{Q}$ is its fraction field and $\overline{\mathbf{Q}}$ the algebraic closure of $\mathbf{Q}$. Typically $\mathbf{D} = \mathbb{Z}$ and $\mathbf{Q} = \mathbb{Q}$.

Definition 1. *Let P and $Q \in \mathbf{D}[x]$ be nonzero polynomials. By a* (generalized) Sturm sequence *for P, Q we mean any pseudo-remainder sequence $\overline{P} = (P_0, P_1, \ldots, P_n)$, $n \geq 1$, such that for all $i = 1, \ldots, n$, we have $a_i P_{i-1} = Q_i P_i + b_i P_{i+1}$ ($Q_i \in \mathbf{D}[x], a_i, b_i \in \mathbf{D}$), such that $a_i b_i < 0$ and $P_0 = P, P_1 = Q, P_{n+1} = 0$. We usually write $\overline{P}_{P_0,P_1}$ if we want to indicate the first two terms in the sequence.*

For a Sturm sequence $\overline{P}$, $V_{\overline{P}}(p)$ denotes the number of sign variations of the evaluation of the sequence at p. The last polynomial in $\overline{P}_{P_0,P_1}$ is the resultant of P_0 and P_1.

Theorem 2. *Let $P, Q \in \mathbf{D}[x]$ be relatively prime polynomials and P square-free. If $a < b$ are both non-roots of P and γ ranges over the roots of P in $[a, b]$, then*

$$V_{P,Q}[a,b] := V_{P,Q}(a) - V_{P,Q}(b) = \sum_{\gamma} \operatorname{sign}(P^{'}(\gamma)Q(\gamma)).$$

where $P^{'}$ is the derivative of P.

Corollary 3. *Theorem 2 holds if in place of Q we use $R = \mathrm{PRem}(Q, P)$, where* $\mathrm{PRem}(Q, P)$*, stands for the pseudo-remainder of Q divided by P.*

Finding isolating intervals for the roots of any polynomial is done below. Still, the computation of a Sturm sequence is quite expensive. In order to accelerate the computation we assume that the polynomials are $P, Q \in \mathbf{D}[a_0, \ldots, a_n, b_0, \ldots, b_m][x]$, where a_i, b_j are the coefficients considered as parameters, and we pre-compute various Sturm sequences ($n, m \leq 4$).

The isolating-interval representation of real algebraic number $\alpha \in \overline{\mathbf{Q}}$ is $\alpha \cong (A(X), I)$, where $A(X) \in \mathbf{D}[X]$ is square-free and $A(\alpha) = 0$, $I = [a, b]$, $a, b, \in \mathbf{Q}$ and A has no other root in I.

Corollary 4. *Assume $B(X) \in \mathbf{D}[X] : \beta = B(\alpha)$, and that $\alpha \cong (A, [a, b])$. By theorem 2,* $\operatorname{sign}(B(\alpha)) = \operatorname{sign}(V_{A,B}[a, b] \cdot A^{'}(\alpha))$.

Let us compare two algebraic numbers $\gamma_1 \cong (P_1(x), I_1)$ and $\gamma_2 \cong (P_2(x), I_2)$ where $I_1 = [a_1, b_1]$ and $I_2 = [a_2, b_2]$. Let $J = I_1 \cap I_2$. When $J = \emptyset$, or only one of γ_1 and γ_2 belong to J, we can easily order the 2 algebraic numbers. All these tests are implemented by the previous corollary and theorem. If $\gamma_1, \gamma_2 \in J$, then $\gamma_1 \geq \gamma_2 \Leftrightarrow P_2(\gamma_1) \cdot P_2^{'}(\gamma_2) \geq 0$. We can easily obtain the sign of $P_2^{'}(\gamma_2)$, and from theorem 2, we obtain the sign of $P_2(\gamma_1)$.

4 Root classification

Before applying the algorithms for root comparison, we analyze each polynomial by determining the number and the multiplicities of its real roots. For this, we use a system of discriminants. For the quadratic polynomial the discrimination system is trivial. For the cubic, it is well known [22, 10]. We study the quartic by Sturm-Habicht sequences, while [23] used a resultant-like matrix. For background see [24, 1]. We factorize the tested quantities by invariants and elements of the Bezoutian matrix. We use invariants in order to provide square-free polynomials defining the algebraic numbers, to compute the algebraic numbers as rationals if this is possible and finally to provide isolating rationals.

Consider the quartic polynomial equation, where $a > 0$.

$$f(X) = aX^4 - 4bX^3 + 6cX^2 - 4dX + e. \tag{1}$$

In the entire paper, we consider as input the coefficients $a, b, c, d, e \in \mathbf{D}$.

For background on invariants see [4], [19]. We consider the rational invariants of f, i.e the invariants in $GL(2, \mathbb{Q})$. They form a graded ring [4], generated by:

$$A = W_3 + 3\Delta_3, \quad B = -dW_1 - e\Delta_2 - c\Delta_3. \tag{2}$$

Every other invariant is an isobaric polynomial in A and B, i.e. it is homogeneous in the coefficients of the quartic. We denote the invariant $A^3 - 27B^2$ by Δ_1 and refer to it as the *discriminant*. The semivariants (which are the leading coefficients of the covariants) are A, B and:

$$\Delta_2 = b^2 - ac, R = aW_1 + 2b\Delta_2, Q = 12\Delta_2^2 - a^2 A. \tag{3}$$

We also derived the following quantities, which are not necessarily invariants but they are elements of the Bezoutian matrix of f and $f^{'}$. Recall that the determinant of the Bezoutian matrix equals the resultant, but its size is smaller than the Sylvester matrix [3].

$$\begin{array}{llll} \Delta_3 &= c^2 - bd & T &= -9W_1^2 + 27\Delta_2\Delta_3 - 3W_3\Delta_2 \\ \Delta_4 &= d^2 - ce & T_1 &= -W_3\Delta_2 - 3\,W_1^2 + 9\,\Delta_2\Delta_3 \\ W_1 &= ad - bc & T_2 &= AW_1 - 9\,bB \\ W_2 &= be - cd & W_3 &= ae - bd \end{array} \tag{4}$$

Proposition 5. *[23] Let $f(X)$ be as in (1). The table gives the real roots and their multiplicities. In case (2) there are 4 complex roots, while in case (8) there are 2 complex double roots (In [23] there is a small error in defining T).*

(1) $\Delta_1 > 0 \wedge T > 0 \wedge \Delta_2 > 0$	$\{1, 1, 1, 1\}$
(2) $\Delta_1 > 0 \wedge (T \leq 0 \vee \Delta_2 \leq 0)$	$\{\}$
(3) $\Delta_1 < 0$	$\{1, 1\}$
(4) $\Delta_1 = 0 \wedge T > 0$	$\{2, 1, 1\}$
(5) $\Delta_1 = 0 \wedge T < 0$	$\{2\}$
(6) $\Delta_1 = 0 \wedge T = 0 \wedge \Delta_2 > 0 \wedge R = 0$	$\{2, 2\}$
(7) $\Delta_1 = 0 \wedge T = 0 \wedge \Delta_2 > 0 \wedge R \neq 0$	$\{3, 1\}$
(8) $\Delta_1 = 0 \wedge T = 0 \wedge \Delta_2 < 0$	$\{\}$
(9) $\Delta_1 = 0 \wedge T = 0 \wedge \Delta_2 = 0$	$\{4\}$

5 Rational isolating points

In what follows, $f \in \mathbb{Z}[X]$ and $a > 0$; the same methods work for any computable real subfield **D**. It is known that for the quadratic $f(X) = a\,X^2 - 2\,b\,X + c$, the rational number $\frac{b}{a}$, isolates the real roots.

Theorem 6. *Consider the cubic $f(X) = a\,X^3 - 3\,b\,X^2 + 3\,c\,X - d$. The rational numbers $\frac{b}{a}$ and $-\frac{W_1}{2\Delta_2}$ isolate the real roots.*

Proof. In [10], we derive the rational isolating points based on the fact that the two extreme points and the inflexion point of $f(X)$ are colinear. Moreover, the line through these points intersects the $x-$axis at a rational number. Notice that the algebraic degree of the isolating points is at most 2. Interestingly, the same points are obtained by applying theorem 7. $\square$

Theorem 7. [21] *Given a polynomial $P(X)$ with adjacent real roots γ_1, γ_2, and any two other polynomials $B(X)$, $C(X)$, let $A(X) := B(X)P'(X) + C(X)P(X)$ where P' is the derivative of P. Then $A(X)$ or $B(X)$ are called* isolating polynomials *because at least one of them has at least one real root in the closed interval $[\gamma_1, \gamma_2]$. In addition, it is always possible to have* $\deg A + \deg B \leq \deg P - 1$.

We now study the case of the quartic. By theorem 7 it is clear how to isolate the roots by 2 quadratic algebraic numbers and a rational. In order to obtain an isolating polynomial, let $B(X) = ax - b$ and $C(X) = -4a$ then

$$A(X) = 3\,\Delta_2\,X^2 + 3\,W_1\,X - W_3. \tag{5}$$

Since $\frac{b}{a}$ is the arithmetic mean of the 4 roots, it is certainly somewhere between the roots. The other two isolating points are the solutions of (5), i.e

$$\sigma_{1,2} = \frac{-3W_1 \pm \sqrt{9W_1^2 + 12\Delta_2 W_3}}{6\Delta_2}. \tag{6}$$

We verify that $\operatorname{sign}\left(f(\frac{b}{b})\right) = \operatorname{sign}\left(a^2 A - 3\Delta_2^2\right)$, so

$$\begin{cases} \sigma_1 < \frac{b}{a} < \sigma_2, \text{ if } f(\frac{b}{a}) > 0; \\ \sigma_1 < \sigma_2 < \frac{b}{a}, \text{ if } f(\frac{b}{a}) < 0 \ \wedge \ R > 0; \\ \frac{b}{a} < \sigma_1 < \sigma_2, \text{ if } f(\frac{b}{a}) < 0 \ \wedge \ R < 0; \end{cases} \tag{7}$$

where R is from (3). If $f(\frac{b}{a}) = 0$ then we know exactly one root and can express the other three roots as roots of a cubic. To obtain another isolating polynomial, we use $B(X) = dx - e, C(X) = -4d$, and

$$A(X) = W_3\,X^3 - 3\,W_2\,X^2 - 3\,\Delta_4\,X.$$

By the theorem at least 2 of the numbers below separate the roots.

$$0, \tau_{1,2} = \frac{3W_2 \pm \sqrt{9W_2^2 + 12\Delta_4 W_3}}{6W_3}. \tag{8}$$

We assume that the roots are > 0, so 0 is not an isolating point. The order of the isolating points is determined similarly as in (7). Let us now find rational isolating points for all relevant cases of prop. 5.

$\{1,1,1,1\}$ Treated below.
$\{1,1\}$ Since $\{1,1,1,1\}$ is harder, we do not examine it explicitly.
$\{2,1,1\}$ The double root is rational since it is the only root of $\mathrm{GCD}(f, f')$ and its value is $\frac{T_1}{T_2}$, see eq (4). In theory, we could divide it out and use the isolating points of the cubic, but in practice we avoid division. When the double root is the middle root then $\frac{b}{a}$ and $-\frac{W_1}{2\Delta_2}$ are isolating points, otherwise we use theorem 7 to find one more isolating point in $\mathbb{Q}$.
$\{2\}$ Compute the double root from $\overline{P}_{f,f'}$; it is rational as a root of $\mathrm{GCD}(f, f')$.
$\{2,2\}$ The roots are the smallest and largest root of the derivative i.e. a cubic. Alternatively, we express them as the roots of $3\Delta_2 X^2 + 3W_1 X - W_3$.
$\{3,1\}$ The triple root is $-\frac{W_1}{2\Delta_2}$ and the single root is $\frac{3\,aW_1+8\,b\Delta_2}{2\,a\Delta_2}$.
$\{4\}$ The one real root is $\frac{b}{a} \in \mathbb{Q}$.

It remains to consider the case where the quartic has 4 simple real roots. We assume that 0 is not a root (otherwise we deal with a cubic), therefore, $e \neq 0$. WLOG, we may consider equation (1) with $b = 0$. Then, specialize equations (6) and (8) using $b = 0$. The only difficult case is when τ_i and σ_j, $i, j \in \{1, 2\}$, isolate the same pair of adjacent roots. WLOG, assume that these are τ_1, σ_1. We combine them by the following lemma.

Lemma 8. *For any $m, n, m', n' \in \mathbb{N}^*$, $0 < \frac{m}{n} < \frac{m'}{n'} \Rightarrow \frac{m}{n} < \frac{m+m'}{n+n'} < \frac{m'}{n'}$.*

$$\mathcal{A} := 9\Delta_4 - 3\,ce, \quad \mathcal{B} := 12\,ae\Delta_4 + 9\,d^2c^2 \tag{9}$$

then, an isolating point is $\frac{3d-3dc+\sqrt{\mathcal{A}}+\sqrt{\mathcal{B}}}{6c+2ae}$. If we find an integer $K \in [\sqrt{\mathcal{A}}, \sqrt{\mathcal{B}}]$, then it suffices to replace $\sqrt{\mathcal{A}} + \sqrt{\mathcal{B}}$ by $2K$ and we denote the resulting rational by $\sigma_i \oplus \tau_j$; notice it has degree 2 in the input coefficients. By prop. 5(1), $\Delta_2 > 0 \Rightarrow c < 0$. Descartes' rule implies that, if $e > 0$, then there are 2 positive and 2 negative roots, while $e < 0$ means there are 3 positive and one negative root or vice versa. We set $K = \lceil \sqrt{\mathcal{A}} \rceil$ to prove theorem 10, provided the following holds:

Theorem 9. *For every quartic in $\mathbb{Z}[X]$ with 4 distinct real roots and $b = 0$, we have $\sqrt{\mathcal{B}} - \sqrt{\mathcal{A}} \geq 1$, using notation (9).*

Proof.

$$\sqrt{\mathcal{B}} \geq 1 + \sqrt{\mathcal{A}} \Leftrightarrow \sqrt{\tfrac{\mathcal{B}}{\mathcal{A}}} \geq 1 + \tfrac{1}{\sqrt{\mathcal{A}}} \Leftarrow \sqrt{\tfrac{\mathcal{B}}{\mathcal{A}}} \geq 2 \;\Leftrightarrow$$
$$g := 4aed^2 - 4ace^2 + 3d^2c^2 - 12d^2 + 16ce \geq 0.$$

First we show that the minimum of $g(a, c, d, e)$ is positive, subject to $-a \leq 1$, $c \leq -5$, and $-e \leq -5$; we treat the case where $c > -5$ and $e < 5$ later. We introduce slack variables y_1, y_2, y_3 and use Lagrange multipliers. So our problem now is

$$\begin{aligned} &\min\ L(a,c,d,e,y_1,y_2,y_3,\lambda_1,\lambda_2,\lambda_3) := \\ &\min\ \left[g(c,e)+\lambda_1(c+y_1^2+5)\lambda_2(-e+y_2^2+5)+\lambda_3(-a+y_3^2+1)\right] \end{aligned} \tag{10}$$

We take partial derivatives,, equate them to zero and the solution of the system, by MAPLE 9, is $(a,c,d,e) = (1,-5,0,5)$ and $g(1,-5,0,5) = 300 > 0$ which is a local minimum. If $-5 < c < 0$ and $0 < e < 5$ we check exhaustively that $\sqrt{\mathcal{B}} - \sqrt{\mathcal{A}} \geq 1$. If $e < 0$ then we use again Lagrange multipliers but with the constraint $e + 1 - y_2^2$. ☐

Theorem 10. *Consider a quartic as in (1). At least three of the rational numbers* $\{0, \frac{b}{a}, \frac{e}{d}, \sigma_i \oplus \tau_j\}$ *isolate the real roots,* $i, j \in \{1, 2\}$.

6 Complexity of computations

The comparison of the roots of two polynomials of degree $d \leq 4$ using Sturm sequences and isolating intervals provides the following bounds in the degree of the tested quantities. We measure degree in terms of the input quartics' coefficients. A lower bound is the degree of the resultant coefficients, which is $2\,d$ in terms of the input coefficients. Recall that the resultant is an irreducible polynomial in the coefficients of an overconstrained system of $n+1$ polynomials in n variables, the vanishing of which is the minimum condition of solvability of the system ([24], [1]).

There is a straightforward algorithm for the comparison of quadratic algebraic numbers, with maximum algebraic degree 4, hence optimal ([14], [5]).

Theorem 11. [10] *There is an algorithm for the comparison of algebraic cubic numbers (including all degenerate cases), with maximum algebraic degree 6, hence optimal.*

Theorem 12. *There is an algorithm that compares any two roots of two square-free quartics with algebraic degree 8 or 9, depending on the degree of the isolating points. When the quartics are not square-free, the algebraic degree is between 8 and 13. The algorithm needs at most 172 additions and multiplications in order to decide the order. These bounds cover all degenerate cases, including when one polynomial drops degree.*

Proof. (*Sketch*) In [10] we derive this bound by considering the evaluation of the Sturm sequence polynomials over the isolating points (see the discussion after cor. 4 for additional details). The isolationg points have degree at most 2. The length of the sequence is at most 6 and so we need at most 12 polynomial evaluations, hence a fixed number of operations. The number of operations is obtained by MAPLE's function `cost`. ☐

7 Applications

We have implemented a software package, including a `root_of` class for algebraic numbers of degrees up to 4 as part of library SYNAPS (v2.1) [6]. Some functionalities pertain to higher degrees. Our implementation is generic in the sense that it can be used with any number type and any polynomial class that supports elementary operations and evaluations. It can handle all degenerate cases and has been extended to arbitrary degree (though without isolating points for now). We developed programs that produce all possible sign combinations of the tested quantities, allow us to test as few quantities as possible, and produce both `C++` code and pseudo-code for the comparison and sign determination functions. We provide the following functionalities, where `UPoly` and `BPoly` stand for arbitrary univariate and quadratic bivariate polynomial respectively:

`Sturm(UPoly` f_1`, UPoly` f_2`)` Compute the Sturm sequence of f_1, f_2, by pseudo-remainders, (optimized) subresultants, or Sturm-Habicht.

`compare(root_of` α`, root_of` β`)` Compare 2 algebraic numbers of degree ≤ 4 using precomputed Sturm sequences. We have precomputed all the possible sign variations so as to be able to decide with the minimum number of tests (and faster than with Sturm-Habicht sequences).

`sign_at(UPoly` f`, root_of` α`)` Determination of the sign of a univariate polynomial of arbitrary degree, over an algebraic number of degree ≤ 4. We use the same techniques, as in `compare`.

`sign_at(BPoly` f`, root_of` γ_x`, root_of` γ_y`)` $\gamma_1 \cong (P_1(x), I_1)$ and $\gamma_2 \cong (P_2(x), I_2)$, where $I_1 = [a_1, b_1]$ and $I_2 = [a_2, b_2]$, are of degrees ≤ 4. We compute the Sturm-Habicht sequence of P_1 and f wrt X. We specialize the sequence for $X = a_1, X = a_2$, and find the sign of all y-polynomials over γ_y as above. The Sturm-Habicht sequence was computed in MAPLE, using the elements of the Bezoutian matrix. Additionally we used the packages `codegeneration, optimize`, in order to produce efficient `C++` code.

`solve(UPoly` f`)` Returns the roots of f as `root_of`. If f has degree ≤ 4 we use the discrimination system, otherwise we use non-static Sturm sequences.

`solve(BPoly` f_1`, BPoly` f_2`)` We consider the resultants R_x, R_y of f_1, f_2 by eliminating y and x respectively, thus obtaining degree-4 polynomials in x and y. The isolating points of R_x, R_y define a grid of boxes, where the intersection points are located. The grid has 1 to 4 rows and 1 to 4 columns. It remains to decide, for boxes, whether they are empty and, if not, whether they contain a simple or multiple root. Multiple roots of the resultants are either rational or quadratic algebraic numbers by prop. 5. Each box contains at most one intersection point. We can decide if a box is empty by 2 calls to the bivariate `sign_at` function. We can do a significant optimization by noticing that the intersections in a column (row) cannot exceed 2 nor the multiplicity of the algebraic number and thus excluding various boxes.

Unlike [2], where the boxes cannot contain any critical points of the intersecting conics, our algorithm does not make any such assumption, hence there is no need to refine them. Our approach can be extended in order to compute intersection points of bivariate polynomials of arbitrary degree, provided that

msec	$\mathbb{Z}$	$\mathbb{Q}$	far	M$-\mathbb{Q}$	M	$\mathbb{Z}$	$\mathbb{Q}$	far	M$-\mathbb{Q}$	M
AXIOM	38.0	57.4	77.3	67.2	82.3					
MAPLE 9	33.2	52.4	69.0	75.5	74.7					
CORE	8.41	5.67	6.76	9.31	10.1					
SYNAPS(gmpq)	1.097	0.820	0.596	1.480	2.114	2.687	2.693	2.780	3.764	5.698
[11](gmpq)	1.249	0.921	0.991	1.582	1.544	23.74	64.4	7.3	4.121	54.7
[11]-FILT(gmpq)	0.346	0.301	0.279	0.313	0.320	1.594	2.130	1.035	2.758	3.980
S^3 (gmpq)	0.077	0.083	0.082	0.074	0.077	0.117	0.190	0.115	0.161	0.129
SYNAPS(gmp mpf)	0.302	0.202	0.195	0.339	0.385					
S^3 (gmp mpf)	0.060	0.064	0.057	0.063	0.061					
SYNAPS(double)	0.055	0.042	0.043	0.061	0.071					
S^3 (double)	0.010	0.011	0.012	0.010	0.010					

Table 2. Running times in *msec* for comparing specific roots of 2 quartics. In parentheses are the number types: `gmpq` / `gmp mpf` stand for GMP rationals / floating point.

we obtain isolating points for the roots of the two resultants, either statically (as above) or dynamically.

8 Experimental results

We performed tests on a 2.6GHz Pentium with 512MB memory. The results are on table 2. AXIOM refers to the implementation of real algebraic arithmetic by Rioboo (current CVS version); it is meant, like MAPLE 9, only for a rough comparison. The package in [11] uses subdivision based on Sturm sequences. Row [11]-FILT is the same but the program tries to decide with `double` arithmetic and, if it cannot, then switches to exact arithmetic. S^3 is our code implemented in SYNAPS 2.1. Every test is averaged over 10000 polynomial pairs. The left part of the table includes results for polynomials with 4 random real roots between -20 and 20, multiplied by an integer $\in [1, 20]$, so the coefficients are integers of absolute value $\leq 32 \cdot 10^5$ and the Mignotte polynomials are $a(x^4 - 2(Lx-1)^2)$, where a, L are chosen randomly from $[3, 30]$. The right part of the table tests polynomials with random roots $\in [10 \cdot 10^4, 11 \cdot 10^4]$ and Mignotte polynomials with parameters a, L both chosen randomly from $[10 \cdot 10^4, 11 \cdot 10^4]$ with uniform distribution.

Column $\mathbb{Z}$ indicates polynomials with 4 integer roots. Column $\mathbb{Q}$ indicates polynomials with 4 rational roots. Column far indicates that one polynomial has only negative roots while the other one has only positive ones. $M - \mathbb{Q}$ indicates that one is a Mignotte polynomial and the other has 4 rational roots. M indicates that we compare the roots of two Mignotte polynomials.

CORE cannot handle multiple roots. This is also the case for the iterative solver of SYNAPS, which also has problems when the roots are endpoints of a subdivision. MAPLE cannot always handle equality for the roots of two Mignotte polynomials. Of course all the implementations have problems when the number type is not an exact one. This is the case of `double` and `gmp mpf`. By considering the left part of table 2, our code is 103, 15 and 16 times faster than CORE, SYNAPS and [11], respectively, in the average case and when no filtering is used. Even in case of filtering ([11]-FILT) our code is faster by a factor of 4 on average. Increasing the magnitude of the coefficients or decreasing the separation bound of the roots, leads to even more dramatic improvement in the efficiency of our implementation over the other ones; this is the case for the right part of table 2.

S^3 has similar running times for all kinds of tests and handles degenerate cases faster that the general ones since we use the discrimination system. This is not the case for any of the other approaches. The running times of our code with an exact number type is less than 8 times slower than when used with doubles. However the latter does not offer any guarantee for the computed result. This is an indication that exact arithmetic imposes a reasonable overhead on efficiency when used in conjunction with efficient algorithms and carefully implemented.

9 Future work

Consider the quintic $ax^5 + 10\,cx^3 - 10\,dx^2 + 5\,ex - f$, with $a > 0$ and $c < 0$ by assuming 5 real roots. Using the techniques above, we obtain 2 pairs of isolating polynomials. Two of them are

$$B_1 = 2\,X^2ca + 3\,dXa + 8\,c^2 \quad B_2 = \left(-4\,df + 4\,e^2\right) X^2 + feX - f^2 \tag{11}$$

One pair of roots may be separated by the roots of B_1, B_2. We combine them by finding an integer between the roots of the respective discriminants: $\Delta_{B_1} = a(9d^2a - 64c^3)$, $\Delta_{B_2} = (17e^2 - 16df)f^2$. It suffices to set K to $2\lceil \Delta_{B_1} \rceil$ or $2\lceil \Delta_{B_2} \rceil$, depending on the signs of e, f. Tests with MAPLE support our choices.

Filtering should lead to running times similar to those of the double number type. Additionally we are planning to compare our software against the software of [18] and NiX, the polynomial library of EXACUS.

The algebraic degree of the resultant is a tight lower bound in checking solvability, but is it tight for comparisons?

Acknowledgments. Both authors acknowledge inspirational comments by R. Rioboo, partial support by INRIA's project "CALAMATA", a bilateral collaboration between the GALAAD group of INRIA Sophia-Antipolis (France) and the National Kapodistrian University of Athens, and by PYTHAGORAS, a project under the EPEAEK program of the Greek Ministry of Education and Religions.

References

1. S. Basu, R. Pollack, and M-F.Roy. *Algorithms in Real Algebraic Geometry*, volume 10 of *Algorithms and Computation in Mathematics*. Springer-Verlag, 2003.
2. E. Berberich, A. Eigenwillig, M. Hemmer, S. Hert, K. Mehlhorn, and E. Schomer. A computational basis for conic arcs and boolean operations on conic polygons. In *ESA*, volume 2461 of *LNCS*, pages 174–186. Springer-Verlag, 2002.

3. P. Bikker and A. Y. Uteshev. On the Bézout construction of the resultant. *J. Symbolic Computation*, 28(1–2):45–88, July/Aug. 1999.
4. J. E. Cremona. Reduction of binary cubic and quartic forms. *LMS J. Computation and Mathematics*, 2:62–92, 1999.
5. O. Deviller, A. Fronville, B. Mourrain, and M. Teillaud. Algebraic methods and arithmetic filtering for exact predicates on circle arcs. *Comp. Geom: Theory & Appl., Spec. Issue*, 22:119–142, 2002.
6. G. Dos Reis, B. Mourrain, R. Rouillier, and P. Trébuchet. An environment for symbolic and numeric computation. In *Proc. of the International Conference on Mathematical Software 2002*, World Scientific, pages 239–249, 2002.
7. L. Dupont, D. Lazard, S. Lazard, and S. Petitjean. Near-optimal parameterization of the intersection of quadrics. In *Proc. Annual ACM Symp. on Comp. Geometry*, pages 246–255. ACM, June 2003.
8. I. Emiris, A. Kakargias, M. Teillaud, E. Tsigaridas, and S. Pion. Towards an open curved kernel. In *Proc. Annual ACM Symp. on Computational Geometry*, pages 438–446, New York, 2004. ACM Press.
9. I. Z. Emiris and E. P. Tsigaridas. Comparison of fourth-degree algebraic numbers and applications to geometric predicates. Tech. Rep ECG-TR-302206-03, INRIA Sophia-Antipolis, 2003.
10. I. Z. Emiris and E. P. Tsigaridas. Methods to compare real roots of polynomials of small degree. Tech. Rep ECG-TR-242200-01, INRIA Sophia-Antipolis, 2003.
11. L. Guibas, M. Karavelas, and D. Russel. A computational framework for handling motion. In *Proc. 6th Workshop (ALENEX)*, Jan. 2004. To appear.
12. M. Hemmer, E. Schömer, and N. Wolpert. Computing a 3-dimensional cell in an arrangement of quadrics: Exactly and actually! In *Proc. Annual ACM Symp. Comput. Geometry*, pages 264–273, 2001.
13. D. Kaplan and J. White. Polynomial equations and circulant matrices. *The Mathematical Association of America (Monthly)*, 108:821–840, November 2001.
14. M. Karavelas and I. Emiris. Root comparison techniques applied to the planar additively weighted Voronoi diagram. In *Proc. Symp. on Discrete Algorithms (SODA-03)*, pages 320–329, Jan. 2003.
15. J. Keyser, T. Culver, D. Manocha, and S. Krishnan. ESOLID: A system for exact boundary evaluation. *Comp. Aided Design*, 36(2):175–193, 2004.
16. D. Lazard. Quantifier elimination: optimal solution for two classical examples. *J. Symb. Comput.*, 5(1-2):261–266, 1988.
17. R. Rioboo. Real algebraic closure of an ordered field: implementation in axiom. In *Proc. Annual ACM ISSAC*, pages 206–215. ACM Press, 1992.
18. F. Rouillier and P. Zimmermann. Efficient isolation of a polynomial real roots. Technical Report 4113, INRIA–Lorraine, 2001.
19. G. Salmon. *Lessons Introductory to the Modern Higher Algebra*. Chelsea Publishing Company, New York, 1885.
20. S. Schmitt. The diamond operator for real algebraic numbers. Technical Report ECG-TR-243107-01, MPI Saarbrücken, 2003.
21. T. W. Sederberg and G.-Z. Chang. Isolating the real roots of polynomials using isolator polynomials. In C. Bajaj, editor, *Algebraic Geometry and Applications*. Springer, 1993.
22. V. Weispfenning. Quantifier elimination for real algebra–the cubic case. In *Proc. Annual ACM ISSAC*, pages 258–263. ACM Press, 1994.
23. L. Yang. Recent advances on determining the number of real roots of parametric polynomials. *J. Symbolic Computation*, 28:225–242, 1999.
24. C. Yap. *Fundamental Problems of Algorithmic Algebra*. Oxford Univ. Press, 2000.