Exercice1 : - Considérons le nombre complexe suivant : z1= 1+2i et z2=3-5j

1) z1+z2=4.0000 -3.000i
2) z1-z2=-2.0000+7.000i
3) z1/z2=-0.2.059+0.3235i
4) z1*z2=13.000+1.0000i

2- Calculez z2 à la puissance 2, z2^2=-16.0000-30.0000i

Exercice2 : - Considérons le nombre complexe suivant :

z=[1+j 2-3j ; 4+2j 5-4j]

Donnez la partie réelle, la partie imaginaire , le conjugué et le module de z sous matlab
produit de z par son conjugué
produit scalaire de z par son conjugué

```
>> z=[1+j 2-3j ; 4+2j 5-4j]
z =
   1.0000 + 1.0000i   2.0000 - 3.0000i
   4.0000 + 2.0000i   5.0000 - 4.0000i
>> r=real(z)      r =     1     2
                          4     5
m=imag(z)     m =    1    -3
                     2    -4
>> conj(z)
ans =     1.0000 - 1.0000i   2.0000 + 3.0000i
          4.0000 - 2.0000i   5.0000 + 4.0000i
>> abs(z)              ans =    1.4142    3.6056
                                4.4721    6.4031
>> conjz=conj(z)    conjz =  1.0000 - 1.0000i
                             2.0000 + 3.0000i
>> produitscalaire=conjz.*z
    produitscalaire =     2    13
                         20    41
>> produit=z*conjz
=   4.0000 -16.0000i  21.0000 - 2.0000i
      18  -28.0000i  43.0000 +16.0000i
```

Exercice 3 : >> A=[6 43 2 11 87 ; 12 6 34 0 5 ; 34 18 7 41 9]

```
A =   6    43     2    11    87
     12     6    34     0     5
     34    18     7    41     9
>>va=A(2,:)
 va =     12     6    34     0     5
>> vb=A(:,4)
vb =   (  11    0   41 )'
vc=A(1:2,:)
 vc            6    43     2    11    87
              12     6    34     0     5
>> vd=A(:,[1,5])     vd =     6    87
                             12     5
                             34     9
>> vg=A(:,1)+A(:,4)
 vg =     17
          12
          75
```

Exercice4 :

On se propose de tracer la courbe suivante $y = cos2x + \frac{1}{2}sin\frac{x}{2}$

La variable x est un vecteur dont les valeurs vont de $-\pi$ à π par pas de $\frac{\pi}{100}$

1. Ecrire le code Matlab qui va tracer la courbe y en fonction de x
2. Donner un titre à cette figure « tracé de la courbe y »
3. Ajouter un label des axes x en abscisses « variable x »
4. Ajouter un label des y en ordonnées « variable y »
5. Utiliser la commande qui permet de réaliser un quadrillage afin de visualiser les valeurs en ordonnées et en abscisses

```
>>x=pi :pi/100 ;pi ;
>>cos(2*x)+0.5*sin(x/2)
>>plot(x,y)
>>grid;
>>xlabel(‘variable x  ‘)
>> ylabel(‘variable y  ‘)
```

Exercice5 :

Soit la fonction suivante à deux variables

$z = \frac{\sin(x^2+y^2)}{x^2+y^2}$

1. Ecrire le code Matlab qui va tracer la courbe z en fonction de x et de y

Remarque :On génère deux matrices carrées X et Y qui définissent le domaine de calcul de z ; On utilise pour ceci une fonction meshgrid

pour x et y variant de $-\pi$ à π par pas de $\frac{\pi}{100}$

\>> x=-pi :pi/100 :pi ;

\>>y=x ;

on génère deux matrices carrées X et Y qui définissent le domaine de calcul de z ; On utilise pour ceci une fonction meshgrid

\>>Z=sin(X.^2+Y.^2)./(X.^2+Y.^2);

\>>mesh(Z)

Exercice 6: Créez un tableau tab contenant les entiers pairs de 0 à 42, puis écrivez un script (une fonction) qui parcourt le tableau et remplace chacune des valeurs par son carré.

Remarque : La fonction size renvoie les dimensions d'une matrice. Ici tab est un tableau d'une ligne et 22 colonnes, mais seule la seconde dimension (nombre de colonnes) nous intéresse, d'où size(tab,2)qui demande la taille de tab dans la deuxième dimension.

où bien on utilise length(tab)

Script :MATLAB\pr1

```
tab = 0:2:42;
for i = 1:size(tab,2)
tab(i) = tab(i).^2
end
```

*Exercice7 : (*Boucle for)

```
vect=[];
for i=1:inf
   if (rem(i,5)==0)
      vect=[vect i];
   end
   if length(vect)==10
      break
   end
end
```

*Exercice 8 : (*While)

```
vect=[];
i=1;
while (length(vect)<10)
   i=i+1;
   if (rem(i,5)==0)
      vect=[vect i];
   end
end
```

*Exercice 9 : (*Factorielle)

```
function fact = factorielle(n)
% cette fonction calcule la factorielle
% d'un nombre entier n
% test de la nature entière et positive
% de l'argument d'entrée
if ~((fix(n)==n) &(n>=0))
   error('le nombre doit être entier positif')
end
% calcul de la factorielle par la commande prod
fact = prod(1:n);
```

*Exercice 10: (*sature)

```
function u_limite = sature(u, u_min, u_max)
 % limitation d'un signal
 % si signal < u_min alors signal = u_min,
 % si signal > u_max alors signal = u_max,
 % si u_min <= signal <= u_max alors signal non modifié
 % expressions logiques retournant 0 ou 1
 expr1 = (u >= u_max);
 expr2 = (u <= u_max);
 expr3 = ((u < u_max) & (u > u_min));

 u_limite = expr1 .* u_max + expr2 .* u_min + expr3.*u;
```

*Exercice 10: (*sature2)

```
function ul = sature2(u, u_min, u_max)
% limitation d'un signal
% si signal < u_min alors signal = u_min
% si signal > u_max alors signal = u_max
% si u_min <= signal <= u_max alors signal non modifié

ul = (u>=u_max).*u_max +
(u<=u_min).*u_min + ((u<u_max)& ...
(u>u_min)).*u;
```

Exercice 11 :

```
% définition du domaine de valeurs de la variable x
x = 0:0.001:10;
% expressions des 2 fonctions y1 et y2
y1=cos(x)-0.5;
y2=sin((-2.5+log(x+1))/3);
```

```
% affichage des 2 courbes des 2
fonnctions y1 et y2
plot(x,y1)
% on garde la même fenêtre pour
tracer y2
hold on
% tracé de y2
plot(x,y2)
% insertion d'une grille
grid
% on cherche les indices des
vecteurs x, y1 ou y2
% qui satisfont l'égalité de y1 et
de y2 avec une
% précision de 0.0005
i=find(abs(y1-y2)<0.0005);
% affichage des indices des
solutions
disp('indices des solutions :')
i
% affichage des solutions
disp('valeurs des solutions :')
sol = x(i)
% tracé des points de rencontre des
2 courbes
% aux valeurs de ces indices
h = plot(x(i), y1(i), '*');
set(h,'Linewidth',5)
% légendes des axes
xlabel('variable x')
ylabel('valeurs des fonctions y1 et
y2')
title('résolution graphique d''un
système d''équations')
```

Exercice 12: ***Fonction calcul qui calcule la* moyenne et la variance**

```
function [moy, var]=
calcul_moy_var(w)

% calcul de la moyenne et de
la variance
moy=mean(w);
var=std(w)^2;
```

Exercice 13: ***tracé de la fonction sinus cardinal***

```
  close all, clear all
% tracé de la fonction sinus
cardinal

% domaine des valeurs de la variable
x
x = -4*pi:pi/100:4*pi;

% valeurs de la fonction
y = (x==0)+sin(x)./(x+(x==0));

% tracé de la fonction sinus
cardinal
plot(x,y)
grid
title('sinus cardinal y = sin(x)/x')
axis([-15 15 -0.4 1.2])

function Nbres_pairs=N_pairs(N)
```

Exercice 13: ***génération de N premiers nombres pairs avec la boucle (For)***

```
% génération de N premiers nombres
pairs
Nbres_pairs=[];
for i=0:N
  if rem(i,2)==0
     Nbres_pairs=[Nbres_pairs i];
  end
end
```

Exercice 13: ***Nombres pairs avec la boucle (While)***

```
function Nbres_pairs=N_pairs3(N)
% génération de N premiers nombres
pairs
Nbres_pairs=[];
i=0;
while i>=0
   if rem(i,2)==0
     Nbres_pairs=[Nbres_pairs i];
  end
  i=i+1;
  if i>N
     break
  end
end
```

Exercice 14: ***(plot3)***

```
% efface les variables de l'espace
de travail
% et les fenêtres graphiques actives
clear all, close all
% x : angle dans l'intervalle [-2pi
2pi]
x=-2*pi:pi/100:2*pi;
% vecteur y (2ème dimension)
y = 2*x;
% vecteur z (3ème dimension)
z = sinc(x-y);
% la fonction plot retourne h
h=plot3(x,y,z);
% spécification de la largeur du
trait
set(h,'LineWidth',2)
grid
```

Exercice 15: ***(meshc)***

```
x = -1:0.1:1;
```

```
y = 2*x;
[X,Y] = meshgrid(x,y);
Z = exp(X.^2 - Y.^2);
meshc(Z)
axis([0 20 0 20 0 3])
```

Exercice 15: *(mesh)*

```
x = -1:0.1:1;
y = 2*x;
[X,Y] = meshgrid(x,y);
Z = sinc(X) .* exp(-X.^2 - Y.^2);
mesh(Z)
axis([0 20 0 20 0 1])
```

Exercice 16:

```
% efface les variables de l'espace de travail
% et les fenêtres graphiques actives
clear all, close all
% x : angle dans l'intervalle [-2pi 2pi]
x=-2*pi:pi/100:2*pi;
% fonction sinc à tracer
y=sinc(x);
% la fonction plot retourne le ponteur h
h=plot(x,y);
% modification de la largeur du trait
set(h,'LineWidth',2)
% fonction sinusoïdale amortie
a=0.5;
z=exp(-a*abs(x)).*sin(x);
% on maintient l'écran graphique pour recevoir
% la deuxième courbe
hold on
plot(x,z)
% choix des dimensions des axes
axis([-7.5 7.5 -0.6 1.2])
% ajout d'une grille
grid
```

Exercice 16:

```
close all
 x=1:12;
 t=[4.9 6.5 9.3 8.9 9.2 15.7 19.8 21.1 20 16 11 7.6000] ;
 bar(x,t)
 ylabel('axe de l''histogramme')
%  h1=gca;
% h2=axes('Position',get(h1,'Position'));
%  y=-0.0804*x.^3+1.11999*x.^2-2.9354*x+7.1162;
%  plot(x,y,'Linewidth',2)
% set(h2,'YaxisLocation','right','Color','none','XtickLabel',[])
% set(h2,'Xlim',get(h1,'Xlim'),'Layer','top')
%  ylabel('axe de la courbe continue')
%  text(2,6,'texte incliné','Rotation',35)
%  grid
%  title('Tracé de 2 courbes dans 2 systèmes d''axes différents','FontSize',12)
```

Exercice 17: *(sin, sinc) avec plot , subplot*

```
close all
x = -2*pi:pi/100:2*pi;
y = sin(x);
z = sinc(x);
w = sin(2*x).*exp(-0.5*abs(x));
figure('Name','Etude de tracés continus','NumberTitle','off')

% matrice d'une ligne et 3 colonnes (3 sous-fenêtres graphiques)
% spécification de la 1ère fenêtre
subplot(211)

% tracé des 3 courbes à la fois avec styles de traits différents
plot(x,y,'-.',x,z,'-',x,w,':')
titre=title('plusieurs courbes avec une commande plot');
set(titre,'FontName','Times')

% gestion des pointeurs sur chacune des courbes
subplot(212)
h1=plot(x,y); % pointeur sur le tracé y(x)
set(h1,'Linewidth',2,'Color',[1 0 0])
% pointeur sur le titre
ht=title('gestion des pointeurs des différentes courbes');
set(ht,'FontName','Arial','FontSize',12)
hold on
h2=plot(x,z);
set(h2,'Linestyle',':')
xlabel('x allant de -2\pi à 2\pi par pas de \pi/100')
```