feuille de travaux pratiques

Séances 1 et 2 : prise en main de MATLAB et GNU OCTAVE

Le symbole $\diamond$ indique un exercice optionnel. Le travail demandé peut être effectué indifféremment avec MATLAB ou le logiciel libre GNU OCTAVE (<http://www.gnu.org/software/octave/>).

Exercice 1 (manipulation et opérations sur les tableaux).

1. On considère les vecteurs et matrices suivants

$$\mathbf{l} = (1 \ 2 \ 3 \ 4 \ 5), \quad \mathbf{v} = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix}, \quad A = \begin{pmatrix} 1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 3 & 1 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 4 & 5 & 6 \\ 3 & 4 & 5 & 6 & 7 \\ 4 & 5 & 6 & 7 & 8 \\ 5 & 6 & 7 & 8 & 9 \end{pmatrix}.$$

- a. Créer des tableaux correspondants à $\mathbf{l}$, $\mathbf{v}$, A et B .
 - b. Extraire la première ligne, la deuxième colonne et la sous-matrice $(a_{ij})_{1 \leq i \leq 3, 1 \leq j \leq 5}$ de la matrice A , les termes diagonaux de la matrice B .
 - c. Commenter le résultat produit par les instructions suivantes : $\mathbf{l} * \mathbf{v}$, $\mathbf{v} * \mathbf{l}$, $\mathbf{l} ./ \mathbf{v}$, $A * B$, B / A , $A .* B$, $A * A'$, $\sin(\mathbf{l})$ et $\exp(A)$.
 - d. Comment déterminer le format (nombres de lignes et de colonnes) de ces tableaux ?
2. MATLAB et GNU OCTAVE permettent également la génération « automatisée » de tableaux particuliers. À titre d'illustration, analyser¹ et commenter le résultat produit par chacune des instructions suivantes :

```
r=[1.3:15.8]
s=[1.3:0.4:15.8]
t=linspace(1.3,15.8,5)
u=ones(size(t))
v=3*ones(1,5)
w=sin([0:pi/6:pi])
x=eye(3,3)
y=rand(1,5)
z=zeros(5,1)
```

3. On considère les vecteurs de $\mathbb{R}^3$ suivants

$$\mathbf{u} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}, \quad \mathbf{v} = \begin{pmatrix} -5 \\ 2 \\ 1 \end{pmatrix} \quad \text{et} \quad \mathbf{w} = \begin{pmatrix} -1 \\ -3 \\ 7 \end{pmatrix}.$$

- a. Créer des tableaux correspondants à ces vecteurs.
- b. Calculer $\mathbf{u} + \mathbf{v}$, $\mathbf{u} + 3\mathbf{v} - 5\mathbf{w}$, $\frac{1}{5}\mathbf{w}$.
- c. En utilisant les commandes appropriées², calculer $\|\mathbf{u}\|_2$, $\|\mathbf{v}\|_1$, $\|\mathbf{w}\|_\infty$ et le cosinus de l'angle formé par les vecteurs $\mathbf{u}$ et $\mathbf{v}$.

¹. Ne pas hésiter à faire appel à l'aide en ligne via l'instruction `help` suivie du nom de la commande concernée pour comprendre son fonctionnement.

². On pourra notamment consulter les aides en ligne des fonctions `norm` et `dot`.

4. On considère les matrices suivantes :

$$A = \begin{pmatrix} 2 & 3 \\ 6 & 5 \end{pmatrix} \text{ et } B = \begin{pmatrix} 2 & 3 & 4 \\ 7 & 6 & 5 \\ 2 & 8 & 7 \end{pmatrix}.$$

- Créer des tableaux correspondants à A et B .
 - En utilisant les commandes appropriées, calculer leurs déterminants, inverses et valeurs propres et vecteurs propres associés³.
5. Créer des tableaux correspondants à la matrice identité d'ordre 6 et à la matrice nulle d'ordre 3.
6. Pour $n \in \mathbb{N}$, $n \geq 2$, on considère la matrice tridiagonale d'ordre n définie par

$$A_n = \begin{pmatrix} 2 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & -1 & 2 & -1 & \\ & & & -1 & 2 \end{pmatrix}.$$

- Que fait la suite d'instructions suivante ?
`S=[eye(n) zeros(n,1)];`
`S=S(:,2:n+1);`
`A=2*eye(n)-S-S'`
- Répondre à la même question avec la suite d'instructions ci-dessous.
`D=diag(ones(n,1));`
`SD=diag(ones(n-1,1),1);`
`A=2*D-SD-SD'`

7. On considère la matrice et les vecteurs

$$A = \frac{1}{8} \begin{pmatrix} 5 & -2 & 1 \\ 2 & 0 & 2 \\ 1 & -2 & 5 \end{pmatrix}, \mathbf{b} = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix} \text{ et } \mathbf{u}^{(0)} = \begin{pmatrix} 5 \\ 2 \\ -4 \end{pmatrix},$$

et la suite de vecteurs $(\mathbf{u}^{(k)})_{k \in \mathbb{N}}$ définie par

$$\mathbf{u}^{(k+1)} = A \mathbf{u}^{(k)} + \mathbf{b}, \forall k \in \mathbb{N}.$$

- Créer des tableaux correspondants à A , $\mathbf{b}$ et $\mathbf{u}^{(0)}$.
- Calculer les premiers termes de la suite $(\mathbf{u}^{(k)})_{k \in \mathbb{N}}$. Qu'observe-t-on ?
- Reprendre les questions précédentes avec

$$A = \begin{pmatrix} 5 & 6 & 3 \\ -1 & 5 & -1 \\ 1 & 2 & 0 \end{pmatrix}, \mathbf{b} = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix} \text{ et } \mathbf{u}^{(0)} = \begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix}.$$

- Interpréter ces résultats.

Exercice 2 (écriture de fonctions). On rappelle que l'on peut définir une nouvelle fonction pour MATLAB ou GNU OCTAVE en écrivant, au moyen d'un simple éditeur de texte, son code source dans un fichier ayant pour extension⁴ `.m`. Un tel fichier de fonction débute par le mot-clé `function`, suivi du nom de la fonction et des paramètres d'entrée et de sortie de cette dernière. Cette déclaration est de la forme :

3. On consultera l'aide en ligne de la commande `eig`, les termes « valeur propre » et « vecteur propre » se traduisant respectivement par « *eigenvalue* » et « *eigenvector* » en anglais.

4. Ce type de fichier permet aussi d'enregistrer une séquence d'instructions à faire exécuter par le logiciel; on parle dans ce cas de fichier de script.

```

function [y_1,y_2,...] = mafonction(x_1,x_2,...)
% Les commentaires situés juste après la déclaration de la fonction
% constituent l'aide obtenue en entrant la commande
% help mafonction

```

Le nom de la fonction et le préfixe du nom du fichier la contenant doivent impérativement être identiques⁵. Ainsi, dans l'exemple ci-dessus, le fichier de fonction doit être nommé `mafonction.m`. La fonction elle-même est appelée depuis la ligne de commande, un fichier de script ou une autre fonction de la manière suivante :

```
[y_1,y_2,...]=mafonction(x_1,x_2,...)
```

On notera qu'il est possible d'afficher le code source d'un fichier de fonction préexistant grâce à la commande `type`.

1. Écrire une fonction nommée `polaire`, prenant comme arguments d'entrée les coordonnées cartésiennes (x, y) d'un point de $\mathbb{R}^2$ et renvoyant en sortie les coordonnées polaires (r, θ) de ce point⁶. Penser à commenter le code source de manière à ce qu'un utilisateur puisse utiliser l'aide en ligne pour s'informer sur cette nouvelle fonction.
2. En utilisant une structure de contrôle de type `if ... else ...`, écrire une fonction nommée `profil` qui associe à toute matrice A à m lignes et n colonnes son profil, c'est-à-dire la suite d'entiers $\{\phi(i)\}_{i=1,\dots,m}$ avec ϕ une application de $\{1, \dots, m\}$ dans $\mathbb{N}$ telle que

$$\phi(i) = \begin{cases} \inf \{j \in \{1, \dots, n\} \mid a_{ij} \neq 0\} & \text{si la } i^{\text{ème}} \text{ ligne de } A \text{ est non nulle,} \\ n + i & \text{sinon.} \end{cases}$$

Exercice 3 (représentation graphique avec la commande `plot`). On cherche à obtenir une représentation graphique de la fonction $f(x) = \exp(-x) \sin(4x)$ sur l'intervalle $[0, 2\pi]$.

1. Que contiennent les tableaux créés via les commandes suivantes ?

```

x=linspace(0,2*pi,101);
y=exp(-x).*sin(4*x);

```

2. Tracer la représentation graphique de la fonction f associées aux tableaux `x` et `y`. En utilisant le zoom, déterminer une valeur approchée du maximum de f sur $[0, 2\pi]$. Comment affiner le tracé pour préciser ce maximum ?
3. En utilisant l'instruction `hold on`, tracer sur une même figure une représentation graphique des fonctions $x \mapsto x^2$ et $x \mapsto x^2 \sin(x) \exp(-x)$ sur l'intervalle $[-1, 1]$, en utilisant des couleurs différentes pour chacune d'entre elles.

Exercice 4 $\diamond$ (manipulation des nombres complexes). Soit u et v les nombres complexes

$$u = 11 - 7i \text{ et } v = -1 + \sqrt{3}i.$$

5. Pour éviter d'éventuels conflits dus à l'emploi de noms déjà affectés à des commandes de `MATLAB` ou de `GNU OCTAVE`, on prendra soin de vérifier qu'une fonction de même nom n'existe pas déjà (grâce à la commande `help` par exemple) et/ou de mettre des majuscules dans le nom (les logiciels étant sensibles à la casse).

6. On rappelle que, en vertu du théorème de Pythagore, on a $r = \sqrt{x^2 + y^2}$ et que, pour obtenir l'angle θ dans l'intervalle $[0, 2\pi[$, on utilise les formules suivantes :

$$\theta = \begin{cases} \arctan\left(\frac{y}{x}\right) & \text{si } x > 0 \text{ et } y \geq 0, \\ \arctan\left(\frac{y}{x}\right) + 2\pi & \text{si } x > 0 \text{ et } y < 0, \\ \arctan\left(\frac{y}{x}\right) + \pi & \text{si } x < 0, \\ \frac{\pi}{2} & \text{si } x = 0 \text{ et } y > 0, \\ \frac{3\pi}{2} & \text{si } x = 0 \text{ et } y < 0. \end{cases}$$

Créer des variables correspondant à nombres et calculer les modules et les arguments de u et de v , les produits $u\bar{v}$ et $\bar{u}v$, ainsi que les parties réelle et imaginaire de u^3+v^2 en utilisant les commandes appropriées.

Exercice 5 $\diamond$ (**formats d'affichage**). MATLAB et GNU OCTAVE utilisent par défaut le format de représentation double précision de la norme IEEE 754 pour les calculs arithmétiques et toute variable est *a priori* stockée dans ce format. On peut afficher des (tableaux de) valeurs numériques ou chaînes de caractères en tapant simplement le nom des variables correspondantes ou en utilisant la commande `disp`, cette dernière réalisant l'affichage sans écrire le nom de la variable concernée.

Il existe différents formats d'affichage prédéfinis, sélectionnables grâce à la commande `format`.

1. Tester ces formats⁷ en exécutant la suite d'instructions suivantes :

```
x=pi^5;
disp(x) % affichage avec la notation par défaut
format long % notation à séparateur fixe et 15 chiffres significatifs
disp(x)
format short e % notation "scientifique" et 5 chiffres significatifs
disp(x)
format long e % notation "scientifique" et 15 chiffres significatifs
disp(x)
format short % retour à la notation par défaut
```

Il est aussi possible de contrôler très précisément la mise en forme d'un affichage avec la commande `fprintf`, qui permet de spécifier explicitement le format via une chaîne de caractères. Dans cette dernière, le format voulu pour chaque variable débute par le signe %, suivi de la longueur minimal du champ (c'est-à-dire le nombre minimal de caractères utilisés pour l'affichage) et d'un caractère de conversion (une lettre) indiquant que la valeur à afficher est un unique caractère (%c), une chaîne de caractères (%s), un entier signé en base 10 (%d ou %i) ou encore un réel en notation à séparateur fixe (%f), en notation « scientifique » (%e) ou « compacte » (%g). Dans les derniers cas, on peut préciser le nombre maximal de caractères utilisés (pour %s), le nombre de chiffres situés à droite du séparateur (pour %f ou %e) ou bien le nombre de chiffres significatifs (pour %g), par un point suivi d'un chiffre. Indiquons que plusieurs autres opérateurs optionnels existent (', -, + ou le caractère « espace »), ainsi que des caractères d'échappement, tels que \n pour un saut de ligne, \t pour une tabulation horizontale, etc... Le tableau de variables à afficher donné en argument de la commande doit contenir au moins autant d'éléments qu'il y en a de prévu dans la chaîne de caractère donnant le format.

2. Tester les possibilités offertes par `fprintf` en exécutant les commandes suivantes :

```
fprintf('%c\n','abcdefghijklmnopqrstuvwxy')
fprintf('%s\n','abcdefghijklmnopqrstuvwxy')
fprintf('%13s\n','abcdefghijklmnopqrstuvwxy')
fprintf('%i %i %i\n',[-10:10])
fprintf('%+.2i %+.2i %+.2i\n',[-10:10])
fprintf('%0f\t%.1f\t%.10f\n',pi,pi,pi)
fprintf('%0e\t%.1e\t%.10e\n',pi,pi,pi)
fprintf('%0g\t%.1g\t%.10g\n',pi,pi,pi)
```

puis en les modifiant.

La commande `format rat` propose enfin un un format affichage pour lequel les valeurs numériques sont (éventuellement) approchées par une fraction rationnelle.

3. Afficher la matrice de Hilbert (obtenue avec la commande `hilb`) d'ordre 10. Quelle est la fraction rationnelle obtenue pour l'approximation de π ? De la constante de Napier e ? De $\sqrt{2}$? De $\ln(2)$? De $\ln(\frac{1}{2})$?

7. Il faut y ajouter deux formats, un court (`format short g`) et un long (`format long g`), utilisant une notation « compacte », pour laquelle le logiciel choisit entre les notations à séparateur fixe et scientifique en fonction de la valeur à afficher.