TP d'initiation à MATLAB n^0 3 : Algèbre linéaire (suite)

On apprend de nouvelles fonctionalités de Matlab concernant les vecteurs et les matrices. On teste ensuite les potentialités du logiciel en travaillant avec des matrices de grande taille, en constatant notamment le gain de temps-machine obtenu en travaillant vectoriellement et matriciellement et non avec des boucles.

Algèbre linéaire

$helpwin$	Ouvre la fenêtre d'aide.
$X = -10 : 0.2 : 10$	Définition d'un vecteur.
$Y = sin(X)$	Fonction d'un vecteur.
$Z = (Y > 0.5)$	On forme un vecteur contenant des 0 et des 1; les 0 correspondent aux coordonnées ne vérifiant pas la propriété $(Y > 0.5)$, les 1 pour celles qui la vérifie.
$l = sum(Z)$	On somme les coordonnées de Z; on obtient ainsi le nombre de coordonnées de Y supérieures à 0.5.
$sum(Y. * (Y > 0))$	Somme des coordonnées positives de Y.
$YY = find((Y > 0))$	Indique le numéro des coordonnées de Y positives.
$Y(YY)$	Affiche seulement les coordonnées positives de Y.
$A = X' * Y$	On crée la matrice A.
$B = sum(A)$	On fait la somme de chaque colonne de A.
$C = sum(A(3, :))$	On somme une ligne précise de A.
$sum(sum(A))$	On somme tous les éléments de A.
$D = inv(A)$	On calcule l'inverse de A (ici, impossible..)
$max(A)$	Détermine les maxima de chaque colonne de A.
$min(max(A))$	Minimum de ces maxima. Comparer avec le maximum des minima par colonne, par ligne.
	Temps de calcul
	Créer un M-fichier appelé *temps.m*.
$\%\ Temps\ de\ calcul$	Commentaire descriptif du programme.
$l = input('ordre\ de\ A?')$	On pourra changer l'ordre de la matrice A.
$A = rand(l);$	Génération de A. A est une matrice contenant des composantes qui sont des nombres choisis au "hasard" (nous reviendrons plus tard sur ce hasard) de façon uniforme dans $[0, 1]$ et indépendamment les uns des autres. En théorie, on peut montrer qu'il y une probabilité nulle que la matrice ne soit pas inversible. Avec le logiciel, du fait qu'on ne travaille pas avec des nombres réels mais avec des décimaux (nombre fini de chiffres après la virgule) et en raison aussi des limites du générateur de nombres au "hasard", la probabilité n'est pas nulle, mais tout de même très faible.
tic	On met la "pendule" à zéro. Le temps de cette pendule est du temps *cpu*.
$B = inv(A);$	On inverse A.
toc	On arrête la pendule. Sauvegarder le programme. Lancer le programme en déterminant grossièrement le temps en fonction de l. Recommencer en affichant B. Conclusion?

Exercices

1. On veut approcher la somme de la série $S = \sum_{k=1}^{\infty} (-1)^k \frac{\ln k}{k^3}$. Faire un programme permettant de calculer la somme partielle $S(n)$ pour n quelconque, d'abord en utilisant une boucle, ensuite, en créant un vecteur de taille n dont les coordonnées sont les $(-1)^k \frac{\ln k}{k^3}$ que l'on somme. Comparer les temps d'exécution pour n de plus en plus grand. Conclusion.

2. Générer une matrice carrée d'ordre (7×7) quelconque M (fonction *rand* ...) puis calculer son carré et sa puissance 100. On appelle λ_M la valeur propre de plus grand module de M (une telle matrice M est diagonalisable avec une probabilité en théorie égale à 1). Vérifier que si $|\lambda_M| > 1$ alors M^n à tendance à "exploser" (en quel sens?) lorsque $n \to \infty$, mais que si $|\lambda_M| \leq 1$ (pour ce cas, il pourra être utile de généerer d'abord les valeurs propres, puis la matrice M), M^n tend vers une matrice limite lorsque $n \to \infty$. Expliquer ceci.