

TD/TP 4 - Première partie

Résolution d'un système linéaire par la méthode de Gauss

Opérations élémentaires sur les matrices

I Opérations élémentaires sur les lignes

On considère le système d'équations linéaires en les inconnues réelles x, y, z, t suivant :

$$\begin{cases} x + 5y + 2z + t = 1 \\ 2x + 10y + 3z - t = 4 \\ 3x + 16y + z - t = 2 \\ -x - 4y - 4z + 2t = -6 \end{cases}$$

- Q.1)** - Résoudre ce système en utilisant la méthode du pivot de Gauss. (Expliciter chacune des étapes.)
- Q.2)** - Traduire en termes matriciels le système précédent. On introduira deux matrices A et B .
- Q.3)** - Reprendre la triangularisation du système précédent sous forme matricielle. Préciser à chacune des étapes l'opération élémentaire effectuée sur les matrices A et B .

Par la suite, on représentera une matrice par un tableau à deux dimensions, c'est-à-dire par le tableau des lignes de la matrice.

- Q.4)** - Décrire un algorithme pour chacune des opérations élémentaires sur les lignes d'une matrice.
- Q.5)** - Donner la complexité de chacun de ces algorithmes en fonction des dimensions de la matrice.

II Pivot de Gauss et algorithme pour la triangularisation

- Q.1)** - Décrire un algorithme retournant la position du plus grand k -ième pivot de Gauss (plus grand en valeur absolue). Quelle est la complexité de cet algorithme ?
- Q.2)** - Adapter au procédé de triangularisation, les opérations élémentaires ci-dessus : lors de l'utilisation du k -ième pivot de Gauss, il n'est pas nécessaire de faire des calculs sur les coefficients à gauche de la k -ième colonne, ni sur la k -ième colonne.
- Q.3)** - Décrire l'algorithme complet de triangularisation. Quelle est sa complexité ?

III Système triangulaire

- Q.1)** - Donner l'algorithme donnant la solution d'un système triangulaire. Quelle est sa complexité ?
- Q.2)** - Quelle est la complexité de la méthode de Gauss pour résoudre un système ?

IV EN TP

Le but de ce TP est d'implémenter la méthode du pivot de Gauss pour résoudre un système linéaire.

- Q.1)** - Créer sous **Netbeans** un nouveau projet que l'on nommera `pivot_de_Gauss`.

- Q.2)** - Dans ce projet, créer une nouvelle classe **Matrice** qui aura pour attributs un tableau à deux dimensions de `double` et deux entiers `nbreDeLignes`, `nbreDeColonnes`.
- Q.3)** - Écrire un constructeur de la classe **Matrice** qui prend en argument un tableau à double entrée. Ce constructeur générera une exception si les lignes du tableau en argument ne sont pas toutes de la même dimension.
- Q.4)** - Écrire une méthode permettant de visualiser facilement sur la sortie standard un objet de type **Matrice**. (On pourra éventuellement redéfinir la méthode `toString()`.)
- Q.5)** - Écrire des méthodes correspondant aux opérations élémentaires sur les matrices vues en première partie.
- Q.6)** - Écrire une méthode retournant la position du plus grand k -ième pivot de Gauss. On utilisera la méthode `Math.abs()`.
- Q.7)** - Comme les calculs fait sur des `double` sont approximatifs, on fixera une constante valant 10^{-8} au dessous de laquelle les valeurs seront considérées comme nulles. En utilisant la méthode `Math.pow()` ajouter cette constante à la classe **Matrice**.
- Q.8)** - Compléter la méthode retournant la position du pivot afin qu'elle génère une exception si le pivot est nulle (c.à.d. plus petit que la constante précédente).
- Q.9)** - Comme décrit lors de la deuxième partie, adapter au procédé de triangularisation, l'une des opérations élémentaires sur les lignes.
- Q.10)** - Ajouter une méthode qui complète l'algorithme de triangularisation.
- Q.11)** - Terminer par une méthode qui retourne la solution d'un système si celui-ci est de Cramer.