Multiplication des matrcies : Algorithme de Strassen

Le but ce projet est de présenter un algorithme efficace (c'est-à-dire économe en coût) pour calculer le produit de deux matrices. Lorsqu'on fait calculer à une machine le produit de deux matrices, les multiplications entre les coefficients des matrices coûtent plus de ressources machines que les additions. L'algorithme de Strassen est donc un algorithme qui utilise moins de multiplication que l'algorithme (formule) classique pour le calcul du produit de deux matrices.

1 Algorithme de Strassen (Matrices 2×2)

Dans cette partie on s'intéresse au nombre de multiplications effectuées lorsqu'on calcule le produit de deux matrices. L'algorithme de Strassen présenté dans cette partie permet de diminuer ce nombre.

Soit le produit de deux matrices de taille 2×2,

$$A \times B = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \times \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix} = C$$

1. Combien de multiplications sont nécessaires pour calculer tous les coefficients c_{ij} par la définition du produit matriciel vue en cours ?

2. Considérons les produits suivants :

$$\text{(Formules de Strassen)} \begin{cases} q_1 = (a_{11} - a_{12})b_{22} \\ q_2 = (a_{21} - a_{22})b_{11} \\ q_3 = a_{22}(b_{11} + b_{21}) \\ q_4 = a_{11}(b_{12} + b_{22}) \\ q_5 = (a_{11} + a_{22})(b_{22} - b_{11}) \\ q_6 = (a_{11} + a_{21})(b_{11} + b_{12}) \\ q_7 = (a_{12} + a_{22})(b_{21} + b_{22}) \end{cases}$$

Vérifier qu'on retrouve la matrice C en effectuant les opérations suivantes :

$$\begin{aligned} c_{11} &= q_1 - q_3 - q_5 + q_7 \\ c_{12} &= q_4 - q_1 \\ c_{21} &= q_2 + q_3 \\ c_{22} &= -q_2 - q_4 + q_5 + q_6 \end{aligned}$$

3. Conclure que les formules de Strassen permettent de calculer le produit $A \times B$ en effectuant 7 multiplications. Qu'a-t-on gagné ?

2 Produit matriciel par blocs

1. Produit par blocs: soit A, B deux matrices de tailles $2n \times 2n$

$$A = \left(\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right) \text{ et } B = \left(\begin{array}{c|c} B_{11} & B_{12} \\ \hline B_{21} & B_{22} \end{array}\right).$$

Les blocs $A_{11}, \dots, B_{22}$ sont des matrices $n \times n$.

Le calcul du produit $A \times B$ peut se faire par blocs à partir des produits des matrices $A_{ik} \times B_{kj}$ de la manière suivante:

$$C = A \times B = \left(\begin{array}{c|c} C_{11} & C_{12} \\ \hline C_{21} & C_{22} \end{array}\right) = \left(\begin{array}{c|c} A_{11} \times B_{11} + A_{12} \times B_{21} & A_{11} \times B_{12} + A_{12} \times B_{22} \\ \hline A_{21} \times B_{11} + A_{22} \times B_{21} & A_{21} \times B_{12} + A_{22} \times B_{22} \end{array}\right)$$

Calculer par blocs $C = A \times B$ où A et B sont les matrices de taille 4×4 définies par

$$A = \begin{pmatrix} 1 & 2 & 0 & 1 \\ 3 & 4 & -1 & 1 \\ 1 & 0 & 1 & 2 \\ 0 & 1 & 3 & 4 \end{pmatrix}, B = \begin{pmatrix} 1 & -1 & 0 & 1 \\ 2 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & -1 \end{pmatrix}$$

3 Algorithme de Strassen (cas général)

1. Soient A et B deux matrices de tailles $n \times n = 2^k \times 2^k$. On découpe ces matrices en blocs (les sous-matrices sont de taille $2^{k-1} \times 2^{k-1}$) :

$$A = \left(\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right) \text{ et } B = \left(\begin{array}{c|c} B_{11} & B_{12} \\ \hline B_{21} & B_{22} \end{array}\right)$$

et on note $C = \left(\begin{array}{c|c} C_{11} & C_{12} \\ \hline C_{21} & C_{22} \end{array}\right)$ le produit $A \times B$.

Écrire les formules de Strassen par blocs et vérifier qu'elles permettent de calculer C.

2. Soit u_k le nombre de multiplications nécessaires pour calculer le produit de deux matrices A et B de tailles $2^k \times 2^k$. Montrer que si on applique les formules de Strassen on a $u_k = 7u_{k-1}$.

3. Toujours en utilisant la méthode de Strassen, que vaut u_1 ?.

4. En déduire qu'il faut $u_k = 7^k$ multiplications pour réaliser le produit de deux matrices de taille $2^k \times 2^k$.

5. Posons $n = 2^k$ et montrer que $u_k = n^{\frac{\ln(7)}{\ln(2)}} \simeq n^{2.81}$ (on pourra poser $u_k = n^\alpha$ et chercher à déterminer α).

6. Si on applique la méthode classique vue en cours pour calculer le produit de deux matrices $A \times B$ de taille $n \times n$ combien de multiplications en fonction de n sont nécessaires ?

7. Application numérique : pour multiplier deux matrices de taille 128×128 combien de multiplications économise-t-on en utilisant les formules de Strassen à la place du calcul classique ?

Remarque : Le cas général où A et B sont des matrices de tailles $n \times n$ avec $2^{k-1} < n < 2^k$ se résout en complétant les matrices par des zéros de sorte à obtenir deux matrices de tailles $2^k \times 2^k$.

4 Implémentation

Une fois le principe de l'algorithme établi vous effecturez les taches suivantes:

- Implémenter l'algorithme de Strassen **et** l'algorithme de multiplication classique pour effecuter le produit de deux matrices **carrées** quelconques.
- Estimer l'entier k à partir du quel l'algorithme de Strassen est "plus rapide" que l'algorithme classique pour calculer le produit de deux matrices carrées de taille 2^k (vous pourrez tester votre estimation en construisant des matrices aléatoires de taille 2^k). On prendra soin de détailler et d'expliquer la démarche et le raisonnement utilisés pour cette estimation.
- Dans quel(s) cas l'utilisation de l'algorithme de Strassen pour le produit de deux matrices n'est pas avantageuse (ou moins efficace) ?
- Exsite-t-il des algorithmes plus performants que l'algorithme de Strassen, en terme de complexité, pour effectuer le produit de deux matrices ?

5 Rendu du projet

Le travail sur ce projet est à faire par binôme (2 personnes maximum). Le travail sera restitué sous la forme d'un rapport et de code source (et éventuels exectuables). Le projet est à rendre sur Moodle avant le 21 janvier 14h, c'est à dire avant le début de la première séance du prochain projet.

Le rapport doit introduire le sujet, répondre à toutes les questions posées dans l'énoncé, expliquer le fonctionnement des algorithmes implémentés, expliquer comment ces algorithmes ont été implémentés en pratique. On pourra terminer le rapport par une discussion des résultats et une conclusion. La qualité, propreté et clarté de la rédaction seront pris en compte.

Le code source relatif aux différentes implémentations demandées devra également être rendu avec le rapport. On choisira un des langages suivants pour l'implémentation des algorithmes : Java, Python, C, C++, Matlab et Maple. Le code devra être commenté, et devra pouvoir être executé et/ou compilé pour vérifier la validité des algorithmes.