TP d'initiation à MATLAB n^0 3 : Algèbre linéaire et Analyse des données

On utilise à nouveau des vecteurs et des matrices, en constatant le gain de temps-machine obtenu en travaillant vectoriellement et non avec des boucles. On utilise ensuite les générateurs de nombres aléatoires pour effectuer des modélisations stochastiques très simples.

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..)
$E = rand(length(A))$	On crée une matrice de la même taille que A constituée de réalisations indépendantes d'une variable aléatoire distribuée suivant une loi uniforme sur $[0, 1]$.
$F = inv(A + E/100)$	On "transforme" A en une matrice inversible.
$G = A * F$	Produit de matrice. Recommencer les deux commandes précédentes en divisant E par des nombres de plus en plus grand. Constatation?
$max(G)$	Détermine les maxima de chaque colonne de G.
$min(max(G))$	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.
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?

Interlude: transférer un fichier sous disquette ou sur son compte

On veut recopier le fichier *temps.m* sur:

1. une disquette. Pour cela, on introduit la disquette, on ouvre le fichier *temps.m*, et dans *file* on clique sur *save as*. On choisit alors le directory *a* :, on change ou pas le nom du programme, et on sauvegarde.

2. le compte *cict* (ou n'importe quel compte internet). On clique sur *Demarrer*, et on choisit *Recherche de documents*. On demande à chercher les fichiers comportant le nom *FTP*. En cliquant sur le fichier *FTP* exécutable, on ouvre une fenêtre dans laquelle on met le nom de la session (par exemple *corail.cict.fr*), son nom (*agreg##* avec ## pour le numéro correspondant) et son mot de passe. Dans le registre local, choisir le fichier *temps.m* (dans $C: /MATLAB/bin/$), et le recopier dans votre compte avec la flèche. Pour vérifier que le fichier a bien été copié ou pour travailler sur votre compte, rechercher *telnet*, lancer l'executable et rentrer dans votre compte.

Exercice

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.

Un brin d'aléatoire

En possédant un générateur de nombres aléatoires, MATLAB permet de réaliser des simulations qui peuvent être nécessaire pour modéliser.

$clear$	On efface tout.
$y = rand(1)$	On génère un nombre aléatoire qui est une réalisation d'une v.a. uniforme sur $[0, 1]$.
$Y = rand(34, 1)$	On génère un vecteur colonne composé de 34 réalisations indépendantes.
$Z = randn(3, 5)$	On génère une matrice de réalisations indépendantes d'une v.a. normale centrée réduite.
$mean(Z)$	Calcule la moyenne par colonne.
$std(Z)$	Calcule l'écart-type par colonne.
$hist(Y)$	Effectue le tracé d'un histogramme des coordonnées de Y.
$hist(Y, 14)$	On obtient un histogramme avec 14 classes et non 10 obtenu par défaut.

Exercices (suite)

1. Faire un programme calculant la moyenne d'un nombre n (arbitraire) de réalisations de variables uniformes sur $[-2, 3]$, et qui trace en fonction de n cette moyenne. Que constate-t-on? Faîtes la même chose avec des réalisation indépendantes d'une v.a. distribuée suivant une loi de Cauchy (que l'on génère à partir de la tangente d'une variable uniforme sur $]-\pi/2, \pi/2[$). Conclusion.

2. Faire un programme générant 1000 moyennes m_n de n (arbitraire) réalisations d'une variable uniforme sur $[0, 2]$, puis qui trace l'histogramme des m_n. Regarder ce qui se passe pour n de plus en plus grand. Conclusion?