Programmer avec Matlab

Les entrées-sorties

1. Les formats d'affichage :

« La commande disp »

On utilise la commande **disp** avec **un tableau** qui est une chaîne de caractères pour afficher un message, comme On peut l'utiliser aussi pour afficher un résultat

```
Command Window
>> A=magic(4);
>> disp('Calcul du déterminant de la matrice A')
Calcul du déterminant de la matrice A
>> disp(['Le déterminant de la matrice A vaut ', num2str(det(A))])
Le déterminant de la matrice A vaut -1.4495e-12
fx >>
```

N.B: un tableau doit être d'un type donné, les éléments d'un même tableau ne peuvent donc être des chaînes de caractères et des valeurs numériques. On a donc recours à la commande **num2str** (<< **number to string** >>) pour convertir une valeur numérique en une chaîne de caractères

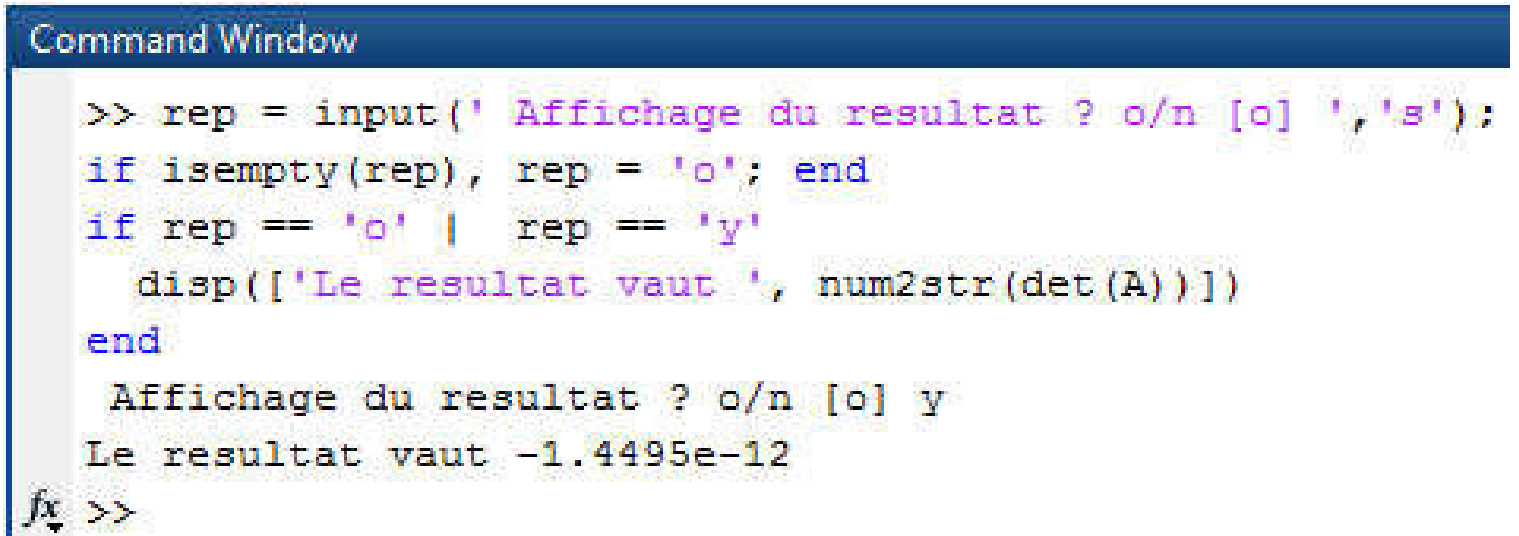
Les entrées-sorties

2. Lecture :

« La commande input »

La commande **input** permet de demander à l'utilisateur d'un programme de fournir des données. La syntaxe est : var = input('une phrase '). La phrase une phrase affichée et MATLAB attend que l'utilisateur saisisse une donnée au clavier. Cette donnée peut être une valeur numérique ou une instruction MATLAB , pour la rendre sous format de chaine de caractère on utilise :

var = input(' une phrase ','s')



Les entrées-sorties

3. Impressions dirigées par format :

« La commande sprintf »

La commande **sprintf** permet l'impression de variables selon un modèle donné. Un modèle d'édition se présente sous la forme du symbole pourcentage **(%)** suivi d'indications permettant de composer le contenu du champ à imprimer, en particulier sa longueur en nombre de caractères.

La syntaxe de la commande **sprintf** est : sprintf(format, variables)

où

- **variables** : est le nom des variables à imprimer suivant le modèle d'édition spécifié dans format;
- **format** : est le format d'édition. Il s'agit d'une chaîne de caractères contenant les modèles d'éditions des variables à imprimer.

Les entrées-sorties

3. Impressions dirigées par format :

« La commande sprintf »

1. **Modèle d'édition de caractères :**

Un modèle d'édition de caractères est de la forme **%Ls** où

- % est le symbole de début de format et
- s le symbole précisant que la donnée est de type chaîne de caractères.
- L est un entier donnant la longueur total du champ (en nombre de caractères).

Par défaut le champ est justifié à droite (si la longueur de la chaîne de caractères est plus petite que la longueur L du champ, des espaces sont insérés après la chaîne de caractères).

Le symbole - (moins) juste après le symbole % permet de justifier à gauche. En l'absence de l'entier L la longueur totale du champ est égale au nombre de caractères de la chaîne.

Les entrées-sorties

3. Impressions dirigées par format :

« La commande sprintf »

1. **Modèle d'édition de caractères « Exemple » :**

Command Window

```
>> sprintf('%s', 'il fera beau à Al Hoceima')

ans =

il fera beau à Al Hoceima
```

```
>> temps = 'il fera beau à Al Hoceima';
>> sprintf('%s',temps)

ans =

il fera beau à Al Hoceima
```

```
>> sprintf('%60s',temps)

ans =

                                il fera beau à Al Hoceima
```

```
>> sprintf('%-30s',temps)

ans =

il fera beau à Al Hoceima
```

```
>> sprintf('Météo : %-30s',temps)

ans =

Météo : il fera beau à Al Hoceima
```

Les entrées-sorties

3. Impressions dirigées par format :

« La commande sprintf »

2. **Modèle d'édition des réelles :**

Un modèle d'édition de réel est de la forme %+- L.D t, où :

- % est le symbole de début de format,
- L est un entier donnant la longueur total du champ (en nombre de caractères, point virgule compris),
- D est le nombre de décimales à afficher
- t spécifie le type de notation utilisée.

→ Le symbole - (moins) permet de justifier à gauche. Le symbole + (plus) provoque l'affichage systématique d'un signe + devant les réels positifs.

Les entrées-sorties

3. Impressions dirigées par format :

« La commande sprintf »

2. **Modèle d'édition des réelles :**

Un modèle d'édition de réel est de la forme %+- L.D t :

Les principales valeurs possibles pour t sont les suivantes:

- %d : Pour les entiers
- %e : Pour une notation à virgule flottante où la partie exposant est délimitée par un e minuscule (ex: 3.1415e+00)
- %E : Même notation que e mais avec E
- %g : La notation la plus compacte entre la notation à virgule flottante et la notation à virgule fixe est utilisée

Les entrées-sorties

3. Impressions dirigées par format :

« La commande sprintf »

2. Modèle d'édition des réelles « Exemple » :

```
>> sprintf('sin(%8.6f) = %f', x,y)

ans =

sin(1.047198) = 0.866025
```

```
>> x = [1:10];
>> sprintf(' %d ,',x)

ans =

 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 , 10 ,
```

→ Si l'on a besoin d'afficher le caractère **%** on le doublera **%%** pour qu'il ne soit pas interprété comme le début d'un format. La commande **fprintf** est l'analogue de **sprintf** pour imprimer de variables selon un modèle donné dans un fichier.

Les instructions de contrôle

1. La boucle « for » (Parcours d'intervalle) :

Une possibilité pour exécuter une séquence d'instructions de manière répétée consiste à effectuer une boucle pour les valeurs d'un indice, incrémenté à chaque itération, variant entre deux bornes données. Ce processus est mis en œuvre par la **boucle for.**

Syntaxe :

```
for indice=borne_inf : borne_sup
    séquence d'instructions
end
```

Où

- *indice* : une variable appelée l'*indice de la boucle*;
- *borne_inf* et *borne_sup* : 2 constantes réelles (appelées *paramètres de la boucle*);
- *séquence d'instructions :* le traitement à effectuer pour les valeurs d'indices variant entre *borne_inf* et *borne_sup* avec un incrément de 1.

Les instructions de contrôle

1. La boucle « for » (exemple) :

Voici un exemple d'utilisation d'une boucle pour calculer *n*!

Command Window

```
>>  n = 4;
nfac = 1;
for k = 1:n
     nfac = nfac*k;
   end
>> nfac

nfac =

      24
```