

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 :

clc ; clear all	efface la feuille command Windows et la mémoire des variables
v1=[4 1 3]	définir un vecteur ligne (v1)
v2=[9 ; 2 ; 5]	définir un vecteur colonne (v2)
vt=v2'	vecteur transpose (v2)
a=[4 8 9 ; 3 5 6 ; 1 2 3]	définir une matrice (a)
at=a'	matrice transpose de (a)
d1=diag(a)	diagonale de la matrice (a)
a1=inv(a)	inverse de matrice (a)
a2=1 ./ a	inverse de chaque élément de la matrice (a)
x=inv(a)*v2	Résoudre système linéaire $[a].\{x\}=\{v2\} \rightarrow \begin{bmatrix} 4 & 8 & 9 \\ 3 & 5 & 6 \\ 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 9 \\ 2 \\ 5 \end{bmatrix}$
d=det(a)	Déterminant de la matrice (a)
[vec_p,val_p]=eig(a)	Calcule les vecteurs propres et les valeurs propres d'une matrice (a)
s1=sum(v2)	la somme des éléments d'un vecteur (v2)
p1=prod(v2)	produit des éléments d'un vecteur (v2)
[m1,i1]=max(v2)	m1 est le max d'un vecteur, i1 est sa position
[m2,i2]=min(v2)	m2 est le min d'un vecteur, i2 est sa position
m3=mean(v2)	la moyenne d'un vecteur
v3=a*v2	produit matrice (a) * vecteur colonne (v2)=vecteur v3
v4=v1*a	produit vecteur ligne (v1)* matrice(a)= vecteur v4
a3=v2* v1	produit vecteur colonne (v2)* vecteur ligne (v1) =matrice
scal=v1* v2	produit vecteur ligne (v1) * vecteur colonne (v2) = scalaire
v5=v1.* v2'	produit vecteur ligne (v1) .* vecteur ligne (transpose v2) = 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

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

Exemple : Calculer la valeur de y

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

```
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 :

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

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

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

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

Exercice 2 : écrire un programme Matlab qui déterminer :

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

```
clc ; clear all
nl=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 linge ===')
for i=1:nl
    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 collons ===')
for j=1:nc
    u(1:nl)=a(1:nl,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 dérive : $\text{diff}(f, x) = \frac{df}{dx} = \frac{d}{dx} \left(\frac{1}{x^2 + 2x + 2} \right)$
- calcule $\int f(x).dx = \int \frac{1}{x^2 + 2x + 2}.dx$
- calcule Transforme Laplace : $TL(f(t)) = \int_0^{\infty} f(t) * \exp(-Pt).dt = F(P)$
- calcule Transforme inverse de Laplace : $TL^{-1}(F(P)) = f(t)$
- résoudre équation différentielle du 1^{ere} ordre : $\frac{dy}{dx} = F(x, y) = (x^2 - 1) * y$
- résoudre équation différentielle du 2^{eme} 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)

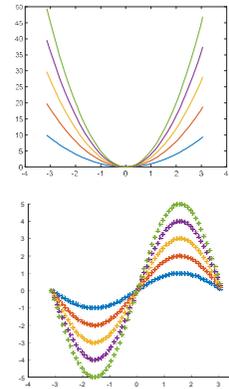
disp('==== integral =====')
syms x ;
intf=int('1/(x^2+2*x+2)',x)

disp('==== Transforme et invese de laplace ====')
syms p t a b;
f=@(t) exp(a*t)+sin(b*t)+t+1; laplace(f,p)
g=@(p) 1/(p-1)^2 ; ilaplace(g,t)

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*x.^2 ; plot(x,y) ;
end
%=====
figure; hold on;
for i=1:5
y=i*sin(x) ; 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');
    
```

