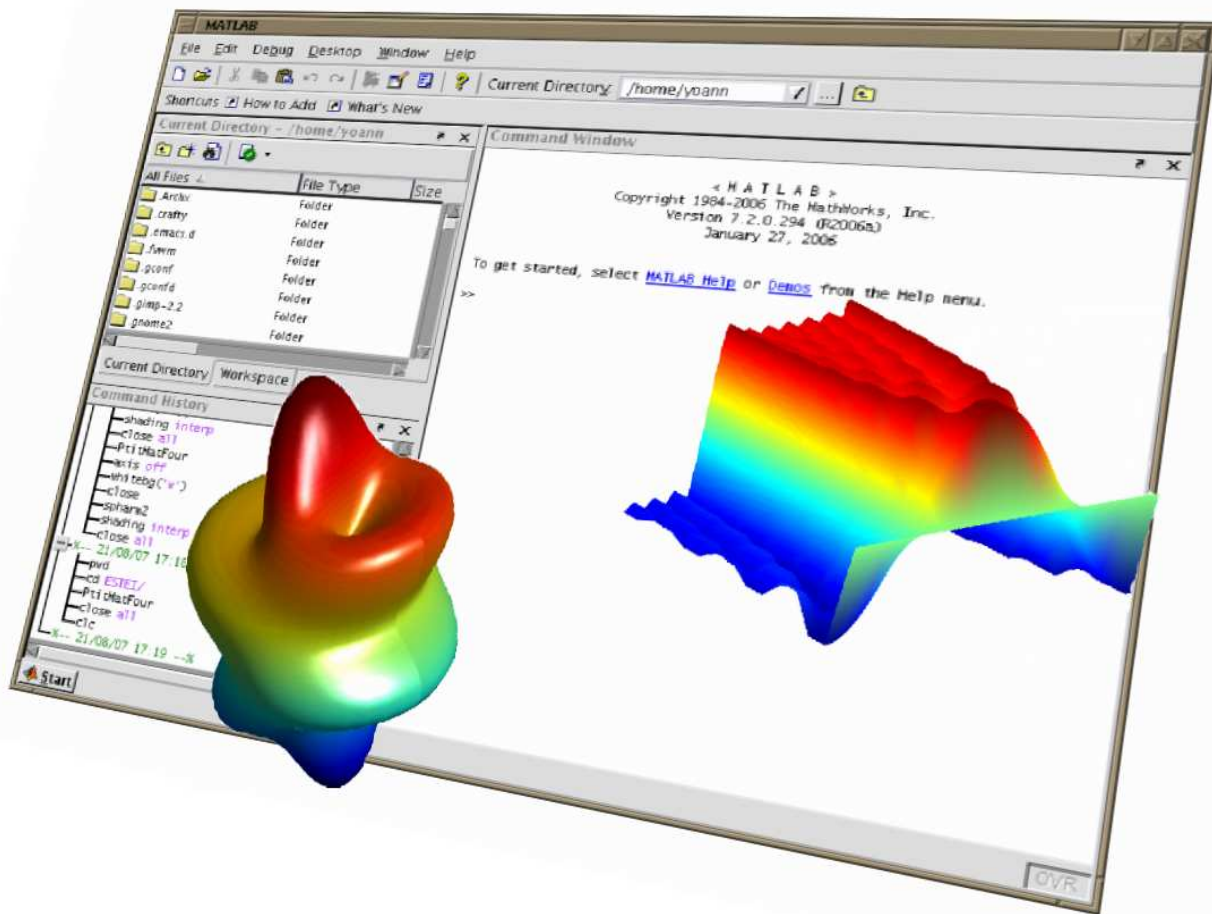


INTRODUCTION ET INITIATION MATLAB



Présentation et introduction à Matlab

Matlab, acronyme de "MATrix LABoratory", est un logiciel et langage de programmation conçu pour fournir un environnement de calcul numérique de haut niveau. Il est particulièrement performant pour le calcul matriciel, et dispose de grandes capacités graphiques pour, par exemple, la visualisation d'objets mathématiques complexes.

Son fonctionnement repose sur un langage de programmation interprété qui permet un développement très rapide. En contre-partie, pour des applications nécessitant des performances plus élevées en temps de calcul, un langage compilé, comme le C++ ou le fortran, est plus adapté.

Sous sa forme "graphique", Matlab dispose d'une interface comprenant l'environnement Matlab à proprement parler, d'où les commandes Matlab peuvent être directement exécutées, ainsi que d'un environnement graphique, pouvant comprendre plusieurs fenêtres : liste des variables en cours d'utilisation, historique des commandes exécutées, ..., et divers menus plus ou moins habituels, "File", ("New", "Open", ...), "Configuration", "Help", ...

Toutes les commandes des différents menus ont leur alternative en "ligne de commande" dans l'environnement propre à Matlab, la réciproque étant fautive.

Dans ce paragraphe d'introduction à Matlab, on ne s'intéressera qu'à l'environnement propre à Matlab, les commandes et syntaxes de base d'instructions Matlab.

Commandes Matlab

Les commandes peuvent se taper directement suite au prompt de Matlab. L'opération est alors immédiatement effectuée et le résultat retourné. Si la commande se termine par un point virgule, la commande est effectuée, mais le résultat obtenu n'est pas retourné.

L'aide

| | |
|------------------------------|--|
| <code>help func</code> | affiche l'aide concernant la fonction <i>func</i> . Voir <i>help help...</i> |
| <code>helpdesk</code> | version graphique et naviguable (html) de l'aide |
| <code>lookfor keyword</code> | lance une recherche sur toutes les fonctions Matlab, et retourne toutes les fonctions dont l'aide contient le mot clé <i>keyword</i> |
| <code>demo</code> | Matlab contient de nombreuses démos. Taper <i>demo</i> , et naviguer... |

Commandes générales

| | |
|------------------------|--|
| <code>cd</code> | change/affiche le repertoire courant |
| <code>which</code> | affiche le chemin complet d'une fonction Matlab |
| <code>path</code> | variable Matlab contenant la liste des repertoires connus, dans lesquels Matlab recherche une fonction lors de son appel |
| <code>addpath</code> | permet d'ajouter un chemin dans le path |
| <code>who</code> | liste des variables de l'espace de travail |
| <code>whos</code> | idem avec tailles en mémoire |
| <code>clear var</code> | supprime la variable <i>var</i> de l'espace de travail |
| <code>clear all</code> | supprime toutes les variables |
| <code>close all</code> | ferme tous les graphiques |

Opérations usuelles

| | |
|-------------------------------------|---|
| <code>=</code> | affectation d'une valeur à une variable (ex. $a = 2$) |
| <code>+, -, *, /</code> | opérations usuelles sur les variables ou valeurs numériques |
| <code>1i</code> | nombre complexe : $(1i)^2 = -1$ |
| <code>abs, angle, real, imag</code> | opérations usuelles sur les nombres complexes |

Définition et opérations sur les vecteurs et matrices

Définition de matrices et vecteurs

| | |
|--------------------------------------|---|
| <code>[, , ; , ,]</code> | définition manuelle d'une matrice (ex. $A = [1, 2, 3; 4, 5, 6]$) |
| <code>deb:pas:fin</code> | définition d'un vecteur régulier balayant l'intervalle $[deb, fin]$ avec le pas pas (ex. $A = 1 : 1 : 6$); par défaut le pas est égal à 1 s'il est omis (ex. $A = 1 : 6$) |
| <code>linspace(deb,fin,N)</code> | définition d'un vecteur balayant l'intervalle $[deb; fin]$ avec N valeurs régulièrement espacées |
| <code>zeros, ones, eye, rand,</code> | matrices particulières (cf. <i>help...</i>) |
| <code>randn, vand, magic...</code> | |

Les opérations usuelles

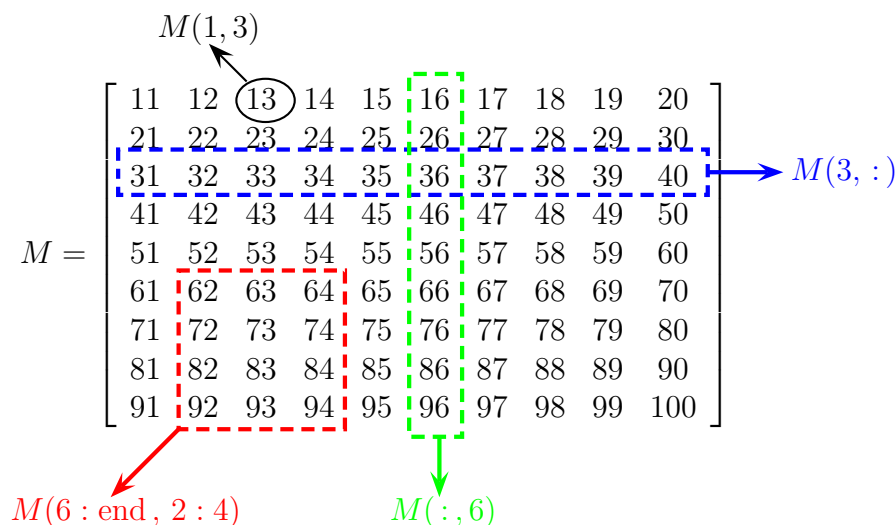
$+, -, *, /, \dots$ agissent indifféremment sur les réels, complexes, ou matrices (à conditions que les dimensions de celles ci le permettent).

D'autres opérations sont également disponibles :

| | |
|---|---|
| <code>.* , ./ , .^ , ...</code> | (les opérateurs usuels précédés d'un point) opérations sur les matrices effectuées terme à terme (ex. calculer $A = [1 : 6] .* [7 : 12]$) |
| <code>length</code> | longueur d'un vecteur |
| <code>size</code> | dimension d'une matrice |
| <code><, <=, >, >=, ==</code> | comparaison des éléments de deux matrices, terme à terme |
| <code>sum, mean, ...</code> | somme, moyenne, ..., des éléments d'un vecteur |
| <code>sin, cos, exp, log, ...</code> | d'une façon générale toutes les fonctions usuelles s'appliquent à des matrices terme à terme (ex. $\log([1, 2, 3]) = [\log 1, \log 2, \log 3]$) |
| <code>find</code> | recherche les éléments non nuls dans une matrice |
| <code>nnz</code> | compte le nombre d'éléments non nuls dans une matrice |

Extraction des éléments d'une matrice

| | |
|--------------------------|---|
| <code>M(i, j)</code> | élément de la matrice M situé sur la ligne i et la colonne j |
| <code>V(end)</code> | dernier élément du vecteur V |
| <code>M(5:9, 3)</code> | les éléments de la matrice M situés de la ligne 5 à 9, et sur la colonne 3 |
| <code>M(:, j)</code> | toutes les lignes de la matrice M , colonne j |
| <code>M(i, :)</code> | toutes les colonnes de la matrice M , ligne i |
| <code>M(1:5, 1:3)</code> | les éléments de la matrice M situés sur les lignes 1 à 5, et sur les colonnes 1 à 3 |



Fonctions graphiques

| | |
|---|---|
| <code>figure</code> | créé une nouvelle figure; <code>figure(n)</code> (ré)initialise la figure n° n |
| <code>plot</code> | tracer d'un ensemble de points (ex. <code>plot([0:0.1:6],sin([0:0.1:6]))</code>) |
| <code>subplot</code> | partionne la figure courante en plusieurs sous-graphiques |
| <code>imagesc</code> , <code>pcolor</code> | affiche via une échelle de couleurs le contenu d'une matrice |
| <code>axis</code> | sélection manuelle de l'échelle d'un graphique |
| <code>title</code> , <code>xlabel</code> , <code>ylabel</code> , <code>legend</code> | titre général, titre sur les axes et légende d'une figure |
| <code>grid on / off</code> | affiche une grille sur le graphique courant |
| <code>hold on /off</code> | permet de superposer des graphiques sur une même figure |
| <code>plot3</code> , <code>semilogx</code> , | autres fonctions graphiques (voir l'aide...) |
| <code>semilogy</code> , <code>loglog</code> , | |
| <code>mesh</code> , <code>surf</code> ,... | |

Scripts et fonctions Matlab

Les commandes ci-dessus peuvent être tapées directement dans l'environnement Matlab, qui l'exécute immédiatement et retourne le résultat (s'il est demandé) quand son calcul est terminé.

Il est aussi possible de regrouper ces commandes dans un fichier "texte" comportant un ensemble de commandes à effectuer. Ce fichier doit être enregistré avec l'extension ".m" (ex. *script.m*), et peut être exécuté directement sous Matlab, si le répertoire dans lequel il est enregistré est soit le répertoire courant (cf. `cd`, ou `pwd`), soit présent dans la variable *path* de Matlab. Un tel fichier peut-être créé en utilisant l'éditeur de Matlab (commande `edit`), soit en utilisant un quelconque éditeur : `vi`, `emacs`, `notepad`, `gedit`, ...

On peut distinguer :

- les **scripts**, qui permettent simplement de regrouper un ensemble de commandes Matlab. A l'appel du script, les commandes sont exécutées séquentiellement.
- les **fonctions**, qui permettent aussi de regrouper un ensemble de commandes effectuant une tâche plus globale. Les fonctions prennent de plus éventuellement un ensemble d'arguments en entrée et retournent un ensemble de valeurs de sortie.

Par défaut, dans une fonction, toutes les variables sont locales, sauf celles explicitement retournées par la fonction.

La structure générale d'une fonction est la suivante :

```
function [y1 y2 ... yp]=nom_func(x1,x2,...,xn)
% Commentaires qui seront affichés lors
% de l'appel "help nom_func"
Corps de la fonction
```

où x_1, x_2, \dots, x_n sont les variables, ou arguments, d'entrée, tandis que y_1, y_2, \dots, y_p sont les valeurs retournées par la fonction.

Une telle fonction s'appelle ensuite par la commande : `[y1,y2,...,yp]=nom_func(x1,x2,...,x4)` ; ou, si on ne souhaite pas affecter les variables de sortie : `nom_func(x1,x2,x3,x4)` ;

Commandes structurées

Les structures usuelles sont bien sûr aussi disponible sous Matlab, par exemple :

```
for ...end , if ...else...end , while ...end , switch ...case...end , ...
```

L'aide (par exemple `help for`) permet de retrouver facilement la syntaxe de ces structures.