



## Module: Calcul Formel sous Matlab

# Chapitre 3: Graphique et Visualisation des données

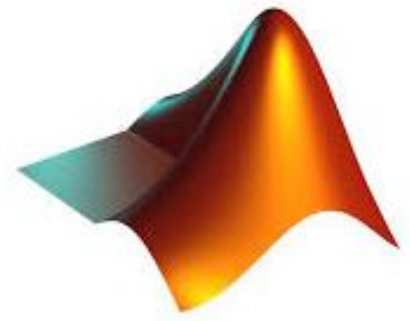
*Pr : A. SLIMANI*

2021/2022

# Plan

- 1- Introduction**
- 2- Représentation graphique 2D**
- 3- Représentation graphique 3D**
- 4- Exercices d'application**

# 1. Introduction



Le but de ce cours est de vous donner les outils permettant d'utiliser Matlab pour rédiger des documents scientifiques. Une compétence importante est donc la capacité à réaliser des graphes de qualité présentant clairement vos données.

MATLAB fournit une grande variété de techniques pour l'affichage des données sous forme graphique.

Le processus de visualisation des données implique une série d'opérations. Ce chapitre fournit une vue générale sur les possibilités que fournit Matlab pour la représentation graphique.

## 2. Représentation graphique 2D

### □ Fonction plot

C'est la fonction la plus utilisée dans la représentation graphique 2D.

- La commande ***plot(x,y)*** permet de tracer le fonction ***y*** en fonction de la variable ***x***.

En toute rigueur, il faut définir auparavant les vecteurs ***x*** et ***y*** qui doivent être de **même dimension**.

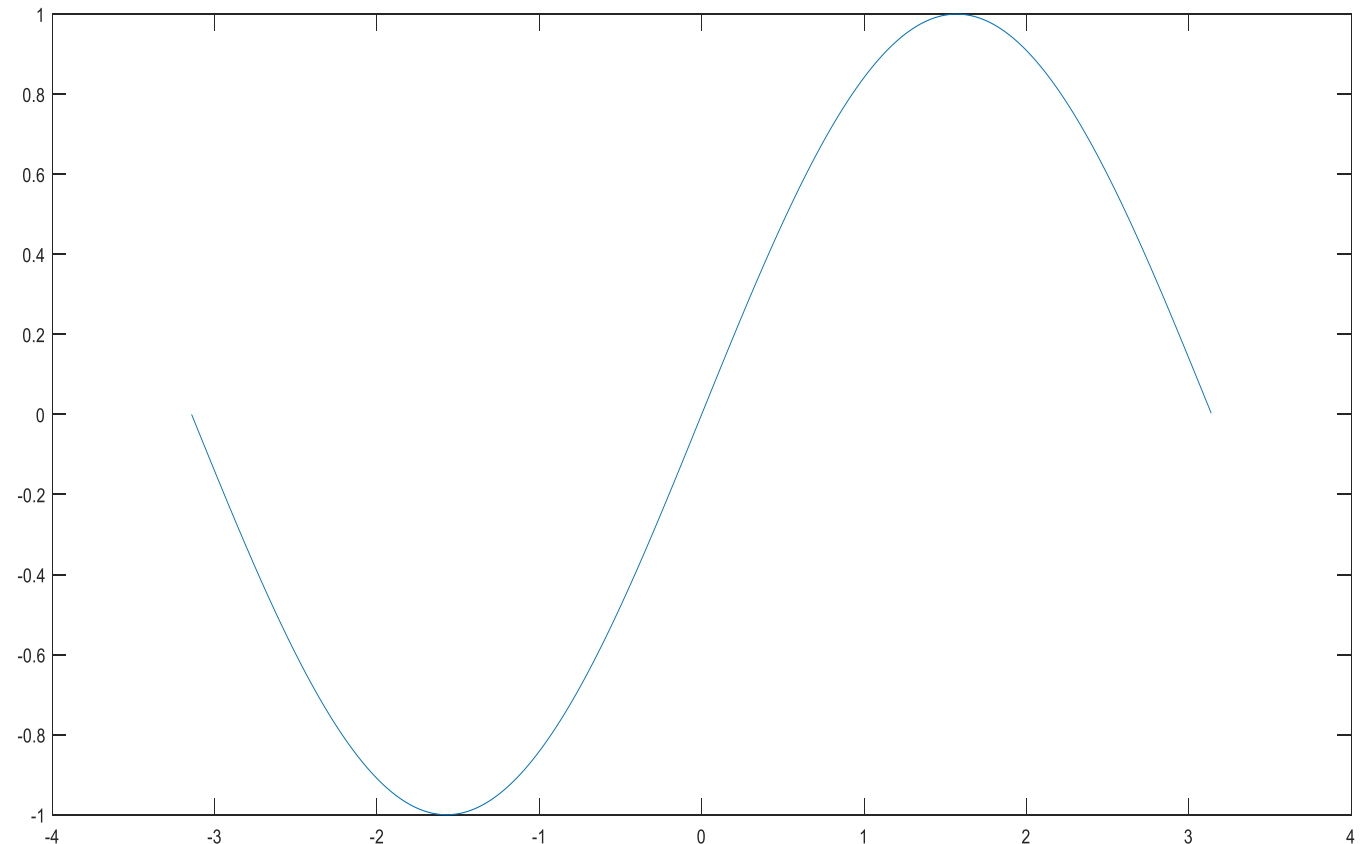
## Exemple d'application :

Le programme ci-dessous permet de représenter le graphe 2D de la fonction  $y = \sin(t)$  sur l'intervalle  $[-\pi, \pi]$  avec un pas de  $\Delta t = 0.1$ .

### Programme

```
t=-pi:0.1:pi;  
y=sin(t);  
plot(t,y)
```

### Graphe



## ❑ Annotation d'une figure

La description textuelle d'une courbe est très important dans une figure, car elle aide l'utilisateur de comprendre la signification de chaque axe et l'intérêt de la représentation concernée.

### Titre à la figure :

Pour donner un titre à la figure, on utilise la commande **title** dans le programme, comme ceci:

```
title ( ' titre de la figure ' )
```

### Titre à la courbe :

Pour donner un titre à la courbe dans la figure, on utilise la commande **legend** dans le programme, comme ceci:

```
legend ( ' titre de la figure ' )
```

## Titre de l'axe des abscisses :

Pour donner une légende à l'axe des abscisses, on utilise la commande **xlabel** dans le programme, comme ceci:

```
xlabel ( ' axe des abscisses ' )
```

## Titre de l'axe des ordonnées :

Pour donner une légende à l'axe des abscisses, on utilise la commande **ylabel** dans le programme, comme ceci:

```
ylabel ( ' axe des ordonnées ' )
```

## Exemple d'application :

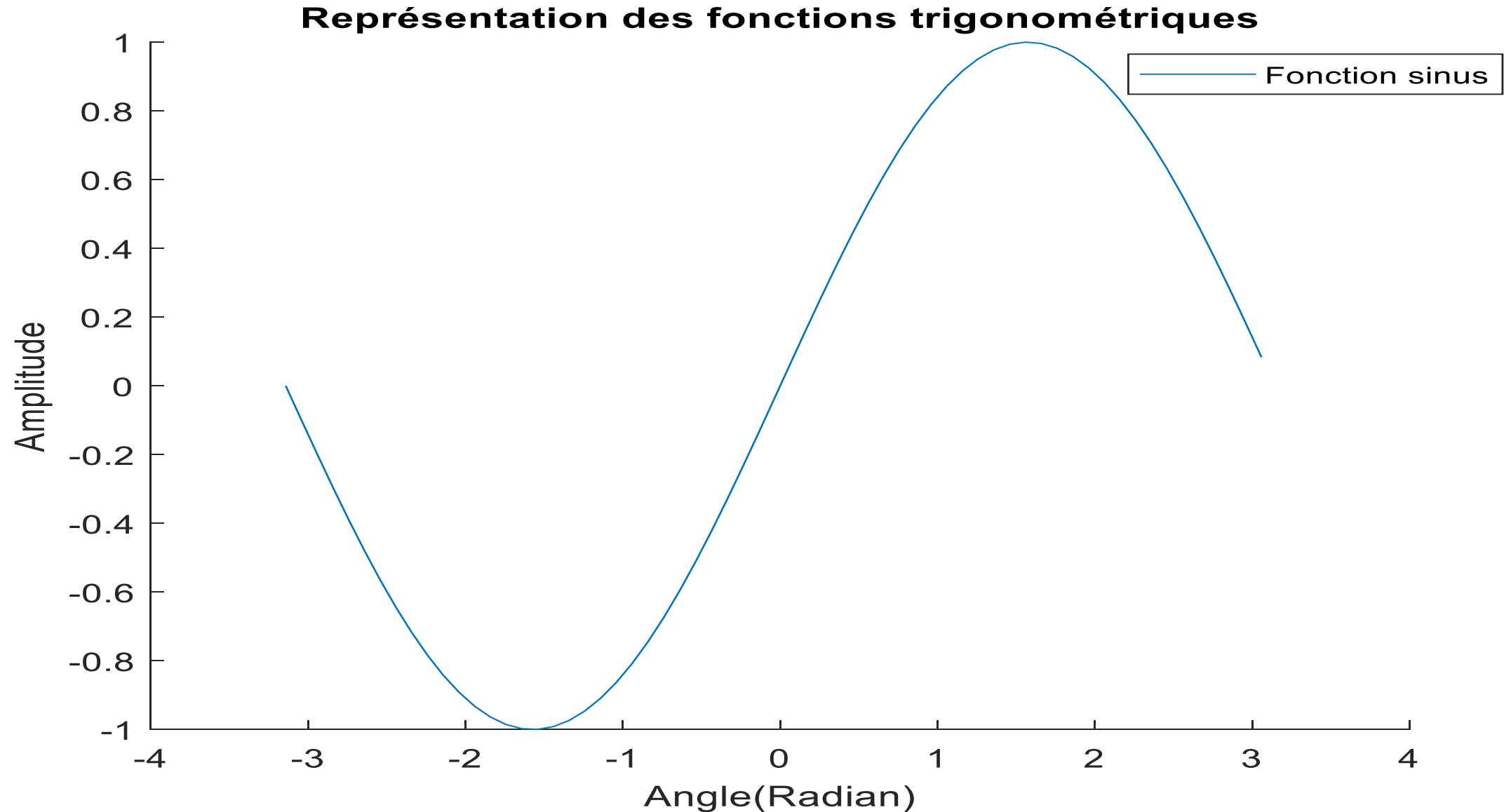
On peut travailler sur le même graphe de la fonction précédente  $y = \sin(t)$  :

### Programme

```
t=-pi:0.1:pi;  
y=sin(t);  
plot(t,y)  
title('Représentation des fonctions trigonométriques')  
xlabel('Angle (Radian)')  
ylabel('Amplitude')  
legend('Fonction sinus')
```



# Graphe obtenu:



## Remarque: Fonction **fplot**

- La fonction **fplot(g)** permet de tracer la courbe de la fonction **g(x)** sur l'intervalle des abscisses **x** par défaut **[-5 5]**.

- On peut tracer sur un intervalle spécifié **[x<sub>min</sub> x<sub>max</sub>]** en l'indiquant au deuxième argument de la fonction:

**fplot(g, [x<sub>min</sub> x<sub>max</sub>])**

- On peut tracer les courbes de deux fonctions **g(x)** et **k(x)** sur l'intervalle par défaut **[-5 5]**.

**fplot(g, k)**

## Exemple:

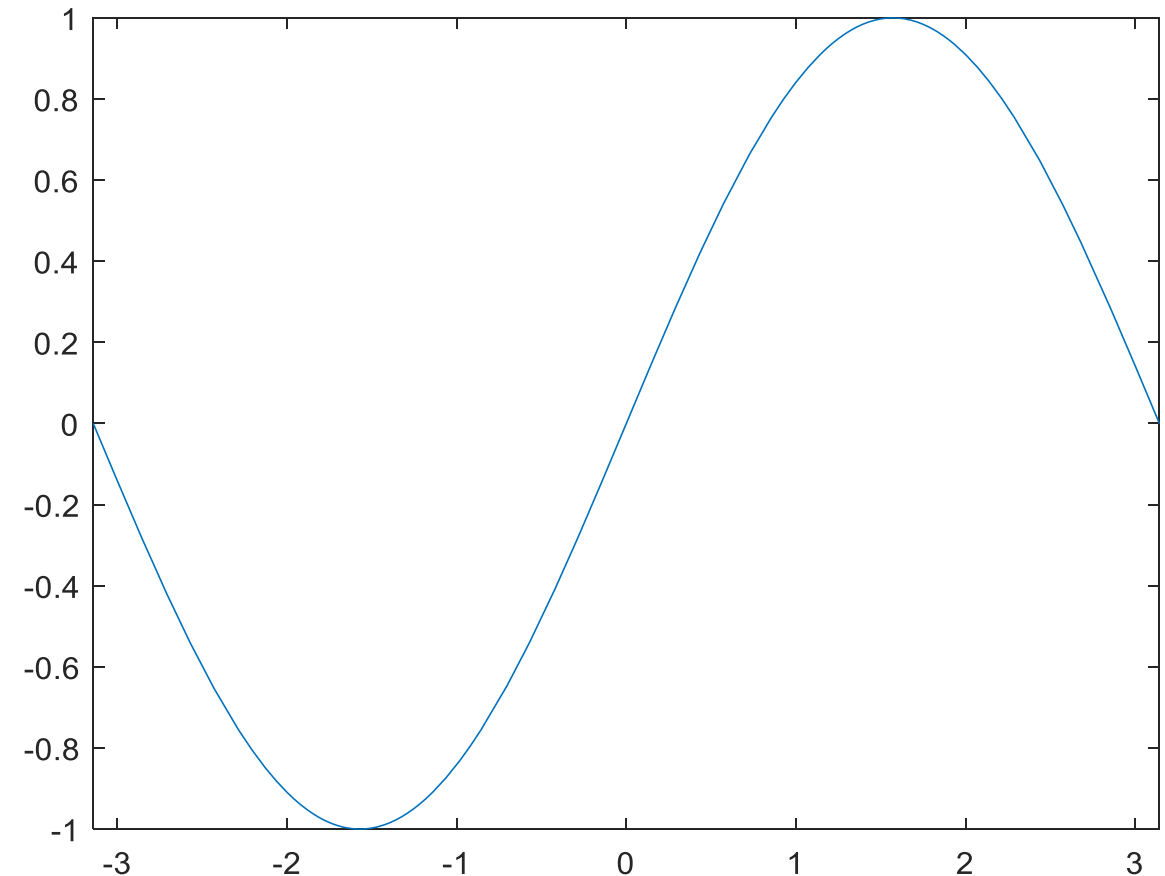
- On peut tracer la courbe de la fonction ***Sinus*** avec différentes méthodes sur  $[-\pi \pi]$ :

Avec `fplot('sin', [-pi pi])`

Ou avec `f=@sin;  
fplot(f, [-pi pi])`

Ou avec `f=@(t) sin(t);  
fplot(f, [-pi pi])`

**Graphe:**



## ❑ Style de courbe

Dans certains cas, on est amené à tracer plusieurs courbes dans une même figure, ce qui nous oblige à représenter chaque courbe avec un style spécial pour distinguer entre les différentes courbes.

Dans Matlab, il est possible de jouer sur l'apparence de chaque courbe, en modifiant sa couleur et sa forme (point, tiret, ...)

Pour cela, on peut ajouter un nouvel argument de type chaîne de caractère à la fonction plot (appelé **marqueur**):

```
plot(variable, fonction, 'marqueur')
```

Le contenu du **marqueur** est une combinaison d'un ensemble de caractères spéciaux rassemblés dans le tableau ci-dessous:

Couleur	
Jaune	y
magenta	m
cyan	c
rouge	r
vert	g
bleu	b
blanc	w
noire	k

Motif	
point	.
cercle	o
x	x
plus	+
étoile	*
carré	s
diamant	d
triangle	^

Styles de lignes	
Ligne solide	-
pointillé	:
Trait d'union point	-.
coupée	--

`plot(t,y,'g o -')`

Couleur      Motif      Style de la ligne

# Taille des courbes:

Pour modifier la taille des courbes, il suffit de spécifier la taille dans la fonction plot à l'aide de l'argument **linewidth** :

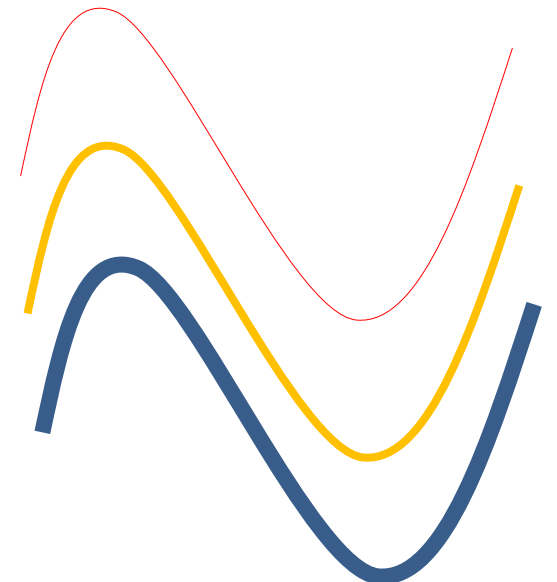
```
plot(variable, fonction, 'marqueur', 'linewidth', taille)
```

## Exemple:

```
plot(t,y, 'r', 'linewidth', 1)
```

```
plot(t,y, 'y', 'linewidth', 2)
```

```
plot(t,y, 'b', 'linewidth', 3)
```



## Remarque:

1. Pour tracer sur la même figure deux courbes ou plus, il suffit d'utiliser la commande **hold on** une seule fois avant les fonctions **plot**, et quand vous voulez la désactiver, il suffit d'ajouter la commande **hold off**.

**Exemple:** Représentation des trois fonctions  $y(t)$ ,  $z(t)$  et  $w(t)$  sur le même graphe

### Syntaxe:

```
hold on
plot(t,y)
plot(t,z)
plot(t,w)
hold off
```

2. Pour mettre un quadrillage (une grille) dans la figure, il suffit d'utiliser la commande **grid on**, et pour la désactiver, il suffit d'ajouter la commande **grid off**.

## Exercice d'application :

Tracer sur la même figure, les courbes des fonctions  $y = 3 * \sin(t)$  et  $z = 2 * \cos(t)$  sur l'intervalle  $[-\pi, \pi]$  avec un pas  $\Delta t = 0.1$ , sachant que:

- Chaque courbe a une **couleur** et un **motif** différent,
- La courbe de la fonction ( $y$ ) a le titre: **Fonction sinus**,
- La courbe de la fonction ( $z$ ) a le titre: **Fonction cosinus**,
- Le titre de la figure: **Représentation des fonctions trigonométriques**,
- Donner la légende à l'axe des abscisses: **Angle (Radian)**,
- Donner la légende à l'axe des ordonnées: **Amplitude**,
- Mettre un quadrillage à la figure.

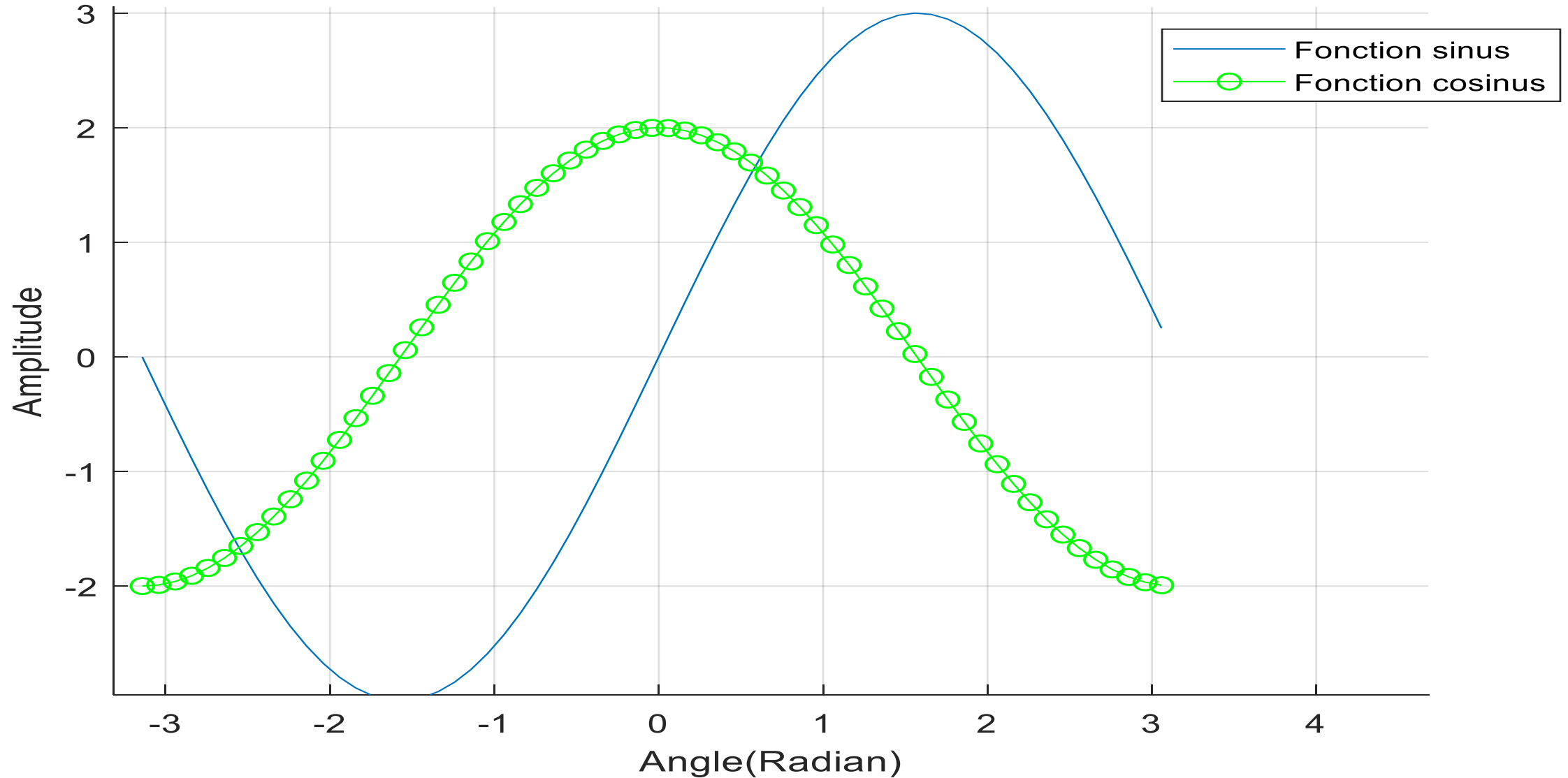


## Solution:

```
t=-pi:0.1:pi;  
y=3*sin(t);  
z=2*cos(t);  
hold on  
plot(t,y)  
plot(t,z,'go-')  
grid on  
legend('Fonction sinus','Fonction cosinus')  
title('Représentation des fonctions trigonométriques')  
xlabel('Angle (Radian)')  
ylabel('Amplitude')
```

# Graphe:

Représentation des fonctions trigonométriques

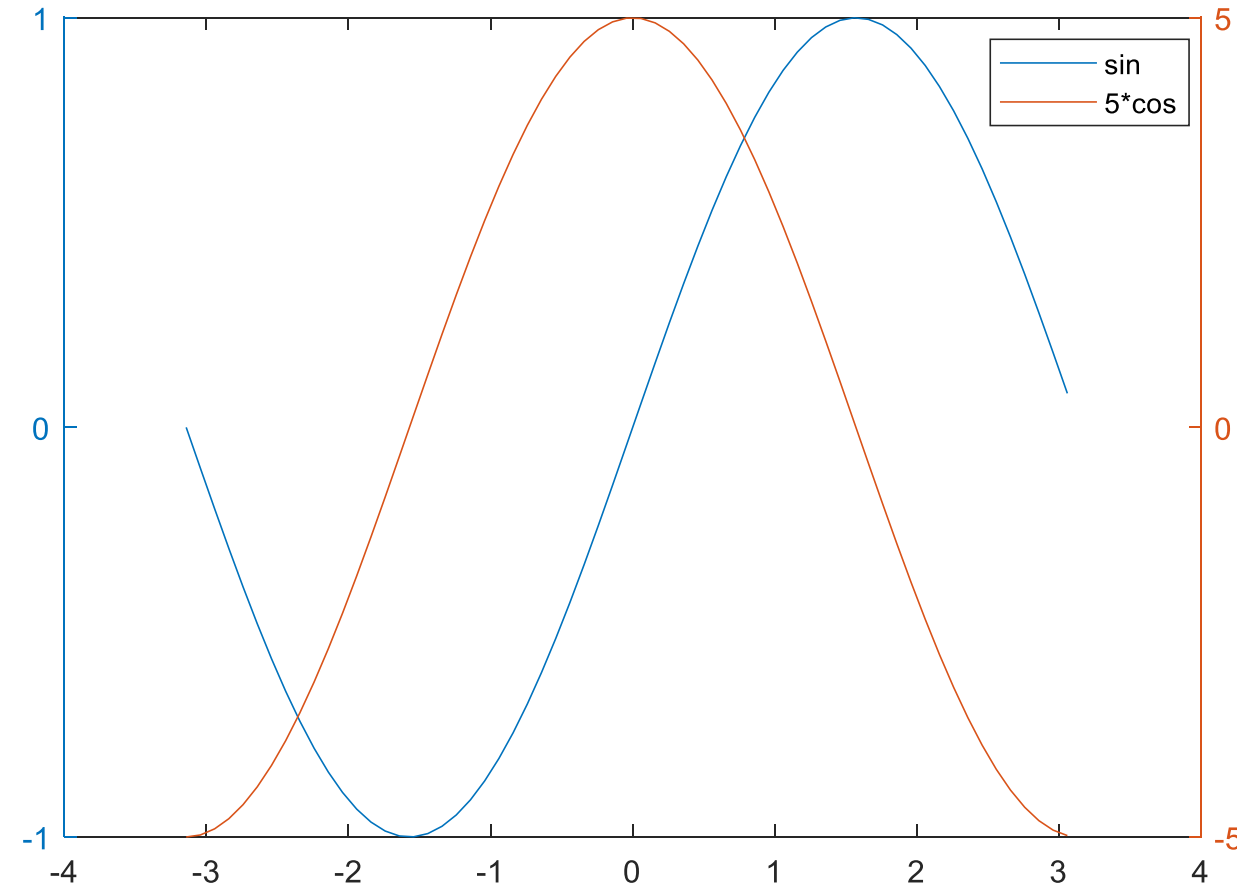


## □ Fonction plotyy

C'est une fonction qui nous permet de tracer deux courbes de données avec des axes ordonnées indépendants et un axe  $x$  commun.

### Exemple:

```
t=-pi:0.1:pi;  
y=sin(t);  
z=5*cos(t);  
plotyy(t,y,t,z)  
legend('sin', '5*cos')
```



## Exercice d'application :

Tracer la fonction  $\sin(t)$  et son module sur la même figure, sachant que chaque fonction possède son axe des ordonnées propre.

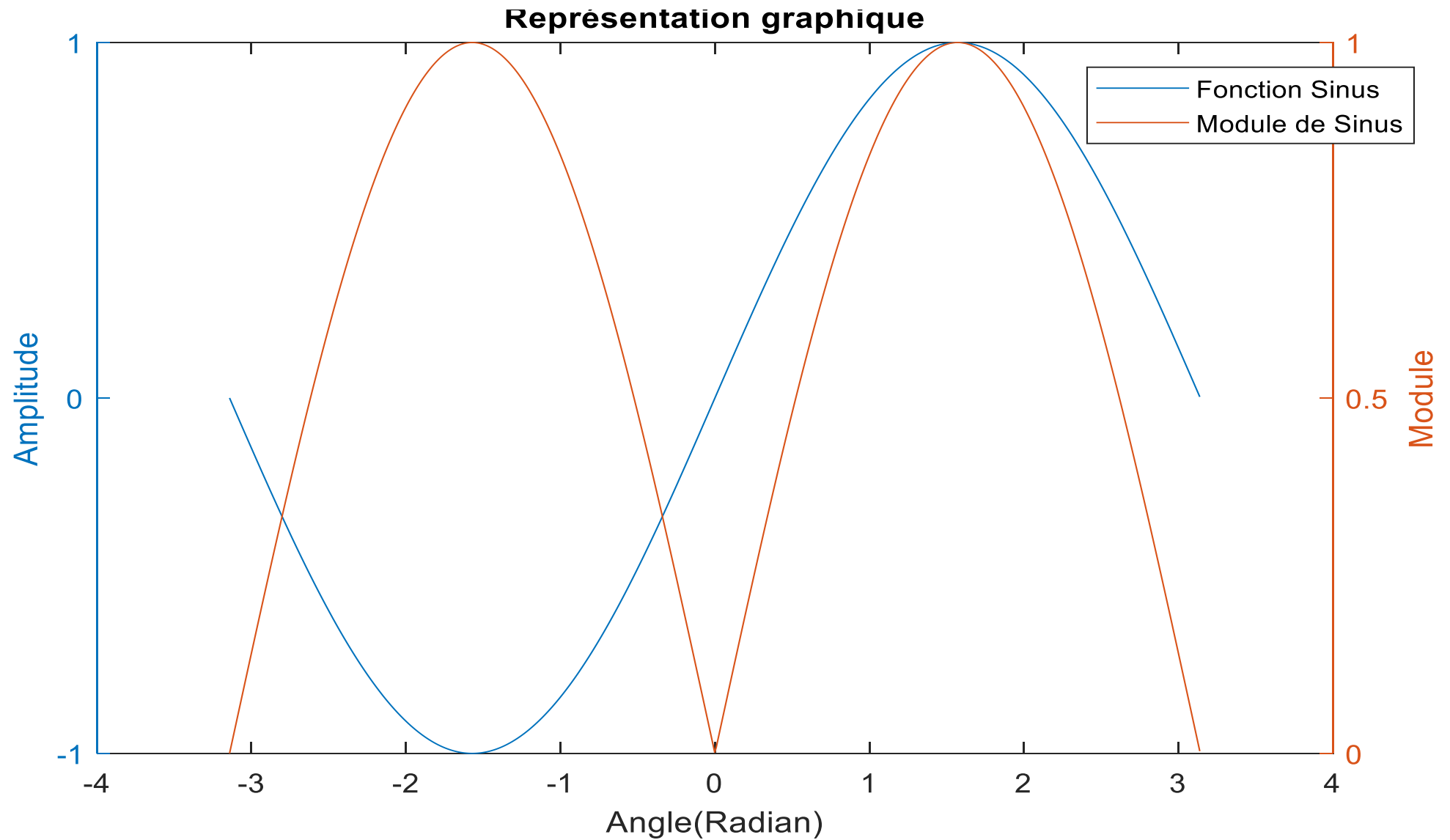
### Remarque:

- Donner un titre pour chaque fonction: `Fonction Sinus`, `Module de Sinus`
- Donner le titre '`Représentation graphique`' à la figure.
- Donner le titre '`Angle (Radian)`' à l'axe des abscisses.
- Donner le titre '`Amplitude`' pour l'axe des ordonnées de la fonction  $\sin(t)$  et le titre '`Module`' pour l'axe des ordonnées de son module.

## Solution:

```
t=-pi:0.01:pi;  
y=sin(t);  
ax=plotyy(t,y,t,abs(y))  
legend('Fonction Sinus','Module de Sinus')  
title('Représentation graphique')  
xlabel('Angle(Radian)')  
ylabel(ax(1),'Amplitude')  
ylabel(ax(2),'Module')
```

# Graphe:



## ❑ Plusieurs figures sur la même fenêtre

Si vous voulez tracer plusieurs courbes, dans la même fenêtre, mais dans des figures séparées, on utilise la fonction **subplot**, de la manière suivante: **subplot (m , n , k)**

- La fonction **subplot(m,n,k)** subdivise la fenêtre graphique en  $m \times n$  ( $m$  lignes et  $n$  colonnes) figures séparées. Le  $k^{\text{ème}}$  figure est choisie de telle sorte que  $k = 1$  correspond à  $(m = 1, n = 1)$ ,  $k = 2 \rightarrow (m=1, n=2)$ , ....

<b>subplot (2 , 2 , 1)</b>	<b>subplot (2 , 2 , 2)</b>
<b>subplot (2 , 2 , 3)</b>	<b>subplot (2 , 2 , 4)</b>

## Exercice d'application :

On peut travailler sur le même exemple de l'exercice précédent:

$$y = 3 * \sin(t) \text{ et } z = 2 * \cos(t) ,$$

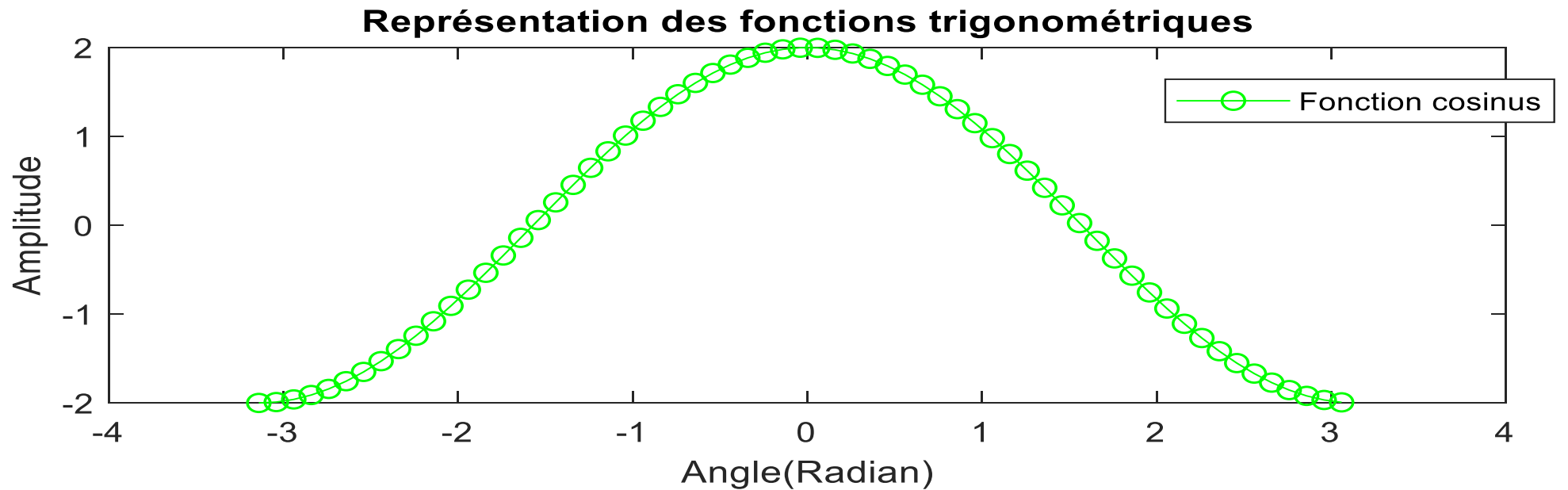
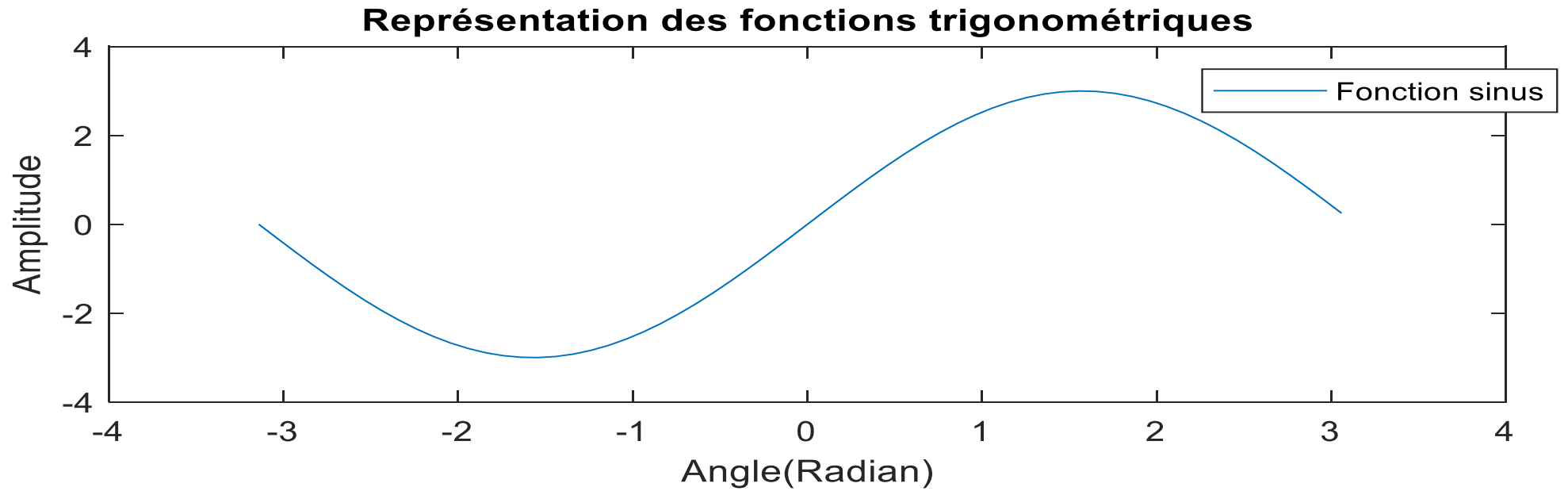
- Tracer les deux fonctions sur la même fenêtre mais sur des figures différentes en gardant les mêmes titres (axes, légendes,...).



# Solution:

```
t=-pi:0.1:pi;
y=3*sin(t);
z=2*cos(t);
subplot(2,1,1)      % la première figure
plot(t,y)
legend('Fonction sinus')
title('Représentation des fonctions trigonométriques')
xlabel('Angle(Radian)')
ylabel('Amplitude')
subplot(2,1,2)      % la deuxième figure
plot(t,z,'go-')
legend('Fonction cosinus')
title('Représentation des fonctions trigonométriques')
xlabel('Angle(Radian)')
ylabel('Amplitude')
```

# Graphe:



## ❑ Plusieurs figures sur des fenêtres séparées

On peut tracer les courbes sur des fenêtres graphiques séparées à l'aide de la commande **figure**. Avant chaque plot, nous mettons une commande **figure** avec le numéro de la figure.

### Exemple:

```
figure(1), plot(t,y), title('Fonction y');  
figure(2), plot(t,z), title('Fonction z');  
figure(3), plot(t,w), title('Fonction w');  
      :           :           :  
figure(k), plot(t,u), title('Fonction u');
```

## Exercice d'application :

On peut travailler sur le même exemple de l'exercice précédent :

$$y = 3 * \sin(t) \quad \text{et} \quad z = 2 * \cos(t) ,$$

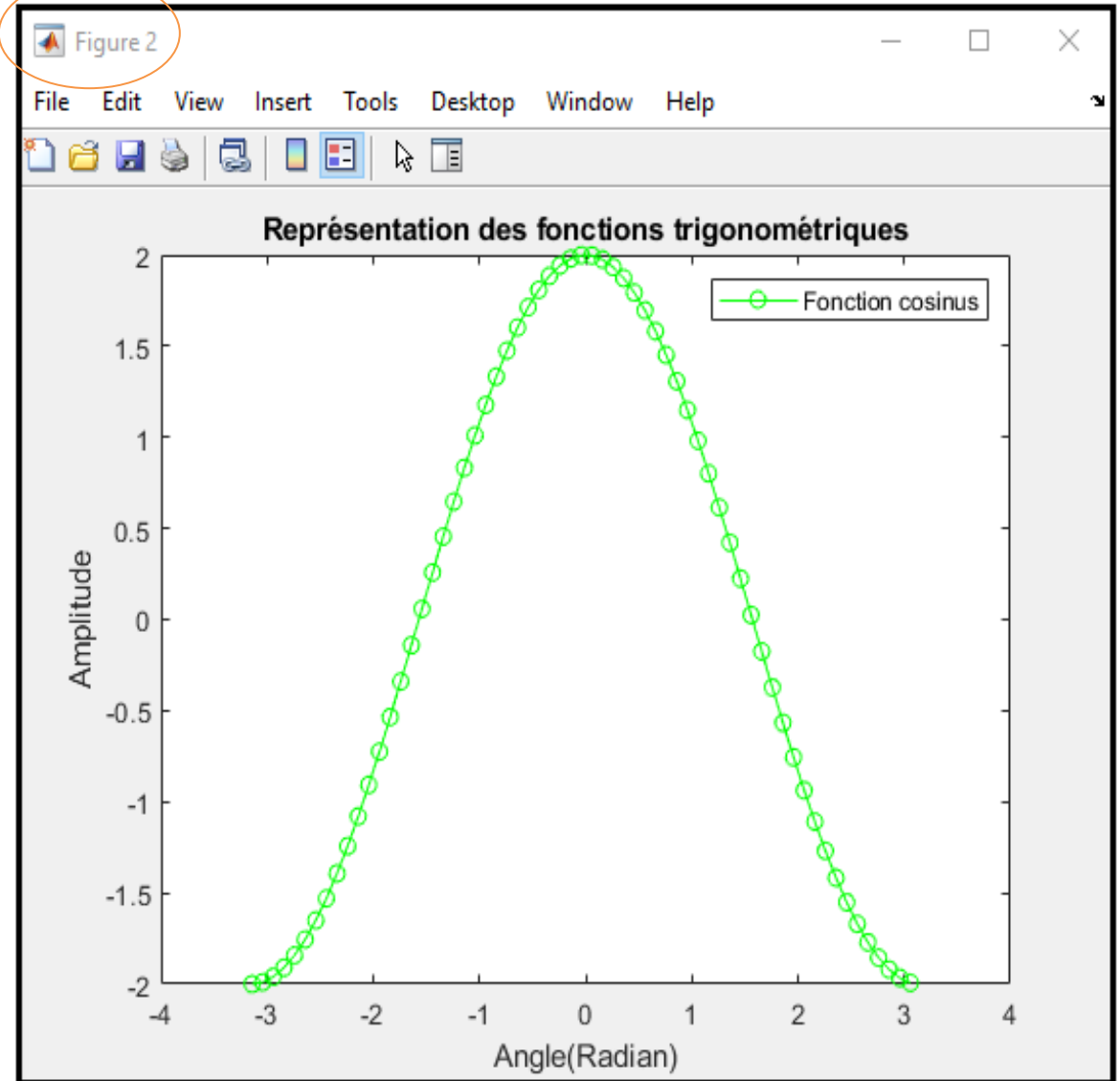
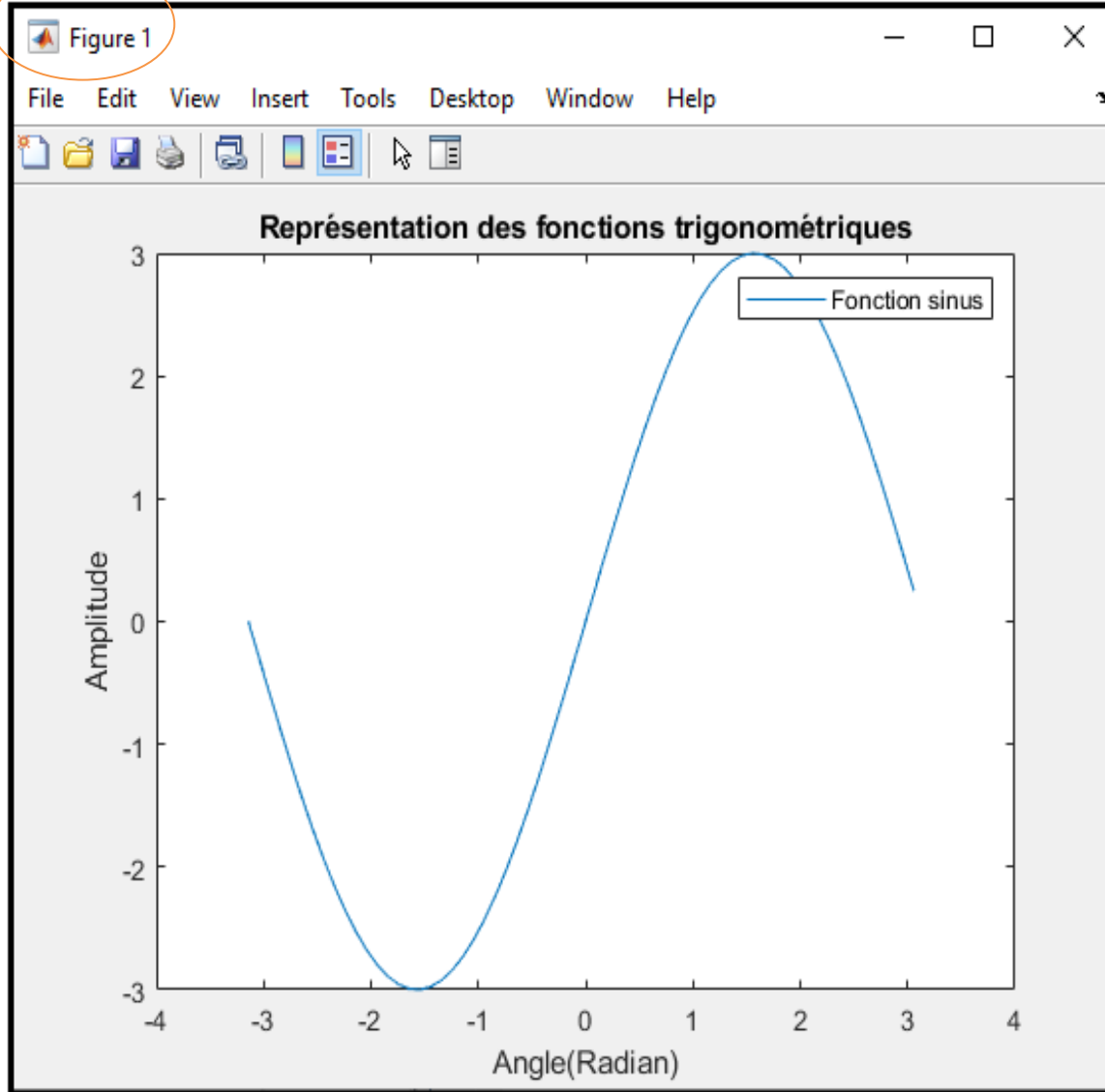
- Tracer les deux fonctions sur des fenêtres séparées en gardant les mêmes titres (axes, légendes,...).

## Solution:

```
t=-pi:0.1:pi;  
y=3*sin(t);  
z=2*cos(t);  
figure (1), plot(t,y);    % la première fenêtre  
legend('Fonction sinus')  
title('Représentation des fonctions trigonométriques')  
xlabel('Angle(Radian)'), ylabel('Amplitude')  
figure (2), plot(t,z,'go-');    % la deuxième fenêtre  
legend('Fonction cosinus')  
title('Représentation des fonctions trigonométriques')  
xlabel('Angle(Radian)'), ylabel('Amplitude')
```

# Graphe:

Numéros des figures



## Remarque:

Parfois, pour la bonne interprétation physique des résultats, il est préférable de représenter les courbes sur les mêmes échelles (axe des abscisses et des ordonnées). Ceci peut se réaliser en utilisant la commande **axis** (après la commande **plot**) dont la syntaxe est :

$$\textit{axis}([X_{min} \ X_{max} \ Y_{min} \ Y_{max}])$$

où,

- $X_{min}$  ,  $X_{max}$  : les graduations minimales et maximales de l'axe des abscisses  $x$ ,
- $Y_{min}$  ,  $Y_{max}$  : les graduations minimales et maximales de l'axe des ordonnées  $y$ .

## Exemple d'application :

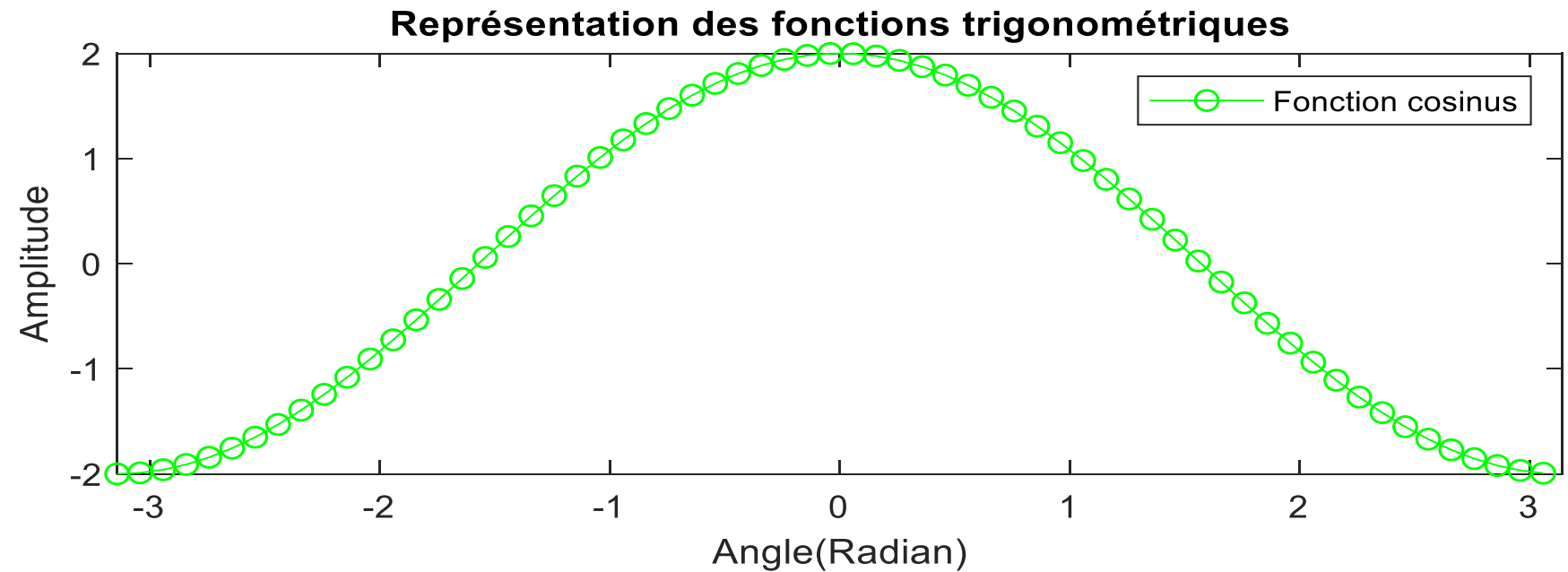
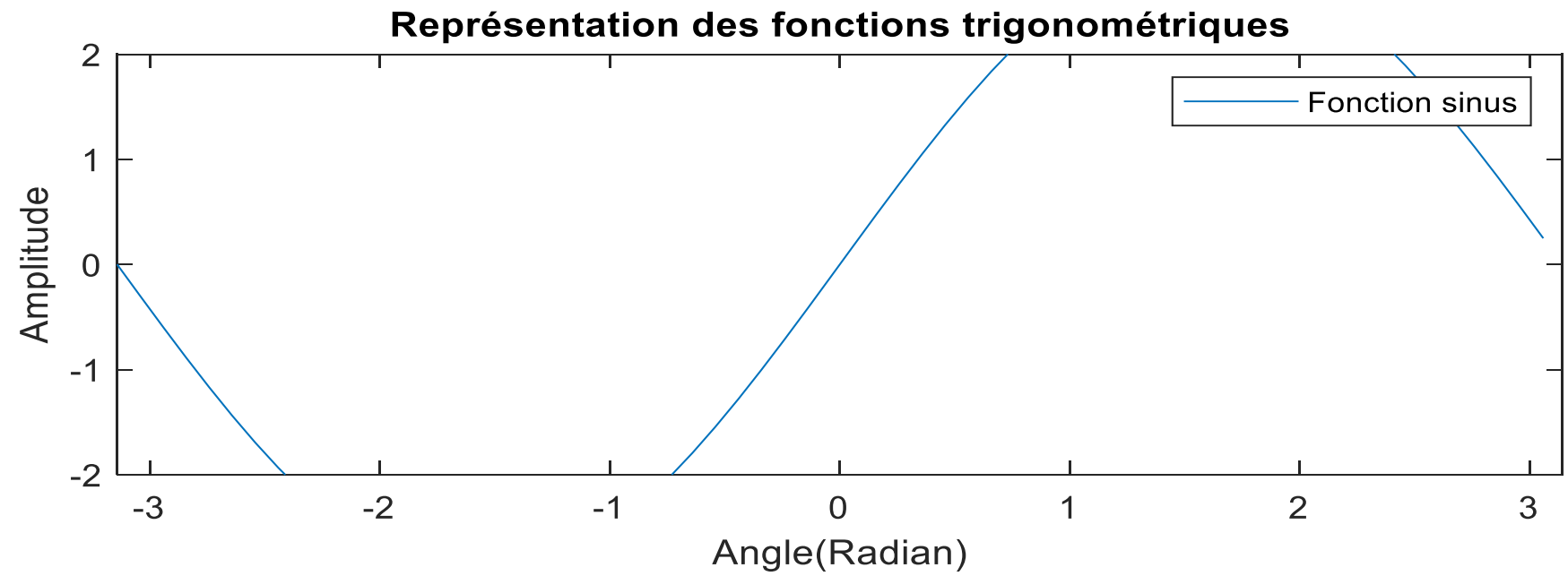
On peut travailler sur le même exemple de l'exercice précédent  $y = 3 * \sin(t)$  et  $z = 2 * \cos(t)$  :

```
t=-pi:0.1:pi; y=3*sin(t); z=2*cos(t);  
subplot(2,1,1), plot(t,y),axis([-pi pi -2 2])  
legend('Fonction sinus')  
title('Représentation des fonctions trigonométriques')  
xlabel('Angle(Radian)'), ylabel('Amplitude')  
subplot(2,1,2), plot(t,z,'go-'),axis([-pi pi -2 2])  
legend('Fonction cosinus')  
title('Représentation des fonctions trigonométriques')  
xlabel('Angle(Radian)'), ylabel('Amplitude')
```



# Graphe:

```
axis([-pi pi -2 2])
```



## ❑ Autres fonctions de représentation 2D

### Fonction **polar** :

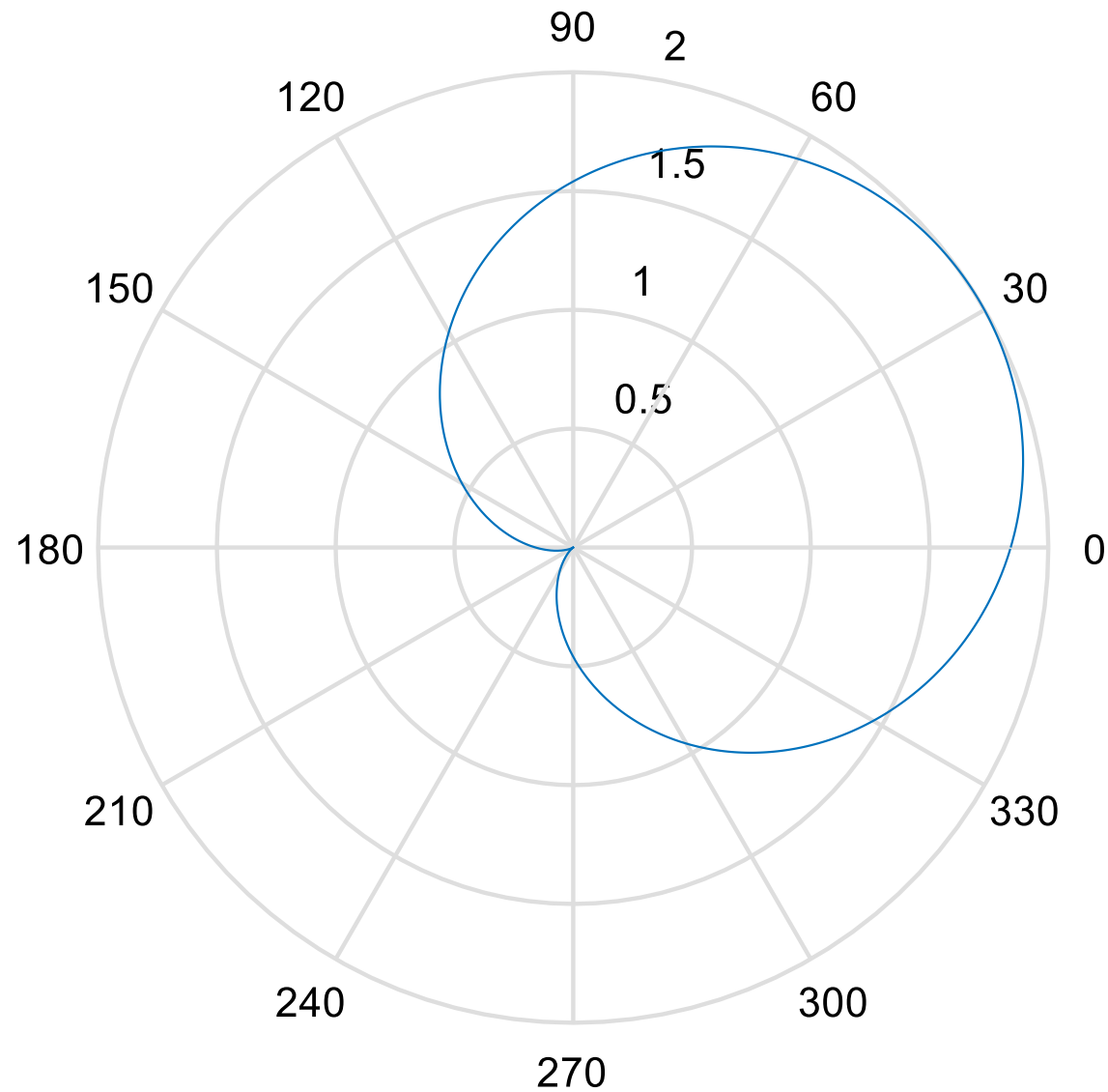
C'est une fonction qui nous permet de représenter des fonctions en coordonnées polaires. Ceci présente un grand intérêt pour les diagrammes de directivité.

### Exemple d'application:

Représentation de la fonction  $f(\theta) = 1 + \sin(\theta + 1)$  sur l'intervalle  $[0, 2\pi]$  avec un pas  $\Delta\theta = 0,01$ .

```
theta = 0:0.01:2*pi;  
f = 1+ sin(theta+1) ;  
polar(theta,f)
```

# Graphe:



Représentation polaire de la fonction:  $f = 1 + \sin(\theta + 1)$

## Exemple d'application:

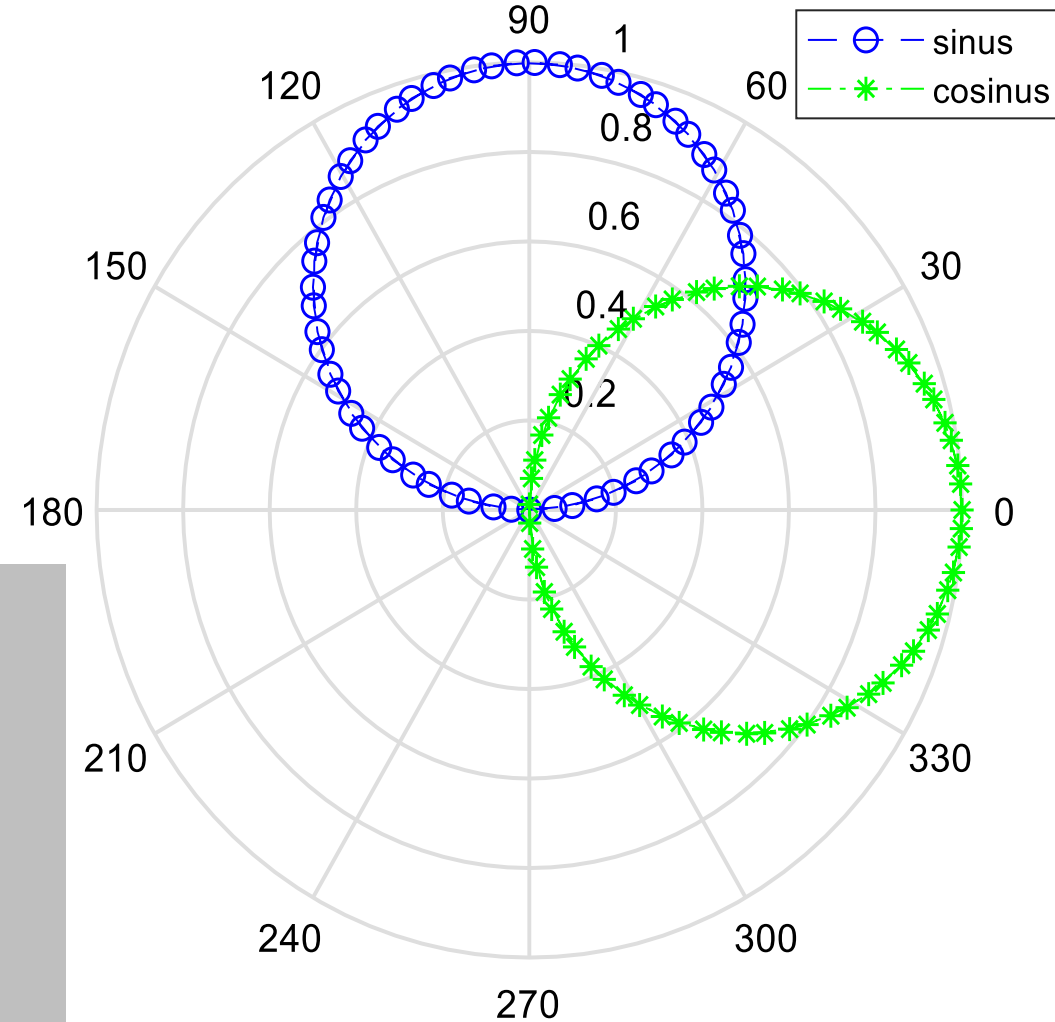
Donner la représentation polaire, sur la même figure des deux fonctions trigonométriques ***sin***( $\theta$ ) et ***cos***( $\theta$ ) sur l'intervalle  $[0, 2\pi]$  avec un pas  $\Delta\theta = 0,1$ .

### Remarque:

- Chaque courbe a une **couleur** et un **motif** différent,
- La courbe de la fonction *Sinus* a le titre: **Sinus**,
- La courbe de la fonction *cosinus* a le titre: **Cosinus**,
- Le titre de la figure: **Représentation polaire des fonctions sin & cos**,

# Solution:

## Représentation polaire des fonctions sin & cos



```
theta = 0:0.1:2*pi;  
f=sin(theta) ; g=cos(theta);  
polar(theta,f,'bo--')  
hold on  
polar(theta,g,'g*-.'  
title('Représentation polaire des  
fonctions sin & cos')  
legend('sinus','cosinus')
```

## Fonction **stem** :

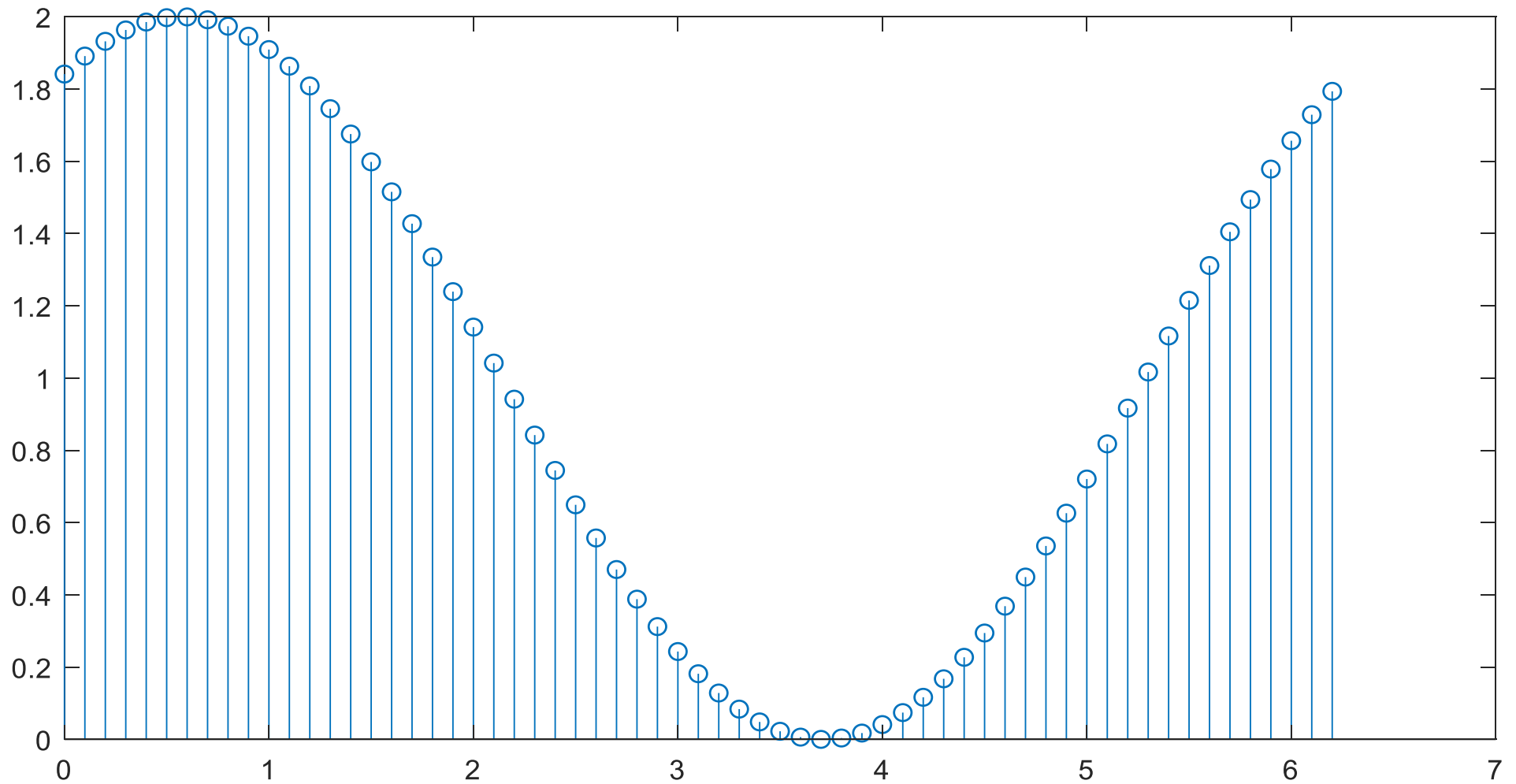
C'est une fonction qui nous permet de représenter (en coordonnées cartésiennes) les ordonnées comme des barres terminées par des cercles.

### Exemple d'application:

Représentation de la fonction précédente  $f(\theta) = 1 + \sin(\theta + 1)$  sur l'intervalle  $[0, 2\pi]$  avec un pas de  $\Delta\theta = 0,01$ .

```
theta = 0:0.01:2*pi;  
f = 1+ sin(theta+1) ;  
stem(theta,f)
```

# Graphe:



# Fonction **hist** :

C'est une fonction qui nous permet de représenter les résultats sous forme d'un histogramme.

***hist(y, N)*** présente un histogramme de  $N$  colonnes de la variable  $y$ . ***hist(y)*** sans argument ( $N$ ), prend par défaut  $N = 10$  colonnes.

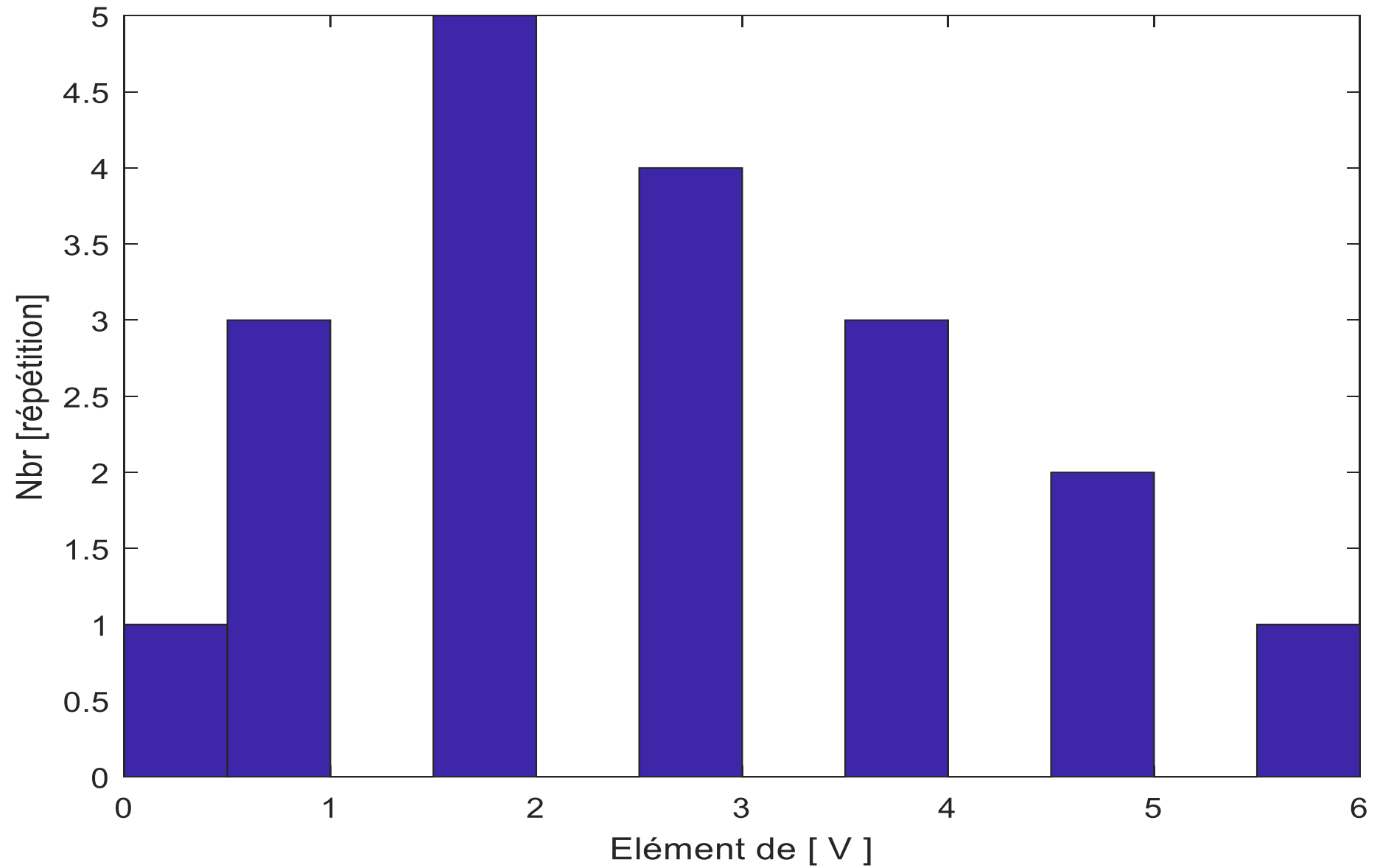
## Exemple d'application:

Représentation des éléments d'un vecteur  $V$  sous forme d'un histogramme.

```
v=[2 0 4 2 5 2 3 6 1 2 1 1 3 4 3 5 2 3 4];  
hist(v,12)  
xlabel('Élément de [ V ]')  
ylabel('Nbr [répétition]')
```



## Graphe:



# Fonctions **logarithmiques** :

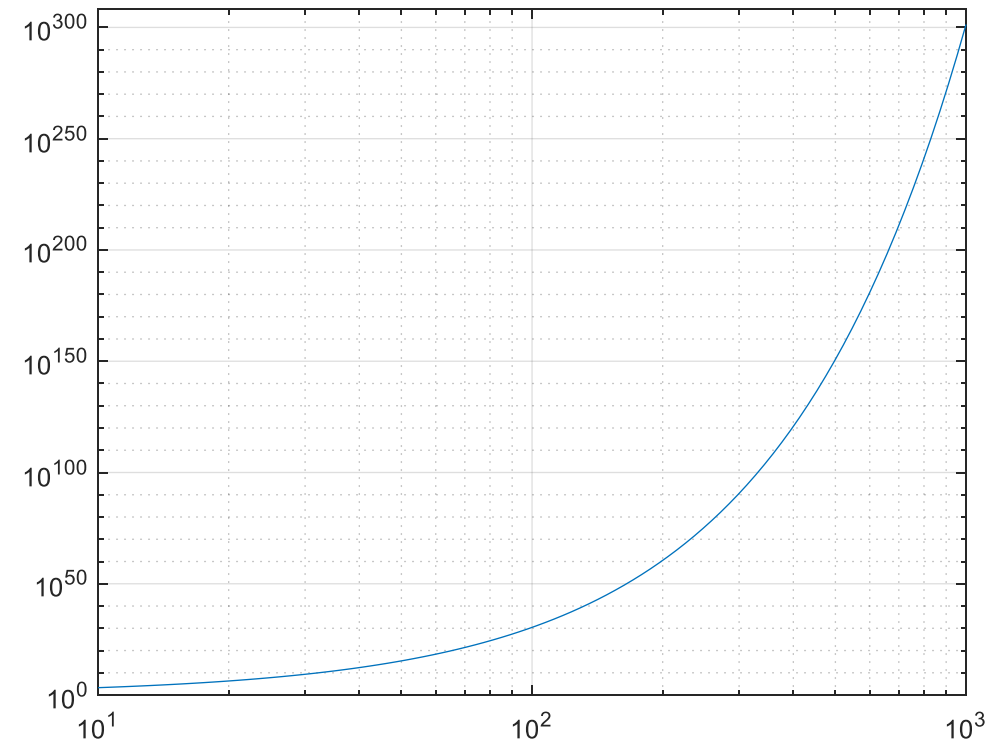
Parmi les fonctions graphiques 2D en Matlab, on trouve les fonctions logarithmiques :

- ❑ ***loglog(x,y)***: trace les coordonnées  $x$  et  $y$  en utilisant une échelle logarithmique de base 10 sur l'axe des  $x$  et des  $y$ .

## Exemple:

```
x = logspace(1, 3, 100);  
f = 2.^(x+1);  
loglog(x, f)  
grid on
```

Grappe



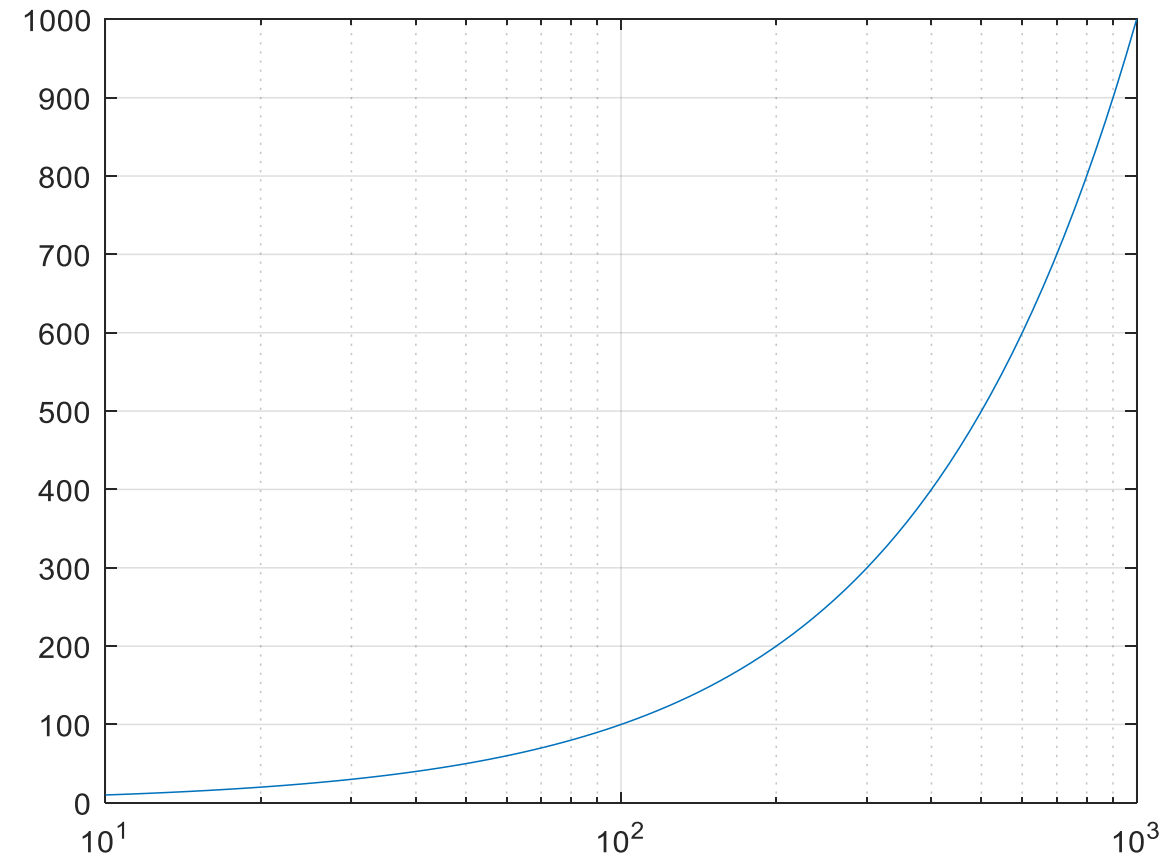
**Remarque:**  $y = \text{logspace}(a, b, n)$  génère  $n$  points entre  $10^a$  et  $10^b$ .

- ❑ ***semilogx(x,y)***: trace les coordonnées  $x$  et  $y$  en utilisant une échelle logarithmique de base 10 sur l'axe des  $x$  et une échelle linéaire sur l'axe des  $y$ .

## Exemple:

```
x = logspace(1,3,100);  
f = x ;  
semilogx(x,f)  
grid on
```

Graphe

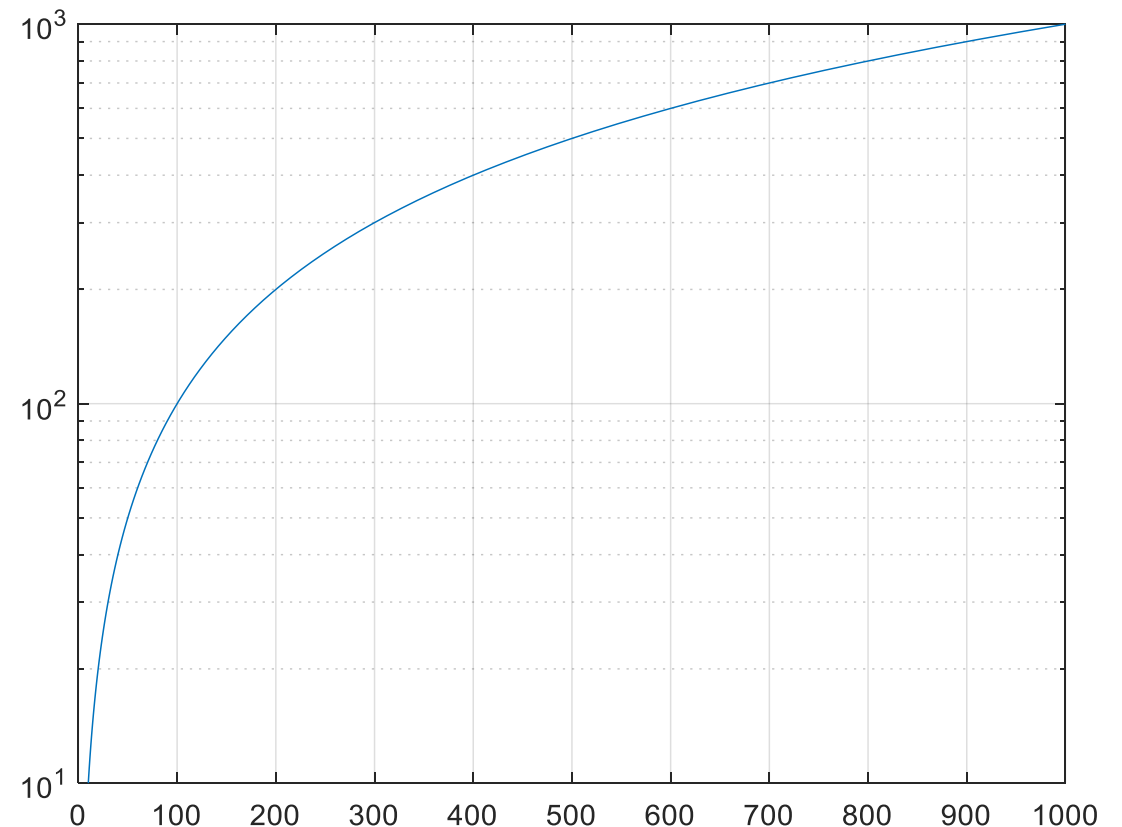


- ❑ ***semilogy(x,y)*** : trace les coordonnées  $x$  et  $y$  en utilisant une échelle logarithmique de base 10 sur l'axe des  $y$  et une échelle linéaire sur l'axe des  $x$ .

## Exemple:

```
x = logspace(1,3,100);  
f = x ;  
semilogy(x,f)  
grid on
```

Grappe



# Note 1:

- Dans Matlab il y a d'autres fonctions qui donnent une représentation graphiques 2D, il suffit de taper **Types of MATLAB Plots** dans la documentation de Matlab pour découvrir les différentes fonctions.
- Utiliser le help de la fonction écrite sur l'image de chaque fonction pour connaître son effet et la façon dont elle est utilisées.
- Vous pouvez aussi taper l'expression **help graph2d** dans la zone de commande pour obtenir la liste des fonctions de représentation graphique 2D disponibles dans MATLAB.

# Plan

- 1- Introduction
- 2- Représentation graphique 2D
- 3- Représentation graphique 3D**
- 4- Exercices d'application

# 3. Représentation graphique 3D

## □ Fonction plot3

C'est une fonction qui donne une représentation graphique sous 3D.

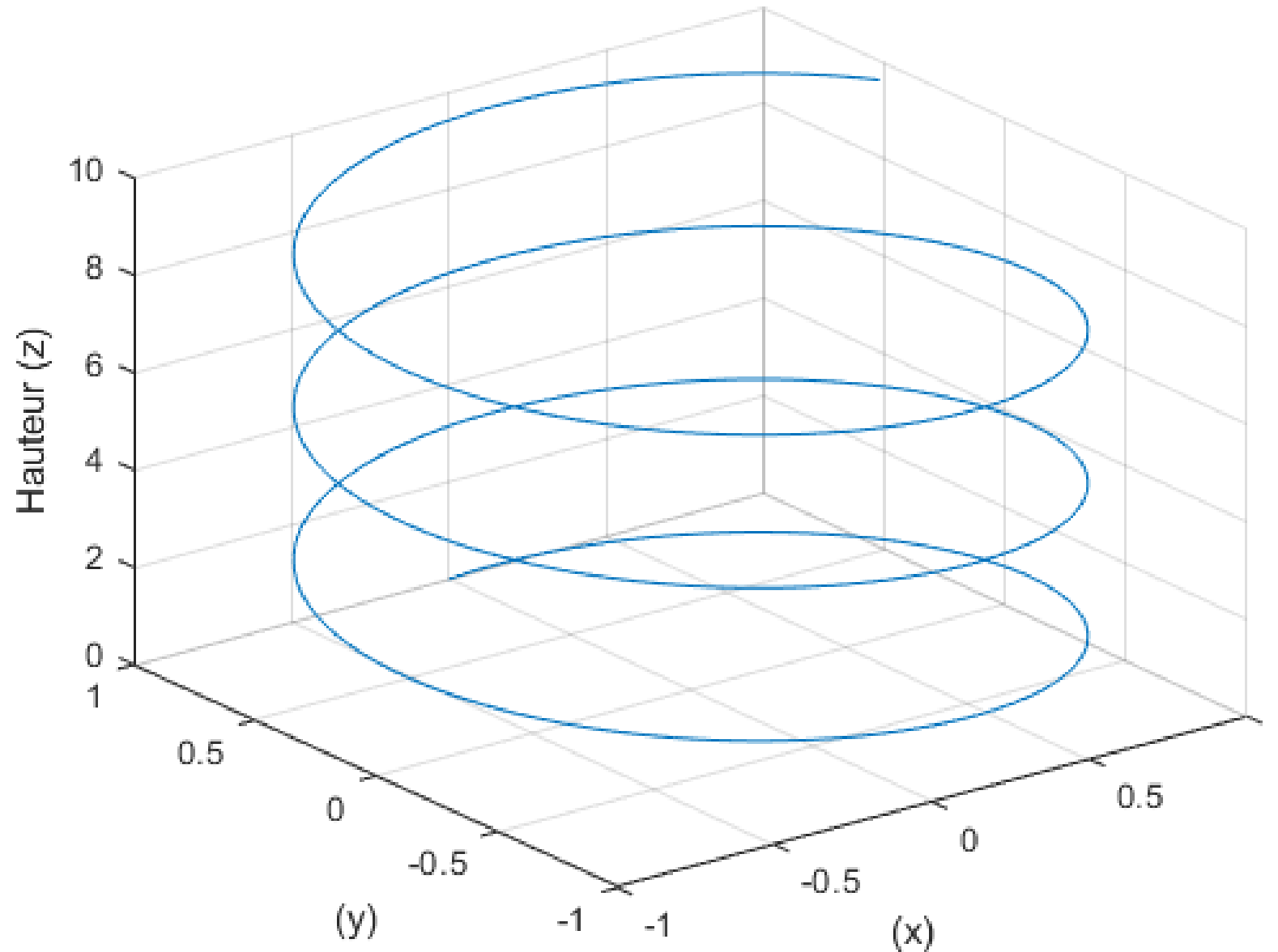
La commande **plot3(x,y,t)** permet de tracer les vecteurs **y**, **x** et **t** reliées par des segments de ligne.

## Exemple d'application:

Représentation sous 3D des deux fonctions x et y et le temps t.

```
z=0:0.01:10;  
x=sin(2*z); y=cos(2*z);  
plot3(x,y,z)  
grid on  
xlabel('(x)'), ylabel('(y)'), zlabel('Hauteur (z)')
```

**Graphe:** Pour chaque  $z$  on vérifie la relation de  $x$  et de  $y$ .





## □ Fonction mesh

C'est une fonction qui permet de tracer un maillage qui est sous forme une surface tridimensionnelle.

*mesh*( $X, Y, Z$ ) :

- permet de tracer le maillage de la fonction  $Z$  en fonction des deux vecteurs  $X$  et  $Y$ ,
- permet de tracer le maillage de  $Z$  au-dessus du plan  $(X, Y)$  (c.-à-d  $Z$  en hauteur) .
- Une couleur déterminée par  $Z$  afin que la couleur soit proportionnelle à la hauteur de la surface.

*surf* est comme la fonction *mesh*, permet de tracer une surface sous trois dimensions.

## Exemple d'application:

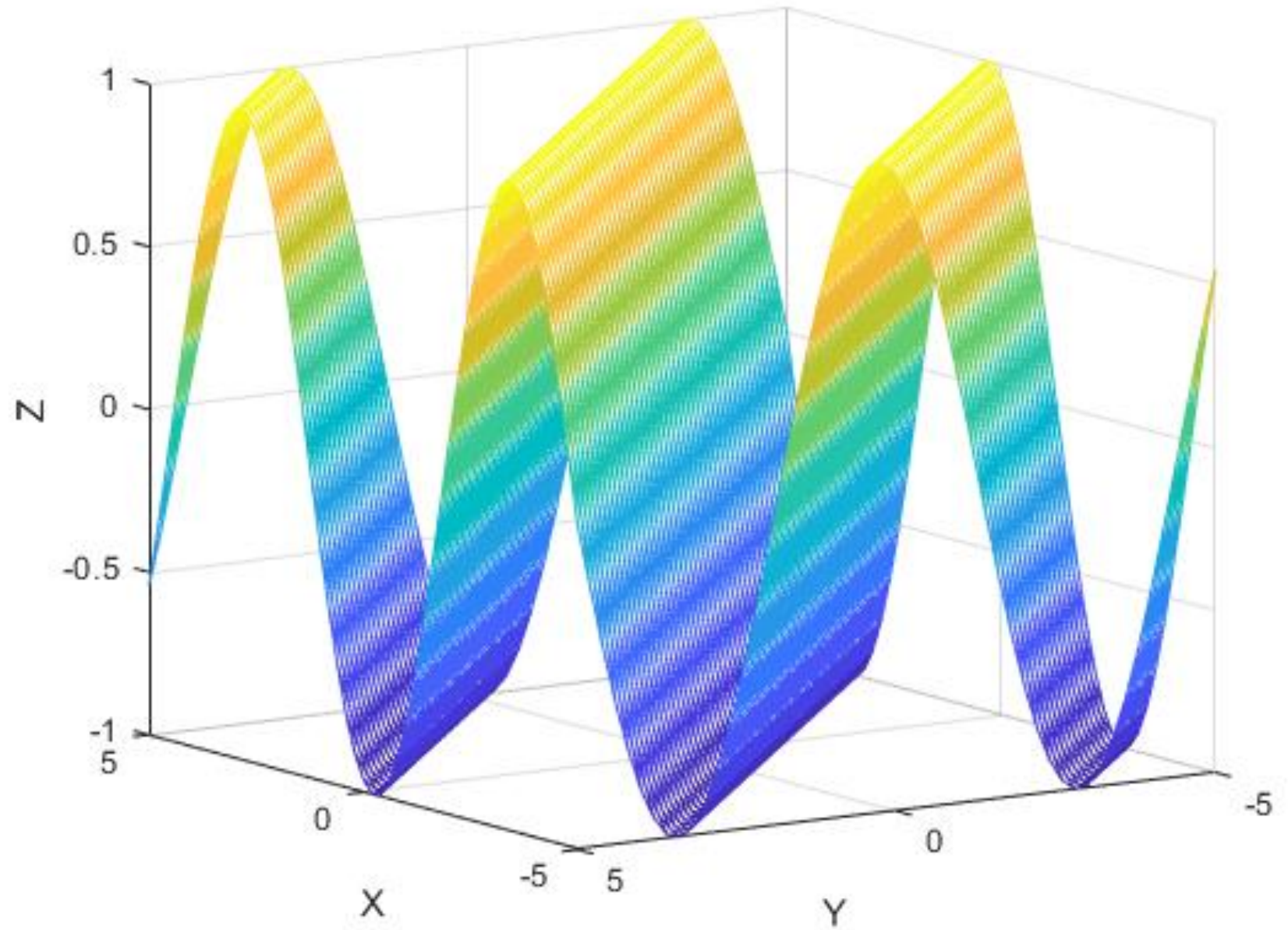
- Tracer le maillage de la fonction  $Z = \sin(X + Y)$ ,
- avec  $X$  et  $Y$  sont deux vecteurs de même dimensions  $X = Y = [-5, 5]$ .

## Programme:

```
[X,Y] = meshgrid(-5:0.2:5);  
Z = sin(X+Y);  
mesh(X,Y,Z)  
xlabel('X'), ylabel('Y'), zlabel('Z')
```

**Meshgrid** est une commande utilisée pour créer un maillage de deux dimensions (X,Y)

**Graphe:**



## Exercice d'application:

Tracer le maillage de la fonction sinus cardinal:  $Z = \sin(X + Y) / (X + Y)$ ,

- avec  $X$  et  $Y$  sont deux vecteurs de même dimensions  $[-25, 25]$ .

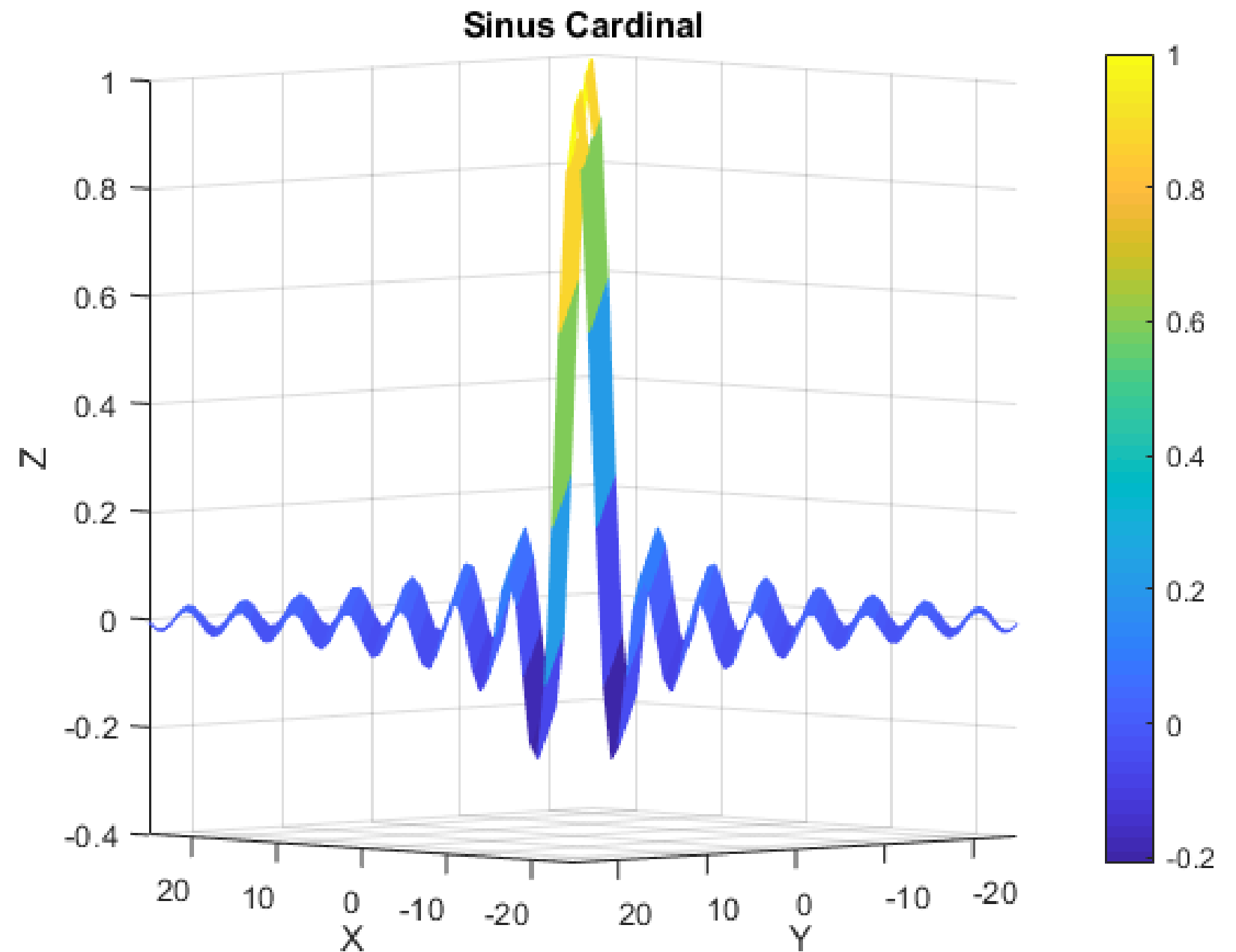
(Essayer d'utiliser la commande **linspace** pour générer les deux vecteurs  $X$  et  $Y$  ).

- Donner un titre à la figure,
- Donner un titre pour chaque axe,
- Essayer la commande **colorbar**.

## Programme:

```
a=linspace(-25,25,60);  
[X,Y] = meshgrid(a);  
Z = sin(X+Y) ./ (X+Y);  
mesh(X,Y,Z)  
title('Sinus Cardinal')  
xlabel('X'), ylabel('Y'), zlabel('Z')  
colorbar
```

**Graphe:**



## ❑ Autres fonctions de représentation 3D

### Fonction **meshc** :

C'est comme la fonction **mesh**, la différence qu'elle permet d'afficher les lignes de niveau sous la surface dans le plan  $Z = Z_{min}$ .

### Fonction **contour** :

Elle permet de tracer les lignes de chaque niveau sous forme une surface.

On peut imposer le nombre de lignes de niveau  $N$  en utilisant la commande **contour(X,Y,Z,N)**.

**Exemple:** `contour(X,Y,Z,10)`

# Fonction **sphere** :

C'est une fonction qui permet de créer une sphère sans la dessiner.

$[X, Y, Z] = \textit{sphere}$  renvoie les coordonnées  $X$ ,  $Y$  et  $Z$  d'une sphère sans la dessiner.

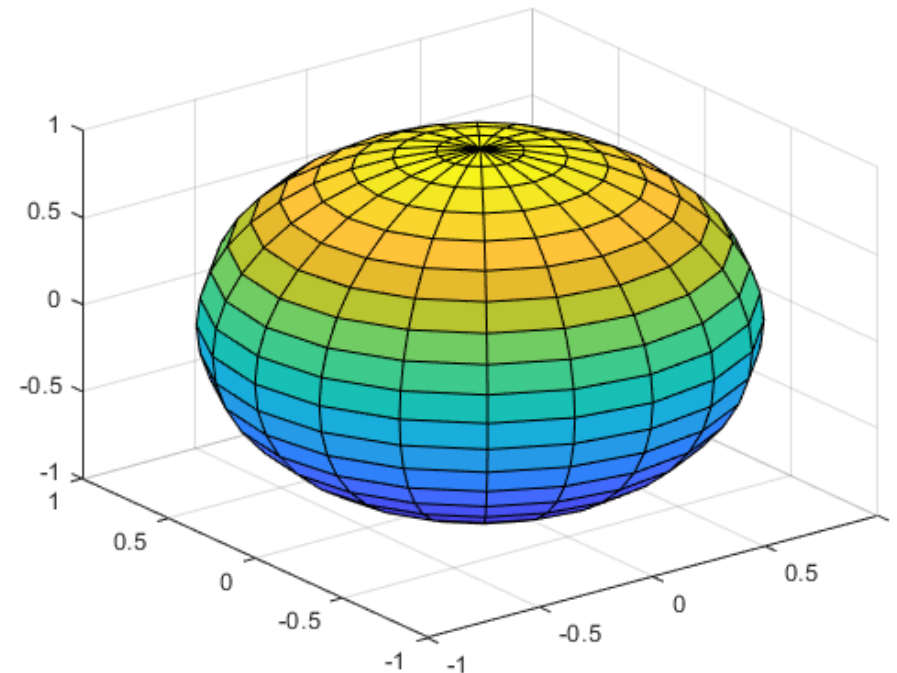
Pour dessiner la sphère en utilisant les coordonnées renvoyées, utilisez les fonctions **surf** ou **mesh**.

## Exemple d'application:

### Programme:

```
[X,Y,Z] = sphere;  
surf(X,Y,Z)
```

### Graphe:





# Fonction **cylinder** :

C'est une fonction qui permet de créer un cylindre sans le dessiner.

$[X, Y, Z] = \text{cylinder}$  renvoie les coordonnées  $X$ ,  $Y$  et  $Z$  d'un cylindre sans le dessiner.

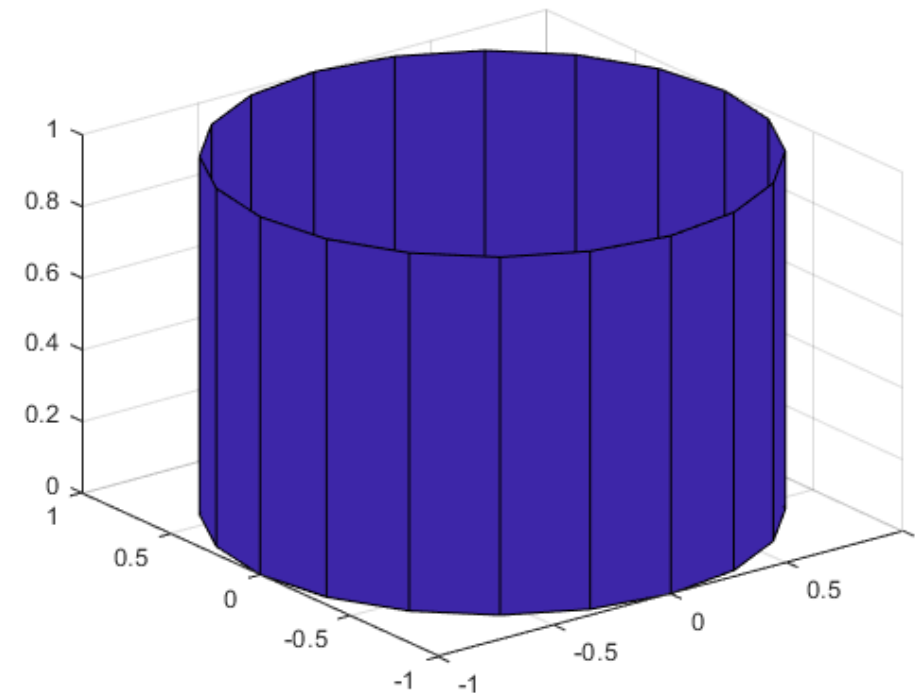
Comme le cas de la sphère, on utilise les fonctions **surf** ou **mesh** pour dessiner le cylindre.

## Exemple d'application:

### Programme:

```
[X,Y,Z] = cylinder;  
surf(X,Y,Z)
```

### Graphe:



## Note 2:

- Dans les paragraphes précédents nous avons vu quelques une des fonctions 3D, il suffit de taper **Types of MATLAB Plots** dans la documentation de Matlab pour découvrir les autres fonctions.
- Vous pouvez aussi taper l'expression **help graph3d** dans la fenêtre de commande pour obtenir la liste des fonctions de représentation graphique 2D disponibles dans MATLAB.

# Plan

- 1- Introduction
- 2- Représentation graphique 2D
- 3- Représentation graphique 3D
- 4- Exercices d'application**

# Exercice 1: **partie 1**

À l'aide de la fonction ***plot***, créer un script qui représente en 2D, les fonctions suivantes dans une seule fenêtre, sachant que chaque fonction est représentée dans une figure indépendante des autres.

➤ **Les fonctions:** Sinus, Cosinus, Tangente, Arcsinus, Arccosinus et Arctangente.

## **Remarque:**

- ☐ Représenter **Sinus, Cosinus, Tangente** sur l'intervalle des abscisses (angle)  $[-2\pi, 2\pi]$ ,
- ☐ Représenter les fonctions inverses sur l'intervalle des abscisses ( $x$ ):  $[-3, 3]$ ,
- ☐ Donner un titre pour chaque figure,
- ☐ Donner des titres pour les axes des abscisses (*Angle* ou  $x$ ) et des ordonnées (*Amplitude*),
- ☐ Utiliser un quadrillage pour chaque figure.

# Exercice 1: **partie 2**

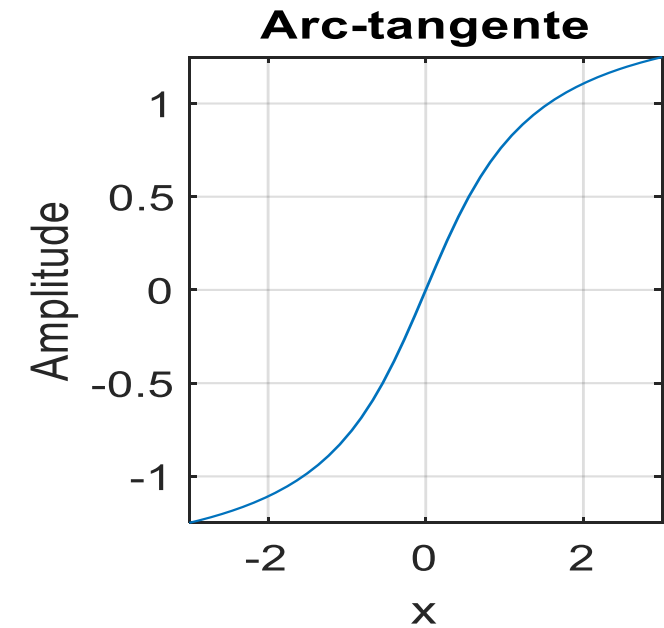
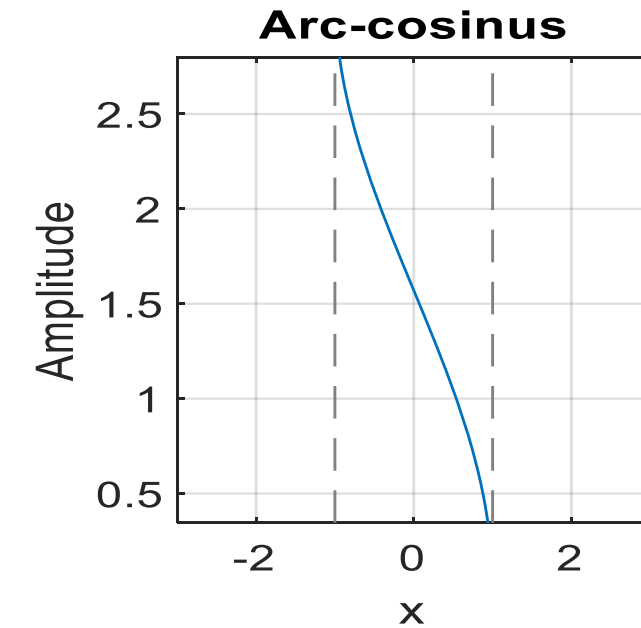
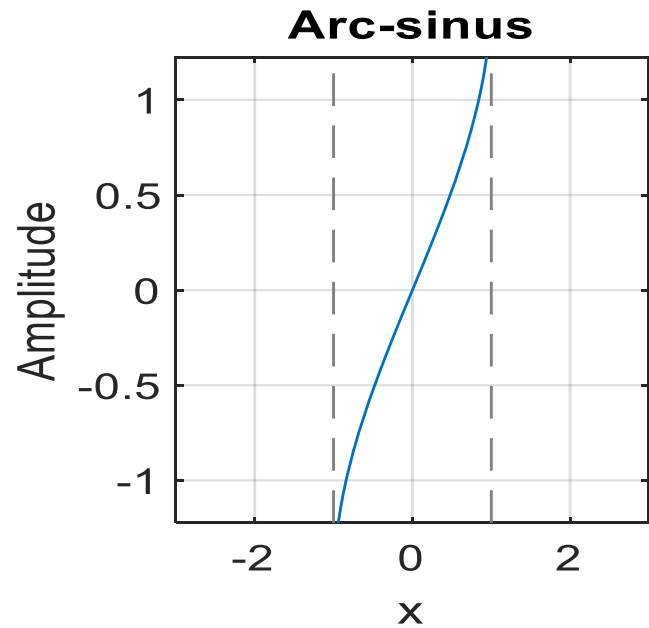
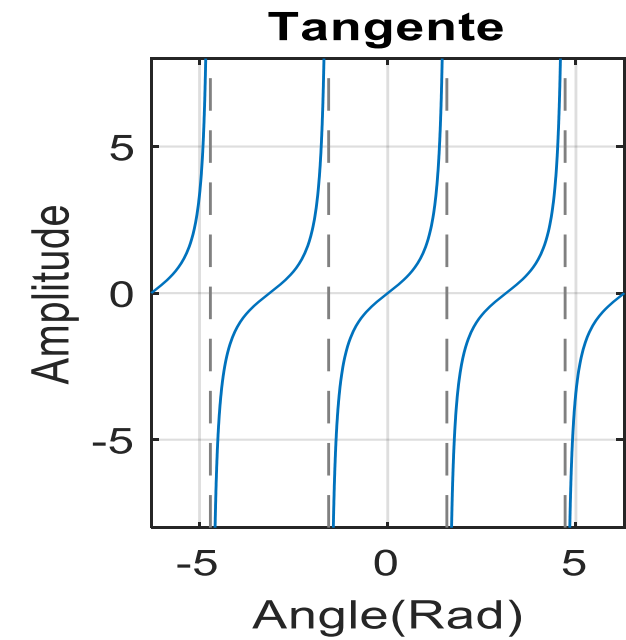
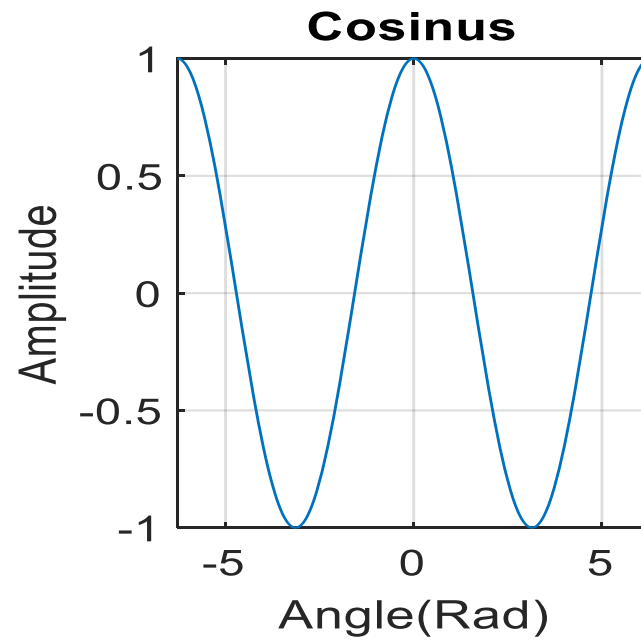
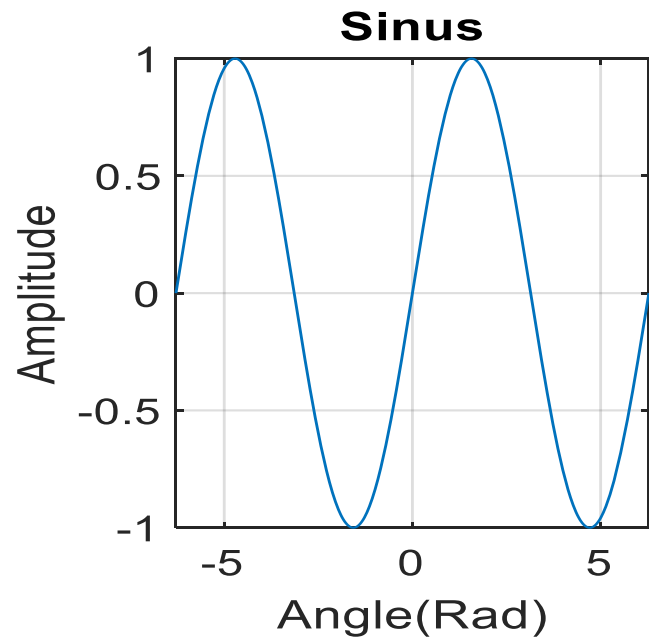
Refaites la partie 1 mais en se basant sur la fonction ***fplot*** au lieu de ***plot***.

➤ **Les fonctions:** Sinus, Cosinus, Tangente, Arcsinus, Arccosinus et Arctangente.

## **Remarque:**

- ☐ Représenter **Sinus, Cosinus, Tangente** sur l'intervalle des abscisses (angle)  $[-2\pi, 2\pi]$ ,
- ☐ Représenter les fonctions inverses sur l'intervalle des abscisses ( $x$ ):  $[-3, 3]$ ,
- ☐ Donner un titre pour chaque figure,
- ☐ Donner des titres pour les axes des abscisses (*Angle* ou  $x$ ) et des ordonnées (*Amplitude*),
- ☐ Utiliser un quadrillage pour chaque figure.

# Graphe obtenu:



## Exercice 2: **partie 1**

Soit la suite définie par :

$$U(n + 1) = M * U(n) + b$$

Sachant que  $M$  est une matrice et  $U$  et  $b$  sont deux vecteurs colonnes:

$$M = \begin{bmatrix} 5/8 & -1/4 & 1/8 \\ 1/4 & 0 & 1/4 \\ 1/8 & -1/4 & 5/8 \end{bmatrix}$$

$$U_1 = \begin{pmatrix} 4 \\ 1 \\ -4 \end{pmatrix}$$

$$b = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

- ☐ Créer un script qui calcule les termes de la suite pour  $n = 20$ .
- ☐ Qu'est ce que vous remarquez à propos des termes de la suite?
- ☐ À l'aide de la fonction ***plot***, représenter l'évolution des composantes (chaque ligne en fonction de toutes les colonnes) de cette suite graphiquement.

## Exercice 2: **partie 2**

On considère la même suite définie par :

$$U(n + 1) = M * U(n) + b$$

- ❑ Créer une fonction nommée ***Fonc\_Suite*** qui possède en entrée ( $M$ ,  $U_1$ ,  $b$  et  $n$ ) et qui calcule en sortie ( $U$ ) les termes de la suite.
- ❑ Dans un nouveau script, tapez la matrice  $M$  et les deux vecteurs colonnes  $U_1$  et  $b$ , puis appeler la fonction pour calculer les termes pour un nombre  $n$  de votre choix.

$$M = \begin{bmatrix} 5/8 & -1/4 & 1/8 \\ 1/4 & 0 & 1/4 \\ 1/8 & -1/4 & 5/8 \end{bmatrix}$$

$$U_1 = \begin{pmatrix} 4 \\ 1 \\ -4 \end{pmatrix}$$

$$b = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

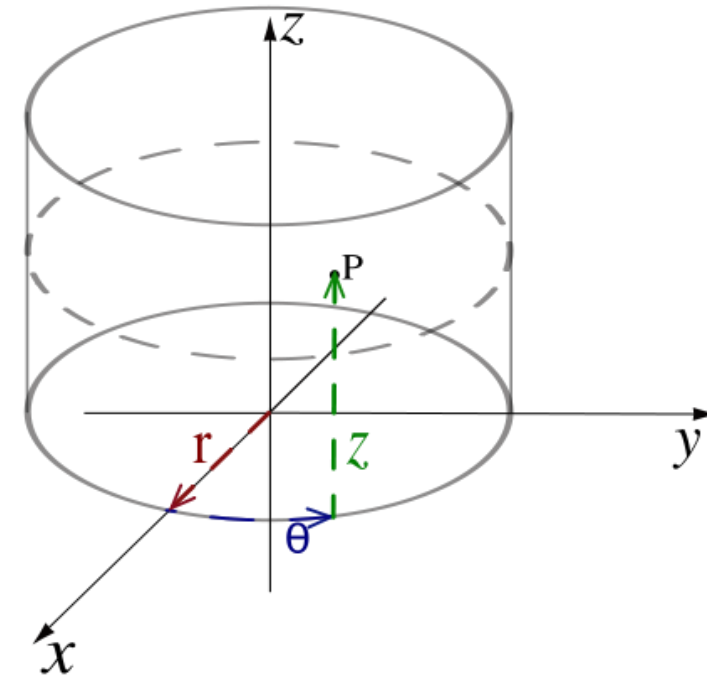
- ❑ Représenter l'évolution des composantes (les trois lignes) de cette suite graphiquement (Déclarer les commandes d'affichage (***plot***) à partir de la fonction ou de script).



# Exercice 3: Partie 1

- ❑ La relation entre les coordonnées cylindriques et cartésiennes est donnée par :

$$\begin{cases} x = r \cos(\theta) \\ y = r \sin(\theta) \\ z = h \end{cases}$$



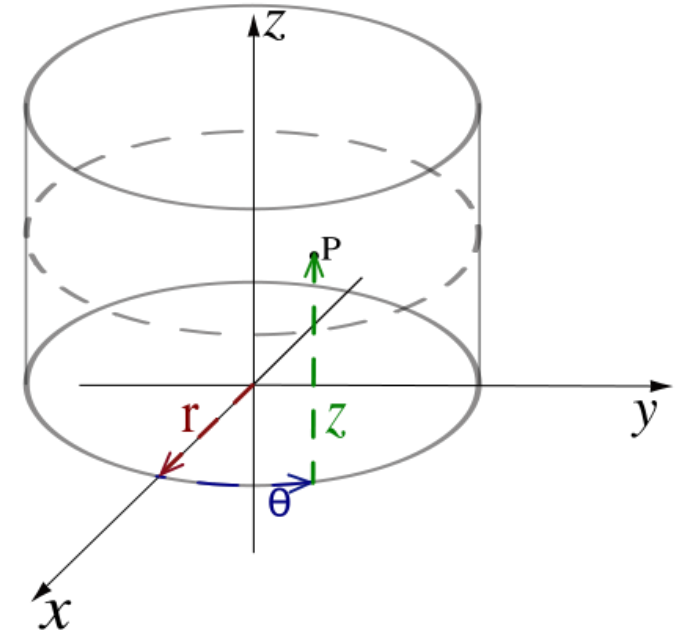
- ❑ Donner un script qui demande la valeur du rayon  $r$  et la hauteur  $h$  du cylindre, puis :

1. Il calcule la surface totale (bases + latérale) de cylindre et son volume,
  2. Il trace le cylindre en 3D en se basant sur les commandes **surf** ou **mesh**,
  3. Il donne un titre pour la figure et pour chaque axe ( $x$ ,  $y$  et  $z$ ),
- Utiliser la commande **colorbar** et comparer la valeur max avec la hauteur  $h$ .

## Exercice 3: **Partie 2**

- ❑ Créer une fonction qui permet de calculer la surface, le volume et de tracer le cylindre, sachant qu'elle possède comme des entrées: le rayon  $r$  et la hauteur  $h$  et comme sortie la surface et le volume de cylindre.
- ❑ Appeler la fonction dans un script en insérant les données d'entrée pour tracer le cylindre.

$$\begin{cases} x = r \cos(\theta) \\ y = r \sin(\theta) \\ z = h \end{cases}$$



# FIN