Decomposition of Finite-Dimensional Matrix Algebras over $\mathbb{F}_q(y)$

by

Ruitong Huang

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2010

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Computing the structure of a finite-dimensional algebra is a classical mathematical problem in symbolic computation with many applications such as polynomial factorization, computational group theory and differential factorization. We will investigate the computational complexity and exhibit new algorithms for this problem over the field $\mathbb{F}_q(y)$, where $\mathbb{F}_q$ is the finite field with q elements.

A finite-dimensional vector space A over a field F is called a finite-dimensional *associative algebra* over F, if A is equipped with a binary associative F-bilinear operation (which is always called multiplication and not necessarily commutative) and the distributive law holds with respect to the addition of linear space and the multiplication. The *matrix algebra* is the subalgebra of the matrix ring $F^{m \times m}$ with the identity matrix.

For an algebra A, there exists a largest nilpotent ideal $\mathrm{Rad}(A)$, called the *radical* of A in every finite dimensional algebra A. $\mathrm{Rad}(A)$ is the set of all strongly nilpotent elements (where an element α is said to be *strongly nilpotent* if for any $\beta \in A$, $\alpha\beta$ is nilpotent). If $\mathrm{Rad}(A) = (0)$ we call the algebra A *semisimple.* So the factor algebra $A/\mathrm{Rad}(A)$ is semisimple. A is called *simple* if A has no proper nonzero ideal. Semisimple algebras admit a very nice structure theorem which is also due to Wedderburn [34].

Theorem 1. *[Wedderburn] Suppose that A is a finite-dimensional semisimple algebra over the field F. Then A can be expressed as a direct sum of simple algebras.*

$$A = A_1 \oplus A_2 \oplus ... \oplus A_t,$$

where $A_1, A_2, \ldots, A_t$ are the minimal nontrivial ideals of A. Each A_i is isomorphic to some full matrix algebra $M_{n_i}(F_i)$, where F_i is an extension division ring of F for $1 \le i \le t$. Such decomposition is called Wedderburn decomposition.

In this thesis we will first present a new probabilistic algorithm for Wedderburn decomposition. The Wedderburn decomposition of separable algebra is solved with almost nearly optimal algorithms by Eberly and Giesbrecht [8, 7]. However, when it comes to the algebra over the field $\mathbb{F}_q(y)$, it becomes non-separable. Ivanyos et al. present a polynomial-time algorithm for Wedderburn decomposition over $\mathbb{F}_q(y)$, but it is not acceptable because of large exponent [24]. We will exhibit a new probabilistic algorithm of Monte Carlo type for decomposition of general semisimple matrix algebras. The idea is inspired by Eberly and Giesbrecht [8, 7]: Demonstrate the large probability to pick up a "good" element randomly, use it to compute the "good" idempotents and then decompose the algebra. Our algorithm is more efficient and easier to implement than the algorithm of Ivanyos et al. [24].

The second part of this thesis is a new probabilistic algorithm for computing the radical of a finite-dimensional algebra. Fröhlich and Shepherdson [10] proved that in general this is algorithmically undecidable over a general computable field. But there are some efficient results over specific fields such as the finite field $\mathbb{F}_q$. The latest paper about computing the radical of the finite-dimensional algebras over $\mathbb{F}_q(y)$ is also developed by Ivanyos et al. [24], which is polynomial-time but

with large exponent too. We give a faster Monte Carlo algorithm in this paper by generalizing the idea of Ivanyos et al. [23, 24] to non-separable cases. We overcome the difficulty of degree explosion by introducing a new decomposition, which we call raw decomposition and has not been discussed in the previous papers. It takes advantage of the idempotents to transform the general algebra into primary algebras.

In the third part of this thesis, we discuss how to modify our algorithms for the field $\mathbb{F}_q(y)$ when q is small. Note that in our algorithms, we have some conditions on the size of q to guarantee the probability of correctness of the algorithms. However, we would like our algorithm to work for general $\mathbb{F}_q(y)$. We present methods to remove the restrictions on q from our algorithms, adapting the algorithms to the general field $\mathbb{F}_q(y)$.

The last chapter of this thesis is about two important questions which remain unsolved as future work. We will discuss the difficulties we encounter when working on these questions. The first open question is how to make our algorithms be of Las Vegas type. Note that our algorithms for decomposition of algebras are of Monte Carlo type. However, we always hope to see that the output is certified correct. The other problem is brought by the inseparability of our algebra. Recall by Wedderburn's Structure Theorem 1 we know each A_i is isomorphic to some full matrix algebra over an extended division ring of F for $1 \leq i \leq t$. Our algorithm does not compute this isomorphic mapping. When $F = \mathbb{F}_q$, i.e. the semisimple algebra is separable, this problem is solved by Friedl et al. [9] and Eberly et al. [7], by a deterministic algorithm and a probabilistic one respectively. However, when it comes to the infinite field $\mathbb{F}_q(y)$ this problem, to our best knowledge, is still open.

Acknowledgments

First, I would like to express my deepest gratitude and sincere appreciations to my supervisor, Mark Giesbrecht, for suggestion of the thesis topic and invaluable explanations, guidance and advice during my study in Waterloo. This thesis would have never been possible without his continued support. And it is indeed an honor to be his student.

I would also like to thank Reinhold Burger, Daniel Ivan and Myung Sub Kim for their selfless help. Thank all the SCG people. I enjoy an excellent life in this lab. My thanks also go to my thesis committee of George Labahn and Ilias Kotsireas for the time spent in reviewing this thesis.

I had a number of helpful discussions with Wei Zhou and Jizhan Hong, whom I would like to thank here for their lots of great suggestions. And I think I should thank Wei Zhou again for the very nice tutorial and continuous help on LyX. To my lovely roommates in Waterloo, Ying Yan, Yujie Zhong, Liqun Diao and Pan Pan Cheng, thanks very much for the happy time and nice meals.

Finally, I thank my parents for their unwavering support and the warm atmosphere they give me; and the lovely Yasi Jiang. You know nothing about the topic of my thesis, but you indeed know how to keep me laughing all the way.

Dedication

This is dedicated to my parents.

Contents

Chapter 1

Introduction

In this thesis we will address two problems in the structure theory of associative matrix algebras over $\mathbb{F}_q(y)$: computing the radical and computing the Wedderburn decomposition, where $\mathbb{F}_q(y)$ is the field of fractions of the polynomial ring $\mathbb{F}_q[y]$. Ivanyos et al. have given polynomial algorithms for these two problems [22, 24], but their costs have large exponents (which have not been completely determined). Our goal is to present new efficient algorithms or improvement of existing algorithms. In order to simplify the discussion, we assume that our algebra contains the identity matrix, and we only calculate the soft order of the complexity, i.e. $O^{\sim}(n^t) = O(n^t)(\log n)^{O(1)}$.

The structures of associative algebras is clarified. Given an algebra, the largest nilpotent ideal is called the *radical* and denoted by $\mathrm{Rad}(A)$. If $\mathrm{Rad}(A) = (0)$ we call the algebra A *semisimple*. Every finite-dimensional algebra A has a largest nilpotent ideal $\mathrm{Rad}(A)$ which satisfies that $A/\mathrm{Rad}(A)$ is a semisimple algebra. $\mathrm{Rad}(A)$ is the set of all the strongly nilpotent elements, where an element α in A is called *strongly nilpotent* if $\alpha\beta$ is nilpotent for any $\beta \in A$. An algebra A is called *simple* if A has no proper nonzero ideal. A main theorem is called Wedderburn's Structure Theorem [34] which clarifies the structure of the semisimple algebras.

Theorem 2. *[Wedderburn] Suppose that A is a finite-dimensional semisimple algebra over a field F. Then A can be expressed as a direct sum of simple algebras:*

$$A = A_1 \oplus A_2 \oplus ... \oplus A_t,$$

where $A_1, A_2, \dots, A_t$ are the minimal nontrivial ideals of A. Each A_i is isomorphic to some full matrix algebra $M_{n_i}(F_i)$, where F_i is an extension ring of F which is a division ring for $1 \le i \le t$. Note that F_i is not necessarily commutative or separable here.

We now make specific the computational description of our problems. The input of our algorithms is the integral basis, $\{a_1, a_2, \dots, a_n\} \subset \mathbb{F}_q[y]^{m\times m}$, of the (semisimple) matrix algebra $A \subset \mathbb{F}_q(y)^{m\times m}$. The output of our algorithm for the problem of computing the radical and the Wedderburn decomposition is a basis of $\mathrm{Rad}(A)$ and those of all the simple components $\{A_1, \dots, A_t\}$.

We examine algorithms which perform exact computations; they accept symbolic representations of an input and return symbolic exact representations of outputs. That means there should be a degree bound on y for the input in $\mathbb{F}_q[y]$, denoted by Δ. In the rest of this chapter we will discuss the model of generating random elements for the probabilistic algorithm.

The exact complexities of some useful symbolic computations are presented in Chapter 2. Particularly, we discuss all the preliminaries in Chapter 2 such as polynomial factorization, solving linear system, and so on. We also develop an efficient algorithm for computing the minimal polynomial of the matrix over $\mathbb{F}_q[y]$ in this chapter. The idea is similar to that of Giesbrecht and Storjohann [18].

Chapter 3 is the main part of this thesis which is devoted to the computation of matrix algebras. Specifically we show how to compute bases of simple components of a semisimple matrix algebra and a basis of the radical of a matrix algebra. We build on the work of Ivanyos et al. [23, 22, 24] and Eberly and Giesbrecht [8]. After reviewing their algorithms and analyzing their complexities, we present an alternative Monte Carlo algorithm over the field $\mathbb{F}_q(y)$ where q is sufficiently large. In particular we prove that it is with large probability to select a "good" element randomly. For the computation of the Wedderburn decomposition the remaining work is much easier: factorizing bivariate polynomial and solving much smaller linear systems. For the computation of the radical, the key idea is reducing the degree bound for the method of Ivanyos [24]. We introduce a new decomposition to do this. We call it a raw decomposition and it has not been discussed before so far as we know. The result is still dependent on the size of q. In the last part of this chapter, we give a generalization of our algorithm. In particular, we present a way to make our algorithms fit the general $\mathbb{F}_q(y)$, i.e. no restriction on the size of q any more.

In the last chapter of this thesis, some possible future work is discussed. We mainly suggest two open questions which we think are important and solvable. We also state the difficulties we have encountered when working on these two questions.

To prove the correctness of our probabilistic algorithms, some technical conditions are required when selecting a random element from a vector space. Let U be a finite dimensional vector space over F. One common way of selecting a random element u from the vector space U over the field F is to first select a sufficiently large finite subset S of F and then select elements uniformly from the finite set S as the coordinates of u. The results to be presented in this thesis rely on the following bound on the number of zeros of a polynomial within a particular set, which is presented by Schwartz. [39]

Lemma 3. *[Schwartz-Zippel Lemma] Suppose $q \in F[x_1, x_2, \ldots, x_n]$ is a polynomial with total degree at most d and that q is not identically zero. Let $c > 0$, and suppose $S \subset F$ is a finite set with size at least cd. Then the number of elements of S^n which are zeros of q is at most $c^{-1}|S|^n$.*

In the procedure of selecting random elements from the algebra A independently, assume that $\{u_1, u_2, \ldots, u_s\}$ is a basis of A over $\mathbb{F}_q(y)$. Given $\delta > 0$, let Ω be a finite subset of $\mathbb{F}_q(y)$ with $|\Omega| \geq D/\delta$. We take $u = \alpha_1 u_1 + \alpha_2 u_2 + \ldots + \alpha_s u_s$, where the coefficients $\alpha_1, \alpha_2, \ldots, \alpha_s$ are drawn uniformly and independently from Ω. Then for every polynomial function $f \in \mathbb{F}_q(y)[x]$ of degree at most D with respect to x, the probability of $f(u) = 0$ is at most $D/|\Omega| \leq \delta$.

We say a probabilistic algorithm is of Las Vegas type if it returns a correct answer on any input with some constant probability (usually greater than 0.5) and otherwise it returns “failure”. A less demanding type of algorithm, called a Monte Carlo algorithm, returns the correct answer with some constant probability (usually greater than 0.5), while it may return either “failure” or incorrect answer otherwise. The principal way to measure the cost of a probabilistic algorithm is to give its deterministic complexity as well as its probability of returning “failure” or incorrect answer. Sometimes this is not convenient, especially when we repeat the algorithm several times to get a correct answer. An alternative measure is the expected complexity, i.e. the expected number of operations required before we get the correct answer. For example, if a Monte Carlo algorithm requires $O(n^c)$ operations with probability ϵ of returning “failure” or incorrect answer, then the expected complexity will be $O(\frac{n^c}{1-\epsilon})$. Obviously, the expected complexity is slightly less informative.

Chapter 2

Fundamental Symbolic Computations

In this chapter we will discuss details about the model of computation and representation of various fields. Computational models for several fields are discussed in Section 2.1. In Section 2.2 we describe an algorithm and its complexity for polynomial factorization over $\mathbb{F}_q[y]$. An algorithm to select the maximal linearly independent subset of vectors is shown in Section 2.3. An algorithm to solve linear systems efficiently is in Section 2.4. At the end of this chapter we will present a new algorithm to compute the minimal polynomial of a matrix over $\mathbb{F}_q[y]$. The algorithms in the last three sections are related to the cost of matrix multiplication over $\mathbb{F}_q[y]$.

Most of the material in this chapter is included for the sake of completeness and is standard or cited from other papers. One exception is the algorithm for computing the minimal polynomial given in the last section. The idea is inspired by the paper of Giesbrecht and Storjohann [18]: it is with large probability to select an element $\alpha \in \mathbb{F}_q$ such that the modular matrix $a \bmod y - \alpha$ keeps the same structure of its Frobenius form. Following this result and these ideas, we can also compute the Frobenius form of an integral matrix over $\mathbb{F}_q(y)$ and the corresponding transformation matrix.

We examine a number of probabilistic algorithms in this thesis. When calculating the cost of a Monte Carlo algorithm, we only focus on the complexity of a single try with error tolerance $\epsilon < 1$. Actually, dividing this complexity by $1 - \epsilon$ gives the expected complexity, i.e. the expected cost to get a correct answer.

In some cases an efficient algorithm for a problem P is difficult. Instead we show that it could be reduced to another problem Q efficiently, by providing an efficient algorithm for P which includes several instances of Q.

Let $M(n)$ denote the number of arithmetical operations over the ring R to multiply two polynomials of degree at most n. Using the practical standard algorithm, $M(n) = O(n^2)$. The algorithm of Schönhage and Strassen [38] for any field R, or the one of Cantor and Kaltofen [3] for any ring R, yields $M(n) = O(n \mathrm{log} n \mathrm{loglog} n)$.

If F is a field, and $f, g \in F[x]$ have degree at most n, then the division with remainder of f by g, i.e. finding $q, r \in F[x]$ such that $f = qg + r$ with $\mathrm{deg} r < \mathrm{deg} g$, will cost $O(M(n))$ operations in F. We can compute the greatest common divisor of f and g with $O(M(n)\text{log}n)$ operations in F [14].

Let $MM(n)$ denote the number of arithmetical operations over the ground field to calculate the product of two $n \times n$ matrices. Using the standard method we have $MM(n) = O(n^3)$ while using the fastest known algorithm $MM(n) = O(n^{2.376})$. We use ω to denote the best known exponent here, so $2 \leq \omega < 3$. This is also the complexity of many fundamental matrix operations, including computing the determinant, the matrix inverse, the rank, the characteristic polynomial as well as the Frobenius normal form [13].

2.1 Computations over Fields

In this section we specify our model of computation and measures of cost which are to be used later. All of our computation is based on the basic computation over $\mathbb{F}_p$. We treat basic operations $(+, -, \times, \div)$, element selection and zero test over $\mathbb{F}_p$ as unit operations which have the same cost. When we analyze the cost of an algorithm, we count the number of these operations used in this algorithm. We analyze the cost in the worst case.

The polynomial $f \in \mathbb{F}_p[x]$ of degree $n-1$ can be represented by an ordered sequence of elements $(a_0, a_2, \dots, a_{n-1}) \subset \mathbb{F}_p^n$. So the addition costs $O(n)$ and the complexity of multiplication is $O(M(n))$ over $\mathbb{F}_p$.

In general, the finite field $\mathbb{F}_q$, where p is a prime and $q = p^l$, can be represented by the isomorphism $\mathbb{F}_q \overset{\sim}{=} \mathbb{F}_p[x]/(f)$, where $f \in \mathbb{F}_p[x]$ is irreducible with degree l. Note that the size of an element of $\mathbb{F}_q$ is $O(\log\, q)$. From the conclusion above, the complexities of addition and multiplication are $O(\log\, q)$ and $O(M(\log\, q))$, respectively. In this thesis we also treat the basic operations over $\mathbb{F}_q$ as unit operations.

We consider the exact representation of elements and the cost of arithmetic for $\mathbb{F}_q(y)$. The rational function $\frac{f}{g}$ $(f, g \in \mathbb{F}_q[y],\ g \neq 0)$ is represented by the ordered pair (f, g), where $f = a_0 + a_1 y + \ldots + a_s y^s$ and $g = b_0 + b_1 y + \ldots + b_t y^t$ $(s \leq t)$ are relatively prime. The polynomial f (or g) is represented by an ordered sequence of elements, $(a_0, a_1, \dots, a_s) \in \mathbb{F}_q^s$ (or respectively $(b_0, b_1 \dots, b_t) \in \mathbb{F}_q^t$). The product of the elements of $\mathbb{F}_q(y)$ having representations of length n is twice as those of polynomials $\mathbb{F}_q[y]$, which uses $O(M(n))$ operations over $\mathbb{F}_q$, i.e. $O(M(n)\text{log}q)$ over $\mathbb{F}_p$. Reducing the result to a standard representation will cost $O(M(n))$ operations over $\mathbb{F}_q$, i.e. $O(M(n)\text{log}q)$ over $\mathbb{F}_p$. Doing addition, we first compute the least common multiple by the Fast Extended Euclidean Algorithm and then doing constant number of multiplications and additions in $\mathbb{F}_q[y]$. Its complexity is $O(M(n)\text{log } n)$ over $\mathbb{F}_q$ or $O(M(n)\text{log } n\text{log } q)$ over $\mathbb{F}_p$.

2.2 Factoring Bivariate Polynomials over $\mathbb{F}_q$

Algorithms for factoring polynomials have made dramatic progress over the past few decades. However there is no known universal algorithm for factorization over all fields. The first polynomial-time multivariate factorization algorithms over integer, rational number and finite fields are due to Kaltofen [25, 26, 27]. In this thesis we only consider the polynomial-time algorithms for bivariate polynomials over finite field. Let R denote a commutative ring, K a field, $R[x, y]$ the ring of polynomials in two variables over R. Also for any $f \in R[x, y]$, we denote by d its total degree and d_y (d_x) its degree with respect to the variable y (x respectively).

Many bivariate polynomials factorization algorithms follow the lifting approach. First, they specify one of the two variables at random. Then the resulting univariate polynomial is factored and its factors are lifted to a sufficiently high degree to get the exact result. The first two steps are classical, following Zassenhaus (1969, 1978), and the final step asks for a recombination procedure to get the factors of the original polynomial. In 2003, Gao [11] presented an algorithm for factoring bivariate polynomials via partial differential equations with a near quadratic running time: $O(N^{2.5})$, where $N = d_x d_y$ is the input size. Bostan et al. [2] follow the lifting approach while getting the first linear bound for the Hensel lifting, and consequentially reduce the cost of the algorithms to $O^{\sim}(d^{\omega})$ when the characteristic of K is 0 or sufficiently large, where $\omega \leq 3$ is the feasible matrix multiplication exponent and $O^{\sim}(d^{\omega}) = O(d^{\omega})(\log d)^{O(1)}$. The most recent result is due to Lecerf [29], which works for a large class of fields with any characteristic and has a lower complexity as follows.

Theorem 4. *[Lecerf]Assume that K has cardinality at least* $2d_xd_y + max(d_x, d_y) + 1$*. Then, given a polynomial* $f = f_1^{e_1} f_2^{e_2} \dots f_r^{e_r}$*, the computation of the irreducible decomposition* $(f_1, e_1), \dots, (f_r, e_r)$ *of* f *reduces to the computation of irreducible decompositions of polynomials in* $K[y]$ *whose degree sum is at most* $d_x + d_y$*, plus*

1. $O^{\sim}((d_xd_y)^{(\omega+1)/2})$ *arithmetic operations in* K *in characteristic* 0*;*
2. $O^{\sim}(l(d_xd_y)^{(\omega+1)/2})$ *arithmetic operations in* $\mathbb{F}_p$ *if* $K = \mathbb{F}_{p^l}$*.*

We will use this result to factor the minimal polynomial of a matrix in this thesis. A more efficient but probabilistic algorithm is also presented in Lecerf's paper [29]. Actually the minimal polynomial we wish to factor is of a special form, called Δ-inc form as defined in Definition 12. So there may be more efficient algorithms for such polynomials. Note that the factorization of polynomials is one of the dominating parts of our Wedderburn decomposition algorithm, Algorithm3.5.

Theorem 5. *[Lecerf] Assume that* K *has cardinality at least* $10d_xd_y$*. Then given a polynomial* $f = f_1^{e_1} f_2^{e_2} \dots f_r^{e_r}$ *the computation of the irreducible decomposition* $(f_1, e_1), \dots, (f_r, e_r)$ *of* f *reduces to the computation of irreducible decompositions of polynomials in* $K[y]$ *whose degree sum is at most* $d_x + d_y$*, plus*

1. $O((d_xd_y)^{1.5})$ *arithmetic operations in* K *and* $R(O(d_xd_y))$ *in characteristic* 0*;*
2. $O^{\sim}(k(d_xd_y)^{1.5})$ *operations in* $\mathbb{F}_p$ *and* $R(O(d_xd_y))$ *if* $K = \mathbb{F}_{p^k}$*.*

The algorithm outputs either nothing or a correct result with a probability at least $1/2$*.*

2.3 Selecting a Maximal Linearly Independent Subset of Vectors over $\mathbb{F}_q[y]$

We now discuss the cost of computing a maximal linearly independent subset from a set of vectors. It is not difficult but frequently used in this thesis. Given $\{a_1, a_2, \ldots, a_n\} \subset \mathbb{F}_q[y]^{1\times m}$, where $\deg(a_i) \leq \Delta$ for $1 \leq i \leq n$, we want to compute its maximal linearly independent subset. The idea is that we have a large probability of keeping the linear dependency of these vectors when specifying the indeterminate y in $\mathbb{F}_q$, as in the following theorem. Let $a = [a_1, \ldots, a_n] \in \mathbb{F}_q[y]^{m\times n}$. So a is a $m \times n$ matrix with degree bound Δ.

Theorem 6. *Given a matrix $a \in \mathbb{F}_q[y]^{m\times n}$ with degree bound Δ and rank k, then a mod $(y - \alpha)$ is a $m \times n$ matrix in $\mathbb{F}_q$ with rank at most k for any $\alpha \in \mathbb{F}_q$. There are at most $k\Delta$ elements in $\mathbb{F}_q$ such that the rank of a mod $(y - \alpha)$ is strictly less than k. Moreover, if rows $t_1,\ t_2,\ \ldots,\ t_k$ of a are linearly independent, then there are at most $k\Delta$ elements in $\mathbb{F}_q$ such that these rows of a mod $(y-\alpha)$ are not linearly independent.*

Proof. Since the rank of a is k, there is a $k \times k$ minor matrix of a, denoted by b such that $\det(b) \neq 0$. Without loss of generality, we assume that $t_i = i$ for $1 \leq i \leq k$, i.e. the first k rows of a are linearly independent. We already know that the entries in b have degree bound Δ. It is clear that $\det(b)$ is a polynomial in $\mathbb{F}_q[y]$ of degree at most $k\Delta$. If α makes the rank of a mod $(y - \alpha)$ strictly less than k, then $\det(a \bmod (y - \alpha)) = 0$, i.e. α is a root of $\det(b)$. So there are at most $k\Delta$ elements in $\mathbb{F}_q$ satisfying such condition. If a mod $(y - \alpha)$ has its rank strictly less than k, then α is a root of $\det(b)$. So there are at most $k\Delta$ elements in $\mathbb{F}_q$ such that the rank of a mod $(y - \alpha)$ is strictly less than k.

□

Thus if q is sufficiently large, we have a Monte Carlo algorithm to select a maximal linearly independent subset of a set of vectors.

Algorithm 2.1 Select Maximal Linearly Independent Subset

```
Input: The set of vectors $\{a_1, a_2, \ldots, a_n\} \subset \mathbb{F}_q[y]^{1\times m}$, where $\deg(a_i) \leq \Delta$ for
    $1 \leq i \leq n$ and $q \geq \frac{\min(m,n)}{\epsilon}\Delta$;
Output: A set of indices $\{t_1, t_2, \ldots, t_k\}$ such that $\{a_{t_1}, a_{t_2}, \ldots, a_{t_k}\}$ satisfies:
        these vectors are linearly independent;
        for any subset $S \subset \{a_1, a_2, \ldots, a_n\}$ such that $\{a_{t_1}, a_{t_2}, \ldots, a_{t_k}\} \subsetneq S$,
        $S$ is not linearly independent.

Choose a random $\alpha \in \mathbb{F}_q$, let $a$ be the matrix constructed by taking $a_i$ as its
    $i$th column for $1 \leq i \leq n$;
Compute $b = a \bmod (y - \alpha)$;
Return the indices of a maximal linearly independent rows of $b$, $\{t_1, t_2, \ldots, t_k\}$.
```

Theorem 7. *Given a set of vectors* $\{a_1, a_2, \ldots, a_n\} \subset \mathbb{F}_q[y]^{1\times m}$*, where* $deg(a_i) \leq \Delta$ *for* $1 \leq i \leq n$ *and* $q \geq \frac{min(m,n)}{\epsilon}\Delta$*, Algorithm 2.1 returns the indices of a maximal linearly independent subset with probability at least* $1 - \epsilon$ *taking* $O(mn\Delta + mn \cdot min(m,n))$ *operations in* $\mathbb{F}_q$*.*

Proof. From Theorem 6 we know there are at most $k\Delta$ elements in $\mathbb{F}_q$ to cause the algorithm to return an incorrect output. So the probability of correctness is at least $1 - \frac{k\Delta}{q} \geq 1 - \frac{k\Delta}{\Delta\min(m,n)/\epsilon} \geq 1 - \epsilon$ since $k \leq \min(m,n)$. Step 2 costs at most $m \times n \times \Delta$ operations in $\mathbb{F}_q$. We can use Gaussian elimination in step 3, which will cost $O(nm \cdot \min(m,n))$ operations in $\mathbb{F}_q$. So the total cost will be $O(mn\Delta + mn \cdot \min(m,n))$.

□

2.4 Solving Systems of Linear Equations

Solving systems of linear equations is a fundamental problem in symbolic computation linked to the basic matrix operations. For nonsingular system, a unique solution exists, while for a singular one, there is either no solution, or an infinite set of solutions generated by a single solution together with a basis of the null space of the coefficient matrix. Given a field F and a system of linear equations $ax = b$, $a \in F^{n\times n}$ and $b \in F^{n\times 1}$, we can solve for x by standard Gaussian elimination taking $O(n^3)$ operations in F. However, this result is not optimal. It is shown that this problem is no harder than the matrix multiplication, i.e. it takes only n^ω operations, where $\omega \leq 2.376$. $O(n^{2.376})$ is the currently best but not practical result. [13]

In this thesis, we need to understand the complexity of solving systems of linear equations over a polynomial ring $\mathbb{F}_q[y]$. Given a ring $F[y]$ over a field F and a system of linear equations $ax = b$, $a \in F[y]^{n\times n}$ and $b \in F[y]^{n\times 1}$, where the degree of the entries here is bounded by Δ. The difficulty comes from the size of the data, intermediate data and the final result. It is known that the matrix multiplication can be done in $O^\sim(n^\omega\Delta)$ operations in F. So do the problems of computing the determinant, the Smith normal form and the linear system solution. Generally, we allow a to be an $m\times n$ matrix with rank r. The following algorithm is by Storjohann and Villard [42] for solving the linear system.

Theorem 8. *[Storjohann & Villard] Let* $a \in K[x]^{m\times n}$ *be a matrix with the degree of all the entries at most* d*. The rank* r *of* a *and* $m-r$ *linearly independent polynomial vectors in the nullspace of* a *can be computed with*

$$O(\frac{nmMM(r,d)}{r^2} + (\frac{m}{r} + logr)(MM(r,d)log(rd) + r^2B(d)logr + rM(rd))),$$

or $O^\sim(nmr^{\omega-2}d)$ *operations in* K *by a randomized Las Vegas (certified) algorithm, where* $B(d)$ *is the cost of solving the extended gcd problem for two polynomials in*

$K[x]$ of degree bounded by d. The degree sum of the computed nullspace vectors is less than $rd\lceil logr \rceil + (m-2r)d$.

We will take advantage of the efficiency of solving linear system in $\mathbb{F}_q[y]$ here. However, we are not satisfied by the stated degree bound. A more precise statement for the degree can be found in the proof of this theorem in Storjohann and Villard's paper [42]. If $m \geq 2r$, there are at most $m-2r$ vectors in the nullspace of degree at most d and the remainder r vectors have a degree sum bound $rd\lceil \log_2 r \rceil$; otherwise, the $m-r$ vectors have a degree sum bound $rd\lceil \log_2 r \rceil$. This result will be applied in three ways later as follows.

- to solve the system as part of our whole algorithm.
- to bound the degree of the null space, thereby to prove the existence of some special element as well as the complexity of the algorithm. Always, we use the conclusion that it will take $O\tilde{}(nm \cdot \min(n,m)^{\omega-2}d)$ to solve the linear system and the degree bound for the output is md.
- to make other relative algorithms of Las Vegas type, as shown below in this section.

Taking advantage of the Las Vegas algorithm for computing the linear nullspace, we can check whether the result of Algorithm 2.1 is correct, i.e. maximal. Recall that Algorithm 2.1 will return the indices of the vectors. We put these vectors as the rows of our matrix $a' \in \mathbb{F}_q[y]^{k\times m}$ with degree upper bound of its entries Δ. One way of checking this result is first computing the nullspace of the matrix a', denoted by $N \in \mathbb{F}_q[y]^{m\times(m-k)}$, and then checking if the product aN equals to zero matrix or not. Here a is the matrix with all the input vectors as its rows. For the complexity, computing N will take $O\tilde{}(k^{\omega-1}m\Delta)$ and computing aN will take $O\tilde{}(n(m-k)mk\Delta) = O\tilde{}(nm^2 \cdot \min(m,n)\Delta)$. So the Las Vegas algorithm for selecting a maximal linearly independent subset costs $O\tilde{}(nm^2 \cdot \min(m,n)\Delta)$.

Theorem 9. *Given a set of m dimensional vectors $\{a_1, a_2, \ldots, a_n\} \subset \mathbb{F}_q[y]^{1\times m}$, where $deg(a_i) \leq \Delta$ for $1 \leq i \leq n$ and $q \geq \frac{min(m,n)}{\epsilon}\Delta$, there exists a probabilistic algorithm of Las Vegas type for computing its maximal linearly independent subset taking $O\tilde{}(nm^2 \cdot min(m,n)\Delta)$ operations in $\mathbb{F}_q$.*

2.5 Computation of the minimal polynomial

The minimal polynomial plays an important role in our procedure of computing the idempotents, which is the key step to decompose the algebra. We will discuss computing the minimal polynomial of a matrix over $\mathbb{F}_q[y]^{m\times m}$ with degree bound Δ. Computing the minimal polynomial has a close relationship with computing the Frobenius form. The algorithm we propose in this section can be also used to compute the Frobenius form of the matrix as well. Recall that the polynomial with respect to the first block of its Frobenius form is the minimal polynomial of the matrix.

Actually the problem of computing the Frobenius form has been well studied. There are many algorithms proposed for this problem over different kinds of fields and rings. The deterministic algorithms are first presented by Lüneburg [31] and Ozello [32] in 1987 taking $O(n^4)$ operations in field F. Storjohann and Villard [40, 41] improved it to $O(n^3)$ in 1998 and 2000. Giesbrecht [17] presented a probabilistic algorithm taking the same number of operations as required for matrix multiplication, that is $O(n^\omega \mathrm{log} n)$, over the field with at least n^2 elements. More recently, an algorithm requiring $O(n^\omega \mathrm{log} n)$ operations over any field for this problem is proposed by Eberly [6].

However, we have another problem when the field K is $\mathbb{F}_q(y)$ and the matrix is integral. In this case we need to pay attention to the size of the intermediate data as well as the final result. The following lemma will be of assistance.

Lemma 10. *Suppose R is an integral domain and K is its rational field. Given a matrix $a \in R^{n\times n}$, then its Frobenius form over K is in $R^{n\times n}$ as well.*

Proof. It is obvious that $\det(xI - a)$ is in $R[x]$. We already know that the minimal polynomial is the divisor of its determinant $\det(xI - a)$. By Gauss's Lemma for polynomials, we know the minimal polynomial has coefficients in R. So the entries in the first block of its Frobenius form are all in R. Following the same path, we can get that the entries of all the blocks are in R.

□

Giesbrecht and Storjohann [18] present an algorithm to compute the Frobenius form of an integer matrix $a \in \mathbb{Z}^{n\times n}$ taking expected $O^\sim(n^4(\log|a|)^2)$ operations. We will deal with the $\mathbb{F}_q(y)$ case in this section. Actually when it comes to the minimal polynomial, we can get an even better result. Suppose we are given a matrix $a \in \mathbb{F}_q[y]^{m\times m}$ with degree bound Δ of its entries.

Lemma 11. *Let $f = x^t + a_1x^{t-1} + \ldots + a_t$, $h = x^v + b_1x^{v-1} + \ldots + b_v$ and $g = x^u + c_1x^{u-1} + \ldots + c_u$ in $\mathbb{F}_q[x, y]$. If $f = gh$ and deg(a_i) $\leq i\Delta$ for $1 \leq i \leq t$, then deg(c_j) $\leq j\Delta$ and deg(b_k) $\leq k\Delta$ for $1 \leq j \leq u$ and $1 \leq k \leq v$.*

Proof. Suppose not all the elements in $\{b_1, \ldots, b_v\}$ satisfy $\deg(b_i) \leq i\Delta$. For a vector $(d_1, d_2, \ldots, d_t) \in \mathbb{Z}^t$ let $\underset{i}{\mathrm{argmax}}\{d_i\}$ denote the indices $\{i_1, \ldots, i_{\bar{t}}\}$ such that d_{i_j} is maximal for $1 \leq j \leq \bar{t}$. Let

$$k_h = \min(\underset{i}{\mathrm{argmax}}\{\deg(b_i) - i\Delta\}).$$

If $\{c_1, \ldots, c_u\}$ satisfies $\deg(c_i) \leq i\Delta$ for $1 \leq i \leq u$, then consider the degree of a_{k_h}, the coefficient of the term $a_{k_h}x^{t-k_h}$. Then

$$a_{v-k_h} = \sum_{j=1}^{\min(k_h,u)} b_{k_h-j}c_j + b_{k_h} = (\delta + b_{k_h}),$$

where $\delta = \sum_{j=1}^{\min(k_h,u)} b_{k_h-j}c_j$. Since $\deg(b_{k_h-j}) - (k_h - j)\Delta < \deg(b_{k_h}) - k_h\Delta$ and $\deg(c_j) \le j\Delta$ for $1 \le j \le \min(k_h, u)$, then

$$\begin{aligned} \deg(\delta) &= \max\{\deg(b_{k_h-j}) + \deg(c_j)\} \\ &< \deg(b_{k_h}) - k_h\Delta + (k_h - j)\Delta + j\Delta \\ &= \deg(b_{k_h}). \end{aligned}$$

So $\deg(a_{k_h}) = \deg(b_{k_h}) > k_h\Delta$, a contradiction.

If not all the elements in $\{c_1, \ldots, c_u\}$ satisfy $\deg(c_i) \le i\Delta$ for $1 \le i \le u$, we denote

$$k_g = \min(\operatorname*{argmax}_i\{\deg(c_i) - i\Delta\}).$$

Then consider the degree of $a_{k_h+k_g}$, the coefficient of the term $x^{t-k_g-k_h}$:

$$a_{k_h+k_g} = \sum_{d=1}^{\min(k_g,v-k_h)} c_{k_g-d}b_{k_h+d} + c_{k_g}b_{k_h} + \sum_{d=1}^{\min(k_h,u-k_g)} c_{k_g+d}b_{k_h-d} = (\delta_1 + b_{k_h}c_{k_g} + \delta_2),$$

where $\delta_1 = \sum_{d=1}^{\min(k_g,v-k_h)} c_{k_g-d}b_{k_h+d}$ and $\delta_2 = \sum_{d=1}^{\min(k_h,u-k_g)} c_{k_g+d}b_{k_h-d}$. Since $\deg(c_{k_g-d}) - (k_g - d)\Delta < \deg(c_{k_g}) - k_g\Delta$ for $1 \le d \le \min(k_g, v - k_h)$ and $\deg(b_j) - j\Delta \le \deg(b_{k_h}) - k_h\Delta$ for $k_h + 1 \le j \le \min(v, k_g + k_h)$, then

$$\begin{aligned} \deg(\delta_1) &= \max_d\{\deg(c_{k_g-d}) + \deg(b_{k_h+d})\} \\ &< \deg(c_{k_g}) - k_g\Delta + (k_g - d)\Delta + \deg(b_{k_h}) - k_h\Delta + (k_h + d)\Delta \\ &= \deg(c_{k_g}) + \deg(b_{k_h}). \end{aligned}$$

Similarly, since $\deg(b_{k_h-d}) - (k_h - d)\Delta < \deg(b_{k_h}) - k_h\Delta$ for $1 \le d \le \min(k_h, u - k_g)$ and $\deg(c_i) - i\Delta \le \deg(c_{k_g}) - k_g\Delta$ for $k_g + 1 \le i \le \min(u, k_g + k_h)$, $\deg(\delta_2) < \deg(c_{k_g}) + \deg(b_{k_h})$. So $\deg(a_{k_h+k_g}) = \deg(c_{k_g}) + \deg(b_{k_h}) > (k_g + k_h)\Delta$, a contradiction.

□

To simplify our statement later in this section, we give a definition for such polynomial with a good upper degree bound for its coefficients.

Definition 12. A polynomial $f \in F[x, y]$ is in Δ-inc form with respect to x, if $f = x^e + a_1x^{e-1} + \ldots + a_e$ and $\deg_y(a_i) \le i\Delta$ for $1 \le i \le e$.

The following two propositions are simple but useful.

Proposition 13. *Let a be a matrix in $F^{m\times m}$. Then the minimal polynomial of a, minpoly(a) $= \frac{det(xI_m - a)}{gcd(g_1, g_2, \ldots, g_{m^2})}$, where $\{g_1, g_2, \ldots g_{m^2}\}$ is the set of determinants of all the $(m-1)\times(m-1)$ minors of $xI_m - a$ and det($xI_m - a$) is the determinant of $xI_m - a$.*

For a given element $\alpha \in \mathbb{F}_q$ and a matrix $a \in \mathbb{F}_q[y]^{m\times m}$, we denote by $a^{(\alpha)} = a \bmod y - \alpha \ \in \mathbb{F}_q^{m\times m}$. Similarly, we denote $f^{(\alpha)} = f \bmod y - \alpha \ \in \mathbb{F}_q[x]$ for $f \in \mathbb{F}_q[x, y]$.

Proposition 14. *gcd(f, g)$^{(\alpha)}$ divides gcd($f^{(\alpha)}, g^{(\alpha)}$) for $f, g \in \mathbb{F}_q[x, y]$ and $\alpha \in \mathbb{F}_q$.*

The key idea of our algorithm is inspired by the paper of Giesbrecht and Storjohann [18] that for a matrix a and a special element $\alpha \in \mathbb{F}_q$, the structure of the Frobenius form of a, thereby its minimal polynomial, will be kept and passed to that of the matrix $a^{(\alpha)}$. Lemma 15 describes such relationship.

Lemma 15. *Given a matrix $a \in \mathbb{F}_q[y]^{m\times m}$ and $\alpha \in \mathbb{F}_q$, deg(minpoly($a^{(\alpha)}$)) $\leq$ deg$_x$(minpoly(a)). Besides, minpoly($a^{(\alpha)}$) $\mid$ minpoly(a)$^{(\alpha)}$.*

Proof. From Proposition 13 we know $\text{minpoly}(a) = \frac{\det(xI_m - a)}{\gcd(g_1, g_2, \ldots, g_{m^2})}$ and $\text{minpoly}(a^{(\alpha)}) = \frac{\det(xI_m - a^{(\alpha)})}{\gcd(g_1^{(\alpha)}, g_2^{(\alpha)}, \ldots, g_{m^2}^{(\alpha)})}$. It is straightforward that $\deg(\det(xI_m - a)) = \deg(\det(xI_m - a^{(\alpha)}))$. From Proposition 14 we have $\gcd(g_1, g_2, \ldots, g_{m^2})^{(\alpha)}$ dividing $\gcd(g_1^{(\alpha)}, g_2^{(\alpha)}, \ldots, g_{m^2}^{(\alpha)})$. So , $\deg(\gcd(g_1, g_2, \ldots, g_{m^2})) \leq \deg(\gcd(g_1^{(\alpha)}, g_2^{(\alpha)}, \ldots, g_{m^2}^{(\alpha)}))$ and $\text{minpoly}(a^{(\alpha)}) = \frac{\det(xI_m - a^{(\alpha)})}{\gcd(g_1^{(\alpha)}, g_2^{(\alpha)}, \ldots, g_{m^2}^{(\alpha)})}$ dividing $\text{minpoly}(a)^{(\alpha)} = (\frac{\det(xI_m - a)}{\gcd(g_1, g_2, \ldots, g_{m^2})})^{(\alpha)}$. □

We would like to choose such good element α, which is defined below, to simplify the computation.

Definition 16. Given $a \in \mathbb{F}_q[y]^{m\times m}$ and $\alpha \in \mathbb{F}_q$, we say α is a good element for a if $\deg(\text{minpoly}(a^{(\alpha)})) = \deg(\text{minpoly}(a))$.

We will show later in this section that the good elements are dense in $\mathbb{F}_q$ when q is sufficiently large. Now suppose we can pick as many good elements as we need from $\mathbb{F}_q$ and we will use these ones to computer the minimal polynomial of $a \in \mathbb{F}_q(y)^{m\times m}$. The algorithm is presented as follows.

Theorem 17. *Given a matrix $a \in \mathbb{F}_q[y]^{m\times m}$ and a set of good elements $\{\alpha_1, \ldots, \alpha_{m\Delta}\}$, Algorithm 2.2 computes the minimal polynomial of a correctly taking $O^{\sim}(m^{\omega+1}\Delta)$ operations in $\mathbb{F}_q$.*

Proof. It is easy to check that $\det(xI_m - a)$ is in the Δ-inc form. So by Lemma 11, the minimal polynomial of a is in Δ-inc form as well. Since α_i is a good element for a, we can compute the degree in x of the minimal polynomial of a, denoted by t. So $\text{minpoly}(a) = x^t + a_1 x^{t-1} + \ldots + a_t$, where $a_i \in F[y]$ and $\deg(a_i) \leq$

Algorithm 2.2 Compute the Minimal Polynomial

```
Input: A matrix $a \in \mathbb{F}_q[y]^{m\times m}$ and a set of good elements $\{\alpha_1, \alpha_2, \ldots, \alpha_{m\Delta}\}$.

Output: The minimal polynomial of $a$.

1: Compute the minimal polynomial of $a^{(\alpha_i)}$, denoted by $f^{(\alpha_i)} = x^t + c_{i1}x^{t-1} + \ldots + c_{it}$ for $i = 1, 2, \ldots, m\Delta$;
2: Interpolate the polynomial $g_j \in \mathbb{F}_q[y]$ of degree at most $j\Delta$ with the values $\{c_{1j}, c_{2j}, \ldots, c_{(j\Delta)j}\}$ for $j = 1, 2, \ldots, t$;
3: Return the polynomial $f = x^t + g_1x^{t-1} + \ldots + g_t \in \mathbb{F}_q[x, y]$.
```

$i\Delta$ for $1 \leq i \leq t$. For any good element α_i, the coefficient of x^{t-j} of $a^{(\alpha_i)}$'s minimal polynomial $c_{ij} = a_j^{(\alpha_i)}$. Thus we can do the interpolation over every a_i to compute minpoly(a). Doing this, we need to first compute $a^{(\alpha_i)}$, which costs $O(m^2 \times M(m\Delta)\text{log}m\Delta) = O^{\sim}(m^3\Delta)$ using the fast polynomial evaluation. Computing the minimal polynomials of $a^{(\alpha_i)}$ requires $O(m\Delta \times m^{\omega}) = O(m^{\omega+1}\Delta)$. Step 2 of interpolation will take $O(M(\Delta)\text{log}\Delta + M(2\Delta)\text{log}2\Delta + \ldots + M(t\Delta)\text{log}t\Delta) = O^{\sim}(t^2\Delta) = O^{\sim}(m^2\Delta)$ operations by fast interpolation. So the algorithm is of complexity $O^{\sim}(m^{\omega+1}\Delta)$.

□

The following theorem shows the density of the good elements in $\mathbb{F}_q$.

Theorem 18. *Given $a \in \mathbb{F}_q[y]^{m\times m}$ of degree Δ in y, if $q \geq \frac{1}{\epsilon}m(m-1)^2\Delta$ then the probability of randomly choosing a good element in $\mathbb{F}_q$ for a is at least $1 - \epsilon$.*

Proof. Given $a \in \mathbb{F}_q(y)^{m\times m}$, suppose $\alpha \in \mathbb{F}_q$ is not a good element, i.e. $\deg(\text{minpoly}(a^{(\alpha)})) \neq \deg_x(\text{minpoly}(a))$. So $\deg(\gcd(g_1, g_2, \ldots, g_{m^2})^{(\alpha)})$ is less than $\deg(\gcd(g_1^{(\alpha)}, g_2^{(\alpha)}, \ldots, g_{m^2}^{(\alpha)}))$. If $\gcd(g_1, g_2, \ldots, g_{m^2}) \neq 1$, then we can divide every g_i by $\gcd(g_1, g_2, \ldots, g_{m^2})$.

So in this proof we assume that $\gcd(g_1, g_2, \ldots, g_{m^2}) = 1$. We also denote g_1 by f_1, $\gcd(f_i, g_{i+1})$ by f_{i+1} for $1 \leq i \leq m^2-1$, so $\gcd(g_1, g_2, \ldots, g_{m^2}) = \gcd(f_{m^2-1}, g_{m^2})$. Since the greatest common divisor is not affected by the order of the polynomials, we may also assume that g_1 is $D(\begin{array}{ccc} 1 & \ldots & m-1 \\ 1 & \ldots & m-1 \end{array})$, the determinant of the first $m-1$ rows and first $m-1$ columns, and the degrees of f_i strictly decrease from f_2 to f_t. It is straightforward that $f_i | g_1$ and $t \leq m$ by checking the degree of f_i with respect to x. So $\{f_1, f_2, \ldots, f_t\}$ are in the Δ-inc form, $f_t = 1$ and $\deg_x(f_i) \leq m - i$ for $1 \leq i \leq t - 1$.

For every step from f_i to f_{i+1}, we want to give a upper bound for the number of elements in $\mathbb{F}_q$ that make the degrees of $\gcd(f_i^{(\alpha)}, g_{i+1}^{(\alpha)})$ strictly less than $\gcd(f_i, g_{i+1})^{(\alpha)}$. Let $c = \gcd(f_i, g_{i+1})$, $f = \frac{f_i}{c}$ and $g = \frac{g_{i+1}}{c}$, then $\gcd(f, g) = 1$ and $\gcd(f^{(\alpha)}, g^{(\alpha)}) \neq 1$. So $\text{res}(f, g) \neq 0$ and $\text{res}(f^{(\alpha)}, g^{(\alpha)}) = 0$, i.e. $y - \alpha | \text{res}(f, g)$. Since $\deg_y(f_i) \leq (m - i)\Delta$ and $\deg_y(g_{i+1}) \leq (m - 1)\Delta$, then $\deg(\text{res}(f, g)) \leq$

$2(m-1)(m-i)\Delta$ (there is a tighter bound that $\deg(\text{res}(f,g)) \leq (m-1)(m-i)\Delta$). So at most there are $2(m-1)(m-i)\Delta$ elements which are not good from f_i to f_{i+1}.

In total, the number of the elements which are not good for a is at most $\sum_{i=1}^{t} 2(m-1)(m-i)\Delta = t(m-1)(2m-1-t)\Delta \leq m(m-1)^2\Delta$. If $q \geq \frac{1}{\epsilon}m(m-1)^2\Delta$, the probability that a random element in $\mathbb{F}_q$ is good is at least $1-\epsilon$.

□

Thus, we can develop a Monte Carlo algorithm to compute the minimal polynomial of the matrix over $\mathbb{F}_q[y]$.

Algorithm 2.3 Compute the minimal polynomial

```
**Input:** An integral matrix $a \in \mathbb{F}_q[y]^{m\times m}$, where the degrees of its entries are at most $\Delta$ and $q \geq \frac{1}{\epsilon}m^2(m-1)^2\Delta^2$;
**Output:** The minimal polynomial of $a$, $f \in \mathbb{F}_q[x,y]$;

Randomly choose $\{\alpha_1, \alpha_2, \ldots, \alpha_{m\Delta}\} \subset \mathbb{F}_q$;
Compute the minimal polynomial of every $a^{(\alpha_i)}$, denoted by $f^{(\alpha_i)} = x^t + c_{i1}x^{t-1} + \ldots + c_{it}$ for $j = 1, 2, \ldots, m\Delta$, where $t$ is the maximal degree of the minimal polynomials of $\alpha_i$ for $1 \leq i \leq m\Delta$;
if Not all the $f^{(\alpha_i)}$s have a same degree $t$ then
    return "failure";
else
    Use the values of $\{c_{1j}, c_{2j}, \ldots, c_{(j\Delta)j}\}$ to interpolate the polynomial $g_j$ of degree at most $j\Delta$ in $\mathbb{F}_q[y]$ for $1 \leq j \leq t$;
    Return the polynomial $f = x^t + g_1x^{t-1} + \ldots + g_t$;
end if
```

Theorem 19. *Given an integral matrix $a \in \mathbb{F}_q[y]^{m\times m}$, where the degrees of its entries are at most Δ and $q \geq \frac{1}{\epsilon}m^2(m-1)^2\Delta^2$, Algorithm 2.3 computes the minimal polynomial of a correctly with probability at least $1-\epsilon$, taking $O^{\sim}(m^{\omega+1}\Delta)$ operations in $\mathbb{F}_q$.*

Proof. The probability that algorithm 2.3 returns the correct minimal polynomial is the same as the probability that $\{\alpha_1, \ldots, \alpha_{m\Delta}\}$ are good elements. From Theorem 18, it is at least with probability $1-\frac{\epsilon}{m\Delta}$ that α_i is a good element. So the probability that $\{\alpha_1, \ldots, \alpha_{m\Delta}\}$ are all good elements is at least $(1-\frac{\epsilon}{m\Delta})^{m\Delta} \geq 1-\frac{\epsilon}{m\Delta}m\Delta = 1-\epsilon$. Thus the probability of correctness of this algorithm is at least $1-\epsilon$. Since $\{\alpha_1, \ldots, \alpha_{m\Delta}\}$ randomly chosen from $\mathbb{F}_q$, so we will not account for its cost. The remaining cost is the same as in Algorithm 2.2, thereby its complexity is $O^{\sim}(m^{\omega+1}\Delta)$ by Theorem 17.

□

Although an efficient probabilistic algorithm of Monte Carlo type for computing the minimal polynomial of a matrix $A \in \mathbb{F}_q[y]^{m\times m}$ has been presented above, we hope we can eliminate the possible error. The remaining part of this section is devoted to a probabilistic algorithm of Las Vegas type. Note that the minimal polynomial of a has such property: $f = x^d + a_1x^{d-1} + a_2x^{d-2} + \ldots + a_d$ where $\deg(a_i) \leq i\Delta$ for $1 \leq i \leq d$. Once we can determine the degree of the minimal polynomial, we can choose a set of $m\Delta$ good elements for the latter procedure. Actually if we find an element whose minimal polynomial has a greater degree, we can report “failure”. Now suppose we already have a set of good elements $\{\alpha_1, \ldots, \alpha_{m\Delta+1}\}$ and the degree of the minimal polynomial of $a^{(\alpha_i)}$ is t for $1 \leq i \leq m\Delta+1$. We need the following proposition to certify our correctness of the minimal polynomial.

Proposition 20. *Given a matrix $a \in \mathbb{F}_q[y]^{m\times m}$ with degree upper bound of its entries Δ and $g = x^d+a_1x^{d-1}+\ldots+a_d \in \mathbb{F}_q[y,x]$ where $a_i \in \mathbb{F}_q[y]$ with deg(a_i) $\leq i\Delta$ and $d \leq m$ for $1 \leq i \leq d$. If g mod $(y-e)$ is the minimal polynomial of $a^{(e)}$ for every element $e \in E$ and the size of E is $m\Delta+1$, then $g(a) = 0$.*

Proof. Assume $E = \{e_1, e_2, \ldots, e_{m\Delta+1}\}$. Note that $g(a)$ is a $m \times m$ matrix with degree of its entries at most $d\Delta < m\Delta+1$ and $|E| = m\Delta+1$. Since g mod $(y-e)$ is the minimal polynomial of $a^{(e)}$ for every element $e \in E$, $g(a) \equiv 0$ mod $(y-e)$ for $e \in E$. So $g(a) \equiv 0$ mod $((y-e_1)\ldots(y-e_{m\Delta+1}))$, i.e. $g(a) = 0$.

□

The proposition above suggests a way to make sure the algorithm will not return a wrong result. We only need to modify a few things in Algorithm 2.3 to make it of Las Vegas type.

Algorithm 2.4 Compute the minimal polynomial

```
Input: An integral matrix $a \in \mathbb{F}_q[y]^{m\times m}$, where the degrees of its entries are bounded by $\Delta$ and $q \geq \frac{1}{\epsilon}m(m-1)^2\Delta(m\Delta+1)$;
Output: The minimal polynomial of $a$, $f \in \mathbb{F}_q[x,y]$;

Randomly choose $\{\alpha_1, \alpha_2, \ldots, \alpha_{m\Delta}, \alpha_{m\Delta+1}\} \subset \mathbb{F}_q$;
Compute the minimal polynomial of every $a^{(\alpha_i)}$, denoted by $f^{(\alpha_i)} = x^t+c_{i1}x^{t-1}+\ldots+c_{it}$ for $j = 1,2,\ldots,m\Delta$, where $t$ is the maximal degree of the minimal polynomials of $\{a^{(\alpha_1)}, \ldots, a^{(\alpha_{m\Delta+1})}\}$;
if Not all the elements in $\{f^{(\alpha_1)}, \ldots, f^{(\alpha_{m\Delta+1})}\}$ have a same degree $t$ then
  Return “failure”;
else
  Use the values of $\{c_{1j}, c_{2j}, \ldots, c_{(m\Delta+1)j}\}$ to interpolate the polynomial $g_j$ of degree at most $m\Delta+1$ in $\mathbb{F}_q[y]$ for $1 \leq j \leq t$;
  if there is some $i$ such that $\deg(g_i) > i\Delta$ then
    Return “failure”;
  else
    Return the polynomial $f = x^t + g_1x^{t-1} + \ldots + g_t$;
  end if
end if
```

Theorem 21. *Given an integral matrix $a \in \mathbb{F}_q[y]^{m\times m}$, where the degrees of its entries are bounded by Δ and $q \geq \frac{1}{\epsilon} m(m-1)^2\Delta(m\Delta+1)$, Algorithm 2.4 computes the minimal polynomial of a correctly with probability at least $1-\epsilon$ and return "failure" with probability at most ϵ, taking $O^{\sim}(m^{\omega+1}\Delta)$ operations in $\mathbb{F}_q$.*

Proof. If the algorithm does not return "failure" at neither step 3 nor step 5, then by Proposition 20 the returned f is the annihilating polynomial of a. So $\mathrm{minpoly}(a) \mid f$. By Lemma 15, $\deg(f) \leq \deg(\mathrm{minpoly}(a))$. So $f = \mathrm{minpoly}(a)$. Note that if $\{\alpha_1, \ldots, \alpha_{m\Delta+1}\}$ are good elements then the algorithm will not return "failure". So the probability of returning the correct minimal polynomial is at least the same as, if not greater than, the probability that all the α_is are good elements. From the proof of Theorem 19 we know the probability of correctness of this algorithm is at least $1-\epsilon$. Since we actually do nothing in step 3 and step 5. The cost of this algorithm is the same as Algorithm 2.3, thereby its complexity is $O^{\sim}(m^{\omega+1}\Delta)$ by Theorem 19.

□

Chapter 3

Computations of Matrix Algebras over $\mathbb{F}_q(y)$

Our main object of study in this thesis is finite-dimensional associative algebras over $\mathbb{F}_q(y)$. These are completely classified by the following representation theorem [37].

Theorem 22. *[Representation Theorem] Let A be a finite-dimensional algebra over a field F and $dim_F A = n$. Then A is isomorphic to a subalgebra of $M_{n+1}(F)$. Moreover, if A has an identity element then A is isomorphic to a subalgebra of $M_n(F)$.*

In the case that A has no identity, we can add one using Dorroh extension [28]. This is why we need $(n+1) \times (n+1)$ matrix in the theorem above. The Representation Theorem is easy to prove using the regular representation of A. Thus, we will focus on the decomposition of the matrix algebras over $\mathbb{F}_q(y)$ with identity. Also, we will assume in this thesis that $m \leq n \leq m^2$ for the dimension of algebra n and the dimension of the matrix m. The structure of finite-dimensional matrix algebras is theoretically clear and fully understood, via Wedderburn's Structure Theorem [34]. Given an algebra A, $A/\mathrm{Rad}(A)$ is semisimple. For a semisimple algebra, Wedderburn's Structure Theorem states that it is isomorphic to the direct sum of its simple components as follows.

Theorem 23. *[Wedderburn] Suppose that A is a finite-dimensional semisimple algebra over the field F. Then*

$$A \cong A_1 \oplus A_2 \oplus ... \oplus A_k$$

for simple algebras $A_1, A_2, ..., A_k \subset A$, and each component satisfies

$$A_i \cong D_i^{t_i \times t_i},$$

where D_i is a division ring over F for some positive integer t_i, for $1 \leq i \leq k$.

Typical proofs in textbook of these results are not constructive thanks to statement like "pick any minimal right ideal". To simplify the discussion, we will first assume q is large enough and discuss the algorithms to compute the radical and the simple components of the algebra for such case. Then in the last section we will discuss how to modify our algorithms for the case that q is small.

3.1 Wedderburn Decomposition

In this section we hope to compute the simple components, i.e. the minimal ideals, of the semisimple algebra. According to Wedderburn's Structure Theorem, every minimal ideal is isomorphic to $D^{t\times t}$, a full matrix ring over some division ring D which is extended from the ground field F. The first (deterministic) general polynomial-time algorithm for computing the simple components of a semisimple algebras over $\mathbb{F}_q$ is presented by Friedl and Rónyai [9] and Rónyai [37]. There is lots of subsequent work related to this problem instigated by Ivanyos, Rónyai, Gianni and so on [16, 21, 33]. It is shown by Rónyai [35, 36] that deciding the existence of the nontrivial idempotents in an algebra over a number field has the same complexity as integer factorization, which is intractable at this time. When the algebra is commutative over $\mathbb{Q}$, Gianni et al. [16] give an efficient algorithm to compute its local components. Other related work such as decomposition of modules are presented by Parker [33] and extended by Holt and Rees [21].

In this thesis we will focus on the Wedderburn decomposition over $\mathbb{F}_q(y)$. We will first introduce some results about the Wedderburn decomposition over $\mathbb{F}_q$ in Subsection 3.1.1. After a quick review on the inefficiency of Friedl and Rónyai's algorithm [37, 9], we will discuss the efficient algorithm by Eberly and Giesbrecht [7] which provides us the key idea when developing our algorithm for the Wedderburn decomposition over $\mathbb{F}_q(y)$. We also discuss a way to modify the algorithm by Friedl and Rónyai [9, 37] to be more efficient but probabilistic. Starting from Subsection 3.1.3, we will bring out our final algorithm step by step. First we will analyze the complexity of the algorithm by Ivanyos [24] In Subsection 3.1.2 which, to our best knowledge, is the only algorithm for the Wedderburn decomposition over $\mathbb{F}_q(y)$. In Subsection 3.1.3 we will discuss generally the key position of the idempotents in the Wedderburn decomposition. Finally we present our probabilistic algorithm for computing the set of idempotents and give a complete algorithm and its analysis in Subsections 3.1.4 and 3.1.5.

Actually our algorithm does not quite compute a complete Wedderburn decomposition. It computes the simple components of the semisimple algebra. However, it does not compute the isomorphic mapping from each of its simple components to some $D^{t\times t}$. This problem is also very important, with applications such as the factorization of the Ore polynomials [19]. Note that for the algebra over the finite field, this problem is solved both in Friedl and Rónyai's paper [9, 37] and Eberly and Giesbrecht's paper [7]. But it is still open in Ivanyos, Rónyai and Szántó's paper [24] when it comes to $\mathbb{F}_q(y)$.

3.1.1 Decomposition of Semisimple Algebras Over $\mathbb{F}_q$

Even though the algorithm of Friedl and Rónyai [9, 37] runs in polynomial time, it is expensive, with a high exponent. The dominating part comes from computing the center of the algebra and solving the invariant space of the mapping $x \longmapsto x^p$ in a standard way. For the computation of its center we need to solve a linear system of size $nm^2 \times n$ and for the invariant space we need to do $n$log p times of $m \times m$ matrix multiplication.

The first improvement in the speed of algorithms for the decomposition of semisimple algebras over a finite field is due to Eberly and Giesbrecht [7]. They give an probabilistic algorithm to compute the set of orthogonal primitive idempotents in $O^{\sim}(m^{\omega} + R(A))$, where $O(R(A))$ is the cost of selecting a random element a uniformly from the algebra A. They then show that using the idempotents the Wedderburn decomposition becomes easy and efficient as presented in the next subsection. The key idea of Eberly and Giesbrecht's work [7] is based on the density of the "good" elements in the simple algebra over $\mathbb{F}_q$, as presented in the following theorem.

Theorem 24. *[Eberly & Giesbrecht] Let $A \subset \mathbb{F}_q^{m \times m}$ be a simple algebra of dimension n. The number of $a \in A$ with f = minpoly(a) such that there exists a factorization $f = f_1 f_2$ for relatively prime polynomials f_1, $f_2 \in \mathbb{F}_q[x]$ with corresponding idempotents ω and $1 - \omega$, and such that $n/2 \leq dim_{\mathbb{F}_q}(A\omega) \leq 3n/4$, is at least $q^n/22$.*

Recall that any simple algebra over $\mathbb{F}_q$ is isomorphic to a full matrix ring $E^{t \times t}$ over an extension field E of $\mathbb{F}_q$ for some positive integer t. This theorem is proved by counting the number of such matrices in $E^{t \times t}$. The primitive idempotents can be computed in expected $O^{\sim}(m^{\omega} + \mathrm{R}(A))$ operations over $\mathbb{F}_q$. Eberly and Giesbrecht [7] showed as well the large probability to split the idempotents in different simple components using a random element from the algebra. Another important thing to be aware of is that not only the simple components of the semisimple part, but the isomorphic mapping from each of its simple component to some $E^{t \times t}$ can be computed from the primitive idempotents.

We also note that there is a simple way to modify the algorithm of Friedl and Rónyai [9, 37]. Recall that the dominating part is from the computation of the center of the algebra. Other than the deterministic standard method, Eberly and Giesbrecht [8] present an efficient algorithm taking $O^{\sim}(m^{\omega})$ operations over $\mathbb{F}_q$ which greatly reduces the complexity. Actually the method here is essentially the same as that in the probabilistic Wedderburn decomposition algorithm [7]. It is also shown that we can avoid computing the invariant space of the mapping $x \longmapsto x^p$, and consequently get the same cost, $O^{\sim}(m^{\omega} + \mathrm{R}(A))$ again. For details of this algorithm please refer to the paper [8].

3.1.2 Algorithm of Ivanyos, Rónyai and Szántó

A polynomial-time algorithm for computing the Wedderburn decomposition of a finite dimensional associative algebra is given in Ivanyos's paper[24]. The basic idea is to first change the algebra into a commutative separable one and then to solve this problem over this much nicer algebra. We denote this commutative separable algebra by A. Let B denote the direct sum of the prime fields in the simple components of our commutative separable algebra A. The algorithm has two cases depending on the character of the ground field p. If p is small, we can solve the linear system for the basis of B. Since B is a $\mathbb{F}_q$-algebra generated by the idempotents of A, we can compute its primitive idempotents (and therefore the primitive idempotents of A) easily. However, in this way we get p as a factor in the complexity. So when p is large, the algorithm uses a lift-and-project method so that we can avoid getting p in the complexity. The lift-and-project method is more efficient but requires a large q as showed later in this subsection.

The first step of the algorithm is to transform the given algebra A' into a commutative separable algebra A.

Algorithm 3.1 Transformation of the algebra

```
Input: An integral basis, $\{a'_1, a'_2, \ldots, a'_n\}$, of an algebra $A' \subset \mathbb{F}_q(y)^{m \times m}$, where $\deg(a'_i) \leq \Delta$ for $1 \leq i \leq n$;
Output: The integral basis, $\{a_1, a_2, \ldots, a_t\}$, of a commutative separable subalgebra $A$ of $A'$ which includes all the idempotents of the center of $A'$;

1: $j = \lfloor \log_p n \rfloor$;
2: Compute the basis $\{c_1, c_2, \ldots, c_k\}$ of the center of $A'$,$C(A')$;
3: Compute the basis of the subalgebra generated by $\{c_1^{p^j}, c_2^{p^j}, \ldots, c_k^{p^j}\}$.
```

Theorem 25. *Given an integral basis $\{a'_1, a'_2, \ldots, a'_n\}$ of an algebra $A' \subset \mathbb{F}_q(y)^{m \times m}$, where deg($a'_i$) $\leq \Delta$ for $1 \leq i \leq n$, Algorithm 3.1 returns the correct result with probability $1 - \epsilon$, taking $O(n^4 m^2 \Delta)$ operations in $\mathbb{F}_q$.*

Proof. Step 2 here could be done by solving a linear system. First, we compute $a'_i a'_j - a'_j a'_i$ for $1 \leq i, j \leq n$, which will takes $O(n^2 \times m^\omega \times M(\Delta)) = O^{\sim}(n^2 m^\omega \Delta)$ operations. Then we solve the following linear system with nm^2 equations and n unknowns in $\mathbb{F}_q[y]$:

$$\sum_{j=1}^{n} c_i(a_i a_j - a_j a_i) = 0 \text{ for } i = 1, 2 \ldots, n,$$

which requires $O^{\sim}(nm^2 n r^{\omega-2} \Delta) = O^{\sim}(n^\omega m^2 \Delta)$ operations in $\mathbb{F}_q$ (r is the rank of our coefficients matrix, i.e. the dimension of C). The degree bound of c_i is $(n-r)\Delta \lceil \log_2(n-r) \rceil$, denoted by d, by Theorem 8.

Step 3 is equivalent to selecting a maximal linearly independent subset from

$\{c_1^{p^j}, c_2^{p^j}, \ldots, c_r^{p^j}\}$. The cost of computing all the $c_i^{p^j}$s for $1 \leq i \leq r$ is

$$\begin{aligned} & O(r \times (m^\omega M(d) + m^\omega M(2d) + \ldots + m^\omega M(2^{\log_2 p^j} d))) \\ = \; & O^{\sim}(r \times m^\omega d(1 + 2 + 4 + \ldots + p^j)) \\ = \; & O^{\sim}(m^\omega r p^j d). \end{aligned}$$

We need to select a maximal linearly independent subset from a set of r m^2-dimension vectors with degree bounded by $d \times p^j$, which will take $O(r \times m^2 \times \min(r, m^2) \times dp^j) = O(r^2 m^2 dp^j)$ operations in $\mathbb{F}_q$ by Theorem 7. When $q \geq \frac{n^2 \lceil \log_2 n \rceil \Delta + n\Delta}{\epsilon} \geq \frac{((n-r)\lceil \log_2 (n-r) \rceil + 1) r\Delta}{\epsilon} \geq \frac{(d+\Delta)\min(r, m^2)}{\epsilon}$, the algorithm works correctly with probability $1 - \epsilon$. So the total cost of this algorithm will be $O(n^\omega m^2 \Delta + m^\omega r p^j d + r^2 m^2 d p^j) = O^{\sim}(n^\omega m^2 \Delta + n^3 m^\omega \Delta + n^4 m^2 \Delta) = O^{\sim}(n^4 m^2 \Delta)$ operations.
□

From the proof of Theorem 25, the degree bound for the basis of the commutative separable algebra A is $n^2 \lceil \log_2 n \rceil \Delta$. Therefore the degree bound for the structure constants could be $n^3 \lceil \log_2 n \rceil \Delta$, denoted by d_c. (This upper bound could be attained from Theorem 8 easily.) We assume that the structure constants are integral in the rest of this subsection. The general case has even higher complexity. The key theorem about the idempotents is as follows.

Theorem 26. *Let A be an n-dimensional commutative separable algebra over the field $\mathbb{F}_q(y)$. Assume that the structure constants with respect to the basis $\{a_1, \ldots, a_n\}$ are from $\mathbb{F}_q[y]$ and their heights are limited by d_c. Then any idempotent $e \in A$ lies in the $\mathbb{F}_q$-space*

$$\left\{ \sum_{i=1}^{n} \frac{\alpha_i}{D} a_i \mid \alpha_i \in \mathbb{F}_q[y], \;\; deg_y \alpha_i \leq (3n-2)d_c \right\},$$

where D is the discriminant $disc_{\{a_1, a_2, \ldots, a_n\}} A$ in this subsection as a polynomial of degree $2nd_c$ with respect to y. It is well-known (see Bastida, [1],pp. 166-168) that D is nonzero for a commutative separable algebra A.

Thus every element in B is generated by $\{\frac{\alpha_i}{D} a_i \mid \alpha_i \in \mathbb{F}_q[y],\ \deg_y \alpha_i \leq (3n-2)d_c,\ 1 \leq i \leq n\}$ over $\mathbb{F}_p$. Hence so is its basis. Let $b = \alpha_1 \frac{a_1}{D} + \alpha_2 \frac{a_2}{D} + \ldots + \alpha_n \frac{a_n}{D}$. We want to solve $\{\alpha_1, \ldots, \alpha_n\}$ from $b^p = b$ as follows. Denote

$$\begin{aligned} \alpha_i & = \alpha_{i0} + \alpha_{i1} y + \ldots + \alpha_{i((3n-2)d_c)} y^{(3n-2)d_c} (\in \mathbb{F}_q[y]) \\ & = \textstyle\sum_{j=0}^{l} \alpha_{i0j} z^j + \sum_{j=0}^{l} \alpha_{i1j} z^j y + \ldots + \sum_{j=0}^{l} \alpha_{i((3n-2)d_c)j} z^j y \\ & = \textstyle\sum_{k=0}^{(3n-2)d_c} \sum_{j=0}^{l} \alpha_{ikj} z^j y^k, \end{aligned}$$

where $\alpha_{ikj} \in \mathbb{F}_p$, z is the generating element of $\mathbb{F}_q$ over $\mathbb{F}_p$ and $q = p^{l+1}$ for $1 \leq i \leq n$, $0 \leq k \leq (3n-2)d_c$ and $0 \leq j \leq l$. Then

$$\alpha_i^p = \sum_{k=0}^{(3n-2)d_c} \sum_{j=0}^{l} \alpha_{ikj} z^{pj} y^{pk} = \sum_{k=0}^{(3n-2)d_c} \sum_{j=0}^{l} \sum_{u=0}^{l} \alpha_{ikj} c_{ju} z^u y^{pk},$$

if we rewrite $z^{pj} = \sum_{u=0}^{l} c_{ju} z^u$ for $1 \le i \le n$ and $0 \le j \le l$. Note that A is commutative, so $b^p = b$ will be

$$\sum_{i=1}^{n} \sum_{k=0}^{(3n-2)d_c} \sum_{j=0}^{l} \sum_{u=0}^{l} \alpha_{ikj} c_{ju} z^u y^{pk} (\frac{a_i}{D})^p = \sum_{i=1}^{n} \sum_{k=0}^{(3n-2)d_c} \sum_{j=0}^{l} \alpha_{ikj} z^j y^k \frac{a_i}{D},$$

which is a linear system of $m^2 \times \max\{p\Delta + p(3n-2)d_c,\, 2nd_c(p-1) + (3n-2)d_c\} \times l = O(m^2 npd_c l)$ equations with $n \times (3n-2)d_c \times l = O(n^2 d_c l)$ unknowns over $\mathbb{F}_p$, where Δ is the degree bound for the entries in a_i. Solving this system will take $O(m^2 npd_c l \times (n^2 d_c l)^{\omega-1}) = O(m^2 n^{2\omega-1} d_c^{\omega} l^{\omega} p) = O^{\sim}(m^2 n^{5\omega-1} \Delta^{\omega} p (\log_p q)^{\omega})$.

So when $p \le 2nd_c$, the algorithm solves the linear system directly with the cost $O^{\sim}(m^2 n^{5\omega+3} \Delta^{\omega+1} (\log_p q)^{\omega})$ in $\mathbb{F}_p$.

When $q > 2nd_c$, since p is not polynomial in the size of our input ($\log p$ here), we need to modify the algorithm. Let I to be the ideal of $\mathbb{F}_q[y]$ generated by $y - c$, where c is the element in $\mathbb{F}_q$ such that its discriminant $D(c) \ne 0$ and M denote the $\mathbb{F}_q[y]$-submodule of A generated by $\{\frac{1}{D}a_1, \frac{1}{D}a_2, \ldots, \frac{1}{D}a_n\}$. The existence of c is guaranteed by $q > 2nd_c$. First we compute the primitive idempotents of the $\mathbb{F}_q$-algebra M/IM with the basis $\{a_1, \ldots, a_n\}$ fixed. Then we lift the idempotents to those in $M/I^k M$ where $k = n(3n-1)d_c + 1$, denoted by $\{\bar{e}_1, \bar{e}_2, \ldots, \bar{e}_r\}$. Ivanyos [24] proved that if we can find an element $u \ne 0$ of degree not greater than $(3n-2)d_c$ in the $\mathbb{F}_q$-algebra generated by $\{\bar{e}_2, \bar{e}_2, \ldots, \bar{e}_r\}$, then u is a zero divisor in A. So A can be split into the proper ideal Au and its complement. Otherwise, A is simple. The above method gives the minimal ideals of A in at most n iterations.

The lifting procedure is granted by the following lemma.

Lemma 27. *Let S be a commutative ring with identity and let J be an ideal of S. Let us define the following lifting mapping $l: S \to S$:*

$$l: \; x \mapsto x^2(3 - 2x).$$

Then for all elements $x, y \in S$, such that $x^2 \equiv x$ and $y^2 \equiv y$ (mod J) and for all positive integers j, we have the following assertions:

- *$l^j(x)^2 \equiv l^j(x)$ (mod J^{2j});*
- *if $x \equiv y$ (mod J), then $l^j(x) \equiv l^j(y)$ (mod J^{2j});*
- *if $xy \equiv 0$ (mod J), then $l^j(x) l^j(y) \equiv 0$ (mod J^{2j});*
- *if $xy \equiv 0$ (mod J), then $l^j(x+y) \equiv l^j(x) + l^j(y)$ (mod J^{2j}).*

To be more detailed, now assume we have primitive idempotents in M/IM, $e_i^{(1)} = c_{i1}\frac{1}{D}a_1 + c_{i2}\frac{1}{D}a_2 + \ldots + c_{in}\frac{1}{D}a_n$ for $1 \le i \le r$, where $c_{ij} \in \mathbb{F}_q[y]/(y-c) = \mathbb{F}_q$. Then we need to compute the following:

1. The structure constants of the algebra A: solve n^2 linear systems generated by $a_i a_j = \sum_{u=1}^{n} \phi_{iju} a_u$, each of which has m^2 equations and n unknowns. Similarly, $\deg(\phi_{iju}) \leq d_c$ for $1 \leq i, j, u \leq n$;

2. For $1 \leq i \leq \lceil \log_2 k \rceil$, $\frac{1}{D}$ mod I^{2^i}. Use the EEA to compute the greatest common divisor of D and $(y-c)^{2^i}$, $sD + t(y-c)^{2^i} = 1$. So $\frac{1}{D} = s_i$ mod I^{2^i}.

Now we can compute a step in the lifting procedure. Suppose we already have the lifted idempotents $\{e_1^{(j)}, e_2^{(j)}, \ldots, e_r^{(j)}\}$ in $M/I^{2^j}M$ for a given j, where $e_i^{(j)} = c_{i1}^{(j)} \frac{1}{D} a_1 + c_{i2}^{(j)} \frac{1}{D} a_2 + \ldots + c_{in}^{(j)} \frac{1}{D} a_n$ with $\deg(c_{iu}^{(j)}) \leq 2^j$ for $1 \leq i \leq r$ and $1 \leq u \leq n$. $e_i^{(j+1)} = (e_i^{(j)})^2(3I - 2e_i^{(j)}) = 3(e_i^{(j)})^2 - 2(e_i^{(j)})^3$. Compute $(e_i^{(j)})^2 = \frac{1}{D} \sum_{u,v=1}^{n} c_{iu}^{(j)} c_{iv}^{(j)} \sum_{s=1}^{n} \phi_{ijs} \frac{1}{D} a_s$ mod $I^{2^{j+1}}$and $(e_i^{(j)})^3$ in the same way. After $\lceil \log_2 k \rceil$ rounds we get the idempotents in $M/I^{2^{\lceil \log_2 k \rceil}}M$, $\{\bar{e}_1, \bar{e}_2, \ldots, \bar{e}_r\}$, where $\bar{e}_i = \bar{c}_{i1} \frac{1}{D} a_1 + \bar{c}_{i2} \frac{1}{D} a_2 + \ldots + \bar{c}_{in} \frac{1}{D} a_n$ with $\deg(\bar{c}_{ij}) \leq 2^{\lceil \log_2 k \rceil}$and $\bar{c}_{ij} = c_{ij0} + c_{ij1} y + \ldots + c_{ij2^{\lceil \log_2 k \rceil}} y^{2^{\lceil \log_2 k \rceil}}$. We will solve a linear system over $\mathbb{F}_q$ as follows:

$$c_{2it} x_2 + \ldots + c_{nit} x_n = 0 \text{ for} 1 \leq i \leq n \text{ and } (3n-2) d_c \leq t \leq 2^{\lceil \log_2 k \rceil}.$$

For its complexity, we will not account for the cost of computing the initial idempotents over $\mathbb{F}_q$ since it is small, based on the paper of Eberly et al. [7]. The computation of structure constants will take $O^{\sim}(m^2 n^3 \Delta + n^2 n^{\omega} n^2 \Delta + n^2 m^{\omega} n^2 \Delta) = O^{\sim}(n^{\omega+4}\Delta)$. The computation of $\frac{1}{D}$ mod I^{2^i} will take $O^{\sim}(M(\max\{2nd_c, 2^{\lceil \log_2 k \rceil}\}))$, thereby $O^{\sim}(n^2 d_c) = O^{\sim}(n^5 \Delta)$. Computing $(e_i^{(j)})^2$ and $(e_i^{(j)})^3$ will take $O(n^3 M(2^{j+1}))$, i.e. $O^{\sim}(n^3 2^j)$. So computing the lifted idempotents $\{\bar{e}_1, \bar{e}_2, \ldots, \bar{e}_r\}$ will take

$$O^{\sim}(\sum_{j=1}^{\lceil \log_2 k \rceil} n^3 2^j) = O^{\sim}(kn^3) = O^{\sim}(n^8 \Delta).$$

At last, solving the linear system above costs $O^{\sim}(n^3 d_c n^{\omega-1}) = O^{\sim}(n^{\omega+5}\Delta)$ in $\mathbb{F}_q$. So the complexity of finding a zero divisor is $O^{\sim}(n^8 \Delta)$ in $\mathbb{F}_q$ when $p \geq 2pd_c$.

We need to keep splitting the algebra until all the subalgebras are simple. Note that we do not need to compute the idempotents again, and also the subalgebra split from u is generated by the idempotents with nonzero coefficients in the expression of u. Besides, the degree bound of the subalgebra is the same as in A. So we need to solve the linear system at most n times. Thus the total cost of this algorithm is $O^{\sim}(n^9 \Delta)$.

Theorem 28. *The algorithm given by Ivanyos et al. [24] for computing the Wedderburn decomposition of n-dimensional matrix algebra $A \subset \mathbb{F}_q(y)^{m \times m}$ with the degree bound Δ for the basis of A costs*

1. $O^{\sim}(m^2 n^{5\omega+3} \Delta^{\omega+1} (log_p q)^{\omega})$ *if* $p \leq 2n^4 \lceil log_2 n \rceil \Delta$*;*

2. $O^{\sim}(n^9\Delta)$ if $q \geq 2n^4 \lceil log_2 n \rceil \Delta + 1$.

3.1.3 Wedderburn Decomposition using Primitive Idempotents

We discuss the close relationship between the computation of idempotents and the decomposition of the semisimple algebra in this part. (Actually the semisimple condition here is not necessary. It is also the case for a general associative algebra, as shown at the end of this subsection.)

Given a semisimple algebra $A \subset F^{m\times m}$, recall that under some isomorphic mapping $A \cong A_1 \oplus A_2 \oplus ... \oplus A_k$ for simple algebras $A_1, A_2, ..., A_k \subset A$, and each component satisfies $A_i \cong D_i^{t_i \times t_i}$. Suppose we are given a complete set of primitive idempotents $\{\omega_1, \omega_2, \ldots, \omega_t\}$ such that $\omega_i\omega_j = 0$ for $i \neq j$ and that $\omega_1 + \omega_2 + \ldots \omega_t = I_m$. There exists a nonsingular matrix U such that

$$\hat{\omega}_i = U^{-1}\omega_i U = \begin{bmatrix} \Delta_{i1} & & & & \\ & \ddots & & & \\ & & \Delta_{ii} & & \\ & & & \ddots & \\ & & & & \Delta_{it} \end{bmatrix} \in F^{m\times m},$$

such that $\Delta_{ij} \in F^{d_j \times d_j}$ is the identity matrix when $j = i$ and zero matrix when $j \neq i$. The same thing happens to the images of ω_i. Thus, there is a new isomorphic mapping ϕ from A to $A_1 \oplus A_2 \oplus ... \oplus A_k$, such that

$$\phi(\omega_j) = \sum_{i=1}^{k} \begin{bmatrix} \Delta_{k_{ij}1} & & & & \\ & \ddots & & & \\ & & \Delta_{k_{ij}k_{ij}} & & \\ & & & \ddots & \\ & & & & \Delta_{k_{ij}t_i} \end{bmatrix} \in \bigoplus_{i=1}^{k} D_i^{t_i \times t_i},$$

where Δ_{ij} is 1 when $j = i$ and $i \neq 0$, and 0 otherwise. Note that $\{\omega_1, \omega_2, \ldots, \omega_t\}$ is the set of primitive idempotents, so there is a unique i such that $k_{ij} \neq 0$ and the dimension of Δ_{ij} is 1. Now we can split and rewrite the idempotents as $\{\omega_{i1}, \omega_{i2}, \ldots, \omega_{it_i}\}$ for $i = 1, 2, \ldots, k$, where $\{\omega_{i1}, \omega_{i2}, \ldots, \omega_{it_i}\}$ are all the primitive idempotents in the ith simple component. We call such a group an *equivalent* class of idempotents associated with the same simple component. Two idempotents in one equivalent class of idempotents associated with the same simple component are called *equivalent* in this thesis. Denote $\bar{\omega}_i = \sum_{j=1}^{t_i} \omega_{ij}$. It is obvious that $A_i = \bar{\omega}_i A \bar{\omega}_i$ for $1 \leq i \leq k$, which is the simple component of the algebra.

The remaining problem is deciding the equivalence classes of the primitive idempotents over $\mathbb{F}_q(y)$. We will first present a probabilistic algorithm and then modify it to be a deterministic one.

Algorithm 3.2 Determining the Equivalent Idempotents

```
Input: Given an integral basis $\{a_1, a_2, \ldots, a_n\}$ of matrix
       algebra $A \subset \mathbb{F}_q(y)^{m\times m}$, and a set of integral elements $\{\omega'_1, \omega'_2, \ldots, \omega'_t\}$
       of $A$ such that each $\omega'_i$ is in a unique simple component of $A$, where
       $q \geq \frac{t}{\epsilon}$, $\deg(a_i) \leq \Delta$ and $\deg(\omega'_j) \leq d\Delta$ for1 $\leq i \leq n$ and $1 \leq j \leq t$;
Output: The indices of all the equivalent elements of $\{\omega'_1, \omega'_2, \ldots, \omega'_t\}$ in each
       class;

Randomly choose element $(c_1, c_2, \ldots, c_n) \in \mathbb{F}_q^n$. Let $\alpha = \sum_{i=1}^n c_i a_i$;
Compute $t_{ij} = \omega'_i \alpha \omega'_j$ for $i, j = 1, 2, \ldots t$ and $i < j$;
$I = \{1, 2, \ldots t\}$; $I_i = \emptyset$ for $1 \leq i \leq t$;
while $I \neq \emptyset$ do
    Pick the smallest $i \in I$
    $I_i = \{j \mid t_{ij} \neq 0,\ j = i, i+1, \ldots, t\}$;
    $I = I \backslash I_i$;
end while
```

It is easy to see that if ω_i and ω_j are in different equivalent classes then for any element $\alpha \in A$, $\omega_i \alpha \omega_j = 0$ for $1 \leq i, j \leq t$. Our algorithm takes advantage of this property. We will analyze the case when $F = \mathbb{F}_q(y)$. It is even simpler when F is finite. Besides, we give an general algorithm for elements each of which is in a unique simple component of A.

Theorem 29. *Given an integral basis $\{a_1, \ldots, a_n\}$ of matrix algebra $A \subset \mathbb{F}_q(y)^{m\times m}$, and a set of integral elements $\{\omega'_1, \omega'_2, \ldots, \omega'_t\}$ of A such that each ω'_i is in a unique simple component of A, where $q \geq \frac{t}{\epsilon}$, deg($a_i$) $\leq \Delta$ and deg(ω'_j) $\leq d\Delta$ for $1 \leq i \leq n$ and $1 \leq j \leq t$, Algorithm 3.2 computes the equivalence classes of $\{\omega'_1, \omega'_2, \ldots, \omega'_t\}$ correctly with probability at least $1 - \epsilon$ taking $O^\sim(t^2 m^\omega d\Delta)$ operations in $\mathbb{F}_q$.*

Proof. First we prove the algorithm gives a correct result with large probability. Given i and j, if ω_i and ω_j are in different equivalent class then $\omega_i \alpha \omega_j = 0$. So the algorithm works correctly in this case. When ω_i and ω_j are in a same equivalence class, we will return the wrong output when $\omega_i \alpha \omega_j = 0$, i.e., the entry at position (i, j) of matrix α is 0. We claim that there is at least one element in the basis of A such that $\omega_i \alpha \omega_j \neq 0$. If not, then there is not matrix in A whose image is e_{ij}, the matrix with position (i, j) being 1 and others being 0, in the simple component of ω_i and ω_j. Suppose $\{a_1, a_2, \ldots, a_s\}$ are all the elements in the basis of A such that $\omega_i a_t \omega_j \neq 0$ for $t = 1, 2, \ldots, s$. Selecting a random element α of A such that $\omega_i \alpha \omega_j = 0$ is equivalent to selecting $\{c_1, c_2, \ldots, c_s\}$ from $\mathbb{F}_q$ such that

$$\sum_{t=1}^{s} c_t \bar{a}_t = 0,$$

where $\bar{a}_t = \omega_i a_t \omega_j \neq 0$ for $t = 1, 2, \dots, s$. So there at most q^{s-1} solutions for this equation and our background set is of q^s element. So by Lemma 3 (Schwartz-Zippel Lemma) the probability that this equation holds, i.e. the probability that this algorithm returns correct output, is at least $1 - \frac{1}{q}$. For the whole algorithm, the probability of correctness is $(1 - \frac{1}{q})^t \geq 1 - \frac{t}{q} \geq 1 - \epsilon$.

To analyze the cost, first we need to compute α with cost $O(m^2\Delta)$. The dominating part is computing t_{ij}, which has cost $O(t^2 \times m^\omega \times M(d\Delta)) = O^\sim(t^2 \times m^\omega \times d\Delta) = O^\sim(t^2 m^\omega d\Delta)$ operations in $\mathbb{F}_q$.

□

From the claim in the proof above, we could modify Algorithm 3.2 to be a deterministic one.

Algorithm 3.3 Determine the Equivalent idempotents

Input: A set of integral elements $\{\omega'_1, \omega'_2, \dots, \omega'_t\}$ of A such that each ω'_i is in a unique simple component of A.
The basis, $\{a_1, a_2, \dots a_n\}$, of the matrix algebra $A \subset \mathbb{F}_q(y)^{m\times m}$, where $\deg(a_i) \leq \Delta$ and $\deg(\omega'_j) \leq d\Delta$ for $1 \leq i \leq n$ and $1 \leq j \leq t$.

Output: The sums of all the equivalent elements of $\{\omega'_1, \omega'_2, \dots, \omega'_t\}$.

```
$I = \{1, 2, \dots t\}$;
while $I \neq \emptyset$ do
  for $i \in I$ do
    $\bar{\omega}_i = \omega'_i$;
    $I = I \backslash \{i\}$;
    for $j = i + 1$ to $t$ do
      for $t = 1$ to $n$ do
        if $\omega'_i a_t \omega'_j \neq 0$ then
          $\bar{\omega}_i = \bar{\omega}_i + \omega'_j$;
          $I = I \backslash \{j\}$;
          Break out the inner "for" loop of $t$;
        end if
      end for
    end for
  end for
end while
Return all the $\bar{\omega}_i$s.
```

It is easy to verify the correctness of Algorithm 3.3 from the claim that there is at least one element a in the basis of A such that $\omega_i a \omega_j \neq 0$ for $1 \leq i, j \leq t$. For the cost, we do the matrix multiplication for at most $t^2 n$ times which cost $O(t^2 n \times m^\omega \times M(d\Delta)) = O^\sim(t^2 n m^\omega d\Delta)$.

Note that we do not really need the idempotents here to be primitive. The algorithms and the conclusions also hold for the set of elements where each element is exactly in one equivalent simple component. We will use this case in our algorithm for the Wedderburn decomposition. Also we do not need the ω_is to be idempotents. But we do need that the sum of all the equivalent ω_is is a unit in some unique simple component. In Subsection 3.1.4 and 3.1.5 we will take advantage of these trivial properties and remove the denominator of the idempotents.

3.1.4 Computing the Idempotents

From Subsection 3.1.3 we know that it is sufficient to compute the primitive idempotents for the Wedderburn decomposition. Moreover, we do not really need the elements to be primitive or idempotent but just to satisfy a special condition: each element is exactly in one simple component and the sum of all the equivalent elements is a unit in some unique simple component. We will discuss a way of computing such a set of elements over $\mathbb{F}_q(y)$ in this subsection.

Recall the Wedderburn's Structure Theorem that

$$A = A_1 \oplus A_2 \oplus ... \oplus A_t.$$

Our idea is inspired by Eberly [8] that it is with large probability to select a "good" splitting element from the algebra. By the "good" here, we mean the expected set of idempotents can be computed in the following way from this element. Suppose the element $a \in A$ satisfies that its minimal polynomial $f \in F[x]$ has a factorization $f = f_1 f_2 \dots f_t$ into two or more monic, pairwise relatively prime $f_i \in F[x]\backslash F$. For $1 \leq i \leq t$, use the Extended Euclidean Algorithm to construct $h_i \in F[x]$ such that $h_i \equiv 1 \bmod f_i$, $h_i \equiv 0 \bmod f_j$ for $j \neq i$, and assign $\omega_i = h_i(\alpha) \in A$. It is easy to prove that $\{\omega_1, \dots, \omega_t\}$ is a set of pairwise orthogonal idempotents and $\omega_1 + \omega_2 + \dots + \omega_t = 1 \in A$. Since $h_i \equiv 1 \bmod f_i$, $h_i \equiv 0 \bmod f_j$ for $i \neq j$, then for $1 \leq i, j \leq t$

$$e_i e_i = h_i(a)^2 = ((t_i f_i + 1) k_i f_1 \dots f_{i-1} f_{i+1} \dots f_k)(a) = (t_i f)(a) + h_i(a) = h_i(a) = e_i,$$

and when $i \neq j$

$$e_i e_j = h_i(a) h_j(a) = (k_i k_j \frac{f}{f_i f_j} f)(a) = 0.$$

But it is not guaranteed that they are primitive, or that each of them is exactly in one simple component. So we need to add more requirement to this "good" element.

Suppose A is a semisimple algebra over $\mathbb{F}_q(y)$ with the basis $\{a_1, a_2, \dots, a_n\}$, where $\deg(a_i) \leq \Delta$, $i = 1, 2, \dots n$. If $\alpha \in A$, then $\alpha = \alpha_1 \oplus \alpha_2 \oplus ... \oplus \alpha_k$, where $\alpha_i \in A_i$. The minimal polynomial of α over F, denoted by $\text{minpoly}_F(\alpha)$, is the least common multiple of the minimal polynomials of all the components:

$$\text{minpoly}_F(\alpha) = \text{lcm}(\text{minpoly}_F(\alpha_1), \text{minpoly}_F(\alpha_2), ..., \text{minpoly}_F(\alpha_k)).$$

We define the "good" element in a formal and clear way as follows.

Definition 30. Let A be a semisimple algebra. An element $\alpha \in A$ is called *decomposing element* in A, if the minimal polynomials of its simple components are pairwise co-prime.

Such decomposing element will provides us a good set of idempotents, i.e. pairwise orthogonal, $\omega_1 + \omega_2 + \ldots + \omega_t = 1$ and each one is exactly in one simple component of A according to the following Theorem 31.

Theorem 31. *If A is a semisimple algebra, and $\alpha \in A$ is the element such that the degree of its minimal polynomial is maximal. Then α is a decomposing element.*

To prove Theorem 31 we will need the following trivial but useful lemma.

Lemma 32. *Let D be a division ring and $F \subset D$ a subfield. The minimal polynomial of $d \in D^{t\times t}$ over F is $f(x) \in F[x]$. Then the minimal polynomial of $d + bI$ over F is $f(x - b) \in F[x]$ for $b \in F$.*

Proof. [Proof of Theorem 31] Suppose α is an element such that the degree of its minimal polynomial is maximal. Recall that $\alpha = \alpha_1 \oplus \alpha_2 \oplus \ldots \oplus \alpha_k$. Now suppose contrarily that $\{\text{minpoly}_F(\alpha_1), \text{minpoly}_F(\alpha_2),\ldots, \text{minpoly}_F(\alpha_k)\}$ are *not* pairwise co-prime.

$$\text{minpoly}_F(\alpha) = \text{lcm}(\text{minpoly}_F(\alpha_1), \text{minpoly}_F(\alpha_2), ..., \text{minpoly}_F(\alpha_k)).$$

We will derive a contradiction. Let $f_i(x) = \text{minpoly}_F(\alpha_i)$. By Lemma 32 the minimal polynomial of $\alpha_i + bI$ is $f_i(x - b) \in F[x]$ for $1 \leq i \leq k$, where $b \in F$ and I is the identity matrix. Since F is infinite, we can choose $b_2 \in F$ such that $f_2(x - b_2)$ and f_1 are co-prime. Then we can choose $b_3 \in F$ such that the minimal polynomial of $\alpha_3 + b_3 I$ and either $f_2(x - b_2)$ or f_1 are co-prime. Similarly, we can choose $b_4, ..., b_k$ such that the minimal polynomials of $\alpha_1, \alpha_2 + b_2 I, ..., \alpha_k + b_k I$ are pairwise co-prime in $F[x]$. Let

$$\alpha' = \alpha_1 \oplus \alpha_2 + b_2 I \oplus ... \oplus \alpha_k + b_k I.$$

Then

$$\begin{aligned}
&\deg(\text{minpoly}_F(\alpha')) \\
&= \deg(\text{lcm}(\text{minpoly}_F(\alpha_1), \text{minpoly}_F(\alpha_2 + b_2 I), ..., \text{minpoly}_F(\alpha_k + b_k I))) \\
&= \deg(\text{minpoly}_F(\alpha_1) \times \text{minpoly}_F(\alpha_2 + b_2 I) \times ... \times \text{minpoly}_F(\alpha_k + b_k I)) \\
&= \deg(\text{minpoly}_F(\alpha_1)) + ... + \deg(\text{minpoly}_F(\alpha_k + b_k I)) \\
&= \deg(\text{minpoly}_F(\alpha_1)) + \deg(\text{minpoly}_F(\alpha_2)) + ... + \deg(\text{minpoly}_F(\alpha_k)) \\
&> \deg(\text{minpoly}_F(\alpha))
\end{aligned}$$

This gives a contradiction.

□

Now we can show that it is with large probability to select a decomposing element randomly from the matrix algebra over $\mathbb{F}_q(y)$, based on Theorem 34.

Lemma 33. *If M is a $n \times m$ matrix on $F[y, y_1, ..., y_n]^{n\times m}$, then its rank is d over $F[y, y_1, ..., y_n]$ if and only if d is the maximal rank over F of the set*

$$\{M|_{(y,y_1,...,y_n)=(c,s_1,...,s_n)} \in F^{n\times m} \mid c, s_1, ..., s_n \in F\}.$$

Theorem 34. *Let $A \subset \mathbb{F}_q(y)^{m\times m}$ be a semisimple algebra of dimension n, with integral basis $a_1, a_2, ..., a_n \in \mathbb{F}_q[y]^{m\times m}$. If the elements $s_1, s_2, ..., s_n$ are randomly chosen uniformly and independently from $\mathbb{F}_q$, where $q \geq \frac{nm^2}{\epsilon}$, then the element*

$$s_1a_1 + s_2a_2 + ... + s_na_n$$

is a decomposing element with probability at least $1 - \epsilon$.

Proof. Denote $F = \mathbb{F}_q(y)$. By Theorem 31 we only need to compute the probability of choosing an element with maximal degree for its minimal polynomial. Note that from Theorem 31, the decomposing element exists. Suppose elements $\hat{s}_1, \hat{s}_2, ..., \hat{s}_n$ is the linear coordinates in F such that the element $\hat{s} = \hat{s}_1a_1 + \hat{s}_2a_2 + ... + \hat{s}_na_n$ is an element with maximal degree for its minimal polynomial in A. The degree of its minimal polynomial is d_m.

Let $y_1, y_2, ..., y_n$ be arguments of σ in F such that

$$\sigma = y_1a_1 + y_2a_2 + ... + y_na_n \in F[y_1, y_2, ..., y_n]^{m\times m},$$

where $\deg_{y_i}(\sigma) = 1$ and the total degree $\deg(\sigma) = 1$ with respect to y_i for $1 \leq i \leq n$. So $\hat{s} = \sigma(\hat{s}_1, \hat{s}_2, ..., \hat{s}_n)$. Now consider the matrix equation $f(\sigma) = \sigma^{d_m} + z_{d_m-1}\sigma^{d_m-1} + ... + z_1\sigma + z_01 = 0$. Let $v_i = \mathrm{Vec}(\sigma^i) \in F[y_1, y_2, ..., y_n]^{m^2\times 1}$ for $1 \leq i \leq d_m + 1$, $M = (I, v_1, ..., v_{d_m-1})$. The corresponding linear system with $\{z_0, \ldots, z_{d_m-1}\}$ as unknowns is

$$M \begin{pmatrix} z_0 \\ z_1 \\ : \\ z_{d_m-1} \end{pmatrix} = v_{d_m}.$$

From the existence of the decomposing element, there is $(y_1, y_2, \ldots, y_n) = (\hat{y}_1, ..., \hat{y}_n)$ such that this system has its unique solution. By Lemma 33 this indicates that d_m is the maximal rank of M over F. Similarly, let $v_i = \mathrm{Vec}(\sigma^i) \in F[y_1, y_2, ..., y_n]^{m^2}$ for $i \geq 1$, $M_t = (I, v_1, ..., v_{d_m+t})$ for $t \geq 0$. Since d_m is the largest degree, $\mathrm{rank}(M_t) = d_m$ for $t \geq 0$ as well. So if the elements $s_1, s_2, ..., s_n$ are randomly chosen uniformly and independently from F_q, and $\mathrm{rank}(M|_{((y_1,...,y_n)=(s_1,...,s_n))}) < d_m$, then $f(s_1, ..., s_n) = \det(M)|_{(y_1,y_2,...y_n)=(s_1,s_2,...,s_n)} = 0$, where $f = \det(M) \in F[y_1, y_2, \ldots, y_n]$.

Now it remains to find the upper bound for the degree of f. It is obvious that the degree of the minimal polynomial of any $\alpha \in A$ is less than m. That is, $d_m \leq m$. $\deg_{y_i}(\sigma^k) \leq k$ since $\deg_{y_i}(\sigma) \leq 1$ for $1 \leq i \leq d_m$. So $\deg_{y_i}(M) \leq d_m \leq m$ and $\deg_{y_i}(f) \leq m^2$ for $1 \leq i \leq d_m$. So the total degree $\deg(f) \leq nm^2$.

By the Schwartz-Zippel lemma, if $q \geq nm^2/\epsilon$ and s_i is selected randomly from a subset of F of size q for $1 \leq i \leq n$, then the probability to get $s_1, s_2, ..., s_n$ such that $f(s_1, ..., s_n) = \det(M)|_{(y_1,y_2,...y_n)=(s_1,s_2,...,s_n)} = 0$ is less than ϵ. So the element $s_1a_1 + s_2a_2 + ... + s_na_n$ is a decomposing element with probability at least $1 - \epsilon$.

□

Theorem 34 gives us a hint of how to compute a good set of idempotents: choose a random element from the algebra and compute its respond set of idempotents. This idea yields the following algorithm.

Algorithm 3.4 Compute a good set of idempotents

```
Input: An integral basis, ${a_1, a_2, . . . a_n}$ , of the algebra $A \subset \mathbb{F}_q(y)^{m\times m}$, where $\deg(a_i) \leq \Delta$ for $1 \leq i \leq n$ and $q \geq \max\{\frac{2nm^2}{\epsilon}, \frac{2m^2(m-1)^2\Delta^2}{\epsilon}, 3m^2\Delta, 7m^2\Delta\}$;

Output: A set of orthogonal idempotents, $\{\omega_1, \omega_2, . . . , \omega_t\}$, such that each idempotent is in a unique simple component;

Select a random vector $(c_1, c_2, . . . , c_n) \in \mathbb{F}_q^n$;
$\alpha = c_1a_1 + c_2a_2 + . . . + c_na_n$;
Compute the minimal polynomial of $\alpha$, $f \in \mathbb{F}_q[y][x]$;
Factor the bivariate polynomial $f = f_1f_2 . . . f_t \in \mathbb{F}_q[y][x]$;
Compute $s_{i,}t_i \in \mathbb{F}_q(y)[x]$ such that $s_if_i + t_i\frac{f}{f_i} = 1$ for $i = 1, 2, . . . , t$;
Return $\omega_i = h_i(\alpha) = I - s_i(\alpha)f_i(\alpha)$ for $i = 1, 2, . . . , t$.
```

Theorem 35. *Given an integral basis $\{a_1, a_2, . . . a_n\}$ of the algebra $A \subset \mathbb{F}_q(y)^{m\times m}$, where deg($a_i$) $\leq \Delta$ and $q \geq$ max$\{\frac{2nm^2}{\epsilon}, \frac{2m^2(m-1)^2\Delta^2}{\epsilon}, 3m^2\Delta, 7m^2\Delta\}$, Algorithm 3.4 computes the good set of idempotents, $\{\omega_1, \omega_2, . . . , \omega_t\}$, where $\omega_i \in \mathbb{F}_q(y)^{m\times m}$ has a same denominator for all of its entries and the degrees of the numerators and denominators of all the entries are at most $m^2\Delta$ for $1 \leq i \leq t$, with probability at least $1 - \epsilon$, taking $O^{\sim}(m^6\Delta + m^{\omega+1}\Delta^{\frac{\omega+1}{2}})$ operations in $\mathbb{F}_q$.*

Proof. The degree bound of the entries of α is Δ. Since $q \geq \frac{2m^2(m-1)^2\Delta^2}{\epsilon}$, step 3 is correct with probability $1 - \frac{\epsilon}{2}$ by Theorem 19. Besides α is a decomposing element with probability at least $1-\frac{\epsilon}{2}$ according to Theorem 34. So Algorithm 3.4 is correct with probability at least $(1 - \frac{\epsilon}{2})^2 \geq 1 - \epsilon$.

For the complexity, the cost of step 3 is $O^{\sim}(m^{\omega+1}\Delta)$ by Theorem 19. By Theorem 4, from Lecerf [29] about the factorization of a bivariate polynomial, step 4 will take $O(m^{\omega+1}\Delta^{\frac{\omega+1}{2}}))$ plus the complexity of factorization of a polynomial in $\mathbb{F}_q[x]$ with degree at most $(m + 1)\Delta$ when $q \geq 2m^2\Delta + m\Delta + 1$. For step 5 and 6, we first use the fast Extended Euclidean Algorithm to compute s_i and t_i and then evaluate $1 - s_if_i$ at α for $1 \leq i \leq t$. Computing $\{\frac{f}{f_1}, . . . , \frac{f}{f_t}\}$ needs $O(mM(m\Delta))$ and when $q \geq 7m^2\Delta$ the fast Extended Euclidean Algorithm costs $O^{\sim}(m^3\Delta)$, gives a cost of $O^{\sim}(m^4\Delta)$ in total for $\{s_1, . . . , s_t\}$. Note that the degree bound of s_i is $m - \deg(f_i)$ in x for $1 \leq i \leq t$. For that of the numerators and denominators of all the coefficients, it is $m^2\Delta$ and note that all the coefficients in one h_i have unique denominator. So the degree bound for h_i is m in x and $m^2\Delta$ in y for the numerators and denominators of all the coefficients for $1 \le i \le t$. First, we compute $\{\alpha, \alpha_2, \ldots, \alpha^m\}$ which costs $O\tilde{}(m^{\omega+1}\Delta)$. Then evaluate $\{h_1, \ldots, h_t\}$ over α with cost $O\tilde{}(t \times m \times m^2 \times M(m^2\Delta)) = O\tilde{}(m^6\Delta)$. The total cost is then $O\tilde{}(m^6\Delta + m^{\omega+1}\Delta^{\frac{\omega+1}{2}})$. It is easy to verify the degree bound of ω_i from the degree bound of f_i and s_i for $1 \le i \le t$.

□

3.1.5 A Complete Algorithm

We present the complete algorithm for the Wedderburn decomposition in this subsection.

Algorithm 3.5 Wedderburn Decomposition

```
Input: An integral basis, $\{a_1, a_2, \ldots, a_n\}$, of a semisimple $m \times m$ matrix algebra over $\mathbb{F}_q(y)$, where deg$a_i \le \Delta$ for $1 \le i \le n$ and $q \ge \max\{\frac{4nm^2}{\epsilon}, \frac{4m^2(m-1)^2\Delta^2}{\epsilon}, 3m^2\Delta, 7m^2\Delta, \frac{4m}{\epsilon}, \frac{4nm(2m^2+4m+1)\Delta}{\epsilon}\}$.
Output: Bases $\{b_{i1}, b_{i2,}, \ldots, b_{it_i}\}$ of all the simple components for $1 \le i \le k$, where $k$ is the number of simple components;

Choose a vector $(c_1, c_2, \ldots, c_n)$ from $\mathbb{F}_q^n$ randomly. Let $\alpha = c_1a_1 + c_2a_2 + \ldots + c_na_n$;
Compute the minimal polynomial of $\alpha$, denoted by $f \in \mathbb{F}_q[y][x]$, via Algorithm 2.3;
Factor $f = f_1f_2 \ldots f_t$ using Algorithm from Lecerf [29], where each $f_i$ is the power of some irreducible polynomial and all the $f_i$s are pairwise co-prime;
Compute $s_{i,}t_i \in \mathbb{F}_q(y)[x]$ such that $s_if_i + t_i\frac{f}{f_i} = 1$ for $i = 1, 2, \ldots, t$;
Return $\omega_i = h_i(\alpha) = I - s_i(\alpha)f_i(\alpha)$ for $i = 1, 2, \ldots, t$.
Modify all the $\omega_i$s to be integral by removing their common denominators, denote by $\omega_i'$;
Determine of the equivalent classes of the idempotents $\omega_i$s via Algorithm 3.2 and compute all the possible $\bar{\omega}_i = \sum_{j \in I_i} \omega_j'$;
Compute the generating set of each simple component $G_i = \{\overline{\omega}_ia_1\overline{\omega}_i, \overline{\omega}_ia_2\overline{\omega}_i, \ldots, \overline{\omega}_ia_n\overline{\omega}_i\}$ for $1 \le i \le k$;
Select the bases $B_i = \{b_{i1}, b_{i2,}, \ldots, b_{it_i}\}$ from each $G_i$ for $1 \le i \le k$.
```

Theorem 36. *Given an integral basis $\{a_1, a_2, \ldots, a_n\}$, of a semisimple $m \times m$ matrix algebra over $\mathbb{F}_q(y)$, where deg(a_i) $\le \Delta$ for $1 \le i \le n$ and $q \ge$ max$\{\frac{4nm^2}{\epsilon}$, $\frac{4m^2(m-1)^2\Delta^2}{\epsilon}, 3m^2\Delta, 7m^2\Delta, \frac{4m}{\epsilon}, \frac{4nm(2m^2+4m+1)\Delta}{\epsilon}\}$, Algorithm 3.5 computes the decomposition of a semisimple algebra correctly with probability $1 - \epsilon$ taking $O(m^{\omega+4}\Delta + m^{\omega+1}\Delta^{\frac{\omega+1}{2}})$ operations in $\mathbb{F}_q$.*

Proof. From Theorem 34, it is with probability at least $1-\frac{\epsilon}{4}$ that α is a decomposing element. When α is a decomposing element, the minimal polynomials of the images of α over each simple components are pairwise co-prime. The probability of computing the minimal polynomial of α correctly in step 2 is $1-\frac{\epsilon}{4}$, by Theorem 19. Then we can compute the identity of each component as the sum of some equivalent idempotents in step 7 which works correctly with probability $1-\frac{\epsilon}{4}$ according Theorem 29. Note that this is a little different from the Wedderburn decomposition using good idempotents. Each $\bar{\omega}_i$ is not the identity but a unit in each simple component. It is easy to verify that the algorithm works correctly. Finally multiply the identities to the basis of A and then pick out a maximal linearly independent set, we get the basis of each component. The final step of computing the bases will be correct with probability $1-\frac{\epsilon}{4m}$ for each G_i, i.e. at least $1-\frac{\epsilon}{4}$ for all the bases, by Theorem 29. So the probability that Algorithm 3.5 returns the correct output is $1-\epsilon$.

The cost of step 2 is $O^{\sim}(m^{\omega+1}\Delta)$ by Theorem 19. Using Lecerf's efficient factorization algorithm, step 3 costs $O^{\sim}((m^2\Delta)^{\frac{\omega+1}{2}})$ by Theorem 4. For step 4 and 5, we first use the fast Extended Euclidean Algorithm to compute s_i and t_i and then evaluate $1-s_if_i$ at α, which take $O^{\sim}(m^6\Delta)$ in total [14]. Nothing happens in step 6. Determining the equivalent classes of idempotents will cost $O^{\sim}(t^2m^{\omega}d\Delta) = O^{\sim}(m^{\omega+4}\Delta)$ by Theorem 29. Computing $\bar{\omega}_i$ takes $O^{\sim}(m^4\Delta)$. There will be $2tm$ matrix multiplication in step 8 with complexity $O^{\sim}(tm \times m^{\omega} \times m^2\Delta) = O^{\sim}(m^{\omega+4}\Delta)$. The final step of selecting the bases will take $O^{\sim}(\sum_{i=1}^{k} t_i^2m^2 + t_im^2m^2\Delta) = O^{\sim}(nm^4\Delta)$ operations by Theorem 7. So the total cost of this algorithm is $O^{\sim}(m^{\omega+4}\Delta + m^{\omega+1}\Delta^{\frac{\omega+1}{2}})$ operations in $\mathbb{F}_q$.

□

3.2 Computation of the Radical

Recall that there is a semisimple subalgebra S of A such that $A = S + \mathrm{Rad}(A)$ and $S \cap \mathrm{Rad}(A) = (0)$. We will develop the algorithm for computing $\mathrm{Rad}(A)$ in this section. Since the radical of A is also a subalgebra of A, it could be represented by its basis. The first polynomial-time algorithm of computing the radical of finite-dimensional associate algebra is attributed to Friedl and Rónyai [9, 37] in 1985 and 1990, where an algorithmic characterisation of the radical is fully developed and used. It is indicated that the problem is much easier when the characteristic of the ground field is zero via the theorem of Dickson [5].

Theorem 37. *[Dickson] Let A be a finite-dimensional algebras of matrices over a field F such that $charF = 0$. Then*

$$Rad(A) = \{x \in A : Tr(xy) = 0 \text{ for every } y \in A\}.$$

This result shows that we can compute the radical by solving a system of linear equations over F.

We will discuss the much more complicated case when the field is of characteristic p in the remaining part of this section. Friedl and Rónyai [9, 37] present the first deterministic polynomial-time algorithm to compute the radical of an finite-dimension associative algebra over $\mathbb{F}_p$. The idea is extended later by Ivanyos, Rónyai and Szántó [24] to $\mathbb{F}_q(Y_1, Y_2, \ldots, Y_m)$. In 1997, Cohen, Ivanyos and Wales [4] generalized the idea of Rónyai [9, 37], reducing this problem to solving systems of semilinear equations over an arbitrary field taking polynomial operations over the ground field. The most current results are due to Ivanyos [23, 22]. A new algorithm in 1999 is for an arbitrary field, reducing the problem of computing the radical of a matrix algebra to computing the radical of a matrix commutative algebra. Note that the latter problem is much easier since the strongly nilpotent elements are equivalent to the nilpotent elements and $(x + y)^p = x^p + y^p$ holds in the algebra. The other new algorithm computes the radical in a probabilistic way using the primitive idempotents over $\mathbb{F}_q$. It reduces the cost to $O^{\sim}(mn^{\omega} + R(A))$.

We will focus on the case when $F = \mathbb{F}_q(y)$. It is a specific case of the problem addressed in Ivanyos, Rónyai and Szántó's paper [24]. In Subsection 3.2.1 we will analyze the complexity of the algorithm of Ivanyos et al. [24]. Then we will adapt the algorithm for reducing the problem to the commutative case by Ivanyos [22] to $F = \mathbb{F}_q(y)$ and analyze its complexity in Subsection 3.2.2. In the remaining part of this section we will then develop a new algorithm for computing the radical of the finite-dimensional matrix algebras over $\mathbb{F}_q(y)$ which is inspired by Ivanyos [23] and based on some intermediate result of Ivanyos, Rónyai and Szántó's paper [24]. First, we introduce the raw decomposition in Subsection 3.2.3, similar to the Wedderburn decomposition, which compute a set of primary subalgebras of A. This idea indeed reduces the degree bound of the method of Ivanyos [24], hence makes the following work efficient. Then in the Subsection 3.2.4 we compute the radical of each primary component to get $\mathrm{Rad}(A)$. There may be more efficient algorithms and improvements based on the computation of the primitive idempotents of A, which is still unsolved in this thesis.

3.2.1 Algorithm of Ivanyos, Rónyai and Szántó

Ivanyos et al. give an algorithm to compute the radical of an algebra over $\mathbb{F}_q(X_1, \ldots, X_m)$ in 1994 [24]. We will discuss its special case that $F = \mathbb{F}_q(y)$ in this thesis. The basic idea is to compute a sequence of ideals of the algebra which converge to its radical. Given A is a $m \times m$ matrix algebra over $\mathbb{F}_q(y)$ with basis $\{a_1, \ldots, a_n\}$. Let $g(x) \in \mathbb{F}_p[x]$ be the minimal polynomial of a generating element of $\mathbb{F}_q$ over $\mathbb{F}_p$. Define the local ring $R = \mathbb{Z}[x][y]_{(p)}/(g(x))$ then $R/pR = \mathbb{F}_q(y)$. For nonnegative integers j, define the following trace functions $\phi_j : M_{m\times m}(R) \to R/p^{j+1}R$ as

$$\phi_j : X \longmapsto \mathrm{Tr}(X^{p^j}) + p^{j+1}R.$$

Let $I_0 = A$, $I_i = \{x \in I_{i-1} |\ \phi_i(xy) = 0,\ \forall y \in A\}$ for $i \geq 1$. The main theorem about $\{I_0, I_1, \ldots\}$ is as follows.

Theorem 38. $\{I_0, I_1, \ldots\}$ *are ideals of* A *and they converge to the Jacobson radical of* A *in the sense that:* $I_j = Rad(A)$ *if* $j \geq \lfloor log_p m \rfloor$.

This theorem provides a way to compute the radical within $\log_p m$ iterations. However we need to be careful about the degree explosion which is discussed in the paper of Ivanyos et al. [24] as well. First let's see the procedure and its cost for the computation of I_e from the basis of I_{e-1}.

A simple but important and powerful result proved in [24] is that $\mathrm{Tr}((aX + bY)^{p^e}) \equiv a^{p^e}\mathrm{Tr}(X^{p^e}) + b^{p^e}\mathrm{Tr}(Y^{p^e}) \bmod p^{e+1}$ and $(X + pY)^{p^e} \equiv X^{p^e} \bmod p^{e+1}$. So when computing I_e we do not need to check all the elements in A but just its basis for the y in $\mathrm{Tr}((xy)^{p^e}) \equiv 0 \bmod p^{e+1}$.

Given the integral basis $\{b_1, b_2, \ldots b_k\}$ of I_{e-1} with degree bound d, let $x = c_1b_1 + c_2b_2 + \ldots + c_kb_k$ is in the I_e, where $\{c_1, \ldots, c_k\}$ are the coordinates from $F_q[y]$. First we compute $\mathrm{Tr}((b_ia_j)^{p^e})$, where $p^e \leq m$ for $1 \leq i \leq k$ and $1 \leq j \leq n$. Then we compute $c_{ij} = \frac{1}{p^e}\mathrm{Tr}((b_ia_j)^{p^e}) = \frac{1}{p^e}\mathrm{Tr}(((b_ia_j)^p)^{p^{e-1}}) = \frac{1}{p^i}\mathrm{Tr}((b_i(a_j(b_ia_j)^{p-1}))^{p^{e-1}}) = \frac{1}{p^i}\mathrm{Tr}((b_ic)^{p^{e-1}})$ where $c = a_j(b_ia_j)^{p-1}$ for $1 \leq i \leq k$ and $1 \leq j \leq n$. Since $\{b_1, \ldots b_k\}$ are in I_{e-1} and $c \in A$, so the c_{ij}s are integral in $R/p^{e+1}R$ with its degree less than $2p^ed$. Rewrite c_{ij} into the following form.

$$\begin{aligned} c_{ij} = \quad & (c_{ij0} + c_{ijp^i}y^{p^e} + \ldots) + (c_{ij1} + c_{ij(p^i+1)}y^{p^e} + \ldots)y \\ & + \ldots + (c_{ij(p-1)} + c_{ij(2p^i-1)}y^{p^e} + \ldots)y^{p^e-1} \\ = \quad & d_{ij0} + d_{ij1}y + \ldots + d_{ij(p^e-1)}y^{p^e-1} \end{aligned}$$

where $d_{ijs} \in \mathbb{F}_q[y^{p^e}]$. We denote y^{p^e} by z, so $d_{ijs} \in \mathbb{F}_q[z]$ for $1 \leq i \leq k$, $1 \leq j \leq n$ and $0 \leq s \leq p^e - 1$. Similarly we denote $c_i^{p^e}$ by $\bar{c}_i \in \mathbb{F}_q[z]$. So the linear system we need to solve is

$$\textstyle\sum_{s=1}^{k} \bar{c}_s c_{si} = \sum_{s=1}^{k} \bar{c}_s(d_{si0} + d_{si1}y + \ldots + d_{si(p^e-1)}y^{p^e-1}) = 0 \quad \text{for } 1 \leq i \leq n$$

Solving this linear system is equivalent to the following linear system in $\mathbb{F}_q[z]$.

$$\sum_{s=1}^{k} \bar{c}_s d_{sij} = 0 \text{ for every } 1 \leq i \leq n \text{ and } 0 \leq j \leq p^e - 1.$$

Solving this linear system we get $\bar{c}_s$. For every coefficient in $\bar{c}_s$, we compute its preimage under the Frobenius endomorphism $\Phi_e : \ x \mapsto x^{p^e}$ of $\mathbb{F}_q$.

To analyze the cost, we note that computing each b_ia_j will require $O^{\sim}(dm^\omega)$ operations and computing X^{p^e}(the degree of X is less than $2d$) will need $O^{\sim}(m^\omega dp^{2e})$, giving $O^{\sim}(nkp^{2e}m^\omega d)$ operations in $\mathbb{F}_q$ in total. Solving the $p^en \times k$ linear system in $\mathbb{F}_q[z]$ of degree $2d$ will cost $O^{\sim}(p^enk^{\omega-1}d)$. Thus, the total cost is $O^{\sim}(nkp^{2e}m^\omega d + p^ek^{\omega-1}nd) = O^{\sim}(nkp^{2e}m^\omega d)$ operations in $\mathbb{F}_q$. Thus, for every e we have the follows.

Theorem 39. *Given the integral basis* $\{b_1, b_2, \ldots b_k\}$ *of* I_{e-1} *with degree bound* d, *as above we can compute the basis of* I_e *taking* $O^{\sim}(nkp^{2e}m^\omega d)$ *operations in* $\mathbb{F}_q$.

Lemma 40. *Let A be an n-dimensional algebra over the field $\mathbb{F}_q(y)$. Assume that the structure constants have numerators of height at most Δ and a common denominator in $\mathbb{F}_q[y]$. Assume further more that for a $0 \leq i \leq \lfloor log_p n \rfloor$ and the ideal I_{i-1} has an integral basis of height at most Γ. Then the ideal I_i has an integral basis of height at most $n(\Gamma + 2\Delta)$.*

Lemma 40 [24] is easy to prove by analyzing the degree swelling of the procedure above for computing I_i. But the result is unacceptable since for our case the degree at the last step could be $O(n^{\log_p m})$, which is not polynomial size. To avoid this problem, Ivanyos et al. [24] find a degree upper bound for the bases of $\{I_1, I_2, \ldots\}$. Every time we get a basis of I_i whose degree is greater than the upper bound, we can modify its basis into a good form and then keep going. However, the degree bound here is not good enough, which make the complexity too high. The following proposition by Ivanyos ([24], Proposition 3.5) concerns the degree bound.

Proposition 41. *Let A be an n-dimension algebra over the field $\mathbb{F}_q(y)$. Assume that the structure constants are integral and their heights are limited by Δ. Then any ideal of A containing Rad(A) has an integral basis of height $O(n^3\Delta)$.*

Since we are dealing with the matrix algebra, the structure constants are completely fixed by its basis. Thus we do not want to make the structure constants as part of input in our problem. Actually, given a general integral basis of degree Δ, the degree of the structure constants could be as large as $O(n\Delta)$, so the degree bound can reach $O(n^4\Delta)$. Now we can give a polynomial-time algorithm.

Algorithm 3.6 Computation of the radical

```
Input: An basis $\{a_1, a_2, \ldots, a_n\}$ of the $m \times m$ matrix algebra $A$ over $\mathbb{F}_q(y)$, where
$\deg(a_i) \leq \Delta$;
Output: The basis $\{r_1, r_2, \ldots, r_t\}$ of the radical of $A$;

1: $I_0 = A$; $t = \lfloor \log_p m \rfloor$;
2: for $i = 1$ to $t$ do
3:     Compute the basis of $I_i = \{x \in I_{i-1} \mid \phi_i(xy) = 0, \forall y \in A\}$;
4:     Find another basis for $I_i$ such that its degree is at most $O(n^4\Delta)$;
5: end for
6: return the basis of $I_t$;
```

The analysis of its complexity is as follows. For the step 3, by Theorem 39, with d being $O(n^4\Delta)$ and k being n the cost will be $O(n^2 p^{2e} m^\omega n^4 \Delta) = O(p^{2e} m^\omega n^6 \Delta)$. So the total cost is $O^{\sim}(\sum_{i=0}^{\lceil \log_p n \rceil} p^{2i} m^\omega n^6 \Delta) = O^{\sim}(m^{\omega+2} n^6 \Delta)$. Note that the basis generated by step 3 is of degree at most $O(n \times n^4\Delta) = O(n^5\Delta)$. Step 4 could be done at the same time when we do step 3 using Storjohann's algorithm for computing the null space. So the total cost for this algorithm is $O^{\sim}(n^6 m^{\omega+2}\Delta)$ operations in $\mathbb{F}_q$.

Note that for a general basis the structure constants are not neccessarily integral as well. Ivanyos et al. present a way to modify the basis such that the construct constants are integral, but it will make the degree of the new basis $O(n\Delta)$ and that of the structure constants $O(n\Delta)$.

3.2.2 New Algorithm of Ivanyos

There is another algorithm by Ivanyos [22] for finding the radical of matrix algebras using Fitting decompositions. The idea is to reduce the problem to the commutative case based on the torus defined as follows. Assume K is a field.
Definition 42. A K-algebra is called *torus* if it is a finite dimensional commutative K-algebra which is separable over K.

The torus and the maximal torus have many good properties which we summary in the following proposition [22].

Proposition 43. *Given a matrix K-algebra A and T a subalgebra of A.*

1. *Let K' be the algebraic closure of K. Then T is a torus if and only if the matrices in T can be simultaneously diagonalized over K';*
2. *If A is commutative then A contains a unique maximal torus. Furthermore, a maximal torus of A contains the maximal torus of the center of A, $Z(A)$;*
3. *If T is a torus, then T is the maximal torus of A if and only if T is the maximal torus of its centralizer $C_A(T)$;*
4. *Let $\phi:\ A \to A/Rad(A)$ be the natural projection and T is a torus, then T is the maximal torus of A if and only if $\phi(T)$ is the maximal torus of $A/Rad(A)$;*
5. *If A is a direct sum of ideals $A_1, A_2, \ldots, A_r$ and T is a torus, then T is the maximal torus of A if and only if $T \cap A_i$ is the maximal torus of A_i for $i = 1, 2, \ldots, r$;*
6. *If Z is a subfield of $Z(A)$ and T is a torus, then T is the maximal torus of A if and only if TZ, considered as a Z-algebra, is a maximal Z-torus of A and Z is a purely inseparable extension of $Z \cap T$;*
7. *If T is the maximal torus of A, then $C_A(T)/Rad(C_A(T))$ is commutative;*
8. *If T is a maximal torus of A and K' is an arbitrary field extension of K, then $K' \otimes_K T$ is a maximal K'-torus of $K' \otimes_K A$.*

The new algorithm of Ivanyos is theoretically based on the structure theory of the algebra with respect to the torus. Given a matrix K-algebra A and $\phi:\ A \to A/\mathrm{Rad}(A)$ the natural projection. Let $\widetilde{C}$ be the set of central separable elements of $A/\mathrm{Rad}(A)$, T a fixed maximal torus of A and $C = \{x \in T \mid \phi(x) \in Z(A/\mathrm{Rad}(A))\}$. Then C is the subalgebra of T and $\phi(C) = \widetilde{C}$. Besides, $A = C_A(C) + [C, A]$, where $[C, A] = \{xy - yx \mid x \in C,\ y \in A\}$. Denote $C_A(C)$, $[C, A]$ and $C_A(T)$ by S, N and H respectively, then we have the following structure theorem [22].

Theorem 44.

1. *$SN \subset N$, $NS \subset N$ and $N \subset$ Rad(A);*
2. *Rad(A) $=$ Rad(S) $+ N$;*
3. *Rad(S) $= S$Rad(H)S and every nilpotent element of H is in Rad(H).*

The paper of Graaf and Ivanyos [20] discusses about computing the maximal torus T in polynomial time over K. When $K = \mathbb{F}_q(y)$ it may suffer degree explosion again. To simplify our discussion, we will not accumulate the cost of computing the maximal torus T and assume the degree bound for the basis of T is the same as that of the basis of A, i.e. Δ. From Theorem 44 if we can compute C then we can compute S, N and H, and hence $\text{Rad}(A)$. However, the definition of C depends on $\text{Rad}(A)$. Another method to describe C given by Ivanyos is that $C = \{x \in T \mid xL \subset L\}$, where $L = [A, A] \cap T$. So we have a computable way to find $\text{Rad}(A)$, presented in the following algorithm.

Algorithm 3.7 Computation of the radical

```
Input: An integral basis, $\{a_1, a_2, \ldots, a_n\}$, of a matrix algebra $A \subset \mathbb{F}_q(y)^{m\times m}$, where $\{a_1, a_2, \ldots, a_k\}$ is the basis of its maximal torus $T$, $\deg(a_i) \le \Delta$ for $1 \le i \le n$;
Output: The basis of $\text{Rad}(A)$, $\{r_1, r_2, \ldots, r_r\}$;

Compute the basis of the centralizer of $T$, $H = C_A(T)$;
Select the basis of $[A, A]$ from $\{[a_i, a_j]\}$ for $i, j = 1, 2, \ldots, n$;
Compute the basis of $L = T \cap [A, A]$;
Compute the basis, $\{c_1, c_2, \ldots c_s\}$, of $C = \{x \in T \mid xL \subset L\}$;
Select a basis of $N = [C, A]$ from $\{[a_i, c_j]\}$ for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, s$;
Select a basis of $I = H[H, H]H$;
Compute a basis, $\{h_1, h_2, \ldots, h_r\}$, of $H_1 = H/I$;
Compute the radical of $H_1$;
Compute the basis of $\text{Rad}(H)$ generated by the bases of $\text{Rad}(H_1)$ and $I$;
Compute the basis of the centralizer of $C$, $S = C_A(C)$;
Select the basis of $S\text{Rad}(H)S$;
Return the union of the basis of $N$ and $S\text{Rad}(H)S$.
```

By Proposition 43, $H/\text{Rad}(H)$ is commutative. So $I \in \text{Rad}(H)$ and H_1 is commutative. Finding the radical of a commutative algebra is much easier and is equivalent to finding the nilpotent elements. We will discuss the details of Algorithm 3.7 in the rest of this subsection. Note we will treat $k = O(n)$.

Let $\mathrm{Vec}(a)$ denote the vector generated by the entries of a in this way: $\mathrm{Vec}(a) = (a_{11}, a_{12}, \dots, a_{1m}, a_{21}, a_{22}, \dots, a_{2m}, a_{31}, \dots, a_{m1}, a_{m2}, \dots, a_{mm})^T$. We will first compute $\gamma_{ij} = a_i a_j - a_j a_i$ for $i, j = 1, 2, \dots, n$, which will take $O^\sim(n^2 m^\omega \Delta)$. For step 1, we do not need to solve the basis over the whole algebra A. Since we already know that T is commutative, we can compute the centralizer of T over the basis $\{a_{k+1}, a_{k+2}, \dots, a_n\}$. So it will just solve the linear system of $\sum_{j=k+1}^{n} \gamma_{ij} x_j = 0$ for $i = 1, 2, \dots, k$ which cost $O^\sim(km^2(n-k)^{\omega-1}\Delta) = O^\sim(n^\omega m^2 \Delta)$ with degree bound $O(n\Delta)$ for the basis of H. Step 2 will take $O^\sim(m^2 n^2 \Delta + n^2 m^4)$ by Theorem 7. Suppose the basis of $[A, A]$ is $\{\bar{a}_1, \dots, \bar{a}_v\}$. For step 3, we first compute the basis, $\{n_1, n_2, \dots, n_{m^2-v}\} \subset \mathbb{F}_q[y]^{m^2}$, of the linear null space of $\{\mathrm{Vec}(\bar{a}_1), \dots, \mathrm{Vec}(\bar{a}_v)\}$, then we solve the linear system

$$\sum_{i=1}^{k} x_i \mathrm{Vec}(a_i)^T n_j = 0,\ j = 1, 2, \dots, m^2 - v$$

which requires

$$\begin{aligned} & O^\sim(m^2 v^{\omega-1}\Delta + k(m^2 - v)m^2 v\Delta + (m^2 - v)k^{\omega-1} v\Delta) \\ = \;& O^\sim(m^2 n^{\omega-1}\Delta + km^4 n\Delta + nm^2 k^{\omega-1}\Delta) \\ = \;& O^\sim(n^2 m^4 \Delta) \end{aligned}$$

operations over $\mathbb{F}_q$ in total. Assume the basis of L is $\{l_1, l_2, \dots, l_u\}$, then $\deg(l_i) = O(n^2\Delta)$ and $u \le v$. For step 4, we first compute the basis of the nullspace of the linear space spanned by $\{\mathrm{Vec}(l_1), \dots, \mathrm{Vec}(l_u)\}$, $\{n'_1, n'_2, \dots, n'_{m^2-u}\}$ with $\deg(n'_i) = O(n^3\Delta)$. Then we need to solve the linear system

$$\sum_{i=1}^{k} x_i \mathrm{Vec}(a_i l_j)^T \mathrm{Vec}(n'_k) = 0,\ j = 1, 2, \dots, u;\ k = 1, 2, \dots, m^2 - u,$$

for the basis $\{c_1, \dots, c_s\}$ of C, whose complexity is

$$\begin{aligned} & O^\sim(u^{\omega-1} m^2 n^2 \Delta + ukm^\omega n^2 \Delta + (m^2 - u)ukm^2 n^3 \Delta + (m^2 - u)uk^{\omega-1} n^3 \Delta) \\ = \;& O^\sim(u^{\omega-1} m^2 n^2 \Delta + ukm^\omega n^2 \Delta + ukm^4 n^3 \Delta + m^2 uk^{\omega-1} n^3 \Delta) \\ = \;& O^\sim(n^5 m^4 \Delta) \end{aligned}$$

with $\deg(c_i) = O(n^4\Delta)$. Similarly, step 5 requires $O^\sim(snm^\omega n^4\Delta + snm^2 n^4\Delta + snm^4) = O^\sim(n^6 m^\omega \Delta)$ operations over $\mathbb{F}_q$. In step 6, when computing the generating set of $H[H, H]H$ we will compute the basis of $[H, H]$, $H[H, H]$ and $H[H, H]H$ successively with complexity $O^\sim(n^2 m^\omega n\Delta + m^2 n^2 n\Delta + n^2 m^4) = O^\sim(n^3 m^\omega \Delta + n^2 m^4)$. We have to compute the projected image in step 7. Suppose the bases of H and I are $\{\bar{h}_1, \bar{h}_2, \dots, \bar{h}_u\}$ and $\{\bar{i}_1, \bar{i}_2, \dots, \bar{i}_v\}$. We will first compute the basis, $\{\hat{n}_1, \hat{n}_2, \dots, \hat{n}_{m^2-v}\}$ of the linear null space of the linear space spanned by $\{\mathrm{Vec}(\bar{i}_1), \dots, \mathrm{Vec}(\bar{i}_v)\}$ and then check the linear combination of $\bar{h}_i$ which is orthogonal to it by solving

$$\sum_{i=1}^{u} x_i \mathrm{Vec}(\bar{h}_i)^T \mathrm{Vec}(\hat{n}_j) = 0,\ j = 1, 2, \ldots, m^2 - v,$$

with $\deg(\hat{n}_j) = O(n^2\Delta)$. Its complexity is

$$\begin{aligned} & O^\sim(m^2 v^{\omega-1} n\Delta) + O^\sim((m^2 - v)um^2n^2\Delta + u^{\omega-1}(m^2 - v)n^2\Delta) \\ = \ & O^\sim(m^2v^{\omega-1}n\Delta + um^4n^2\Delta + u^{\omega-1}m^2n^2\Delta) \\ = \ & O^\sim(m^4n^3\Delta) \end{aligned}$$

with $\deg(h_i) = O(n^3\Delta)$. We will analyze the cost of step 8 separately. So the total cost of steps 1-7 is $O^\sim(n^5m^4\Delta + n^6m^\omega\Delta)$, where we treat $k = O(n)$.

Let $j = \lceil \log_p m \rceil$, so x is nilpotent if and only if $x^{p^j} = 0$ in H_1. Let $x = \sum_{i=1}^{r} x_i h_i$, then first compute $t_i = h_i^{p^j}$, with $\deg(t_i) = O(p^j n^3 \Delta)$. Then we compute $d_{ik} = \mathrm{Vec}(t_i)^T \mathrm{Vec}(\hat{n}_k)$ for $1 \le i \le r$ and $1 \le k \le m^2 - v$. Note that $\deg(d_{ik}) = O(n^3\Delta p^j)$. For each d_{ik}, we rewrite it into the following form.

$$\begin{aligned} d_{ik} = \ & (d_{ik0} + d_{ikp^i} y^{p^j} + \ldots) + (d_{ik1} + d_{ik(p^j+1)} y^{p^j} + \ldots)y \\ & + \ldots + (d_{ik(p^j-1)} + d_{ik(2p^j-1)} y^{p^j} + \ldots) y^{p^j-1} \\ = \ & e_{ik0} + e_{ik1}y + \ldots + e_{ik(p^j-1)} y^{p^j-1}, \end{aligned}$$

where $e_{iku} \in \mathbb{F}_q[y^{p^j}]^{m\times m}$ for $1 \le u \le p^j - 1$. Again we denote y^{p^j} by z. So $e_{iku} \in \mathbb{F}_q[z]$ and $\deg_z(e_{iku}) = O(n^3\Delta)$. Also we denote $x_i^{p^j}$ by $\bar{x}_i \in \mathbb{F}_q[z]$. So the linear system we need to solve in $\mathbb{F}_q[z]$ is

$$\sum_{i=1}^{r} \bar{x}_i d_{ik} = \sum_{i=1}^{r} \bar{x}_i (e_{ik0} + e_{ik1}y + \ldots + e_{ik(p^j-1)} y^{p^j-1}) = 0,\ 1 \le k \le m^2 - v, \quad (3.2.1)$$

where the $\bar{x}_i$s are unknowns in $\mathbb{F}_q[z]$. Thus, the linear system 3.2.1 equals to the following linear system in $\mathbb{F}_q[z]$.

$$\sum_{i=1}^{r} \bar{x}_i e_{iks} = 0,\ 0 \le s \le p^j - 1 \text{ and } 1 \le k \le m^2 - v.$$

Solving this linear system we get $\bar{x}_v$. Then for every coefficient in $\bar{x}_v$, we compute its preimage under $\mathbb{F}_q$'s Frobenius endomorphism $\Phi_j : x \mapsto x^{p^j}$. For its cost, the system has r unknowns and $(m^2 - v)p^j$ equations. Thus its complexity is $O^\sim((m^2 - v)p^j r^{\omega-1} n^3\Delta) = O^\sim(m^3 n^{\omega+2}\Delta)$ and $\deg(x_i) = O(n^4\Delta)$.

Step 9 will do nothing but just take the union of $\{\bar{i}_1, \bar{i}_2, \ldots, \bar{i}_v\}$ and the basis of $\mathrm{Rad}(H_1)$ to be the basis of $\mathrm{Rad}(H)$, denoted by $\{h_1^{(r)}, h_2^{(r)}, \ldots, h_u^{(r)}\}$. Step 10 again requires solving a linear system $\sum_{i=1}^{n} x_i(a_ic_j - c_ja_i) = 0$ for $1 \le j \le s$, which will take $O^\sim(nsm^\omega n^4\Delta + sm^2n^{\omega-1}n^4\Delta) = O^\sim(n^{\omega+4}m^2\Delta)$ with $\deg(x_i) = O(n^5\Delta)$. Suppose the basis of S is $\{s_1, s_2, \ldots, s_w\}$. The basis of $\mathrm{Rad}(S)$ is selected in the step 11. We will first compute $s_i h_t^{(r)} s_j$ for $1 \le i, j \le w$ and $1 \le t \le u$ where s_i is the basis of S, which requires $O^{\sim}(w^2 u m^{\omega} n^5 \Delta) = O^{\sim}(n^8 m^{\omega} \Delta)$ operations in $\mathbb{F}_q$. Then by Theorem 7, it will cost $O^{\sim}(w^2 u m^2 n^5 \Delta + n^2 u m^4)$. So the complexity of Algorithm 3.7 is $O^{\sim}(n^8 m^{\omega} \Delta)$ operations in $\mathbb{F}_q$.

Note we already assume that the first part of the basis of A is that of its fixed maximal torus, which is rarely to be satisfied in practice. So in general we need to compute a basis of the torus. Graaf and Ivanyos give an algorithm to compute the maximal torus in another paper [20], which is in polynomial time but again we need to be careful about the degree explosion. So in general, the new algorithm by Ivanyos of computation of the radical requires more than $O^{\sim}(n^8 m^{\omega} \Delta)$ operations in $\mathbb{F}_q$.

3.2.3 Raw Decomposition

We would like to introduce a new decomposition, which we call raw decomposition and has not been discussed in the previous papers. It will act as the first stage of our algorithm for computing the radical. Given a matrix algebra A over $\mathbb{F}_q(y)$, we want to compute its decomposition such that

$$A = P_1 + P_2 + \ldots + P_k + N,$$

where N is a linear subspace of its radical and each P_i is a primary subalgebra, i.e. $P_i/\mathrm{Rad}(P_i)$ is simple.

Let $\phi : A \to A/\mathrm{Rad}(A)$ the natural projection. By the Wedderburn's Structure Theorem [34] we have

$$A/\mathrm{Rad}(A) \cong (A_1 \oplus A_2 \oplus \ldots \oplus A_k).$$

Let $\bar{e}_i$ be the preimage of $\delta_i(1) \oplus \ldots \oplus \delta_i(k) \in A_1 \oplus A_2 \oplus \ldots \oplus A_k$ for $1 \le i \le k$, where $\delta_i(i) = 1$ and $\delta_i(t) = 0$ for $t \ne i$. Thus, $\sum_{i=1}^{k} \bar{e}_i = 1 \in A/\mathrm{Rad}(A)$. Choose proper $\{e_1, \ldots, e_k\}$ from $\phi^{-1}(\bar{e}_i)$s such that $\sum_{i=1}^{k} e_i = 1$. So

$$A = (\sum_{i=1}^{k} e_i) A (\sum_{i=1}^{k} e_i) = e_1 A e_1 + \ldots + e_k A e_k + (\sum_{i \ne j; 1 \le i,j \le k} e_i A e_j).$$

Let ϕ_{ij} be the induced projection of ϕ over $e_i A e_j$, $\phi_{ij}(e_i A e_j) = \bar{e}_i (A/\mathrm{Rad}(A)) \bar{e}_j$, which is equal to A_i if $i = j$ or 0 if $i \ne j$. Denote $e_i A e_i$ by P_i and $\sum_{i \ne j; 1 \le i,j \le k} e_i A e_j$ by N, then the raw decomposition exists.

For any $\alpha \in A$, let $s = \phi(\alpha)$. We denote the minimal polynomial of α by

$f = x^t + a_1x^{t-1} + \ldots + a_t$ and that of s by $g = x^s + b_1x^{s-1} + \ldots + b_s$. Then

$$\begin{aligned} f(\alpha) = & \; f(s+r) \\ = & \; (s+r)^t + a_1(s+r)^{t-1} + \ldots + a_t \\ = & \; s^t + a_1s^{t-1} + \ldots + a_t + r(ts^{t-1} + \ldots) \\ = & \; f(s) + r\delta = 0. \end{aligned}$$

So $f(s) \in \mathrm{Rad}(A)$, thereby $g|f$ and

$$\begin{aligned} g(\alpha) = & \; g(s+r) \\ = & \; g(s) + r\delta' \\ = & \; 0 + r\delta'. \end{aligned}$$

Since $r\delta' \in \mathrm{Rad}(A)$, then there exists a k such that $(r\delta')^k = 0$. So $g^k(\alpha) = 0$, i.e. $f|g^k$. Note that the irreducible components of f and g are totally the same since $g \mid f \mid g^k$, so we can use the algorithm for decomposition of semisimple algebra to compute the idempotent elements for a general algebra.

We compute the idempotent elements using Algorithm 3.4 without any modification and its projected image is also an idempotent in $A/\mathrm{Rad}(A)$ as well. Now assume $f = f_1f_2\ldots f_t$ such that each f_i is the power of some irreducible polynomial and they are pairwise co-prime. Similarly, denote $h_i = 1 - s_if_i$ where $s_if_i + t_i\frac{f}{f_i} = 1$. Then $\{h_1(\alpha), \ldots, h_t(\alpha)\}$ are pairwise orthogonal idempotents with sum I_m. It is easy to check that $\{\phi(h_1(\alpha)), \ldots, \phi(h_t(\alpha))\}$ are pairwise orthogonal idempotents with sum I_m in $A/\mathrm{Rad}(A)$. Denote $A/\mathrm{Rad}(A)$ by S, $h_i(\alpha)$ by ω_i and $\phi(h_i(\alpha))$ by $\bar{\omega}_i$.

In the approach here, however, it is more complicated to provide the condition that the idempotents split S in a good way: each $\bar{\omega}_i$ is in unique simple component of S. The difficulty comes from the computation of the equivalence classes of these idempotents.

Definition 45. Two idempotents ω_1 and ω_2 are called *equivalent*, if $\phi(\omega_1)$ and $\phi(\omega_2)$ are in a same simple component of S.

Note that even if ω_i and ω_j are in nonequivalent classes, for any $\alpha \in A$, $\omega_i\alpha\omega_j$ is not necessarily equal to zero. It could just be strongly nilpotent as stated in Lemma 48. Similar to Definition 30, we have the following definition to describe decomposing element in a general algebra.

Definition 46. Given an algebra A, an element $a \in A$ is called decomposing element for A if the minimal polynomials of $\phi(\alpha)$'s simple components in $A/\mathrm{Rad}(A)$ are pairwise co-prime.

It is also with large probability to get such decomposing element by selecting an element from A randomly. As what we did in Theorem 31, an element α, such that the degree of the minimal polynomial of $\phi(\alpha)$ is maximal in those of the elements in S, is a decomposing element, thereby each $\bar{\omega}_i$ is in unique simple component of S.

Theorem 47. *Let $A \subset \mathbb{F}_q(y)^{m\times m}$ be an matrix algebra of dimension n, with integral basis $a_1, a_2, ..., a_n \in \mathbb{F}_q(y)^{m\times m}$. If the elements $s_1, s_2, ..., s_n$ are randomly chosen uniformly and independently from $\mathbb{F}_q$, where $q \geq \frac{n^3}{\epsilon}$, then the element*

$$s_1a_1 + s_2a_2 + ... + s_na_n$$

is a decomposing element with probability at least $1-\epsilon$.

Proof. Let $F = \mathbb{F}_q(y)$. We can choose a new basis of A, $\{\beta_1, \beta_2, \ldots, \beta_n\}$, such that $\{\phi(\beta_1), \phi(\beta_2), \ldots, \phi(\beta_k)\}$ is the basis of S and $\{\beta_{k+1}, \beta_{k+2}, \ldots, \beta_n\}$ is the basis of Rad(A). By the Representation Theorem 22, S is isomorphic to a subalgebra of $M_k(F)$. Let $\hat{s}' = \hat{s}_1\phi(\beta_1) + \hat{s}_2\phi(\beta_2) + \ldots + \hat{s}_k\phi(\beta_k)$. Then by Theorem 34, if $\{\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_k\}$ is chosen from a subset of F with size $\frac{k^3}{\epsilon}$, then $\hat{s}'$ is a decomposing element in S with probability at least $1-\epsilon$.

Suppose elements $\hat{s}_1, \hat{s}_2, ..., \hat{s}_n \in \mathbb{F}_q(y)$ are the linear coordinates and the element $\hat{s} = \hat{s}_1\beta_1 + \hat{s}_2\beta_2 + ... + \hat{s}_n\beta_n$. Thus, if $\hat{s}_1, \hat{s}_2, ..., \hat{s}_n$ are selected randomly from a set of size at least $\frac{k^3}{\epsilon}$, then the element $\hat{s}'$ is a decomposing element with probability at least $1-\epsilon$, so is the $\hat{s}$.

Now it remains to show that our way of selecting random element for the basis $\{a_1, a_2, ..., a_n\}$ is equivalent to selecting $\hat{s}_i$ from a subset of $\mathbb{F}_q(y)$ of size at least q for $1 \leq i \leq n$. Assume the transformation matrix from $\{a_1, a_2, \ldots, a_n\}$ to $\{\beta_1, \beta_2, \ldots, \beta_n\}$ is U . So

$$\begin{pmatrix} c_1 & c_2 & \cdots & c_n \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} c_1 & c_2 & \cdots & c_n \end{pmatrix} U \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{pmatrix},$$

which indicates that selecting $\begin{pmatrix} c_1 & c_2 & \cdots & c_n \end{pmatrix}$ to be the coordinates of $\{a_1, a_2, \ldots, a_n\}$ is equivalent to selecting $\begin{pmatrix} c_1 & c_2 & \cdots & c_n \end{pmatrix}U$ for $\{\beta_1, \beta_2, \ldots, \beta_n\}$. Since U is invertible, the size of the set $\{\begin{pmatrix} c_1 & c_2 & \cdots & c_n \end{pmatrix}\}$ is the same as the size of the set $\{\begin{pmatrix} \hat{s}_1 & \hat{s}_2 & \cdots & \hat{s}_n \end{pmatrix} = \begin{pmatrix} c_1 & c_2 & \cdots & c_n \end{pmatrix}U\}$. So the size of the set that we select the coordinates for $\{\beta_1, \beta_2, \ldots, \beta_n\}$ is at least $(\frac{n^3}{\epsilon})^n$. Therefore we have at least probability $1-\epsilon$ to get a decomposing element since that $k \leq n$.

□

To determine the equivalence classes of the idempotents we encounter new problem, compared to semisimple case. Even when ω_i and ω_j are not in the same equivalence class, $\omega_i\alpha\omega_j$ is not guaranteed to be zero. We therefore need to modify our criterion, based on the following Lemma.

Lemma 48. *Given a set of idempotents $\{\omega_1, \ldots, \omega_t\}$ such that each $\phi(\omega_i)$ in only one simple component of $\phi(A)$ for $1 \leq i \leq t$, then for any $\alpha \in A$*

- *if ω_i and ω_j are in different simple components then $\omega_i\alpha\omega_j$ is strongly nilpotent.*

- *if ω_i and ω_j are in a same simple component, then $\omega_i \alpha \omega_j$ is strongly nilpotent if and only if $\phi(\omega_i)\phi(\alpha)\phi(\omega_j) = 0$.*

Proof. For any $\alpha \in A$, if $\phi(\omega_i)$ and $\phi(\omega_j)$ are in different simple components, then $\phi(\omega_i)\phi(\alpha)\phi(\omega_j) = 0$, thereby $\omega_i\alpha\omega_j \in \mathrm{Rad}(A)$. So $\omega_i\alpha\omega_j$ is strongly nilpotent. If ω_i and ω_j are in the same simple component, without loss of generality we assume it is the first component, then $\phi(\omega_i\alpha\omega_j) = \bar{s}$. It is obvious that when $\bar{s} = 0$, $\omega_i\alpha\omega_j$ is strongly nilpotent. Note that when $\bar{s} \neq 0$, $\omega_i\alpha\omega_j \notin \mathrm{Rad}(A)$, therefore $\omega_i\alpha\omega_j$ is not strongly nilpotent.

□

Lemma 48 suggests a way of determining the equivalent classes: randomly select an element from the algebra and check if it is strongly nilpotent. In order to check this we need the result of Ivanyos [24]. Let $g(x) \in \mathbb{F}_p[x]$ be the minimal polynomial of a generating element of $\mathbb{F}_q$ over $\mathbb{F}_p$. Define the local ring $R = \mathbb{Z}[x][y]_{(p)}/(g(x))$ then $R/pR = \mathbb{F}_q(y)$. Then we have the following proposition describing the nilpotent elements in $M_{n\times n}(R)$.

Proposition 49. *$l = \lfloor log_p n \rfloor$.*

1. *If $X \in M_{n\times n}(R)$ satisfies $Tr(Y^{p^l}) \equiv 0$ mod p^{l+1} for $Y \in \{X, X^2, \ldots, X^n\}$. Then the image of the matrix X is nilpotent in the residue class ring $M_{n\times n}(R/pR) \cong M_{n\times n}(R)/pM_{n\times n}(R)$;*

2. *If the image of the matrix X is nilpotent in the residue class ring $M_{n\times n}(R/pR) \cong M_{n\times n}(R)/pM_{n\times n}(R)$, then $Tr(X^{p^j}) \equiv 0$ mod p^{j+1} for all the nonnegative integers j;*

Given a basis $\{a_1, a_2 \ldots, a_n\}$ for A, if $X \in A$ satisfies that $\mathrm{Tr}((Xa_i)^{p^l}) \equiv 0 \bmod p^{l+1}$, then for any $Y \in A$, $\mathrm{Tr}((C)^{p^l}) \equiv 0 \bmod p^{l+1}$ for every $C \in \{XY, (XY)^2, \ldots, (XY)^n\}$. Since we can express C over the basis of $\{a_1, a_2 \ldots, a_n\}$, we have the following proposition.

Proposition 50. *Given the basis $\{a_1, a_2 \ldots, a_n\}$ of a finite-dimensional matrix algebra $A \subset \mathbb{F}_q(y)^{m\times m}$, an element $X \in A$ is strongly nilpotent in A if and only if $Tr((Xa_i)^{p^l}) \equiv 0$ mod p^{l+1} for $1 \leq i \leq n$ and $l = \lfloor log_p m \rfloor$.*

For any element $\beta \in A$, β is nilpotent if and only if $Tr(\beta^{p^l}) \equiv 0 \bmod p^{l+1}$, where $l = \lfloor \log_p m \rfloor$. We want to check if, for any β in the basis of A, $Tr((\omega_i\alpha\omega_j\beta)^{p^l}) \equiv 0 \bmod p^{l+1}$.

Theorem 51. *Given an integral basis $\{a_1, a_2, \ldots, a_n\}$ of the matrix algebra $A \subset \mathbb{F}_q(y)^{m\times m}$, and integral elements $\{\omega_1, \omega_2, \ldots, \omega_t\}$ with $t \leq q\epsilon$ such that each ω_i is in a unique simple component of A, where $deg(a_i) \leq \Delta$ and $deg(\omega_i) \leq d\Delta$, Algorithm 3.8 computes the equivalence classes of $\{\omega_1, \omega_2, \ldots, \omega_t\}$ correctly with probability at least $1 - \epsilon$, taking $O^{\sim}(t^2 nm^{\omega+1} d\Delta)$ operations in $\mathbb{F}_q$.*

Algorithm 3.8 Determine the equivalence classes

Input: An integral basis of the matrix algebra $A \subset \mathbb{F}_q(y)^{m\times m}$, $\{a_1, a_2, \dots, a_n\}$;
Integral elements $\{\omega_1, \omega_2, \dots, \omega_t\}$ with $t \leq q\epsilon$ such that each ω_i is in a unique simple component of A;
$\deg(a_i) \leq \Delta$ and $\deg(\omega_i) \leq d\Delta$;

Output: The sums of all the equivalence classes of $\{\omega_1, \omega_2, \dots, \omega_t\}$;

```
Choose random elements $c_i \in \mathbb{F}_q$ for $i = 1, 2, \dots, n$. Let $\alpha = \sum_{i=1}^{n} c_i a_i$.
Compute $t_{ij} = \omega_i \alpha \omega_j$, for $i, j = 1, 2, \dots, t$ and $i \lneq j$.
$l = \lfloor \log_p m \rfloor$
for $i = 1$ to $t$ do
    for $j = 1$ to $t$ do
        $\text{temp}_{ij} = 0$;
        for $k = 1$ to $n$ do
            if $\mathrm{Tr}((t_{ij} a_k)^{p^l}) \not\equiv 0 \bmod p^{l+1}$ then
                $\text{temp}_{ij} = 1$;
                Break out of the inner "for" loop of $k$;
            end if
        end for
    end for
end for
$I = \{1, 2, \dots, t\}$;
while $I \neq \emptyset$ do
    for $i \in I$ do
        compute $\bar{\omega}_i = \sum_{\substack{j = i, i+1, \dots, t \\ \text{temp}_{ij} = 1}} \omega_j$;
        $I = I \backslash \{j | \text{temp}_{ij} = 1,\ j = i, i+1, \dots, t\}$;
    end for
end while
return all the $\bar{\omega}_i$;
```

Proof. When ω_i and ω_j are in different simple components then $\omega_i \alpha \omega_j$ is strongly nilpotent. The algorithm works correctly under this case. When ω_i and ω_j are in a same simple component, then once $\omega_i \alpha \omega_j$ is not strongly nilpotent then the return result is correct. Otherwise, $\omega_i \alpha \omega_j$ is strongly nilpotent, i.e. $\phi(\omega_i s \omega_j) = 0$ by Lemma 48. We can choose a new basis of A, $\{\beta_1, \beta_2, \ldots, \beta_n\}$, such that $\{\phi(\beta_1), \phi(\beta_2), \ldots, \phi(\beta_k)\}$ is the basis of S and $\{\beta_{k+1}, \beta_{k+2}, \ldots, \beta_n\}$ is the basis of $\mathrm{Rad}(A)$. Similarly, there is at least one element in the basis of A such that $\phi(\omega_i \alpha \omega_j)$ is not zero. Without loss of generality, assume $\{\beta_1, \beta_2, \ldots, \beta_s\}$ are all the elements in the basis satisfying such condition. $s \le k$. Let $\alpha = \sum_{i=1}^{s} c_i' \beta_i$ and $\phi(\omega_i \alpha \omega_j) = 0$.

By the Representation Theorem 22, S is isomorphic to a subalgebra of $M_k(F)$ under some mapping ψ. Thus

$$\sum_{t=1}^{s} c_t' \bar{\beta}_t = 0,$$

where $\bar{\beta}_t = \psi(\phi(\omega_i \beta_t \omega_j))$ for $t = 1, 2, \ldots, s$. Selecting a random element of A in the 1st step of Algorithm 3.8 is equivalent to selecting $\{c_1', c_2', \ldots, c_s'\}$ for $\{\beta_1, \beta_2, \ldots, \beta_s\}$ from a subset of $\mathbb{F}_q(y)$ with size q. So there are at most q^{s-1} solutions for this equation and the size of the ground set is q^s. By Lemma 3 (Schwartz-Zippel Lemma) the probability that this equation holds, i.e. the algorithm works incorrectly, is at most $\frac{1}{q}$. So for the whole algorithm, the probability of correctness is $(1 - \frac{1}{q})^t \ge 1 - \frac{t}{q} \ge 1 - \epsilon$.

To analyze the complexity, step 2 will take t^2 times matrix multiplications, taking $O(t^2 \times m^\omega \times M(d\Delta)) = O^\sim(t^2 m^\omega d\Delta)$ operations in $\mathbb{F}_q$ and $\deg(t_{ij}) \le O(d\Delta)$. Checking if $\omega_i \alpha \omega_j$ is strongly nilpotent requires computation of the p^lth power of $t^2 n$ matrices. It will cost $O(t^2 n \times m^w \times M(d\Delta) \times p^l) \le O^\sim(t^2 n m^{\omega+1} d\Delta)$. So the total complexity of Algorithm 3.8 is $O^\sim(t^2 n m^{\omega+1} d\Delta)$ operations in $\mathbb{F}_q$.

□

Similarly to Algorithm 3.3, we can modify it to be a deterministic algorithm. Now we can give our algorithm of raw decomposition. From Theorem 47 it is very possible to get a decomposing element α. Suppose $\{\omega_1, \omega_2, \ldots, \omega_k\}$ is the idempotents computed from α, then

$$A = \sum_{i=1}^{k} e_i A e_i + \sum_{i \ne j;\ i,j=1}^{k} e_i A e_j;$$

Let $N = \sum_{i \ne j;\ i,j=1}^{k} e_i A e_j$, and $P_i = e_i A e_i$, then we get the raw decomposition. The algorithm is given as follows.

Theorem 52. *Given an integral basis $\{a_1, a_2, \ldots, a_n\}$ of $A \subset \mathbb{F}_q(y)^{m \times m}$, where $q \ge max\{\frac{5n^3}{\epsilon}, \frac{5m}{\epsilon}, \frac{5m^2(m-1)^2\Delta^2}{\epsilon}, 3m^2\Delta, 7m^2\Delta, \frac{5nm(2m^2+4m+1)\Delta}{\epsilon}, \frac{5m^2(2m^2+4m+1)\Delta}{\epsilon}\}$ and $deg(a_i) \le \Delta$, for $1 \le i \le n$, Algorithm 3.9 computes the raw decomposition correctly with probability at least $1 - \epsilon$, taking $O^\sim(m^{\omega+1}\Delta^{\frac{\omega+1}{2}} + n^2 m^{\omega+4}\Delta)$ operations in $\mathbb{F}_q$.*

Proof. From the analysis above, the algorithm returns the correct result when a decomposing element is selected and the equivalent classes are correctly determined.

Algorithm 3.9 Raw Decomposition

```
Input: An basis $\{a_1, a_2, \ldots, a_n\}$ of $A \subset \mathbb{F}_q(y)^{m\times m}$, where $\deg(a_i) \leq \Delta$, for $1 \leq i \leq n$ and $q \geq \max\{\frac{5n^3}{\epsilon}, \frac{5m}{\epsilon}, \frac{5m^2(m-1)^2\Delta^2}{\epsilon}, 3m^2\Delta, 7m^2\Delta, \frac{5nm(2m^2+4m+1)\Delta}{\epsilon}, \frac{5m^2(2m^2+4m+1)\Delta}{\epsilon}\}$;

Output: Bases of all the primary subalgebras $P_i$s of $A$ and the basis of $N$ which is the subalgebra of Rad($A$) such that $A = P_1 + P_2 + \ldots + P_k + N$.

1: Choose a random $(c_1, c_2, \ldots, c_n)$ from $\mathbb{F}_q^n$ and Compute $\alpha = \sum_{i=1}^{n} c_i a_i$;
2: Compute the set of idempotents $\{\omega'_1, \omega'_2, \ldots, \omega'_t\}$ using Algorithm 3.4;
3: Modify $\{\omega'_1, \omega'_2, \ldots, \omega'_t\}$ to be integral by deleting the common denominator of each $\omega'_i$ for $1 \leq i \leq t$, denoted by $\{\omega_1, \omega_2, \ldots, \omega_t\}$;
4: Determine the equivalent classes of $\{\omega_1, \omega_2, \ldots, \omega_t\}$ using Algorithm 3.8;
5: Compute $G_i = \{\overline{\omega}_i a_1 \overline{\omega}_i, \overline{\omega}_i a_2 \overline{\omega}_i, \ldots, \overline{\omega}_i a_n \overline{\omega}_i\}$ for $1 \leq i \leq k$ and $N = \{\overline{\omega}_i a_s \overline{\omega}_j\}$ for $i \neq j$, $1 \leq i, j \leq k$ and $1 \leq s \leq n$;
6: Return a maximal linearly independent subset of all the $G_i$s and $N$ using Algorithm 2.1.
```

By Theorem 47, step 1 selects a decomposing element with the large probability at least $1 - \frac{\epsilon}{5}$. Then by Theorem 35 and 51, step 3 and 4 compute and determine the equivalent classes correctly with probability greater than $1 - \frac{2\epsilon}{5}$. Since $k \leq m$, $\deg(\omega_i) \leq (m^2 + 2m)\Delta$ and $q \geq \frac{5m(2m^2+4m+1)\Delta \min\left(m^2, n\right)}{\epsilon}$, for each G_i, the probability of selecting the basis correctly is at least $1 - \frac{\epsilon}{5m}$ from Theorem 7. Thereby the probability of correctness of step 4 is $(1 - \frac{\epsilon}{5m})^m \geq 1 - \frac{\epsilon}{5}$. Similarly, $q \geq \frac{5(2m^2+4m+1)\Delta \min\left(m^2, m^2 n\right)}{\epsilon}$, the basis of N is computed correctly with probability $1 - \frac{\epsilon}{5}$. So the algorithm is correct with probability $(1 - \frac{\epsilon}{5})^3(1 - \frac{2\epsilon}{5}) \geq 1 - \epsilon$.

For the complexity, step 2 will cost $O^\sim(m^6\Delta + m^{\omega+1}\Delta^{\frac{\omega+1}{2}})$ by Theorem 35. Nothing happens in step 3. Note the degree of ω_i is $O(m^2\Delta)$. Using Algorithm 3.8 the cost of step 4 is $O^\sim(t^2 n m^{\omega+3}\Delta) = O^\sim(n m^{\omega+5}\Delta)$ by Theorem 51. And since the degree of ω_i is $O(m^2\Delta)$, so is the $\bar{\omega}_i$. So the degree of $\bar{\omega}_i a_s \bar{\omega}_j$ is $O(m^2\Delta)$. Step 5 will requires $O^\sim(k^2 n m^\omega m^2 \Delta) = O^\sim(n m^{\omega+4}\Delta)$ operations over $\mathbb{F}_q$. According to Theorem 7, step 6 will take $O^\sim(k \times (nm^2m^2\Delta + nm^2n) + (nk^2m^2m^2\Delta + nk^2m^2\min(nk^2, m^2))) = O^\sim(nm^6\Delta)$. So the complexity of this algorithm is $O^\sim(m^{\omega+1}\Delta^{\frac{\omega+1}{2}} + nm^{\omega+5}\Delta)$ operations in $\mathbb{F}_q$.

□

3.2.4 Computing the Radical

We end this section with a complete algorithm and its analysis for computing the radical of A. Given a primary matrix algebra $A \subset \mathbb{F}_q(y)^{m\times m}$, we follow the algorithm in the paper of Ivanyos et al.[24]. Theorem 38 suggests an algorithm (Algorithm 3.6) to compute the radical of a general matrix algebra. However, it suffers the explosion of the degree. Without any modification, the degree could grow in every iteration and reach $n^{\log_p n}$. One way of modifying this method is finding a good degree bound and transforming the resulting basis into a lower degree one in each iteration. Rónyai gives a degree bound of $O(n^3\Delta)$ based on the degree that the structure constants are integral with degree at most Δ. We already know that the degree bound here for our problem could reach as large as $O(n^4\Delta)$. The degree bound here is unacceptable. So we would like to find another way to avoid such degree explosion. Notice the following useful lemma in Rónyai's paper.

Theorem 53. *Given a primary algebra $A \subset \mathbb{F}_q[y]^{m\times m}$ over the field $\mathbb{F}_q(y)$. $I_1, \ldots, I_j$, where $j > \lfloor log_p n \rfloor$, are the same as that in Theorem 38. Then the case that $I_i \subsetneq I_{i-1}$ happens at most once in the subset chain.*

We will first modify the algorithm of Rónyai to compute the radical of a primary algebra while avoiding the degree explosion problem. Then a complete algorithm is given as a combination of doing raw decomposition first and then computes the radical of each primary component A_i. The basis of $\mathrm{Rad}(A)$ will be the union of the bases of the radicals of all the primary algebras and the basis of N. The idea of transforming the decomposition of a general algebra into the decomposition of primary algebra is inspired by Eberly's paper [7], where he transforms the computation of the radical of a general algebra over $\mathbb{F}_q$ into that of local algebras. Recall that for a general algebra A, $A = S + \mathrm{Rad}(A)$, where $\mathrm{Rad}(A)$ is the radical of A and S is a semisimple algebra. Since S is a semisimple algebra, $S = A_1 \oplus A_2 \oplus \ldots \oplus A_k$ where A_is are simple algebras. From the raw decomposition, Theorem 52, we can compute the bases of A_is and N with degree bound $O(m^2\Delta)$ for their bases in $O^\sim(m^{\omega+1}\Delta^{\frac{\omega+1}{2}} + nm^{\omega+5}\Delta)$ such that $A = (A_1 \oplus A_2 \oplus \ldots \oplus A_k) + N$ where N is the linear subspace of its radical and every A_i is primary (i.e. $A_i/\mathrm{Rad}(A_i)$ is simple).

Theorem 54. *Given a basis $\{a_1, a_2, \ldots, a_s\}$ of a primary algebra $A \subset \mathbb{F}_q(y)^{m\times m}$, where deg($a_i$) $= O(m^2\Delta)$, Algorithm 3.10 computes the radical of a primary algebra correctly taking $O^\sim(\min\{s,m\}^2 s^2 m^{\omega+2}\Delta)$ operations in $\mathbb{F}_q$.*

Proof. The correctness of this algorithm is proved in the paper of Ivanyos et al. [24] since the only difference is just that we are addressing a special input and the algorithm works for the general case. Note that from Theorem 53, once we detect the difference between I_{i-1} and I_i, then we can break out of the loop and I_i is the radical. This part is expressed in steps 4-9.

For the analysis of its complexity, it is similar to that in Subsection 3.2.1 except the degree explosion. If $\dim(I_i) = n$, then $I_i = A$, we can keep the basis for the next iterations. If $\dim(I_i) < n$, according to Theorem 53, I_i is the radical, we do not need to do the next iteration. So by Theorem 39, step 3 takes $O^\sim(skp^{2i}m^\omega d)$ for $d = m^2\Delta$, $k = s$ and $1 \le i \le t$. So the total cost will be $O^\sim(\sum_{i=1}^{t} skp^{2i}m^\omega d) = O^\sim(\min\{s,m\}^2 s^2 m^{\omega+2}\Delta)$ operations in $\mathbb{F}_q$.

□

Algorithm 3.10 Computation of Rad(A) of primary algebra

```
Input: A basis $\{a_1, a_2, \ldots, a_s\}$ of a primary algebra $A \subset \mathbb{F}_q(y)^{m \times m}$, where $\deg(a_i) = O(m^2\Delta)$;
Output: A basis $\{r_1, r_2, \ldots, r_t\}$ of the radical of $A$;

$I_0 = A$; $t = \min\{ \lfloor \log_p s \rfloor , \lfloor \log_p m \rfloor \}$;
for $i = 1$ to $t$ do
    Compute the basis of $I_i = \{x \in I_{i-1} \mid \mathrm{Tr}((xy)^{p^i}) \equiv 0 \bmod p^{i+1}, \forall y \in A\}$;

    if $\dim(I_i) \lneq n$ then
        $I_t = I_i$
        continue;
    else
        $I_i = A$;
    end if
end for
Return the basis of $I_t$.
```

Now we can present our complete algorithm to compute the radical of a general algebra. We will use the raw decomposition to separate the general algebra into primary ones and use Algorithm 3.10 to compute the radical of each primary subalgebra, then make a union to get a basis of its radical.

Algorithm 3.11 Computation of the radical

```
Input: An integral basis $\{a_1, a_2, \ldots, a_n\}$ of the $m \times m$ matrix algebra $A$ over $\mathbb{F}_q(y)$, where $q \geq \max\{\frac{5n^3}{\epsilon}, \frac{5m}{\epsilon}, \frac{5m^2(m-1)^2\Delta^2}{\epsilon}, 3m^2\Delta, 7m^2\Delta, \frac{5nm(2m^2+4m+1)\Delta}{\epsilon}, \frac{5m^2(2m^2+4m+1)\Delta}{\epsilon}\}$ and $\deg(a_i) \leq \Delta$ for $1 \leq i \leq n$;
Output: A basis $\{r_1, r_2, \ldots, r_t\}$ of its radical;

1: Compute the Raw Decomposition using Algorithm 3.9;
2: for each primary subalgebra $P_i$ for $1 \leq i \leq k$, compute its radical using Algorithm 3.10;
3: Return the basis of $N$ and all the Rad($P_i$) for $1 \leq i \leq k$.
```

Theorem 55. *Given an integral basis $\{a_1, a_2, \ldots, a_n\}$ of the $m \times m$ matrix algebra A over $\mathbb{F}_q(y)$, where $q \geq \max\{\frac{5n^3}{\epsilon}, \frac{5m}{\epsilon}, \frac{5m^2(m-1)^2\Delta^2}{\epsilon}, 3m^2\Delta, 7m^2\Delta, \frac{5nm(2m^2+4m+1)\Delta}{\epsilon}, \frac{5m^2(2m^2+4m+1)\Delta}{\epsilon}\}$ and $\deg(a_i) \leq \Delta$ for $1 \leq i \leq n$, Algorithm 3.11 computes the radical of a general matrix algebra correctly with probability at least $1 - \epsilon$, taking $O(m^{\omega+1}\Delta^{\frac{\omega+1}{2}} + n^2m^{\omega+4}\Delta)$ operations in $\mathbb{F}_q$.*

Proof. By Theorem 52, step 1 works correctly with probability $1 - \epsilon$. Step 2 is a deterministic procedure. So the whole algorithm returns the correct result with probability at least $1 - \epsilon$.

For the complexity, step 1 requires $O^{\sim}(m^{\omega+1}\Delta^{\frac{\omega+1}{2}}+n^2m^{\omega+4}\Delta)$ operations over $\mathbb{F}_q$ by Theorem 52 and gives the basis of each A_i with degree upper bound $O(m^2\Delta)$. Then step 2 needs $O^{\sim}(\sum_{i=1}^{k}\min\{s,m\}^2s^2m^{\omega+2}\Delta)$ operations in $\mathbb{F}_q$, where s_i is the dimension of A_i and $\sum_{i=1}^{k}s_i\leq n$. $O^{\sim}(\sum_{i=1}^{k}\min\{s,m\}^2s^2m^{\omega+2}\Delta)=O(n^2m^{\omega+4}\Delta)$. Thus the complexity will be $O^{\sim}(m^{\omega+1}\Delta^{\frac{\omega+1}{2}}+n^2m^{\omega+4}\Delta)$ operations in $\mathbb{F}_q$.

□

3.3 Modified algorithms for the case of small q

There are many restrictions on the size of q in our algorithms above, from choosing the decomposing element to selecting maximal linearly independent vector subset. We discuss how to modify the algorithms in this section to adapt them to the small finite field case.

The size of q is always related to the probability of failure in our algorithm. Now assume q is not large enough in the algorithm for selecting a maximal linearly independent subset of vectors (Algorithm 2.1). By the proof of Theorem 6, there are at most $k\Delta$ elements in the ground field $\mathbb{F}_q$ that change the linear dependency. If we extend $\mathbb{F}_q$ to $\mathbb{F}_{q^l}$, the number of such elements will still stay the same $k\Delta$. Thus, we can follow the same path of Algorithm 7 but modify the first step to be "Choose a random $\alpha\in\mathbb{F}_{q^l}$", then the probability of correctness of this algorithm is $1-\frac{k\Delta}{q^l}$. We can pick a sufficiently large l such that $q^l\geq\frac{\Delta\min(m,n)}{\epsilon}$ for a given $\epsilon>0$.

The second algorithm concerning the size of q is that of computing the minimal polynomial (Algorithm 2.4). Similarly by the proof of Theorem 18, the number of bad elements we choose from the ground field will be at most $m(m-1)^2\Delta$. Now assume we are computing the minimal polynomial of a over $\mathbb{F}_{q^l}$ for some positive $l\in\mathbb{Z}$ such that $q^l\geq\frac{1}{\epsilon}m(m-1)^2\Delta(m\Delta+1)$ for a given $\epsilon>0$. We follow Algorithm 2.4 but choose $\{\alpha_1,\alpha_2,\ldots,a_{m\Delta},\alpha_{m\Delta+1}\}\subset\mathbb{F}_{q^l}$ in the first step. It is easy to show that the probability of success is at least $1-\epsilon$. But we need to prove the result we get in this way is indeed in $\mathbb{F}_q[y,x]$, i.e. the result of the modified algorithm is correct, as the following lemma.

Lemma 56. *Given a matrix $a\in\mathbb{F}_q[y]^{m\times m}$, its minimal polynomial over $\mathbb{F}_q[y]$ is the same as that over $\mathbb{F}_{q^l}[y]$ when we treat a as a matrix in $F_{q^l}[y]$.*

Proof. Assume the minimal polynomial of a over $\mathbb{F}_q(y)$ is f and that over $\mathbb{F}_{q^l}(y)$ is

$$g=x^t+b_{t-1}x^{t-1}+\ldots+b_0,$$

where $b_i\in\mathbb{F}_{q^l}(y)$ for $0\leq i\leq t$. From Lemma 10 we know $g\in\mathbb{F}_{q^l}[y][x]$. Note that $g|f$, therefore $\deg(f)\geq t$. Assume $h(z)$ is the minimal polynomial with degree $l-1$ of a generating element of $\mathbb{F}_{q^l}$ over $\mathbb{F}_q$. So $b_i=b_{i0}+b_{i1}z+\ldots+b_{i(l-1)}z^{l-1}\in\mathbb{F}_q[y][z]$

where $b_{ij} \in \mathbb{F}_q[y]$ for $0 \leq i \leq t$ and $0 \leq j \leq l-1$. Since $a \in \mathbb{F}_q[y]$,

$$g(a) = (a^t + b_{(t-1)0}a^{t-1} + \ldots + b_{00}) + \ldots + (b_{(t-1)(l-1)}a^{t-1} + \ldots + b_{0(l-1)})z^{l-1} = 0.$$

So $b_{(t-1)j}a^{t-1} + \ldots + b_{0j} = 0$, i.e. $b_{(t-1)j}x^{t-1} + \ldots + b_{0j}$ is an annihilating polynomial of a or a zero polynomial for $1 \leq j \leq l-1$. If there is $u \in \{0, \ldots, t-1\}$ and $v \in \{1, \ldots, l-1\}$ such that $b_{uv} \neq 0$, then $b_{(t-1)v}x^{t-1} + \ldots + b_{0v}$ is an annihilating polynomial of a, so is $\bar{f} = x^{t-1} + \frac{b_{(t-2)v}}{b_{(t-1)v}}x^{t-2} + \ldots + \frac{b_{0v}}{b_{(t-1)v}} \in \mathbb{F}_q(y)[x]$. This is contradiction to f is the minimal polynomial of a over $\mathbb{F}_q(y)$ and $\deg(f) \geq t$. So $g = x^t + b_{(t-1)0}x^{t-1} + \ldots + b_{00} \in \mathbb{F}_q[y, x]$ with $b_{i0} \in \mathbb{F}_q[y]$. Thus $f = g$.

□

To modify Algorithm 3.4 for selecting decomposing element in Section 3.1, note that from the proof of Theorem 34, the number of bad elements (non-decomposing elements) we may select is fixed, independent to what the ground field is. Thus, we can just extend our ground set $\mathbb{F}_q$ where we choose the random element to $\{f \in \mathbb{F}_q[y] | \operatorname{Deg}(f) < l\}$, i.e. select $(c_1, c_2, \ldots, c_n) \in \{f \in \mathbb{F}_q[y] | \operatorname{Deg}(f) < l\}^n$ in the first step of Algorithm 3.4, for some sufficiently large l such that $q^l \geq \frac{nm^2}{\epsilon}$.

Note that there are also requirements for q in the Lecerf's algorithm of bivariate polynomial factorization and the algorithm of extended Euclidean algorithm. For the algorithm of bivariate polynomial factorization, when q is small, we can replace the algorithm of Lecerf with other algorithms which have no requirement about q, such as the algorithm of Lenstra [30] and the one of von Zur Gathen and Kaltofen [15]. Now assume $p \leq q \leq 3m^2\Delta$, we will use Lenstra's result here (Theorem 2.18, [30]), as follows:

Theorem 57. *Let f be a polynomial in $\mathbb{F}_q[x, y]$. Then the factorization of f into irreducible factors in $\mathbb{F}_q[x, y]$ can be determined in $O(d_x^6 d_y^2 + d_x^3 pl + d_y^3 pl)$ arithmetic operations in $\mathbb{F}_q$, where d_x (d_y) is the degree of f with respect to x (y) and $q = p^l$.*

Investigating into the proof of this theorem, we know that for our minimal polynomial, primitive with respect to x, the cost is $O(d_x^6 d_y^2 + d_x^3 pl) = O^{\sim}(m^8\Delta^2 + m^5\Delta) = O^{\sim}(m^8\Delta^2)$. We prefer Lenstra's result [30] to that of Gathen and Kaltofen [15] here. Because the result of Gathen and Kaltofen [15] focuses on the total degree of f, which makes the complexity has a large exponent for Δ.

For the extended Euclidean algorithm, given $f, g \in \mathbb{F}_q[y, x]$, we would like to compute s,$t \in \mathbb{F}_q(y)[x]$ such that $sf + tg = 1$. Treating f, g as polynomials in $\mathbb{F}_{q^l}[y, x]$ for a sufficiently large l we can compute $\bar{s}$,$\bar{t} \in \mathbb{F}_{q^l}(y)[x]$ such that $\bar{s}f + \bar{t}g = 1$. By the proof of the theorem about the degree bound of $\bar{s}$ and $\bar{t}$ ([14], Theorem 6.54, pp. 175), $\sigma_k\bar{s}f + \sigma_k\bar{t}g = \sigma_k r_k$ and $\sigma\bar{s}$ and $\sigma\bar{t}$ are in $\mathbb{F}_{q^l}[y]$ with degree bound $m^2\Delta$ where $r_k = 1$. So after we compute the value of $\bar{s}$ and $\bar{t}$ we can first compute the lcm of their denominators, denoted by σ. Thus, $\sigma\bar{s}f + \sigma\bar{t}g = \sigma$. Again assume $h(z)$ is the minimal polynomial with degree $l-1$ of a generating element of $\mathbb{F}_{q^l}$ over $\mathbb{F}_q$.

Let the jth coefficients of $\sigma\bar{s}$, $\sigma\bar{t}$ and σ is as follows,

$$\begin{aligned}[]
[\sigma\bar{s}]_j &= a_{j(m^2\Delta)}y^{m^2\Delta} + \ldots + a_{j0};\\
[\sigma\bar{t}]_j &= b_{j(m^2\Delta)}y^{m^2\Delta} + \ldots + b_{j0};\\
[\sigma]_j &= c_{j(m^2\Delta)}y^{m^2\Delta} + \ldots + c_{j0},
\end{aligned}$$

where a_{ji}, b_{ji} and c_{ji} are in $\mathbb{F}_q[z]$ for $0 \leq i \leq m^2\Delta$ with degree bound $l-1$. Rewrite $\sigma\bar{s}$, $\sigma\bar{t}$ and σ as the polynomial with indeterminate z,

$$\begin{aligned}
\sigma\bar{s} &= \bar{a}_{l-1}z^{l-1} + \ldots + \bar{a}_0;\\
\sigma\bar{t} &= \bar{b}_{l-1}z^{l-1} + \ldots + \bar{b}_0;\\
\sigma &= \bar{c}_{l-1}z^{l-1} + \ldots + \bar{c}_0,
\end{aligned}$$

where $\bar{a}_i$ and $\bar{b}_i$ in $\mathbb{F}_q[y,x]$ and $\bar{c}_i \in \mathbb{F}_q[y]$ for $0 \leq i \leq l-1$ with degree bound with respect to y as $m^2\Delta$. Since $\sigma \neq 0$, there at least one $\bar{c}_i \neq 0$. Suppose it is $\bar{c}_j$. So $\bar{a}_j f + \bar{b}_j g = \bar{c}_j$ are an equation in $\mathbb{F}_q[y,x]$. Therefore $\frac{\bar{a}_j}{\bar{c}_j}f + \frac{\bar{b}_j}{\bar{c}_j}g = 1$ in $\mathbb{F}_q(y)[x]$ with degree bound of y for the coefficients of $\frac{\bar{a}_j}{\bar{c}_j}$ and $\frac{\bar{b}_j}{\bar{c}_j}$ as $m^2\Delta$. Hence we can compute the extended Euclidean algorithm over $\mathbb{F}_{q^l}(y)$ first and then get the final result in $\mathbb{F}_q(y)$ with the same degree bound.

Thus, based on what we state above, we can modify our algorithm for the Wedderburn decomposition for the algebra over $\mathbb{F}_q(y)$ when q is small.

For Algorithm 47 in Section 3.2, it is similar to Algorithm 34 of Section 3.1. So we can apply the same trick here to modify the algorithm to adapt our algorithm for computing the radical to the case of small q.

To analyze its complexity, note that the lower bound for the size of q is all polynomial in m, n, Δ and $\frac{1}{\epsilon}$. Thus, l is polynomial in $\log m$, $\log n$, $\log \Delta$ and $\log \frac{1}{\epsilon}$. Since we only care about the soft complexity, we ignore the part of $\log m$, $\log n$, $\log \Delta$ except $\log \frac{1}{\epsilon}$. For the algorithm for the Wedderburn decomposition, we select $(c_1, c_2, \ldots, c_n) \in \{f \in \mathbb{F}_q[y] | \operatorname{Deg}(f) < l\}^n$, therefore the degree of α will be $\Delta + l$. Then the complexity is $O^{\sim}(m^{\omega+4}(\Delta + \log \frac{1}{\epsilon}) + m^8(\Delta + \log \frac{1}{\epsilon})^2) = O^{\sim}(m^8(\Delta + \log \frac{1}{\epsilon})^2)$. Similarly for the algorithm for computing the radical, it requires $O^{\sim}(m^8(\Delta + \log \frac{1}{\epsilon})^2 + n^2m^{\omega+4}(\Delta + \log \frac{1}{\epsilon}))$ operations over $\mathbb{F}_q$ and the degree bound of the returned bases is $O(m^2(\Delta + \log \frac{1}{\epsilon}))$. So the complexity of computation of the radical over $\mathbb{F}_q$ is $O^{\sim}(m^8(\Delta + \log \frac{1}{\epsilon})^2 + n^2m^{\omega+4}(\Delta + \log \frac{1}{\epsilon})) = O^{\sim}(n^2m^{\omega+4}(\Delta + \log \frac{1}{\epsilon})^2)$.

Chapter 4

Future Work

Recall that there are at least two questions remaining unsolved. One is to make our algorithms of Las Vegas type. One possible way is to make all the steps, which are probabilistic in our algorithm, to be of Las Vegas type. We already propose the algorithms of Las Vegas type for selecting a maximal linearly independent vector subset, computing the minimal polynomial as well as determining the equivalent classes of idempotents. The remaining question is how to check that the random element we choose from the algebra is really a decomposing element. In the Wedderburn decomposition, recall that the decomposing element is the element with highest degree for its minimal polynomial. Suppose the basis of the semisimple algebra $A \in \mathbb{F}_q(y)^{m\times m}$ is $\{a_1, a_2, \ldots, a_n\}$. Then, by Lemma 33, the maximal rank of an element in A is the same as that of the multinomial polynomial matrix $a_1z_1 + a_2z_2 + \ldots + a_nz_n \in \mathbb{F}_q(y)[z]^{m\times m}$, where z_i are the indeterminants of the polynomial. However we have no efficient algorithm to compute the rank of this matrix. Also, this way does not work for computing the radical directly thanks to the more general definition of decomposing element in Subsection 3.2.3. Another possible method to make it of Las Vegas type is developing an algorithm to check the result. When the ground field $F = \mathbb{F}_q$, Eberly and Giesbrecht [7] present an analysis to check the result. When $F = \mathbb{F}_q(y)$, it is still unsolved.

The other open question is to compute the isomorphic mapping from the simple component to a full matrix algebra $M_t(D)$, where D is an extended division ring of $\mathbb{F}_q(y)$. Actually this question is equivalent to computing the primitive idempotents, or computing the zero divisors. In the paper of Giesbrecht and Zhang [19], this isomorphic mapping is taken advantage of to factor the Ore polynomials. A similar idea of computing the primitive elements is selecting a random element from the simple algebra and then proving it is with large probability to decompose the algebra, so that we can compute a zero divisor (or decompose the idempotents in the algebra until they are primitive). Let $T(n, d, q)$ be the set of all d-inc polynomials in $\mathbb{F}_q[x, y]$ of degree in x being n, $t(n, d, q) = |T(n, d, q)|$ and $r(n, d, q)$ the number of reducible polynomials in $T(n, d, q)$. The following proposition, similar to Proposition 2.1 in Gao and Lauder's paper [12], shows that the reducible polynomials in $T(n, d, q)$ is sparse when n is large.

Proposition 58. *for $n \geq 6$ and $q \geq 2$,*

$$\frac{r(n,d,q)}{t(n,d,q)} \leq \frac{4}{3} \cdot \frac{1}{q^{d(n-1)}}.$$

Proof. Let $f = x^n + a_{n-1}x^{n-1} + \ldots + a_0 \in T(n,d,q)$. It is straightforward that $t(n,d,q) = q^{\sum_{i=1}^{n} id} = q^{d\frac{n(n+1)}{2}}$. If a polynomial in $T(n,d,q)$ is reducible, then one of its factor must have degree in x between 1 and $\frac{n}{2}$ and all of its factors are in d-inc form by Lemma 11. So

$$\begin{aligned} r(n,d,q) &\leq \textstyle\sum_{i=1}^{n/2} t(i,d,q)t(n-i,d,q) \\ &= \textstyle\sum_{i=1}^{n/2} q^{d(\frac{i(i+1)}{2}+\frac{(n-i)(n-i+1)}{2})}. \end{aligned}$$

It follows that

$$\frac{r(n,d,q)}{t(n,d,q)} \leq \sum_{1 \leq i \leq \frac{n}{2}} \frac{1}{q^{d(n-i)i}}$$

By the proof of Proposition 2.1 in Gao and Lauder's paper [12], $\frac{r(n,d,q)}{t(n,d,q)} \leq \frac{4}{3} \cdot \frac{1}{q^{d(n-1)}}$ for $q \geq 2$ and $n \geq 6$.

□

Besides, we need to analyze the number of matrices which have a "good" Frobenius form over D, where D is an extended infinite division ring of $\mathbb{F}_q(y)$. When the ground field is $\mathbb{F}_q$, we only need to do the analysis over a nice field, finite field. However, when $F = \mathbb{F}_q(y)$, D is infinite, which makes it much more difficult. A way to deal with the infinite ground field when analyzing the decomposing elements is presented in Theorem 34 and Theorem 47. But a similar way to analyze the "good" element, which decomposes the idempotent in an unique simple algebra, is not known yet.

Another possible application of primitive idempotents is computing the radical of a general matrix algebra, following the idea of Ivanyos's paper [23]. In this way, we also need to be careful about the degree explosion. Let $A = S + \mathrm{Rad}(A)$ and $C = C_A(S)$, the centralizer of S in A. Another problem of this idea is from the inseparability. For a local algebra A over $\mathbb{F}_q$, $A/\mathrm{Rad}(A)$ is a field, which is the case in the paper of Ivanyos [23] and makes the following key property holds: every element $c \in C$ can be uniquely written in the form $c = c_s + c_n$, where $c_s \in S$ and $c_n \in \mathrm{Rad}(A)$. Since $A/\mathrm{Rad}(A)$ is a field, $S \subset C$. However, when the ground field is $\mathbb{F}_q(y)$, $A/\mathrm{Rad}(A)$ is an inseparable division ring over $\mathbb{F}_q(y)$. We need to develop an alternative way to compute the radical of the local algebra A.

Bibliography

[1] J. R. Bastida. *Field Extensions and Galois Theory.* Cambridge University Press and Addison-Wesley 1984. 21

[2] A. Bostan, G. Lecerf, B. Salvy, É. Schost, and B. Wiebelt. Complexity issues in bivariate polynomial factorization. In ISSAC '04: *Proceedings of the 2004 international symposium on Symbolic and algebraic computation, pages 42-49.* ACM Press, 2004. 6

[3] D. G. Cantor and E. Kaltofen. On fast multiplication of polynomials over arbitrary algebras. *Acta Informatica, Volume 28, Number 7. pp. 693-701, 1991.* 4

[4] A. M. Cohen, G. Ivanyos and D. Wales. Finding the radical of an algebra of linear transformations. *Journal of Pure and Applied Algebra 117& 118 (1997) 177-193.* 33

[5] L. E. Dickson, *Algebras and Their Arithmetics.* Chicago, IL: University of Chicago Press. 32

[6] W. Eberly. *Asymptotically Efficient Algorithms for the Frobenius Form.* Department of Computer Science, Universiyt of Calgary, Technical Report, 2000. 10

[7] W. Eberly and M. Giesbrecht. Efficient decomposition of associative algebras over finite fields. *Journal of Symbolic Computation, v. 29, 2000, pp. 441-458.* iii, iv, 18, 19, 23, 47, 52

[8] W. Eberly and M. Giesbrecht. Efficient decomposition of separable algebras. *Journal of Symbolic Computation. Volume 37, Issue 1, pp. 35-81, 2004.* iii, 2, 19, 27

[9] K. Friedl and L. Rónyai. Polynomial time solutions of some problems in computational algebra. *Proc. 17th ACM STOC, pp. 153-162,1985.* iv, 18, 19, 32, 33

[10] A. Fröhlich and J. C. Shepherdson, Effective procedures in field theory, *Roy. Soc. London Phil. Trans. A 248 (1955-56) 407-432.* iii

[11] S. Gao. Factoring multivariate polynomials via partial differential equations. *Math. Comp.*, 72:801-822, 2003. 6

[12] S. Gao and A. G. B. Lauder, *Hensel Lifting and Bivariate Polynomial Factorisation over Finite Fields.* Mathematics of Computation. 71 (2002), no. 240, 1663-1676. 52, 53

[13] J. von zur Gathen. Algebraic complexity theory. *Annual Review of computer Science (1988) Vol. 3:317-348.* 5, 8

[14] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra.* Cambridge University Press, 1999. 5, 32, 50

[15] J. von zur Gathen and E. Kaltofen. Factorization of multivariate polynomials. J. Comput. System Sci., 31:265-287, 1985. 50

[16] P. Gianni, V. Miller and B. Trager. *Decomposition of Algebras.* Proc. ISSAC'88, p. 300-308, 1988. 18

[17] M. Giesbrecht. *Nearly optimal algorithms for canonical matrix forms.* SIAM Journal of Computing 24 (1995), 948-969. 10

[18] M. Giesbrecht and A. Storjohann. Computing rational forms of integer matrices. *Journal of Symbolic Computation. Vol. 34, No. 3, September 1, 2002.* 2, 4, 10, 12

[19] M. Giesbrecht and Y. Zhang, *Factoring and Decomposing Ore Polynomial over $\mathbb{F}_q(t)$.* ACM International Symposium on Symbolic and Algebraic Computation (ISSAC), pp. 127-134, 2003. 18, 52

[20] W. A. de Graaf and G. Ivanyos. Finding splitting elements and maximal tori in matrix algebras. In *F. van Oystaeyen and M. Saorin (Eds.), Interactions Between Ring Theory and Representations of Algebras, Lecture Notes in Pure and Applied Mathematics 210, Marcel Dekker, pp. 95-105, 2000.* 37, 40

[21] D. F. Holt and S. Rees. *Testing modules for irreducibility.* J. Aust. Math. Soc. 57, 1-16, 1994. 18

[22] G. Ivanyos. Finding the radical of matrix algebras using fitting decompositions. *Journal of Pure and Applied Algebra 139 (Proc. MEGA'98) (1999), 159-182.* 1, 2, 33, 36

[23] G. Ivanyos. Fast Randomized Algorithms for the Structure of Matrix Algebras over Finite Fields. *Proc. 2000 Int. Symp. on Symbolic and Algebraic Computation (ISSAC 2000), 175-183.* iv, 2, 33, 53

[24] G. Ivanyos, L. Rónyai and Á. Szántó. Decomposition of algebras over $\mathbb{F}_q(X_1, X_2, \ldots, X_m)$. *Algebra in Engineering, Communication and Computing 5 (1994), 71-90.* iii, iv, 1, 2, 18, 20, 22, 23, 33, 34, 35, 43, 47

[25] E. Kaltofen. A polynomial reduction from multivariate to bivariate integral polynomial factorization. *Proc. 14th ACM STOC, pp. 261-266. 1982.* 5

[26] E. Kaltofen. A polynomial-time reduction from bivariate to univariate integral polynomial factorization. In *Proceedings of the 23rd symposium on Foundations of Computer Science, pp. 57-64. IEEE, 1982.* 5

[27] E. Kaltofen. Sparse Hensel lifting. In *EUROCAL'85 Vol.2 (Linz, 1985)*, volume 204 of *LNCS*, *pp. 4-17.* Springer-Verlag, 1985. 5

[28] A. Kertész. *Lectures on Artinian Rings (ed. R. Wiegandt).* Budapest: Akadémiai Kiadó, 1987. 17

[29] G. Lecerf. New Recombination Algorithms for Bivariate Polynomial Factorization Based on Hensel Lifting. *Applicable Algebra in Engineering, Communication and Computing, Volume 21, Number 2, pp. 801-822, 2010.* 6, 30, 31

[30] A. K. Lenstra. *Factoring multivariate polynomials over finite fields.* J. Comput. System Sci. 2:235-248, 1985. 50

[31] H. Lüneburg, *On Rational Normal Form of Endomorphisms: a Primer to Constructive Algebra.* Wissenschaftsverlag, Mannheim, 1987. 10

[32] P. Ozello, *Calcul Exact Des Forms De Jordan et de Frobenius d'une Matrice.* PhD thesis, Universitpé Scientifique Technologique et Medicale de Grenoble, 1987. 10

[33] R. A. Parker. *The computer calculation of modular characters (the meat-axe).* In: Computational Group Theheory. Academic Press, London, pp. 267-274, 1984. 18

[34] R. S. Pierce, *Associative Algebras.* Springer-Verlag (Heidelberg), 1982. iii, 1, 17, 40

[35] L. Rónyai. *Simple algebras are difficult.* In: Proceedings, 19th ACM Symposium on Theory of Computing. New York, pp. 398-408, 1987. 18

[36] L. Rónyai. *Zero divisors in quaternion algebras.* J. Algorithms 9, 494-506, 1988. 18

[37] L. Rónyai. Computing the Structure of Finite Algebras. *Journal of Symbolic Computation (1990) 9, 355-373.* 17, 18, 19, 32, 33

[38] A. Schönhage and V. Strassen. Schnelle Multiplikation großer zahlen. *Journal of Computing, 7, pp. 281-292, 1971.* 4

[39] J. T. Schwartz, Fast probabilistic algorithms for verification of polynomial identities. *J. Assoc. Comput. 701-717, March 27, 1980.* 2

[40] A. Storjohann. *An $O(n^3)$ algorithm for Frobenius normal form.* In O.Gloor, editor, Proc. ISSAC'98, pp. 101-104. ACM Press, New York, 1998. 10

[41] A. Storjohann and G. Villard. *Algorithms for Similarity Transforms.* Rhine Workshop on Computer Algebra, Bregenz, Austria, 2000. 10

[42] A. Storjohann & G. Villard. Computing the rank and a small nullspace basis of a polynomial matrix. *ISSAC'05, Beijing, China, pp. 309-316.* ACM Press, 2005.

8, 9