Le produit médian

Paul Zimmermann

(avec Guillaume Hanrot et Michel Quercia)

Séminaire Complexité et Cryptographie, École Normale Supérieure, 29 janvier 2004

Complexité des opérations élémentaires

Multiplication, division, fonctions algébriques (carré, racine carrée, ...), fonctions élémentaires (exponentielle, logarithme) ...

... sur entiers, flottants, polynômes ou séries formelles.

On se ramène usuellement à $M(n)$.

Ex. : l'inversion par Newton coûte $3M(n)$ $[I(n) = I(n/2) + 3M(n/2)]$.

Multiplication tronquée (*short product*)

flottants $\equiv$ séries de Taylor tronquées.

Brent : coûts exprimés en fonction de $M^*(n)$.

Cohérent, mais constantes plus élevées : $M^*(n) \sim cM(n)$ pour $c < 1$?

Mulders : oui pour Karatsuba et Toom-Cook, mais pas pour FFT.

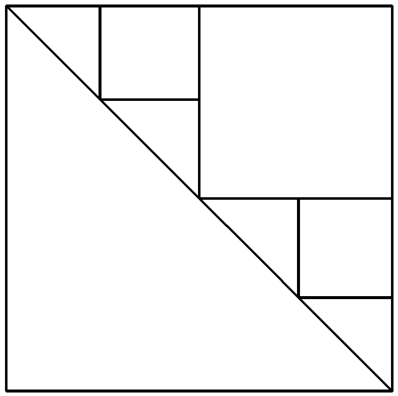
Multiplication détendue

Van der Hoeven : intermédiaire entre calcul paresseux (n^2 pour la multiplication) et zélé ($M(n)$).

Idée : similaire au calcul en ligne, mais on anticipe les calculs dès que possible, avec une multiplication rapide.

b_3	3			
b_2	2	3		
b_1	1	2	3	
b_0	0	1	2	3
	a_0	a_1	a_2	a_3

3	3	3	3
2	3	2	3
1	1	3	3
0	1	2	3
a_0	a_1	a_2	a_3



Produit médian

On considère les n chiffres médians d'un produit $n \times (2n)$.

$$321110693270 \;\times\; 474256143563558458718976$$

$$= 152288719047250904987423059464491520$$

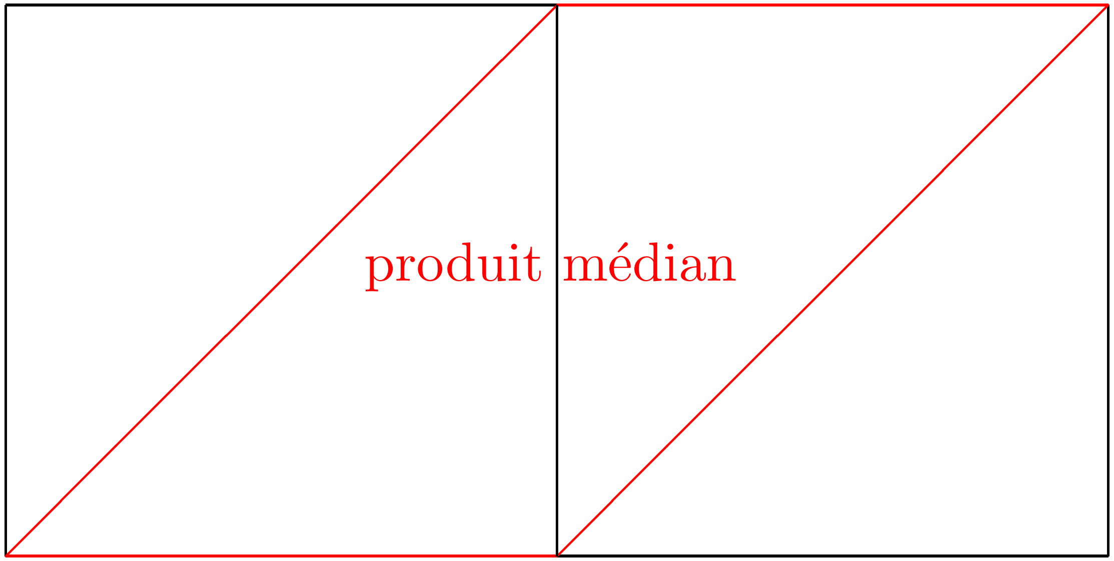
Produit classique : $n \times n \to 2n$

Produit court : $n \times n \to n$

Produit médian : $n \times 2n \to n$

($2n \to 2n-1$ pour les polynômes)

Un dessin vaut mieux...



$$M(n/2) \leq MP(n) \leq 2M(n) \text{ et } M^*(n) \leq MP(n) \leq 2M^*(n)$$

Question : peut-on calculer $MP(n)$ plus rapidement que $2M(n)$?

Motivation

Itération de Newton pour $1/A$:

$$x_{k+1} = x_k + x_k(1 - Ax_k).$$

Si x_k a n chiffres corrects, on prend $2n$ chiffres de A. Le produit Ax_k fait $3n$ chiffres, dont les n chiffres de poids fort s'annulent avec 1, et les n chiffres de poids faible sont inintéressants.

Exemple

Calcul de $1/1997451013 \bmod 2^{32}$. $x_0 = 1 = 1/A \bmod 2^1$,
$x_1 = 1 = 1/A \bmod 2^2$, $x_2 = 13 = 1/A \bmod 2^4$,
$x_3 = 205 = 1/A \bmod 2^8$, $x_4 = 50637 = 1/A \bmod 2^{16}$.

$$Ax_4 = 101101111111101\textcolor{red}{1010001101110001}0000000000000001$$

$$= 23549 \cdot 2^{32} + 41841 \cdot 2^{16} + 1$$

$$x_4(1 - Ax_4) = x_4(-41841 \cdot 2^{16}) \bmod 2^{32} = -2^{16}(41841 x_4 \bmod 2^{16}).$$

Cas de base ($n = 2$)

$$(a_0 + a_1 x) \times (b_0 + b_1 x + b_2 x^2) =$$

$$a_0 b_0 + (a_0 b_1 + a_1 b_0) x + (a_0 b_2 + a_1 b_1) x^2 + a_1 b_2 x^3$$

Peut-on calculer les deux termes médians en 3 multiplications ?

$$\alpha \leftarrow a_1(b_0 + b_1), \quad \beta \leftarrow a_0(b_1 + b_2), \quad \gamma \leftarrow (a_0 - a_1) b_1$$

$$\alpha + \gamma = a_0 b_1 + a_1 b_0, \quad \beta - \gamma = a_0 b_2 + a_1 b_1$$

Cas général

L'astuce pour $n = 2$ se généralise à n quelconque :

Algorithme $\texttt{MP-Kara}([y_0, \ldots, y_{n-1}], [z_0, \ldots, z_{2n-2}])$.

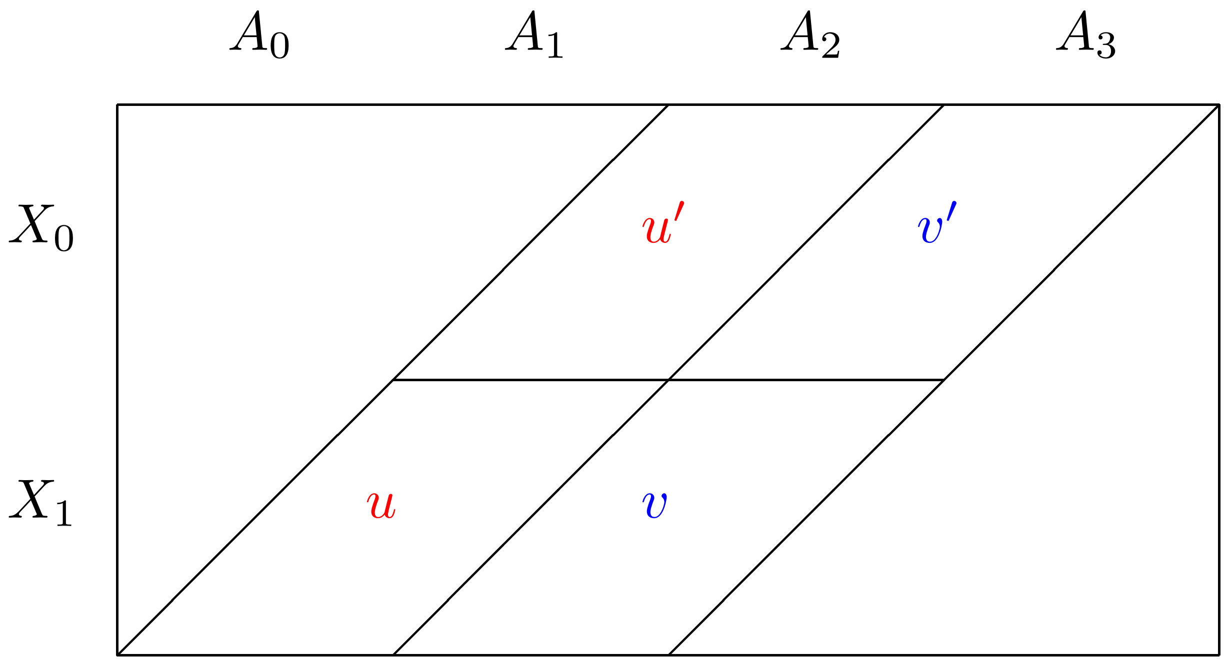
0. If $n = 1$, return $[y_0 z_0]$.
1. $n_0 \leftarrow \lfloor n/2 \rfloor, \quad n_1 \leftarrow \lceil n/2 \rceil$.
2. $\alpha \leftarrow \texttt{MP-Kara}([y_{n_0}, \ldots, y_{n-1}], [z_0 + z_{n_1}, \ldots, z_{2n_1-2} + z_{3n_1-2}])$
3. If n is even

$$\beta \leftarrow \texttt{MP-Kara}([y_{n_1} - y_0, \ldots, y_{n-1} - y_{n_0-1}], [z_{n_1}, \ldots, z_{3n_1-2}])$$

else

$$\beta \leftarrow \texttt{MP-Kara}([y_{n_0}, y_{n_1} - y_0, \ldots, y_{n-1} - y_{n_0-1}], [z_{n_1}, \ldots, z_{3n_1-2}])$$

4. $\gamma \leftarrow \texttt{MP-Kara}([y_0, \ldots, y_{n_0-1}], [z_{n_1} + z_{2n_1}, \ldots, z_{n_1+2n_0-2} + z_{2n-2}])$
5. Return $[\alpha_0 - \beta_0, \ldots, \alpha_{n_1-1} - \beta_{n_1-1}, \gamma_0 + \beta_0, \ldots, \gamma_{n_0-1} + \beta_{n_0-1}]$.



$$
\begin{aligned}
u + v &= \alpha = \mathtt{MP}(X_1,\ [A_0, A_1] + [A_1, A_2]) \\
v - u' &= \beta = \mathtt{MP}(X_1 - X_0,\ [A_1, A_2]) \\
u' + v' &= \gamma = \mathtt{MP}(X_0,\ [A_1, A_2] + [A_2, A_3])
\end{aligned}
$$

Modèles Karatsuba et FFT

(Résultats obtenus avec Hanrot et Quercia, AEECC 2004).

Produit médian en $K(n)$ mult. scalaires, ou $FFT(n)$.

Carré en $K(n)/2$ multiplications scalaires.

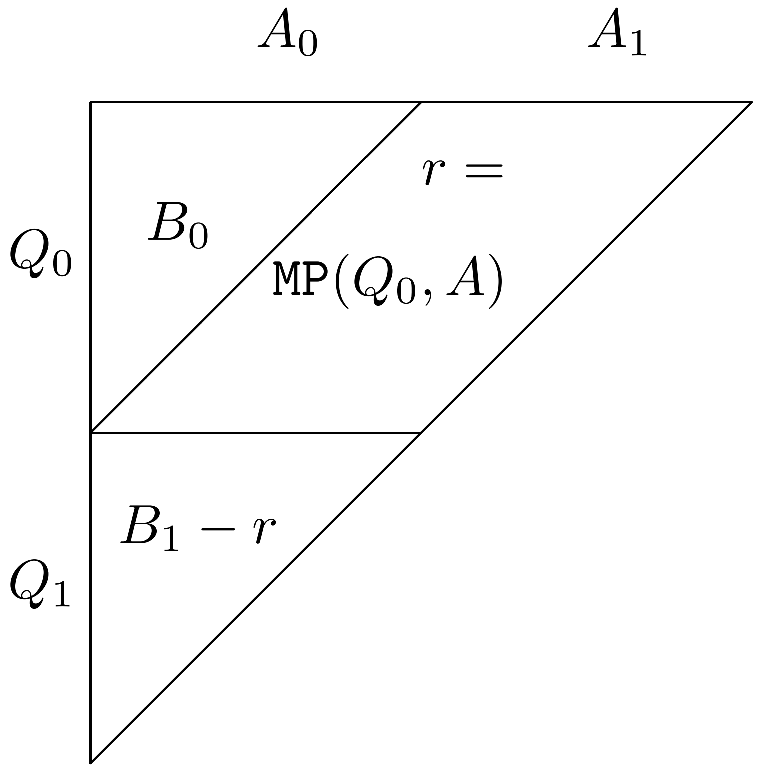
Inverse en $K(n)$, ou $2FFT(n)$.

Quotient en $K(n)$ avec MP-divide, ou $\frac{5}{2}FFT(n)$.

Racine carrée en $\frac{3}{4}K(n)$, ou $3FFT(n)$.

MP-divide

$$B_0 + B_1 t = (Q_0 + Q_1 t)(A_0 + A_1 t) \bmod t^2$$



$\boxed{\textbf{MP-FFT}}$

Algorithme `MP-FFT`$([y_0, \ldots, y_{n-1}], [z_0, \ldots, z_{2n-2}])$.

0. $\omega \leftarrow$ primitive $(2n)$-th root of unity
1. $\alpha \leftarrow \texttt{FFT}(\omega, [y_{n-1}, \ldots, y_0, \underbrace{0, \ldots, 0}_{n}])$
2. $\beta \leftarrow \texttt{FFT}(\omega^{-1}, [z_0, \ldots, z_{2n-2}, 0])$
3. $\gamma \leftarrow \texttt{FFT}(\omega, [\alpha_0\beta_0, \ldots, \alpha_{2n-1}\beta_{2n-1}])$
4. Return $[\gamma_0/2n, \ldots, \gamma_{n-1}/2n]$.

Implantation

Facile pour les polynômes/séries pour Karatsuba et FFT (`http://www.loria.fr/~zimmerma/papers/MP.mpl`).

Plus ardu pour Toom-Cook et variantes.

Difficile pour entiers/flottants à cause des retenues :

$$MP(321110693270, 474256143563558458718976) = 250904987423$$

$$\alpha = MP(693270, 474256143563 + 143563558458) = 864820$$

$$\beta = MP(321110, 143563558458 + 558458718976) = 373506$$

$$\gamma = MP(693270 - 321110, 143563558458) = 613915$$

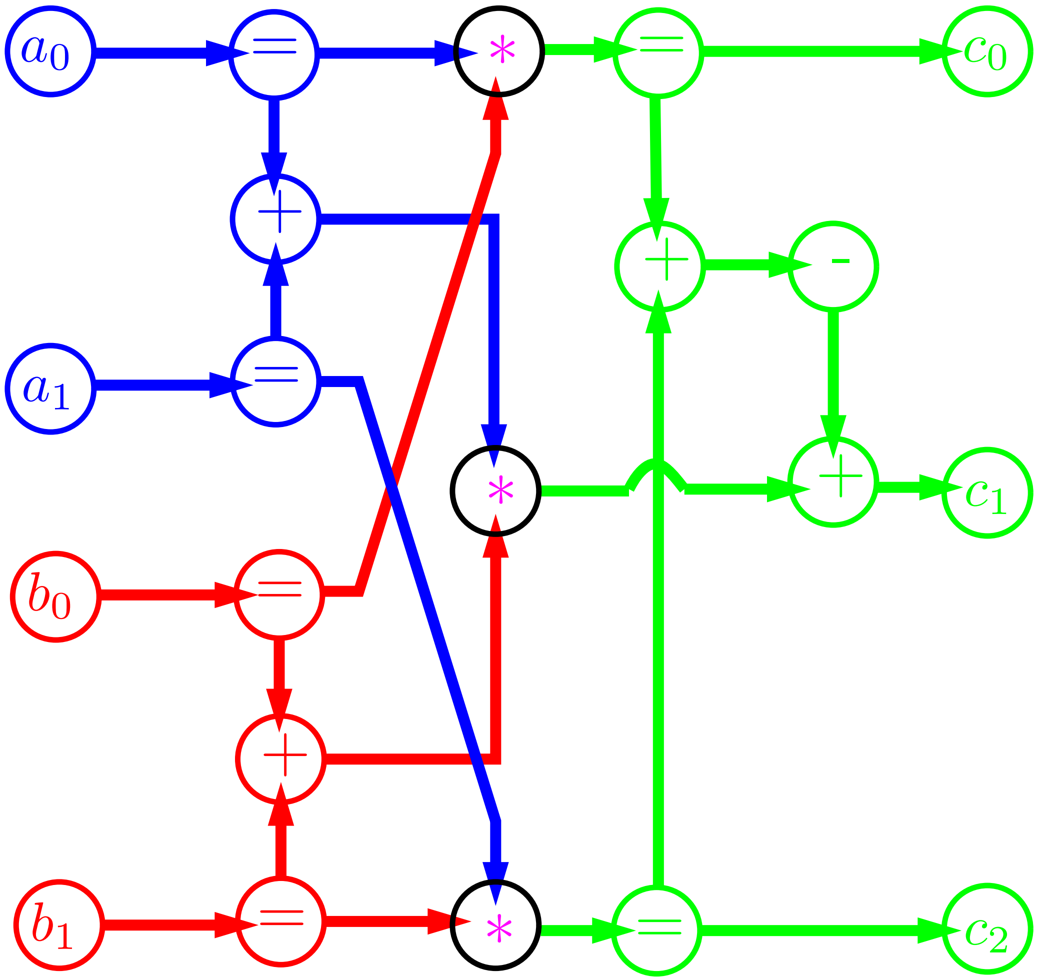
$$\alpha - \gamma = 250905, \qquad \beta + \gamma = 987421$$

En termes de circuit

Un algorithme de multiplication peut être vu comme un circuit prenant en entrée $a_0, ..., a_{n-1}$, et $b_0, ..., b_{n-1}$, et donnant en sortie $c_0, ..., c_{2n-2}$.

Portes $\oplus$, $\otimes$, $\ominus$ (unaire), $\ominus\!\!\!\!=$ (duplication).

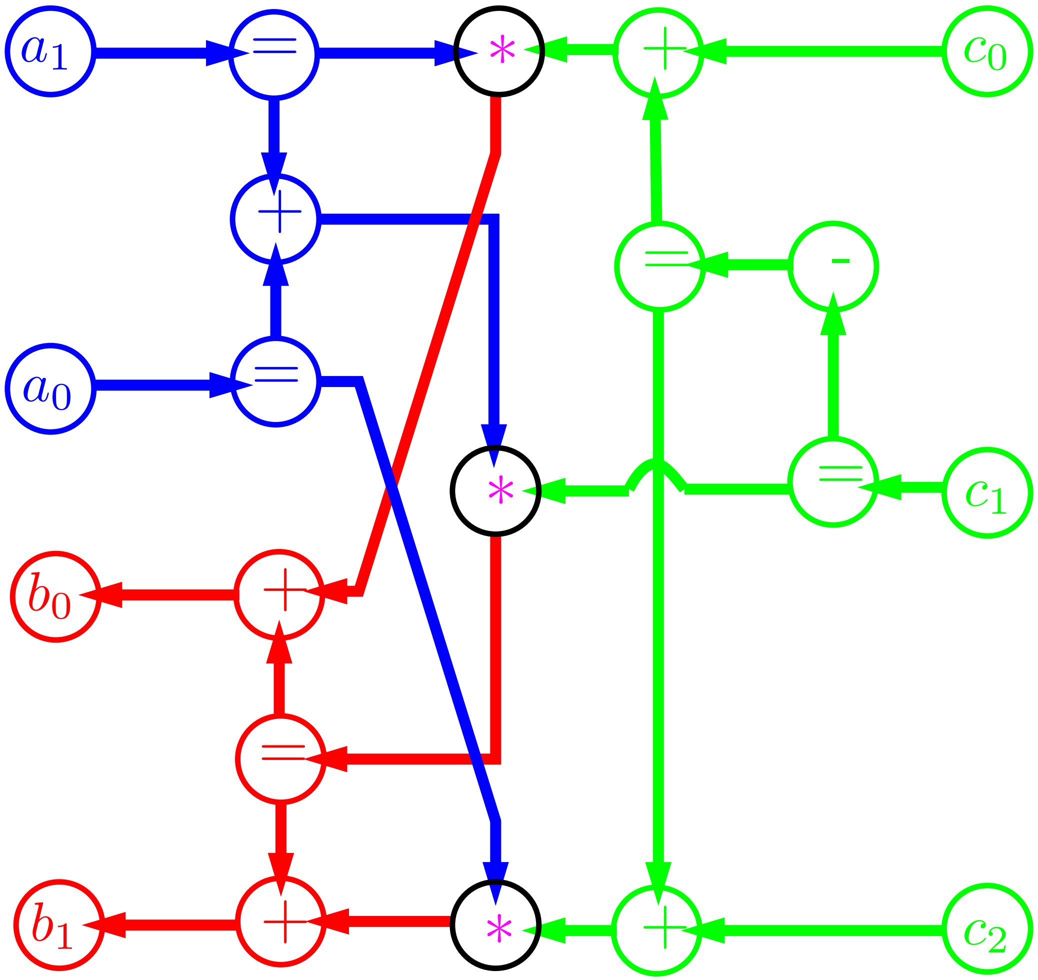
Circuit Karatsuba



Produit classique $\Longrightarrow$ produit médian :

- le circuit partant de a_i jusqu'aux $\otimes$ reste inchangé
- les flèches des circuits b_i et c_i sont renversées
- les $\oplus$ deviennent $\ominus$, les $\ominus$ deviennent $\oplus$

Circuit MP-Karatsuba



Kara($[a_0, a_1], [b_0, b_1]$)	MP-Kara($[a_0, a_1], [c_0, c_1, c_2]$)
$a_2 \leftarrow a_0 + a_1$	$a_2 \leftarrow a_0 + a_1$
$b_0, b'_0 \leftarrow b_0$	$c_1, c'_0, c'_2 \leftarrow c_1$
$b_1, b'_1 \leftarrow b_1$	$c_2 \leftarrow c_2 - c'_2$
$b_2 \leftarrow b'_0 + b'_1$	$c_0 \leftarrow c_0 - c'_0$
$c_0 \leftarrow a_0 \times b_0$	$b_2 \leftarrow a_2 \times c_1$
$c_2 \leftarrow a_1 \times b_1$	$b_1 \leftarrow a_1 \times c_2$
$c_1 \leftarrow a_2 \times b_2$	$b_0 \leftarrow a_0 \times c_0$
$c_0, -c'_0 \leftarrow c_0$	$b'_0, b'_1 \leftarrow b_2$
$c_2, -c'_2 \leftarrow c_2$	$b_1 \leftarrow b_1 + b'_1$
$c_1 \leftarrow c_1 + c'_0 + c'_2$	$b_0 \leftarrow b_0 + b'_0$
Return $[c_0, c_1, c_2]$.	Return $[b_0, b_1]$.

En termes matriciels

Principe de Tellegen (Tellegen, 1952) : tout algorithme calculant un produit matrice-vecteur peut être transposé en un algorithme qui calcule le produit par la matrice transposée.

Aussi appelé principe de transposition (Kaminski, Kirkpatrick, Bshouty 1988).

Souvent utilisé en théorie de la complexité, rarement de manière effective.

Multiplication $n \times n \Longrightarrow$ matrice de Toeplitz $(2n) \times n$:

$$\begin{pmatrix} a_0 & 0 \\ a_1 & a_0 \\ a_2 & a_1 \\ 0 & a_2 \end{pmatrix} \times \begin{pmatrix} b_0 \\ b_1 \end{pmatrix} = \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix}$$

Le produit par la matrice transposée correspond au produit médian :

$$\begin{pmatrix} a_0 & a_1 & a_2 & 0 \\ 0 & a_0 & a_1 & a_2 \end{pmatrix} \times \begin{pmatrix} 0 \\ b_1 \\ b_0 \\ 0 \end{pmatrix} = \begin{pmatrix} c_2 \\ c_1 \end{pmatrix}$$

Le transposé du produit court $n \times n \rightarrow n$

$$\begin{pmatrix} a_0 & 0 \\ a_1 & a_0 \end{pmatrix} \times \begin{pmatrix} b_0 \\ b_1 \end{pmatrix} = \begin{pmatrix} c_0 \\ c_1 \end{pmatrix}$$

... est lui-même :

$$\begin{pmatrix} a_0 & a_1 \\ 0 & a_0 \end{pmatrix} \times \begin{pmatrix} b_1 \\ b_0 \end{pmatrix} = \begin{pmatrix} c_1 \\ c_0 \end{pmatrix}$$

Généralisation à d'autres opérations

(Bostan, Lecerf, Schost, 2003)

$$\text{opération } \sigma \iff \text{opération transposée } \sigma^t$$

$$\text{algorithme pour } \sigma \iff \text{algorithme transposé pour } \sigma^t$$

Évaluation multipoint

Étant donné $f(x) = b_0 + b_1 x + \cdots + b_n x^n$, évaluer $f(a_0), ..., f(a_n)$.

Produit par matrice de Vandermonde :

$$\begin{pmatrix} 1 & a_0 & a_0^2 \\ 1 & a_1 & a_1^2 \\ 1 & a_2 & a_2^2 \end{pmatrix} \times \begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} f(a_0) \\ f(a_1) \\ f(a_2) \end{pmatrix}.$$

Opération transposée : coefficients de Taylor de $\displaystyle\sum_{j=0}^{n} \frac{b_j}{1 - a_j x}$

$\Rightarrow$ algorithme en $\frac{3}{2} M(n) \log_2 n$ pour l'évaluation multipoint.

Meilleur algo. connu auparavant : $\frac{7}{2} M(n) \log_2 n$ (Montgomery, 1992).

Résumé

Nouvelle opération « produit médian ».

Tout algorithme de multiplication se transpose en un algorithme de produit médian de même complexité.

D'autres opérations se transposent aussi (évaluation multipoint, interpolation, évaluation-interpolation).

Gains d'un facteur 2 à 3 sur le coût asymptotique.

S'implante bien pour les polynômes/séries, plus dur pour les entiers/flottants.

Références

An FFT extension of the elliptic curve method of factorization, Montgomery, PhD Thesis, 1992.

Relax, but don't be too lazy, van der Hoeven, Journal of Symbolic Computation, 2002.

The Middle Product Algorithm, I. Speeding up the division and square root of power series, Hanrot, Quercia, Z., rapport INRIA 4664 (2002), à paraître dans AAECC.

Tellegen's Principle into Practice, Bostan, Lecerf, Schost, ISSAC'2003.

Thèse d'Alin Bostan, 2003.