Initiation Matlab pour les stats

Le but n'est pas de faire de vous des pros de Matlab,

mais de savoir l'utiliser rapidement

Corinne MAILHES, TéSA/INPT

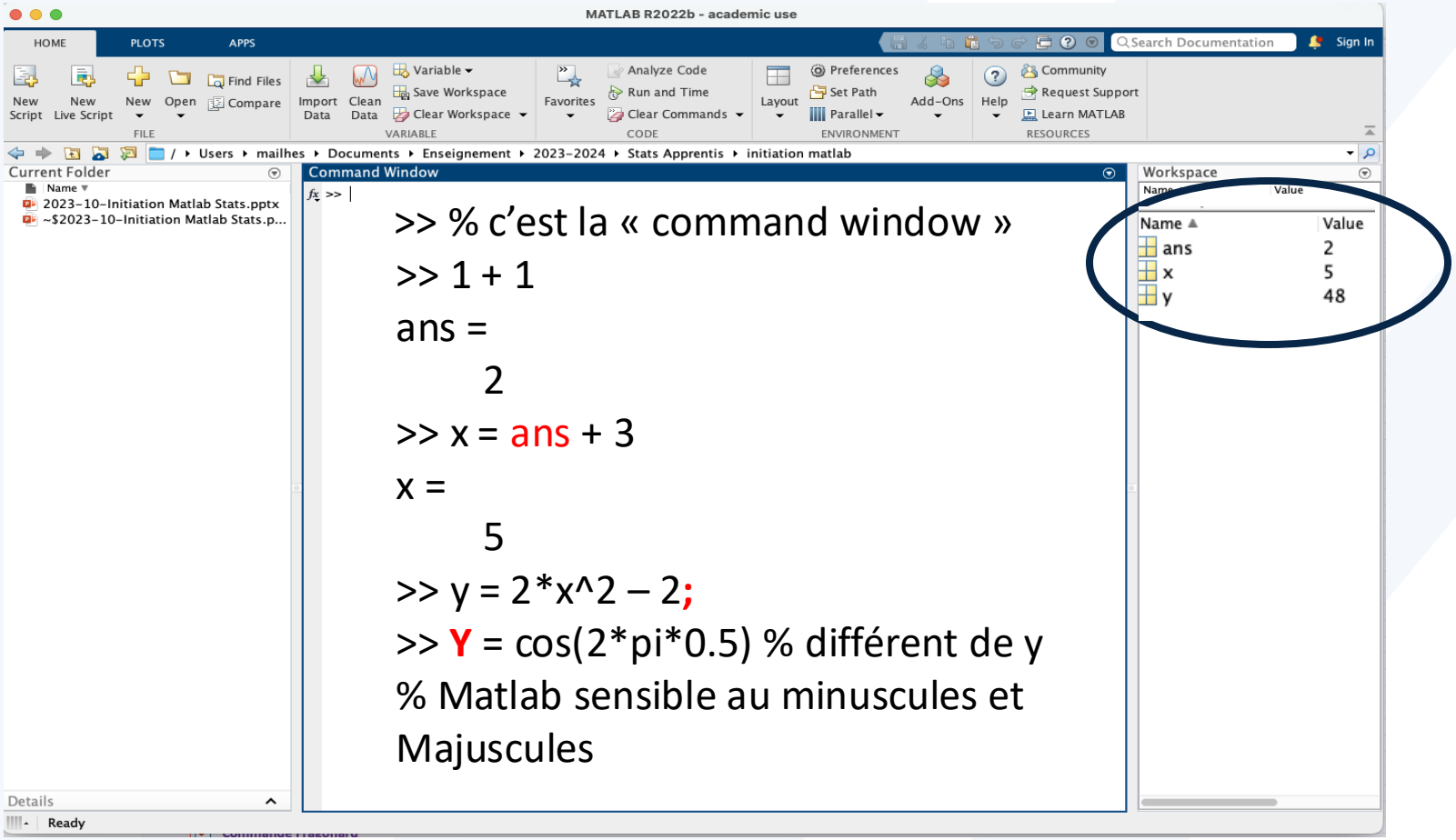
Février 2025

corinne.mailhes@tesa.prd.fr

Généralités

Lancer Matlab

Matlab = Matrix Laboratory avec ses Toolboxes



Généralités

Type de variables : pas de déclaration nécessaire…. ☺ ou ☹ ?

Il existe cinq grands types de variables sous Matlab : les entiers, les réels, les complexes, les chaînes de caractères et le type logique. Définissons une variable de chaque type :

```
>> a = 1.3; b = 3+i; c = 'bonjour';
>> d1 = true(1==1); d2 = logical(1);
>> e = int8(2);
```

```
>> who   ou   whos   % testez les deux commandes
```

Définir un vecteur :

```
>> v = [1 2 3.3 4 5 6]; % vecteur ligne
>> v % tapez enter pour le voir
>> w = [1 ; 2 ; 3] % vecteur colonne
>> length(w) % tapez enter
>> size(w) % tapez enter
```


```
>> help ones
>> help zeros
```

Concaténer 2 vecteurs :

```
>> vw = [v w'] % un seul vecteur ligne
>> vw_encolonne = [v' ; w ] % ou alors vw' ☺
```

Généralités

Opérations en plus de + - / *

Voici une liste (non exhaustives) des fonctions incorporées dans Matlab :

`exp(x)`	:	exponentielle de x
`log(x)`	:	logarithme néperien de x
`log10(x)`	:	logarithme en base 10 de x
`x^n`	:	x à la puissance n
`sqrt(x)`	:	racine carrée de x
`abs(x)`	:	valeur absolue de x
`sign(x)`	:	1 si $x > 0$ et 0 si $x \leq 0$
`sin(x)`	:	sinus de x
`cos(x)`	:	cosinus de x
`tan(x)`	:	tangente de x
`asin(x)`	:	sinus inverse de x (arcsin de x)
`sinh(x)`	:	sinus hyperbolique de x
`asinh(x)`	:	sinus hyperbolique inverse de x

On pourra également utiliser les fonctions d'arrondis :

`round(x)`	:	entier le plus proche de x
`floor(x)`	:	arrondi par défaut de x
`ceil(x)`	:	arrondi par excés de x

Enfin lorsque l'on travaille avec des complexe on pourra utiliser :

`conj(z)`	:	conjugué de z
`abs(z)`	:	module de z
`angle(z)`	:	argument de z
`real(z)`	:	partie réelle de z
`imag(z)`	:	partie imaginaire de z

Généralités

La force de Matlab : les opérations sur les vecteurs (et les matrices)

```
% créer un vecteur qui va de 0 à 1 sur 1024 points
>> vect1 =  linspace(0,1,1024); % remarque : plusieurs façons de faire
>> vect2 = (1 : 3 : 1 + 3 * (1024-1)); % un autre vecteur de taille 1024
        >> somme = vect1+vect2; % par ex
                >> prodTermATerm = vect1 .* vect2; % le « . » avant l'opérateur = terme à terme
                        >> prodScal = vect1 * vect2'     % ça fait une valeur = produit scalaire
                                >> M = vect1' * vect2;  % ça fait une matrice ☺
```

`sum(x)`	:	somme des éléments du vecteur x
`prod(x)`	:	produit des éléments du vecteur x
`max(x)`	:	plus grand élément du vecteur x
`min(x)`	:	plus petit élément du vecteur x
`mean(x)`	:	moyenne des éléments du vecteur x
`sort(x)`	:	ordonne les éléments du vecteur x par ordre croissant
`fliplr(x)`	:	renverse l'ordre des éléments du vecteur x

Généralités

La force de Matlab : les opérations sur les vecteurs (et les matrices)

>> A = [16 2 3 13 ; 5 11 10 8 ; 9 7 6 12 ; 4 14 15 1] % une matrice particulière

A =

16	2	3	13
5	11	10	8
9	7	6	12
4	14	15	1

>> A(3,2) % attention, indices commencent à 1 !!!
>> A'
>> diag(A)
>> sum(diag(A))
>> sum(A,1)
>> sum(A,2)
>> inv(A)
>> B = magic(4) % pour s'amuser

>> help min

Les opérations classiques résumées ci-dessous peuvent être appliquées sur les variables scalaires, vectorielles ou matricielles.

Opérations matricielles	
+	addition
-	soustraction
*	multiplication
/	division
^	puissance

Opérations élément par élément	
.*	multiplication élément par élément
./	division élément par élément
.^	puissance élément par élément

Généralités

On crée un « script » (programme) :
New script
Save as Initiation.m

On démarre un programme : en commentaire les infos
% Initiation matlab – octobre 2023
% auteur : Prénom NOM

% exemple : créer un vecteur comme une fonction type cosinus et la représenter

N = input(‘ Nombre de points du signal : ‘); % pour poser une question
t = linspace(0,2*pi, N); % vecteur en abscisse
b=-0.4; a = 0.4/2/pi; % autres paramètres

```
for indice = valeur initiale:pas:valeur finale
  ...suite d'instructions...
end
```

```
for k = 1 : N
    x(k) = cos(t(k));
    y(k) = a*t(k) + b; % autre fonction
end
```

% mieux en une ligne !!! x = cos(t); y = a*t + b; % on peut souvent éviter for/end

Généralités

Visualisation des courbes

```
% de nombreuses possibilités et options !!!
% a minima
figure;
plot(t,x); hold on; plot(t,y);
xlabel('phase en radians');
ylabel('amplitude');
title('Evolution de deux fonctions');
legend('cosinus','linéaire');
v=axis; v(2) = pi; axis(v); % utile pour bien dimensionner la figure

% autre possibilité utile
figure('Name','Visualisation en sous-figures');
subplot(2,3,1); plot(t,x); title('sous-figure 1');
subplot(2,3,2); plot(t,y); title('sous-figure 2');
subplot(2,3,3); plot(t,x,'r'); title('sous-figure 3');
subplot(2,3,4); plot(t,y,'g*'); title('sous-figure 4');
subplot(2,3,5); plot(t,x,'s--'); title('sous-figure 5');
subplot(2,3,6); plot(t,y); title('sous-figure 6');
```

Voir ensuite toutes les options en faisant :
>> help plot
Et cliquer sur Documentation for plot

Généralités

```
if expression à tester
  ...suite d'instructions à exécuter si l'expression est vraie...
else
  ...suite d'instructions à exécuter si l'expression est fausse...
end
```

```
Test = zeros(size(x));
for k=1:N
      if x(k) > y(k)
            test(k) = 1;
      else
            test(k) = -1;
      end
end
% figure (1); % pour y revenir
figure; % une nouvelle
plot(t,x,'r',t,y,'b',t,test,'c');
```

Opérateurs relationnels et logiques de Matlab

==	égalité
~=	inégalité
<	strictement plus petit que
>	strictement plus grand que
<=	plus petit ou égal
>=	plus grand ou égal
|	ou logique
&	et logique

```
% plus fort !!!
test = (x>y); % variable logique (mais qu'on peut visualiser comme les autres)
Test = 2*test -1; % devient réelle ☺
% ou alors
[test indices] = find(x>y); Test = length(test);
Plot(t(indices), test,`*');
disp(['Le résultat est ' num2str(Test/N*100) '%']); % pour montrer « disp »
```

Généralités

Matlab et les fichiers de données

Fichiers au format Matlab : **.mat** **(option '–ascii')**

save et load

Mais Matlab sait lire et écrire beaucoup de formats de fichiers

Categories

Text Files
Delimited and formatted text files

Spreadsheets
Microsoft® Excel® spreadsheets

Images imread, imwrite, …
JPEG, TIFF, PNG, and other formats

Scientific Data
netCDF, HDF, FITS, and CDF formats

Audio and Video audioread, audiowrite, …
Read and write video and audio files; Record and play audio

Structured Data and XML Documents
Work with structured data and Extensible Markup Language documents

JSON Format
JavaScript® Object Notation format

Et aussi dans sa forme la plus « brute »
fopen, fread, fclose,…

Généralités

Matlab et les fonctions (sous-programmes)

Exemple

```
%%% ceci est le main
clear all; close all;

nech = 1000; amp = 1; f0 = 0.21;
x = genersignal(nech,amp,f0); % c'est la fonction
plot(x);
```

Et dans un fichier séparé appelé **genersignal.m**

```
function signal = genersignal(n,a,f0)

signal = a*cos(2*pi*f0*(0:n-1));
signal = signal.^2; % pour corser
signal = signal + a^3;
```

MAIS dans le cadre des TDM/BE/projet Stats, on va faire un NoteBook : **LiveEditor**

https://fr.mathworks.com/help/matlab/matlab_prog/create-live-scripts.html

En français : **https://fr.mathworks.com/products/matlab/live-editor.html**

Vidéo (en anglais) comment utiliser le live editor :
https://fr.mathworks.com/videos/using-the-live-editor-117940.html

Généralités

En conclusion, pour savoir comment faire quelque chose sous matlab :
>> help mean
>> lookfor mean

Google

matlab calcul moyenne

YouTube

Faire des stats avec Matlab

```
>> ver
>> help stats
>> help mean
>> help var
```

Simulation de variables aléatoires :

Soient U_1 et U_2 deux v.a.r. indépendantes de loi uniforme $\mathcal{U}([0, 1])$.

Posons

$$\begin{cases} X_1 = \sqrt{-2\ln(U_1)}\cos(2\pi U_2), \\ X_2 = \sqrt{-2\ln(U_1)}\sin(2\pi U_2). \end{cases}$$

Alors X_1 et X_2 sont deux v.a.r. indépendantes de loi normale centrée réduite $\mathcal{N}(0, 1)$.

Charger template sur site JYT

C'est à VOUS!

Pour vérifier, utiliser :

histogram (mieux que hist)

(on peut faire hold on en affichant la densité de probabilité ☺)

Ou mieux,

histfit

(faire help histfit ou hist pour les paramètres)

Nombre de bins pour l'histogramme recommandé :

round(sqrt(Nombre d'observations))