

UNIVERSITÉ DE BÉJAÏA

Méthodes de Monte-Carlo

Master1 PSA : 2018–2019

Tutoriel MATLAB

Table des matières

1 Les entrées-sorties	2
2 Vecteurs et Matrices	2
2.1 Définitions de vecteurs et de matrices	2
2.2 Concaténations de matrices ou de vecteurs	2
2.3 Extraction d'une sous-matrice	3
2.4 Création de matrices particulières	3
3 Graphiques	3
3.1 Graphiques simples	3
3.2 Graphiques avancés	3
3.3 Graphiques multiples	4
4 Statistiques et Probabilités	5
4.1 Affichage du nuages de points	5
4.2 Histogramme et autres	6
4.3 Générateur de variables aléatoires	7
5 Fonction Matlab	8
5.1 Syntaxe d'une fonction matlab	8
5.2 l'appel d'une fonction matlab dans un programme	8

Ce tutoriel contient les bases permettant l'utilisation du logiciel Matlab dans le domaine de la simulation statistique.

1 Les entrées-sorties

```
x = input('x =');           % Lire x.
A=[1 2; 4 5];              % matrice A
disp('message');           % Afficher un message.
disp(A);                   % Permet d'afficher la matrice A.
disp(['det A vaut', num2str(det(A))]); % Permet d'afficher un résultat.
sprintf('format, variables') % Permet l'impression de variables selon
                               un modèle donné.
```

2 Vecteurs et Matrices

2.1 Définitions de vecteurs et de matrices

Pour définir un vecteur sous Matlab, il suffit de mettre ses valeurs entre crochets et de les séparer par un espace. Pour obtenir un vecteur vertical, il suffit de le transposer via l'ajout d'un apostrophe après le crochet fermant.

```
%-----Vecteurs-----
u=[1 2 3];    % Vecteur horizontal
v=[1 2 3]';   % Vecteur vertical

%-----Matrices-----
M=[1 2 3; 4 5 6]; % Matrice
Ma=[1 2 3; 4 5 6]'; % Matrice transposée
```

2.2 Concaténations de matrices ou de vecteurs

Les dimensions d'une matrice ou d'un vecteur peuvent être augmentées en introduisant de nouveaux éléments empruntés à une autre matrice ou un autre vecteur. Soit par exemple le vecteur $x = [1 \ 2 \ 3]$. Les commandes :

```
x = [x 4 5 6];
y = [x; 1 : 6];
```

fournissent les résultats suivants :

```
x =
    1    2    3    4    5    6
```

```
y =
    1    2    3    4    5    6
    1    2    3    4    5    6
```

2.3 Extraction d'une sous-matrice

On peut utiliser les deux points pour extraire une sous-matrice d'une matrice M .

$M(i, j);$: Extrait les éléments d'indice (i, j) de M .
$M(:, j);$: Extrait la $j^{\text{ème}}$ colonne de M .
$M(i, :);$: Extrait la $i^{\text{ème}}$ ligne de M .
$M(:, j : k);$: Extrait la sous – matrice de M formée des colonnes j à k .
$M(i : k, :);$: Extrait la sous – matrice de M formée des lignes i à k .
$M(i : k, j : l);$: Extrait la sous – matrice de M formée des éléments situés dans les lignes i à k et dans les colonnes j à l .

2.4 Création de matrices particulières

Matlab permet de créer de manière simplifiée des matrices spéciales comme, par exemple, des matrices remplies de 1 ou de 0, etc. (m et n sont les dimensions de la matrice souhaitée et M est une matrice). Les dimensions d'une matrice sont données par la commande :

$$[m, n] = \text{size}(M);$$

$\text{eye}(m, n);$: Matrice unité.
$\text{ones}(m, n);$: Matrice dont tous les éléments sont égaux à un.
$\text{zeros}(m, n);$: Matrice dont tous les éléments sont égaux à zéro.

3 Graphiques

3.1 Graphiques simples

Les quelques commandes suivantes montrent comment afficher un graphique simple. Matlab ne sachant travailler qu'en valeurs discrètes, il faut d'abord commencer par définir l'intervalle de valeurs de l'abscisse. Une fois ceci fait, on calcul la valeur de la fonction (dans notre cas une équation de droite) pour chaque échantillon de x et l'on arrive au vecteur y .

$x = [0 : \pi/40 : 2 * \pi];$	% Abscisse de 0 à $2 * \pi$ par pas de $\pi/40$.
$y = \cos(x);$	% Valeur de la fonction.
$\text{figure};$	% Nouvelle fenêtre.
$\text{plot}(x, y);$	% Tracer.
$\text{title}('tracé de la fonction \cos');$	% Titre de la figure.
$\text{xlabel}('x');$	% Légende en abscisse.
$\text{ylabel}('y');$	% Légende en ordonnée.
$\text{gtext}('texte');$	% Ajouter du texte.

3.2 Graphiques avancés

Matlab permet de personnaliser les graphiques. Par exemple, il est possible de changer la couleur du trait, de marquer les points avec différents symboles tels que x , o , $*$, ... etc. Voici un aperçu simple de ce qui est possible de faire.

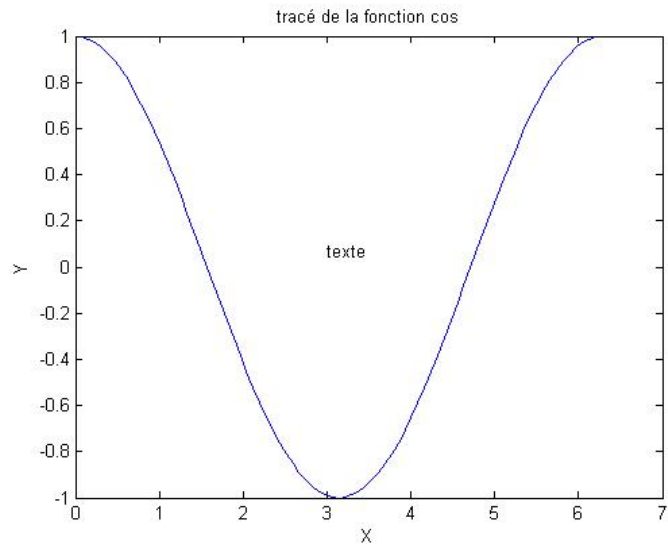


FIGURE 1 – Graphique simple

```
figure; % Nouvelle fenêtre.
x = [0 : pi/40 : 2 * pi];
y = cos(x);

subplot(2, 2, 1); % Zone supérieure gauche.
plot(x, y); % Tracé.
title('Normal'); % Titre de la figure.

subplot(2, 2, 2); % Zone supérieure droite.
plot(x, y, 'r'); % Tracé en rouge.
title('En rouge'); % Titre de la figure.

subplot(2, 2, 3); % Zone inférieure gauche.
plot(x, y, 'or'); % Tracé points ronds en rouge.
title('Cercle en rouge'); % Titre de la figure.

subplot(2, 2, 4); % Zone inférieure droite.
plot(x, y, 'm'); % Tracé points étoiles relié en vert.
title('Etoiles en vert'); % Titre de la figure.
```

3.3 Graphiques multiples

Il est possible de tracer plusieurs courbes sur le même graphique. Pour ce faire, il existe deux méthodes :

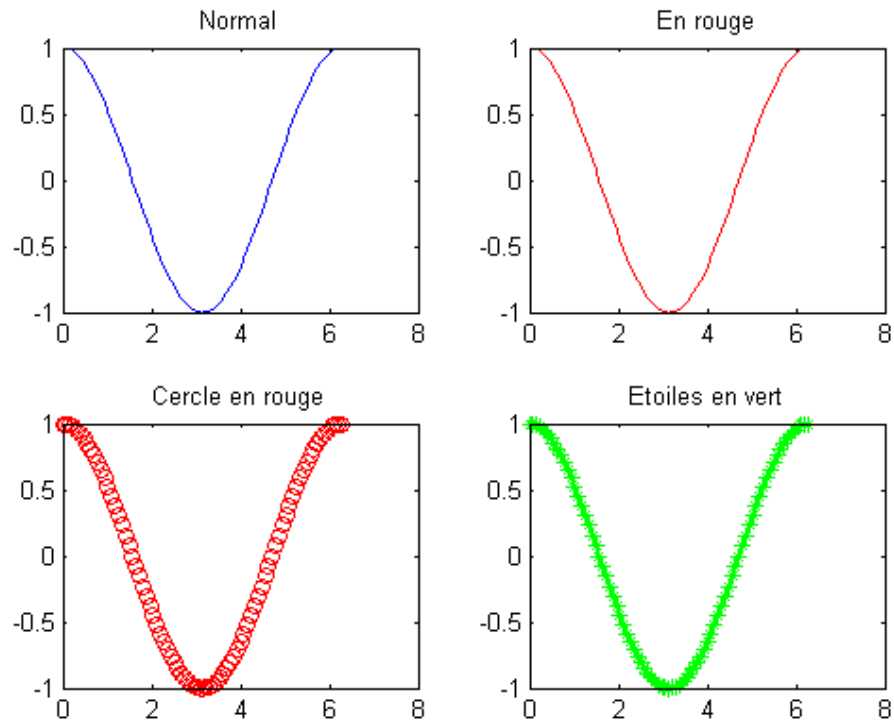


FIGURE 2 – Graphique avancé

```
figure; % Nouvelle fenêtre.
x = [0 : pi/40 : 2 * pi]; y1 = cos(x); % Fonction 1.
y2 = sin(x); % Fonction 2.
hold on; % Maintient du graphique.
plot(x, y1); % Tracé de la droite y1.
plot(x, y2, 'r'); % Tracé de la droite y2 en rouge.
legend('y1 = cos(x)', 'y2 = sin(x)'); % Légende sur graphique.
```

Ou

```
figure; % Nouvelle fenêtre.
x = [0 : pi/40 : 2 * pi];
y1 = cos(x); % Fonction 1.
y2 = sin(x); % Fonction 2.
plot(x, y1, x, y2, 'r'); % Tracé des deux fonctions y1 et y2.
legend('y1 = cos(x)', 'y2 = sin(x)'); % Légende sur graphique.
```

4 Statistiques et Probabilités

4.1 Affichage du nuages de points

Matlab permet l'affichage de nuages de points. Pour ce faire, il faut lui fournir les coordonnées des points en deux vecteurs x et y .

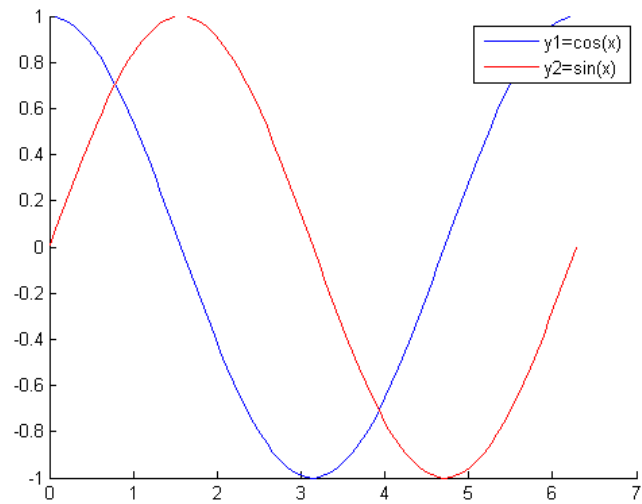
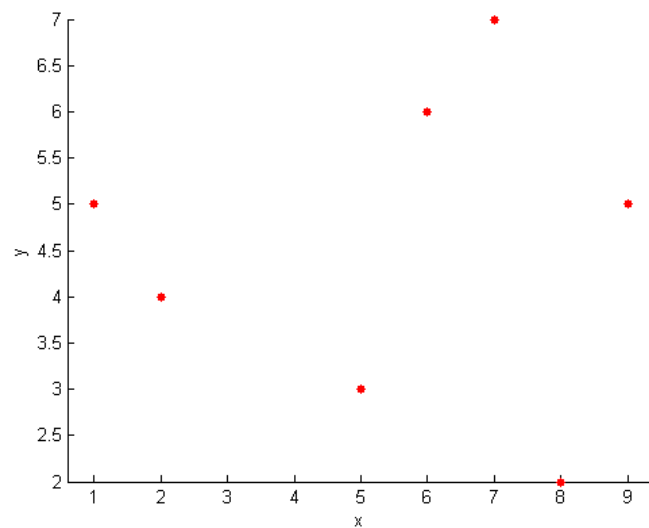


FIGURE 3 – Graphiques multiples

```
x=[1 2 5 6 8 7 9 5]; % Coordonnée en x des points.
y=[5 4 3 6 2 7 5 3]; % Coordonnée en y des points.
gscatter(x,y);        % Affichage.
```



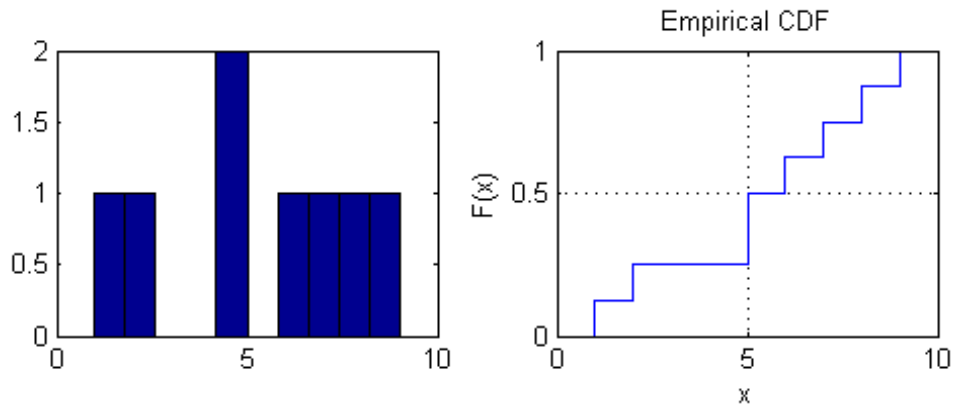
4.2 Histogramme et autres

Il est aussi possible d'afficher les données de différentes manière comme par exemple en histogramme ou en distribution cumulative (fonction de répartition).

```

x=[1 2 5 6 8 7 9 5]; % Coordonnée en x des points.
subplot(2,2,1);      % Zone supérieure gauche.
hist(x);              % Histogramme.
subplot(2,2,2);      % Zone supérieure droite.
cdfplot(x);           % Fonction de répartition.

```



4.3 Générateur de variables aléatoires

Matlab dispose en particulier de certains outils utiles en probabilités et statistiques comme plusieurs générateurs de nombres aléatoires dont **rand**, associé à la loi uniforme sur l'intervalle $\mathcal{U}_{[0,1]}$ et **randn** associée à la loi normale $\mathcal{N}(0, 1)$.

```

rand(n);      % crée une matrice de n×n
               dont les éléments sont uniformément distribués entre 0 et 1.
rand(n,m);    % crée une matrice de n×m
               dont les éléments sont uniformément distribués entre 0 et 1.
rand(n,1);    % crée un vecteur verticale de n éléments.
rand(1,n);    % crée un vecteur horizontale de n élément.
randn(n);     % crée une matrice de n×n
               dont la distribution est gaussienne centrée de variance 1.

```

5 Fonction Matlab

5.1 Syntaxe d'une fonction matlab

La syntaxe d'une fonction Matlab est donnée par :

```
function [ variables de sorties ] = Nomfonction(variables d'entrées )  
    Corps de la fonction.  
end
```

5.2 l'appel d'une fonction matlab dans un programme

On fait appel à une fonction Matlab dans un programme de la façon suivante :

```
[ variables de sorties ] = Nomfonction(variables d'entrées ) ;
```