

Corrigé des exercices de familiarisation avec Matlab

Exercice 1 :

Soit la série de nombres {17 8 12 15 6 11 9 18 16 10 13 19}.

```
% pour commencer dans un environnement propre
clear all % supprime toutes les variables de la mémoire
close all % ferme toutes les fenêtres graphiques
clc % nettoie la fenêtre de commande
```

```
% pour un affichage plus lisible par la suite
% ne change rien au stockage des variables
format short g
```

1. Entrer ces valeurs dans le vecteur x ;

```
x=[17 8 12 15 6 11 9 18 16 10 13 19]
```

2. Calculer la longueur N de ce vecteur;

```
N=length(x)
```

3. Calculer la somme S des éléments;

```
S=sum(x)
```

4. Calculer la moyenne $\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$;

```
xbarre=S/N % version pedestre
```

```
xbarre=mean(x) % version rapide
```

5. Calculer l'écart-type $\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}$;

```
sigma=sqrt(sum((x-xbarre).^2)/(N-1)) % version pedestre
```

```
sigma=std(x) % version rapide
```

6. Calculer le vecteur $dx = \{x_{i+1} - x_i\}$ pour $i = \{1, 2, \dots, N - 1\}$.

```
dx=x(2:end)-x(1:end-1) % version pedestre
```

```
dx=diff(x) % version rapide
```

Exercice 2 :

```
clear all
close all
clc
format short g
```

1. Définir un vecteur t qui contient 51 valeurs équidistantes entre -25 et 25 ;

```
t=linspace(-25,25,51);
```

2. Calculer le vecteur $x = t^2$;

```
x=t.^2;
```

3. Calculer le vecteur y qui contient t^3 mais dans l'ordre inverse ;

```
y=t(end:-1:1).^3; % possibilité la plus simple
```

```
y=fliplr(t).^3; % autre possibilité
```

4. Représenter x et y en fonction de t ;

```
figure
subplot(2,1,1)
plot(t,x)
xlabel('t'); ylabel('x');
subplot(2,1,2)
plot(t,y)
xlabel('t'); ylabel('y');
```

5. Calculer la somme des éléments pairs de x ;

```
% l'une des trois lignes au choix
```

```
sum(x(fix(x/2)==x/2))
```

```
sum(x(rem(x,2)==0))
```

```
sum(x(mod(x,2)==0))
```

6. Calculer la somme des éléments positifs de y .

```
sum(y(y>0))
```

Exercice 3 :

```
clear all
close all
clc
format short g
```

1. Générer le vecteur colonne t qui va de 1 à 10 par pas de 0,5 ;

```
t=(1:0.5:10)';
```

2. Créer une matrice A dont les colonnes sont t, t^2, t^3 et t^4 ;

```
A=[t t.^2 t.^3 t.^4];
```

3. Ajouter une colonne qui vaut 1 lorsque $t > 5$, 0 le reste du temps ;

```
A(:,5)=t>5;
```

4. Ajouter une colonne qui vaut 5 lorsque t est entier, 0 le reste du temps.

```
% l'une des trois lignes au choix
```

```
A(:,6)=(fix(t)==t)*5;
```

```
A(:,6)=(rem(t,1)==0)*5;
```

```
A(:,6)=(mod(t,1)==0)*5;
```

Exercice 4 :

1. Créer une fonction `t=matrace(A)` :
 - qui calcule la trace d'une matrice A si A est carré ;
 - qui sort la valeur -1 si A n'est pas carré ;

```
function t=matrace(A)
% fonction qui calcule la trace de A

if size(A,1)==size(A,2)
    t=trace(A); % méthode rapide
    t=sum(diag(A)); % méthode plus pédestre
else
    t=-1;
end
```

2. Générer une matrice aléatoire de dimension 10×10 ;

```
% matrice aléatoire de dimension 10*2 à valeurs entre 0 et 1
B=randn(10,2);
% matrice aléatoire de dimension 10*10 à valeurs entre 0 et 1
C=rand(10,10);
% matrice aléatoire de dimension 10*10 à valeurs entre 0 et 10
D=round(rand(10,10)*10);
```

3. Calculer la trace de cette matrice, de sa transposée et de son inverse.

```
t0=matrace(B)
t1=matrace(D)
t2=matrace(D')
t3=matrace(inv(D))
```