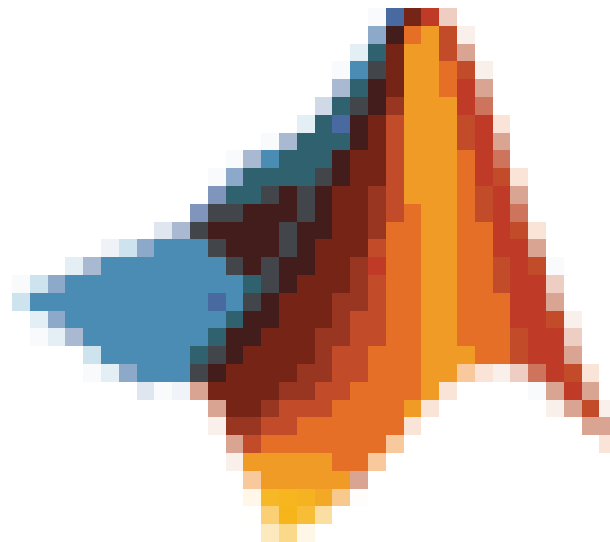
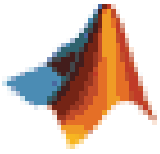
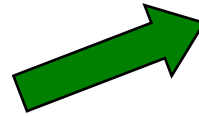
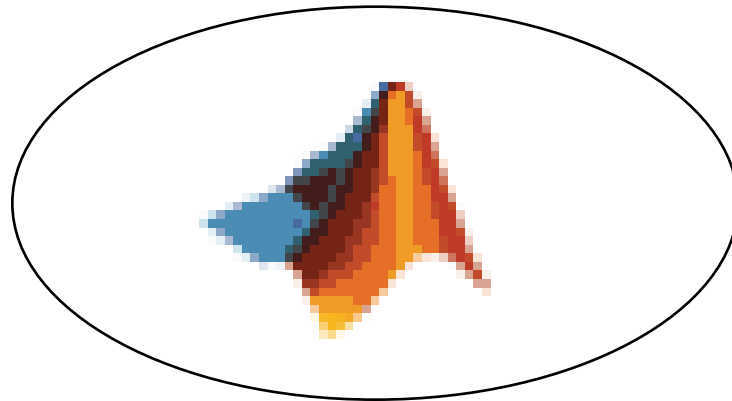


MATLAB: le Laboratoire des Matrices

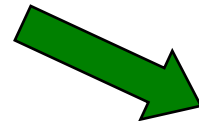




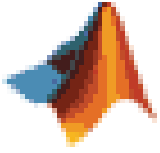
Introduction à Matlab



Logiciel Matlab



SIMULINK



Introduction à Matlab

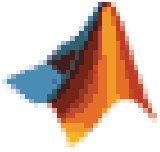
- Le Logiciel Matlab:

Comprend une vaste bibliothèque (de milliers de fonctions) de fonctions spécialisées (Toolboxes).

Nombre de problèmes de l'ingénieur

– ne peuvent être résolus que numériquement

– sont où doivent être formulés de manière matricielle

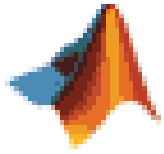


Introduction à Matlab

- Simulink:

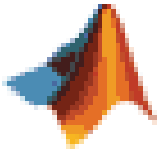
Un environnement puissant de modélisation basé sur les schemas blocs et la simulation de systèmes dynamiques linéaires et non linéaires

Des bibliothèques de blocs Simulink (Blocksets)



Presentation de Matlab

- MATLAB est un environnement complet, orienté vers le calcul matriciel, ouvert et extensible pour le calcul et la visualisation. Il dispose de plusieurs fonctions mathématiques, scientifiques et techniques. L'approche matricielle de MATLAB permet de traiter les données sans aucune limitation de taille et de réaliser des calculs de façon fiable et rapide. Grâce aux fonctions graphiques de MATLAB, il devient très facile de modifier interactivement les différents paramètres des graphiques pour les adapter selon nos souhaits.
- Les domaines couverts sont très variés et comprennent notamment :
 - le traitement du signal, l'automatique, l'identification de systèmes, les réseaux de neurones, la logique floue, le calcul de structure, les statistiques, la simulation des modèles, etc.

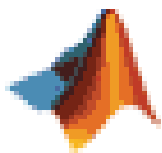


Presentation de Matlab

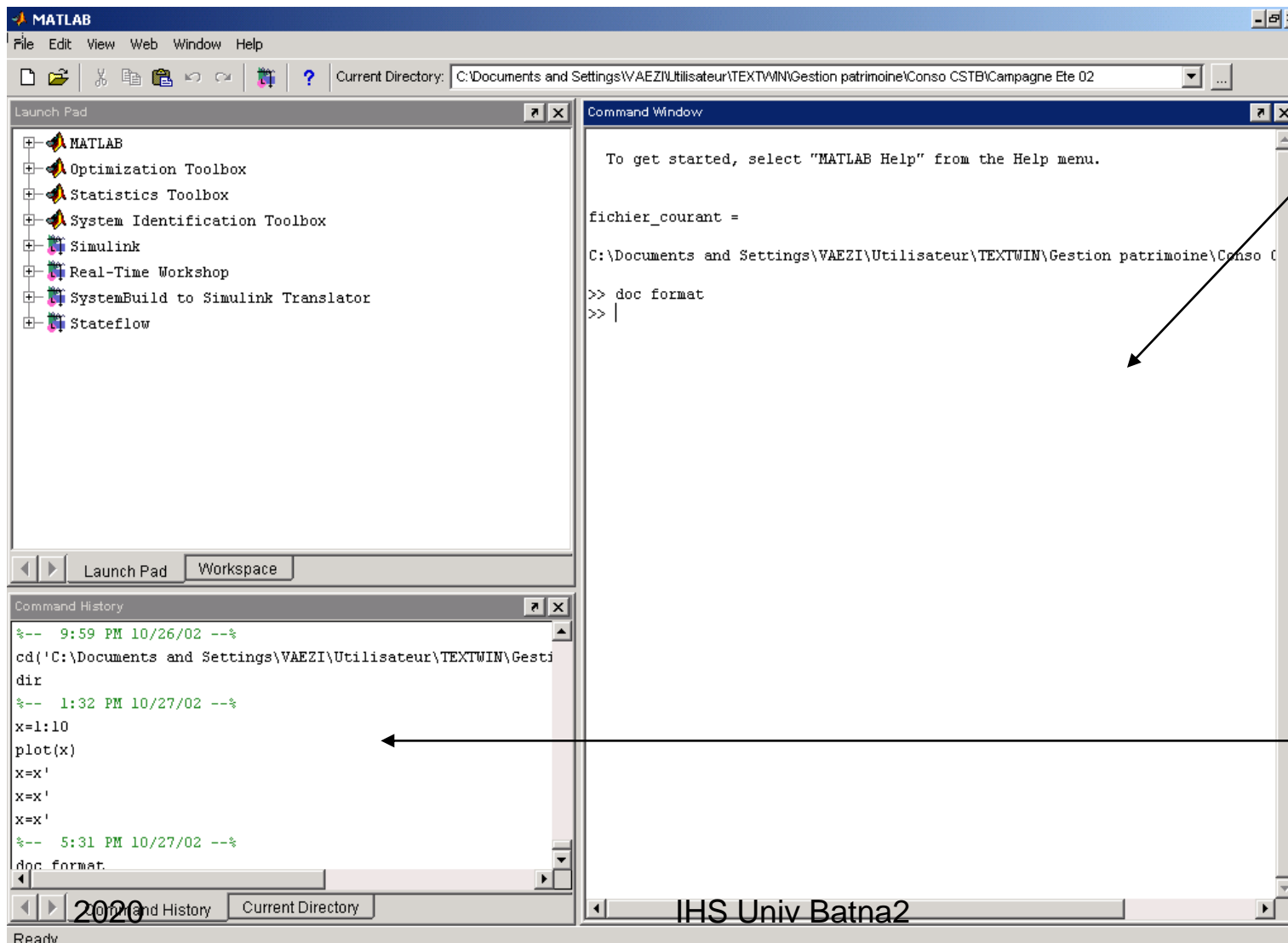
- **Quelles sont les particularités de MATLAB ?**

MATLAB permet le travail soit en mode commande, soit en mode programmation ; tout en ayant toujours la possibilité de faire des visualisations graphiques. Considéré comme un des meilleurs langages de programmations (C ou Fortran), MATLAB possède les particularités suivantes par rapport à ces langages :

- la programmation facile,
- la gamme étendue des nombres et leurs précisions,
- la bibliothèque mathématique très compréhensive (rec2pol(x,y)),
- l'outil graphique qui inclus les fonctions d'interface graphique et les utilitaires,
- la possibilité de liaison avec les autres langages classiques de programmations (C ou Fortran).
- Dans MATLAB, aucune déclaration n'est à effectuer sur les nombres. En effet, il n'existe pas de distinction entre les nombres entiers, les nombres réels, les nombres complexes et la simple ou double précision (En Fortran par exemple, une subroutine est presque nécessaire pour chaque variable simple ou double précision, entière, réelle ou complexe).

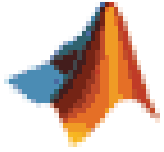


Presentation de Matlab



Fenetre de
commande

Fenetre
Historique



Presentation de Matlab

1. Démarrage de MATLAB

Pour lancer l'exécution de MATLAB :

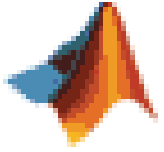
sous Windows, il faut cliquer sur Démarrage, ensuite Programme, ensuite MATLAB,

L'invite '>>' de MATLAB doit alors apparaître, à la suite duquel on entrera les commandes.

La fonction "**quit**" permet de quitter MATLAB :

```
>>quit
```

La commande "**help**" permet de donner l'aide sur un problème donné.



Presentation de Matlab

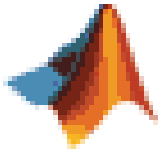
Exemple :

```
>> help cos  
COS Cosine.  
COS(X) is the cosine of the elements of X.
```

clock : affiche l'année, le mois, le jour, l'heure, les minutes et les secondes.

```
>>clock  
ans =  
1.0e+003 *  
2.0090 0.0200 0.0020 0.0210 0.0120 0.0098
```

```
>>date  
ans =  
02-Feb-2009
```



Presentation de Matlab

Calcul en mode Commande dans MATLAB :

Soit à calculer le volume suivant : où $R=4\text{cm}$

Pour calculer V , on exécute les commandes suivantes :

```
>>R=4
```

```
R =
```

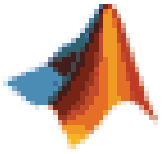
```
4
```

```
>>V=4/3*pi*R^3
```

```
V =
```

```
268.0826
```

(Ici, $\text{pi}=\pi$).



Presentation de Matlab

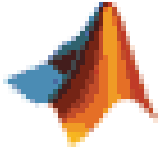
Symboles des Calcul arithmétique :

+ plus

- moins

/ division

* multiplication



Presentation de Matlab

Test 'if' Ce test s'emploie, souvent. Un test '*if*' est toujours suivi par un '*end*'.

Exemple :

```
>>V=268.0826
```

```
V =
```

```
268.0826
```

```
>>if V>150, surface=pi*R^2, end
```

```
surface =
```

```
50.2655
```

L'opérateur 'égal' (==) dans 'if' Il est noté (ou symbolisé) par '**==**'.

Exemple :

```
>>R=4
```

```
R =
```

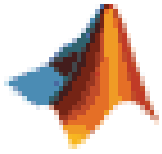
```
4
```

```
>>if R==4, V=4/3*pi*R^3;end
```

L'opérateur 'ou' Il est noté (ou symbolisé) par '**|**'

Exemple : Si R=4 ou m=1, alors

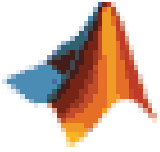
```
>>if R==4 | m==1, V=4/3*pi*R^3;end
```



Presentation de Matlab

Autres opérateurs :

<= inférieur ou égal > supérieur à	.* produit élément par élément de matrices
< inférieur à	.^ puissance élément par élément de matrices
>=supérieur ou égal	./ division élément par élément de matrices
<=inférieur ou égal	xor OU exclusif (XOR)
Error affiche le message : 'error'	message spécifié, émet un 'bip' et interrompt l'exécution du programme



Presentation de Matlab

Exemples :

Si $g > 2$ ou $g < 0$, alors $a = 4$

```
>> if g>2 |g<0, a=4, end
```

Si $a > 3$ et $C < 0$, alors $b = 15$

```
>> if a>3 & c<0, b=15, end
```

Les opérateurs ‘&’ et ‘|’ peuvent être utilisés dans la même chaîne :

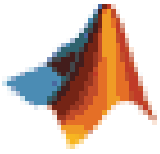
```
>> if (a==2 | b==3)&(c<5),g=1,end
```

L’opérateur ‘if.....else.....elseif.....end’:

Exemples :

```
>> R=2, if R>3, b=1 ; elseif R==3, b=2, else b=0, end
```

l’instruction ‘elseif’ peut être répétée dans un programme autant de fois.



Presentation de Matlab

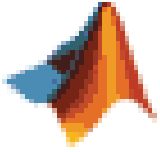
Variables et noms de variables :

Les variables et les noms de variables n'ont pas à être déclarés, car dans MATLAB, il n'y a aucune distinction entre variable 'entière', variable 'réelle' ou variable 'complexe'.

$\text{eps} = 2.2204 \cdot 10^{-16}$

Variables complexes :

Traditionnellement, en *Fortran* les variables i, j, k, l, m et n sont réservées aux variables entières. Dans MATLAB, i et j sont réservées aux unités imaginaires. Mais, on peut également les utiliser comme d'autres variables (entières ou réelles) si on les précise.



Presentation de Matlab

Opérateurs 'for/end' et 'while/end'

Exemples :

```
>>for R=1 :5, V=4/3*pi*R^3; disp([R,V]), end
```

Dans ce cas, R varie de 1 à 5, et la commande "`disp([R,V])`" retourne la matrice suivante :

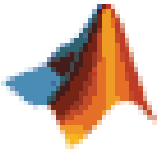
```
[R=1 :5, V (V(1) :V(5))]
```

On peut définir également l'instruction 'length' qui représente la taille de la variable. En effet, dans ce cas, on a :

`length(R)=5 ; (R=1 :5)` et `length(R)-1=4` (4 intervalles de pas 1).

```
>>while R<5, R=R+1 ; V=4/3*pi*R^3; disp([R,V]), end
```

`while` exécute l'instruction qui suit tant que le test logique est vrai.



Presentation de Matlab

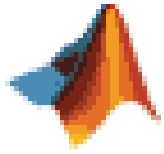
Exemple de pas dans la boucle for :

```
>>for R=5 :-1 :1, V=4/3*pi*R^3; disp([R,V]), end
```

Ici, le pas utilisé est dégressif (= -1). On peut utiliser les imbrications de 'for' autant de fois que l'on souhaite.

Exemple :

```
>> for i=0 :10, for j=1 :5, V=4/3*pi*R^3;disp([R,V]);end,end
```



Presentation de Matlab

break :

Interrompt l'exécution d'un 'for' ou d'un 'while'

goto :

Dans MATLAB l'instruction 'goto' est remplacée par 'break'.

Exemple :

```
while R==1
```

```
    . if x>Xlimite, break, end
```

```
    .
```

```
    .
```

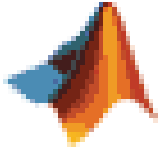
```
end
```

clear :

Efface toutes les variables existantes en mémoire

clc :

Efface l'écran (fenêtre) de MATLAB



Presentation de Matlab

Utilisation de nombres ou de variables complexes :

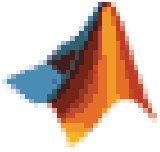
MATLAB, peut utiliser et gérer les variables ou les nombres complexes. La plupart des fonctions implicites définies pour les réels existent pour les complexes, y compris la puissance.

Exemple :

```
>>z=3.5-1.25i ;
>>log(z)
ans =
1.3128 - 0.3430i
>>cos(2-i)
ans =
-0.6421 + 1.0686i
:
>>i^2
ans =
-1.0000 + 0.0000i
```

Soit z un nombre complexe. Son conjugué est donné par la fonction 'conj(z)'.

```
>>z=3.5-1.25i
z =
3.5000 - 1.2500i
>>conj(z)
ans =
3.5000 + 1.2500i
```



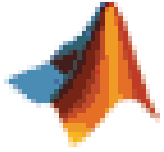
Presentation de Matlab

Module et argument d'un nombre complexe :

Dans MATLAB, les fonctions '*abs*' et '*angle*' permettent l'obtention directe du module et de l'argument d'un nombre complexe. L'angle theta est en radians.

Exemple :

```
>>z1
z1 =
3.5000 - 1.2500i
>>r=abs(z1)
r =
3.7165
>>theta=angle(z1)
theta =
-0.3430
```



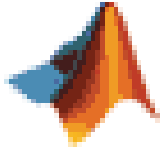
Presentation de Matlab

Objet de base : la matrice

L'objet de base de Matlab est la matrice, donc un tableau de réels ou de complexes.

Il n'est pas nécessaire de déclarer préalablement la taille, ni la nature de l'objet, sa définition suffit. Pour définir une matrice, on introduit ses coefficients entre des crochets. Le saut de ligne est fait par un point virgule. Ainsi pour obtenir la matrice suivante :

```
A =  
[1 2 3  
4 5 6  
7 8 9]
```



Presentation de Matlab

il faut exécuter :

```
A=[1 2 3 ; 4 5 6 ; 7 8 9]
```

Une matrice peut se définir par concaténation horizontale, ou verticale, d'autres

matrices. La matrice précédente peut donc être obtenue par :

```
A1=[1 2 3;4 5 6]
```

```
A2=[7 8 9]
```

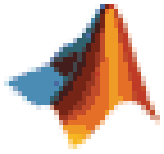
```
A=[A1;A2]
```

ou

```
A1=[1 2;4 5;7 8]
```

```
A2=[3;6;9]
```

```
A
```



Presentation de Matlab

il faut exécuter :

```
A=[1 2 3 ; 4 5 6 ; 7 8 9]
```

Une matrice peut se définir par enchainement horizontale, ou verticale, d'autres matrices. La matrice précédente peut donc être obtenue par :

```
A1=[1 2 3;4 5 6]
```

```
A2=[7 8 9]
```

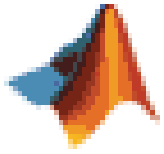
```
A=[A1;A2]
```

ou

```
A1=[1 2;4 5;7 8]
```

```
A2=[3;6;9]
```

```
A=[A1 A2]
```



Presentation de Matlab

La composante de A , située sur la ligne i , et la colonne j , est désignée par $A(i,j)$. La numérotation des lignes et des colonnes commence à 1. Pour extraire une sous matrice, on peut indiquer les indices des lignes et des colonnes sélectionnées dans un vecteur, ou avec ':'. Pour extraire les 2e et 3e lignes et la 1ere colonne de A on peut exécuter :

```
L=[2 3]
```

```
C=1
```

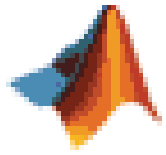
```
A(L,C)
```

ou

```
A(2:3,1)
```

Pour sélectionner toutes les colonnes (idem pour les lignes) il suffit de mettre ':' sans préciser la première ni la dernière. Pour obtenir la dernière ligne de A , il suffit

```
d'exécuter : A(3,:)
```

Presentation de Matlab

Pour générer un vecteur ligne de valeurs comprises entre a et b espacées par un pas fixe p, on utilise le symbole ':'. Par exemple pour obtenir les nombres pairs (pas de 2), entre 0 et 14, on exécute la commande ci-dessous.

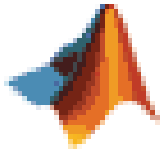
```
x = 0 : 2 : 14
```

```
> x=[0 2 4 6 8 10 12 14]
```

Par défaut, le pas est fixé à 1.

```
x = 0 : 4
```

```
> x=[0 1 2 3 4]
```



Presentation de Matlab

Quelques fonctions liées aux matrices:

`size(A)` : renvoie un vecteur ligne indiquant les dimensions de A .

`length(v)` : renvoie la longueur du vecteur v .

`eye(n)` : définit la matrice unité, de dimension $n \times n$.

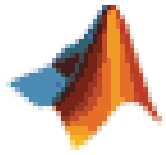
`ones(m,n)` : définit une matrice de dimension $m \times n$, dont toutes les composantes sont égales à 1.

`zeros(m,n)` : définit une matrice de dimension $m \times n$, dont toutes les composantes sont égales à 0.

`rand(m,n)` : définit une matrice de dimension $m \times n$, dont les composantes sont choisies aléatoirement entre 0 et 1.

A' : transpose la matrice A .

`inv(A)` : permet d'inverser la matrice A (résolution de systèmes de n équations à n inconnues).



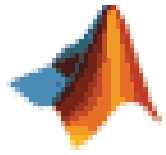
Presentation de Matlab

$\exp(A)$: calcule l'exponentielle de la matrice A .

A^n : calcule la puissance n de A : A^n .

$\min(v)$: calcule la valeur minimale des composantes du vecteur v .

$\max(v)$: calcule la valeur maximale des composantes du vecteur v .



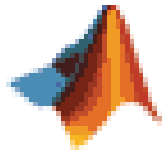
Presentation de Matlab

Écriture de fonctions personnelles :

Dans MATLAB, les programmes qui sont sauvegardés comme des fichiers `.m` sont équivalentes à des sous-programmes et des fonctions dans d'autres langages.

Fonction retournant une seule variable :

$$f(x) = \frac{2x^3 + 7x^2 + 3x + 1}{x^2 - 3x + 5e^{-x}}$$



Presentation de Matlab

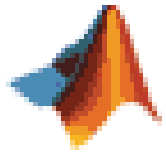
Supposons que le programme MATLAB correspondant est sauvegardé sous le nom `demof_.m`. ce programme est le suivant :

```
function y=demof_(x)
```

```
y=(2*x.^3+7*x.^2+3*x-1)/(x.^2-3*x+5*exp(-x));
```

Pour déterminer $f(x=3)$ par exemple, il suffit d'exécuter la commande : "`y=demof_(3)`", et la réponse sera :

```
>>y=  
502.1384
```



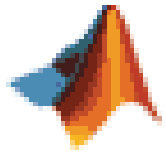
Presentation de Matlab

Fonction retournant plusieurs variables :

Une fonction peut retourner plus qu'une variable.

Considérons la fonction qui permet d'évaluer la moyenne et l'écart type de données.

Pour retourner les 2 variables, un vecteur est utilisé du côté gauche de la fonction, comme l'illustre le cas suivant :



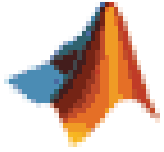
Presentation de Matlab

```
function [mean, stdv]=mean_st(x)
    n=length (x) ;
    mean=sum(x)/n ;
    stdv=sqrt(sum(x.^2)/n-mean.^2) ;
```

Exemple :

```
>>x=[1 5 3 4 6 5 8 9 2 4];
>>[m,s]=mean_st(x)
```

La réponse est : $m=4.7000$ et $s=2.3685$



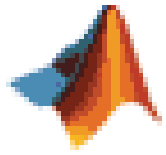
Presentation de Matlab

Écriture d'un programme MATLAB :

En MATLAB, les programmes se terminent par une extension '*.m*' dans le nom du fichier programme. Aucune compilation n'est à faire avant l'exécution du programme. Au cours de l'exécution, un message d'erreur apparaît et indique les lieux où se trouvent les erreurs. Pour lancer l'exécution du programme, il faut se mettre toujours dans le même répertoire où se trouve ce programme.

Exemple :

ce dernier se trouve dans `c:\utilisateur` ; il faut changer tout d'abord de répertoire après avoir lancé MATLAB en tapant :
`"cd c:\utilisateur"` ou en indiquant à Matlab son chemin d'accès à l'aide de File/Set Path.

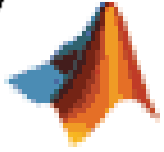


Presentation de Matlab

Génération de graphique avec MATLAB

MATLAB est un outil très puissant et très convivial pour la gestion des graphiques, que ce soit en une dimension, en deux dimensions ou en trois dimensions. Pour tracer une courbe $y=\sin(x)$ par exemple, où $x=0 :50$; il suffit de faire :

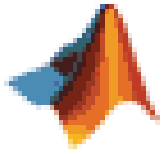
```
>>x=0:50;y=sin(x);plot(x,y)
```

$M \begin{pmatrix} 2 \\ M \begin{pmatrix} 2 \\ 4 \end{pmatrix} \end{pmatrix}$ 

Presentation de Matlab

résumé très succinct concernant le traçage des graphiques et la manipulation des axes et des échelles

- `xlabel('temps')` pour donner un titre à l'axe x,
- `ylabel('vitesse')` pour donner un titre à l'axe y,
- `title('évolution de la vitesse')` pour donner un titre au graphique,
- `text(2,4,'+++Température T1')` au point (2,4) , écrire la légende de la courbe tracée avec "+++",

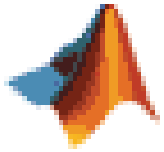


Presentation de Matlab

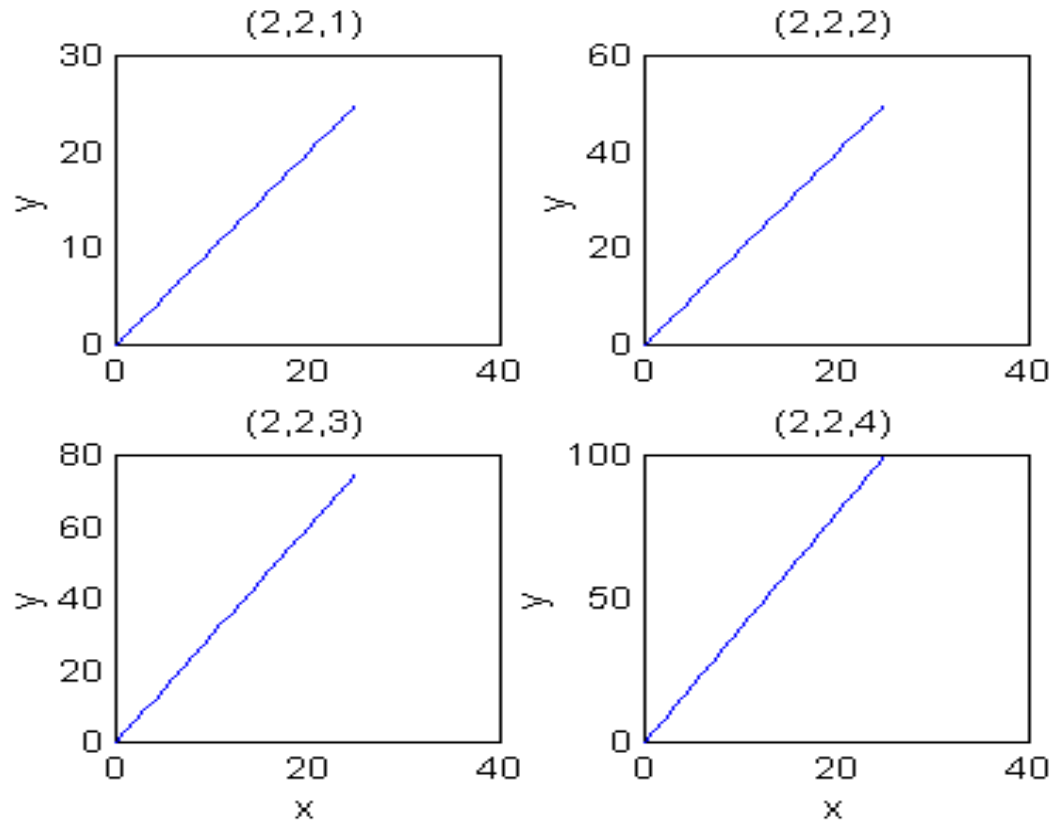
`subplot(m,n,k)` à tracer $m \times n$ graphiques séparés dans la feuille, où k est le numéro de chaque graphiques.

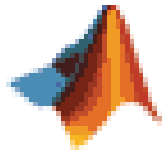
Exemple :

```
x=0 :25 ;y=x ;  
    subplot(2,2,1),plot(x,y) ;  
ylabel('y') ;title('(2,2,1)') ;  
    subplot(2,2,2),plot(x,2*y) ;  
ylabel('y') ;title('(2,2,2)') ;  
    subplot(2,2,3),plot(x,3*y) ;  
xlabel('x') ;ylabel('y') ;title('(2,2,3)') ;  
    subplot(2,2,4),plot(x,4*y) ;  
xlabel('x') ;ylabel('y') ;title('(2,2,4)') ;
```



Presentation de Matlab

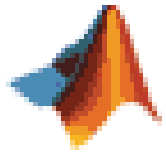




Presentation de Matlab

Pour définir une variable comp valant 1 si x est supérieur à 5, valant 0 si $x = 5$, et -1 sinon :

```
if x>5
    comp=1
elseif x==5
    comp=0
else
    comp=-1
end
```



Presentation de Matlab

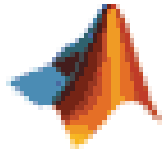
Pour faire la somme des k^2 , pour $k = 1, 2, \dots, 100$:

```
somme=0
```

```
for k=1:1:100
```

```
    somme=somme+k^2
```

```
end
```



Presentation de Matlab

Pour calculer cette même somme jusqu'à ce qu'elle dépasse la valeur 1000

```
somme=0
```

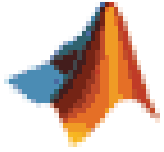
```
k=0
```

```
while somme < 1000
```

```
    k=k+1
```

```
    somme=somme+k^2
```

```
end
```



Presentation de Matlab

Manipulation du logiciel

- **Après avoir démarré Matlab, défini un répertoire courant de travail, et l'avoir inclus**
- **dans le chemin d'accès, faire les opérations suivantes :**
- **Créer une matrice A de dimension 3×3 , et définir des sous matrices (par exemple sélectionner la première ligne et les deux dernières colonnes.**
- **Définir la matrice $B = A^T$, et les multiplier entre elles (vérifier que le produit matriciel n'est pas nécessairement commutatif).**
- **Définir un vecteur t, dont la première composante vaut 0, la dernière vaut 10, avec un pas de 0.1. Définir un vecteur y, dont chaque composante $y(k)$ est donnée par $y(k) = \sin(x(k))$. Cela peut se faire par une boucle for, ou par une commande unique sur y.**
- **Tracer $\sin(t)$ en fonction de t, pour $t \in [0, 4_]$. Donner un titre au graphique, et aux axes. Superposer ce graphique avec d'autres tracés (sinusoïdes de fréquences différentes par exemple).**
- **Faire une fonction qui trace la fonction dans une figure la fonction $\cos(5t)$. La fonction aura pour paramètres d'entrée : t_0 la valeur minimale de t, t_f : la valeur finale de t, et nb le nombre de points de la courbe. Les paramètres de sortie sont les vecteurs x et y contenant respectivement les valeurs de t et de $\sin(5t)$.**