# Computational Complexity of Matrix Multiplication

Andy He and Evan Williams

CS 6810 Fall 2023

**Abstract**

This survey provides an overview of the computational complexity of matrix multiplication, a pivotal operation in computational disciplines. It begins with an introduction to the fundamental principles of matrix multiplication and progresses through a series of influential works that have significantly advanced this field. Key highlights include an examination of Strassen's algorithm, Bini's contributions, Schonhage's methods, and the innovative laser method by Copperfield and Winograd. Each of these works is explored in terms of its impact on reducing computational complexity and improving efficiency. Additionally, the survey addresses the trade-offs inherent in algorithm design, particularly balancing computational speed with resource usage. It concludes with insights into how these seminal works continue to influence contemporary algorithm development and the future of matrix multiplication complexity.

# 1 Introduction

Algebraic complexity theory is the study of computation using algebraic models - namely, algebraic circuits. In this model, the inputs are variables $x_1, ..., x_n$ and computations are performed using the arithmetic operations $\times$, $+$, and can have constants from a field $\mathbb{F}$. The output of an arithmetic circuit is a polynomial in the input variables. The complexity measures associated with such circuits are **size** and **depth**, which represent the number of operations and the maximal distance between an input and output, respectively.

The study of algebraic complexity theory is largely concerned with two types of tasks: proving lower bounds on the computational complexity of algebraic problems and developing techniques to construct fast algorithms for computational problems with an algebraic structure. In this survey, we focus on the former with respect to an important computational problem: *matrix multiplication.* The multiplication of two $n \times n$ matrices, using the default algorithm, takes $O(n^3)$ operations. We will show how techniques developed over the past few decades can be used to construct an algorithm that multiplies two $n \times n$ matrices over a field $\mathbb{F}$ in $O(n^{2.38})$ time.

## 1.1 The Exponent of Matrix Multiplication

For the task of computing the matrix product of two $n \times n$ matrices $A$ and $B$ with entries in the field $\mathbb{F}$, we assume that the $2n^2$ entries $a_{ij}$ and $b_{ij}$ are given as input and we want to compute the value:

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

corresponding to the entry in the $i$-th row and $j$-th column of the product $C = AB$ for all $i, j \in 1...n$. Consider that the computational complexity of this problem is $O(n^\alpha)$.

**Definition 1.1 (Exponent of Matrix Multiplication)** *Let $mam(n)$ be the size of the smallest arithmetic circuit computing the matrix product of two $n \times n$ matrices. Then, the exponent of matrix multiplication $\omega$ is defined as:*

$$\omega = \inf\{\alpha : mam(n) = O(n^\alpha\}$$

From the trivial algorithm, we know that $\omega \leq 3$. Further, since there must be $n^2$ entries in the output, there cannot be any fewer operations than this in total matrix multiplication, so we have that $2 \leq \omega \leq 3$.

# 2 Bilinear Complexity Theory

## 2.1 Strassen's Algorithm

Strassen's algorithm improves on the naive matrix multiplication algorithm through a divide-and-conquer approach. The key insight is that multiplying two $n \times n$ matrices can be done using only seven multiplications as opposed to the eight required by the trivial algorithm. Consider the multiplication of two $2 \times 2$ matrices:

1. First compute the following seven terms:

$$\begin{aligned}
m_1 &= a_{11} \cdot (b_{12} - b_{22}) \\
m_2 &= (a_{11} + a_{12}) \cdot b_{22} \\
m_3 &= (a_{21} + a_{22}) \cdot b_{11} \\
m_4 &= a_{22} \cdot (b_{21} - b_{11}) \\
m_5 &= (a_{11} + a_{22}) \cdot (b_{11} + b_{22}) \\
m_6 &= (a_{12} - a_{22}) \cdot (b_{21} + b_{22}) \\
m_7 &= (a_{11} - a_{21}) \cdot (b_{11} + b_{12})
\end{aligned}$$

2. Then, output:

$$\begin{aligned}
c_{11} &= -m_2 + m_4 + m_5 + m_6 \\
c_{12} &= m_1 + m_2 \\
c_{21} &= m_3 + m_4 \\
c_{22} &= m_1 - m_3 + m_5 - m_7
\end{aligned}$$

Strassen's algorithm can be applied recursively to compute the product of two $2^k \times 2^k$ matrices for any $k \geq 1$. If the given matrices are not of size $2^k \times 2^k$, we simply zero-pad the input matrices to reach the next highest power of 2. The algorithm takes advantage of *blocking*, a common technique in numerical linear algebra in which we split a given matrix into four equally-sized sub-matrices:

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

Analyzing the complexity of recursion we have that:

$$\begin{aligned}
T(n) &\leq 7T(\frac{n}{2}) + cn^2 \\
&= \sum_{i=1}^{\log_2 n} 7^i((\frac{1}{2})^i n)^2 \\
&= \sum_{i=1}^{\log_2 n} 7^i(\frac{1^{2i}}{2^{2i}})n^2 \\
&= n^2 \sum_{i=1}^{\log_2 n} (\frac{7}{4})^i \\
&= O(n^2(\frac{7}{4})^{\log_2 n}) \\
&= O(n^{2.80735\ldots})
\end{aligned}$$

In other words, Strassen showed that $\omega \leq 2.80735$.

## 2.2 Bilinear Algorithms

A **bilinear algorithm** is an algebraic algorithm that proceeds in two steps. First, we compute $t$ products of the form:

$$\begin{aligned}
m_1 &= (\text{linear combination of } a_{ij}\text{'s}) \times (\text{linear combination of } b_{ij}\text{'s}) \\
&\vdots \\
m_t &= (\text{linear combination of } a_{ij}\text{'s}) \times (\text{linear combination of } b_{ij}\text{'s})
\end{aligned}$$

Then, each entry of the product $c_{ij}$ is computed by linear combinations of $m_1, ..., m_t$. The integer $t$ is called the **bilinear complexity** of this algorithm. Strassen's algorithm is just a bilinear algorithm that computes the product of two $2 \times 2$ matrices with bilinear complexity $t = 7$. In fact, Strassen's approach can be generalized to any bilinear algorithm for matrix multiplication. Let $A_{ik}$ be an $m \times n$ matrix and $B_{kj}$ be a $n \times p$ matrix. Then the elementary formula for matrix multiplication $C = AB$ can be represented by the following $np$ bilinear forms:

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

one for each entry in the product. This gives us the following proposition:

**Proposition 2.1** *Let $n$ be a positive integer. Suppose that there exists a bilinear algorithm that computes the product of two $n \times n$ matrices with bilinear complexity $t$. Then,*

$$\omega \leq \log_n(t)$$

To prove this proposition we will need to build up some machinery relating to bilinear maps.

## 2.3 Bilinear Maps

**Definition 2.2** *Let $U$, $V$, $W$ be vector spaces over a field $\mathbb{F}$. A* **bilinear map** *is a map: $\phi : U \times V \to W$ satisfying:*

$$\phi(\lambda_{11} u_1 + \lambda_{12} u_2, \lambda_{21} v_1 + \lambda_{22} v_2) = \sum_{i,j \leq 2} \phi(u_i, v_j)$$

*for all $\lambda_{i,j} \in \mathbb{F}$, $u_i \in U$, $v_j \in V$.*

Given this definition, it is easy to see that matrix multiplication is really a bilinear map.

**Definition 2.3** *Let $\phi : U \times V \to W$ be a bilinear map over field $\mathbb{F}$. For $i \in 1...r$, let $f_i \in U^*$, $g_i \in V^*$, $w_i \in W$ be such that:*

$$\phi(u, v) = \sum_{i=1}^{r} f_i(u) g_i(v) w_i$$

*for all $u \in U$, $v \in V$. Then, $(f_1, g_1, w_i; ...; f_r, g_r, w_r)$ is called a* **bilinear algorithm** *of length $r$ for $\phi$.*

**Definition 2.4** *The length of the shortest bilinear algorithm for $\phi$ is called the* **rank** *of $\phi$ and is denoted $R(\phi)$.*

Let's rewrite Strassen's algorithm as a bilinear algorithm:

- $f_1 = A_{11} + A_{22}, g_1 = B_{11} + B_{22}$
- $f_2 = A_{21} + A_{22}, g_2 = B_{11}$
- $f_3 = A_{11}, g_3 = B_{12} - B_{22}$
- $f_4 = A_{22}, g_4 = -B_{11} + B_{21}$
- $f_5 = A_{11} + A_{12}, g_5 = B_{22}$
- $f_6 = -A_{11} + A_{21}, g_6 = B_{11} + B_{12}$
- $f_7 = A_{12} - A_{22}, g_7 = B_{21} + B_{22}$

$$w_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad w_2 = \begin{pmatrix} 0 & 0 \\ 0 & -1 \end{pmatrix}, \quad w_3 = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}, \quad w_4 = \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix},$$

$$w_5 = \begin{pmatrix} -1 & 0 \\ -1 & 0 \end{pmatrix}, \quad w_6 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \quad w_7 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}.$$

Thus, the rank of the bilinear algorithm computing $2 \times 2$ matrix multiplication is at most 7. For the trivial algorithm, we have that the rank is at most 8. We denote the problem of multiplying an $m \times n$ matrix by a $n \times p$ matrix as $\langle m, n, p\rangle$. So we have that:

$$R(\langle 2, 2, 2\rangle) \leq 2$$

Next, we'll show that the rank of a concise bilinear map is greater than $\max(dim(U), dim(V), dim(W))$.

**Definition 2.5** *A bilinear map is* **concise** *if and only if the left kernel* $\{u \in U | \phi(u, v) = 0 \forall v \in V\} = 0$ *and the right kernel* $\{v \in V | \phi(u, v) = 0 \forall u \in W\} = 0$ *and if the span of* $\phi(U, V) = W$.

**Lemma 2.6** *The rank of a concise bilinear map is greater than* $\max(dim(U), dim(V), dim(W))$.

*Proof.* If the rank of a concise bilinear map is less than the dimension of $U$ then the $f_i$ do not form a basis for $U^*$. So one can always find a non-zero $u \in U$ such that $f_i(u) = 0$ for all $f_i$, and thus $\phi$ will have a non-zero kernel, contradicting the fact that the bilinear map is concise. An analogous argument holds true for $V$. If the rank of a bilinear map is less than the dimension of $W$, then the dimension of the image of $\phi(U, V)$ will be less than the dimension space of $W$, also contradicting conciseness. ■

We will continue to work with rank thorughout the survey, as it is better behaved than the number of operations. We define $\omega$ in terms of rank:

**Proposition 2.7** *For every field* $\mathbb{F}$, *we have:*

$$\omega(\mathbb{F}) = \inf\{\alpha \in \mathbb{R} | R(\langle h, h, h\rangle) = O(h^\alpha)\}$$

*Proof.* Using a bilinear algorithm, it can be shown that $mam(m^{i+1})$ is:

$$r \cdot mam(m^i) + cm^{2i}$$

for some $c$ that depends on $m$ and $r$. Solving the recurrence gives us:

$$mam(m^i) \leq \alpha r^i + \beta m^{2i}$$

where $\alpha = mam(1) + m^2 c/(r - m^2)$ and $\beta = -m^2 c/(r - m^2)$, which yields $mam(m^i) = O(r^i)$. Manipulating logs and plugging in the definition of $\omega$ gives us the desired result. ■

**Lemma 2.8** *The rank is invariant when permuting the sizes of matrices. i.e.,*

$$R(\langle e, h, l\rangle) = R(\langle h, l, e\rangle) = R(\langle l, e, h\rangle)$$

The proof of this lemma will require us to introduce new notation.

## 2.4 Tensors

**Fact 2.9** *If* $U$, $V$, $W$ *are vector spaces over a field* $\mathbb{F}$, *there exists a unique isomorphism* $U^* \otimes V^* \otimes W \to Bil(U, V; W)$ *which sends* $f \otimes g \otimes w$ *to the bilinear map* $(u, v) \to f(u)g(v)w$.

We'll omit this proof to focus on more interesting results, but this allows us to construct a tensor in $U^* \otimes V^* \otimes W$ instead of an explicit map.

**Definition 2.10** *Consider three finite-dimensional vector spaces* $U$, $V$ *and* $W$ *over the field* $\mathbb{F}$. *Take a basis* $\{x_1, \ldots, x_{\dim(U)}\}$ *of* $U$, *a basis* $\{y_1, \ldots, y_{\dim(V)}\}$ *of* $V$, *and a basis* $\{z_1, \ldots, z_{\dim(W)}\}$ *of* $W$. *A* **tensor** *over* $(U, V, W)$ *is an element of* $U \otimes V \otimes W$ *or, equivalently, a formal sum*

$$T = \sum_{u=1}^{\dim U} \sum_{v=1}^{\dim V} \sum_{w=1}^{\dim W} d_{uvw}\, x_u \otimes y_v \otimes z_w$$

*with coefficient* $d_{uvw} \in \mathbb{F}$ *for each* $(u, v, w) \in \{1, \ldots, \dim(U)\} \times \{1, \ldots, \dim(V)\} \times \{1, \ldots, \dim(W)\}$.

It is now possible to correspond matrix multiplication and tensors.

**Definition 2.11** *The tensor corresponding to the multiplication of an $m \times n$ matrix by an $n \times p$ matrix is*

$$\sum_{i=1}^{m}\sum_{j=1}^{p}\sum_{k=1}^{n} a_{ik} \otimes b_{kj} \otimes c_{ij}.$$

*This tensor is denoted $\langle m, n, p\rangle$.*

Let $T$ be a 3-dimensional tensor $(t_{i,j,k})$ where $i, j, k$ vary over a finite-index cube. A series of bilinear algorithms can be aggregated as a 3-dimensional tensor where each "slice" of the tensor represents a computation for $c_{ij}$ in matrix multiplication. The tensor $T$ can thus be decomposed into a sum of $r$ simpler tensors. We denote this decomposition as:

$$T = \sum_{s=1}^{r} u_i \otimes v_i \otimes w_i$$

where $u_i$, $v_i$, and $w_i$ are vectors and $\otimes$ denotes their tensor product. Thus, $r$ is the **rank** of the tensor decomposition. Decompositions of rank $r$ correspond to matrix multiplication algorithms with $r$ multiplications. We use $R(\langle m, n, p\rangle)$ to denote the rank of the minimal tensor decomposition. We now have a proof for Lemma 2.8. It is easy to see that the rank of $T$ is invariant under permutation of the coordinates, hence the rank of $\phi$ is also invariant under permutation. ■

We can see now that Strassen's algorithm can also be formulated as a tensor decomposition:

$$\begin{aligned}\langle 2,2,2\rangle = {} & a_{11} \otimes (b_{12} - b_{22}) \otimes (c_{12} + c_{22}) \\ & + (a_{11} + a_{12}) \otimes b_{22} \otimes (-c_{11} + c_{12}) \\ & + (a_{21} + a_{22}) \otimes b_{11} \otimes (c_{21} - c_{22}) \\ & + a_{22} \otimes (b_{21} - b_{11}) \otimes (c_{11} + c_{21}) \\ & + (a_{11} + a_{22}) \otimes (b_{11} + b_{22}) \otimes (c_{11} + c_{22}) \\ & + (a_{12} - a_{22}) \otimes (b_{21} + b_{22}) \otimes c_{11} \\ & + (a_{11} - a_{21}) \otimes (b_{11} + b_{12}) \otimes (-c_{22})\end{aligned}$$

We'll now examine an important theorem:

**Theorem 2.12** *Let $m$, $n$, $p$, and $t$ be four positive integers. If $R(\langle m, n, p\rangle) \leq t$, then:*

$$(mnp)^{\omega/3} \leq t$$

*Proof.* Recall that rank is multiplicative and that matrix-multiplication tensors are symmetric in $m$, $n$, and $p$. Thus, $R(\langle mnp, mnp, mnp\rangle) = R(\langle m, n, p\rangle)^3$. The proof then follows directly from Proposition 2.1. ■

## 2.5 Trilinear Aggregation

Until Pan discovered this method, nobody was able to improve upon Strassen's algorithm. We have the following crucial observation:

**Lemma 2.13** *Matrix multiplication is equivalent to finding the trace of the product of three matrices $A$, $B$, and $C$ of size $m \times n$, $n \times p$, and $p \times m$, respectively.*

*Proof.* Let $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{n\times p}$, and $C \in \mathbb{R}^{p\times m}$. The trace of $ABC$ is given by:

$$\mathrm{Tr}(ABC) = \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{p} A_{ij}B_{jk}C_{ki}$$

Set $C_{ki}$ to 1 and the other entries of $C$ to 0. Then, we find $\sum_{i=1}^{n} A_{ij}B_{jk}$, or $AB_{ik}$. Thus, we have that the problem of matrix multiplication is embedded in the problem of finding the trace of three matrices.

In the reverse direction, if the rank of matrix multiplication of two matrices $A$ and $B$ is $r$, then the product $ABC$ can be written as: '

$$\sum_{i=1}^{r} f_i(A)g_i(B)w_iC$$

for some $f_i \in \mathbb{F}^{m\times n^*}$, $g_i \in \mathbb{F}^{m\times p^*}$, $w_i \in \mathbb{F}^{p\times m}$. To find the trace of this product, we have that:

$$\mathrm{Tr}\sum_{i=1}^{r} f_i(A)g_i(B)w_iC = \sum_{i=1}^{r} f_i(A)g_i(B)\mathrm{Tr}(w_iC)$$

where we have that the function $\mathrm{Tr}(w_iC)$ is in the dual of $\mathbb{F}^{p\times m}$, so the overall problem of finding the trace of the product of three matrices is a **trilinear map** of the form:

$$\sum_{i=1}^{r} f_i(A)g_i(B)h_i(C)$$

where $h_i$ is found by taking the trace of each product of $w_iC$. So the rank of the two problems is equal. ■

The equation for the trace of a matrix given above is a **trilinear map** $\phi : U \times V \times W \to K$. The rank $r$ is the minimal number for which $f_i \in U^*$, $g_i \in V^*$, $h_i \in W^*$:

$$\phi(u, v, w) = \sum_{i=1}^{r} f_i(A)g_i(B)h_i(C)$$

Recall the trivial algorithm for finding the trace of the product of three matrices:

$$\sum_{i,j,k} a_{ij}b_{jk}c_{ki}$$

Each term in this sum is called **desirable**. Consider the product:

$$(a_{ij} + a_{k_1i_i})(b_{jk} + b_{i_ij_1})(c_{ki} + c_{j_1k_1})$$

which is equal to:

$$a_{ij}b_{jk}c_{ki} + a_{k1i}b_{i1j}c_{j1k1} + a_{k1i}b_{i1j}c_{jk} + a_{ij}b_{i1j}c_{j1k1} + a_{ij}b_{jk}c_{j1k1} + a_{k1i}b_{jk}c_{j1k1} + a_{ij}b_{jk}c_{j1k1}.$$

Observe that $i_1$, $j_1$, and $k_i$ are functions of $i$, $j$, and $k$, respectively. In other words, we can select $i_1$, $j_1$, $k_1$ and $i$, $j$, $k$ such that all possible combinations of $i$, $j$, and $k$ in the desired range are obtained. We call terms $(a_{ij} + a_{k_1i_i})$, $(b_{jk} + b_{i_ij_1})$, and $(c_{ki} + c_{j_1k_1})$ *aggregates* and the process of collecting together undesirable terms *uniting.*

Pan used this result to create an asymptotically faster algorithm than Strassen's. Assume that we intend to multiiply two $n \times n$ matrices, where $n = 2s$ is even. Summing over the set:

$$S^1 = \{(i, j, k) \mid 0 \le j < k \le s - 1\} \cup \{(i, j, k) \mid 0 \le k < j \le i \le s - 1\}$$

we can obtain the trace of the product of three matrices in $\frac{n^3-4n}{3} + 6n^2$ operations. Letting $n = 70$, we have that $\omega \le \log_{70} 143640 = 2.79512$, which is lower than Strassen's.

# 3 Approximate Bilinear Algorithms

## 3.1 Border Rank and Degeneration

In 1979, Bini introduced arbitrary precision approximation (APA) methods for matrix multiplication. More explicitly, he discovered that one could obtain algorithms with fewer scalar multiplications to compute at the cost of being only close approximations of the "correct" result. Let $\lambda$ be an indeterminate and $\mathbb{F}[\lambda]$ be the ring of polynomials in $\lambda$ with coefficients in the field $\mathbb{F}$.

**Definition 3.1 (Border Rank)** *Let $T$ be a tensor over $(U, V, W)$. The border rank of $T$, denoted $\underline{R}(T)$, is the minimal integer $t$ for which there exists an integer $c \geq 0$ and a tensor $T''$ such that $T$ can be written as:*

$$\lambda^c T = \sum_{s=1}^{t} \left[ \left( \sum_{u=1}^{\dim(U)} \alpha_{su} x_u \right) \otimes \left( \sum_{v=1}^{\dim(V)} \beta_{sv} y_v \right) \otimes \left( \sum_{w=1}^{\dim(W)} \gamma_{sw} z_w \right) \right] + \lambda^{c+1} T''$$

*for some constants $\alpha_{su}$, $\beta_{sv}$, $\gamma_{sw}$ in $\mathbb{F}[\lambda]$*

In order to motivate the notion of border rank, consider $\mathbb{F} = \mathbb{R}$. Then it can happen that the limit of a converging sequence of tensors has a higher rank than all of its approximants. This would normally be described by means of a variable $\lambda$ representing "small numbers". In this setting $\lambda$ is an extra indeterminate over $\mathbb{F}$.

We call the representation:

$$\sum_{i=1}^{r} f_i(\lambda) + g_i(\lambda) + h_i(\lambda) = \lambda^h t + O(\lambda^{h+1})$$

an **approximate decomposition** of order $h \geq 0$ of length $r$ of a tensor $T$ or its trilinear form $\psi$. This is also called an approximate algorithm for $\psi$. The minimal $r$ of this kind is the **approximate rank of order h**. The minimum over all possible values of $h$ is the border rank. Using this method, Bini found an approximation for the multiplication of two $3 \times 3$ matrices that takes only 21 multiplications, as opposed to the 23 required by the exact rank:

$$\begin{aligned}
F_1(\lambda) = {} & (a_{11} + \lambda a_{12})(\lambda b_{11} + b_{21})c_{11} \\
& + (a_{21} + \lambda a_{22})(\lambda b_{12} + b_{22})c_{22} \\
& + (a_{31} + \lambda a_{32})(\lambda b_{13} + b_{23})c_{33} \\
& - a_{11}(b_{21} + b_{31})(c_{11} + c_{12} + c_{13}) \\
& - a_{21}(b_{22} + b_{32})(c_{21} + c_{22} + c_{23}) \\
& - a_{31}(b_{23} + b_{33})(c_{31} + c_{32} + c_{33}) \\
& + (a_{11} + \lambda a_{22})(b_{21} - \lambda b_{12})c_{12} \\
& + (a_{21} + \lambda a_{12})(b_{22} - \lambda b_{11})c_{21} \\
& + (a_{11} + \lambda a_{32})(b_{21} - \lambda b_{13})c_{13} \\
& + (a_{31} + \lambda a_{12})(b_{23} - \lambda b_{11})c_{31} \\
& + (a_{21} + \lambda a_{32})(b_{22} - \lambda b_{13})c_{23} \\
& + (a_{31} + \lambda a_{22})(b_{23} - \lambda b_{12})c_{32} \\
& + (a_{31} + \lambda a_{22})(b_{23} - \lambda b_{12})c_{32} \\
& + (a_{11} + \lambda a_{23})(b_{31} + \lambda b_{12})(c_{12} + \lambda c_{21}) \\
& + (a_{21} + \lambda a_{13})(b_{32} + \lambda b_{11})(c_{21} + \lambda c_{12}) \\
& + (a_{11} + \lambda a_{33})(b_{31} + \lambda b_{13})(c_{13} + \lambda c_{31}) \\
& + (a_{31} + \lambda a_{13})(b_{33} + \lambda b_{12})(c_{31} + \lambda c_{13}) \\
& + (a_{21} + \lambda a_{33})(b_{32} + \lambda b_{13})(c_{23} + \lambda c_{32}) \\
& + (a_{31} + \lambda a_{23})(b_{33} + \lambda b_{12})(c_{32} + \lambda c_{23}) \\
& + (a_{11} + \lambda a_{13})b_{31}(c_{11} - \lambda c_{31} - \lambda c_{21}) \\
& + (a_{21} + \lambda a_{23})b_{32}(c_{22} - \lambda c_{32} - \lambda c_{12}) \\
& + (a_{31} + \lambda a_{33})b_{33}(c_{33} - \lambda c_{13} - \lambda c_{23}) \\
= {} & \lambda^2(\mathrm{Trace}(ABC) + \lambda G(\lambda)).
\end{aligned}$$

Clearly, $R(T) \leq \underline{R}(T)$ for any tensor $T$. Further, border rank is also invariant to cyclic permutation and we also have that:

$$\underline{R}(T \otimes T') = \underline{R}(T) \times \underline{R}(T')$$

Consider the example Bini produced:

$$\begin{aligned}T_{\text{Bini}} &= \sum_{\substack{1\le i,j,k\le 2\\(i,j,k)\neq(2,2)}} a_{ik}\otimes b_{kj}\otimes c_{ij}\\ &= a_{11}\otimes b_{11}\otimes c_{11} + a_{12}\otimes b_{21}\otimes c_{11} + a_{11}\otimes b_{12}\otimes c_{12} + a_{12}\otimes b_{22}\otimes c_{12}\\ &+ a_{21}\otimes b_{11}\otimes c_{21} + a_{21}\otimes b_{12}\otimes c_{22}\end{aligned}$$

This corresponds exactly to a matrix product of two $2\times 2$ matrices where one entry in the first matrix is zero. It can be shown that $R(T_{\text{Bini}}) = 6$. Bini showed that $\underline{R}(T_{\text{Bini}}) \le 5$ by exhibiting the identity:

$$\lambda T_{\text{Bini}} = T' + \lambda^2 T''$$

where we have that:

$$\begin{aligned}T' &= (a_{12}+\lambda a_{11})\otimes(b_{12}+\lambda b_{22})\otimes c_{12}\\ &+ (a_{21}+\lambda a_{11})\otimes b_{11}\otimes(c_{11}+\lambda c_{21})\\ &- a_{12}\otimes b_{12}\otimes(c_{11}+c_{12}+\lambda c_{22})\\ &- a_{21}\otimes(b_{11}+b_{12}+\lambda b_{21})\otimes c_{11}\\ &+ (a_{12}+a_{21})\otimes(b_{12}+\lambda b_{21})\otimes(c_{11}+\lambda c_{22})\end{aligned}$$

and:

$$T'' = a_{11}\otimes b_{22}\otimes c_{12} + a_{11}\otimes b_{11}\otimes c_{21} + (a_{12}+a_{21})\otimes b_{21}\otimes c_{22}$$

**Proposition 3.2** *There exists a constant $\alpha$ such that $R(T) \le \alpha \underline{R}(T)$ for any tensor $T$.*

Combining two copies of $T_{\text{Bini}}$, we have that:

$$\underline{R}(\langle 3,3,2\rangle) \le 10$$

Recall that we have the following two properties for border rank:

- $\underline{R}(T\otimes T') \le \underline{R}(T)\times\underline{R}(T')$
- $\underline{R}(\langle m,n,p\rangle) = \underline{R}(\langle m,p,n\rangle) = \underline{R}(\langle n,m,p\rangle) = \underline{R}(\langle n,p,m\rangle) = \underline{R}(\langle p,m,n\rangle) = \underline{R}(\langle p,n,m\rangle)$

So we have that:

$$\underline{R}(\langle 12,12,12\rangle) \le 1000$$

and thus:

$$R(\langle 12,12,12\rangle) \le \alpha \times 1000$$

by Proposition 3.2. Unfortunately this does not yield an interesting bound on $\omega$. Instead, consider the tensor:

$$\langle 12,12,12\rangle^{\otimes N} \approx \langle 12^N, 12^N, 12^N\rangle$$

for a large $N$. This gives us:

$$R(\langle 12^N,12^N,12^N\rangle) \le \alpha\times\underline{R}(\langle\rangle 12^N,12^N,12^N) \le \alpha\times 1000^N$$

Using Theorem 2.12, we have:

$$\begin{aligned}\omega &\le \log_{12}(\alpha^{\frac{1}{N}}\times 1000)\\ &\le \log_{12}(1000)\\ &< 2.78\end{aligned}$$

where we take the limit as $N\to\infty$. This is precisely the bound Bini obtained.

This analysis suggests that when deriving an upper bound on $\omega$, it suffices to consider border rank instead of rank. Indeed, we have the following theorem.

**Theorem 3.3** *Let $m$, $n$, $p$, and $t$ be four positive integers. If $\underline{R}(\langle m,n,p\rangle) \leq t$, then $(mnp)^{\omega/3} \leq t$*

*Proof.* Since we may symmetrize, we show this for $e = h = l = n$. By definition, we have that:

$$\underline{R}(\langle n,n,n\rangle) \leq t$$

And from above:

$$\underline{R}(\langle n^N, n^N, n^N\rangle) \leq t^N$$

And again using Theorem 2.12 we attain:

$$n^{\omega N} \leq (N+1)^2 t^N$$

where letting $N$ grow and taking the $N$-th roots achieves the desired result. ■

A lower bound on the border rank may be obtained by considering the dimensions of the spaces $U$, $V$, and $W$.

**Theorem 3.4** *Let the matrix product $\phi : U \times V \to W$ be a degeneration of order $q$ of $\phi'$ with border rank $\underline{R}$. Then,*

$$\underline{R} \geq \max\{dim(U), dim(V), dim(W)\}$$

*Proof.* We have that $\phi \leq \phi'$ and that the border rank of $\phi$ is $\underline{R}$. Raising $\phi$ to the $N$-th power, we find that the border rank of $\phi^N$ is $\underline{R}^N$. Using the same result as before, we have that the exact rank of $\phi^N$ is at most $(N+1)^2\underline{R}^N$. Since the rank of a bilinear map is greater than the maximum of the sizes of the dimensions, it must be the case that:

$$\max\{dim(U), dim(V), dim(W)\} \leq (N+1)^2\underline{R}^N$$

Letting $N$ tend to infinity and taking the $N$-th roots gives the desired result. ■

## 3.2 Additivity Conjecture for the Exact Rank

**Fact 3.5** *Border rank is non-additive:*

$$\underline{R}(\phi_1 + \phi_2) \neq \underline{R}(\phi_1) + \underline{R}(\phi_2)$$

Schönhage used the *Addivity Conjecture for the Exact Rank* to derive a better bound for $\omega$. The conjecture states:

$$R(\bigoplus_{i=1}^{N}\langle m_i, n_i, p_i\rangle) = \sum_{i=1}^{N} R(\langle m_i, n_i, p_i\rangle)$$

The conjecture is unproven, but it is useful for deriving a bound on $\omega$. It will later be shown how to derive the same bound without relying on the unproven conjecture.

**Theorem 3.6** *The exponent of matrix multiplication $\omega \leq 2.548$ if the additivity conjecture holds*

*Proof.* We begin by observing that exponentiating $(e,1,l) \otimes (1,h,1)$ by $N$ results in

$$\bigotimes_{s=0}^{N} \binom{N}{s} \otimes (e^s, h^{N-s}, l^s),$$

where the symbol $\otimes$ signifies that we are considering all distinct instances of the tensor product that coincide in the value of $s$. By harnessing the border rank properties as introduced, it follows that

$$R\left(\bigotimes_{s=0}^{N} \binom{N}{s} \otimes (e^s, h^{N-s}, l^s)\right) > (1+2N)^2(el+1)^N.$$

At this juncture, we utilize the additivity conjecture for the exact rank to deduce that

$$\sum_{s=0}^{N} \binom{N}{s} R\left((e^s, h^{N-s}, l^s)\right) \leq (1+2N)^2(el+1)^N.$$

Employing equation 1.5, we deduce

$$\sum_{s=0}^{N} \binom{N}{s} (el)^{s/3} h^{(N-s)/3} \leq (1+2N)^2(el+1)^N,$$

which, by extracting the $s$-th roots and considering the limit as $s$ approaches infinity, simplifies to

$$el^{w/3} + h^{w/3} \leq kn + 1.$$

Setting $e = l = 4$ (and consequently $h = 9$), we acquire the sought-after result. ■

## 3.3 Schönhage's Asymptotic Sum Inequality

As stated above, however, Schönhage found it possible to prove the same bound without relying on the additivity conjecture for the exact rank. We state and prove the theorem:

**Theorem 3.7 (Schönhage's Asymptotic Sum Inequality)** *Let $k, t$ be two positive integers, and let $m_1, \ldots, m_k, n_1, \ldots, n_k, p_1, \ldots, p_k$ be $3k$ positive integers. If*

$$R\left(\bigoplus_{i=1}^{k} (m_i, n_i, p_i)\right) \leq t$$

*then*

$$\sum_{i=1}^{k} (m_i \cdot n_i \cdot p_i)^{\omega/3} \leq t.$$

*Proof.* Note that for some $q \in \mathbb{N}$,

$$\bigoplus_{i=1}^{k} (m_i, n_i, p_i) \leq_q (t),$$

this when raised to the $N$th power, yields

$$\left(\bigoplus_{i=1}^{k} (m_i, n_i, p_i)\right)^N \leq_{(q-1)N+1} (t^N),$$

and we deduce

$$R\left(\bigoplus_{i=1}^{k} (m_i, n_i, p_i)^N\right) \leq ((q-1)N+1)^2 t^N.$$

Considering a vector $\mu = (\mu_1, \ldots, \mu_k)$ that satisfies $\sum_i \mu_i = N$,

$$R\left(\bigotimes_{\mu} \bigotimes_{i=1}^{k} (m_i, n_i, p_i)^{\mu_i}\right) \leq ((q-1)N+1)^2 t^N.$$

For every $\varepsilon > 0$, there is a constant $c_\varepsilon \in \mathbb{N}$ such that for all $n$,

$$R((n, n, n)) \leq c_\varepsilon n^{2+\varepsilon}.$$

We choose $P = \left(\binom{N}{\mu}\right)^{\frac{1}{2+\varepsilon}}$ so that

$$R((P, P, P)) \leq c_\varepsilon \left(\binom{N}{\mu}\right).$$

This leads to the conclusion that the multiplication

$$\bigotimes_{i=1}^{k}(P^{m_i\mu_i}, P^{n_i\mu_i}, P^{p_i\mu_i})$$

can be performed with fewer than $c_\varepsilon((q-1)N+1)^2t^N$ operations in $k$. For the "U" matrix, we view the elements as being products of $\bigotimes_{i=1}^{k} m_i^{n_i\mu_i}$ matrices, and similarly for the "V" and "W" matrices, the total number of $\bigotimes_{i=1}^{k}(m_i, n_i, p_i)^{\mu_i}$ matrix products is

$$\leq c_\varepsilon((q-1)N+1)^2t^N.$$

we see that

$$(P^3\bigotimes_{i=1}^{k}(m_in_ip_i)^{\mu_i})^{1/3} \leq c_\varepsilon((q-1)N+1)^2t^N.$$

After multiplying through by $\left(\binom{N}{\mu}\right)^{-\frac{\varepsilon}{2+\varepsilon}}$ and utilizing the facts that $\left(\binom{N}{\mu}\right) \leq k^N$ and that $a/2 \leq \lceil a \rceil$ to obtain

$$\left(\binom{N}{\mu}\bigotimes_{i=1}^{k}(m_in_ip_i)^{\mu_i}\right)^{1/3} \leq 2^\omega k^{\omega+\varepsilon}c_\varepsilon((q-1)N+1)^2t^N.$$

Summing over all feasible distributions of $\mu$, we conclude

$$\left(\sum_{i=1}^{k}(m_in_ip_i)^{\mu_i}\right)^{1/3} N \leq \left(\frac{N+k-1}{k-1}\right)^{\omega} k^{\omega+\varepsilon}c_\varepsilon((q-1)N+1)^2t^N.$$

Taking the $N$th roots and allowing $N$ to approach infinity, we derive the result that

$$\sum_{i=1}^{k}(m_in_ip_i)^{\omega/3} \leq s^\varepsilon t,$$

and as $\varepsilon$ approaches 0, we obtain the desired result. ■

We now show how Schönhage proved the bound on $\omega$. Consider the following tensor:

$$T_{\text{Scön}} = \sum_{i,j=1}^{3} a_i \otimes b_i \otimes c_i + \sum_{i,j=1}^{4} v_k \otimes v_k \otimes w$$

Note that the first part is isomorphic to $\langle 3,1,3\rangle$ and the second part is isomorphic to $\langle 1,4,1\rangle$. The first and second part do not share variables, so the sum is direct:

$$T_{\text{Schön}} \approx \langle 3,1,3\rangle + \langle 1,4,1\rangle$$

Since $\underline{R}(\langle 3,1,3\rangle) = 9$ and $\underline{R}(\langle 1,4,1\rangle) = 4$, we immediately obtain that: $\underline{R}(T_{\text{Schön}}) \leq 13$. Schönhage attained $\underline{R}(T_{\text{Schön}}) \leq 10$ by exhibiting the following decomposition:

$$\epsilon^2 T_{\text{Schon}} = T' + \epsilon^3 T''$$

for:

$$\begin{aligned}
T' = &(\alpha_1 + \epsilon u_1) \otimes (\beta_1 + \epsilon v_1) \otimes (w + \epsilon^2 c_{11}) \\
&+ (\alpha_1 + \epsilon u_2) \otimes (\beta_2 + \epsilon v_2) \otimes (w + \epsilon^2 c_{12}) \\
&+ (\alpha_2 + \epsilon u_3) \otimes (\beta_1 + \epsilon v_3) \otimes (w + \epsilon^2 c_{21}) \\
&+ (\alpha_2 + \epsilon u_4) \otimes (\beta_2 + \epsilon v_4) \otimes (w + \epsilon^2 c_{22}) \\
&+ (\alpha_3 - \epsilon u_1 - \epsilon u_3) \otimes \beta_1 \otimes (w + \epsilon^2 c_{31}) \\
&+ (\alpha_3 - \epsilon u_2 - \epsilon u_4) \otimes \beta_2 \otimes (w + \epsilon^2 c_{32}) \\
&+ \alpha_1 \otimes (\beta_3 - \epsilon v_1 - \epsilon v_2) \otimes (w + \epsilon^2 c_{13}) \\
&+ \alpha_2 \otimes (\beta_3 - \epsilon v_3 - \epsilon v_4) \otimes (w + \epsilon^2 c_{23}) \\
&+ \alpha_3 \otimes \beta_3 \otimes (w + \epsilon^2 c_{33}) \\
&- (\alpha_1 + \alpha_2 + \alpha_3) \otimes (\beta_1 + \beta_2 + \beta_3) \otimes w
\end{aligned}$$

and some tensor $T''$. Applying the asymptotic sum inequality to the $T_{\text{Schön}}$ gives:

$$9^{\omega/3} + 4^{\omega/3} \leq 10$$

which implies $\omega \leq 2.60$. ■

# 4 Laser Method

We will now show how the techniques developed so far, together with a new approach called the *laser method* can be used to obtain an upper bound of $\omega \leq 2.38$. This upper bound was obtained by Coppersmith and Winograd.

## 4.1 A First Construction

Let $q$ be a positive integer and consider three vector spaces $U$, $V$, and $W$ of dimension $q+1$ over some field $\mathbb{F}$. Takes a basis $\{x_0, ..., x_q\}$ of $U$, $\{y_0, ..., y_q\}$ of $V$, and $z_0, ..., z_q$ of $W$. Consider the following tensor:

$$T_{\text{easy}} = T^{011}_{\text{easy}} + T^{101}_{\text{easy}} + T^{110}_{\text{easy}}$$

where we have that:

$$\begin{aligned} T^{011}_{easy} &= \sum_{i=1}^{q} x_0 \otimes y_i \otimes z_i \approx \langle 1, 1, q \rangle, \\ T^{101}_{easy} &= \sum_{i=1}^{q} x_i \otimes y_0 \otimes z_i \approx \langle q, 1, 1 \rangle, \\ T^{110}_{easy} &= \sum_{i=1}^{q} x_i \otimes y_i \otimes z_0 \approx \langle 1, q, 1 \rangle. \end{aligned}$$

Observe:

$$\lambda^3 T_{\text{easy}} = T' + \lambda^4 T''$$

where:

$$\begin{aligned} T' = &\sum_{i=1}^{q} \lambda(x_0 + \lambda x_i) \otimes (y_0 + \lambda y_i) \otimes (z_0 + \lambda z_i) \\ &- \left( x_0 + \lambda^2 \sum_{i=1}^{q} x_i \right) \otimes \left( y_0 + \lambda^2 \sum_{i=1}^{q} y_i \right) \otimes \left( z_0 + \lambda^2 \sum_{i=1}^{q} z_i \right) \\ &+ (1 - q\lambda) x_0 \otimes y_0 \otimes z_0. \end{aligned}$$

and $T''$ is some tensor. So we have that $\underline{R}(T_{\text{easy}}) \leq q + 2$.

While the tensor $T_{\text{easy}}$ is the sum of three parts, the sum is not direct because the parts share variables. The key insight made by Coppersmith and Winograd was that we can consider several copies of $T_{\text{easy}}$ to obtain the following result:

**Theorem 4.1** *For $N$ large enough, the tensor $T^{\otimes N}_{easy}$ can be converted into a direct sum of:*

$$2^{(H(\frac{1}{3}, \frac{2}{3}) - o(1))N}$$

*terms, each ismorphic to*

$$\left[ T^{011}_{easy} \otimes \frac{N}{3} \right] \otimes \left[ T^{101}_{easy} \otimes \frac{N}{3} \right] \otimes \left[ T^{110}_{easy} \otimes \frac{N}{3} \right] \approx \left\langle q^{\frac{N}{3}}, q^{\frac{N}{3}}, q^{\frac{N}{3}} \right\rangle.$$

This gives us:

$$2^{\left(H\left(\frac{1}{3},\frac{2}{3}\right)-o(1)\right)N} \times q^{\frac{N\omega}{3}} \leq R\left(T_{easy}^{\otimes N}\right) \leq (q+2)^N$$

so:

$$2^{H\left(\frac{1}{3},\frac{2}{3}\right)} \times q^{\omega/3} \leq q+2$$

and thus:

$$\omega \leq 2.41$$

for $q = 8$.

## 4.2 A Second Construction

The second construction from Coppersmith and Winograd's paper yields $\omega \leq 2.387$. Let $q$ be a positive integer and consider three vector spaces $U$, $V$, and $W$ of dimension $q+2$ over some field $\mathbb{F}$. Takes a basis $\{x_0, ..., x_q\}$ of $U$, $\{y_0, ..., y_q\}$ of $V$, and $z_0, ..., z_q$ of $W$. Consider the following tensor:

$$T_{CW} = T_{CW}^{011} + T_{CW}^{101} + T_{CW}^{110} + T_{CW}^{002} + T_{CW}^{020} + T_{CW}^{200}$$

where we have:

$$\begin{aligned}
T_{CW}^{011} &= \sum_{i=1}^{q} x_0 \otimes y_i \otimes z_i \approx \{1, 1, q\}, \\
T_{CW}^{101} &= \sum_{i=1}^{q} x_i \otimes y_0 \otimes z_i \approx \{1, q, 1\}, \\
T_{CW}^{110} &= \sum_{i=1}^{q} x_i \otimes y_i \otimes z_0 \approx \{q, 1, 1\}, \\
T_{CW}^{002} &= x_0 \otimes y_0 \otimes z_{q+1} \approx \{1, 1, 1\}, \\
T_{CW}^{020} &= x_0 \otimes y_{q+1} \otimes z_0 \approx \{1, 1, 1\}, \\
T_{CW}^{200} &= x_{q+1} \otimes y_0 \otimes z_0 \approx \{1, 1, 1\}
\end{aligned}$$

Observe that:

$$\lambda^3 T_{CW} = T' + \lambda^4 T''$$

where we have:

$$\begin{aligned}
T' = &\sum_{i=1}^{q} \lambda(x_0 + \lambda x_i) \otimes (y_0 + \lambda y_i) \otimes (z_0 + \lambda z_i) \\
&- (x_0 + \lambda^2 \sum_{i=1}^{q} x_i) \otimes (y_0 + \lambda^2 \sum_{i=1}^{q} y_i) \otimes (z_0 + \lambda^2 \sum_{i=1}^{q} z_i) \\
&+ (1 - q\lambda)(x_0 + \lambda^3 x_{q+1}) \otimes (y_0 + \lambda^3 y_{q+1}) \otimes (z_0 + \lambda^3 z_{q+1}).
\end{aligned}$$

and $T''$ is some tensor. So we have $\underline{R}(T_{CW}) \leq q+2$. Just as in the last construction, even though $T'$ is a sum of six parts, the sum is not direct, so we cannot directly use the asymptotic sum inequality. Again, consider many copies of $T_{CW}$.

**Theorem 4.2** *For any $0 \leq \alpha \leq 1/3$ and for large enough $N$, the tensor $T_{CW}^{\otimes N}$ can be converted into a direct sum of:*

$$2^{(H(\frac{2}{3}-\alpha, 2\alpha, \frac{1}{3}-\alpha)-o(1))N}$$

*terms, each isomorphic to:*

$$\begin{aligned}
&\left[T_{CW}^{011} \otimes \alpha N\right] \otimes \left[T_{CW}^{101} \otimes \alpha N\right] \otimes \left[T_{CW}^{110} \otimes \alpha N\right] \\
&\quad \otimes \left[T_{CW}^{002} \otimes \left(\frac{1}{3} - \alpha\right) N\right] \otimes \left[T_{CW}^{020} \otimes \left(\frac{1}{3} - \alpha\right) N\right] \otimes \left[T_{CW}^{200} \otimes \left(\frac{1}{3} - \alpha\right) N\right] \\
&\quad \approx \langle \alpha N, \alpha N, \alpha N \rangle.
\end{aligned}$$

Theorem 4.2, via the asymptotic sum inequality, gives us:

$$2\left(H\left(\frac{2}{3}-\alpha, 2\alpha, \frac{1}{3}-\alpha\right)-o(1)\right) N \times \alpha N\omega \leq R\left(T_{CW}^{\otimes N}\right) \leq (q+2)^N.$$

which gives us:

$$2^{H\left(\frac{2}{3}-\alpha, 2\alpha, \frac{1}{3}-\alpha\right)} \times q^{\alpha\omega} \leq q+2,$$

which yields:

$$\omega \leq 2.38719$$

for $q = 6$ and $\alpha = 0.3173$.

# 5 Further Work

Since Coppersmith and Winograd achieved a bound of $\omega \leq 2.3755$ in 1990, we have not had many substantial improvements in upper bounds for $\omega$ in the last thirty years. Notably, the best known upper bound is $\omega \leq 2.371552$. Here is a summary of results from the past few decades: Interestingly, it was shown that the laser method could not be used to show $\omega \leq 2.3725$, but Duan, Wu, and Zhou used a new approach to the laser method that addressed what is called *combination loss*. They then showed that addressing the laser method with combination loss could not show a bound better than $\omega \leq 2.3078$.

| **Year** | **Bound on omega** | **Authors** |
|---|---|---|
| 1969 | 2.8074 | Strassen |
| 1978 | 2.796 | Pan |
| 1979 | 2.780 | Bini, Capovani, Romani |
| 1981 | 2.522 | Schönhage |
| 1981 | 2.517 | Romani |
| 1981 | 2.496 | Coppersmith, Winograd |
| 1986 | 2.479 | Strassen |
| 1990 | 2.3755 | Coppersmith, Winograd |
| 2010 | 2.3737 | Stothers |
| 2013 | 2.3729 | Williams |
| 2014 | 2.3728639 | Le Gall |
| 2020 | 2.3728596 | Alman, Williams |
| 2022 | 2.371866 | Duan, Wu, Zhou |
| 2023 | 2.371552 | Williams, Xu, Xu, and Zhou |

Table 1: Timeline of matrix multiplication exponent

Other researchers put the matrix multiplication problem in a group theoretic context. Cohn, Kleinberg, Szegedy, and Umans utiilized triples of subsets of finite groups which satisfy a disjointness property called the triple product property (TPP). They also give conjectures that, if true, would imply that there are matrix multiplication algorithms with essentially quadratic complexity.

Perhaps even more cutting edge, Google DeepMind developed a method to automatically find new matrix multiplication algorithms through reinforcement learning. Alpha Tensor, as it has been coined, is a system able to autonomously search for provably correct multiplication algorithms. The RL-agent plays **TensorGame**: Given any tensor $T$ we want to find a decomposition of $T$ as a sum of $R$ outer products with $R$, which corresponds to the number of multiplications in the algorithm. This method has been used to find surprisingly practical algorithms for matrix multiplication that achieve lower constants than the work we've discussed in this survey.

# 6 Acknowledgements

We wanted to thank Eshan and Mohit for giving us the inspiration for this project, as well as for providing resources to us in our initial stages of research. More broadly, we want to thank them for a wonderful

experience in CS 6810 this semester - we truly enjoyed the course this semester and thoroughly appreciate your contributions to our learning!

# 7 References

- Strassen, Volker (1969). "Gaussian Elimination is not Optimal". Numer. Math. 13 (4): 354–356. doi:10.1007/BF02165411. S2CID 121656251.
- Dario Andrea Bini; Milvio Capovani; Francesco Romani; Grazia Lotti (Jun 1979). "$O(n^{2.7799})$ complexity for $n \times n$ approximate matrix multiplication". Information Processing Letters. 8 (5): 234–235. doi:10.1016/0020-0190(79)90113-3.
- A. Schönhage (1981). "Partial and total matrix multiplication". SIAM Journal on Computing. 10 (3): 434–455. doi:10.1137/0210032.
- D. Coppersmith; S. Winograd (Mar 1990). "Matrix multiplication via arithmetic progressions". Journal of Symbolic Computation. 9 (3): 251–280. doi:10.1016/S0747-7171(08)80013-2
- Stothers. "On the Complexity of Matrix Multiplication". https://www.maths.ed.ac.uk/sites/default/files/atoms/fil
- Le Gall. "Algebraic Complexity Theory and Matrix Multiplication". http://francoislegall.com/LeGallISSAC14-tutorial-handout.pdf