

1) Notion de base de MATLAB

Les opérations arithmétiques de base de MATLAB sont :

+ pour l'addition -Pour la soustraction *Pour la multiplication / ou \ pour la division. Mais la division à droite et à gauche ne donnent pas les mêmes résultats, se sont donc deux opérations différentes
a/b c'est a qui divise b

```
>> 4/3
ans =
    1.3333
```

Par contre a\b c'est b qui divise a

```
>> 4\3
ans =
    0.7500
```

Le moyen le plus simple de saisir un vecteur est d'entrer ses éléments en les séparant par des blancs

```
>> x=[6 4 5]
x =
    6    4    5
```

Afin d'éviter l'affichage du résultat d'une expression quelconque, nous terminons celle-ci par un point virgule

```
>> x=[6 4 5];
```

Autre façon de saisir un vecteur ligne

```
>> x=[6,4,5];
```

Ce vecteur est considéré comme une matrice à une ligne et trois colonnes

```
>> size(x)
ans =
    1    3
```

Les dimensions d'un tableau quelconque peuvent être récupérées sous forme d'un vecteur [m,n], m et n étant respectivement le nombre de lignes et de colonnes

```
>> [m,n]=size(x)
m =
    1
n =
    3
```

La longueur d'un tableau quelconque est, par définition, sa plus grande dimension

```
>> longueur_x=length(x)
Longueur_x =
    3
```

L'addition et la soustraction de vecteurs de mêmes dimensions se font élément par élément

```
>> x=[0 4 3];
>> y=[2 5 7];
>> x-y
ans =
   -2   -1   -4
>> x+y
ans =
    2    9   10
```

La transformation d'un vecteur ligne en un vecteur colonne et inversement, sera réalisé à l'aide de l'opérateur de transposition « ' »

```
>> tx=x'
tx =
    0
    4
    3
```

Le produit d'un vecteur par sa transposée donne le carré de la norme de celui-ci

```
>> x*tx
ans =
    25
```

Le produit d'un vecteur colonne de taille n par un vecteur ligne de taille m donne une matrice de dimension (n,m)

```
>> tx*y
```

```
ans =  
 0  0  0  
 8 20 28  
 6 15 21
```

2) Matrices particulières
Matrice identité

```
>> ident=eye(3)  
ident =  
 1  0  0  
 0  1  0  
 0  0  1
```

Matrice nulle

```
>> Matrice_nulle=zeros(2,3)  
Matrice_nulle =  
 0  0  0  
 0  0  0
```

Matrice unité

```
>> Matrice_unite=ones(3,2)  
Matrice_unite =  
 1  1  
 1  1  
 1  1
```

Matrice aléatoire

```
>> Matrice_aleatoire=rand(2,3)  
Matrice_aleatoire =  
 0.8147  0.1270  0.6324  
 0.9058  0.9134  0.0975
```

3) Les polynômes

Le polynôme p d'expression $p(x) = x^2 - 6x + 9$ est représenté par le tableau à une dimension

```
>> p=[1 -6 9]  
p =  
 1 -6 9
```

Nous pouvons déterminer les racines du polynôme à l'aide de la fonction roots

```
>> roots(p)  
ans =  
 3.0000 + 0.0000i  
 3.0000 - 0.0000i
```

Pour évaluer un polynôme en un point, nous utilisons la fonction polyval

```
>> polyval(p,1)  
ans =  
 4
```

4) Représentation graphique

Pour tracer la représentation graphique d'un polynôme $K(x) = x^3 - 6x^2 + 11x - 6$

Définissons un domaine pour la variable x qui contient les racines de K . Ensuite, on calcule y les valeurs du polynôme $K(x)$ aux différents point de x

```
>>x=0:0.1:4 ;  
>>K=[1 -6 11 -6] ;  
>>y=polyval(K,x);  
>>plot(x,y);
```

Les courbes tracées sur l'écran peuvent avoir des titres sur les axes, un titre du graphique et des lignes graduées
Dans le cas où nous ne précisons pas l'axe des abscisses, MATLAB le génère automatiquement

```
>> x=0:10:100  
>> y=[4 4 6 9 13 17 22 30 37 51 56]  
>> plot(y)  
>> title('Caractéristique d"une vanne');grid
```

La fonction hold permet de superposer 2 courbes

```
>>x=0:0.1:1 ;  
>>K=[1 0 0] ;  
>>y=polyval(K,x);  
>>plot(x,y,'r');
```

```
>>hold on
>>K=[1 0] ;
>>y=polyval(K,x);
>>plot(x,y,'g')
>>hold off
```

Hold on signifie la mise en marche de la figure courante. Cette fonction permet l'affichage de plusieurs courbes sur la même figure

Hold off signifie l'arrêt de la figure courante. Attention si vous oubliez de faire cette fonction toutes les courbes seront affichées sur la même figure

Avec MATLAB, il est possible de tracer plusieurs courbes graphiques dans une même fenêtre à l'aide de la commande subplot en divisant cette dernière en plusieurs zones

Subplot(m,n,p) divise la fenêtre graphique en m ligne et n colonnes et trace le graphique qui suit dans la zone de numéro p

Par exemple si nous voulons tracer les fonctions précédentes sur 2 fenêtres positionnées en lignes, nous procédons comme suit :

```
>>x=0:0.1:1 ;
>>K=[1 0 0] ;
>>y=polyval(K,x);
>>subplot(211);
>>plot(x,y);
>>K=[1 0] ;
>>y=polyval(K,x);
>>subplot(212);
>>plot(x,y)
```

Par exemple si nous voulons tracer les 2 courbes sur 2 fenêtres positionnées en colonne, nous procédons comme suit :

```
>>x=0:0.1:1 ;
>>K=[1 0 0] ;
>>y=polyval(K,x);
>>subplot(121);
>>plot(x,y);
>>K=[1 0] ;
>>y=polyval(K,x);
>>subplot(122);
>>plot(x,y)
```

5) L'outil Simulink

Comme extension de MATLAB, SIMULINK comprend plusieurs caractéristiques des systèmes dynamiques. Nous pouvons considérer SIMULINK comme une version graphique de MATLAB. Ainsi, avec SIMULINK, pour intégrer un modèle il suffit de dessiner le schéma bloc correspondant à ce système

Pour démarrer **SIMULINK**, il suffit de taper la commande "**simulink**" sous environnement **MATLAB** ou

cliquer sur l'icône suivante  dans la fenêtre de **MATLAB**.

Les blocs usuels sont :

