The Explicit Isomorphism Problem

Péter Kutas

Supervisor: Gábor Ivanyos

Associate Supervisor: Lajos Rónyai

A Dissertation Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy in Mathematics

Central European University
Budapest, Hungary

March 6, 2017

Dedicated to the memory of László Hannák

Acknowledgements

First and foremost I would like to thank my supervisor Gábor Ivanyos and my associate advisor Lajos Rónyai. Without their immense help and support I could not have completed this thesis.

I am also indebted to the Mathematics Department at CEU. It was extremely to be enrolled to such a nice place where I was surrounded by extraordinary people who made my life considerably easier.

I am grateful to Gergő Nemes, who not only provided the template for this thesis, but also was a constant help to me throughout my graduate years.

I'm privileged to have worked with and consulted with extremely talented people, some of which I had the pleasure to coauthor a paper with. The list includes Emil Kiss, Gergely Zábrádi, Zoltán Boros, Włodzimierz Fechner, José Gómez-Torrecillas, Francisco Javier Lobillo, Gabriel Navarro, Till Mitzow, Márton Erdélyi and many more.

Finally, I would like to thank my wife Edina, and my friends and family.

ABSTRACT

In this thesis we consider the following algorithmic problem. Let K be a field and let $\mathcal{A}$ be an algebra over K which is given by structure constants and is isomorphic to $M_n(K)$, the algebra of $n \times n$ matrices over K. The task is to find an explicit isomorphism between $\mathcal{A}$ and $M_n(K)$. We propose a polynomial time algorithm for the case where $K = \mathbb{F}_q(t)$, the field of rational functions over a finite field. Using an oracle for integer factorization, we provide an algorithm for $K = \mathbb{Q}(\sqrt{d})$ and $n = 2$. This algorithm reduces the original problem to finding nontrivial zeros of quadratic forms in several variables over $\mathbb{Q}$. Since the reduction procedure works over every field of characteristic different from 2, we concern ourselves with finding nontrivial zeros of quadratic forms over $\mathbb{F}_q(t)$, where q is odd. We propose a polynomial time algorithm for finding nontrivial zeros of quadratic forms over $\mathbb{F}_q(t)$. We apply the algorithm to compute the Witt decomposition of a quadratic form and decide equivalence of quadratic forms. Also, in the case two quadratic forms are equivalent, we provide a transition matrix. Finally, the algorithm is applied to compute an explicit isomorphism in the case $\mathcal{A} \cong M_2(L)$, where L is a quadratic extension of $\mathbb{F}_q(t)$ (where q is odd).

Besides these results, we also obtained some minor results as well (such as lattice reduction over the field of formal Laurent-series) and we have implemented two of our main algorithms.

PREFACE

Finite dimensional associative algebras arise in various branches of mathematics, including number theory, group theory, topology and algebraic geometry. Computing the structure of algebras is an important task which can be applied for example to computing irreducible representations of finite groups (by computing the Wedderburn decomposition of the underlying group algebra). An algebra can be considered as an input of an algorithm in many ways. In this thesis we always consider them given as a collection of structure constants, i.e., by a vector space basis and a multiplication table of the basis elements. Several natural questions arise. How can we compute the radical of an algebra? Can we compute the minimal left ideals of its semisimple part? Can we compute an explicit isomorphism between simple algebras? The answer is highly dependent on the ground field over which the algebra is considered. As it turns out, computationally the most difficult part is computing an isomorphism between simple algebras which are known apriori to be isomorphic. By the general theory of central simple algebras this reduces to the following problem, called the *explicit isomorphism problem*:

Problem 1 (Explicit isomorphism problem)**.** *Let K be a field, and let $\mathcal{A}$ be an algebra over K given by structure constants. Suppose that $\mathcal{A} \cong M_n(K)$. Compute an isomorphism between $\mathcal{A}$ and $M_n(K)$.*

This problem is interesting on its own as well, however, it has various applications in different areas of mathematics. Here we give three examples (the first two are quite recent applications).

Example 1. Let E be an elliptic curve defined over K, where K is an algebraic number field (for an introduction to the arithmetic theory of elliptic curves the reader is referred to [65]). Then by the Mordell-Weil theorem, $E(K)$, the group of K-rational points, is a finitely generated abelian group. It is a natural question, that, given a curve, how can one compute generators for $E(K)$? This is a question of high importance in algorithmic number theory, and is related to the Birch–Swinnerton-Dyer conjecture (which postulates a relation between the rank of $E(K)$ and the vanishing of a certain L-function at $s = 1$). It is known

that the factor groups $E(K)/nE(K)$ are finite (this is usually called the weak Mordell-Weil theorem). Via the descent theorem (which is the standard way of proving the Mordell-Weil theorem) one can see that it is sufficient to have an algorithm for finding generators of the finite groups $E(K)/nE(K)$ (for all n). However, as of today, no such procedure is known. On the other hand, there is the following exact sequence:

$$0 \to E(K)/nE(K) \to S^{(n)}(E/K) \to Ш(E/K)[n] \to 0$$

where $S^{(n)}(E/K)$ is the n-Selmer group of E and $Ш(E/K)[n]$ is the n-torison of the Tate-Shafarevich group of E. It is known (see [65, Chapter X, Theorem 4.2.]) that the n-Selmer group of an elliptic curve defined over K is finite. Now we have that the group $E(K)/nE(K)$ is embedded in a finite group. However, computing the Selmer-group is a difficult task. There is a technique, called n-descent, which reduces the question of finding generators to finding rational points on homogeneous spaces associated to the elliptic curve. If $n = 2$, then this is an efficient way for computing $E(K)/nE(K)$, as in this case, homogeneous spaces satisfy the local-global principle. For further details see [65, Chapter X].

Higher descents are more complicated. An important series of papers in this field is [11], [12], [13]. They represent the elements of the n-Selmer group by projective curves of degree n in $\mathbb{P}^{n-1}$ via the following method. The n-Selmer group is a subgroup of the cohomology group $H^1(K, E[n])$ (where $E[n]$ denotes the n-torsion subgroup of $E(\overline{K})$). The obstruction map Ob is a map from $H^1(K, E[n])$ to the Brauer group of K which has the property that it maps an element of the n-Selmer group to $M_n(K)$ (it is in the 'kernel' of this map). Let $\xi \in H^1(K, E[n])$. Then in [13] the authors can compute structure constants for the central simple algebra $Ob(\xi)$. From an explicit isomorphism between $Ob(\xi)$ and $M_n(K)$ they can compute equations for the curve ξ. Their algorithm is effective in the case when $n = 3$ and $K = \mathbb{Q}$. They note that one of the key obstacles for making the algorithm effective in the general case lies in solving the explicit isomorphism problem for number fields. Due to this connection, one of the authors of [11], [12], [13] (John Cremona) invited the paper [38] to be submitted to the journal Foundations of Computational Mathematics ([38] includes a large part of this thesis).

Example 2. Let $\mathbb{F}_q$ be the finite field with q elements. Let $\mathbb{F}_q(t)$ be the field of rational functions over $\mathbb{F}_q$. Suppose that σ is an automorphism of $\mathbb{F}_q(t)$. Let $R = \mathbb{F}_q(t)[x, \sigma]$ denote the ring of Ore polynomials, i.e., the usual polynomial ring (with variable x) over $\mathbb{F}_q(t)$ where multiplication is induced by the relation $xr = \sigma(r)x$ ($r \in \mathbb{F}_q(t)$). As in the case of ordinary polynomials, one may consider the problem of factorization into irreducible polynomials. This ques-

tion is equivalent to the problem of computing the structure of an algebra over a finite extension of $\mathbb{F}_q(t)$. As it turns out, the hardest part is when an algebra is simple. Therefore, this problem reduces to solving the explicit isomorphism problem when K is a finite extension of $\mathbb{F}_q(t)$. The factorization problem of Ore polynomials has several applications. A recent application is the construction of certain convolutional codes [27].

Example 3. Let K be a field and let L be a finite Galois extension of K. Let $N_{L|K}$ denote the norm map from L to K. Then an equation of the type $N_{L|K}(x) = a$, where $a \in K$ is called a norm equation. Solving norm equations (a is given, x is unknown) is a classical problem in computational number theory. Many fundamental results originate from the research of the Number Theory School of Debrecen, through the work of Győry [6], Pethő, Bérczes [2], Gaál ([23]) and many more (this list is just an excerpt of their work, it is far from a complete list). They mostly consider the case where a is an algebraic integer and look for integral solutions. This is a more complicated matter as one cannot say that if the norm equation $N_{L|K}(x) = a$ is solvable (here a is an algebraic integer), then there exists an integral solution as well. We give an example which can be found in the introduction of [64]. Let $K = \mathbb{Q}$, $L = \mathbb{Q}(\sqrt{34})$ and $a = -1$. Then $N_{L|K}(x) = a$ has no integral solution as the fundamental unit in L has norm 1. On the other hand $x = \frac{\sqrt{34}+5}{3}$ has norm -1. Besides norm equations being of theoretical interest, they can also be used for cryptographic purposes [3].

The first algorithm for solving norm equations in the case $K = \mathbb{Q}$ was proposed by Fincke [21] in his Thesis. This algorithm was later extended to number fields by Fieker, Jurk and Pohst [20]. A different approach is due to Simon [64]. Simon proves that if S is a suitably large set of primes then if S-unit is a norm in K then it is the norm of an S-unit in L. The question of solving norm equations over global function fields is considered in [23].

The relation of norm equations to central simple algebras is a classical result: a cyclic algebra is isomorphic to a full matrix algebra if and only if a certain norm equation is solvable. Moreover, from an explicit isomorphism (between the cyclic algebra and the full matrix algebra) a solution to the norm equation can be retrieved. Thus an algorithm for the explicit isomorphism problem can be used to solve norm equations. It is an intriguing research problem how this approach compares to the algorithms from [20] and [64]. Thus besides obtaining new algorithms and theoretical results, implementations of existing algorithms is also an important task.

In this thesis we are concerned with the explicit isomorphism problem over global fields (i.e., finite extensions of $\mathbb{Q}$ or $\mathbb{F}_q(t)$). When considering algorithms, we restrict ourselves to algorithms (possibly randomized) which run in polynomial time (meaning the running time of the algorithm can be bounded by a

suitably large constant power of the size of the input) which is sometimes allowed to call oracles for certain tasks. We have the following two definitions which were introduced by Rónyai in [59]:

Definition 4. *An f-algorithm is a deterministic polynomial time algorithm which is allowed to call oracles for factoring polynomials over finite fields. The cost of the call is the size of the input.*

Definition 5. *An ff-algorithm is a deterministic polynomial time algorithm which is allowed to call oracles for factoring integers and polynomials over finite fields. In both cases, the cost of the call is the size of the input.*

First we give a brief overview of the known results, then we list the new results of the thesis.

We start with the case where K is an algebraic number field. Let $\mathcal{A}$ be an algebra isomorphic to $M_n(K)$, which is given by structure constants. Then the ff-algorithm from [33] solves the explicit isomorphism problem in polynomial time if three parameters are bounded: the dimension of $\mathcal{A}$ over K, the degree of K over $\mathbb{Q}$ and the discriminant of K. The algorithm was improved in [35], but the boundedness assumptions were not relaxed.

When K is a finite extension of $\mathbb{F}_q(t)$, then only the case where $\mathcal{A} \cong M_2(\mathbb{F}_q(t))$ was known. More precisely, in [11] the authors propose a randomized polynomial time algorithm for finding nontrivial zeros of quadratic forms in 3 variables (which is the same as the aforementioned explicit isomorphism problem in the $n = 2$ case).

Now we turn our attention to the novel results in this thesis. We propose a polynomial time f-algorithm for solving the explicit isomorphism problem if $\mathcal{A} \cong M_n(\mathbb{F}_q(t))$. Note that we do not assume n to be bounded.

We propose an algorithm for the explicit isomorphism problem in the case where $\mathcal{A} \cong M_2(\mathbb{Q}(\sqrt{d}))$. The algorithm is randomized and runs in polynomial time if one is allowed to call an oracle for factoring integers. This provides an efficient algorithm for infinitely many cases of the explicit isomorphism problem over number fields, which was not known before.

The method of the algorithm is reducing the explicit isomorphism problem over $\mathbb{Q}(\sqrt{d})$ to finding nontrivial zeros of quadratic forms in several variables over $\mathbb{Q}$. The reduction procedure works over every field of characteristic different from 2. This motivated us to search for an algorithm which finds a nontrivial zero of a quadratic form in several variables over $\mathbb{F}_q(t)$ (where q is an odd prime power). Such an algorithm was only known in the case where the number of variables is at most 3. We provide an algorithm for finding a nontrivial zero of a quadratic form in four variables. Then we use that algorithm to find a nontrivial zero of a quadratic form in five variables. As a quadratic form in 5

variables is always isotropic, this already provides an algorithm which works for any number of variables. We apply these results to decide equivalence of quadratic forms, and in the case they are isometric, we also output a transition matrix. All these algorithms are randomized and run in polynomial time.

The thesis is divided into seven chapters and an Appendix. The first two chapters are introductory and do not contain any new results. The first chapter contains all the necessary theoretical notions and theorems which are used in later chapters. Some results which are more specific and are not part of general knowledge, are not introduced here but in later chapters, where they are applied (for example a formula on the number of irreducible polynomials in a given residue class). The second chapter is restricted to algorithmic problems. Here we establish the computational model we are working with and describe important algorithms related to the explicit isomorphism problem. In the first two chapters we omitted proofs in most cases. We give however references where the reader can look up the proofs.

The third chapter is mostly based on the paper [38]. In the first section we recall Lenstra's algorithm for finding a reduced basis of an $\mathbb{F}_q[t]$-lattice in $\mathbb{F}_q[t]^n$. In later sections we describe and analyze our algorithm for the explicit isomorphism problem over $\mathbb{F}_q(t)$. In the final section we extend Lenstra's lattice reduction algorithm to vector spaces over $\mathbb{F}_q((\frac{1}{t}))$ (the field of formal Laurent-series over $\mathbb{F}_q$ in $\frac{1}{t}$) and use it to find a nontrivial lattice point in a parallelepiped. This is a result which is not incorporated in the paper [38].

The fourth chapter is based on the paper [39]. In the first section we prove lemmas concerning the isotropy of quadratic forms over $\mathbb{F}_q(t)$ and their completions. In the proceeding sections we propose an algorithm for finding isotropic vectors of quadratric forms over $\mathbb{F}_q(t)$ and apply our algorithms to find an explicit isometry between quadratic forms.

The fifth chapter describes the reduction procedure of [42] (and [43]) for arbitrary fields of characteristic different from 2. Let K be a field whose characteristic is different from 2. Let L be a quadratic extension of K. Let $\mathcal{A} \cong M_2(L)$ be given by structure constants. Then we show that finding an explicit isomorphism between $\mathcal{A}$ and $M_2(L)$ can be reduced to finding nontrivial zeros of quadratic forms over K. Finally, using the algorithm from [63] and our algorithm from Chapter 4, we solve the explicit isomorphism problem for quadratic extensions of $\mathbb{Q}$ and $\mathbb{F}_q(t)$. We note that this general framework could also be useful for other fields as well.

The sixth chapter is devoted to implementations of our algorithms. We implemented some of the main algorithms from Chapters 4 (the algorithm which finds a nontrivial zero of a quadratic form in four variables) and 5 (the case where $K = \mathbb{Q}$) in the computational algebra system MAGMA [47]. We discuss

the implementations in detail and give a brief description of experimental results.

The seventh chapter contains open problems. Several natural questions arose which could be new directions for future research.

The Appendix contains the program codes.

Contents

List of Figures

List of Tables

CHAPTER 1

PRELIMINARIES

This chapter contains all the necessary notions and theorems which are needed in later chapters. We stick to theoretical results. Most of the chapter is based on [50].

1.1 Semisimple algebras

Throughout the section K will be a field and $\mathcal{A}$ a finite-dimensional associative algebra over K. First we would like to state structural results about algebras.

We define elements of special types:

Definition 6. *A pair of elements $a, b \in \mathcal{A}$ is a pair of zero divisors if $ab = 0$. An element $x \in \mathcal{A}$ is nilpotent if there exists a positive integer k such that $x^k = 0$. An element x is strongly nilpotent if for every $y \in \mathcal{A}$, xy is nilpotent. An element $e \in \mathcal{A}$ is an idempotent if $e^2 = e$. Two idempotents e and f are orthogonal if $ef = fe = 0$. An idempotent e is primitive if it cannot be written as sum of two orthogonal nonzero idempotents.*

Definition 7. *The intersection of all the maximal left ideals of $\mathcal{A}$ is called the radical of $\mathcal{A}$. It is denoted by $Rad(\mathcal{A})$.*

Proposition 8. 1. *$Rad(\mathcal{A})$ is an ideal of $\mathcal{A}$*

2. *$Rad(\mathcal{A})$ is the set of strongly nilpotent elements.*

Remark 9. There are several other characterizations of the radical, however we only need these two facts later.

Definition 10. *An algebra $\mathcal{A}$ is semisimple if $Rad(\mathcal{A}) = 0$.*

It is easy to see that the factor $\mathcal{A}/Rad(\mathcal{A})$ is always semisimple. Semisimple algebras admit the following well-known structure theorem due Wedderburn and Artin:

Definition 11. *A finite dimensional algebra $\mathcal{D}$ over K is a division algebra if every nonzero element in $\mathcal{D}$ has a multiplicative inverse.*

Theorem 12. *Let K be a field, and $\mathcal{A}$ a finite dimensional semisimple algebra over K. Then $\mathcal{A}$ is expressible as a direct sum:*

$$\mathcal{A} = \mathcal{A}_1 \oplus \cdots \oplus \mathcal{A}_k \,, \tag{1.1}$$

where the $\mathcal{A}_i$ are exactly the minimal nonzero ideals of $\mathcal{A}$. Moreover, $\mathcal{A}_i$ is isomorphic to $M_{n_i}(D_i)$ where D_i is a division algebra over K.

Remark 13. We did not assume that an algebra contains a multiplicative identity element. However, one can show that a finite dimensional semisimple algebra always contains an identity element (see [55, Proposition 2.1.]).

1.2 Central simple algebras

The theory of central simple algebras over fields is vast. In this section we would only like to recall some basic facts about them which are necessary for the upcoming chapters. All the results stated here can be found in [50, Chapter 12].

First we define the center of an algebra.

Definition 14. *Let $\mathcal{A}$ be an algebra over a field K. Then $Z(\mathcal{A})$, the center of $\mathcal{A}$, consists of those elements which commute with every element of the algebra.*

Note that the identity element 1 is always contained in $Z(\mathcal{A})$. Moreover, $K \cdot 1$ is also contained in the center of $\mathcal{A}$. Identifying K with $K \cdot 1$, we may assume that $K \subseteq Z(\mathcal{A})$.

Definition 15. *An algebra $\mathcal{A}$ over the field K is simple if it contains no proper two-sided ideals.*

The center of a simple algebra is always a field. This motivates the following definition.

Definition 16. *Let K be a field and let $\mathcal{A}$ be a simple algebra over K. Then $\mathcal{A}$ is a central simple algebra over K if $Z(\mathcal{A}) = K$.*

Recall that $K \subseteq Z(\mathcal{A})$, whence $\mathcal{A}$ is a central simple K-algebra if its center is equal to K.

Now we define the Brauer group of a field. The first observation is that the tensor product of central simple algebras is also a central simple algebra over K.

Proposition 17. *Let $\mathcal{A}$ and $\mathcal{B}$ be central simple algebras over K. Then $\mathcal{A} \otimes_K \mathcal{B}$ is also a central simple algebra over K.*

Definition 18. *We call two central simple K-algebras $\mathcal{A}$ and $\mathcal{B}$ Brauer equivalent (or simply* equivalent*) if there exist integers m, l such that $\mathcal{A} \otimes M_m(K) \cong \mathcal{B} \otimes M_l(K)$.*

Another way to formulate Brauer equivalence is the following. A central simple algebra is isomorphic to $M_n(\mathcal{D})$ where $\mathcal{D}$ is a division algebra over the field K. Furthermore, if $\mathcal{A} \cong M_n(\mathcal{D})$ and $\mathcal{A} \cong M_m(\mathcal{D}')$ then $m = n$ and $\mathcal{D} \cong \mathcal{D}'$. Hence one can associate to any central simple algebra a unique division ring. Two central simple K-algebras $\mathcal{A}$ and $\mathcal{B}$ are Brauer equivalent if their associated division rings are isomorphic.

Proposition 19. *Equivalence classes of central simple algebras form a group with the group operation being the tensor product. The identity element is the equivalence class of K. The inverse of a central simple algebra $\mathcal{A}$ is $\mathcal{A}^{op}$ which is its opposite algebra (i.e., it is the same as an abelian group but multiplication is reversed). This group is called the Brauer group of K, and is denoted by $Br(K)$.*

Determining the Brauer group of a field is usually an extremely difficult task. The Brauer group of an algebraically closed field is always trivial. The Brauer group of a finite field is also trivial by a theorem of Wedderburn. A result of Frobenius implies that the Brauer group of the reals is the group with 2 elements generated by Hamilton's quaternions.

If $\mathcal{A}$ is a central simple algebra over K, L is a field extension of K then $\mathcal{A} \otimes_K L$ is a central simple L-algebra. A field L is called a splitting field for $\mathcal{A}$ if $\mathcal{A} \otimes_K L \cong M_n(L)$ for some integer n. It is easy to see that those central simple algebras which are split by L form a subgroup of the Brauer group of K (actually it is isomorphic to a certain cohomology group). Every central simple algebra admits a splitting field, furthermore:

Proposition 20. *Let $\mathcal{A}$ be a central simple algebra over the field K. Then there exists a separable Galois extension L of K which splits $\mathcal{A}$.*

We conclude the section with the classification of automorphisms of central simple algebras which is due to Noether and Skolem:

Theorem 21 (Noether-Skolem)**.** *Let $\mathcal{A}$ be a central simple algebra over K. Then for every K-algebra automorphism σ there exists an invertible element $y \in \mathcal{A}$ such that $\sigma(x) = y^{-1}xy$. Moreover, if $\mathcal{B}$ is simple subalgebra of $\mathcal{A}$ then any algebra homomorphism from $\mathcal{B}$ to $\mathcal{A}$ can be extended to an automorphism of $\mathcal{A}$.*

1.3 Maximal orders

Maximal orders in finite dimensional semisimple algebras are non-commutative analogues of the ring of integers in algebraic number fields. We discuss some basic properties of maximal orders which we need later in Chapter 3. This section is based on [53, Chapter 2]. Throughout the section let R be an integral domain with quotient field K and let $\mathcal{A}$ be an algebra over the field K.

Definition 22. *Let V be a finite dimensional vector space over K. A finitely generated R-submodule of V is a full R-lattice if $KM = \{\sum \alpha_i m_i (\text{finite sum}) |\ \alpha_i \in K,\ m_i \in M\}$ is equal to V.*

Definition 23. *A subring Λ of $\mathcal{A}$ including the identity element of $\mathcal{A}$ is an R-order if it is also a full R-lattice. An order is called maximal if it is maximal with respect to inclusion.*

Example 24.

- In $M_n(\mathbb{Q})$ the subring $M_n(\mathbb{Z})$ is a maximal $\mathbb{Z}$-order
- Let K be an algebraic number field. Then the ring of integers O_K is the only maximal $\mathbb{Z}$-order inside K.
- Let G be a finite group and let $\mathbb{Q}[G]$ be the group algebra over the rationals. Then $\mathbb{Z}[G]$ is a $\mathbb{Z}$-order (but not necessarily maximal).

If L is a full R-lattice in $\mathcal{A}$, then $O_l(L) = \{a \in \mathcal{A} : aL \leq L\}$ is always an order in $\mathcal{A}$ called the left order of L. Every semisimple algebra contains a maximal order. However, the usual proof uses Zorn's lemma, hence is ineffective. An algorithm for computing maximal orders in semisimple algebras over $\mathbb{Q}$ was proposed in [34].

Let R be a principal ideal domain with quotient field K. Next we characterize maximal orders in $M_n(K)$. The first statement is a rephrasing of [53, Theorem 21.6].

Proposition 25. *Let R be a principal ideal domain with quotient field K. Let $\mathcal{A} = \mathrm{Hom}_K(V, V)$ where V is a vector space of dimension n over K. Let L be any full R-lattice in V. Then $\mathrm{Hom}_R(L, L)$, identified with the subring*

$$O_l(L) = \{a \in \mathcal{A} : aL \leq L\}$$

of $\mathcal{A}$, is a maximal R-order in $\mathcal{A}$, and all maximal orders are of this form.

Remark 26. This proposition also holds if R is a Dedekind domain (an integral domain where every nonzero proper ideal is the product of prime ideals).

In terms of matrices, the second statement of the theorem gives the following.

Corollary 27. *Let R be a principal ideal domain with quotient field K. Assume that Λ is a maximal R-order in $M_n(K)$. Then there exists an invertible matrix $P \in M_n(K)$ such that $\Lambda = PM_n(R)P^{-1}$.*

Proof. The theorem with with $V = K^n$ gives that every maximal R-order in $M_n(K)$ is $\mathcal{O}(L)$ for a full R-lattice L in K^n. Let P be a matrix whose columns are an R-basis of L. □

Remark 28. This corollary does not hold in general for Dedekind domains. In fact, if R has class number $c > 1$, then there exist at least c maximal orders in $M_n(K)$ which are pairwise non-conjugates.

We conclude the section by introducing the concept of localization of orders:

Definition 29. *Let R be an integral domain with quotient field K. Let $\mathcal{A}$ be a semisimple algebra over K. Denote by R_P the localization of R at the prime ideal P (embedded in K). Let Λ be an R-order in $\mathcal{A}$. Then the localization of Λ at P is $\Lambda_P = R_P\Lambda$.*

Computationally, the following statement is of high importance:

Proposition 30. *Let R be an integral domain with quotient field K. Let $\mathcal{A}$ be a semisimple algebra over K. Suppose that Λ is an R-order in $\mathcal{A}$. Then Λ is a maximal R-order if and only if Λ_P is maximal R_P order for every prime ideal P of R.*

1.4 Quaternion algebras

The dimension of every central simple algebra is a square [50, Chapter 12], hence every nontrivial central simple algebra has dimension at least 4. Four dimensional central simple algebras are called quaternion algebras. In this section we give a short introduction to quaternion algebras (over fields whose characteristic is different from 2) and their relations with ternary quadratic forms. This is based on [66, Chapter I].

A quaternion algebra has a special special K-basis as stated below:

Proposition 31. *Let $\mathrm{char}(K) \neq 2$ and let $\mathcal{H}$ be a quaternion algebra over K. Then $\mathcal{H}$ has a K-basis $1, u, v, uv$ such that $uv = -vu$ and u^2 and v^2 are nonzero and are in the center of $\mathcal{H}$. We call the basis $1, u, v, uv$ a quaternion basis of $\mathcal{H}$.*

Remark 32. This result is well known, a proof can be found in [66]. There is a similar presentation if $\mathrm{char}(K) = 2$, however we do not need that later on.

From now on we assume that $\mathrm{char}(K) \neq 2$. Since the center of $\mathcal{H}$ is K, we have that $u^2 \in K$ and $v^2 \in K$. This motivates the following notation:

Definition 33. *Let $\mathcal{H}$ be a quaternion algebra over K with quaternion basis $1, u, v, uv$. Let $u^2 = \alpha$ and $v^2 = \beta$. Note that α and β are in K^*. Then we denote H by $\mathcal{H}_K(\alpha, \beta)$.*

$\mathcal{H}_K(\alpha, \beta)$ is well-defined, i.e. all quaternion algebras which have a quaternion basis $1, u, v, uv$ such that $u^2 = \alpha$ and $v^2 = \beta$ are isomorphic.

The Wedderburn-Artin theorem implies that every quaternion algebra is either isomorphic to $M_2(K)$ or it is a division algebra over K. There is a nice criterion which tells us when a quaternion algebra is split (i.e., is isomorphic to $M_2(K)$). First we recall some definitions.

Definition 34. *Let $\mathcal{H}$ be a quaternion algebra over K, with quaternion basis $1, u, v, uv$. Let $s = \lambda_1 + \lambda_2 u + \lambda_3 v + \lambda_4 uv$. Then let $\sigma(s) = \lambda_1 - \lambda_2 u - \lambda_3 v - \lambda_4 uv$ be the* conjugate of s. The map σ is a K-linear map with the following properties:

1. For every $x, y \in \mathcal{H}$, $\sigma(xy) = \sigma(y)\sigma(x)$,
2. For every $x \in \mathcal{H}$, $\sigma(\sigma(x)) = x$
3. σ fixes every element of the center of $\mathcal{H}$.

Maps with these properties are called involutions of the first kind.

We call $tr\ s = s + \sigma(s)$ the *trace of s and $N(s) = s\sigma(s)$ the* norm of s. Note that both $tr\ s$ and $N(s)$ are in K. We call an element $x \in \mathcal{H}$ *traceless* if $tr\ x = 0$. The trace and norm defined this way coincides with the usual reduced trace and norm which can be defined in the following way. Every central simple K-algebra $\mathcal{A}$ of dimension n^2 can be embedded into $M_n(\overline{K})$, where $\overline{K}$ denotes the algebraic closure of K. The reduced trace of an element $\mathcal{A}$ is the trace of the image of that elemenet via the embedding, the reduced norm is the determinant. The Noether-Skolem theorem guarantees that the reduced norm and trace are well-defined. For a more detailed discussion of reduced norms and traces, the reader is referred to [53, Section 9].

Proposition 35. *The following statements are equivalent:*

1. $\mathcal{H}_K(\alpha, \beta) \cong M_2(K)$,
2. *There exists a nonzero element $s \in \mathcal{H}_K(\alpha, \beta)$ such that $N(s) = 0$,*
3. *The quadratic form $x_1^2 - \alpha x_2^2 - \beta x_3^2 + \alpha\beta x_4^2$ has a nontrivial zero over K,*

4. *There exists a nonzero element* $s \in \mathcal{H}_K(\alpha, \beta)$ *such that* $tr\ s = 0$ *and* $N(s) = 0$,

5. *The quadratic form* $\alpha x^2 + \beta y^2 - z^2$ *has a nontrivial zero over* K.

If we write out condition (2) in terms of the quaternion basis we obtain (3). Condition (4), if written out would give the equation $\alpha x^2 + \beta y^2 - \alpha\beta z^2 = 0$ (since every traceless element is the linear combination of u, v and uv). By a change of variables we arrive at (5) . Details can be found in [66] (or [8], [56]). Note that this shows that there is a strong connection between quaternion algebras and quadratic forms in three variables over K.

We recall some facts about cyclic algebras, which are higher dimensional analogues of quaternion algebras. This is based on [50][Chapter 15].

Definition 36. *A central simple algebra* $\mathcal{A}$ *over the field* K *is called cyclic if there is a maximal subfield* L *of* $\mathcal{A}$ *such that* L *is a cyclic extension of* K *(i.e.* $L|K$ *is a Galois extension whose Galois group is cyclic).*

Observe that quaternion algebras are also cylic algebras. Indeed, the quaternion algebra $\mathcal{H}_K(\alpha, \beta)$ either contains the subfield $K(\sqrt{\alpha})$ if α is not a square in K, or is a full matrix algebra (if α is a square) which is naturally cyclic. The next proposition states that cyclic algebras admit a presentation similar to the presentation of quaternion algebras:

Proposition 37. *Let* L *be a cyclic extension of* K *and let* σ *be a generator of the Galois group of the extension. Let* $\mathcal{A}$ *be an algebra containing* L *as a maximal subfield. Then there exists an invertible element* $u \in \mathcal{A}$ *such that:*

1. $\mathcal{A} = \oplus_{i=1}^{n} u^i L$,

2. $u^{-1} d u = \sigma(d)$ *for all* $d \in L$,

3. $u^n = a$ *for some* $a \in K$.

We use the notation $\mathcal{A} = (L, \sigma, a)$.

We conclude the section with a proposition similar to Proposition 35 for cyclic algebras. This is important to us because it establishes a connection between cyclic algebras and norm equations.

Proposition 38. *The cyclic algebra* $\mathcal{A} = (L, \sigma, a)$ *is isomorphic to* $M_n(K)$ *if and only if* a *is in the image of the norm map* $N_{L|K} : L \to K$.

1.5 Quadratic forms and Quadratic spaces

This section is based on Chapter I of [45]. Here F will denote a field such that char $F \neq 2$.

A *quadratic form* over F is a homogeneous polynomial Q of degree two in n variables (say, $x_1, \ldots, x_n$) for some n. Two quadratic forms are called *equivalent* if they can be obtained from each other by a homogeneous linear change of the variables. By such a change we mean that each variable x_j is substituted by the polynomial $\sum_{i=1}^{n} b_{ij}x_i$ ($j = 1, \ldots, n$). The $n \times n$ matrix $B = (b_{ij})$ over F has to be invertible as otherwise there is no appropriate substitution in the reverse direction. The *matrix* of Q is the (unique) symmetric n by n matrix $A = (a_{ij})$ with $Q(x_1, \ldots, x_n) = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{i,j}x_ix_j$. We will also refer to this as the Gram matrix of the quadratic form. The *determinant* of a quadratic form is the determinant of its matrix. We call Q *regular* if its matrix has nonzero determinant and *diagonal* if its matrix is diagonal. We say that Q is *isotropic* if the equation $Q(x_1, \ldots, x_n) = 0$ admits a nontrivial solution and *anisotropic* otherwise. Two quadratic forms with Gram matrices A_1 resp. A_2 are then equivalent if and only if there exists an invertible n by n matrix $B \in M_n(F)$, such that $A_2 = B^T A_1 B$ (or, equivalently, $A_1 = {B^{-1}}^T A_2 B^{-1}$). Here B is just the matrix of the change of variables defined above. We will use the term *transition matrix* for such a B. Two regular unary quadratic forms ax^2 and bx^2 are equivalent if and only if a/b is a square in F^*. In other words, equivalence classes of regular unary quadratic forms correspond to the elements of the factor group $F^*/(F^*)^2$.

Every quadratic form is equivalent to a diagonal one, see the discussion of Gram–Schmidt-orthogonalization in the context of quadratic spaces below and in Subsection 4.1.1. A regular diagonal quadratic form $Q(x_1, x_2) = a_1x_1^2 + a_2x_2^2$ is isotropic if and only if $-a_2/a_1$ is a square in F^*. Binary quadratic forms that are regular and isotropic at the same time are called *hyperbolic*. If (β_1, β_2) is a nontrivial zero of Q then $\gamma = 2(a_1\beta_1^2 - a_2\beta_2^2)$ is nonzero and the substition $x_1 \leftarrow \beta_1 x_1 + \frac{\beta_1}{\gamma}x_2$, $x_2 \leftarrow \beta_2 x_1 - \frac{\beta_2}{\gamma}x_2$ provides an equivalence of Q with the form x_1x_2. Another, diagonal standard hyperbolic form is $x_1^2 - x_2^2$. The standard forms x_1x_2 and $x_1^2 - x_2^2$ are equivalent via the substitution $x_1 \leftarrow \frac{1}{2}x_1 + \frac{1}{2}x_2$, $x_2 \leftarrow \frac{1}{2}x_1 - \frac{1}{2}x_2$ (the inverse of this substitution is $x_1 \leftarrow x_1 + x_2$, $x_2 \leftarrow x_1 - x_2$).

Now we introduce the notion of *quadratic spaces*. This offers a coordinate-free approach to quadratic forms. A quadratic space over F is a pair (V, h) consisting of a vector space V over F and a symmetric bilinear function $h : V \times V \to F$. Throughout this section all vector spaces will be finite dimensional. To a quadratic form Q having Gram matrix A one can associate the bilinear function $h(u, v) = u^T Av$ on F^n. Conversely, if (V, h) is an n-dimensional quadratic space

then for any basis $v_1, \ldots, v_n$ we can define its *Gram matrix* $A = (a_{ij})$ with respect to the given basis by putting $a_{ij} = h(v_i, v_j)$. Then $Q(x_1, \ldots, x_n) = \underline{x}^T A \underline{x}$ is a quadratic form where $\underline{x}$ stands for the column vector $(x_1, \ldots, x_n)^T$ of variables. The quadratic form obtained from h using another basis will be a form equivalent to Q. Let (V, h) and (V', h') be quadratic spaces. Then a linear bijection $\phi : V \to V'$ is an *isometry* if $h'(\phi(v_1), \phi(v_2)) = h(v_1, v_2)$ for every $v_1, v_2 \in V$. We say that (V, h) and (V', h') are *isometric* if there is an isometry $\phi : V \to V'$. Equivalent quadratic forms give isometric quadratic spaces and to isometric quadratic spaces equivalent quadratic forms are associated. Moreover, the following holds. Let (V, h) and (V', h') be quadratic spaces. Let $v_1, \ldots, v_n$ be a basis of V and let $v'_1, \ldots, v'_n$ be a basis of V'. Suppose that ϕ is an isometry between V and V'. Then $\phi(v_i) = \sum_{j=1}^{n} b_{ij} v'_j$ where $b_{ij} \in \mathbb{F}$. Let A be the Gram matrix of h in the basis $v_1, \ldots, v_n$ and let A' be the Gram matrix of h' in the basis $v'_1, \ldots, v'_n$. If $B \in M_n(F)$ is equal to the matrix (b_{ij}) then $A = B^T A' B$.

Let (V, h) be a quadratic space. We say that two vectors u and v from V are *orthogonal* if $h(u, v) = 0$. An orthogonal basis is a basis consisting of pairwise orthogonal vectors. The well-known Gram–Schmidt-orthogonalization procedure provides an algorithm for constructing orthogonal bases (we will discuss some details in the context of quadratic spaces over $\mathbb{F}_q(t)$ in Subsection 4.1.1). With respect to an orthogonal basis, the Gram matrix is diagonal. Therefore the Gram–Schmidt-procedure gives a way of computing diagonal forms equivalent to given quadratic forms. The *orthogonal complement* of a subspace $U \leq V$ is the subspace

$$U^{\perp} = \{v : h(u, v) = 0 \text{ for every } u \in U\}.$$

The subspace $V^{\perp}$ is called the *radical* of (V, h). (V, h) is called regular if its radical is zero. A quadratic space is regular if and only if at least one of, or equivalently, each of the quadratic forms associated to it using various bases is regular.

The *orthogonal sum* of (V, h) and (V', h') is the quadratic space $(V \oplus V', h \oplus h')$ where

$$h \oplus h'((v_1, v'_1), (v_2, v'_2)) = h(v_1, v_2) + h'(v'_1, v'_2)$$

(here $v_1, v_2 \in V$ and $v'_1, v'_2 \in V'$). The inner version of this is a decomposition of V into the direct sum of two subspaces V and V' with $V \leq V'^{\perp}$ and $V' \leq V^{\perp}$. An orthogonal basis gives a decomposition into the orthogonal sum of one-dimensional quadratic spaces.

A nonzero vector in a quadratic space is called *isotropic* if it is orthogonal to itself. Isotropic vectors correspond to nontrivial zeros of quadratic forms. A quadratic space is isotropic if it admits isotropic vectors and *anisotropic* otherwise. A quadratic space (V, h) is *totally isotropic* if h is identically zero on

$V \times V$. This is equivalent to that every (nonzero) vector in V is isotropic (note that char $F \neq 2$). Every subspace $U \leq V$ in a quadratic space (V,h) is also a quadratic space with the restriction of h to U. A subspace of V is called isotropic, anisotropic, totally isotropy, etc. if it is isotropic, anisotropic, totally isotropic as a quadratic space with the restriction of h. A quadratic space can be decomposed as an orthogonal sum of a totally isotropic subspace (this is necessarily the radical of the whole space) and a regular space (this can actually be any of the direct complements of the radical). A two-dimensional quadratic space is called a *hyperbolic plane* if it is regular and isotropic. Such spaces correspond to hyperbolic binary forms.

Theorem 39 (Witt)**.** *Let (V,h) be a quadratic space over F. Then V can be decomposed as the orthogonal sum of V_0, a totally isotropic space, V_h, which is an orthogonal sum of hyperbolic planes, and an anisotropic space V_a. Such a decomposition is called a Witt decomposition of (V,h) and the number $\frac{1}{2}\dim(V_h)$ is called the Witt index of (V,h). Here V_0 is the radical. The Witt index and the isometry class of the anisotropic part V_a do not depend on the particular Witt decomposition. In turn, two quadratic spaces are isometric if and only if their radical have the same dimension, their Witt indices coincide and their anisotropic parts are isometric.*

A proof of this theorem can be found in [45, Chapter I, Theorem 4.1.]. There is another interpretation of the Witt index concerning totally isotropic subspaces.

Proposition 40. *Let (V,h) be a regular quadratic space with Witt index m. Then the dimension of every maximal totally isotropic subspace is m.*

The proof of this proposition can be found in [45, Chapter I, Corollary 4.4.]. By the following fact, the Witt decomposition has implications to equivalence of quadratic forms.

Proposition 41. *Two regular quadratic spaces (V,h) and (V',h') having the same dimension are isometric if and only if the orthogonal sum of (V,h) and $(V',-h')$ can be decomposed as an orthogonal sum of hyperbolic planes.*

The proof of this proposition can be found in [24, Proposition 2.46.].

Thus deciding isotropy of quadratic spaces (or, equivalently, deciding equivalence of quadratic forms) can be reduced to computing Witt decompostions. In Chapter 5 we will show that such a reduction exists even for computing isometries (and for computing transition matrices) explicitly.

1.6 Lattices

The use of lattices in number theory is an extremely powerful tool. One of the first results which was proven in such a way is probably Minkowski's proof of

the finiteness of the ideal class group of a number field. These techniques are also extremely useful algorithmically, one of the most important examples being the LLL-algorithm which has various applications (for example factorization of polynomials with rational coefficients) [46].

Here we recall some of the basic notions and introduce the concept of a reduced basis for integer lattices and lattices over $\mathbb{F}_q[t]$.

First we consider $\mathbb{Z}$-lattices. We recall the definition of a full lattice. L is a full $\mathbb{Z}$-lattice if $L = \{\alpha_1\mathbf{b}_1 + \cdots + \alpha_m\mathbf{b}_m |\ \alpha_i \in \mathbb{Z}\}$ where $\mathbf{b}_1, \ldots, \mathbf{b}_m$ is a basis (over $\mathbb{R}$) in $\mathbb{R}^m$.

An important algorithmic question in the theory of $\mathbb{Z}$-lattices is the following. Given a lattice L by a basis (i.e. linearly independent (over $\mathbb{R}$) generators of L as an abelian group) can one compute a shortest vector in L (in terms of the usual euclidean norm). This task is NP-hard, which was proven by Ajtai [1]. One of the reasons why this is difficult is the following. It may happen that a short vector, when expressed as a $\mathbb{Z}$-linear combination of the given basis, has large coefficients. A good way to avoid, or at least decrease, this phenomenon is to compute another basis where the size of the coefficients can be controlled. We prove the following lemma:

Lemma 42. *Let Γ be a full lattice in $\mathbb{R}^m$. Suppose that we have a basis $\mathbf{b}_1, \ldots, \mathbf{b}_m$ of Γ over $\mathbb{Z}$ such that*

$$|\mathbf{b}_1| \cdot |\mathbf{b}_2| \cdots |\mathbf{b}_m| \leq c_m \cdot \det(\Gamma) \tag{1.2}$$

holds for a real number $c_m > 0$. Suppose that

$$\mathbf{c} = \sum_{i=1}^{m} \gamma_i \mathbf{b}_i \in \Gamma, \quad \gamma_i \in \mathbb{Z}.$$

Then we have $|\gamma_i| \leq c_m \frac{|\mathbf{c}|}{|\mathbf{b}_i|}$ for $i = 1, \ldots, m$.

Proof. From Cramer's rule we obtain

$$|\gamma_i| = \frac{|\det(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{i-1}, \mathbf{c}, \mathbf{b}_{i+1}, \ldots, \mathbf{b}_m)|}{\det(\Gamma)} \leq \frac{|\mathbf{b}_1| \cdots |\mathbf{b}_{i-1}| \cdot |\mathbf{c}| \cdot |\mathbf{b}_{i+1}| \ldots |\mathbf{b}_m|}{\det(\Gamma)} =$$

$$= \frac{|\mathbf{c}|}{|\mathbf{b}_i|} \cdot \frac{|\mathbf{b}_1| \cdots |\mathbf{b}_{i-1}| \cdot |\mathbf{b}_i| \cdot |\mathbf{b}_{i+1}| \cdots |\mathbf{b}_m|}{\det(\Gamma)} \leq \frac{|\mathbf{c}|}{|\mathbf{b}_i|} \cdot c_m \cdot \frac{\det(\Gamma)}{\det(\Gamma)} = c_m \cdot \frac{|\mathbf{c}|}{|\mathbf{b}_i|}.$$

□

We remark that the LLL algorithm gives a basis with $c_m = 2^{m(m-1)/4}$ in formula (1.2), see [46].

Now we turn our attention to $\mathbb{F}_q[t]$-lattices.

Definition 43. *Let $f, g \in \mathbb{F}_q[t]$. Then we set $|\frac{f}{g}| = \deg(f) - \deg(g)$. We will refer to $|.|$ as the* valuation *(or* degree*) of an element of $\mathbb{F}_q(t)$. We set $|0| = -\infty$. Let $\mathbf{v} = (v_1, \ldots, v_m)^T \in \mathbb{F}_q(t)^m$. Then the* valuation *(or* degree*) of the vector $\mathbf{v}$ is $|\mathbf{v}| = \max(|v_1|, \ldots, |v_m|)$.*

Definition 44. *L is a* full lattice *in $\mathbb{F}_q(t)^m$ if $L = \{\alpha_1\mathbf{b}_1 + \cdots + \alpha_m\mathbf{b}_m |\ \alpha_i \in \mathbb{F}_q[t]\}$ where $\mathbf{b}_1, \ldots, \mathbf{b}_m$ is a basis (over $\mathbb{F}_q(t)$) in $\mathbb{F}_q(t)^m$.*

Definition 45. *Let $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m \in \mathbb{F}_q(t)^m$. Then the* orthogonality defect *denoted by $OD(\mathbf{b}_1, \ldots, \mathbf{b}_m)$ is defined as $OD(\mathbf{b}_1, \ldots, \mathbf{b}_m) = \sum_{i=1}^m |\mathbf{b}_i| - |\det(B)|$ where B is the matrix whose columns are the $\mathbf{b}_i$ $(i = 1, \ldots, m)$.*

The following lemma is from [44]. However, there it is stated in a slightly weaker form than we need it in this thesis. So we state and prove the lemma here as well. The proof is also from [44].

Lemma 46. *Let $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m \in \mathbb{F}_q(t)^m$ be linearly independent and $\mathbf{a} = \sum_{i=1}^m \alpha_i\mathbf{b}_i$ where $\alpha_i \in \mathbb{F}_q[t]$. Then the following holds for every i:*

$$|\alpha_i| \leq |\mathbf{a}| + OD(\mathbf{b}_1, \ldots, \mathbf{b}_m) - |\mathbf{b}_i| \tag{1.3}$$

Proof. Consider the α_i as unknowns. Then we have m linear equations and m variables so we can use Cramer's rule. Note that $\{\mathbf{b}_i\}_{i=1}^m$ is a basis so the determinant of the coefficient matrix B is non-zero. By Cramer's rule α_i is equal to the quotient of two determinants. In other words α_i multiplied by the determinant of the lattice is equal to the determinant where the ith column of B is switched to $\mathbf{a}$. Since these two sides are equal, their valuations are equal also (on both sides we have elements from $\mathbb{F}_q(t)$). Note that the valuation of a determinant can be bounded from above by the sum of the valuations of its columns. To formalize this last sentence:

$$\begin{aligned} |\alpha_i| + |\det(B)| &\leq |\mathbf{b}_1| + |\mathbf{b}_2| + \cdots + |\mathbf{b}_{i-1}| + |\mathbf{b}_{i+1}| + \cdots + |\mathbf{b}_m| + |\mathbf{a}| \\ &= \sum_{i=1}^m |\mathbf{b}_i| - |\mathbf{b}_i| + |\mathbf{a}|. \end{aligned}$$

After rearranging we obtain the result. □

An implication of this lemma is the following. If we have a vector with small valuation, then the coefficients corresponding to a basis are also small, if the orthogonality defect of the basis is small. This also suggests that an ideal basis is one whose orthogonality defect is 0. This motivates the following definition.

Definition 47. *A basis $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m \in \mathbb{F}_q(t)^m$ is called* reduced *if the orthogonality defect $OD(\mathbf{b}_1, \ldots, \mathbf{b}_m) = 0$.*

These are the basic notions which we need later for our algorithms. Algorithmic results concerning finding a reduced basis will be stated in Chapter 3.

1.7 Valuations and the local-global principle

In this section we recall some basic facts about valuations, local and global fields and introduce the local-global principle. This section is based on [48, Chapter 2].

Definition 48. *Let K be a field. An* exponential valuation *is a function $v : K \to \mathbb{R} \cup \infty$ with the following properties:*

1. $v(x) = \infty \iff x = 0$
2. $v(xy) = v(x) + v(y)$
3. $v(x + y) \geq min(v(x), v(y))$

where for every $a \in \mathbb{R}, a < \infty$, $a + \infty = \infty$ and $\infty + \infty = \infty$. A valuation is called discrete *if its image in $\mathbb{R}$ is a discrete subgroup.*

Sometimes an exponential valuation is just called valuation in the literature. However, we would like to distinguish this notion from the valuation defined in Definition 43, which is actually the negative of an exponential valuation. The reason we distinguish these two notions is that in terms of lattices we follow the terminology of [44] which is computationally more natural.

A related notion is an absolute value on a given field:

Definition 49. *Let K be a field. An absolute value is a function $\|.\| : K \to \mathbb{R}$ with the following properties:*

1. $\|x\| = 0 \iff x = 0$
2. $\|xy\| = \|x\|\|y\|$
3. $\|x + y\| \leq \|x\| + \|y\|$.

An absolute value is non-archimedean *if the following stronger version of (3) holds:*

$$\|x + y\| \leq max(\|x\|, \|y\|).$$

Two absolute values $\|.\|_1$ and $\|.\|_2$ are called equivalent *if $\|.\|_2 = \|.\|_1^r$ for some $r > 0$ real number.*

It is easy to see that if v is an exponential valuation than s^{-v} (where $s > 1$ is a real number) is an absolute value (by the convention that $s^{-\infty} = 0$). However, not all absolute values arise this way (such an absolute value is always non-archimedean). We give certain important examples of absolute values and then state two theorems that classify absolute values on $\mathbb{Q}$ and $\mathbb{F}_q(t)$ respectively (up to equivalence).

Example 50.
- Let $K = \mathbb{Q}$. Then every $\frac{a}{b} \in \mathbb{Q}$, with a, b being integers satisfying $(a, b) = 1$, can be written in the form $\frac{a}{b} = p^k \frac{a'}{b'}$ where p is a prime number, k is an integer and neither a' nor b' is divisible by p. The the *p-adic absolute value* defined on $\mathbb{Q}$ is given by $\|\frac{a}{b}\| = p^{-k}$. This absolute value comes from the exponential valuation defined by $v(\frac{a}{b}) = k$.
- Let $K = \mathbb{F}_q(t)$. Then the same construction as above is valid by replacing the prime number p with a monic irreducible polynomial $f \in \mathbb{F}_q[t]$.
- There is another important exponential valuation on $\mathbb{F}_q(t)$ defined by the following relation: let $\frac{f}{g} \in \mathbb{F}_q(t)$ (here $f, g \in \mathbb{F}_q[t]$), then $v_\infty(\frac{f}{g}) = deg(g) - deg(f)$ where $deg(f)$ denotes the degree of the polynomial f. Note that the degree of the zero polynomial is defined to be $-\infty$. This is called the valuation at infinity. Hence the corresponding absolute value is defined as $\|\frac{f}{g}\|_\infty = q^{deg(f)-deg(g)}$.

Theorem 51 (Ostrowski)**.** *Any absolute value on $\mathbb{Q}$ is equivalent to exactly one of the following three:*

1. *The trivial absolute value, i.e., $\|x\| = 1$ for all $0 \neq x \in \mathbb{Q}$,*
2. *A p-adic absolute value for some prime number p,*
3. *The usual absolute value, i.e. $\|x\| = x$ if $x \geq 0$ and $\|x\| = -x$ if $x < 0$.*

The next statement is the polynomial analogue of Ostrowski's theorem ([24, Theorem 3.15, (ii)]):

Theorem 52. *Any absolute value on $\mathbb{F}_q(t)$ is equivalent to exactly one of the following three:*

1. *The trivial absolute value, i.e. $\|x\| = 1$ for all $0 \neq x \in \mathbb{F}_q(t)$,*
2. *A f-adic absolute value for some monic irreducible polynomial f,*
3. *The absolute value $\|.\|_\infty$.*

Let $K = \mathbb{Q}$ or $K = \mathbb{F}_q(t)$. A nontrivial absolute value makes these fields a metric space. However, neither or of them is complete (i.e., not every Cauchy sequence is convergent) with respect to a nontrivial absolute value. We may complete them with the following standard procedure. We consider two Cauchy sequences equivalent if their difference converges to zero. We may define addition and multiplication naturally on the equivalence classes of Cauchy sequences. If we complete $\mathbb{Q}$ with respect to the usual absolute value, we obtain the field of

real numbers $\mathbb{R}$. The completion of $\mathbb{Q}$ with respect to a p-adic absolute is the field of p-adic numbers $\mathbb{Q}_p$. If we complete $\mathbb{F}_q(t)$ with respect to $\|-\|_\infty$, we get the field $\mathbb{F}_q((\frac{1}{t}))$, the field of formal Laurent-series in $\frac{1}{t}$. For a monic irreducible polynomial f, the completion is just the field of formal Laurent series in f. This justifies the terminology for $\|.\|_\infty$.

We recall some definitions about complete valued fields.

Let K be a field equipped with a non-archimedean absolute value. Then the elements whose absolute value is at most 1 form a subring O, called the *valuation ring* of K. This has a maximal ideal P, consisting of those elements whose absolute value is strictly smaller than 1. The factor O/P is a field, called the residue field of K and is denoted by k.

If K is equipped with a discrete exponential valuation, then the ring O is a principal ideal domain and P is the unique maximal ideal of O. The generator π of the ideal P is called the prime element of K. A polynomial $f(x) \in O[x]$ is called primitive if not every coefficient of f is divisible by π.

The next statement shows a close relation between decomposing polynomials with coefficients from O to decomposing polynomials over k. This is extremely useful, as in a lot of cases (all the cases considered later) the residue field is a finite field.

Lemma 53 (Hensel)**.** *Let K be a complete valued field, O its valuation ring and P the unique maximal ideal in O. If a primitive polynomial $f(x) \in O[x]$ admits modulo P a factorization*

$$f(x) \equiv \overline{g}(x)\overline{h}(x) \mod O$$

into relatively prime polynomials $\overline{g}, \overline{h} \in \kappa[x]$ then $f(x)$ admits a factorization

$$f(x) = g(x)h(x)$$

into polynomials $g, h \in O[x]$ such that $\deg(g) = \deg(\overline{g})$ *and*

$$g(x) \equiv \overline{g}(x) \mod P, \quad h(x) \equiv \overline{h}(x) \mod P$$

We will use this fact later on.

Now we would like to describe a principle for finding zeros of multivariate polynomials over $\mathbb{Z}$ or $\mathbb{F}_q[t]$.

Let $F = \mathbb{Q}$ or $F = \mathbb{F}_q(t)$. If $f \in F[x_1, \ldots, x_n]$, we may look at the equation $f(x_1, \ldots, x_n) = 0$. A natural way to show that the equation is not solvable is to show that it is not solvable in a certain completion of F. What happens however if the equation $f(x_1, \ldots, x_n) = 0$ is solvable in every completion? Can we conclude that it is also solvable in F^n? The first result in this direction is due to Hasse:

Theorem 54. *A non-degenerate quadratic form over $\mathbb{Q}$ is isotropic over $\mathbb{Q}$ if and only if it is isotropic over every completion of $\mathbb{Q}$.*

In other words the answer to the previous question is in the affirmative for homogeneous polynomials of degree 2. A similar result for $F = \mathbb{F}_q(t)$ (where q is odd) is due to Rauter, a doctoral student of Hasse. This principle is called the *local-global principle*.

It is to be noted that the local-global principle holds also for quadratic forms over global fields (finite extensions of $\mathbb{Q}$ and $\mathbb{F}_q(t)$). Another important case where a local-global principle is valid is the splitting of central simple algebras over algebraic number fields:

Theorem 55 (Albert-Brauer-Hasse-Noether)**.** *Let K be an algebraic number field and $\mathcal{A}$ a central simple algebra of degree d (i.e., of dimension d^2) over K. Denote by K_v the completion of K with respect to the valuation v. If for every v*

$$\mathcal{A} \otimes_K K_v \cong M_d(K_v),$$

then $\mathcal{A} \cong M_d(K)$.

This statement has a similar flavour as the Hasse-Minkowski theorem but it concerns isomorphisms of central simple algebras, not zeros of homogeneous polynomials. However, the splitting of a central simple algebra can always be associated to the existence of a nontrivial zero of a homogeneous polynomial (a statement similar to Propositions 35 and 38 is true for higher dimensional central simple algebras as well).

Unfortunately, the local-global principle does not hold in general. The smallest degree counterexample is due to Selmer:

Example 56 (Selmer). The equation $3x^3 + 4y^3 + 5z^3 = 0$ is solvable in every completion of $\mathbb{Q}$ but has no rational solution.

Understanding the failure of the local-global principle for higher degree forms is an important research problem in modern number theory.

CHAPTER 2

COMPUTING THE STRUCTURE OF ALGEBRAS

In this chapter we give a brief overview of some of the most important known algorithms for computing the structure of algebras over finite fields and global fields. In the first section we define the computational model we will be working with. In the rest of the chapter we give a brief summary of certain algorithms which are relevant to the topic.

2.1 The computational model

We specify the way we consider an algebra.

Definition 57. *Let $\mathcal{A}$ be an algebra over K. Let $a_1, \ldots, a_l$ be a K-basis of $\mathcal{A}$. Then every $a_i a_j$ can be expressed as the linear combination of the a_i:*

$$a_i a_j = \sum_{k=1}^{l} \gamma_{i,j,k} a_k \tag{2.1}$$

The $\gamma_{i,j,k}$ are called structure constants.

We consider an algebra to be given as a collection of structure constants.

Example 58. Let $G = \mathbb{Z}/3\mathbb{Z}$ be the cyclic group with 3 elements. Let g be the generator of G. Let $\mathbb{Q}[G]$ denote the group algebra of G over the rationals. This is an associative algebra with the $\mathbb{Q}$-basis $1, g, g^2$. Then the structure constant representation is given by the Cayley-table of the group.

The following proposition says that a structure constant representation can be thought of as a representation by matrices.

Proposition 59. *Let $\mathcal{A}$ be an algebra over the field K such that $\dim_K \mathcal{A} = n$. Then $\mathcal{A}$ is isomorphic to a subalgebra of $M_{n+1}(K)$.*

Proof. We shall use the regular representation. Multiplication from the left by an element x is a K-linear transformation from $\mathcal{A}$ to itself. Hence it can be represented by an $n \times n$ matrix. This is injective if $\mathcal{A}$ has an identity element. If not then we adjoin one to $\mathcal{A}$ in the following way. Let $\mathcal{A}^*$ denote the set of pairs (a, λ) where $a \in \mathcal{A}$ and $\lambda \in K$. Addition and multiplication is defined as follows:

$$(a_1, \lambda_1) + (a_2, \lambda_2) = (a_1 + a_2, \lambda_1 + \lambda_2)$$

$$(a_1, \lambda_1)(a_2, \lambda_2) = (a_1 a_2 + \lambda_1 a_2 + \lambda_2 a_1, \lambda_1 \lambda_2)$$

Note that $\mathcal{A}$ naturally injects into $\mathcal{A}^*$ and $(0, 1)$ is the identity element of $\mathcal{A}^*$. This proves our claim. □

Remark 60. The previous construction of A^* is called the Dorroh extension.

We also would like to remark that from a representation by matrices, a structure constant representation can be obtained. Indeed, we multiply two matrices and express it as a linear combination of the basis elements. This can be accomplished by solving a system of linear equations (if $\dim_K(\mathcal{A}) = n$, then we have n equations and n variables). The coefficients in the linear combination are the structure constants (this is the definition of structure constants).

Now as we have established our computational model we are ready to state our goals. Assume an algebra $\mathcal{A}$ over a field K is given by structure constants:

1. Compute the radical of $\mathcal{A}$,

2. Compute the Wedderburn decomposition of $\mathcal{A}/Rad(\mathcal{A})$ (i.e., its minimal ideals),

3. Compute an explicit isomorphism between its simple components $\mathcal{A}_i$ and $M_n(D_i)$.

Naturally, we have to specify what kind of fields we are working with. In this thesis we are concerned with finite fields, algebraic number fields and finite (separable) extensions of $\mathbb{F}_q(t)$, the field of rational functions over the finite field $\mathbb{F}_q$.

2.2 Computing the radical

In this section we sketch the algorithm from [10] for computing the radical. We assume that $\mathcal{A}$ is a subalgebra of a full matrix algebra over the field K. By Proposition 59 and the discussion succeeding it, this is equivalent to $\mathcal{A}$ being given by structure constants. Throughout the section K will denote either an algebraic number field, a finite field or $\mathbb{F}_q(t)$. Also let $\dim_K \mathcal{A} = n$.

For any $x \in \mathcal{A}$ let $tr\ x$ denote the trace of x (note that x is matrix and the trace function is invariant under conjugation) and let $\chi(x,t) = \sum_{i=1}^{n}(-1)^i a_{i,x} t^i$ be the characteristic polynomial of x. We define the set $\mathcal{A}_1 \subseteq \mathcal{A}$ in the following way. An element x is in $\mathcal{A}_1$ if $tr\ x = 0$ and for every $y \in \mathcal{A}$ we have that $tr\ xy = 0$. Since the map $y \mapsto tr\ xy$ is linear, computing a basis for $\mathcal{A}_1$ boils down to solving a system of linear equations over K. Therefore computing a basis for $\mathcal{A}_1$ can be accomplished in polynomial time.

It is easy to see that $\mathcal{A}_1$ is an ideal of $\mathcal{A}$. Also $\mathcal{A}_1$ contains the radical of $\mathcal{A}$ by Proposition 8. In addition, one has that $Rad(\mathcal{A}) = Rad(\mathcal{A}_1)$.

We have the following structural result:

Lemma 61. *Let a be a matrix with entries from K. If $tr\ a^j = 0$ for every positive integer j then the following hold:*

1. *If* char $K = 0$ *then a is nilpotent,*
2. *if* char $K = p$ *then the multiplicities of nonzero eigenvalues (in the characteristic polynomial) of a are divisible by p,*
3. *every element in $\mathcal{A}_1$ has this property.*

Proof. The third statement is trivial by the definition of $\mathcal{A}_1$. Now let $\lambda_1, \ldots, \lambda_r$ be the nonzero eigenvalues of a with multiplicities $m_1, \ldots, m_r$. Then the condition that $tr\ a^s = 0$ translates into the following equation:

$$\sum_{i=1}^{r} m_i \lambda_i^s = 0. \tag{2.2}$$

Consider the $r \times r$ matrix M with the entry λ_j^s at the (s,j) position. Since M is a Vandermonde matrix, and the λ_i are distinct, M is non-singular. However, the vector $(m_1, \ldots, m_r)$ is in its nullspace. Hence if char $K = 0$ all the m_i are zero. This means that a has no nonzero eigenvalues (over the algebraic closure of K), so it is nilpotent. If char $K = p$ then all the m_i are divisible by p. □

An immediate consequence of this result is that if char $K = 0$ then $\mathcal{A}_1 = Rad(\mathcal{A})$. This means that in zero characteristic we already know how to compute the radical in polynomial time.

From now on we assume that char $K = p$. Our goal is to define a descending chain of subalgebras $\mathcal{A}_1 \supseteq \cdots \supseteq \mathcal{A}_j = Rad(\mathcal{A})$ which are effectively computable. First we introduce a notation:

Definition 62. *Let $x \in \mathcal{A}$. Then we define the modified characteristic polynomial to be $\overline{\chi(x,t)} = det(tx + I)$, where I is the identity matrix. Let $\overline{\chi(x,t)} = \sum_{i=0}^{n} C(i,x)t^i$. In other other words we denote by $C(i,x)$ the ith coefficient of the modified characteristic polynomial.*

Now we define the A_i in the following way. $\mathcal{A}_0 = \mathcal{A}$. Suppose we have already defined $\mathcal{A}_{i-1}$. Then we define $\mathcal{A}_i$ to consist of those elements $x \in \mathcal{A}_{i-1}$ for which $C(p^{i-1}, x) = 0$ and $C(p^{i-1}, xy) = 0$ for every $y \in \mathcal{A}_{i-1}$. Note that this coincides with our definition of $\mathcal{A}_1$ if we choose $i = 1$. We have the following theorem:

Theorem 63. *The following statements hold for $i \geq 1$:*

1. *Each $\mathcal{A}_i$ is an ideal in $\mathcal{A}_{i-1}$*
2. *The multiplicities of nonzero eigenvalues of elements of $\mathcal{A}_i$ are multiples of p^i*
3. *For $x, y \in \mathcal{A}_i$ we have that $C(p^i, x + y) = C(p^i, x) + C(p^i, y)$ and $C(k, x) = 0$ for all k not divisible by p^i.*

Remark 64. An immediate consequence of this is that if $i > \log_p n$ then $\mathcal{A}_i = Rad(\mathcal{A})$. Indeed, since then all the elements of $\mathcal{A}_i$ are nilpotent and the radical of $\mathcal{A}$ is contained in $\mathcal{A}_i$.

The proof of Theorem 63 can be found in [10]. Now we show how one can compute $\mathcal{A}_i$ from $\mathcal{A}_{i-1}$ in deterministic polynomial time. Note that by the previous remark this is the only thing we have to show since the number of rounds is $O(\log\ n)$.

Observe that $C(p^i, \lambda x) = \lambda^{p^i} C(p^i, x)$ where $\Lambda \in K$. Hence the function C is semilinear. Hence checking whether $a \in \mathcal{A}_i$ it is enough to check that $C(p^{i-1}, ab_l) = 0$ where $b_1, \ldots, b_k$ is a basis of $\mathcal{A}_{i-1}$. Therefore, by the third statement of Theorem 63, checking the properties of being in $\mathcal{A}_i$ boils down to solving a system of semilinear equations over K. If K is a finite field of characteristic p, then the map $\lambda \mapsto \lambda^p$ is an automorphism, hence solving a system of of semilinear equations boils down to solving a system of linear equations. When $K = \mathbb{F}_q(t)$ than a method for solving a system of semilinear equations is described in the paper [36] (solving a sytem of semilinear equations can be reduced to solving a system of linear equations over the subfield $\mathbb{F}_q(t)^{p^j}$). If char $K = 0$, then we have already established that $\mathcal{A}_1 = Rad(\mathcal{A})$.

The second statement of Theorem 63 shows that for large enough i we have that $\mathcal{A}_i$ is equal to the radical and the third statement gives us a way to compute $\mathcal{A}_i$ from $\mathcal{A}_{i-1}$.

2.3 Computing the Wedderburn decomposition

Assume now that $\mathcal{A}$ is a semisimple algebra over K. Our goal is to compute the minimal ideals of $\mathcal{A}$. This question is closely related to factoring univariate polynomials over K. Indeed, consider a squarefree polynomial $f \in K[x]$. Then $\mathcal{A} = K[x]/(f)$ is a semisimple algebra (it does not even contain any nilpotent elements as f was square-free). Computing the minimal ideals of $\mathcal{A}$ is equivalent to computing the irreducible factors of f. Hence a natural question would be the following. Can computing the Wedderburn decomposition of $\mathcal{A}$ be reduced to factoring polynomials over (finite extensions of) K. By reduction we mean a polynomial time reduction (possibly randomized).

First we sketch such an algorithm in the case of finite fields. Afterwards we outline how one has to modify the algorithm in the case of number fields. Finally we state a result of Ivanyos, Rónyai and Szánto [36], which deals with the case of $K = \mathbb{F}_q(t)$ (actually they even consider the field of rational functions in several variables).

Theorem 65. *Let $\mathcal{A}$ be a finite dimensional semisimple algebra over $\mathbb{F}_q$. Then there exists a randomized polynomial time algorithm for computing the minimal ideals of $\mathcal{A}$.*

Remark 66. Note that polynomials over $\mathbb{F}_q$ can be factored in polynomial time by a randomized algorithm [4].

Proof. This is an outline of the proof, which can be found in [22]. Let $Z(\mathcal{A})$ denote the center of $\mathcal{A}$. Let $\mathcal{A} = \mathcal{A}_1 \oplus \cdots \oplus \mathcal{A}_k$ where the $\mathcal{A}_i$ are the minimal ideals. Observe that

$$Z(\mathcal{A}) = Z(\mathcal{A}_1) \oplus \cdots \oplus Z(\mathcal{A}_k) \tag{2.3}$$

and that $Z(\mathcal{A}_i)\mathcal{A} = \mathcal{A}_i$. The center of an algebra can be computed efficiently. Indeed, let $b_1, \ldots, b_n$ be a basis of $\mathcal{A}$. Then $a \in \mathcal{A}$ is in the center of $\mathcal{A}$ if $ab_i - b_i a = 0$ for every i. Therefore finding such an a is equivalent to solving a system of homogeneous linear equations (note that the maps $a \mapsto ab_i - b_i a$ are linear and we have to compute the intersection of their kernels). Hence we may assume that $\mathcal{A}$ is commutative.

We describe a method which decomposes $\mathcal{A}$ into the direct sum of smaller ideals $\mathcal{A} = I \oplus J$ unless $k = 1$. We iterate this cutting procedure until the Wedderburn decomposition is found. Assume that $a_1, \ldots, a_n$ is a basis of $\mathcal{A}$.

Assume that the subalgebra generated by $1, a_1, \ldots, a_i$ is a field K_i. We set $K_0 = \mathbb{F}_q$. If $i = n$ then $\mathcal{A}$ is a field and we are done. If $i < n$ then let f be the minimal polynomial of a_{i+1} over K_i. The polynomial f can be calculated in polynomial time. If f is irreducible then $K_i(a_{i+1})$ is a field. If not, then $f = gh$ $(g, h \in K_i[t])$ where $gcd(g, h) = 1$.

Then there exists $g', h' \in K_i[t]$ such that $gg' + hh' = 1$. Finally $I = \mathcal{A}g(a_{i+1})$ and $J = \mathcal{A}h(a_{i+1})$ will be suitable. □

Now we turn our attention to the case $K = \mathbb{Q}$. If we look at the previous proof closely, than we see that this cutting procedure works in exactly the same fashion as in the case of finite fields, only we need to factor polynomials with rational coefficients. However, when we iterate the cutting procedure it may happen that the size of the basis of the ideals grows rapidly. Therefore the bit complexity may not be polynomial. This phenomenon is amended by the following lemma from [22]:

Lemma 67. *Suppose that I is an ideal of $\mathcal{A}$ given by the basis $c_1, \ldots, c_k$. Let $a_1, \ldots, a_n$ be the basis of $\mathcal{A}$. Let $c_i = \sum_{j=1}^{n} \lambda_{i,j} a_j$. Assume that all the numerators and denominators of the $\lambda_{i,j}$ is bounded by N. Also if $\gamma_{i,j,k}$ are the structure constants for the basis $a_1, \ldots, a_n$, then $\left|\gamma_{i,j,k}\right| \leq K$. Then there exists a polynomial $p(x, y)$ and a polynomial time algorithm (polynomial in n, N and K) which computes a new basis of I with size bounded by $p(n, K)$.*

The previous cutting procedure combined with these reduction steps yields a polynomial time algorithm for this problem. In addition this idea extends naturally to algebraic number fields.

Finally we conclude our chapter by stating a result of [36]:

Theorem 68. *Let K be a finite extension of $\mathbb{F}_q(t)$ and let $\mathcal{A}$ be a semisimple algebra over K. Then there exists a deterministic polynomial time reduction from computing the minimal ideals of $\mathcal{A}$ to factoring polynomials over finite fields.*

Remark 69. An alternative approach (when K is either an algebraic number field or a finite extension of $\mathbb{F}_q(t)$) is the following. One may assume that $\mathcal{A}$ is an n-dimensional commutative and separable algebra over the field K (a semisimple algebra $\mathcal{A}$ over K is separable if for every field extension L of K, $\mathcal{A} \otimes_K L$ is a semisimple L-algebra). As discussed in Section 1.4, $\mathcal{A}$ can be embedded into $M_n(\overline{K})$, where $\overline{K}$ is the algebraic closure of K. An element $a \in \mathcal{A}$ is called a splitting element if it has n distinct eigenvalues (as a matrix in $M_n(\overline{K})$). Let f be the minimal polynomial of a. Then one has that $\mathcal{A} \cong K[x]/(f(x))$. Therefore finding the Wedderburn decomposition of $\mathcal{A}$ reduces to factoring f. A splitting element can be found by a randomized algorithm if the ground field K is large enough ([18]), as most elements are splitting elements. This method was can be derandomized by a textbook method which is described for example in [29].

2.4 Simple algebras

Now we assume that $\mathcal{A} \cong M_n(K)$, where $\mathcal{A}$ is given by structure constants. Our goal is to compute an explicit isomorphism between $\mathcal{A}$ and $M_n(K)$. This is the main question that motivated all the new results in this thesis.

We start with some general observations for arbitrary fields. Then we continue by a short survey on the current situation of the problem for various fields.

We recall some facts about idempotent matrices. If e is a nontrivial idempotent (i.e., not equal to 0 or 1), then its minimal polynomial is $x^2 - x$. This has 2 distinct roots over every field, namely 0 and 1. Hence e is diagonalizable and its diagonal form contains 0-s and 1-s in the diagonal. An idempotent is primitive if and only if its diagonal entries are 0-s except for one. Naturally, the rank of an idempotent is the number of 1-s in the diagonal. If e is an idempotent of rank k, then e can be decomposed into the sum of k pairwise orthogonal ($e_i e_j = e_j e_i = \delta_{ij} e_i$) primitive idempotents. The characteristic polynomial of e is then $(x-1)^k x^{n-k}$.

Proposition 70. *Let $\mathcal{A} \cong M_n(K)$ and let e be a primitive idempotent in $\mathcal{A}$. Let $V = \mathcal{A}e$. Then $dim_K V = n$ and the left action of A on V provides an isomorphism between $\mathcal{A}$ and $M_n(K)$.*

Proof. Let V' be the K vector space of matrices in $M_n(K)$ which contains nonzero element only in the first column. Since e is a primitive idempotent it is a conjugate of

$$\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix}$$

Hence the vector spaces V and V' are conjugates, thus have the same dimension. The second part of the claim is trivial. □

A consequence of the previous proposition is that it suffices to find a primitive idempotent in $\mathcal{A}$. Actually, if one already has an element $r \in \mathcal{A}$ of rank 1 then a primitive idempotent can be computed efficiently. Indeed, since a right identity element of the left ideal $\mathcal{A}r$ is a primitive idempotent which can be computed by solving a system of linear equations. Assume now that we have a zero divisor r in $\mathcal{A}$ of rank k. Then, in a similar fashion, we can compute an idempotent of rank k.

Proposition 71. *Let e be an idempotent of rank k in $\mathcal{A} \cong M_n(K)$. Then $e\mathcal{A}e \cong M_k(K)$.*

Proof. There exists an invertible matrix f such that

$$f^{-1}ef = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & 0 \end{pmatrix}$$

where the number of 1-s is exactly k. Naturally, $e\mathcal{A}e$ is isomorphic to $f^{-1}e\mathcal{A}ef$. Also since $\mathcal{A}$ is isomorphic to $f\mathcal{A}f^{-1}$ we have that $e\mathcal{A}e \cong f^{-1}ef\mathcal{A}f^{-1}ef$. Hence $e\mathcal{A}e$ is isomorphic to

$$\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & 0 \end{pmatrix} M_n(K) \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & 0 \end{pmatrix}$$

which is isomorphic to $M_k(K)$. □

The proposition above immediately suggests the following. If we are able to compute zero divisors then we can compute primitive idempotents as well. Indeed, one first constructs an idempotent of some rank k. Then one reduces the problem to finding a primitive idempotent to a full matrix algebra of smaller dimension. This idea works for finite fields as we will see later in this chapter. However, for number fields and function fields, the size of structure constants obtained by this operation may blow up. Therefore this idea usually only works if n is assumed to be bounded.

Now we turn our attention to finite fields. The result by Rónyai [57] shows that there exists a randomized polynomial time algorithm which solves the problem. Actually we can say little bit more about the complexity. Recall the following definition:

Definition 72. *An f-algorithm is a deterministic algorithm which is allowed to call oracles for factoring polynomials over finite fields. The cost of the call is the size of the input.*

Theorem 73. *Let $\mathcal{A} \cong M_n(\mathbb{F}_q)$ be given by structure constants. Then there exists a polynomial time f-algorithm which finds a zero divisor in $\mathcal{A}$.*

Remark 74. In the case of finite fields we can compute a primitive idempotent from a zero divisor using the previous procedure.

We sketch the main ideas of the algorithm. First, it computes a maximal subfield $\mathbb{F}$ of $\mathcal{A}$ (or returns a zero divisor). Then the Frobenius automorphism of $\mathbb{F}$ extends to an $\mathbb{F}_q$-algebra automorphism of $\mathcal{A}$, thus by the Noether-Skolem theorem the Frobenius automorphism can be obtained by conjugation via an element $c \in \mathcal{A}$. Therefore we obtain a cyclic algebra presentation of $\mathcal{A}$. Finding a zero divisor in a cyclic algebra reduces to solving a norm equation (which in the case of finite fields can reduced to factoring a suitable polynomial).

For bounded n there is an improvement of this result [32]. That algorithm is deterministic but its complexity is exponential in n (it runs in polynomial time however, if we restrict that the prime divisors of n are bounded by some constant).

Now we turn our attention to the case where $K = \mathbb{Q}$. The first interesting result in this topic is unfortunately a negative one [56]:

Theorem 75. *Assume that $\mathcal{A} \cong M_2(\mathbb{Q})$ is given by structure constants. Then there is a randomized polynomial time reduction from factoring square-free integers to finding a zero divisor in $\mathcal{A}$.*

Note that there is no current polynomial time algorithm for factoring integers (unless we consider quantum computers as well). However, one may ask the following question. Do the two tasks have the same complexity, i.e. what can we do if we are allowed to call an oracle for factoring integers. Recall the following definition:

Definition 76. *An ff-algorithm is a deterministic algorithm which is allowed to call oracles for factoring integers and polynomials over finite fields. Th cost of the call is the size of the input.*

The first positive result was given by Ivanyos and Szántó [37]:

Theorem 77. *Let $\mathcal{A} \cong M_2(\mathbb{Q})$ be given by structure constants. Then there exists a polynomial time ff-algorithm which finds a zero divisor in $\mathcal{A}$.*

The algorithm uses the following extremely important result by Ivanyos and Rónyai [34]:

Theorem 78. *Let $\mathcal{A}$ be a semisimple algebra over $\mathbb{Q}$. Then there exists a polynomial time ff-algorithm which finds a maximal order in $\mathcal{A}$.*

Then Ivanyos and Szántó use lattice reduction for indefinite forms to compute zero divisors. This paper introduced LLL-type algorithms to this topic which became a fruitful contribution. A different algorithm for the same task was proposed by Cremona and Rusin [15].

Then de Graaf et al. [28] and Pilnikova [51] solved the cases where $n = 3$ and $n = 4$ respectively using norm equation solvers. However, the complexity of solving norm equations is not yet fully known. For instance it is not known whether norm equations over higher degree number fields can be solved by polynomial time ff-algorithms.

Here we give a brief summary of the result of Ivanyos, Rónyai and Schicho [33] which generalizes all the previous results. First we state the result and then we go into details:

Theorem 79. *Let K be an algebraic number field, with discriminant D and degree d over $\mathbb{Q}$. Let $\mathcal{A} \cong M_n(K)$ be given by structure constants. Then there exists an ff-algorithm which computes an explicit isomorphism between $\mathcal{A}$ and $M_n(K)$. The running time of the algorithm is exponential in d,n and $\log D$ and polynomial in the size of the structure constants.*

This algorithm solves the explicit isomorphism problem in polynomial time if we assume d, n and D to be bounded. We describe the algorithm in the case where $K = \mathbb{Q}$. The idea is similar for any number field but it is fairly more technical. The following is the main structural result:

Theorem 80. *Let $\mathcal{A}$ be a $\mathbb{Q}$-subalgebra of $M_n(\mathbb{R})$ isomorphic to $M_n(\mathbb{Q})$. Let Λ be a maximal $\mathbb{Z}$-order in $\mathcal{A}$. Then there exists an element $C \in \Lambda$ which has rank 1 as a matrix, and whose Frobenius norm $\|C\|$ is less than n.*

Remark 81. The Frobenius norm of $X \in M_n(\mathbb{R})$ is $\|X\| = \sqrt{tr\ X^T X}$. Let $\mathcal{A}$ be an algebra isomorphic to $M_n(\mathbb{Q})$. We may define the Frobenius norm of an element of $\mathcal{A}$ by embedding $\mathcal{A}$ into $M_n(\mathbb{R})$ and assigning to each element its Frobenius norm as a real matrix.

Proof. We sketch the proof here (a more detailed version can be found in [33]).

Observe that every maximal $\mathbb{Z}$-order Λ of $\mathcal{A}$ is a conjugate (by an invertible matrix $P \in M_n(\mathbb{R})$) of $M_n(\mathbb{Z})$. This follows from Corollary 27 (every maximal $\mathbb{Z}$-order of $M_n(\mathbb{Q})$ is a conjugate of $M_n(\mathbb{Z})$ since $\mathbb{Z}$ is a principal ideal domain) and the Noether-Skolem theorem. Let $\Lambda' = M_n(\mathbb{Z})$. Then we have that

$$\Lambda = P\Lambda' P^{-1}.$$

for some $P \in M_n(\mathbb{R})$. Let $Q = P/(|\det P|)^{1/n}$. Then we have $\Lambda = Q\Lambda' Q^{-1}$, where the determinant of Q is 1.

Let ρ be the set of all integer matrices which have 0 everywhere except in the first column. Observe that ρ is a lattice of determinant 1 in the vector space S of all real matrices having nonzeros only in the first column. The lattice $L = Q\rho$ will be a lattice in S, with determinant 1 (as the determinant of Q is equal to 1).

The key idea of the proof is the application of Minkowski's theorem on lattice points in convex bodies to L in S. The volume of the ball of radius $\sqrt{n}$ in S centered at the zero matrix is more than 2^n, as it contains 2^n internally disjoint copies of the n-dimensional unit cube, and more. Hence there exists an element $B \in \rho$ such that QB is a nonzero matrix whose length is less than $\sqrt{n}$. Since B was of rank 1 so is QB (as it is nonzero).

Observe that by a "transpose" of this argument (with Q^{-1} in the place of Q), there exists a nonzero integer matrix B', which is zero everywhere except in the first row, $B'Q^{-1}$ is nonzero, and has Frobenius norm less than $\sqrt{n}$.

Now consider

$$C = PBB'P^{-1} = QBB'Q^{-1}.$$

C is in Λ because $BB' \in M_n(\mathbb{Z})$. It has length less than n because the Frobenius norm is submultiplicative. Obviously, C has rank at most 1, as B and B' are of rank 1. Finally, from the shape of B and B' we see, that $BB' \neq 0$, hence rank BB' = rank $C = 1$. This finishes the proof. □

We may interpret the result in the following way. Every maximal order contains a short vector which is a rank 1 element. How can we turn this observation into an algorithm? We compute a maximal order in $\mathcal{A}$. Then we embed $\mathcal{A}$ into $M_n(\mathbb{R})$ in order to obtain a norm on $\mathcal{A}$. We scan through all the short vectors to find a rank 1 element. In order to make this efficient we need lattice reduction techniques. Since our lattice vectors may have nonrational coordinates we use an approximate version of the LLL-algorithm. We also need the following observation from [33], that too short vectors in a maximal order are automatically zero divisors:

Lemma 82. *Let $X \in M_n(\mathbb{R})$ be a matrix such that* det X *is an integer, and $\|X\| < \sqrt{n}$. Then X is a singular matrix.*

Now we outline the algorithm:

---

1. Compute a maximal order Λ in $\mathcal{A}$ using the Ivanyos-Rónyai algorithm [34]

2. Compute an embedding of $\mathcal{A}$ into $M_n(\mathbb{R})$. One can use the randomized polynomial time algorithm of [18] or its derandomiezd version [29] (this is a slightly modified version of the procedure described at the end of Section 2.3, as we need an embedding into $M_n(\mathbb{R})$, not $M_n(\mathbb{C})$). This way we have a Euclidean norm on $\mathcal{A}$: for $X \in \mathcal{A}$ we put $\|X\| = \sqrt{tr\ X^T X}$. Thus Λ can be viewed as a full lattice in $\mathbb{R}^m$, where $m = n^2$. The length $|\mathbf{c}|$ of a lattice vector $\mathbf{c}$ is just the Frobenius norm of $\mathbf{c}$ as a matrix.

3. Compute a rational approximation B_0 of the lattice basis of Λ and then compute a reduced basis $\mathbf{b}_1, \ldots, \mathbf{b}_m$ of the lattice $\Lambda \subset \mathbb{R}^m$ by applying the LLL algorithm to the rational approximation B_0.

4. If there exists an i such that $|\mathbf{b}_i| < \sqrt{n}$, then by Lemma 82 $\mathbf{b}_i$ is a zero divisor. If $\mathbf{b}_i$ has rank 1 then we output $\mathbf{b}_i$. If $\mathbf{b}_i$ has rank $k > 1$, then we proceed as described at the beginning of the section (we reduce to the explicit isomorphism problem of $M_k(\mathbb{Q})$).

5. At this point we know that $|\mathbf{b}_i| \geq \sqrt{n}$ holds for every i. Scan through all integral linear combinations $C' = \sum_{i=1}^{m} \gamma_i \mathbf{b}_i$, where the γ_i are integers, $|\gamma_i| \leq c_m \frac{n}{|\mathbf{b}_i|} \leq c_m \sqrt{n}$ (this is the point where we use that $\mathbf{b}_1, \ldots, \mathbf{b}_m$ is a reduced basis) until a C is found with rank $C = 1$ (checking whether an element has rank one can be accomplished in polynomial time via the method described at the beginning of the section). The existence of such a C is guaranteed by Theorem 80. Output this C.

The discussion of the running time of this algorithm can be found in [33]. The reason we discussed this algorithm in slightly more detail than the previous one, is that this result is the starting point for the next chapter which is devoted to solving the explicit isomorphism problem in the case $K = \mathbb{F}_q(t)$. We conclude by saying that there is an improvement of the result from [33] due to Ivanyos, Lelkes and Rónyai [35]. However, the general question for arbitrary n, d and D remains open.

CHAPTER 3

COMPUTING EXPLICIT ISOMORPHISMS WITH FULL MATRIX ALGEBRAS OVER $\mathbb{F}_q(t)$

In the previous chapter we outlined an algorithm for computing the radical of an algebra over $\mathbb{F}_q(t)$ and referred to [36] where the authors compute the Wedderburn decomposition of a semisimple algebra over $\mathbb{F}_q(t)$. In this section we deal with the following problem. Let $\mathcal{A} \cong M_n(\mathbb{F}_q(t))$ be given by structure constants. Compute an explicit isomorphism between $\mathcal{A}$ and $M_n(\mathbb{F}_q(t))$. As we have seen in the previous chapter, this task is equivalent to finding a primitive idempotent in $\mathcal{A}$. This chapter is based on [38].

The starting point of this result is the paper from Ivanyos, Rónyai and Schicho on splitting full matrix algebras over the rational numbers [33]. In that paper, which was outlined in the previous chapter, the authors describe an algorithm for finding a primitive idempotent in a full matrix algebra over $\mathbb{Q}$ which is given by structure constants. They compute a maximal order and then embed the algebra into $M_n(\mathbb{R})$ using the splitting element method from Eberly [18] which gives a way to define a norm on the original algebra. They show that every maximal order contains a rather small rank 1 element. Finally, using lattice reduction techniques they scan through all the small elements. This scanning part is rather time consuming and that is why the running time is exponential in n, the degree of the matrix algebra.

When started research in the problem our initial goal was to imitate this process. Which means computing a maximal order and scanning through short vectors using lattice reduction. However, as it turns out, stronger structural results simplify matters considerably. One can show that short vectors in a maxi-

mal order form a finite ring which contains a primitive idempotent from $\mathcal{A}$. So instead of scanning through all the elements we can compute its structure using the algorithm from [57] and use this to our advantage. The main result of this chapter is the following:

Theorem 83. *Let $\mathcal{A}$ be isomorphic to $M_n(\mathbb{F}_q(t))$, and given by structure constants. Then there exists a polynomial (in n and in the size of the structure constants) time f-algorithm which finds an explicit isomorphism between $\mathcal{A}$ and $M_n(\mathbb{F}_q(t))$.*

Again, lattices this time over $\mathbb{F}_q(t)$, play an important role in our algorithm. For the purpose of computing an explicit isomorphism we only need a slight extension of Lenstra's original reduction algorithm [44]. However, his results can be generalized further. We give such a generalization in the last section and also provide an application for this result.

The structure of this chapter is as follows. In the first section we describe Lenstra's lattice reduction algorithm ([44]) and extend it slightly to lattices in $\mathbb{F}_q(t)^m$. In Section 2 we propose an algorithm which finds a maximal order in an algebra $A \cong M_n(\mathbb{F}_q(t))$. In Section 3 we prove our main structural theorem and give a description of our algorithm together with its complexity analysis. In the final section we extend Lenstra's algorithm to lattices in $\mathbb{F}_q((1/t))^m$. We apply it to find lattice vectors in parallelepipeds.

3.1 Lattice reduction

Let L be a lattice in $\mathbb{F}_q(t)^m$ generated by $\mathbf{b}_1, \ldots, \mathbf{b}_m \in \mathbb{F}_q[t]^m$. Recall (Section 1.6.), that the orthogonality defect of a basis $\mathbf{b}_1, \ldots, \mathbf{b}_m \in \mathbb{F}_q[t]^m$ is defined as $OD(\mathbf{b}_1, \ldots, \mathbf{b}_m) = \sum_{i=1}^m |b_i| - \det(L)$ and that a basis is reduced, if its orthogonality defect is 0. Lenstra ([44]) proposed an algorithm which finds a reduced basis in a lattice given by another lattice basis. Now we outline this algorithm:

Theorem 84. *Let L be a lattice in $\mathbb{F}_q(t)^m$ given by the basis $\mathbf{b}_1, \ldots, \mathbf{b}_m$ where $\mathbf{b}_i \in \mathbb{F}_q[t]^m$. Assume that $|\mathbf{b}_i| < B$ for all i. Then there exists an algorithm which takes $O(Bm^3(OD(\mathbf{b}_1, \ldots, \mathbf{b}_m) + 1))$ arithmetic operations in $\mathbb{F}_q$ and returns a reduced basis of L.*

Proof. Denote by b_{ij} the jth coordinate of $\mathbf{b}_i$. Let us assume that an integer $k \in \{0, 1, \ldots, m\}$ is given (with the convention $|\mathbf{b}_0| = -\infty$) such that the following hold:

1. $|\mathbf{b}_i| \leq |\mathbf{b}_j|$ for $1 \leq i < j \leq k$,
2. $|\mathbf{b}_k| \leq |\mathbf{b}_j|$ for $k < j \leq m$,

3. $|b_{ii}| \geq |b_{ij}|$ for $1 \leq i \leq k$ and $i < j \leq m$,

4. $|b_{ii}| > |b_{ij}|$ for $1 \leq j < i \leq k$.

Initially these conditions are satisfied for $k = 0$. If $k = m$ then the basis $\mathbf{b}_1, \ldots, \mathbf{b}_m$ is reduced (one has the terms of largest valuation in the diagonal of the matrix of the lattice, and all other expansion terms have strictly smaller valuation). Suppose that $k < m$. Renumber $\mathbf{b}_{k+1}, \ldots, \mathbf{b}_m$ in a way that the minimum of $|\mathbf{b}_{k+1}|, \ldots, |\mathbf{b}_m|$ becomes $|\mathbf{b}_{k+1}|$ (i.e. the one with the smallest index should be the shortest). Let a_{ij} be the coefficient of $t^{|\mathbf{b}_i|}$ in b_{ij} for $1 \leq i \leq k+1$ and $1 \leq j \leq k$. It follows from the third and fourth condition on k that $a_{ii} \neq 0$ for $1 \leq i \leq k$ and that $a_{ij} = 0$ for $1 \leq j < i \leq k$. This implies that a solution $(r_1, \ldots, r_m) \in \mathbb{F}_q^m$ of the following triangular system of linear equations exists:

$$\sum_{i=1}^{k} a_{ij} r_i = a_{(k+1)j} \text{ for } 1 \leq j \leq k. \tag{3.1}$$

We put $\mathbf{b}^*_{k+1} = \mathbf{b}_{k+1} - \sum_{i=1}^{k} r_i \mathbf{b}_i t^{|\mathbf{b}_{k+1}| - |\mathbf{b}_i|}$. Then $|\mathbf{b}^*_{k+1}| \leq |\mathbf{b}_{k+1}|$ and the first two conditions imply that $\mathbf{b}^*_{k+1} \in \mathbb{F}_q[t]^m$. Furthermore equation (3.1) implies that $|b^*_{k+1i}| < |\mathbf{b}_{k+1}|$ for $1 \leq i \leq k$. We distinguish two cases. If $|\mathbf{b}^*_{k+1}| = |\mathbf{b}_{k+1}|$ then we replace $\mathbf{b}_{k+1}$ by $\mathbf{b}^*_{k+1}$, we permute the coordinates of $\mathbf{b}_1, \ldots, \mathbf{b}_m$ in such a way that $|b_{k+1,k+1}| = |\mathbf{b}_{k+1}|$ (this does not affect the first k coordinates), and finally we replace k by $k+1$. If $|\mathbf{b}^*_{k+1}| < |\mathbf{b}_{k+1}|$ then we replace $\mathbf{b}_{k+1}$ by $\mathbf{b}^*_{k+1}$ and we replace k by the largest index $l \in \{0, 1, \ldots, k\}$ such that $|\mathbf{b}_l| \leq |\mathbf{b}_{k+1}|$. Now all 4 four conditions are satisfied and we proceed are algorithm from here. Now we prove the bound on the running time of the algorithm. Let $S = \sum_{i=1}^{m} |\mathbf{b}_i|$. Then while passing through the main loop S either remains unaltered (first case) or decreases by 1 (second case). Since the value of k is increased by 1 in the first case, a particular value of S can only occur at most $(n+1)$ times. On the other hand S can have at most $OD(\mathbf{b}_1, \ldots, \mathbf{b}_m)$+1 values, whence the number of passes through the main loop is $O(m(OD(\mathbf{b}_1, \ldots, \mathbf{b}_m) + 1))$. The result now follows from the fact that solving the system of linear equations (3.1) takes $O(k^2)$ operations in $\mathbb{F}_q$, while computing b^*_{k+1} takes $O(mkB)$ operations in $\mathbb{F}_q$. □

This result can be extended to find a reduced basis of an arbitrary full lattice in $\mathbb{F}_q(t)^m$. Let us assume that we have a basis $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m$ in $\mathbb{F}_q(t)^m$. Let L be the $\mathbb{F}_q[t]$-lattice generated by these vectors and let B be the matrix with columns $\mathbf{b}_1, \ldots, \mathbf{b}_m$. Let γ be the least common multiple of all the denominators of the entries of B. We consider the lattice L' generated by $\gamma\mathbf{b}_1, \ldots, \gamma\mathbf{b}_m$. Note that $L' \in \mathbb{F}_q[t]^m$. So using Lenstra's algorithm one can find a reduced basis $\mathbf{c}_1, \ldots, \mathbf{c}_m$ in L'. Note that $|\det L'| = |\det L| + m|\gamma|$. This implies that choosing $\mathbf{b}'_i = \frac{1}{\gamma}\mathbf{c}_i$

we get a reduced basis of L. Since the orthogonality defect of $\mathbf{b}_1, \ldots, \mathbf{b}_m$ is the same as the orthogonality defect of $\gamma\mathbf{b}_1, \ldots, \gamma\mathbf{b}_m$, we obtain the following:

Corollary 85. *Let $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m$ be a basis in $\mathbb{F}_q(t)^m$ and let L be the $\mathbb{F}_q[t]$-lattice they generate. Let γ be the least common multiple of all the denominators for the entries of $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m$. Let $M = |\gamma| + \max_{1\leq i\leq m}(|\mathbf{b}_i|)$ and let $M' = \max(M, 1)$. Then there exists an algorithm which takes $O(m^3 M'(OD(\mathbf{b}_1, \ldots, \mathbf{b}_m) + 1))$ arithmetic operations in $\mathbb{F}_q$ and returns a reduced basis of L.*

Given an integer k, the set of elements of the lattice whose valuation is smaller than k is a finite dimensional $\mathbb{F}_q$-vector space (this is a consequence of Lemma 46), and a basis of this vector space can also be computed efficiently.

The algorithm of Corollary 85 finds a reduced basis of a lattice which is given by a basis. However, one can ask the following question: what happens if the lattice is only given by an $\mathbb{F}_q[t]$-module generating system? In such situations an algorithm by Paulus [49, Algorithm 3.1.] is applicable. It finds a reduced basis of a lattice in $\mathbb{F}_q(t)^m$ given by a system of generators. We shall make use of the fact that the valuations of the reduced basis obtained by Paulus' algorithm will not be greater than those of the given generators.

3.2 Maximal orders over $\mathbb{F}_q[t]$

3.2.1 Preliminaries

In this subsection we assume that R is a principal ideal domain with quotient field K and $\mathcal{A}$ is a central simple algebra isomorphic to $M_n(K)$. Recall that an *R-order* in $\mathcal{A}$ is a full R-lattice which is at the same time a subring of $\mathcal{A}$ containing the identity element. Maximal orders are orders maximal with respect to inclusion. Recall also that in this specialized setting every maximal order is the conjugate of the standard maximal order $M_n(R)$ (Corollary 27).

Our eventual aim is to construct a maximal R-order in $M_n(\mathbb{K})$. We will construct an initial order Λ_0 in a rather straightforward way and iteratively enlarge it. Strictly speaking, our initial object Λ_0 will not be an order. We say that an R-subalgebra Λ of $\mathcal{A}$ is an *almost R-order* in $\mathcal{A}$ if it is a full R-lattice in $\mathcal{A}$. Thus orders are almost orders containing the identity element of $\mathcal{A}$. It turns out that if Λ_0 is an almost R-order, then the R-lattice generated by Λ_0 and the identity element of $\mathcal{A}$ is an R-order.

Discriminants enable us to control the depth of chains of (almost) orders and will also be useful in representing orders efficiently. The *reduced trace*, $tr\ a$, of an element a of an $\mathcal{A}$ is simply the trace of a as an n by n matrix (this is well defined by the Noether-Skolem theorem, see Section 1.4 in this thesis). To

compute reduced traces efficiently, it is not necessary to know an isomorphism $\mathcal{A} \cong M_n(K)$. If n is not divisible by the characteristic of K, then $tr\ a$ is $\frac{1}{n}$ times the trace of the image of a under the regular representation of a. In general, the reduced trace can be computed by taking an appropriate coefficient of the nth root of the characteristic polynomial of the regular representation. This is because the regular representation of $\mathcal{A}$ decomposes as a direct sum of n copies of the standard n-dimensional (irreducible) representation.

The *bilinear trace form* on $\mathcal{A}$ is the symmetric bilinear function $(a, b) \mapsto tr\ ab$. As the matrix corresponding to an element of an almost R-order Λ is similar to a matrix with entries of R, the reduced trace of any element of Λ is from R. The discriminant $d(\Lambda)$ can be defined as the principal ideal of R generated by the determinant of the Gram matrix $\left(tr\ b_i b_j\right)_{i,j=1}^{n^2}$ where $b_1, \ldots, b_{n^2}$ are an R-basis for Λ. It is nonzero and independent of the choice of the basis. We can loosely think of $d(\Lambda)$ as an element of R, defined up to a unit of R. As the bilinear trace form is non-degenerate, we have the following (see [53, Exercise 10.3]).

Proposition 86. *Let Λ and Γ be almost R-orders in $\mathcal{A}$ such that $\Lambda \subseteq \Gamma$. Then $d(\Gamma)|d(\Lambda)$ and $\Lambda = \Gamma$ if and only if $d(\Gamma) = d(\Lambda)$.*

The following statement gives an R-lattice as an upper bound for R-orders containing a given almost order. An extension to more general rings R is used in the proof of [53, Theorem 10.3]. For orders over principal ideal domains it is stated explicitly in [34, Proposition 2.2]. As we need a slight generalization to almost orders, we give a proof for completeness.

Proposition 87. *Let Λ and Γ be almost R-orders in $\mathcal{A}$ such that $\Lambda \subseteq \Gamma$. Then $\Gamma \subseteq \frac{1}{d}\Lambda$ where $d = d(\Lambda)$.*

Proof. Let $b_1, \ldots, b_{n^2}$ be an R-basis for Λ. Then an element $a \in \Gamma$ can be written as $a = \sum_{i=1}^{n^2} \alpha_i b_i$ with $\alpha_i \in K$ $(i = 1, \ldots, n^2)$. For $j = 1, \ldots, n^2$ put $\beta_j = tr\ ab_j$. Then the elements β_j are in R because the elements ab_j are in the almost order Γ which is contained in an R-order and hence have reduced trace from R. By linearity, we have $\sum_i \alpha_i tr\ b_i b_j = \beta_j$. Cramer's rule gives that each α_i is a quotient of an element of R and d, which means that $a \in \frac{1}{d}\Lambda$. □

An algorithmic consequence is that it is possible to represent R-orders containing a given almost order Λ as submodules of the factor module $\frac{1}{d}\Lambda/\Lambda$. This will be particularly useful when $R = \mathbb{F}_q[t]$, in which case this factor is an $n^2 \deg d$-dimensional vector space over $\mathbb{F}_q$.

Our algorithm for computing maximal orders is an adaptation of the method proposed by Ivanyos and Rónyai for the case $R = \mathbb{Z}$ in [34]. The method is discussed in the context of global fields in the Ph. D. thesis of Ivanyos [31]. For

completeness, we include proofs of statements that are not rigorously proved for general principal ideal rings in [34].

Let M be a full R-lattice in $\mathcal{A}$. Then the left order of M is defined by

$$O_l(M) = \{a \in \mathcal{A} |\ aM \subseteq M\}.$$

The set $O_l(M)$ is known to be an R-order of $\mathcal{A}$, see [53, Chapter 8]. It actually follows from the fact that $O_l(M)$ is isomorphic to the intersection of two R-algebras: the image of $\mathcal{A}$ under the left regular representation and $\mathrm{Hom}_R(M, M)$ (embedded into $\mathrm{Hom}_K(\mathcal{A}, \mathcal{A})$).

The next two lemmas will be important tools for the algorithm which finds maximal orders. The first one reduces the question of enlarging an order over R to a similar task for R_π-orders where π is a prime element of R. Here $R_\pi \leq K$ denotes the localization of R at the prime ideal $R\pi$, that is, $R_\pi = \{\frac{\alpha}{\beta} : \alpha, \beta \in R \text{ with } \pi \not| \beta\}$. If Γ is an R-order in $\mathcal{A}$, then $\Gamma_\pi = R_\pi\Gamma$ is an R_π-order.

Lemma 88. *Let π be a prime element of R and Γ be an R-order in $\mathcal{A}$. Suppose that J is an ideal of Γ_π such that $J \geq \pi\Gamma_\pi$ and $O_l(J) > \Gamma_\pi$. Put $I = \Gamma \cap J$. Then we have $I \geq \pi\Gamma$ and $O_l(I) > \Gamma$.*

This lemma is stated for $R = \mathbb{Z}$ in [34, Lemma 2.7]. The proof goes through for any principal domain R. We include it for completeness.

Proof. Clearly $I \geq \pi\Gamma$ and I is an ideal of Γ. We also have $J = R_\pi I$. Let $a \in O_l(J)\backslash\Gamma_\pi$. Let $a_1, a_2, \ldots, a_s$ be a generating set of I as an R-module. Then these elements generate J as an R_π-module whence for $i = 1, \ldots, s$ we have

$$aa_i = \frac{\alpha_{i1}}{\beta_{i1}}a_1 + \cdots + \frac{\alpha_{it}}{\beta_{it}}a_s, \tag{3.2}$$

where $\alpha_{ij}, \beta_{ij} \in R$ and π does not divide β_{ij}. Now put $\beta = \prod_{i,j}\beta_{ij}$. Then $\beta a a_i$ is in I ($i = 1, \ldots, s$), whence $\beta a I \leq I$ and consequently $\beta a \in O_l(I)$. Finally we observe that βa is not in Γ since β is not divisible by π and therefore $\beta a \in \Gamma$ would imply $a \in \Gamma_\pi$. The proof is complete. □

The next simple statement is stated in [34, Proposition 2.8] for $R = \mathbb{Z}$. It enables us to use Λ in place of Λ_π in computations regarding sufficiently large one or two-sided ideals of Λ_π.

Proposition 89. *Let Λ be an R-order in $\mathcal{A}$ and π be a prime of R. Then the map $\Phi : x \mapsto x + \pi\Lambda_\pi (x \in \Lambda)$ induces an isomorphism of rings $\Lambda/\pi\Lambda \cong \Lambda_\pi/\pi\Lambda_\pi$.*

Proof. Clearly $\Phi : \Lambda \to \Lambda_\pi/\pi\Lambda_\pi$ is an epimorphism of rings. It is straightforward to check that its kernel is $\pi\Lambda$. □

Now we quote some further theorems and definitions from [34]. The next statement is [34, Proposition 3.1].

Proposition 90. *Let Λ_π be an R_π-order in $\mathcal{A}$. Then $\overline{\Lambda}_\pi = \Lambda_\pi/\pi\Lambda_\pi$ is an algebra with identity element over the residue class field $\overline{R}_\pi = R_\pi/\pi R_\pi$ (which is also isomorphic to $R/\pi R$) and $\dim_{\mathbb{K}}\mathcal{A} = \dim_{\overline{R}_\pi}\overline{\Lambda}_\pi$. If $\Phi : \Lambda_\pi \to \overline{\Lambda}_\pi$ is the canonical epimorphism, then $\pi\Lambda_\pi \subseteq Rad(\Lambda_\pi) = \Phi^{-1}Rad(\overline{\Lambda}_\pi)$ and Φ induces a ring isomorphism $\Lambda_\pi/Rad(\Lambda_\pi) \cong \overline{\Lambda}_\pi/Rad(\overline{\Lambda}_\pi)$.*

Now we introduce the important concept of extremal orders:

Definition 91. *Let Λ_π and Γ_π be R_π-orders in $\mathcal{A}$. We say that Γ_π* radically contains *Λ_π if and only if $\Gamma_\pi \supseteq \Lambda_\pi$ and $Rad(\Gamma_\pi) \supseteq Rad(\Lambda_\pi)$. This is a partial ordering on the set of R_π-orders. Orders maximal with respect to this partial ordering are called* extremal.

The next statement is [34, Proposition 4.1].

Proposition 92. *An R_π-order Λ_π of $\mathcal{A}$ is extremal if and only if $\Lambda_\pi = O_l(Rad(\Lambda_\pi))$.*

Finally, we quote [34, Proposition 4.5].

Proposition 93. *Let $\Lambda_\pi \subset \Gamma_\pi$ be R_π-orders in $\mathcal{A}$. Suppose that Λ_π is extremal and Γ_π is minimal among the R_π-orders properly containing Λ_π. Then there exists a two-sided ideal I of Λ_π minimal among those containing $Rad(\Lambda_\pi)$ such that $O_l(I) \supseteq \Gamma_\pi$*

3.2.2 The algorithm

We start with a high-level description of the algorithm over a general principal ideal domain R. Let R be a principal ideal domain, K its field of fractions. Suppose that an algebra $\mathcal{A}$, isomorphic to $M_n(K)$ is given by structure constants γ_{ij}^k $(i,j,k = 1,\ldots,n^2)$ from K with respect to a basis $a_1,\ldots,a_{n^2}$. We assume that these structure constants are represented as fractions of pairs of elements from R. Let δ be a common multiple (e.g., the product or the l. c. m.) of the denominators. Then $a_i' = \delta a_i$ $(i = 1,\ldots,n^2)$ will be a basis with structure constants $\delta\gamma_{ij}^k \in R$. Therefore the R-submodule Λ_0 of $\mathcal{A}$ with basis $a_1',\ldots,a_{n^2}'$ is an almost R-order.

We shall compute the discriminant $d = d(\Lambda_0)$. Let $S = \{\pi_1,\ldots,\pi_r\}$ be the set of the prime factors of d. Observe that the discriminant of any R-order conjugate to $M_n(R)$ is 1. This also holds for R_π-orders for any prime element π. Therefore, by Corollary 27 and by Proposition 86, $\Lambda_{0\pi}$ is a maximal R_π-order for any prime π not in S.

Starting with the order Λ obtained by taking the R-module generated by Λ_0 and the identity element, for each prime in S we test constructively whether Λ_π is a maximal R_π order using the two tests described below. By constructiveness we mean that in the "no" case we construct an R-order $\Gamma \supsetneq \Lambda$. If any of the tests finds such a Γ, then we proceed with Γ in place of Λ. Otherwise, if Λ_π passes the tests for every $\pi \in S$ then we conclude that Λ is already maximal. By Proposition 86 the number of such rounds is at most the number of the prime divisors of d, counted with multiplicities.

The first test is used to constructively decide whether Λ_π is an extremal R_π-order by checking if $O_l(Rad(\Lambda_\pi)) = \Lambda_\pi$ (Proposition 92). To this end, we compute the ideal $I = Rad(\Lambda_\pi) \cap \Lambda$. By Lemma 88, Λ passes the test if and only if $O_l(I) = \Lambda$. Otherwise $\Gamma = O_l(I)$ is an order strictly containing Λ. To compute I, we work with the n^2-dimensional $R/\pi R$-algebra $\mathcal{B} = \Lambda/\pi\Lambda$. From Propositions 89 and 90 we infer that I is the inverse image of $Rad(\mathcal{B})$ with respect to the canonical map $\Lambda \to \mathcal{B}$.

If Λ_π passes the first test, then we proceed with the test of Proposition 93: if there exists an ideal J of Λ_π minimal among the two-sided ideals properly containing $Rad(\Lambda_\pi)$ such that $O_l(J) > \Lambda_\pi$, then we construct an R-order Γ that properly contains Λ. Like for the first test, we can work in the $R/\pi R$-algebra $\mathcal{B} = \Lambda/\pi\Lambda$. Let $J_1, \ldots, J_m$ denote the minimal two-sided ideals of $\mathcal{B}$ which contain $Rad(\mathcal{B})$. We have $m \leq n^2$. Let I_i denote the inverse image of J_i with respect to the map $\Lambda \to \mathcal{B}$. As in the first case we obtain, that we have to compute the rings $O_l(I_i)$ for $i = 1, \ldots, m$. We can stop when $\Lambda < O_l(I_i)$ is detected, because then we have an order properly containing Λ.

3.2.3 The case $R = \mathbb{F}_q[t]$

We continue with details of the key ingredients of an efficient algorithm for $R = \mathbb{F}_q[t]$ following the lines above. These will give an f-algorithm whose running time is polynomial in the size of the input. The input is an array of n^6 structure constants represented as fractions of polynomials. We assume that the numerators are of degree at most d_N and the denominators are of degree at most d_D. Thus the size of the input is around $n^6(d_D + d_N)\log q$.

The l. c. m. of the denominators and hence a basis for the initial almost order Λ_0 can be computed in polynomial time. The degree of this common denominator is at most $n^6 d_D$, whence Λ_0 will have a basis $a'_1, \ldots, a'_{n^2}$, where each a'_j is a_j, multiplied by a polynomial of degree at most $n^6 d_D$. The structure constants for the basis $a'_1, \ldots, a'_{n^2}$ are polynomials of degree at most $n^6 d_D + d_N$. The discriminant $d = d(\Lambda_0)$ can be efficiently computed in a direct way following the definition. The entries of the matrices for the images of a'_j at the regular repre-

sentation, written in terms of the basis $a'_1, \ldots, a'_{n^2}$ are just structure constants for the basis $a'_1, \ldots, a'_{n^2}$. Therefore these entries are polynomials of degree bounded by $n^6 d_D + d_N$ and hence the entries of the Gram matrix of the bilinear trace form are polynomials of degree $2n^6 d_D + 2d_N$. To compute $d(\Lambda_0)$, let $n = p^r k$ where p is the characteristic of $\mathbb{F}_q$ and k is relatively prime to p. Then the characteristic polynomial of $a'_i a'_j$ (in the regular representation), is the nth power of the characteristic polynomial of $a'_i a'_j$ as an n by n matrix. Therefore it is of the form

$$\begin{aligned}(X^n - (tr\ a'_i a'_j) X^{n-1} + \cdots)^n &= (X^{nk} - k(tr\ a'_i a'_j) X^{nk-1} + \cdots)^{p^r} \\ &= X^{n^2} - (ktr\ a'_i a'_j)^{p^r} X^{n^2 - p^r} + \cdots .\end{aligned}$$

It follows that $d(\Lambda_0)$ is a polynomial D_0 of degree at most $2n^8 d_D + 2n^2 d_N$.

By Proposition 87, we have $\Lambda \leq \frac{1}{D_0}\Lambda_0$ for any $\mathbb{F}_q[t]$-order $\Lambda \geq \Lambda_0$. Therefore we can represent Λ as the $\mathbb{F}_q[t]$-submodule Λ/Λ_0 of the factor module $\frac{1}{D_0}\Lambda_0/\Lambda_0$. This factor module is an $n^2 \deg D_0$-dimensional vector space over the field $\mathbb{F}_q$. In fact, the elements $\frac{t^k}{D_0} a'_i + \Lambda_0$ ($i = 1, \ldots, n^2$, $k = 0, \ldots, \deg D_0 - 1$ form an $\mathbb{F}_q$-basis) and we represent Λ/Λ_0 by an $\mathbb{F}_q$-basis written in terms of this basis. Notice that the ideals I whose left order $O_l(I)$ we compute throughout the algorithm are all (left) Λ_0-submodules of $\frac{1}{D_0}\Lambda_0$ containing $D_0\Lambda_0$. Observe next that the multiplication of $\mathcal{A}$ induces an $\mathbb{F}_q$-bilinear map μ from $\frac{1}{D_0}\Lambda_0/\Lambda_0 \times I/D_0 I$ to $\frac{1}{D_0} I/I$. For $a \in \frac{1}{D_0}\Lambda$ and $b \in I$, one can set

$$\mu(a + \Lambda_0, b + D_0 I) = ab + I.$$

This is well defined as $(\frac{1}{D_0}\Lambda_0)(D_0 I) = \Lambda_0 I \subseteq I$. Taking an $\mathbb{F}_q$-basis $b_1, \ldots, b_s$ of $I/D_0 I$, the factor $O_l(I)/\Lambda_0$ can be computed as the intersection of the kernels of the linear maps $\mu(\cdot, b_i)$ ($i = 1, \ldots, s$). As the dimensions are bounded by polynomials in n and in the degree of D_0, for every I possibly occurring in the algorithm, $O_l(I)$ is computable in polynomial time. Given an intermediate order Λ, we can compute the candidate ideals I by computing the radical of $\mathcal{B} = \Lambda/g\Lambda$ for the irreducible factors g of D_0 and the minimal two-sided ideals of $\mathcal{B}$ containing the radical and finally by taking inverse images of these at the map $\Lambda \to \mathcal{B}$. As $\mathcal{B}$ is an $n^2 \deg g$-dimensional vector space over $\mathbb{F}_q$, its radical and the minimal two-sided ideals containing it can be computed in time polynomial in the input size using for example the deterministic method [57]. The minimal two-sided ideals containing the radical, that is, the simple components of $\mathcal{B}/Rad(\mathcal{B})$ can be found by the deterministic f-algorithm [22].

For $\alpha_{ik} \in \mathbb{F}_q$ ($i = 1, \ldots, n^2$, $k = 0, \ldots, \deg D_0 - 1$), the combination

$$\sum_{i=1}^{n^2} \sum_{k=0}^{\deg D_0 - 1} \alpha_{ik} \frac{t^k}{D_0} a'_i$$

of $a'_1, \ldots, a'_{n^2}$ has coefficients whose numerators and denominators are polynomials of degree at most $\deg D_0 \leq 2n^8 d_D + 2n^2 d_N$. Together with $a'_1, \ldots, a'_{n^2}$, such representatives for an $\mathbb{F}_q$-basis of Λ/Λ_0 give a system of generators over $\mathbb{F}_q[t]$ for Λ. When Λ turns out to be maximal, then we can use the lattice reduction algorithm by Paulus [49] to obtain a basis for Λ consisting of combinations of $a'_1, \ldots, a'_{n^2}$ with coefficients having numerators and denominators also of degree at most $\deg D_0 \leq 2n^8 d_D + 2n^2 d_N$. (Here we make use of the nature of the reduction algorithm: it never increases the maximum degree of the coordinates of the intermediate generators.) This gives us the following theorem:

Theorem 94. *Let $\mathcal{A}$ be isomorphic to $M_n(\mathbb{F}_q(t))$ given by structure constants having numerators and denominators of degree at most $d_C \geq 1$. A maximal $\mathbb{F}_q[t]$-order Λ can be constructed by an f-algorithm running in time $(n + d_C + \log q)^{O(1)}$. The output of the algorithm is an $\mathbb{F}_q[t]$-basis for Λ whose elements are linear combinations in the original basis of $\mathcal{A}$ with coefficients which are ratios of polynomials of degree at most $(2n^8 + n^6 + 2n^2)d_C$.*

Notice that $\sum_{j=0}^{d} \alpha_j t^j = t^d \sum_{j=0}^{d} \alpha_{d-j} \frac{1}{t^j}$. Therefore a fraction of two polynomials in t of degree at most d can also be written as a fraction of two polynomials in $\frac{1}{t}$ also of degree at most d. Therefore Theorem 94 gives the following.

Corollary 95. *Let $\mathcal{A}$ and d_C be as in Theorem 94. Then a maximal $\mathbb{F}_q[\frac{1}{t}]$-order Δ can be constructed by an f-algorithm running in time $(n + d_C + \log q)^{O(1)}$. The output of the algorithm is an $\mathbb{F}_q[\frac{1}{t}]$-basis for Δ whose elements are linear combinations in the original basis of $\mathcal{A}$ with coefficients which are ratios of polynomials (in t) of degree at most $(2n^8 + n^6 + 2n^2)d_C$.*

We remark that later on we will actually need an $\mathbb{F}_q[\frac{1}{t}]_{(\frac{1}{t})}$-basis for a maximal $\mathbb{F}_q[\frac{1}{t}]_{(\frac{1}{t})}$-order. Obviously, for this an $\mathbb{F}_q[\frac{1}{t}]$-basis for an $\mathbb{F}_q[\frac{1}{t}]$-order Δ whose localization at the the prime $\frac{1}{t}$ is maximal, will do. Therefore it will be actually sufficient to apply the main steps of the order increasing algorithm only for the prime $\frac{1}{t}$ of $\mathbb{F}_q[\frac{1}{t}]$.

3.3 Finding a rank 1 idempotent in $\mathcal{A}$

Let $R \subseteq \mathbb{F}_q(t)$ be the set of rational functions having degree at most 0 (note that the 0 polynomial has degree $-\infty$ hence it also belongs to R). Thus, if $f, g \in \mathbb{F}_q[t]$,

$g \neq 0$, then $\frac{f}{g} \in R$ iff $\deg f \leq \deg g$. It is easy to see that R is a subring of $\mathbb{F}_q(t)$. Actually R is the valuation ring for the exponential valuation $-\deg$ of $\mathbb{F}_q(t)$. An alternative view is that $R = \mathbb{F}_q[\frac{1}{t}]_{(\frac{1}{t})}$, the localization of the ring $\mathbb{F}_q[\frac{1}{t}]$ at the prime ideal $(\frac{1}{t})$. (In fact, one readily verifies that the elements of R are precisely the functions of the form $f(\frac{1}{t})/g(\frac{1}{t})$, where f, g are univariate polynomials over $\mathbb{F}_q$ and the constant term of g is not 0.) Thus R is a discrete valuation ring, and as such, a principal ideal ring.

The following theorem is the main structural result. It identifies a finite subalgebra C of modest size in $\mathcal{A}$, which contains a primitive idempotent of $\mathcal{A}$.

Theorem 96. *Let $\mathcal{A} \cong M_n(\mathbb{F}_q(t))$ and let Λ be a maximal $\mathbb{F}_q[t]$-order in $\mathcal{A}$. Also, let R be the subring of $\mathbb{F}_q(t)$ discussed above, that is, the set of rational functions of degree at most zero. Let Δ be a maximal R-order in $\mathcal{A}$. Let $b_1, \ldots, b_{n^2}$ be an $\mathbb{F}_q[t]$-basis of Λ, and for $j = 1, \ldots, n^2$ let d_j be the smallest integer such that $\frac{1}{t^{d_j}} b_j \in \Delta$. Let $d_{min} = \min\{d_j : 1 \leq j \leq n^2\}$, $d_{max} = \max\{d_j : 1 \leq j \leq n^2\}$. Then*

(i) For every element $a \in \Lambda \cap \Delta$ we have $a = \sum \alpha_i b_i$, where the α_i are polynomials in $\mathbb{F}_q[t]$ of degree at most $n^2 d_{max} - d_{min}$.

(ii) $\Lambda \cap \Delta$ contains a primitive idempotent of $\mathcal{A}$.

Proof. Let $\phi : \mathcal{A} \to M_n(\mathbb{F}_q(t))$ be an algebra isomorphism such that $\phi(\Delta) = M_n(R)$. (Such a ϕ exists by Corollary 27.) We show that the $\mathbb{F}_q[t]$-lattice $\phi(\Lambda)$ in $M_n(\mathbb{F}_q(t))$ (the latter considered as $\mathbb{F}_q(t)^{n^2}$) has determinant 1. To see this, let B be the matrix whose columns form an $\mathbb{F}_q[t]$-basis for the $\mathbb{F}_q[t]$ lattice $\phi(\Lambda)v \subset \mathbb{F}_q(t)^n$ where v is a nonzero vector from $\mathbb{F}_q(t)^n$. Then $\phi(\Lambda) = BM_n(\mathbb{F}_q[t])B^{-1}$. The claim on the determinant follows from that the standard lattice $\mathbb{F}_q[t]^{n^2}$ has determinant one and from that the conjugation $X \mapsto BXB^{-1}$, considered as an $\mathbb{F}_q(t)$-linear transformation on $\mathbb{F}_q(t)^{n^2}$, has determinant one. For the latter, notice that multiplication by B^{-1} from the right is similar to a block diagonal matrix consisting of n copies of B^{-1}, and hence has determinant $(\det B^{-1})^n$, while multiplication by B from the left has determinant $(\det B)^n$.

Let $C = \Lambda \cap \Delta$. As $\Delta = \phi^{-1}(M_n(R))$, C can be characterized as the set of the elements a of Λ such that $\phi(a)$ has no entries of positive degree. As both Δ and Λ are $\mathbb{F}_q$-algebras, so is C.

Notice that for $0 \neq a \in \mathcal{A}$ the degree of $\phi(a) \in M_n(\mathbb{F}_q(t))$ (the maximum of the degrees of the entries of the matrix $\phi(a)$) is just the minimal (possibly negative) integer r such that $\frac{1}{t^r}\phi(a) \in M_n(R)$, or, equivalently, $t^{-r}a \in \Delta$. It follows that the degrees of the entries of $\phi(b_j)$ are bounded by d_{max} and hence the orthogonality defect of the basis $\phi(b_1), \ldots, \phi(b_{n^2})$ for $\phi(\Lambda)$ is at most $n^2 d_{max}$,

because $|\det \phi(\Lambda)| = 0$. Therefore, for $a = \sum_{j=1}^{n^2} \alpha_j b_j \in C$ Lemma 46 gives that α_j has degree at most $n^2 d_{max} - d_{min}$, showing statement (i).

To establish statement (ii), consider an invertible matrix $B \in M_n(\mathbb{F}_q(t))$ for which $\phi(\Lambda) = B^{-1} M_n(\mathbb{F}_q[t]) B$. Let us consider the lattice $L_1 = B^{-1}\mathbb{F}_q[t]^n$ in $\mathbb{F}_q(t)^n$. The determinant of L_1 is obviously $\det B^{-1}$. Let us denote by δ be the degree of $\det B$. Let $B^{-1}u_1, \ldots, B^{-1}u_n$, with $u_i \in \mathbb{F}_q[t]^n$, be an $\mathbb{F}_q[t]$-basis of orthogonality defect zero for L_1. One can obtain such a basis by lattice basis reduction. Similarly, let $L_2 = B^T\mathbb{F}_q[t]$. Then L_2 is an $\mathbb{F}_q[t]$-lattice having determinant $\det B$. Let $B^T u'_1, \ldots, B^T u'_n$, with $u'_i \in \mathbb{F}_q[t]^n$, be a basis of defect zero for L_2. Now we define a graph. We connect u_i with u'_j with an edge if ${u'_j}^T u_i \neq 0$. This defines a bipartite graph having these $2n$ vectors as vertices satisfying Hall's criterion for having a perfect matching. (A set of s vectors from $u_1, \ldots, u_n$ having less than s neighbors would span a subspace of dimension s having an orthocomplement having dimension larger than $n - s$.) By changing the order of u'_js we arrange that ${u'_i}^T u_i \neq 0$ $(i = 1, \ldots, n)$. We have

$$\sum_{j=1}^{n} (|B^{-1}u_j| + |B^T u'_j|) = \sum_{j=1}^{n} |B^{-1}u_j| + \sum_{j=1}^{n} |B^T u'_j| = -\delta + \delta = 0,$$

whence there exists at least one index i, such that the maximum degree of the coordinates of $B^{-1}u_i$ and the maximum degree of the coordinates of $B^T u'_i$ add up to at most zero. Let i be such an index and let S resp. S' be the matrix whose first column is u_i resp. u'_i, and whose remaining entries are zero. Now $Z = B^{-1}S{S'}^T B$ is a matrix whose entries are of degree at most zero. Also, $Z \in \phi(\Lambda)$. Therefore $\phi^{-1}(Z)$ is in C. Furthermore, Z has rank one as it is similar to $S{S'}^T = u_i {u'_i}^T$. Also, as $(u_i {u'_i}^T)^2 = \mu u_i {u'_i}^T$ where $\mu = {u'_i}^T u_i \neq 0$. It follows that the minimal polynomial of Z over $\mathbb{F}_q(t)$ as well as that of $\phi^{-1}(Z)$ is $X^2 - \mu X$ with a nonzero $\mu \in \mathbb{F}_q(t)$. As $\phi^{-1}(Z) \in \Lambda \cap \Delta$, we have $\mu \in \mathbb{F}_q[t] \cap R = \mathbb{F}_q$. Now $e = \frac{1}{\mu}\phi^{-1}(Z)$ is an idempotent in C such that $\phi(e)$ has rank one. □

Remark 97. We give an example of a C which is not isomorphic to a full matrix algebra over $\mathbb{F}_q(t)$. Let $\Lambda = B^{-1}M_2(\mathbb{F}_q[t])B$ where B is the following matrix:

$$\begin{pmatrix} \frac{1}{t} & 0 \\ 0 & t \end{pmatrix}.$$

Let $\Gamma = M_2(R)$, i.e. those matrices whose degree is at most 0. Then $C = \Gamma \cap \Lambda$ is generated as an $\mathbb{F}_q$ vector space by the following matrices:

$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ \frac{1}{t} & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ \frac{1}{t^2} & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

Note that C has dimension 5 over $\mathbb{F}_q$, hence it cannot be isomorphic to $M_2(\mathbb{F}_q)$. As a matter of fact, it is not even semisimple. The radical of C consists of those matrices whose diagonal entries are 0. Finally, note that $C/Rad\ C \cong \mathbb{F}_q \oplus \mathbb{F}_q$.

For finding a primitive idempotent of $\mathcal{A}$ inside C we can use the method described in the proof of the following lemma.

Lemma 98. *Let C be the finite $\mathbb{F}_q$-algebra from Theorem 96, and let $e_1, \ldots, e_r$ be a complete system of orthogonal primitive idempotents in C. Then there exists an i such that e_i is a rank 1 idempotent in $\mathcal{A}$.*

Having a basis of C at hand (a subset of $\mathcal{A}$), one can find such an idempotent by a polynomial time f-algorithm.

Proof. We note first, that the identity element of $\mathcal{A}$ is in C, hence C has idempotents. Let $x \in C$ be an element which is a rank 1 idempotent in $\mathcal{A}$. By Theorem 96 such an x exists. Next observe that there exists an index i, for which $e_i x$ is not in the radical of C. For, otherwise $\sum_{i=1}^{r} e_i x = x$ would be in the radical of C, which is impossible, as x is not nilpotent. Let us denote this primitive idempotent e_i by e. Since ex is not in the radical of C, the right ideal exC it generates in C contains a nonzero idempotent f. Indeed, we can consider this right ideal as an $\mathbb{F}_q$-algebra which is not nilpotent. Hence if we factor out its radical, then we have a nonzero idempotent there ([17, Corollary 2.2.5]), which can be lifted to an idempotent in exC ([17, Corollary 3.1.2]). Write $f = exy$ with a suitable $y \in C$. We have $ef = e(exy) = e^2xy = exy = f$. We verify now that both fe and $e - fe$ are idempotent elements:

$$(fe)^2 = fefe = f(ef)e = ffe = fe$$

and

$$(e - fe)^2 = e^2 + (fe)^2 - efe - fee = e + fe - fe - fe = e - fe.$$

Furthermore, they are orthogonal:

$$fe(e - fe) = (fee) - (fe)^2 = fe - (fe)^2 = 0$$

and

$$(e - fe)fe = (ef)e - (fe)^2 = fe - (fe)^2 = 0.$$

Since e is a primitive idempotent, one has either $fe = 0$ or $fe = e$. We show that the first case cannot happen. If $fe = 0$ then $fef = 0$. However, $fef = f^2 = f$ which is not zero. This implies that $fe = e$, and $e = exye$. Since x had rank 1 in $\mathcal{A}$, e also has rank 1 in $\mathcal{A}$.

As for the computational part of the statement, first one has to compute a Wedderburn-Malcev complement in C: a subalgebra B of C which is isomorphic to $C/Rad(C)$. This can be done in deterministic polynomial time using the algorithm of [30, Theorem 3.1]. Then we can use for example the polynomial time f-algorithms of [22] and [57] to compute a complete system of primitive idempotents in B. To calculate ranks, we can use the fact that for $a \in \mathcal{A}$ the left ideal $a\mathcal{A}$ has dimension rn over $\mathbb{F}_q(t)$ where r is the rank of a (considered as an n by n matrix). □

We prove a bound on d_{min} and d_{max} in the case when Λ and Δ are the maximal orders constructed in Theorem 94 and Corollary 95, respectively. Λ is an $\mathbb{F}_q[t]$-order and Δ is viewed as an R-order here.

Lemma 99. *For the pair of maximal orders as above, we have* $d_{max} \leq (2n^8 + 2n^6 + 2n^2)d_C$ *and* $d_{min} \geq -2(2n^8 + n^6 + 2n^2)d_C$

Proof. For short, we write $L = (2n^8 + n^6 + 2n^2)d_C$. Let $a_1, \ldots, a_{n^2}$ be the input basis of $\mathcal{A}$ we use in the algorithms of Theorem 94 and Corollary 95. We know that the numerators and denominators of the structure constants for $\mathcal{A}$ are polynomials of degree at most d_C. Let $g^*(1/t)$ be the smallest common denominator of the structure constants when written as rational functions in $\frac{1}{t}$. The degree of g^* is at most $n^6 d_C$. We know that the $g^*(1/t)a_i$ are in the starting almost $\mathbb{F}_q[\frac{1}{t}]$-order Δ_0, hence they are also in Δ. Also, one can then write

$$g^*\left(\frac{1}{t}\right) = \frac{1}{t^\ell} h\left(\frac{1}{t}\right)$$

where $h(y) \in \mathbb{F}_q[y]$ and $h(0) \neq 0$. We have here $\ell \leq n^6 d_C$. We claim that $\frac{1}{t^{n^6 d_C}} a_i \in \Delta$ hold for every i. Indeed

$$\frac{1}{t^{n^6 d_C}} a_i = \frac{\frac{1}{t^{n^6 d_C - \ell}}}{h(\frac{1}{t})} \cdot g^*\left(\frac{1}{t}\right) a_i.$$

Here the first factor is in R, the second is in Δ, thus giving the claim.

We know from Theorem 94 that every basis element b_j of Λ is a linear combination of the a_i with coefficients $\alpha_i \in \mathbb{F}_q(t)$, and the numerator as well as the denominator of α_i has degree at most L. We claim now that $\frac{1}{t^{n^6 d_C + L}} b_j \in \Delta$. Indeed, we have

$$\frac{1}{t^{n^6 d_C + L}} \alpha_i a_i = \left(\frac{1}{t^{n^6 d_C}} a_i\right) \cdot \left(\frac{1}{t^L} \alpha_i\right).$$

The first factor is in Δ, the second is in R and the upper bound follows.

As for d_{min}, we observe that the coefficients for the elements of Λ in the basis $\{a_i\}$ are rational functions of degree at least $-L$ (Theorem 94). Similarly, by Corollary 95 the coefficients for the elements of Δ in the basis $\{a_i\}$ are rational functions of degree at most L. It follows that for $d < -2L$ the element $\frac{1}{t^d}b_j$ can not be in Δ, as the coefficient $\frac{1}{t^d}\alpha_i$ has degree at least $L+1$. □

Now we turn to the algorithmic task of finding (an $\mathbb{F}_q$-basis of) C.

Lemma 100. *Let $b_1, \ldots, b_{n^2}$ be the $\mathbb{F}_q[t]$-basis of Λ constructed by the algorithm of Theorem 94, and let $u_1, \ldots, u_{n^2}$ be the R-basis of Δ constructed by the method of Corollary 95. From these data we can construct an F_q-basis of C in deterministic polynomial time.*

Proof. We consider the elements of $\mathcal{A}$ as vectors in the basis $u_1, \ldots, u_{n^2}$. This way the elements of $\mathcal{A}$ can be viewed as vectors from $\mathbb{F}_q(t)^{n^2}$ in the usual way: an element $a \in \mathcal{A}$ with $a = \sum_{j=1}^{n^2} \alpha_j u_j$ is represented by the vector

$$(\alpha_1, \ldots, \alpha_{n^2})^T \in \mathbb{F}_q(t)^{n^2}.$$

Observe, that a vector as above represents an element of Δ iff $|\alpha_i| \leq 0$ holds for every i. Consider now the vectors $b_i' \in \mathbb{F}_q(t)^{n^2}$ representing the basis elements b_i of Λ. They generate a full $\mathbb{F}_q[t]$-lattice (corresponding to Λ) in $\mathbb{F}_q(t)^{n^2}$. We next compute a reduced basis $c_1, \ldots, c_{n^2}$ of this lattice. An element

$$a = \sum_{i=1}^{n^2} \beta_i c_i \quad \text{with } \beta_i \in \mathbb{F}_q[t] \text{ for } i = 1, \ldots, n^2$$

represents an element of $C = \Lambda \cap \Delta$ iff $|a| \leq 0$. We claim that this latter condition is equivalent to the set of inequalities

$$|\beta_i c_i| = |\beta_i| + |c_i| \leq 0, \quad i = 1, \ldots, n^2.$$

Indeed, as the $\{c_i\}$ is a reduced $\mathbb{F}_q[t]$-basis, from Lemma 46 we obtain that

$$|\beta_i| \leq |a| + OD(c_1, \ldots, c_{n^2}) - |c_i| = |a| - |c_i| \tag{3.3}$$

for every i, hence if $|a| \leq 0$ then $|\beta_i c_i| \leq 0$ for every i. Conversely, $|\beta_i c_i| \leq 0$ for every i obviously implies that $|a| \leq 0$. We conclude that the elements $t^j c_i$ such that $1 \leq i \leq n^2$ and j is a natural number with $j + |c_i| \leq 0$ form an $\mathbb{F}_q$-basis of C. Theorem 96 and Lemma 99 provide a polynomial upper bound for the dimension of C over $\mathbb{F}_q$, and hence on the number of such elements $t^j c_i$.[1]

[1] A polynomial bound for the dimension of C follows also simply from the polynomiality of the algorithm described here.

The algorithmic subtasks involved here: change of basis from the input basis to the basis $\{u_i\}$, and the lattice basis reduction both can be done in deterministic polynomial time, hence from Λ and Δ we obtain C in polynomial time. □

The main steps of our algorithm for finding a rank 1 idempotent element $e \in \mathcal{A}$ are as follows.

---

1. Construct a maximal $\mathbb{F}_q[t]$-order Λ and a maximal R-order Δ, by the f-polynomial time algorithms of Theorem 94, and Corollary 95, respectively.
2. Compute an $\mathbb{F}_q$-basis of the finite algebra $C = \Lambda \cap \Delta$ using the polynomial time algorithm of Lemma 100.
3. With the polynomial time f-algorithm of Lemma 98 find a complete system $e_1, \ldots, e_r$ of orthogonal primitive idempotents in C, and then select an e_i among them which has rank 1 in $\mathcal{A}$. Finally output this element $e = e_i$.

Proof of Theorem 83. The correctness and the timing for the first Step follows immediately from Theorem 94, and Corollary 95. These, and Lemma 99 imply that C admits polynomial size description. Then Lemma 100 settles Step 2. Correctness and polynomiality for the last step is provided by Lemma 98. □

3.4 Lattices in $\mathbb{F}_q((\frac{1}{t}))^m$

In this section we propose an algorithm for finding a reduced basis of a lattice in $\mathbb{F}_q((\frac{1}{t}))^m$ and give a simple application of this result.

The field $\mathbb{F}_q((\frac{1}{t}))$ is an analogue of $\mathbb{R}$ so as one sometimes considers $\mathbb{Z}$-lattices in $\mathbb{R}^m$ it may be worthwile to look at lattices in $\mathbb{F}_q((\frac{1}{t}))^m$. As we will see later it is quite straightforward to generalize Lenstra's reduction algorithm to this setting.

First we restate some of the definitions and propositions. Note that the valuation $|.|$ we defined naturally generalizes to vectors in $\mathbb{F}_q((\frac{1}{t}))^m$. Indeed we define the degree of an $h = \sum_{i=n}^{-\infty} a_i t^i$ (where $a_n \neq 0$) to be n. The degree of a vector is just the largest degree among its components. Therefore the notion of orthogonality defect also makes sense in this setting.

Definition 101. *A basis* $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m \in \mathbb{F}_q((\frac{1}{t}))^m$ *is called* reduced *if the orthogonality defect* $OD(\mathbf{b}_1, \ldots, \mathbf{b}_m) = 0$.

Lemma 102. *Let $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m \in \mathbb{F}_q((\frac{1}{t}))^m$ be linearly independent and assume that $\mathbf{x} = \sum_{i=1}^m r_i \mathbf{b}_i$ where $r_i \in \mathbb{F}_q[t]$. Then the following holds for every i:*

$$|r_i| \leq |\mathbf{x}| + OD(\mathbf{b}_1, \ldots, \mathbf{b}_m) - |\mathbf{b}_i| \tag{3.4}$$

The proof of this lemma is essentially the same as the proof of Lemma 46.

Now we show how to apply Lenstra's algorithm (the algorithm described in Section 3.1, or [44, Algorithm 1.7]) to find a reduced basis of a full lattice in $\mathbb{F}_q((\frac{1}{t}))^m$.

Assume that we have a basis $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m$ in $\mathbb{F}_q((\frac{1}{t}))^m$. Let L be the $\mathbb{F}_q[t]$-lattice they generate. We propose an algorithm which transforms this basis into a reduced one.

Observe that we have a lower bound on the valuation of the shortest nonzero vector in L. Indeed, if one applies Lemma 102 (considering that $|r_i| \geq 0$), one gets that $c = \min\{|\mathbf{b}_i|\} - OD(\mathbf{b}_1, \ldots, \mathbf{b}_m)$ will be such a bound.

Let L' be an $\mathbb{F}_q[t]$-lattice in $\mathbb{F}_q[t]^m$ given by a basis $\mathbf{c}_1, \ldots, \mathbf{c}_m$. Lenstra's algorithm transforms this basis into a reduced basis $\mathbf{c}'_1, \ldots, \mathbf{c}'_m$. Now let us observe certain things about this algorithm. An inspection of the steps of the algorithm reveals that during the computation none of the $|\mathbf{c}_i|$-s increase. Put $M = \max\{|\mathbf{c}_i|\} \geq max\{|\mathbf{c}'_i|\}$. Then there are polynomials $r_{ij} \in \mathbb{F}_q[t]$ such that:

$$c'_i = \sum_{j=1}^{m} r_{ij} \mathbf{c}_j \text{ for } i = 1, \ldots, m \tag{3.5}$$

Due to Lemma 102 one has that $|r_{ij}| \leq M + OD(\mathbf{c}_1, \ldots, \mathbf{c}_m) - \min\{|\mathbf{c}_i|\}$.

Now consider our original lattice $L \subset \mathbb{F}_q((\frac{1}{t}))^m$. From the original input basis $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m$ we create a new basis $\mathbf{c}_1, \ldots, \mathbf{c}_m \in \mathbb{F}_q[t]^m$ in the following way. From each coordinate we omit each term of degree smaller than

$$\begin{aligned} K = \min(&m \min\{|\mathbf{b}_i|\} - (m-1) \max\{|\mathbf{b}_i|\} - OD(\mathbf{b}_1, \ldots, \mathbf{b}_m) - 1, \\ &c - \max\{|\mathbf{b}_i|\} - OD(\mathbf{b}_1, \ldots, \mathbf{b}_m) + \min\{|\mathbf{b}_i|\} - 1) \end{aligned}$$

Then we multiply each basis element with t^{-K}. Let L' be the $\mathbb{F}_q[t]$-lattice they generate. Observe that this new basis $\mathbf{c}_1, \ldots, \mathbf{c}_m$ consists of vectors from $\mathbb{F}_q[t]^m$. Now let us apply Lenstra's algorithm with this basis. Let $\mathbf{c}'_i$ and r_{ij} be the same as in the previous discussion. We prove the following proposition:

Proposition 103. *Let $\mathbf{b}'_i = \sum_{j=1}^m r_{ij} \mathbf{b}_j$ for $i = 1, \ldots, m$. Then $\mathbf{b}'_1, \ldots, \mathbf{b}'_m$ is a reduced basis.*

Proof. Let $b_{ij} = d_{ij} + \epsilon_{ij}$ where ϵ_{ij} is the sum of the terms of b_{ij} with degree smaller than K. Let $\mathbf{d}_i = (d_{i1}, \ldots, d_{im})$. Let $\mathbf{d}'_i = \sum_{j=1}^m r_{ij} \mathbf{d}_j$, for $i = 1, \ldots, m$.

Note that $\mathbf{c}_i = t^{-K}\mathbf{d}_i$ and so $\mathbf{c}'_i = t^{-K}\mathbf{d}'_i$. Hence $\mathbf{d}'_1, \ldots, \mathbf{d}'_m$ is also a basis of orthogonality defect zero. Let L'' be the lattice generated by the $\mathbf{d}_i$.

We show that $OD(\mathbf{d}_1, \ldots, \mathbf{d}_m) = OD(\mathbf{b}_1, \ldots, \mathbf{b}_m)$. For this to be true one only has to prove that the determinant of the lattice L and L'' have the same valuation, since $|\mathbf{d}_i| = |\mathbf{b}_i|$. Observe that the following holds:

$$K < |\det(L)| - (m-1)\max\{|\mathbf{b}_i|\}.$$

Indeed, K was chosen in a way such that

$$K < m\min\{|\mathbf{b}_i|\} - (m-1)\max\{|\mathbf{b}_i|\} - OD(\mathbf{b}_1, \ldots, \mathbf{b}_m)$$

and

$$OD(\mathbf{b}_1, \ldots, \mathbf{b}_m) = \sum_{i=1}^{m} |b_i| - \det(L) \geq m\min\{|\mathbf{b}_i|\} - \det(L).$$

By rearranging one gets the desired inequality. If we omit the terms of degree smaller than $|\det(L)| - (m-1)\max\{|\mathbf{b}_i|\}$ than the valuation of the determinant of the lattice does not change since even if we multiply such a small term with $m-1$ entries of maximal degree its valuation will be still smaller than the valuation of the determinant hence will not change its valuation. This proves that $|\det(L)| = |\det(L')|$.

Clearly $OD(\mathbf{c}_1, \ldots, \mathbf{c}_m) = OD(\mathbf{d}_1, \ldots, \mathbf{d}_m) = OD(\mathbf{b}_1, \ldots, \mathbf{b}_m)$. Due to the discussion at the beginning of the proof one has that

$$|r_{ij}| \leq \max\{|\mathbf{b}_i|\} + OD(\mathbf{b}_1, \ldots, \mathbf{b}_m) - \min\{|\mathbf{b}_i|\}.$$

Indeed, $\max\{|\mathbf{b}_i|\} - \min\{|\mathbf{b}_i|\} = \max\{|\mathbf{c}_i|\} - \min\{|\mathbf{c}_i|\}$ (omitting terms does not change the valuations and multiplying by t^K does not change the difference between maximum and minimum). Therefore

$$|r_{ij}\epsilon_{kl}| = |r_{ij}| + |\epsilon_{kl}| \leq \max\{|\mathbf{b}_i|\} + OD(\mathbf{b}_1, \ldots, \mathbf{b}_m) - \min\{|b_i|\} + K < c. \quad (3.6)$$

Let $\mathbf{e}_i = (\epsilon_{i1}, \ldots, \epsilon_{im})$. We have that $\mathbf{b}_i = \mathbf{d}_i + \mathbf{e}_i$. This implies that

$$\mathbf{b}'_i = \sum_{j=1}^{m} r_{ij}\mathbf{d}_j + \sum_{j=1}^{m} r_{ij}\mathbf{e}_j = \mathbf{d}'_i + \sum_{j=1}^{m} r_{ij}\mathbf{e}_j.$$

Equation 3.6 implies that $|\sum_{j=1}^{m} r_{ij}\mathbf{e}_j| < c$. Note that every vector in L'' has valuation at least c since

$$\min\{|\mathbf{d}_i|\} - OD(\mathbf{d}_1, \ldots, \mathbf{d}_m) = \min\{|\mathbf{b}_i|\} - OD(\mathbf{b}_1, \ldots, \mathbf{b}_m) = c.$$

Thus $|\mathbf{b}'_i| = |\mathbf{d}'_i|$ which implies that $OD(\mathbf{b}'_1, \ldots, \mathbf{b}'_m) = OD(\mathbf{d}'_1, \ldots, \mathbf{d}'_m) = 0$ as we have already shown that $\det(L) = \det(L'')$. This proves our claim. □

Now we have the following theorem:

Theorem 104. *Let $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m$ be a basis in $\mathbb{F}_q((\frac{1}{t}))^m$ and let L be the $\mathbb{F}_q[t]$-lattice they generate. Let $B = \max_{1 \leq i \leq m}(|b_i|)$. There exists an algorithm which takes $O(m^3 B(OD(\mathbf{b}_1, \ldots, \mathbf{b}_m) + 1))$ arithmetic operations in $\mathbb{F}_q$ and returns a reduced basis of L.*

Proof. Correctness follows from the previous discussion and Proposition 103. The estimate on the running time follows from the discussion on the running time of Lenstra's original algorithm (see Section 3.1). □

Let now $\mathbf{b}_1, \ldots, \mathbf{b}_m$ be a reduced basis of a lattice L ordered in a way that $|\mathbf{b}_1| \leq |\mathbf{b}_2|, \cdots \leq |\mathbf{b}_m|$. Then observe that b_1 is a shortest vector in L. Indeed, let $\mathbf{x} \in \mathbb{F}_q((\frac{1}{t}))^m$ be a nonzero vector and let $\mathbf{x} = \sum_{i=1}^m \alpha_i \mathbf{b}_i$. Assume that $\alpha_i \neq 0$ for some i (such an i exists since $\mathbf{x}$ is nonzero). Then by Lemma 102 we have that:

$$|\mathbf{x}| \geq |\alpha_i| + |\mathbf{b}_i| \geq |\mathbf{b}_1|$$

Note that $\alpha_i \in \mathbb{F}_q[t]$, hence its valuation is nonnegative.

Therefore Lenstra's algorithm may be interpreted as finding a lattice point in a cube centered around the origin. We would like to apply it to slightly more general object:

Definition 105. *Let $\mathbf{g}_1, \ldots, \mathbf{g}_m$ be linearly independent vectors in $\mathbb{F}_q((\frac{1}{t}))^m$. Then the $\mathbb{F}_q[[\frac{1}{t}]]$-module generated by $\mathbf{g}_1, \ldots, \mathbf{g}_m$ in $\mathbb{F}_q((\frac{1}{t}))^m$ is called the* parallelepiped generated by $\mathbf{g}_1, \ldots, \mathbf{g}_m$.

Assume that we have an $\mathbb{F}_q[t]$-lattice L in $\mathbb{F}_q((\frac{1}{t}))^m$. Also assume that a parallelepiped P is given by linearly independent generators $\mathbf{g}_1, \ldots, \mathbf{g}_m$. We are interested in the following problem: compute a nonzero element from the intersection $L \cap P$ or conclude that it is trivial (i.e., it only contains the zero vector).

This can be solved in the following way. Write the $\mathbf{b}_i$ as a linear combination of the $\mathbf{g}_i$:

$$\mathbf{b}_i = \sum_{j=1}^{m} \alpha_{ij} g_j$$

Let $\mathbf{c}_i = (\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{im})$. Consider the lattice L' generated by $\mathbf{c}_1, \ldots, \mathbf{c}_m$ in $\mathbb{F}_q((\frac{1}{t}))^m$. Find a shortest vector in L' using the algorithm from Theorem 104. Let this vector be $\mathbf{x}$. If $|x| \leq 0$ then $x \in P$ since this means that it can be expressed as a linear combination of the $\mathbf{g}_i$ where every coefficient has negative or zero valuation. If $|x| > 0$ then the intersection $L \cap P$ is trivial since if $y \in L \cap P$ than every coefficient of y in the basis $\mathbf{g}_1, \ldots, \mathbf{g}_m$ has negative or zero valuation, hence its valuation in L' cannot be positive.

CHAPTER 4

EXPLICIT EQUIVALENCE OF QUADRATIC FORMS OVER $\mathbb{F}_q(t)$

In this chapter we consider algorithmic questions concerning quadratic forms over $\mathbb{F}_q(t)$ where q denotes an odd prime power. The main focus is on the problem of finding a nontrivial zero of a quadratic form. The complexity of the problem of finding nontrivial zeros of quadratic forms in three variables has already been considered in ([14], [38]). However the same problem concerning quadratic forms of higher dimensions remained open.

Similarly, in the the case of quadratic forms over $\mathbb{Q}$ the algorithmic problem of finding nontrivial zeros of 3-dimensional forms was considered in several papers ([15], [37]) and afterwards Simon and Castel proposed an algorithm for finding nontrivial zeros of quadratic forms of higher dimensions ([63], [8]). The algorithms for the low-dimensional cases (dimension 3 and 4) run in polynomial time if one is allowed to call oracles for integer factorization. Surprisingly, the case where the quadratic form is of dimension at least 5, Castel's algorithm runs in polynomial time without the use of oracles (this is however, dependent on the Generalized Riemann Hypothesis). Note that (by the classical Hasse-Minkowski theorem) a 5-dimensional quadratic form over $\mathbb{Q}$ is always isotropic if it is indefinite.

Here we consider the question of isotropy of quadratic forms in 4 or more variables over $\mathbb{F}_q(t)$. The main idea of the algorithm is to split the form into two forms and find a common value they both represent. Here we apply two important facts. There is an effective bound on the number of irreducible polynomials in an arithmetic progression of a given degree. An asymptotic formula (which is effective for large q) was proven by Kornblum [41], but for our purposes, we apply a version with a much better error term, due to Rhin [54, Chapter 2, Sec-

tion 6, Theorem 4]. However, that statement is slightly more general, hence we cite a specialized version from [68]. A short survey on the history of this result can be found in [19, Section 5.3.]. The other fact we use is the local-global principle for quadratic forms over $\mathbb{F}_q(t)$ due to Rauter [52].

Finally we solve these two equations separately using the algorithm from [14] (and our Algorithm 1 in the 5-variable case). In the 4-dimensional case we are also able to detect if a quadratic form is anisotropic (a 5-dimensional form over $\mathbb{F}_q(t)$ is always isotropic). The algorithms are randomized and run in polynomial time. We also give several applications of these algorithms. Most importantly, we propose an algorithm which computes a transition matrix of two equivalent quadratic forms.

The chapter is structured as follows. The first section contains the necessary theoretical results concerning quadratic forms over $\mathbb{F}_q(t)$. For a brief introduction to quadratic forms over fields (whose characteristic is different from 2) we refer to Section 1.5 of this thesis. The second section contains the crucial ingredients of our algorithms. In the third section we describe the steps of our main algorithms together with their complexity analyses over $\mathbb{F}_q(t)$. In the final section we propose an algorithm for computing isometries between quadratic spaces using our main algorithms.

4.1 Quadratic forms over $\mathbb{F}_q(t)$

In this section we recall some basic facts about quadratic forms over $\mathbb{F}_q(t)$ (and over its completions) where q is an odd prime power. The main focus is on the question of isotropy of such forms. We start with two easy but useful facts concerning quadratic forms over finite fields. The first one was already established earlier in Section 2.1.

Fact 106. *(a) Let $a_1x_1^2 + a_2x_2^2$ be a non-degenerate quadratic form over a field $\mathbb{F}$. Then it is isotropic if and only if $-a_1a_2$ is a square in $\mathbb{F}$.*

(b) Every non-degenerate quadratic form over $\mathbb{F}_q$ with at least three variables is isotropic.

Remark 107. If $\mathbb{F} = \mathbb{F}_q$ (q is an odd prime power) then one can check easily if an element $s \neq 0$ in $\mathbb{F}$ is a square or not. Indeed, compute $s^{\frac{q-1}{2}}$ and check whether it is 1 or -1. Hence due to Fact 106 there is a deterministic polynomial time algorithm for checking whether $a_1x_1^2 + a_2x_2^2 = 0$ is solvable over $\mathbb{F}_q$ or not.

Now we turn our attention to quadratic forms over $\mathbb{F}_q(t)$ and their completions. The first lemma deals with quadratic forms in three variables:

Lemma 108. *Let $a_1, a_2, a_3 \in \mathbb{F}_q[t]$ be nonzero polynomials. Let f be a monic irreducible polynomial. Let $\mathbb{F}_q(t)_{(f)}$ denote the f-adic completion of $\mathbb{F}_q(t)$. Let $v_f(a_i)$ denote the multiplicity of f in the prime decomposition of a_i. Then the following hold:*

1. *If $v_f(a_1) \equiv v_f(a_2) \equiv v_f(a_3) \pmod 2$ then the equation $a_1x_1^2 + a_2x_2^2 + a_3x_3^2 = 0$ is solvable in $\mathbb{F}_q(t)_{(f)}$.*

2. *Assume that not all the $v_f(a_i)$ have the same parity. Also suppose that $v_f(a_i) \equiv v_f(a_j) \pmod 2$. Then the equation $a_1x_1^2 + a_2x_2^2 + a_3x_3^2 = 0$ is solvable in $\mathbb{F}_q(t)_{(f)}$ if and only if $-f^{-v_f(a_ia_j)}a_ia_j$ is a square modulo f.*

Proof. First assume that all $v_f(a_i)$ have the same parity. By a change of variables (replacing a_i by a_i/f^{k_i} for suitable k_i) we may assume that either $v_f(a_i) = 0$ for all i or $v_f(a_i) = 1$. In the second case we can divide the equation by f so we may assume that none of the a_i are divisible by f. We obtain an equivalent form whose coefficients are units in $\mathbb{F}_q(t)_{(f)}$. An equation $a_1x_1^2 + a_2x_2^2 + a_3x_3^2 = 0$ where all a_i are units in $\mathbb{F}_q(t)_{(f)}$ is solvable by [45, Chapter VI, Corollary 2.5.].

Now we turn to the second claim. By a change of variables we may assume that all the a_i are square-free. This results in two cases. Either f divides exactly one of the a_i or f divides exactly two of the a_i. First we consider the case where f divides exactly one, say a_1 (hence now $v_f(a_2) = v_f(a_3) = 0$ and $v_f(a_1) = 1$).

The necessity of $-a_2a_3$ being a square modulo f is trivial since otherwise the equation $a_1x_1^2 + a_2x_2^2 + a_3x_3^2 = 0$ is not solvable modulo f (one may assume the existence of a solution from the valuation ring where at least one value is a unit). Now assume that $-a_2a_3$ is a square modulo f. This implies that $-\frac{a_2}{a_3}$ is a square as well. Note that $-\frac{a_2}{a_3}$ is a unit in $\mathbb{F}_q(t)_{(f)}$. Hence by Hensel's lemma $-\frac{a_2}{a_3}$ is a square in $\mathbb{F}_q(t)_{(f)}$ (since q is odd). Now solvability follows from Fact 106.

Now let us consider the case where f divides exactly two coefficients, say a_2 and a_3. We apply the following change of variables: $x_2 \leftarrow x_2/f$ and $x_3 \leftarrow x_3/f$. Now we have the equivalent equation $a_1x_1^2 + a_2(x_2/f)^2 + a_3(x_3/f)^2 = 0$. We multiply this equation by f and get the equation $fa_1x_1^2 + a_2/fx_2^2 + a_3/fx_3^2 = 0$. This equation is solvable in $\mathbb{F}_q(t)_{(f)}$ if and only if $\frac{-a_2a_3}{f^2}$ is a square modulo f by the previous point, since f only divides one of the coefficients. This is what we wanted to prove. □

The previous lemma characterized solvability at a finite prime. The next one considers the question of solvability at infinity.

Lemma 109. *Let $a_1, a_2, a_3 \in \mathbb{F}_q[t]$ be nonzero polynomials. Then the following hold:*

1. *If the degrees of the a_i all have the same parity then the equation $a_1x_1^2 + a_2x_2^2 + a_3x_3^2 = 0$ admits a nontrivial solution in $\mathbb{F}_q((\frac{1}{t}))$.*

2. *Assume that not all of the degrees of the a_i have the same parity. Also assume that $deg(a_i) \equiv deg(a_j) \pmod 2$. Let c_i and c_j be the leading coefficients of a_i and a_j respectively. Then the equation $a_1x_1^2 + a_2x_2^2 + a_3x_3^2 = 0$ has a nontrivial solution in $\mathbb{F}_q((\frac{1}{t}))$ if and only if $-c_ic_j$ is a square in $\mathbb{F}_q$.*

Proof. Let $u = 1/t$ and $d_i = deg(a_i)$. Substitute $x_i \leftarrow y_iu^{d_i}$. The coefficient of y_i^2 becomes $a_i' = u^{2d_i}a_i$. Notice that $a_i' = u^{d_i}b_i$ where b_i is a polynomial in u with nonzero constant term c_i. It follows that $v_u(a_i') = d_i$ and the residue of $u^{-d_i}a_i$ modulo u is c_i. Thus both statements follow from Lemma 108 applied to $f = u$ in $\mathbb{F}_q[u]$. □

Remark 110. A form in four variables is always isotropic at infinity if three of its coefficients have the same degree parity. Indeed, let a_i be the coefficient whose degree parity is different. Then setting $x_i = 0$ and applying Lemma 109, (1) implies the desired result.

The next lemmas deal with local solvability of quadratic forms in 4 variables.

Lemma 111. *Let $a_1, a_2, a_3, a_4 \in \mathbb{F}_q[t]$ be square-free polynomials. Let $f \in \mathbb{F}_q[t]$ be a monic irreducible dividing exactly two of the coefficients, a_i and a_j. Let the other two coefficients be a_k and a_l. Then the equation $a_1x_1^2 + a_2x_2^2 + a_3x_3^2 + a_4x_4^2 = 0$ is solvable in $\mathbb{F}_q(t)_{(f)}$ if and only if at least one of the two conditions holds:*

1. *$-a_ka_l$ is a square modulo f*

2. *$-(a_i/f)(a_j/f)$ is a square modulo f*

Proof. First we prove that if any of these conditions hold, the equation is locally solvable at f. If the first condition holds we apply Lemma 108 to show the existence of a nontrivial solution with $x_i = 0$. If the second condition holds we apply the following change of variables: $x_i \leftarrow x_i/f, x_j \leftarrow x_j/f$. With these variables we have the following equation:

$$a_i(x_i/f)^2 + a_j(x_j/f)^2 + a_kx_k^2 + a_lx_l^2 = 0$$

By multiplying this equation by f we get an equation where the coefficients of x_i and x_j are not divisible by f and the the other two are. Now applying Lemma 108 again proves the result.

Now we prove the reverse direction. If the equation $a_1x_1^2 + a_2x_2^2 + a_3x_3^2 + a_4x_4^2 = 0$ has a solution in $\mathbb{F}_q(t)_{(f)}$ then it has a solution in the valuation ring of

$\mathbb{F}_q(t)_{(f)}$. We denote this ring by O. Let $u_1, u_2, u_3, u_4 \in O$ be a solution satisfying that not all of them are divisible by f. Let us consider the equation modulo f:

$$a_1u_1^2 + a_2u_2^2 + a_3u_3^2 + a_4u_4^2 \equiv 0 \ (mod\ f) \tag{4.1}$$

The rest of the proof is divided into subcases depending on how many of u_1, u_2, u_3, u_4 are divisible by f.

If none are divisible by f then we get that $a_ku_k^2 + a_lu_l^2 \equiv 0 \ (mod\ f)$. Therefore $-a_ka_l$ is a square modulo f.

Assume that f divides exactly one of the u_r. If $r = i$ or $r = j$ we again have that $a_ku_k^2 + a_lu_l^2 \equiv 0 \ (mod\ f)$, so $-a_ka_l$ is again a square modulo f. Observe that r cannot be k or l since then equation 4.1 would not be satisfied.

Now consider the case where f divides exactly two of the u_r. If f divides u_i and u_j we have again that $a_ku_k^2 + a_lu_l^2 \equiv 0 \ (mod\ f)$. The next subcase is when f divides exactly one of u_i and u_j, and exactly one of u_k and u_l. Assume that u_i and u_k are the ones divisible by f. This cannot happen since then $a_iu_i^2 + a_ju_j^2 + a_ku_k^2 + a_lu_l^2 \equiv a_lu_l^2 \ (mod\ f)$ and hence the left-hand side of equation 4.1 would not be divisible by f. Finally assume that u_k and u_l are divisible by f. Let $u_k' := u_k/f$ and $u_l' := u_l/f$. We have that $a_1u_1^2 + a_2u_2^2 + a_3u_3^2 + a_4u_4^2 = 0$. We divide this equation by f and obtain the equation $(a_i/f)u_i^2 + (a_j/f)u_j^2 + fa_ku_k'^2 + fa_lu_l'^2 = 0$. We have already seen that this implies that $-(a_i/f)(a_j/f)$ is a square modulo f.

Now suppose that three of the u_r are divisible by f. Observe that u_k and u_l must be divisible by f since otherwise (1) would not be satisfied. Assume that u_i is not divisible by f. However, this cannot happen, because $a_1u_1^2 + a_2u_2^2 + a_3u_3^2 + a_4u_4^2 \equiv a_iu_i^2 \not\equiv 0 \ (mod\ f^2)$.

□

The next lemma is the version of Lemma 111 at infinity.

Lemma 112. *Let $a_1, a_2, a_3, a_4 \in \mathbb{F}_q[t]$ be square-free polynomials. Assume that a_i and a_j are of even degree and the other two, a_k and a_l are of odd degree. Let c_m be the leading coefficient of a_m for $m = 1, \ldots, 4$. Then the quadratic form $a_1x_1^2 + a_2x_2^2 + a_3x_3^2 + a_4x_4^2$ is anisotropic in $\mathbb{F}_q((\frac{1}{t}))$ if and only if $-c_ic_j$ and $-c_kc_l$ are both non-squares in $\mathbb{F}_q$.*

Proof. Let $u = 1/t$. First we do the following change of variables. We substitute $x_r \leftarrow x_rt^{\lceil \frac{-deg(a_r)}{2} \rceil}$ $(r = 1, 2, 3, 4)$. By this substituion we obtain new coefficients $a_r' \in \mathbb{F}_q[u]$. Observe that the u does not divide a_i' and a_j' and the multiplicity of u in a_k' and a_l' is 1. The remainder of a_i' modulo u is c_i, the remainder of a_j' modulo u is c_j. The remainder of a_k'/u modulo u is c_k and the remainder a_l'/u modulo u is c_l. Hence we may apply Lemma 111 with $f = u$ in $\mathbb{F}_q[u]$.

□

Remark 113. If $q \equiv 1 \ (mod\ 4)$ then the lemma says that anisotropy occurs if and only if exactly one of c_i and c_j is a square and the same holds for c_k and c_l. If $q \equiv 3 \ (mod\ 4)$ then the lemma says that anisotropy occurs if and only c_i and c_j are either both squares or both non-squares and the same holds for c_k and c_l. The reason for this is that -1 is a square in $\mathbb{F}_q$ if and only if $q \equiv 1 \ (mod\ 4)$.

We also have the following well-known fact [45, Chapter VI, Theorem 2.2]:

Fact 114. *Let K be a complete field with respect to a discrete valuation whose residue field is a finite field with odd characteristic. Then every non-degenerate quadratic form over K in 5 variables is isotropic.*

We state a variant of the Hasse-Minkowski theorem over $\mathbb{F}_q(t)$ [45, Chapter VI, 3.1]. It was proven by Hasse's doctoral student Herbert Rauter in 1926 [52].

Theorem 115. *A non-degenerate quadratic form over $\mathbb{F}_q(t)$ is isotropic over $\mathbb{F}_q(t)$ if and only if it is isotropic over every completion of $\mathbb{F}_q(t)$.*

For ternary quadratic forms there exists a slightly stronger version of this theorem which is a consequence of the product formula for quaternion algebras or Hilbert's reciprocity law [45, Chapter IX, Theorem 4.6]:

Fact 116. *Let $a_1x_1^2 + a_2x_2^2 + a_3x_3^2$ be a non-degenerate quadratic form over $\mathbb{F}_q(t)$. Then if it is isotropic in every completion except maybe one then it is isotropic over $\mathbb{F}_q(t)$.*

There is a useful fact about local isotropy of a quadratic form [45, Chapter VI, Corollary 2.5]:

Fact 117. *Let $Q(x_1, \ldots, x_n) = a_1x_1^2 + \cdots + a_nx_n^2$ $(n \geq 3)$ be a non-degenerate quadratic form over $\mathbb{F}_q(t)$ where $a_i \in \mathbb{F}_q[t]$. If $f \in \mathbb{F}_q[t]$ is a monic irreducible not dividing $a_1 \cdots a_n$ then Q is isotropic in the f-adic completion.*

We finish the subsection with a formula on the number of monic irreducible polynomials of given degree in a residue class ([68, Theorem 5.1.]):

Fact 118. *Let $a, m \in \mathbb{F}_q[t]$ be such that $deg(m) > 0$ and the $gcd(a, m) = 1$. Let N be a positive integer and let*

$$S_N(a, m) = \#\{f \in \mathbb{F}_q[t] \text{ monic irreducible} \mid f \equiv a \ (mod\ m),\ deg(f) = N\}.$$

Let $M = deg(m)$ and let $\Phi(m)$ denote the number of polynomials in $\mathbb{F}_q[t]$ relative prime to m whose degree is smaller than M. Then we have the following inequality:

$$|S_N(a, m) - \frac{q^N}{\Phi(m)N}| \leq \frac{1}{N}(M + 1)q^{\frac{N}{2}}.$$

As indicated in the Introduction, this fact is an extremely effective bound on the number of irreducible polynomials of a given degree in an arithmetic progression. A similar error term for prime numbers from an arithmetic progression (in a given interval) is not known.

4.1.1 Gram-Schmidt orthogonalization

We propose a version of the Gram–Schmidt orthogonalizition procedure and prove a bound on the size of its output over $\mathbb{F}_q(t)$.

Lemma 119. *Let (V,h) be an n-dimensional quadratic space over $\mathbb{F}_q(t)$. We assume that h is given by its Gram-matrix with respect to a basis $v_1, v_2, \ldots, v_n$ whose entries are represented as fractions of polynomials. Suppose that all the numerators occurring in the Gram matrix have degree at most Δ while the degrees of the denominators are bounded by Δ'. Then there is a deterministic polynomial time algorithm which finds an orthogonal basis $w_1, \ldots, w_n$ with respect to h such that the maximum of the degrees of the numerators and the denominators of the $h(w_i, w_i)$ is $O(n(\Delta + \Delta'))$.*

Proof. We may assume that h is regular. Indeed, we can compute the radical of V by solving a system of linear equations and then continue in a direct complement of it. It is easy to select a basis for this direct complement as a subset of the original basis.

We find an anisotropic vector v'_1 in the following way. If one of the v_i is anisotropic then we choose $v'_1 := v_i$. If all of them are isotropic then there must be an index i such that $h(v_i, v_1) \neq 0$ (otherwise h would not be regular). Since q is odd $v'_1 := v_i + v_1$ will suffice.

Afterwards, we transform the basis $v_1, \ldots, v_n$ into a basis $v'_1, \ldots, v'_n$ which has the property that for every k, the subspace generated by $v'_1, \ldots, v'_k$ is regular. We start with v'_1 which is already anisotropic. Then we proceed inductively. We choose v'_{k+1} in the following way. If some j between $k+1$ and n has the property that the subspace spanned by $v'_1, \ldots, v'_k$ and v_j is regular then we choose $v'_{k+1} := v_j$ where j is the smallest such index. Otherwise we claim that there exists an index j between $k+1$ and n, that $v'_{k+1} = v_{k+1} + v_j$ is suitable. Note that if this is true then this can be checked in polynomial time. Indeed, the cost of the computation is dominated by that of computing $O(n)$ determinants (those of the Gram matrices of the restriction of h to the subspace spanned by $v'_1, \ldots, v'_k$ together with the candidate v'_{k+1}).

Now we prove the claim. Let U be now the subspace generated by $v'_1, \ldots, v'_k$ and let ϕ_U be the orthogonal projection onto the subspace U. (Note that by our assumptions U is a regular subspace and hence V can be decomposed as the orthogonal sum of the subspaces U and $U^\perp$.) Let $v^* = v - \phi_U(v)$, so v^* is in the orthogonal complement of U. We have to prove that if neither v_j is a suitable choice for v'_{k+1} then there exists a j such that $v_{k+1} + v_j$ is suitable. Note that if v_{k+1} is not a suitable choice then the subspace generated by U and v^*_{k+1} is not regular (they generate the same subspace as U and v_{k+1}) hence v^*_{k+1} is isotropic (U was regular). If for any j between $k+1$ and n, the vector v^*_j is anisotropic, we can choose $v'_{k+1} = v^*_j$. Otherwise there must be a j between $k+1$

and n such that $h(v_{k+1}^*, v_j^*) \neq 0$ since h is regular. This implies that $v_{k+1}^* + v_j^*$ is anisotropic since $h(v_{k+1}^* + v_j^*, v_{k+1}^* + v_j^*) = 2h(v_{k+1}^*, v_j^*) \neq 0$. Observe that $v_{k+1}^* + v_j^* = (v_{k+1} + v_j)^*$ so $(v_{k+1} + v_j)^*$ is anisotropic. This implies that the subspace generated by U and $v_{k+1} + v_j$ is regular.

Now we compute an orthogonal basis $w_1, \ldots, w_n$ from the starting basis $v_1', \ldots, v_n'$. We start with $w_1 := v_1'$. Let $w_k := v_k' - u_k$ where u_k is the unique vector from the subspace generated by $v_1', \ldots, v_{k-1}'$ with the property that $h(u_i, v_j') = h(v_i', v_j')$ for every j between 1 and k (uniqueness comes from the fact that the vectors $v_1', \ldots, v_{k-1}'$ span a regular subspace).

Finding w_k is solving a system of k linear equations with k variables. Since the coefficient matrix of the system is non-singular (we chose $v_1', \ldots, v_k'$ in this way) Cramer's rule applies. The same bounds on degrees apply to the Gram-matrix obtained from the v_i' as the original Gram-matrix obtained from the v_i, since the transition matrix $T \in GL_n(\mathbb{F}_q)$. Hence Cramer's rule gives us the bounds on the w_i as claimed. □

4.1.2 Effective isotropy of binary and ternary quadratic forms over $\mathbb{F}_q(t)$

We can efficiently diagonalize regular quadratic forms over $\mathbb{F}_q(t)$ using the version of the Gram–Schmidt-orthoginalization procedure discussed in Subsection 4.1.1. Then a binary form can be made equivalent to $b(x_1^2 - ax_2^2)$ for some $a, b \in \mathbb{F}_q(t)$. The coefficient a is represented as the product of a scalar from $\mathbb{F}_q$ with the quotient of two monic polynomials. We can use the Euclidean algorithm to make the quotient reduced. Then testing whether a is a square can be done in deterministic polynomial time by computing the squarefree factorization of the two monic polynomials and by computing the $\frac{q-1}{2}$th power of the scalar. If a is a square then a square root of it can be computed by a randomized polynomial time method, the essential part of this is computing a square root of the scalar constituent ([4], [62]). Using this square root, linear substitutions "standardizing" hyperbolic forms (making them equivalent to $x_1^2 - x_2^2$ or to x_1x_2, whichever is more desirable) can be computed as discussed in Section 1.5.

Nontrivial zeros of isotropic ternary quadratic forms can be computed in randomized polynomial time using the method of of Cremona and van Hoeij from [14]. Through the connection with quaternion algebras described in Section 1.5, the paper [38] offers an alternative approach. Here we cite the explicit bound on the size of a solution from [14, Section 1].

Fact 120. *Let $Q(x_1, x_2, x_3) = a_1x_1^2 + a_2x_2^2 + a_3x_3^2$ where $a_i \in \mathbb{F}_q[t]$. Then there is a randomized polynomial time algorithm which decides if Q is isotropic and if it is, then computes a nonzero solution (b_1, b_2, b_3) to $Q(x_1, x_2, x_3) = 0$ with polynomials $b_1, b_2, b_3 \in \mathbb{F}_q[t]$ having the following degree bounds:*

1. $deg(b_1) \leq deg(a_2a_3)/2$
2. $deg(b_2) \leq deg(a_3a_1)/2$
3. $deg(b_3) \leq deg(a_1a_2)/2$

4.2 Minimization and splitting

In this section we describe the key ingredients needed for our algorithms for finding nontrivial zeros in 4 or 5 variables. First we do some basic minimization to the quadratic form. Then we split the form $Q(x_1, \ldots, x_n)$ (where $n = 4$ or $n = 5$) into two forms and show the existence of a certain value they both represent assuming the original form is isotropic. The section is divided in two parts. The first deals with quadratic forms in 4 variables, the second with quadratic forms in 5 variables.

4.2.1 The quaternary case

We consider a quadratic form $Q(x_1, x_2, x_3, x_4) = a_1x_1^2 + a_2x_2^2 + a_3x_3^2 + a_4x_4^2$. We assume that all the a_i are in $\mathbb{F}_q[t]$ and are nonzero.

We now give a simple algorithm which minimizes Q in a certain way. We start with definitions:

Definition 121. *We call a polynomial $h \in \mathbb{F}_q[t]$* cube-free *if there do not exist any monic irreducible $f \in \mathbb{F}_q[t]$ such that f^3 divides h.*

Our goal is to replace Q with another quadratic form Q' which is isotropic if and only if Q was isotropic and which has the property that from a nontrivial zero of Q' a nontrivial zero of Q can be retrieved in polynomial time. For instance if we apply a linear change of variables to Q (i.e. we replace Q with an explicitly equivalent form), then this will be the case. However, we may further relax the notion of equivalence by allowing to multiply the quadratic form with a nonzero element from $\mathbb{F}_q(t)$.

Definition 122. *Let Q and Q' be diagonal quadratic forms in n variables. We call Q and Q' projectively equivalent if Q' can be obtained from Q using the following two operations:*

1. *multiplication of Q by a nonzero $g \in \mathbb{F}_q(t)$*
2. *linear change of variables*

We call these two operations projective substitutions.

Definition 123. *We call a diagonal quaternary quadratic form $Q(x_1, x_2, x_3, x_4) = a_1x_1^2 + a_2x_2^2 + a_3x_3^2 + a_4x_4^2$ minimized if it satisfies the following four properties:*

1. *All the a_i are square-free,*
2. *The determinant of Q is cube-free,*
3. *If a monic irreducible f does not divide a_i and a_j (two of the coefficients), but divides the other two, then $-a_ia_j$ is a square modulo f,*
4. *The number of square leading coefficients among the a_i is at least the number of non-square leading coefficients among the a_i.*

Remark 124. By Lemma 108 and Lemma 111, a minimized quadratic form is locally isotropic at any finite prime.

Lemma 125. *There is a randomized algorithm running in polynomial time which either shows that Q is anisotropic at a finite prime or returns the following data:*

1. *a minimized diagonal quadratic form Q' which is projectively equivalent to Q,*
2. *a projective substitution which turns Q into Q'.*

Proof. We factor each a_i. If for a monic irreducible polynomial f, f^{2k} (where $k \geq 1$) divides a_i then we substitute $x_i \leftarrow \frac{x_i}{f^k}$. By iterating this process through the list of primes dividing the a_i we obtain a new equivalent diagonal quadratic form where all the coefficients are square-free polynomials.

Let f be a monic irreducible polynomial in $\mathbb{F}_q[t]$ dividing the determinant of Q. If every a_i is divisible by f then we divide Q by f. Now let us assume that a_1 is the only coefficient not divisible by f. Then we make the following substitution: $x_1 \leftarrow fx_1$. The form obtained this way is still diagonal, and every coefficient is divisible by f. Moreover, f^2 divides exactly one of the coefficients. Divide the form by f. Then the multiplicity of f in the determinant of the new form is exactly 1. If we do this for all monic irreducibles f, whose third power divides the determinant of Q, we obtain a new form whose determinant is cube-free.

Let us assume that each a_i is square-free and that there exists a monic irreducible f which divides exactly two of the a_i. We may assume that f divides a_1

and a_2 but does not divide the other two coefficients. If $-a_3a_4$ is a square modulo f we do nothing. If not, we do a change of variables $x_1 \leftarrow x_1/f, x_2 \leftarrow x_2/f$. If $-\frac{a_1}{f}\frac{a_2}{f}$ is not a square modulo f then we can conclude that Q is anisotropic in the f-adic completion by Lemma 111. Otherwise we continue with the equivalent quadratic form $Q'(x_1, x_2, x_3, x_4) = \frac{a_1}{f}x_1^2 + \frac{a_2}{f}x_2^2 + fa_3x_3^2 + fa_4x_4^2$. This is locally isotropic at f due to Lemma 111.

If the third condition is not satisfied then we multiply the quadratic form by a non-square element from $\mathbb{F}_q$.

Now we consider the running time of the algorithm. First we need to factor the determinant. There are factorisation algorithms which are randomized and run in polynomial time ([4], [7]). We might need a non-square element from $\mathbb{F}_q$. Such an element can be found by a randomized algorithm which runs in polynomial time. The rest of the algorithm runs in deterministic polynomial time (see Remark 1). □

The next lemma is the key observation for our main algorithm.

Lemma 126. *Assume that $a_1x_1^2 + a_2x_2^2 + a_3x_3^2 + a_4x_4^2$ is an isotropic minimized quadratic form with the property that $a_ix_i^2 + a_jx_j^2$ is anisotropic for every $i \neq j$. We define $D = a_1a_2a_3a_4$. Then there exists a permutation $\sigma \in S_4$, an $\epsilon \in \{0, 1\}$ and a residue class b modulo D such that for every monic irreducible $a \in \mathbb{F}_q[t]$ satisfying $a \equiv b \pmod D$ and $deg(a) \equiv \epsilon \pmod 2$, the following equations are both solvable:*

$$a_{\sigma(1)}x_{\sigma(1)}^2 + a_{\sigma(2)}x_{\sigma(2)}^2 = f_1 \cdots f_k g_1 \cdots g_l a \tag{4.2}$$

$$-a_{\sigma(3)}x_{\sigma(3)}^2 - a_{\sigma(4)}x_{\sigma(4)}^2 = f_1 \cdots f_k g_1 \cdots g_l a \tag{4.3}$$

Here $f_1, \ldots, f_k$ are the monic irreducible polynomials dividing both $a_{\sigma(1)}$ and $a_{\sigma(2)}$. Also $g_1, \ldots, g_l$ are the monic irreducibles dividing both $a_{\sigma(3)}$ and $a_{\sigma(4)}$. In addition, b, σ and ϵ can be found by a randomized polynomial time algorithm.

Remark 127. The meaning of this lemma is that if we split the original quaternary form in an appropriate way into two binary quadratic forms then we can find this type of common value they both represent.

Proof. First we show that with an arbitrary splitting into equations 4.2 and 4.3 we can guarantee local solvability (of equations 4.2 and 4.3) everywhere (by choosing a in a suitable way) except at infinity and at a. Then we choose σ and ϵ in a way that local solvability is satisfied at infinity as well. Finally, Fact 116 shows local solvability everywhere (that is at a as well).

For the first part we assume that σ is the identity (this simplifies notation).

Since $a_1x_1^2 + a_2x_2^2$ or $a_3x_3^2 + a_4x_4^2$ are anisotropic over $\mathbb{F}_q[t]$ the question whether equation 4.2 or 4.3 is solvable is equivalent to the following quadratic forms being isotropic over $\mathbb{F}_q(t)$:

$$a_1x_1^2 + a_2x_2^2 - f_1 \cdots f_k g_1 \cdots g_l a z^2 \tag{4.4}$$

$$-a_3x_3^2 - a_4x_4^2 - f_1 \cdots f_k g_1 \cdots g_l a z^2 \tag{4.5}$$

Due to the local-global principle (Theorem 115) the quadratic forms 4.4 and 4.5 are isotropic over $\mathbb{F}_q(t)$ if they are isotropic locally everywhere. Hence equations 4.2 and 4.3 are solvable if and only if they are solvable locally everywhere.

Now we go through the set of primes excluding a and infinity. We check local solvability at every one of them. We have 4 subcases for equation 4.2: the primes f_i; the primes g_j; primes dividing exactly one of a_1 and a_2; remaining primes. The list is similar for equation 4.3. First we show that 4.2 is solvable at all these primes.

Solvability at the f_i

Equation 4.2 is solvable at any f_i since we can divide by f_i and obtain a quadratic form whose determinant is not divisible by f_i. By Fact 117 this is solvable at f_i.

Solvability at a prime g which divides exactly one of a_1 and a_2

We may assume that g divides a_1. Due to Lemma 108 equation 4.2 is solvable in the g-adic completion if $a_2 f_1 \cdots f_k g_1 \cdots g_l a$ is a square modulo g (meaning in the finite field $\mathbb{F}_q[t]/(g)$). Since $(\frac{a_2 f_1 \cdots f_k g_1 \cdots g_l}{g})$ is fixed this gives the condition on a that $(\frac{a}{g}) = (\frac{a_2 f_1 \cdots f_k g_1 \cdots g_l}{g})$. This can be thought of as a congruence condition on a modulo g (this gives a condition whether a should be a square element modulo g or not). Due to the Chinese Remainder Theorem these congruence conditions on a can be satisfied simultaneously. This implies that a has to be in one of certain residue classes modulo the product of these primes. We choose a to be in one of these residue classes.

Solvability at the g_i

Now consider equation 4.2 modulo the g_i. Note that due to minimization neither a_1 nor a_2 are divisible by the g_i. Hence equation 4.2 has a solution in the g_i-adic completion if and only if $-a_1a_2$ is a square modulo g_i. This is satisfied since we have a minimized quadratic form (condition 3 of Definition 123).

Solvability at the remaining primes

Solvability at these primes is satisfied by Fact 117.

Note that solvability of 4.2 holds independently of the choice of a except for primes dividing exactly one of a_1 and a_2. Thus, in the analogous case of

the solvability of 4.3 we have only to consider the case of primes which divide exactly one of a_3 and a_4. These impose congruence conditions again on a. A problem can occur if these congruence conditions are contradictory. We show that this cannot happen. Assume that a monic irreducible polynomial g divides one of a_1, a_2 and one of a_3, a_4, say a_1 and a_3. By the previous discussion we have that in this case $-a_2 a f_1 \cdots f_k g_1 \cdots g_l$ should be a square modulo g and that $a_4 a f_1 \cdots f_k g_1 \cdots g_l$ should be a square modulo g. These can always be satisfied by choosing a to be in a suitable residue class modulo g except if $-a_2 a_4$ is not a square modulo g. However, this cannot happen since our form was minimized (condition 3 of Definition 123).

Now we have proven that for any splitting, equations 4.2 and 4.3 are solvable locally everywhere for suitable primes a except maybe at a or at infinity. We now choose σ and the parity of the degree of a in a way that both 4.2 and 4.3 are solvable at infinity. Then, by Fact 116, 4.2 and 4.3 will be solvable at a as well.

First assume that all a_i have odd degrees. Then we can pick σ arbitrarily and we choose a in a way that $f_1 \cdots f_k g_1 \cdots g_l a$ has odd degree. Then both equations are solvable in $\mathbb{F}_q((\frac{1}{t}))$ by Lemma 109, (1).

Next assume that one coefficient is of even degree and all the others are of odd degree. Pick σ in a way that $a_{\sigma(1)}$ is of even degree and the leading coefficient of $a_{\sigma(2)}$ is a square in $\mathbb{F}_q$. This can be achieved since we have a minimized quadratic form (here we use the fourth condition of Definition 123). Choose a in a way that $f_1 \cdots f_k g_1 \cdots g_l a$ has odd degree. Then equation 4.3 is solvable in $\mathbb{F}_q((\frac{1}{t}))$ due to the same reason as before. Equation 4.2 is also solvable due to Lemma 109, (2).

Now assume that there are two odd degree coefficients and two even degree ones among the a_i. We have that at least two of the a_i has a leading coefficient which is a square (again due to the fact that the form is minimized). We choose σ in such a way that in equations 4.2 and 4.3 one coefficient is of odd degree and the other is of even degree. Assume $a_{\sigma(1)}$ and $-a_{\sigma(3)}$ are of odd degree. Let the leading coefficient of a_i be c_i. If $c_{\sigma(1)}$ and $-c_{\sigma(3)}$ are both squares then we pick a in a way that $f_1 \cdots f_k g_1 \cdots g_l a$ has odd degree. If $c_{\sigma(2)}$ and $-c_{\sigma(4)}$ are both squares we pick a in such a way that $f_1 \cdots f_k g_1 \cdots g_l a$ has even degree. It may occur that $c_{\sigma(1)}, c_{\sigma(2)}, -c_{\sigma(3)}, -c_{\sigma(4)}$ are all squares. In this case there is no degree constraint on a. In these two cases both equations are solvable at infinity by Lemma 109. The only problem occurs if $c_{\sigma(1)}$ and $-c_{\sigma(3)}$ are not both squares and the same holds for $c_{\sigma(2)}$ and $-c_{\sigma(4)}$.

We distinguish two cases depending on whether $q \equiv 1 \ (mod\ 4)$ or $q \equiv 3 \ (mod\ 4)$. First suppose that $q \equiv 1 \ (mod\ 4)$. In this case -1 is square element in $\mathbb{F}_q$. If neither $c_{\sigma(1)}$ nor $c_{\sigma(3)}$ is a square in $\mathbb{F}_q$ then $c_{\sigma(2)}$ and $c_{\sigma(4)}$ must be both squares (we use the fourth condition of Definition 123). Therefore, $-c_{\sigma(4)}$ is a

square since -1 is a square and we have a contradiction (we assumed that one of $c_{\sigma(2)}$ and $-c_{\sigma(4)}$ is not a square). If neither $c_{\sigma(2)}$ nor $c_{\sigma(4)}$ is a square in $\mathbb{F}_q$ then $c_{\sigma(1)}$ and $c_{\sigma(3)}$ must be both squares which is again, a contradiction. The only problem occurs if exactly one of $c_{\sigma(1)}$ and $c_{\sigma(3)}$ is a square and the same is true for $c_{\sigma(2)}$ and $c_{\sigma(4)}$. However, in this case, the form $a_1x_1^2 + a_2x_2^2 + a_3x_3^2 + a_4x_4^2$ is anisotropic by Remark 113.

Suppose that $q \equiv 3 \ (mod\ 4)$. Note that in this case -1 is not a square in $\mathbb{F}_q$. If $c_{\sigma(1)}$ and $-c_{\sigma(3)}$ are non-squares then we have that $c_{\sigma(3)}$ is a square (since -1 is not a square). Then let $\sigma' = \sigma \circ (13)$ (i.e. swap $a_{\sigma(1)}$ with $a_{\sigma(3)}$). Now $c_{\sigma'(1)}$ is a square and so is $-c_{\sigma'(3)}$, hence again we choose a in a way that $f_1 \cdots f_k g_1 \cdots g_l a$ has odd degree and equations 4.2 and 4.3 are solvable at infinity due to Lemma 109. If $c_{\sigma(2)}$ and $-c_{\sigma(4)}$ are non-squares then the situation is essentially the same (let $\sigma' = \sigma \circ (24)$ and choose a in a way that $f_1 \cdots f_k g_1 \cdots g_l a$ has even degree). If exactly one of $c_{\sigma(1)}$ and $-c_{\sigma(3)}$ is a square and the same holds for $c_{\sigma(2)}$ and $-c_{\sigma(4)}$ then the form $a_1x_1^2 + a_2x_2^2 + a_3x_3^2 + a_4x_4^2$ is anisotropic by Remark 113. Indeed, $c_{\sigma(1)}$ and $c_{\sigma(3)}$ are either both squares or both non-squares and the same holds for $c_{\sigma(2)}$ and $c_{\sigma(4)}$.

The cases where there is 1 odd degree one or no odd degree ones amongst the a_i are esentially the same when there are three odd degree ones, or all are of odd degree.

This shows that choosing σ in this way equations 4.2 and 4.3 are solvable locally everywhere, except maybe at a, hence are solvable over $\mathbb{F}_q(t)$ as well by Fact 116.

We conclude by verifying that b, σ and ϵ can be found by a polynomial time algorithm. The computation of a residue class b involves finding non-square elements in finite fields and Chinese remaindering. Both can be accomplished in polynomial time, the first using randomization. Choosing σ and ϵ can be achieved in constant time (by looking at the parity of the degrees of the a_i). □

Remark 128. As seen in the proof there is not just one residue class b modulo D that would satisfy the necessary conditions. Assume that D is divisible by k different monic irreducible polynomials. Then $q^{deg(D)}/3^k$ is a lower bound on the number of appropriate residue classes. Indeed, since modulo each prime half of the nonzero residue classes are squares. However, we will not use this fact later on.

4.2.2 The 5-variable case

We consider a quadratic form $Q(x_1, x_2, x_3, x_4, x_5) = a_1x_1^2 + a_2x_2^2 + a_3x_3^2 + a_4x_4^2 + a_5x_5^2$, where the $a_i \in \mathbb{F}_q[t]$ are nonzero polynomials.

Lemma 129. *There exists a randomized polynomial time algorithm that returns a projectively equivalent diagonal quadratic form Q' whose coefficients are square-free polynomials and whose determinant is cube-free, and a projective substitution which transforms Q into Q'.*

Proof. Making the coefficients of Q' square-free is done a similar fashion as in Lemma 125. If every coefficient is divisible by a monic irreducible f we divide Q by f. If at most 2 coefficients are not divisible by f we do the same trick as in Lemma 125. To implement this for every irreducible polynomial f, we need to factor the determinant. This can be achieved in polynomial time by a randomized algorithm [4]. All the other steps run in deterministic polynomial time. □

Now we prove a Lemma similar to Lemma 126.

Lemma 130. *Let $Q(x_1, x_2, x_3, x_4, x_5) = a_1x_1^2 + a_2x_2^2 + a_3x_3^2 + a_4x_4^2 + a_5x_5^2$, where $D = a_1a_2a_3a_4a_5$ is cube-free and all the a_i are square-free polynomials from $\mathbb{F}_q[t]$. Suppose, $a_ix_i^2 + a_jx_j^2 + a_kx_k^2$ is anisotropic for every $1 \leq i < j < k \leq 5$. Then there exists a permutation $\sigma \in S_5$, an $\epsilon \in \{0, 1\}$ and a residue class b modulo D such that for every monic irreducible $a \in \mathbb{F}_q[t]$ satisfying $a \equiv b \pmod{D}$ and $deg(a) \equiv \epsilon \pmod 2$ the following equations are both solvable:*

$$a_{\sigma(1)}x_{\sigma(1)}^2 + a_{\sigma(2)}x_{\sigma(2)}^2 = f_1 \cdots f_k a \tag{4.6}$$

$$-a_{\sigma(3)}x_{\sigma(3)}^2 - a_{\sigma(4)}x_{\sigma(4)}^2 - a_{\sigma(5)}x_{\sigma(5)}^2 = f_1 \cdots f_k a \tag{4.7}$$

Here $f_1, \ldots, f_k$ are the monic irreducible polynomials dividing both $a_{\sigma(1)}$ and $a_{\sigma(2)}$. In addition, b, σ and ϵ can be found by a randomized polynomial time algorithm.

Remark 131. Assuming that $a_ix_i^2 + a_jx_j^2 + a_kx_k^2$ is anisotropic for every i, j, k allows us to consider the solvability of equations 4.6 and 4.7 as the isotropy of the quadratic forms $a_{\sigma(1)}x_{\sigma(1)}^2 + a_{\sigma(2)}x_{\sigma(2)}^2 - f_1 \cdots f_k az^2$ and $-a_{\sigma(3)}x_{\sigma(3)}^2 - a_{\sigma(4)}x_{\sigma(4)}^2 - a_{\sigma(5)}x_{\sigma(5)}^2 - f_1 \cdots f_k az^2$ hence we can use our lemmas and theorems from the previous sections.

Proof. First we show that for any $\sigma \in S_5$ equation 4.6 is solvable for suitable a at any prime except maybe at infinity and at a. Also if a is suitably chosen then equation 4.7 is solvable everywhere except maybe at infinity. In order to simplify notation we can assume that σ is the identity.

First consider equation 4.6. It is solvable at any of the f_i since a_1 and a_2 are square-free (Lemma 108). It is solvable at any prime not dividing $a_1a_2f_1 \cdots f_k a$ by Fact 117. Let g be a prime that divides a_1 but not a_2. In order to ensure that

4.6 is solvable in the g-adic completion $-a_2 a f_1 \cdots f_k$ has to be a square modulo g. This imposes a congruence condition on a. The situation is the same when looking at a prime dividing a_2 but not a_1.

Now consider equation 4.7. Again if a prime does not divide any of the coefficients then the equation is locally solvable at that prime. The equation is solvable at every f_i (using (1) of Lemma 108 with $z = 0$) since none of the f_i divide a_3, a_4, a_5. Similarly it is also solvable at a (we choose a to differ from the primes occurring in $a_3 a_4 a_5$). If a prime g divides exactly one of a_3, a_4, a_5 then similarly the equation is locally solvable at that prime. Finally consider the case where a prime h divides exactly two out of a_3, a_4, a_5 (say a_3 and a_4). This gives a congruence condition on a. Specifically, $-a f_1 \cdots f_k a_5$ has to be a square modulo h. Note that since for every prime f, f^3 does not divide the determinant of the original quadratic form, the congruence conditions on a coming from equations 4.6 and 4.7 cannot be contradictory.

Now we choose σ and ϵ in a way that both 4.6 and 4.7 become solvable at infinity at the cost of possibly restricting the parity of the degree of a. Then by Fact 116 equation 4.6 will become solvable at a as well. Finally by the local-global principle (Theorem 115) both equations are solvable over $\mathbb{F}_q[t]$.

First if all a_i have odd degree then σ can be chosen arbitrarily and we choose a in a way that $f_1 \cdots f_k a$ has odd degree. This way both equations are solvable at infinity by Lemma 109, (1).

Now consider the case where one coefficient has even degree and the others are of odd degree. Then we choose σ in a way that $a_{\sigma(3)}$ has even degree (and the others are of odd degree). We choose a in a way that $f_1 \cdots f_k a$ has an odd degree. Due to Lemma 109 both equations are solvable at infinity (with $x_{\sigma(3)} = 0$).

Finally assume that there are two a_i-s with even degree. We choose σ in a way that $a_{\sigma(1)}$ and $a_{\sigma(2)}$ are of even degree. We choose a in such a way that $f_1 \cdots f_k a$ has even degree. Now equations 4.6 and 4.7 are solvable at infinity. The remaining cases are essentially the same, we systematically swap "odd" and "even" in the preceding arguments.

Note that b, σ and ϵ can be found in polynomial time (using randomization) by the same reasoning as described at the end of the proof of Lemma 126. □

4.3 The main algorithms

In this section we describe two algorithms. One for solving a quadratic equation in 4 variables and one for 5 variables. The algorithms are similar, however the second uses the first algorithm. The idea of the algorithms is the following. Split the original equation into two and find a common value they both represent and then solve the two equations.

The input of the first algorithm is a diagonal quadratic form defined as $Q(x_1, x_2, x_3, x_4) = a_1x_1^2 + a_2x_2^2 + a_3x_3^2 + a_4x_4^2$ where all a_i are in $\mathbb{F}_q[t]$.

Algorithm 1 (Quaternary case).

1. *Minimize Q using the algorithm from Lemma 125. Minimization either yields that Q is anisotropic (then stop) or returns a new projectively equivalent quadratic form. Let the new minimized form be $Q'(x_1, x_2, x_3, x_4) = b_1x_1^2 + b_2x_2^2 + b_3x_3^2 + b_4x_4^2$. If $b_1, b_2, b_3, b_4 \in \mathbb{F}_q$ then return a nontrivial zero of Q' using the algorithm of [69].*
2. *Check solvability at infinity (Remark 110 and Lemma 112). Check if $b_ix_i^2 + b_jx_j^2$ is isotropic for every pair $i \neq j$. If it is for a pair (i, j) then return a solution.*
3. *Split the quadratic form into equations 4.2 and 4.3 (i.e. find a suitable permutation $\sigma \in S_4$) as discussed in Lemma 126.*
4. *List the congruence conditions on a (as described in Lemma 126) and solve this system of linear congruences. Obtain a residue class b modulo $b_1b_2b_3b_4$ as a result.*
5. *Let d be the degree of $b_1b_2b_3b_4$ and let $N = 4d$ or $N = 4d + 1$ (depending on the degree parity ϵ we need by Lemma 126). Pick a random polynomial f of degree N of the residue class b modulo $b_1b_2b_3b_4$ and check whether it is irreducible. If f is irreducible, then proceed. If not, then repeat this step.*
6. *Solve equations 4.2 and 4.3 using the method of [14].*
7. *By subtracting equation 4.3 from equation 4.2 find a nontrivial zero of Q'.*
8. *Return a nontrivial zero of Q using the reverse substitutions of the substitutions obtained by the algorithm from Lemma 125.*

The input of the second algorithm is a diagonal quadratic form defined as $Q(x_1, x_2, x_3, x_4, x_5) = a_1x_1^2 + a_2x_2^2 + a_3x_3^2 + a_4x_4^2 + a_5x_5^2$ where all a_i are nonzero polynomials in $\mathbb{F}_q[t]$.

Algorithm 2. *1. Minimize Q using the algorithm from Lemma 129. Lemma 129 implies that minimization returns a new projectively equivalent diagonal quadratic form $Q'(x_1, x_2, x_3, x_4, x_5) = b_1x_1^2 + b_2x_2^2 + b_3x_3^2 + b_4x_4^2 + b_5x_5^2$ whose determinant is cube-free and whose coefficients are square-free. If $b_1, b_2, b_3, b_4, b_5 \in \mathbb{F}_q$ then return a nontrivial zero of Q' using the algorithm of [69].*

2. *Split the quadratic form into equations 4.6 and 4.7 (i.e pick $\sigma \in S_5$) as discussed in the proof of Lemma 130. Check if the quadratic forms on the left-hand side of equations of 4.6 and 4.7 are isotropic or not. If one of them is then return a nontrivial solution. Use the algorithm from [14].*

3. *List the congruence conditions on a (as described in Lemma 130) and solve this system of linear congruences. Obtain a residue class b modulo $b_1b_2b_3b_4b_5$ as a result.*

4. *Let d be the degree of $b_1b_2b_3b_4b_5$ and let $N = 4d$ or $N = 4d + 1$ (according to degree parity ϵ we need by Lemma 130). Pick a random polynomial f of degree N of the residue class b modulo $b_1b_2b_3b_4b_5$ and check whether it is irreducible. If f is irreducible, then proceed. If not, then repeat this step.*

5. *Solve equations 4.6 and 4.7 using the method of [14] and Algorithm 1.*

6. *By subtracting equation 4.7 from equation 4.6 find a nontrivial zero of Q'.*

7. *Return a nontrivial zero of Q using the reverse substitutions of the substitutions obtained by the algorithm from Lemma 129.*

Theorem 132. *Algorithm 1 and Algorithm 2 are randomized algorithms (of Las Vegas type) which run in polynomial time in the size of the quadratic form (the largest degree of the coefficients) and in* log *q. Let D be the determinant of the quadratic form. Let $d = deg(D)$. Then both algorithms return a solution of size $O(d)$ (Algorithm 1 also detects if the form is isotropic or not), that is an array of 4 (or 5) polynomials of degree $O(d)$.*

Proof. The correctness of the algorithms follows from Lemmas 126 and 130. We start analyzing the running times of the algorithms. First we deal with Algorithm 1. We consider its running time step by step. The first part of Step 1 runs in polynomial time (is however randomized) as proven in Lemma 125. The second part of Step 1 is deterministic and runs in polynomial time (see [69]). From now on we suppose that the determinant of the minimized form has degree

at least 1. The first part of Step 2 can be executed in deterministic polynomial time (using Fact 106 combined with Lemma 109 and 112). The second part is checking whether a polynomial is a square due to Fact 106. This can be done in polynomial time by computing the square-free factorization of the polynomial ([70]) and checking whether the leading coefficient is a square or not (Remark 107). Step 3 runs in deterministic polynomial time since we only need to check whether certain leading coefficients are squares in $\mathbb{F}_q$ or not. In Step 4 in order to obtain congruence conditions we may have to present a non-square element in a finite field (an extension of $\mathbb{F}_q$ which has degree smaller than the determinant of Q'). This can be done by a randomized algorithm which runs in polynomial time. Note that the probability that a nonzero element in a finite field (whose characteristic is odd) is a square is $1/2$. In the other part of Step 4 we have to solve a system of linear congruences. This can be done in deterministic polynomial time by Chinese remaindering.

Step 5 needs more explanation. After solving the linear congruences we obtain a residue class b modulo D (Lemma 126). By Fact 118 we have that (note that $d \geq 1$):

$$\left| S_N(b, D) - \frac{q^N}{\Phi(D)N} \right| \leq \frac{1}{N}(d+1)q^{\frac{N}{2}}.$$

We choose the degree of a to be $N = 4d$ or $N = 4d+1$ (depending on the parity we need for the degree of a which is discussed in the proof of Lemma 126). We give an estimate on the probability that a polynomial in this given residue class is irreducible. We have the following:

$$\frac{S_N(b, D)}{q^{N-d}} \geq \frac{q^N}{q^{N-d}\Phi(D)N} - \frac{(d+1)q^{\frac{N}{2}}}{Nq^{N-d}} \geq \frac{1}{N} - \frac{d+1}{Nq^{\frac{N}{2}-d}} \geq \frac{1}{N} - \frac{d+1}{Nq^d} \geq \frac{1}{3N}.$$

Here we used the fact that $\frac{d+1}{q^d} \leq 2/3$ since $q \geq 3$ and the function $\frac{d+1}{q^d}$ is decreasing (as a function of d). We also used that $q^d \geq \Phi(D)$.

We pick a uniform random monic element a from the residue class b modulo D. This can be done in the following way. We pick a random polynomial $r(t) \in \mathbb{F}_q[t]$ of degree $N-d$ whose leading coefficient is the inverse of the leading coefficient of D. We consider the polynomial $r' := rD + b$. Then r' has degree N, is monic and is congruent to b modulo D.

The probability that a is irreducible is at least $1/3N$ by the previous calculation. Irreducibility can be checked in deterministic polynomial time [4]. This means that the probability that we do not obtain an irreducible polynomial after $3N$ tries is smaller than $1/2$. Hence this step runs in polynomial time (it is, however, randomized).

The last two steps use the algorithm from [14]. This algorithm is randomized and runs in polynomial time.

The discussion for Algorithm 2 is similar.

Now we turn to the question of the size of solutions. First we consider Algorithm 1. The previous discussion shows that N (the degree of a) can be chosen to be of size $O(d)$. Finally when solving equations 4.2 and 4.3 we use the algorithm from [14]. By Fact 120 we obtain that the solution for 4.2 and 4.3 have size $O(d)$. In the case of Algorithm 2 the same reasoning is valid, except that we have to use Algorithm 1 for solving 4.7. □

Remark 133. Due to Fact 114 and Theorem 115 we have that every quadratic form in 5 or more variables is isotropic over $\mathbb{F}_q(t)$. Hence Algorithm naturally works for diagonal quadratic forms in more than 5 variables. Indeed, we set some variables to zero and use Algorithm 2.

Corollary 134. *Assume that Q is a regular quadratic form (not necessarily diagonal) in either 4 or 5 variables. Let D be the determinant of Q. Let d_1 be the largest degree of all numerators of entries of the Gram-matrix of Q. Let d_2 be the largest degree of all denominators of entries of the Gram-matrix of Q. Then there is randomized polynomial time algorithm which finds a nontrivial zero of Q of size $O(d_1 + d_2)$.*

Proof. First we diagonalize Q using Lemma 119. As a result we obtain a quadratic form with determinant D'. The degree of the numerator and the denominator of D' are both of size $O(d_1 + d_2)$. By clearing the denominators we obtain a quadratic form Q'' with polynomial coefficients and of determinant $O(d_1 + d_2)$. Using Algorithm 1 or 2 (depending on the dimension) we find an isotropic vector. By Theorem 132 the size of the solution vector is $O(d_1 + d_2)$. □

Remark 135. Corollary 134 can be extended to higher dimensions as well. We diagonalize the quadratic form and then set all x_i to zero except 5. Then apply Algorithm 2. Due to diagonalization the size of the solution in this case is $O(n(d_1 + d_2))$.

4.4 Equivalence of quadratic forms

In this section we use the algorithms from the previous sections to compute the following: the Witt decomposition of a quadratic form, a maximal totally isotropic subspace and the transition matrix for two equivalent quadratic forms. We use a presentation in the context of quadratic spaces. We assume that a quadratic space is input by the Gram matrix with respect to a basis.

Theorem 136. *Let (V, h) be a regular quadratic space, $V = \mathbb{F}_q(t)^n$. There exists a randomized polynomial time algorithm which finds a Witt decomposition of (V, h).*

Proof. First we find an orthogonal basis using Lemma 119. This basis can be used to decompose the space into the orthogonal sum of subspaces of dimension 5 and possibly one quadratic form of dimension at most 4 (division with remainder), each with an already computed orthogonal basis. In every 5 dimensional subspace we find an isotropic vector using Algorithm 2. Then we find a hyperbolic plane in each of these subspaces. The subspace generated by this isotropic vector and one of the basis elements from the orthogonal basis of the subspace will be suitable (otherwise h would not be regular restricted to this subspace). We compute its orthogonal complement inside this 5 dimensional subspace. These are all of dimension 3. We find an orthogonal basis in each of these 3 dimensional subspaces using Lemma 119. For their direct sum we again have an orthogonal basis and we iterate the process (we again group by 5 and find hyperbolic planes). We have that V is the orthogonal sum of hyperbolic planes and a subspace of dimension at most 4. Using Algorithm 1 for the quaternary case, the algorithm from [14] for the ternary case, and the method of Subsection 4.1.2 if the dimension is 2, we either conclude that it is anisotropic or find a decomposition into hyperbolic planes and anisotropic part.

Now consider the running time of the algorithm. Assume that h was given by a Gram-matrix where the maximum degree of the numerators is Δ and the maximum degree of the denominators is Δ'. Diagonalization is done in polynomial time via Lemma 119. Also, it produces a diagonal Gram-matrix where every numerator and denominator has degree at most $n(\Delta + \Delta')$. Afterwards we only diagonalize in dimension at most 5. Hence in each step the degrees only grow by a constant factor by Corollary 134. The number of iterations is $O(\log n)$ so the algorithm will run in polynomial time (is however randomized since Algorithm 1 and 2 are randomized). □

Corollary 137. *Let h be a regular bilinear form on the vector space $V = \mathbb{F}_q(t)^n$. Then there exists a randomized polynomial time algorithm which finds a maximal totally isotropic subspace for h.*

Proof. We compute the Witt decomposition of h using Theorem 136. Then we take an isotropic vector from each hyperbolic plane. They generate a maximal totally isotropic subspace [45, Chapter I, Corollary 4.4.]. □

Here we only considered regular bilinear forms. Now we deal with the case where h is not regular.

Corollary 138. *Let (V, h) be a quadratic space. There exists a randomized polynomial time algorithm which finds a Witt decomposition of h.*

Proof. The radical of V can be computed by solving a system of linear equations. Then h restricted to a direct complement of the radical is regular, thus Theorem 136 applies. □

We conclude the section by proposing an algorithm for explicit equivalence of quadratic forms. For simplicity we restrict our attention to regular bilinear forms.

Theorem 139. *Let (V_1, h_1) and (V_2, h_2) be regular quadratic forms over $\mathbb{F}_q(t)$. Then there exists a randomized polynomial time algorithm which decides whether they are isometric, and, in case they are, computes an isometry between them.*

Proof. The quadratic spaces (V_1, h_1) and (V_2, h_2) are equivalent if and only if the orthogonal sum of (V_1, h_1) and $(V_2, -h_2)$ can be decomposed into the orthogonal sum of hyperbolic planes ([45, Chapter I, Section 4]). Hence the question of deciding isometry can be solved using Theorem 136. We turn our attention to the second part of the theorem, to computing an isometry.

First we consider the case of quadratic spaces whose Witt decomposition consist only of the orthogonal sum of hyperbolic planes (i.e., hyperbolic spaces). As shown in Section 1.5, we can transform each of the corresponding binary forms into the standard diagonal form, $x_1^2 - x_2^2$. This results in new bases for the two spaces in which h_1 and h_2 have block diagonal matrices with 2×2 diagonal blocks

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

The linear extension of an approriate bijection between these bases is an isometry. We can efficiently compute the matrix of this map in terms of the original bases.

Assume now that (V_1, h_1) and (V_2, h_2) are isometric anisotropic quadratic spaces. Isometry implies that $(V_1 \oplus V_2, h_1 \oplus -h_2)$ is the orthogonal sum of hyperbolic planes. We find a basis of $V_1 \oplus V_2$ in which the Gram matrix of $h_1 \oplus -h_2$ is of a block diagonal form like above. Then the substitution described in Section 1.5 for equivalence of the two standard binary hyperbolic forms $x_1^2 - x_2^2$ and $x_1 x_2$ can be used to construct a new basis $b_1, b_2, \ldots, b_{2n}$ in which the Gram matrix becomes block diagonal with blocks

$$\begin{pmatrix} 0 & \frac{1}{2} \\ \frac{1}{2} & 0 \end{pmatrix}.$$

(Here n is the common dimension of V_1 and V_2.) Every b_i can be uniquely written in the form $b_i = u_i + v_i$ where $u_i \in V_1$ and $v_i \in V_2$. These can be found

by orthogonal projection. We claim that the vectors $u_1, u_3 \ldots, u_{2n-1}$ are linerly independent. To see this, assume that

$$\lambda_1 u_1 + \lambda_3 u_3 + \ldots + \lambda_{2n-1} u_{2n-1} = 0$$

for some $\lambda_1, \ldots, \lambda_{2n-1}$ not all zero. Then the vector $b = \lambda_1 b_1 + \lambda_3 b_3 + \ldots + \lambda_{2n-1} b_{2n-1}$ is nonzero as the b_i are linearly independent. The orthogonal projection of b to V_1 is zero, whence b is a nonzero vector from V_2. The vector b, as a member of the totally isotropic subspace spanned by $b_1, b_3, \ldots, b_{2n-1}$, must be isotropic. This however contradicts to the anisotropy of $(V_2, -h_2)$. Therefore $u_1, u_3, \ldots, u_{2n-1}$ is a basis of V_1. By symmetry, $v_1, v_3, \ldots, v_{2n-1}$ is a basis of V_2. Now we prove that the Gram matrix of the quadratic form h_1 in the basis $u_1, u_3 \ldots, u_{2n-1}$ is the same as the Gram matrix of h_2 in the basis $v_1, v_3, \ldots, v_{2n-1}$. Observe that since the Gram matrix of $h_1 \oplus -h_2$ had zeros in the diagonal $h_1(u_i, u_i) = h_2(v_i, v_i)$. Since we chose only the odd indices (i.e there are no two indices which differ by 1) we also have that $h_1(u_i, u_j) = h_2(v_i, v_j)$. Thus the linar extension of the map $u_i \to v_i$ ($i = 1, 3, \ldots, 2n-1$) is an isometry between V_1 and V_2. One only has to compute the matrix of this map in terms of the original bases for V_1 and V_2.

In order to find isometries of possibly isotropic quadratic spaces we first compute their Witt decomposition. Then by [45, Chapter I, Section 4] we know that they are isometric if and only if their hyperbolic and anisotropic parts are isometric respectively. An isometry can be found by taking the direct sum of a pair of isometries between the respective parts. Again, one can finish with computing the matrix of this direct sum map in terms of the original bases for V_1 and V_2. □

Remark 140. Theorem 139 can be extended to degenerate quadratic spaces (using Corollary 138). Also, the proof actually shows existence of a reduction from computing isometries to three instances of computing Witt decompositions of quadratic spaces over an arbitrary field of characteristic different from 2.

CHAPTER 5

SPLITTING QUATERNION ALGEBRAS OVER QUADRATIC EXTENSIONS

In this chapter we describe an algorithm for solving the explicit isomorphism problem in the case where $\mathcal{A} \cong M_2(L)$, where L is a quadratic extension of either $\mathbb{Q}$ or $\mathbb{F}_q(t)$, where q is odd. This chapter is based on the conference paper [43], which was later improved in [42]. The case $K = \mathbb{F}_q(t)$ is considered in the last section of [39].

In this thesis, we present these results in a slightly more general framework. Let K be a field such that char $K \neq 2$. Let L be a quadratic extension of K. Then there exists a polynomial time reduction from finding zero divisors in $\mathcal{A} \cong M_2(L)$ to finding nontrivial zeros of quadratic forms over K in several variables. As we have seen in Proposition 35, this can be rephrased as follows. The task of finding a nontrivial zero of a ternary quadratic form over L can be reduced to finding nontrivial zeros of quadratic forms in several variables over K. In other words the case of ternary quadratic forms over a bigger field L can be reduced to computing zeros of quadratic forms over K, however the dimension of the form increases. Fortunately, in the case where $K = \mathbb{Q}$ or $K = \mathbb{F}_q(t)$ there exist algorithms for computing nontrivial zeros of quadratic forms of arbitrary dimension. If $K = \mathbb{Q}$, then the algorithms are due to Simon and Castel ([63], [8]). The low dimensional cases (i.e. where the dimension is 3 or 4) are randomized algorithms which run in polynomial time if one is allowed to call an oracle for factoring integers. Note that there is a randomized polynomial time reduction from computing isotropic vectors for ternary quadratic forms to factoring square-free integers [56] (and finding nontrivial zeros of quaternary quadratic forms is at least as hard as finding nontrivial zeros of ternary quadartic forms [8, Section 1.2]). Surprisingly, if the dimension of a quadratic form is at

least 5, Castel's algorithm [8] runs in polynomial time without the use of oracles assuming the Generalized Riemann Hypothesis (or GRH for short). Simon [63] also proposes an algorithm for higher dimensional quadratic forms but it uses oracles for integer factorisation. However its complexity analysis does not rely on GRH. The case where $K = \mathbb{F}_q(t)$ is considered in Chapter 4.

The chapter is divided into three sections. In the first we recall all the known algorithmic results concerning finding nontrivial zeros of quadratic forms over $\mathbb{Q}$ and $\mathbb{F}_q(t)$. In the second we describe the general procedure, which reduces finding zero divisors in L to finding nontrivial zeros of quadratic forms over K in several variables. In the third section we consider the complexity of the algorithm described in the second section when $K = \mathbb{Q}$ or $K = \mathbb{F}_q(t)$.

5.1 Known algorithmic results

Assume a quaternion algebra $\mathcal{H}$ over a field K (whose characteristic is different from 2) is given by a structure constant representation. We start by describing an algorithm for finding a quaternion basis of $\mathcal{H}$, i.e. a K-basis $1, u, v, uv$ of H such that $u^2, v^2 \in K$ and $uv = -vu$. This algorithm is from [58].

Let $u_0 \in \mathcal{H}$ be an element which is not in the center of $\mathcal{H}$. Note that from a structure constant representation the center of an algebra can be computed efficiently (see Section 2.3).

It is known that the degree of the minimal polynomial of u_0 is exactly 2 ([66, Section 1.1]). Hence there exists $\lambda, \mu \in K$ such that $u_0^2 + \lambda u_0 + \mu = 0$. We rewrite the equation in the following form: $(u_0 + \frac{\lambda}{2})^2 - \frac{\lambda^2}{4} + \mu = 0$. Therefore $u = u_0 + \frac{\lambda}{2}$ is an element whose square is in the center of $\mathcal{H}$, despite the fact that u is not. If $u^2 = 0$ then we have found a zero divisor in $\mathcal{H}$, hence $\mathcal{H}$ is a full matrix algebra over K. By the general procedure described in Chapter 2, an explicit isomorphism between $\mathcal{H}$ and $M_2(K)$ can be computed (from which a quaternion representation is trivial to find). From now on we assume that $u^2 = a \neq 0$.

Now look at the map $\sigma : \mathcal{H} \to \mathcal{H}, w \mapsto uw + wu$. First of all this is a K-linear map which is not the zero map, since $2u^2$ and $2u$ lie in the image. By previous assumptions, neither $2u^2$ nor $2u$ is zero, furthermore, they are linearly independent over K. Thus we have shown that the image of σ has dimension at least 2 over K. Observe that every element in the image of σ commutes with u. Indeed, $u(uw + wu) = uuv + uwu = wa + uwu = (uw + wu)u$. Thus the dimension of the image is at most 2 (for example by the double centralizer theorem, see [50]). Putting these facts together we obtain that the image of σ is of dimension 2 over K. Therefore the kernel of σ is also a two dimensional vector space over K. Let

v be in the kernel of σ, i.e. $uv + vu = 0$. Since char $K \neq 2$ and u is invertible, v must be non-central.

Observe that v^2 commutes with u. Indeed, $uv^2 = (uv)v = -(vu)v = -v(uv) = v(vu) = v^2u$. Therefore v^2 is the K-linear combination of 1 and u. However, since v is not in the center of $\mathcal{H}$, v^2 is also a K-linear combination of 1 and v. Since $uv = -vu$ this can only happen if v^2 is in K. Let $v^2 = b$. Since v was an arbitrary element of the kernel of σ, a suitable v can be found by computing the kernel of σ, which boils down to solving a system of homogeneous linear equations over K. Again, if $b = 0$ we already have a zero divisor. Suppose that $b \neq 0$. Finally, it is easy to see that $1, u, v, uv$ are linearly independent, therefore this is a quaternion basis of $\mathcal{H}$.

Now we list some known algorithmic results about quadratic forms over $\mathbb{Q}$. We start with ternary forms. In the introduction of this chapter we already mentioned that this task is in a sense at least as hard as factoring integers (as proven by Rónyai in [56]). Ivanyos and Szántó [37] proposed a polynomial time ff-algorithm to solve this problem. They construct a maximal order in the quaternion algebra (using the algorithm from [34]) and use lattice reduction to find a zero divisor. Cremona and Rusin gave a different algorithm [15] for the same task. They proposed an algorithm which finds nontrivial solutions of homogeneous quadratic equations in three variables (these two tasks are essentially the same as seen in Section 1.4). Now we state the results of Simon and Castel for quadratic forms of higher dimension:

Fact 141 (Simon [63])**.** *There is a randomized algorithm which finds a nontrivial zero of a quadratic form over $\mathbb{Q}$ in dimension 4 (or higher) if one exists, or concludes that the form is anisotropic. The running time is polynomial if one is allowed to call oracles for factoring integers.*

This task is also at least as hard as factoring integers since quadratic forms in dimension 4 with square discriminant correspond to quadratic forms of dimension 3 (see [8, Section 1.2]). Castel [8] improved these algorithms and obtained an algorithm which works in dimension 5 (and above) and does not depend on factoring integers:

Fact 142 (Castel [8])**.** *Assuming GRH, there is a randomized polynomial time algorithm which finds an isotropic vector for an indefinite quadratic form (over $\mathbb{Q}$) in dimension 5 (or more).*

All the algorithms we use in the $K = \mathbb{F}_q(t)$ case in this chapter are described in Chapter 4.

5.2 The general algorithm

Let K be a field such that char $K \neq 2$. Let L be a quadratic extension of K, i.e. $L = K(\sqrt{d})$, where $d \in K$ and d is not a square in K. We consider the problem of finding a zero divisor in a central simple L-algebra $\mathcal{A}$, which is isomorphic to $M_2(L)$ and is given by structure constants. We show how this problem can be reduced (in polynomial time) to computing nontrivial zeros of quadratic forms in several variables over K.

First we construct a subalgebra $\mathcal{B}$ in $\mathcal{A}$ which is a quaternion algebra over K. Then, with this information at our hands, we construct a zero divisor. We outline the main steps of the algorithm:

Algorithm 3.

- *Find an element $u \in \mathcal{A}$ such that tr $u = 0$ (recall that the trace of an element is the sum of the element and its conjugate, for deatils see Section 1.4) and $u^2 \in K$ and $u \neq 0$. If $u^2 = r^2$, where $r \in K$, then return the zero divisor $u - r$.*
- *Find a nonzero element v such that $uv = -vu$ and $v^2 \in K$. If $v^2 = r^2$ such that $r \in K$, then return the zero divisor $v - r$.*
- *Let $\mathcal{B}$ be the K-subspace generated by $1, u, v, uv$. $\mathcal{B}$ is a quaternion algebra over K. If $\mathcal{B} \cong M_2(K)$ then find a zero divisor in $\mathcal{B}$.*
- *If $\mathcal{B}$ is a division algebra then find an element $s \in \mathcal{B}$ such that $s^2 = d$. Return the zero divisor $s - \sqrt{d}$.*

The key to each step is finding a nontrivial zero of a quadratic form in several variables. First observe that taking square roots is essentially the same as finding a nontrivial zero of a quadratic form in 2 variables. In Step 1 we solve a homogeneous quadratic equation in 6 variables, in Step 2 and 3 an equation in 3 variables and finally in Step 4 an equation in 4 variables. Now we proceed step by step.

Proposition 143. *Let $\mathcal{A} \cong M_2(K(\sqrt{d}))$ be given by structure constants. Then finding a traceless nonzero $l \in \mathcal{A}$ (i.e., tr $l = 0$), such that $l^2 \in K$ can reduced to finding a nontrivial zero of a certain quadratic form in 6 variables over K. Morevover, the existence of such an l is equivalent to the form being isotropic over K.*

Proof. First we construct a quaternion basis $1, w, w', ww'$ of $\mathcal{A}$. We have the following:

$$w^2 = r_1 + t_1\sqrt{d},\ w'^2 = r_2 + t_2\sqrt{d}$$

Every traceless element is in the $K(\sqrt{d})$-subspace generated by w, w' and ww'. The condition $l^2 \in K$ gives the following equation ($s_1, \ldots, s_6 \in K$):

$$((s_1 + s_2\sqrt{d})w + (s_3 + s_4\sqrt{d})w' + (s_5 + s_6\sqrt{d})ww')^2 \in K$$

If we expand this we obtain:

$$\begin{aligned}((s_1 + s_2\sqrt{d})w + (s_3 + s_4\sqrt{d})w' + (s_5 + s_6\sqrt{d})ww')^2 = \\ (s_1^2 + ds_2^2 + 2s_1s_2\sqrt{d})(r_1 + t_1\sqrt{d}) + (s_3^2 + ds_4^2 + 2s_3s_4\sqrt{d})(r_2 + t_2\sqrt{d}) - \\ (s_5^2 + ds_6^2 + 2s_5s_6\sqrt{d})(r_1 + t_1\sqrt{d})(r_2 + t_2\sqrt{d})\end{aligned}$$

The coefficient of $\sqrt{d}$ has to be zero:

$$t_1s_1^2 + t_1ds_2^2 + 2r_1s_1s_2 + t_2s_3^2 + t_2ds_4^2 + 2r_2s_3s_4 - (r_1t_2 + t_1r_2)s_5^2 - \tag{5.1}$$

$$(r_1t_2 + t_1r_2)ds_6^2 - 2(r_1r_2 + t_1t_2d)s_5s_6 = 0 \tag{5.2}$$

The left hand side of this equation is a quadratic form in 6 variables which is isotropic over K if and only if there exists an $l \in \mathcal{A}$ such that $tr\ l = 0$ and $l^2 \in K$.

Remark 144. If $K = \mathbb{Q}$, one can actually show that the left-hand side of equation 5.1 is an indefinite quadratic form, hence it is always isotropic. This implies that every quaternion algebra over $\mathbb{Q}(\sqrt{d})$ contains a traceless element whose square is in $\mathbb{Q}$.

□

We proceed to the next step:

Proposition 145. *Let $\mathcal{B} = \mathcal{H}_{K(\sqrt{d})}(a, b + c\sqrt{d})$ given by: $u^2 = a, v^2 = b + c\sqrt{d}$, where $a, b, c \in K$, $c \neq 0$. Then finding a nonzero element v' such that $uv' + v'u = 0$ and $v'^2 \in K$ is equivalent to finding a zero divisor in the quaternion algebra $\mathcal{H}_K((\frac{b}{c})^2 - d, a)$.*

Remark 146. Finding a zero divisor in a quaternion algebra over K is equivalent to finding a nontrivial zero of a quadratic form in three variables over K by Proposition 35. The reason we stated it this way (and not as a reduction to finding isotropic vectors of quadratic forms) is the following. The quaternion algebra $\mathcal{H}_K((\frac{b}{c})^2 - d, a)$ is split if and only if the quaternion algebra $\mathcal{H}_K(b^2 - cd^2, a)$ is split. Note that this means that $\mathcal{H}_K(N_{L|K}(b + c\sqrt{d}), a)$ is split where $N_{L|K}$ is the norm map from L to K. A similar statement can be found in ([16, Part II, Theorem 7]) in a slightly more general context. Basically, this says that the existence a quaternion subalgebra over a smaller field K is equivalent to the splitting of a different quaternion algebra over K.

The reason we emphasize this here, is that we believe that this step could be generalized to field extension of degree greater than 2 in a similar fashion. However, this statement is somewhat stronger than just a pure existential theorem, since if we find a zero divisor in $\mathcal{H}_K((\frac{b}{c})^2 - d, a)$ then we also can compute a quaternion subalgebra over K in $\mathcal{B}$ (not just conclude that it exists).

Proof. Since v' anticommutes with u (i.e. $uv' + v'u = 0$) it must be a $K(\sqrt{d})$-linear combination of v and uv. This implies we have to search for $s_1, s_2, s_3, s_4 \in K$ such that:

$$((s_1 + s_2\sqrt{d})v + (s_3 + s_4\sqrt{d})uv)^2 \in K$$

Expanding this expression we obtain the following:

$$((s_1 + s_2\sqrt{d})v + (s_3 + s_4\sqrt{d})uv)^2 =$$
$$(s_1^2 + s_2^2 d + 2s_1 s_2\sqrt{d})(b + c\sqrt{d}) - (s_3^2 + s_4^2 d + 2s_3 s_4\sqrt{d})a(b + c\sqrt{d})$$

The coefficient of $\sqrt{d}$ has to be zero, which implies the following equation:

$$c(s_1^2 + s_2^2 d) + 2bs_1 s_2 - ac(s_3^2 + s_4^2 d) - 2abs_3 s_4 = 0$$

First we divide by c. Note that c is nonzero. Let $f = b/c$.

$$s_1^2 + s_2^2 d + 2fs_1 s_2 - a(s_3^2 + s_4^2 d) - 2afs_3 s_4 = 0 \tag{5.3}$$

In order to diagonalize the left hand side of equation 5.3, consider the following change of variables: $x := s_1 + fs_2$, $y := s_2$,$z := s_3 + s_4 f$, $w := s_4$. Note that the transition matrix of this change is an upper triangular matrix with 1-s in the diagonal so it has determinant 1 (this means that we obtain a quadratic form on the left hand-side of the equation which is equivalent to the original one). In terms of these new variables the equation takes the following form:

$$x^2 + (d - f^2)y^2 - az^2 - a(d - f^2)w^2 = 0.$$

Finding a solution of this is equivalent to finding a zero divisor in the quaternion algebra $\mathcal{H}_K(f^2 - d, a)$ by Proposition 35. □

We have the following lemma:

Lemma 147. *Let $\mathcal{A} \cong M_2(K(\sqrt{d}))$. Let $l \in \mathcal{A}$ be such that $l^2 \in K$ and l is not in the center of $\mathcal{A}$ and l^2 is not a square in K. Then there exists an element l' such that $l'^2 \in K$ and $ll' + l'l = 0$.*

Proof. There exists a subalgebra $\mathcal{A}_0$ in $\mathcal{A}$ which is isomorphic to $M_2(K)$. In this subalgebra there is an element l_0 for which l and l_0 have the same minimal polynomial over $K(\sqrt{d})$. Moreover, there exists an $m \in \mathcal{A}$ such that $l = m^{-1}l_0m$ ([66, Theorem 2.1.]). There exists a nonzero $l_0' \in \mathcal{A}_0$ such that $l_0l_0' + l_0'l_0 = 0$. Let $l' = m^{-1}l_0'm$. We have that $l'^2 = m^{-1}l_0'mm^{-1}l_0m = m^{-1}l_0^2m = l_0^2$, hence $l'^2 \in K$. Since conjugation by m is an automorphism we have that $ll' + l'l = m^{-1}(l_0l_0' + l_0'l_0)m = m^{-1}0m = 0$. Thus we have proven the existence of a suitable element l'. □

Now let us take a look at Algorithm 3. We showed in Proposition 143 that Step 1 can be reduced to finding a nontrivial zero of a quadratic form in 6 variables. In Proposition 145 we showed that Step 2 is equivalent to finding a nontrivial zero of a quadratic form in 3 variables. Lemma 147 shows that Step 2 always returns a solution assuming $\mathcal{A} \cong M_2(K(\sqrt{d}))$. The second part of Step 3 is finding an isotropic vector for a quadratic form in 3 variables over K by Proposition 35. Finally, we show how to reduce tha last step to finding a nontrivial zero of a quadratic form in 4 variables:

Proposition 148. *Let $\mathcal{A} \cong M_2(K(\sqrt{d}))$ and let $\mathcal{H}$ be a subalgebra of $\mathcal{A}$ which is quaternion algebra over K. Moreover, assume that $\mathcal{H}$ is a division algebra. Then $\mathcal{H}$ contains an element u which is not in the center of $\mathcal{A}$ and $s^2 = d$. Such an element u can be constructed by finding a nontrivial zero of a quadratic form in 4 variables.*

Proof. The existence of such an s follows from the fact that $\mathcal{H}$ is split by $K(\sqrt{d})$ (as $\mathcal{A} \cong K(\sqrt{d}) \otimes_K \mathcal{H}$) and therefore contains $K(\sqrt{d})$ as a subfield [66, Theorem 1.2.8].

Now let $1, u, v, uv$ be a quaternion basis of $\mathcal{H}$ with $u^2 = a, v^2 = b$. Every non-central element whose trace is zero (in $\mathcal{H}$) is a K-linear combination of u, v and uv. Hence finding an element s such that $s^2 = d$ is equivalent to solving the following equation:

$$ax_1^2 + bx_2^2 - abx_3^2 = d \tag{5.4}$$

Since $\mathcal{H}$ is a division algebra, the quadratic form $ax_1^2 + bx_2^2 - abx_3^2$ is anisotropic. Thus solving equation 5.4 is equivalent to finding a nontrivial zero of the quadratic form $ax_1^2 + bx_2^2 - abx_3^2 - dx_4^2$. □

5.3 Complexity analysis over $\mathbb{Q}$ and $\mathbb{F}_q(t)$

Theorem 149. *Let $\mathcal{A} \cong M_2(\mathbb{Q}(\sqrt{d}))$ be given by structure constants, where d is a square -free integer. Then there exists an randomized algorithm running in polynomial time, if one is allowed to call an oracle for factoring integers, which finds a zero divisor in $\mathcal{A}$.*

Proof. First we compute a quaternion basis of $\mathcal{A}$. This can be done in polynomial time by the algorithm described at the beginning of Section 5.1 (see also [58]). Now we go through the steps of Algorithm 3.

Proposition 143 implies that the first part of the first step can be executed in polynomial time if one is allowed to call oracles for integer factorization using the algorithm from [63]. Actually, this step can also be carried out without the use of oracles, using Castel's algorithm [8], but the running time of that algorithm is only polynomial if GRH holds. Checking that a rational number is a square can also be achieved in polynomial time. The second step can be completed via the algorithm from [15] or [33] (this is a consequence of Proposition 145). They are both ff-algorithms which run in polynomial time. However, an ff-algorithm can be made into a randomized polynomial time algorithm using oracles for integer factorization by applying Berlekamp's algorithm [4] for factoring polynomials over finite fields. The same is true for step 4. The final step involves finding a nontrivial zero of a quadratic form in 4 variables by Proposition 148. Here we invoke again the algorithm of Simon [63] which runs in polynomial time if one is allowed to call oracles for factoring integers. □

Theorem 150. *Let $\mathcal{A} \cong M_2(\mathbb{F}_q(t)(\sqrt{d}))$ be given by structure constants where $d \in \mathbb{F}_q[t]$ is a square-free polynomial. Then there exists a randomized polynomial time algorithm which finds a zero divisor in $\mathcal{A}$.*

Proof. The proof is similar to the proof of Theorem 149. The only difference is that we use the algorithms from the previous chapter for finding nontrivial zeros of 4 and 6-variable quadratic forms, and the algorithm from [11] for 3-variable quadratic forms (or the algorithm from Chapter 3). Also note that computing square roots in this setting (which one may need in Step 1) can be done in polynomial time [62]. □

CHAPTER 6

IMPLEMENTATIONS AND COMPUTATIONAL EXPERIMENTS

In this chapter we consider implementations for Algorithm 1 and Algorithm 3. We implemented both algorithms in the computational algebra system MAGMA [47].

This chapter is divided into two sections, one for each algorithm. In both sections we consider the running times of the algorithms for various types of inputs and discuss the most important steps of implementation. Our goal with this chapter is to illustrate that both algorithms are easy to implement and perform well in practice. However, we also address some theoretical questions via computation (for example what percentage of quaternion algebras splits over a quadratic extension of $\mathbb{Q}$). Codes of both algorithms can be found in the Appendix of this thesis.

6.1 Finding nontrivial zeros of quadratic forms in four variables over $\mathbb{F}_q(t)$

We start with the implementation. The main function's input are four polynomials a_1, a_2, a_3, a_4, defined over a finite field. It either outputs that the equation $a_1x_1^2 + a_2x_2^2 + a_3x_3^2 + a_4x_4^2 = 0$ has no solutions, or returns the following data: the minimization b_1, b_2, b_3, b_4 of a_1, a_2, a_3, a_4 (see Lemma 125) and a nonzero solution vector to the equation $b_1x_1^2 + b_2x_2^2 + b_3x_3^2 + b_4x_4^2 = 0$. From a solution vector for the minimized equation it is easy to compute a nonzero solution to the original equation $a_1x_1^2 + a_2x_2^2 + a_3x_3^2 + a_4x_4^2 = 0$.

There are four major auxiliary functions. The function "minimization" ei-

ther detects anisotropy at a finite prime or returns a minimized quadratic form (future improvement could be a function that also returns the transition matrix of the minimization). The function "infinity" checks isotropy at infinity. The running time of both functions is quite fast. The only computational challenge is the factorization of the determinant of the quadratic form. Using these two functions one can already decide whether the quadratic form is isotropic or not. Computational experiments suggest that a quaternary quadratic form chosen at random should be isotropic. This also follows from the lemmas in Section 4.1.

The third major function is "splitting". This function returns a sequence consisting of the following: the splitting defined in Lemma 126, an indicator on the degree of the polynomial a (using the terminology of Lemma 126, this is the degree parity of $f_1 \cdots f_k g_1 \cdots g_l a$) and the leading coefficient of a. The function "solving" returns a nontrivial zero of a minimized quadratic form. This is the most time consuming part of the algorithm as it uses a built-in algorithm for finding a nontrivial zero of two ternary quadratic forms (based on the algorithm from [14]). This function also contains a step for generating an irreducible polynomial of a certain residue class (not a built-in MAGMA function). This performs surprisingly well (even for large finite fields). We use the degree bounds from Theorem 132. For technical reasons this step is not a separate function, however, it can easily made into one. Also, a function generating an irreducible polynomial from a residue class, may be of independent interest.

We provide some remarks on the running time of the algorithm. Checking whether a given quadratic form in four variables is isotropic usually takes less then one second if the coefficient polynomials are randomly chosen polynomials of degree at most 100 and the cardinality of the finite field is a 7 digit decimal number. Actually, the dominant parameter here is not the size of the finite field, but the size of the characteristic of the finite field. The reason is that the hardest computational task here is the factorization of the determinant.

Finding a nontrivial zero of an isotropic form takes more time. However, computational experiments show that the most time consuming part is the solution of the two ternary forms. If the degree of the inputs was at most D, then the degree of the output was at most $9D$ (assuming the input is minimized). This is the first step in implementing the algorithms from Chapter 4, as all other algorithms there are based on Algorithm 1. Implementing those algorithms could be a goal for a future project.

We conclude by a table of running times (Table 6.1). In each row, we have the following data: the degree of the input polynomials (which are minimized), the size of the finite field, the degree of the solution vector (i.e., the largest degree among the components of the solution vector) and the running time in seconds. The input polynomials were chosen at random (the degrees of the input poly-

Table 6.1. *Running times of Algorithm 1*

deg a_1	deg a_2	deg a_3	deg a_4	Degree of the solution	q	Running time
29	29	29	29	246	1009694033	400,767
9	9	9	9	76	1009694033	17,190
9	9	9	9	76	3^8	10,420
70	71	67	63	574	3	82,600
47	44	44	50	391	3	33,603
29	29	29	29	246	1009694033	403,450
99	99	99	99	no solution	3	0,062
101	101	101	101	839	3	410,969

nomials were not chosen randomly, just their coefficients).

6.2 Finding zero divisors in quaternion algebras over $\mathbb{Q}(\sqrt{d})$

First, we comment on the implementation, which is based on Algorithm 3. Note that a key ingredient of Algorithm 3 is finding nontrivial zeros of quadratic forms in several variables over $\mathbb{Q}$. This task is accomplished by the MAGMA-function "IsotropicSubspace" which is based on the algorithms from [63].

The input of the main function is 5 integers d, a_1, a_2, a_3, a_4: d defines the quadratic field, a_1, a_2, a_3, a_4 define the quaternion algebra $\mathcal{H} = \mathcal{H}_{\mathbb{Q}(\sqrt{d})}(a_1 + a_2\sqrt{d}, a_3 + a_4\sqrt{d})$. First, one finds an element $u \in \mathcal{H}$ such that $u^2 \in \mathbb{Q}$. This is is not a separate function, but is incorporated in the main function for certain technical reasons. However, it may be of independent interest. Recall that such an element always exists (even if $\mathcal{H}$ is a division algebra). Finding an element v which anticommutes with u, and $v^2 \in \mathbb{Q}$, is implemented as a separate function entitled "anticommute". Several small auxiliary functions were also implemented. For instance a function, which finds a zero divisor in a quaternion algebra over $\mathbb{Q}$ (it returns a zero divisor in terms of the quaternion basis), and a function which finds an anticommuting element for a traceless element (an element whose trace is zero). The algorithms used here are not novel, but it seems they have not yet been implemented into MAGMA.

One can easily verify the correctness of the calculation in the following way. First, MAGMA has a built-in function "IsMatrixRing" (based on the paper [67]), which decides whether a quaternion algebra splits or not. However, this function does not return a zero divisor if the quaternion algebra splits (its output

is either "true" or "false"). If Algorithm 3 outputs an element claiming to be a zero divisor then this claim can be checked by applying the built-in MAGMA function "Norm" to it. If it returns zero, then we have indeed found a zero divisor (by computing the trace of the element, a zero divisor pair can also be found instantly).

Now we share some remarks about the running time. Algorithm 3 decides whether $\mathcal{H}$ splits or not. However, computational experiments have shown that deciding splitting can be accomplished faster via the function "IsMatrixRing". The reason for this is pretty simple. Algorithm 3 always finds an element whose square is rational, hence it always has to find a nontrivial zero of a quadratic form in 6 variables. Then it still needs to solve quadratic equations in 3 and 4 variables and it may happen that the ternary equation is solvable and only the equation in 4 variables is not. So if we are only interested in splitting, then "IsMatrixRing" is more efficient. However, the data Algorithm 3 computes is far from irrelevant. Computing a zero divisor if $\mathcal{H}$ is split seems to be quite fast if d is around 2^{30} (the running time is around 1 second).

Some questions arose while experimenting with this algorithm. Assume that $\mathcal{H}$ is a division quaternion algebra over $\mathbb{Q}(\sqrt{d})$ which contains a quaternion subalgebra over $\mathbb{Q}$. Does Algorithm 3 compute such a quaternion subalgebra of $\mathcal{H}$ over $\mathbb{Q}$? This is not apriori clear, since we only proved this claim if $\mathcal{H}$ is a full matrix algebra. We will show that this is indeed true for division algebras as well. However, our proof is not as elementary as the proof of Lemma 147.

The other natural question is, if the isomorphism class of the quaternion subalgebra is unique or not. If $\mathcal{H}$ is split then the answer to this is negative. We give an example for a division algebra for which the answer is negative as well.

We start with a definition.

Definition 151. *Let K be a field, and let L be a quadratic separable extension of K. Let τ be the non-trivial automorphism of L fixing K. Let $\mathcal{A}$ be an algebra over L. Then σ is an* involution of the second kind *if the following hold:*

1. $\sigma(x+y) = \sigma(x) + \sigma(y)$ *for all* $x, y \in \mathcal{A}$,
2. $\sigma(xy) = \sigma(y)\sigma(x)$ *for all* $x, y \in \mathcal{A}$,
3. $\sigma(\sigma(x)) = x$ *for all* $x \in \mathcal{A}$,
4. σ *restricted to* L *is* τ.

Let $\mathcal{H}$ be a quaternion algebra over $\mathbb{Q}(\sqrt{d})$. If $\mathcal{H}$ has a quaternion subalgebra over $\mathbb{Q}$ with quaternion basis $1, u, v, uv$ then an involution σ of the second kind

can be constructed on $\mathcal{H}$ as follows:

$$x = (\lambda_1 + \mu_1\sqrt{d}) + (\lambda_2 + \mu_2\sqrt{d})u + (\lambda_3 + \mu_3\sqrt{d})v + (\lambda_4 + \mu_4\sqrt{d})uv \quad (6.1)$$

$$\sigma : x \mapsto (\lambda_1 - \mu_1\sqrt{d}) - (\lambda_2 - \mu_2\sqrt{d})u + (\lambda_3 - \mu_3\sqrt{d})v + (\lambda_4 - \mu_4\sqrt{d})uv \quad (6.2)$$

The converse is also true [40, Chapter I, Proposition 2.22], meaning that $\mathcal{H}$ has a rational quaternion subalgebra if it possesses an involution of the second kind (one composes this involution with the usual quaternion conjugation and considers the elements fixed under this map).

There is a theorem characterizing the existence of an involution of the second kind (special case of [40, Theorem 3.1]). The theorem uses the notion of corestriction of central simple algebras. We do not define the corestriction of an algebra here (for a definition see [40, Chapter I, Section 3B] where it is called the norm of an algebra), as we do not use it later on. The only thing we use here is that it is a map which maps a central simple algebra over $\mathbb{Q}(\sqrt{d})$ to a central simple algebra over $\mathbb{Q}$.

Fact 152. *Let $\mathcal{H}$ be a quaternion algebra over $\mathbb{Q}(\sqrt{d})$. Then $\mathcal{H}$ contains a subalgebra $\mathcal{B}$ which is a quaternion algebra over $\mathbb{Q}$ if and only if $Cor_{\mathbb{Q}(\sqrt{d})|\mathbb{Q}}(\mathcal{H})$ (the corestriction of $\mathcal{H}$ with respect to the field extension $\mathbb{Q}(\sqrt{d})|\mathbb{Q}$) splits.*

We also need the following fact, called the projection formula [16, Part II, Theorem 7]:

Fact 153. *Let $\mathcal{H}_{\mathbb{Q}(\sqrt{d})}(a, b + c\sqrt{d})$ be a quaternion algebra over $\mathbb{Q}(\sqrt{d})$ where $a, b, c \in \mathbb{Q}$. Then $Cor_{\mathbb{Q}(\sqrt{d})|\mathbb{Q}}(\mathcal{H})$ is Brauer equivalent to $\mathcal{H}_{\mathbb{Q}}(a, b^2 - c^2 d)$.*

Now we are ready to prove the following:

Proposition 154. *Let $\mathcal{H}$ be a quaternion algebra over $\mathbb{Q}(\sqrt{d})$ which contains a quaternion subalgebra over $\mathbb{Q}$. Let $s \in \mathcal{H}$ such that $s^2 \in \mathbb{Q}$. Then there exists an element r such that $sr + rs = 0$ and $r^2 \in \mathbb{Q}$.*

Remark 155. Proposition 154 implies that Algorithm 3 computes a quaternion subalgebra over $\mathbb{Q}$ even if $\mathcal{H}$ is division algebra containing a quaternion subalgebra over $\mathbb{Q}$.

Proof. Let $s^2 = a$, where $a \in \mathbb{Q}$. Let $s' \in \mathcal{H}$ be such that $ss' + s's = 0$ and $s'^2 = b + c\sqrt{d}$. We have that $\mathcal{H} \cong \mathcal{H}_{\mathbb{Q}(\sqrt{d})}(a, b + c\sqrt{d})$. Proposition 145 says that a suitable r exists if and only if $\mathcal{H}_{\mathbb{Q}}(a, b^2 - c^2 d)$ splits. So if we show that this is indeed the case then we are done. By Fact 152 we have that $Cor_{\mathbb{Q}(\sqrt{d})|\mathbb{Q}}(\mathcal{H})$ splits since $\mathcal{H}$ contains a quaternion subalgebra over $\mathbb{Q}$. By the projection formula

Table 6.2. *Running times of Algorithm 3, where d defines the quadratic field, and the columns a_i contain the number of digits of a_i*

d	a_1	a_2	a_3	a_4	Running time
5	2	2	6	4	0,25
5	5	3	2	3	0,12
223303	16	9	2	3	1,20
645945847	3	3	25	16	1,23
18050605201	28	17	2	3	5,66
6759916343	27	16	2	3	72,94
4985824399	26	15	2	3	3,10
92641259	23	14	2	3	10,82

(Fact 153) we have that $Cor_{\mathbb{Q}(\sqrt{d})|\mathbb{Q}}(\mathcal{H})$ is Brauer equivalent to $\mathcal{H}_{\mathbb{Q}}(a, b^2 - c^2 d)$, hence $\mathcal{H}_{\mathbb{Q}}(a, b^2 - c^2 d)$ splits. This proves the existence of a suitable element r. □

Proposition 154 also implies that Algorithm 3 can be used do decide if $\mathcal{H}$ contains a quaternion subalgebra over $\mathbb{Q}$ or not.

The reason that these statements are located in this chapter and not in the previous one, is that it was discovered while computing with the implementation of Algorithm 3. The same holds for the next example:

Example 156. Let $\mathcal{H} = \mathcal{H}_{\mathbb{Q}(\sqrt{5})}(7, 11)$. One can check that $\mathcal{H}$ is a division algebra. Clearly it contains the rational quaternion algebra $\mathcal{H}_{\mathbb{Q}}(7, 11)$. Let $1, u, v, uv$ be a quaternion basis of $\mathcal{H}$ for which $u^2 = 7$ and $v^2 = 11$. Let $s = (11 + \sqrt{5})u + (1 + \sqrt{5})uv$. Then $s^2 = 420$. However, $\mathcal{H}_{\mathbb{Q}}(7, 11)$ is not split by $\mathbb{Q}(\sqrt{420})$ (this can be verified using the MAGMA function "IsMatrixRing") hence it does not contain an element whose square is 420. Proposition 154 then implies that $\mathcal{H}$ contains two non-isomorphic rational quaternion subalgebras.

We conclude by a table containing running times of Algorithm 3 (Table 6.2). Each row consists of the following data: d, which defines the quadratic field, the number of digits (decimal digits) of a_1, a_2, a_3, a_4, which define the quaternion algebra $\mathcal{H}_{\mathbb{Q}(\sqrt{d})}(a_1 + a_2\sqrt{d}, a_3 + a_4\sqrt{d})$ and the running time of the algorithm in seconds. As a random quaternion algebra is almost certainly a division algebra, the parameters are not chosen at random. We chose a split quaternion algebra and computed a different quaternion basis of that algebra. So in all cases the quaternion algebra splits.

CHAPTER 7

PROBLEMS FOR FURTHER RESEARCH

We would like to conclude this dissertation with some open problems which naturally arose during our research.

Problem 2. *Is the statement of Theorem 96 valid for finite extensions K of $\mathbb{F}_q(t)$. More precisely, is it true that the intersection of a maximal $\mathbb{F}_q[t]$ order and a maximal R-order (here R consists of those elements of K whose degree is at most 0) is a finite dimensional $\mathbb{F}_q$-algebra which contains a primitive idempotent? Finiteness is true, this follows from lattice reduction techniques. However, it is not clear whether it contains any zero divisors at all. If this would be true (it is known to be true in certain special cases, when the ring of integers of K is a unique factorization domain) in general, then one could derive an algorithm for general function fields as well. Moreover, this statement is also of pure theoretical interest.*

Problem 3. *In Chapter 3 we introduced $\mathbb{F}_q[t]$-lattices in $\mathbb{F}_q((\frac{1}{t}))^m$ and proposed an algorithm for finding a reduced basis. We gave an application of this algorithm which finds a lattice point inside a parallelepiped. In 2012, Chonoles [9] proved a function field analogue of Minkowski's convex body theorem. It roughly says that if a set is closed under subtraction and has large enough volume (larger than the determinant of the lattice) than it contains a lattice point. Chonoles's proof is not effective. It would be interesting to provide an algorithm for finding such a lattice point in certain different cases. Another alternative improvement of our result could be to consider parallelepipeds not to be given by generators but by some separation oracle.*

Problem 4. *Give an algorithm for finding nontrivial zeros of quadratic forms in several variables over quadratic extensions of $\mathbb{Q}$ or $\mathbb{F}_q(t)$. The first step in this direction is naturally the ternary case which is resolved in Chapter 5. A similar approach to that of Chapter 4 could be applied, however, in the case of quadratic extensions, unique factorization is no longer necessarily true. On the other hand, the local-global principle*

is still valid. Also note that resolving this question would immediately solve the explicit isomorphism problem for quaternion algebras over degree 4 extensions of $\mathbb{Q}$ or $\mathbb{F}_q(t)$ applying the procedure from Chapter 5.

Problem 5. *The algorithms from Chapter 4 are probabilistic. Is there an f-algorithm (which is by definition deterministic but uses oracles for factoring polynomials over finite fields) which finds nontrivial zeros of quadratic forms in several variables. Such an algorithm exists for ternary quadratic forms ([11], [38]).*

Problem 6. *Assume that K is a global field of characteristic different from 2 (i.e., a finite extension of $\mathbb{Q}$ or $\mathbb{F}_q(t)$, where q is odd). Let $\mathcal{H}$ be a quaternion algebra over K. Find a quaternion subalgebra of $\mathcal{H}$ over $\mathbb{Q}$ or $\mathbb{F}_q(t)$. This is an easier problem than the general explicit isomorphism problem for global fields. However, as in the quadratic case, this could be a first step. As indicated in Chapter 5, this may be achieved by generalizing the algorithm from Chapter 5. Also, a slightly more conceptual version of Algorithm 3 could be of independent interest (by more conceptual we mean an algorithm whose validity does not require much calculation). It is even interesting how one can find an element in $\mathcal{H}$ whose square is rational (or at least its minimal polynomial has rational coefficients).*

Problem 7. *Generalize our algorithms to finding nontrivial zeros of quadratic forms in characteristic 2. The algorithm from Chapter 3 solves the case of ternary equations. Also the explicit isomorphism problem for quadratic extensions of $\mathbb{F}_q(t)$, where q is now even, is interesting.*

These problems are all somewhat related to solving the explicit isomorphism problem for global fields. It seems that global fields of positive characteristics are easier to handle than number fields. However, there might exist a universal approach. Stating a conjecture is probably too early at this stage but we believe that the function field case could be resolved by a polynomial time algorithm. In the case of number fields, we believe that there exist ff-algorithms which run in polynomial time in the degree and discriminant of the number field. Note that we left out a parameter , namely the dimension of the matrix algebra. We believe that finding an algorithm which is polynomial in the dimension of the matrix algebra should be considerably more difficult (probably even intractable). However, these are only beliefs of the author, which do not yet have a solid foundation.

BIBLIOGRAPHY

[1] M. Ajtai: The shortest vector problem in L_2 is NP-hard for randomized reductions; Proceedings of the 30th annual ACM symposium on Theory of computing (1998). Dallas, Texas, United States: ACM. pp. 10-19.

[2] A. Bérczes, L. Hajdu, A. Pethő: Arithmetic progressions in the solution sets of norm form equations; Rocky Mountain Math. Journal 40 (2010), pp. 383-396.

[3] A. Bérczes, J. Ködmön, A. Pethő: A one-way function based on norm equations, Periodica Mathematica Hungarica 49 (2004), pp.1-13.

[4] E.R. Berlekamp: Factoring polynomials over finite fields; Bell System Technical Journal 46 (1967), pp. 1853-1859.

[5] J. Buchmann: Reducing lattice bases by means of approximations, in: Algorithmic number theory, LNCS 877, Springer-Verlag (1994), pp. 160-168.

[6] Y. Bugeaud, K. Győry: Bounds for the solutions of Thue–Mahler equations and norm form equations; Acta Arithmetica 74 (1996), pp. 273-292.

[7] D.G. Cantor, H. Zassenhaus: A new algorithm for factoring polynomials over finite fields; Mathematics of Computation 36 (1981), pp. 587-592.

[8] P. Castel: Un algorithme de résolution des équations quadratiques en dimension 5 sans factorisation; Phd thesis, Université de Caen, 2011.

[9] Z. Chonoles: Hermite's theorem for function fields, Honors Thesis, Brown University, 2012.

[10] A. M. Cohen, G. Ivanyos, D. B. Wales: Finding the radical of an algebra of linear transformations; Journal of Pure and Applied Algebra 117-118 (1997), pp. 177-193.

[11] J.E. Cremona, T.A. Fisher, C. O'neill, D. Simon, M. Stoll: Explicit n-descent on elliptic curves I. Algebra; Journal für die reine und angewandte Mathematik 615 (2008), pp. 121-155.

[12] J.E. Cremona, T.A. Fisher, C. O'neill, D. Simon, M. Stoll: Explicit n-descent on elliptic curves II. Geometry; Journal für die reine und angewandte Mathematik 632 (2009), pp. 63–84.

[13] J.E. Cremona, T.A. Fisher, C. O'neill, D. Simon, M. Stoll: Explicit n-descent on elliptic curves III. Algorithms; Mathematics of Computation 84 No. 292 (2015), pp. 895-922.

[14] J. Cremona, M. van Hoeij: Solving conics over function fields; Journal de Théorie des Nombres de Bordeaux 18(3) (2006), pp. 595-606.

[15] J.E. Cremona, D. Rusin: Efficient solution of rational conics, Mathematics of Computation, 72 (2003), pp. 1417-1441.

[16] P. K. Draxl: Skew Fields; Cambridge University Press, 1983.

[17] Y. Drozd, V.V. Kirichenko: Finite dimensional algebras; Vyshcha Shkola, Kiev, 1980.

[18] W. M. Eberly: Decompositions of algebras over $\mathbb{R}$ and $\mathbb{C}$; Computational Complexity 1(1991), pp. 207-230.

[19] G. W. Effinger, D. R. Hayes: Additive Number Theory of Polynomials over a Finite Field; Oxford Science Publications, 1991.

[20] C. Fieker, A. Jurk, M. Pohst: On solving relative norm equations in algebraic number fields, Mathematics of Computation 66 (1997), pp. 399-410.

[21] U. Fincke: Ein Ellipsoidverfahren zur Lösung von Normgleichungen in algebraischen Zahlkörpern, Phd Thesis, Düsseldorf, 1984.

[22] K. Friedl, L. Rónyai: Polynomial time solutions of some problems in computational algebra; Proceedings of the 17th annual ACM symposium on Theory of computing (1985). Providence, Rhode Island, United States: ACM. pp. 153-162.

[23] I. Gaál, M. Pohst: On solving norm equations in global function fields; Journal of Mathematical Cryptology 3 (2009), pp. 237-248.

[24] L.J. Gerstein: Basic quadratic forms; Graduate Studies in Mathematics, vol. 90, American Mathematical Society, Providence, RI, 2008.

[25] M. Giesbrecht, Y. Zhang: Factoring and decomposing Ore polynomials over $\mathbb{F}_q(T)$; Proceedings of the 2003 International Symposium on Symbolic and Algebraic Computation (ISSAC2003), New York, NY, USA: ACM. pp. 127-134.

[26] J. Gómez-Torrecillas, F. J. Lobillo, G. Navarro: Factoring Ore polynomials over $\mathbb{F}_q(t)$ is difficult; (2015) Preprint arXiv:1505.07252.

[27] J. Gómez-Torrecillas, F. J. Lobillo, G. Navarro: A New Perspective of Cyclicity in Convolutional Codes; IEEE Transactions on Information Theory 62 (2016), pp. 2702-2706.

[28] W. A. de Graaf, M. Harrison, J. Pílnikova, J. Schicho: A Lie algebra method for rational parametrization of Severi-Brauer surfaces; Journal of Algebra 303(2006), pp. 514-529.

[29] W. A. de Graaf, G. Ivanyos: Finding maximal tori and splitting elements in matrix algebras; Interaction between Ring Theory and Representations of Algebras, Lecture Notes in Pure and Applied Mathematics 210 (1998), pp. 95-105.

[30] W. A. de Graaf, G. Ivanyos, A. Küronya, L. Rónyai: Computing Levi decompositions; Applicable Algebra in Engineering, Communication and Computing 8 (1997), pp. 291-304.

[31] G. Ivanyos: Algorithms for algebras over global field; Ph. D. thesis, Hungarian Academy of Sciences 1996.

[32] G. Ivanyos, M. Karpinski, L. Rónyai, N. Saxena: Trading GRH for algebra: algorithms for factoring polynomials and related structures; Mathematics of Computation 81 (2012), pp. 493-531.

[33] G. Ivanyos, L. Rónyai, J. Schicho: Splitting full matrix algebras over algebraic number fields; Journal of Algebra 354 (2012), pp. 211-223.

[34] G. Ivanyos, L. Rónyai: On the complexity of finding maximal orders in semisimple algebras over $\mathbb{Q}$; Computational Complexity 3 (1993), pp. 245-261.

[35] G. Ivanyos, Á. Lelkes, L. Rónyai: Improved algorithms for splitting full matrix algebras; JP Journal of Algebra, Number Theory and Applications 28 (2013), pp. 141-156.

[36] G. Ivanyos, L. Rónyai, Á. Szántó: Decomposition of algebras over $\mathbb{F}_q(x_1, ..., x_m)$; Applicable Algebra in Engineering, Communication and Computing 5 (1994), pp. 71-90.

[37] G. Ivanyos, Á. Szántó: Lattice basis reduction for indefinite forms and an application; Discrete Mathemathics 153 (1996), pp. 177-188.

[38] G. Ivanyos, P. Kutas, L. Rónyai: Computing explicit isomorphisms with full matrix algebras over $\mathbb{F}_q(x)$; Foundations of Computational Mathematics, accepted (doi:10.1007/s10208-017-9343-2).

[39] G. Ivanyos, P. Kutas, L. Rónyai: Explicit equivalence of quadratic forms over $\mathbb{F}_q(t)$; (2016) Preprint arXiv: arXiv:1610.08671.

[40] M-A. Knus, A. Merkurjev, M. Rost, J-P. Tignol: The book of involutions; American Mathematical Society Colloquium Publications, 44. American Mathematical Society, Providence, RI, 1998.

[41] Heinrich Kornblum: Über die Primfunktionen in einer arithmetischen Progression; Mathematische Zeitschrift 5 (1919), pp. 100-111.

[42] P. Kutas: Splitting quaternion algebras over quadratic number fields; (2016) Preprint: arXiv:1606.01053.

[43] P. Kutas: Some Results Concerning the Explicit Isomorphism Problem over Number Fields; International Conference on Mathematical Aspects of Computer and Information Sciences, Springer International Publishing (2015), pp. 143-148.

[44] A.K. Lenstra: Factoring multivariate polynomials over finite fields; Journal of Computer and System Sciences 30 (2) (1985), pp. 235-248.

[45] T. Y. Lam: Introduction to quadratic forms over fields; Graduate Studies in Mathematics, vol. 67, American Mathematical Society, Providence, RI, 2005.

[46] A. K. Lenstra, H. W. Lenstra, L. Lovász, Factoring polynomials with rational coefficients, Mathematische Annalen 261 (1982), pp. 515–534.

[47] http://magma.maths.usyd.edu.au/magma/handbook/text/840

[48] J. Neukirch: Algebraic Number Theory; Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], 322. Springer-Verlag, Berlin, 1999.

[49] S. Paulus: Lattice basis reduction in function fields; Proceedings of the Third Symposium on Algorithmic Number Theory, Portland, Oregon, United States: ANTS-III, Springer LNCS 1423 (1998), pp. 567-575.

[50] R. S. Pierce, Associative algebras, Springer-Verlag, 1982.

[51] J. Pílniková: Trivializing a central simple algebra of degree 4 over the rational numbers, Journal of Symbolic Computation 42 (2007), pp. 579-586.

[52] H. Rauter:Über die Darstellbarkeit durch quadratische Formen im Körper der rationalen Funktionen einer Unbestimmten über dem Restklassenkörper mod p; Phd thesis, Halle, 1926.

[53] I. Reiner: Maximal orders; Academic Press, 1975.

[54] G. Rhin: Répartition modulo 1 dans un corps de séries formelles sur un corps fini; Dissertationes Mathematicae (Rozprawy Matematyczne), No. 95., Mathematical Institute of the Polish Academy of Aciences, 1972.

[55] J. C. Robson: A unified approach to unity; Communications in Algebra 7 (1979), pp. 1245-1255.

[56] L. Rónyai: Simple algebras are difficult; Proc. of the 19th Annual ACM Symposium on the Theory of Computing, New York (1987), pp. 398-408.

[57] L. Rónyai: Computing the structure of finite algebras; Journal of Symbolic Computation 9 (1990), pp. 355-373.

[58] L. Rónyai: Zero Divisors in Quaternion Algebras; Journal of Algorithms 9 (1988), pp. 494-506.

[59] L. Rónyai: Algorithmic properties of maximal orders in simple algebras over $\mathbb{Q}$; Computational Complexity 2 (1992), pp. 225-243.

[60] M. Rosen: Number Theory in Function Fields, Springer-Verlag, New York, 2002.

[61] A. Schönhage: The fundamental theorem of algebra in terms of computational complexity; Preliminary report, Universität Tübingen, 1982.

[62] Daniel Shanks: Five Number Theoretic Algorithms; Proceedings of the Second Manitoba Conference on Numerical Mathematics 51 (1973), pp. 51-70.

[63] D. Simon: Quadratic equations in dimensions 4, 5 and more; preprint (2005).

[64] D. Simon: Solving norm equations in relative number fields using S-units; Mathematics of Computation 71 No. 239 (2002), pp. 1287-1305.

[65] J. H. Silverman: The Arithmetic of Elliptic Curves; Springer-Verlag, 2009.

[66] M-F. Vignéras: Arithmétique des Algèbres de Quaternions; Springer-Verlag, 1980.

[67] J. Voight: Identifying the matrix ring: algorithms for quaternion algebras and quadratic forms; Quadratic and higher degree forms, Springer, New York (2013), pp. 255-298.

[68] D. Wan: Generators and irreducible polynomials over finite fields; Mathematics of Computation 66 No. 219 (1997), pp. 1195-1212.

[69] C. van de Woestijne: Deterministic equation solving over finite fields; Phd Thesis Universiteit Leiden, 2006.

[70] D. Y. Y. Yun: On square-free decomposition algorithms; Proceedings of the third ACM symposium on Symbolic and Algebraic Computation, ACM (1976), pp. 26-35.

Program code for Algorithm 1

```
issquare:=function(f);
F:=FiniteField(3);
P<x>:=PolynomialRing(F);
L:=Factorisation(f);
s:=0;
if (IsSquare(LeadingCoefficient(f)) eq false) then
return 0;
end if;
if (Degree(f) ne 0) then
for i:=1 to #L do
if (L[i][2] mod 2 eq 1) then
s:=1;
end if;
end for;
if (s eq 1) then return 0;
else return 1;
end if;
end if;
if (Degree(f) eq 0) then
if (IsSquare(LeadingCoefficient(f))) then return 1;
else return 0;
end if;
end if;
end function;


squarefree:=function(c);
F:=FiniteField(3);
P<x>:=PolynomialRing(F);
```

```
Q:=Factorization(c);
S:=[];
for i:=1 to #Q do
S[i]:=Q[i][2];
end for;
T:=[];
for i:=1 to #S do
T[i]:=S[i] mod 2;
end for;
U:=[];
for i:=1 to #Q do
U[i]:=Q[i][1];
end for;
M:=[];
for i:=1 to #Q do
M[i]:=<U[i],T[i]>;
end for;
return Facpol(M);
end function;
F:=FiniteField(3);
P<x>:=PolynomialRing(F);

minimization:=function(a1,a2,a3,a4);
F:=FiniteField(3);
P<x>:=PolynomialRing(F);
b1:=squarefree(a1);
b2:=squarefree(a2);
b3:=squarefree(a3);
b4:=squarefree(a4);
S:=[P|];
B:=[];
B[1]:=b1;
B[2]:=b2;
B[3]:=b3;
B[4]:=b4;
L1:=[];
L1:=Factorisation(b1*b2*b3*b4);
for i:=1 to #L1 do
if (L1[i][2] eq 2) then
if (b1 mod L1[i][1] eq 0) then S[1]:=1;
```

```
else
S[1]:=0;
end if;
if (b2 mod L1[i][1] eq 0) then S[2]:=1;
else
S[2]:=0;
end if;
if (b3 mod L1[i][1] eq 0) then S[3]:=1;
else
S[3]:=0;
end if;
if (b4 mod L1[i][1] eq 0) then S[4]:=1;
else
S[4]:=0;
end if;
k:=0;
s:=1;
while (k eq 0) do
if (S[s] eq 0) then s:=s+1;
else k:=1;
end if;
end while;
s2:=s+1;
while (k eq 1) do
if (S[s2] eq 0) then s2:=s2+1;
else k:=2;
end if;
end while;
l:=0;
r:=1;
while (l eq 0) do
if (S[r] eq 1) then r:=r+1;
else l:=1;
end if;
end while;
r2:=r+1;
while (l eq 1) do
if (S[r2] eq 1) then r2:=r2+1;
else l:=2;
end if;
```

```
end while;

baj:=0;
if (JacobiSymbol(-B[r]*B[r2],L1[i][1]) eq -1) then baj:=1;
end if;
if (baj eq 1) then
f:=-B[s]*B[s2] div L1[i][1]^2;
if (JacobiSymbol(f,L1[i][1]) eq -1) then baj:=2;
end if;
end if;
LL:=[P|];
LL:=[0,0,0,0];
if (baj eq 2) then return LL;
end if;
if (baj eq 1) then
B[r]:=L1[i][1]*B[r];
B[r2]:=L1[i][1]*B[r2];
B[s]:=B[s] div L1[i][1];
B[s2]:=B[s2] div L1[i][1];
end if;
end if;
if (L1[i][2] eq 3) then
S2:=[P|];
if (b1 mod L1[i][1] eq 0) then S2[1]:=1;
else
S2[1]:=0;
end if;
if (b2 mod L1[i][1] eq 0) then S2[2]:=1;
else
S2[2]:=0;
end if;
if (b3 mod L1[i][1] eq 0) then S2[3]:=1;
else
S[3]:=0;
end if;
if (b4 mod L1[i][1] eq 0) then S2[4]:=1;
else
S2[4]:=0;
end if;
seged:=0;
```

```
k2:=1;
while (seged eq 0) do
if (S2[k2] eq 1) then k2:=k2+1;
else seged:=1;
end if;
end while;
for j:=1 to 4 do
if (S2[j] eq 1) then B[j]:=B[j] div L1[i][1];
else B[j]:=B[j]*L1[i][1];
end if;
end for;
end if;
if (L1[i][2] eq 4) then
for j2:=1 to 4 do
B[j2]:=B[j2] div L1[i][1];
end for;
end if;
end for;
return B;
end function;

infinity:=function(a1,a2,a3,a4);
F:=FiniteField(3);
P<x>:=PolynomialRing(F);
A:=[];
A[1]:=a1;
A[2]:=a2;
A[3]:=a3;
A[4]:=a4;
S:=[];
for i:=1 to 4 do
S[i]:=LeadingTerm(A[i]);
end for;
k:=0;
for j:=1 to 4 do
if (Degree(S[j]) mod 2 eq 0) then
k:=k+1;
end if;
end for;
if (k eq 2) then
```

```
r:=0;
l:=1;
while (r eq 0) do
if (Degree(S[l]) mod 2 eq 0) then
r:=l;
else l:=l+1;
end if;
end while;
r2:=0;
l2:=l+1;
while (r2 eq 0) do
if (Degree(S[l2]) mod 2 eq 0) then
r2:=l2;
else l2:=l2+1;
end if;
end while;
s:=0;
m:=1;
while (s eq 0) do
if (Degree(S[m]) mod 2 eq 1) then
s:=m;
else m:=m+1;
end if;
end while;
s2:=0;
m2:=m+1;
while (s2 eq 0) do
if (Degree(S[m2]) mod 2 eq 1) then
s2:=m2;
else m2:=m2+1;
end if;
end while;

baj:=0;
C:=[];
for i2:=1 to 4 do
C[i2]:=LeadingCoefficient(A[i2]);
end for;
if (IsSquare(-C[r]*C[r2])) then return 1;
else baj:=baj+1;
```

```
end if;
if (IsSquare(-C[s]*C[s2])) then return 1;
else baj:=baj+1;
end if;
if (baj eq 2) then return 0;
end if;
else return 1;
end if;
end function;

check:=function(a1,a2,a3,a4);
F:=FiniteField(3);
P<x>:=PolynomialRing(F);
A:=[];
A[5]:=x;
A[1]:=a1;
A[2]:=a2;
A[3]:=a3;
A[4]:=a4;
s:=0;
r:=0;
for i:=1 to 3 do
for j:=i+1 to 4 do
m:=-A[i]*A[j];
if (issquare(m) eq 1) then
s:=i;
r:=j;
return r,s,m,-A[r]*A[s];
end if;
end for;
end for;
if (s eq 0) then
return 0;
end if;
end function;

splitting:=function(a1,a2,a3,a4);
F:=FiniteField(3);
P<x>:=PolynomialRing(F);
B:=[P|];
```

```
A:=[P|];
S:=[P|];
A[1]:=a1;
A[2]:=a2;
A[3]:=a3;
A[4]:=a4;
s:=0;
for i:=1 to 4 do
if (Degree(A[i]) mod 2 eq 1) then
s:=s+1;
end if;
end for;
case s:
when 0:
B[1]:=A[1];
B[2]:=A[2];
B[3]:=-A[3];
B[4]:=-A[4];
B[5]:=0;
B[6]:=1;
when 1:
h:=0;
j:=1;
while (h eq 0) do
if (Degree(A[j]) mod 2 eq 1) then h:=j;
else j:=j+1;
end if;
end while;


S[1]:=0;
k1:=1;
while (S[1] eq 0) do
if ((Degree(A[k1]) mod 2 eq 0)) then S[1]:=A[k1];
else k1:=k1+1;
end if;
end while;
S[2]:=0;
k2:=k1+1;
while (S[2] eq 0) do
```

```
if ((Degree(A[k2]) mod 2 eq 0)) then S[2]:=A[k2];
else k2:=k2+1;
end if;
end while;
S[3]:=0;
k3:=k2+1;
while (S[3] eq 0) do
if ((Degree(A[k3]) mod 2 eq 0)) then S[3]:=A[k3];
else k3:=k3+1;
end if;
end while;
B[1]:=A[h];
B[2]:=S[1];
B[3]:=-S[2];
B[4]:=-S[3];
B[5]:=0;
B[6]:=LeadingCoefficient(S[1]);
when 2:
L1:=[];
L2:=[];
l1:=1;
l2:=1;
for l:=1 to 4 do
if (Degree(A[l]) mod 2 eq 0 ) then
L1[l1]:=A[l];
l1:=l1+1;
else
L2[l2]:=A[l];
l2:=l2+1;
end if;
end for;
B[1]:=L1[1];
B[2]:=L2[1];
B[3]:=-L1[2];
B[4]:=-L2[2];
C:=[];
for m:=1 to 4 do
C[m]:=LeadingCoefficient(B[m]);
end for;
if (IsSquare(C[1]*C[3])) then
```

```
B[5]:=0;
B[6]:=C[1];
else
B[5]:=1;
B[6]:=C[2];
end if;
when 3:
h2:=0;
j2:=1;
while (h2 eq 0) do
if (Degree(A[j2]) mod 2 eq 0) then h2:=j2;
else j2:=j2+1;
end if;
end while;

T:=[P|];
T[1]:=0;
k12:=1;
while (T[1] eq 0) do
if ((Degree(A[k12]) mod 2 eq 1)) then T[1]:=A[k12];
else k12:=k12+1;
end if;
end while;
T[2]:=0;
k22:=k12+1;
while (T[2] eq 0) do
if ((Degree(A[k22]) mod 2 eq 1)) then T[2]:=A[k22];
else k22:=k22+1;
end if;
end while;
T[3]:=0;
k32:=k22+1;
while (T[3] eq 0) do
if ((Degree(A[k32]) mod 2 eq 1)) then T[3]:=A[k32];
else k32:=k32+1;
end if;
end while;
B[1]:=A[h2];
B[2]:=T[1];
B[3]:=-T[2];
```

```
B[4]:=-T[3];
B[5]:=1;
B[6]:=LeadingCoefficient(T[1]);
when 4:
B[1]:=A[1];
B[2]:=A[2];
B[3]:=-A[3];
B[4]:=-A[4];
B[5]:=1;
B[6]:=1;
end case;
return B;
end function;

randompoli:=function(d);
F:=FiniteField(3);
P<x>:=PolynomialRing(F);
L:=[];
for i:= 1 to d do
L[i]:=Random(F);
end for;
return Polynomial(L);
end function;

solving:=function(a1,a2,a3,a4);
F:=FiniteField(3);
P<x>:=PolynomialRing(F);
A:=splitting(a1,a2,a3,a4);
gcd1:=GreatestCommonDivisor(A[1],A[2]);
gcd2:=GreatestCommonDivisor(A[3],A[4]);
L1:=Factorization(gcd1);
L2:=Factorization(gcd2);
M1:=[];
j:=1;
for i:=1 to #L1 do
M1[i]:=L1[i][1];
end for;
M2:=[];
for i2:=1 to #L2 do
M2[i2]:=L2[i2][1];
```

```
end for;

a12:=A[1] div gcd1;
a22:=A[2] div gcd1;
A12:=Factorisation(a12);
A22:=Factorisation(a22);

a32:=A[3] div gcd2;
a42:=A[4] div gcd2;
A32:=Factorisation(a32);
A42:=Factorisation(a42);

Prod1:=Identity(P);
for s:=1 to #M1 do
Prod1:=Prod1*M1[s];
end for;
Prod2:=Identity(P);
for s2:=1 to #M2 do
Prod2:=Prod2*M2[s2];
end for;
Prod:=Prod1*Prod2;
I1:=[];
for o1:=1 to #A12 do
I1[o1]:=<A12[o1][1],JacobiSymbol(A[2]*Prod,A12[o1][1])>;
end for;

I2:=[];
for o2:=1 to #A22 do
I2[o2]:=<A22[o2][1],JacobiSymbol(A[1]*Prod,A22[o2][1])>;
end for;

I3:=[];
for o3:=1 to #A32 do
I3[o3]:=<A32[o3][1],JacobiSymbol(A[4]*Prod,A32[o3][1])>;
end for;

I4:=[];
for o4:=1 to #A42 do
I4[o4]:=<A42[o4][1],JacobiSymbol(A[3]*Prod,A42[o4][1])>;
end for;
```

```
I:=I1 cat I2 cat I3 cat I4;

J:=[CartesianProduct(P,P)|];
J[1]:=<0,0>;
for p:=1 to #I do
if (I[p][2] eq 1) then
J[p]:=<I[p][1],1>;
else
seged:=0;
while (seged eq 0) do
g:=randompoli(Degree(I[p][1]));
if (JacobiSymbol(g,I[p][1]) eq -1) then
seged:=1;
J[p]:=<I[p][1],g>;
end if;
end while;
end if;
end for;
J1:=[];
J2:=[];
for w:=1 to #J do
J1[w]:=J[w][2];
J2[w]:=J[w][1];
end for;
for p1:=1 to #J2 do
for p2:=p1+1 to #J2 do
if (J2[p1] eq J2[p2]) then
J2[p2]:=1;
end if;
end for;
end for;
for p3:=1 to #J2 do
if (J2[p3] eq 1) then
Remove(J2,p3);
Remove(J1,p3);
end if;
end for;

remainder:=ChineseRemainderTheorem(J1,J2);
D:=1;
```

```
for w2:=1 to #J2 do
D:=D*J2[w2];
end for;

ind:=Degree(Prod) mod 2;
if (A[5] eq ind) then
dd:= 3*Degree(D);
else dd:=3*Degree(D)+1;
end if;
Q:=[];
Q[dd+1]:=A[6];
repeat
for v:=1 to dd do
Q[v]:=Random(F);
end for;
q:=P! Q;
a:=D*q+remainder;
until IsIrreducible(a) eq true;

K<x>:=FieldOfFractions(P);
P2<x1,x2,x3>:=ProjectiveSpace(K,2);
ff1:=Conic(P2,A[1]*x1^2+A[2]*x2^2-a*Prod*x3^2);
ff2:=Conic(P2,A[3]*x1^2+A[4]*x2^2-a*Prod*x3^2);
CC1:=Coordinates(RationalPoint(ff1));
CC2:=Coordinates(RationalPoint(ff2));
zero:=A[1]*CC1[1]^2+A[2]*CC1[2]^2-A[3]*CC2[1]^2-A[4]*CC2[2]^2;
AA:=[A[1],A[2],-A[3],-A[4]];
return zero,AA,CC1[1],CC1[2],CC2[1],CC2[2];
end function;

main:=function(a1,a2,a3,a4);
if (minimization(a1,a2,a3,a4) eq [0,0,0,0]) then return "no solutions";
else
A:=minimization(a1,a2,a3,a4);
end if;
if (infinity(A[1],A[2],A[3],A[4]) eq 0) then return "no solutions";
else
if (check(A[1],A[2],A[3],A[4]) eq 0) then
return solving(A[1],A[2],A[3],A[4]);
else
```

```
return check(A[1],A[2],A[3],A[4]);
end if;
end if;
return A;
end function;

main2:=function(a,c);
s:=0;
for i:=1 to c do
a1:=randompoli(a);
a2:=randompoli(a);
a3:=randompoli(a);
a4:=randompoli(a);
if (minimization(a1,a2,a3,a4) eq [0,0,0,0]) then s:=s+1;
else
A:=minimization(a1,a2,a3,a4);
end if;
if (infinity(A[1],A[2],A[3],A[4]) eq 0) then s:=s+1;
end if;
end for;
return s;
end function;
```

Program code for Algorithm 3

```
antikomm:=function(D,u,v,w);
Q:=Rationals();
K<z>:=QuadraticField(D);
A<i,j>:=QuaternionAlgebra<K| u,v+w*z>;
P:=PolynomialRing(Q,4);
a:=P.1;
b:=P.2;
c:=P.3;
d:=P.4;
g:=w*a^2+w*D*b^2+2*v*a*b-u*w*c^2-u*w*D*d^2-2*u*v*c*d;
V:=IsotropicSubspace(g);
H:=Generators(V);
I:=SetToSequence(H);
r:=Dimension(V);
if (r ne 0) then
s:=Eltseq(I[1]);
q:=[];
q[1]:=s[1]+s[2]*z;
q[2]:=s[3]+s[4]*z;
return q;
else
q2:=[];
q2[1]:=0;
q2[2]:=0;
return q2;
end if;
end function;
```

```
zerodiv:=function(a,b);
Q:=RationalField();
H<i,j,k>:=QuaternionAlgebra<Q| a,b>;
P:=PolynomialRing(Q,3);
x:=P.1;
y:=P.2;
z:=P.3;
g:=a*x^2+b*y^2-a*b*z^2;
V:=IsotropicSubspace(g);
if (Dimension(V) ne 0) then
H:=Generators(V);
I:=SetToSequence(H);
s:=Eltseq(I[1]);
r:=s[1]*i+s[2]*j+s[3]*k;
return s;
else
s2:=[];
s2[1]:=0;
s2[2]:=0;
s2[3]:=0;
return s2;
end if;
end function;

bazis:=function(D,u,v,w,t,a,b,c);
Q:=Rationals();
K<z>:=QuadraticField(D);
A<i,j,k>:=QuaternionAlgebra< K | u+v*z,w+t*z>;
r:=a*i+b*j+c*k;
S:=Coordinates(r*i+i*r);
T:=Coordinates(r*j+j*r);
R:=Coordinates(r*k+k*r);
M:=[];
M[1]:=Vector(S);
M[2]:=Vector(T);
M[3]:=Vector(R);
N:=Matrix(M);
U:=NullSpace(N);
V:=Basis(U);
P:=Eltseq(V[1]);
```

```
r1:=P[1]*i+P[2]*j+P[3]*k;
return P;
end function;


rational:=function(D,u,v,w,t);
Q:=Rationals();
K<z>:=QuadraticField(D);
A<i,j,k>:=QuaternionAlgebra< K | u+v*z,w+t*z>;
if (IsMatrixRing(A) eq false) then return "division algebra";
else
P:=PolynomialRing(Q,6);
a:=P.1;
b:=P.2;
c:=P.3;
d:=P.4;
e:=P.5;
f:=P.6;
g:=v*a^2+v*D*b^2+2*u*a*b+t*c^2+t*D*d^2+2*w*c*d-u*t*e^2-v*w*e^2-u*t*D*f
V:=IsotropicSubspace(g);
H:=Generators(V);
I:=SetToSequence(H);
s:=Eltseq(I[1]);
q:=(s[1]+s[2]*z)*i+(s[3]+s[4]*z)*j+(s[5]+s[6]*z)*i*j;
if (q^2 eq 0) then
return q;
else
S:=Coordinates(q);
R:=bazis(D,u,v,w,t,s[1]+s[2]*z,s[3]+s[4]*z,s[5]+s[6]*z);
W:=R[1]*i+R[2]*j+R[3]*k;
T:=[];
T[1]:=Trace(q^2)/2;
T[2]:=Trace(W^2)/2;
O:=[];
O[1]:=Trace(T[1],RationalField())/2;
O[2]:=Trace((T[2]+Conjugate(T[2]))/2,RationalField())/2;
O[3]:=Trace(1/z*(T[2]-Conjugate(T[2]))/2,RationalField())/2;
seged:=antikomm(D,O[1],O[2],O[3]);
segedlista:=[];
segedlista[1]:=0;
```

```
segedlista[2]:=0;
if (seged eq segedlista) then
return "division algebra";
end if;
q2:=seged[1]*W+seged[2]*q*W;
if (q2^2 eq 0) then
return q2;
else
szam1:=T[1];
szam2:=Trace(q2^2)/2;
szam11:=Trace((szam1)/2,RationalField());
szam22:=Trace((szam2)/2,RationalField());
segedlista2:=zerodiv(szam11,szam22);
nullista:=[];
nullista[1]:=0;
nullista[2]:=0;
nullista[3]:=0;
if (segedlista2 ne nullista) then
return segedlista2[1]*q+segedlista2[2]*q2+segedlista2[3]*q*q2, "nulloszto
end if;
P2:=PolynomialRing(Q,4);
a1:=P2.1;
b1:=P2.2;
c1:=P2.3;
d1:=P2.4;
pol:=szam11*a1^2+szam22*b1^2-szam11*szam22*c1^2-D*d1^2;
V2:=IsotropicSubspace(pol);
if (Dimension(V2) eq 0) then
return "division algebra";
end if;
H2:=Generators(V2);
I2:=SetToSequence(H2);
S2:=Eltseq(I2[1]);
r2:=(S2[1]/S2[4])*q+(S2[2]/S2[4])*q2+(S2[3]/S2[4])*q*q2-z;
return r2;
end if;
end if;
end if;
end function;
```