Research in Number Theory

RESEARCH

Explicit isomorphisms of quaternion algebras over quadratic global fields

Tímea Csahók[1], Péter Kutas[2*], Mickaël Montessinos[3] and Gergely Zábrádi[4]

*Correspondence:
kutasp@gmail.com
Eötvös Loránd University and University of Birmingham, Birmingham, UK
Full list of author information is available at the end of the article

Abstract

Let L be a separable quadratic extension of either $\mathbb{Q}$ or $\mathbb{F}_q(t)$. We exhibit efficient algorithms for finding isomorphisms between quaternion algebras over L. Our techniques are based on computing maximal one-sided ideals of the corestriction of a central simple L-algebra.

1 Introduction

In this paper we consider a special case of the following algorithmic problem. Let K be a global field and let A and B be central simple algebras over K given by a K-basis and a multiplication table of the basis elements. The task is to decide whether A and B are isomorphic, and if so, find an explicit isomorphism between them. A special case of this problem when $B = \mathcal{M}_n(K)$ is referred to as the *explicit isomorphism problem* which has various applications in arithmetic geometry [4,10,12], computational algebraic geometry [7] and coding theory [18,19].

In 2012, Ivanyos et al. [24] exhibited an algorithm for the explicit isomorphism problem in the case where K is an algebraic number field. Their algorithm is a polynomial-time ff-algorithm (which is a deterministic algorithm that is allowed to call an oracle for factoring integers and polynomials over finite fields) in the case where the dimension of the matrix algebra, the degree of the number field and the discriminant of the number field are all bounded. More concretely, the running time of the algorithm is exponential in all these parameters. They also show that finding explicit isomorphisms between central simple K-algebras A and B of dimension n^2 over K can be reduced to finding an explicit isomorphism between the algebra $A \otimes B^{op}$ and $\mathcal{M}_{n^2}(K)$ (where B^{op} denotes the opposite algebra of B).

Then in [11] and independently in [30] an algorithm polynomial in $\log(d)$ was provided when A is isomorphic to $\mathcal{M}_2(\mathbb{Q}(\sqrt{d}))$. The case where $K = \mathbb{F}_q(t)$, the field of rational functions over a finite field was considered in [26] where the authors propose a randomized polynomial-time algorithm. This algorithm is somewhat analogous to that of [24] but it is polynomial in the dimension of the matrix algebra. Similarly to the number field case, this was extended to quadratic extensions (now with a restriction to odd characteristics) in [27].

Springer

In this paper we initiate a new method for dealing with field extensions which is analogous to Galois descent. It is known that finding an explicit isomorphism between A and $\mathcal{M}_n(K)$ is polynomial-time equivalent to finding a rank 1 element in A. Thus if one could find a subalgebra of A isomorphic to $\mathcal{M}_n(\mathbb{Q})$ or $\mathcal{M}_n(\mathbb{F}_q(t))$, then one could apply the known algorithms for the subalgebra and that would give an exponential speed-up in both cases. Furthermore, these types of methods should work equally for the function field and number field case which have completely different applications. In [27,30] this type of method is studied. In both cases one finds a simple subalgebra of A which is central simple over a subfield of the center of A. This subalgebra is not necessarily a full matrix algebra, but it is at least split by the center of A. This can be leveraged to compute zero divisors in A. The disadvantage of these methods is that they are based on explicit calculations and reductions to finding nontrivial zeros of quadratic forms which do not generalize easily.

In this paper we prove results of [30] in a more conceptual way and extend them to the isomorphism problem of two quaternion algebras over a quadratic global field. The main technique is to compute a maximal right ideal of the corestriction of the algebra A (which is an explicit construction corresponding to the usual corestriction on cohomology groups) and apply it to construct an involution of the second kind on A. In general this might not be useful, but when A possesses a canonical involution of the first kind, then composing the two kinds of involutions and taking fixed points gives us a central simple subalgebra over a smaller field. Fortunately, tensor products of quaternion algebras carry a canonical involution of the first kind which is exactly what we need. This provides an example of the explicit isomorphism problem when the degree of the field over $\mathbb{Q}$ or $\mathbb{F}_q(t)$ is fixed but the discriminant does not need to be bounded.

We also implement our algorithm in Magma [2]. In particular, this also involved implementing the main algorithm from [18,26]. The same implementation was used in [5] for matrix algebras of degree 2 in even characteristic. Here we use it for algebras of higher degree and study its efficiency. Even though our main algorithm runs in polynomial time, the implementation is not practical. The bottleneck of the computation seems to be computing maximal orders in higher degree split central simple algebras. The computationally expensive part is not the factorization of the discriminant of the starting order (which in the rational function field case is particularly fast), just the fact that the currently known maximal order algorithms run in polynomial time but with a large exponent. We analyze the complexity of maximal order algorithms given in [26, Sect. 3] and [14, Sect. 3 and 4] and we also provide some substantial speed-ups in the case relevant to our main algorithm (when the algebra is obtained as a corestriction).

The paper is structured as follows. Section 2 contains number theoretic and algorithmic preliminaries. Section 3 is devoted to the general method of computing involutions of the second kind and computing Galois descents of quaternion algebras. In Sect. 4 we describe our main algorithm for finding explicit isomorphisms between quaternion algebras over quadratic extensions of either $\mathbb{Q}$ or $\mathbb{F}_q(t)$ (where q can be even as well). Section 5 is devoted to complexity estimates and optimisation tricks to speed-up the computations. Section 6 contains some details about our Magma implementation[1].

[1] https://github.com/QuaternionIsomorphisms/QuaternionIsomorphisms/

2 Preliminaries

2.1 General algebraic background

Definition 2.1 Let K be a field and let A be a finite dimensional algebra over K. Then A is a *central simple algebra* over K if it is simple and its center $Z(A)$ equals K (central). A central simple algebra A over the field K that has dimension 4 over K is called a *quaternion algebra*.

By a fundamental result of Wedderburn, a central simple algebra A is isomorphic to the full matrix algebra $\mathcal{M}_n(D)$ for some division ring D. In particular, a quaternion algebra over K is either a division algebra or is isomorphic to the algebra of 2×2 matrices over K.

Definition 2.2 Let A be a central simple algebra over K. We say that A is *split* by a field extension L/K if $A \otimes_K L \cong \mathcal{M}_n(L)$ for a suitable n. If a central simple algebra over K is isomorphic to $\mathcal{M}_n(K)$, then we call the algebra *split* (i.e. a shorter version of split by the extension K/K).

Now we recall some facts about the Brauer group. Our main reference is [16].

Definition 2.3 We call the central simple K-algebras A and B *Brauer equivalent* if there exist integers $m, m' > 0$ such that $A \otimes_K \mathcal{M}_m(K) \cong B \otimes_K \mathcal{M}_{m'}(K)$. The Brauer equivalence classes of central simple K-algebras form a group under tensor product over K. This group is called the *Brauer group* $\mathrm{Br}(K)$ of K.

In order to state the cohomological interpretation of the Brauer group we need to introduce some further notation. For a field K we put K_{sep} for a fixed separable closure of K and $G_K := \mathrm{Gal}(K_{sep}/K)$ for the absolute Galois group.

Theorem 2.4 [16, Thm. 4.4.3] *Let K be a field. Then the Brauer group* $\mathrm{Br}(K)$ *is naturally isomorphic to the second Galois cohomology group* $H^2(G_K, K_{sep}^{\times})$.

For specific fields one can even determine the Brauer group explicitly. The case of local fields is treated by the following famous result of Hasse.

Proposition 2.5 (Hasse) [16, Prop. 6.3.9, Rem. 6.5.6] *Let K be a nonarchimedean local field. Then we have a canonical isomorphism*

$$\mathrm{Br}(K) \cong \mathbb{Q}/\mathbb{Z}\,.$$

Moreover for a finite separable extension L/K there are commutative diagrams

$$\begin{tikzcd} \mathrm{Br}(L) \arrow[r, "\cong"] \arrow[d, "\mathrm{Cor}"'] & \mathbb{Q}/\mathbb{Z} \arrow[d, "\mathrm{id}"] \\ \mathrm{Br}(K) \arrow[r, "\cong"] & \mathbb{Q}/\mathbb{Z} \end{tikzcd} \quad \textit{and} \quad \begin{tikzcd} \mathrm{Br}(K) \arrow[r, "\cong"] \arrow[d, "\mathrm{Res}"'] & \mathbb{Q}/\mathbb{Z} \arrow[d, "|L\colon K|"] \\ \mathrm{Br}(L) \arrow[r, "\cong"] & \mathbb{Q}/\mathbb{Z}\,, \end{tikzcd}$$

where the right vertical map in the second diagram is the multiplication by the degree $|L\colon K|$.

The map inducing the isomorphism $\mathrm{Br}(K) \cong \mathbb{Q}/\mathbb{Z}$ is classically called the *Hasse invariant map*. Note that in the archimedean case Frobenius' Theorem on division rings over the field of real numbers $\mathbb{R}$ is equivalent to the fact $\mathrm{Br}(\mathbb{R}) = \frac{1}{2}\mathbb{Z}/\mathbb{Z} \subset \mathbb{Q}/\mathbb{Z}$. Finally, since $\mathbb{C}$ is algebraically closed, we have $\mathrm{Br}(\mathbb{C}) = 0$.

Now let K be a *global field*, i.e either a number field (finite extension of $\mathbb{Q}$) or the function field $K = \mathbb{F}(C)$ of a smooth projective curve C over a finite field $\mathbb{F}$. Denote by $\mathcal{P}$ the set of (finite and infinite) places of K, i.e. in the function field case $\mathcal{P}$ is the set C_0 of closed points on C and in the number field case $\mathcal{P}$ consists of the prime ideals in the ring of integers of K and the set of equivalence classes of archimedean valuations on K. For a place $P \in \mathcal{P}$ we denote by K_P the completion of K at P. If A is a central simple algebra over K then $A_P := A \otimes_K K_P$ is a central simple algebra over K_P. This induces a natural map $\mathrm{Br}(K) \to \mathrm{Br}(K_P) \overset{\mathrm{inv}_P}{\to} \mathbb{Q}/\mathbb{Z}$. Note that every central simple algebra A splits at all but finitely many places, i.e. we have $\mathrm{inv}_P([A_P]) = 0$ for all but finitely many P. Using the main results of class field theory one obtains the following classical theorem of Hasse.

Theorem 2.6 (Hasse) [16, Cor. 6.5.4, Rem. 6.5.6] *For any global field K we have an exact sequence*

$$0 \to \mathrm{Br}(K) \to \bigoplus_{P \in \mathcal{P}} \mathrm{Br}(K_P) \overset{\sum \mathrm{inv}_P}{\to} \mathbb{Q}/\mathbb{Z} \to 0\,.$$

Note that the Hasse-invariant of a nonsplit quaternion algebra over a local field is $\frac{1}{2}$. In particular, any quaternion algebra A over K splits at an even number of places. Further, for any finite subset $S \subset \mathcal{P}$ of even cardinality there exists a unique quaternion algebra (upto isomorphism) over K that splits exactly at the places in $\mathcal{P} \setminus S$. This is usually referred to as Hilbert's reciprocity law.

Finally, we briefly recall the definition and basic properties of orders in central simple algebras over local and global fields.

Definition 2.7 Let R be a Dedekind domain and K be its field of fractions. An *R-order* in a central simple algebra A over K is a subring O in A that is a finitely generated R-submodule in A such that $K \cdot O = A$ (i.e. O is a full R-lattice in the K-vectorspace A). We call an order $O \subset A$ *maximal* if it is maximal with respect to inclusion.

By the following result, being a maximal order is a local property.

Theorem 2.8 *[32, Cor. 11.2] An R-order O in A is maximal if and only if for each maximal ideal P in R the localization O_P is a maximal R_P-order in A.*

2.2 The corestriction of a central simple algebra

Due to the fact that the Brauer group admits a cohomological interpretation, one can use standard techniques from Galois cohomology to analyze central simple algebras. Let L be a finite Galois extension of K (contained in the fixed separable closure K_{sep}). Let G_K and G_L be the absolute Galois group of K and L respectively. There are two standard maps to analyze: restriction, which is a map from $H^2(G_K, K_{sep}^\times)$ to $H^2(G_L, K_{sep}^\times)$ and corestriction which is a map from $H^2(G_L, K_{sep}^\times)$ to $H^2(G_K, K_{sep}^\times)$.

For our purposes we need explicit descriptions of these maps on central simple algebras. The restriction map is easy, one just considers the extensions of scalars by L (i.e. the map $A \mapsto A \otimes_K L$). However the corestriction map is more complicated. We describe the corestriction map when L is a separable quadratic extension of K. This discussion is taken from [29, Sect. 3B] (in that book the corestriction is called the norm of an algebra).

Let L be a separable quadratic extension of a field K. Let σ be a generator of $\mathrm{Gal}(L/K)$. Let A be a central simple algebra over L. Then we define A^σ to be the algebra you apply σ to every entry in the multiplication table of A. Alternatively, one can define A^σ as a collection of elements $\{a^\sigma \mid a \in A\}$ with the following properties:

$$a^\sigma + b^\sigma = (a+b)^\sigma,\ a^\sigma b^\sigma = (ab)^\sigma\ , (\lambda \cdot a)^\sigma = \sigma^{-1}(\lambda) a^\sigma\ (\lambda \in L)\,.$$

A^σ is also a central simple L-algebra and the map $A \to A^\sigma$ given by $a \mapsto a^\sigma$ is a K-algebra isomorphism (but is visibly not L-linear).

Definition 2.9 Let L be a separable quadratic extension of K. Let A be a central simple L-algebra. The *switch map s* is the K-linear endomorphism of $A \otimes_L A^\sigma$ defined on elementary tensors by $s(a \otimes b^\sigma) = b \otimes a^\sigma$, extended K-linearly.

Proposition 2.10 [29, Proposition 3.13.] *The elements of $A \otimes_L A^\sigma$ invariant under the switch map form a subalgebra which is a central simple algebra over K of dimension* $\dim_L(A)^2$ *over K.*

The algebra in Proposition 2.10 is called the *corestriction* of A (with respect to the extension L/K). It induces the corestriction map of Galois cohomology. Our main application of the corestriction maps concerns involutions of central simple algebras. Recall that an involution of the central simple algebra A of the *second kind* is an involution whose restriction to the center L of A is nontrivial. For an overview of involutions the reader is referred to [29, Chapter 1, Sects. 1–3]. The main result we use is the following:

Theorem 2.11 *Let L/K be a quadratic Galois extension and let A be a central simple algebra over L. Then A admits an involution of the second kind if and only if the corestriction of A is split.*

The proof of this theorem in [29] is constructive which we will exploit in later sections.

2.3 Corestriction of maximal orders

For the purpose of optimising maximal order computation in the corestriction of a matrix algebra (see Sect. 5 for details), we need to consider the intersection of the corestriction of a central simple algebra with a maximal order over a Dedekind domain strictly contained in the base global field. In this section, we deal with the situation at unramified primes. This case is already well known, and we rely on the exposition given in [13]. We discuss the situation at ramified primes in Proposition 5.5.

The unramified case is the case where S is a Galois R-algebra. We quote as a definition of Galois extensions of rings the characterization given by point (6) of [13, Theorem 12.2.9]:

Definition 2.12 Let R be a commutative ring, and S a commutative R-algebra. Let G be a finite group of R-algebra automorphisms of S. Then S is a *Galois extension* of R with group G if the following conditions are verified:

1. $S^G = R$
2. for each maximal ideal $\mathfrak{m}$ of S and for each non-trivial $\sigma \in G$, there is an $x \in S$ such that $\sigma(x) - x \notin \mathfrak{m}$.

Let K be a global field and L be a Galois extension of K with Galois group G. Assume that $R \subsetneq K$ is a Dedekind domain and S is the integral closure of R in L. We have the following

Lemma 2.13 *The ring S is a Galois extension of R with group G if and only if no prime ideal of R is ramified in S.*

Proof Since $R = S \cap K$ it is clear that $R = S^G$. Now, we let $\mathfrak{P}$ be a prime ideal of S, lying above a prime $\mathfrak{p}$ in R. Then if $\mathfrak{p}$ does not ramify in L, then for each $1 \neq \sigma \in G$ either $\sigma(\mathfrak{P}) \neq \mathfrak{P}$ or σ induces a non-trivial automorphism of the residue field of $\mathfrak{P}$. In both cases, we may find some $x \in S$ such that $\sigma(x) - x \notin \mathfrak{P}$. Conversely, if $\mathfrak{p}$ ramifies in S then the inertia subgroup $I_{\mathfrak{P}}$ is nontrivial, ie. there exists an element $1 \neq \sigma \in G$ such that $\sigma(\mathfrak{P}) = \mathfrak{P}$ and σ acts trivially on the residue field of $\mathfrak{P}$. So for all $x \in S$, $\sigma(x) - x \in \mathfrak{P}$. □

For the remainder of this subsection, assume that L is a quadratic extension and $G = \{1, \sigma\}$. We now give a definition of the corestriction of an S-order in A. Note that by [13, Theorem 14.1.12], this agrees with the more general construction [13, Definition 14.1.1] given in the case that S is a Galois extension of R. However, we give a definition which does not require S to be Galois over R, as we deal with such cases in Subsection 5.2.

Definition 2.14 Let A be a central simple algebra over L and let $\mathcal{O}$ be an S-order in A. Then we call *corestriction of* $\mathcal{O}$ the intersection of $\mathcal{O} \otimes_S \mathcal{O}^\sigma$ and the corestriction of A.

We may now state and prove the main result of this subsection:

Proposition 2.15 *Let $A = \mathcal{M}_n(L)$, and let $\mathcal{O}$ be a maximal S-order in A. Assume that no prime of R ramifies in L. Then the corestriction of $\mathcal{O}$ is a maximal R-order in the corestriction of A.*

Proof Recall that an Azumaya algebra over a ring R' is an R'-algebra A' that is finitely generated, projective and faithful as an R'-module and such that the map $s : A' \otimes_{R'} A'^{op} \to End_{R'}(A')$ is an isomorphism, where s is defined by $s(a \otimes b)(x) = axb$ for $a, b, x \in A'$ (see [13, Theorem 7.1.4 (3)] and [13, Corollary 1.1.16 (1)]).

Since A and its corestriction are matrix algebras (respectively over L and K), their maximal orders are Azumaya algebras (respectively over S and R). This follows from [13, Theorem 11.3.14], since the Brauer class of a matrix algebra is trivial in the Brauer group of its base field. Furthermore, any R-order that is an Azumaya R-algebra is a maximal order in the corestriction of A. This is the content of [13, Theorem 11.3.11].

Lemma 2.13 states that S is a Galois extension of R. In particular, by [13, Theorem 12.2.9], S is a separable R-algebra. Furthermore, by [13, Theorem 6.4.6], S is an R-progenerator module. It follows that [13, Theorem 14.1.9 (1)] applies, which states that the corestriction of an Azumaya S-algebra is an Azumaya R-algebra. □

2.4 Algorithmic preliminaries

In this subsection we give a brief overview of known algorithmic results in this context and provide more details of the algorithms specifically used in this paper.

Let K be a field and let A be an associative K-algebra given by the following presentation. One is given a K-basis $b_1, \ldots, b_m$ of A and a multiplication table of the basis elements, i.e. b_ib_j expressed as a linear combination $\sum_{k=1}^m \gamma_{i,j,k} b_k$. These $\gamma_{i,j,k}$ are called structure constants and we consider our algebra given by structure constants. It is a natural algorithmic problem to compute the structure of A, i.e., compute its Jacobson radical rad A, compute the Wedderburn decomposition of $A/\operatorname{rad} A$ and finally compute an explicit isomorphism between the simple components of $A/\operatorname{rad} A$ and $\mathcal{M}_n(D_i)$ where the D_i are division algebras over K and $\mathcal{M}_n(D_i)$ denotes the algebra of $n \times n$ matrices over D_i. The problem has been studied for various fields K, including finite fields, the field of complex and real numbers, global function fields and algebraic number fields. There exists a polynomial-time algorithm for computing the radical of A over any computable field [3]. There also exist efficient algorithms for every task over finite fields [14,34] and the field of real and complex numbers [8]. Finally, when $K = \mathbb{F}_q(t)$, the field of rational functions over a finite field $\mathbb{F}_q$, then there exist efficient algorithms for computing Wedderburn decompositions [23].

This motivates the algorithmic study of computing isomorphisms between simple algebras. Over finite fields every simple algebra is a full matrix algebra. Finding isomorphisms between full matrix algebras can be accomplished in polynomial time using the results from [14,34].

For more general fields, non-trivial central simple algebras exist. However, the discussion given in [24, Sect. 4] gives a polynomial-time reduction from the computation of isomorphism of matrix algebras to the general computation of isomorphism between central simple algebras. We record the result below.

Theorem 2.16 *Let A_1 and A_2 be isomorphic central simple algebras of degree n over an infinite field K. Then there is a polynomial-time reduction from computing an explicit isomorphism between A_1 and A_2 to computing an explicit isomorphism between $A_1 \otimes A_2^{op}$ and $\mathcal{M}_{n^2}(K)$.*

We now examine existing algorithms for the case where K is a global field.

2.5 Number fields

Over number fields there is an immediate obstacle. Rónyai [33] showed that this task is at least as hard as factoring integers. However, in most interesting applications factoring is feasible, thus it is a natural question to ask whether such an isomorphism can be computed if one is allowed to call an oracle for factoring integers.

In [20] the connection between norm equations and split cyclic algebras is exploited to compute an explicit isomorphism between a full matrix algebra and a split cyclic algebra. This method might be practical in certain cases but there is no known proven polynomial-time algorithm for solving norm equation. Also for general central simple algebras (other than degree 2 or 3) there is no known efficient algorithm to turn a structure constant representation into a cyclic algebra representation (thus [24] is more general). The main difficulty here is that in general not every central simple algebra is a cyclic algebra, such a statement is only true for global fields (thus an efficient algorithm would have to exploit the special structure of the field).

In [30] a polynomial-time algorithm (modulo factoring integers) is proposed for the $n = 2$ case when K is a quadratic field. This algorithm also uses an explicit descent method for finding a $\mathbb{Q}$-subalgebra of A. However, the subalgebra is computed by solving quadratic forms over $\mathbb{Q}$, while our approach applies more conceptual methods detailed in Sect. 3.

In [24] the authors propose such an algorithm when $A \cong \mathcal{M}_n(K)$ where K is a number field. This algorithm involves a search step for elements of small norm in a maximal order of A. However, when K is a non-trivial extension of $\mathbb{Q}$, the expected duration of the search grows exponentially in all the parameters (n, the degree and the size of the discriminant of K). It follows that [24] does not provide a polynomial-time algorithm for the problem over a general number field, or even over extensions of $\mathbb{Q}$ of bounded degree.

2.6 Function fields

In the case that $K = \mathbb{F}_q(t)$, where q is a prime power, an algorithm is given by [26]. In contrast to the number field case, this algorithm is polynomial in n and $\log q$ due to the fact that there is no exhaustive search step at the end. Instead of the search step, the intersection of two maximal orders, one over $\mathbb{F}_q[t]$ and one over the valuation ring for the degree valuation is computed. This yields a finite algebra over $\mathbb{F}_q$ which contains a rank one idempotent element. This techniques fails for function fields of positive genus.

When K is a finite extension of $\mathbb{F}_q(t)$, the only known case is the case of separable quadratic extensions. When q is odd, then [27] proposes a polynomial-time algorithm for finding zero divisors in split quaternion algebras over K using techniques similar to the ones developed in [30]. When q is even, then an analogous polynomial-time algorithm is presented in [5].

We emphasize that some of the previously mentioned algorithms (e.g., the main algorithm from [26]) have not been implemented and have no precise complexity estimate (beyond running in polynomial time). In this work we provide an implementation of [26] and analyze the complexity of certain subroutines (such as maximal order computation) in more detail.

3 The descent method

Let K be a field and let L be a separable quadratic extension of K. Let A be a central simple algebra over L given by structure constants. Our goal in this section is to find a subalgebra of A which is a central simple algebra over K. In other words, we would like to decompose A as a tensor product $B \otimes_K L$ when this is possible. Our main technical tool is an algorithm that computes the corestriction of a central simple algebra. We apply this the case of quadratic extensions.

Our first step is to construct an involution of the second kind on A if such an involution exists. The following lemma [29, Theorem 3.17] provides a useful relationship between certain right ideals of the corestriction of A and involutions of the second kind:

Lemma 3.1 *Let A be a central simple algebra over L of dimension n^2 where L is a separable quadratic extension of the field K. Put B for the corestriction of A with respect to L/K. Assume that there exists a right ideal I of B such that $A^\sigma \otimes_L A = I_L \oplus (1 \otimes A)$ where $I_L = I \otimes_K L$. Then A admits an involution of the second kind.*

Proof We sketch the proof here. For each $a \in A$ there exists a unique element $\tau_I(a) \in A$ such that

$$a^\sigma \otimes 1 - 1 \otimes \tau_I(a) \in I_L.$$

One can check that the map $a \mapsto \tau_I(a)$ is indeed an involution of the second kind on A. □

Now we propose an algorithm which either returns an involution of the second kind, or a zero divisor of A:

Algorithm 3.2 *Let L be a separable quadratic extension of a field K. Let A be a central simple algebra over L of dimension n^2 and let B be its corestriction with respect to the field extension $L|K$. Finally, suppose that A admits an involution of the second kind (i.e., B is isomorphic to $\mathcal{M}_{n^2}(K)$ by Theorem 2.11). Then the following algorithm computes either a zero divisor or an involution of the second kind of A:*

1. *Compute a maximal right ideal I in B.*
2. *Let $I_L = I \otimes L$ be the scalar extension of I in $A^\sigma \otimes A$. Compute the intersection of I_L and $1 \otimes A$.*
3. *If $I_L \cap 1 \otimes A \neq 0$, then we have computed a zero divisor in A*
4. *If $I_L \cap 1 \otimes A = 0$, then I is a right ideal with the property that $A^\sigma \otimes_L A = I_L \oplus (1 \otimes A)$ by dimension considerations which allows us to construct an involution of the second kind.*

The following theorem shows that if one is allowed to call an oracle for the first step, which is essentially equivalent to finding an explicit isomorphism between the corestriction B and $\mathcal{M}_{n^2}(K)$), then the rest of the algorithm runs in polynomial time.

Theorem 3.3 *Let L be a separable quadratic extension of a field K. Let A be a central simple algebra over L of dimension n^2 which admits an involution of the second kind. Then Algorithm 3.2 gives a polynomial-time reduction from the problem of computing an involution of the second kind in A to the problem of computing a maximal right ideal in A.*

Proof Let B be the corestriction of A. Our assumptions together with Theorem 2.11 imply that B is split. The correctness of Algorithm 3.2 follows mostly from Lemma 3.1, we only have to show that every element of I_L is a zero divisor. Every element of I is a zero divisor as B is a full matrix algebra and non-units are automatically zero divisors. Now I_L is obtained from I by extensions of scalars hence every element of I_L is a zero divisor as well.

Now we discuss the complexity of the steps of the algorithm. Computing a right ideal is a subroutine required by the statement of the theorem, thus Step 1 can be carried out in polynomial time. Step 2 computes the intersection of two L-subspaces which can be accomplished by solving a system of linear equations over L. Finally, the last step runs in polynomial time by Lemma 3.1. □

The above proof is particularly interesting when one is looking for zero divisors in quaternion algebras.

Proposition 3.4 *Let L be a separable quadratic extension of K. Then there exists a polynomial-time reduction from the problem of finding a K-sulbalgebra of a quaternion algebra over L that is a quaternion algebra over K to the problem of finding an explicit isomorphism between a degree 4 split central simple algebra over K given by structure constants and $\mathcal{M}_4(K)$.*

Proof Let A be a quaternion algebra over L which contains a K-subalgebra that is a quaternion algebra over K, and let B be the corestriction of A. B is then a split central simple algebra of degree 4 over K. Computing an explicit isomorphism with $\mathcal{M}_4(K)$ allows us to find a maximal right ideal of B, which we use as input for Algorithm 3.2.

Algorithm 3.2 returns either a zero divisor or an involution of the second kind on A. If it returns a zero divisor, then one can efficiently construct an explicit isomorphism between A and $\mathcal{M}_2(L)$ which provides a subalgebra isomorphic to $\mathcal{M}_2(K)$. If Algorithm 3.2 returns an involution of the second kind, then one can compose that with the canonical involution (conjugation) on A. Then the fixed points of this map form a quaternion subalgebra over K. □

When L is a quadratic extension of $K = \mathbb{Q}$ or $K = \mathbb{F}_q(t)$, then there already existed efficient algorithms for computing quaternion subalgebras over K in quaternion algebras over L ([30, Corollary 19], [27, Proposition 42]) using explicit calculations and utilizing algorithms for finding nontrivial zeros of quadratic forms. Proposition 3.4 shows a more conceptual method for computing subalgebras which avoids tedious calculations. Furthermore, this proposition applies to quaternion algebras in characteristic 2 as well.

Corollary 3.5 *Let L be a separable quadratic extension of* $K = \mathbb{F}_{2^k}(t)$ *and A a quaternion algebra over L. There exists a polynomial-time algorithm which computes a quaternion subalgebra over K of A if such a quaternion algebra exists.*

Proof The statement follows from Proposition 3.4 and the fact that there exists a polynomial-time algorithm for finding explicit isomorphisms between an algebra A given by structure constants and $\mathcal{M}_4(\mathbb{F}_{2^k}(t))$ [26]. □

Let L be a quadratic extension of $K = \mathbb{F}_{2^k}(t)$ and A be an algebra isomorphic to $\mathcal{M}_2(L)$ given by structure constants. Combining Corollary 3.5 with [5, Theorem 3.19] one has the following result:

Theorem 3.6 *Let L be a quadratic extension of* $K = \mathbb{F}_{2^k}(t)$ *and A an algebra isomorphic to* $\mathcal{M}_2(L)$ *given by structure constants. Then there exists a polynomial-time algorithm that computes a zero divisor in A.*

4 The main algorithm

In this section we propose our main algorithm for computing explicit isomorphisms between quaternion algebras over quadratic global fields.

We start with a small observation regarding the isomorphism problem of rational quaternion algebras. It is known that there is a polynomial-time algorithm for this task if one is allowed to call an oracle for factoring integers. Furthermore, there is a polynomial-time reduction from the problem of computing explicit isomorphisms of rational quaternion algebras to factoring, which implies that the factoring oracle is indeed necessary.

Let $B_{p,\infty}$ be the rational quaternion algebra which is ramified exactly at p and at infinity. In [9] the authors study the following problem: if we are given two quaternion algebras isomorphic to $B_{p,\infty}$ and we are also given a maximal order in both quaternion algebras, can we compute an explicit isomorphism between them without relying on a factoring oracle. The motivation for this problem comes from the fact that the endomorphism ring of a supersingular elliptic curve over a field of characteristic p> 0 is a maximal order in $B_{p,\infty}$. The authors propose a heuristic algorithm which does not rely on factoring. Here we propose an algorithm for this task which does not rely on any heuristics:

Proposition 4.1 *Let A, B be rational quaternion algebras both known to be isomorphic to $B_{p,\infty}$ and let O_1, O_2 be maximal orders in A and B respectively. Then there exists a polynomial-time algorithm which computes an isomorphism between A and B.*

Proof In [24] the authors show that finding an isomorphism between A and B can be reduced to finding a primitive idempotent in $C = A \otimes_{\mathbb{Q}} B^{op}$. First observe that $O_1 \otimes O_2^{op}$ is an order in C which is locally maximal at every prime except at p. Thus we can find a maximal order containing $O_1 \otimes O_2^{op}$ in polynomial time without factoring using the algorithm from [36] (in the general algorithm one needs to factor the discriminant of the order but in this case the factorization is already known as the discriminant of both O_1 and O_2 is p^2). Finally, we use the algorithm from [25] which finds a primitive idempotent. □

Remark 4.2

1. At the end of the above proof we could use the algorithm from [24] but then it might only find a zero divisor which is not enough for our purposes (as it reduces to finding a zero divisor in a quaternion algebra where we do not have a maximal order).
2. The same reasoning applies to the case where A and B are isomorphic rational quaternion algebras and one knows the places at which the algebras ramify.

The main goal of the remainder of the section is to design an efficient algorithm which computes an explicit isomorphism between isomorphic quaternion algebras over quadratic extensions L of $\mathbb{Q}$ or $\mathbb{F}_q(t)$ (where q is a prime power and can be even).

Thus if one is given two quaternion algebras A_1 and A_2 over L which is a separable quadratic extension of either $K = \mathbb{Q}$ or $K = \mathbb{F}_q(t)$, then by Theorem 2.16 it is enough to find an explicit isomorphism between $A_1 \otimes A_2^{op}$ and $\mathcal{M}_4(L)$. Note that when $K = \mathbb{Q}$ the paper [24] proposes such an algorithm but it is exponential in the size of the discriminant of $L/\mathbb{Q}$. We will get around this issue by exploiting the fact that in this case $\mathcal{M}_4(L)$ is not given by a usual structure constant representation but as a tensor product of two quaternion algebras.

First we identify three algorithmic problems on which the main algorithm will rely:

Problem 1 *Let K be a field and let A be an algebra over K known to be isomorphic to $\mathcal{M}_4(K)$ or $\mathcal{M}_{16}(K)$ given by structure constants (without an isomorphism). Compute a maximal right ideal of A.*

Remark 4.3 Problem 1 is equivalent to finding an explicit isomorphism between A and $\mathcal{M}_4(K)$ or $\mathcal{M}_{16}(K)$.

Problem 2 *Let K be a field and let D be a quaternion division algebra over K. Let A be an algebra over K isomorphic to $\mathcal{M}_2(D)$ given by structure constants. Compute a zero divisor in A.*

Problem 3 *Let K be a field and let L be a separable quadratic extension of K. Let A be a split quaternion algebra over L given by structure constants. Compute a zero divisor in A.*

Let K be a global field and let L be a separable quadratic extension of K. We show that if one can find efficient algorithms for these problems then there exists an efficient algorithm for computing explicit isomorphisms between quaternion algebras over L.

Remark 4.4 In our applications K will be either $\mathbb{Q}$ or $\mathbb{F}_q(t)$, but we prefer to state the above problems in this generality for the following two reasons. First, both algorithms would follow the exact same outline, only the subroutine for the aforementioned Problems 1, 2 and 3 would be different. Second, a general framework might have applications over global fields other than $\mathbb{Q}$ or $\mathbb{F}_q(t)$. For example when $K = \mathbb{Q}(\sqrt{2})$, Problem 1 admits a polynomial-time algorithm and thus only the other two have to be dealt with.

Theorem 4.5 *Let A_1 and A_2 be isomorphic quaternion algebras over L where L is a quadratic extension of a global field K. Suppose there exist polynomial-time algorithms (with an oracle for factoring integers in the case that K has characteristic zero) for Problems 1, 2 and 3. Then there exists a polynomial-time algorithm for computing an isomorphism between A_1 and A_2.*

Proof We provide an algorithm for computing an explicit isomorphism between $A_1^{op} \otimes A_2$ and $\mathcal{M}_4(L)$. Then Theorem 2.16 implies that one can compute an explicit isomorphism between A_1 and A_2 in polynomial time.

Let $B = A_1^{op} \otimes A_2$. Then one can compute an involution of the first kind on B since it is given as a tensor product of quaternion algebras (i.e., we take the canonical involution on each component of the tensor product).

Applying Theorem 3.3 one can either construct an involution of the second kind or a zero divisor in B using an efficient algorithm for Problem 1. Suppose first that the algorithm from Theorem 3.3 finds a zero divisor a in B. If the zero divisor has rank 1 or 3 (here rank means its rank as a matrix which can be computed by computing the dimension of the left ideal it generates), then one can find either a rank 1 or a rank 3 idempotent by computing the left unit of the right ideal (i.e., an element e such that for every element b in the right ideal, $eb = b$) generated by a. Observe that if an idempotent e has rank 3, then $1 - e$ has rank 1, and thus one has actually found a primitive idempotent in both cases, which implies an explicit isomorphism between B and $\mathcal{M}_4(L)$. If a has rank 2, then we construct an idempotent e of rank 2 in a similar fashion. Then $eBe \cong \mathcal{M}_2(L)$ and computing an explicit isomorphism between them can be used to construct an explicit isomorphism between B and $\mathcal{M}_4(L)$ (as a rank one element in $eBe \cong \mathcal{M}_2(L)$ has rank 1 in B). Computing an explicit isomorphism between eBe and $\mathcal{M}_2(L)$ is exactly Problem 3. Note that the discussion also implies that it is enough to find a zero divisor in B as it can be used for constructing an explicit isomorphism between B and $\mathcal{M}_4(L)$.

Now we can suppose that the algorithm from Theorem 3.3 has computed an involution of the second kind on B. We then have an involution of the second kind and an involution of the first kind on A. Composing them and taking fixed points finds a subalgebra C of B which is a central simple algebra of degree 4 over K and $C \otimes_K L = B$. There are 3 kinds of central simple algebras of degree 4: full matrix algebras, division algebras, and 2×2 matrix algebras over a division quaternion algebra. When C is a full matrix algebra over K, then one can use an algorithm for Problem 1 to compute a zero divisor. When C is a 2×2 matrix algebra over a division quaternion algebra, then computing a zero divisor in C is an instance of Problem 2. Finally, C is never a division algebra as it is split by a quadratic extension (the smallest splitting field of a degree 4 central simple division algebra has degree 4 over the ground field for global fields). □

After obtaining a general algorithm our goal is to look at the Problems 1, 2 and 3 in the cases where $K = \mathbb{Q}$ or $K = \mathbb{F}_q(t)$.

4.1 Rational function fields

We begin with the case when $K = \mathbb{F}_q(t)$ and q is odd:

1. Problem 1 can be solved in polynomial time using the main algorithm from [26, Sect. 4].
2. Problem 2 can be obtained in polynomial time using the algorithm from [18, Corollary 17].
3. Problem 3 admits a polynomial-time algorithm derived in [27, Proposition 43].

Now we look at the case where q is even :

1. Problem 1 can be accomplished in polynomial time using the main algorithm from [26, Sect. 4].
2. Problem 2 admits a polynomial-time algorithm by [5, Corollary 3.22].
3. Problem 3 admits a polynomial-time algorithm by Theorem 3.6.

All these imply the following:

Corollary 4.6 *Let L be a separable quadratic extension of $\mathbb{F}_q(t)$ where q is a prime power (which can be even). Let A_1 and A_2 be two isomorphic quaternion algebras over L. Then there exists a randomized polynomial-time algorithm which computes an isomorphism between A_1 and A_2.*

4.2 The rationals

Now we turn our attention to the $K = \mathbb{Q}$ case. Problem 1 can again be accomplished in polynomial time (with the help of an oracle for factoring integers) using the algorithm from [24, Sect. 2]. Problem 3 can also be obtained in polynomial time using an oracle for factoring integers. One has to use the algorithm [30, Corollary 19].

There is no known algorithm for Problem 2 in the rational case. In the rest of this section we propose a polynomial-time algorithm for this task which is analogous to [18, Corollary 17]. The key ingredient of the algorithm is a special case of a result in [1] (see also the Master's thesis of Schwinning [35] where the construction is made explicit):

Lemma 4.7 *Suppose one is given a list of places $v_1, \ldots, v_k$ of $\mathbb{Q}$ where k is even. Then there exists a polynomial-time algorithm which constructs a quaternion algebra over $\mathbb{Q}$ which ramifies at exactly those places.*

Proposition 4.8 *Let A be an algebra isomorphic to $\mathcal{M}_2(D)$ where D is a quaternion division algebra over $\mathbb{Q}$. Then there exists a polynomial-time algorithm which is allowed to call an oracle for factoring integers which computes a zero divisor in A.*

Proof First we compute a maximal order in A using the algorithm from [22, Corollary 6.5.4]. An extension of this algorithm [21] computes the places where the algebra A ramifies. Now we use Lemma 4.7 to compute a division algebra D_0 which ramifies at exactly those places as A, which implies that $A \cong \mathcal{M}_2(D_0)$. Now we proceed in a similar fashion as in [18, Theorem 16] or [5, Corollary 3.22] but by invoking the algorithm from [24] for computing the required explicit isomorphism. □

An immediate corollary is the following:

Corollary 4.9 *Let L be a quadratic extension of $\mathbb{Q}$ and let A_1 and A_2 be isomorphic quaternion algebras over L. Then there exists a polynomial-time algorithm which is allowed* *to call an oracle for factoring integers, that computes an explicit isomorphism between A_1 and A_2.*

5 Complexity questions and optimisations

In this section, we give complexity estimates for the computation of maximal orders in separable algebras over function fields. We then present optimisations that are relevant to our use case. Namely, in Algorithm 1 our algebra A is a tensor product of quaternion algebras which allows for case specific optimizations. More precisely, we compute maximal orders for the smallest possible algebras and use them to construct orders with small discriminant in the algebras that we generate throughout execution of Algorithm 1.

5.1 Complexity of maximal order computation

The complexity bottleneck of our algorithm is the computation of various maximal orders. Although polynomial-time algorithms exist for this task (see [15,26]), the actual complexity makes them rather impractical as soon as the degree of A increases. Throughout the execution of Algorithm 1, we may encounter two K-algebras of degree 16. One is the corestriction of $A = B_1 \otimes B_2$ and the other is $A_K \otimes \mathcal{M}_2(D)$, which appears when A_K itself is isomorphic to some $\mathcal{M}_2(D)$, with D a division quaternion algebra (see Sects. 4 and 6 for more details). In both cases, we need to compute a zero divisor and therefore we need to compute maximal orders (in fact, we compute a maximal order over the ring $\mathbb{F}_q[t]$ and another one over the valuation ring corresponding to the degree valuation). We review descriptions of the algorithm used for maximal order computations in Magma, and give an upper bound for its complexity.

The algorithm used for computing maximal orders over Dedekind domains in associative algebras over global function fields is the one given in Sects. 3 and 4 of [15], which is similar to the algorithm described in Sect. 3 of [26]. The computation proceeds from a starting order Λ_0. Letting μ be half the degree of the discriminant of Λ_0, the algorithm has a worst-case complexity of $O(\mu n^5)$, where n is the dimension of the input algebra (see [15, Proposition 3.17 and Remark 4.18]).

If no starting order is given, one is computed from the given basis of the input algebra. However, according to the discussion in Subsection 3.3 of [26], an upper bound for μ is then $(n^8 d_D + n^2 d_N)$, where d_D and d_N are upper bounds respectively of the degrees of the denominators and of the numerators of the structure constants of A. Note that in [26], n is the degree of the algebra, while the convention used in [15] is that n is the dimension. We obtain the following:

Proposition 5.1 *The cost of computing a maximal order in a separable $\mathbb{F}_q(t)$-algebra A of dimension n, such that the numerators and denominators of the structure constants of A are bounded above by a constant $C \in \mathbb{R}_{>0}$ is $O(n^9)$.*

The cost of computing a maximal overorder of an order with discriminant μ, however, is $O(\mu n^5)$.

Remark 5.2 [26] states its result for algebras that are isomorphic to matrix algebras, but this hypothesis is not used in the estimation of bounds for the degree of the discriminant. The estimates are therefore valid for more general separable algebras.

5.2 Optimisation of the maximal order computations

As suggested by Proposition 5.1, computing maximal orders in degree 16 matrix algebras is the computational bottleneck of our algorithm. However, this complexity depends on the degree of the discriminant of the order with which we start our computation. We use this to our advantage, by computing maximal orders for the input quaternion algebras, and then passing their bases through the various operations we execute on the algebras (tensor product, corestriction and Galois descent). While it is not true that after applying these operations we always get maximal orders, we may control the growth of the discriminant, and therefore the complexity of the later maximal order computations.

We now give results concerning the discriminant of orders passing through our various operations. In this context, R is a Dedekind domain, and K is the fraction field of R. We stress that the results given here are targeted for function fields of odd characteristic, as this is the use case of our implementation.

Proposition 5.3 *Let A and B be central simple algebras over K, respectively of dimension m and n, and let O_A and O_B be R-orders respectively of A and B. Then $O_A \otimes_R O_B$ is an R-order in $A \otimes_K B$, and*

$$\mathrm{Disc}(O_A \otimes_R O_B) = \mathrm{Disc}(O_A)^n \mathrm{Disc}(O_B)^m.$$

Proof This is [28, Eq. 3.5]. □

Next, we consider the computation of the corestriction of a matrix algebra $A = \mathcal{M}_n(K)$ on a quadratic extension K of a rational function field $\mathbb{F}_q(t)$ in odd characteristic. Let σ be the non-trivial $\mathbb{F}_q(t)$-automorphism of K. We let $R \subsetneq \mathbb{F}_q(t)$ be a Dedekind domain, and we call S the integral closure of R in K. Let O be a maximal S-order in A. Then $O \otimes_R O^\sigma$ embeds in $A \otimes_R A^\sigma$ in an obvious manner and is stable under the switch map (see Definition 2.9). Following Definition 2.14, we call $\mathrm{Cor}(O) = (O \otimes_R O^\sigma) \cap \mathrm{Cor}(A)$ the corestriction of O. We may easily construct a basis of $\mathrm{Cor}(O)$ in $\mathrm{Cor}(A)$ from a basis of O in A. Unfortunately, $\mathrm{Cor}(O)$ is not a maximal R-order in $\mathrm{Cor}(A)$. However, we compute its discriminant, whose degree only depends on the quadratic field K. We first need a lemma:

Lemma 5.4 *With notations as above, let us assume further that R is a DVR, and that its corresponding valuation in $\mathbb{F}_q(t)$ ramifies in K. Then S admits a uniformizer π such that $\sigma(\pi) = -\pi$.*

Proof Since q is odd, we may find $\theta \in K \setminus \mathbb{F}_q(t)$ such that $\theta^2 \in \mathbb{F}_q(t)$. That is, $\sigma(\theta) = -\theta$. Up to multiplication by an element of $\mathbb{F}_q(t)$, we may assume that $\theta \in S$ and that its valuation is 0 or 1. Let k be the residue field of S, and then σ induces the identity on k. In k, we therefore have $\overline{\sigma(\theta)} = \overline{\theta} = -\overline{\sigma(\theta)}$ and since k has odd characteristic, $\overline{\theta} = \overline{\sigma(\theta)} = 0$. Therefore, θ is a uniformizer of S and $\sigma(\theta) = -\theta$. □

Proposition 5.5 *Let the notations be as above. Then let $p_1, \ldots, p_m$ be the irreducible elements of R that ramify in S. Then*

$$\mathrm{Disc}(\mathrm{Cor}(O)) = \prod_{1 \le i \le m} p_i^{\frac{n^4-n^2}{2}}.$$

Proof We first prove the result in the case that R is a DVR. Let v be the valuation corresponding to R in K.

If v does not ramify in S, then this is Proposition 2.15. We now assume that v ramifies in S.

For the computation that follows, we will use the delta symbol for tuples. By this, we mean that if (i, j) and (o, p) are couples of indices, then $\delta_{(i,j),(o,p)}$ is 1 if $(i, j) = (o, p)$ and is zero otherwise. The definition is extended to tuples with more than two elements in the obvious manner. We also will use the lexicographic order on tuples of indices.

Let π be a uniformizer of S such that $\sigma(\pi) = -\pi$, which exists by Lemma 5.4. Up to conjugation by an automorphism, we may assume that $O = \mathcal{M}_n(S)$. Let $(E_{i,j})_{1 \le i,j \le n}$ be the canonical matrix basis of $\mathcal{M}_n(S)$ over S. Then a basis of $\mathrm{Cor}(O)$ is

$$\begin{aligned} B = {} & (E_{i,j} \otimes E_{i,j})_{(1,1) \le (i,j) \le (n,n)} \\ & \cup (E_{i,j} \otimes E_{k,l} + E_{k,l} \otimes E_{i,j})_{(1,1) \le (i,j) < (k,l) \le (n,n)} \\ & \cup (\pi (E_{i,j} \otimes E_{k,l} - E_{k,l} \otimes E_{i,k}))_{(1,1) \le (i,j) < (k,l) \le (n,n)}. \end{aligned}$$

The discriminant of $\mathrm{Cor}(O)$ is then the ideal of R generated by

$$\det(tr(b_i b_j))_{1 \le i,j \le n^4}.$$

Since R is a DVR, we in fact only need to compute the valuation of this determinant in R.

We now compute the value of $tr(b_i b_j)$ for the various choices of b_i and b_j in B. We use the general fact that $tr(E_{i,j} E_{k,l}) = \delta_{(i,j),(l,k)}$. For what follows, we consider the indices $1 \le i, j, k, l, o, p, q, r \le n$. We also make the assumptions that $(i, j) \ne (k, l)$ and that $(o, p) \ne (q, r)$. It is then straightforward to check the following identities.

$$\begin{aligned} tr((E_{i,j} \otimes E_{i,j})(E_{o,p} \otimes E_{o,p})) &= \delta_{(i,j),(p,o)} \\ tr((E_{i,j} \otimes E_{i,j})(E_{o,p} \otimes E_{q,r} + E_{q,r} \otimes E_{o,p})) &= 0 \\ tr((E_{i,j} \otimes E_{i,j})(E_{o,p} \otimes E_{q,r} - E_{q,r} \otimes E_{o,p})) &= 0 \\ tr((E_{i,j} \otimes E_{k,l} + E_{k,l} \otimes E_{i,j})(E_{o,p} \otimes E_{q,r} - E_{q,r} \otimes E_{o,p})) &= 0 \\ tr((E_{i,j} \otimes E_{k,l} - E_{k,l} \otimes E_{i,j})(E_{o,p} \otimes E_{q,r} + E_{q,r} \otimes E_{o,p})) &= 0 \\ tr((E_{i,j} \otimes E_{k,l} + E_{k,l} \otimes E_{i,j})(E_{o,p} \otimes E_{q,r} + E_{q,r} \otimes E_{o,p})) &= 2(\delta_{(i,j,k,l),(p,o,r,q)} + \delta_{(i,j,k,l),(r,q,p,o)}) \\ tr((E_{i,j} \otimes E_{k,l} - E_{k,l} \otimes E_{i,j})(E_{o,p} \otimes E_{q,r} - E_{q,r} \otimes E_{o,p})) &= 2(\delta_{(i,j,k,l),(p,o,r,q)} - \delta_{(i,j,k,l),(r,q,p,o)}) \end{aligned}$$

Now, the last two lines represent the trace of the product of two elements of B if and only if the inequalities $(i, j) < (k, l)$ and $(o, p) < (q, r)$ are satisfied. Given i, j, k, l such that $(i, j) < (k, l)$, either $(j, i) < (l, k)$ or $(l, k) < (j, i)$.

It follows that each line of the matrix $\left(tr(b_\alpha b_\beta)_{1 \le \alpha, \beta < n^4}\right)$, has only one non-zero coefficient. The non-zero coefficient has valuation 0 in S, unless the index of the line is larger than $\frac{n^4+n^2}{2}$, in which case the valuation is 2. Since the matrix is symmetric, this property is also true for its columns. It follows that there exists a permutation of the columns such that the resulting matrix is diagonal. Therefore, the valuation of $\det\left(tr(b_\alpha b_\beta)_{1 \le \alpha, \beta < n^4}\right)$ is $n^4 - n^2$ in S. As a result, letting $\mathfrak{p}$ be the unique maximal ideal of R, we get

$$\mathrm{Disc}(\mathrm{Cor}(O)) = \mathfrak{p}^{\frac{n^4-n^2}{2}}.$$

Now, let R be a Dedekind domain. Then for any R-order O', it is well known that $\mathrm{Disc}(O') = \bigcap_{\mathfrak{p} \in \mathrm{Spec}(R)} \mathrm{Disc}(O'_{\mathfrak{p}})$. Therefore, the result will follow from the DVR case if we prove that for a prime $\mathfrak{p}$ of R, $\mathrm{Cor}(R_{\mathfrak{p}} O) = R_{\mathfrak{p}} \mathrm{Cor}(O)$. However, this is immediate as multiplication by an element of $R_{\mathfrak{p}}$ commutes with the switch map. □

The last operation to consider is the Galois descent operation, using an involution of the second kind. It does not seem possible here to obtain such explicit results as we have had before. A reason for that is that the discriminant of the resulting R-order largely depends on the choice of involution of the second kind. In [17], the situation is studied in the case of quaternion algebras.

Following results from this subsection, we make the following optimisations to our algorithm: Maximal orders of quaternion algebras B_1 and B_2 are immediately computed. Furthermore, after applying any operation to one of our algebras, we apply the same operation to its maximal orders and then compute a maximal order of the new algebra from the order we obtain.

We may now compare the efficiency of the optimised version of our algorithm and that of the naive one. Let B_1 and B_2 be two central simple K-algebras of equal dimensions, and let $A = B_1 \otimes_K B_2$. We assume that B_1 and B_2 are such that A is isomorphic to a matrix algebra. We compare the cost of the computation of a maximal order in $\mathrm{Cor}A$. The complexity estimates are given assuming that the degrees of the discriminants of B_1, B_2 and K are bounded by a constant C independent of n. For this discussion we let $n = \dim A$. Hence $\dim_{\mathbb{F}_q(t)} A = 2n$, $\dim_{\mathbb{F}_q(t)} B_i = 2n^{\frac{1}{2}}$ for $i \in \{1, 2\}$, and $\dim(\mathrm{Cor}A) = n^2$. Aside from the maximal order computations, we compute tensor products and a basis of fixed points by a linear map. We consider the cost of these linear algebra operations to be negligible compared to the cost of maximal order computations.

In the naive approach, we directly compute maximal orders of the corestriction of the algebra $A = B_1 \otimes B_2$. By Proposition 5.1, this maximal order computation has a cost $O(n^{18})$. The optimised approach first computes maximal orders $\mathcal{O}_1$ (resp. $\mathcal{O}_2$) of B_1 (resp. B_2), which has a cost $O(n^{\frac{9}{2}})$. We must then compute a maximal overorder of $\mathcal{O}_1 \otimes_R \mathcal{O}_2$. By Proposition 5.3, the degree of the discriminant of this order is bounded by $2Cn^{\frac{1}{2}}$. Hence, by the second statement in Proposition 5.1, this computation has a cost in $O(n^{\frac{11}{2}})$. Finally, we must compute a maximal order in the corestriction of A. This time, using Proposition 5.5 we start from an order with discriminant of degree $O(n^2)$. This operation therefore has a cost $O(n^{12})$. This last call is by far the most expensive of the optimised computation. We record this result as

Proposition 5.6 *With notations as above, directly computing a maximal order in* $\mathrm{Cor}A$ *using Friedrichs' algorithm [15] has complexity* $O(n^{18})$*, where n is the dimension of A.*

Using the approach described in this section, the complexity of this computation is reduced to $O(n^{12})$.

Remark 5.7 The algorithm from Theorem 4.5 uses the case where B_1 and B_2 are quaternion algebras. That is, $n = 4$. Therefore, some multiplicative constants (powers of 2 coming from the difference between dimension over $\mathbb{F}_q(t)$ and dimension over K) that disappear with the O notation still have a non negligible influence in the cost of the computation for the optimised method. In Subsection 6.3, we give concrete timing comparisons to show that the optimised method still remains more efficient than the naive approach, even with this additional overhead.

6 Implementation

In this section we present our implementation[2] of Algorithm 1 in Magma. This includes an implementation of the main algorithm from [26] for computing an explicit isomorphism of a central simple algebra to a matrix algebra. This implementation, which is also used in [5] (but in that case only on quaternion algebras), may be of independent interest.

We stress that due to the impracticality of algorithms for maximal order computation in algebras of dimension 256, our implementation of Algorithm 1 currently does not terminate in reasonable time. This highlights the interest of improving the results of [26, Sect. 3] and [15], as the existence of a more efficient algorithm for this task would render our own algorithm practical. We stress that any algorithm for maximal order computation with complexity depending on the discriminant of a starting order would benefit from the optimisation described in Subsection 5.2.

In the first subsection, we detail the subroutines we implement for Algorithm 1, and in the second subsection we give results of computational experiments.

6.1 Implementation details

For clarity of exposition, we present as Algorithm 1 a succinct pseudo-code description of the main function in our implementation of the algorithm from Theorem 4.5.

```
Input: $(B_1, B_2)$ two quaternion algebras defined on a quadratic field $L$ over $K = \mathbb{F}_q(t)$,
       with $q$ odd.
Output: An $L$-algebra isomorphism $B_1 \to B_2$.
$A \leftarrow B_1 \otimes_L B_2$;
$z, s \leftarrow$ InvolutionSecondKind($A$);
if $z = 0$ then
    $A_K \leftarrow$ Descent($A, s$) ;
    $z \leftarrow$ ZeroDivisor($A_K$);
end
$e \leftarrow$ RankOneIdempotent($A, z$);
return IsomorphismFromIdempotent($B_1$, $B_2$, e)
```

Algorithm 1: Main algorithm

We now detail our implementation of the subroutines in Algorithm 1. In what follows, L will be a quadratic extension of $\mathbb{F}_q(t)$.

- Tensor product computation is straightforward: one defines the algebra of dimension 16 over L, with basis $(b_{1,i} \otimes b_{2,j})_{1 \le i,j \le 4}$. The structure constants of $A = B_1 \otimes B_2$ are then products of the structure constants of A and B. We also construct the canonical injections from B_1 and B_2 to $B_1 \otimes B_2$. These maps are useful to give a succinct description of the conjugation involution over $B_1 \otimes B_2$ and to compute a basis of $O_1 \otimes O_2$, where O_1 and O_2 are maximal orders in B_1 and B_2.
- *Descent* Given an L-algebra A and a semi-linear algebra automorphism f, we return the K-subalgebra of elements of A fixed by f. We also compute low discriminant orders in this subalgebra by taking the fixed points of maximal orders of A if such orders are known. The only subtlety regarding the implementation is that in order to

[2] https://github.com/QuaternionIsomorphisms/QuaternionIsomorphisms/

make it efficient in Magma, the map f must be defined on a K-vector space representing the algebra A, since it is only semi-linear over L.

- *Corestriction* Computing the corestriction of an L-algebra A is a straightforward application of Proposition 2.10. We apply the non-trivial $\mathbb{F}_q(t)$-automorphism σ of L to the structure constants of A to compute A^σ, and a map between A and A^σ. Then maximal orders of A are computed, and from them we directly obtain maximal orders of A^σ. The algebra $A \otimes A^\sigma$ and its maximal orders are computed as described above. The switch map is then computed in a straightforward manner using maps $A \to A^\sigma$, $A \to A \otimes A^\sigma$ and $A^\sigma \to A \otimes A^\sigma$. We then apply the Descent subroutine to $A \otimes A^\sigma$ and the switch map to obtain the corestriction of A, orders with small discriminant and a map from the corestriction to $A \otimes A^\sigma$.
- *InvolutionSecondKind* This is Algorithm 3.2. Details of the computation of the corestriction are given above. Once the corestriction is computed, we compute a rank one idempotent e. Then $1-e$ generates a maximal right ideal I of B. We therefore compute the ideal generated by $1-e$ in $A \otimes A^\sigma$. The rest is a straightforward implementation of Algorithm 3.2.
- RankOneIdempotent when $A \cong \mathcal{M}_n(K)$: This is the main algorithm from [25, Sect. 4]. This algorithm uses many subroutines: we implement lattice reduction algorithms described in [26, Sect. 2] and [31, Sect. 1], and the computation of the Wedderburn-Malcev complement of a finite algebra following [6, Sect. 3]. The only remaining technical part is then to compute the intersection of maximal orders in A following [26, Lemma 25], and to express its structure constants as an algebra over $\mathbb{F}_q$.
- ZeroDivisor when $A \cong \mathcal{M}_n(D)$, with D a division quaternion algebra over K: Following [18, Theorem 18], we compute local indices of A and use this information to construct a quaternion algebra D' isomorphic to D, and then a representation of $\mathcal{M}_m(D')$ with structure constants. We then use the RankOneIdempotent subroutine described above and the IsomorphismFromIdempotent subroutine described below to compute an isomorphism $A \cong \mathcal{M}_m(D')$ and return a zero divisor. Note that the hypothesis from [18, Theorem 18] on the splitting places of A is not needed here since we restrict to the case that D is a quaternion algebra, and we therefore only need to compute local indices instead of Hasse invariants.
- RankOneIdempotent when $A \cong \mathcal{M}_4(L)$ and a zero divisor z is given: Following the discussion in the proof of Theorem 4.5, we compute e, the left unit of the right ideal zA. If z has rank 1 or 3, we are done as per the discussion. If z has rank 2, we apply the algorithm from [27, Proposition 43] to the split quaternion algebra eBe.
- *IsomorphismFromIdempotent* Given a rank one idempotent in algebra $A = B_1 \otimes B_2^{op}$, we compute an explicit isomorphism $B_1 \cong B_2^{op}$. Note that we in fact computed $A = B_1 \otimes B_2$, but since B_2 is a quaternion algebra, the conjugation gives an explicit isomorphism $B_2 \cong B_2^{op}$. This is an implementation of the algorithm given by [24, Corollary 10].

6.2 Computation examples

We present here two computations which employ some of the subroutines used in our implementation of Algorithm 1. For both computation, we exhibit here the input used and the result that was found. The reader interested in seeing intermediate steps of the computation may find these details in the examples/output directory of our implementation's repository.

6.2.1 Spliting a matrix algebra

We demonstrate our implementation of the main algorithm from [26]. In our algorithm, it is used for splitting the corestriction in the InvolutionSecondKind procedure. Execution in that case is not tractable because of the time needed to compute maximal orders is such a large algebra. We illustrate the method by recovering an explicit isomorphism to a matrix algebra for a 9-dimensional algebra over $\mathbb{F}_5(t)$.

We generate a split algebra by taking a random basis of the matrix algebra $\mathcal{M}_3(\mathbb{F}_5(t))$ and computing structure constants corresponding to this basis. We then discard the explicit basis and use the structure constants to generate a degree 3 central simple algebra. For this example, we get A from the following basis of $\mathcal{M}_3(\mathbb{F}_5(t))$:

$$\begin{pmatrix} \frac{4t}{t+2} & 0 & \frac{1}{t+3} \\ 0 & 0 & \frac{2}{t^2+3t+3} \\ 3 & 2t+1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & \frac{2t+1}{t} & 0 \\ 0 & 0 & 3t \\ 0 & 0 & 2t^2+t+2 \end{pmatrix},$$

$$\begin{pmatrix} 2t+3 & 0 & 0 \\ \frac{2}{t+3} & \frac{4}{t^2+2} & 0 \\ 0 & 0 & \frac{4}{t+1} \end{pmatrix}, \begin{pmatrix} 0 & 0 & 2t^2+2t+3 \\ 0 & 0 & 0 \\ \frac{3}{t^2+3} & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 4t^2+t+1 \\ 2t+4 & \frac{4}{t+2} & 2t+4 \\ 0 & 0 & \frac{3t^2}{t+4} \end{pmatrix},$$

$$\begin{pmatrix} 0 & \frac{t+3}{t^2+t+1} & 2t^2+t+3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 3 & 0 & 0 \\ 2 & 0 & 0 \\ 0 & 0 & \frac{4}{t^2+3t+3} \end{pmatrix}, \begin{pmatrix} 0 & \frac{t+3}{t+1} & 0 \\ \frac{4t+4}{t^2+1} & t+4 & \frac{4t+3}{t+3} \\ 0 & 0 & 3 \end{pmatrix}$$

We then compute a rank one idempotent element $e \in A$. We get an element with the following coordinates:

$$e = \begin{pmatrix} 0 \\ \frac{t^{23}+2t^{22}+4t^{18}+4t^{15}+2t^{14}+t^{13}+3t^{12}+2t^{11}+4t^{10}+3t^9+2t^8+3t^7+4t^6+t^5+t^4+t^3+2t^2+3t+2}{t^{25}+3t^{24}+4t^{23}+t^{22}+3t^{21}+t^{20}+3t^{19}+4t^{18}+3t^{16}+3t^{15}+t^{14}+t^{13}+3*t^{12}+3*t^9+3*t^8+2*t^7+3*t^6+3*t^5+3*t^4+2*t^3+3*t+3} \\ \frac{t^{18}+3t^{17}+2t^{16}+t^{14}+2t^{13}+t^{11}+2t^{10}+t^9+3t^6+2t^5+3t^4+4t^3+4t^2+4}{t^{22}+4t^{21}+4t^{20}+t^{19}+t^{18}+t^{17}+2t^{16}+4t^{15}+3t^{14}+4t^{13}+3t^{12}+4t^{11}+t^{10}+4t^9+3t^8+4t^7+3t^6+t^5+4t^4+4t^3+3t^2+3t+1} \\ \frac{3t^{22}+4t^{19}+t^{18}+4t^{17}+4t^{15}+2t^{13}+4t^{12}+t^{11}+t^{10}+4t^9+t^8+4t^7+t^6+t^5+t^4+3t^3+2t^2+4}{t^{23}+4t^{22}+4t^{20}+2t^{19}+t^{18}+3t^{17}+4t^{16}+t^{14}+4t^{13}+2t^{12}+t^{11}+3t^{10}+t^8+4t^7+4t^6+4t^5+t^3+4t^2+3t+1} \\ \frac{3t^{25}+t^{24}+4t^{23}+3t^{22}+2t^{20}+t^{18}+2t^{17}+t^{16}+4t^{15}+2t^{14}+4t^{12}+2t^{11}+2t^{10}+t^9+4t^6+2t^5+3t^3+4t^2+2t+3}{t^{25}+3t^{24}+4t^{23}+t^{22}+3t^{21}+t^{20}+3t^{19}+4t^{18}+3t^{16}+3t^{15}+t^{14}+t^{13}+3t^{12}+3t^9+3t^8+2t^7+3t^6+3t^5+3t^4+2t^3+3t+3} \\ \frac{2t^{18}+3t^{17}+4t^{16}+t^{15}+3t^{14}+3t^{12}+t^{11}+4t^8+3t^7+3t^6+t^5+t^4+3t+3}{t^{21}+4t^{20}+4t^{19}+3t^{17}+t^{16}+3t^{14}+3t^{12}+4t^{11}+4t^{10}+2t^9+4t^8+3t^7+2t^6+t^5+2t^4+3t^3+3t^2+3t+1} \\ \frac{2t^{22}+t^{21}+t^{20}+3t^{18}+2t^{17}+2t^{16}+3t^{15}+4t^{14}+t^{13}+2t^{12}+t^{10}+4t^9+2t^6+3t^5+t^4+4t^3+2}{t^{22}+t^{21}+t^{20}+3t^{19}+2t^{18}+2t^{16}+3t^{15}+4t^{14}+2t^{13}+2t^{12}+3t^{11}+2t^{10}+3t^9+4t^8+2t^7+2t^6+4t^4+2t^3+2t^2+2t} \\ \frac{2t^{20}+t^{19}+2t^{17}+2t^{16}+t^{15}+4t^{14}+t^{12}+2t^{11}+4t^{10}+2t^9+3t^6+3t^5+2t^4+t^3+t^2+3}{t^{23}+4t^{22}+4t^{20}+2t^{19}+t^{18}+3t^{17}+4t^{16}+t^{14}+4t^{13}+2t^{12}+t^{11}+3t^{10}+t^8+4t^7+4t^6+4t^5+t^3+4t^2+3t+1} \\ \frac{3t^{18}+4t^{17}+t^{14}+t^{13}+4t^{12}+3t^{11}+2t^{10}+4t^8+4t^6+2t^5+4t^3+2t^2+2}{t^{22}+2t^{21}+t^{20}+2t^{19}+3t^{18}+3t^{16}+3t^{15}+4t^{14}+3t^{13}+3t^{12}+t^{11}+4t^{10}+t^7+2t^6+4t^4+2t^3+2t^2+3} \end{pmatrix}$$

6.2.2 Finding a rational quaternion subalgebra

The most novel part of our algorithm is the descent method described in Sect. 3. Here we illustrate the method applied to an algebra $A \cong \mathcal{M}_2(L)$ where $L = \mathbb{F}_3(t)(\sqrt{t^3+2t^2+2t+2})$. For the rest of this example, we denote $\theta = \sqrt{t^3+2t^2+2t+2}$.

As above, we generate A from a random basis of $\mathcal{M}_2(L)$, by computing the corresponding structure constants. For this example, we use the following matrices as basis elements:

$$\begin{pmatrix} (t^2+1)\theta + \frac{2t^2+t+1}{t^2+1} & 0 \\ \frac{2}{t^2+2t+2}\theta + \frac{t^2+2t+1}{t+2} & 0 \end{pmatrix}$$
$$\begin{pmatrix} \theta + 2t + 2 & \frac{2}{t+2}\theta + \frac{1}{t} \\ 0 & \frac{2}{t+2}\theta + 1 \end{pmatrix}$$
$$\begin{pmatrix} \theta + \frac{t^2+2}{t^2+1} & \frac{2}{t}\theta + \frac{2t+2}{t+2} \\ 0 & 2\theta + 1 \end{pmatrix}$$
$$\begin{pmatrix} 0 & \frac{t^2+2t+1}{t^2+t+2}\theta + \frac{1}{t^2+2} \\ 0 & 0 \end{pmatrix}$$

We then apply the method suggested by Proposition 3.4, and compute $B \subset A$, a quaternion algebra over $\mathbb{F}_3(t)$. The subalgebra B is given by a K-bases formed of the following elements:

$$\begin{pmatrix} 1 \\ 0 \\ \frac{t^4+2t^2+1}{t^2}\theta + \frac{t^{12}+2t^{11}+2t^8+t^7+2t^4+t^3+2t^2+1}{t^5+t^3+2t^2} \\ \frac{t^{29}+t^{27}+2t^{24}+t^{23}+t^{22}+2t^{21}+t^{20}+t^{18}+t^{16}+t^{14}+2t^{13}+2t^{12}+2t^{11}+2t^9+2t^8+2t^7+2t^5+t+2}{t^{21}+2t^{20}+2t^{19}+t^{18}+t^{17}+2t^{15}+2t^{14}+t^{13}+2t^{11}+t^9+t^8+2t^7+t^6+2t^3+2t^2}\theta + \frac{t^{33}+t^{32}+t^{31}+t^{30}+t^{29}+t^{28}+t^{27}+2t^{24}+2t^{21}+t^{19}+2t^{16}+2t^{14}+t^{13}+2t^{12}+2t^{11}+2t^6+2t^5+t^3+2t^2+2}{t^{25}+2t^{24}+2t^{22}+t^{21}+2t^{20}+2t^{19}+t^{18}+t^{15}+2t^{14}+2t^{12}+t^{10}+t^9+2t^8+t^7+2t^6+2t^5+t^3} \end{pmatrix},$$

$$\begin{pmatrix} \theta \\ 0 \\ \frac{2t^2+t+1}{t^2}\theta + \frac{t^{12}+2t^{11}+t^{10}+t^9+2t^8+2t^7+t^6+2t^4+t^2+1}{t^5+t^3+2t^2} \\ \frac{t^{30}+t^{29}+2t^{28}+t^{27}+2t^{26}+t^{24}+t^{22}+t^{21}+t^{19}+t^{18}+t^{17}+t^{16}+2t^{14}+2t^{13}+2t^{11}+2t^9+2t^8+t^7+2t^5+2t+2}{t^{22}+t^{20}+2t^{18}+t^{17}+2t^{16}+t^{15}+t^{13}+2t^{12}+2t^{11}+t^{10}+2t^9+t^6+2t^4+t^3+2t^2}\theta + \frac{t^{33}+t^{32}+t^{31}+t^{30}+2t^{28}+2t^{27}+t^{26}+t^{24}+t^{23}+2t^{22}+2t^{21}+2t^{19}+2t^{18}+t^{17}+2t^{16}+2t^{15}+2t^{14}+t^{12}+2t^{10}+2t^9+t^7+t^6+2t^5+t^4+2t^3+2}{t^{25}+2t^{24}+2t^{22}+t^{21}+2t^{20}+2t^{19}+t^{18}+t^{15}+2t^{14}+2t^{12}+t^{10}+t^9+2t^8+t^7+2t^6+2t^5+t^3} \end{pmatrix},$$

$$\begin{pmatrix} 0 \\ 1 \\ \frac{t^3+t^2+t+1}{t^3+2t^2}\theta + \frac{t^8+t^6+2t^5+t^4+t+2}{t^3+2t^2} \\ \frac{t^{28}+t^{27}+t^{26}+2t^{25}+t^{24}+t^{23}+2t^{21}+2t^{20}+t^{18}+2t^{17}+2t^{16}+2t^{15}+t^{14}+t^{12}+t^{10}+t^9+t^7+2t^4+2t^3+t^2+2t+2}{t^{22}+t^{21}+2t^{19}+2t^{17}+2t^{16}+2t^{14}+2t^{13}+2t^{12}+t^{11}+t^{10}+t^8+2t^7+2t^6+2t^4+t^2}\theta + \frac{t^{29}+2t^{28}+t^{27}+2t^{26}+2t^{25}+t^{24}+2t^{23}+t^{22}+t^{20}+t^{19}+2t^{18}+2t^{17}+2t^{16}+t^{14}+t^{11}+t^9+2t^8+2t^7+2t^5+2t^3+2t^2+1}{t^{23}+t^{22}+2t^{20}+2t^{18}+2t^{17}+2t^{15}+2t^{14}+2t^{13}+t^{12}+t^{11}+t^9+2t^8+2t^7+2t^5+t^3} \end{pmatrix},$$

Table 1 Running time for computing maximal orders in the corestriction of degree 2 matrix algebras

Naive version	Optimised version
95.180	7.160
1128.870	46.990
2338.350	155.520

Table 2 Runtime for the RankOneIdempotent subroutine

n	Maximal $\mathbb{F}_{17}[t]$-order computation	Maximal R-order computation	Running time
2	4.690	0.390	5.510
3	7245.840	401.000	7706.890

and

$$\begin{pmatrix} 0 \\ \theta \\ \frac{2t^2+2}{t}\theta+ \\ \frac{2t^8+t^7+2t^6+t^5+2t^2+2}{t^2+2t} \\ \frac{2t^{27}+2t^{26}+2t^{25}+2t^{23}+2t^{21}+t^{19}+2t^{18}+t^{16}+t^{15}+2t^{12}+t^{11}+2t^{10}+t^7+t^6+2t^4+2t^2+t+2}{t^{20}+2t^{19}+2t^{18}+t^{17}+t^{16}+2t^{14}+2t^{13}+t^{12}+2t^{10}+t^8+t^7+2t^6+t^5+2t^2+2t}\theta+ \\ \frac{2t^{29}+2t^{28}+t^{27}+t^{26}+t^{24}+t^{20}+t^{17}+t^{15}+t^{14}+2t^{13}+2t^{12}+t^{10}+2t^9+t^7+2t^5+2t^3+2t+1}{t^{22}+t^{21}+2t^{19}+2t^{17}+2t^{16}+2t^{14}+2t^{13}+2t^{12}+t^{11}+t^{10}+t^8+2t^7+2t^6+2t^4+t^2} \end{pmatrix}$$

6.3 Running times comparisons

In Table 1 we give running times for the task of computing maximal orders in the corestriction of a degree 2 matrix algebra over $K = \mathbb{F}_q(t)(\sqrt{D})$, with D a polynomial of degree 2. The running time includes the computation of the corestriction itself. Running times are given in seconds.

The *naive version* column corresponds to the running time of the direct approach to the task, and the *optimised version* column refers to using the methods described in Subsection 5.2.

In Table 2 we give running times for executions of the RankOneIdempotent subroutine from Algorithm 1. We execute it on a $\mathbb{F}_{17}(t)$-algebra A isomorphic to $\mathcal{M}_n(\mathbb{F}_{17}(t))$. We recall that this subroutine is an implementation of the main algorithm from [26]. It begins with the computation of a maximal $\mathbb{F}_{17}[t]$-order and a maximal R-order of A, where R is the valuation ring for the degree valuation. That is, R is the ring of elements in $\mathbb{F}_{17}(t)$ that have a denominator of higher degree than their numerator.

Running times are again given in seconds. We also give the running time of the maximal order computations.

The results from Table 2 show that the complexity bottleneck of this subroutine is indeed the computation of maximal orders. We recall that our use case involves running this computation on algebras isomorphic to $\mathcal{M}_{16}(\mathbb{F}_q)$. We conclude that our algorithm would be made practical by the discovery of a fast algorithm for computing maximal orders in separable algebras over $\mathbb{F}_q(t)$.

Acknowledgements

We would like to thank John Voight for helpful suggestions and comments on an earlier version of this manuscript.

Funding Open access funding provided by Eötvös Loránd University. Gergely Zábrádi was supported by the Rényi Institute Lendület Automorphic Research Group, by the NKFIH Research Grants FK-127906 and K-135885, and by Project ED 18-1-2019-0030 (Application-specific highly reliable IT solutions) under the Thematic Excellence Programme funding scheme. Péter Kutas was supported by EPSRC grant EP/V011324/1 and. Péter Kuta was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences and by the ÚNKP-22-5 New National Excellence Program. Péter Kutas was also supported by the Ministry of Innovation and Technology and the National Research, Development and Innovation Office within the Quantum Information National Laboratory of Hungary.

Data availability Data sharing not applicable to this article as no datasets were generated or analysed during the current study.,

Declarations

Conflict of interests

The author asserts that there are no conflicts of interest.

Author details

[1]University of Oxford, Oxford, UK, [2]Eötvös Loránd University and University of Birmingham, Birmingham, UK, [3]Faculty of Mathematics and Informatics, Institute of Mathematics, Vilnius University, Vilnius, Lithuania, [4]Eötvös Loránd University and Rényi Institute of Mathematics, Lendület "Automorphic" Research Group, Budapest, Hungary.

Received: 18 August 2022 Accepted: 1 September 2022 Published online: 1 October 2022

References

1. Böckle, G., Gvirtz, D.: Division algebras and maximal orders for given invariants. LMS J. Comput. Math. **19**(A), 178–195 (2016)
2. Bosma, W., Cannon, J., Playoust, C.: The magma algebra system I: the user language. J. Symb. Comput. **24**(3–4), 235–265 (1997)
3. Cohen, A.M., Ivanyos, G., Wales, D.B.: Finding the radical of an algebra of linear transformations. J. Pure Appl. Algebra **117**, 177–193 (1997)
4. Cremona, J., Fisher, T., O'Neil, C., Simon, D., Stoll, M.: Explicit n-descent on elliptic curves III. Algorithms. Math. Comput. **84**(292), 895–922 (2015)
5. Csahók, T., Kutas, P., Montessinos, M., and Zábrádi, G.: Finding nontrivial zeros of quadratic forms over rational function fields of characteristic 2. http://arxiv.org/abs/2203.04068 (2022)
6. de Graaf, W., Ivanyos, G., Küronya, A., Rónyai, L.: Computing Levi decompositions in Lie algebras. Appl. Algebra Eng. Commun. Comput. **8**, 291–303 (1997)
7. De Graaf, W.A., Harrison, M., Pílniková, J., Schicho, J.: A Lie algebra method for rational parametrization of Severi-Brauer surfaces. J. Algebra **303**(2), 514–529 (2006)
8. Eberly, W.: Decompositions of algebras over $\mathbb{R}$ and $\mathbb{C}$. Comput. Complex. **1**(3), 211–234 (1991)
9. Eisenträger, K., Hallgren, S., Lauter, K., Morrison, T., and Petit, C.: Supersingular isogeny graphs and endomorphism rings: reductions and solutions. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 329–368. Springer (2018)
10. Fisher, T.: Explicit 5-descent on elliptic curves. Open Book Ser. **1**(1), 395–411 (2013)
11. Fisher, T.: Higher descents on an elliptic curve with a rational 2-torsion point. Math. Comput. **86**(307), 2493–2518 (2017)
12. Fisher, T., Newton, R.: Computing the Cassels-Tate pairing on the 3-Selmer group of an elliptic curve. Int. J. Number Theory **10**(07), 1881–1907 (2014)
13. Ford, T.J.: Separable Algebras Volume 183 of Graduate Studies in Mathematics. American Mathematical Society, Providence (2017)
14. Friedl, K., Rónyai, L.: Polynomial time solutions of some problems of computational algebra. In: Proceedings of the seventeenth annual ACM symposium on Theory of computing, pp. 153–162 (1985)
15. Friedrichs, C.: Berechnung von Maximalordnungen über Dedekindringen. PhD thesis, Technische Universität Berlin (2000)
16. Gille, P., Szamuely, T.: Central Simple Algebras and Galois Cohomology, vol. 165. Cambridge University Press, Cambridge (2017)
17. Granath, H.: Lattices and orders in quaternion algebras with involution. J. Algebra **304**(2), 927–949 (2006)
18. Gómez-Torrecillas, J., Kutas, P., Lobillo, F., Navarro, G.: Primitive idempotents in central simple algebras over $\mathbb{F}_q(t)$ with an application to coding theory. Finite Fields Appl. **77**, 101935 (2022)

19. Gómez-Torrecillas, J., Lobillo, F., Navarro, G.: A new perspective of cyclicity in convolutional codes. IEEE Trans. Inf. Theory **62**(5), 2702–2706 (2016)
20. Hanke, T.: The isomorphism problem for cyclic algebras and an application. In: Proceedings of the 2007 international symposium on symbolic and algebraic computation, pp. 181–186 (2007)
21. Ivanyos, G.: Algorithms for algebras over global fields. PhD thesis, Hungarian Academy of Sciences (1996)
22. Ivanyos, G., Rónyai, L.: Finding maximal orders in semisimple algebras over $\mathbb{Q}$. Comput. Complex. **3**(3), 245–261 (1993)
23. Ivanyos, G., Rónyai, L., Szántó, Á.: Decomposition of algebras over $\mathbb{F}_q(x_1, \ldots, x_m)$. Appl. Algebra Eng. Commun. Comput. **5**(2), 71–90 (1994)
24. Ivanyos, G., Rónyai, L., Schicho, J.: Splitting full matrix algebras over algebraic number fields. J. Algebra **354**(1), 211–223 (2012)
25. Ivanyos, G., Lelkes, Á., Rónyai, L.: Improved algorithms for splitting full matrix algebras. JP J. Algebra Number Theory Appl. **28**(2), 141–156 (2013)
26. Ivanyos, G., Kutas, P., Rónyai, L.: Computing explicit isomorphisms with full matrix algebras over $\mathbb{F}_q(x)$. Found. Comput. Math. **18**(2), 381–397 (2018)
27. Ivanyos, G., Kutas, P., Rónyai, L.: Explicit equivalence of quadratic forms over $\mathbb{F}_q(t)$. Finite Fields Appl. **55**, 33–63 (2019)
28. Janusz, G.J.: Tensor products of orders. J. Lond. Math. Soc. **20**(2), 186–192 (1979)
29. Knus, M.-A., Merkurjev, A., Rost, M., Tignol, J.-P.: The book of involutions. AMS Colloq. Publ. **44**, 17 (1998)
30. Kutas, P.: Splitting quaternion algebras over quadratic number fields. J. Symb. Comput. **94**, 173–182 (2019)
31. Lenstra, A.K.: Factoring multivariate polynomials over finite fields. J. Comput. Syst. Sci. **30**(2), 235–248 (1985)
32. Reiner, I.: Maximal Orders. London Mathematical Society Monographs. Oxford University Press, Oxford (2003)
33. Rónyai, L.: Simple algebras are difficult. In: Proceedings of the nineteenth annual ACM symposium on Theory of computing, pp. 398–408 (1987)
34. Rónyai, L.: Computing the structure of finite algebras. J. Symb. Comput. **9**(3), 355–373 (1990)
35. Schwinning, N.: Ein Algorithmus zur Berechnung von Divisionsalgebren über $\mathbb{Q}$ zu vorgegebenen Invarianten. Master's thesis, Universität Duisburg-Essen, Germany (2011)
36. Voight, J.: Identifying the matrix ring: algorithms for quaternion algebras and quadratic forms. In: Quadratic and higher degree forms, pp. 255–298. Springer (2013)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.