

# TP 1 : prise en main de GNU Octave

GNU Octave (voir <http://www.gnu.org/software/octave>) est un interpréteur de commandes. Il est capable de comprendre une certaine quantité d'instructions algorithmiques si ces dernières respectent un format bien déterminé. Le langage d'Octave (qui est très similaire à celui de MatLab, solution commerciale plus connue dans l'industrie) est particulièrement adapté à l'analyse numérique. Ceci en fait un outil puissant tant pour les calculs complexes que pour la visualisation de données.

Lorsque vous démarrez GNU Octave, un terminal de commande prêt à recevoir du code GNU Octave se lance (cf. Figure 1). C'est dans cette fenêtre que vous demanderez à GNU Octave d'exécuter vos programmes pour voir le résultat. Cet interpréteur vous permet d'effectuer des calculs en les tapant directement dans la console. On valide une opération en tapant sur entrée.

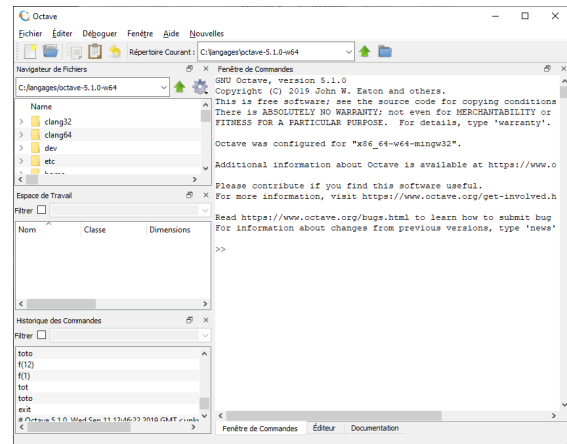


Figure 1: Environnement GNU Octave 5.1.0

Lorsqu'une opération est effectuée, le système renvoie la réponse sous cette forme :

*ans* = ...

Si on désire programmer une séquence d'opérations en une seule fois, il suffit de séparer chaque opération par le caractère « ; ». Toutefois, ce mode ne permet de connaître **que le résultat de la dernière opération** (cf. Figure 2).

```
>> a=45;b=55;a+b
ans = 100
>> a=45;b=55;a-b
ans = -10
>> |
```

Figure 2 : Séquence d'opérations dans une console Octave

## Exercice de démarrage

Saisissez les commandes suivantes dans l'interpréteur Octave

```
a = 28;
b = 62;
a + b
c = a + b;
```

Qu'en déduisez-vous ? Que fait cette suite de commande ? Quelles sont les variables définies ? (vous pouvez utiliser la commande « whos »).

Tapez « a » dans l'invite de commande. Quelle est sa valeur ?

Maintenant tapez « `clear` ». Quelle est maintenant la valeur de « `a` » ? Qu'en déduisez-vous sur l'effet de la commande `clear` ?

A l'aide de GNU Octave, résolvez l'équation suivante :

$$\sqrt{8} + 94x = 45$$

Trouvez la valeur de `x` et mettez-la dans une variable du même nom.

Avec Octave  
 $\sqrt{8}$  s'écrit  
`sqrt(8)`

## Programmer dans un fichier .m

Vous avez pu observer que l'édition du texte dans la console est particulièrement **limitée** et **inefficace**. Il est donc à réserver pour les programmes courts et pour les tests ponctuels. Pour pallier à ce problème, il est possible d'éditer vos programmes dans un fichier (.m) et de demander à octave d'exécuter le code qui est à l'intérieur. Pour travailler de la sorte, suivez les étapes suivantes :

1) Créer un répertoire sous Windows où les fichiers contenant vos programmes seront stockés (*dans votre espace personnel par exemple*).

2) Spécifier à GNU Octave l'emplacement de ce répertoire. Pour cela il est nécessaire de connaître l'arborescence de fichier de votre ordinateur. Vous pouvez la visualiser avec n'importe quelle fenêtre Windows. Par défaut, Octave s'exécute depuis le répertoire qui le contient, il est donc capable d'accéder uniquement aux fichiers de ce répertoire (C:\Octave\4.0.3\bin dans l'exemple de la Figure 3.). Deux solutions sont possibles pour spécifier votre répertoire de travail dans Octave :

## Positionnement du répertoire de travail en ligne de commande

Utilisez la commande `pwd` (Path to Working Directory) pour connaître le répertoire dans lequel vous vous situez (par défaut le répertoire où Octave est installé sur l'ordinateur)

Utilisez la commande `cd` pour vous positionner dans le répertoire de votre choix.

**ATTENTION** : il est fortement recommandé de choisir un chemin d'accès à votre répertoire **où il n'y a pas d'espace** (par exemple évitez quelque chose de ce genre : « C:\Les TPs\Exercice 1\ »)

```
>> pwd
ans = c:\dev\octave
>> |
```

**Figure 3 : Exemple d'utilisation de commandes dans GNU Octave**

Une autre option consiste à utiliser l'environnement graphique de Windows pour définir votre répertoire de travail (c'est déjà un peu plus pratique). Comme le montre la Figure 4, vous pouvez glisser l'icône de votre dossier personnel dans GNU Octave. Cela vous évite de le taper et de faire des erreurs éventuelles. Comme pour la version ligne de commande, précédez le chemin vers votre dossier personnel de la commande `cd` et tapez entrée.

## Positionnement du répertoire de travail graphiquement

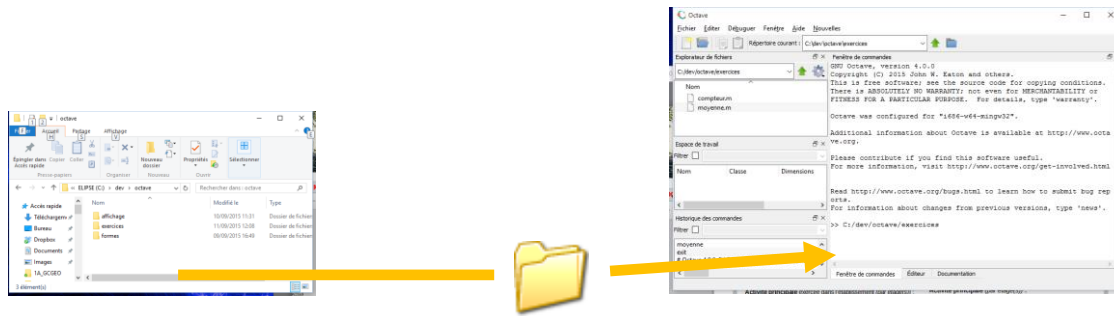


Figure 4 : Glisser le chemin d'accès vers son répertoire de travail graphiquement

Une fois GNU Octave positionné dans votre répertoire personnel, vous n'avez plus qu'à exécuter les programmes que vous aurez préalablement saisis dans un fichier texte. Pour cela, il vous suffit de taper dans la fenêtre GNU Octave le nom du fichier contenant votre programme pour qu'il s'exécute. Si votre fichier contenant votre programme s'appelle `test.m`, il suffit de taper `test` dans GNU Octave et il s'exécute. Voici à quoi votre environnement de travail final devrait ressembler (cf. figure 5).

Avec Octave  
les nombres complexes  
s'écrivent  $a + i * b$   
tout simplement

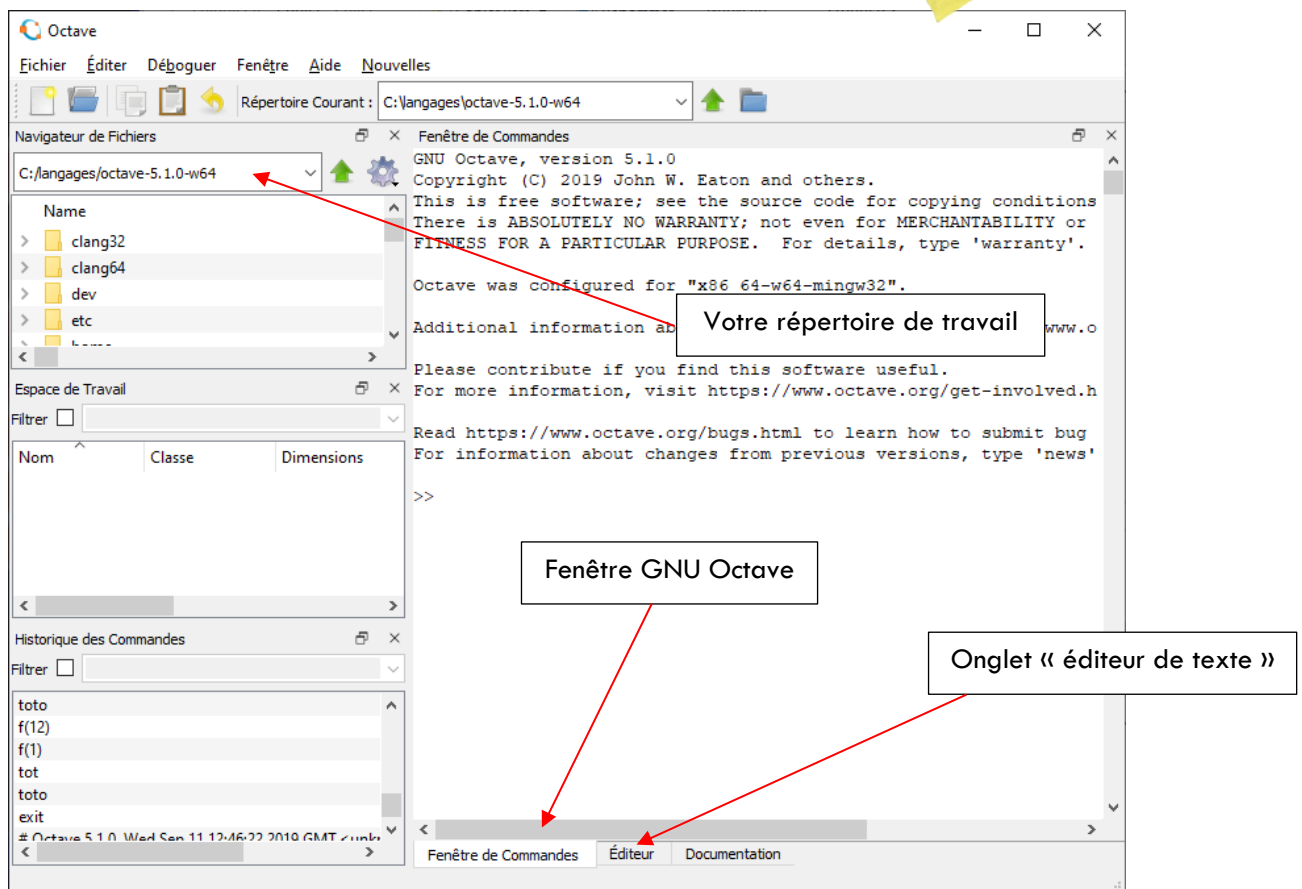


Figure 5 : environnement de travail GNU Octave 5.1.0

**Attention :** dans les programmes GNU Octave, vous pouvez sauter des lignes mais seuls les résultats des lignes qui ne sont pas terminées par le caractère « ; » seront affichés...

## Exercice de démarrage, la suite :

Reprenez l'exercice précédent et insérez les commandes suivantes dans un fichier `.m` (utilisez l'onglet *Editeur*), sauvez le sous le nom « `essai.m` » puis exécutez-le.

```
a = 28 ;
b = 62 ;
a + b
c = a + b ;
```

Une fois le programme exécuté, tapez « `a` », puis « `c` ». Quelles sont leurs valeurs ?

- Comme une calculatrice, GNU Octave connaît déjà une quantité de fonctions de manière native telles que `cos`, `log` (logarithme népérien), `log2` ou `log10` ainsi qu'une quantité de constantes telles que `pi`. Saisissez et donnez le résultat de l'opération suivante :

$$1.5 * \cos(90 * \frac{\pi}{180}) + \log(8.12^2) ;$$

- On considère maintenant les nombres complexes suivants :

$$a = \sqrt{3} + i$$

$$b = \sqrt{2} - i\sqrt{2}$$

Déterminer le module (fonction `abs`) et l'argument (fonction `arg` ou `angle`) de `a`, `b` et `a/b`

- Soit le prix en euros suivant : 9,99 € TTC. Calculez son prix hors taxes (avec une taxe appliquée de 20 %).

## L'affichage basique de données

Il faut savoir dissocier l'affichage et le contenu « véritable » des variables. La plupart du temps, un affichage approché des valeurs est suffisant pour l'utilisateur.

**Par défaut**, GNU Octave **n'affiche que les 4 premières valeurs** après la virgule (affichage de la valeur approchée donc).

Il est possible de modifier ce paramètre en tapant « **format long** » pour afficher **15** valeurs significatives (et revenir au paramètre par défaut en tapant « **format short** »).

Néanmoins, Octave stocke en réalité beaucoup plus de valeurs (**48**) après la virgule qu'il est possible d'afficher avec des fonctions particulières.

Une fonction proche du **langage C** est très utile pour afficher ces données : la fonction « **printf** » Tapez :

```
pi
format long
pi
printf("%.10f\n",pi);
printf("%.50f\n",pi);
```

Quel sont les résultats affichés ? **Qu'en déduisez-vous ?**

**Nota :** La partie « `%.10f\n` » correspond à la sortie « formatée » de la fonction ; `%.10f` signifiant que nous souhaitons afficher un nombre flottant (à virgule) avec 10 chiffres après la virgule et `\n` signifiant que nous voulons revenir à la ligne après l'affichage (`\n` → *newline* – revenir à la ligne).

Le caractère `\t` permet de créer une tabulation.

**Exercice :** Créer un « vecteur » entre les valeurs 0 et 3 avec un pas de 0,2. Afficher chaque valeur du vecteur séparée par une tabulation avec 5 valeurs maximum par ligne.

## Un exercice complet de devinette pour finir

Ecrire dans un fichier `.m` dont vous choisirez le nom le script (programme) du jeu suivant :

L'ordinateur choisit au hasard un nombre entier entre 1 et 100. Le joueur (humain) doit découvrir ce nombre par essais successifs. A chaque essai, le programme indique au joueur (affiche) si la valeur saisie est supérieure ou inférieure à la valeur recherchée.

Vous utiliserez les instructions `rand()` pour générer un nombre aléatoire réel compris entre 0 et 1, et `ceil(a)` qui renvoie la partie entière par excès de l'argument `a`.

Afficher enfin dans un graphique (commande `plot(x,y)`) la valeur indiquée par l'utilisateur en fonction du numéro de l'essai (nous verrons la prochaine fois comment améliorer le processus d'affichage)

**Nota :** vous devrez stocker les valeurs données par l'utilisateur dans un « vecteur »

## Quelques commandes GNU Octave utiles

<b>cd</b>	<code>cd MonRepertoire</code> : vous descendez dans le répertoire <i>MonRepertoire</i>  <code>cd ..</code> : vous remontez dans le répertoire parent  <code>cd ~</code> : vous revenez directement dans le répertoire personnel
<b>clc</b>	Efface la console sans effacer pour autant vos variables (très pratique quand trop d'informations sont affichées et que l'on n'y voit plus rien).
<b>clear</b>	Efface la mémoire de GNU Octave, toutes vos variables sont effacées
<b>exit</b>	<code>exit</code> ferme GNU Octave (voir aussi la commande <code>quit</code> ).
<b>help</b>	<code>help</code> suivi du nom d'une fonction vous permet d'accéder à la documentation de la fonction en question (voir aussi la commande <code>lookfor</code> ).
<b>pwd</b>	Montre le répertoire de travail courant ( <i>idéalement</i> votre dossier personnel)
<b>dir</b>	Liste le contenu du répertoire de travail