Algorithme de

STRASSEN

" the man who changed Matrix Multiplication game "

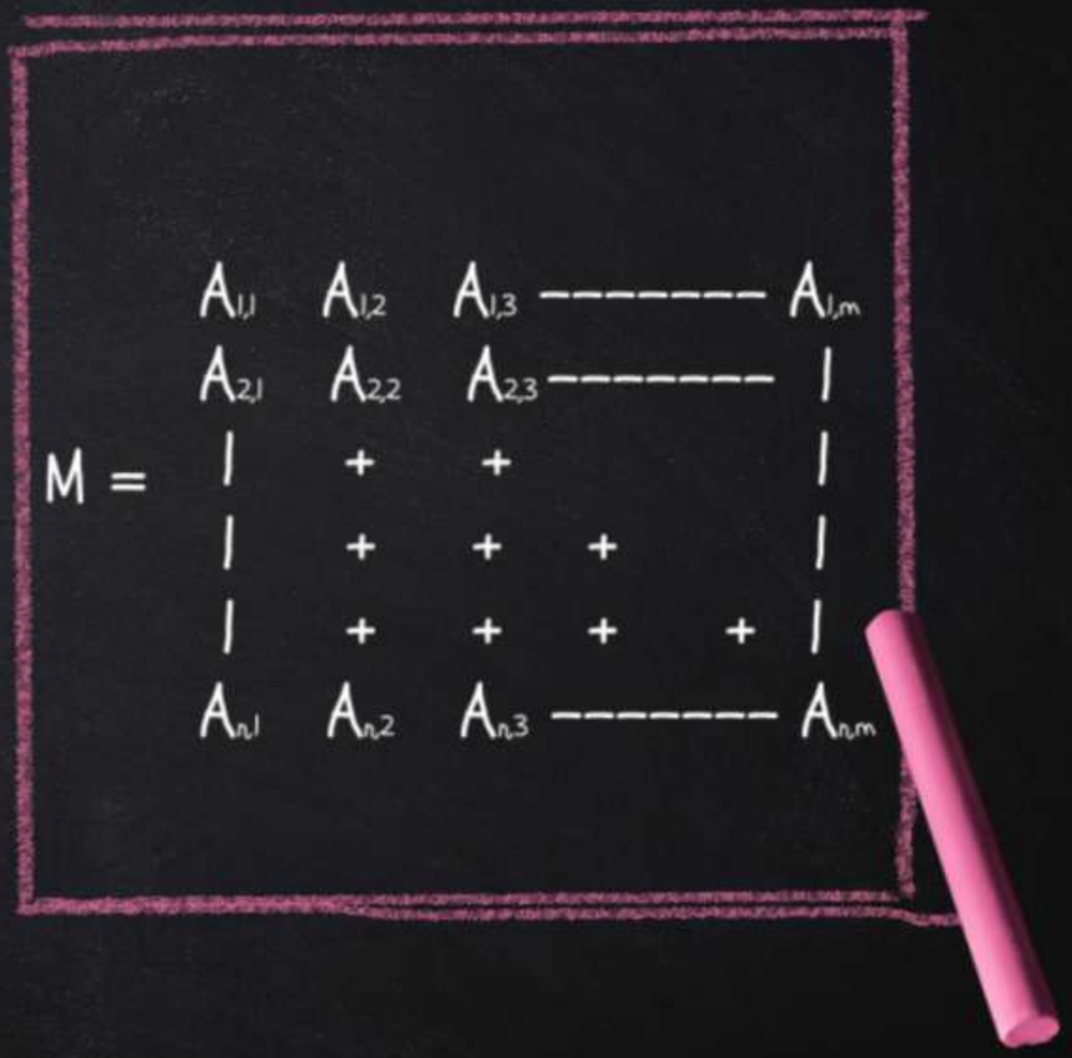
INDEX

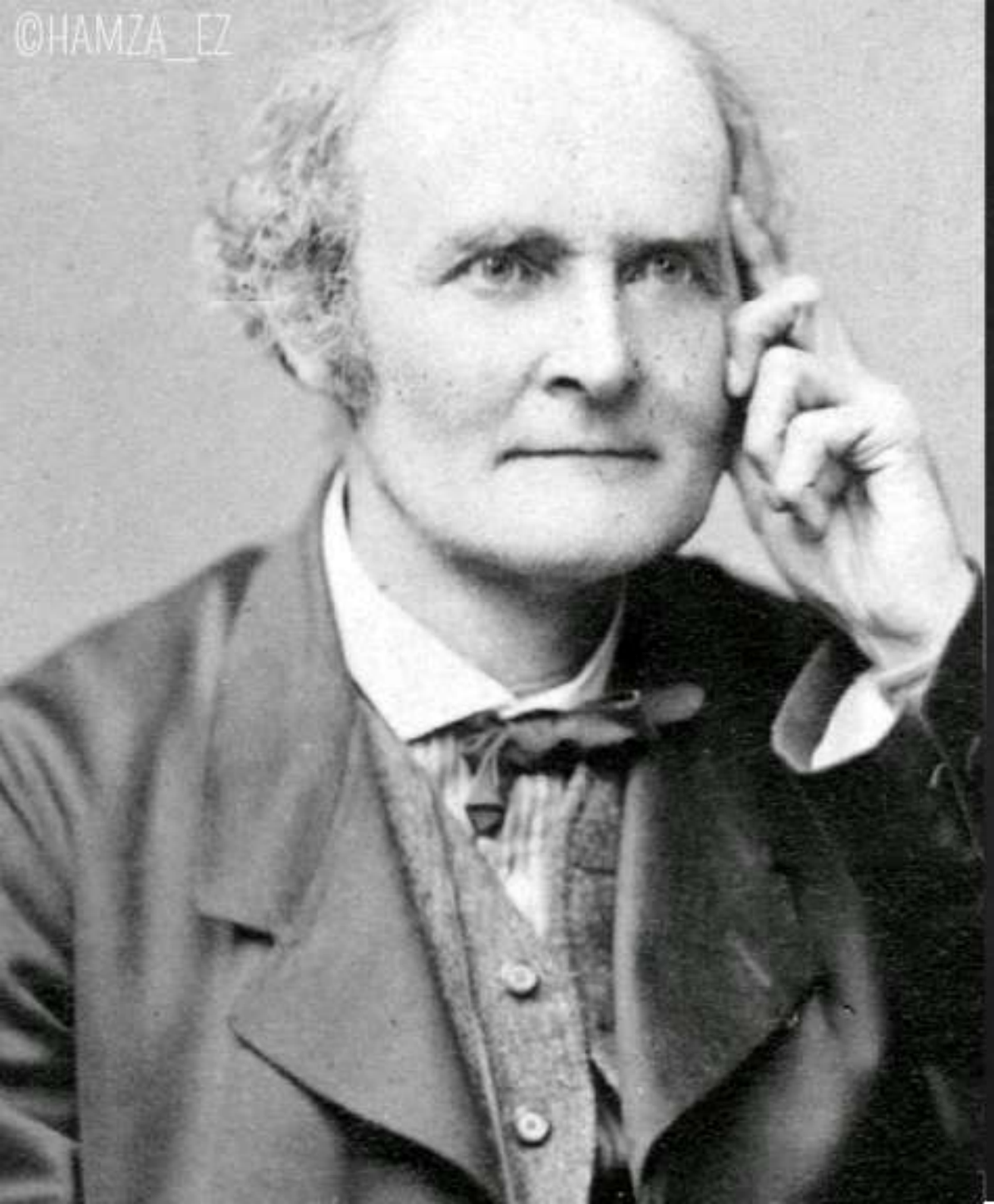
1 INTRODUCTION DE BASE

1. La notion Matrice
2. Le produit des Matrices
3. Méthode Classique de produit matriciel

1

LA NOTION MATRICE





PETITE HISTORIQUE :

- le calcul matriciel est apparaisse qu'au début du xixe siècle, les matrices, en tant que tableaux de nombres.

En 1854, Arthur Cayley définit les opérations usuelles du calcul matriciel (addition, multiplication et division).

puis, beaucoup de traitement et opération entre les matrices est apparu, l'une les plus connue c'est le produit matriciel.

Matrice (Matrix)

Une matrice à m lignes et n colonnes est un tableau rectangulaire de n × m nombres, rangés ligne par ligne. Dans chaque lignes on a m éléments (m colonne).

$$M = \begin{pmatrix} A_{1,1} & A_{1,2} & A_{1,3} & \cdots & A_{1,m} \\ A_{2,1} & A_{2,2} & A_{2,3} & \cdots & | \\ | & + & + & & | \\ | & + & + & + & | \\ | & + & + & + & + \ | \\ A_{n,1} & A_{n,2} & A_{n,3} & \cdots & A_{n,m} \end{pmatrix}$$

2

LE PRODUIT MATRICIEL

LE PRODUIT MATRICIEL

- Le produit de matrices est une opération qui intervient souvent dans les calculs informatiques et numériques. C'est typiquement le genre d'opération assez lourd à mener. Ce n'est pas sa complexité qui est embarassante mais son aspect répétitif dès que la taille de la matrice devient importante .

- Pour que le produit de deux matrices soit défini, il faut que le nombre de colonnes de la première matrice soit égal au nombre de lignes de la deuxième. C-à-d :

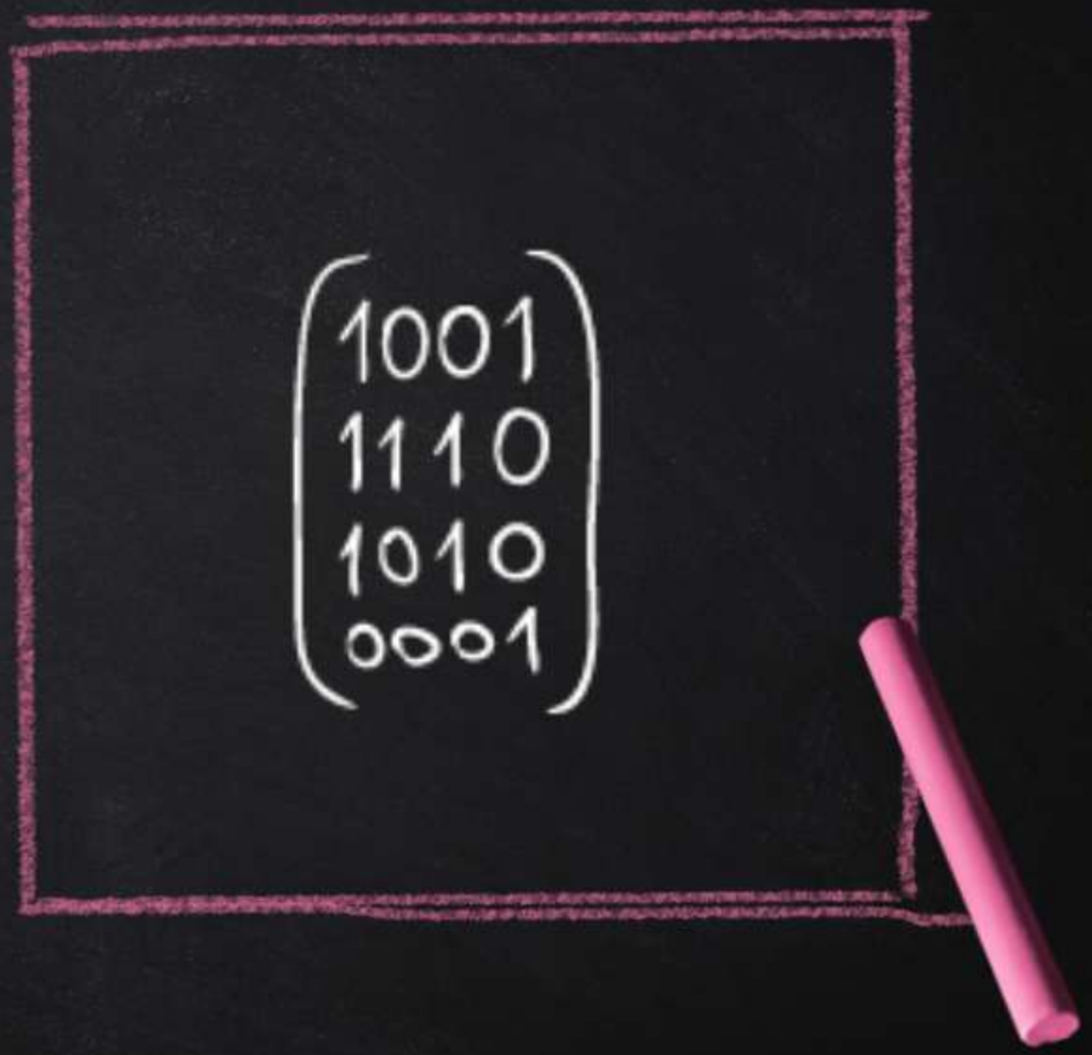
  Si A une matrice de taille (n,m) et B Matrice de taille (p,q) , le produit Matriciel A*B est défini Si et seulement si m = q .

$$(m, n) * (n, q)$$

Le produit est défini

3

METHODE CLASSIQUE



METHODE CLASSIQUE

Soient A et B des matrices de taille successivement (n , m) et (m , p), Le produit matriciel est une matrice C = A * B de taille (n , p) :

Le principe est de multiplie chaque element de chaque ligne de A par chaque element de chaque colone de B, puis fait la somme C-à-d :

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix} * \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1p} \\ b_{21} & b_{22} & \cdots & b_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{mp} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1p} \\ c_{21} & c_{22} & \cdots & c_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{np} \end{bmatrix}$$

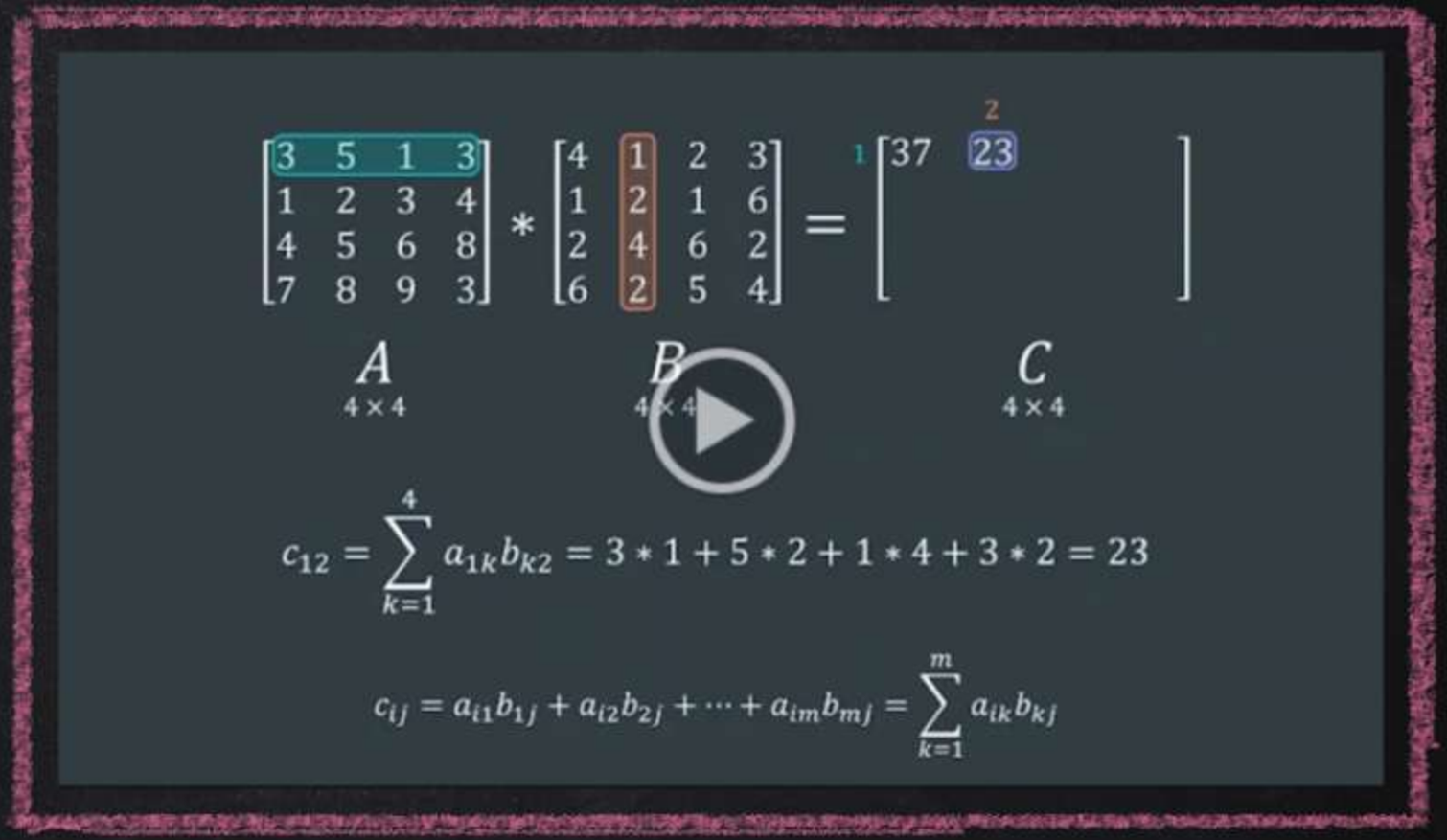
$$\underset{n \times m}{A} \qquad \underset{m \times p}{B} \qquad \underset{n \times p}{C}$$

Avec :

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{im}b_{mj} = \sum_{k=1}^{m} a_{ik}b_{kj}$$

EXEMPLE ANIME

Produit de deux matrice A et B de taille 4 * 4 :



CONCLUSION ET COMPLEXITE

- (M , N) * (N , P) : Produit défini.
- (M , N) * (Q , P) : Produit n'es Pas défini

- La Complexite de La méthode Classique est :

$$O(n3)$$

Volker Strassen a proposé un algorithme calculant le produit de deux matrices carrées de taille n_0 et moins complexité que la méthode classique.

En 1969, l'Algorithme de strassen est proposé, La complexité de l'algorithme est en O(n2,807), avec pour la première fois un exposant inférieur à celui de la multiplication classique.

Historique

Volker Strassen, né le 29 avril 1936 à Düsseldorf1, est un mathématicien allemand, actuellement professeur émérite à l'université de Constance.

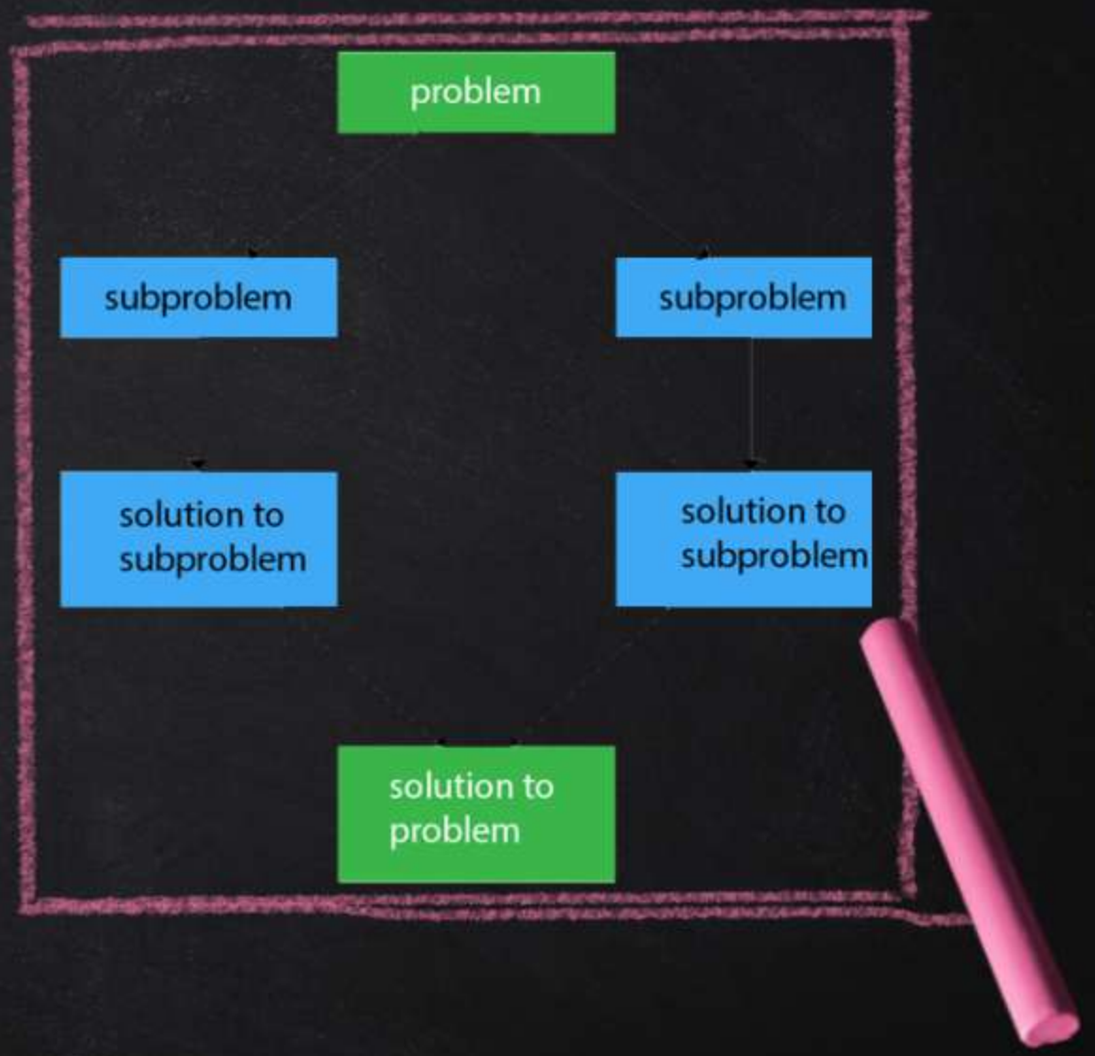
Il est célèbre pour son travail sur la complexité algorithmique des opérations de base en calcul formel et en théorie algorithmique des nombres. Parmi ses contributions majeures, on peut citer l'algorithme de Strassen (1969) pour le produit matriciel,

METHODE STRASSEN

1. Divide and conquer
2. Les étape de l'Algorithme
3. complexité
4. Algorithme Strassen

1

DIVIDE AND CONQUER



DIVIDE AND CONQUER

On prend un problème (généralement complexe à résoudre), on divise ce problème en une multitude de petits problèmes, l'idée étant que les "petits problèmes" seront plus simples à résoudre que le problème original. Une fois les petits problèmes résolus, on recombine les "petits problèmes résolus" afin d'obtenir la solution du problème de départ.

Le paradigme "diviser pour régner" repose donc sur 3 étapes :

" *découper un problème initial en sous-problèmes* "

Diviser

" *résoudre les sous-problèmes (récursivement ou directement s'ils sont assez petits)* "

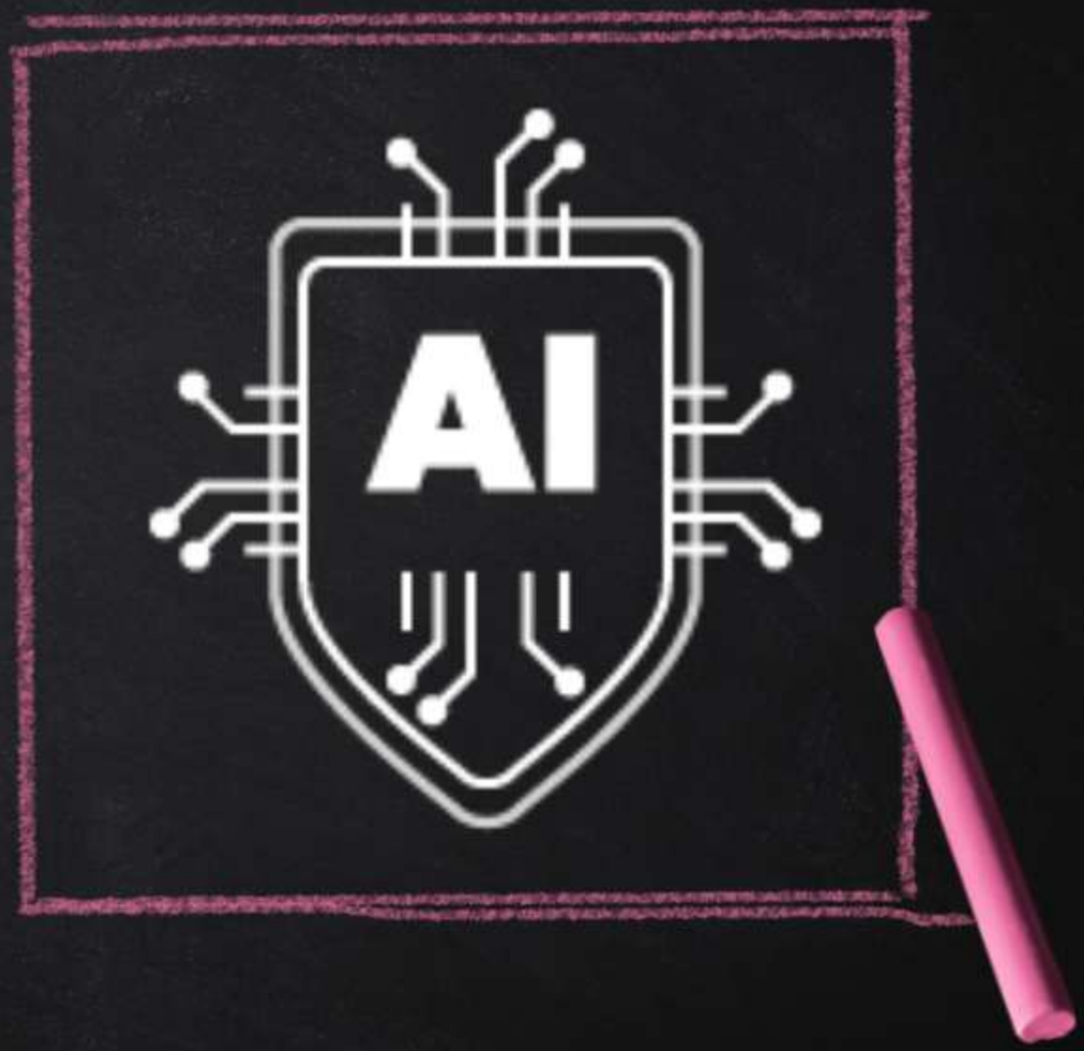
Régner

" *calculer une solution au problème initial à partir des solutions des sous-problèmes.* "

Combiner

1

LES ETAPES DE L'ALGORITHME



Principe

En 1969, Volker Strassen imagine un algorithme permettant de descendre le nombre de multiplications en dessous de N3.
L'algorithme est basée sur le produit Classique de matrices 2x2:

$$\underset{A}{\begin{bmatrix} a & b \\ c & d \end{bmatrix}} * \underset{B}{\begin{bmatrix} e & f \\ g & h \end{bmatrix}} = \underset{C}{\begin{bmatrix} ae + bg & af + bh \\ ce + dg & cf + dh \end{bmatrix}}$$

$A_{2\times 2}$ $B_{2\times 2}$ $C_{2\times 2}$

Cela nécessite 8 multiplications et 4 additions. L'idée de Volker Strassen est de trouver un moyen de multiplier les matrices A et B en utilisant seulement 7 multiplications au lieu de 8 par des combinaisons d'opérations astucieuses entre les éléments de A et B.

COMBINAISON DE STRASSEN

- M1 = (a + d) * (e + h)
- M2 = (c + d) * e
- M3 = a * (f - h)
- M4 = d * (g - e)
- M5 = (a + b) * h
- M6 = (c - a) * (e + f)
- M7 = (b - d) * (g + h)

$$\underset{2\times2}{A} * \underset{2\times2}{B} = \underset{2\times2}{C}$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} * \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae + bg & af + bh \\ ce + dg & cf + dh \end{bmatrix}$$

$$C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

- en utilisant seulement 7 multiplications au lieu de 8. AVEC :

$$C_{11} = M_1 + M_4 - M_5 + M_7 \qquad C_{12} = M_3 + M_5$$

$$C_{21} = M_2 + M_4 \qquad C_{22} = M_1 - M_2 + M_3 + M_6$$

ALGORITHME DE STRASSEN

l'idée de Strassen est de appliquer la méme logique meme si A et B sont de taille n, mais cette fois les a, b, c, d, e, f et g serons des matrices de taille inferieur de n et exactement de taille n/2.

En général, soit A et B deux matrices carrées de taille n, Pour calculer le produit C = A * B, nous suivons les étapes suivantes

ETAPE 01

Vérifier la parité de n :

- Si n est pair on ne fair rien dans cette étape.
- Si n est impaire, il suffit d'augmenter la taille de la matrice A et B à n+1, puis en ajoutant les colonnes et les lignes de ZEROS nécessaires.

ETAPE 02

Division des matrices A, B et C :

- devise A, B et C en des 4 Matrice de taille n/2.

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \quad B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{11} \end{bmatrix} \quad C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

ALGORITHME DE STRASSEN

avec Aij , Bij et Cij de taille n/2, la méthode Classique serait :

$$\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} = \begin{bmatrix} A_{11}*B_{11} + A_{12}*B_{21} & A_{11}*B_{12} + A_{12}*B_{22} \\ A_{21}*B_{11} + A_{22}*B_{21} & A_{21}*B_{12} + A_{22}*B_{22} \end{bmatrix}$$

encore fois besoin de 8 multiplications de blocs matriciels pour calculer les matrices C .

ETAPE 03

réduit le nombre de multiplication par la combinaison de l'algorithme de Strassen.

$$M_1 = (A_{11} + A_{22}) * (B_{11} + B_{22})$$
$$M_2 = (A_{21} + A_{22}) * B_{11}$$
$$M_3 = A_{11} * (B_{12} - B_{22})$$
$$M_4 = A_{22} * (B_{21} - B_{11})$$
$$M_5 = (A_{11} + A_{12}) * B_{22}$$
$$M_6 = (A_{21} - A_{11}) * (B_{11} + B_{12})$$
$$M_7 = (A_{12} - A_{22}) * (B_{21} + B_{22})$$

ALGORITHME DE STRASSEN

en n'utilisant que 7 multiplications (une pour chaque M_k) au lieu de 8. Nous pouvons maintenant exprimer le Cij en termes de M_k

ETAPE 03

$$C_{11} = M_1 + M_4 - M_5 + M_7$$

$$C_{22} = M_1 - M_2 + M_3 + M_6$$

$$C_{21} = M_2 + M_4 \; ; \; C_{12} = M_3 + M_5$$

ETAPE 04

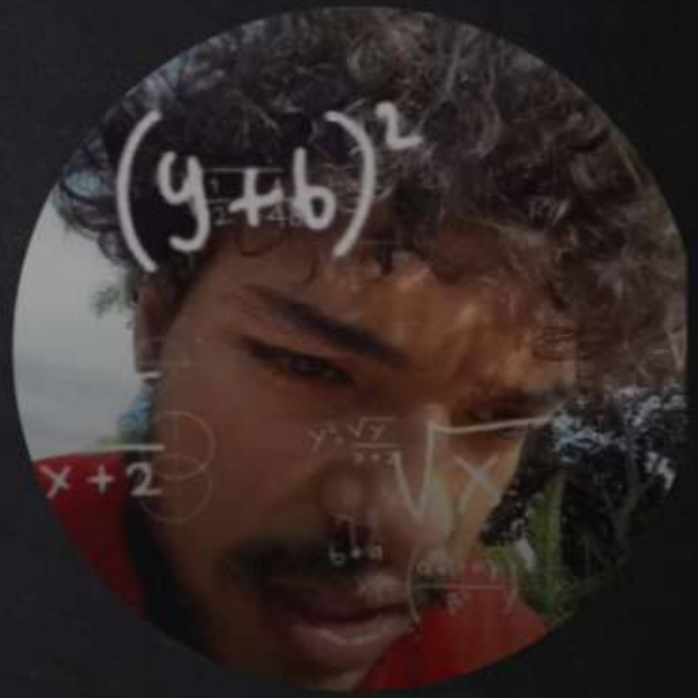
Enfin :

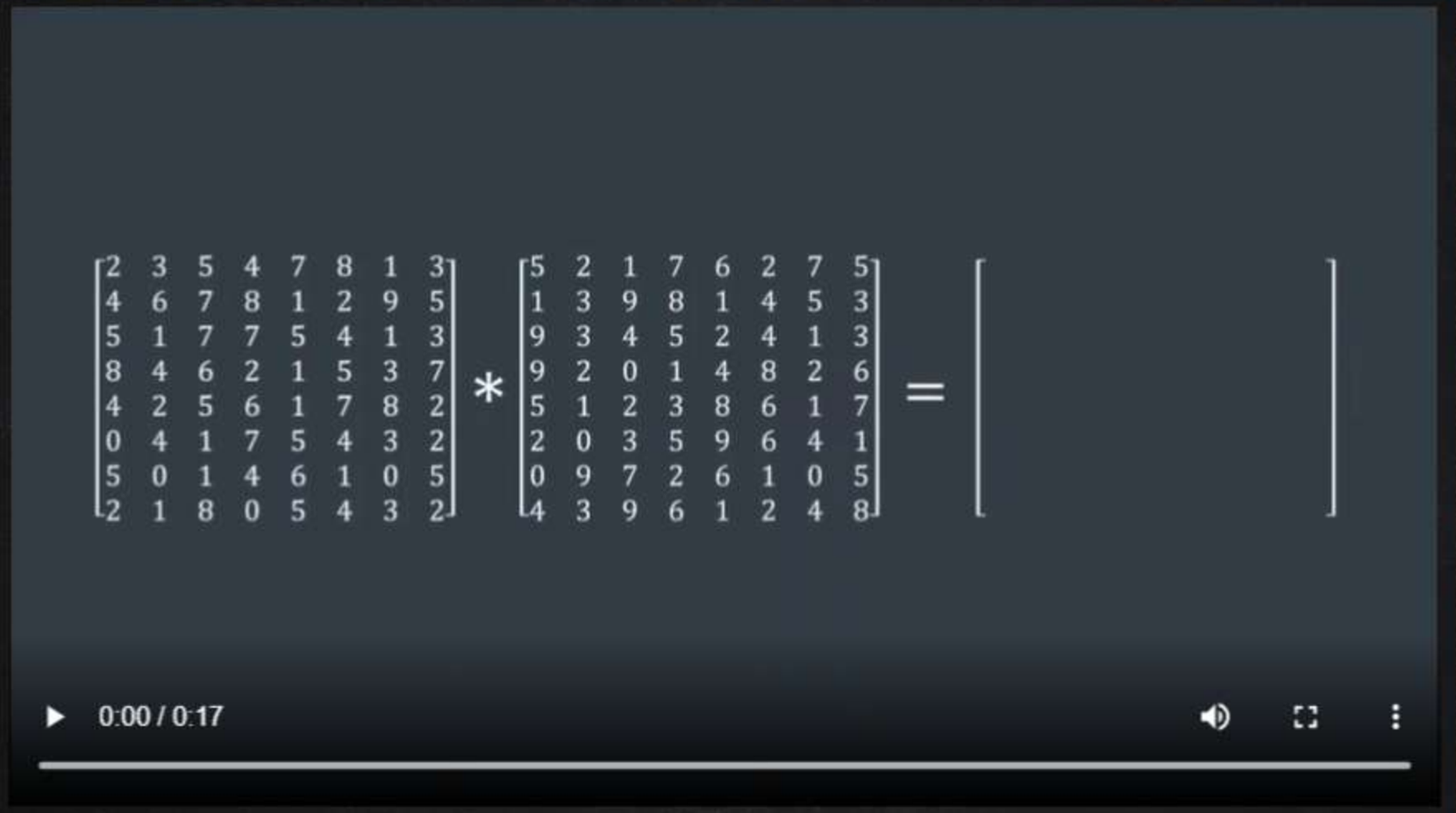
Nous itérons récursivement ce processus de division jusqu'à ce que les sous-matrices dégénèrent en nombres. Si, comme mentionné précédemment, la matrice d'origine avait une taille qui n'était pas une puissance de 2, alors le produit résultant aura zéro lignes et colonnes tout comme A et B, et celles-ci seront ensuite supprimées à ce stade pour obtenir la matrice C nous voulions vraiment.

EXEMPLE

- calculer le produit matriciel de A et B de taille 8*8 à l'aid de l'Algorithme de Strassen.

a				b					e				f			
2	3	5	4	7	8	1	3		5	2	1	7	6	2	7	5
4	6	7	8	1	2	9	5		1	3	9	8	1	4	5	3
5	1	7	7	5	4	1	3		9	3	4	5	2	4	1	3
8	4	6	2	1	5	3	7		9	2	0	1	4	8	2	6
4	2	5	6	1	7	8	2		5	1	2	3	8	6	1	7
0	4	1	7	5	4	3	2		2	0	3	5	9	6	4	1
5	0	1	4	6	1	0	5		0	9	7	2	6	1	0	5
2	1	8	0	5	4	3	2		4	3	9	6	1	2	4	8
c				d					g				h			





+info

ANIMATION DE PROCESSUS

LE CODE DE L'ALGORITHME

```
Fonction Strassen(A,B: matrices; N: entier): matrices;

Si N n'est pas une puissance de 2
   Alors on ajoute des lignes et des colonnes de 0 afin d'acceder a la
   puissance de 2 la plus proche superieurement.

Si N=1 Alors Strassen = A * B;

Sinon {
        On partitionne A et B en blocs de taille N/2;

        M1 = Strassen(A11 + A22, B11 + B22, N/2);
        M2 = Strassen(A21 + A22, B11, N/2);
        M3 = Strassen(A11, B12 - B22, N/2);
        M4 = Strassen(A22, B21 - B11, N/2);
        M5 = Strassen(A11 + A12, B22, N/2);
        M6 = Strassen(A21 - A11, B11 + B12, N/2);
        M7 = Strassen(A12 - A22, B21 + B22, N/2);

        C11 = M1 + M4 - M5 + M7;
        C12 = M3 + M5;
        C21 = M2 + M4;
        C22 = M1 - M2 + M3 + M6;
      }
Fin Fonction
```

COMPARAISON AVEC LA MÉTHODE CLASSIQUE

Méthode de Classique

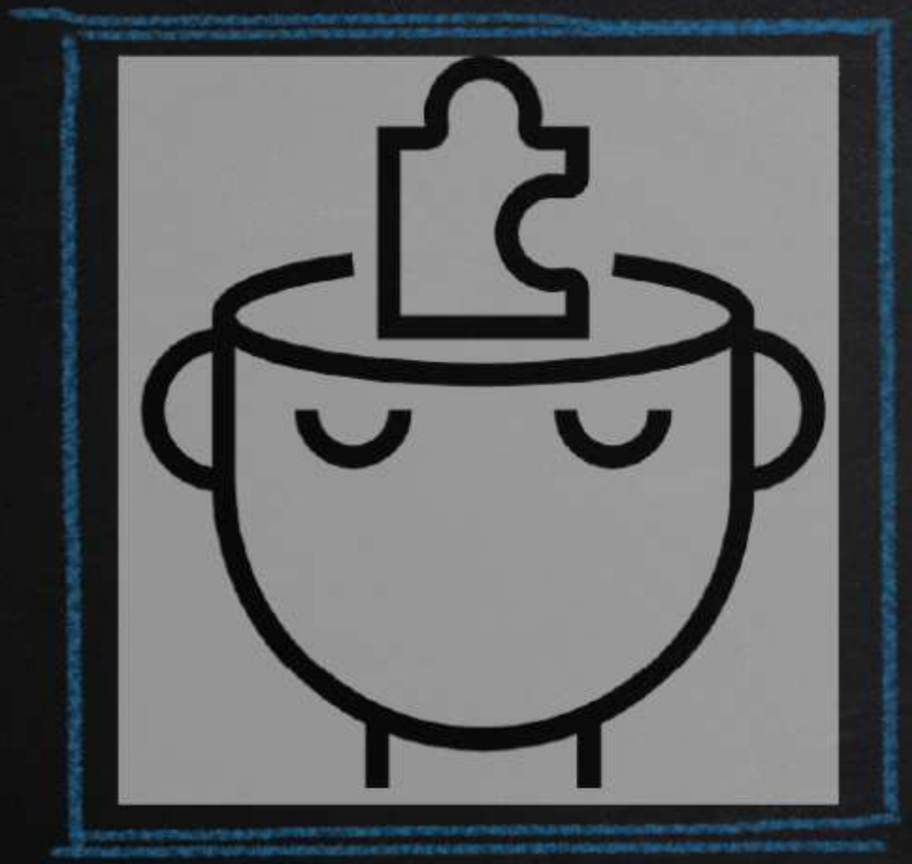
- 8 multiplications et 4 additions
- O(n^3) time complexity
- non efficace pour les matrices de taille n > 100

VS

Méthode Strassen

- juste 7 multiplications et 18 additions
- O(N^2,8074) time complexity
- les sous-matrices recursives occuppent de l'espace et prennent beaucoup de temps a repondre
- risque d'erreurs pour les nombres non naturels

Le meilleur cas de méthode classique C'est le pire de cas de méthode Strassen



VS

Le meilleur cas de méthode Strassen C'est le pire de cas de méthode classique

3 IMPLEMENTATION SOUS PYTHON

- Fonctions Utilisée.
- Exemple d'exécusion.

```python
def split(Matrix):
    n = len(Matrix)
    return (
        Matrix[: n // 2, : n // 2],
        Matrix[: n // 2, n // 2 :],
        Matrix[n // 2 :, : n // 2],
        Matrix[n // 2 :, n // 2 :],
    )
```

Diviser La Matrice en 4 sous Matrice de taille N/2

```python
def Add_Zeros(Matrix):
    n = len(Matrix)
    if n % 2 == 0:
        return Matrix
    else:
        B = np.array([np.zeros(n)])
        D = np.array([np.zeros(n + 1)])
        c = np.concatenate((Matrix, B), axis=0)
        c = np.concatenate((c, D.T), axis=1)
        return c
```

Ajoute des lignes et des colonnes des ZEROS

```python
# X : La taille de matrice
# et Y : la taille de matrice apres l'ajoute des zeros

def dlt_Zeros(Matrix, x, y):
        i = int(math.log2(y) - 1)
        Matrix = Matrix[
            : (x + 1) - int(math.log2(x + 1) - i),
            : (x + 1) - int(math.log2(x + 1) - i),
        ]
    return Matrix
```

Supprimer les lignes et les colonnes de ZEROS

```python
def Strassen(A, B):
    n = len(A)
    k = len(A)
    if n % 2 != 0:
        A = Add_Zeros(A)
        B = Add_Zeros(B)
        n = n + 1
    if n <= 2:
        C = np.dot(A, B)
    else:
        A11, A12, A21, A22 = split(A)      #Division des Matrices
        B11, B12, B21, B22 = split(B)
        M1 = Strassen(A11 + A22, B11 + B22)   #Application des combinaison de Strassen
        M2 = Strassen(A21 + A22, B11)
        M3 = Strassen(A11, B12 - B22)
        M4 = Strassen(A22, B21 - B11)
        M5 = Strassen(A11 + A12, B22)
        M6 = Strassen(A21 - A11, B11 + B12)
        M7 = Strassen(A12 - A22, B21 + B22)
        C11 = M1 + M4 - M5 + M7
        C12 = M3 + M5
        C21 = M2 + M4
        C22 = M1 - M2 + M3 + M6
        C = np.vstack((np.hstack((C11, C12)), np.hstack((C21, C22)))) #Elaboration du Matrice C
        C = dlt_zero(C, k, n)    #supprimer les ligne et les colonne des ZEROS
    return C
```

Fonction main de L'algorithme

```
La Matrice A :

[[ 1  2  3  2  3]
 [ 0  4  6  2  3]
 [ 1  5  1  5  6]
 [ 0  4  6  2  3]
 [ 9 10 11  6  7]]

La Matrice B :

[[ 5  2  1  6  7]
 [ 1  3  9 10 11]
 [ 9  3  4  1  5]
 [ 1  5  1  5  6]
 [ 1  2  3  2  3]]

Le Produit A * B avec A'lgo Strassen :

[[ 39.  33.  42.  45.  65.]
 [ 63.  46.  71.  62.  95.]
 [ 30.  57.  73.  94. 115.]
 [ 63.  46.  71.  62.  95.]
 [167. 125. 170. 209. 285.]]
```

EXEMPLE