

Résumé de cours MATLAB

Introduction

Ce résumé de cours a pour but de fournir les bases nécessaires à l'utilisation du logiciel MATLAB. Le logiciel MATLAB (Matrix Laboratory) est un langage interprété : il propose des facilités de programmation et de visualisation, ainsi qu'un grand nombre de fonctions réalisant diverses méthodes numériques. Tous les objets définis dans MATLAB sont donc au moyen de vecteurs et de matrices/tableaux. Un ensemble important d'opérateurs et de fonctions MATLAB de base facilitent leur manipulation et des opérations comme par exemple le produit et l'inversion matricielles `inv`, la transposition (`'`) ou encore le calcul des valeurs propres (`eig`) font parties de la bibliothèque standard. D'autres fonctions servant à la création et à la manipulation de matrices et des tableaux (`diag`, `fliplr`, `flipud`, `rot90`, `rand`, `ones`, `zeros`, `linspace`, `tril`, `triu`, etc...) sont également disponibles.

MATLAB offre également plusieurs fonctions destinées à la résolution (numérique) d'équations différentielles linéaires ou non linéaires, notamment par la méthode de Runge-Kutta (`ode23` et `ode45`), l'intégration numérique (`trapz`, `quad` et `quad8`), la recherche des solutions d'équations algébriques (`roots`) ou transcendantes (`fzero`), la création et manipulation de polynômes (`poly`, `polyder`, `polyval`, `conv`, `deconv`)

Remarque 1 *une aide en ligne est disponible pour chaque commande de MATLAB. Il suffit d'introduire après l'invité de commande » :*

```
help nom_de_commande, exemple: help fzero; help poly
```

L'environnement MATLAB

l'interface de Matlab selon la version utilisée, par défaut on retrouve quatre fenêtres :

1. Command Window : est la fenêtre principale dans laquelle on tape les commandes directement dans l'interpréteur MATLAB.
2. Command History : garde la trace de toutes les commandes entrées par l'utilisateur, Il est donc possible de récupérer des instructions déjà saisies (et ensuite de les modifier dans le but de

les réutiliser)

```
→, ↓, ←, ←, permet de se déplacer dans les lignes de commandes
```

3. Current Directory : indique le répertoire courant ainsi que les fichiers existants
4. Workspace : indique toutes les variables et valeurs mises en mémoire avec leurs types, leur dimensions et leur nom.

Remarque 2 *l'interface Matlab peut changer selon la version, mais les principales fenêtres resteront identiques. La figure 1 illustre l'interface générale du Matlab.*

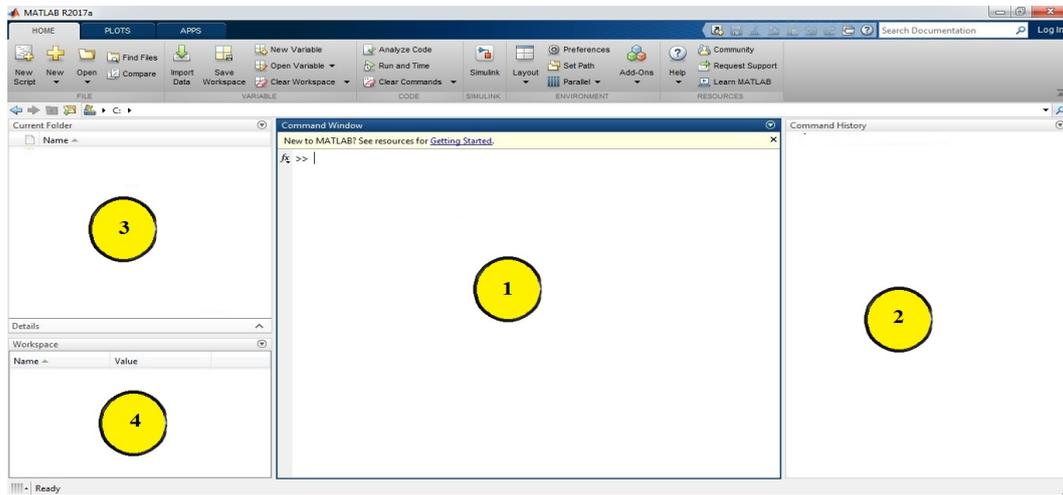


FIGURE 1.1 – Interface de Matlab

Editeur intégré à MATLAB

MATLAB possède un éditeur intégré que l'on appelle sélectionnant `File-New-Script`. De cette façon, l'utilisateur peut rajouter ses propres fonctions. Cette approche est définie en Matlab par les M-Files, qui sont des fichiers pouvant contenir les données, les programmes (scripts) ou les fonctions que nous développons. Pour créer un M-Files il suffit de taper la commande `edit`, ou tout simplement aller dans le menu : `File`→ `New` → `M-Files` (ou cliquer sur l'icône ).

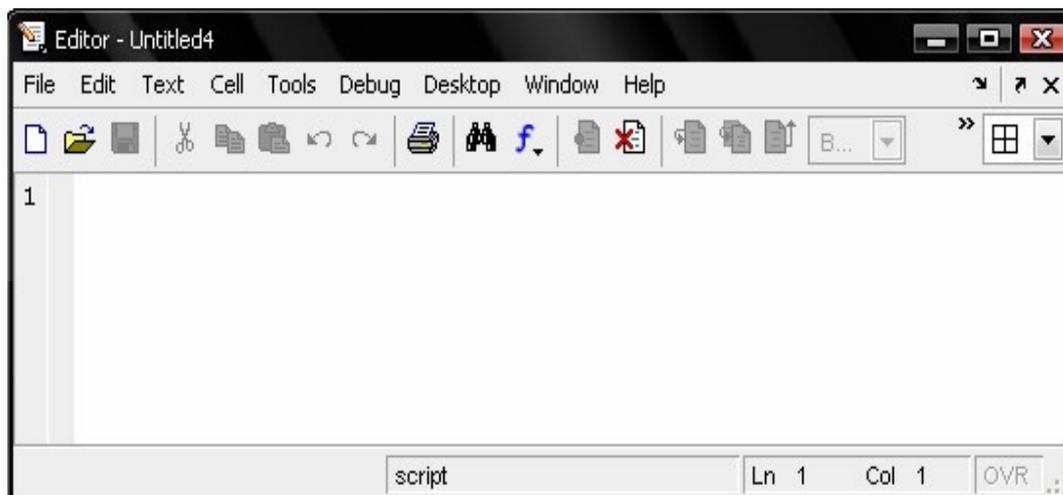


FIGURE 1.2 – La fenêtre d'édition MATLAB

Les bases de MATLAB

Création et calcul de vecteurs et matrices

vecteurs

Un vecteur-ligne ($1 \times n$)(resp colonne ($n \times 1$)) est introduit de la façon suivante :

```
v = [ 6 , 2 , 1 3 , 5 ]; w = [ 1 ; 2 ; 3 ; 4 ];
```

Si l'introduction est terminée par un point-virgule, on évite l'affichage du vecteur v ou w . L'accès aux composantes d'un vecteur s'effectue directement par des commandes du genre :

```
v(2); v(1: 3); w(4); w(1:2:end);
```

Si comme dans le cas présent, $v(2)$ n'est pas affecté à une variable, par une commande de la forme $y=v(3)$ MATLAB copie le résultat dans la variable système **ans**

matrices

Une matrice peut être construite de différentes manières :

```
M=[ 1, 2, 1 2, 4; 7, 8, 3, 1 0; 0, 1, 1, 9]
N=[1 2 3 4;5 6 7 8; 9 10 11 12; 13 40 42 43]
```

où les ; séparent les lignes d'une matrice. On peut construire une matrice à partir de plusieurs vecteurs de même longueur

```
x=[ 1 2 3]; y=[2, 5 6]; z=[8 3 1]; mat=[x;y;z]
```

vecteurs de chaîne de caractères

La chaîne de caractères est un vecteur ligne. Pour le créer, on entre les caractères entre deux (quote (')), exemple »ch='Matlab'

Les nombres complexes

Dans Matlab, un nombre complexe est de la forme : $z = a + ib$. exemple $z=3+i$. La commande **conj(z)**, **real(z)**, **imag(z)**, **abs(z)** et **angle(z)** calcul le conjugué, la partie réel, la partie imaginaire le module et l'angle en radian (respectivement) du nombre complexe z

Remarque 3 La commande **format** permet d'afficher un nombre à un format donné. Les différents formats possibles sont :

- *short* (valeur par défaut) : représentation d'un nombre réel sous forme fixe avec 4 chiffres de précision après la virgule ($x\dots x.y1y2y3y4$).
- *long* : représentation d'un nombre réel sous forme fixe avec 14 chiffres de précision après la virgule.
- *short e* : représentation d'un nombre réel en notation scientifique avec 4 chiffres de précision après la virgule ($x\dots x.y1y2y3y4e+z1z2z3$).
- *long e* : représentation d'un nombre réel en notation scientifique avec 14 chiffres de précision après la virgule.
- *format rat* : représentation d'un nombre au format rationnel (x/y).

Les polynômes

Matlab représente un polynôme sous forme d'un vecteur ligne contenant les coefficients classés dans l'ordre des puissances décroissances. Par exemple le polynôme $P = x^2 - 7x + 9$ est représenté par [1 -7 9]

Remarque 4 Dans MATLAB, les indices des vecteurs et matrices doivent être des entiers positifs. L'indice zéro n'est donc pas plus admis que les indices négatifs.

Exercice 1 Construire

1. une suite partant de 8 et allant à 5 par pas de 0 :25.
2. une suite décroissante d'entiers de 16 à 5.
3. une suite de longueur 50 de π à π .

Opérations vectorielles

Les tableaux suivants résument certaines commandes couramment utilisées.

Vecteurs

<code>n:m</code>	nombre de n à m par pas de 1
<code>n:p:m</code>	nombre de n à m par pas de p
<code>linspace(n,m,p)</code>	p nombres de n à m Le pas est calculé automatiquement par Matlab selon la formule : $(m-n)/(p-1)$...
<code>length(x)</code>	longueur de x
<code>x(i)</code>	ième composante de x
<code>x(i1:i2)</code>	composantes i1 à i2 de x
<code>x(i1:i2)=[]</code>	supprimer les composantes i1 à i2 de x
<code>[x,y]</code>	concaténer les vecteurs x et y
<code>x*y'</code>	produit scalaire des vecteurs lignes x et y
<code>x'*y</code>	produit scalaire des vecteurs colonnes x et y
<code>reshape(x,u,v)</code>	construire une matrice de taille [u,v], a partir de x
<code>X = diag(v,k)</code>	Lorsque v est un vecteur de n composantes, renvoie ... une matrice carrée X d'ordre $n+abs(k)$, avec les éléments de v sur ... la kième diagonale. $k = 0$ représente la diagonale principale, $k > 0$... au-dessus de la diagonale principale et en $k < 0$ dessous de la ... diagonale principale.
<code>v = diag(X,k)</code>	pour X matrice, renvoie un vecteur colonne v formé à ... partir des éléments de la kième diagonale de X.

Fonctions mathématiques propres aux vecteurs

Il existe aussi des commandes qui sont propres aux vecteurs

<code>sum(x)</code>	somme des éléments du vecteur x
<code>prod(x)</code>	produit des éléments du vecteur x
<code>max(x)</code>	plus grand élément du vecteur x
<code>min(x)</code>	plus petit élément du vecteur x
<code>mean(x)</code>	moyenne des éléments du vecteur x
<code>sort(x)</code>	ordonne les éléments du vecteur x par ordre croissant
<code>fliplr(x)</code>	renverse les éléments du vecteur x

Matrices

<code>size(A)</code>	nombre de lignes et de colonnes de A
<code>A(i,j)</code>	coefficient d'ordre i,j de A
<code>A(i1:i2,:)</code>	lignes i1 à i2 de A
<code>A(i1:i2,:) = []</code>	supprimer les lignes i1 à i2 de A
<code>A(:,j1:j2)</code>	colonnes j1 à j2 de A
<code>A(:,j1:j2) = []</code>	supprimer les colonnes j1 à j2 de A
<code>A(:)</code>	indexation linéaire de A, (concaténation des vecteurs colonnes de A)
<code>A(i)</code>	coefficient d'ordre i dans l'indexation linéaire
<code>diag(A)</code>	coefficients diagonaux de A

Matrices particulières

<code>zeros(m,n)</code>	matrice nulle de taille m,n
<code>ones(m,n)</code>	matrice de taille m,n dont tous les coefficients valent 1
<code>eye(n)</code>	matrice identité de taille n
<code>diag(x)</code>	matrice diagonale dont la diagonale est le vecteur x
<code>magic(n)</code>	carre magique de taille n
<code>rand(m,n)</code>	matrice de taille m,n a coefficients sur [0; 1]

Exercice 2 Entrez la matrice $A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & 1 & 2 \end{bmatrix}$ Quels sont les résultats des commandes suivantes ?

- `A([2 3],[1 3])`
- `A([2 3],1 :2)`
- `A([2 3], :)`
- `A([2 3],end)`
- `A(:)`
- `A(5)`
- `reshape(A(:),size(A))`

Exercice 3 Que produit cette ligne de commande
`diag(-m : m) + diag(ones(2 * m, 1), 1) + diag(ones(2 * m, 1), - 1)`

Exemple 1 Créez des matrices particulières. Exemple de création d'une matrice par blocs :
`C=[A, zeros(3,2); zeros(2,3), eye(2)]`

Exercice 4 Ecrire la matrice carrée M d'ordre 12 contenant les entiers de 1 à 144 rangés par ligne. Extraire de cette matrice les matrices suivantes :

- la sous-matrice formée par les coefficients $a_{i,j}$ de M pour $i = 1, \dots, 6$ et $j = 7, \dots, 12$
- celles des coefficients $a_{i,j}$ pour $(i,j) \in \{1, 2, 5, 6, 9, 10\}^2$
- celle des coefficients $a_{i,j}$ pour $i + j$ pair.

Opérations matricielles et fonctions

Les opérations matricielles

<code>A'</code>	transposée de A
<code>rank(A)</code>	rang de A
<code>inv(A)</code>	inverse de A
<code>expm(A)</code>	exponentielle de A
<code>det(A)</code>	déterminant de A
<code>trace(A)</code>	<code>trace</code> de A
<code>poly(A)</code>	polynome caractéristique de A
<code>eig(A)</code>	valeurs propres de A
<code>[U,D]=eig(A)</code>	vecteurs propres et valeurs propres de A
<code>+</code> <code>-</code>	addition, soustraction
<code>*</code> <code>^</code>	multiplication, puissance (matricielles)
<code>.*</code> <code>.^</code>	multiplication, puissance terme à terme
<code>A\b</code>	solution de $Ax = b$
<code>./</code>	division terme à terme