# A Strassen-like Matrix Multiplication

## Suited for Squaring and Higher Power Computation

Marco Bodrato

ISSAC 2010 - July 28$^{\mathrm{th}}$

see appendices

# Matrix $2 \times 2$ product ...

We start from two matrices $A, B \in M_{2\times 2}$
and we need the product:

$$M_{2\times 2} \ni C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}$$

**Naïve vs. Strassen**

The naïve algorithm requires 8 multiplications:

$$C = \begin{pmatrix} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\ A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \end{pmatrix};$$

thanks to Strassen, we can use only 7.

# Matrix $2 \times 2$ product and squaring

We start from one matrix $A \in M_{2\times 2}$
and we need the square of it:

$$M_{2\times 2} \ni C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}^2$$

**Naïve vs. Strassen**

The naïve algorithm requires 8 multiplications:

commutative case

$$C = \begin{pmatrix} A_{11}A_{11} + A_{12}A_{21} & A_{11}A_{12} + A_{12}A_{22} \\ A_{21}A_{11} + A_{22}A_{21} & A_{21}A_{12} + A_{22}A_{22} \end{pmatrix};$$

thanks to Strassen, we can use only 7.

# Recall Strassen method

Strassen's method trades one multiplication with many pre- and post- linear combinations, it does not assume commutativity and it can be used recursively.

$$\begin{cases} P_1 &= (A_{21}+A_{22})B_{11} \\ P_2 &= A_{11}(B_{12}-B_{22}) \\ P_3 &= (A_{12}-A_{22})(B_{21}+B_{22}) \\ P_4 &= (A_{11}+A_{12})B_{22} \\ P_5 &= A_{22}(B_{21}-B_{11}) \\ P_6 &= (A_{21}-A_{11})(B_{11}+B_{12}) \\ P_7 &= (A_{11}+A_{22})(B_{11}+B_{22}) \end{cases} \qquad \begin{cases} C_{11} &= P_7-P_4+P_5+P_3 \\ C_{12} &= P_2+P_4 \\ C_{21} &= P_1+P_5 \\ C_{22} &= P_7+P_2-P_1+P_6 \end{cases}$$

# Strassen-like $2 \times 2$ matrix multiplication algorithm

## Number of operations, for the product of $2 \times 2$ matrices

| **Method** | **additions** | **multiplications** | *complexity* |
|---|---|---|---|
| Naïve | 4 | 8 | $d^3$ |
| Strassen's | 18 | 7 | $7d^{\log_2 7}$ |
| Winograd's | **15** | **7** | $6d^{\log_2 7}$ |

Winograd variant **is** optimal for $2 \times 2$ multiplication.
That's why research on $2 \times 2$ matrix product basically stopped (except some works on scheduling), the focus moved on bigger matrices.

# The search for a new sequence

## What about matrix squaring?

| Squaring method | additions | multiplications | squaring |
|---|---|---|---|
| Naïve | 4 | 6 | 2 |
| Strassen's | 13 | 6 | 1 |
| Winograd's | 15 | 6 | 1 |

Winograd's variant **is not** optimal for $2 \times 2$ squaring.
That's why we started the search for a new sequence.
The sequence was searched for $2 \times 2$ matrices in $\mathrm{GF}(2)$, as a sequence of additions and multiplications only, then lifted to $\mathbb{Z}$.
We can not use commutativity.
We are **not** trying to reduce the big-O complexity.

# The new sequence for $C = AB$

## Symmetry!

look at Winograd's

$$A=\begin{pmatrix}A_{11} & A_{12}\\ A_{21} & A_{22}\end{pmatrix}\begin{cases}S_1=A_{22}+A_{12}\\ S_2=A_{22}-A_{21}\\ S_3=S_2+A_{12}\\ S_4=S_3-A_{11}\end{cases}\quad B=\begin{pmatrix}B_{11} & B_{12}\\ B_{21} & B_{22}\end{pmatrix}\begin{cases}T_1=B_{22}+B_{12}\\ T_2=B_{22}-B_{21}\\ T_3=T_2+B_{12}\\ T_4=T_3-B_{11}\end{cases}$$

$$\begin{cases}P_1 = S_1 T_1\\ P_2 = S_2 T_2\\ P_3 = S_3 T_3\\ P_4 = A_{11} B_{11}\\ P_5 = A_{12} B_{21}\\ P_6 = S_4 B_{12}\\ P_7 = A_{21} T_4\end{cases}\quad \begin{cases}U_1 = P_3 + P_5\\ U_2 = P_1 - U_1\\ U_3 = U_1 - P_2\\ C_{11} = P_4 + P_5\\ C_{12} = U_3 - P_6\\ C_{21} = U_2 - P_7\\ C_{22} = P_2 + U_2\end{cases}\rightsquigarrow C=\begin{pmatrix}C_{11} & C_{12}\\ C_{21} & C_{22}\end{pmatrix}$$

# What's new in the new sequence

The new sequence is:

- equivalent to Winograd's variant for plain product;
- symmetric;
- optimal for squaring;

| Squaring method | additions | multiplications | squaring |
|---|---|---|---|
| Naïve | 4 | 6 | 2 |
| Strassen's | 13 | 6 | 1 |
| Winograd's | 15 | 6 | 1 |
| New sequence | **11** | **3** | **4** |

# What's new in the new sequence

The new sequence is:

- equivalent to Winograd's variant for plain product;
- symmetric;
- optimal for squaring;
  - number of multiplications and squarings is minimal (7),
  - number of multiplications not being squarings is minimal (3),
  - because of symmetry pre-computations on the operand is halved.

| **Squaring method** | **additions** | **multiplications** | | **squaring** |
|---|---|---|---|---|
| Naïve | 4 | 6 | | 2 |
| Strassen's | 13 | 6 | | 1 |
| Winograd's | 15 | 6 | | 1 |
| New sequence | **11** | (**3** | + | **4**) = 7 |

# The new sequence for $C = A^2$

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \left\{ \begin{array}{l} S_1 = A_{22} + A_{12} \\ S_2 = A_{22} - A_{21} \\ S_3 = S_2 + A_{12} \\ S_4 = S_3 - A_{11} \end{array} \right.$$

$$\left\{ \begin{array}{lcl} P_1 & = & S_1 S_1 \\ P_2 & = & S_2 S_2 \\ P_3 & = & S_3 S_3 \\ P_4 & = & A_{11} A_{11} \\ P_5 & = & A_{12} A_{21} \\ P_6 & = & S_4 A_{12} \\ P_7 & = & A_{21} S_4 \end{array} \right. \quad \left\{ \begin{array}{lcl} U_1 & = & P_3 + P_5 \\ U_2 & = & P_1 - U_1 \\ U_3 & = & U_1 - P_2 \\ C_{11} & = & P_4 + P_5 \\ C_{12} & = & U_3 - P_6 \\ C_{21} & = & U_2 - P_7 \\ C_{22} & = & P_2 + U_2 \end{array} \right. \rightsquigarrow C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$$

## The three products...

Thanks to symmetry, when dealing with the three products, we can keep using a half the pre-computation on operands, and this is true for any recursion level.

$$\begin{cases} P_5 & = & A_{12}A_{21} \\ P_6 & = & S_4A_{12} \\ P_7 & = & A_{21}S_4 \end{cases}$$

Matrix multiplication is not commutative, but the sequence is symmetric

# Reduce linear operations for chain products

When one needs to compute $M^3$ or $M_1 M_2 M_3$, one usually need some intermediate results: $M^2 M$, $(M_1 M_2) M_3$.

# Reduce linear operations for chain products

When one needs to compute $M^3$ or $M_1 M_2 M_3$, one usually need some intermediate results: $M^2 M$, $(M_1 M_2) M_3$.

$$\begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix} \left\{ \begin{matrix} S_1 & \rightsquigarrow & P_1 \\ S_2 & \rightsquigarrow & P_2 \\ S_3 & \rightsquigarrow & P_3 \\ S_4 & \rightsquigarrow & P_4 \\ M_{11} & \rightsquigarrow & P_5 \\ M_{12} & \rightsquigarrow & P_6 \\ M_{21} & \rightsquigarrow & P_7 \end{matrix} \right\} \begin{pmatrix} \widetilde{M}_{11} & \widetilde{M}_{12} \\ \widetilde{M}_{21} & \widetilde{M}_{22} \end{pmatrix} \left\{ \begin{matrix} \widetilde{S}_1 & \rightsquigarrow & \widetilde{P}_1 \\ \widetilde{S}_2 & \rightsquigarrow & \widetilde{P}_2 \\ \widetilde{S}_3 & \rightsquigarrow & \widetilde{P}_3 \\ \widetilde{S}_4 & \rightsquigarrow & \widetilde{P}_4 \\ \widetilde{M}_{11} & \rightsquigarrow & \widetilde{P}_5 \\ \widetilde{M}_{12} & \rightsquigarrow & \widetilde{P}_6 \\ \widetilde{M}_{21} & \rightsquigarrow & \widetilde{P}_7 \end{matrix} \right.$$

# Reduce linear operations for chain products

When one needs to compute $M^3$ or $M_1 M_2 M_3$, one usually need some intermediate results: $M^2 M$, $(M_1 M_2) M_3$.

$$\begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix} \left\{ \begin{array}{lcl} S_1 & \rightsquigarrow & P_1 \\ S_2 & \rightsquigarrow & P_2 \\ S_3 & \rightsquigarrow & P_3 \\ S_4 & \rightsquigarrow & P_4 \\ M_{11} & \rightsquigarrow & P_5 \\ M_{12} & \rightsquigarrow & P_6 \\ M_{21} & \rightsquigarrow & P_7 \end{array} \right. \qquad \begin{array}{lcl} \widetilde{S}_1 & \rightsquigarrow & \widetilde{P}_1 \\ \widetilde{S}_2 & \rightsquigarrow & \widetilde{P}_2 \\ \widetilde{S}_3 & \rightsquigarrow & \widetilde{P}_3 \\ \widetilde{S}_4 & \rightsquigarrow & \widetilde{P}_4 \\ \widetilde{M}_{11} & \rightsquigarrow & \widetilde{P}_5 \\ \widetilde{M}_{12} & \rightsquigarrow & \widetilde{P}_6 \\ \widetilde{M}_{21} & \rightsquigarrow & \widetilde{P}_7 \end{array}$$

If we don't need the intermediate value, we should skip it.

# How to skip the unneeded values

The linear post- and pre- computation can collapse.

$$\begin{cases} \widetilde{M}_{11} = P_4 + P_5 \\ \widetilde{M}_{12} = P_3 - P_2 - P_6 + P_5 \\ \widetilde{S}_2 \quad = P_2 + P_7 \\ \widetilde{S}_1 \quad = P_1 - P_6 \\ \hline \widetilde{S}_3 \quad = \widetilde{S}_2 + \widetilde{M}_{12} \\ \widetilde{M}_{21} = \widetilde{S}_1 - \widetilde{S}_3 \\ \widetilde{S}_4 \quad = \widetilde{S}_3 - \widetilde{M}_{11} \end{cases}$$

Before we had $7 + 4 = 11$ operations, now we have 9.

# How to skip the unneeded values

The linear post- and pre- computation can collapse.

$$\left\{\begin{array}{l} \left.\begin{array}{ll} \widetilde{M}_{11} &= P_4 + P_5 \\ \widetilde{M}_{12} &= P_3 - P_2 - P_6 + P_5 \\ \widetilde{S}_2 &= P_2 + P_7 \\ \widetilde{S}_1 &= P_1 - P_6 \end{array}\right\} \text{Depending on products} \\ \hline \left.\begin{array}{ll} \widetilde{S}_3 &= \widetilde{S}_2 + \widetilde{M}_{12} \\ \widetilde{M}_{21} &= \widetilde{S}_1 - \widetilde{S}_3 \\ \widetilde{S}_4 &= \widetilde{S}_3 - \widetilde{M}_{11} \end{array}\right\} \text{Depending on other values} \end{array}\right.$$

Before we had $7 + 4 = 11$ operations, now we have 9.
Moreover we can group operations . . .

# How to save linear operations

## with no more memory needs

Before we had an unneeded value

$$\begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix} \left\{ \begin{array}{ccc} S_1 & \rightsquigarrow & P_1 \\ S_2 & \rightsquigarrow & P_2 \\ S_3 & \rightsquigarrow & P_3 \\ S_4 & \rightsquigarrow & P_4 \\ M_{11} & \rightsquigarrow & P_5 \\ M_{12} & \rightsquigarrow & P_6 \\ M_{21} & \rightsquigarrow & P_7 \end{array} \right\} \begin{pmatrix} \widetilde{M}_{11} & \widetilde{M}_{12} \\ \widetilde{M}_{21} & \color{red}{\widetilde{M}_{22}} \end{pmatrix} \left\{ \begin{array}{ccc} \widetilde{S}_1 & \rightsquigarrow & \widetilde{P}_1 \\ \widetilde{S}_2 & \rightsquigarrow & \widetilde{P}_2 \\ \widetilde{S}_3 & \rightsquigarrow & \widetilde{P}_3 \\ \widetilde{S}_4 & \rightsquigarrow & \widetilde{P}_4 \\ \widetilde{M}_{11} & \rightsquigarrow & \widetilde{P}_5 \\ \widetilde{M}_{12} & \rightsquigarrow & \widetilde{P}_6 \\ \widetilde{M}_{21} & \rightsquigarrow & \widetilde{P}_7 \end{array} \right.$$

# How to save linear operations

## with no more memory needs

Before we had an unneeded value, now we have none.

$$\begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix} \left\{ \begin{array}{ccc} S_1 & \rightsquigarrow & P_1 \\ S_2 & \rightsquigarrow & P_2 \\ S_3 & \rightsquigarrow & P_3 \\ S_4 & \rightsquigarrow & P_4 \\ M_{11} & \rightsquigarrow & P_5 \\ M_{12} & \rightsquigarrow & P_6 \\ M_{21} & \rightsquigarrow & P_7 \end{array} \right\} \begin{pmatrix} \widetilde{M}_{11} & \widetilde{M}_{12} \\ \widetilde{S}_2 & \widetilde{S}_1 \end{pmatrix} \left\{ \begin{array}{ccc} \widetilde{S}_1 & \rightsquigarrow & \widetilde{P}_1 \\ \widetilde{S}_2 & \rightsquigarrow & \widetilde{P}_2 \\ \widetilde{S}_3 & \rightsquigarrow & \widetilde{P}_3 \\ \widetilde{S}_4 & \rightsquigarrow & \widetilde{P}_4 \\ \widetilde{M}_{11} & \rightsquigarrow & \widetilde{P}_5 \\ \widetilde{M}_{12} & \rightsquigarrow & \widetilde{P}_6 \\ \widetilde{M}_{21} & \rightsquigarrow & \widetilde{P}_7 \end{array} \right.$$

We save 2 operations by replacing the sequences.
Because of symmetry of the method we do not care of computation order: $(M_1 M_2) M_3$ or $M_1 (M_2 M_3)$.

# Intermediate and standard representation are linked

The intermediate representation is obtained from the standard one with a simple linear function.

$$\psi\left(\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}\right)=\begin{pmatrix} A_{11} & A_{12} \\ A_{22}-A_{21} & A_{22}+A_{12} \end{pmatrix}$$

$$\psi^{-1}\left(\begin{pmatrix} A_{11} & A_{12} \\ S_2 & S_1 \end{pmatrix}\right)=\begin{pmatrix} A_{11} & A_{12} \\ (\mathbf{S_1}-\mathbf{A_{12}})-S_2 & (\mathbf{S_1}-\mathbf{A_{12}}) \end{pmatrix}$$

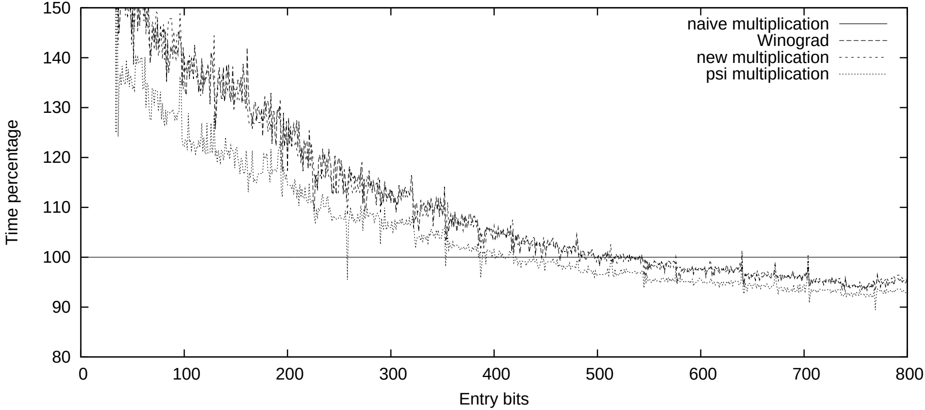
## This representation is optimal

- it uses as much memory as standard one;
- the number of linear operation for Strassen-like multiplications is minimal.

It's linear: it can be used for general polynomial computations too.

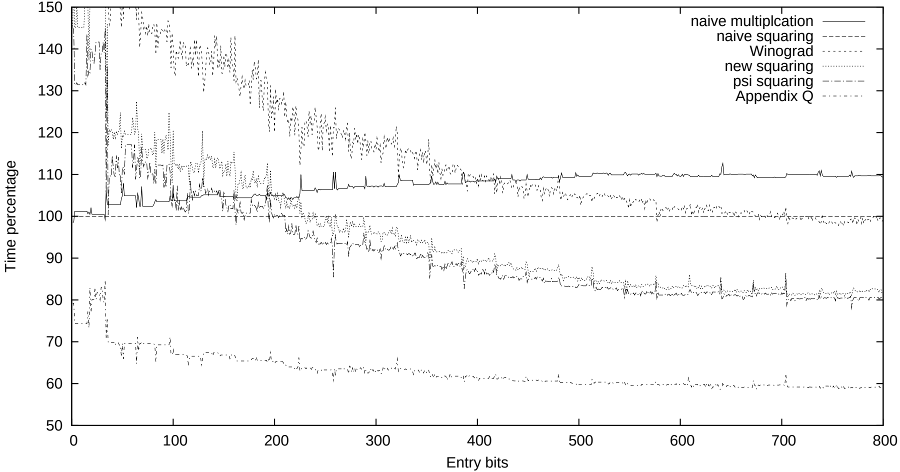
# Matrix-Matrix Multiplication

## Time ratio with respect to naïve multiplication ($2 \times 2$)



# Matrix Squaring

## Time ratio with respect to naïve squaring ($2 \times 2$)



# Current implementations of the new sequence

1. M4RI: for big matrices in $\mathrm{GF}(2)$ (early 2009);
2. GMP: for $2 \times 2$ matrices, (used in HGCD code);
3. `fastmm` library: matrices with float entries (licence problems);

None of them implement chain product nor any trick for powers, nevertheless they give some small but measurable improvement with respect to previous Strassen-Winograd implementations.
(There are others good side-effects of symmetry)

# Conclusions

A new Strassen-like sequence was proposed. It is:

- symmetric;
- optimal for plain product (not new);
- optimal for squaring;
- optimal for chain products.

Software implementation is as easy as Winograd's variant.
Consider implementing it if you need a variant of Strassen.

# That's all !

Thank you very much for your kind attention

Questions?

Presentation will be available on the web:
http://bodrato.it/papers/#ISSAC2010,


Full paper too is available on web.

## 4 Some more details

- Commutative matrix squaring
- Winograd's variant

back to index

# Matrix $2 \times 2$ squaring, exploiting commutativity

## The simple formula for $2 \times 2$ matrices

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^2 = \begin{pmatrix} a^2 + bc & b(a+d) \\ c(a+d) & d^2 + bc \end{pmatrix}$$

It can be generalised obtaining:

$$d \text{ squares, } d^3 - d^2 - \binom{d}{2} \text{ products, } d^3 - d^2 - \binom{d}{2} \text{ additions,}$$

but it is fast only for $2 \times 2$ or $3 \times 3$ matrices.

# Winograd's variant

## No symmetry. . .

back to new sequence

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{cases} S_1 = A_{21} + A_{22} \\ S_2 = S_1 - A_{11} \\ S_3 = A_{11} - A_{21} \\ S_4 = A_{12} - S_2 \end{cases} \quad B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} \begin{cases} T_1 = B_{12} - B_{11} \\ T_2 = B_{22} - T_1 \\ T_3 = B_{22} - B_{12} \\ T_4 = B_{21} - T_2 \end{cases}$$

$$\begin{cases} P_1 &= A_{11} B_{11} \\ P_2 &= A_{12} B_{21} \\ P_3 &= S_1 T_1 \\ P_4 &= S_2 T_2 \\ P_5 &= S_3 T_3 \\ P_6 &= S_4 B_{22} \\ P_7 &= A_{22} T_4 \end{cases} \quad \begin{cases} U_1 &= P_1 + P_4 \\ U_2 &= U_1 + P_5 \\ U_3 &= U_1 + P_3 \\ C_{11} &= P_2 + P_1 \\ C_{12} &= U_3 + P_6 \\ C_{21} &= U_2 + P_7 \\ C_{22} &= U_3 + P_5 \end{cases} \rightsquigarrow C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$$