

Initiation à matlab

Créer un répertoire `matlab` et s'y placer. Lancer alors matlab à l'aide de la commande `matlab`.

I. Manipulation de vecteurs et matrices

On peut utiliser matlab comme une super-calculatrice qui opère sur des matrices (les nombres réels sont considérés comme des matrices 1×1). L'ajout d'un point-virgule en fin de ligne de commande supprime l'affichage.

<p>DÉFINITION DE VECTEURS ET MATRICES</p> <pre>>> x=[1,2,3] >> y=[4;5;6] >> z=[i,2;4,5]; >> z >> t=1:4;t >> u=0:0.1:1</pre> <p>MATRICES PARTICULIÈRES</p> <pre>>> id=eye(5,5) >> zz=zeros(5,3) >> zzz=zeros(size(y)) >> un=ones(6,2)</pre> <p>EXTRACTION DE COEFFICIENTS</p> <pre>>> x(2) >> z(1,2)</pre>	<pre>>> x(2:3) >> z(1:1,1:2) >> u(t) >> z(t(1:2),x(2))</pre> <p>OPÉRATIONS D'ALGÈBRE LINÉAIRE</p> <pre>>> z' >> z^2 >> x+y' >> z*y >> z\y</pre> <p>OPÉRATIONS COEFFICIENT PAR COEFFICIENT</p> <pre>>> x.^2 >> x./y' >> 1./x >> z+1</pre>
---	--

Exercice : construire les matrices A et B de taille 4×4 telles que

$$a_{ij} = 2 \text{ si } i \neq j \text{ et } a_{ii} = 1 \quad \text{et} \quad b_{ij} = j.$$

II. Programmation avec matlab

Il est possible d'utiliser des fichiers pour regrouper des instructions matlab. Pour les rédiger, on utilise un *éditeur de texte* : soit celui intégré à Matlab, soit un éditeur externe comme NEdit ou emacs ; des commandes matlab permettent d'utiliser ces fichiers.

<p>EXÉCUTION DE SCRIPTS</p> <p>Créer le fichier <code>script.m</code> contenant les instructions suivantes</p> <pre style="margin-left: 40px;">n = 5; A = zeros(n,n); for i = 1 : n A(i,i) = i; end</pre> <p>L'exécution dans matlab s'effectue comme suit</p> <pre>>> script</pre>	<p>UTILISATION DE FONCTIONS</p> <p>Créer le fichier <code>f.m</code> :</p> <pre style="margin-left: 40px;">function y = f(x) y = x.^3;</pre> <p>On utilise directement <code>f</code> dans matlab :</p> <pre>>> f([1,2,3])</pre> <p>Dans le cas des fonctions comme dans le cas des scripts, les fichiers doivent être contenus dans le répertoire courant.</p>
---	---

III. Outils graphiques

De puissants outils graphiques sont fournis par matlab ; on en donne ici quelques exemples.

GRAPHES DE FONCTIONS	GESTION DES AXES
<pre>>> x=linspace(0,4,100); >> y=sin(x);z=cos(x); >> plot(x,y) >> loglog(x,abs(y)) >> plot(x,z) >> hold on >> plot(x,y) >> clf >> plot(x,y,x,z) >> clf >> plot(x,y,'r',x,z,'b--')</pre>	<pre>>> axis off >> axis on >> axis equal >> grid on</pre>
	PARTITIONNEMENT DE LA FENÊTRE
	<pre>>> figure >> subplot(1,3,1) >> plot(x,y) >> subplot(1,3,2) >> plot(x,y,'r--') >> subplot(1,3,3) >> plot(x,y,'go')</pre>
TEXTES ET LÉGENDES	
<pre>>> legend('sinus','cosinus') >> xlabel('x') >> ylabel('f(x)') >> title('Graphes trigonometriques')</pre>	
	IMPRESSION DANS UN FICHER
	<pre>>> print -f1 -deps fic >> print -f2 -djpeg fic</pre>

IV. Utilisation de l'aide en ligne

Une aide en ligne est disponible grâce à l'instruction `help nom_de_commande`.

Exercice : on se donne une matrice $A=\text{rand}(5,5)$. En utilisant la commande `diag`, construire – en une seule ligne d'instructions – la matrice diagonale D telle que $D_{ii} = A_{ii}$.

Quand on ne connaît pas le nom exact de la commande, on peut utiliser l'instruction

```
>> lookfor mot_clef
```

qui effectue une recherche dans les entêtes des rubriques d'aide (en anglais).

Exercice : quelles commandes permettent de

- définir des points à l'aide de la souris ;
- calculer les valeurs propres d'une matrice ;
- changer le format d'affichage des nombres ;
- tracer un graphe en coordonnées polaires ;
- créer une matrice de Vandermonde ;
- calculer la longueur d'un vecteur ;
- ouvrir une boîte de dialogue pour entrer des paramètres ;
- calculer le temps d'exécution d'une tâche.