ALGEBRA, LOGIC AND NUMBER THEORY
BANACH CENTER PUBLICATIONS, VOLUME 108
INSTITUTE OF MATHEMATICS
POLISH ACADEMY OF SCIENCES
WARSZAWA 2016

ALGORITHMS FOR QUADRATIC FORMS OVER REAL FUNCTION FIELDS

KONRAD JAŁOWIECKI and PRZEMYSŁAW KOPROWSKI

Institute of Mathematics, University of Silesia
Bankowa 14, 40-007 Katowice, Poland
E-mail: kjalowiecki@us.edu.pl, przemyslaw.koprowski@us.edu.pl

Abstract. This paper presents algorithms for quadratic forms over a formally real algebraic function field K of one variable over a fixed real closed field $\Bbbk$. The algorithms introduced in the paper solve the following problems: test whether an element is a square, respectively a local square, compute Witt index of a quadratic form and test if a form is isotropic/hyperbolic. Finally, we remark on a method for testing whether two function fields are Witt equivalent.

1. Introduction. Computational methods have been prominent in the theory of quadratic forms from its early dawn. One may, for example, mention Legendre's descent method for solving trivariate quadratic equations. This subjects continues to be an area of active research today. The problem of finding isotropic vectors and subspaces are treated (among others) in papers by J. E. Cremona and D. Rusin [5], D. Simon [14] or P. Castel [4]. Given a base field K, the main aim of the theory is to develop algorithms for the following tasks:

(1) testing if a non-degenerate quadratic form ξ (with coefficients in K) is isotropic;
(2) testing if ξ is hyperbolic;
(3) computing Witt index of ξ;
(4) determining the anisotropic part of ξ;
(5) finding an isotropic vector of ξ.

The problems listed above are ordered roughly with respect to their level of difficulty. Indeed, being able to solve (5), one may inductively knock out hyperbolic planes from the quadratic space (K^n, ξ), until the anisotropic part of ξ is found. Knowing the dimension of the anisotropic part one immediately computes the Witt index (see Section 3, below) and knowing the Witt index one verifies whether the form is isotropic/hyperbolic.

2010 *Mathematics Subject Classification*: 11E25, 14P05, 14Q99, 68W30.
The paper is in final form and no version of it will be published elsewhere.

In this paper, we deal with these problems for a formally real algebraic function field K of one variable over a real closed field $\Bbbk$. It was proved in [10] that already for the field $\Bbbk(X)$ of rational functions, there cannot exist an algorithm solving (4), hence there is also no such algorithm for (5). Therefore, in this paper we concentrate entirely on the first three tasks.

A function field K may be presented in various manners. The basic representations are the following ones:

(1) As a finite extension of a rational function field $\Bbbk(X)$, i.e. $K = \Bbbk(x,y) = {}^{\Bbbk(X)[Y]}\!/_{p}$, where $p \in \Bbbk(X)[Y]$ is an irreducible polynomial. The elements of K are then the remainders of polynomials modulo p and are represented as polynomials of degrees smaller than $\deg p$.
(2) As the fraction field of the quotient ring of the bivariate polynomial ring $\Bbbk[X,Y]$, i.e. $K = \Bbbk(x,y) = \operatorname{qf}\big({}^{\Bbbk[X,Y]}\!/_{P}\big)$, where $P \in \Bbbk[X,Y]$ is a bivariate irreducible polynomial. This way we may think of K as the function field of an affine curve $C = \{P = 0\} \subset \mathbb{A}^2\Bbbk(\sqrt{-1})$.
(3) As the function field of a projective algebraic curve $\hat{C} = \{P^h = 0\} \subset \mathbb{P}^2\Bbbk(\sqrt{-1})$, where $P^h \in \Bbbk[X,Y,Z]$ is a homogeneous polynomial.

Conversions between all these three representations are straightforward and will be implicit in presented algorithms, when needed. The points of $\hat{C}$ are in one-to-one correspondence with non-trivial valuation rings of K containing $\Bbbk$. We will frequently utilize this correspondence.

2. Square tests. Our first and most basic task is to develop an algorithm testing whether a given element $\alpha \in K$ is a square. We achieve the goal by the following procedure.

ALGORITHM 1. Let $K = \Bbbk(x,y) = {}^{\Bbbk(X)[Y]}\!/_{p}$ be an algebraic function field over $\Bbbk$ represented by an irreducible polynomial $p \in \Bbbk(X)[Y]$. For an element $\alpha \in \dot{K}$, represented by a polynomial $a \in \Bbbk(X)[Y]$, $\deg a < \deg p$, this algorithm returns true, when α is a square in K and false otherwise.

(1) Take $s := 1$ and compute
$$N := \operatorname{res}_Y\big((T - s\cdot Y)^2 - a, p\big) \in \Bbbk(X)[T].$$
If N is square-free, proceed to the next step. Otherwise increment s and compute N again. Repeat this step until N becomes a square-free polynomial.
(2) Let $T' := T - s\cdot Y$ and let $M \in \Bbbk[X]$ be the gcd of the denominators of the coefficients of N, considered as a polynomial in T' over $\Bbbk(X)$.
(3) Let $F \in \Bbbk(X)[T']$ be the primitive part of $M \cdot N \in \Bbbk(X)[T']$.
(4) Use [1, Alg. 4.4.1] to check if F is irreducible (by checking if the ideal (F) is prime) in $\Bbbk[X,T']$ as a bivariate polynomial. If it is irreducible then return false, otherwise return true.

Proof of correctness. There are only finitely many $s \in \Bbbk$ for which N is not square-free (see e.g. [16, Theorem 5.4.5]), hence the step (1) terminates in finite time. Now, α is a square in K if and only if $(T')^2 - \alpha$ factors in $K[T']$ and [16, Theorem 5.4.2] asserts

that this is the case if and only if N is reducible in $\Bbbk(X)[T']$. In turn, N is irreducible in $\Bbbk(X)[T']$ if and only if F is irreducible in $\Bbbk[X,T']$. Indeed, suppose F is reducible. Then, since F is primitive, it follows that it can be written as a product GH of two non-constant polynomials $G,H\in\Bbbk[X,T']$, so $N=\frac{1}{M}GH$ factors in $\Bbbk(X)[T']$. Conversely, if F is irreducible in $\Bbbk[X,T']$, it follows from the Gauss lemma that it is also irreducible in $\Bbbk(X)[T']$ and this is only the case if N is irreducible, since factoring N in $\Bbbk(X)[T']$ would clearly yield a factorization of F over the same field. ∎

REMARK. If the polynomial a in the above algorithm has rational coefficients, then for the irreducibility test in step (4) one can use [6, §5], which is much faster than [1, Alg. 4.4.1].

The previous algorithm tests whether an element is a square in K. One may consider it as a "global" problem. Its "local" counterpart is to check if α is a square in the completion $K_{\mathfrak{p}}$ for some real place $\mathfrak{p}$ of K. Let $\Bbbk$ be a real closed field. For $Q\in\Bbbk[X]$, $\deg(Q)>0$, let us denote by $\operatorname{Der}(Q)$ the set $\{Q,Q',\dots,Q^{\deg Q-1}\}$ of consecutive derivatives of Q. We assume the following notation: if (T,σ) is triangular Thom encoding specifying a point $(x_0,y_0)\in\Bbbk^2$, we denote elements of the list T by T_1 and T_2 and elements of the list σ by σ_1,σ_2. For the notion of Thom encoding we refer the reader to [2, Chapter 10]. Recall that $T_1,T_2,\sigma_1,\sigma_2$ satisfy the following conditions:

(1) $T_1\in K[X]$, $T_2\in K[X,Y]$ and $T_1(x_0)=0$, $T_2(x_0,y_0)=0$,
(2) σ_1 is a sign condition on $\operatorname{Der}(T_1)$ specifying x_0,
(3) σ_2 is a sign condition on $\operatorname{Der}\big(T_2(x_0,Y)\big)$ specifying y_0.

Before we present an algorithm for local square test we introduce a method to perform a particularly useful change of variables in a triangular Thom encoding.

ALGORITHM 2. Let $\Bbbk$ be an arbitrary real closed field. Let (T,σ) be a triangular Thom encoding of a point $(x_0,y_0)\in\Bbbk^2$. For a given $a\in\Bbbk$ this algorithm returns the triangular Thom encoding of the point (x_0+ay_0,y_0).

(1) Let $P(Y):=\operatorname{res}_X(T_1,T_2)$.
(2) Characterize all real roots $y_1,\dots,y_m$ of P by the sign conditions $\tau_1,\dots,\tau_m$ on $\operatorname{Der}(P)$ using [2, Alg. 10.14].
(3) Characterize all real roots $x_1,\dots,x_n$ of T_1 by the sign conditions $\overline{\tau}_1,\dots,\overline{\tau}_n$ on $\operatorname{Der}(T_1)$ using [2, Alg. 10.14].
(4) Comparing necessary sign conditions using [2, Alg. 10.16], find j such that $\overline{\tau}_j=\sigma_1$.
(5) Comparing necessary sign conditions using [2, Alg. 10.16], find k such that the sign condition realized on $\operatorname{Der}_Y(T_2)$ at the point specified by $\big((T_1,P),(\overline{\tau}_j,\tau_k)\big)$ is equal to σ_2.
(6) Let $Q:=T_1(X+aY)$.
(7) Let $R:=\operatorname{res}_Y(Q,P)$.
(8) Describe all real roots $\overline{x}_1,\dots,\overline{x}_l$ of R by the sing conditions $\overline{\sigma}_1,\dots,\overline{\sigma}_l$ on $\operatorname{Der}(R)$ realized at these points using [2, Alg. 10.14].
(9) Choose an increasing sequence $m_1,\dots,m_n$ of natural numbers, such that points $(\overline{x}_{m_1},y_k),\dots,(\overline{x}_{m_n},y_k)$ correspond to common roots of Q and P using [2, Alg. 11.8] for sign determination.
(10) Return the triangular Thom encoding $\big((R,P),(\overline{\sigma}_{m_j},\tau_k)\big)$.

Proof of correctness. It is clear that the algorithm terminates in finite time. Existence of j and k in steps (4) and (5) is also clear. After performing a transformation of the form $X \mapsto X + aY$, the ordinate of every point remains fixed and so does the number of common roots of T_1 and P. Transformations of this form also preserve order on $\Bbbk \times \{y_j\}$. Since the abscissa of every common root of Q and P can be encoded using sign conditions on $\mathrm{Der}(R)$ it is sufficient to determine sign conditions of those roots of the form (x, y_j) and then choose k-th of them, which is what the algorithm does. ∎

Recall (see e.g. [7], [3] or [12, Section 1.4]) that a real curve C can be equipped with a topology (called *strong* or *euclidean* topology) in which the elements of K regarded as real functions on C are continuous. The curve C consists of finitely many semi-algebraically connected components. Every component can be independently oriented in two different ways. Having an orientation fixed, for every two points $\mathfrak{p}$, $\mathfrak{q}$ belonging to the same component, one defines an interval $(\mathfrak{p}, \mathfrak{q})$ to be a subset consisting of all the points lying between $\mathfrak{p}$ and $\mathfrak{q}$ with respect to the given orientation. In order to present an algorithm for a local square test we need one more tool. The following proposition gives a necessary and sufficient condition for $\alpha \in K$ to be a local square at a point $\mathfrak{p}$.

Proposition 2.1. *Let $\mathfrak{p}$ be a point on a curve C and $\alpha \in K$. Then α is a square in the completion $K_\mathfrak{p}$ of K at point $\mathfrak{p}$ if and only if it takes strictly positive values in some neighborhood of* $\mathfrak{p}$ excluding $\mathfrak{p}$.

Proof. There exist exactly two orderings of K that are compatible with $\mathfrak{p}$ (see e.g. [9, p. 332]), namely:

$$P_\mathfrak{p}^- = \Big\{\alpha \in K\colon\ \exists\, \mathfrak{q} < \mathfrak{p}\ \forall\, \mathfrak{r} \in (\mathfrak{q}, \mathfrak{p})\ \alpha(\mathfrak{r}) > 0\Big\},$$
$$P_\mathfrak{p}^+ = \Big\{\alpha \in K\colon\ \exists\, \mathfrak{q} > \mathfrak{p}\ \forall\, \mathfrak{r} \in (\mathfrak{p}, \mathfrak{q})\ \alpha(\mathfrak{r}) > 0\Big\}.$$

Now, $\alpha \in \dot{K}_\mathfrak{p}$ is a square if and only if it belongs to both orderings. This condition is equivalent to the statement of the proposition. ∎

We are now ready to present an algorithm for local square test.

Algorithm 3. Let P be an irreducible bivariate polynomial over a real closed field $\Bbbk$, representing a function field $K = \mathrm{qf}(\Bbbk[X,Y]/P)$. Assume that the curve $C = \{P = 0\}$ has no self-intersections. Given a real point $\mathfrak{p}$ lying on C, represented by a triangular Thom encoding (T, σ) and an element $\alpha \in \dot{K}$ represented by a polynomial $a \in \Bbbk[X,Y]$, this algorithm returns true if α is square in the completion $K_\mathfrak{p}$ of K at $\mathfrak{p}$ and false otherwise.

(1) Evaluate the sign s of a at the point $\mathfrak{p}$ using [2, Alg. 11.8].
(2) If $s > 0$ return true, if $s < 0$ return false. Proceed to the next step only when $s = 0$.
(3) Using [2, Alg. 11.8] determine the sign of $\frac{\partial P}{\partial Y}$ at the point $\mathfrak{p}$, if it is 0, then perform a change of coordinates $(X, Y) \mapsto (Y, X)$.
(4) Let $k := 1$.
(5) Determine signs realized by $\frac{\partial P}{\partial Y}$ above the point represented by Thom encoding (T_1, σ_1) using [2, Alg. 11.12] and [2, Alg. 11.8]. If none of them is 0 go to step (8).
(6) Perform a linear change of coordinates $(X, Y) \mapsto (X + kY, Y)$ on P and a and point $\mathfrak{p}$ using Algorithm 2.

(7) Let $k := k + 1$, go to step (5).
(8) Use [2, Alg. 11.2] to perform a cylindrical algebraic decomposition adapted to the system $\{a, P, T_1\}$. Denote points determining intervals of first level cells by $x_1 < \ldots < x_L$ and let j be such that x_j is the abscissa of the point $\mathfrak{p}$. Denote sample points above x_j by $A_1, \ldots, A_d$, sample points above (x_{j-1}, x_j) by $B_1, \ldots, B_e$ and sample points above (x_j, x_{j+1}) by $C_1, \ldots, C_f$ (putting $x_0 = -\infty$ and $x_{l+1} = +\infty$ if necessary).
(9) Choose an increasing subsequence $m_1, \ldots, m_n$ of $\{1, \ldots, d\}$ consisting of all such indices that the sign of P at points $A_{m_1}, \ldots, A_{m_n}$ is 0.
(10) Choose an increasing subsequence $\overline{m}_1, \ldots, \overline{m}_n$ of $\{1, \ldots, e\}$ consisting of all such indices that the sign of P at points $B_{\overline{m}_1}, \ldots, B_{\overline{m}_n}$ is 0.
(11) Choose an increasing subsequence $\tilde{m}_1, \ldots, \tilde{m}_n$ of $\{1, \ldots, f\}$ consisting of all such indices that the sign of P at points $C_{\tilde{m}_1}, \ldots, C_{\tilde{m}_n}$ is 0.
(12) Find l such that the point A_{m_l} corresponds to the input point $\mathfrak{p}$.
(13) Extract the sings s_1 and s_2 of a respectively at points $B_{\overline{m}_l}$ and $C_{\tilde{m}_l}$, return $s_1 > 0 \wedge s_2 > 0$.

Proof of correctness. Denote the coordinates of $\mathfrak{p}$ by (p_x, p_y). Proposition 2.1 asserts that it suffices to check if the polynomial a takes only positive values in some neighborhood of $\mathfrak{p}$ on curve C except possibly $\mathfrak{p}$ itself. Therefore, if it is positive at $\mathfrak{p}$ it is a local square and if it takes a negative value at this point it certainly is not a local square. The only remaining case is when curves $\{P = 0\}$ and $\{a = 0\}$ intersect at $\mathfrak{p}$. In this case we make sure that tangent to P at $\mathfrak{p}$ is not vertical. In case it is not, polynomial P crosses line $X = p_x$ and we only need to determine sign of a on two intervals of C adjacent to $\mathfrak{p}$ from both sides. The algorithm first makes sure that there are no vertical tangent lines to P above p_x, and since there are only finitely many points at which it may happen, loop in steps (4)–(7) clearly stops in finite time. In case when there are no vertical tangents above the point p_x, the algorithm performs cylindrical algebraic decomposition adapted to $\{a, P, T_1\}$. Since there are no critical points of C above p_x, the number of cells above p_x at which P vanishes is the same as the number of cells at which P vanishes above the intervals adjacent to p_x. Thus, in order to determine the required sign conditions for a, it suffices to find sample points for the cells corresponding to two intervals adjacent to $\mathfrak{p}$. This is what the algorithm does. ∎

It is also worth pointing out that a test if an element is a square can be performed not only in K, but also in $\Bbbk[X, Y]$. The next algorithm checks whether a given $F \in \Bbbk[X, Y]$ is a square.

Recall (see e.g. [16, Ch 5.2]) that Kronecker transform of a polynomial $F \in \Bbbk[X, Y]$ is a mapping of the form $F(X, Y) \mapsto F(T, T^d)$, where $d > 1$ is a fixed natural number. For a given d let us denote this transform by S_d. Observe that S_d is a ring homomorphism from $\Bbbk[X, Y]$ to $\Bbbk[T]$. In general, it is not injective. However, one may get an injective map by restricting S_d to the family $\big\{F \in \Bbbk[X, Y] \colon \max\{\deg_X(F), \deg_Y(F)\} \leq d\big\}$.

It is clear that if $F \in \Bbbk[X, Y]$ is a square, then for every $d \in \mathbb{N}$, $d \geq 2$, its image $S_d(F)$ is also a square. The opposite implication is false in general. Nevertheless, by fixing a sufficiently large d and computing the square root g of $S_d(F)$ and finding its preimage

$S_d^{-1}(g)$, one can eliminate "false positives". The details of the procedure are formulated in the following algorithm.

ALGORITHM 4. Let $\Bbbk$ be an arbitrary real closed field and let $F \in \Bbbk[X,Y]$. This algorithm returns true if F is a square in $\Bbbk[X,Y]$ and false otherwise.

(1) Compute $d = \max\{\deg_X(F), \deg_Y(F)\}$.
(2) Perform Kronecker transform $S_d(F)$ and denote the image by f.
(3) Perform the square-free factorization $a_1 a_2^2 \cdots a_n^n$ of f.
(4) Using the square-free factorization of f, decide if it is a square in $\Bbbk[T]$, if not return false, otherwise compute g such that $f = g^2$.
(5) Compute the preimage $G = S_d^{-1}(g)$. Check if $F = G^2$, if yes return true, otherwise return false.

Proof of correctness. Consider f computed in step (2) of the algorithm. If it is not a square in $\Bbbk[T]$ then certainly F cannot be a square in $\Bbbk[X,Y]$. Hence, the only thing that needs to be shown is that if $f = g^2$ and $F \neq G^2$ then F is not a square. Suppose otherwise, $F = H^2$ for some $H \in \Bbbk[X,Y]$. But then $f = S_d(H)^2$, so $S_d(H) = g$ or $S_d(H) = -g$. In both cases $F = S_d^{-1}(h)^2 = G^2$. ∎

3. Isotropy, hyperbolicity and Witt index. Our next aim is to develop algorithms checking if a given form ξ is isotropic/hyperbolic or more generally computing the dimension of the anisotropic part (hence consequently also the Witt index) of ξ. To this end, we need a tool determining the signatures $\operatorname{sgn}_{\mathfrak{p}} \xi$ of ξ at almost all real points $\mathfrak{p} \in C$. Clearly, the signatures may possibly change only at the zeros/poles of the coefficients of the form ξ. Other than that, it is constant on the intervals between the zeros/poles. Hence, the given form ξ admits only finitely many signatures. We may compute them using the following variant of cylindrical algebraic decomposition.

ALGORITHM 5. Let $\Bbbk$ be a real closed field, let K be a formally real algebraic function field over $\Bbbk$ represented by an irreducible polynomial $P \in \Bbbk[X,Y]$ and let $C_r = \{(x,y) \in \mathbb{A}^2\Bbbk\colon P(x,y) = 0\}$. Given a non-degenerate quadratic form $\xi = \langle \alpha_1, \ldots, \alpha_d \rangle$, where the coefficients α_i are represented by polynomials $A_1, \ldots, A_d$, the algorithm returns the list S of signatures of ξ at all but finitely many points on C_r.

(1) Compute the resultants $r_j = \operatorname{res}_Y(P, A_j)$, for $j = 1, \ldots, d$.
(2) Let $R := r_1 \cdots r_d$ and $S := ()$ be an initially empty list.
(3) Compute an ordered list of Thom encodings $\sigma_1, \ldots, \sigma_m$ of all real roots $x_1 < x_2 < \ldots < x_m$ of R using [2, Alg. 10.14].
(4) Compute Thom encodings $\overline{\sigma}_1, \ldots, \overline{\sigma}_l$ of all the roots $\overline{x}_1 < \ldots < \overline{x}_l$ of R' using [2, Alg. 10.14].
(5) Using [2, Prop. 2.28] for comparison of Thom encodings in lists $\{\sigma_i\}$ and $\{\overline{\sigma}_j\}$, choose an increasing subsequence $k_1, \ldots, k_{m-1}$ of $1, 2, \ldots, m-1$ such that $\overline{x}_{k_j}$ lies between x_j and x_{j+1}.
(6) To simplify notation denote Thom encodings $\overline{\sigma}_{k_j}$ by θ_j. Compute Thom encodings θ_0 and θ_m of points q_0 and q_m respectively smaller and larger than every real root of R. This can be done for example by encoding respectively smallest root of $R(X+1)$ and largest root of $R(X-1)$.

(7) For each θ_j with $0 \leq j < m$ which specifies a root of $\operatorname{res}_Y(P, \frac{\partial P}{\partial Y})$ do the following: compute Thom encoding of an arbitrary point between the point specified by θ_j and x_{j+1} using [2, Alg. 10.15]. Replace θ_j by this Thom encoding. Repeat until the point specified by θ_j is no more a root of $\operatorname{res}_Y(P, \frac{\partial P}{\partial Y})$.

(8) For each $j \in \{1, \ldots, m\}$ proceed as follows:
(a) Denote by q_j the point specified by θ_j. Compute triangular Thom encodings $\widetilde{\sigma}_1, \ldots, \widetilde{\sigma}_{n-1}$ specifying points $(q_j, y_1), \ldots, (q_j, y_{n-1})$ where $y_1, \ldots, y_{n-1}$ are points lying in the intervals between the consecutive real roots of the polynomial $f(Y) = P(q_j, Y) \cdot A_1(q_j, Y) \cdots A_d(q_j, Y)$ (one point in each such interval) using [2, Alg. 11.11].
(b) Compute Thom encodings $\widetilde{\sigma}_0, \widetilde{\sigma}_n$ of arbitrary points $(q_j, y_0), (q_j, y_n)$ with the property that $y_0 < y_1$ and $y_{n-1} < y_n$. This can be done for example by encoding the smallest root of the polynomial $f(Y+1)$ and the largest root of the polynomial $f(Y-1)$.
(c) For every $0 < i \leq n$ if $\operatorname{sgn} P(q_j, y_i) \cdot \operatorname{sgn} P(q_j, y_{i+1}) < 0$ (which can be checked using [2, Alg. 11.8]) compute, using the same algorithm, $s_1 = \operatorname{sgn} A_1(q_j, y_i), \ldots, s_d = \operatorname{sgn} A_d(q_j, y_i)$. Append $s_1 + \ldots + s_d$ to S.

(9) Return S.

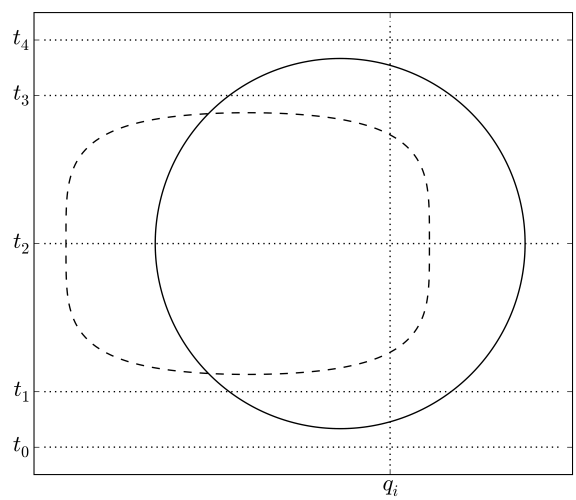

Fig. 1. Illustration of Algorithm 5. Curve C_r is shown with a solid line and curve $\{A_k = 0\}$ is shown with a dashed line.

REMARK. Figure 1 illustrates how the above algorithm works. Observe that, if C is not monotonic in the cylinder over some (q_i, q_{i+1}), then the signatures of ξ on some intervals may be computed multiple times. In other words, the algorithm computes the signatures of ξ on all intervals of C with endpoints either at "critical points" (i.e. points where tangent to C is strictly vertical) or points with abscissas equal to the projection of some zero/pole of at least one coefficient of ξ.

Proof of correctness. The curve C intersects curves $\{A_1 = 0\}, \ldots, \{A_d = 0\}$ in only finitely many points. By the basic properties of the resultant, the roots of R are precisely the abscissas of all these intersection points. Fix now a point (q_j, y_i) as computed in step (6), (7) and (8a) of the algorithm. All the roots of f are distinct, hence if $P(q_j, y_i) \cdot P(q_j, y_{i+1})$ is negative, this means that y_i is not a root of any $A_k(q_j, Y)$. Consequently, the sign of $A_k(q_j, y_i)$, or equivalently the sign of $A_k(q_j, y_{i+1})$, is just the sign of α_k on the whole interval of C lying between (q_j, y_i) and (q_j, y_{i+1}). ∎

Recall (see e.g. [13, Chapter i, §4]) that any non-degenerate quadratic form ξ can be uniquely (up to an isometry) decomposed as $\xi = \eta \perp H$, where η is an anisotropic form, called the *anisotropic part* of ξ and H is hyperbolic. The number of hyperbolic planes making up H (i.e. half of the dimension of H) is called the *Witt index* of ξ and denoted by $\operatorname{ind}(\xi)$, in other words $\operatorname{ind}\xi = {}^1\!/_2 \cdot (\dim\xi - \dim\eta)$. The above algorithm can be used to answer three questions: about isotropy, hyperbolicity and Witt index of a given quadratic form.

PROPOSITION 3.1. *Let $\xi = \langle \alpha_1, \ldots, \alpha_d \rangle$ be a non-degenerate quadratic form over the real function field $K = \Bbbk(x, y)$.*

(1) *If $d = 1$, then ξ is neither isotropic nor hyperbolic and* $\operatorname{ind}\xi = 0$.
(2) *If $d = 2$, then ξ is isotropic, and equivalently hyperbolic, if and only if $-\alpha_1\alpha_2$ is a square in K (this condition can be checked by Algorithm* 1*). If it is the case then* $\operatorname{ind}\xi = 1$, *otherwise* $\operatorname{ind}\xi = 0$.
(3) *Assume that $d > 2$ and $s = \max\{|\sigma| \mid \sigma \in S\}$, where S is the list returned by Algorithm* 5*. The form ξ is isotropic if and only if $s < d$. The form ξ is hyperbolic if and only if $s = 0$. The Witt index of ξ equals* $\operatorname{ind}\xi = {}^{(d-s)}\!/_2$.

Proof. The first two assertions are trivial, while the third one follows immediately from the Witt theorem (see e.g. [8, Theorem 9.4]). The computation of the Witt index follows from the above equality $\operatorname{ind}\xi = {}^1\!/_2 \cdot (\dim\xi - \dim\eta)$. ∎

4. Witt equivalence. Recall (see e.g. [13] or [15]) that the Witt ring of a field (resp. a ring) K is the ring of similarity classes of non-degenerate bilinear forms with the addition induced by the orthogonal sum and the multiplication induced by the tensor product. Two fields (resp. rings) are said to be *Witt equivalent* if their Witt rings are isomorphic (see e.g. [12, Chapter 5] or [15, Chapter 20] for further information concerning Witt equivalence). One of the fundamental problems in algebraic theory of quadratic forms is to find criteria for two fields/rings to be Witt equivalent. It is known (see e.g. [9, Theorem 5.1]) that two function fields K, L with a common real closed field of constants $\Bbbk$ are Witt equivalent if and only if their are both formally real (i.e. the associated real curves are not empty) or both non-real (i.e. the curves are both empty). Suppose now that both K and L are formally real. Denote by $\mathcal{R}_K := \bigcap_{\mathfrak{p}\ -\ \text{real}} \mathcal{O}_{\mathfrak{p}}(K)$ the intersection of all the residually real valuation rings of K, and similarly $\mathcal{R}_L := \bigcap_{\mathfrak{p}\ -\ \text{real}} \mathcal{O}_{\mathfrak{p}}(L)$. These are the rings of functions regular in all real points of the curves $\hat{C}_K$ associated to K and $\hat{C}_L$ associated to L. It is known (see [11, Corollary 4.2]) that $\mathcal{R}_K$ and $\mathcal{R}_L$ are Witt equivalent if and only $\hat{C}_K$ and $\hat{C}_L$ have the same number of semi-algebraically connected components. The algorithm in

[6, §6] checks if a real curve is non-empty and the number of components of the curve may be computed by the means of cylindrical algebraic decomposition[1]. Hence we proved:

OBSERVATION 4.1. *Using Algorithm* [2, Alg. 15.13] *one may verify whether two function fields K, L with a common real closed field of constants $\Bbbk$ are Witt equivalent or not. If they are, the same algorithm may be used to check if their sub-rings $\mathcal{R}_K$, $\mathcal{R}_L$ are Witt equivalent.*

References

[1] W. W. Adams, P. Loustaunau, *An Introduction to Gröbner Bases*, Grad. Stud. Math. 3, Amer. Math. Soc., Providence, RI 1994.

[2] S. Basu, R. Pollack, M.-F. Roy, *Algorithms in Real Algebraic Geometry*, Algorithms Comput. Math. 10, Springer, Berlin 2003.

[3] J. Bochnak, M. Coste, M.-F. Roy, *Real Algebraic Geometry*, Ergeb. Math. Grenzgeb. (3) 36, Springer, Berlin 1998.

[4] P. Castel, *Solving quadratic equations in dimension* 5 *or more without factoring*, in: ANTS X—Proceedings of the Tenth Algorithmic Number Theory Symposium, Open Book Ser. 1, Math. Sci. Publ., Berkeley, CA 2013, 213–233.

[5] J. E. Cremona, D. Rusin, *Efficient solution of rational conics*, Math. Comp. 72 (2003), 1417–1441 (electronic).

[6] E. Kaltofen, *Computing the irreducible real factors and components of an algebraic curve*, Appl. Algebra Engrg. Comm. Comput. 1 (1990), 135–148.

[7] M. Knebusch, *On algebraic curves over real closed fields*, I, Math. Z. 150 (1976), 49–70.

[8] M. Knebusch, *On algebraic curves over real closed fields*, II, Math. Z. 151 (1976), 189–205.

[9] P. Koprowski, *Witt equivalence of algebraic function fields over real closed fields*, Math. Z. 242 (2002), 323–345.

[10] P. Koprowski, *Algorithms for quadratic forms*, J. Symbolic Comput. 43 (2008), 140–152.

[11] P. Koprowski, *Witt equivalence of rings of regular functions*, Ann. Math. Sil. 22 (2008), 45–57.

[12] P. Koprowski, *Witt Morphisms*, Wydawnictwo Uniwersytetu Śląskiego, Katowice 2012.

[13] T. Y. Lam, *Introduction to Quadratic Forms over Fields*, Grad. Stud. Math. 67, Amer. Math. Soc. Providence, RI 2005.

[14] D. Simon, *Solving quadratic equations using reduced unimodular quadratic forms*, Math. Comp. 74 (2005), 1531–1543 (electronic).

[15] K. Szymiczek, *Bilinear Algebra. An Introduction to the Algebraic Theory of Quadratic Forms*, Algebra, Logic and Applications, Gordon and Breach, Amsterdam 1997.

[16] F. Winkler, *Polynomial Algorithms in Computer Algebra*, Texts Monogr. Symbol. Comput., Springer, Vienna 1996.

[1] In fact the algorithm in [6] works in archimedean case only. In a non-archimedean case one may however use CAD for both: testing if a curve is non-empty and computing the number of its components.