

TP 1 : Introduction à Scilab

Étant donné un entier $n \geq 3$, on considère la matrice $A = (a_{i,j})_{1 \leq i,j \leq n} \in \mathcal{M}_n(\mathbb{R})$ définie par

$$\forall(1 \leq i, j \leq n) \quad a_{i,j} = \begin{cases} 2 & \text{si } i = j, \\ -1 & \text{si } |i - j| = 1, \\ 0 & \text{sinon.} \end{cases}$$

1. Afficher dans la console de Scilab la matrice A obtenue pour $n = 3, n = 4, n = 5$.
2. Pour chacune de ces trois matrices, entrer dans la console les instructions $A'==A$. Quel résultat obtient-on? Comment peut-on l'interpréter?
3. À l'aide de la commande `spec`, vérifier que chacune de ces matrices est définie positive.
4. Créer une fonction $A = \mathbf{f1}(n)$, dont l'argument d'entrée est l'entier n et l'argument de sortie est la matrice A . Pour construire la matrice, cette fonction utilisera uniquement des boucles `for`.
5. Afficher le déterminant de A pour quelques valeurs de n (en utilisant la fonction précédente `f1`). Que remarque-t-on? Proposer une formule pour le déterminant de A , pour tout n .
6. Pour tout n on définit le vecteur $b \in \mathbb{R}^n$ par $b_j = 1$ pour tout $1 \leq j \leq n$, et on note $x \in \mathbb{R}^n$ la solution du système linéaire $Ax = b$. Calculer x avec Scilab, pour différentes valeurs de n (par exemple, $n = 10, n = 100, n = 200$, etc¹) en utilisant :
 - a) soit la commande `inv(A)`,
 - b) soit la commande `A\b`.
 Les vecteurs x obtenus par les deux méthodes sont-ils identiques? On pourra utiliser la commande `norm` pour le vérifier. Comment peut-on interpréter ces résultats?
7. *Inverser une matrice vs résoudre un système linéaire.* On va se servir ici de la commande `timer` pour mesurer le temps de calcul de nos programmes.
 - (i) Reprendre la question précédente et comparer l'efficacité des méthodes a) et b) (en termes de temps de calcul) pour la résolution du système $Ax = b$. On pourra choisir quelques valeurs de n assez grandes ($n = 100, 200, \dots$).
 - (ii) Plotter un graphique comparant l'évolution du temps de calcul des méthodes a) et b) en fonction de n . On pourra se servir des lignes de code suivantes, qu'il faudra compléter :

```
N=[500:50:1500]'; T1=zeros(length(N),1); T2=zeros(length(N),1);
for i=1:length(N), n=N(i); A=f1(n); ..... T2(i)=timer(); end,
plot2d(N,[T1,T2],style=[color('red'),color('blue')]);
```

8. *Le coût des boucles for.* Créer une autre fonction $A = \mathbf{f2}(n)$, qui fournira le même résultat que la fonction `f1`, mais sans utiliser de boucle `for`. On utilisera pour cela les commandes `diag` et `ones` (penser à utiliser `help` pour comprendre leur utilisation).

1. Attention, pour $n > 1000$ le temps de calcul peut devenir long.

Comparer l'efficacité des fonctions **f1** et **f2** en termes de temps de calcul pour de grandes valeurs de n ($n = 500, 1000, \dots$). On pourra essayer de comparer les deux méthodes à l'aide d'un graphe, à l'instar de la question précédente.

9. On note $\lambda_{min}, \lambda_{max}$ la plus petite et la plus grande valeur propre de A . En utilisant la fonction **f2**, estimer les limites quand n tend vers l'infini de λ_{min} et λ_{max} .

Démarrage rapide avec Scilab

Une matrice $A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$ se définit avec **A = [1 2 3;4 5 6]**; Noter que cette commande n'affiche pas de résultat, car elle se termine par un point-virgule ";". Si on veut voir le résultat d'une fonction, ou d'une expression, il faut conclure la commande avec une virgule ",".

La définition d'un vecteur se fait de manière similaire, car Scilab considère que les vecteurs de taille n sont des matrices de taille $n \times 1$.

Quelques commandes de base pour la manipulation de matrices :

- Transposée d'une matrice : **A'** (Utile pour passer d'un vecteur ligne à colonne).
- Produit de deux matrices (ou matrice fois vecteur) : **A*B**
- Puissance d'une matrice : **A^3**
- Inverse d'une matrice : **A^-1** ou **inv(A)**
- Solution de l'équation $Ax = b$: **A\b** (Attention : le vecteur doit être en colonne!)
- Déterminant d'une matrice : **det(A)**
- Spectre d'une matrice : **S=spec(A)** retourne un vecteur contenant toutes les valeurs propres de A rangées par ordre croissant.
- Norme euclidienne d'un vecteur : **norm(x)**

Quelques commandes pour générer des matrices :

- Matrice identité de taille $n \times n$: **eye(n,n)**
- Matrice aléatoire de taille $n \times m$: **rand(n,m)**
- Vecteur défini par progression arithmétique : **I=[3:0.1:4]** renverra le vecteur ligne [3 3.1 3.2 ... 3.9 4]

Syntaxe de la boucle **for** :

```
for i=1:100,
    INSTRUCTIONS;
end
```

Écrire et utiliser une fonction dans un fichier :

1. Utiliser le navigateur de fichiers de Scilab pour se placer dans un dossier dans lequel on va travailler.
2. Depuis la console Scilab, cliquer sur l'icône "Démarrer Scinotes".
3. Écrire sa fonction dans Scinotes. La syntaxe est la suivante :

```
function [variables-de-sortie] = nom-de-la-fonction(variables-entrantes)
    INSTRUCTIONS;
end
```

Si besoin est, on commentera une ligne en la faisant commencer par `\%`
4. Enregistrer le fichier dans le dossier, en lui donnant le nom de la fonction.
5. Cliquer sur l'icône "Exécuter".
6. On peut alors faire appel à la fonction depuis la console ou d'autres fonctions.

Astuce : si Scilab mouline sur un calcul trop difficile, on peut tuer l'exécution depuis la console via **Contrôle > Abandonner**.