Real algebraic numbers and polynomial systems of small degree ⋆

Ioannis Z. Emiris [1], Elias P. Tsigaridas [1]

Department of Informatics and Telecommunications, National Kapodistrian University of Athens, HELLAS

Abstract

We present exact and complete algorithms based on precomputed Sturm-Habicht sequences, discriminants and invariants, that classify, isolate with rational points and compare the real roots of polynomials of degree up to 4. We have closed formulas for all isolating points. Moreover we combine these results with a simple version of rational univariate representation so as to isolate and compute the multiplicity of all common real roots of a bivariate system of integer polynomials of total degree ≤ 2. We present our implementation within SYNAPS and we perform experimentation and comparison with all available software. Our package is 2–10 times faster, even when compared to inexact software or to sofware with intrinsic filtering.

Key words: algebraic number, real solving, bivariate polynomial, quartic, Sturm sequence, discriminant

1 Introduction

Lazard (1988) derived the necessary condition under which a quartic polynomial takes only positive values and the necessary condition under which an ellipse lies inside a unit circle. That paper, by providing optimal solutions,

⋆ This paper is the final presentation of the results in (Emiris and Tsigaridas, 2004a,b)

Email addresses: emiris@di.uoa.gr (Ioannis Z. Emiris), et@di.uoa.gr (Elias P. Tsigaridas).

[1] Partially supported by "CALAMATA", a bilateral collaboration between the GALAAD group of INRIA and the Department of Informatics of National University of Athens, and "PYTHAGORAS", project no. 70/3/7392, funded by the Greek Ministry of National Education and the European Union.

showed that general purpose algorithms do not always provide optimal solutions (algorithms) to specific problems. Inspired by the approach of D. Lazard, we tackle the problem of enumerating, isolating and comparing the real roots of integer polynomials of degree up to 4, in an efficient way leading to the best available implementation. Moreover, using these results, we derive an efficient algorithm (and implementation) for isolating in rational boxes all common real roots of systems of bivariate integer polynomials of total degree up to 2. For each root we also output its multiplicity.

An important application of these special-purpose algorithms 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 (Keyser et al., 2004), EXACUS (Hemmer et al., 2001; Eigenwillig et al., 2004), and the upcoming curved kernel of CGAL (Emiris et al., 2004). Predicates must be decided exactly in all cases, including degeneracies. Efficiency is critical because comparisons of algebraic numbers and solution of polynomial systems of small degree, lie in the inner loop of several geometric algorithms, including those for computing the arrangement of algebraic curves, arcs or surfaces, the Voronoi diagrams of curved objects, eg. (Eigenwillig et al., 2004; Dupont et al., 2003; Lazard et al., 2004; Hemmer et al., 2001; Karavelas and Emiris, 2003) and related kinetic data-structures (Guibas et al., 2004).

Our work also provides a special-purpose quantifier elimination method for one or two variables and for parametric polynomial equalities and inequalities of low degree. Our approach extends the one of Weispfenning (1994) because the 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 quadratic bivariate polynomial systems (sec. 6).

Our method is based on pre-computed (static) Sturm sequences; essentially, we implement straight-line programs for each computation. One contribution is finding isolating points of low algebraic degree (and rational for polynomials of degree ≤ 4) which is a problem of independent interest. It provides starting points for iterative algorithms and has direct applications, e.g. (Dupont et al., 2003; Lazard et al., 2004). The isolating points are given as functions of the coefficients of the polynomial. 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, via MAPLE.

We have implemented a package of algebraic numbers and bivariate polyno-

mial system solving and show that it compares favorably with other software. Our implementation is part of the SYNAPS [2] v2.1 library (Dos Reis et al., 2002), which is an open source library for symbolic and numeric computations. 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 as well as our main contributions. Next, we formalize Sturm sequences, the representation of algebraic numbers and the algorithms for comparison. Sect. 4 studies discrimination systems, their connection to the invariants of the polynomial and to root classification. Sect. 5 obtains rational isolating points for polynomials of degree ≤ 4 and bounds the algebraic degree and the number of operations for the comparison. Sect. 6 applies our tools to solve a system of bivariate polynomials of total degree up to 2. Sect. 7 sketches our implementation and sect. 8 illustrates our implementation with experimental results. Future work is mentioned throughout.

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 (Kaplan and White, 2001). Our approach allows us to isolate and determine the multiplicity of a specific root of a polynomial, without computing all the roots. 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, there are seminal works that optimize low level, operations, eg. (Lazard, 1988; Weispfenning, 1994). However, by those approaches, the comparison of real algebraic numbers requires 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, or to compare two such numbers.

Rioboo (1992), implemented real closure in AXIOM, an arithmetic of real algebraic numbers of arbitrary degree with coefficients from a real closed field, which is the only package that can handle non-trivial examples. The extension, Rioboo (2002), proposed for the sign evaluation, is essentially the same as theorem 1.

[2] www-sop.inria.fr/galaad/logiciels/synaps/

Iterative methods based on the approach of Descartes / Uspensky seem to be the fastest means of isolating real roots, in general (Rouillier and Zimmermann, 2003). Such a method, based on the Bernstein basis, is implemented in SYNAPS 2.1 (Mourrain et al., 2002). An iterative method that uses subdivision and Sturm sequences, has been implemented by Guibas et al. (2004). The latter two methods are tested in Sec. 8, since their source code is available.

LEDA and CORE [3] 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 and Netwon's method (Schmitt, 2003). But it faces efficiency problems in computing isolating intervals for degree 3 and 4, since Newton's iteration cannot always be applied with interval coefficients. CORE recently provided a `rootOf` operator for dealing with algebraic numbers using Sturm sequences.

Precomputed quantities for the comparison of quadratic algebraic numbers were used by Karavelas and Emiris (2003), with static Sturm sequences. 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. Here we settle these problems.

The basis of our work are the discrimination systems, which are the same as in (Yang, 1999), but they are derived differently and we also correct a small typographical error concerning the quartic. For a polynomial of degree up to 4, we use the quantities involved in its discrimination system not only to determine the number of its roots, but also to compute their multiplicity, to express them as rationals when this is possible, to compute the polynomial's square-free part and to provide rational points that isolate its roots. The derivation of rational isolating points, allows us to compare two algebraic numbers using a *single* Sturm-Habicht sequence (see theorem 1).

For quadratic numbers and for the efficiency of our implementation see (Emiris et al., 2004). For algebraic numbers of degree 3 and 4, preliminary results are in (Emiris and Tsigaridas, 2003, 2004a,b), where details can be found, which cannot fit here for reasons of space. Here we compare our software with the univariate solver of SYNAPS (Mourrain et al., 2002), GKR of (Guibas et al., 2004), RS (Rouillier and Zimmermann, 2003) [4], CORE, and NiX, the polynomial library of EXACUS [5]. Our software is 2–10 times faster, even compared to software packages that have intrinsic filtering.

Solving polynomial systems in a real field is an active area of research. There

[3] www.algorithmic-solutions.com/enleda.htm, www.cs.nyu.edu/exact/core
[4] http://fgbrs.lip6.fr/∼rouillie/Software/RS
[5] www.mpi-sb.mpg.de/projects/EXACUS/

are several algorithms that tackle this problem, c.f. Basu et al. (2003) and references therein. In order to solve exactly and efficiently quadratic bivariate polynomial systems, without the assumption of generic position, we precompute resultants and Sturm-Habicht sequences in two variables and we combine the rational isolating points with a simple version of rational univariate representation.

For real-solving of bivariate systems, we performed experiments against existing solvers in SYNAPS, that is NEWMAC, which is based on normal forms (Mourrain and Trébuchet, 2002), STH, which is based on the work of Gonzalez Vega and Necula (2002), RES, which is based on computing the generalized eigenvalues of a Bézoutian matrix (Busé et al., 2005). Additionally we test against GBRS [6], through its MAPLE interface, which uses Groebner bases and rational univariate representation (Rouillier, 1999). In short, our software is 2–10 times faster, even compared to inexact software.

3 Sturm Sequences and real algebraic numbers

Sturm sequences is a well known and useful tool for isolating the roots of any polynomial. Additionally, the reader can refer to (Karavelas and Emiris, 2003) where Sturm sequences are used for comparing algebraic numbers of degree 2. 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 $\mathbf{Q} = \mathbb{Q}$. Let $V_{P_1,P_2}(a)$ denote the number of sign variations of the evaluation of the Sturm sequence of polynomials P_1 and P_2, over a.

Theorem 1 *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*

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

where P' is the derivative of P. The theorem also holds if we replace Q by $R = \text{PRem}(Q, P)$, where $\text{PRem}(Q, P)$, stands for the pseudo-remainder of Q divided by P.

For a proof see (Basu et al., 2003) or (Rioboo, 2002). Actually th. 1 expresses the computation of the Cauchy index of P and Q, over the interval $[a, b]$.

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.

[6] http://fgbrs.lip6.fr

Let $B(X) \in \mathbf{D}[X]$ and a real algebraic number $\beta = B(\alpha)$, where $\alpha \cong (A, [a, b])$. By theorem 1, $\text{sign}(B(\alpha)) = \text{sign}(V_{A,B}[a, b] \cdot A^{'}(\alpha))$.

Here is our method to 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 theorem 1. 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 1, we obtain the sign of $P_2(\gamma_1)$. This approach is similar to (Rioboo, 2002).

4 Root classification

We analyze each given 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 (Weispfenning, 1994). We study the quartic by Sturm-Habicht sequences, while Yang (1999) used a resultant-like matrix. For background refer to Basu et al. (2003). 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 $a, b, c, d, e \in \mathbf{D}$ and $a > 0$:

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

For background on invariants see Cremona (1999) and Salmon (1885). We consider the rational invariants of f, i.e the invariants in $GL(2, \mathbb{Q})$. They form a graded ring (Cremona, 1999), generated by, $A = W_3 + 3\Delta_3$ and $B = -dW_1 - e\Delta_2 - c\Delta_3$. Every other invariant is an isobaric polynomial in A and B, i.e. it is homogeneous in the coefficients of the quartic. Let $\Delta_1 = A^3 - 27B^2$ be the *discriminant*. The semivariants (which are the leading coefficients of the covariants) are A, B and $\Delta_2 = b^2 - ac$, $R = aW_1 + 2b\Delta_2$ and $Q = 12\Delta_2^2 - a^2A$. We also derived the following quantities, which are not necessarily invariants but they are elements of the Bézoutian matrix of f and $f^{'}$.

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

Since our discrimination system is based on Sturm-Habicht sequence, basically on the principal subresultant coefficients, we use the Bézoutian matrix to compute them in a symbolic way, because its size is smaller than the Sylvester matrix. Extending this approach to higher degree is a work in progress.

In (Yang, 1999) there is a small typographical error in defining T.

Proposition 2 *(Yang, 1999) 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*

(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 and bit complexity of comparison

In what follows, $f \in \mathbb{Z}[X]$ and $a > 0$ in Eq. (1); the same methods work for any computable real subfield $\mathbf{D}$. In this section we provide rational isolating points for polynomials of degree up to 4. For the quadratic $f(X) = a\,X^2 - 2\,b\,X + c$, the rational number $\frac{b}{a}$, isolates the real roots.

Theorem 3 *(Emiris and Tsigaridas, 2004a) 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.*

We now study the quartic and derive rational isolating points.

Theorem 4 (Sederberg and Chang, 1993) *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* $\deg A + \deg B \leq \deg P - 1$.

By theorem 4 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

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

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 (3), denoted by

$\sigma_{1,2}$. We verify that $\text{sign}\left(f(\frac{b}{b})\right) = \text{sign}\,(a^2A - 3\Delta_2^2)$, so

$$\begin{cases} \sigma_1 < \frac{b}{a} < \sigma_2, \text{ if } f(\frac{b}{a}) > 0; \\ \sigma_1 < \sigma_2 < \frac{b}{a}, \text{ if } f(\frac{b}{a}) < 0 \ \wedge \ R > 0; \\ \frac{b}{a} < \sigma_1 < \sigma_2, \text{ if } f(\frac{b}{a}) < 0 \ \wedge \ R < 0; \end{cases} \tag{4}$$

where R is a seminvariant defined previously. 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 now

$$A(X) = W_3\,X^3 - 3\,W_2\,X^2 - 3\,\Delta_4\,X. \tag{5}$$

By the theorem at least 2 of $\{0, \tau_1, \tau_2\}$ separate the roots, where $\tau_{1,2}$ are the non-zero roots of $A(X)$. Wlog we assume that the roots of Eq. (1) are > 0, so 0 is not an isolating point. The order of the isolating points, τ_1 and τ_2, is determined similarly as in (4). Hence we have determined quadratic isolating points. Let us now find rational isolating points for all relevant cases of prop. 2.

$\{1,1,1,1\}$ Treated below.
$\{1,1\}$ Subsumed by case $\{1,1,1,1\}$, so we do not examine it explicitly.
$\{2,1,1\}$ The double root is rational since it is the only root of $\text{GCD}(f, f')$ and its value is $\frac{T_1}{T_2}$, see eq (2). 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 4 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 $\text{GCD}(f, f')$. If $\Delta_2 = 0$ then the root is $\frac{W_3}{3W_1}$, else it is $-\frac{T_2}{3T_1}$.
$\{2,2\}$ The roots are the smallest and largest root of the derivative i.e. a cubic. Alternatively, we express them as the roots of $3\Delta_2 X^2 + 3W_1X - W_3$.
$\{3,1\}$ 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 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 (3) and (5) 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 5 *For any* $m, n, m', n' \in \mathbb{N}^*$, $0 < \frac{m}{n} < \frac{m'}{n'} \Rightarrow \frac{m}{n} < \frac{m+m'}{n+n'} < \frac{m'}{n'}$.

In order to derive rational isolating points for proving th. 7 we set:

$$\mathcal{A} := 9\Delta_4 - 3\,ce, \quad \mathcal{B} := 12\,ae\Delta_4 + 9\,d^2c^2 \tag{6}$$

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. 2, $\Delta_2 > 0$ $\Rightarrow c < 0$. 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 7, provided the following holds:

Lemma 6 *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 (6).*

PROOF.

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

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

$$\begin{aligned} &\min \;\; L(a, c, d, e, y_1, y_2, y_3, \lambda_1, \lambda_2, \lambda_3) := \\ &\min \;\; [g(c, e) + \lambda_1(c + y_1^2 + 5)\lambda_2(-e + y_2^2 + 5) + \lambda_3(-a + y_3^2 + 1)] \end{aligned} \tag{7}$$

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 $e + 1 - y_2^2$. $\square$

Theorem 7 *Consider a quartic as in (1), with four distinct real roots. At least three of the rational numbers $\{0, \frac{b}{a}, \frac{e}{d}, \sigma_i \oplus \tau_j\}$ isolate the real roots, $i, j \in \{1, 2\}$.*

We measure complexity by the degree of the tested quantities in terms of the input polynomial's coefficients; assume a univariate polynomial of degree d. A lower bound is the degree of the resultant coefficients, which is $2\,d$, and is an open question if a better lower bound exists. There is a straightforward algorithm for the comparison of quadratic algebraic numbers, with maximum algebraic degree 4, hence optimal, e.g (Karavelas and Emiris, 2003).

Theorem 8 (Emiris and Tsigaridas, 2003) *There is an algorithm for the comparison of algebraic cubic numbers (including all degenerate cases), with maximum algebraic degree 6, hence optimal.*

Theorem 9 *(Emiris and Tsigaridas, 2004a) 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. These bounds cover all degenerate cases, including when one polynomial drops degree.*

6 Real solution of a bivariate quadratic polynomial system

We consider the system $f_1 = f_2 = 0$, where $f_{1,2} \in \mathbf{D}[X, Y]$ are bivariate polynomials of total degree at most 2. In what follows we assume that the system is $0-$dimensional (we can easily detect if it is not, since then the resultants that we compute below would not be univariate polynomials). The real solution of the system are points in $\overline{\mathbf{Q}}^2$.

In order to compute the real solutions of the system, we compute 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. We find the real solutions of R_x, R_y and their isolating points, define a grid of boxes, where the common roots of f_1 and f_2 are located. The grid has 1 to 4 rows and 1 to 4 columns in $\mathbb{R}^2$. It remains to decide, for boxes, whether they are empty and, if not, whether they contain a simple or multiple root, that is to match the algebraic numbers, computed as solutions of the resultants.

The hardest (computational) cases are when R_x and R_y do not have multiple roots. However in this case f_1 and f_2 are in generic position (the intersection points have distinct $x-$coordinates) and thus we can solve the system using a simple version of rational univariate representation, e.g (Rouillier, 1999; Gonzalez Vega and Necula, 2002). Now the $y-$coordinate is the solution of the first subresultant, which is univariate with respect to Y and its coefficients are univariate polynomials evaluated over the solutions of R_x, that is $\gamma_y = F(\gamma_x) = \frac{-B(\gamma_x)}{A(\gamma_x)}$. This implicit representation of γ_y is complicated and thus in order to have an isolating interval representation of it, we use the following trick. Since we have the solutions of R_y and their isolating points, we find the isolating interval at which each $F(\gamma_x)$ lies. This can be done with testing the signs of univariate polynomials evaluated over algebraic numbers.

In our case the computation of the two resultants is an easy computational task, since the degree is small and we have precomputed the appropriate quantities. Unlike e.g (Eigenwillig et al., 2004), where the boxes cannot contain any critical points of f_1 and f_2, 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 we obtain isolating points for the roots of the two resultants, either statically (as above) or dynamically. This is being implemented in SYNAPS.

7 Implementation

We have implemented a software package, S^3, as part of library SYNAPS (v2.1) (Dos Reis et al., 2002), for dealing with algebraic numbers and bivariate polynomial system solving, which is optimized for small degree. 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 and can handle all degenerate cases. We developed programs that produce all possible sign combinations of the tested quantities, so as to test as few quantities as possible, and produce both `C++` code and pseudo-code for solving, comparison and sign determination functions. In what follows `root_of` is a class that represents real algebraic numbers, computed as roots of polynomial, `UPoly` and `BPoly` are classes for univariate and multivariate polynomial. All classes are parametrized by the ring number type (`RT`); the reader may refer to the documentation of SYNAPS for more details. We provide the following functionality:

`Seq<root_of<RT> > solve(UPoly<RT>` f`)` Solves a univariate polynomial f.

`int compare(root_of<RT>` α`, root_of<RT>` β`)` Compares two algebraic numbers. For degree up to 4 we use static Sturm sequences. For higher degree we use Sturm-Habicht sequences, computed on the fly.

`int sign_at(UPoly<RT>` f`, root_of<RT>` α`)` Computes the sign of a univariate polynomial evaluated over an algebraic number.

`int sign_at(BPoly<RT>` f`, root_of<RT>` γ_x`, root_of<RT>` γ_y`)` Computes the sign of a bivariate polynomial evaluated over two real algebraic numbers. We use cascaded Stum-Habicht sequences.

`Seq < pair<root_of<RT> > > solve(BPoly<RT>` f_3`, BPoly<RT>` f_2`)` Computes the real solutions of a bivariate polynomial system.

8 Experimental results

We performed all tests on a 2.6GHz Pentium with 512MB memory, running Linux, with kernel version 2.6.10. We compiled the programs with `g++`, v. 3.3.5, with option `-O3`. Competitive algorithms are described in Sec. 2.

Table 1
Univariate root comparison

msec	A	B	C	D
f-S^3	0.142	0.153	0.150	0.177
S^3	0.291	0.320	0.142	0.112
RS	5.240	6.320	4.930	5.180
SYNAPS	1.058	1.011	0.717	1.850
CORE	3.050	3.520	2.240	1.470
GKR	2.287	2.973	2.212	1.595
NiX	0.358	0.362	0.215	0.377

Table 2
Bivariate real-solving

msec	A	B
f-S^3	0.17	0.18
S^3	0.14	0.54
GBRS	6.40	6.90
STH	0.51	0.57
RES	0.36	-
NEWMAC	3.19	3.26

Univariate case We perform four kinds of tests concerning the solution of quartic univariate polynomials and comparison of real algebraic numbers of degree up to 4. For every polynomial we "compute" all its real roots, with every package, since except S^3, no other package can compute a specific root only. We performed each test 10000 times. Column A refers to polynomials with exactly 4 distinct rational roots in $[-1, 1]$, the bit size of the coefficients is 40 bits. Column B refers to random polynomials, produced by interpolation in $[-1, 1] \times [-1, 1]$, the bit size of the coefficients is 90 bits. Column C refers to Mignotte polynomials, of the form $a(x^4 - 2(Lx - 1)^2)$, where the bit size of a and L is 40 bits. Finally, Column D, refers to degenerate polynomials, that is polynomials with at least one multiple root. All the roots are in $[-1, 1]$ and the bit size of the coefficients is 30 bits. The results are on table 1.

GKR is the package of Guibas et al. (2004) NiX is the polynomial library of EXACUS that has intristic filtering, since it is based on LEDA. CORE is version 1.7 and RS is the package of Rouillier and Zimmermann (2003) used through its MAPLE interface. SYNAPS refers to the algorithm of Mourrain et al. (2002) in SYNAPS. We have also tested MAPLE and AXIOM, but we do not show their timings here, since they are too slow, see (Emiris and Tsigaridas, 2004a) for details. S^3 is our code implemented in SYNAPS 2.1 and f-S^3, is our code using the filtered number type `Lazy_exact_nt` from CGAL[7].

SYNAPS has some problems when the roots are endpoints of a subdivision, while CORE has some problems with subdivision, since it uses Newton's method for refinement. By considering table 1, the exact version of our code (S^3), is clearly faster than CORE, SYNAPS and GKR, even when we do not use filtering. Also, in the case of filtering (NiX) S^3 is faster. Special attention must be paid to column D, where our code is remarkably faster, since it handles degenerate cases fast. The slow times of RS are due to the fact that we use its MAPLE

[7] www.cgal.org

interface in order to call the appropriate functions, since the source code is not available.

Now consider the first row of Table 1. The adoption of a filtered number type improves the running times in most cases, otherwise it leaves them essentially unchanged. Even in the hardest case, which is Column B, due to the big bit size of the coefficients, f-S^3 is 2 times faster than the next fastest software NiX, which has intrinsic filtering. We are planning to use more sophisticated filtering techniques in the next version of S^3.

Bivariate case We performed two kinds of experiments concerning real solving of bivariate polynomial systems of degree ≤ 2, and the results are on table 2. For every test we picked 2 polynomial at random and solve them; we repeat this 10000 times. Column A refers to 1000 bivariate polynomialas, with integer coefficients sampled in $[-10, 10]$, with not many intersections on average (every polynomial has common real roots with 135 others in the list, on average). Column B refers to 1000 conics sampled by 5 random integer points in $[-10, 10] \times [-10, 10]$, where 2 random conics probably intersect (every conic has common real roots with 970 others in the list, on average).

We test against NewMac (Mourrain and Trébuchet, 2002). It is a general purpose polynomial system solver, based on normal forms. sth, refers to a code in synaps, based on Sturm-Habicht sequences and subresultants, following Gonzalez Vega and Necula (2002). res is a bivariate polynomial solver based on the Bézoutian matrix and the eigensolver of lapack, see (Busé et al., 2005). GbRs, uses Gröbner bases and rational univariate representation (Rouillier, 1999). We use its maple interface, since the source code is not freely available, which explains the slow times of this package. S^3 refers to our code, while f-S^3 is our code based on `Lazy_exact_nt`.

We have to emphasize that our approach is exact, i.e it outputs isolating boxes with rational endpoints containing a unique root whose multiplicity is also calculated. On the other hand sth, uses a double approximation in order to compute the ordinate of the solution. res works only with doubles, since it has to compute the generalized eigenvalues and the eigenvectors based on lapack and that is why it cannnot perform the tests of Column B. NewMac, also relies on the computation of eigenvalues and computes also the complex solutions of the system. S^3 is considerably faster on both data sets, and always produces the correct results. This is not always the case for sth and res. When we use filtering, then our code is at least 3 times faster than any other approach. We have also done some preliminary experiments with exacus, but since we do not have yet a permission to present the code and the running times, we do not present the results here. However, our code seems to be faster.

Acknowledgments: We thank M-F. Roy for discussions with the 2nd author during ICPSS 2004 and B. Mourrain for his help with the implementation and experiments.

References

Basu, S., Pollack, R., M-F.Roy, 2003. Algorithms in Real Algebraic Geometry. Vol. 10 of Algorithms and Computation in Mathematics. Springer-Verlag.

Busé, L., Khalil, H., Mourrain B., 2005. Resultant-based methods for curves intersection problems, Manuscript.

Cremona, J. E., 1999. Reduction of binary cubic and quartic forms. J. Computation and Mathematics 2, 62–92.

Dos Reis, G., Mourrain, B., Rouillier, R., Trébuchet, P., 2002. An environment for symbolic and numeric computation. In: Proc. Int. Conf. Math. Software. World Scientific. pp. 239–249.

Dupont, L., Lazard, D., Lazard, S., Petitjean, S., 2003. Near-optimal parameterization of the intersection of quadrics. In: Proc. SoCG ACM, pp. 246–255.

Eigenwillig, A., Kettner, L., Schömer, E., Wolpert, N., 2004. Complete, exact, and efficient computations with cubic curves. In: Proc. SoCG ACM, pp. 409–418.

Emiris, I. Z., Tsigaridas, E. P., 2003. Comparison of fourth-degree algebraic numbers and applications to geometric predicates. Tech. Rep ECG-TR-302206-03, INRIA.

Emiris, I., Kakargias, A., Teillaud, M., Tsigaridas, E., Pion, S., 2004. Towards an open curved kernel. ACM Press, New York, pp. 438–446.

Emiris, I., Tsigaridas, E., 2004a. Computing with real algebraic numbers of small degree. In: Proc. ESA. LNCS. Springer Verlag, pp. 652–663.

Emiris, I. Z., Tsigaridas, E. P., 2004b. Computations with real algebraic numbers of degree up to 4. In: ICPSS (in honor of D. Lazard).

Gonzalez Vega, L., Necula, I., 2002. Efficient topology determination of implicitly defined algebraic plane curves. CAGD 19 (9), 719–743.

Guibas, L., Karavelas, M., Russel, D., 2004. A computational framework for handling motion. In: Proc. 6th Workshop Algor. Engin. & Experim.

Hemmer, M., Schömer, E., Wolpert, N., 2001. Computing a 3-dimensional cell in an arrangement of quadrics: Exactly and actually! In: Proc. SoCG, pp. 264–273.

Kaplan, D., White, J., 2001. Polynomial equations and circulant matrices. The Mathematical Association of America (Monthly) 108, 821–840.

Karavelas, M., Emiris, I., 2003. Root comparison techniques applied to the planar additively weighted Voronoi diagram. In: Proc. SODA. pp. 320–329.

Keyser, J., Culver, T., Manocha, D., Krishnan, S., 2004. ESOLID: A system for exact boundary evaluation. Comp. Aided Design 36 (2), 175–193.

Lazard, D., 1988. Quantifier elimination: optimal solution for two classical examples. J. Symb. Comput. 5 (1-2), 261–266.

Lazard, S., Peñaranda, L. M., Petitjean, S., 2004. Intersecting quadrics: an efficient and exact implementation. In Proc: SoCG ACM pp. 419–428.

Mourrain, B., Trébuchet, P., 2002. Algebraic methods for numerical solving. In: Proc. of the 3rd Int. Workshop on Symbolic and Numeric Algorithms for Scientific Computing'01 (Timisoara, Romania). pp. 42–57.

Mourrain, B., Vrahatis, M., Yakoubsohn, J.C, 2002. On the complexity of isolating

real roots and computing with certainty the topological degree. J. Complexity 18 (2).

Rioboo, R., 1992. Real algebraic closure of an ordered field: implementation in axiom. In: Proc. Annual ACM ISSAC. ACM Press, pp. 206–215.

Rioboo, R., 2002. Towards faster real algebraic numbers. In: Mora, T. (Ed.), Proc. Annual ACM ISSAC. ACM Press, New York, USA, pp. 221–228.

Rouillier, F., 1999. Solving zero-dimensional systems through the rational univariate representation. J. Appl. Algebra Engin, Comm. and Comp. 9 (5), 433–461.

Rouillier, F., Zimmermann, P., 2003. Efficient isolation of polynomial real roots. Journal of Computational and Applied Mathematics 162 (1), 33–50.

Salmon, G., 1885. Lessons Introductory to the Modern Higher Algebra. Chelsea Pub. Co, New York.

Schmitt, S., 2003. The diamond operator for real algebraic numbers. ECG-TR-243107-01, MPI Saarbrücken.

Sederberg, T. W., Chang, G.-Z., 1993. Isolating the real roots of polynomials using isolator polynomials. In: Bajaj, C. (Ed.), Algebraic Geometry and Applications.

Weispfenning, V., 1994. Quantifier elimination for real algebra–the cubic case. In: Proc. Annual ACM ISSAC. ACM Press, pp. 258–263.

Yang, L., 1999. Recent advances on determining the number of real roots of parametric polynomials. J. Symbolic Computation 28, 225–242.

Elias