

Rappels d'utilisation de MATLAB

✿ Utilisation de l'environnement graphique de MATLAB

- demander de l'aide :
 - `help plot` pour obtenir l'aide de la fonction plot dans la fenêtre de commande ;
 - `doc plot` pour obtenir la documentation de la fonction plot dans une fenêtre graphique.
- accéder à la liste des variables :
 - regarder le contenu de la fenêtre « Workspace » ;
 - utiliser les commandes `who` et `whos`.
- changer le répertoire de travail :
 - utiliser la fenêtre « Current Directory » ;
 - `pwd` pour savoir dans quel répertoire on se trouve ;
 - `cd` pour changer le répertoire courant.
- réinitialiser l'espace de travail :
 - `clear all` pour supprimer toutes les variables dans la mémoire ;
 - `close all` pour fermer toutes les fenêtres graphiques ;
 - `clc` pour nettoyer la fenêtre de commande.
- taper ses instructions dans un script :
 - créer un fichier dans l'éditeur de texte (File/New/M-File) ;
 - taper les commandes dans l'éditeur de texte puis enregistrer ;
 - exécuter les commandes du script avec la touche F5.



s'assurer que le script ne porte pas le même nom qu'une fonction prédéfinie !

✿ Création de matrices et de vecteurs sous MATLAB

- créer une matrice M :
 - `M=[[1,2,3] ; [4,5,6] ; [7,8,9]]`
 - `M=[1 2 3 ; 4 5 6 ; 7 8 9]`
- créer un vecteur ligne L :
 - `L=[1,2,3]`
 - `L=[1 2 3]`
- créer un vecteur colonne C :
 - `C=[1 ; 2 ; 3]`
 - `C=[1 2 3]'`
- créer un vecteur à l'aide d'une suite de nombres :
 - `t1=1 : 1 : 10` \Leftrightarrow `t1=[1 2 3 4 5 6 7 8 9 10]`
 - `t2=1 : 2 : 10` \Leftrightarrow `t2=[1 3 5 7 9]`
 - `t3=10 : -1 : 1` \Leftrightarrow `t3=[10 9 8 7 6 5 4 3 2 1]`
- fonction `size` pour connaître les dimensions d'une matrice
- fonction `length` pour connaître directement la dimension d'un vecteur



Les opérateurs de création de matrices sont les crochets [] et non les parenthèses () ou les accolades { } !

✿ **Extraction d'éléments à l'aide des parenthèses ()**

Il y a deux manières de procéder à l'extraction d'éléments d'une matrice : soit en créant un masque booléen de mêmes dimensions que la matrice ; soit en spécifiant directement les indices.

➔ extraction d'éléments sur le vecteur $x=1 : 10$

masque	indice	effet
<code>x(logical([0 0 0 0 1 0 0 0 0 0]))</code>	<code>x(5)</code>	extrait le cinquième élément
<code>x(logical([1 0 0 0 1 0 0 0 0 1]))</code>	<code>x([1 5 10])</code>	extrait les éléments 1, 5 et 10
<code>x(logical([1 1 1 1 1 0 0 0 0 0]))</code>	<code>x(1 : 5)</code>	extrait les cinq premiers éléments
<code>x(logical([1 0 1 0 1 0 1 0 1 0]))</code>	<code>x(1 : 2 : end)</code>	extrait les éléments d'indice impair
<code>x(logical([0 1 0 1 0 1 0 1 0 1]))</code>	<code>x(2 : 2 : end)</code>	extrait les éléments d'indice pair
<code>x(x>10)</code>	<code>x(find(x>10))</code>	extrait les éléments supérieurs à 10
<code>x(rem(x,2)==0)</code>	<code>x(find(rem(x,2)==0))</code>	extrait les éléments pairs
<code>x(rem(x,2)==1)</code>	<code>x(find(rem(x,2)==1))</code>	extrait les éléments impairs

➔ extraction d'éléments sur la matrice aléatoire $M=rand(10,5)$

`M(5,2)` pour extraire l'élément de la cinquième ligne et de la deuxième colonne ;

`M(:,2)` pour extraire l'ensemble des éléments de la deuxième colonne ;

`M(1 :5,2)` pour extraire les cinq premiers éléments de la deuxième colonne ;

`M(5, :)` pour extraire l'ensemble des éléments de la cinquième ligne ;

`M(5,3 :end)` pour extraire les trois derniers éléments de la cinquième ligne.



⊥ Pour extraire les éléments dans l'ordre inverse, on peut utiliser la commande `x(end : -1 : 1)` !

✿ **Opérations sur les matrices / vecteurs**

➔ bien distinguer la différence entre les opérations matricielles et les opérations élément par élément ;

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix} \quad A * B = \begin{pmatrix} 3 & 6 & 9 \\ 6 & 12 & 18 \\ 9 & 18 & 27 \end{pmatrix} \quad A .* B = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 9 \end{pmatrix}$$

➔ ne pas confondre la transposée et l'inverse : $A' \neq \text{inv}(A)$;

➔ penser à utiliser les fonctions prédéfinies telles que `sum()`, `cumsum()`, `mean()`, `var()`, `std()`, etc.

✿ **Création de fonctions** Il faut créer un fichier qui porte nécessairement le même nom que le nom de la fonction. La première ligne du fichier doit être la suivante :

`function [s1, s2, ...] = mafonction(e1, e2, ...)`

où $e1, e2, \dots$ correspondent aux paramètres d'entrée de la fonction, tandis que $s1, s2, \dots$ correspondent aux paramètres de sortie de la fonction. Pour utiliser la fonction, il suffit de taper

`[s1, s2, ...] = mafonction(e1, e2, ...)`

dans la fenêtre de commande après avoir naturellement pris soin de définir les paramètres d'entrée.



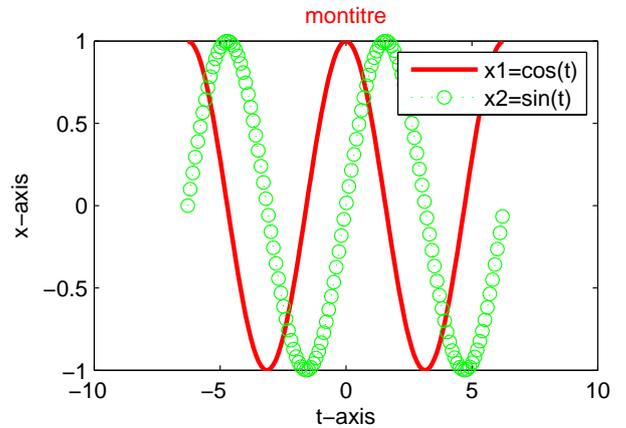
⊥ Les arguments de la fonction sont des variables muettes. Il est donc possible de choisir des noms différents lors de la déclaration et de l'appel !

✿ **Utilisation des fonctions plot et subplot**

➔ superposer deux courbes dans une même figure :

```
t=-2*pi:0.1:2*pi;
x1=cos(t);
x2=sin(t);
```

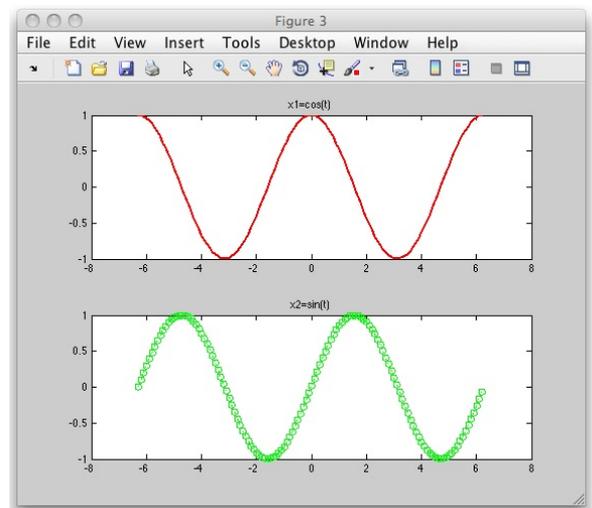
```
figure
plot(t,x1,'Color','r','LineWidth',2)
hold on
plot(t,x2,'go')
hold off
title('montitre','Color','r')
xlabel('t-axis')
ylabel('x-axis')
legend('x1=cos(t)','x2=sin(t)')
```



➔ tracer deux courbes dans deux sous-fenêtres distinctes :

```
t=-2*pi:0.1:2*pi;
x1=cos(t);
x2=sin(t);
```

```
figure
subplot(2,1,1)
plot(t,x1,'Color','r','LineWidth',2)
title('x1=cos(t)')
subplot(2,1,2)
plot(t,x2,'go')
title('x2=sin(t)')
```



⚠ La fonction subplot(m,n,p) permet de diviser une fenêtre graphique en $m \times n$ sous-figures. Les indices m et n correspondent respectivement au nombre de divisions verticales et horizontales. La position des sous-figures est renseignée par l'indice p . La numérotation s'effectue de gauche à droite en partant du haut.