

Prise en main de MATLAB

MATLAB (MATrix LABoratory) est un logiciel de calcul scientifique, c'est avant tout un programme de calcul matriciel. MATLAB est un interpréteur : un programme (ou une commande) un fois appelée reste en mémoire.

MATLAB affiche une fenêtre de commande dans laquelle on peut taper les commandes à exécuter.

Pour MATLAB toute variable est considérée comme un tableau d'éléments d'un type donné. MATLAB différencie trois formes particulières de tableaux. Les scalaires qui sont des tableaux à une ligne et une colonne. Les vecteurs qui sont des tableaux à une ligne et plusieurs colonnes. Les matrices qui sont des tableaux ayant plusieurs lignes et plusieurs colonnes. Ces tableaux peuvent être à éléments réels ou complexes.

Attention, il faut respecter les minuscules et les majuscules dans la dénomination des variables. Les fonctions MATLAB sont toujours définies en minuscules. Attention à ne pas utiliser les termes i et j car ces constantes sont initialisées dans MATLAB comme les variables complexes.

Pour nettoyer l'espace de travail il faut taper la commande `clear all`. Il est possible d'effacer seulement une variable en tapant la commande `clear nom-de-la-variable`. Il est possible également d'obtenir une aide en ligne sur une commande en tapant : `help nom de la commande`. La présence d'un `;` à la fin de la ligne de commande supprime l'affichage du résultat d'une opération. Le symbole `%` permet d'écrire des commentaires dans le programme.

1 Les vecteurs

Un vecteur est défini en donnant sa liste d'éléments entre crochets (`[]`). Ces éléments sont séparés par des espaces ou des virgules. On définit un vecteur colonne en donnant la liste de ses éléments séparés par des points virgules (`;`). `x=x(:)` transforme un vecteur ligne en vecteur colonne. Un vecteur colonne peut être transformé en vecteur ligne et réciproquement en tapant `x'` (`'` est le symbole de transposition).

On peut accéder à une composante d'un vecteur, par exemple le $k^{\text{ème}}$ élément est indentifié par `x(k)`. Attention le premier élément a obligatoirement comme indice 1.

La longueur d'un vecteur peut être obtenue par la commande `length`. Taper la commande `length(z)`

vecteurs spéciaux

Certains vecteurs sont prédéfinis dans MATLAB:

ones(1,n) est un vecteur ligne de 1 de longueur n.

ones(m,1) est un vecteur colonne de 1 de longueur m.

zeros(1,n) est un vecteur ligne de 0 de longueur n.

zeros(m,1) est un vecteur colonne de 0 de longueur m.

rand(1,n) est un vecteur ligne de longueur n dont les valeurs sont générées de manière aléatoires entre 0 et 1.

rand(m,1) est un vecteur colonne de longueur m dont les valeurs sont générées de manière aléatoires entre 0 et 1.

2 Les matrices

Une matrice est définie en donnant la liste de ses éléments entre crochets. Les éléments d'une ligne peuvent être séparés par des blancs ou par une virgule. Les lignes peuvent être séparées par un point virgule ou un retour chariot.

Un élément d'une matrice est référencé par ses numéros de ligne et de colonne: $A(2,3)=6$
la commande size permet de connaître la dimension de la matrice : $\text{size}(A) = 2 \ 3$

matrices spéciales

Certaines matrices sont prédéfinies dans MATLAB :

eye(n) est la matrice identité (n,n)

ones(m,n) est la matrice à m lignes et n colonnes dont tous les éléments sont des 1

zeros(m,n) est la matrice à m lignes et n colonnes dont tous les éléments sont des 0

rand(m,m) est la matrice à m lignes et n colonnes dont les éléments sont générés de manière aléatoire entre 0 et 1.

Les fonctions arrondies sont :

round(x) : entier le plus proche de x

floor : arrondi par défaut

ceil(x) : arrondi par excès

Pour obtenir une matrice (3,3) de nombre aléatoire de 1 et de 0, on peut utiliser la commande $\text{floor}(2*\text{rand}(3,3))$. On multiplie par deux les valeurs de la matrice rand (3,3) et on effectue l'arrondi par défaut à l'entier le plus proche par la commande floor .

Le $j^{\text{ème}}$ vecteur colonne de la matrice D est écrit $D(:,j)$.

Les symboles + , - , * , ^ sont respectivement les opérations d'addition de soustraction de multiplication et d'exponentiation sur les matrices.

3 Programmation

Des que le calcul à effectuer entraîne une suite de commandes un peu compliquée, il est nécessaire de sauvegarder ces commandes dans un fichier. Une suite de commandes peut être regroupée dans un fichier texte nom.m sous le répertoire C/MATLAB/WORK. Il faut taper la commande nom dans la fenêtre de commande pour exécuter le fichier. Une fenêtre d'édition de fichier peut également être ouverte.

Les instructions de contrôle sont très proches de celles existant dans d'autres langages de programmation.

La boucle for (for ...end) permet de répéter une séquence d'instructions pour les valeurs d'un indice, incrémenté à chaque itération, variant entre deux bornes données. On peut également utiliser des boucles for imbriquées.

```
for t=1 : 50
at=2*0.01*t
x(t)=5-cos(at+pi/2)
end
```

La boucle while (while ...end) effectue une suite de commande jusqu'à ce qu'une condition soit satisfaite.

L'instruction if .. elseif..else...end permet de choisir plusieurs options en fonction d'une d'expressions logiques.

4 Les graphiques

On utilise l'instruction plot pour tracer des courbes en deux dimensions :

plot(x,t) : Tracer de x en fonction de t

plot(t,x,t,y) permet de tracer des courbes x et y en fonction de t sur le même graphique.

graphique multiple

On peut tracer plusieurs graphiques dans la même fenêtre en utilisant l'instruction subplot pour diviser la fenêtre en plusieurs parties.

Pour diviser la fenêtre en deux parties (basse et haute), on utilise les commandes subplot(2,1,1) pour la fenêtre du haut et subplot(2,1,2) pour la fenêtre du bas.

De même, on peut diviser la fenêtre en fenêtre droite et gauche par les commandes subplot(1,2,1) et subplot(1,2,2).

Pour diviser la fenêtre en quatre parties on utilise alors les commandes subplot(2,2,1) subplot(2,2,2) subplot(2,2,3) subplot(2,2,4).

On peut ainsi diviser la fenêtre graphique en partie égale en jouant sur indices de la commande subplot(m,n,i) où

m est le nombre de sous fenêtre verticalement

n est le nombre de sous fenêtre horizontalement

i sert à spécifier dans quelle sous fenêtre doit s'effectuer l'affichage. Les fenêtres sont numérotés de gauche à droite et de haut en bas..

On peut ajouter du texte au graphique en utilisant les commandes:

title('titre du graphique') permet de donner un titre au graphique

xlabel('temps') permet de donner une étiquette à x

ylabel('signal ') permet de donner une étiquette à y

gtext('graphique') permet d'ajouter un texte au graphique à l'aide de la souris.

Autres commandes

semilogx(f,A) permet de tracer A en fonction de f avec une échelle log(f)

polar(teta,r) permet de tracer r en fonction de teta en coordonnées polaires ...

grid permet d'ajouter une grille

axis([0 1000 -0.2 1.2]) permet de redéfinir les axes avec une nouvelle échelle, x allant de 0 à 1000 y étant compris entre -0,2 et 1,2.

Bibliographie :

Débuter avec MATLAB, Stéphane Balac, centre de mathématique INSA de Lyon.

Le traitement du signal sous MATLAB, André Quinquis, Hermes.