Décomposition QR : Householder
Analyse Numérique : Devoir 1

Minh-Phuong Tran

October 2018

1 Introduction

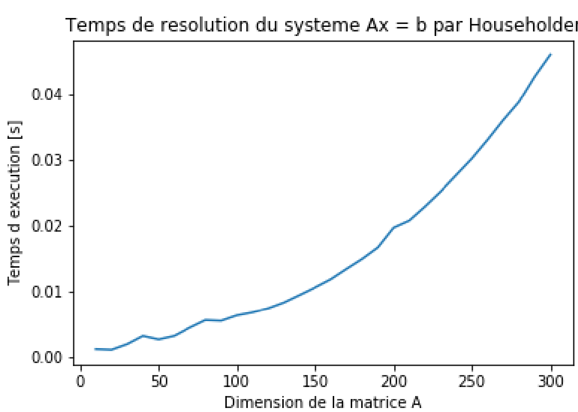
Suite à l'implémentation de la résolution de système A*b = x grâce à la décomposition A = QR, nous aborderons à présent l'analyse de la complexité temporelle du programme.

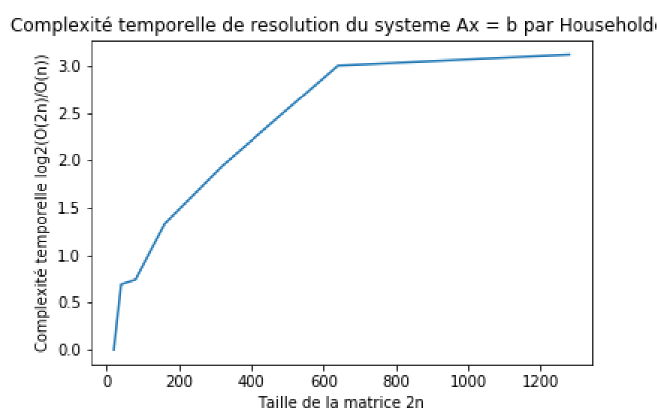
2 Complexité temporelle

Afin d'étudier la complexité temporelle de notre algorithme, nous pouvons tout d'abord diviser le programme en plusieurs étapes : la décomposition A = QR, le calcul de Q*b, le calcul de x. Pour chacune de ces 3 étapes, nous parcourons les n colonnes de A, V ou R en effectuant des opérations matricielles (produit etc.). Or, les multiplications matricielles pour une matrice de taille n x n, prennent un temps de l'ordre de O(n^2). Cela fait au total une complexité de l'ordre de O(n^3), ce qui est bien en accord avec la complexité vue au cours de O($\frac{4}{3}n^3$).

Afin de vérifier cela, nous avons implémenté des fonctions permettant de nous indiquer le temps nécessaire à la résolution d'un système par rapport à sa dimension : genMat (générateur de matrices aléatoires A et b) et complexTemp (fonction mettant en graphe le temps utilisés).

Nous obtenons alors :





La seconde figure a été obtenue en appliquant la formule $\frac{O(2n)}{O(n)} = 2^p$ nous voyons donc bien que notre p asymptotiquement tend vers 3 ce qui est en accord avec notre prédiction de O(n^3).

3 Conclusion

Nous pouvons donc confirmer que la résolution par décomposition QR (par Householder) prend un temps de l'ordre de O(n^3).