

## TP 1 (initiation programmation Matlab)

### Exercice 1

Dans cet exercice, on a choisi quelque problème mathématique sur les vecteurs et les matrices résolu directement en langage Matlab :

<code>clc ; clear all</code>	efface la feuille command Windows et la mémoire des variables
<code>v1=[4 1 3]</code>	définir un vecteur ligne ( <b>v1</b> )
<code>v2=[9 ; 2 ; 5]</code>	définir un vecteur colonne ( <b>v2</b> )
<code>vt=v2'</code>	vecteur transpose ( <b>v2</b> )
<code>a=[4 8 9 ; 3 5 6 ; 1 2 3]</code>	définir une matrice ( <b>a</b> )
<code>at=a'</code>	matrice transpose de ( <b>a</b> )
<code>d1=diag(a)</code>	diagonale de la matrice ( <b>a</b> )
<code>a1=inv(a)</code>	inverse de matrice ( <b>a</b> )
<code>a2=1 ./ a</code>	inverse de chaque élément de la matrice ( <b>a</b> )
<code>x=inv(a)*v2</code>	Résoudre système linéaire $[a].\{x\}=\{v2\} \rightarrow \begin{bmatrix} 4 & 8 & 9 \\ 3 & 5 & 6 \\ 1 & 2 & 3 \end{bmatrix} \cdot \begin{Bmatrix} x1 \\ x2 \\ x3 \end{Bmatrix} = \begin{Bmatrix} 9 \\ 2 \\ 5 \end{Bmatrix}$
<code>d=det(a)</code>	Déterminant de la matrice ( <b>a</b> )
<code>[vec_p,val_p]=eig(a)</code>	Calcule les vecteurs propres et les valeurs propres d'une matrice ( <b>a</b> )
<code>s1=sum(v2)</code>	la somme des éléments d'un vecteur ( <b>v2</b> )
<code>p1=prod(v2)</code>	produit des éléments d'un vecteur ( <b>v2</b> )
<code>[m1,i1]=max(v2)</code>	m1 est le max d'un vecteur, i1 est sa position
<code>[m2,i2]=min(v2)</code>	m2 est le min d'un vecteur , i2 est sa position
<code>m3=mean(v2)</code>	la moyenne d'un vecteur
<code>v3=a*v2</code>	produit matrice ( <b>a</b> ) * vecteur colonne ( <b>v2</b> )=vecteur <b>v3</b>
<code>v4=v1*a</code>	produit vecteur ligne ( <b>v1</b> )* matrice( <b>a</b> )= vecteur v4
<code>a3=v2* v1</code>	produit vecteur colonne ( <b>v2</b> )* vecteur ligne ( <b>v1</b> ) =matrice
<code>scal=v1* v2</code>	produit vecteur ligne ( <b>v1</b> ) * vecteur colonne ( <b>v2</b> ) = scalaire
<code>v5=v1.* v2'</code>	produit vecteur ligne ( <b>v1</b> ) .* vecteur ligne ( <b>transpose v2</b> ) = vecteur (produit deux vecteur même type élément par élément sans faire la sommation)

## NOTION DE BASE

- **INSTRUCTION DE CONTROLE (TEST)**

L'instruction **if** exécute un ensemble de commande si une condition est satisfaite.

La syntaxe générale est

<code>if(expression ou condition logique ) instruction end</code>	<code>If (expression ou condition logique ) instruction 1 else instruction 2 end</code>
---	---

**Exemple :** Calculer la valeur de y

$$y = \begin{cases} \exp(x) + \sin(x) & ; \quad x < 1 \\ x^2 - \ln(x) & ; \quad x \geq 1 \end{cases}$$

```
x = - 3
if (x < 1)
    y = exp(x) + sin(x)
else
    y = x^2 - log(x)
end
```

**Remarque :** expression ou condition logique c'est des instructions qui contiennent **Les opérateurs de comparaison :**

>	Plus grand que	<=	Plus petit ou égal à
<	Plus petit que	==	Egal à
>=	Plus grand ou égal à	~=	Différent de

• **INSTRUCTION DE REPETITION (BOUCLES)**

Une boucle permet de répéter la même commande un grand nombre de fois en faisant varier un paramètre. C'est par exemple très utile pour la définition des suites par récurrences comme nous le verrons. Différentes syntaxes existent. La boucle s'écrit de la manière suivante :

<b>while</b> ( <i>expression ou condition logique</i> ) <i>Instruction</i> <b>end</b>	<b>for</b> <i>indic</i> = <i>valeur initiale</i> : <i>le pas</i> : <i>valeur final</i> <i>instruction</i> <b>end</b>
---	--

**Exemple :** en utilisant la boucle while ou la boucle for , écrire un programme en Matlab qui:

- Calculer la somme :  $\sum_{k=1}^N K = 1+2+3+...+N$
- Calculer le factorielle de N  $N! = 1*2*3*...*N$

<b>clc ; clear all</b> n=5 ; s=0 ; fact=1 ; k=1 ; <b>while</b> (k<=n ) s=s+k ; fact=fact*k ; k=k+1 ; <b>end</b> <b>s</b> <b>fact</b>	<b>clc ; clear all</b> n=5 ; s=0 ; fact=1 ; <b>for</b> k=1 :1 :n s=s+k ; fact=fact*k ; <b>end</b> <b>s</b> <b>fact</b>
--	---

**Exercice 2 :** écrire un programme Matlab qui déterminer :

- le max et sa position de chaque ligne
- le min et sa position de chaque colonne

```

clc ; clear all
n1=3 ; nc=4 ;
a=[4 5 7 2 ; 3 9 8 -3 ; -2 0.5 11 2.7 ]
disp('==== le max et sa position pour chaque ligne ===')
for i=1:n1
    v(1:nc)=a(i,1:nc) ;
    [max_l(i),loc_l(i)]=max(v);
end
max_l
loc_l
disp('==== le min et sa position pour chaque colonne ===')
for j=1:nc
    u(1:n1)=a(1:n1,j) ;
    [min_c(j),loc_c(j)]=min(u);
end
min_c
loc_c

```

### Exercice 3

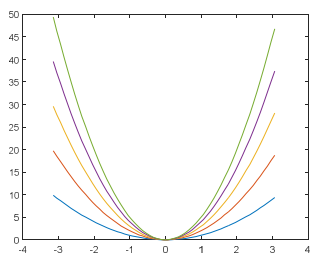
Dans cette exercice, on a choisi quelque problème mathématique résolu directement en langage Matlab sans faire discrétisation numérique comme :

- équation non linéaire analytiquement :  $f(x) = x^2 + 5x + 6$
- calcule la dérivée :  $\text{diff}(f, x) = \frac{df}{dx} = \frac{d}{dx} \left( \frac{1}{x^2 + 2x + 2} \right)$
- calcule  $\int f(x).dx = \int \frac{1}{x^2 + 2x + 2}.dx$
- résoudre équation différentielle du 1<sup>ère</sup> ordre :  $\frac{dy}{dx} = F(x, y) = (x^2 - 1) * y$
- résoudre équation différentielle du 2<sup>ème</sup> ordre  $\frac{d^2y}{dx^2} - 5 \cdot \frac{dy}{dx} + 6 \cdot y = x^2$

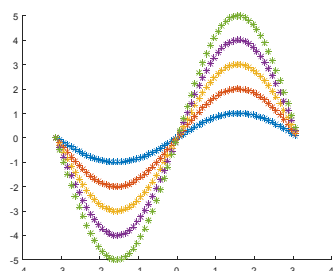
```
clc ;clear all
disp('== resoudre equation non lineaire analytiquement ==')
syms x
xc=eval(solve(x.^2+5*x+6))
disp('==== la deriver d un fonction ====')
syms x ;
derive=diff('1/(x^2+2*x+2)',x)
disp('==== integral ====')
syms x ;
intf=int('1/(x^2+2*x+2)',x)

disp('== solution analytiquement eq diff 1 ere ordre ====')
dsolve('D1y=(x^2-1)*y','x')
disp('=== solution analytiquement eq diff 2 eme ordre ====')
dsolve('D2y-5*D1y+6*y=x^2','x')
```

Exercice 4 : Ecrire un programme qui trace plusieurs courbes un seul variable



En utilisant la commande  
Hold on



```
clc
clear all
x=[-pi:0.1:pi] ;
%=====
y1= x.^2 ; y2= 2*x.^2 ; y3= 3*x.^2 ;
y4= 4*x.^2 ; y5= 5*x.^2 ;
figure ; plot(x,y1,x,y2,x,y3,x,y4,x,y5) ;

%=====
figure; hold on;
for i=1:5
y=i*sin(x) ;
fig4=plot(x,y);
end
```

## Exercice 5

Ecrire un programme qui trace les courbes à deux variables  $z=f(x,y)$

```
clc ; clear all ; close all ;
[x,y]=meshgrid(-10:0.1:10,-10:0.1:10);
%=====
z = x.^2 + y.^2 ;
figure ; surf(x,y,z,'lineStyle','none');
%=====
z = x.^2 - y.^2 ;
figure ; surf(x,y,z,'lineStyle','none');
%=====
r = sqrt(x.^2 + y.^2) ;
z = sin(r)./r;
figure ; surf(x,y,z,'lineStyle','none');
```

