

ANALYSE NUMÉRIQUE

Travaux Pratiques 2024 – 2025

Séance 1

1. Entrez les instructions suivantes dans la fenêtre de commandes Octave, tâchant de comprendre leur fonctionnement. Aidez-vous de

```
help commande
```

pour comprendre ce que fait une *commande* particulière (exemple : `help clc`).

```
%%% 1 -- quelques opérations de base
clear; clc % notez que ; permet deux opérations sur une ligne
a = 1+pi
format long
b = exp(2); % notez que ; évite l'affichage de b
b
format short; c = log(b)+2*i % notez que le résultat est un complexe
whos
clear; clc;
whos
%%% 2 -- vecteurs et matrices
v = [1;2;3;4] % matrice 4x1 (vecteur)
w = [1 2 3 4] % matrice 1x4
w = 1:4 % une manière plus concise de définir w
w' % transposée de w (à ne pas copier-coller)
M = [1 2; 3 4; 5 6] % matrice 3x2 ; notez le rôle des ; et d'espaces
length(v)
size(M)
v(2) % élément 2 de v
v(1:3) % sous-vecteur des éléments de 1 à 3 de v
M(2,2) % élément 2,2 de M
M(1:2,:) % sous-matrice contenant lignes 1 et 2 et toutes les colonnes
w = [v(1:3); v(4)] % concaténation verticale (car ; ) de v(1:3) et v(4)
w = [v(1:3) v(4)] % concaténation horizontale: pourquoi une erreur?
v'*v % produit matriciel (à ne pas copier-coller)
v*v % produit matriciel: pourquoi une erreur?
v.*v % . pour une opération élément par élément
%%% 3 -- graphes 2D
x = 0:0.5:4*pi; % vecteur d'éléments 0, 0.5, 1, ..., 12.5 (≈ 4*pi)
y = sin(x); % vecteur d'éléments sin(0), sin(0.5), sin(1), ..., sin(12.5)
plot(x,y,'*') % affiche les points (x(i),y(i))
hold on % garde l'affichage précédent
plot(x,y,'-') % graphe reliant les points (x(i),y(i)) entre eux
% traçons le graphe de (sinx)^2
y2 = y*y; % doit-on utiliser la multiplication vectorielle?
y2 = y.*y; % ou élément par élément? justifiez!
hold off % efface l'affichage précédent
plot(x,y2,'-r')
exit % sort d'Octave; pour forcer la sortie utilisez plutôt Ctrl+c
```

2. Effectuez les opérations matricielles suivantes.

a) Créez une matrice 5×5 aléatoire $A = (a_{ij})$;
(aidez-vous de la commande **rand**).

b) Créez une matrice identité L de mêmes dimensions ;
(aidez-vous de la commande **eye**).

c) Remplacez la première colonne de L par la première colonne de A divisée par a_{11} ; inversez le signe des éléments hors diagonale de L ;
(utilisez $A(a:b, c)=B(a:b, c)$ pour remplacer $A(a, c)$, $A(a+1, c)$, ..., $A(b, c)$ par les éléments correspondants de B).

d) Calculez le produit matriciel LA . Que pouvez-vous dire de sa première colonne ?

e) Calculez L^{-1} (aidez-vous de la commande **inv**). Comparez L^{-1} et L .

3. Ecrivez une fonction Octave qui retourne le produit d'éléments diagonaux d'une matrice. Utilisez cette fonction pour vérifier que le déterminant (commande **det**) d'une matrice triangulaire est donné par le produit de ses éléments diagonaux.

(Sauvegardez la fonction dans un fichier du même nom avec l'extension **.m**; pour pouvoir l'utiliser, déplacez-vous dans le répertoire où elle se trouve à l'aide des instructions **ls**, **cd** répertoire ou **cd ..**).

4. Ecrivez une fonction Octave qui calcule

a) $f : \mathbb{R} \mapsto \mathbb{R} : x \mapsto \frac{e^{x/\pi}}{\log(x + \pi)}$

b) $f : \mathbb{R} \mapsto \mathbb{R} : x \mapsto \begin{cases} 1 & \text{si } x = 0 \\ \frac{\sin(x)}{x} & \text{si non} \end{cases}$

(aidez-vous des commandes **pi**, **exp**, **log10** et **sin**). Affichez les graphes de ces fonctions sur l'intervalle $[-1, 1]$ à l'aide de la commande **plot**.

Pour ce qui est des logarithmes, Octave dispose des fonctions pour les logarithmes naturels (**log**), en base 2 (**log2**) et 10 (**log10**); est-ce limitatif ?