Strassen's algorithm is not optimaly accurate

Jean-Guillaume Dumas Clément Pernet Alexandre Sedoglavic

ISSAC, Raleigh, NC, USA.
July 17, 2024

Université Grenoble Alpes, Grenoble INP – UGA, Université de Lille, France

Strassen's algorithm and a long lasting defiance

2×2 matrix multiplication	$\#$ Multiplications	$\#$ Additions
Conventional	8	4
[Strassen'69]	7	18
[Winograd'70]	7	15

Strassen's algorithm and a long lasting defiance

2×2 matrix multiplication	# Multiplications	# Additions	Divide & Conquer
Conventional	8	4	$2n^3$
[Strassen'69]	7	18	$7n^{2.8074}$
[Winograd'70]	7	15	$6n^{2.8074}$

Strassen's algorithm and a long lasting defiance

2×2 matrix multiplication	$\#$ Multiplications	$\#$ Additions	Divide & Conquer
Conventional	8	4	$2n^3$
[Strassen'69]	7	18	$7n^{2.8074}$
[Winograd'70]	7	15	$6n^{2.8074}$

Speed ? improving exponent while degrading leading constant $\rightsquigarrow$ only for large n

Strassen's algorithm and a long lasting defiance

2×2 matrix multiplication	# Multiplications	# Additions	Divide & Conquer
Conventional	8	4	$2n^3$
[Strassen'69]	7	18	$7n^{2.8074}$
[Winograd'70]	7	15	$6n^{2.8074}$

Speed ? improving exponent while degrading leading constant $\leadsto$ only for large n

[Knuth, The Art of Computer Programming vol.2]

advantageous for $n > 20$, and saved 18 percent when $n = 100$. He estimated that Strassen's scheme (36) would not begin to excel over (35) until $n \approx 250$; and such enormous matrices rarely occur in practice unless they are very sparse, when other techniques apply. Furthermore, the known methods of order n^ω

Strassen's algorithm and a long lasting defiance

2×2 matrix multiplication	# Multiplications	# Additions	Divide & Conquer
Conventional	8	4	$2n^3$
[Strassen'69]	7	18	$7n^{2.8074}$
[Winograd'70]	7	15	$6n^{2.8074}$

Speed ? improving exponent while degrading leading constant $\rightsquigarrow$ only for large n

Unstability: aggregation-cancelation, coefficient growth $\rightsquigarrow$ numerically unstable

Strassen's algorithm and a long lasting defiance

2×2 matrix multiplication	# Multiplications	# Additions	Divide & Conquer
Conventional	8	4	$2n^3$
[Strassen'69]	7	18	$7n^{2.8074}$
[Winograd'70]	7	15	$6n^{2.8074}$

Speed ? improving exponent while degrading leading constant ⇝ only for large n

Unstability: aggregation-cancelation, coefficient growth ⇝ numerically unstable

[ATLAS BLAS mailing list]

```
>>And anybody knows about other implementations as MKL, ACML or Goto(Open)BLAS?
>
> They should all be using the standard algorithm.  Strassen (and all the other
> fast matmuls) are illegal in standard libraries because they are not as
> numerically stable.  There are some high performance libraries that optionally
> provide fast/unstable multiplies (in particular 3-M for complex), but they
> aren't supposed to do so by default.
```

Strassen's algorithm and a long lasting defiance

[Huss-Lederman, Jacobson, Johnson, Tsao, Turnbull'96]

Strassen's algorithm has long suffered from the erroneous assumptions that it is not efficient for matrix sizes that are seen in practice and that it is unstable. Both of these assumptions have been questioned in recent work. By stopping the Strassen recursions early

Strassen's algorithm and a long lasting defiance

[Cormen-Leiserson-Rivest-Stein, Introduction to Algorithms]

From a practical point of view, Strassen's algorithm is often not the method of choice for matrix multiplication, for four reasons:

1. The constant factor hidden in the $\Theta(n^{\lg 7})$ running time of Strassen's algorithm is larger than the constant factor in the $\Theta(n^3)$-time SQUARE-MATRIX-MULTIPLY procedure.
2. When the matrices are sparse, methods tailored for sparse matrices are faster.
3. Strassen's algorithm is not quite as numerically stable as SQUARE-MATRIX-MULTIPLY. In other words, because of the limited precision of computer arithmetic on noninteger values, larger errors accumulate in Strassen's algorithm than in SQUARE-MATRIX-MULTIPLY.
4. The submatrices formed at the levels of recursion consume space.

Strassen's algorithm and a long lasting defiance

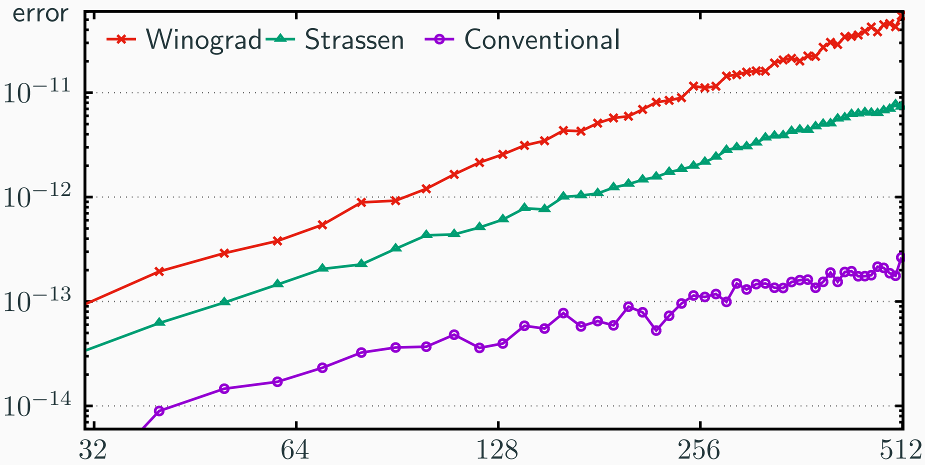
[Cormen-Leiserson-Rivest-Stein, Introduction to Algorithms]

From a practical point of view, Strassen's algorithm is often not the method of choice for matrix multiplication, for four reasons:

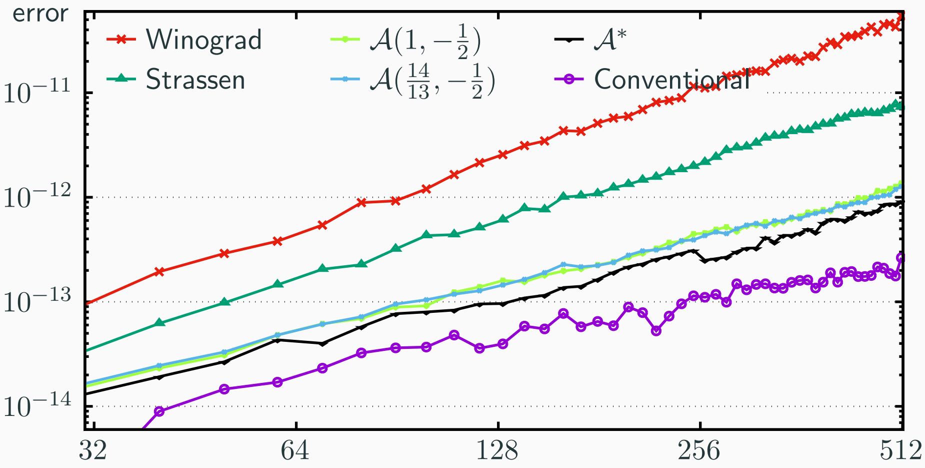
1. The constant factor hidden in the $\Theta(n^{\lg 7})$ running time of Strassen's algorithm is larger than the constant factor in the $\Theta(n^3)$-time SQUARE-MATRIX-MULTIPLY procedure.
2. When the matrices are sparse, methods tailored for sparse matrices are faster.
3. Strassen's algorithm is not quite as numerically stable as SQUARE-MATRIX-MULTIPLY. In other words, because of the limited precision of computer arithmetic on noninteger values, larger errors accumulate in Strassen's algorithm than in SQUARE-MATRIX-MULTIPLY.
4. The submatrices formed at the levels of recursion consume space.

1. [Schwarz et al.] and [here] $\rightsquigarrow 5n^{\log_2 7}$
3. [Brent 70], [Bini Lotti'80], [Demmel'93], [Higham'02], [Demmel et al.'07] ... and [here]
4. [Dumas Grenet'24] in previous talk

Accuracy of recursive 2×2 matrix multiplication algorithms



Contribution: new algorithms with improved accuracy and leading constant



Fast Matrix Multiplication algorithms and their representation

Fast Matrix Multiplication algorithms and their representation

Bilinear program

$$\rho_1 \leftarrow a_{11} \cdot b_{11}$$
$$\rho_2 \leftarrow a_{12} \cdot b_{21}$$
$$\rho_3 \leftarrow (-a_{11} - a_{12} + a_{21} + a_{22}) \cdot b_{22}$$
$$\rho_4 \leftarrow a_{22} \cdot (-b_{11} + b_{12} + b_{21} - b_{22})$$
$$\rho_5 \leftarrow (a_{21} + a_{22}) \cdot (-b_{11} + b_{12})$$
$$\rho_6 \leftarrow (-a_{11} + a_{21}) \cdot (b_{12} - b_{22})$$
$$\rho_7 \leftarrow (-a_{11} + a_{21} + a_{22}) \cdot (-b_{11} + b_{12} - b_{22})$$

Fast Matrix Multiplication algorithms and their representation

Bilinear program

$$\rho_1 \leftarrow a_{11} \cdot b_{11}$$
$$\rho_2 \leftarrow a_{12} \cdot b_{21}$$
$$\rho_3 \leftarrow (-a_{11} - a_{12} + a_{21} + a_{22}) \cdot b_{22}$$
$$\rho_4 \leftarrow a_{22} \cdot (-b_{11} + b_{12} + b_{21} - b_{22})$$
$$\rho_5 \leftarrow (a_{21} + a_{22}) \cdot (-b_{11} + b_{12})$$
$$\rho_6 \leftarrow (-a_{11} + a_{21}) \cdot (b_{12} - b_{22})$$
$$\rho_7 \leftarrow (-a_{11} + a_{21} + a_{22}) \cdot (-b_{11} + b_{12} - b_{22})$$

Equivalent L, R, P representation

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & -1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ -1 & 0 & 1 & 0 \\ -1 & 0 & 1 & 1 \end{bmatrix} \quad R = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 1 & 1 & -1 \\ -1 & 1 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ -1 & 1 & 0 & -1 \end{bmatrix}$$

Fast Matrix Multiplication algorithms and their representation

Bilinear program

$$\rho_1 \leftarrow a_{11} \cdot b_{11}$$
$$\rho_2 \leftarrow a_{12} \cdot b_{21}$$
$$\rho_3 \leftarrow (-a_{11} - a_{12} + a_{21} + a_{22}) \cdot b_{22}$$
$$\rho_4 \leftarrow a_{22} \cdot (-b_{11} + b_{12} + b_{21} - b_{22})$$
$$\rho_5 \leftarrow (a_{21} + a_{22}) \cdot (-b_{11} + b_{12})$$
$$\rho_6 \leftarrow (-a_{11} + a_{21}) \cdot (b_{12} - b_{22})$$
$$\rho_7 \leftarrow (-a_{11} + a_{21} + a_{22}) \cdot (-b_{11} + b_{12} - b_{22})$$

$$\begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} = \begin{bmatrix} \rho_1 + \rho_2 & \rho_1 - \rho_3 + \rho_5 - \rho_7 \\ \rho_1 + \rho_4 + \rho_6 - \rho_7 & \rho_1 + \rho_5 + \rho_6 - \rho_7 \end{bmatrix}$$

Equivalent L, R, P representation

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & -1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ -1 & 0 & 1 & 0 \\ -1 & 0 & 1 & 1 \end{bmatrix} \quad R = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 1 & 1 & -1 \\ -1 & 1 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ -1 & 1 & 0 & -1 \end{bmatrix}$$

$$P = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 1 & 0 & -1 \\ 1 & 0 & 0 & 1 & 0 & 1 & -1 \\ 1 & 0 & 0 & 0 & 1 & 1 & -1 \end{bmatrix}$$

Transformations of a Fast Matrix Multiplication Algorithm

Isotropies : exploiting symmetries of the matrix multiplication tensor

$[\mathsf{L}; \mathsf{R}; \mathsf{P}]$ an $n \times n$ Matrix Multiplication Representation

$$\Big\downarrow \ \diamond\, (\mathsf{U}, \mathsf{V}, \mathsf{W}) \in \mathrm{SL}^{\pm}(n, \mathbb{K})^3$$

$$\left[\mathsf{L} \cdot \left(\mathsf{V}^{\intercal} \otimes \mathsf{U}^{-1}\right); \mathsf{R} \cdot \left(\mathsf{W}^{\intercal} \otimes \mathsf{V}^{-1}\right); \left(\mathsf{U} \otimes \mathsf{W}^{-\intercal}\right) \cdot \mathsf{P}\right]$$

Transformations of a Fast Matrix Multiplication Algorithm

Isotropies : exploiting symmetries of the matrix multiplication tensor

$[\mathsf{L};\mathsf{R};\mathsf{P}]$ an $n \times n$ Matrix Multiplication Representation

$$\Big\downarrow \diamond\, (\mathsf{U},\mathsf{V},\mathsf{W}) \in \mathrm{SL}^{\pm}(n,\mathbb{K})^3$$

$$\left[\mathsf{L}\cdot\left(\mathsf{V}^{\intercal}\otimes\mathsf{U}^{-1}\right);\mathsf{R}\cdot\left(\mathsf{W}^{\intercal}\otimes\mathsf{V}^{-1}\right);\left(\mathsf{U}\otimes\mathsf{W}^{-\intercal}\right)\cdot\mathsf{P}\right]$$

[de Groote'78]: All 2×2 matrix products with 7 multiplications lie in the same orbit

$\rightsquigarrow$ usefull for exploring matrix product algorithms in 7 multiplications

Transformations of a Fast Matrix Multiplication Algorithm

Sparsification via alternative basis [Karstadt and Schwartz'17]

[L; R; P] an $n \times n$ Matrix Multiplication Representation

$$\Big\downarrow \diamond\, (\mathsf{U}, \mathsf{V}, \mathsf{W}) \in \mathrm{GL}(n^2, \mathbb{K})^3$$

[LU; RV; WP]

Transformations of a Fast Matrix Multiplication Algorithm

Sparsification via alternative basis [Karstadt and Schwartz'17]

$[\mathsf{L}; \mathsf{R}; \mathsf{P}]$ an $n \times n$ Matrix Multiplication Representation

$$\Big\downarrow \diamond (\mathsf{U}, \mathsf{V}, \mathsf{W}) \in \mathrm{GL}(n^2, \mathbb{K})^3$$

$$[\mathsf{LU}; \mathsf{RV}; \mathsf{WP}]$$

Choose $(\mathsf{U}, \mathsf{V}, \mathsf{W})$ making $[\mathsf{L}; \mathsf{R}; \mathsf{P}]$ sparser

- ✓ reduces the number of additions
- ✓ reduces the leading constant
- ✗ No longer a Matrix Multiplication alg.
  $\rightsquigarrow$ apply the inverse change of basis on the input in $O(n^2 \log n)$

Transformations of a Fast Matrix Multiplication Algorithm

Sparsification via alternative basis [Karstadt and Schwartz'17]

$[\mathsf{L};\mathsf{R};\mathsf{P}]$ an $n \times n$ Matrix Multiplication Representation

$$\Big\downarrow \ \diamond\,(\mathsf{U},\mathsf{V},\mathsf{W}) \in \mathrm{GL}(n^2,\mathbb{K})^3$$

$[\mathsf{LU};\mathsf{RV};\mathsf{WP}]$

Choose $(\mathsf{U},\mathsf{V},\mathsf{W})$ making $[\mathsf{L};\mathsf{R};\mathsf{P}]$ sparser

✓ reduces the number of additions

✓ reduces the leading constant

✗ No longer a Matrix Multiplication alg.
$\rightsquigarrow$ apply the inverse change of basis on the input in $O(n^2 \log n)$

$$\begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & -1 & 1 \\ 0 & 1 & -1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 1 & -1 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \longrightarrow \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 1 \\ 0 & 0 & -1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

- 12 additions $\rightsquigarrow 5n^{\log_2 7} + O(n^2 \log n)$

Improving accuracy

Accuracy bounds

Notations:

- ε**:** machine precision
- $\widehat{C}$**:** the approximation of $C = A \times B$ computed using floating pt arithmetic

Accuracy bounds

Notations:

ε**:** machine precision

$\widehat{C}$**:** the approximation of $C = A \times B$ computed using floating pt arithmetic

A first (strict) definition of accuracy [Miller'75]

For the conventional product:

$$|\widehat{\mathsf{C}} - \mathsf{C}| \leq f_{\mathsf{alg}}(n, \varepsilon)|\mathsf{A}| \times |\mathsf{B}| \qquad \text{(coefficient-wise)}$$

Accuracy bounds

Notations:

ε: machine precision

$\widehat{C}$: the approximation of $C = A \times B$ computed using floating pt arithmetic

A first (strict) definition of accuracy [Miller'75]

For the conventional product:

$$|\widehat{\mathsf{C}} - \mathsf{C}| \leq f_{\mathsf{alg}}(n, \varepsilon)|\mathsf{A}| \times |\mathsf{B}| \qquad \text{(coefficient-wise)}$$

- Conventionnal $O(n^3)$ product: $f_{\mathsf{alg}}(n, \varepsilon) = \frac{(2n-1)\varepsilon}{1-(2n-1)\varepsilon}$
- Moreover any algorithm matching this accuracy must be $\Omega(n^3)$

$\rightsquigarrow$ Long lasting impression that sub-cubic algorithms were unstable

Accuracy bounds

A second definition (commonly used for forward accuracy)

$$\left\| \widehat{\mathsf{C}} - \mathsf{C} \right\|_\infty \le f_{\mathsf{alg}}(n) \|\mathsf{A}\|_\infty \|\mathsf{B}\|_\infty \varepsilon + O(\varepsilon^2)$$

- Conventional $O(n^3)$ product: $f_{\mathsf{alg}}(n) = n^2$

Accuracy bounds

A second definition (commonly used for forward accuracy)

$$\left\|\widehat{\mathsf{C}} - \mathsf{C}\right\|_\infty \leq f_{\mathsf{alg}}(n)\|\mathsf{A}\|_\infty\|\mathsf{B}\|_\infty\varepsilon + O(\varepsilon^2)$$

- Conventional $O(n^3)$ product: $f_{\mathsf{alg}}(n) = n^2$
- Strassen: $f_{\mathsf{alg}}(n) = O(n^{\log_2 12}\log n)$ [Bini Lotti'80] [Demmel et al.07], [Ballard et al.'16]

Accuracy bounds

A second definition (commonly used for forward accuracy)

$$\left\|\widehat{\mathsf{C}} - \mathsf{C}\right\|_\infty \leq f_{\mathsf{alg}}(n)\|\mathsf{A}\|_\infty\|\mathsf{B}\|_\infty\varepsilon + O(\varepsilon^2)$$

- Conventional $O(n^3)$ product: $f_{\mathsf{alg}}(n) = n^2$
- Strassen: $f_{\mathsf{alg}}(n) = O(n^{\log_2 12}\log n)$ [Bini Lotti'80] [Demmel et al.07], [Ballard et al.'16]
  $f_{\mathsf{alg}}(n) = O(n^{\log_2 12})$ [Brent'70], [Higham'02]

Accuracy bounds

A second definition (commonly used for forward accuracy)

$$\left\|\widehat{\mathsf{C}} - \mathsf{C}\right\|_\infty \le f_{\mathsf{alg}}(n)\|\mathsf{A}\|_\infty\|\mathsf{B}\|_\infty\varepsilon + O(\varepsilon^2)$$

- Conventional $O(n^3)$ product: $f_{\mathsf{alg}}(n) = n^2$
- Strassen: $f_{\mathsf{alg}}(n) = O(n^{\log_2 12}\log n)$ [Bini Lotti'80] [Demmel et al.07], [Ballard et al.'16]
  $f_{\mathsf{alg}}(n) = O(n^{\log_2 12})$ [Brent'70], [Higham'02]
- Winograd: $f_{\mathsf{alg}}(n) = O(n^{\log_2 18}\log n)$ [Bini Lotti'80] [Demmel et al.07], [Ballard et al.'16]
  $f_{\mathsf{alg}}(n) = O(n^{\log_2 18})$ [Higham'02]

Searching for Fast Matrix Multiplication Algorithms with better accuracy

$$\left\|\widehat{\mathsf{C}} - \mathsf{C}\right\|_\infty \leq f_{\mathsf{alg}}(n) \|\mathsf{A}\|_\infty \|\mathsf{B}\|_\infty \varepsilon + O(\varepsilon^2)$$

with

$$f_{\mathsf{alg}}(n) = O(n^{\log_2 \gamma})$$

and

$$\gamma = \max_{k=1..4} \sum_{i=1}^{7} \|\mathsf{L}_i\|_1 \|\mathsf{R}_i\|_1 |p_{i,k}|$$

Searching for Fast Matrix Multiplication Algorithms with better accuracy

$$\left\|\widehat{\mathsf{C}}-\mathsf{C}\right\|_\infty \leq f_{\mathsf{alg}}(n)\|\mathsf{A}\|_\infty\|\mathsf{B}\|_\infty\varepsilon + O(\varepsilon^2)$$

with

$$f_{\mathsf{alg}}(n) = O(n^{\log_2 \gamma})$$

and

$$\gamma = \max_{k=1..4} \sum_{i=1}^{7} \|\mathsf{L}_i\|_1 \|\mathsf{R}_i\|_1 |p_{i,k}|$$

- ✓ $\|\cdot\|_\infty$ produces the tightest bounds
- ✓ $\gamma = 12$ (as for Strassen's algorithm) is minimal
- ✓ $\gamma = 12$ is attained by many other variants

Searching for Fast Matrix Multiplication Algorithms with better accuracy

$$\left\|\widehat{\mathsf{C}} - \mathsf{C}\right\|_\infty \leq f_{\mathsf{alg}}(n)\|\mathsf{A}\|_\infty\|\mathsf{B}\|_\infty\varepsilon + O(\varepsilon^2)$$

with

$$f_{\mathsf{alg}}(n) = O(n^{\log_2 \gamma})$$

and

$$\gamma = \max_{k=1..4} \sum_{i=1}^{7} \|\mathsf{L}_i\|_1 \|\mathsf{R}_i\|_1 |p_{i,k}|$$

- ✓ $\|\cdot\|_\infty$ produces the tightest bounds
- ✓ $\gamma = 12$ (as for Strassen's algorithm) is minimal
- ✓ $\gamma = 12$ is attained by many other variants

- ✗ no room for improvement
- ✗ does not seem to fully capture the accuracy of the algorithms

Searching for Fast Matrix Multiplication Algorithms with better accuracy

$$\left\|\widehat{\mathsf{C}} - \mathsf{C}\right\|_\infty \leq f_{\mathsf{alg}}(n) \left\|\mathsf{A}\right\| \left\|\mathsf{B}\right\| \varepsilon + O(\varepsilon^2)$$

with

$$f_{\mathsf{alg}}(n) = O(n^{\log_2 \gamma})$$

and

$$\gamma = \max_{k=1..4} \sum_{i=1}^{7} \left\|\mathsf{L}_i\right\|_* \left\|\mathsf{R}_i\right\|_* |p_{i,k}|$$

- Bound holds for any norm $\|\cdot\|$ and related dual norm $\|\cdot\|_*$:

Searching for Fast Matrix Multiplication Algorithms with better accuracy

$$\left\|\widehat{\mathsf{C}} - \mathsf{C}\right\|_\infty \leq f_{\mathsf{alg}}(n)\|\mathsf{A}\|_\infty\|\mathsf{B}\|_\infty\varepsilon + O(\varepsilon^2)$$

with

$$f_{\mathsf{alg}}(n) = O(n^{\log_2 \gamma_{1,1,\infty}})$$

and

$$\gamma_{1,1,\infty} = \max_{k=1..4} \sum_{i=1}^{7} \|\mathsf{L}_i\|_1 \|\mathsf{R}_i\|_1 |p_{i,k}|$$

- Bound holds for any norm $\|\cdot\|$ and related dual norm $\|\cdot\|_*$:
  - $(\|\cdot\|_\infty, \|\cdot\|_1)$

Searching for Fast Matrix Multiplication Algorithms with better accuracy

$$\left\|\widehat{\mathsf{C}} - \mathsf{C}\right\|_\infty \leq f_{\mathsf{alg}}(n) \left\|\mathsf{A}\right\|_1 \left\|\mathsf{B}\right\|_1 \varepsilon + O(\varepsilon^2)$$

with

$$f_{\mathsf{alg}}(n) = O(n^{\log_2 \gamma_{\infty,1,\infty}})$$

and

$$\gamma_{\infty,1,\infty} = \max_{k=1..4} \sum_{i=1}^{7} \|\mathsf{L}_i\|_\infty \|\mathsf{R}_i\|_\infty |p_{i,k}|$$

- Bound holds for any norm $\|\cdot\|$ and related dual norm $\|\cdot\|_*$:
  - $(\|\cdot\|_\infty, \|\cdot\|_1)$
  - $(\|\cdot\|_1, \|\cdot\|_\infty)$

Searching for Fast Matrix Multiplication Algorithms with better accuracy

$$\left\|\widehat{\mathsf{C}} - \mathsf{C}\right\|_\infty \leq f_{\mathsf{alg}}(n) \left\|\mathsf{A}\right\|_2 \left\|\mathsf{B}\right\|_2 \varepsilon + O(\varepsilon^2)$$

with

$$f_{\mathsf{alg}}(n) = O(n^{\log_2 \gamma_{2,1,\infty}})$$

and

$$\gamma_{2,1,\infty} = \max_{k=1..4} \sum_{i=1}^{7} \left\|\mathsf{L}_i\right\|_2 \left\|\mathsf{R}_i\right\|_2 |p_{i,k}|$$

- Bound holds for any norm $\|\cdot\|$ and related dual norm $\|\cdot\|_*$:
  - $(\|\cdot\|_\infty, \|\cdot\|_1)$
  - $(\|\cdot\|_1, \|\cdot\|_\infty)$
  - $(\|\cdot\|_2, \|\cdot\|_2)$

Agenda

Search for algorithms

- among the 2×2 multiplication algorithms in 7 products,
- using Isotropies and Sparsification transformations,
- improving accuracy $\rightsquigarrow$ **optimize** $\gamma_{2,1,\infty}$ w.r.t. norm 2,
- and with a competitive complexity's leading constant.

Agenda

Search for algorithms

- among the 2×2 multiplication algorithms in 7 products,
- using Isotropies and Sparsification transformations,
- improving accuracy $\rightsquigarrow$ **optimize** $\gamma_{2,1,\infty}$ w.r.t. norm 2,
- and with a competitive complexity's leading constant.

Simplified optimization program

- 1 isotropy $= 3 \times 4$ variables (up to permutations and rotations) $\rightsquigarrow$ too hard to solve
- Optimize on the subvariety where $\|\mathsf{L}\| = \|\mathsf{R}\| = \|\mathsf{P}\|$ $\rightsquigarrow$ 4 variables $\rightsquigarrow$ 2 variables

Optimizing the growth factor in norm 2

Weaker majorations (to optimize a smoother function):

$$\underbrace{\max_{k=1..4} \sum_{i=1}^{7} \|\mathsf{L}_i\|_2 \, \|\mathsf{R}_i\|_2 \, |p_{i,k}|}_{\gamma_{2,1,\infty}} \;\leq\; \underbrace{\sum_{i=1}^{7} \|\mathsf{L}_i\|_2 \, \|\mathsf{R}_i\|_2 \, \|\mathsf{P}^\intercal{}_i\|_2}_{\gamma_{2,1}}$$

Optimizing the growth factor in norm 2

Weaker majorations (to optimize a smoother function):

$$\underbrace{\max_{k=1..4} \sum_{i=1}^{7} \|\mathsf{L}_i\|_2 \, \|\mathsf{R}_i\|_2 \, |p_{i,k}|}_{\gamma_{2,1,\infty}} \leq \underbrace{\sum_{i=1}^{7} \|\mathsf{L}_i\|_2 \, \|\mathsf{R}_i\|_2 \, \|\mathsf{P}^\intercal{}_i\|_2}_{\gamma_{2,1}}$$

Proposition

The minimum of $\gamma_{2,1}$ *on the subvariety* $\|\mathsf{L}\| = \|\mathsf{R}\| = \|\mathsf{P}\|$ *is* $\gamma^*_{2,1} = \frac{16}{\sqrt{3}} + \frac{4}{\sqrt{2}} \approx 12.066,$

reached by the Algorithm $\mathcal{A}^* = \begin{bmatrix} \frac{\sqrt{3}}{2} & \frac{1}{2} & \frac{1}{2} & \frac{\sqrt{3}}{6} \\ 0 & 0 & 1 & -\frac{\sqrt{3}}{3} \\ 0 & 1 & 0 & \frac{\sqrt{3}}{3} \\ 0 & 0 & 0 & -\frac{2}{\sqrt{3}} \\ -\frac{\sqrt{3}}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{\sqrt{3}}{2} \\ -\frac{\sqrt{3}}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{\sqrt{3}}{6} \\ -\frac{\sqrt{3}}{2} & \frac{1}{2} & \frac{1}{2} & -\frac{\sqrt{3}}{6} \end{bmatrix}; \begin{bmatrix} 0 & \frac{2}{\sqrt{3}} & 0 & 0 \\ -1 & \frac{\sqrt{3}}{3} & 0 & 0 \\ 0 & \frac{\sqrt{3}}{3} & 0 & -1 \\ \frac{1}{2} & -\frac{\sqrt{3}}{6} & \frac{\sqrt{3}}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{\sqrt{3}}{6} & \frac{\sqrt{3}}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{\sqrt{3}}{6} & -\frac{\sqrt{3}}{2} & -\frac{1}{2} \end{bmatrix}; \begin{bmatrix} \frac{\sqrt{3}}{6} & \frac{1}{2} & \frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{\sqrt{3}}{3} & 0 & -1 & 0 \\ \frac{\sqrt{3}}{3} & -1 & 0 & 0 \\ \frac{\sqrt{3}}{6} & -\frac{1}{2} & -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{\sqrt{3}}{6} & -\frac{1}{2} & \frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{2}{\sqrt{3}} & 0 & 0 & 0 \end{bmatrix}^\intercal.$

Optimizing the growth factor in norm 2

Weaker majorations (to optimize a smoother function):

$$\underbrace{\max_{k=1..4} \sum_{i=1}^{7} \|\mathsf{L}_i\|_2 \, \|\mathsf{R}_i\|_2 \, |p_{i,k}|}_{\gamma_{2,1,\infty}} \;\leq\; \underbrace{\sum_{i=1}^{7} \|\mathsf{L}_i\|_2 \, \|\mathsf{R}_i\|_2 \, \|\mathsf{P}^{\intercal}{}_i\|_2}_{\gamma_{2,1}}$$

How far is $\gamma_{2,1}^{*} \approx 12.066$ from a global optimum?

Optimizing the growth factor in norm 2

Weaker majorations (to optimize a smoother function):

$$\underbrace{\max_{k=1..4} \sum_{i=1}^{7} \|\mathsf{L}_i\|_2 \, \|\mathsf{R}_i\|_2 \, |p_{i,k}|}_{\gamma_{2,1,\infty}} \leq \underbrace{\sum_{i=1}^{7} \|\mathsf{L}_i\|_2 \, \|\mathsf{R}_i\|_2 \, \|\mathsf{P^{\intercal}}_i\|_2}_{\gamma_{2,1}} \leq \underbrace{\|L\|_{2,3} \, \|R\|_{2,3} \, \|P\|_{2,3}}_{\|(\mathsf{L},\mathsf{R},\mathsf{P})\|_{2,3}}$$

How far is $\gamma^*_{2,1} \approx 12.066$ from a global optimum?

Proposition (Holder's inequalities)

For any algorithm $[\mathsf{L}; \mathsf{R}; \mathsf{P}]$ *in Strassen's orbit,*

$$\|(\mathsf{L}, \mathsf{R}, \mathsf{P})\|_{\Sigma} \leq \gamma_{2,1} \leq \|(\mathsf{L}, \mathsf{R}, \mathsf{P})\|_{2,3}$$

Optimizing the growth factor in norm 2

Weaker majorations (to optimize a smoother function):

$$\underbrace{\max_{k=1..4} \sum_{i=1}^{7} \|\mathsf{L}_i\|_2 \, \|\mathsf{R}_i\|_2 \, |p_{i,k}|}_{\gamma_{2,1,\infty}} \quad \leq \quad \underbrace{\sum_{i=1}^{7} \|\mathsf{L}_i\|_2 \, \|\mathsf{R}_i\|_2 \, \|\mathsf{P^T}_i\|_2}_{\gamma_{2,1}} \quad \leq \quad \underbrace{\|L\|_{2,3} \, \|R\|_{2,3} \, \|P\|_{2,3}}_{\|(\mathsf{L},\mathsf{R},\mathsf{P})\|_{2,3}}$$

How far is $\gamma^*_{2,1} \approx 12.066$ from a global optimum?

Proposition (Holder's inequalities)

For any algorithm $[\mathsf{L};\mathsf{R};\mathsf{P}]$ *in Strassen's orbit,*

$$\|(\mathsf{L},\mathsf{R},\mathsf{P})\|_\Sigma \quad \leq \quad \gamma_{2,1} \quad \leq \quad \|(\mathsf{L},\mathsf{R},\mathsf{P})\|_{2,3}$$

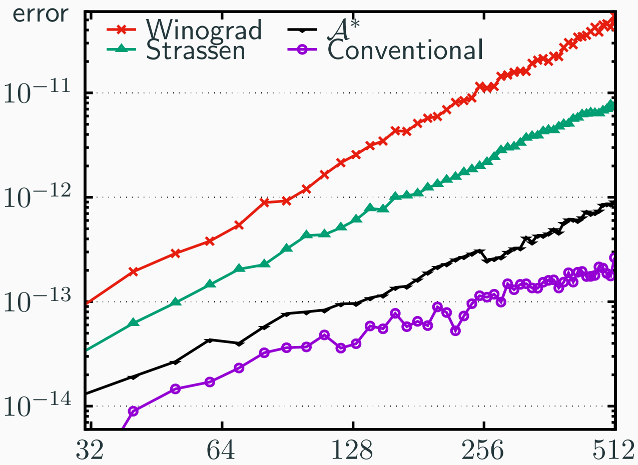
$$\|(\mathsf{L},\mathsf{R},\mathsf{P})\|_\Sigma \xrightarrow{min} 11.755 \qquad \|(\mathsf{L},\mathsf{R},\mathsf{P})\|_{2,3} \xrightarrow{min} 12.066$$

A new algorithm with improved accuracy

Algorithm	$\gamma_{1,1,\infty}$	$\gamma_{2,1,\infty}$	$\gamma_{2,1}$
Winograd'70	18	8	17.854
Strassen'69	12	6.829	14.829
$\mathcal{A}^*$	17.475	5.966	12.066

A new algorithm with improved accuracy

Algorithm	$\gamma_{1,1,\infty}$	$\gamma_{2,1,\infty}$	$\gamma_{2,1}$
Winograd'70	18	8	17.854
Strassen'69	12	6.829	14.829
$\mathcal{A}^*$	17.475	5.966	12.066



Improving speed while preserving accuracy

Improving speed while preserving accuracy

With isotropies

Any isotropy $(\mathsf{U}, \mathsf{V}, \mathsf{W})$ made of permutations and rotations preserves the growth factor γ.
$\rightsquigarrow$ helped finding the sparse algorithm $\mathcal{A}^*$ $\rightsquigarrow$ 24 ADD ($+$12 scalar MUL)

Improving speed while preserving accuracy

With isotropies

Any isotropy $(\mathsf{U}, \mathsf{V}, \mathsf{W})$ made of permutations and rotations preserves the growth factor γ.
$\rightsquigarrow$ helped finding the sparse algorithm $\mathcal{A}^*$ $\rightsquigarrow$ 24 ADD (+12 scalar MUL)

With alternative basis [Karstadt, Schwartz'17]

- Reduces to the optimal 12 ADD $\rightsquigarrow$ $5n^{\log_2 7} + O(n^2 \log n)$ complexity
- [Schwartz, Toledo, Vaknim and Wiernik'24]: Mostly preserves accuracy (still $O(n^{\log_2 \gamma})$)

$$\begin{bmatrix} 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 0 & \frac{2}{\sqrt{3}} \\ 0 & 1 & 0 & \frac{\sqrt{3}}{3} \\ 0 & 0 & 1 & -\frac{\sqrt{3}}{3} \\ -\frac{\sqrt{3}}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \quad ; \quad \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & \frac{2}{\sqrt{3}} & 0 & 0 \\ 1 & -\frac{\sqrt{3}}{3} & 0 & 0 \\ 0 & \frac{\sqrt{3}}{3} & 0 & -1 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} & -\frac{1}{2} \end{bmatrix} \quad ; \quad \begin{bmatrix} 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}^{\intercal} \times \begin{bmatrix} -\frac{2}{\sqrt{3}} & 0 & 0 & 0 \\ \frac{\sqrt{3}}{3} & -1 & 0 & 0 \\ -\frac{\sqrt{3}}{3} & 0 & -1 & 0 \\ \frac{\sqrt{3}}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix}^{\intercal}$$

Rational approximations of $\mathcal{A}^*$

$$\mathcal{A}^* = \mathsf{argmin}(\mathcal{A}(x, y)) = \mathcal{A}\left(\frac{\sqrt{2}}{\sqrt[4]{3}}, -\frac{1}{2}\right)$$

Rational approximations of $\mathcal{A}^*$

$$\mathcal{A}^* = \mathrm{argmin}(\mathcal{A}(x,y)) = \mathcal{A}\left(\frac{\sqrt{2}}{\sqrt[4]{3}}, -\frac{1}{2}\right)$$

Rational approximations of this minimal point $\rightsquigarrow$ Rational approximations of $\mathcal{A}^*$

- $\frac{\sqrt{2}}{\sqrt[4]{3}} \approx 1$ (first convergent) $\rightsquigarrow \gamma_{2,1} \approx 12.203$ and $\gamma_{2,1,\infty} \approx 6.046$

$$\mathcal{A}(1, -\frac{1}{2}) = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & -1 \end{bmatrix} \times \begin{bmatrix} 1 & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{4} \\ 0 & 0 & 1 & \frac{1}{2} \\ 0 & 0 & 1 & -\frac{1}{2} \\ 0 & -1 & 1 & 0 \end{bmatrix} \quad ; \quad \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} -\frac{1}{2} & -\frac{1}{4} & 1 & \frac{1}{2} \\ 0 & -\frac{1}{2} & 0 & 1 \\ 1 & \frac{1}{2} & 0 & 0 \\ 1 & 0 & 0 & -1 \end{bmatrix} \quad ; \quad \begin{bmatrix} 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}^{\mathsf{T}} \times \begin{bmatrix} -\frac{1}{2} & -1 & 0 & 0 \\ \frac{1}{2} & -1 & 0 & 0 \\ \frac{1}{4} & -\frac{1}{2} & -\frac{1}{2} & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}^{\mathsf{T}}$$

Rational approximations of $\mathcal{A}^*$

$$\mathcal{A}^* = \mathrm{argmin}(\mathcal{A}(x,y)) = \mathcal{A}\left(\frac{\sqrt{2}}{\sqrt[4]{3}}, -\frac{1}{2}\right)$$

Rational approximations of this minimal point $\rightsquigarrow$ Rational approximations of $\mathcal{A}^*$

- $\frac{\sqrt{2}}{\sqrt[4]{3}} \approx 1$ (first convergent) $\rightsquigarrow \gamma_{2,1} \approx 12.203$ and $\gamma_{2,1,\infty} \approx 6.046$

$$\mathcal{A}(1, -\frac{1}{2}) = \begin{bmatrix} 0&0&0&1 \\ 1&0&0&0 \\ 0&0&1&0 \\ 0&0&1&-1 \\ 0&1&0&0 \\ 1&0&0&1 \\ 0&1&0&-1 \end{bmatrix} \times \begin{bmatrix} 1 & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{4} \\ 0 & 0 & 1 & \frac{1}{2} \\ 0 & 0 & 1 & -\frac{1}{2} \\ 0 & -1 & 1 & 0 \end{bmatrix} \quad ; \quad \begin{bmatrix} 0&0&0&1 \\ 0&0&1&0 \\ 0&-1&0&0 \\ -1&0&0&0 \\ 0&0&1&-1 \\ 0&1&0&1 \\ 1&0&0&1 \end{bmatrix} \times \begin{bmatrix} -\frac{1}{2} & -\frac{1}{4} & 1 & \frac{1}{2} \\ 0 & -\frac{1}{2} & 0 & 1 \\ 1 & \frac{1}{2} & 0 & 0 \\ 1 & 0 & 0 & -1 \end{bmatrix} \quad ; \quad \begin{bmatrix} 0&0&0&1 \\ -1&0&0&0 \\ 0&0&1&0 \\ 1&0&0&1 \\ 0&0&1&1 \\ 0&1&0&0 \\ 0&1&0&1 \end{bmatrix}^{\mathsf{T}} \times \begin{bmatrix} -\frac{1}{2} & -1 & 0 & 0 \\ \frac{1}{2} & -1 & 0 & 0 \\ \frac{1}{4} & -\frac{1}{2} & -\frac{1}{2} & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}^{\mathsf{T}}$$

- $\frac{\sqrt{2}}{\sqrt[4]{3}} \approx \frac{14}{13}$ $\rightsquigarrow \gamma_{2,1} \approx 12.0662$ and $\gamma_{2,1,\infty} \approx 6.000043$
- ...

Implementations

Remaining challenge

Find the best operation schedule from a matrix representation ?

Remaining challenge

Find the best operation schedule from a matrix representation ?

- PlinOpt: https://github.com/jgdumas/plinopt

$$\begin{bmatrix} \frac{\sqrt{3}}{2} & \frac{1}{2} & \frac{1}{2} & \frac{\sqrt{3}}{6} \\ 0 & 0 & 1 & -\frac{\sqrt{3}}{3} \\ 0 & 1 & 0 & \frac{\sqrt{3}}{3} \\ 0 & 0 & 0 & -\frac{2}{\sqrt{3}} \\ -\frac{\sqrt{3}}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{\sqrt{3}}{2} \\ -\frac{\sqrt{3}}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{\sqrt{3}}{6} \\ -\frac{\sqrt{3}}{2} & \frac{1}{2} & \frac{1}{2} & -\frac{\sqrt{3}}{6} \end{bmatrix};\begin{bmatrix} 0 & \frac{2}{\sqrt{3}} & 0 & 0 \\ -1 & \frac{\sqrt{3}}{3} & 0 & 0 \\ 0 & \frac{\sqrt{3}}{3} & 0 & -1 \\ \frac{1}{2} & -\frac{\sqrt{3}}{6} & \frac{\sqrt{3}}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{\sqrt{3}}{6} & \frac{\sqrt{3}}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{\sqrt{3}}{6} & -\frac{\sqrt{3}}{2} & -\frac{1}{2} \end{bmatrix};\begin{bmatrix} \frac{\sqrt{3}}{6} & \frac{1}{2} & \frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{\sqrt{3}}{3} & 0 & -1 & 0 \\ \frac{\sqrt{3}}{3} & -1 & 0 & 0 \\ \frac{\sqrt{3}}{6} & -\frac{1}{2} & -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{\sqrt{3}}{6} & -\frac{1}{2} & \frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{2}{\sqrt{3}} & 0 & 0 & 0 \end{bmatrix}^{\intercal} \rightsquigarrow$$

$$\begin{array}{llll} t_1 = \frac{\sqrt{3}}{3}a_{22} & t_2 = a_{21} + t_1 & s_1 = \frac{\sqrt{3}}{3}b_{21} & s_2 = s_1 - b_{11} \\ t_3 = a_{12} + t_2 & l_1 = \frac{\sqrt{3}}{2}a_{11} + \frac{1}{2}t_3 & s_3 = s_2 + b_{22} & r_1 = 2s_1 \\ l_2 = a_{12} - t_1 & l_3 = t_2 & r_2 = s_2 & r_3 = s_1 - b_{22} \\ l_4 = 2t_1 & l_5 = l_2 - l_1 & r_4 = \frac{1}{2}s_3 - \frac{\sqrt{3}}{2}b_{12} & r_5 = r_3 + r_4 \\ l_6 = l_5 + l_4 & l_7 = l_5 + l_3 & r_6 = r_1 - r_5 & r_7 = r_5 - r_2 \end{array}$$

$$\begin{array}{llll} p_1 = l_1 \cdot r_1 & p_2 = l_2 \cdot r_2 & p_3 = l_3 \cdot r_3 & p_4 = l_4 \cdot r_4 \\ p_5 = l_5 \cdot r_5 & p_6 = l_6 \cdot r_6 & p_7 = l_7 \cdot r_7 & \end{array}$$

$$\begin{array}{llll} w_2 = p_5 + p_1 + p_6 & w_1 = p_7 + p_6 & w_3 = w_2 - p_2 & w_5 = \frac{p_4 + w_2}{2} \\ c_{12} = p_1 - p_3 - w_5 & c_{21} = w_3 - w_5 & c_{22} = \sqrt{3}w_5 & \end{array}$$

$$c_{11} = \frac{\sqrt{3}}{3}(w_3 - c_{12} - 2w_1)$$

Implementations

Remaining challenge

Find the best operation schedule from a matrix representation ?

- `PlinOpt`: `https://github.com/jgdumas/plinopt`

$$
\begin{bmatrix}
\frac{\sqrt{3}}{2} & \frac{1}{2} & \frac{1}{2} & \frac{\sqrt{3}}{6} \\
0 & 0 & 1 & -\frac{\sqrt{3}}{3} \\
0 & 1 & 0 & \frac{\sqrt{3}}{3} \\
0 & 0 & 0 & -\frac{2}{\sqrt{3}} \\
-\frac{\sqrt{3}}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{\sqrt{3}}{2} \\
-\frac{\sqrt{3}}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{\sqrt{3}}{6} \\
-\frac{\sqrt{3}}{2} & \frac{1}{2} & \frac{1}{2} & -\frac{\sqrt{3}}{6}
\end{bmatrix};
\begin{bmatrix}
0 & \frac{2}{\sqrt{3}} & 0 & 0 \\
-1 & \frac{\sqrt{3}}{3} & 0 & 0 \\
0 & \frac{\sqrt{3}}{3} & 0 & -1 \\
\frac{1}{2} & -\frac{\sqrt{3}}{6} & \frac{\sqrt{3}}{2} & -\frac{1}{2} \\
-\frac{1}{2} & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} & -\frac{1}{2} \\
\frac{1}{2} & \frac{\sqrt{3}}{6} & \frac{\sqrt{3}}{2} & \frac{1}{2} \\
\frac{1}{2} & \frac{\sqrt{3}}{6} & -\frac{\sqrt{3}}{2} & -\frac{1}{2}
\end{bmatrix};
\begin{bmatrix}
\frac{\sqrt{3}}{6} & \frac{1}{2} & \frac{1}{2} & \frac{\sqrt{3}}{2} \\
-\frac{\sqrt{3}}{3} & 0 & -1 & 0 \\
\frac{\sqrt{3}}{3} & -1 & 0 & 0 \\
\frac{\sqrt{3}}{6} & -\frac{1}{2} & -\frac{1}{2} & \frac{\sqrt{3}}{2} \\
\frac{\sqrt{3}}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{\sqrt{3}}{2} \\
-\frac{\sqrt{3}}{6} & -\frac{1}{2} & \frac{1}{2} & \frac{\sqrt{3}}{2} \\
-\frac{2}{\sqrt{3}} & 0 & 0 & 0
\end{bmatrix}^{\intercal} \rightsquigarrow
$$

$$
\begin{array}{llll}
t_1 = \frac{\sqrt{3}}{3}a_{22} & t_2 = a_{21} + t_1 & s_1 = \frac{\sqrt{3}}{3}b_{21} & s_2 = s_1 - b_{11} \\
t_3 = a_{12} + t_2 & l_1 = \frac{\sqrt{3}}{2}a_{11} + \frac{1}{2}t_3 & s_3 = s_2 + b_{22} & r_1 = 2s_1 \\
l_2 = a_{12} - t_1 & l_3 = t_2 & r_2 = s_2 & r_3 = s_1 - b_{22} \\
l_4 = 2t_1 & l_5 = l_2 - l_1 & r_4 = \frac{1}{2}s_3 - \frac{\sqrt{3}}{2}b_{12} & r_5 = r_3 + r_4 \\
l_6 = l_5 + l_4 & l_7 = l_5 + l_3 & r_6 = r_1 - r_5 & r_7 = r_5 - r_2
\end{array}
$$

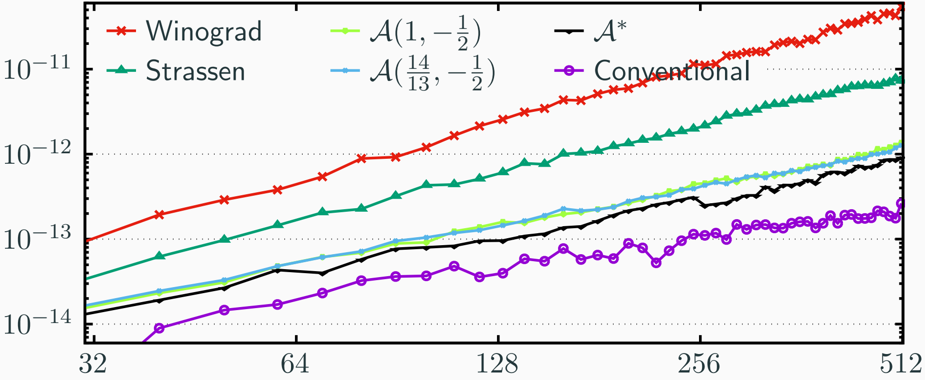
$$
\begin{array}{llll}
p_1 = l_1 \cdot r_1 & p_2 = l_2 \cdot r_2 & p_3 = l_3 \cdot r_3 & p_4 = l_4 \cdot r_4 \\
p_5 = l_5 \cdot r_5 & p_6 = l_6 \cdot r_6 & p_7 = l_7 \cdot r_7 &
\end{array}
$$

$$
\begin{array}{llll}
w_2 = p_5 + p_1 + p_6 & w_1 = p_7 + p_6 & w_3 = w_2 - p_2 & w_5 = \frac{p_4 + w_2}{2} \\
c_{12} = p_1 - p_3 - w_5 & c_{21} = w_3 - w_5 & c_{22} = \sqrt{3}w_5 &
\end{array}
$$

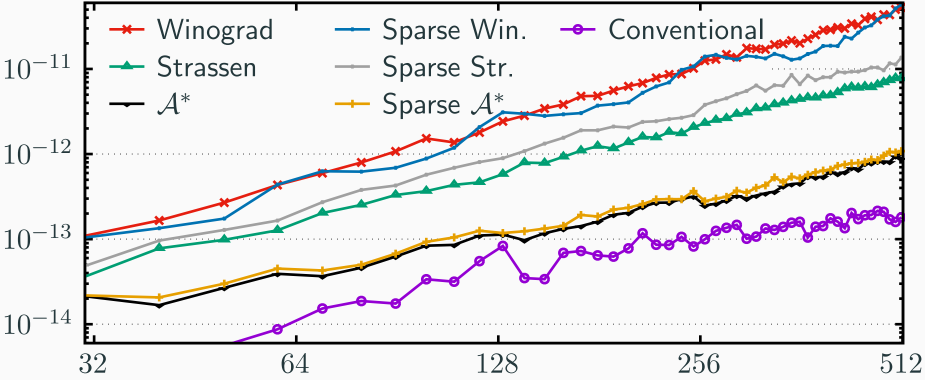
$$
c_{11} = \frac{\sqrt{3}}{3}(w_3 - c_{12} - 2w_1)
$$

- 24 ADD and 12 MUL, instead of 45 ADD and 57 MUL (naive MatVec application)

Accuracy of rational approximations

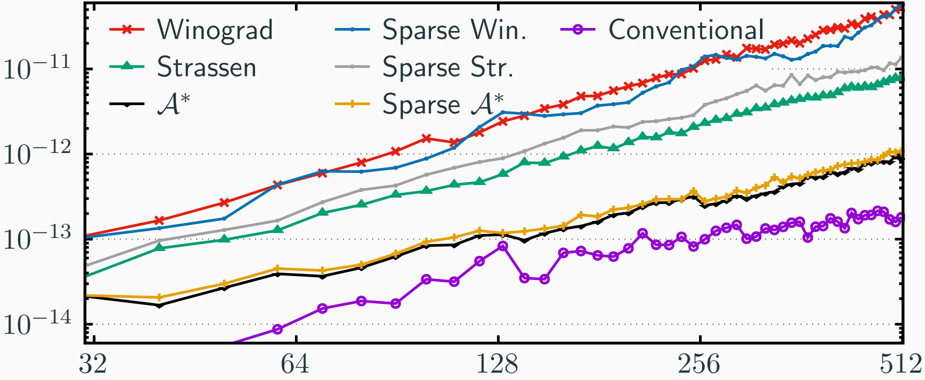


New algorithms, fast and accurate



All MatMul algorithms (LRP representations and schedule), maple optimization programs, PlinOpt scripts and Matlab benchmarks are available on `https://github.com/jgdumas/Fast-Matrix-Multiplication`.

New algorithms, fast and accurate



All MatMul algorithms (LRP representations and schedule), maple optimization programs, PlinOpt scripts and Matlab benchmarks are available on `https://github.com/jgdumas/Fast-Matrix-Multiplication`.

Thank you