Designing Strassen's Algorithm For Matrix Multiplication

Joint w/ Cris Moore
arXiv:1708.09398, 1612.01527

LA Combinatorics & Complexity Seminar
November, 2020

Joshua A. Grochow

University of Colorado **Boulder**

Multiplying matrices

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nn} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{pmatrix}$$

n multiplications and $n-1$ additions per dot product

n^2 dot products

$\Rightarrow O(n^3)$ steps

Multiplying matrices

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nn} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{pmatrix}$$

n multiplications and $n-1$ additions per dot product
n^2 dot products
$\Rightarrow O(n^3)$ steps

Theorem [Klyuyev & Kokovkin-Scherbak '65]: Optimal if only allowed to work on rows and columns as a whole.

Theorem [Strassen '69]: Can do better! $O\left(n^{\log_2 7}\right) = O(n^{2.81})$.

Multiplying matrices

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nn} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{pmatrix}$$

Used in scientific computing*, graphics, GPUs, combinatorial algorithms, other algebraic algorithms, deep learning

Matrix multiplication can even be the bottleneck

Multiplying matrices in practice

In practice

- Often have sparse or structured matrices → better algorithms available
- Even with dense, unstructured matrices, memory and communication are more frequently the bottleneck than number of operations

Multiplying matrices in practice

In practice

- Often have sparse or structured matrices → better algorithms available
- Even with dense, unstructured matrices, memory and communication are more frequently the bottleneck than number of operations
- Strassen's algorithm actually used in practice
- Other algorithms today aren't, but future ones could be!

Multiplying matrices

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nn} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{pmatrix}$$

Used in scientific computing*, graphics, GPUs, combinatorial algorithms, other algebraic algorithms, deep learning

Matrix multiplication can even be the bottleneck

More importantly: gives us insight into the nature of computing & complexity!

Multiplying matrices

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nn} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{pmatrix}$$

Theorem [Strassen '69]: Can do better! $O\left(n^{\log_2 7}\right) = O(n^{2.81})$.

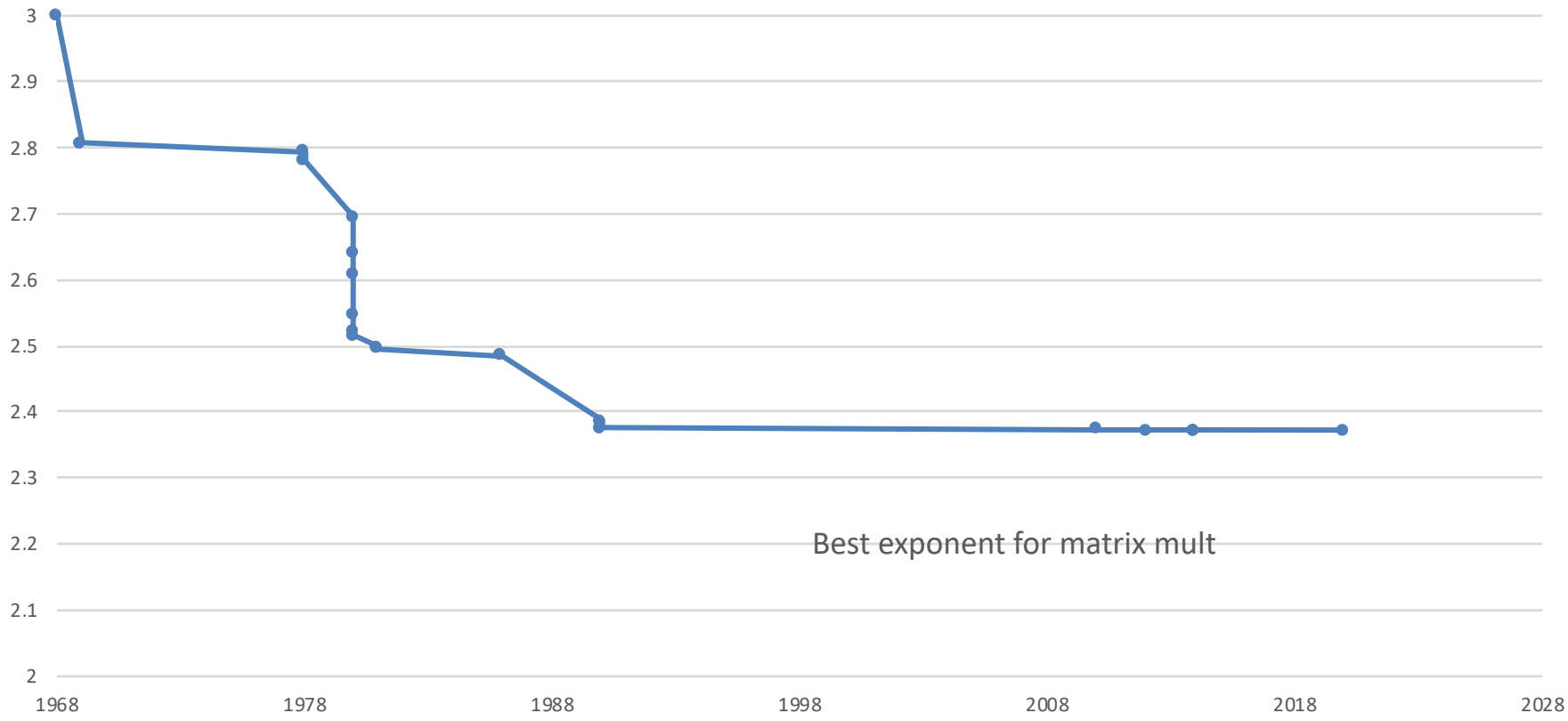
Definition: The exponent of matrix multiplication is

$$\omega = \inf\{e : MM_n \; in \; O(n^{e+\varepsilon}) \forall \varepsilon > 0\}$$

Conjecture (folklore): $\omega = 2$.

In principle, ω depends on the characteristic.

Improvements (in theory)



Current record [Alman-Williams '20]: $\omega < 2.372859$

Strassen's Algorithm I: A magical 2x2 trick

Ordinary 2x2 product: 8 products, 4 sums

$$I = (a_{11}+a_{22})(b_{11} + b_{22}) \qquad II = (a_{21} + a_{22})(b_{11})$$

$$III = (a_{11})(b_{12} - b_{22}) \qquad IV = (a_{22})(-b_{11} + b_{21})$$

$$VI = (-a_{11} + a_{21})(b_{11} + b_{12}) \qquad V = (a_{11} + a_{12})(b_{22})$$

$$VII = (a_{12} - a_{22})(b_{21} + b_{22})$$

$$c_{11} = I + IV - V + VII \qquad c_{12} = III + V$$

$$c_{21} = II + IV \qquad c_{22} = I + III - II + VI$$

Strassen 2x2 product: **7** products, 18 sums

Strassen's Algorithm II: Recurse

Works correctly even if entries are from a noncommutative ring

Suppose the entries are from $M_n(\mathbb{C})$

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} b_{11} & a_{12} \\ b_{21} & a_{22} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}$$

Strassen's Algorithm II: Recurse

Works correctly even if entries are from a noncommutative ring

Suppose the entries are from $M_n(\mathbb{C})$

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} b_{11} & a_{12} \\ b_{21} & a_{22} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}$$

$$\begin{pmatrix} \begin{pmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{n1} & \alpha_{n2} & \cdots & \alpha_{nn} \end{pmatrix} & \begin{pmatrix} \beta_{11} & \beta_{12} & \cdots & \beta_{1n} \\ \beta_{21} & \beta_{22} & \cdots & \beta_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{n1} & \beta_{n2} & \cdots & \beta_{nn} \end{pmatrix} \\ \begin{pmatrix} \gamma_{11} & \gamma_{12} & \cdots & \gamma_{1n} \\ \gamma_{21} & \gamma_{22} & \cdots & \gamma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{n1} & \gamma_{n2} & \cdots & \gamma_{nn} \end{pmatrix} & \begin{pmatrix} \delta_{11} & \delta_{12} & \cdots & \delta_{1n} \\ \delta_{21} & \delta_{22} & \cdots & \delta_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \delta_{n1} & \delta_{n2} & \cdots & \delta_{nn} \end{pmatrix} \end{pmatrix}$$

$$M_2\big(M_n(\mathbb{C})\big) \cong M_{2n}(\mathbb{C})$$

Strassen's Algorithm II: Recurse

Works correctly even if entries are from a noncommutative ring

Suppose the entries are from $M_n(\mathbb{C})$

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} b_{11} & a_{12} \\ b_{21} & a_{22} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}$$

$$M_2\big(M_n(\mathbb{C})\big) \cong M_{2n}(\mathbb{C})$$

Multiply $2n \times 2n$ matrices using 7 $n \times n$ products, and 18 $n \times n$ additions

Let $R(n) =$ # products ("rank"), $A(n) =$# additions to mult two $n \times n$ matrices

$$R(2n) \leq 7R(n) \qquad\qquad R(n) \leq n^{\log_2 7}$$

$$A(2n) \leq 18n^2 + 7A(n) \quad \rightarrow \quad A(n) \leq 6\left(7n^{\log_2 7} - 4n^2\right) = \Theta(n^{\log_2 7})$$

Strassen's Algorithm II: Origin story

How did Strassen find this?

Trying to prove it couldn't be done mod 2, by hand, (clever) brute force.

Found solution mod 2.

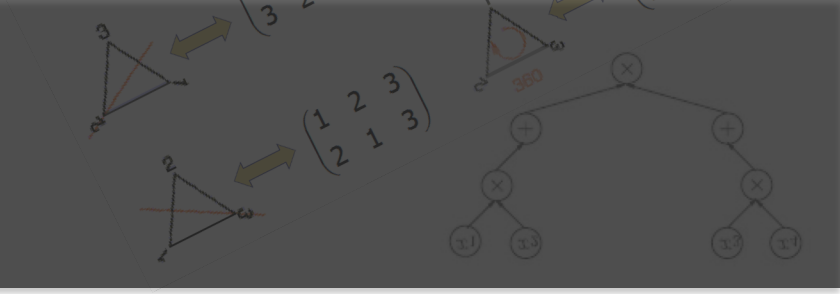
Figured out signs to work over $\mathbb{Z}$.

Mystery:
Math behind the magic?

Where does the magical 2×2 trick

“really” come from?

Origin of
The math behind the magic



Matrix multiplication is characterized by its symmetries (important in Geometric Complexity Theory approach to P versus NP).

Let $\mu: M_n \otimes M_n \to M_n$ be matrix multiplication. Then:

$$\mu(A,B) = X^{-1}\mu(XAY^{-1}, YBZ^{-1})Z$$

for all $X, Y, Z \in GL_n$.

Any bilinear map $f: M_n \otimes M_n \to M_n$ such that

$$f(A,B) = X^{-1}f(XAY^{-1}, YBZ^{-1})Z$$

for all $X, Y, Z \in GL_n$ is a scalar multiple of μ.

Our starting point: For V an irrep of G, write $M_n = V \otimes V^*$, then invariance under G^3 (rather than GL_n^3) suffices!

Proof

First rephrase matrix multiplication more symmetrically

It is a bilinear map $M_n \otimes M_n \to M_n$, so we can treat it as an element of $M_n \otimes M_n \otimes M_n^*$, namely

$$T_{MM} = \sum_{ijk} E_{ij} \otimes E_{jk} \otimes E_{ki}$$

$$\langle T_{MM} | A \otimes B \otimes C \rangle = tr(ABC)$$

Note $(AB)_{ik} = tr(ABE_{ki})$

Rephrasing Matrix Multiplication Symmetrically

$$\sum_{ijk} E_{ij} \otimes E_{jk} \otimes E_{ki}$$

If we write $M_n \cong U \otimes V^*$, then this lives in

$$M_n \otimes M_n \otimes M_n^*$$
$$\cong (U \otimes V^*) \otimes (V \otimes W^*) \otimes (W \otimes U^*)$$
$$\cong U \otimes (V \otimes V^*) \otimes (W^* \otimes W) \otimes U^*$$

In this decomposition, MM is

$$Id_U \otimes Id_V \otimes Id_W$$

Rephrasing Matrix Multiplication Symmetrically

$$\sum_{ijk} E_{ij} \otimes E_{jk} \otimes E_{ki}$$

If we write $M_n \cong U \otimes V^*$, then this lives in

$$M_n \otimes M_n \otimes M_n^*$$
$$\cong (U \otimes V^*) \otimes (V \otimes W^*) \otimes (W \otimes U^*)$$
$$\cong U \otimes (V^* \otimes V) \otimes (W^* \otimes W) \otimes U^*$$

In this decomposition, MM is

$$Id_U \otimes Id_V \otimes Id_W$$

Rephrasing Matrix Multiplication Symmetrically

$$\sum_{ijk} E_{ij} \otimes E_{jk} \otimes E_{ki}$$

If we write $M_n \cong U \otimes V^*$, then this lives in

$$M_n \otimes M_n \otimes M_n^*$$
$$\cong (U \otimes V^*) \otimes (V \otimes W^*) \otimes (W \otimes U^*)$$
$$\cong U \otimes (V^* \otimes V) \otimes (W^* \otimes W) \otimes U^*$$

In this decomposition, MM is

$$Id_U \otimes Id_V \otimes Id_W$$

Rephrasing Matrix Multiplication Symmetrically

$$\sum_{ijk} E_{ij} \otimes E_{jk} \otimes E_{ki}$$

If we write $M_n \cong U \otimes V^*$, then this lives in

$$M_n \otimes M_n \otimes M_n^*$$
$$\cong (U \otimes V^*) \otimes (V \otimes W^*) \otimes (W \otimes U^*)$$
$$\cong U \otimes (V^* \otimes V) \otimes (W^* \otimes W) \otimes U^*$$

In this decomposition, MM is

$$Id_U \otimes Id_V \otimes Id_W$$

Proof
Using group orbits

G: Finite group, acting irreducibly on $\mathbb{C}^n$.
S: G-orbit of in $\mathbb{C}^n$. Then (Schur's Lemma)

$$\frac{n}{|S|}\sum_{v\in S}|v\rangle\langle v| = Id$$

Then we have

$$\frac{n^3}{|S|^3}\sum_{u,v,w\in S}|u\rangle\langle u|\otimes|v\rangle\langle v|\otimes|w\rangle\langle w| = Id\otimes Id\otimes Id$$

$$\frac{n^3}{|S|^3}\sum_{u,v,w\in S}|u\rangle\langle v|\otimes|v\rangle\langle w|\otimes|w\rangle\langle u| = MM$$

Proof

Suppose $S \subseteq \mathbb{C}^n$ satisfies

$$\frac{1}{|S|}\sum_{v\in S}|v\rangle\langle v| = \frac{1}{n}Id$$

Then we have

$$\frac{n^3}{|S|^3}\sum_{u,v,w\in S}|u\rangle\langle u|\otimes|v\rangle\langle v|\otimes|w\rangle\langle w| = Id\otimes Id\otimes Id$$

$$\frac{n^3}{|S|^3}\sum_{u,v,w\in S}|u\rangle\langle v|\otimes|v\rangle\langle w|\otimes|w\rangle\langle u| = MM$$

Proof

Suppose $S \subseteq \mathbb{C}^n$ satisfies

$$\frac{1}{|S|}\sum_{v\in S}|v\rangle\langle v| = \frac{1}{n}Id$$

Then we have

$$\frac{n^3}{|S|^3}\sum_{u,v,w\in S}|u\rangle\langle u|\otimes|v\rangle\langle v|\otimes|w\rangle\langle w| = Id\otimes Id\otimes Id$$

$$\frac{n^3}{|S|^3}\underbrace{\sum_{u,v,w\in S}}_{|S|^3 \text{ summands}}|u\rangle\langle v|\otimes|v\rangle\langle w|\otimes|w\rangle\langle u| = MM$$

Proof

Consider

$$\sum_{u,v,w \in S} |u\rangle\langle v-u| \otimes |v\rangle\langle w-v| \otimes |w\rangle\langle u-w|$$

Proof

Consider

$$\sum_{u,v,w \in S} |u\rangle\langle v-u| \otimes |v\rangle\langle w-v| \otimes |w\rangle\langle u-w|$$

Why consider this?

- Looks kinda like matrix multiplication. Almost G^3-invariant.
- Only need to sum of triples of *distinct* $u, v, w \in S$. $\rightarrow |S|(|S|-1)(|S|-2)$ summands < $|S|^3$

Proof

$$\sum_{u,v,w \in S} |u\rangle\langle v - u| \otimes |v\rangle\langle w - v| \otimes |w\rangle\langle u - w|$$

Expand out, giving 4 kinds of terms:

1. Matched $-|u\rangle\langle u| \otimes |v\rangle\langle v| \otimes |w\rangle\langle w|$
2. 1 mismatch $|u\rangle\langle v| \otimes |v\rangle\langle v| \otimes |w\rangle\langle w|$
3. 2 mismatches $|u\rangle\langle v| \otimes |v\rangle\langle w| \otimes |w\rangle\langle w|$
4. 3 mismatches $|u\rangle\langle v| \otimes |v\rangle\langle w| \otimes |w\rangle\langle u|$

Proof

$$\sum_{u,v,w \in S} |u\rangle\langle v - u| \otimes |v\rangle\langle w - v| \otimes |w\rangle\langle u - w|$$

Expand out, giving 4 kinds of terms:

1. Matched $-|u\rangle\langle u| \otimes |v\rangle\langle v| \otimes |w\rangle\langle w|$
2. 1 mismatch $|u\rangle\langle v| \otimes |v\rangle\langle v| \otimes |w\rangle\langle w|$
3. 2 mismatches $|u\rangle\langle v| \otimes |v\rangle\langle w| \otimes |w\rangle\langle w|$
4. 3 mismatches $|u\rangle\langle v| \otimes |v\rangle\langle w| \otimes |w\rangle\langle u|$

Proof

$$\sum_{u,v,w \in S} |u\rangle\langle v - u| \otimes |v\rangle\langle w - v| \otimes |w\rangle\langle u - w|$$

Expand out, giving 4 kinds of terms:

1. Matched $-|u\rangle\langle u| \otimes |v\rangle\langle v| \otimes |w\rangle\langle w|$
2. 1 mismatch $|u\rangle\langle v| \otimes |v\rangle\langle v| \otimes |w\rangle\langle w|$
3. 2 mismatches $|u\rangle\langle v| \otimes |v\rangle\langle w| \otimes |w\rangle\langle w|$
4. 3 mismatches $|u\rangle\langle v| \otimes |v\rangle\langle w| \otimes |w\rangle\langle u|$

Proof

$$\sum_{u,v,w \in S} |u\rangle\langle v - u| \otimes |v\rangle\langle w - v| \otimes |w\rangle\langle u - w|$$

Expand out, giving 4 kinds of terms:

1. Matched $-|u\rangle\langle u| \otimes |v\rangle\langle v| \otimes |w\rangle\langle w|$
2. 1 mismatch $|u\rangle\langle v| \otimes |v\rangle\langle v| \otimes |w\rangle\langle w|$
3. 2 mismatches $|u\rangle\langle v| \otimes |v\rangle\langle w| \otimes |w\rangle\langle w|$
4. 3 mismatches $|u\rangle\langle v| \otimes |v\rangle\langle w| \otimes |w\rangle\langle u|$

Proof

$$\sum_{u,v,w \in S} |u\rangle\langle v-u| \otimes |v\rangle\langle w-v| \otimes |w\rangle\langle u-w|$$

Expand out, giving 4 kinds of terms:

1. Matched $-|u\rangle\langle u| \otimes |v\rangle\langle v| \otimes |w\rangle\langle w|$ $\rightarrow Id^{\otimes 3}$
2. 1 mismatch $|u\rangle\langle v| \otimes |v\rangle\langle v| \otimes |w\rangle\langle w|$
3. 2 mismatches $|u\rangle\langle v| \otimes |v\rangle\langle w| \otimes |w\rangle\langle w|$
4. 3 mismatches $|u\rangle\langle v| \otimes |v\rangle\langle w| \otimes |w\rangle\langle u|$

Proof

$$\sum_{u,v,w \in S} |u\rangle\langle v-u| \otimes |v\rangle\langle w-v| \otimes |w\rangle\langle u-w|$$

Expand out, giving 4 kinds of terms:

1. Matched $-|u\rangle\langle u| \otimes |v\rangle\langle v| \otimes |w\rangle\langle w|$ $\rightarrow Id^{\otimes 3}$
2. 1 mismatch $|u\rangle\langle v| \otimes |v\rangle\langle v| \otimes |w\rangle\langle w|$
3. 2 mismatches $|u\rangle\langle v| \otimes |v\rangle\langle w| \otimes |w\rangle\langle w|$
4. 3 mismatches $|u\rangle\langle v| \otimes |v\rangle\langle w| \otimes |w\rangle\langle u|$ $\rightarrow$MM

Proof

$$\sum_{u,v,w \in S} |u\rangle\langle v-u| \otimes |v\rangle\langle w-v| \otimes |w\rangle\langle u-w|$$

Expand out, giving 4 kinds of terms:

1. Matched $-|u\rangle\langle u| \otimes |v\rangle\langle v| \otimes |w\rangle\langle w|$ $\rightarrow Id^{\otimes 3}$
2. 1 mismatch $|u\rangle\langle v| \otimes |v\rangle\langle v| \otimes |w\rangle\langle w|$ $\rightarrow$0?
3. 2 mismatches $|u\rangle\langle v| \otimes |v\rangle\langle w| \otimes |w\rangle\langle w|$ $\rightarrow$0?
4. 3 mismatches $|u\rangle\langle v| \otimes |v\rangle\langle w| \otimes |w\rangle\langle u|$ $\rightarrow$MM

Proof

$$\sum_{u,v,w \in S} |u\rangle\langle v-u| \otimes |v\rangle\langle w-v| \otimes |w\rangle\langle u-w|$$

Expand out, giving 4 kinds of terms:

1. Matched $-|u\rangle\langle u| \otimes |v\rangle\langle v| \otimes |w\rangle\langle w|$ $\rightarrow Id^{\otimes 3}$
2. 1 mismatch $|u\rangle\langle v| \otimes |v\rangle\langle v| \otimes |w\rangle\langle w|$ $\rightarrow$0?
3. 2 mismatches $|u\rangle\langle v| \otimes |v\rangle\langle w| \otimes |w\rangle\langle w|$ $\rightarrow$0?
4. 3 mismatches $|u\rangle\langle v| \otimes |v\rangle\langle w| \otimes |w\rangle\langle u|$ $\rightarrow$MM

Proof

If

$$\sum_{u \in S} u = 0 \qquad (*)$$

Then those terms vanish, as desired.

Given $u \in \mathbb{C}^n$, (*) is its projection onto trivial rep.

→If $\mathbb{C}^n$ is a *nontrivial irrep*, the sum must be 0

Proof

If S is a G-orbit in a nontrivial irrep,

then

$$MM = Id^{\otimes 3} + \frac{n^3}{|S|^3} \sum_{u,v,w \in S} |u\rangle\langle v-u| \otimes |v\rangle\langle w-v| \otimes |w\rangle\langle u-w|$$

Proof

If

Unitary 2-Design!

$$\sum_{u \in S} u = 0$$

$$\frac{1}{|S|} \sum_{v \in S} |v\rangle\langle v| = \frac{1}{n} id$$

Then

$$Id^{\otimes 3} + \frac{n^3}{|S|^3} \sum_{u,v,w \in S} |u\rangle\langle v - u| \otimes |v\rangle\langle w - v| \otimes |w\rangle\langle u - w|$$

$$= MM$$

The math behind the magic

Theorem [G.-Moore]: For $S \subseteq \mathbb{C}^n$ a unitary 2-design, $s = |S|$, $n \times n$ matrices can be multiplied using at most $s(s-1)(s-2)+1$ multiplications.

The math behind the magic

Theorem [G.-Moore]: For $S \subseteq \mathbb{C}^n$ a unitary 2-design, $s = |S|$, $n \times n$ matrices can be multiplied using at most $s(s-1)(s-2)+1$ multiplications.

Observe: If G finite group, V nontrivial irrep, $v \in V$ w/ $|v| = 1$, then the orbit Gv is a unitary 2-design.

The math behind the magic

Theorem [G.-Moore]: For $S \subseteq \mathbb{C}^n$ a unitary 2-design, $s = |S|$, $n \times n$ matrices can be multiplied using at most $s(s-1)(s-2)+1$ multiplications.

The action of S_3 on $\mathbb{C}^2$ has an orbit of size 3 (equilaterial triangle).

Corollary [G.-Moore]: $3(3-1)(3-2)+1 = 7$.

The math behind the magic

Theorem [G.-Moore]: For $S \subseteq \mathbb{C}^n$ a unitary 2-design, $s = |S|$, $n \times n$ matrices can be multiplied using at most $s(s-1)(s-2)+1$ multiplications.

The action of S_{n+1} on $\mathbb{C}^n$ has an orbit of size n+1. Smallest possible, since must sum to zero.

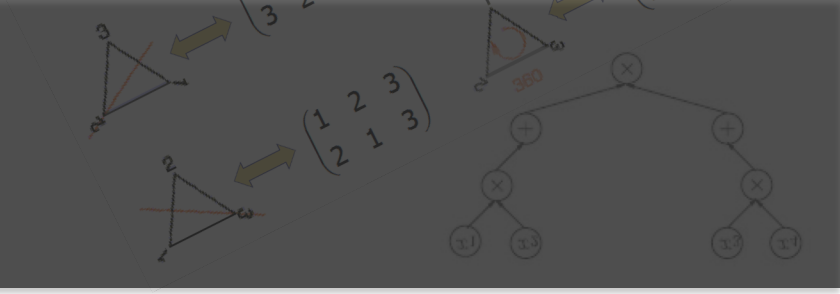
Corollary [G.-Moore]: $(n+1)n(n-1)+1 = n^3 - n + 1 < n^3$.

Summary of
The math behind the magic

Okay, but is that the same as Strassen's algorithm?

- Yes, by a theorem of de Groote '78
- But…we don't care! It gives a conceptual explanation of the upper bound of 7.

Summary of
The math behind the magic



Gave conceptual explanation of Strassen's 7.

Open: Find a similar explanation that works over arbitrary rings (as Strassen's algorithm does; ours needs ½, $1/3$, and $\sqrt{3}$).

Further Ideas:

- Use of unitary d-designs for $d > 2$?
- Other symmetric algorithms? See also Burichenko; Chiantini-Ikenmeyer-Landsberg-Ottaviani.

Other explanations

Gastinel 1971
Yuval 1978
Chatelin 1986
Clausen 1988
Alekseyev 1997
Gates & Kreinovich 2000
Paterson 2009
Minz 2015
Chiantini, Ikenmeyer, Landsberg, Ottaviani 2016
Ikenmeyer & Lysikov 2017

Uses/reveals symmetries

EXTRA SLIDES

Strassen's Algorithm III: The math behind the magic

Observe: If G finite group, V nontrivial irrep, $v \in V$ w/ $|v| = 1$, then the orbit Gv is a unitary 2-design.

Proof:

1. $\sum_{g \in G} gv$ is the projection onto the trivial representation, so must be 0.
2. $\varphi = \sum_{g \in G} |gv\rangle\langle gv|$ is an G-endomorphism of V, so must be scalar by Schur's Lemma. *QED*

Interlude on Recursion

Recursive algorithms

Strassen's recursion shows:

Using m products to multiply $n_0 \times n_0$ matrices

$\rightarrow$

Use $O(n^{\log_{n_0} m})$ operations to multiply $n \times n$ matrices ($n \rightarrow \infty$)

Recursive algorithms

n	Lower bound on # mults	Upper bound	Needed to beat records
2	7 [HK71, Win71]	7 [Str69]	-
3	19 [Blä03]	23 [Lad76, JM86, CBH11]	-
4	34 [Blä03]	49 [Str69]	-
8	136 [Blä03]	7^3 [Str69]	138 would beat Le Gall '14
16	592 [Blä03]	7^4 [Str69]	600 would beat 2.3078 (cf. [AFLG15])

Theorem [Bläser '03]: Needs at least $\frac{5}{2}n^2 - 3n$.

Current best lower bound (asymptotically):

Theorem [Landsberg '13]: Needs $\geq 3n^2 - o(n^2)$. (Stronger when $n > 84$.)