

Importation/Exportation de données vers Tableur

Le type fondamental de données de Matlab est le tableau/matrice. Ce type est très proche des feuilles Excel destinées à créer des tables de données. De nombreux autres logiciels utilisent des formats d'échange voisins des concepts utilisés sur Excel.

L'échange de données peut se faire de plusieurs manières:

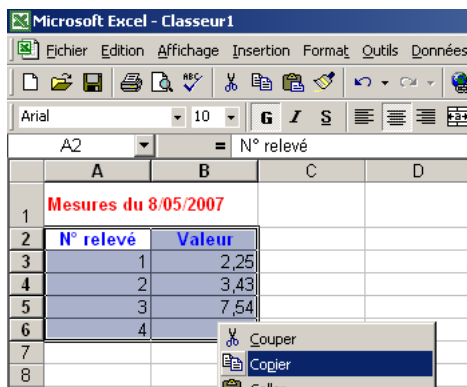
- par la méthode "copier/coller" d'un logiciel vers l'autre (peu conseillé).
- par enregistrement sur disque (fichier) depuis l'application source des données puis relecture sur l'autre application. Matlab met à disposition de l'utilisateur divers outils d'automatisation (wizard, opérations déclenchées par programmation/macro-commande)
- par échange direct de données d'une application à l'autre sous forme échange de type DDE (Direct Dynamic Exchange).

Importation et exportation directe par copier/coller

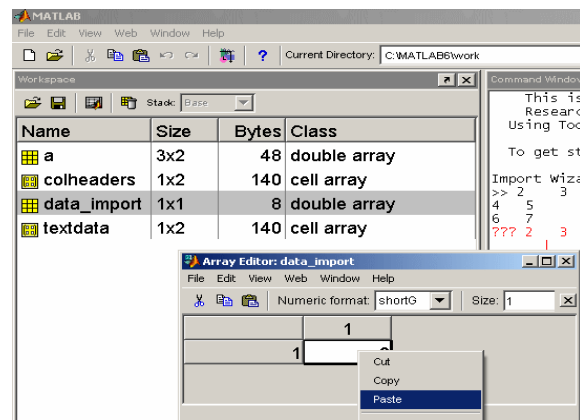
Les données de Matlab étant des tableaux analogues de ceux d'Excel, il semble possible d'utiliser directement "copier/coller". Attention: seules les données numériques sont "collables".

Excel-->Matlab :

- Sélectionner dans Excel les données à transférer , puis faire "copier"
- Dans Matlab, créer une variable non-vide, par exemple `data_import = 0;` (les données initiales seront écrasées par le collage)
- Sélectionner cette variable dans le workspace et demander son édition (double clic sur la variable). Par simple "clic droit" dans l'éditeur de la variable d'importation, sélectionner "paste" (collage). Les données sont recopiées dans la variable Matlab.



	A	B	C	D
1	Mesures du 8/05/2007			
2	N° relevé	Valeur		
3	1	2,25		
4	2	3,43		
5	3	7,54		
6	4			
7				
8				

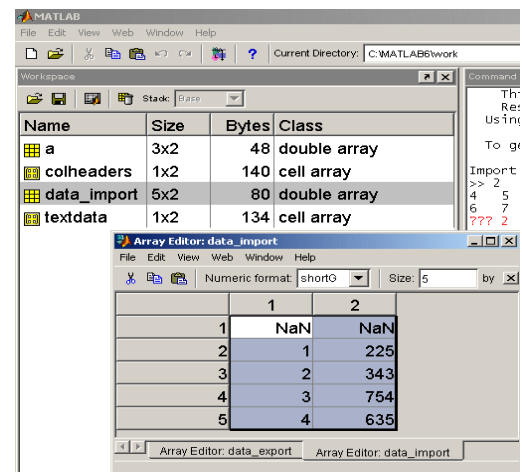


Name	Size	Bytes	Class
a	3x2	48	double array
colheaders	1x2	140	cell array
data_import	1x1	8	double array
textdata	1x2	140	cell array

Array Editor: data_import
1

Le collage sera en échec si les données ne sont pas utilisables par Matlab (données textuelles d'en-tête de colonne par exemple). Ces données sont signalées comme "NaN" (Not a Number)

Résultat de l'exemple: (attention: il faut forcer les nombres à virgule avec un nombre de chiffres après la virgule fixe, la virgule n'étant pas correctement interprétée sur les version 6 et 7).



Name	Size	Bytes	Class
a	3x2	48	double array
colheaders	1x2	140	cell array
data_import	5x2	80	double array
textdata	1x2	134	cell array

Array Editor: data_import		
	1	2
1	NaN	NaN
2	1	225
3	2	343
4	3	754
5	4	635

Le "copier/coller" Matlab--> Excel pose encore plus de problème : la virgule est mal interprétée ainsi que les exposants (les nombres sont restitués sous forme chaîne!)

Note: pour passer des chaînes de caractères vers Matlab, il faut utiliser une structure de cellules, chaque cellule contenant un texte (création de la structure initiale par: `data_import = { ' ' } ;`).

Importation par fichier Excel

Matlab peut lire directement un fichier Excel, sous réserve que son contenu représente un tableau de données. La procédure de travail est la suivante:

- mettre les données en tableau dans une feuille Excel (un seul tableau sur une seule feuille dans le classeur), de préférence sans en-tête (ne pas mettre ni titre dans la feuille, ni commentaire ni tout autre information que le tableau des données).

- enregistrer les données Excel avec l'option "Excel97"

- reprendre ces données sous Matlab avec l'instruction :

```
mes_données = xlsread('mon_fichier.xls');
```

Exemple:

	A	B	C
1	Pierre	1,92	14
2	Paul	1,75	12
3	Jacques	1,55	10
4	Henri	1,76	13
5			

```
>> [data , noms] = xlsread('c:\data_matlab.xls')
```

```
data =      1.9200  14.0000
           1.7500  12.0000
           1.5500  10.0000
           1.7600  13.0000
```

```
noms =      'Pierre'
           'Paul'
           'Jacques'
           'Henri'
```

Sur version 2007, il est possible de sélectionner la feuille et une partie d'un tableau :

```
donnees = xlsread('c:\mafeuille.xls' , 'feuille2' , 'a2:j5' )
```

La relecture des dates pose aussi problème , étant donné des formats différents. La proposition de Matlab est la suivante :

```
excelDates = xlsread('mon_fichier.xls')
matlabDates = datenum( '30-Dec-1899' ) + excelDates
les_dates = datestr(matlabDates , 2)
les dates      04/12/00
                04/13/00
                04/14/00
```

Importation par fichier texte

L'importation par fichier texte est une solution simple et éprouvée. Elle permet de transférer de grandes quantités de données *numériques*.

La procédure de travail est la suivante:

- mettre les données dans une feuille Excel (une seule feuille dans le classeur), de préférence sans en-tête (ne pas mettre ni titre dans la feuille, ni commentaire ni tout autre information que le tableau des données).

- enregistrer les données Excel avec l'option "Texte" (format txt) ou "Texte avec séparateur" (format csv)

- reprendre ces données sous Matlab avec l'instruction :

```
load c:\mes_datas.txt          la variable créée prend le nom du fichier
ou mes_datas = load c:\monfichier.txt
```

Les difficultés sont nombreuses (point décimal...) ; il est conseillé de passer par un éditeur de texte (Notepad, Wordpad ou autre...) pour mettre en forme visuellement les données.

Tout autre moyen pour écrire de tels fichiers texte est possible, en particulier, certains logiciels prévoient la sauvegarde en fichier texte .

Lecture par fonctions Entrée/Sortie Matlab

Pour des fichiers particuliers, la lecture de fichiers texte peut se faire ligne à ligne avec analyse de la ligne lue pour en retirer les données utiles.

La lecture ligne à ligne d'un fichier texte se fait par :

```
id_fich = fopen('mon_fichier.txt')
ligne_lue = '';
while (ligne_lue ~= -1)
    ligne_lue = fgets(id_fich);
    mon_analyse(ligne_lue)
end
fclose(id_fich)
```

L'analyse des lignes lues pourra utiliser la fonction `sscanf` (lecture de données formatées) ou les fonctions `strfind` ou `strread` pour rechercher dans la ligne les "balises" permettant de récupérer les données (convertir les données textuelles numériques en nombre par `str2num`)

Exportation de données

Les principes sont voisins de ceux de l'importation par fichier. Les deux formats sont :

- exportation directe en format Excel97 d'un tableau Matlab (version 7 et sous réserve que Excel soit installé):

```
no_erreur = xlswrite( 'c:\monclasseur.xls' , mon_tableau_Matlab , 'Feuille1' , 'A2:C4' )
```

Le tableau peut contenir des nombres ou des chaînes; il est possible d'exporter aussi des structures "cells". Si le bloc de cellule est de taille insuffisante, seule la partie correspondante du tableau est copiée. Le nom de la feuille peut être omis; Matlab crée automatiquement une nouvelle feuille si le fichier Excel existe déjà. En l'absence d'Excel, la sauvegarde est faite en format *csv* .

- exportation en format texte d'un tableau

```
save nom_du_fichier.txt ma_matrice -ascii
```

Echange dynamique DDE

L'échange dynamique a pour objet l'écriture directe de données dans une feuille Excel depuis Matlab ou inversement la lecture directe depuis une feuille.

La liaison directe s'appelle un lien. C'est une structure d'échange dans laquelle les deux applications Matlab et Excel sont ouvertes simultanément. Matlab peut écrire directement dans les cellules Excel, sans intervention de l'opérateur sur Excel. Les données écrites sur la feuille Excel peuvent être sauvegardées comme pour toute feuille Excel.

- Ecriture dans une feuille Matlab:

```
mon_channel = ddeinit('Excel' , 'mon_classeur.xls')
ok = ddepoke(mon_channel, 'r5c1:r10c5', ma_matrice);
ok = ddeterm(mon_channel)
```

- Lecture depuis une feuille Matlab:

```
mon_channel = ddeinit('Excel' , 'mon_classeur.xls')
ma_matrice = ddereq(mon_channel, 'r1c1:r5c5');
ok = ddeterm(mon_channel)
```

Ces techniques d'échange ont un intérêt pour utiliser Matlab comme outil de calcul sur des données Excel qui sert d'interface utilisateur. Dans ce cas, Matlab doit rester actif et "boucler" sur sa fonction de calcul, ce qui est consommateur de ressource CPU.

Pour éviter de charger inutilement le CPU, le mécanisme DDE prévoit un signalement à Matlab d'un changement de valeur dans Excel. Le changement provoque l'exécution d'une fonction de "callback" dans Matlab. Si le paramètre 'nom de variable Matlab' ('mon_tableau' dans l'exemple) est défini dans la fonction, la valeur de la plage de cellule (matrice) est passé à Matlab; sinon, seule la fonction de callback est exécutée.

```
mon_channel = ddeinit('Excel' , 'mon_classeur.xls')
ok = ddeadv(channel, 'r1c1:r5c5', 'ma_fonction(mon_tableau)', 'mon_tableau');
```

Le lien est arrêté par :

```
ok = ddeunadv(channel, 'r1c1:r5c5');
```

Ne pas oublier de fermer la liaison DDE avec Excel.