Complexité

Algèbre linéaire I : on compte les opérations

Exercice 1 : Soit $A \in \mathrm{GL}(n, \mathbf{k})$. On résoud $Ax = b$ en utilisant la méthode de Gauss.
Compter le nombre d'additions/soustractions et le nombre de multiplications/divisions faites sur chaque ligne lors de la k^e étape, pour l'ensemble de la k^e étape, pour l'ensemble de la résolution (on ne compte pas les éventuels échanges de lignes).
Montrer que les nombres d'opérations pour l'ensemble de la résolution sont des polynômes de degré 3 en n qu'on calculera.

Exercice 2 : Reprendre l'exercice précédent pour le calcul de la décomposition LU de A (on suppose qu'il n'y a pas d'échanges de lignes, cf. td méthodes directes de résolution des systèmes linéaires).
Application : Quel est le coût du calcul de l'inverse de A par la méthode de Gauss en résolvant un seul système ? par la méthode de Gauss en résolvant n systèmes ? par la décomposition LU de A ?

Algèbre linéaire II : la taille des opérandes

Exercice 3 : On suppose maintenant $A \in \mathcal{M}_{n,n}(\mathbf{Z})$. Comment adapter la méthode de Gauss pour ne pas introduire de nombres rationnels ? Pourquoi ne veut-on pas introduire de nombres rationnels ?
Si on suppose que les entiers de A ont au plus l chiffres quelle est la taille (maximale) des entiers calculés lors de la k^e étape ? (on verra en tp qu'on peut diminuer considérablement cette taille)
Les opérations sur des entiers de grande taille demandent plusieurs opérations de la machine. En utilisant les algorithmes "naïfs" on peut considérer qu'une addition/soustraction d'entiers à l chiffres demandent $c_1 l$ additions/soustractions de la machine et qu'une multiplication/division d'entiers à l chiffres demandent $c_2 l^2$ multiplications/divisions de la machine (c_1 et c_2 sont des constantes). Pourquoi (cf. Demazure) ?
Reprendre l'exercice 1 en tenant compte de n et l.

Le pgcd

Exercice 4 : L'algorithme d'Euclide peut ne prendre qu'un pas même avec des entiers grands (ex : n et $n-1$). À l'inverse quelle doit être la relation entre deux restes consécutifs pour que le nombre de pas soit le plus grand possible ? En déduire que si deux entiers $x > y > 0$ ont pour pgcd d et que l'algorithme d'Euclide prend n pas alors $x \geq dF_{n+2}$ et $y \geq dF_{n+1}$ où (F_i) est la suite de Fibonacci. Inversement, majorer le nombre de pas en fonction de y. Qu'en est-il si on tient compte de la taille des opérandes ? (cf. Demazure ou Childs et Knuth vol. 2 pour une discussion du coût en moyenne)

Un classique : les tris

Exercice 5 : On veut ranger une suite $(a_1, \ldots, a_n)$ en ordre croissant. On utilise l'algorithme suivant (tri par bulles) : on parcourt la suite en échangeant a_i et a_{i+1} s'ils ne sont pas ordonnés, puis on recommence sur la suite modifiée $(a_1, \ldots, a_{n-1})$. Pourquoi ? Combien de comparaisons fait-on ?
On retient le premier i où l'on a fait un échange et on ne recommence qu'à partir de là. Pourquoi ? Combien de comparaisons fait-on au mieux, au pire ?
Quel sens donner à l'affirmation "le tri par fusion (quicksort) est optimal" ? (cf Knuth vol. 3)

Références

Ciarlet, *Introduction à l'analyse numérique matricielle et à l'optimisation*, Dunod
Childs, *A concrete introduction to higher algebra*, Springer
Demazure, *Cours d'algèbre*, Cassini
Knuth, *The art of computer programming, vol. 2 semi-numerical algorithms*, Addison-Wesley
Knuth, *The art of computer programming, vol. 3 sorting and searching*, Addison-Wesley
Serre, *Les matrices*, Dunod