

Initiation à MATLAB

TP3

Exercice 1

$$g(x) = \begin{cases} 0 & x < -1 \\ 1/(e^{-15(x+3/4)} + 1) & -1 \leq x < -1/2 \\ 1 & -1/2 \leq x < 3 \\ e^{-15(x-7/2)^2} & 3 \leq x < 4 \\ 0 & x \geq 4 \end{cases}$$

1. Ecrire un M file pour la fonction

sur l'intervalle $[-2, 5]$

2. Tester la fonction sur quelques valeurs, par exemple $g(-1)=0.0230$, $g(3)=0.0235$.

3. Créer un tableau d'abscisses x de -2 à 5 par pas de 0.01 .

4. Représenter la fonction g aux points x_i , avec la commande `plot(x,g(x))`.

Corrigé de l'exercice 1

1. La fonction $x \mapsto g(x)$ est définie par morceaux. Son expression change en fonction de l'intervalle considéré. En choisissant de l'étudier sur l'intervalle $[-2, 5]$, il y a 5 intervalles différents qui partitionnent le domaine d'étude. Sur chacun d'eux la fonction admet une expression différente.

On peut définir la fonction g à l'aide de la notion de test logique à base des conditions `if_else_end` sur la valeur de x ,

On peut aussi définir cette fonction directement à l'aide des fonctions indicatrices des sous intervalles du domaine d'étude. Le M-file suivant qui porte le nom `fonction_1.m` permet de définir explicitement la fonction $x \mapsto g(x)$ sur le domaine $[-2, 5]$.

```
function [gdex]=fonction_1(x)

c1=x>=-2 & x<-1;

c2=x>=-1 & x<-1/2;

c3=x>=-1/2 & x<3;

c4=x>=3 & x<4;
```

```
gdex=c1*0+c2*(1/(exp(-15*(x+3/4))+1))+c3*1+c4*exp(-15*(x-7/2)^2)+c5*0;
```

```
end
```

2.

```
>> fonction_1(-1)
```

```
ans =
```

```
0.0230
```

```
>> fonction_1(3)
```

```
ans =
```

```
0.0235
```

3.

```
>> x=-2:0.01:5
```

4. Pour représenter la courbe de la fonction $x \mapsto g(x)$ sur l'intervalle d'étude, il faut donner au M-file qui définit cette fonction la possibilité de calculer l'image d'un tableau. Notons à cet effet que les tests logiques définissant les variables logiques $c1$ à $c5$ s'appliquent sans aucun problème à des tableaux. D'où l'adaptation suivante du M-file fonction_1.m

```
function [gdex]=fonction_1(x)
```

```
c1=x>=-2 & x<-1;
```

```
c2=x>=-1 & x<-1/2;
```

```
c3=x>=-1/2 & x<3;
```

```
c4=x>=3 & x<4;
```

```
c5=x>=4 & x<=5;
```

```
gdex=c1.*0+c2.*(1./(exp(-15*(x+3/4))+1))+c3.*1+c4.*exp(-15*(x-7/2).^2)+c5.*0;
```

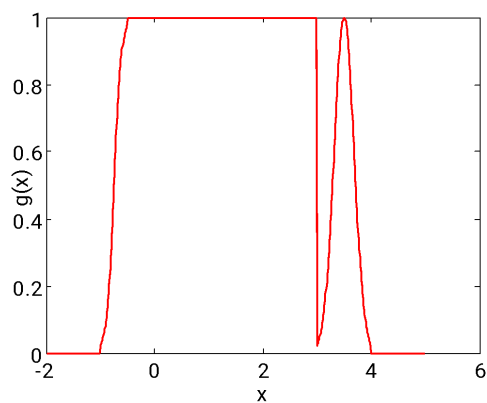
```
end
```

Il suffit maintenant de créer un script qui appelle la fonction fonction_1 c'est-à-dire g, en permettant de tracer son graphe à l'aide des antécédents regroupés dans le tableau x.

```
clear all;
```

```
clc;  
  
format long;  
  
x=-2:0.01:5;  
  
y=fonction_1(x);  
  
plot(x,y,'r-')
```

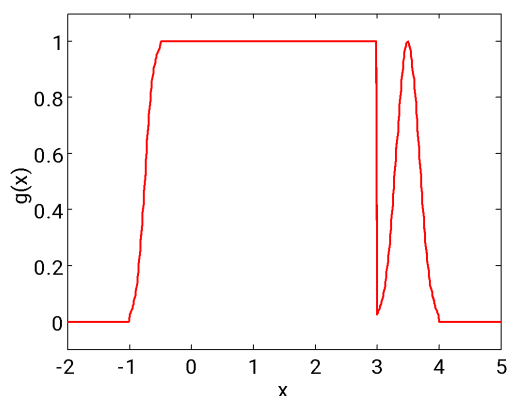
On obtient alors le graphe suivant:



On peut noter ici un problème d'échelle sur l'axe des y qui empêche de voir clairement la courbe de la fonction g. On peut y remédier en modifiant l'échelle à l'aide de la commande *axis*. Il suffit alors de rajouter en fin du script la ligne suivante

```
axis([-2 5 -0.1 1.1]);
```

La figure devient alors



Exercice 2

En utilisant la commande *subplot*, tracer sur l'intervalle $[0, 2\pi]$ et sur la même figure contenant quatre pavés les quatre fonctions suivantes:

$$f_1(x) = \sin(x) ; f_2(x) = \sin(2x) ; f_3(x) = \sin(3x) \text{ et } f_4(x) = \sin(4x)$$

Corrigé de l'exercice 2

On a ici quatre courbes différentes qu'on veut tracer sur quatre pavés (pour par exemple mieux les comparer). Ceci se fait très simplement à l'aide de la commande *subplot* de *Matlab*.

On peut proposer le script suivant pour tracer ces quatre courbes sur la même figure.

```
clear all;

close all;

clc;

format long;

x=linspace(0,2*pi,100);

y1=sin(x);

y2=sin(2*x);

y3=sin(3*x);

y4=sin(4*x);

subplot(2,2,1);

plot(x,y1,'r-');

axis([0 2*pi -1 1]);

subplot(2,2,2);

plot(x,y2,'r-');

axis([0 2*pi -1 1]);

subplot(2,2,3);

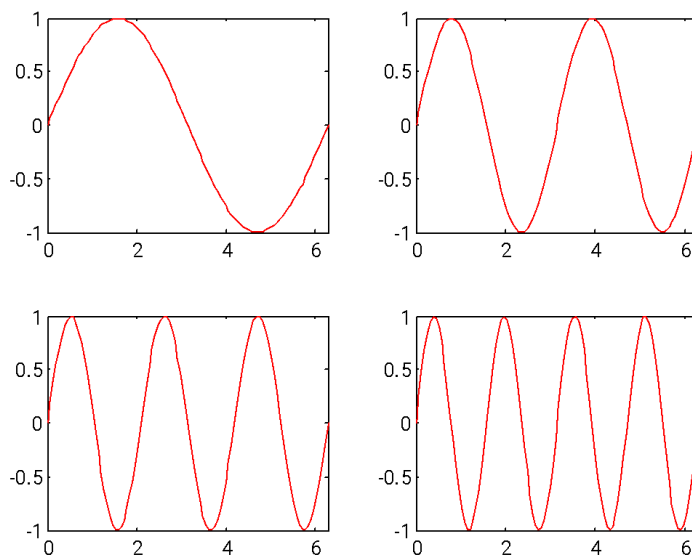
plot(x,y3,'r-');

axis([0 2*pi -1 1]);

subplot(2,2,4);

plot(x,y4,'r-');
```

L'exécution du script permet d'obtenir le graphe en pavé suivant.



Exercice 3

Tracer sur le domaine plan $[-2,2] \times [-3,3]$ la surface répondant à l'équation cartésienne:

$$z = e^{-x^2 - y^2}$$

Corrigé de l'exercice 3

Pour tracer la surface $(x,y) \mapsto z = e^{-x^2 - y^2}$, il suffit de créer deux tableaux: l'un des abscisses x_i sur le domaine $[-2,2]$, l'autre des ordonnées y_j sur le domaine $[-3,3]$ et de calculer ensuite la cote z_{ij} correspondant à chaque pair (x_i, y_j) avant de faire la représentation graphique de la surface par la commande *surf* ou *mesh*.

L'ensemble de ces opérations sont regroupées dans le script suivant :

```
clear all;
close all;
clc;
```

```
N=60;

x=(linspace(-2,2,N))'*ones(1,N);

y=ones(N,1)*linspace(-3,3,N);

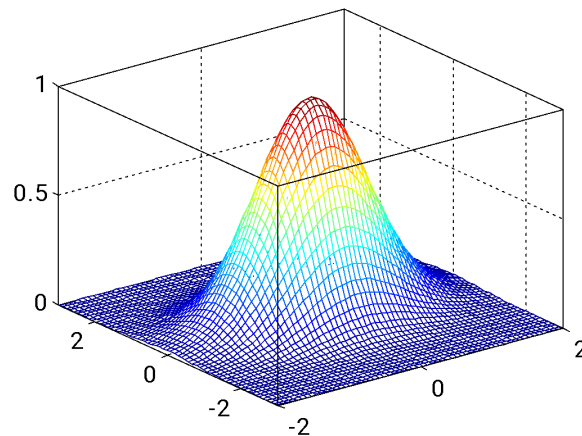
z=exp(-x.^2-y.^2);

surf(x,y,z)

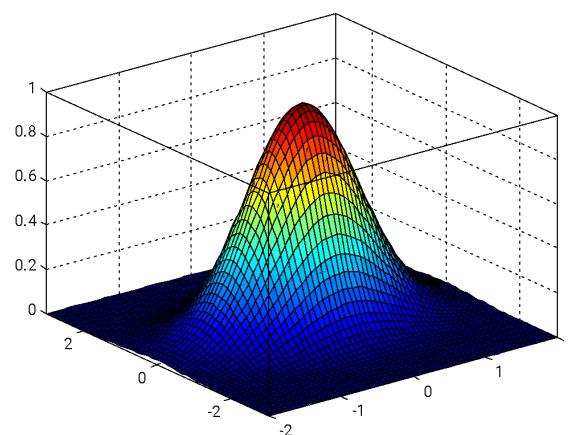
axis([-2 2 -3 3 0 1]);

box;
```

On obtient alors le graphe suivant :



En substituant la commande *surf* à la commande *mesh*, le résultat devient



Noter bien ici l'astuce qui permet de transformer le tableau des abscisses en une matrice contenant toutes les abscisses des points qui discrétisent le domaine $[-2,2] \times [-3,3]$. Cette matrice admet des

ansformation permet de récupérer une matrice des ordonnées avec cette fois-ci les colonnes qui sont identiques. Le calcul de z peut alors être effectué en bloc à l'aide de l'expression $\exp(-x.^2-y.^2)$ constante par ligne de la matrice des x et constante par colonne de la matrice des y . On verra dans la section 7, qu'il y a possibilité de générer les deux matrices x et y à l'aide de la structure boucle *for*. Ici, on s'est débrouillé, conformément à l'esprit de *Matlab* en évitant les boucles, pour générer les tableaux des points qui parcourent la surface $(x,y) \Rightarrow z=e^{-x^2-y^2}$. Une règle d'or dit qu'il faut toujours chercher à utiliser peu de moyens pour faire le maximum!