INFO 430 MULTIMEDIA

INFO430

Dr Siba Haidar

PRISE EN MAIN MATLAB

INFO430

Introduction à Matlab

- ***Matlab*** est un logiciel de calcul numérique produit par MathWorks, disponible sur plusieurs plateformes
- ***Matlab*** est un langage simple et très efficace, optimisé pour le traitement des matrices, d'où son nom
- Pour le calcul numérique, ***Matlab*** est beaucoup plus concis,
  - → plus besoin de programmer des boucles pour modifier un à un les éléments d'une matrice
  - on peut traiter la matrice comme une simple variable

Introduction à Matlab

- ***Matlab*** contient également une interface graphique puissante, ainsi qu'une grande variété d'algorithmes scientifiques
- On peut enrichir ***Matlab*** en ajoutant des "boîtes à outils" (*toolbox*)
  - des ensembles de fonctions supplémentaires, profilées pour des applications particulières (traitement des images, analyses statistiques, optimisation, etc.)

Aspects élémentaires

Aides

- *help* → donne de l'aide sur une fonction ou un toolkit (help help)
- *helpdesk* → documentation en hypertexte (requiert Netscape ou autre)
- *helpwin* → aide en ligne dans une fenêtre séparée
- *lookfor* → recherche d'un mot clé (lent)
- *which* → localise fonctions et fichiers
- *what* → liste des fichiers matlab dans le répertoire courant
- *exist* → check si une fonction ou une variable existe dans le workspace
- *who, whos* → liste des variables dans le workspace

Variables scalaires, workspace, opérations élémentaires

- >> var=2
- var = 2
- >> autre=3;
- >> who % fournit la liste des variables définies dans le workspace
- Your variables are:
- autre var
- >>whos % donne plus d'informations que who
- Sous Windows, vous avez accès au “Workspace browser” dans la barre d’outils.
- >> clear autre
- >> who
- Your variables are:
- var
- >>clear % efface toutes les variables du workspace

Opérations élémentaires:

- + - * / or \ ^
- >> 4/2
- ans =
- 2
- >> 2\4
- ans =
- 2

Commentaires, ponctuation

- >> s=2+3 % je fais une somme
- s = 5
- >> cout_moyen = cout ... % commande sur deux lignes
- / nombre;

Variables spéciales

- *pi inf i* or *j realmin realmax eps ans*
- >> eps
- ans =
- 2.2204e-16
- >> realmax
- ans =
- 1.7977e+308
- >> realmin
- ans =
- 2.2251e-308

Nombres complexes

- >> c1 = 1-2i
- c1 =
- 1.0000 - 2.0000i
- >>c2 = 3*(2-sqrt(-1)*3)
- c2 =
- 6.0000 - 9.0000i
- real(c1) imag(c1) abs(c1)

Fonctions mathématiques

- *sin cos tan ...*
- *exp log log10 sqrt*
- *fix floor ceil round mod rem sign*
- *cart2sph cart2pol pol2cart sph2cart*
- *factor isprime primes gcd (*pgcd*) lcm (*ppcm*)*

Fonctions mathématiques

- *nchoosek* (nombre de combinaisons différentes de N éléments pris k à k)
- >> nchoosek(30,4)
- ans =
- 27405

Fonctions mathématiques

- perms (toutes les permutations possibles d'un vecteur V)
- >> perms([1 2 3])
- ans =
- 3 2 1
- 2 3 1
- 3 1 2
- 1 3 2
- 2 1 3
- 1 2 3

Fonctions mathématiques

- *dot* (produit vectoriel
- >> v1=[1 3 5]
- v1 =
- 1 3 5
- >> v2=[2 4 6]
- v2 =
- 2 4 6
- >> dot (v1, v2) %meme que v1*v2’
- ans =
- 44

- >>whos

- name – size – bytes – class
- les memes que dans le workspace

Affichage

- FORMAT Set output format.
- All computations in MATLAB are done in double precision.
- FORMAT may be used to switch between different output

- display formats as follows:
- FORMAT SHORT (default) Scaled fixed point format with 5 digits.
- FORMAT LONG Scaled fixed point format with 15 digits.
- FORMAT SHORT G Best of fixed or floating point format with 5 digits.
- FORMAT LONG G Best of fixed or floating point format with 15 digits.
- Autres format : cf help format
- Spacing:
- FORMAT COMPACT Suppress extra line-feeds.
- FORMAT LOOSE Puts the extra line-feeds back in.

Exemples

- >> pi
- ans =
- 3.1416
- >> format long g
- >> pi
- ans =
- 3.14159265358979

Entrées-sorties

- Deux commandes utiles pour gérer le workspace, dont la taille dépend de votre espace de swap:
- >> save % écrit toutes les variables du workspace dans le fichier matlab.mat
- >> load % charge dans le workspace toutes les variables du fichier matlab.mat
- Entrées-sorties sur des fichiers disques:
- fopen (ouverture d'un fichier) fclose (fermeture d'un fichier)
- fscanf (lecture formatée) fprintf (écriture formatée)
- N = input('Nombre de boucles désirées >'); % entrée interactive
- disp(N) % affiche la valeur de N

Terminer Matlab

- >> quit % retourne le nombre d'opérations effectuées pendant la session
- 5340584 flops.

Personnaliser Matlab

- Si, au démarrage, Matlab trouve le fichier startup.m, il l'exécute.
- Exemple de startup.m :

```
disp(' Executing STARTUP.M')
addpath(' H:/Matlab/Cours');
addpath(' H:/Matlab/Exercices');
cd Matlab; % (set current directory to ./Matlab)
format compact
disp(' STARTUP.M completed')
```

Vecteurs

Création de vecteurs

- Par défaut, le vecteur est une ligne à plusieurs colonnes
- ***a) vecteur ligne par énumération des composantes***:
- >> v = [1 3.4 5 -6]
- ***b) vecteur ligne par description***:
- >> x = [0 : pi/10 : pi] % [valeur-initiale : incrément : valeur-finale]
- ***c) vecteur colonne***:
- >> xcol = x'
- ***d) génération de vecteurs métriques***
- >> x = linspace(0, pi, 11) % génère le même x que ci-dessus (11 valeurs. réparties de 0 à pi)
- ausso logspace

Adressages et indexages

- >> x(3) % 3ème élément du vecteur x
- >> x(2 : 4) % un bloc de composantes
- >> x([8 3 9 1]) % une sélection de composantes (on les désigne avec un autre vecteur!)

Combinaison de vecteurs

- ***a) Accolage de deux vecteurs***:
- >> a = [1:3]
- >> b=[10:10:30]

- >> c = [a b]

- >> d=[a(2:-1:1) b] % on accole b avec une portion de a dans l'ordre renversé

Tableau 1: Notations

[ ]	énumération d'éléments
:	descripteur d'éléments de vecteur/matrice
( )	ensemble d'arguments
,	séparateur d'arguments
;	séparateur des lignes dans les matrices supression du résultat de l'évaluation d'une instruction
'	transposition de matrice
.	force l'opérateur à s'appliquer sur chaque élément du vecteur/matrice
%	délimitateur de commentaire
...	continuation de l'instruction sur la ligne suivante

Matrices

Création de matrices

- Un matrice est un ensemble de lignes comportant toutes le même nombre de colonnes
- Matlab, depuis la version 5, supporte les matrices à n dimensions (n>2)
- ***a) Par énumération des éléments***
- >> m1 = [1 2 3 ; 4 5 6 ; 7 8 9] % on sépare les lignes par des point-virgules
  - On peut étendre aux matrices les autres manières de définir des vecteurs. Par exemple:
- >> m2 = [1:1:3 ; 11:1:13]

Transposition

- l'opérateur apostrophe utilisé pour créer un vecteur colonne est en fait l'opérateur transposition:
- >> m2'
- ans =
- 1 11
- 2 12
- 3 13

Opérations scalaires-matrices

- Une telle opération agit sur chaque élément de la matrice:
- >> m2’ * 10 % de même: 4*m2 m2-10 m2/4
- ans =
- 10 110
- 20 120
- 30 130

Une exception:

- >> m2^2
- ??? Error using ==> ^
- Matrix must be square.
- Dans ce cas, Matlab veut calculer le produit matriciel m2 * m2
- La solution est l'usage du point qui force l'opération sur chaque élément:
- >> m2 .^ 2

Opérations entre matrices

- ***a) Multiplications***
- >> m1 * m2' % le produit matriciel n'est possible que lorsque les dimensions sont cohérentes
- Multiplication élément par élément:
- >> m2 .*m3 % (m2 et m3 ont les mêmes dimensions)
- ***b) Divisions***
- >> m2/m3 % division matricielle à droite
- >> m2\m3 % division matricielle à gauche (cf algèbre linéaire!)
- m2./m3 % chaque élément de m2 est divisé par l'élément équivalent de m3
- >> m2.\m3 % chaque élément de m3 est divisé par l'élément équivalent m2
- m3./m2 % chaque élément de m3 est divisé par l'élément équivalent m2

Matrices particulières

- >> ones(3)
- >> zeros(2,5)
- >> eye(4) % aussi: eye(2,4)
- >> diag([1 : 4])
- rand(1,7) % nombres aléatoires entre 0 et 1

Caractéristiques des matrices

- >> size(m3) % dimensions
- >> length(m3) % equivalent à max(size(m3)) : dimension maximum

Exercice

- *Définissez A une matrice 3x3*
- *Mettez à zéro l'élément (3,3)*
- *Changez la valeur de l'élément dans la 2ème ligne, 6ème colonne, que se passe-t-il?*
- *Mettez tous les éléments de la 4ème colonne à 4*
- *Créez B en prenant les lignes de A en sens inverse*
- *Créer C en accolant toutes les lignes de la première et troisième colonne de B à la droite de A*
- *Créer D sous-matrice de A faite des deux premières lignes et les deux dernières colonnes de A. Trouvez aussi une manière de faire qui ne dépende pas de la taille de A.*
- ***Note**: chacun de ces exercices se fait en une seule instruction, sans boucles ittératives.*

Manipulations de matrices et sous-matrices

Fonctions de manipulation des matrices:

- >> A = [1 2 3 ; 4 5 6 ; 7 8 9]
- A =
- 1 2 3
- 4 5 6
- 7 8 9
- >> flipud(A) % flip up-down
- >> fliplr(A) % flip left-right
- >> rot90(A,2) %2 rotations de 90 degres (sens trigo)
- >> reshape(A,1,9) % change la forme de la matrice
- >> diag(A) % extrait la diagonale de A
- >> diag (ans) % diag travaille dans les 2 sens !
- >> triu(A) % extrait le triangle supérieur de A
- >> tril(A) % triangle inférieur

Exercice (avancé)

- *Sans utiliser de boucles d'ittération, ajouter aux éléments d'une matrice l'indice de leur colonne.*

Programmer en Matlab

Opérateurs logiques et de relation

- < plus petit
- > plus grand
- <= plus petit ou égal
- >= plus grand ou égal
- == égal
- ~= pas égal
- & et
- | ou
- ~ not
- xor(x,y) ou exclusif
- any(x) retourne 1 si un des éléments de x est non nul
- all(x) retourne 1 si tous les éléments de x sont non nuls
- isequal(A,B), ischar etc...

Contrôler l'exécution

- ***a)For***
- for n = 1:5
- for m = 5:-1:1
- A(n,m) = n^2 + m^2;
- end
- disp(n)
- end

- ***b) While***
- while expression
- (commands)
- end

- ***c) If-then-else***
  - if expression1
  - (commandes à exécuter si expression1 est“vrai”)
  - elseif expression2
  - (commandes à exécuter si expression2 est“vrai”)
  - else
  - (commandes à exécuter si aucune expression est “vrai”)
  - end

M-Files ou scripts

- Un script (ou M-file) est un fichier (message.m par exemple) contenant des instructions Matlab.
- Voici un exemple de script:

- % message.m affiche un message
- % ce script affiche le message que s'il fait beau
- beau_temps=1;
- if beau_temps~=0
- disp('Hello, il fait beau')
- end
- return % (pas nécessaire à la fin d'un M-file)

M-Files ou scripts

- Matlab vous offre un éditeur pour écrire et mettre au point vos M-files:
- >> edit % lance l'éditeur de MatLab
- Les M-files sont exécutés séquentiellement dans le "workspace", c'est à dire qu'ils peuvent accéder aux variables qui s'y trouvent déjà, les modifier, en créer d'autres etc.
- On exécute un M-file en utilisant le nom du script comme commande:
- >> message
- Hello, il fait beau

Fonctions

- On peut écrire des fonctions MatLab que l'on peut ensuite appeler depuis un script.
- Voici une fonction" temps" définie dans le fichier temps.m:

```
function y=temps(x)
% TEMPS(X) affiche un message suivant le temps qu'il fait
% et retourne le paramètre d'entrée X changé de signe
if length(x)>1 error('X doit être un scalaire'); end
if x~=0
disp('Hello, il fait beau')
else
disp('Espérons que demain sera meilleur!')
end
y=-x;
return
```

Fonctions

- Utilisation de cette fonction:
- >> clear
- >> help temps
- TEMPS(X) affiche un message suivant le temps qu'il fait et retourne le paramètre d'entrée X changé de signe
- >> temps(1)
- Hello, il fait beau
- -1

Fonctions

- Remarquez que les variables internes d'une fonction sont locales et n'entrent pas dans le workspace.
- Vous pouvez, dans une fonction, déclarer une ou plusieurs variables globales afin de pouvoir les utiliser depuis l'extérieur de la fonction.
  - global x y

Fonctions

- Matlab offre plusieurs moyens de vérifier les arguments d'entrées et de sorties d'une fonction:
  - nargin retourne le nombre d'arguments d'entrée
  - nargout retourne le nombre d'arguments de sortie
  - nargchk vérifie le nombre d'arguments d'entrée
  - error Affiche un message d'erreur
  - inputname retourne le nom d'un argument d'entrée

Exercice:

- *Ecrivez une fonction “index_of_max” qui retourne l’index de l’élément le plus grand d’un vecteur A donné comme argument d’entrée.*

Gestion du système de fichiers

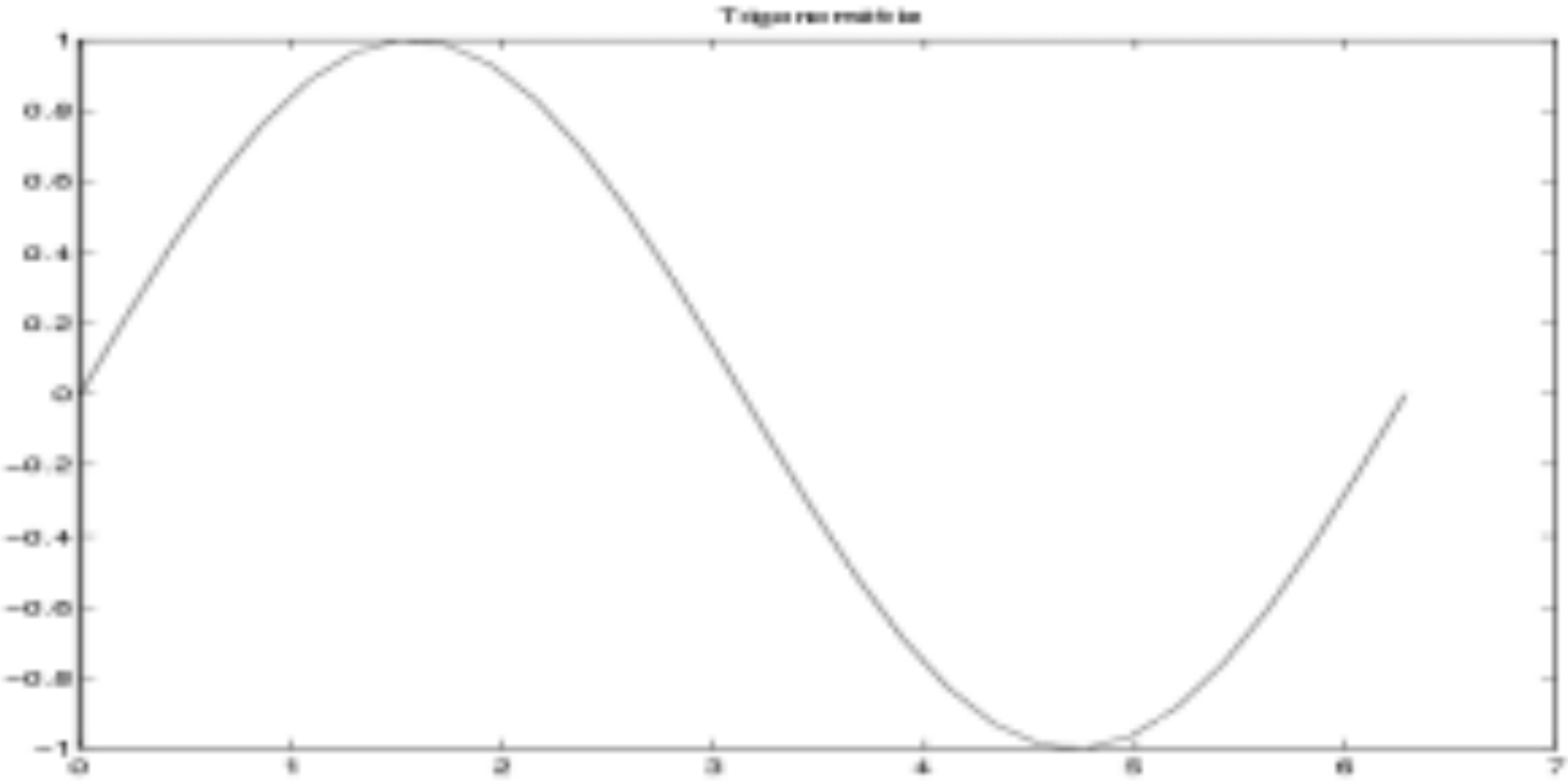
- Matlab utilise la syntaxe Linux/Unix (DOS est assez semblable à ce niveau) pour la gestion des dossiers.
- Ainsi on peut connaître le dossier courant et se déplacer dans l'arbre des dossiers à l'aide des commandes suivantes :
  - pwd permet de connaître le dossier courant (*Print Working Directory*)
  - cd .. remonte au dossier supérieur
  - cd sub sélectionne le dossier inférieur nommé *sub*
  - ls donne la liste des fichiers dans le dossier courant (*LiSt*)

Graphisme

- Nous donnons ici les indications minimum.
- Utilisez help et les autres commandes d'aide pour affiner vos connaissances et vos graphiques.

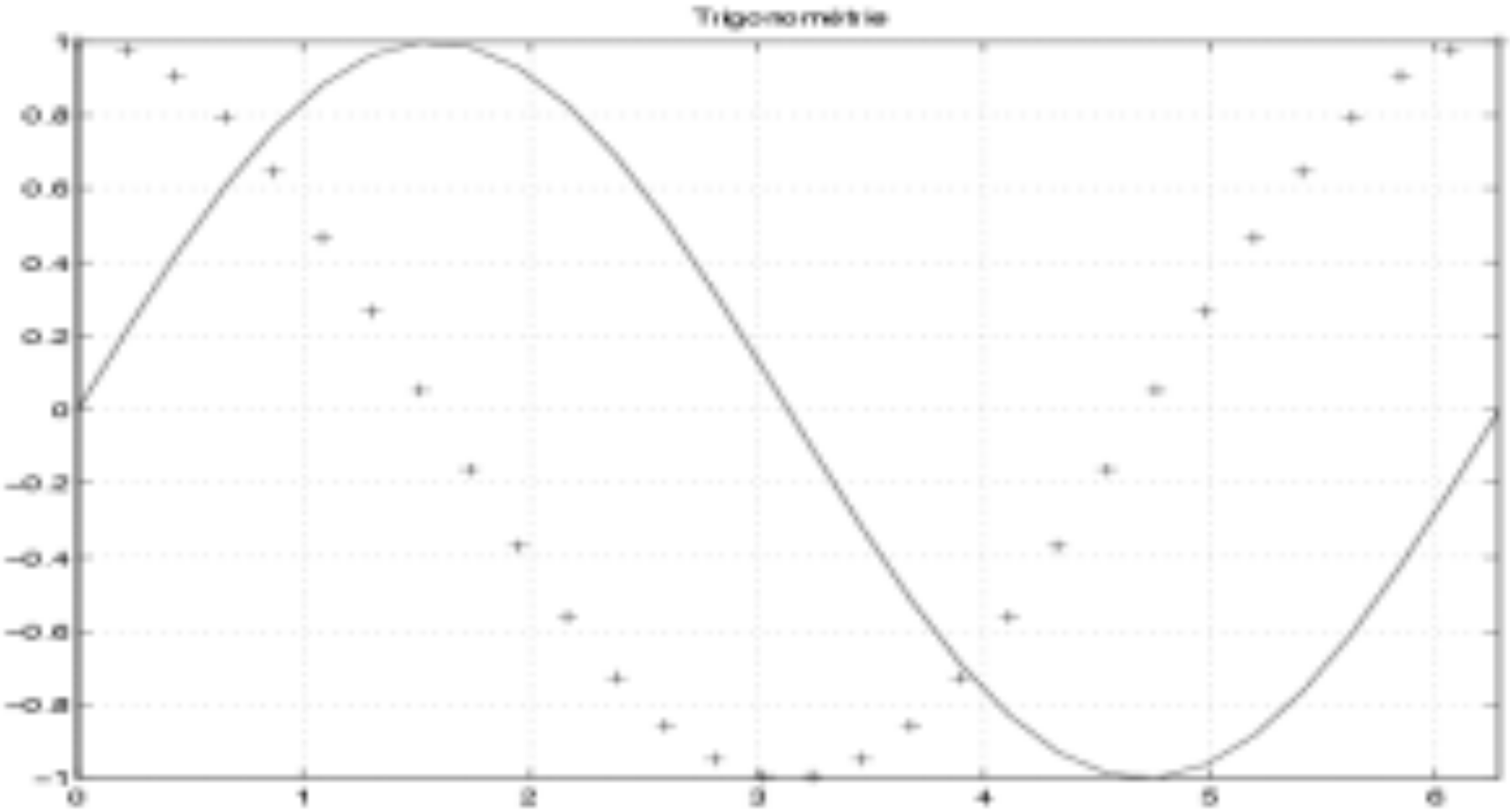
Graphiques à 2D

- >> help graph2d % intro. au graphisme 2D et tableau des fonctions disponibles
- a) courbes: ***plot***
- >> x=linspace(0,2*pi,30);
- >> y=sin(x);
- % un plot rudimentaire:
- >> plot(x,y)
- >> title('Trigonométrie')


1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
0
1
2
3
4
5
6
7

Graphiques à 2D

- % quelques améliorations:
- >> grid on
- >> axis([0 2*pi -1 1])
- % add a second curve
- >> hold on
- >> z=cos(x)
- >> plot(x,z,'c+')
- >> clf % efface la figure


Trigonométrie
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
1
2
3
4
5
6