**INF4705 : ANALYSE ET CONCEPTION D’ALGORITHMES**

***RAPPORT DE LABORATOIRE DU TP1***

**Réalisé par :**

xxxxxxxxxx#yyyyyyyyyyy

CODO Hermann R.C #1715995

**Remis-le :**

15/10/2017



*Description du sujet et objectifs de ce travail*

Le laboratoire 1 du cours INF4705 porte sur l’analyse empirique et hybride des algorithmes de multiplication de matrice *conventionnel* et celle de Strassen faite à partir du patron de conception diviser pour régner.

Ainsi il nous a été demandé une implantation des algorithmes conventionnel, de Strassen et celle de Strassen avec un seuil de récursivité dont les valeurs en dessous demanderont celle conventionnel. Ces algorithmes sont implantés sur des exemplaires pour effectuer des produits de matrices carrées de taille 2n, n allant de 6 à 10.

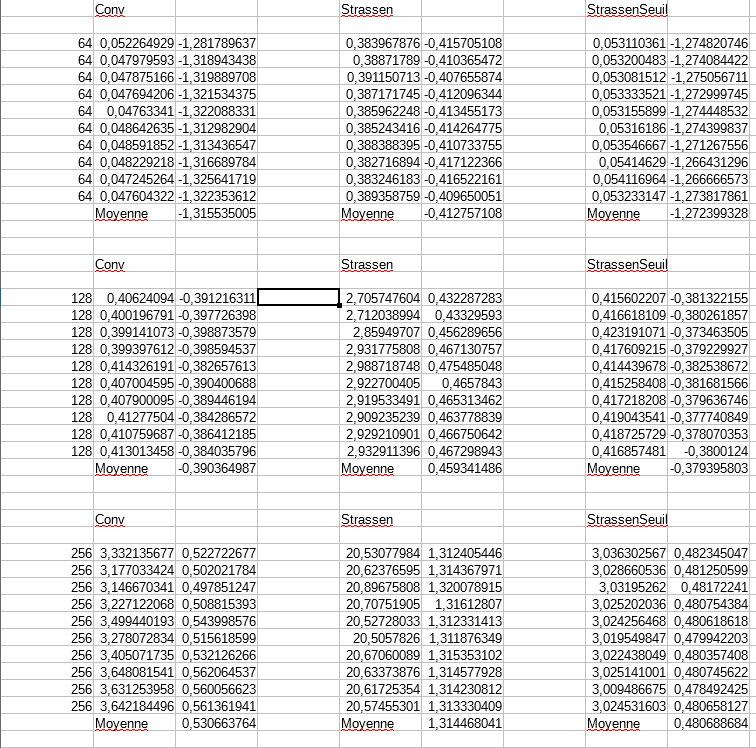
Pour l'analyse de ces algorithmes trois approches d’analyse sont demandées à savoir le test de puissance, le test de rapport et le test des constantes ; tout cela dans l'objectif de faire une comparaison des consommations de ressource en temps entre les algorithmes ci-dessus.

*Description complète du jeu de données*

Pour réaliser nos tests, nous avons par taille de matrice un dossier contenant cinq fichiers de données de nomenclature ex\_n.x ( n etant 2n de la taille de matrice et x allant de 1 a 5 ).

Dans chacun des 5 dossiers nous avons donc 5 matrices que nous multiplions deux a deux donc un total de 10 multiplications par taille de matrice.

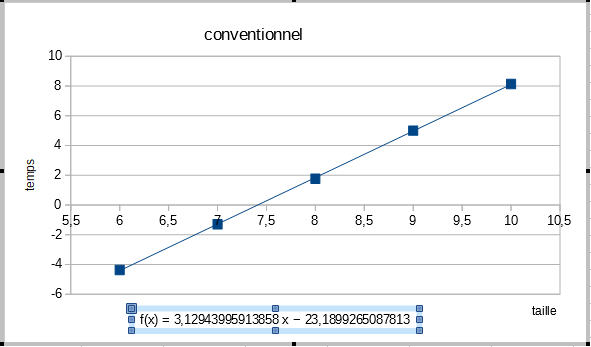
*Résultats expérimentaux*

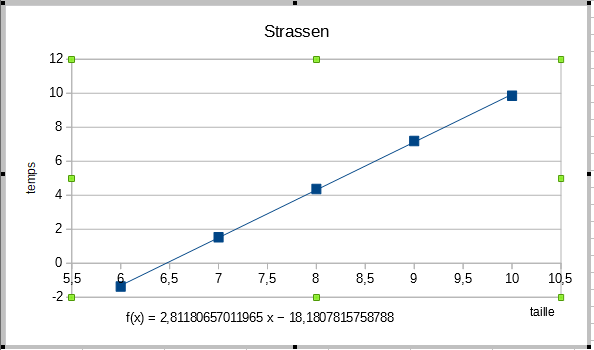


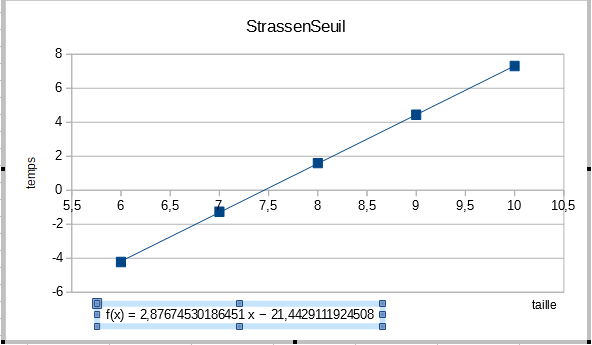


*Analyse*

I) Test de puissance





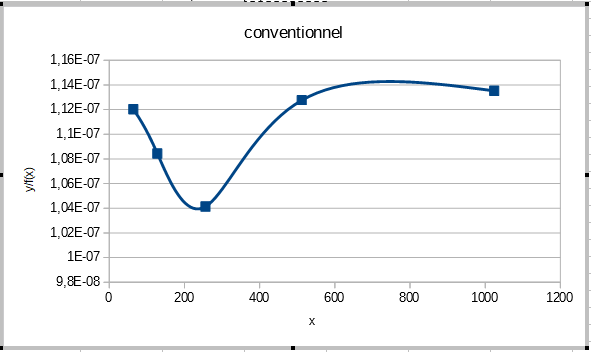


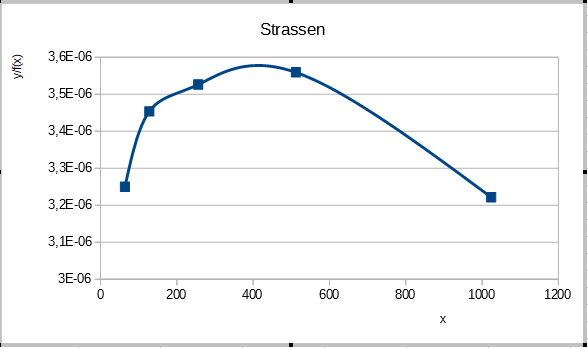
Pour effectuer notre test de puissance nous avons calculé les logarithmes en base 2 de nos x et y respectivement la taille et le temps d'exécution. On peut aisément constater à partir des différents graphes obtenus pour chacun des algorithmes que tous les points appartiennent à une même droite ; on peut donc conclure que chacun des algorithmes s'exécute en temps polynomial.

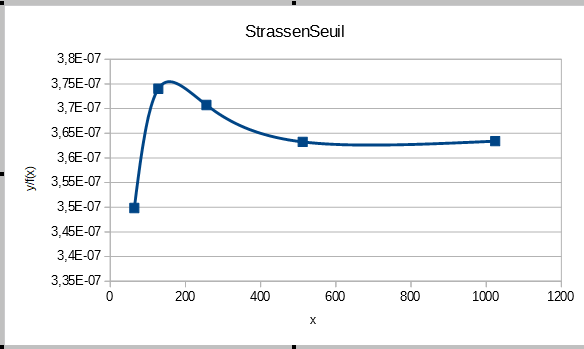
II)

* Pour notre algorithme conventionnel : O (n3,12)
  + Ab = 2-23,19 = 1.145E-7 , M = 3,12
* Pour l'algorithme de Strassen : O(n2.81)
  + Ab = 2-18.18 = 3,36E-6 , M = 2,81
* Pour l'algorithme de StrassenSeuil : O(n2.87)
  + Ab = 2-21.44 = 3,5149E-7 , M = 2,87

III) Test de de rapport





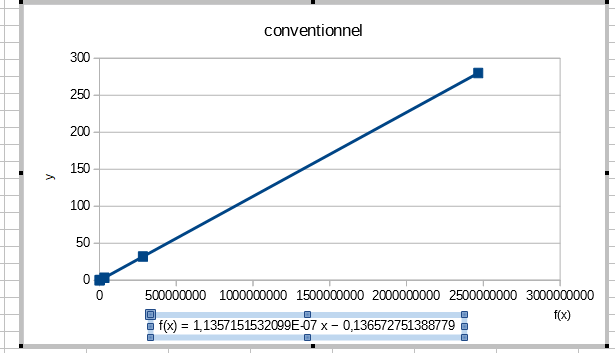


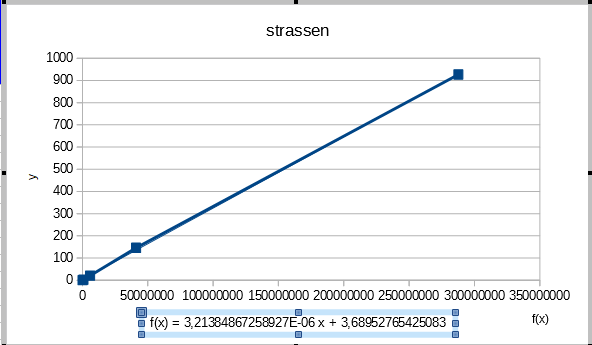
Pour chacun des algorithmes, grâce au test de rapport on clairement remarquer que nos courbes tendent vers 2b ,avec b la valeur de l'ordonnée a l'origine de la régression obtenue lors du test de puissance.

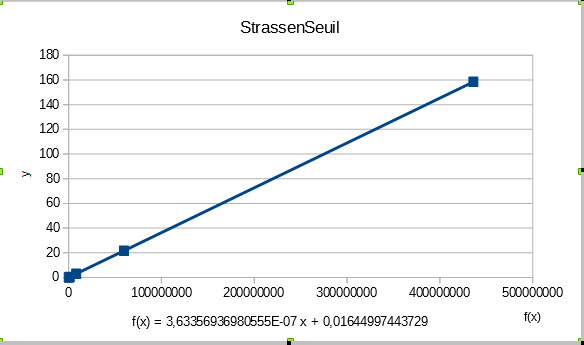
Notons bien que la fonction f (x) que nous avons utilisée est toujours égale à xm , avec m la valeur du coefficient directeur de la régression obtenue lors du test de puissance pour chaque algorithme.

Ce test nous permet ainsi donc de nous rassurer de nos hypothèses lors du test de puissance comme quoi le temps de consommation pour chacun de nos algorithmes est polynomial.

IV) Test de constantes







Pour chacun de nos algorithmes on peut remarquer que nous arrivons à faire passer une droite pour chacun des points du graphe ( f(x),y ). Ces observations nous permettent sans aucun doute de confirmer que nos consommations sont polynomiales et de confirmer les valeurs des constantes et exposant pour chacun de nos algorithmes.

V) Le seuil de récursivité est uniquement calculé dans le cas de l'algorithme de Strassen. Avec le choix d'un seuil égal à 2, on remarque que le temps d’exécution moyen pour les différentes valeurs de n est de loin meilleur par rapport à l’algorithme conventionnel, et on remarque aussi que le temps d’exécution des tailles inférieures à ce seuil de récursivité et strictement inférieure à ceux obtenues pour les mêmes grandeurs en utilisant l’algorithme conventionnel. Cependant en bas de ce seuil l’algorithme conventionnel garanti un temps meilleur.

VI) Au bout de cette analyse nous avons décidé de que pour des valeurs de n inférieures à 2 on utilisera l'algorithme conventionnel et pour les valeurs de n supérieures a 2 il est préférentiel d'utiliser celui de Strassen.