# Multiplication Rapide

travaux-pratiques d'algorithmique n. 4

4 novembre 2003

**Objectifs.** Implanter la méthode multiplicative de Kara{c,z, ts}uba-Ofman.
**Lectures.**
*Algorithmes en Langage C*, R. Sedgewick.
*Algorithm and complexity*, S. H. Wilf.
*The Art of Computer Programming*, D. Knuth.
**Durée:** 3 heures.

L'algorithme de multiplication naïf que nous connaissons tous calcule le produit de deux polynômes en temps quadratique. On peut faire beaucoup mieux, $\Theta(n \log n)$, par des méthodes d'interpolation à base de transformation de Fourier. Dans les années soixante, Karacuba et Ofman ont décrit un algorithme beaucoup plus facile à implanter, de complexité intermédiaire. Il s'agit de l'algorithme récursif un peu plus bas. L'astuce consiste à remarquer que les expressions intermédiaires qui apparaissent au cours des calculs peuvent être obtenues à moindre coût (3 multiplications et 3 additions) grâce à la relation,

$$(x_H + x_L)(y_H + y_L) = x_H * y_H + (x_L y_H + x_H y_L) + x_L y_L$$

D'après le cours, sa complexité est $\Theta(n^{\log_2 3})$, et donc, proche de $n\sqrt{n}$ puisque $\log_2 3 = 1.58496254$. Il calcule le produit des deux polynômes $x$ et $y$ de taille $n$ dans un troisième de taille $2n$. Dans la suite, nous supposerons toujours que $n$ est une puissance de 2. Lors d'un appel externe, les polynômes $x$, $y$ et $z$ sont vus comme des tableaux mais à l'intérieur de la procédure, il vaut mieux penser en termes d'adresses pour découper les tableaux des coefficients en zones. Les polynômes $x$ et $y$ en deux zones : faible et forte. Le polynôme $z$ en trois.

1. Implanter un produit naïf.
2. Comment tester efficacement la correction d'une procédure de calcul d'un produit ?
3. Les deux premiers paramètres de `KARACUBA(x, y, Z)` sont de taille $n$, $n$ et le troisième de taille $2n$. En déduire, les tailles des variables auxiliaires de l'algorithme.
4. Implanter l'algorithme de Karacuba-Ofman.

```
KARACUBA( x, y, z )
polynome sx, sy;
polynome tmp;
DEBUT
   n := TAILLE( x );
   SI ( n < 2 ) ALORS
      z := x * y;
      RETOUR;
   FSI
   ADD( xL, xH, sx);
   ADD( yL, yH, sy);
   KARACUBA( sx, sy, zx);
   KARACUBA( xL, yL, tmp);
   SUB( zx, tmp);
   ADD( zl, tmp);
   KARACUBA( xH, yH, tmp);
   SUB( zx, tmp);
   ADD( zh, tmp);
FIN
```

$$
\begin{array}{cccc}
 & & \boxed{y_H} & \boxed{y_L} \\
\times & & & \\
 & & \boxed{x_H} & \boxed{x_L} \\
\hline
 & & \boxed{y_H x_L} & \boxed{y_L x_L} \\
+ & & & \\
 & \boxed{y_H x_H} & \boxed{y_L x_H} & \\
\hline
= & \boxed{z_h} & \boxed{z_x} & \boxed{z_l}
\end{array}
$$

FIG. 1 – Principe et algorithme de Karacuba-Ofman.

5. Faire des mesures de temps de calcul pour mettre en évidence les complexités des algoritmes implantés.
6. Quand vaut-il mieux utiliser le produit naïf ?
7. Modifier l'algorithme de Karacuba.
8. Ecrire un algorithme de calcul multiplication pour les grands entiers.
9. Optimiser la procédure de `KARACUBA()`

Les plus curieux pourront partir à la recherche de l'article original : A. Karatsuba et Y. Ofman, *Multiplication of Many-Digital Numbers by Automatic Computers*. Doklady Akad. Nauk SSSR 145, 293-294, 1962. Translation in Physics-Doklady 7, 595-596, 1963.