Méthode du pivot

Rappels de PYTHON : Avec le module `numpy`, on peut définir et utiliser des tableaux (`array`).

Construction de tableaux (vecteurs et matrices) : `X = array([1,1,1]) ; Y = zeros(3)`

`A = array([[1,2,3],[1,1,1]]) ; B = zeros((2,3))  # attention : noter le couple` $(2,3)$

Exemple : `A = array([[1,2,3],[1,1,1]])` définit la matrice $A = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 1 & 1 \end{pmatrix}$.

`A[i][j]` ou `A[i,j]` est le coefficient situé sur ligne d'indice i et la colonne d'indice j de la matrice A.

`A[i]` ou `A[i,:]` est le vecteur ligne d'indice i de A ; `A[:,j]` est le vecteur colonne d'indice j de A.

`shape(A)` renvoie le couple (n,p), où n est le nombre de lignes de A et p le nombre de colonnes.

`len(X)` ou `shape(X)` renvoie la longueur du vecteur `X`.

Remarque : `len(A)` est le nombre n de lignes et `len(A[0])` est le nombre n de lignes

On peut opérer directement sur les vecteurs : Par exemple `A[i] = A[i] + 2*A[0]` permet d'ajouter à la i-ième ligne de A la ligne d'indice 0 multipliée par 2. Le coût d'une telle opération est linéaire en la longueur des vecteurs.

`dot(A,X)` renvoie la valeur du produit matriciel AX (produit de deux matrices ou d'une matrice et d'un vecteur).

Ici, on considère un système linéaire $AX = Y$ de n équations à n inconnues : $\begin{cases} a_{11}x_1 + ... + a_{1n}x_n &= y_1 \\ a_{21}x_1 + ... + a_{2n}x_n &= y_2 \\ \quad\quad ... & \\ a_{n1}x_1 + ... + a_{nn}x_n &= y_n \end{cases}$

On désignera par le couple (A, Y) un tel système.

On dit que deux systèmes sont équivalents ssi ils ont les mêmes solutions.

A. Résolution d'un système inversible triangulaire

1) On suppose $A \in \mathcal{M}_n(\mathbb{R})$ triangulaire supérieure.

Ecrire une fonction `resout(A,Y)` qui étant donnés une matrice $A \in \mathcal{M}_n(\mathbb{R})$ triangulaire supérieure inversible et un vecteur $Y \in \mathbb{R}^n$, renvoie l'unique vecteur X solution du système $AX = Y$.

2) Evaluer le nombre d'opérations effectuées (en fonction de n).

B. Résolution d'un système inversible par la méthode du pivot

On suppose $A \in \mathcal{M}_n(\mathbb{R})$ inversible. On se propose de résoudre $AX = Y$ par la méthode du pivot.

On dit que A admet r pivots ssi pour tout $j \in \{0, 1, ..., r-1\}$, le coefficient diagonal a_{jj} n'est pas nul, et si tous les coefficients situés sous les pivots sont nuls (c'est-à-dire $\forall j \in \{0, 1, ..., r-1\}$, $\forall i \geq j$, $a_{ij} = 0$).

$$A = \begin{pmatrix} \blacksquare & * & * & * & * \\ 0 & \blacksquare & * & * & * \\ 0 & 0 & \square & * & * \\ 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & * & * \end{pmatrix}$$

$\blacksquare$ = pivot (coefficient non nul) ; $\square$ = nouveau pivot

3) Soit A une matrice admettant r pivots, avec $0 \leq r < n-1$.

a) Ecrire une fonction `select(A,r)` qui renvoie le plus petit $s \geq r$ tel que $a_{sr} \neq 0$.

Remarque : Comme A est inversible, il existe nécessairement un tel coefficient situé dans la j-ième colonne.

b) Ecrire une procédure `echange(A,Y,r,s)` qui modifie le système (A, Y) en un système équivalent obtenu en échangeant les équations d'indices r et s. Si initialement $a_{sr} \neq 0$, on obtient ainsi une matrice A vérifiant $a_{rr} \neq 0$.

Remarque : Ne pas oublier de modifier Y aussi.

c) On suppose encore que A une matrice admet r pivots, et que de plus $a_{rr} \neq 0$. Ecrire une procédure `pivot(A,B,r)` qui modifie le système (A, Y) en un système équivalent où A admet $(r+1)$ pivots. On procède en retranchant à chaque équation d'indice $i > r$ un multiple de l'équation d'indice r multipliée par un coefficient choisi de sorte à annuler a_{ir}.

4) a) Ecrire une procédure `triangulaire(A,Y)` qui modifie le système (A, Y) en un système équivalent de sorte que A est une matrice triangulaire supérieure inversible.

b) Evaluer le coût pour une matrice A d'ordre n.

Remarque : Une opération vectorielle sur un vecteur de longueur n nécessite $O(n)$ opérations élémentaires.

5) Ecrire une fonction `solve(A,Y)` qui renvoie l'unique vecteur X vérifiant $AX = Y$.

Remarque culturelle importante : Noter au passage que l'algorithme proposé effectue $O(n^3)$ opérations.

C. Résolution d'un système tridiagonal en temps linéaire

Soit $A = \begin{pmatrix} a_1 & c_1 & 0 & \dots & 0 \\ b_1 & a_2 & c_2 & \ddots & \vdots \\ 0 & b_2 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & a_{n-1} & c_{n-1} \\ 0 & \dots & 0 & b_{n-1} & a_n \end{pmatrix}$ une matrice tridiagonale et le sytème $(S) : AX = Y$.

On suppose A donnée par les trois listes a, b, c définies par $a = [a_1, ..., a_n]$, $b = [b_1, ..., b_{n-1}]$ et $c = [c_1, ..., c_{n-1}]$.

Remarque : Il s'agit d'un exemple de matrice "creuse" qu'on code de sorte à optimiser la complexité en espace.

On suppose Y donné par un vecteur de longueur n.

On souhaite résoudre le sytème (S), en supposant ici qu'en appliquant la méthode du pivot, les coefficients diagonaux obtenus sont toujours non nuls (sans faire de permutations de lignes) et peuvent donc être choisis comme pivots.

Remarque culturelle : Notons Δ_i le déterminant de la sous-matrice obtenue en ne gardant que les i premières lignes et les i premières colonnes. La condition choisie revient en fait à supposer que pour tout $i \in \{1, 2, ., .., n\}$, $\Delta_i \neq 0$.

En appliquant la méthode du pivot, on se ramène à un système équivalent $BX = Z$, avec B est de la forme

$$B = \begin{pmatrix} d_1 & c_1 & 0 & \dots & 0 \\ 0 & d_2 & c_2 & \ddots & \vdots \\ 0 & 0 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & d_{n-1} & c_{n-1} \\ 0 & \dots & 0 & 0 & d_n \end{pmatrix}, \text{ où les } d_i \text{ sont non nuls (ce sont les pivots)}$$

6) Ecrire une fonction `pivots(a,b,c,Y)` qui étant données les listes a, b, c et le vecteur Y renvoie le couple (d, Z), où $d = [d_1, ..., d_n]$ est la liste des coefficients diagonaux de B.

7) En déduire une fonction `solve(a,b,c,Y)` qui renvoie l'unique vecteur X vérifiant $AX = Y$.

8) Préciser la complexité de l'algorithme.