

Chapitre 5: Le graphisme sous Matlab

V.1 Tracer le graphe d'une fonction; la commande fplot

La commande `fplot` permet de tracer le graphe d'une fonction sur un intervalle donné. La syntaxe est:

```
fplot('nomf', [xmin , xmax])
```

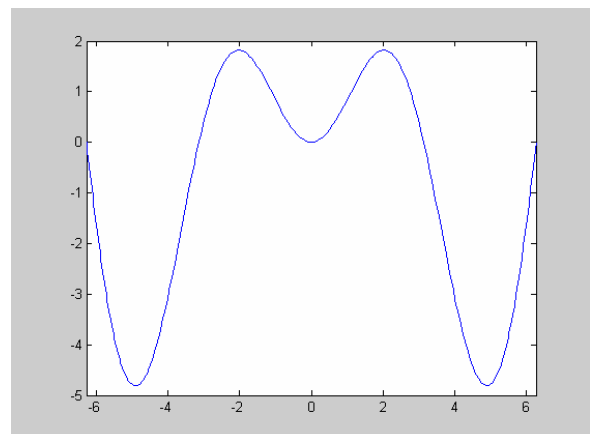
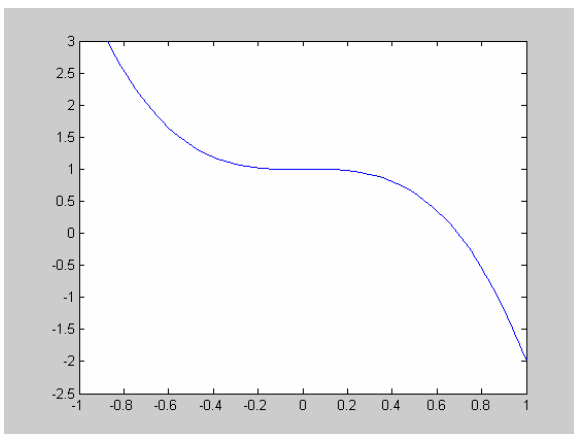
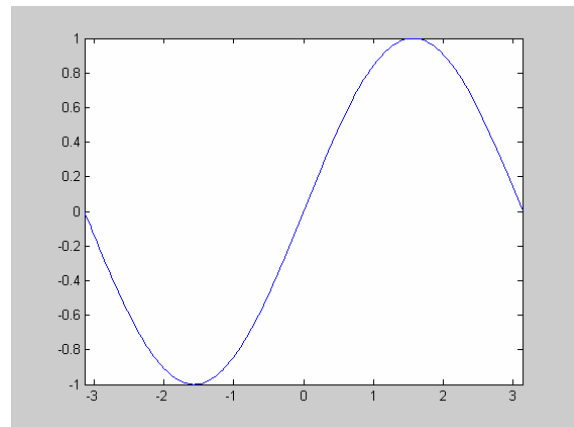
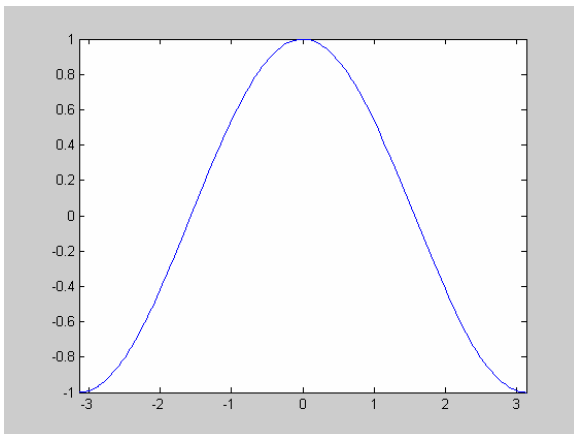
où

- `nomf` est soit le nom d'une fonction MATLAB incorporée, soit une expression définissant une fonction de la variable `x`, soit le nom d'une fonction utilisateur.
- `[xmin , xmax]` est l'intervalle pour lequel est tracé le graphe de la fonction.

Exemples :

```
figure(1),fplot('cos',[pi -pi])  
figure(2),fplot('sin',[pi -pi])  
figure(3),fplot('-3*x^3+1',[-1 +1 -2.5 3])  
figure(4),fplot('exp',[pi -pi])
```

```
function y=MaFonction(x)  
y=x.*sin(x);
```

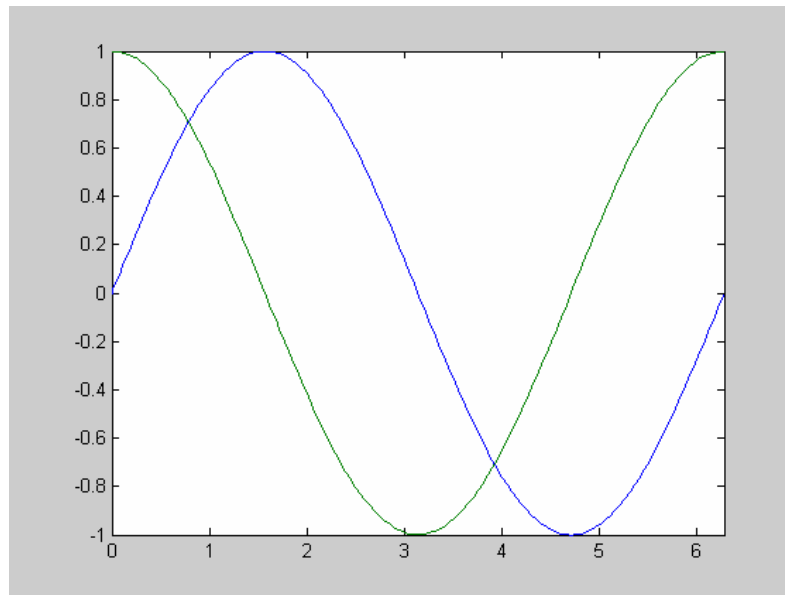


Il est possible de tracer plusieurs fonctions sur la même figure. Il faut pour cela utiliser la commande `fplot` de la manière suivante:

`fplot('nom_f1 , nom_f2 , nom_f3', [x_min , x_max])` où `nom_f1`, `nom_f2`, `nom_f3` est soit le nom d'une fonction MATLAB incorporée, soit une expression définissant une fonction de la variable `x`, soit le nom d'une fonction utilisateur.

Il est également possible de gérer les bornes des valeurs en ordonnées. Pour limiter le graphe aux ordonnées comprises entre les valeurs `y_min` et `y_max` on passera comme second argument de la commande `fplot` le tableau `[x_min , x_max , y_min , y_max]`.

Exemple : `fplot('sin(x), cos(x)', [0 2*pi])`



V.2 La commande plot

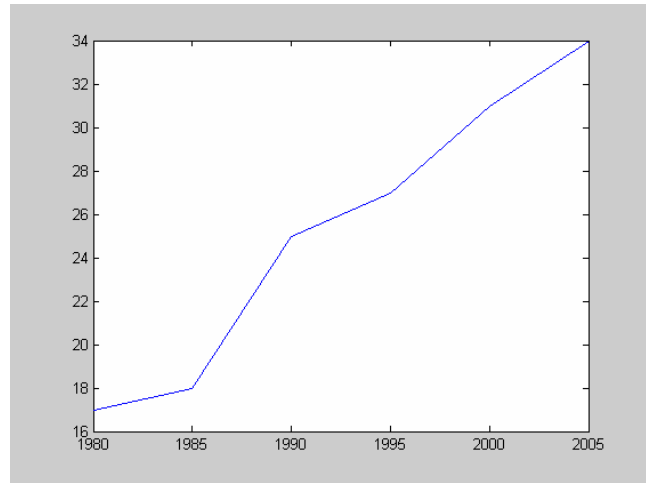
La commande `plot` permet de tracer un ensemble de points de coordonnées (x_i, y_i) $i = 1, \dots, N$. La syntaxe est `plot(x,y)` où `x` est le vecteur contenant les valeurs x_i en abscisse et `y` est le vecteur contenant les valeurs y_i en ordonnée. Bien entendu les vecteurs `x` et `y` doivent être de même dimension mais il peut s'agir de vecteurs lignes ou colonnes. Par défaut, les points (x_i, y_i) sont reliés entre eux par des segments de droites.

Voici par exemple une autre façon de tracer le graphe de la fonction $h(x) = x \sin(x)$ entre -2π et 2π ,

```
>> x=[-2*pi:0.01:2*pi]; y = x.*sin(x);  
>> plot(x,y)
```

Exemple 01:

```
>> A=[1980 1985 1990 1995 2000 2005];  
>> P=[17 18 25 27 31 34];  
>> plot(A,P) % par défaut
```



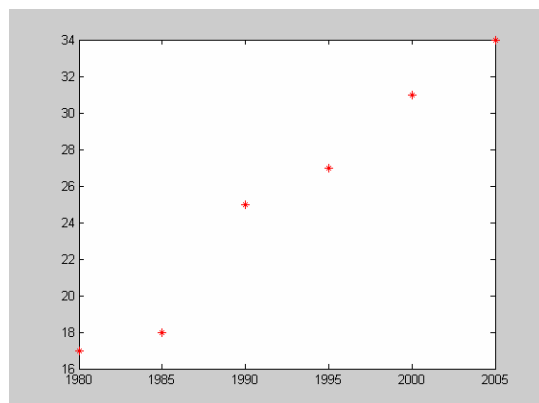
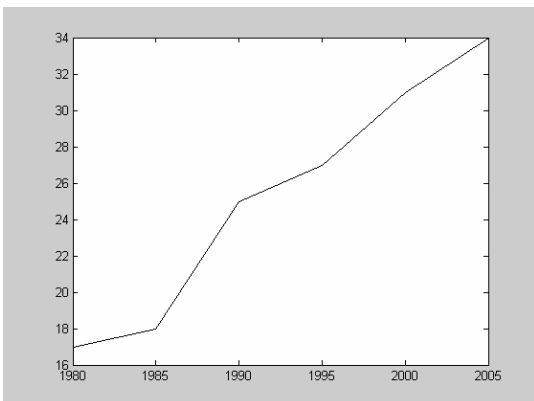
On peut spécifier à MATLAB quelle doit être la couleur d'une courbe, quel doit être le style de trait et/ou quel doit être le symbole à chaque point (x_i, y_i) . Pour cela on donne un troisième paramètre d'entrée à la commande `plot` qui est une chaîne de 3 caractères de la forme '`cst`' avec `c` désignant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

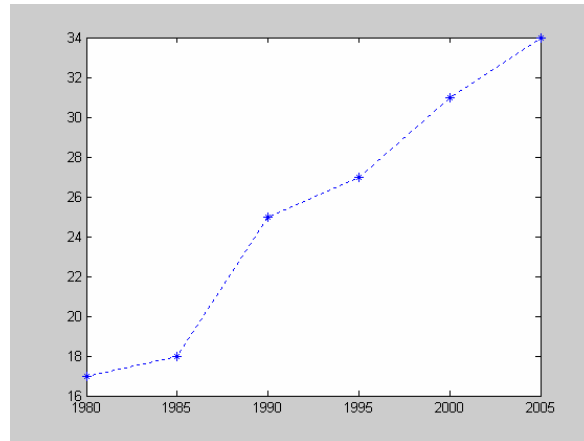
y	jaune	.	point	-	trait plein
m	magenta	o	cercle	:	pointillé court
c	cyan	x	marque x	--	pointillé long
r	rouge	+	plus	-.	pointillé mixte
g	vert	*	étoile		
b	bleu	s	carré		
w	blanc	d	losange		
k	noir	v	triangle (bas)		
		^	triangle (haut)		
		<	triangle (gauche)		
		>	triangle (droit)		
		p	pentagone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

Les valeurs par défaut sont `c = b`, `s = .` et `t = -` ce qui correspond à un trait plein

Exemple 02:

```
>> A=[1980 1985 1990 1995 2000 2005]; P=[17 18 25 27 31 34];  
>> plot(A,P)% par défaut  
>> plot(A,P,'k')% couleur(noir)  
>> plot(A,P,'r*')%couleur(rouge) + Symbole en chaque point (xi,yi) (étoile)  
>> plot(A,P,'b*:')% couleur + Symbole en chaque point (xi,yi) + Style de trait (pointillé court)
```





V.3 Afficher plusieurs courbes dans une même fenêtre

Il est possible d'afficher plusieurs courbes dans une même fenêtre graphique grâce à la commande `hold on`. Les résultats de toutes les instructions graphiques exécutées après appel à la commande `hold on` seront superposés sur la fenêtre graphique active. Pour rétablir la situation antérieure (le résultat d'une nouvelle instruction graphique remplace dans la fenêtre graphique le dessin précédent) on tapera `hold off`.

Voici un exemple d'utilisation de la commande `hold on`. Le résultat est présenté à la figure 10.

```
>> e = exp(1);
>> figure
>> hold on
>> fplot('exp',[-1 1])
>> fplot('log',[1/e e])
>> plot([-1:0.01:e],[-1:0.01:e])
>> grid
>> hold off
```

V.4 LA COMMANDE SUBPLOT

Il est possible de décomposer une fenêtre en sous-fenêtres et d'afficher une figure différente sur chacune de ces sous-fenêtres grâce à la commande `subplot`. La syntaxe est

`subplot(m,n,i)`

où

- `m` est le nombre de sous-fenêtres verticalement;
- `n` est le nombre de sous-fenêtres horizontalement;
- `i` sert à spécifier dans quelle sous-fenêtre doit s'effectuer l'affichage. Les fenêtres sont numérotées de gauche à droite et de haut en bas.

```
subplot(2,2,1)
fplot('cos',[pi -pi]),title('fonction cosinus')
subplot(2,2,2)
fplot('sin',[pi -pi]),title('fonction sinus')
subplot(2,2,3)
fplot('-3*x^3+1',[-1 +1 -2.5 3]),title('fonction -3x3+1')
subplot(2,2,4)
fplot('exp',[pi -pi]),title('fonction exp')
```

