

TP 2 : : Définition et traçage des courbes de quelques fonctions test en optimisation

2.1 Introduction :

Le principe général des représentations graphiques est de se ramener à des calculs sur des matrices ou des vecteurs. Ainsi la représentation d'une fonction de $\mathbb{R}$ dans $\mathbb{R}^2$ commencera par la création d'un vecteur d'abscisses, en général régulièrement espacées, auxquelles on applique la fonction pour créer le vecteur des ordonnées.

Pour la représentation d'une surface, il faudra créer la matrice des points d'une grille rectangulaire dans $\mathbb{R}^2$. La commande hold permet de superposer des tracés successifs sur une même figure.

2.2 Graphisme :

Il existe plusieurs commandes pour le traçage des courbes à 2 dimensions ou à 3 dimensions suivant le cas.

2.2.1 Graphisme en deux dimensions :

Dans le cas d'un graphe en deux dimensions en retrouve les instructions suivantes :

plot	trace point par point un graphe 2D
semilogx	identique à <i>plot</i> mais avec échelle logarithmique pour l'axe des abscisses
semilogy	identique à <i>plot</i> mais avec échelle logarithmique pour l'axe des ordonnées
loglog	identique à <i>plot</i> mais avec échelles logarithmiques pour les deux axes
grid	ajoute une grille
xlabel	ajoute une légende pour l'axe des abscisses
ylabel	ajoute une légende pour l'axe des ordonnées
title	ajoute un titre
axis	modifie les échelles des axes
zoom	effectue un zoom
gtext	place une légende avec la souris
hold	ajoute un graphe dans la fenêtre courante
figure	crée une nouvelle fenêtre

Il existe de nombreuses manières qui permettent de tracer des courbes variées, des exemples illustratifs sont donnés :

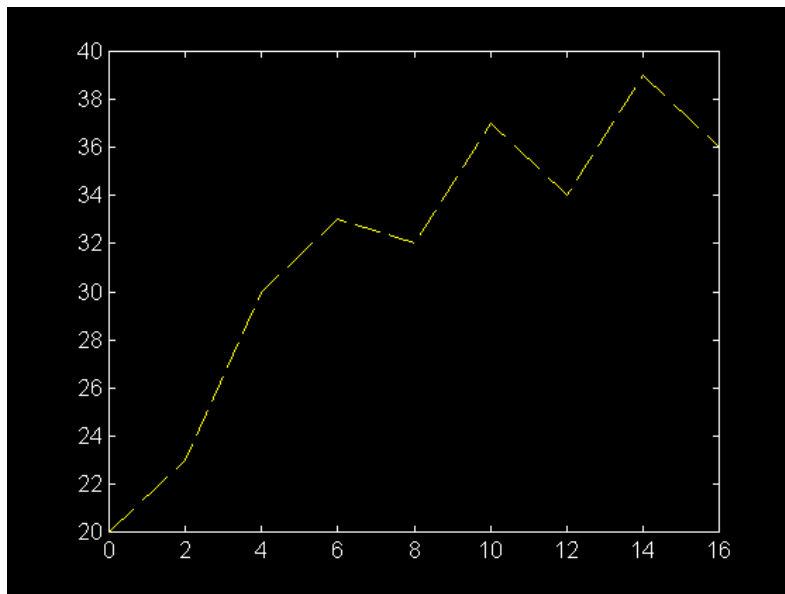
Exemple 1 :

Soit l'exemple suivant ; on veut tracer l'évolution des températures en fonction du temps.

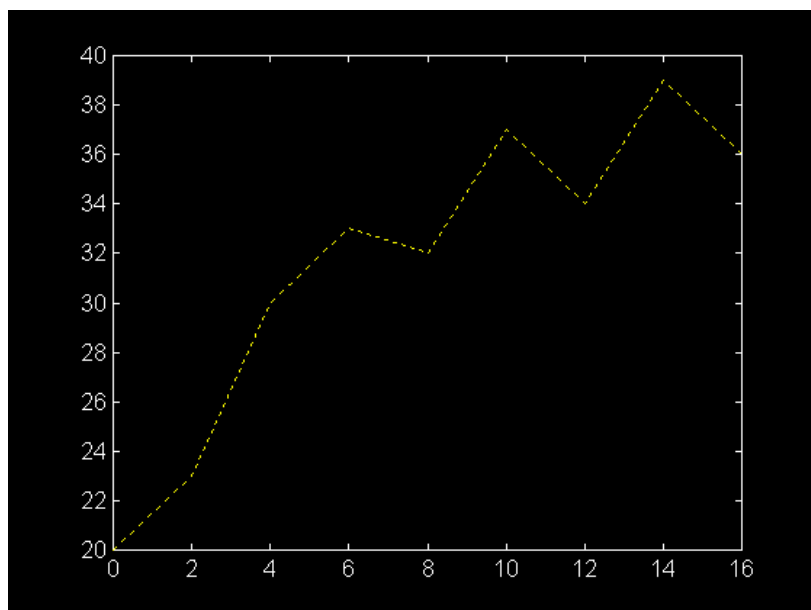
Temps (heures)	0	2	4	6	8	10	12	14	16
Température (°C)	20	23	30	33	32	37	34	39	36

Il faut introduire ces données sous forme de vecteurs.

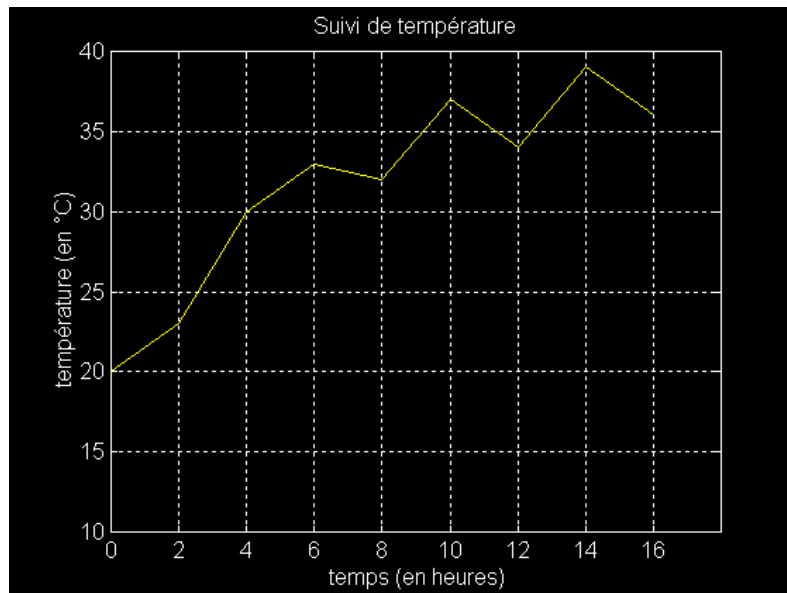
```
>> temps = [0 2 4 6 8 10 12 14 16]
temps =
0 2 4 6 8 10 12 14 16
>> temperature = [20 23 30 33 32 37 34 39 36]
temperature =
20 23 30 33 32 37 34 39 36
>> plot ( temps , temperature , '--' )
```



```
>> plot ( temps , temperature , ':' )
```



```
>> plot (temps , temperature)
>> grid on
>> xlabel ( 'temps (en heures)' )
>> ylabel ( 'température (en °C)' )
>> title ( 'Suivi de température' )
>> axis ( [ 0 18 10 40 ] )
```



On peut spécifier à MATLAB quelle doit être la couleur d'une courbe, quel doit être le style de trait et/ou quel doit être le symbole à chaque point (x_i, y_i). Pour cela on donne un troisième paramètre d'entrée à la commande plot qui est une chaîne de 3 caractères de la forme '**cst**' avec **c** désignant la couleur du trait, **s** le symbole du point et **t** le type de trait. Les possibilités sont les suivantes:

y jaune	. point	- trait plein
m magenta	o cercle	: pointillé court
c cyan	x marque x	- pointillé long
r rouge	+ plus	-. pointillé mixte
g vert	* étoile	
b bleu	s carré	
w blanc	d losange	
k noir	v triangle (bas)	
^ triangle (haut)		
< triangle (gauche)		
> triangle (droit)		
p pentagone		
h hexagone		

Les valeurs par défaut sont **c = b**, **s = .** et **t = -** ce qui correspond à un trait plein bleu reliant les points entre eux. Il n'est pas obligatoire de spécifier chacun des trois caractères. On peut se contenter d'en spécifier un ou deux. Les autres seront les valeurs par défaut.

Exemple 2 :

C'est l'exemple d'un graphe d'une fonction à une variable, soit la fonction suivante :

$$y = \frac{t(1+t^2)}{10}$$

Création du vecteur t :

```
>> t = 0 : 0.01 : 1
```

t =

Columns 1 through 7

```
0 0.0100 0.0200 0.0300 0.0400 0.0500 0.0600
```

...

Columns 99 through 101

```
0.9800 0.9900 1.0000
```

Création du vecteur y :

Attention : ne pas oublier le point
devant les opérateurs
multiplication, puissance et division

:

.*

.^

./

```
>> y = t.*(1 + t.^2)./10
```

y =

Columns 1 through 7

```
0 0.0010 0.0020 0.0030 0.0040
```

```
0.0050 0.0060
```

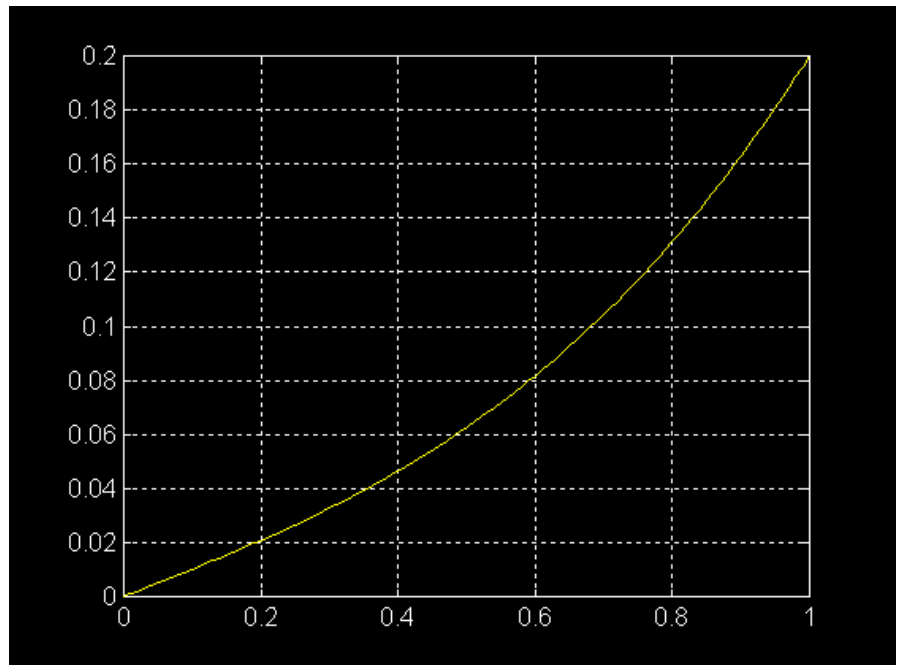
...

Columns 99 through 101

```
0.1921 0.1960 0.2000
```

```
>> plot ( t , y )
```

```
>> grid on
```



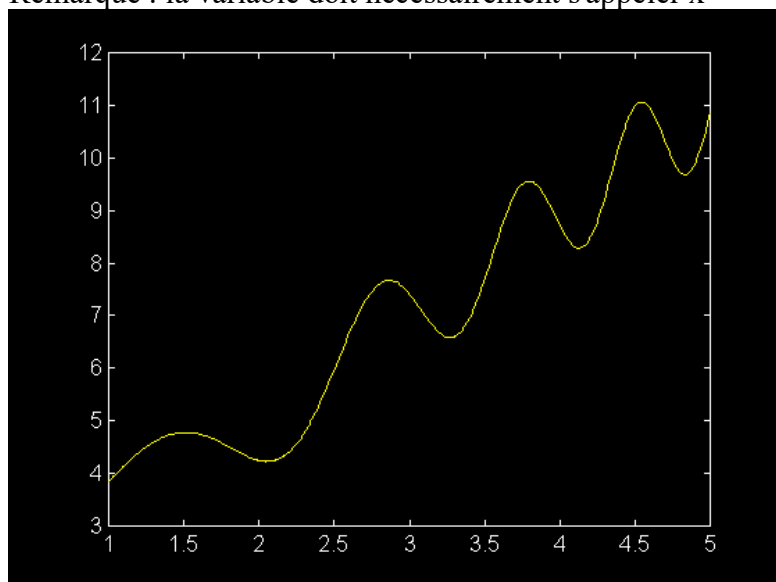
Exemple 3 :

On veut tracer le graphe de la fonction :

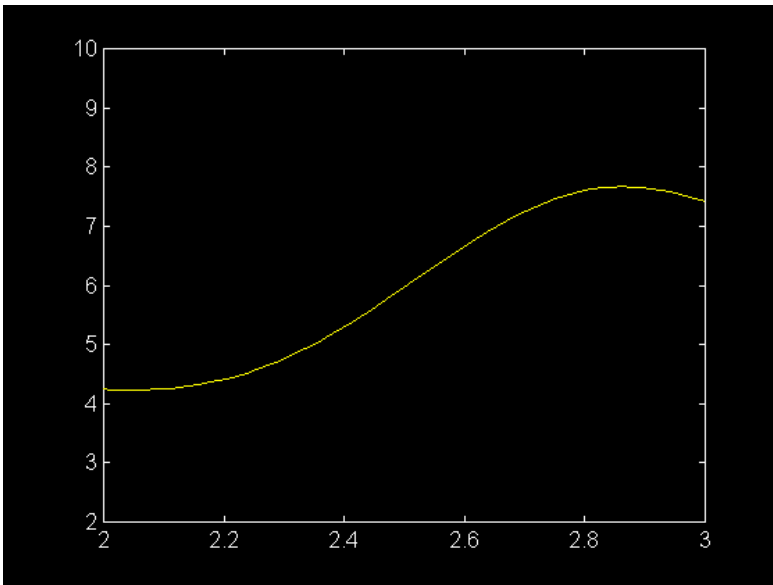
$$y = f(x) = 1 + 2x + \sin(x^2)$$

```
>> fplot('1+ 2*x + sin(x*x)', [ 1 5 ])
```

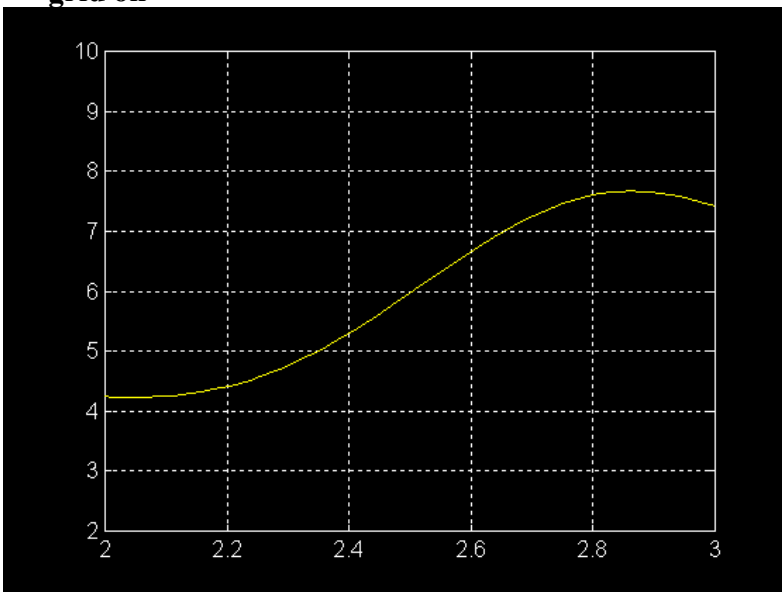
Remarque : la variable doit nécessairement s'appeler x



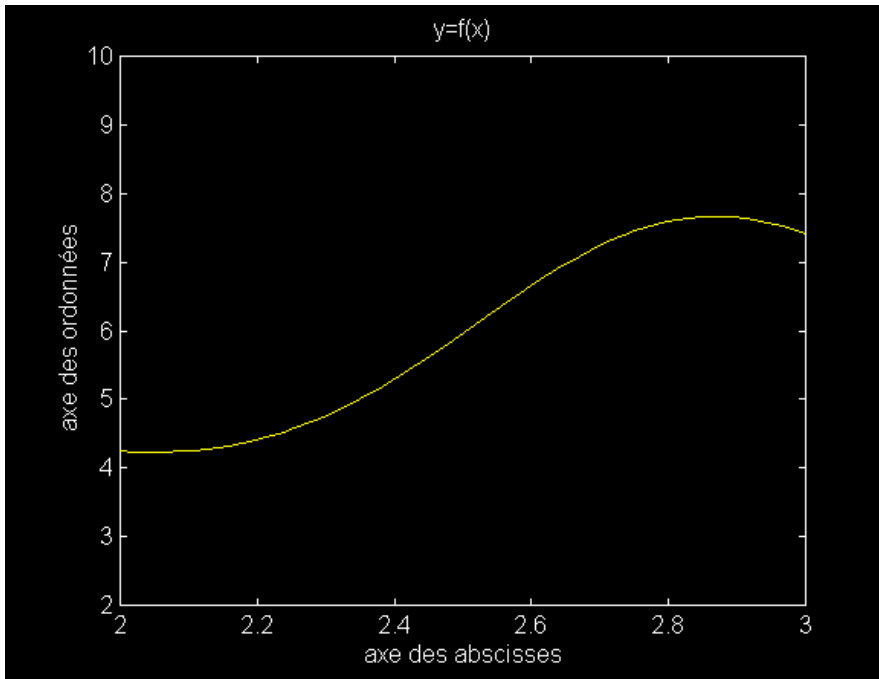
```
>> fplot('1+ 2*x + sin(x*x)', [ 2 3 2 10 ])
```



```
>> grid on
```



```
>> grid off  
>> xlabel('axe des abscisses')  
>> ylabel('axe des ordonnées')  
>> title('y=f(x)')
```



>> zoom on

click droit : zoom arrière

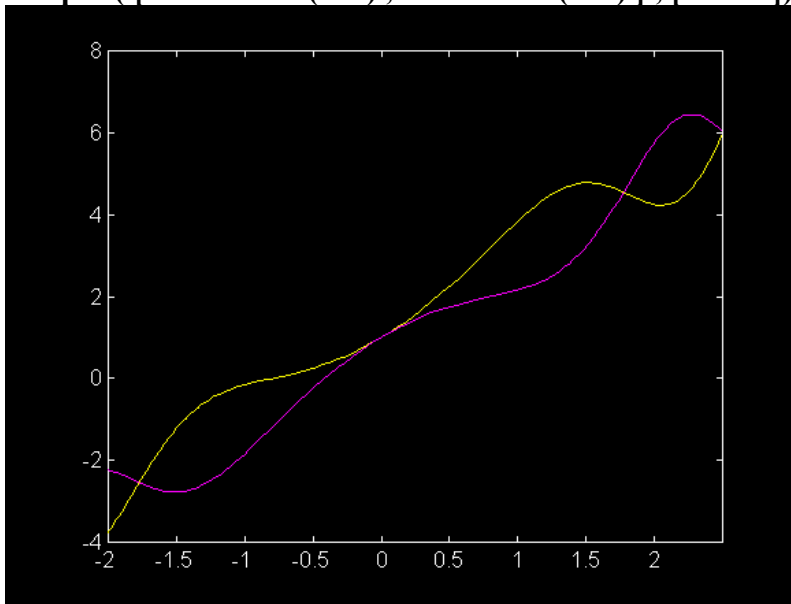
click gauche : zoom avant

click gauche et glissé : zoom d'une zone

>> zoom off

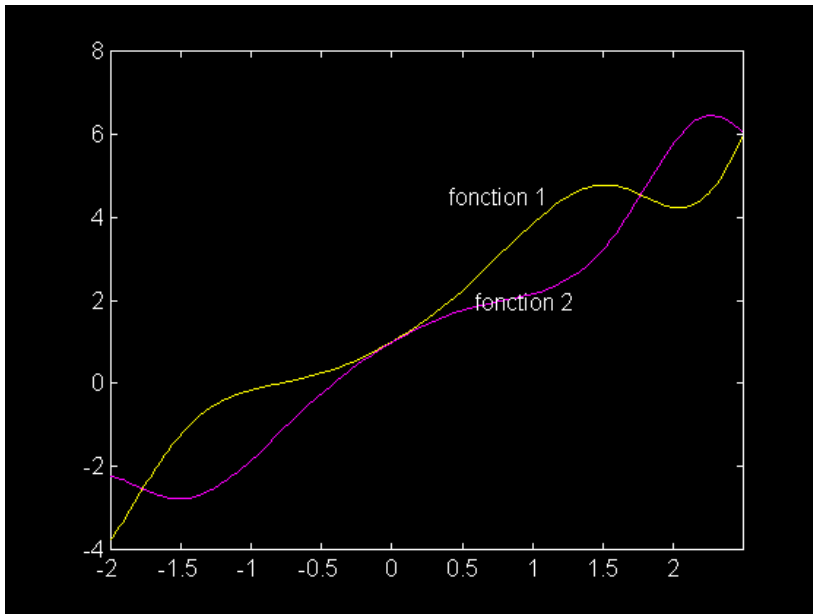
Pour tracer plusieurs graphes dans la même fenêtre :

>> fplot('1+ 2*x + sin(x*x) , 1+ 2*x - sin(x*x) ', [-2 2.5])



>> gtext('fonction 1')

>> gtext('fonction 2')



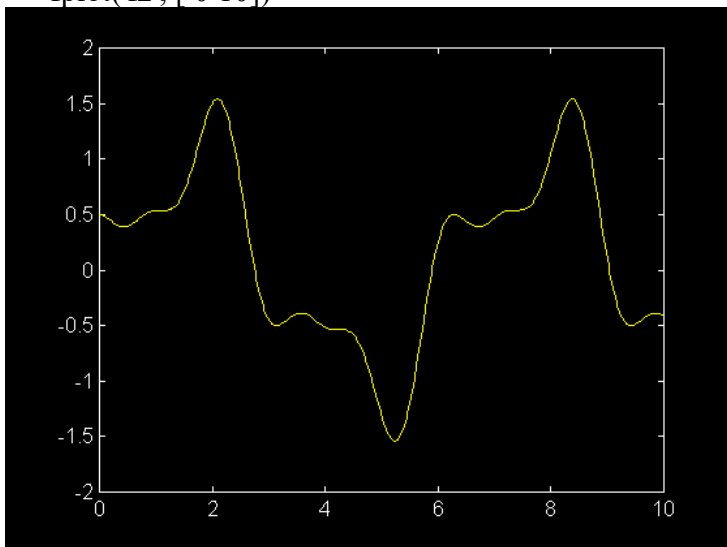
Exemple 4 :

Soit la fonction suivante :

$$y = f(x) = \sin(x) + \frac{\cos(3x)}{2} - \frac{\sin(5x)}{5}$$

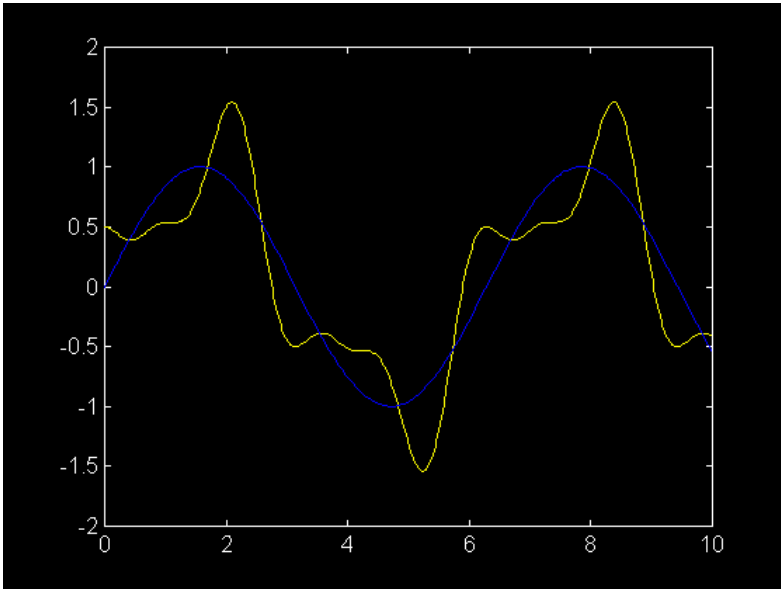
```
f2.m - Bloc-notes
Fichier Edition Format Affichage ?
function y=f2(x) ;
y = sin(x) + 0.5.*cos(3.*x) - 0.2.*sin(5.*x) ;
```

>> fplot('f2', [0 10])

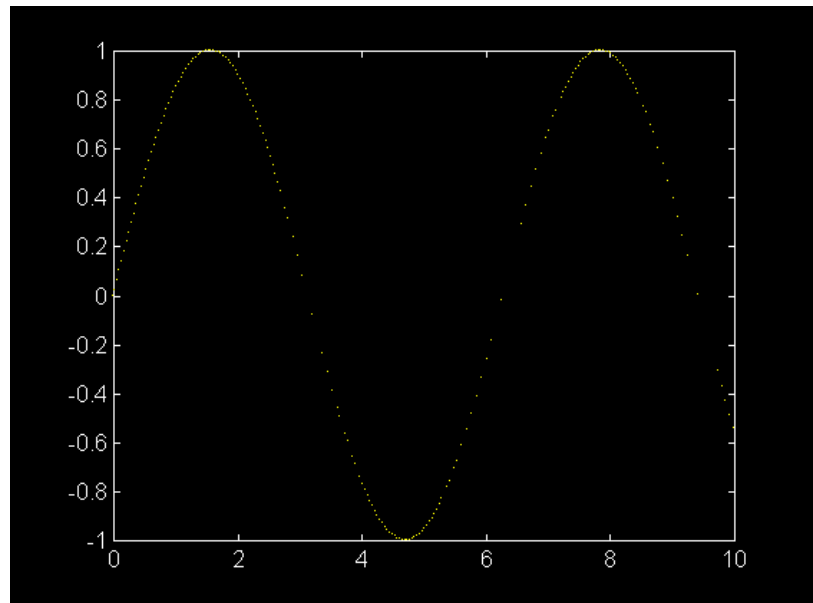


>> hold on

>> fplot('sin(x)', [0 10], 'b')



```
>> hold off
>> [X Y] = fplot ( 'sin(x)' , [ 0 10 ]
)
X =
0
0.0200
0.0600
...
9.8700
9.9350
10.0000
Y =
0
0.0200
0.0600
...
-0.4307
-0.4884
-0.5440
>> fplot ( 'sin(x)' , [ 0 10 ] , '.' )
```



2.2.2 Graphisme en trois dimensions :

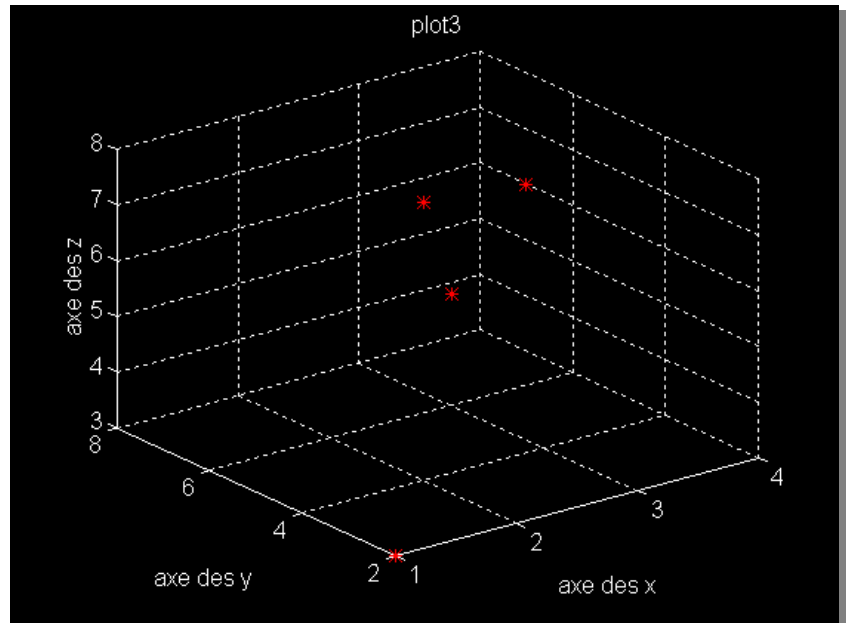
Dans le cas d'un graphe en trois dimensions en retrouve les instructions suivantes :

meshgrid	(voir l'exemple)
mesh	(voir l'exemple)
meshc	(voir l'exemple)
meshz	(voir l'exemple)
contour	(voir l'exemple)
view	ajuste l'angle de vue
grid	ajoute une grille
xlabel	ajoute une légende pour l'axe des abscisses
ylabel	ajoute une légende pour l'axe des ordonnées

zlabel	ajoute une légende pour l'axe des z
title	ajoute un titre
hold	ajoute un graphe dans la fenêtre courante
figure	crée une nouvelle fenêtre

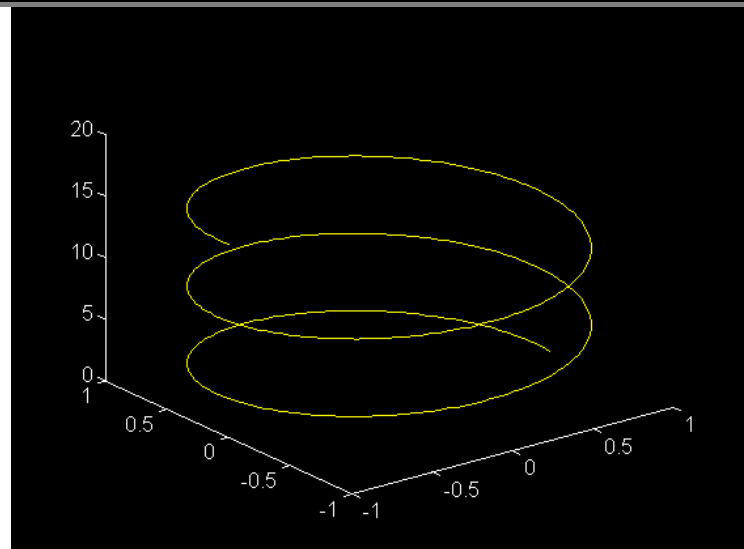
Exemple 1 :

```
>> x = [ 1 2 3 4 ]
x =
1 2 3 4
>> y = [ 2 4 6 7 ]
y =
2 4 6 7
>> z = [ 3 8 5 6 ]
z =
3 8 5 6
>> plot3 (x , y , z , 'r*')
>> grid on
>> xlabel('axe des x')
>> ylabel('axe des y')
>> zlabel('axe des z')
>> title('plot3')
```



Exemple 2 : c'est le cas d'une équation paramétrique

```
x = cos (t)
y = sin (t)
z = t
>> t = 0 : pi/100 : 5*pi
>> plot3 (cos(t) , sin(t) , t)
```



Exemple 3 : c'est le graphe d'une fonction à deux variables $z = f(x,y)$

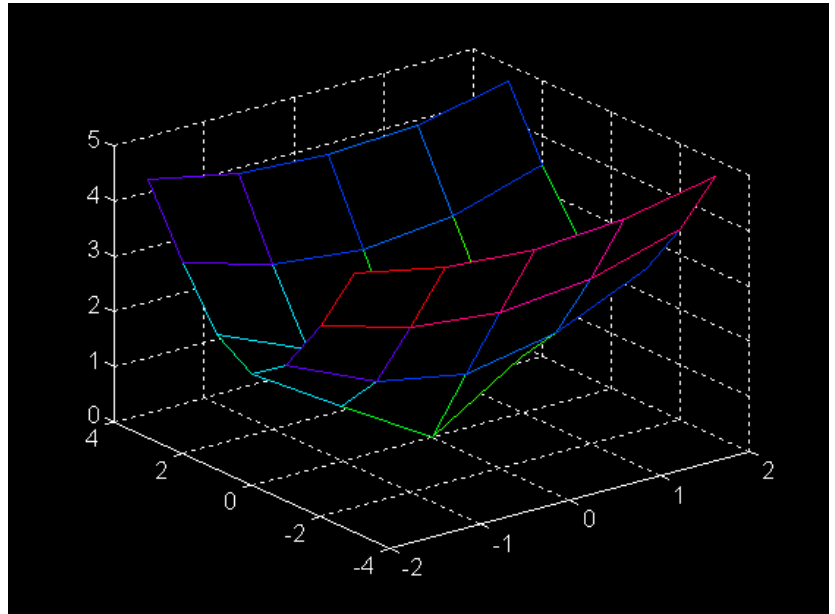
$$z = f(x, y) = \sqrt{x^2 + 2y^2}$$

```
>> x = -2 : 2
x =
-2 -1 0 1 2
>> y = -3 : 3
y =
-3 -2 -1 0 1 2 3
>> [X , Y] = meshgrid(x , y)
X =
-2 -1 0 1 2
```

```

-2 -1 0 1 2
-2 -1 0 1 2
-2 -1 0 1 2
-2 -1 0 1 2
-2 -1 0 1 2
-2 -1 0 1 2
Y =
-3 -3 -3 -3 -3
-2 -2 -2 -2 -2
-1 -1 -1 -1 -1
0 0 0 0 0
1 1 1 1 1
2 2 2 2 2
3 3 3 3 3
>> Z = sqrt(X.^2 + 2.*Y.^2)
Z =
4.6904 4.3589 4.2426 4.3589 4.6904
3.4641 3.0000 2.8284 3.0000 3.4641
2.4495 1.7321 1.4142 1.7321 2.4495
2.0000 1.0000 0 1.0000 2.0000
2.4495 1.7321 1.4142 1.7321 2.4495
3.4641 3.0000 2.8284 3.0000 3.4641
4.6904 4.3589 4.2426 4.3589 4.6904
>> mesh(X, Y, Z)
>> grid on

```

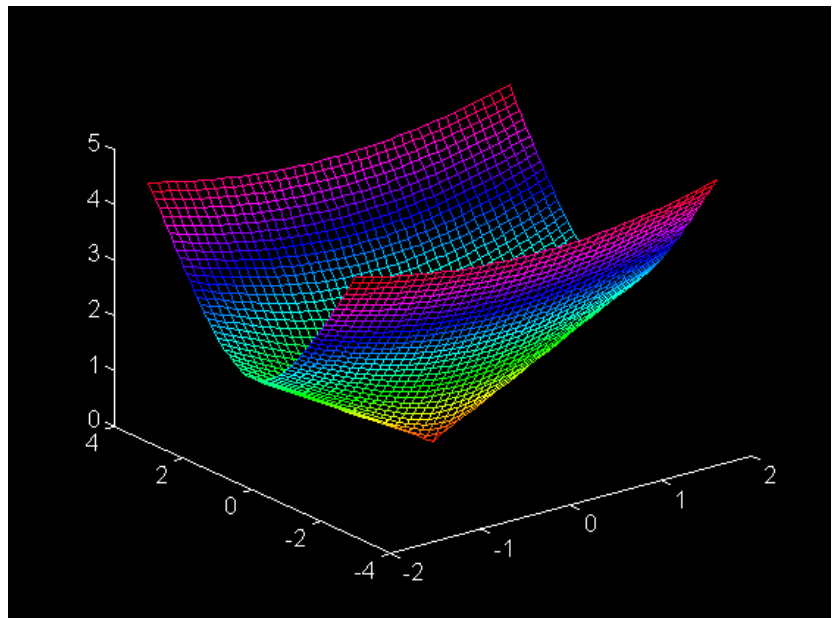


Pour une meilleure résolution :

```

>> x = -2 : 0.1 : 2
>> y = -3 : 0.1 : 3
>> [X, Y] = meshgrid(x, y)
>> Z = sqrt(X.^2 + 2.*Y.^2)
>> mesh(X, Y, Z)

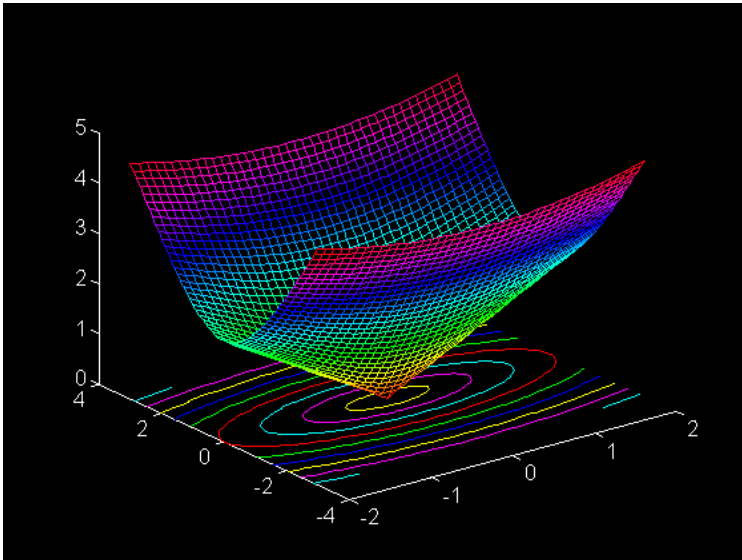
```



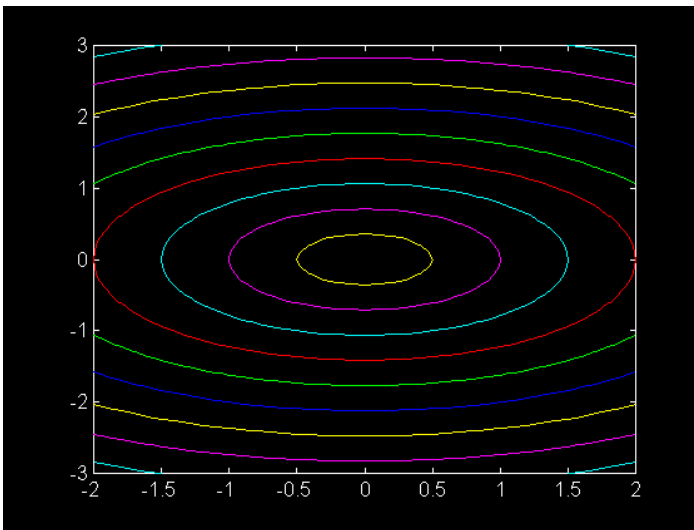
```

>> meshc(X, Y, Z)

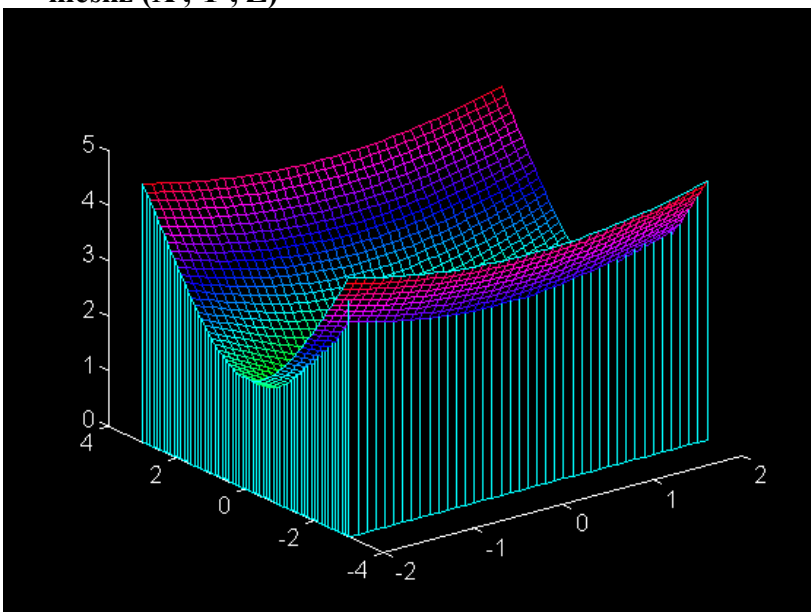
```



```
>> contour (x , y , Z)
```

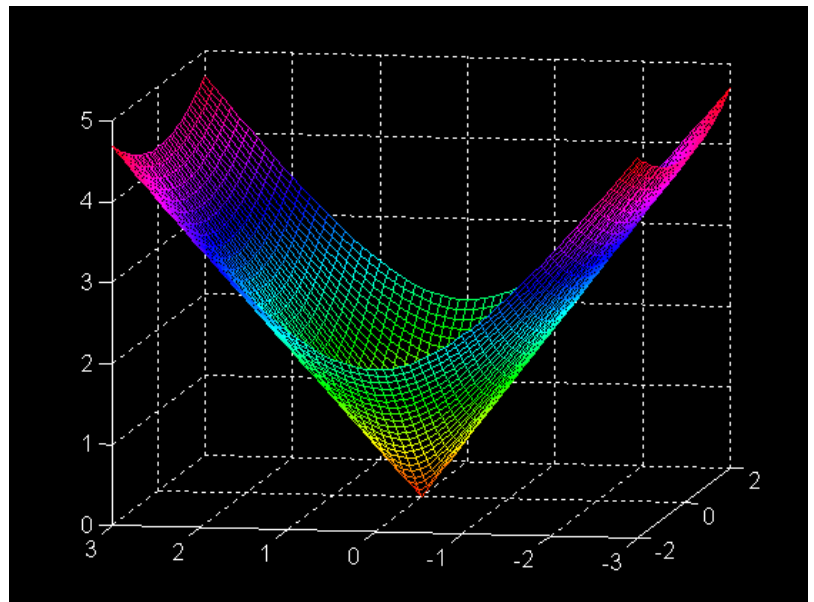


```
>> meshz (X , Y , Z)
```



```
>> mesh (X , Y , Z)
>> grid on
>> view(-80 , 10)
```

Remarque : view (-37.5 , 30) par défaut

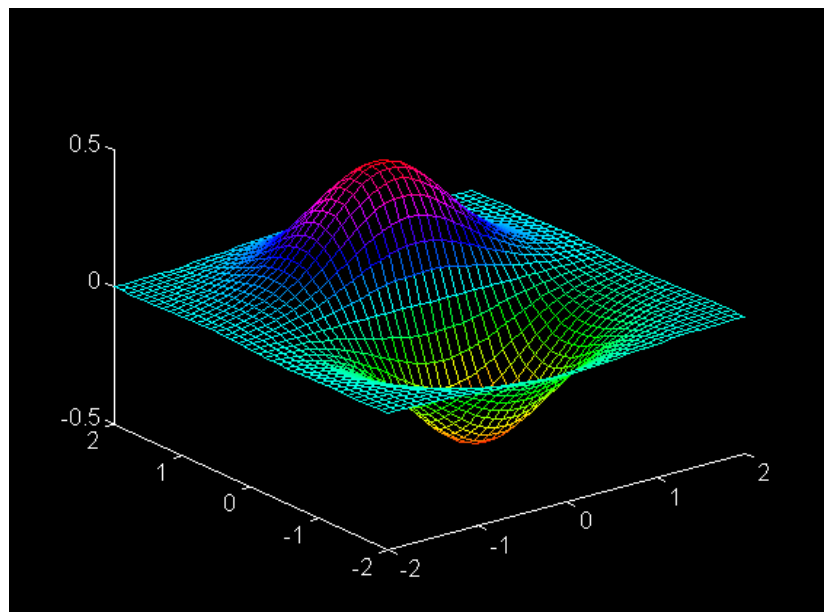


Exemple 4 :

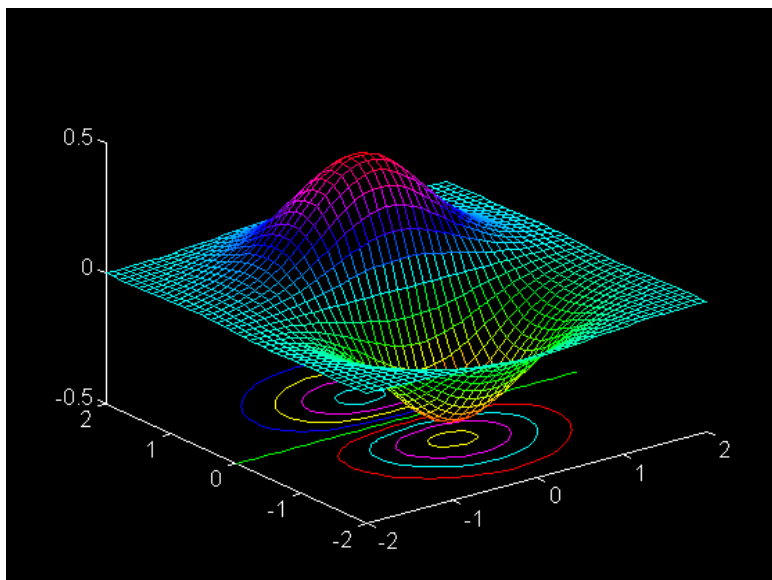
Soit l'exemple suivant :

$$z = f(x, y) = y \cdot \exp(-x^2 - y^2)$$

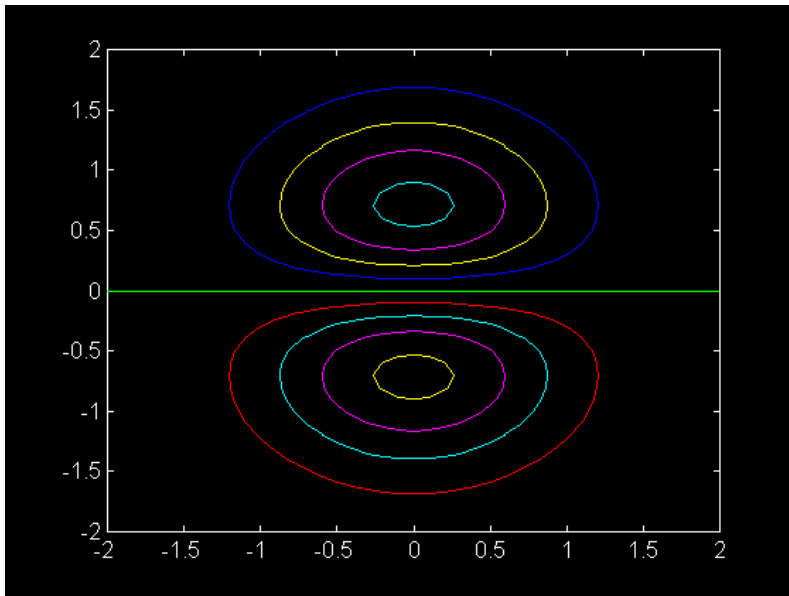
```
>> x = -2 : 0.1 : 2
>> y = -2 : 0.1 : 2
>> [X , Y] = meshgrid(x , y)
>> Z = Y.*exp ( -X.^2 - Y.^2)
>> mesh (X , Y , Z)
```



```
>> meshc (X , Y , Z)
```



```
>> contour (x , y , Z)
```



2.3 Minimum d'une fonction f(x)

La commande qui permet de déterminer le minimum d'une fonction est : **Fminbnd**, un exemple illustratif est donné :

Soit la fonction suivante :

$$y = -\frac{1}{10} + \frac{1}{2+(5-x)^2} + \frac{1}{1+(1+x)^2}$$

Dessignons maintenant le graphe de cette fonction :

>> fplot('f1', [-10 10])

>> grid on

Cherchons le minimum de cette fonction

dans l'intervalle 0 à 5 :

>> fmin('f1', 0, 5)

ans =

2.1177

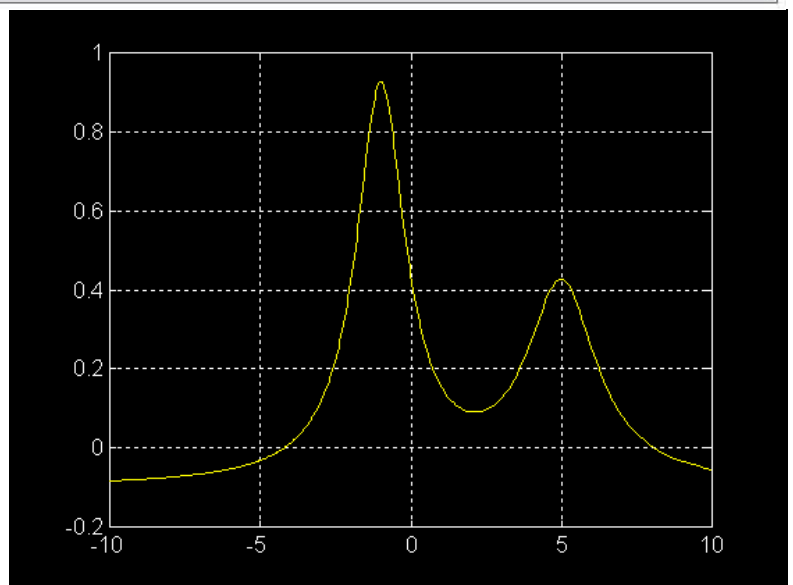
>> f1(ans)

ans =

0.0903

```
f1.m - Bloc-notes
Fichier Edition Format Affichage ?
function y=f1(x) ;

y = -0.1 + 1./(2+(5-x).^2) + 1./(1+(1+x).^2) ;
```



2.4 Equation non linéaire à une inconnue et racines d'une fonction

Soit à résoudre l'équation :

$$-6 + 2t + \sin(t^2) = 0$$

Cela revient à trouver les racines de la fonction :

$$y(t) = -6 + 2t + \sin(t^2)$$

```
f3.m - Bloc-notes
Fichier Edition Format Affichage ?
function y=f3(t);
y = -6 + 2.*t + sin(t.^2);
```

Représentation graphique de la fonction :

```
>> fplot('f3', [ 0 5 ])
>> grid on
```

Cela fait apparaître 3 racines, aux environs de 2,7 3,1 et 3,4.

```
>> fzero('f3', 2.7)
ans =
2.6545
Pour les maniaques de la précision :
>> format longe
>> fzero('f3', 2.7)
ans =
2.654461193307640e+000
>> format short
>> fzero('f3', 3.1)
ans =
3.1039
>> fzero('f3', 3.4)
ans =
3.4078
```

Vérification :

```
>> f3(ans)
ans =
5.5511e-016
```

