

# Chapitre O : BRÈVE INTRODUCTION À OCTAVE

## 1 Introduction

- Introduction
- Manipulations de base
- Opérations matricielles
- Constructions
- Fonctions et fichiers \*.m

# INTRODUCTION

- Qu'est-ce que Octave ?

D'après le site web du projet <sup>1</sup>, *GNU Octave is a high-level interpreted language, primarily intended for numerical computations.*

---

1. <http://www.gnu.org/software/octave/>

# INTRODUCTION

- Qu'est-ce que Octave ?

D'après le site web du projet <sup>1</sup>, *GNU Octave is a high-level interpreted language, primarily intended for numerical computations.*

- ▶ **interpreted** :  
chaque instruction est interprétée séparément (similaire à une calculatrice) ;
- ▶ **intended for numerical computations** :  
conçu pour le calcul numérique

- **Caractéristiques** :

- ▶ logiciel libre
- ▶ similaire à Matlab (sauf que ce dernier est commercial)
- ▶ déclaration implicite de variables
- ▶ syntaxe proche des notations mathématiques
- ▶ nombreuses fonctions de calcul numérique

- **Plus d'information** : cf. site du cours, et en particulier

Eléments de syntaxe Octave ([an\\_syntaxe\\_octave.pdf](#)) .

---

1. <http://www.gnu.org/software/octave/>

# MANIPULATIONS DE BASE

- Les manipulations se font dans un **terminal Octave**.
- **Démarrage** : cliquer sur l'icône dédiée ou entrer la commande **octave** dans un terminal unix.
- **Sortie** : commandes **exit/quit** (utilisez **Ctrl+c** pour forcer la sortie)
- Octave choisit par défaut un **répertoire de travail** ; utilisez
  - ▶ **pwd** pour afficher sa localisation
  - ▶ **ls** pour afficher son contenu
  - ▶ **cd nouveau répertoire** pour aller vers un nouveau répertoire
  - ▶ **cd ..** pour aller vers le répertoire parent
  - ▶ **mkdir répertoire** pour créer un répertoire (et **rmdir** pour le supprimer)

La plupart des commandes (y compris celles déjà mentionnées) sont documentées ; la documentation est affichée avec

**help commande**

Exemple :

```
help pwd % ce que fait pwd
pwd % afficher répertoire de travail
mkdir calnum % creer répertoire calnum
cd calnum % aller dans calnum
cd .. % sortir
rmdir calnum % effacer calnum
```

# OPÉRATIONS MATRICIELLES

```
c = [1; -1; 0]           % c - vecteur-colonne
l = [1  -1  0]           % l - vecteur-ligne
M = [1 2 3; 4 5 6]       % M - matrice 2x3
M'                        % transposée de M
M'*M                     % mult. matricielle
M.*M                     % mult. élément par élément
c(1)                     % élément 1 du vecteur c
M(1,1)                   % élément 1,1 de la matrice M
v = 1:3                  % => v = [1, 2, 3]
w = 1:3:10                % => w = [1, 1+3, 1+6, 1+9]
N=M(:,1:2)               % sous-matrice principale 2x2 de M
N^2                      % = N*N (^ pour exposant)
N.^2                     % = N.*N
length(c)                % longueur de c (= 3)
[n,m]=size(M)            % n - nbr. lignes, m - nbr. col.
norm(c)                  % norme vectorielle euclidienne
norm(M)                  % norme matricielle euclidienne
```

illustrations et compléments  $\Rightarrow$  travaux pratiques!

# CONSTRUCTIONS

- Conditions

```
if (condition 1)
    code pour if
elseif (condition 2)
    code pour premier elseif
else
    code pour else
end
```

```
if (x<=2 && x>=0)
    disp('x entre 0 et 2')
elseif (x > 0)
    disp('x superieur à 2')
else
    disp('x est negatif')
end
```

- Boucle for

```
for i = vect
    code pour for
end
```

```
for i = 1:10
    disp(i)
end
```

- Boucle while

```
while(condition 1)
    code pour while
end
```

```
i=1;
while (i <= 10)
    disp(i) ; i++ ;
end
```

# FONCTIONS ET FICHIERS \*.m

Multiples instructions peuvent être groupées dans les fichiers avec une extension \*.m.

Un fichier `nom.m` peut contenir :

- un `script`
  - ⇒ l'instruction `nom` équivaut à copier-coller le contenu du fichier dans le terminal octave
- une `fonction`
  - ⇒ `nom` doit correspondre au nom de la fonction

Les `fonctions` qui peuvent être définies :

- en ligne de commande (fonctions simples)
  - ex : `nom = @(x) sin(2*x);`
- dans un fichier `nom.m` (ci-contre)

```
% fichier nom.m
function y = nom (x)
y = sin(2*x);
```

Dans les deux cas, l'instruction `nom(pi)` sera équivalente à `sin(2 $\pi$ ) (= 0)`.

**Attention :** les fonctions peuvent être passées comme arguments à d'autres fonctions; la syntaxe est alors différente pour ces deux types de définitions.

⇒ cf. `an_syntaxe_octave.pdf` pour plus de détails.