Московский государственный университет
имени М. В. Ломоносова

Факультет вычислительной математики и кибернетики
Кафедра математической кибернетики

Отчет по научно-исследовательской работе

# «Поиск симметричного алгоритма для умножения матриц порядка 3 над полем из 2 элементов»

студента группы 281мк_дса

Иванова Андрея Александровича

**Научный руководитель:**

канд. физ.-мат. наук

Чокаев Б. В.

Москва, 2025

# Содержание

# 1 Введение

Вычисление произведения матриц является крайне распространённой задачей. Эта операция широко применяется в таких областях как компьютерная графика, моделирование физических процессов, обучение нейронных сетей и т.д.

Пусть необходимо вычислить произведение двух матриц размеров $n \times n$. Получаемый непосредственно из определения алгоритм «строка на столбец» требует $O(n^3)$ операций. В 1969 году Ф. Штрассен предложил алгоритм [1], который требует всего $O(n^{\log_2 7})$ операций для того же вычисления ($\log_2 7 \approx 2.807 < 3$). В этой же работе было показано, что на основе быстрого алгоритма умножения матриц могут быть построены алгоритмы вычисления определителя и обратной матрицы с той же асимптотической сложностью. Ключевым шагом алгоритма Штрассена является возможность умножить две матрицы $2 \times 2$ за 7 умножений вместо стандартных 8.

Пусть даны матрицы $A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$ и $B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}$. Требуется вычислить $C = AB$. Согласно алгоритму Штрассена вначале вычисляются линейные комбинации элементов матриц $\alpha_k = \sum u_{ij}^{(k)} a_{ij}$, $\beta_k = \sum v_{ij}^{(k)} b_{ij}$. Затем вычисляются произведения этих линейных комбинаций $\gamma_k = \alpha_k \beta_k$. Ответ вычисляется как $C = \sum \gamma_k W_k^T$. Где $W_k$ — 7 матриц $2 \times 2$ и $1 \leqslant i, j \leqslant 2$; $1 \leqslant k \leqslant 7$. Для того чтобы данная процедура корректно вычисляла произведение матриц необходимо и достаточно чтобы элементы матриц $U_k, V_k, W_k$ удовлетворяли трилинейной системе уравнений $\sum u_{i_1 j_1}^{(k)} v_{i_2 j_2}^{(k)} w_{i_3 j_3}^{(k)} = \delta_{i_1 j_3} \delta_{i_2 j_1} \delta_{i_3 j_2}$, где суммы берутся по $k$, $\delta_{ij}$ — символы Кронекера и уравнения должны быть выполнены для любого набора $(i_1, j_1, i_2, j_2, i_3, j_3)$.

Алгоритмы такого вида получили название билинейных. Далее задача умножения матрицы размера $m \times n$ на матрицу размера $n \times p$ будет обозначаться как $\langle m, n, p \rangle$. Можно показать, что минимальное количество умножений в билинейном алгоритме не меняется при перестановке $m, n, p$, и что имея алгоритм билинейной сложности $L$ для решения задачи $\langle m, n, p \rangle$ можно построить алгоритм умножения матриц произвольного размера $N$ со сложностью $O(N^\omega)$, где $\omega = 3 \log_{(mnp)} L$.

В 1971 году Ш. Виноград доказал, что для матриц данного размера алгоритм Штрассена оптимален, то есть невозможно умножить две матрицы $2 \times 2$, использовав менее 7 умножений [2].

Для задач $\langle m, n, 1\rangle$ стандартный алгоритм сложности $mn$ оптимален. Для задач $\langle 2, 2, n\rangle$ алгоритм Штрассена даёт верхнюю оценку $\lceil \frac{7n}{2} \rceil$. В работе [3] была доказана её оптимальность над полем из двух элементов. Над произвольным полем вопрос остаётся открытым. На настоящий момент известны также точные оценки сложности задач $L(\langle 2, 2, 3\rangle) = 11$ [5], $L(\langle 2, 2, 4\rangle) = 14$ [7], $L(\langle 2, 3, 3\rangle) = 15$ [10]. Для бóльших размеров матриц известные нижние и верхние оценки уже не совпадают: $17 \leqslant L(\langle 2, 2, 5\rangle) \leqslant 18$ [8], $19 \leqslant L(\langle 3, 3, 3\rangle) \leqslant 23$ [4, 6], $34 \leqslant L(\langle 4, 4, 4\rangle) \leqslant 49$ [6] [1].

Запишем коэффициенты алгоритма в таблицу

| | | |
|---|---|---|
| $U_1$ | $V_1$ | $W_1$ |
| $U_2$ | $V_2$ | $W_2$ |
| $\vdots$ | $\vdots$ | $\vdots$ |
| $U_r$ | $V_r$ | $W_r$ |

Обозначим столбцы буквами $\{u,\ v,\ w\}$, а строки числами $1..r$. Заметим, что система уравнений на коэффициенты алгоритма инвариантна относительно произвольной перестановки строк и циклической перестановки столбцов таблицы, а решение, соответствующее алгоритму Штрассена, является неподвижной точкой группы, порождённой перестановкой с цикленной записью $(u, v, w)(1)(2, 3, 4)(5, 6, 7)$. Также существует решение, являющееся неподвижной точкой группы, порождённой перестановкой $(u, v, w)(1)(2)(3)(4)(5, 6, 7)$.

Пусть некоторое решение системы (без повторяющихся строк) является неподвижной точкой какой-то группы $G \neq \{e\}$. Легко убедиться, что такая группа порождена элементом порядка 3. Такая перестановка состоит из циклов длин 1 и 3. Будем искать решение над полем $\mathbb{F}_2$, являющееся неподвижной точкой такой группы.

## 2 Основные понятия

Пусть $R$ — кольцо. Задача умножения матриц $\langle m, n, p\rangle$ состоит в том, чтобы по данным матрицам $A \in R^{m\times n}$ и $B \in R^{n\times p}$ вычислить матрицу $C = AB \in R^{m\times p}$.

[1] Указаны нижние оценки для произвольного поля и верхние оценки для кольца $\mathbb{Z}$.

В 2022 году была получена оценка $L(\langle 4, 4, 4\rangle) \leqslant 47$ над $\mathbb{F}_2$ [9]

В 2025 году была получена оценка $L(\langle 4, 4, 4\rangle) \leqslant 48$ над $\mathbb{Z}[\frac{1}{2}, i]$ [11], позже над $\mathbb{Z}[\frac{1}{2}]$ [12]

Билинейным алгоритмом называется алгоритм вида

$$\alpha_k := \sum_{i=1}^{m}\sum_{j=1}^{n} u_{ij}^{(k)} a_{ij}, \quad 1 \leqslant k \leqslant r$$

$$\beta_k := \sum_{i=1}^{n}\sum_{j=1}^{p} v_{ij}^{(k)} b_{ij}, \quad 1 \leqslant k \leqslant r$$

$$\gamma_k := \alpha_k \beta_k, \quad 1 \leqslant k \leqslant r$$

$$c_{ji} = \sum_{k=1}^{r} \gamma_k w_{ij}^{(k)}$$

Где $U_k = ||u_{ij}^{(k)}||,\ V_k = ||v_{ij}^{(k)}||,\ W_k = ||w_{ij}^{(k)}||$ — матрицы коэффициентов. Наименьшее $r$, при котором для задачи умножения матриц $\langle m, n, p\rangle$ существует билинейный алгоритм с коэффициентами из $R$, называется билинейной сложностью задачи умножения матриц над кольцом $R$ и обозначается $L_R(\langle m, n, p\rangle)$.

Будем записывать билинейные алгоритмы в виде таблиц коэффициентов с 3 строчками и $r$ столбцами, где в каждой клетке записана матрица коэффициентов.

Пусть $C_3$ — группа состоящая из трёх циклических перестановок столбцов таблицы коэффициентов, а $S_r$ — группа всех перестановок строк матрицы. Будем называть алгоритм (без повторяющихся строк) симметричным, если он является неподвижной точкой некоторой неединичной подгруппы $G \leqslant C_3 \times S_r,\ G \neq \{e\}$.

# 3 Постановка задачи

Необходимо разработать и протестировать программу, позволяющую найти или подтвердить отсутствие решения с указанными симметриями над $\mathbb{F}_2$, соответствующего алгоритму умножения матриц порядка 3 билинейной сложности 19, 20, 21 и 22.

# 4 Выполненная работа и полученные результаты

Разработана программа, основанная на методе, предложенном проф. В. Б. Алексеевым. Краткое изложение принципа работы. На первом шаге строятся все возможные способы заполнения диагоналей матриц коэффициентов. При этом способы противоречащие уравнениям, в которые входят только диагональные элементы отбрасываются, а из всех способов, переводимых друг в друга симметриями оставляется только один. На втором

шаге выбирается набор из $r$ переменных с одинаковыми индексами, и составляется система уравнений в которые эти переменные входят линейно, а значения остальных переменных уже зафиксированы. Эта система решается методом Гаусса. В пространстве её решений перебираются все наборы. Проверяются нелинейные уравнения, значения всех переменных в которых были зафиксированы. В случае невыполнения одного из уравнений дальнейшее рассмотрение этой ветки прерывается. Шаг 2 выполняется рекурсивно для следующего набора из $r$ переменных. Если значения всех переменных зафиксированы и не было найдено невыполненного уравнения, найдено решение.

Некоторые особенности реализации. Все уравнения строятся автоматически при помощи пакета sagemath, которые преобразуются в код на Си, что позволяет как предотвратить неизбежную ошибку при составлении тысяч уравнений вручную, так и проводить символьные вычисления лишь один раз. В реализации метода Гаусса векторы над $\mathbb{F}_2$ кодируются единственным целым числом, над которым проводятся побитовые операции. Пространство решений СЛАУ, представляющее собой булев куб, обходится по гамильтонову пути, для минимизации объёма вычислений. После решения каждой СЛАУ размерность пространства решений сравнивается с минимально допустимой на этом шаге. Если размерность меньше минимально допустимой, рассмотрение этой ветки немедленно прекращается. Минимально допустимая размерность определяется следующей леммой.

**Лемма.** Пусть $U_1, \ldots, U_r, V_1, \ldots, V_r, W_1, \ldots, W_r$ — коэффициенты билинейного алгоритма умножения матриц. $(i_1, j_1), \ldots, (i_t, j_t)$ — различные наборы индексов. Тогда векторы $(w^{(1)}_{i_1 j_1}, \ldots, w^{(r)}_{i_1 j_1}), \ldots, (w^{(1)}_{i_t j_t}, \ldots, w^{(r)}_{i_t j_t})$ линейно независимы.

Доказательство. Линейными комбинациями матриц $W_1^T, \ldots, W_r^T$ можно представить любую матрицу $C \in \mathbb{F}_2^{m \times p}$. Следовательно среди них есть $mp$ линейно независимых. Следовательно матрица

$$\begin{pmatrix} w^{(1)}_{11} & \ldots & w^{(1)}_{1m} & \ldots & w^{(1)}_{p1} & \ldots & w^{(1)}_{pm} \\ \vdots & & \vdots & & \vdots & & \vdots \\ w^{(r)}_{11} & \ldots & w^{(r)}_{1m} & \ldots & w^{(r)}_{p1} & \ldots & w^{(r)}_{pm} \end{pmatrix}$$

имеет ранг $mp$, т. к. у неё есть $mp$ линейно независимых строк. Следовательно все её $mp$ столбцов линейно независимы.

**Следствие.** Пусть векторы $(w^{(1)}_{i_1 j_1}, \ldots, w^{(r)}_{i_1 j_1}), \ldots, (w^{(1)}_{i_t j_t}, \ldots, w^{(r)}_{i_t j_t})$ удовлетворяют одной и той же однородной СЛАУ. Тогда размерность пространства её решений $\geqslant t$.

**Замечание.** Тот же результат верен и для двух других столбцов таблицы коэффициентов.

Эта же лемма позволяет не рассматривать нулевые решения СЛАУ.

Программа протестирована на хорошо исследованной задаче $\langle 2,\ 2,\ 2\rangle$. Для задачи $\langle 3,\ 3,\ 3\rangle$ проверен случай $r = 21$ причём все строчки таблицы коэффициентов разбиваются на группы по 3, на которые циклический сдвиг столбцов действует циклически. Вычисления производились на одном узле кластера IBM Polus [2], суммарно все вычисления заняли порядка 4.5 часов. В результате было установлено отсутствие решений с такой симметрией.

Исходный код размещён по адресу https://github.com/AIV5/cw2026

# 5 План дальнейших работ и ожидаемые результаты

В рамках магистерской диссертации планируется закончить перебор, тем самым установив точную оценку сложности задачи умножения матриц порядка 3 над полем из двух элементов в классе симметричных алгоритмов.

# Список литературы


[1] Strassen V. Gaussian elimination is not optimal // Numer. Math. 1969, № 13. p. 354–356.

[2] Winograd S. On multiplication of $2 \times 2$ matrices // Linear Algebra and Appl. 1971, № 4. p. 381–388.

[3] Hopcroft J. E., Kerr L. R. On minimizing the number of multiplications necessary for matrix multiplication // SIAM Journal on Applied Mathematics. 1971, T. 20, № 1. p. 30–36.

[4] Laderman J. D. A noncommutative algorithm for multiplying $3 \times 3$ matrices using 23 multiplications // Bull. Amer. Math. Soc. 1976, № 82 p. 126–128.

[5] Alekseyev V. B. On the complexity of some algorithms of matrix multiplication // Journal of Algorithms. 1985, № 6. p. 71–85.


[2] https://hpc.cs.msu.ru/polus

[6] Bläser M. On the complexity of the multiplication of matrices of small formats // Journal of Complexity. 2003, Т. 19. № 1. p. 43–60.

[7] Алексеев В. Б., Смирнов А. В. О точной и приближённой сложностях умножения матриц размеров $4 \times 2$ и $2 \times 2$ // Совр. проблемы матем. 2013, Вып. 17. С. 135–152.

[8] Алексеев В. Б. О билинейной сложности умножения матриц размеров $m \times 2$ и $2 \times 2$ // Чебышевский сборник. 2015, Т. 16. № 4 (56). С. 11–27.

[9] Fawzi A. et al. Discovering faster matrix multiplication algorithms with reinforcement learning // Nature. 2022, Т. 610. – №. 7930. – С. 47–53.

[10] Буриченко В. П. О билинейной сложности умножения $3\times2$ матрицы на $2\times3$ матрицу // Дискрет. Матем. 2024, № 36:1, С. 15–45.

[11] Novikov A. et al. AlphaEvolve: A coding agent for scientific and algorithmic discovery. [pdf]

[12] Dumas J.-G., Pernet C., Sedoglavic A. A non-commutative algorithm for multiplying $4\times4$ matrices using 48 non-complex multiplications. Technical Report 2506.13242, arXiv, June 2025. [arXiv]