

Fast and accurate subcubic matrix multiplication

Jean-Guillaume Dumas  Clément Pernet Alexandre Sedoglavic

Séminaire Pascaline LIP,
Lyon, France. 18 Décembre 2025

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



Subcubic matrix multiplication in practice: a long lasting defiance

2×2 matrix multiplication	# Multiplications	# Additions
Classic $O(n^3)$	8	4
[Strassen'69]	7	18
[Winograd'70]	7	15

Subcubic matrix multiplication in practice: a long lasting defiance

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

Subcubic matrix multiplication in practice: a long lasting defiance

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

Speed ? better exponent but worse leading constant

\rightsquigarrow only for large n

Subcubic matrix multiplication in practice: a long lasting defiance

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

Speed ? better exponent but worse leading constant

\rightsquigarrow 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

Subcubic matrix multiplication in practice: a long lasting defiance

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

Speed ? better exponent but worse leading constant
Stability ?: aggregation-cancellation, coefficient growth

\rightsquigarrow only for large n
 \rightsquigarrow less accurate

Subcubic matrix multiplication in practice: a long lasting defiance

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

Speed ? better exponent but worse leading constant

\rightsquigarrow only for large n

Stability ?: aggregation-cancelation, coefficient growth

\rightsquigarrow less accurate

[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.
```

Subcubic matrix multiplication in practice: 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

Subcubic matrix multiplication in practice: 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.

Subcubic matrix multiplication in practice: a long lasting defiance

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

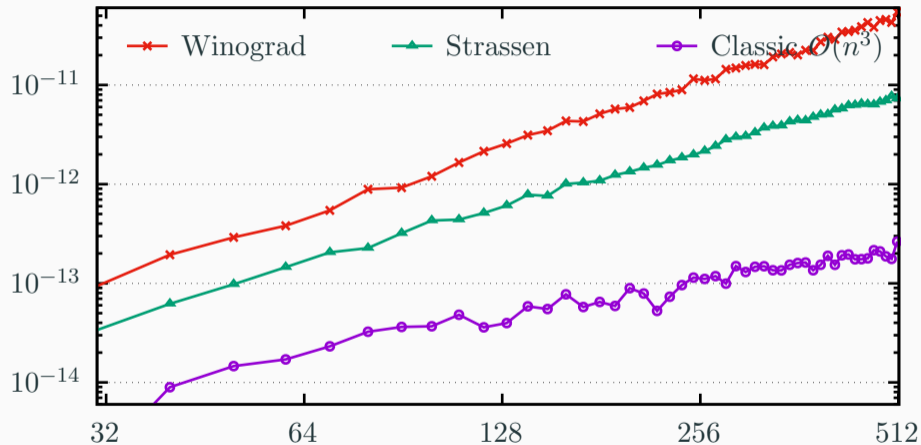
From a practical point of view, Strassen's algorithm is often not the method of choice for matrix multiplication, for four reasons:

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.

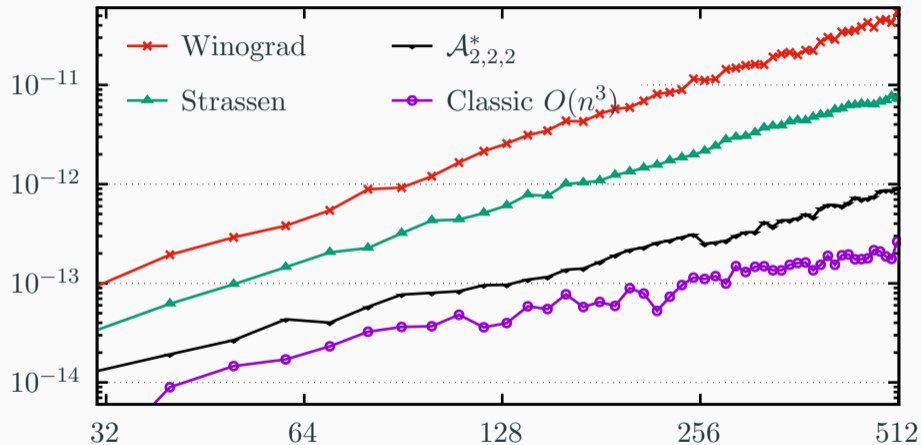
Debunking in progress

1. [Schwarz et al.] and [here] $\rightsquigarrow 5n^{\log_2 7}$ [here] $\rightsquigarrow 7n^{\log_4 48}$
3. [Brent 70], [Bini Lotti'80], [Demmel'93], [Higham'02], [Demmel et al.'07] ... and [here]
4. Pebble games (or [Dumas Grenet'24] $\rightsquigarrow 8n^{\log_2 7}$ fully inplace) ^{2/29}

Accuracy of recursive 2×2 matrix multiplication algorithms



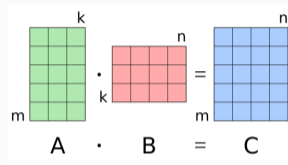
Contribution: new algorithms with improved accuracy and leading constant



Fast Matrix Multiplication algorithms and their representation

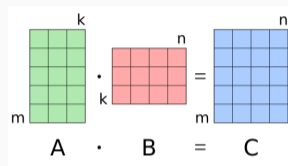
Small Matrix Multiplication

- ▶ Algorithm $\langle m \times k \times n : r \rangle$ or $\langle m \times k \times n : r \mid a \rangle$:
 - ◊ Multiplies $m \times k$ by $k \times n$ matrices
 - ◊ with r multiplications (and a additions).
- ▶ Standard MM algorithm:
 - ◊ $\langle m \times k \times n : mkn \mid m(k-1)n \rangle$,
 - ◊ $\langle n \times n \times n : n^3 \mid n^3 - n^2 \rangle$
 - ◊ $\rightsquigarrow \langle 2 \times 2 \times 2 : 8 \mid 4 \rangle, \langle 5 \times 3 \times 4 : 60 \mid 40 \rangle$,



Small Matrix Multiplication

- ▶ Algorithm $\langle m \times k \times n : r \rangle$ or $\langle m \times k \times n : r \mid a \rangle$:
 - ◊ Multiplies $m \times k$ by $k \times n$ matrices
 - ◊ with r multiplications (and a additions).
- ▶ Standard MM algorithm:
 - ◊ $\langle m \times k \times n : mkn \mid m(k-1)n \rangle$,
 - ◊ $\langle n \times n \times n : n^3 \mid n^3 - n^2 \rangle$
 - ◊ $\rightsquigarrow \langle 2 \times 2 \times 2 : 8 \mid 4 \rangle, \langle 5 \times 3 \times 4 : 60 \mid 40 \rangle$,

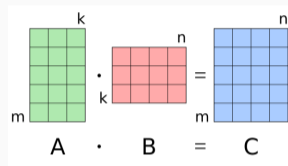


Theorem (Strassen'69, Winograd'77, Groote'78)

$\langle 2 \times 2 \times 2 : 7 \mid 15 \rangle$ is minimal (favoring multiplications)

Small Matrix Multiplication

- ▶ Algorithm $\langle m \times k \times n : r \rangle$ or $\langle m \times k \times n : r \mid a \rangle$:
 - ◊ Multiplies $m \times k$ by $k \times n$ matrices
 - ◊ with r multiplications (and a additions).
- ▶ Standard MM algorithm:
 - ◊ $\langle m \times k \times n : mkn \mid m(k-1)n \rangle$,
 - ◊ $\langle n \times n \times n : n^3 \mid n^3 - n^2 \rangle$
 - ◊ $\rightsquigarrow \langle 2 \times 2 \times 2 : 8 \mid 4 \rangle, \langle 5 \times 3 \times 4 : 60 \mid 40 \rangle$,



Theorem (Strassen'69, Winograd'77, Groote'78)

$\langle 2 \times 2 \times 2 : 7 \mid 15 \rangle$ is minimal (favoring multiplications)

Open problems: for given $m > 2$, $k > 2$ and $n > 2$

- ▶ best upper bound on the rank r
- ▶ minimality of $r(m, n, k)$
- ▶ for a given r , best upper bound on a
- ▶ minimality of $a(m, n, k, r)$

Matrix Multiplication Asymptotics

Lemma

A non-commutative $\langle i \times i \times i : r \rangle$ algorithm can multiply $n \times n$ matrices in $O(n^{\log_i(r)})$ operations, if $\log_i(r) > 2$

Proof.

Divide & Conquer recursively:

$$\text{Cost}(n) = r \cdot \text{Cost}\left(\frac{n}{i}\right) + O(n^2), \text{ and } \text{Cost}(1) = 1$$
$$\rightsquigarrow \text{Cost}(n) = O(r^{\log_i(n)}) = O(n^{\log_i(r)})$$



Matrix Multiplication Asymptotics

Lemma

A non-commutative $\langle i \times i \times i : r \rangle$ algorithm can multiply $n \times n$ matrices in $O(n^{\log_i(r)})$ operations, if $\log_i(r) > 2$

Proof.

Divide & Conquer recursively:

$$\begin{aligned}\text{Cost}(n) &= r \cdot \text{Cost}\left(\frac{n}{i}\right) + O(n^2), \text{ and } \text{Cost}(1) = 1 \\ \rightsquigarrow \text{Cost}(n) &= O(r^{\log_i(n)}) = O(n^{\log_i(r)})\end{aligned}$$

□

Strassen's subcubic $\langle 2 \times 2 \times 2 : 7 \rangle$ algorithm

MM in $O(n^\omega)$ operations with $\omega = \log_2(7) \approx 2.807354922$

Some Fast Matrix Multiplication Exponents

2.81	[Strassen'69]
2.79	[Pan'78]
2.78	[Bini'79]
2.55	[Schönage'81]
2.50	[Pan-Romani,Coppersmith-Winograd'84]
2.48	[Strassen'87]
2.38	[Coppersmith-Winograd'90]
2.3730	[Stothers'10]
2.3728640	[Williams'12]
2.3728639	[Le Gall'14]
2.372859	[Alman-Williams'21]
2.371866	[Duan-Wu-Zhou'22]
2.371552	[Williams-Xu-Xu-Zhou'23]
2.371339	[Alman-Duan-Williams-Xu-Xu-Zhou'24]

Some Fast Matrix Multiplication Exponents

2.81	[Strassen'69]
2.79	[Pan'78]
2.78	[Bini'79]
2.55	[Schönage'81]
2.50	[Pan-Romani,Coppersmith-Winograd'84]
2.48	[Strassen'87]
2.38	[Coppersmith-Winograd'90]
2.3730	[Stothers'10]
2.3728640	[Williams'12]
2.3728639	[Le Gall'14]
2.372859	[Alman-Williams'21]
2.371866	[Duan-Wu-Zhou'22]
2.371552	[Williams-Xu-Xu-Zhou'23]
2.371339	[Alman-Duan-Williams-Xu-Xu-Zhou'24]

Theorem (Alman-Yu'25)

if $\omega < 2.372$, $n \times n$ matrices can be multiplied over any field using at most

$$c_\epsilon^{O(\epsilon^2)} M^{\omega+\epsilon} + c_\epsilon^{O(1/\epsilon^2)} M^2$$

field operations, with $c_\epsilon \approx 2^{2^{\Theta(1/\epsilon^2)}}$.

Some Feasible Fast Matrix Multiplication Algorithms

$(m \times k \times n : r)$	r	naive	construction	exponent
$(2 \times 2 \times 2 : 7)$	7	8	Strassen-1969	2.807354922
$(3 \times 3 \times 3 : 23)$	23	27	Laderman-1976	2.854049830
$(4 \times 4 \times 4 : 48)$	48	64	D.-P.-S.-2025	2.792481250
$(5 \times 5 \times 5 : 93)$	93	125	Moosbauer-Poole-2025	2.816262409
$(6 \times 6 \times 6 : 153)$	153	216	Moosbauer-Poole-2025	2.807540860
$(7 \times 7 \times 7 : 249)$	249	343	$(4 \times 4 \times 4 : 48) + 3(3 \times 3 \times 4 : 29) + 3(3 \times 4 \times 4 : 38)$	2.835409898
$(8 \times 8 \times 8 : 336)$	336	512	$(2 \times 2 \times 2 : 7) + (4 \times 4 \times 4 : 48)$	2.797439141
$(9 \times 9 \times 9 : 498)$	498	729	$6(3 \times 3 \times 4 : 29) + 9(3 \times 3 \times 5 : 36)$	2.826565905
$(10 \times 10 \times 10 : 651)$	651	1000	$(2 \times 2 \times 2 : 7) \otimes (5 \times 5 \times 5 : 93)$	2.813580989
$(11 \times 11 \times 11 : 873)$	873	1331	$(6 \times 6 \times 6 : 153) + 3(5 \times 5 \times 6 : 110) + 3(5 \times 6 \times 6 : 130)$	2.824116479
$(12 \times 12 \times 12 : 1040)$	1040	1728	$(2 \times 4 \times 4 : 26) \otimes (6 \times 3 \times 3 : 40)$	2.795668800
$(13 \times 13 \times 13 : 1426)$	1426	2197	$(4 \times 4 \times 6 : 73) + (5 \times 5 \times 6 : 110) + (5 \times 5 \times 7 : 127) + 4(4 \times 4 \times 7 : 85) + 4(4 \times 5 \times 6 : 90) + 4(4 \times 5 \times 7 : 104)$	2.831490056
$(14 \times 14 \times 14 : 1719)$	1719	2744	$(7 \times 7 \times 7 : 249) + 3TA((7 \times 7 \times 7), (7 \times 7 \times 7))$	2.822287486
$(15 \times 15 \times 15 : 2058)$	2058	3375	$6(5 \times 5 \times 7 : 127) + 9(5 \times 5 \times 8 : 144)$	2.817336958
$(16 \times 16 \times 16 : 2304)$	2304	4096	$(4 \times 4 \times 4 : 48) \otimes (4 \times 4 \times 4 : 48)$	2.792481250
$(17 \times 17 \times 17 : 2934)$	2934	4913	$2(8 \times 8 \times 8 : 336) + 3(8 \times 9 \times 9 : 430) + TA((9 \times 9 \times 9), (9 \times 9 \times 9))$	2.818044739
$(18 \times 18 \times 18 : 3200)$	3200	5832	$(3 \times 3 \times 6 : 40) \otimes (6 \times 6 \times 3 : 80)$	2.792341873
$(19 \times 19 \times 19 : 4033)$	4033	6859	$(9 \times 10 \times 10 : 600) + (10 \times 10 \times 10 : 651) + 3(9 \times 9 \times 10 : 534) + TA((9 \times 10 \times 10), (10 \times 9 \times 10))$	2.819642673
$(20 \times 20 \times 20 : 4340)$	4340	8000	Drevet-NazrullIslam-Schost-2011	2.795853856
$(21 \times 21 \times 21 : 5240)$	5240	9261	$(12 \times 12 \times 12 : 1040) + 3(9 \times 9 \times 12 : 600) + 3(9 \times 12 \times 12 : 800)$	2.812945857
$(22 \times 22 \times 22 : 5566)$	5566	10648	Drevet-NazrullIslam-Schost-2011	2.790137008
$(23 \times 23 \times 23 : 6724)$	6724	12167	$Proj([1, 14], [14], (24 \times 24 \times 24 : 7000))$	2.810861028
$(24 \times 24 \times 24 : 7000)$	7000	13824	Drevet-NazrullIslam-Schost-2011	2.785876483
$(25 \times 25 \times 25 : 8359)$	8359	15625	$Proj([1, 15], [15], (26 \times 26 \times 26 : 8658))$	2.805667125
$(26 \times 26 \times 26 : 8658)$	8658	17576	Drevet-NazrullIslam-Schost-2011	2.782679679
$(27 \times 27 \times 27 : 10234)$	10234	19683	$Proj([1, 16], [16], (28 \times 28 \times 28 : 10550))$	2.801555770
$(28 \times 28 \times 28 : 10550)$	10550	21952	Schwartz-Zwecher-2025	2.780105816
$(29 \times 29 \times 29 : 12365)$	12365	24389	$Proj([1, 17], [17], (30 \times 30 \times 30 : 12688))$	2.798276616
$(30 \times 30 \times 30 : 12688)$	12688	27000	Schwartz-Zwecher-2025	2.777966370
$(31 \times 31 \times 31 : 14768)$	14768	29791	$Proj([1, 18], [18], (32 \times 32 \times 32 : 15096))$	2.795647562
$(32 \times 32 \times 32 : 15096)$	15096	32768	Schwartz-Zwecher-2025	2.776375742

Some Feasible Fast Matrix Multiplication Algorithms

$\langle m \times k \times n : r \rangle$	r	naive	construction	exponent
$\langle 2 \times 2 \times 2 : 7 \rangle$	7	8	Strassen-1969	2.807
$\langle 2 \times 6 \times 6 : 56 \rangle$	56	72	Kauers-Moosbauer-2023	2.824
$\langle 2 \times 6 \times 7 : 66 \rangle$	66	84	Kauers-Wood-2025	2.837
$\langle 3 \times 3 \times 3 : 23 \rangle$	23	27	Laderman-1976	2.854
$\langle 3 \times 5 \times 5 : 58 \rangle$	58	75	S.-Smirnov-2021	2.821
$\langle 3 \times 4 \times 7 : 63 \rangle$	63	84	D.-P.-S.-2025	2.806
$\langle 4 \times 4 \times 4 : 48 \rangle$	48	64	D.-P.-S.-2025	2.792
$\langle 6 \times 6 \times 6 : 153 \rangle$	153	216	Moosbauer-Poole-2025	2.808

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

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

$$\left. \begin{array}{c} \\ \\ \end{array} \right) \diamond (U, V, W) \in \text{GL}(n, \mathbb{K})^3$$
$$[L \cdot (V^T \otimes U^{-1}); R \cdot (W^T \otimes V^{-1}); (U \otimes W^{-T}) \cdot P]$$

Transformations of a Fast Matrix Multiplication Algorithm

Isotropies : exploiting symmetries of the matrix multiplication tensor

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

$$\left. \begin{array}{l} \\ \\ \end{array} \right) \diamond (U, V, W) \in GL(n, \mathbb{K})^3$$
$$[L \cdot (V^T \otimes U^{-1}); R \cdot (W^T \otimes V^{-1}); (U \otimes W^{-T}) \cdot P]$$

[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

$$\left. \begin{array}{l} \\ \\ \end{array} \right) \diamond (U, V, W) \in GL(n^2, \mathbb{K})^3$$

$[LU; RV; WP]$

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

$$\left. \begin{array}{l} \\ \\ \end{array} \right) \diamond (U, V, W) \in GL(n^2, \mathbb{K})^3$$

$[LU; RV; WP]$

Choose (U, V, W) making $[L; R; P]$ 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]

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

$$\left. \begin{array}{c} \\ \\ \\ \end{array} \right) \diamond (U, V, W) \in GL(n^2, \mathbb{K})^3$$
$$[LU; RV; WP]$$

Choose (U, V, W) making $[L; R; P]$ 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 $O(n^2 \log n)$

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

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

▶ optimal

Improving accuracy

Accuracy bounds

Notations:

ε : machine precision

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

Accuracy bounds

Notations:

ε : machine precision

\hat{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:

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

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:

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

- ▶ Classic $2n^3$ product: $f_{\text{alg}}(n, \varepsilon) = n\varepsilon + O(\varepsilon^2)$
- ▶ Moreover any algorithm matching this accuracy must be $\Omega(n^3)$

↪ Long lasting impression that sub-cubic algorithms were unstable

A second definition (commonly used for forward accuracy)

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

where $\|A\|_{\infty} = \max_{i,j} |a_{i,j}|$ is the max-norm

- ▶ Classic $O(n^3)$ product: $f_{\text{alg}}(n) = n^2$

A second definition (commonly used for forward accuracy)

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

where $\|A\|_{\infty} = \max_{i,j} |a_{i,j}|$ is the max-norm

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

A second definition (commonly used for forward accuracy)

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

where $\|A\|_{\infty} = \max_{i,j} |a_{i,j}|$ is the max-norm

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

A second definition (commonly used for forward accuracy)

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

where $\|\mathbf{A}\|_{\infty} = \max_{i,j} |a_{i,j}|$ is the max-norm

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

Generalizing the accuracy bound for Fast Matrix Multiplication

$$\|\hat{C} - C\|_{\infty} \leq f_{\text{alg}}(n) \|A\|_{\infty} \|B\|_{\infty} \varepsilon + O(\varepsilon^2)$$

with

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

and

$$\gamma = \max_{k=1..4} \sum_{i=1}^7 \|L_i\|_1 \|R_i\|_1 |p_{i,k}|$$

Generalizing the accuracy bound for Fast Matrix Multiplication

$$\|\hat{\mathbf{C}} - \mathbf{C}\|_{\infty} \leq f_{\text{alg}}(n) \|\mathbf{A}\|_{\infty} \|\mathbf{B}\|_{\infty} \varepsilon + O(\varepsilon^2)$$

with

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

and

$$\gamma = \max_{k=1..4} \sum_{i=1}^7 \|L_i\|_1 \|R_i\|_1 |p_{i,k}|$$

- ✓ $\|\cdot\|_{\infty}$ in right-hand side produces the tightest bounds
- ✓ $\gamma = 12$ (as for Strassen's algorithm) is minimal [Bini Lotti'80]
- ✓ $\gamma = 12$ is reached by many other variants

Generalizing the accuracy bound for Fast Matrix Multiplication

$$\|\hat{C} - C\|_{\infty} \leq f_{\text{alg}}(n) \|A\|_{\infty} \|B\|_{\infty} \varepsilon + O(\varepsilon^2)$$

with

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

and

$$\gamma = \max_{k=1..4} \sum_{i=1}^7 \|L_i\|_1 \|R_i\|_1 |p_{i,k}|$$

- ✓ $\|\cdot\|_{\infty}$ in right-hand side produces the tightest bounds
- ✓ $\gamma = 12$ (as for Strassen's algorithm) is minimal [Bini Lotti'80]
- ✓ $\gamma = 12$ is reached by many other variants
- ✗ no room for improvement
- ✗ does not seem to fully capture the accuracy of the algorithms

Generalizing the accuracy bound for Fast Matrix Multiplication

$$\|\hat{\mathbf{C}} - \mathbf{C}\|_{\infty} \leq f_{\text{alg}}(n) \|\mathbf{A}\| \|\mathbf{B}\| \varepsilon + O(\varepsilon^2)$$

with

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

and

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

- ▶ Bound holds for any norm $\|\cdot\|$ and related dual norm $\|\cdot\|^*$ over \mathbb{R}^{n^2} :

Generalizing the accuracy bound for Fast Matrix Multiplication

$$\|\hat{\mathbf{C}} - \mathbf{C}\|_{\infty} \leq f_{\text{alg}}(n) \|\mathbf{A}\|_{\infty} \|\mathbf{B}\|_{\infty} \varepsilon + O(\varepsilon^2)$$

with

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

and

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

- ▶ Bound holds for any norm $\|\cdot\|$ and related dual norm $\|\cdot\|^*$ over \mathbb{R}^{n^2} :
 - ◊ $(\|\cdot\|_{\infty}, \|\cdot\|_1)$

Generalizing the accuracy bound for Fast Matrix Multiplication

$$\|\hat{\mathbf{C}} - \mathbf{C}\|_{\infty} \leq f_{\text{alg}}(n) \|\mathbf{A}\|_1 \|\mathbf{B}\|_1 \varepsilon + O(\varepsilon^2)$$

with

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

and

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

- ▶ Bound holds for any norm $\|\cdot\|$ and related dual norm $\|\cdot\|^*$ over \mathbb{R}^{n^2} :
 - ◊ $(\|\cdot\|_{\infty}, \|\cdot\|_1)$
 - ◊ $(\|\cdot\|_1, \|\cdot\|_{\infty})$

Generalizing the accuracy bound for Fast Matrix Multiplication

$$\|\hat{\mathbf{C}} - \mathbf{C}\|_{\infty} \leq f_{\text{alg}}(n) \|\mathbf{A}\|_2 \|\mathbf{B}\|_2 \varepsilon + O(\varepsilon^2)$$

with

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

and

$$\gamma = \max_{k=1..4} \sum_{i=1}^7 \|\mathbf{L}_i\|_2 \|\mathbf{R}_i\|_2 |p_{i,k}|$$

- ▶ Bound holds for any norm $\|\cdot\|$ and related dual norm $\|\cdot\|^*$ over \mathbb{R}^{n^2} :
 - ◇ $(\|\cdot\|_{\infty}, \|\cdot\|_1)$
 - ◇ $(\|\cdot\|_1, \|\cdot\|_{\infty})$
 - ◇ $(\|\cdot\|_2, \|\cdot\|_2)$

Generalizing the accuracy bound for Fast Matrix Multiplication

$$\left\| \widehat{\mathbf{C}} - \mathbf{C} \right\|_p \leq f_{\text{alg}}(n) \|\mathbf{A}\|_q \|\mathbf{B}\|_q \varepsilon + O(\varepsilon^2)$$

with

$$f_{\text{alg}}(n) = O(n^{\log_2 \gamma_{p,q}})$$

and

$$\gamma_{p,q} = \left\| \sum_{i=1}^7 \|\mathbf{L}_i\|_q^* \|\mathbf{R}_i\|_q^* |p_{i,k}| \right\|_p$$

- ▶ Bound holds for any norm $\|\cdot\|$ and related dual norm $\|\cdot\|^*$ over \mathbb{R}^{n^2} :
- ▶ arbitrary $\|\cdot\|_p$ on LHS $\|\cdot\|_q$ on RHS

Generalizing the accuracy bound for Fast Matrix Multiplication

$$\left\| \widehat{\mathbf{C}} - \mathbf{C} \right\|_p \leq f_{\text{alg}}(n) \|\mathbf{A}\|_q \|\mathbf{B}\|_q \varepsilon + O(\varepsilon^2)$$

with

$$f_{\text{alg}}(n) = O(n^{\log_2 \gamma_{p,q}})$$

and

$$\gamma_{p,q} = \left\| \sum_{i=1}^7 \|\mathbf{L}_i\|_q^* \|\mathbf{R}_i\|_q^* |p_{i,k}| \right\|_p$$

- ▶ Bound holds for any norm $\|\cdot\|$ and related dual norm $\|\cdot\|^*$ over \mathbb{R}^{n^2} :
- ▶ arbitrary $\|\cdot\|_p$ on LHS $\|\cdot\|_q$ on RHS
- ▶ **no log** whenever alg performs a Matrix Multiplication

Search for algorithms

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

Optimizing the growth factor in norm 2

Weaker majorations (to optimize a smoother function):

$$\underbrace{\max_{k=1..4} \sum_{i=1}^7 \|L_i\|_2 \|R_i\|_2 |p_{i,k}|}_{\gamma_{\infty,2}} \leq \underbrace{\sum_{i=1}^7 \|L_i\|_2 \|R_i\|_2 \|P^T_i\|_2}_{\gamma_2}$$

Optimizing the growth factor in norm 2

Weaker majorations (to optimize a smoother function):

$$\underbrace{\max_{k=1..4} \sum_{i=1}^7 \|L_i\|_2 \|R_i\|_2 |p_{i,k}|}_{\gamma_{\infty,2}} \leq \underbrace{\sum_{i=1}^7 \|L_i\|_2 \|R_i\|_2 \|P^T_i\|_2}_{\gamma_2}$$

Optimization program over all isotropies (U, V, W)

Numerical optimization: approximate solution showing a structure: $\|L\| = \|R\| = \|P\|$

Exact optimization: solving a polynomial system with Gröbner basis

- ▶ 1 isotropy = 3×4 variables (up to permutations and rotations)
 \rightsquigarrow too hard to solve
- ▶ Projective property of tensor: matrices of $\det 1 \rightsquigarrow 9$ free variables
- ▶ Optimize on the subvariety where $\|L\| = \|R\| = \|P\|$
 $\rightsquigarrow 3$ free variables

Optimizing the growth factor in norm 2

Weaker majorations (to optimize a smoother function):

$$\underbrace{\max_{k=1..4} \sum_{i=1}^7 \|L_i\|_2 \|R_i\|_2 |p_{i,k}|}_{\gamma_{\infty,2}} \leq \underbrace{\sum_{i=1}^7 \|L_i\|_2 \|R_i\|_2 \|P^T_i\|_2}_{\gamma_2}$$

Proposition

The minimum of γ_2 on the subvariety $\|L\| = \|R\| = \|P\|$ is $\gamma_2^* = \frac{16}{\sqrt{3}} + \frac{4}{\sqrt{2}} \approx 12.066$,

reached by the Algorithm $\mathcal{A}_{2,2,2}^*$ =

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

Optimizing the growth factor in norm 2

Weaker majorations (to optimize a smoother function):

$$\underbrace{\max_{k=1..4} \sum_{i=1}^7 \|L_i\|_2 \|R_i\|_2 |p_{i,k}|}_{\gamma_{\infty,2}} \leq \underbrace{\sum_{i=1}^7 \|L_i\|_2 \|R_i\|_2 \|P^T_i\|_2}_{\gamma_2}$$

Proposition

The minimum of γ_2

$$\text{is } \gamma_2^* = \frac{16}{\sqrt{3}} + \frac{4}{\sqrt{2}} \approx 12.066,$$

reached by the Algorithm $\mathcal{A}_{2,2,2}^* =$

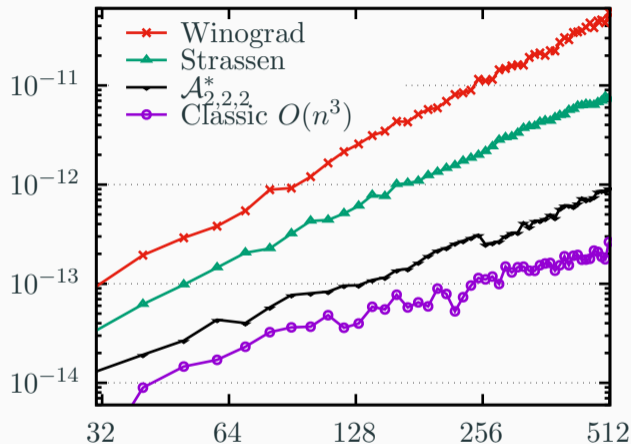
$$\begin{bmatrix} \frac{\sqrt{3}}{2} & \frac{1}{2} & \frac{1}{2} & \frac{\sqrt{3}}{6} \\ 0 & 0 & 1 & -\frac{\sqrt{3}}{3} \\ 0 & 1 & 0 & \frac{\sqrt{3}}{3} \\ 0 & 0 & 0 & -\frac{2}{\sqrt{3}} \\ -\frac{\sqrt{3}}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{\sqrt{3}}{2} \\ -\frac{\sqrt{3}}{2} & -\frac{1}{2}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

A new algorithm with improved accuracy

Algorithm	$\gamma_{\infty, \infty}$	$\gamma_{\infty, 2}$	γ_2
Winograd'70	18	8	17.854
Strassen'69	12	6.829	14.829
$\mathcal{A}_{2,2,2}^*$	17.475	5.966	12.066

A new algorithm with improved accuracy

Algorithm	$\gamma_{\infty, \infty}$	$\gamma_{\infty, 2}$	γ_2
Winograd'70	18	8	17.854
Strassen'69	12	6.829	14.829
$\mathcal{A}_{2,2,2}^*$	17.475	5.966	12.066



Coefficients sampled from a normal distribution

**Improving speed while preserving
accuracy**

Improving speed while preserving accuracy

- ▶ Sparsify the L,R,P representation
 - ◇ via isotropies
 - ◇ via alternative basis
- ▶ Schedule operations into a Straight line program (SLP)
 - ◇ factor out as many common sub-expressions as possible
- ▶ Optimize the memory footprint of the SLP
 - ◇ Pebble games

Sparsify the L,R,P representation

Theorem

The 2-norm of a matrix is invariant by orthogonal transformation

Sparsify the L,R,P representation

Theorem

The 2-norm of a matrix is invariant by orthogonal transformation

- ▶ Explore the sub-space of orthogonal isotropies for sparse L,R,P algorithms

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

$$\left. \begin{array}{l} \\ \\ \end{array} \right) \diamond (\mathbf{U}, \mathbf{V}, \mathbf{W}) \in \text{ORTHOGONAL}$$
$$[\mathbf{L} \cdot (\mathbf{V} \otimes \mathbf{U}^{-\top}); \mathbf{R} \cdot (\mathbf{W} \otimes \mathbf{V}^{-\top}); (\mathbf{U}^{\top} \otimes \mathbf{W}^{-1}) \cdot \mathbf{P}]$$

Factor with alternative basis

- ▶ Reduces back to the optimal recursive **12 ADD** for L^Φ, R^Ψ, P^ν
 $\rightsquigarrow 5n^{\log_2 7} + O(n^2 \log n)$ complexity bound
- ▶ No longer a Matrix Multiplication alg.
 \rightsquigarrow apply recursive change of basis, Φ, Ψ, ν in additional $O(n^2 \log n)$
- ▶ Accuracy?

$$\underbrace{\begin{bmatrix} 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}}_{L^\Phi} \times \underbrace{\begin{bmatrix} 0 & 0 & 0 & \frac{2}{\sqrt{3}} \\ 0 & 1 & 0 & \frac{\sqrt{3}}{3} \\ 0 & 0 & 1 & -\frac{\sqrt{3}}{3} \\ -\frac{\sqrt{3}}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix}}_{\Phi} ; \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 1 \end{bmatrix}}_{R^\Psi} \times \underbrace{\begin{bmatrix} 0 & \frac{2}{\sqrt{3}} & 0 & 0 \\ 1 & -\frac{\sqrt{3}}{3} & 0 & 0 \\ 0 & \frac{\sqrt{3}}{3} & 0 & -1 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} & -\frac{1}{2} \end{bmatrix}}_{\Psi} ; \underbrace{\begin{bmatrix} 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}^T}_{P^\nu} \times \underbrace{\begin{bmatrix} -\frac{2}{\sqrt{3}} & 0 & 0 & 0 \\ \frac{\sqrt{3}}{3} & -1 & 0 & 0 \\ -\frac{\sqrt{3}}{3} & 0 & -1 & 0 \\ \frac{\sqrt{3}}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix}^T}_{\nu}$$

Accuracy of the alternative basis variant

- ▶ [Schwartz, Toledo, Vaknim and Wiernik'24]: claims accuracy is preserved $O(n^{\log_2 \gamma} \log n)$
- ▶ Unable to verify the claim, instead best proven bound

$$\gamma^{(AltBasisMatMul)} = \gamma^{(AltBasisTensor)} \|\Phi\|_q \|\Psi\|_q \|\nu^T\|_p$$

Accuracy of the alternative basis variant

- ▶ [Schwartz, Toledo, Vaknim and Wiernik'24]: claims **accuracy is preserved** $O(n^{\log_2 \gamma} \log n)$
- ▶ **Unable to verify the claim**, instead best proven bound

$$\gamma^{(AltBasisMatMul)} = \gamma^{(AltBasisTensor)} \|\Phi\|_q \|\Psi\|_q \|\nu^T\|_p$$

		Standard	Alternative basis
Strassen	$\gamma_{\infty, \infty}$	12	80
	$f_{\text{alg}}(n)$	$10.6n^{3.69}$	$(1 + 14 \log n)n^{6.33}$
Winograd	$\gamma_{\infty, \infty}$	18	270
	$f_{\text{alg}}(n)$	$12.25n^{4.17}$	$(1 + 15 \log n)n^{8.08}$
$\mathcal{A}_{2,2,2}^*$	$\gamma_{\infty, \infty}$	17.48	184
	$f_{\text{alg}}(n)$	$17.94n^{4.13}$	$(1 + 20 \log n)n^{7.52}$

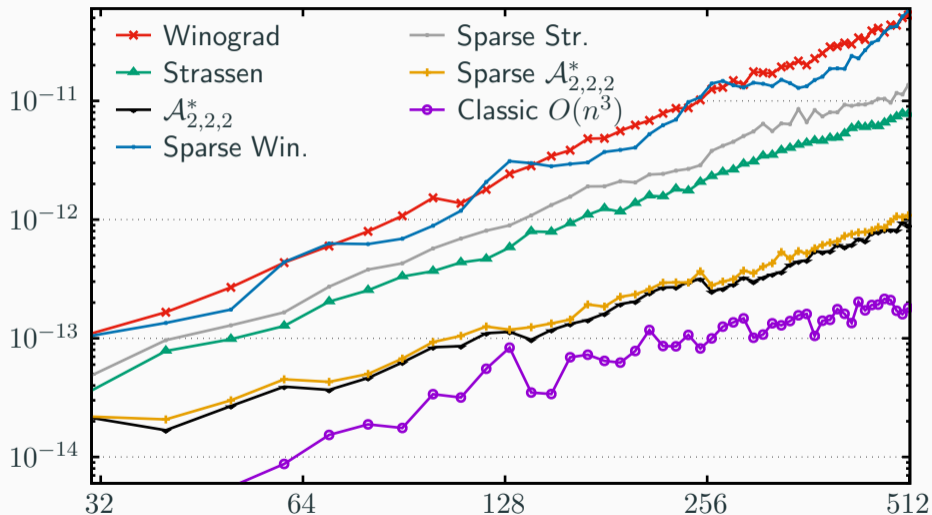
Accuracy of the alternative basis variant

- ▶ [Schwartz, Toledo, Vaknim and Wiernik'24]: claims **accuracy is preserved** $O(n^{\log_2 \gamma} \log n)$
- ▶ **Unable to verify the claim**, instead best proven bound

$$\gamma^{(\text{AltBasisMatMul})} = \gamma^{(\text{AltBasisTensor})} \|\Phi\|_q \|\Psi\|_q \|\nu^T\|_p$$

		Standard	Alternative basis
Strassen	$\gamma_{\infty,2}$	6.83	84
	$f_{\text{alg}}(n)$	$10.38n^{2.78}$	$(1 + 14 \log n)n^{6.40}$
Winograd	$\gamma_{\infty,2}$	8	108
	$f_{\text{alg}}(n)$	$12.43n^3$	$(1 + 15 \log n)n^{6.76}$
$\mathcal{A}_{2,2,2}^*$	$\gamma_{\infty,2}$	5.97	95.6
	$f_{\text{alg}}(n)$	$17.74n^{2.58}$	$(1 + 20 \log n)n^{6.58}$

Accuracy of Alternative Basis variants



Beyond Strassen's algorithm

$4 \times 4 \times 4$ Matrix Multiplication schemes: complexity

$N \times N \times N$ Algorithm	N	field	# Mul	# Add	Recursively
Classic $O(n^3)$	2	any	8	4	$2n^3$
[Strassen'69]	2	any	7	18	$7n^{2.8074}$
[Winograd'70]	2	any	7	15	$6n^{2.8074}$

$4 \times 4 \times 4$ Matrix Multiplication schemes: complexity

$N \times N \times N$ Algorithm	N	field	# Mul	# Add	Recursively
Classic $O(n^3)$	2	any	8	4	$2n^3$
[Strassen'69]	2	any	7	18	$7n^{2.8074}$
[Winograd'70]	2	any	7	15	$6n^{2.8074}$
Winograd ^{⊗2}	4	any	49	165	$6n^{2.8074}$

$4 \times 4 \times 4$ Matrix Multiplication schemes: complexity

$N \times N \times N$ Algorithm	N	field	# Mul	# Add	Recursively
Classic $O(n^3)$	2	any	8	4	$2n^3$
[Strassen'69]	2	any	7	18	$7n^{2.8074}$
[Winograd'70]	2	any	7	15	$6n^{2.8074}$
Winograd ^{⊗2}	4	any	49	165	$6n^{2.8074}$
[Fawzi et al. 22]	4	char= 2	47		$O(n^{2.7773})$

$4 \times 4 \times 4$ Matrix Multiplication schemes: complexity

$N \times N \times N$ Algorithm	N	field	# Mul	# Add	Recursively
Classic $O(n^3)$	2	any	8	4	$2n^3$
[Strassen'69]	2	any	7	18	$7n^{2.8074}$
[Winograd'70]	2	any	7	15	$6n^{2.8074}$
Winograd ^{⊗2}	4	any	49	165	$6n^{2.8074}$
[Fawzi et al. 22]	4	char= 2	47		$O(n^{2.7773})$
[Novikov et al. 25]	4	\mathbb{C}	48		$O(n^{2.7925})$

$4 \times 4 \times 4$ Matrix Multiplication schemes: complexity

$N \times N \times N$ Algorithm	N	field	# Mul	# Add	Recursively
Classic $O(n^3)$	2	any	8	4	$2n^3$
[Strassen'69]	2	any	7	18	$7n^{2.8074}$
[Winograd'70]	2	any	7	15	$6n^{2.8074}$
Winograd ^{⊗2}	4	any	49	165	$6n^{2.8074}$
[Fawzi et al. 22]	4	char= 2	47		$O(n^{2.7773})$
[Novikov et al. 25]	4	\mathbb{C}	48		$O(n^{2.7925})$
$\mathcal{A}_{4,4,4}^*$	4	char \neq 2	48	292	$10.125n^{2.7925}$

$4 \times 4 \times 4$ Matrix Multiplication schemes: complexity

$N \times N \times N$ Algorithm	N	field	# Mul	# Add	Recursively	Alt-basis
Classic $O(n^3)$	2	any	8	4	$2n^3$	
[Strassen'69]	2	any	7	18	$7n^{2.8074}$	$5n^{2.8074}$
[Winograd'70]	2	any	7	15	$6n^{2.8074}$	$5n^{2.8074}$
Winograd ^{⊗2}	4	any	49	165	$6n^{2.8074}$	
[Fawzi et al. 22]	4	char= 2	47		$O(n^{2.7773})$	
[Novikov et al. 25]	4	\mathbb{C}	48		$O(n^{2.7925})$	
$\mathcal{A}_{4,4,4}^*$	4	char \neq 2	48	292	$10.125n^{2.7925}$	$7n^{2.7925}$

$4 \times 4 \times 4$ Matrix Multiplication schemes: accuracy

$N \times N \times N$ Algorithm	N	# Mul	Recursively	$\gamma_{\infty,2}$	$f_{\text{alg}}(n)$
Classic $O(n^3)$	2	8	$2n^3$	2	$O(n)$
[Strassen'69]	2	7	$7n^{2.8074}$	6.83	$O(n^{2.7716})$
[Winograd'70]	2	7	$6n^{2.8074}$	8	$O(n^3)$
$\mathcal{A}_{2,2,2}^*$	2	7	$13n^{2.8074}$	5.97	$O(n^{2.58})$

$4 \times 4 \times 4$ Matrix Multiplication schemes: accuracy

$N \times N \times N$ Algorithm	N	# Mul	Recursively	$\gamma_{\infty,2}$	$f_{\text{alg}}(n)$
Classic $O(n^3)$	2	8	$2n^3$	2	$O(n)$
[Strassen'69]	2	7	$7n^{2.8074}$	6.83	$O(n^{2.7716})$
[Winograd'70]	2	7	$6n^{2.8074}$	8	$O(n^3)$
$\mathcal{A}_{2,2,2}^*$	2	7	$13n^{2.8074}$	5.97	$O(n^{2.58})$
Strassen \otimes^2	4	49	$7n^{2.8074}$	46.65	$O(n^{2.7716})$

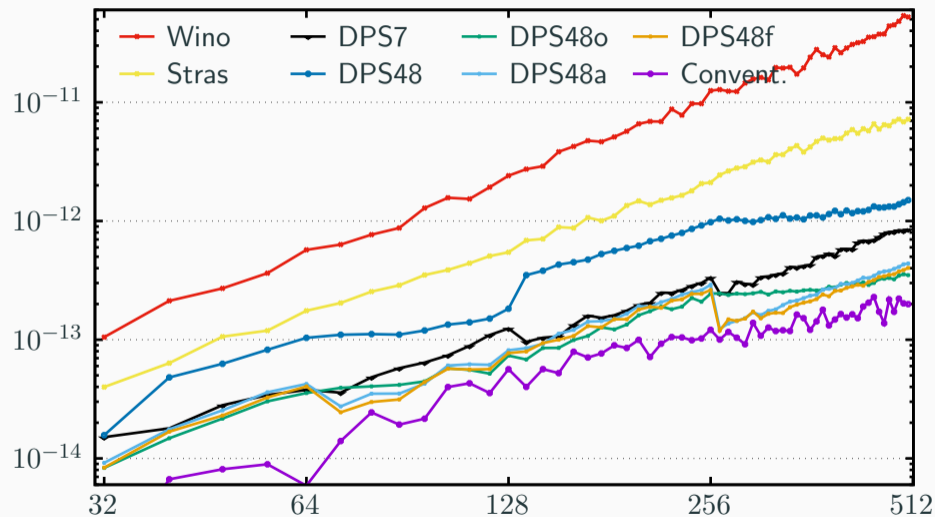
$4 \times 4 \times 4$ Matrix Multiplication schemes: accuracy

$N \times N \times N$ Algorithm	N	# Mul	Recursively	$\gamma_{\infty,2}$	$f_{\text{alg}}(n)$
Classic $O(n^3)$	2	8	$2n^3$	2	$O(n)$
[Strassen'69]	2	7	$7n^{2.8074}$	6.83	$O(n^{2.7716})$
[Winograd'70]	2	7	$6n^{2.8074}$	8	$O(n^3)$
$\mathcal{A}_{2,2,2}^*$	2	7	$13n^{2.8074}$	5.97	$O(n^{2.58})$
Strassen $^{\otimes 2}$	4	49	$7n^{2.8074}$	46.65	$O(n^{2.7716})$
$\mathcal{A}_{4,4,4}$	4	48	$10.25n^{2.7925}$	45.65	$O(n^{2.7561})$

$4 \times 4 \times 4$ Matrix Multiplication schemes: accuracy

$N \times N \times N$ Algorithm	N	# Mul	Recursively	$\gamma_{\infty,2}$	$f_{\text{alg}}(n)$
Classic $O(n^3)$	2	8	$2n^3$	2	$O(n)$
[Strassen'69]	2	7	$7n^{2.8074}$	6.83	$O(n^{2.7716})$
[Winograd'70]	2	7	$6n^{2.8074}$	8	$O(n^3)$
$\mathcal{A}_{2,2,2}^*$	2	7	$13n^{2.8074}$	5.97	$O(n^{2.58})$
Strassen $^{\otimes 2}$	4	49	$7n^{2.8074}$	46.65	$O(n^{2.7716})$
$\mathcal{A}_{4,4,4}$	4	48	$10.25n^{2.7925}$	45.65	$O(n^{2.7561})$
$\mathcal{A}_{4,4,4}^*$	4	48	$12.125n^{2.7925}$	27.314	$O(n^{2.3857})$

Accuracy of $\langle 4 \times 4 \times 4 : 48 \rangle$ variants



DPS48 fastest
DPS48o best accuracy
DPS48a $(48 \times 16)(16 \times 16)$
DPS48f $(48 \times 47)(47 \times 16)$

Find the best operation schedule from a matrix representation ?

Common Subexpression Elimination

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

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

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

$$w_2 = p_5 + p_1 + p_6 \quad w_1 = p_7 + p_6 \quad w_3 = w_2 - p_2 \quad w_5 = \frac{p_4 + w_2}{2}$$

$$c_{12} = p_1 - p_3 - w_5 \quad c_{21} = w_3 - w_5 \quad c_{22} = \sqrt{3}w_5$$

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

Find the best operation schedule from a matrix representation ?

Common Subexpression Elimination

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

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

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

$$w_2 = p_5 + p_1 + p_6 \quad w_1 = p_7 + p_6 \quad w_3 = w_2 - p_2 \quad w_5 = \frac{p_4 + w_2}{2}$$

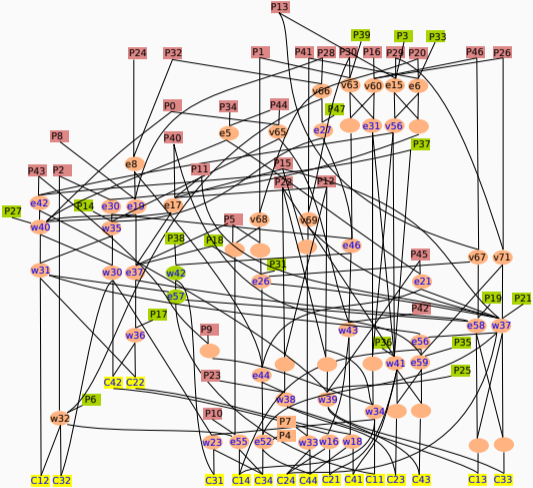
$$c_{12} = p_1 - p_3 - w_5 \quad c_{21} = w_3 - w_5 \quad c_{22} = \sqrt{3} w_5$$

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

$2 \times 2 \times 2$: ADD : 45 \rightsquigarrow 24 scalar MUL: 57 \rightsquigarrow 12,

$4 \times 4 \times 4$: ADD : 960 \rightsquigarrow 258 scalar MUL: 280 \rightsquigarrow 38,

Memory placement



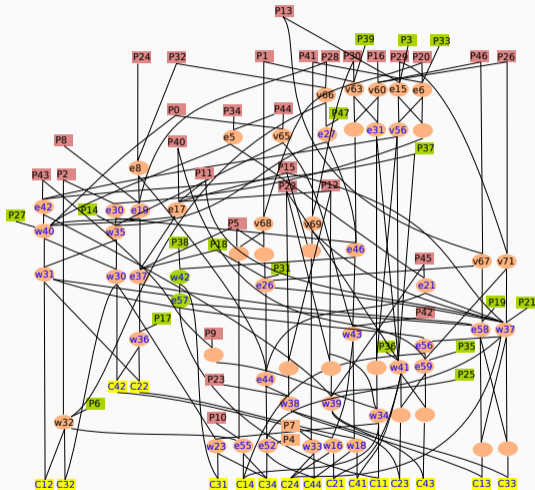
Optimizing memory placement

Pebble games

with randomized exhaustive search

DAG of the post-additions in $\langle 4 \times 4 \times 4 : 48 \mid 341 \rangle$

Memory placement



Optimizing memory placement

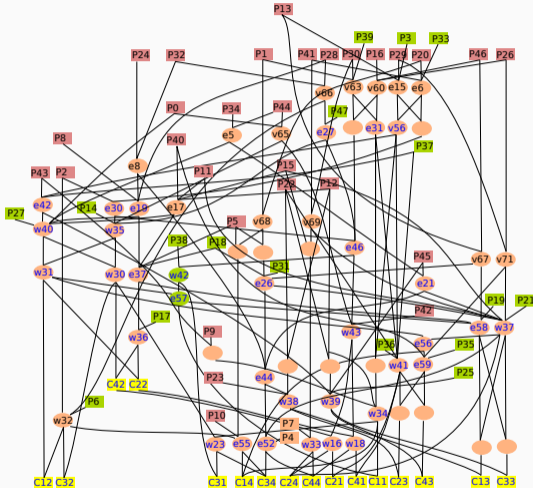
Pebble games

with randomized exhaustive search

- ▶ $\langle 4 \times 4 \times 4 : 48 \mid 296 \rangle$
 \rightsquigarrow 38 temp \rightsquigarrow $2.53n^2$ overhead
- ▶ $\langle 4 \times 4 \times 4 : 48 \mid 356 \rangle$ accurate
 \rightsquigarrow 36 temp. \rightsquigarrow $2.4n^2$ overhead

DAG of the post-additions in $\langle 4 \times 4 \times 4 : 48 \mid 341 \rangle$

Memory placement



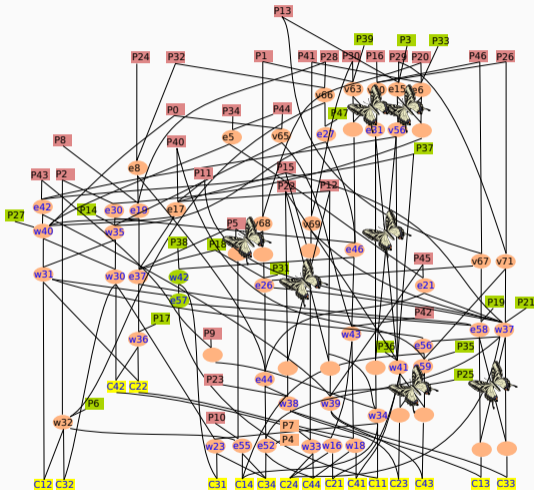
DAG of the post-additions in $\langle 4 \times 4 \times 4 : 48 \mid 341 \rangle$

Optimizing memory accesses

Short supply chain

- ▶ moving data production closer to its consumption

Memory placement



DAG of the post-additions in $\langle 4 \times 4 \times 4 : 48 \mid 341 \rangle$

Optimizing memory accesses

Short supply chain

- ▶ moving data production closer to its consumption

Butterflies

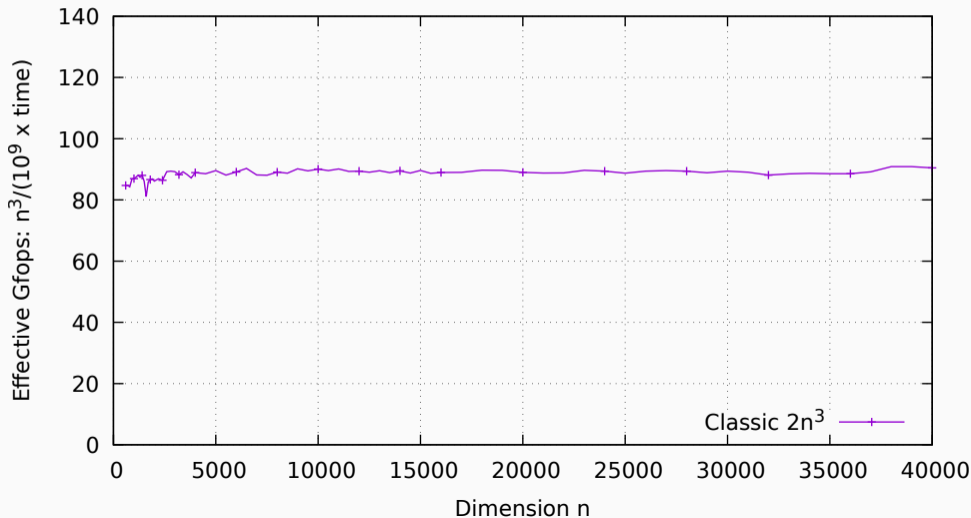


$$\begin{aligned} x &= a+b; \\ y &= a-b \end{aligned} \quad \rightsquigarrow \quad \begin{aligned} x, y &= a+b, a-b \end{aligned}$$

- ▶ 2 operations for 2 load 2 stores
- ▶ in-place: $a, b = a+b, a-b$;
- ▶ done at the SIMD level

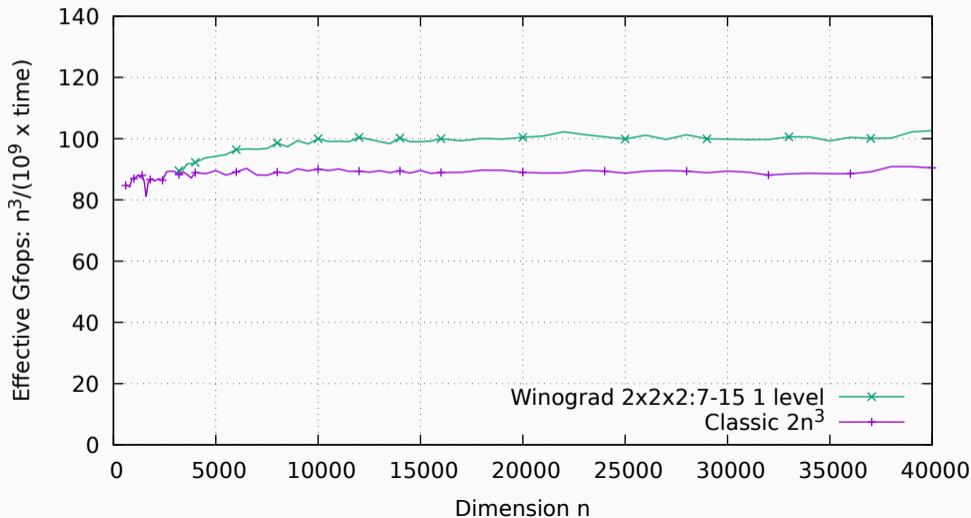
Computation speed (fflas-ffpack + OpenBLAS)

Matrix Multiplication in ZRing<double> on an Xeon Gold 6330



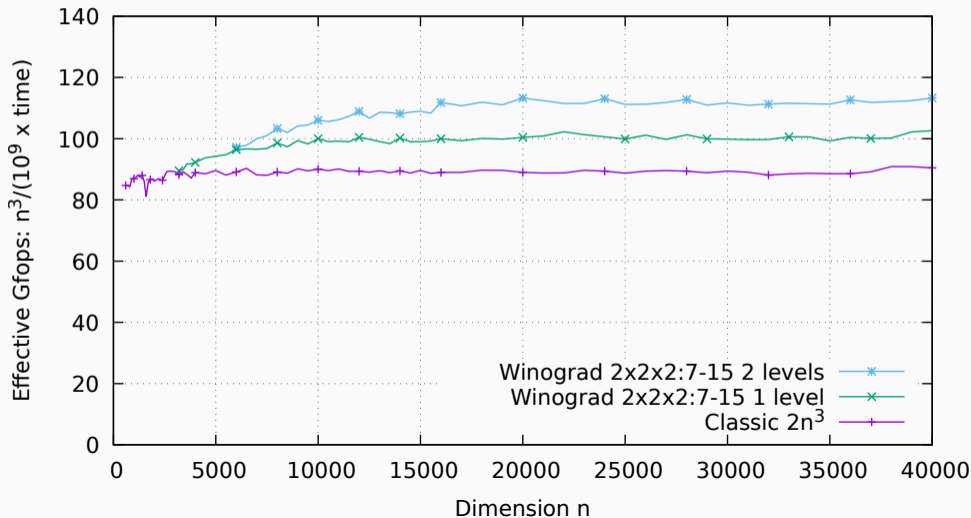
Computation speed (fflas-ffpack + OpenBLAS)

Matrix Multiplication in ZRing<double> on an Xeon Gold 6330



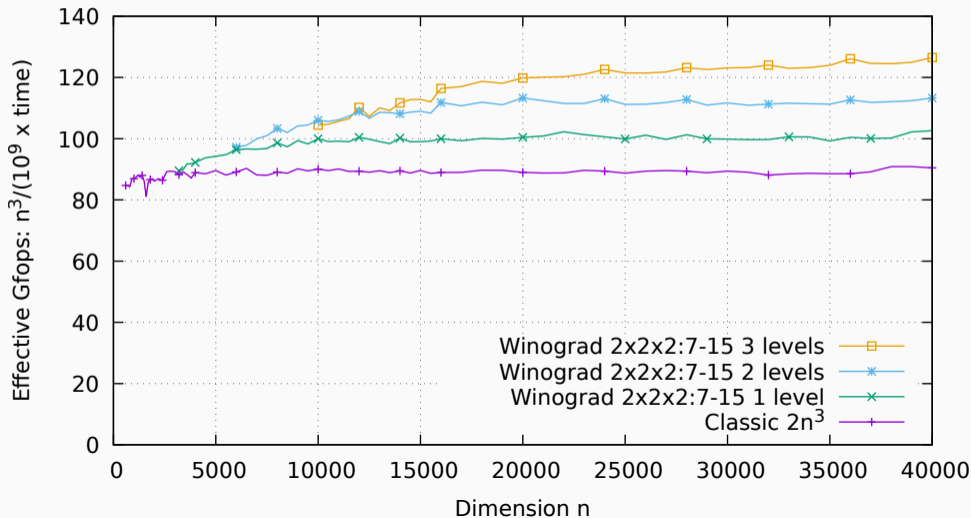
Computation speed (fflas-ffpack + OpenBLAS)

Matrix Multiplication in ZRing<double> on an Xeon Gold 6330



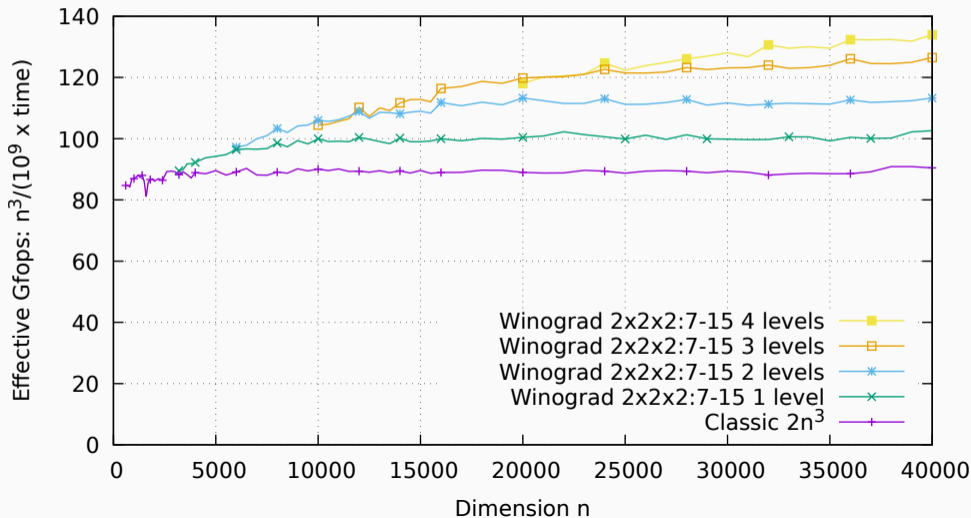
Computation speed (fflas-ffpack + OpenBLAS)

Matrix Multiplication in ZRing<double> on an Xeon Gold 6330



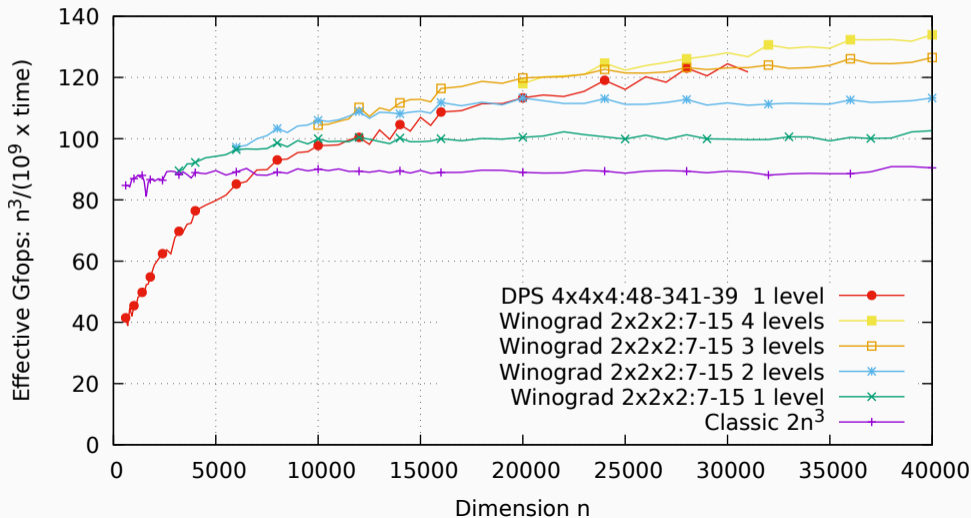
Computation speed (fflas-ffpack + OpenBLAS)

Matrix Multiplication in ZRing<double> on an Xeon Gold 6330



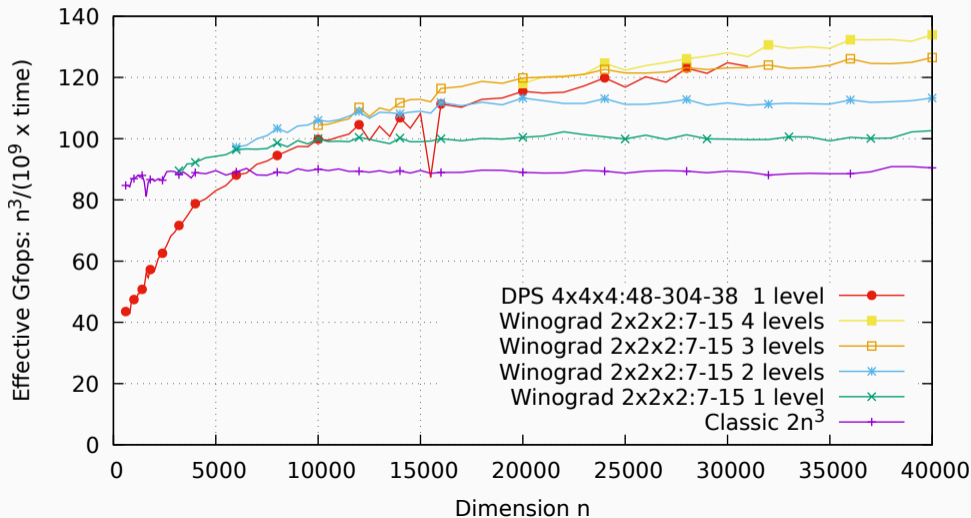
Computation speed (fflas-ffpack + OpenBLAS)

Matrix Multiplication in ZRing<double> on an Xeon Gold 6330



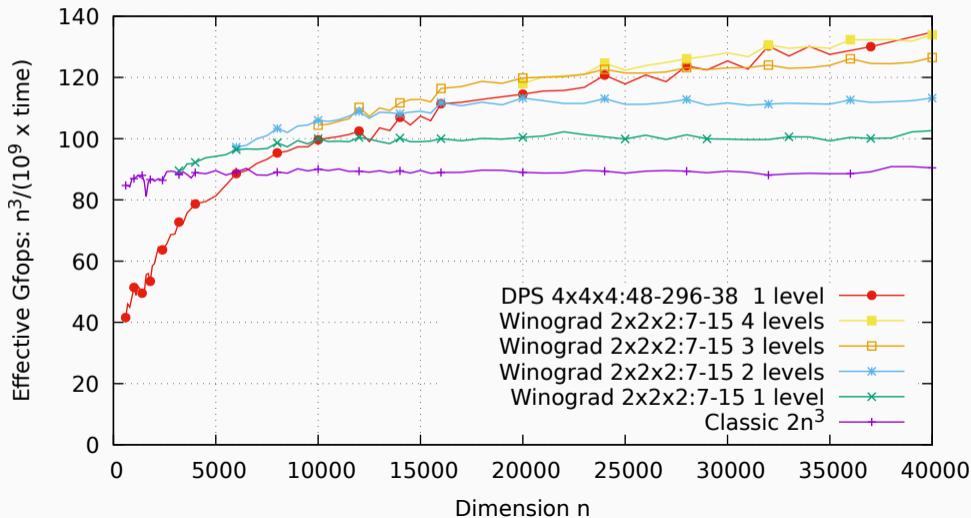
Computation speed (fflas-ffpack + OpenBLAS)

Matrix Multiplication in ZRing<double> on an Xeon Gold 6330



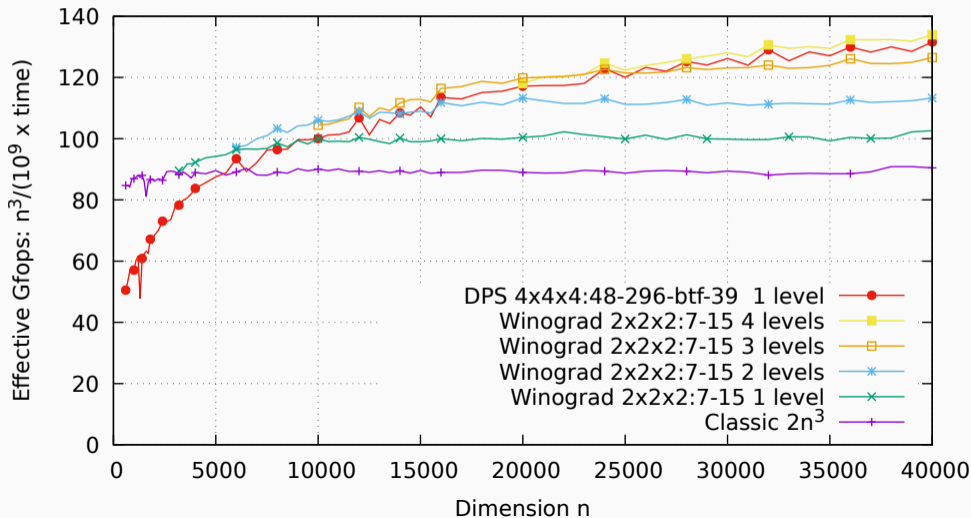
Computation speed (fflas-ffpack + OpenBLAS)

Matrix Multiplication in ZRing<double> on an Xeon Gold 6330



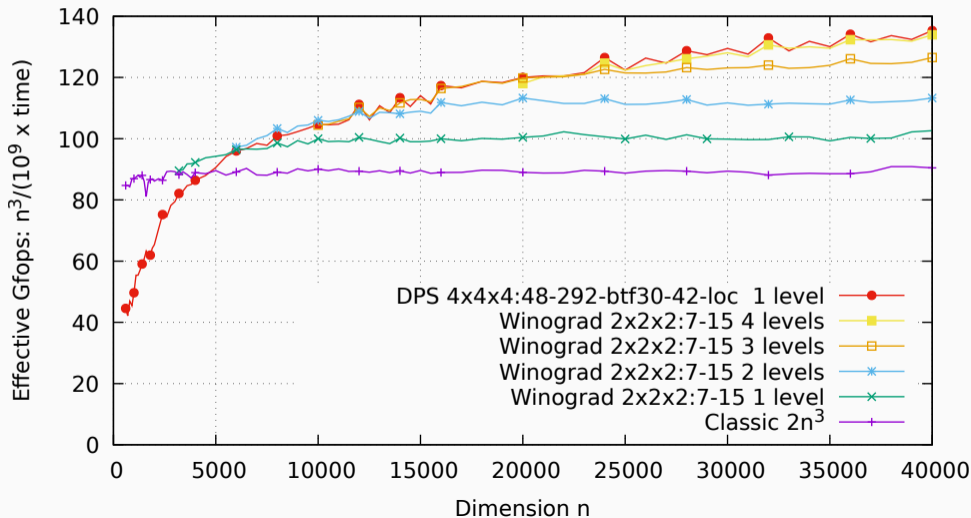
Computation speed (fflas-ffpack + OpenBLAS)

Matrix Multiplication in ZRing<double> on an Xeon Gold 6330



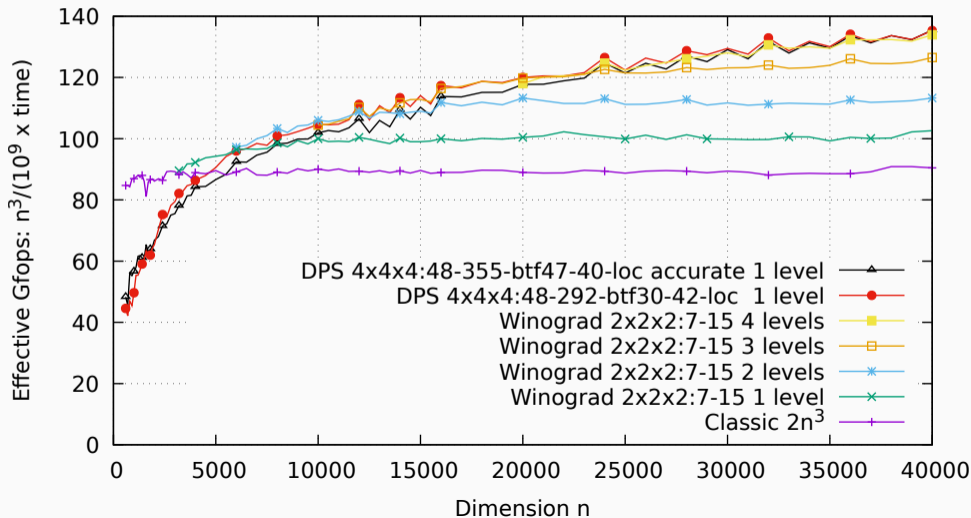
Computation speed (fflas-ffpack + OpenBLAS)

Matrix Multiplication in ZRing<double> on an Xeon Gold 6330



Computation speed (fflas-ffpack + OpenBLAS)

Matrix Multiplication in ZRing<double> on an Xeon Gold 6330



Thank you

$\mathcal{A}_{2,2,2}^*$ in [D.-P.-S. 2026]

Towards automated generation of fast and accurate algorithms for recursive matrix multiplication.
J. Symbolic Comp. (134). 2026. <https://hal.science/hal-04995684v1>

$\mathcal{A}_{4,4,4}^*$ in [D.-P.-S. 2025]

A non-commutative algorithm for multiplying 4x4 matrices using 48 non-complex multiplications.
<https://hal.science/hal-05112145>

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>.
- ▶ <https://github.com/jgdumas/plinopt>.

Rational approximations of $\mathcal{A}_{2,2,2}^*$

$$\mathcal{A}^* = \operatorname{argmin}(\gamma_2(\mathcal{A}(x, y))) = \mathcal{A}\left(\frac{\sqrt{2}}{\sqrt[4]{3}}, -\frac{1}{2}\right)$$

Rational approximations of $\mathcal{A}_{2,2,2}^*$

$$\mathcal{A}^* = \operatorname{argmin}(\gamma_2(\mathcal{A}(x, y))) = \mathcal{A}\left(\frac{\sqrt{2}}{\sqrt[4]{3}}, -\frac{1}{2}\right)$$

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 $\gamma_{2,1,\infty} \approx 6.046$

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

Rational approximations of $\mathcal{A}_{2,2,2}^*$

$$\mathcal{A}^* = \operatorname{argmin}(\gamma_2(\mathcal{A}(x, y))) = \mathcal{A}\left(\frac{\sqrt{2}}{\sqrt[4]{3}}, -\frac{1}{2}\right)$$

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 $\gamma_{2,1,\infty} \approx 6.046$

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

▶ $\frac{\sqrt{2}}{\sqrt[4]{3}} \approx \frac{14}{13}$

$\rightsquigarrow \gamma_{2,1} \approx 12.0662$

and $\gamma_{2,1,\infty} \approx 6.000043$

▶ ...

Accuracy of rational approximations

