Cohn and Umans' Matrix Multiplication and the Matrix Isomorphism Problem

Khai Dong

June 6, 2023

Abstract

Matrices are used in a large array of disciplines including mathematics, physics, and computer science to model multi-dimensional properties. Specifically, modeling with matrices allows for complex properties of subject matter to be translated into basic mathematical operations. With its diverse applications, improving the run-time of the operation is of great importance. Henry Cohn and Christopher Umans [3, 4] have proposed a theoretical framework for faster matrix multiplication algorithms and conjectured an upper bound of $O(n^{2+o(1)})$ on its runtime. Since this framework allows faster matrix multiplication, it is an active area of research. However, most of the steps remained unexplored and unimplemented. My research aims to solve the explicit isomorphism problem, the last of these remaining steps. In this proposal, I will first provide a high-level overview of Cohn and Umans' proposed approach and the details regarding the explicit isomorphism problem.

Contents

1 Introduction

Matrices have a wide range of applications across various fields including, but not limited to, physics, economics, and computer science. For instance, a direct application of matrices and matrix multiplication in machine learning is linear fitting, which is the foundation for many machine learning models [1]. Due to matrix multiplication's wide computational use, improvements in the runtime of this operation improve the efficiency of a wide array of computation tasks.

A matrix is a rectangular array of numbers arranged into rows and columns. A matrix with m rows and n columns has $m \times n$ entries (numbers) where m and n are arbitrary positive integers. An example of a 2×3 matrix with 6 entries is

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

Matrix multiplication between two matrices is defined when the number of columns in the first matrix is equal to the number of rows in the second matrix. In more mathematical terms, the multiplication of matrix A of size $m \times n$ and matrix B of size $n \times p$ is defined as A and B both having n columns and rows, respectively. The resulting matrix C is of size $m \times p$. This process of multiplying two matrices involves taking the dot products of the first matrix's rows with the second matrix's columns. For example, we have,

$$\begin{bmatrix} 1 & 2 \end{bmatrix} \cdot \begin{bmatrix} 3 & 5 \\ 4 & 6 \end{bmatrix} = \begin{bmatrix} 1 \cdot 3 + 2 \cdot 4 & 1 \cdot 5 + 2 \cdot 6 \end{bmatrix} = \begin{bmatrix} 11 & 17 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 3 & 5 \\ 4 & 6 \end{bmatrix} = \begin{bmatrix} 1 \cdot 3 + 2 \cdot 4 & 1 \cdot 5 + 2 \cdot 6 \\ 0 \cdot 3 + 0 \cdot 4 & 0 \cdot 5 + 0 \cdot 6 \end{bmatrix} = \begin{bmatrix} 11 & 17 \\ 0 & 0 \end{bmatrix}$$

Generally, we can observe that matrix multiplication of non-square matrices can be reduced to the problem of multiplying square matrices by adding entries 0 in the missing rows and columns. Therefore, it is sufficient to solely focus on multiplications of square matrices.
Let A and B be $n \times n$ square matrices to be of the form

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{bmatrix} \qquad B = \begin{bmatrix} b_{1,1} & b_{1,2} & \dots & b_{1,n} \\ b_{2,1} & b_{2,2} & \dots & b_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n,1} & b_{n,2} & \dots & b_{n,n} \end{bmatrix}$$

The product of these 2 matrices, $C = A \cdot B$, is defined as an $n \times n$ matrix, with entries

$$c_{i,j} = \sum_{k=1}^{n} a_{i,k} \cdot b_{k,j}$$

where $a_{i,j}$, $b_{i,j}$, and $c_{i,j}$ are entries at i^{th} row and j^{th} column of matrices A, B, and C, respectively. Naively, following the definition, each entry of C takes n multiplications and $n-1$ sums, which is a total of $2n-1$ operations. Accordingly, the computation of the whole matrix C costs $n^2 \cdot (2n-1) = O(n^3)$ operations. For brevity, we define ω to be the power of the runtime $O(n^\omega)$. For this naive algorithm, $\omega = 3$. Since the matrix A and B both have n^2 entries, we also have a natural lower bound of 2. Some researchers question whether we could multiply matrices at $\omega = 2$.
In 1969, Strassen [9] derived an improved algorithm yielding $\omega = 2.81$. This breakthrough raises the question of what the smallest value of ω that can be achieved. In 1987, Strassen [9]'s approach is further improved by Coppersmith and Winograd [5], yielding $\omega = 2.38$. In 2003 and 2005, Cohn and Umans [4, 3] proposed a framework for matrix multiplication and conjectured an upper bound of $\omega = 2$. However, as impressive as it is, this framework is largely unimplemented. Anderson et al. [2] have been able to implement the first and last steps of this framework, and Dubinsky [6] has partially implemented the remaining steps.
The goal of this research is to finish the remaining work: Solving the explicit isomorphism problem. By solving this problem, I hope to perform Cohn and Umans' matrix multiplication that yields $\omega < 3$. The context and details of this problem are presented in Sections 2 and 3.

2 Background and Related Works

This section presents some background information to build up to the explicit isomorphism problem. We noted that the abstract algebra used in this the rest of this proposal is based on *Abstract Algebra* by Dummit and Foote [7].
Firstly, one important result of representation theory given in Cohn and Umans [4] is the Wedderburn-Artin Theorem.

2.1 Wedderburn-Artin Theorem

The **key takeaway** from this section is the Wedderburn-Artin Theorem guarantees the existence of Wedderburn decomposition Φ (a bijective function), which, informally, takes a matrix and decomposes it into smaller matrices such that matrix multiplication holds under Φ. This decomposition of matrices into smaller matrices plays a significant part in the speed-up provided by Cohn and Umans' matrix multiplication framework. Specifically, in multiplying $n \times n$ matrices, we get efficiency improvement if the cost of naively multiplying 2 matrices is more than that of multiplying the matrices in the Wedderburn Decomposition. In mathematical notation, we gain efficiency if the following inequality holds:

$$n^3 > \sum_{i=1}^{k} d_i^3 \tag{1}$$

where n is the size of the original matrix, and d_i's are the sizes of matrices in the original matrix's Wedderburn decomposition.

The specific details regarding the Wedderburn-Artin Theorem and the Wedderburn Decomposition are as follows:

Theorem 1 *(Wedderburn-Artin) A semisimple group algebra $\mathbb{F}[G]$ can be decomposed into the direct product of full matrix algebras:*

$$\mathbb{F}[G] \cong \bigoplus_{i=1}^{k} \mathbb{D}_i^{d_i \times d_i} \tag{2}$$

where d_i is the order of matrix algebra $\mathbb{D}_i^{d_i \times d_i}$.

Here, a group G is a collection of elements defined under an operation $*$ such that the following holds among other properties:

$$\forall g_1, g_2 \in G, g_1 * g_2 \in G.$$

A group algebra $\mathbb{F}[G]$ is a collection of sums of products of elements in $\mathbb{F}$ and those of G. For example, if $G = \{g_1, g_2, g_3, g_4\}$ and $\mathbb{F} = \mathbb{Q}$ (the rationals), then one element of $\mathbb{Q}[G]$ is

$$\frac{1}{2}g_1 + 2g_2 + 4g_3 + 0g_4.$$

A matrix algebra $\mathbb{D}_i^{d_i \times d_i}$ is a collection of $d_i \times d_i$ matrices with entries in $\mathbb{D}_i$. An element of $\mathbb{Q}^{2\times 2}$ could be:

$$\begin{bmatrix} \frac{1}{2} & 2 \\ 4 & 0 \end{bmatrix}.$$

Equation (2) means $\mathbb{F}[G]$ is isomorphic to $\bigoplus_{i=1}^{k} \mathbb{D}_i^{d_i \times d_i}$. This means there exists a isomorphism $\Phi : \mathbb{F}[G] \to \bigoplus_{i=1}^{k} \mathbb{D}_i^{d_i \times d_i}$. An isomorphism, informally, is a bijective function from the domain to the codomain ($\mathbb{F}[G]$ to $\bigoplus_{i=1}^{k} \mathbb{D}_i^{d_i \times d_i}$, in this case) such that all operations in the domain holds in the codomain. This isomorphism Φ is the Wedderburn decomposition. How the direct sum, $\oplus$, works will be demonstrated in the following example:

Consider the semisimple group algebra $\mathbb{C}[G]$ with some group G over the complex numbers $\mathbb{C}$ with the following decomposition:

$$\mathbb{C}[G] \cong \mathbb{C}^{2\times 2} \oplus \mathbb{C}^{3\times 3}.$$

Noted that the D_is in the Wedderburn decomposition of $\mathbb{C}[G]$ are just $\mathbb{C}$.
For $C \in \mathbb{C}[G]$, we have that

$$\Phi(C) = A \oplus B = \begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix} & 0 \\ 0 & \begin{bmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,1} & b_{3,2} & b_{3,3} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} a_{1,1} & a_{1,2} & 0 & 0 & 0 \\ a_{2,1} & a_{2,2} & 0 & 0 & 0 \\ 0 & 0 & b_{1,1} & b_{1,2} & b_{1,3} \\ 0 & 0 & b_{2,1} & b_{2,2} & b_{2,3} \\ 0 & 0 & b_{3,1} & b_{3,2} & b_{3,3} \end{bmatrix}$$

where A and B are 2×2 and 3×3 matrices with entries in $\mathbb{C}$. Similarly, with $C' \in \mathbb{C}[G]$, we have a similar decomposition:

$$\Phi(C') = A' \oplus B' = \begin{bmatrix} A' & 0 \\ 0 & B' \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} a'_{1,1} & a'_{1,2} \\ a'_{2,1} & a'_{2,2} \end{bmatrix} & 0 \\ 0 & \begin{bmatrix} b'_{1,1} & b'_{1,2} & b'_{1,3} \\ b'_{2,1} & b'_{2,2} & b'_{2,3} \\ b'_{3,1} & b'_{3,2} & b'_{3,3} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} a'_{1,1} & a'_{1,2} & 0 & 0 & 0 \\ a'_{2,1} & a'_{2,2} & 0 & 0 & 0 \\ 0 & 0 & b'_{1,1} & b'_{1,2} & b'_{1,3} \\ 0 & 0 & b'_{2,1} & b'_{2,2} & b'_{2,3} \\ 0 & 0 & b'_{3,1} & b'_{3,2} & b'_{3,3} \end{bmatrix}$$

It is the case that $\Phi(C) \cdot \Phi(C') = \Phi(C \cdot C') = \begin{bmatrix} A \cdot A' & 0 \\ 0 & B \cdot B' \end{bmatrix}$ by naively multiplying out the 2 matrices in the decomposition.
Generally, we have that if $\Phi(A) = A_1 \oplus A_2 \oplus \cdots \oplus A_k$ and $\Phi(B) = B_1 \oplus B_2 \oplus \cdots \oplus B_k$ such that A_i and B_i are matrices of dimension $d_i \times d_i$,

$$\Phi(A) \cdot \Phi(B) = \Phi(A \cdot B) = \begin{bmatrix} A_1 \cdot B_1 & 0 & \dots & 0 \\ 0 & A_2 \cdot B_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A_k \cdot B_k \end{bmatrix}$$

This can be proven by naively multiplying out the matrices. This multiplying in the decomposition provides some speed up, which will be explained in the next subsection. We noted that Φ is an isomorphism (a bijection), so there exists its inverse Φ^{-1}, and we can obtain $A \cdot B$ by applying Φ^{-1} on $\Phi(A \cdot B)$:

$$\Phi^{-1}\left(\begin{bmatrix} A_1 \cdot B_1 & 0 & \dots & 0 \\ 0 & A_2 \cdot B_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A_k \cdot B_k \end{bmatrix} \right) = A \cdot B$$

Moreover, the equality,

$$\dim(\mathbb{F}[G]) = |G| = \dim\left(\bigoplus_{i=1}^{k} \mathbb{D}_i^{d_i \times d_i} \right) = \sum_{i=1}^{k} d_i^2, \tag{3}$$

holds since $\mathbb{F}[G]$ and $\bigoplus_{i=1}^{k} \mathbb{D}_i^{d_i \times d_i}$ are isomorphic.

2.2 Cohn and Umans' Matrix Multiplication

Cohn and Umans' matrix multiplication framework is as follows:
Let A and B be $n \times n$ matrices over a field $\mathbb{F}$, and G be a group of size at least n^2 satisfying a certain property that is described by Cohn and Umans [4] such that $\mathbb{F}[G]$ is a semisimple algebra. Let $C = A \cdot B$. We have the following approach:

Step 1: Map A and B to their respective elements in $\mathbb{F}[G]$, denoted $\overline{A}, \overline{B} \in \mathbb{F}[G]$, respectively.

Step 2: Compute the Wedderburn decompositions of $\overline{A}$ and $\overline{B}$, $\Phi(\overline{A})$ and $\Phi(\overline{B})$, respectively.

Step 3: Compute $\Phi(\overline{A}) \cdot \Phi(\overline{B}) = \Phi(\overline{A} \cdot \overline{B})$ as point-wise product as described in Section 2.1. Notice that each product is that of 2 matrices, so we can send them back to step 1 or do the naive matrix multiplication, whichever is more efficient.

Step 4: Compute $\Phi^{-1}(\Phi(\overline{A} \cdot \overline{B})) = \overline{A} \cdot \overline{B} = \overline{C} \in \mathbb{F}[G]$.

Step 5: Map $\overline{C} \in \mathbb{F}[G]$ to its respective matrix C.

Steps 4 and 5 hold since the mapping in Step 1 [4] and Φ are invertible. The improvement of the run-time comes from doing the point-wise product in Step 3 as described in Section 2.1. Moreover, since we are recursively multiplying the matrices in Step 3 to gain more efficiency, if this inequality holds for every pair of matrices, we gain further speed up.
In this framework, Steps 1 and 5 have been implemented by Anderson et al. [2]. Steps 2 and 4 have been partially implemented by Dubinsky [6]. Whereas, Step 3 is either a naive matrix multiplication or a recursion back to Step 1. The remaining work is to solve the explicit isomorphism problem, which is the final problem in Steps 2 and 4 of computing the Wedderburn decomposition.
It is worth noting that if group G is of size exactly n^2, since Equation 3 holds, we have that

$$n^2 = |G| = \sum_{i=1}^{k} d_i^2.$$

Then, Equation 1 always holds. This means if we find group G of size n^2, we gain speed-up regardless. However, it has been shown that the smallest group G can get is $n^{2+o(1)} > n^2$, but we do not know how close we can get to n^2. The problem of finding such group G of the smallest size possible is a separate problem from that which I am proposing to solve. However, it is worth mentioning since Cohn and Umans [4] asserted that small groups G could yield $\omega = 2$. This motivates my research to allow for a complete implementation of Cohn and Umans matrix multiplication.

2.3 Computing the Wedderburn Decomposition Φ

Before going into the explicit isomorphism problem, we first discuss how to compute the Wedderburn decomposition.
Let $\mathcal{A} = \mathbb{F}[G]$ be a semisimple algebra. There exists elements $e_1, e_2, \ldots, e_k \in \mathcal{A}$ satisfying certain properties where Peirce's Decomposition states that

$$\mathcal{A} \cong \bigoplus_{i=1}^{k} e_i \mathcal{A} e_i$$

and for each i, $e_i \mathcal{A} e_i \cong \mathbb{D}_i^{d_i \times d_i}$, a full matrix algebra of order d_i over some field $\mathbb{D}_i$. These e_i's are referred to as idempotents. The algorithm to calculate these idempotents has been developed by Dubinsky [6]. The remaining work is the explicit isomorphism problem which is to explicitly produce the isomorphism

$$\phi : e_i \mathcal{A} e_i \to \mathbb{D}_i^{d_i \times d_i}.$$

By producing all isomorphisms ϕ, we can produce Wedderburn Decomposition Φ.

3 Methods

In this section, I am going to formally define the explicit isomorphism problem and outline a plan of how I am going to approach this problem alongside the timeline of my thesis.

3.1 Explicit Isomorphism Problem

Given semisimple algebra $\mathcal{A} = \mathbb{F}[G]$ and one of its idempotent e. Find the isomorphism

$$\phi : e\mathcal{A}e \to \mathbb{D}^{d \times d}$$

where $\mathbb{D}$ and d are the field and the order of the matrix algebras, respectively.
In general, we want to solve the general problem where $\mathbb{F} = \mathbb{R}$, the set of real numbers. However, $\mathbb{R}$ contains transcendental elements (π and e for example) that can not be represented with a finite decimal representation or patterns. Doing computation with these transcendentals would ultimately result in numerical instability. To avoid the issue of numerical instability with $\mathbb{R}$, my research will focus on solving a

sub-problem of this with $\mathbb{F} = \mathbb{Q}$ where any element can be described using a pair of integers as the numerator and the denominator.
In the case of $\mathbb{F} = \mathbb{Q}$, $\mathbb{D} = \mathbb{Q}$ or $\mathbb{Q}(\zeta_n)$, a cyclotomic field. Cyclotomic fields, $\mathbb{Q}(\zeta_n)$, are the usual rationals along with the n^{th} roots of unity (all roots of x to $x^n = 1$). We can break this problem into 2 parts:

- The first part of the problem is identifying D which is either $\mathbb{Q}$ or $\mathbb{Q}(\zeta_n)$ and d, the order of the matrix algebra. This first part can be solved by library Wedderga in GAP and is also partially implemented by Dubinsky [6].
- The second part of the problem is to produce the isomorphism ϕ knowing $\mathbb{D}$ and d. Ivanyos et al. [8] provides one such algorithm in producing this isomorphism. However, there is no existing implementation of this algorithm.

Since the first part has been done and its implementation is publicly available, I am going to focus on the second part of the problem where we need to produce the isomorphism knowing the algebra $\mathcal{A}$, the idempotent e, the field of the matrix algebra $\mathbb{D}$, and the order the corresponding matrix algebra d.
For testing, generating test cases for the algorithm is simple since there are specific properties that ϕ has to satisfy being an isomorphism. Specifically, one of such properties is that when we send 2 elements in $e\mathcal{A}e$ through ϕ to their respective matrices in $\mathbb{D}^{d\times d}$, the product of those 2 matrices can be mapped back to the product of the 2 elements in $e\mathcal{A}e$ i.e., $A', B' \in e\mathcal{A}e$ implies

$$\phi(A') \cdot \phi(B') = \phi(A' \cdot B').$$

3.2 Approach and Timeline

Since Ivanyos et al. [8] provides one such algorithm, firstly, I need to understand the involved steps in his algorithm and attempt to recreate the process by hand with some simple examples. Currently, I plan to break the second part of explicitly computing ϕ into 2 cases:

- <u>Case 1:</u> $\mathbb{D}^{d\times d} = \mathbb{Q}^{d\times d}$ with $\mathbb{D} = \mathbb{Q}$ and arbitrary d.
- <u>Case 2:</u> $\mathbb{D}^{d\times d} = \mathbb{Q}(\zeta_n)^{d\times d}$ with $\mathbb{D} = \mathbb{Q}(\zeta_n)$ for arbitrary d and n.

By solving Case 2, we can solve Case 1 since $\mathbb{Q} = \mathbb{Q}(\zeta_1)$. However, since the explicit isomorphism problem is complex, it might be more palatable to break down the problem into cases. The following is a timeline to guide my research:

- Fall 2023:
  - Weeks 1-4: Reading topics in abstract algebra including but not limited to field and ring theories, and representation theory to gather the necessary background information to understand this research. Reviewing literature cited in [8] to understand the related subroutine.
  - Weeks 5-8: Replicating the approach by hand for Case 1 with some small groups following the results from weeks 1-4. Making posters.
  - Weeks 9-10: Continue to work on Case 1 or move to Case 2 if Case 1 has been fully understood. Writing thesis report.
- Winter 2023:
  - Week 1: Replicating the approach by hand for Case 2 for some small groups following the results from weeks 1-4 of Fall 2023.
  - Weeks 2-5: Implement the algorithm.
  - Weeks 6-10: Writing thesis report, making posters, and preparing for thesis defense. Testing the implemented algorithm in isolation.

Since this algorithm's implementation is not publicly available, this research pursues a novelty. This may result in adjustments to the timeline later on. Moreover, since I have a Summer Research Fellowship to work on the direction of this research for 8 weeks after the end of Spring 2023, I might be able to free up some of the first 4 weeks of Fall 2023 to focus more on the implementation of the algorithm and testing. This may also free up some time for me to assemble the full framework to actually perform Cohn and Umans' matrix multiplication, which is the end goal of this project.

3.3 Limitations

A limitation of this project is that since we are using $\mathbb{F} = \mathbb{Q}$, this algorithm does not work on $\mathbb{F} = \mathbb{C}$, the complex numbers and can only provide an approximation for the transcendental elements in $\mathbb{R}$, the real number. Until the problem of numerical instability is solved, it is not possible to completely solve this problem for $\mathbb{F} = \mathbb{C}$ or $\mathbb{R}$.

Due to time constraints, this research will only focus on producing a correct implementation of the algorithm. Further analysis, improvements of this algorithm, or assembling the pieces to perform Cohn and Umans' matrix multiplication is not included in the timeline. In my opinion, these tasks are less important at the moment and should be pursued after having a working implementation.

Moreover, since this project is heavily math-oriented, the end result may not be completely accessible to the general reader. For the sake of communication, the attempt to make the end result accessible may result in oversimplification in the final reports and presentations.

4 Why me? and Conclusion

By solving this problem, we will finally have all the pieces needed to perform Cohn and Umans' matrix multiplication. I think this project is possible within 20 weeks since I have been working with Professor Matthew Anderson's research team since the start of Fall 2022 and have been working towards this problem since Winter 2023. Moreover, I have been having discussions regarding this problem with Dubinsky [6] over Spring 2023 which helps me have a meaningful understanding of the background of the problem and the problem itself. As my Summer Research Fellowship provides me the opportunity to work on the direction of this project over the summer, I think before the start of Fall 2023, I could gather enough knowledge to pull this off.

References

[1] Charu C. Aggarwal. *Linear Algebra and Optimization for Machine Learning [electronic resource] : A Textbook / by Charu C. Aggarwal.* eng. 1st ed. 2020. Cham: Springer International Publishing, 2020. ISBN: 3-030-40344-0.

[2] Matthew Anderson, Zongliang Ji, and Anthony Xu. "Matrix Multiplication: Verifying Strong Uniquely Solvable Puzzles". In: June 2020, pp. 464–480. ISBN: 978-3-030-51824-0. DOI: `10.1007/978-3-030-51825-7_32`.

[3] H. Cohn et al. "Group-theoretic algorithms for matrix multiplication". In: *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)* (2005), pp. 379–388.

[4] Henry Cohn and Christopher Umans. "A group-theoretic approach to fast matrix multiplication". In: *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.* (2003), pp. 438–449.

[5] Don Coppersmith and Shmuel Winograd. "Matrix multiplication via arithmetic progressions". In: *Proceedings of the nineteenth annual ACM symposium on Theory of computing* (1987).

[6] Zachary Dubinsky. *Fast Matrix Multiplication and The Wedderburn-Artin Theorem*. 2023.

[7] David S. Dummit and Richard M. Foote. *Abstract Algebra*. en. 3rd ed. Nashville, TN: John Wiley Sons, 2003. ISBN: 9780471433347.

[8] Gábor Ivanyos, Lajos Rónyai, and Josef Schicho. "Splitting full matrix algebras over algebraic number fields". In: *Journal of Algebra* 354.1 (2012), pp. 211–223. ISSN: 0021-8693. DOI: https://doi.org/10.1016/j.jalgebra.2012.01.008. URL: https://www.sciencedirect.com/science/article/pii/S0021869312000300.

[9] V. Strassen. "Gaussian Elimination is not Optimal." In: *Numerische Mathematik* 13 (1969), pp. 354–356. URL: http://eudml.org/doc/131927.