Présentation de Matlab

1 Présentation générale

Matlab est à la fois un logiciel de calcul et un langage de programmation haut niveau. C'est un logiciel payant, dont il existe deux équivalents gratuits

- **Octave** est un logiciel qui utilise le langage de matlab et peut donc utiliser les fonctions écrites en matlab. Il est plus lent et un peu moins beau.
- **Scilab** est développé par l'INRIA et la syntaxe diffère un peu de celle de matlab mais l'esprit est le même. Il est de mon point de vue encore un peu moins pratique que matlab.

Matlab est un logiciel de calcul numérique pas de calcul formel à la différence de Maple. Il ne sait résoudre que des équations numériques.
Le nom Matlab vient de Matrix Laboratory. En Matlab les objets sont tous par défaut des matrices. Une variable réelle est donc vu par Matlab comme une matrice 1×1. Le produit est donc par défaut un produit matriciel. Il faut se méfier.
Le type des variable n'est pas très important. Matlab peut additionner un booléen et un réel, multiplier un entier par un complexe sans problème.
Pour lancer Matlab, il suffit de lancer la commande *matlab* dans un terminal. S'ouvre alors la fenêtre principale de matlab. On peut y lancer directement des lignes de commande mais la plupart du temps on passera par l'éditeur de Matlab qui permet de réaliser des scripts et des fonctions.

2 Premières commandes

Dans la fenêtre principale tapez les lignes suivantes :

```
>>a=1;
>>b=3
```

La commande $=$ est la commande d'affectation et le point virgule permet de ne pas afficher le résultat de la commande.

```
>>c=a+2*b
```

donne le résultat attendu. Pour construire un vecteur ligne on utilise des crochets et des virgules :

```
>>X=[1,-9,13,0]
```

Pour faire un vecteur colonne on remplace les virgule par des points virgule :

```
>>Y=[1;4;7;2]
```

Essayer de prévoir ce vont donner les deux instructions suivantes :

```
>>Z=X*Y
>>W=Y*X
```

La construction d'une matrice se fait a aussi simplement :

```
>>A=[2,3;4,5]
```

La mutiplication de deux matrices ou vecteurs termes à terme se fait par la commande suivante :

```
>>B=A.*A
```

qui ne renvoit pas le même résultat que

```
>>C=A*A
```

La commande *size* renvoie les deux dimensions d'une matrice, nombre de lignes et nombre de colonne. La commande suivante permet de construire une matrice remplie de zéros :

```
>>D=zeros(2,3)
```

La commande *ones* renvoie une matrices remplie de 1.
Si vous avez un doute sur l'emploi d'une commande vous pouvez utiliser l'aide de matlab en tapant help suivi du nom de la commande :

```
>>help size
```

Attention, dans l'aide les commandes sont inscrites en majuscule mais il faut les taper en miniscule dans la fenêtre. Matlab est sensible à la casse.
On peut construire des vecteurs dont les composantes sont régulièrement espacées de la manière suivante :

```
>>a=0;b=1;d=0.1;
>>t=[a:d:b]
```

Question : En utilisant le produit terme à terme, construire un vecteur ligne contenant les carrés des entiers de 1 à 10.
La commande *rand* fournit des valeurs aléatoires distribuées selon une loi uniforme sur $[0,1]$. Lancer plusieurs fois la commande suivante

```
>>A=rand(5,5)
```

La commande *randn* permet d'obtenir des variables distribuées selon une loi gaussienne centrée et réduite.
On peut extraire une partie de la matrice facilement :

```
>>B=A(1:3,1:4)
```

On peut même ne sélectionner que certaines colonnes :

```
>>C=A(:,[1,3])
```

Les deux points servant à indiquer qu'on prend toutes les lignes et le vecteur $[1,3]$ qu'on ne prend que les colonnes indicées par 1 et 3.
Remarque : On peut multiplier une matrice par un nombre réel. La multiplication se fait alors terme à terme. On peut aussi additioner un nombre à une matrice, le nombre est alors ajouté à tous les coefficients de la matrice.
Question : Construire une matrice ayant 6 lignes et 3 colonnes et dont les coefficients sont uniquement des 4.
On peut renvoyer un booléen dans une variable :

```
>>t=3;
>>r=(t<2)
```

On peut le faire également sur un vecteur

```
>>t=[0,3,5,7,1.5,-4];
>>r=(t>2)
```

La valeur absolue d'un réel (et le module d'un complexe) sont calculés par la commande *abs*

```
>>b=abs(t)
```

2.1 Affichage

La commande d'affichage classique est plot :

```
>>t=[0:0.01:2*pi];
>>x=sin(t);
>>plot(x)
```

permet de visualiser la fonction sinus sur $[0,2\pi]$. Par défaut les abscisses sont les numéros des composantes du vecteur mais si on utilise

```
>>plot(t,x)
```

Matlab relie les points dont les abscisses sont dans t et les ordonnées sont dans x. Si on ne veut pas que les points soient reliés ou changer la couleur on peut utiliser

```
>>plot(t,x,'+r');
```

Voir l'aide de plot pour plus de détails. Pour afficher deux courbes sur la même figure on peut utiliser les commandes suivantes :

```
>>y=sin(2*t);
>>plot(t,x);hold on;plot(t,y,'r');hold off;
```

On peut également visualiser des images avec les commandes *image* et *imagesc*.
Remarque : si on demande à matlab d'afficher un vecteur complexe, il affiche la partie imaginaire en fonction de la partie réelle, ce qui donne souvent des figures surprenantes.
Pour ouvrir une nouvelle figure ; on utilise la commande *figure*.

2.2 Fonctions usuelles

- sqrt(x) : racine carrée de x,
- log(x) : logarithme népérien de x,
- exp(x) : exponentielle de x.
- mod(n,m) : n modulo m, reste de la division euclidienne de m par n.
- abs(z) : module de z,
- angle(z) : argument d'un nombre complexe z.
- real(z) : partie réelle de z,
- imag(z) : partie imaginaire de z,
- i ou j : nombre complexe $i = e^{i\pi/2}$, sauf si on change la valeur de i ou j dans une boucle par exemple,
- sort : permet de trier un vecteur (regarder l'aide), la commande sort renvoie également la permutation qui permet de passer du vecteur au vecteur ordonné. Il arrive souvent que seule cette permutation soit intéressante.
- length(A) : donne la plus grande dimension d'une matrice A,
- sum(V) : somme les composantes d'un vecteur V,
- diff(V) : renvoie un vecteur dont les composantes sont les différences des composantes consécutive du vecteur V,
- while ... end : permet de faire une boucle while,
- a==b : renvoie le booléen 1 si $a = b$ et 0 si $a \neq b$,
- A' : matrice transposée de A,
- floor(x) : partie entière de x.
- ceil(x) : partie entière supérieure de x.
- max(x) : maximum d'un vecteur x.
- min(x) : minimum d'un vecteur x.
- mean(x) : moyenne d'un vecteur x.

3 Editeur, scripts et fonctions

Dès qu'on veut enchainer des instructions de manière un peu évoluée en matlab on crée des programmes matlab. La manière le plus simple de le faire est de créer un script ou une fonction Matlab. Tout d'abord il est préférable de créer un répertoire de travail où seront sauvés les fichiers matlab. Vous pouvez vous déplacer dans l'arborescence de votre ordinateur à l'aide de la commande *cd* qui fonctionne comme sous linux dans la fenêtre principale de Matlab et créer un répertoire avec la commande *mkdir*. Ensuite on ouvre l'éditeur de matlab et on peut créer une fonction ou un script. Ces programmes matlab doivent être sauvés sous un nom avec l'extension .m .

3.1 Les fonctions

Une fonction matlab est un programme matlab qui prend en entrée un ensemble de variables (souvent des nombres, vecteurs ou matrices) et qui renvoie un autre ensemble de variables (souvent des nombres, vecteurs ou matrices.)
Remarque : les variables utilisées dans chaque fonction sont des variables locales. C'est à dire qu'elles peuvent avoir une valeur différentes voire un type différent qu'une variable de même nom dans le programme principal. Ainsi même si dans la fenêtre principale une variable a existe, on peut utiliser une variable a dans une fonction sans modifier la valeur de celle ci dans le programme principal. Ainsi on ne peut faire appelle dans une fonction qu'aux variables d'entrées de la fonction à moins d'utiliser l'instruction *GLOBALE*. Voici un exemple de fonction élémentaire :

```
function [s,d]=toto(a,b)
s=a+b;
d=a-b;
```

Cette fonction sera sauvé impérativement sous le nom *toto.m*. Il faut prendre l'habitude de sauver les fonctions sous le nom utilisé en première ligne.
Pour utiliser cette fonction on pourra taper dans la fenêtre principale :

```
>>[x,y]=toto(5,7)
```

Les variables de sorties sont séparées par des virgules et mises entre crochets et les variables d'entrée sont également séparées par des virgules et mises entre parenthèse.
Question : Créer et utiliser les fonctions suivantes en essayant de prévoir la valeur de sortie pour des valeurs d'entrées raisonnables.

1. ```
   function S=mystere(n)
   S=0;
   for k=1:n
   S=S+k;
   end
   ```
   ```

2.
```
function S=test1(n)
if n>5
   S=10;
else
   S=3;
end
```
3.
```
function T=Seuillage(V,a)
T=V.*(abs(V)>=a);
```

3.2 Les scripts

A la différence des fonctions les scripts sont des sériées d'instruction qui ne prennent pas de valeur d'entrée et pas de valeur de sortie. Lancer un script est équivalent à recopier les lignes de commande dans la fenêtre principale. Si une variable est modifiée par un script elle est modifiée après l'exécution du script. Il n'y a pas d'entête à un script.

4 Exercices d'application

1. Ecrire une fonction matlab Exo1.m qui prend en entrée en vecteur ligne V et deux entiers $a$et b et qui renvoie un vecteur identique à V en dehors des indices dans l'intervalle $[a,b]$ où le vecteur de sortie est nul :

   `function W=Exo1(V,a,b)`
2. Ecrire une fonction matlab Exo2.m permettant d'afficher sur la même figure avecdes couleurs différentes les fonctions x,x^2 et x^3 sur un intervalle $[a,b]$ et sur N points.
3. Ecrire une fonction matlab Exo3.m qui prend en entrée une matrice M et qui renvoie le vecteur V du nombre de changement de signes de chaque ligne de A. Par exemple Si A est définie par la commande $A = [2,3,-1;-4,5,-3]$ le nombre de changement de signe dans la première ligne est 1 et dans la deuxième ligne il est 2.
4. Soit W_n la suite de matrice définie par $W_0 = (1)$ et $W_n = \begin{pmatrix} W_n & W_n \\ W_n & -W_n \end{pmatrix}$.

   Ces matrices sont appelées matrices de Walsh. Ecrire une fonction matlab Exo4.m qui prend en entrée un entier k et qui renvoie la matrice W_k.
5. Qu'obtient on quand on multiplie W_k par elle même ?
6. Quelle est la particularité du vecteur formé du nombre de changements de signes des lignes de W_k ?
7. Ecrire une fonction Exo5.m qui ne conserve que les n plus grandes composantes en valeur absolue d'un vecteur V et qui met les autres à 0.