

# A non-commutative algorithm for multiplying 7 x 7 matrices using 250 multiplications

Alexandre Sedoglavic

## ► To cite this version:

Alexandre Sedoglavic. A non-commutative algorithm for multiplying 7 x 7 matrices using 250 multiplications. 2017. hal-01572046v1

## HAL Id: hal-01572046 https://hal.science/hal-01572046v1

Preprint submitted on 4 Aug 2017 (v1), last revised 18 Jan 2019 (v3)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Public Domain

## A non-commutative algorithm for multiplying $7 \times 7$ matrices using 250 multiplications

Alexandre.Sedoglavic@univ-lille.fr

August 4, 2017

### Abstract

We present a non-commutative algorithm for multiplying  $7\times7$  matrices using 250 multiplications.

### 1 Introduction

We use the Strassen's matrix multiplication algorithm [7] to *divide* the  $7 \times 7$  matrix multiplication problem into smaller sub-problems; the use of Smirnov's matrix multiplication algorithms [5, 6] allows to *conquer* a new upper bound on the number of necessary non-commutative multiplications.

Hence, we present a scheme that evaluate the product  $P = N \cdot M$ :

$$N = \begin{pmatrix} n_{11} & \cdots & n_{17} \\ \vdots & & \vdots \\ n_{71} & \cdots & n_{77} \end{pmatrix}, \ M = \begin{pmatrix} m_{11} & \cdots & m_{17} \\ \vdots & & \vdots \\ m_{71} & \cdots & m_{77} \end{pmatrix}, \ P = \begin{pmatrix} p_{11} & \cdots & p_{17} \\ \vdots & & \vdots \\ p_{71} & \cdots & p_{77} \end{pmatrix}, \ (1)$$

using 250 multiplications. This algorithm improves slightly the previous known upper bound 258 presented in [1] and likely obtained with the same kind of technics presented in the next section.

## 2 Divide

For any  $2 \times 2$  matrices:

$$A = (a_{ij})_{1 \le i,j \le 2}, \quad B = (b_{ij})_{1 \le i,j \le 2} \quad \text{and} \quad C = (c_{ij})_{1 \le i,j \le 2}, \tag{2}$$

V. Strassen shows in [7] that the matrix product  $C = A \cdot B$  could be computed by performing the following operations:

$$t_{1} = (a_{11} + a_{22})(b_{11} + b_{22}), \quad t_{2} = (a_{12} - a_{22})(b_{21} + b_{22}), t_{3} = (-a_{11} + a_{21})(b_{11} + b_{12}), \quad t_{4} = (a_{11} + a_{12})b_{22}, t_{5} = a_{11}(b_{12} - b_{22}), \quad t_{6} = a_{22}(-b_{11} + b_{21}), \quad t_{7} = (a_{21} + a_{22})b_{11},$$
(3)
$$\begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix} = \begin{pmatrix} t_{1} + t_{2} - t_{4} + t_{6} & t_{6} + t_{7}, \\ t_{4} + t_{5} & t_{1} + t_{3} + t_{5} - t_{7} \end{pmatrix},$$

in the considered coefficients algebra.

,

To construct our algorithm, we are going to work with the algebra of  $4 \times 4$ matrices and thus, we have to adapt our inputs P, N, M to that end. So we rewrite the matrices (1) in the following equivalent form:

$$X = \begin{pmatrix} x_{11} & x_{12} & x_{13} & 0 & x_{14} & \cdots & x_{17} \\ x_{21} & x_{22} & x_{23} & 0 & x_{24} & \cdots & x_{27} \\ x_{31} & x_{32} & x_{33} & 0 & x_{34} & \cdots & x_{37} \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ \hline x_{41} & x_{42} & x_{34} & 0 & x_{44} & \cdots & x_{34} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{71} & x_{72} & x_{77} & 0 & x_{74} & \cdots & x_{77} \end{pmatrix}, \quad X \in \{P, M, N\}, \quad (4)$$

in which we have just added a line and a column of zeros. After that padding, the product  $P = N \cdot M$  is unchanged.

**Notations 2.1** — Hence, in the sequel P (resp. M, N) designates the  $8 \times 8$  matrices defined in (4) and  $P_{ij}$  (resp.  $M_{ij}, N_{ij}$ ) designates  $4 \times 4$  matrices (e.g.  $P_{11}$ stands for the upper left submatrix of P, etc).

**Remark 2.1** — The process of pealing (removing rows and columns) the result of our computations might be better understood if we use the classical equivalence between the bilinear application  $\mathcal{B}: \mathbb{K}^{8 \times 8} \times \mathbb{K}^{8 \times 8} \mapsto \mathbb{K}^{8 \times 8}$  with indeterminates N and M that defines the matrix multiplication  $\mathcal{B}(N, M) = N \cdot M = P$ and the trilinear form  $\mathbb{K}^{8\times8} \times \mathbb{K}^{8\times8} \times \mathbb{K}^{8\times8} \mapsto \mathbb{K}$  with indeterminates N, Mand P defined by Trace  $(N \cdot M \cdot P)$  (see [4, § 2.5.2] and [2] for a complete description of this equivalence).

Computationally, this equivalence induces that the following relation holds:

$$\operatorname{Trace}(N \cdot M \cdot P) = \operatorname{Trace}((N_{11} + N_{22}) \cdot (M_{11} + M_{22}) \cdot (P_{11} + P_{22}))$$
(5a)

+ Trace 
$$((N_{12} - N_{22}) \cdot (M_{21} + M_{22}) \cdot P_{11})$$
 (5b)

+ Trace
$$((-N_{11} + N_{21}) \cdot (M_{11} + M_{12}) \cdot P_{22})$$
 (5c)

+ Trace
$$((N_{11} + N_{12}) \cdot M_{22} \cdot (-P_{11} + P_{21}))$$
 (5d)

+ Trace
$$(N_{11} \cdot (M_{12} - M_{22}) \cdot (P_{21} + P_{22}))$$
 (5e)

+ Trace
$$(N_{22} \cdot (-M_{11} + M_{21}) \cdot (P_{11} + P_{12}))$$
 (5f)

+ Trace 
$$(N_{21} + N_{22}) \cdot M_{11} \cdot (P_{12} - P_{22}))$$
 (5g)

and as the bilinear application and the trilinear form are equivalent, one could retrieve directly the algorithm from this last form. Our original problem is now divided in 7 lower dimensional subproblems encoded by trilinear forms. In the next section, we presents the algorithms used to compute the matrix products (5a-5g).

#### 3 Conquer

The first summand (5a) involves unstructured  $4 \times 4$  matrix multiplication that could be computed using Strassen's algorithm and could be done with  $7^2$  multiplications. Before studying the other summands, we emphasise the following trivial remarks:

**Remarks 3.1** — As we consider some  $4 \times 4$  matrices with zero last column and/or row, recall that the product of two  $4 \times 4$  matrices  $(x_{ij})_{1 \le i,j \le 4}$  and matrix  $(y_{ij})_{1 < i,j < 4}$  is equivalent to the product of the:

- $4 \times 3 \text{ matrix } (x_{ij})_{1 \le i \le 4, 1 \le j \le 3}$  with the  $3 \times 4 \text{ matrix } (y_{ij})_{1 \le i \le 3, 1 \le, j \le 4}$  when for  $1 \le i \le 4$  we have  $y_{i4} = 0$  (zero last row);
- $3 \times 4 \text{ matrix } (x_{ij})_{1 \leq i \leq 3, 1 \leq j \leq 4}$  with the  $4 \times 3 \text{ matrix } (y_{ij})_{1 \leq i \leq 3, 1 \leq , j \leq 4}$  when for  $1 \leq j \leq 4$  we have  $y_{4j} = 0$  (zero last column).

Let us now review the matrices involved in the summands (5b-5g).

**Remarks 3.2** — We notice that:

- the last row and column of  $X_{11}$  are only composed by zeros;
- $X_{22}$  is a  $4 \times 4$  matrix;
- the last column of  $-X_{11} + X_{21}$  is only composed by zeros;
- the last line of  $X_{11} + X_{12}$  is only composed by zeros;
- $X_{12} X_{22}$  and  $X_{21} X_{22}$  are  $4 \times 4$  matrices without zero row or column.

Hence, taking into account Remarks 3.1 and 3.2, the summand:

- (5b) involves the product of a  $3 \times 4$  by a  $4 \times 3$  by a  $3 \times 3$  matrix;
- (5c) involves the product of  $4 \times 3$  by  $3 \times 4$  by  $4 \times 4$  matrix;
- (5d) involves the product of  $3 \times 4$  by  $4 \times 4$  by  $4 \times 3$  matrix;
- (5e) involves the product of  $3 \times 3$  by  $3 \times 4$  by  $4 \times 3$  matrix;
- (5f) involves the product of  $4 \times 4$  by  $4 \times 3$  by  $3 \times 4$  matrix;
- (5g) involves the product of  $4 \times 3$  by  $3 \times 3$  by  $3 \times 4$  matrix.

**Remark 3.3** — We rely our construction on the representation of matrix multiplication algorithm by trilinear forms and the underlying tensor representation because, as quoted in [3, § 4.6.4 p. 507]:

"For example, a normal scheme for evaluating an  $m \times n$  times  $n \times s$ matrix product implies the existence of a normal scheme to evaluate an  $n \times s$  times  $s \times m$  matrix product using the same number of chain multiplications."

This is exactly what we are using in the sequel. To do so and in order to evaluate the summands (5b-5g), we introduce the following notations:

**Notation 3.1** — For any matrices U, V and W of size  $u \times v, v \times w$  and  $w \times u$ , we denote by  $\langle u, v, w \rangle$  the known supremum on the multiplication necessary for computing Trace  $(U \cdot V \cdot W)$  (that is the number of multiplication used by the best known algorithm allowing to compute  $U \cdot V$  (a.k.a. tensor rank)).

Using this notation, we see that:

- (5b) can be computed using  $\langle 3, 4, 3 \rangle$  multiplications;
- (5c) can be computed using  $\langle 4, 3, 4 \rangle$  multiplications;
- (5d) can be computed using  $\langle 3, 4, 4 \rangle$  multiplications;
- (5e) can be computed using  $\langle 3, 3, 4 \rangle$  multiplications;
- (5f) can be computed using  $\langle 4, 4, 3 \rangle$  multiplications;
- (5g) can be computed using  $\langle 4, 3, 3 \rangle$  multiplications.

The remark 3.3 is a direct consequence of the Trace property:

$$\operatorname{Trace}\left(U\cdot V\cdot W\right) = \operatorname{Trace}\left(W\cdot U\cdot V\right) = \operatorname{Trace}\left(V\cdot W\cdot U\right).$$
(6)

To be more precise, we recall the following well-known result:

Lemma 3.1 With the notations 3.1, the following relations hold:

$$\langle u, v, w \rangle = \langle w, u, v \rangle = \langle v, w, u \rangle. \tag{7}$$

This allows us to state that algorithm (5b-5g) requires:

$$\langle 4, 4, 4 \rangle + 3 \langle 3, 3, 4 \rangle + 3 \langle 3, 4, 4 \rangle \tag{8}$$

multiplications. As A. V. Smirnov states that  $\langle 3, 3, 4 \rangle = 29$  and  $\langle 3, 4, 4 \rangle = 38$  in [5, Table 1, N<sup>o</sup> 13 and 21], we conclude that our algorithm required 250 multiplications (7<sup>2</sup> + 3 · 29 + 3 · 38). Furthermore, Smirnov provides in [6] the explicit description of these algorithms, allowing us to do the same with the algorithm constructed in this note at url:

http://www.fil.univ-lille1.fr/ sedoglav/MUL7x7x7250.txt.

### References

- DREVET, C.-É., NAZRUL ISLAM, M., AND SCHOST, É. Optimization techniques for small matrix multiplication. *Theoretical Computer Sci*ence 412, 22 (May 2011), 2219–2236.
- [2] DUMAS, J.-G., AND PAN, V. Y. Fast matrix multiplication and symbolic computation. Tech. rep. 1612.05766, arXiv, Dec. 2016.
- [3] KNUTH, D. E. The Art of Computer Programming. Seminumerical Algorithms, 3 ed., vol. 2 of Computer Science and Information Processing. Addison Wesley, Reading, Mass., 1997.
- [4] LANDSBERG, J. M. Tensors: geometry and applications, vol. 128 of Graduate Studies in Mathematics. American Mathematical Society, 2010.
- [5] SMIRNOV, A. V. The bilinear complexity and practical algorithms for matrix multiplication. Computational Mathematics and Mathemacital Physics 53, 2 (Dec. 2013), 1781–1795.
- [6] SMIRNOV, A. V. Several bilinear algorithms for matrix multiplication. Tech. rep., ResearchGate, DOI: 10.13140/RG.2.2.30005.06886, Jan. 2017.
- [7] STRASSEN, V. Gaussian elimination is not optimal. Numerische Mathematik 13, 4 (Aug. 1969), 354–356.