

Intitulé de la matière : **TP CODAGE ET COMPRESSION**

---

**TP1— INITIATION**

---

## OBJECTIFS DU TP

Ce TP, intitulé *Initiation*, est un pré-acquis pour les TPs ultérieurs. Il vise en particulier à :

- familiariser l'étudiant avec la manipulation de données numériques, en particulier : fichiers **.mat**, fichiers **ascii** (chaines de caractères), fichiers images, fichiers audio, etc.
  - initier l'étudiant au développement de fonctions Matlab en utilisant les fichiers numériques sus-indiqués. Le contenu de ces fonctions concerne essentiellement certains rappels fondamentaux de la théorie de l'information qui sont nécessaires à la compression des données numériques.
- 

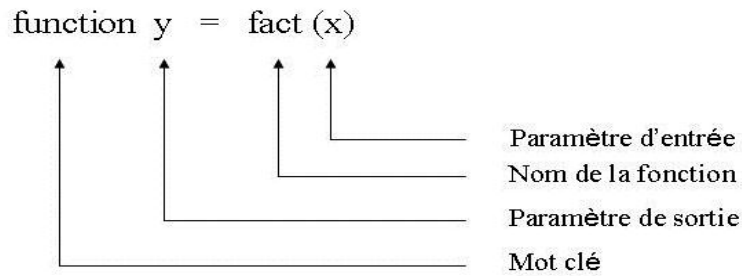
## 1. RAPPELS SUR LES FONCTIONS

L'exemple suivant montre les différentes parties constituant une fonction Matlab.

```
1  function f = fact(n)
2  % Calcule la valeur du factoriel
3  % fact(N) retourne le factoriel de N, souvent noté N!
4
5  -   clc;
6  -   close all;
7
8  %%% Corps de la fonction
9  -   f = 1;
10 -   for i = 1:n
11 -       f = f * i;
12 -   end
```

### 1) Function Definition Line

Cette ligne de définition de la fonction informe MATLAB que le fichier **.m** contient une fonction. Elle spécifie le nom de la fonction ainsi que le nombre et l'ordre des paramètres d'entrée et de sortie. Elle contient le mot clé **function** et doit toujours être la première ligne d'une fonction Matlab. Dans le cas de l'exemple précédent, on a :



## ■ *Nom de la fonction*

Doit commencer par une lettre et peut contenir des caractères alphanumériques. Sa taille ne doit pas dépasser la longueur autorisée par la fonction **namelengthmax**. Puisque les variables obéissent aux mêmes règles, on utilise la fonction **isvarname** pour vérifier la validité du nom des fonctions. Par exemple: **isvarname fact**

Les fonctions ne doivent pas porter le nom des mots clés de MATLAB. Utiliser la fonction **iskeyword** pour voir la liste de tous les keywords.

Si le nom du fichier qui contient la fonction MATLAB et le nom de la fonction défini dans la ligne de définition sont différents, le nom interne (de la fonction) est alors ignoré. Par exemple, si le nom du fichier contenant la fonction **fact** est **factoriel.m**, alors on appelle ou exécute la fonction en tapant **factoriel**. Ainsi, afin d'éviter toute confusion il est recommandé de donner au fichier **.m** le nom de la fonction défini dans la ligne de définition.

## ■ *Paramètres*

Si la fonction retourne plusieurs valeurs de sortie, on met la liste des paramètres entre crochets. Les paramètres d'entrée, s'ils existent, sont placés entre parenthèses à la suite du nom de la fonction. Utiliser des virgules pour séparer les paramètres multiples d'entrée ou de sortie. Si la fonction ne retourne aucun paramètre de sortie, utiliser des crochets vides ou bien laisser un blanc tout simplement. Par exemple:

**function** [] = printresult (x)    ou bien    **function** printresult (x)

## 2) **Line H1**

La ligne H1 est appelée ainsi car c'est la première ligne texte d'aide. C'est une ligne de commentaire suivant immédiatement la ligne de définition de la fonction. Puisque elle contient un texte commentaire, elle commence obligatoirement par le symbole **%**. Elle est la première ligne de texte qui apparaît lorsqu'on tape la commande **help** suivie du nom de la fonction.

## 3) **Help Text**

Il s'agit d'un help on ligne pour la fonction. C'est un texte de commentaires qui suit immédiatement la ligne **H1**. La première ligne sans le signe **%** termine le **help**.

## 4) **Function Body**

Le corps de la fonction contient tout le code MATLAB qui performe les calculs et affecte les valeurs aux paramètres de sortie. Il peut contenir, en outre, des commentaires et des lignes de blancs.

## 2. IMPORTER ET EXPORTER DES FICHIERS .mat

### 2.1. Exporter

Matlab fournit un format de fichier très pratique pour enregistrer et charger des données : les fichiers “.mat”. Les avantages de ces fichiers sont la rapidité de chargement et la petite taille qu’ils occupent. Il est possible de sauver tout le **workspace**.

▪ **Exemple :**

```
save data ;
```

- ✓ Toutes les variables du **workspace** seront enregistrées dans le fichier “data.mat” dans le dossier courant.
- ✓ Il est également possible de spécifier un chemin si le dossier existe, car il ne sera pas automatiquement créé. Il peut également être créé par Matlab à l’aide de la commande

```
[a,b,c] = mkdir ('newdossier') ;      save newdossier \ data
```

```
mkdir D:\chemin\newdossier\data ;   save D:\chemin\newdossier\data
```

- ✓ Pour sauvegarder uniquement certaines variables, il faut les lister à la suite du nom du fichier :

```
save variables var1 var2 var3
```

### 2.2. Importer

L’importation des données est effectuée par la commande **load**.

▪ **Exemple :**

```
load variables ;
```

- ✓ Toutes les variables préalablement sauvegardées sont restaurées avec leur nom d’origine. Afin de ne pas écraser d’autres variables présentes dans le **workspace**,

▪ **Exemple**

Ecrire la matrice A suivante et la sauvegarder dans le dossier courant.

```
A = [2 5 1 2 4 0 ; 2 5 4 3 0 0 ; 5 1 1 4 2 3 ; 3 2 4 5 1 3] ;
```

Taper successivement dans la fenêtre de commandes (**Command Window**), et observer le résultat :

```
save A ;
```

```
clear all ;
```

```
whos ;
```

```
load A ;
```

```
whos ;
```

### 3. FICHIERS ASCII

#### 3.1. Création d'une chaîne de caractères

Une chaîne de caractères est toute suite de caractères placés entre guillemets.

- **Exemple**

**chaîne = 'Ce TP concerne la compression des données numériques sous MATLAB' ;**

Une chaîne de caractères est considérée par MATLAB comme un vecteur ligne dont on peut calculer la taille, la longueur, le vecteur transposé, etc.

**size (chaîne) ; length (chaîne) ; chaîne'**

#### 3.2. Conversion d'une chaîne ASCII en nombres

On utilise la fonction **abs** qui retourne le module d'un complexe ou le code ascii d'un caractère.

- **Exemple**

**abs ('Ce TP concerne la compression des données numériques sous MATLAB') ;**

**abs (chaîne) ;**

#### 3.3. Conversion de nombres en chaîne ASCII

La commande **char** convertit un tableau d'entiers non négatifs en caractères. Réciproquement, connaissant les codes ASCII des caractères d'une chaîne, on peut obtenir la chaîne de caractères correspondante par la fonction **setstr**.

- **Exemple**

**S\_chaine = chaîne (1 : 14) ; S\_chaine = Ce TP concerne**

**s = abs (S\_chaine); s = [67 101 32 84 80 32 99 111 110 99 101 114 110 101]**

**m = char (s) ; m = Ce TP concerne**

**m = setstr (s) ; m = Ce TP concerne**

#### 3.4. Conversion décimale binaire et vis versa

- ✓ La commande **dec2bin** donne la représentation binaire d'un nombre décimal.
- ✓ La commande **bin2dec** convertit un nombre binaire sous forme de chaîne en sa valeur décimale.

- **Exemple**

**S\_chaine = chaîne (1 : 14) ; S\_chaine = Ce TP concerne**

**b = dec2bin (S\_chaine) ; c = dec2bin (S\_chaine , 8) ; d = bin2dec (b) ; d = bin2dec (c) ;**

S_chaine	s	b	c	d
C	67	1000011	01000011	67
e	101	1100101	01100101	101
	32	0100000	00100000	32
T	84	1010100	01010100	84
P	80	1010000	01010000	80
	32	0100000	00100000	32
c	99	1100011	01100011	99
o	111	1101111	01101111	111
n	110	1101110	01101110	110
c	99	1100011	01100011	99
e	101	1100101	01100101	101
r	114	1110010	01110010	114
n	110	1101110	01101110	110
e	101	1100101	01100101	101

### 3.5. Importer et exporter des fichiers ASCII

#### i. Création

La première étape est la création du fichier en utilisant la commande **fopen**. Les paramètres de la fonction sont le nom du fichier (avec le chemin d'accès) ainsi que la permission. Taper **help fopen** pour connaître les types de permissions possibles pour cette fonction. La fonction retourne un identifiant pour ce fichier.

**fid = fopen (filename, permission) ;**

#### ii. Ecriture des données formatées

L'écriture dans un fichier se fait avec la fonction **fprintf**. Les paramètres sont l'identifiant du fichier, le format, puis les données :

**taille = fprintf (fid, format, A, ...)**

L'argument du format est un texte contenant des caractères et des spécifications de conversion des données. La sortie est la longueur du texte écrit. Les spécifications de conversion contrôlent la notation, l'alignement, le nombre de chiffres significatifs, la largeur du champ, ainsi que d'autres aspects d'affichage. Pour en savoir davantage taper **help fprintf**.

Une autre fonction d'écriture dans un fichier ASCII est **fwrite**. Pour en savoir davantage taper **help fwrite**.

### ■ Exemple

```
chaine = 'Ce TP concerne la compression des données numériques sous MATLAB';  
fid = fopen('TP1chaine.txt', 'wt');  
count = fprintf(fid, '%s', chaine);  
fclose(fid);
```

### iii. Lecture d'un fichier

La fonction **fscanf** est l'équivalent de la fonction **fprintf** pour la lecture des fichiers.

**A = fscanf (fid, format)**

La fonction lit tout le fichier et converti les données au format spécifié, puis les retourne à la matrice A.

**[A , longueur] = fscanf (fid, format, size)**

Une autre fonction de lecture de fichiers ASCII est **fread**. Pour en savoir davantage taper **help fread**.

### ■ Exemple

```
fid = fopen('TP1chaine.txt', 'r');  
w = fscanf(fid, '%c');  
fclose(fid);
```

## 4. LECTURE/ECRITURE D'UNE IMAGE

Pour lire une image à partir d'un fichier graphique on utilise la commande **imread**. Pour en savoir davantage taper **help imread**.

**a = imread (filename, format)**

Pour écrire une matrice image dans un fichier graphique, on utilise la commande **imwrite**. Pour en savoir davantage taper **help imwrite**.

**imwrite (a, filename, format)**

## 5. LECTURE/ECRITURE D'UN FICHIER AUDIO

Pour lire un fichier son **'.wav'**, on utilise la commande **wavread**. Pour lire un fichier son **'.au'**, on utilise la commande **auread**. Pour en savoir davantage taper **help wavread ; help auread**.

**a = wavread (file) ; a = auread (file)**

Pour écrire un fichier son **'.wav'**, on utilise la commande **wavwrite**. Pour écrire un fichier son **'.au'**, on utilise la commande **auwrite**. Pour en savoir davantage taper **help wavwrite ; help auwrite**.

**wavwrite (a, f, nbits, wavefile) ; auwrite (a, f, nbits, audiofile)**