Ce TP a pour objectif de vous introduire à la conception d'algorithme. Pour ce faire, nous allons implémenter la méthode de Strassen afin de calculer rapidement des produits matriciels.

1 Présentation

Soit A et B deux matrices carrées de taille $n * n$. On supposera dans un premier temps que n est une puissance de 2.

L'algorithme naïf de multiplication de matrices calcul les n^2 coefficient en effectuant n opérations à chaque fois. On effectue donc $O(n^3)$ opérations.

L'idée de l'algorithme de Strassen est de calculer ce produit récursivement comme suit :

Si $A = \begin{pmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{pmatrix}$ et $B = \begin{pmatrix} B_{1,1} & B_{1,2} \\ B_{2,1} & B_{2,2} \end{pmatrix}$, l'algorithme de Strassen consiste à calculer $C = AB = \begin{pmatrix} C_{1,1} & C_{1,2} \\ C_{2,1} & C_{2,2} \end{pmatrix}$ en effectuant des calculs par blocs.

On pose :

$$\begin{aligned}
M_1 &= (A_{1,1} + A_{2,2})(B_{1,1} + B_{2,2}) \\
M_2 &= (A_{2,1} + A_{2,2})B_{1,1} \\
M_3 &= A_{1,1}(B_{1,2} - B_{2,2}) \\
M_4 &= A_{2,2}(B_{2,1} - B_{1,1}) \\
M_5 &= (A_{1,1} + A_{1,2})B_{2,2} \\
M_6 &= (A_{2,1} - A_{1,1})(B_{1,1} + B_{1,2}) \\
M_7 &= (A_{1,2} - A_{2,2})(B_{2,1} + B_{2,2})
\end{aligned}$$

Et on vérifie alors qu'on a

$$\begin{aligned}
C_{1,1} &= M_1 + M_4 - M_5 + M_7 \\
C_{1,2} &= M_3 + M_5 \\
C_{2,1} &= M_2 + M_4 \\
C_{2,2} &= M_1 - M_2 + M_3 + M_6
\end{aligned}$$

Les matrices M_1 à M_7 sont également calculées avec la méthode se Strassen récursivement. Le nombre d'opérations effectué vérifie alors $c(n) = 7c(n/2) + O(n^2)$. Ce qui donne alors $O(n^{\log_2(7)})$ opérations.

2 Implémentation naïve

Pour ce TP, on supposera $N = 8$ et on manipulera donc des matrices de cette taille. Nos matrices seront des double**.

Commencez donc votre programme par un :

```
#Define N 8

double** A;
double** B;
double** C;
```

Question 0

Allouez l'espace mémoire de A, B et C dans le main.

Question 1

Écrire une fonction affiche_mat de type void qui prend en entrée une matrice A de taille $N * N$ et un entier $k \leq N$ et qui affiche la matrice extraite de A entre les lignes et colonnes 0 à $(k-1)$.

Question 2

Écrire une fonction genere_matrice de type void qui prend en entrée une matrice A de taille $N * N$ et la remplie de coefficient aléatoire entre -10 et 10. On utilisera la fonction $rand()$ de la librairie stdlib.

Question 3

Écrire une fonction mult_naive de type void qui prend en entrée trois matrices de tailles $N * N$ A, B et C et qui remplace C par $A * B$.

3 Construction de l'algorithme

Le but de cette section est de réfléchir à la décomposition de l'algorithme afin de l'écrire de façon optimale.

Question 4

Quelle(s) opération(s) est(sont) à effectuer plusieurs fois dans la méthode de Strassen et nécessiteraient une fonction auxiliaire ?

Question 5

Écrire le pseudo-code de l'algorithme de Strassen. Quelles opérations "faciles" du pseudo-code nécessiteraient une fonction auxiliaire en C pour être codées ?

Question 6

Écrire une fonction Strassen de type void qui prend en entrée trois matrices A, B et C et qui remplace C par le produit AB et utilisant la méthode de Strassen. On pourra commencer par coder les fonctions auxiliaires qui nous semblent utiles.

Question 7

Améliorer la fonction précédente pour qu'elle fonctionne également si N n'est pas une puissance de 2. On pourra considérer le plus petit entier N' plus grand que N qui est une puissance de 2 et construire une matrice de taille N'.