

# Introduction au programmation Scilab

Professeur ESSADDOUKI Mostafa (essaddouki@gmail.com)  
<http://www.developpement-informatique.com>

10 mai 2018

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Qu'est-ce que Scilab?	2
1.2	Espace de travail	2
1.2.1	Console	2
1.2.2	L'éditeur	3
1.2.3	La fenêtre graphique	3
<b>2</b>	<b>Éléments de base du langage</b>	<b>5</b>
2.1	Expressions	5
2.2	Nom du variable	5
2.3	Commentaires et lignes de suite	6
2.4	Fonctions mathématiques	6
2.5	Types de données	7
2.5.1	Booléen	7
2.5.2	Nombres complexes	7
2.5.3	Entiers	7
2.5.4	Conversions entre entiers	7
2.5.5	Virgules flottantes	9
2.5.6	Chaines de caractères	9
2.6	Fonctions de lecture et ecriture	9
2.6.1	disp	9
2.6.2	input	9

# 1 Introduction

## 1.1 Qu'est-ce que Scilab ?

**Scilab** est un logiciel libre de calcul numérique multi-plate-forme fournissant un environnement de calcul pour des applications scientifiques.

Ce qui suit est une courte liste de ses capacités :

- Algèbre linéaire
- Polynômes et fonctions rationnelles,
- Interpolation, approximation,
- Résolution des équations différentielles,
- Traitement de signal
- Statistiques

## 1.2 Espace de travail

L'espace de travail de Scilab est constitué de plusieurs fenêtres :

- La console pour faire des calculs,
- L'éditeur pour écrire des programmes,
- Les fenêtres graphiques pour l'affichage des graphiques,
- L'aide incorporée.

### 1.2.1 Console

Après avoir double-cliqué sur l'icône pour lancer Scilab, l'environnement Scilab se compose par défaut des fenêtres suivantes - console, navigateurs de fichiers et de variables, historique des commandes. Il est possible de revenir à tout moment

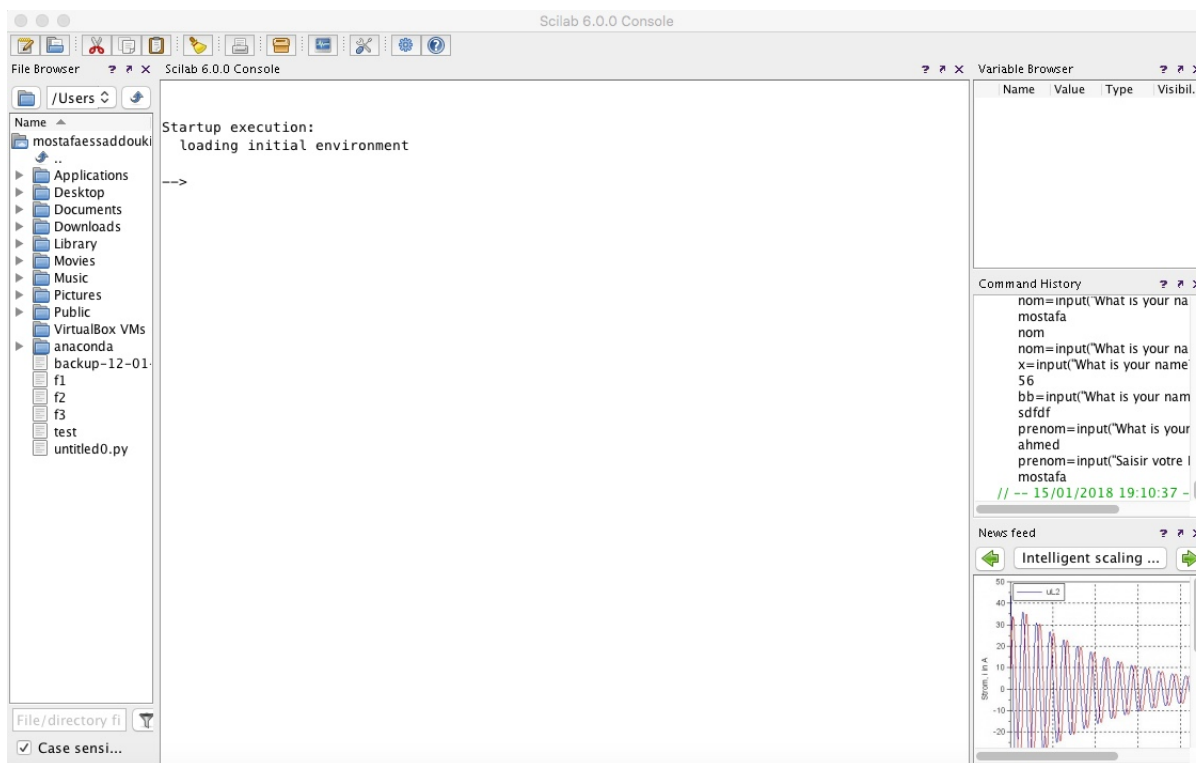


FIGURE 1 – Page d'accueil de Scilab

avec les flèches du clavier  $\leftarrow \uparrow \rightarrow \downarrow$  ou avec la souris. Les touches gauche et droite sont utilisées pour changer les instructions et les touches haut et bas sont utilisées pour revenir à une commande précédemment exécutée.

Dans la console après l'invite "-->", tapez simplement une commande et appuyez sur la touche Entrée pour obtenir le résultat correspondant.

```
1 --> 3+5
2   ans =
3     8.
4
5 --> 5/4+3
6   ans =
7     4.25
```

### 1.2.2 L'éditeur

Taper directement dans la console a deux inconvénients : il n'est pas possible de sauvegarder les commandes et il n'est pas facile d'éditer plusieurs lignes d'instruction. L'éditeur est l'outil approprié pour exécuter plusieurs instructions.

#### Ouverture de l'éditeur

Pour ouvrir l'éditeur à partir de la console, cliquez sur la première icône de la barre d'outils ou sur **Applications**> **SciNotes** dans la barre de menus. L'éditeur s'ouvre avec un fichier par défaut nommé "**Untitled 1**".

#### Écrire dans l'éditeur

Taper dans l'éditeur est comme dans n'importe quel traitement de texte. Dans l'éditeur de texte, les parenthèses d'ouverture et de fermeture, les boucles de fin, les fonctions et les commandes de test sont ajoutées automatiquement. Cependant, ces fonctionnalités peuvent être désactivées dans **Options**> **Auto-complétion** dans le menu, en cliquant sur les deux entrées ci-dessous activées par défaut :

- (, [, ...
- si, fonction, ...

Alors qu'en principe chaque instruction doit être entrée sur une ligne séparée, il est possible de taper plusieurs instructions sur une même ligne séparant chaque instruction avec un point-virgule ";" ". Un espace au début de la ligne appelé indentation est automatique lorsqu'une boucle ou un test est démarré.

#### Remarque 1

- *Les commentaires précédés de "//" ne seront pas pris en compte dans les calculs.*
- *Pour changer la police, cliquez sur **Options**> **Préférences**.*
- *Lors de l'écriture d'un programme, l'indentation est automatique. Si ce n'est pas le cas, cliquez sur **Edition**> **Corriger l'indentation** pour la restaurer (Ctrl I sous Windows et Linux ou Commande I sous Mac OS X).*

#### sauvegarde

Tout fichier peut être sauvegardé en cliquant sur **File** > **Save as**. L'extension ".sce" à la fin d'un nom de fichier lancera automatiquement Scilab lors de l'ouverture (sauf sous Linux et Mac OS X). **Copie dans la console, exécution d'un programme** En cliquant sur Exécuter dans la barre de menu, trois options sont disponibles :

- Exécuter "...file with no echo" (Ctrl Shift E sous Windows et Linux, Cmd Shift E sous Mac OS X) : le fichier est exécuté sans écrire le programme dans la console (sauvegarder le fichier en premier est obligatoire).
- Exécuter "... file with echo" (Ctrl L sous Windows et Linux, Cmd L sous Mac OS X) : réécrire le fichier dans la console et l'exécuter.
- Exécuter "...until the caret, with echo" (Ctrl E sous Windows et Linux, Cmd E sous Mac OS X) : réécrire la sélection choisie avec la souris dans la console et l'exécuter ou exécuter les données du fichier jusqu'au curseur position définie par l'utilisateur.

Copier / coller standard peut également être utilisé.

### 1.2.3 La fenêtre graphique

**Ouvrir une fenêtre graphique** Une fenêtre graphique s'ouvre automatiquement lors de la création de n'importe quel tracé. Il est possible de tracer des courbes, des surfaces, des séquences de points.

Pour obtenir un exemple de courbe, tapez dans la console :

```
1 --> plot
```

#### Remarque 2

- *Pour effacer un tracé, tapez **clf** ("clear figure").*
- *Pour ouvrir une autre fenêtre graphique, tapez **scf**; ("set current figure").*

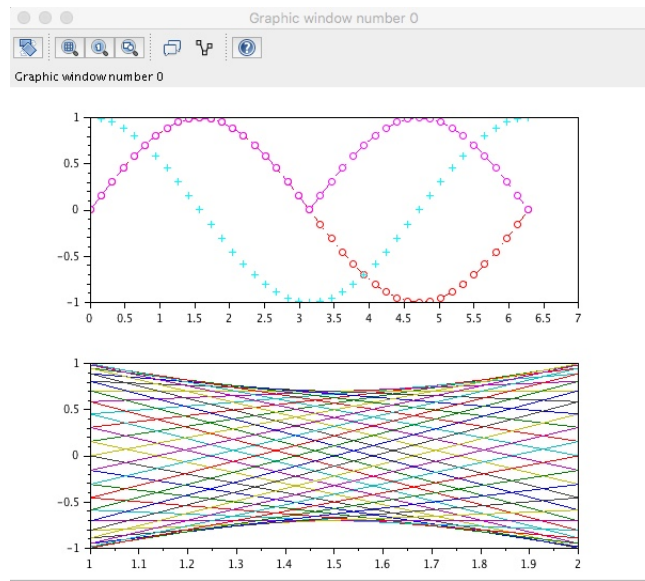





FIGURE 2 – La fenêtre graphique

— Si plusieurs fenêtres graphiques sont ouvertes, vous pouvez choisir dans lequel le tracé sera dessiné en tapant `scf (n)` ; `n` pour le numéro de fenêtre graphique (mentionné en haut à gauche).

**Modifier une fenêtre** La loupe  permet un zoom. Pour zoomer en deux dimensions, cliquez sur l’outil et avec la souris créez un rectangle qui constituera la nouvelle vue agrandie. Pour zoomer en trois dimensions, cliquez sur l’outil et créez un parallélépipède qui constituera la nouvelle vue agrandie. Il est également possible de zoomer en utilisant la molette de la souris. Pour revenir à l’écran initial, cliquez sur l’autre loupe . L’icône  permet la rotation de la figure (particulièrement utile en 3-D) avec des actions de clic droit guidées par des messages en bas de la fenêtre graphique.

Pour des modifications plus précises, cliquez sur **Edit > Figure properties** de la figure ou **Axes properties** et laissez-vous guider **Aide en ligne** Pour accéder à l’aide en ligne, cliquez sur **? > Scilab Help** dans la barre de menu, ou tapez dans la console :

```
1 --> help
```

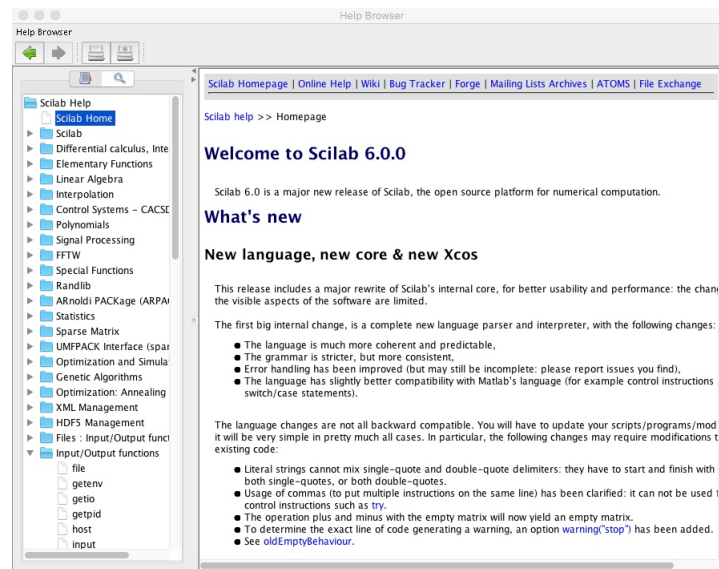


FIGURE 3 – La fenêtre d’aide

Pour obtenir de l'aide sur n'importe quelle fonction, tapez help dans la console suivi du nom de la fonction appropriée. Par exemple :

```
1 --> help sin
```

affiche l'aide pour la fonction sin (sinus).

## 2 Éléments de base du langage

Scilab calcule uniquement avec des nombres. Tous les calculs sont effectués avec des matrices, bien que cela puisse passer inaperçu. Même si le concept de matrices est inconnu, les vecteurs et les suites de nombres peuvent l'expliquer, car ce sont en fait des matrices de dimension  $1 \times n$  ou  $n \times 1$  et un nombre est lui-même une matrice de dimension  $1 \times 1$ . Les variables n'ont pas besoin d'être déclarées à l'avance, mais toute variable doit avoir une valeur. Par exemple, l'obtention de la valeur d'une variable qui n'a pas reçu de valeur génère une erreur :

```
1 --> f
2 Undefined variable: f
```

### 2.1 Expressions

Évaluation des données ou faire les opérations arithmétiques.

Exemple :

```
1 --> 2+6/2
2     ans =
3     5.
```

Opérateurs arithmétiques :

+ - *	addition, soustraction, multiplication	$2+3, 5-2, 2*3$
/	division à droite	$X/Y^{-1}$
\	division à gauche	$X^{-1}/Y$
^	puissance	$3\backslash 2$
modulo	c'est à dire le reste de n divisé par m	modulo(4,2)

### 2.2 Nom du variable

Les noms de variables peuvent être aussi longs que l'utilisateur le souhaite.

— Toutes les lettres ASCII de "a" à "z", de "A" à "Z" et les chiffres de "0" à "9" sont autorisés, avec les caractères supplémentaires %, \_, #, \$, ?.

Notez bien que les noms de variables, dont la première lettre est "%", ont une signification spéciale dans Scilab

— Scilab est sensible à la casse

```
1 -->A = 2
2     A=
3     2.
4 -->a = 1
5     a=
6     1.
7 -->A
8     A=
9     2.
10 -->a
11     a=
12     1.
```

## 2.3 Commentaires et lignes de suite

Toute ligne commençant par deux barres obliques "//" est considérée par Scilab comme un commentaire et est ignorée. Quand une instruction exécutable est trop longue pour être écrite sur une seule ligne, la deuxième et les suivantes lignes sont appelées lignes de continuation. Dans Scilab, toute ligne se terminant par deux points est considérée comme le début d'une nouvelle ligne de continuation.

```

1  -->// c'est un commentaire.
2  -->x=1..
3  - - >+2..
4  - - >+3..
5  -->+4
6      x=
7      10.

```

## 2.4 Fonctions mathématiques

```

exp   expm  log   log10  log1p  log2  logm  max
maxi  min   mini modulo pmodulo sign  signm sqrt
sqrtm

```

FIGURE 4 – fonctions mathématiques

La plupart de ces fonctions prennent un argument d'entrée et retournent un argument de sortie. Ces fonctions sont vectorisées dans le sens où leurs arguments d'entrée et de sortie sont des matrices. Cela permet de calculer des données avec des performances plus élevées, sans aucune boucle.

```

%i    the imaginary number i
%e    Euler's constant e
%pi   the mathematical constant π

```

FIGURE 5 – variables mathématiques pré-définies

Dans l'exemple suivant, nous utilisons les fonctions cos et sin et vérifions l'égalité  $\cos(x)^2 + \sin(x)^2 = 1$

```

1  -->x = cos (2)
2      x =
3      - 0.4161468
4  -->y = sin (2)
5      y =
6      0.9092974
7  -->x^2+y^2
8      ans =
9      1.

```

## 2.5 Types de données

### 2.5.1 Booléen

Les variables booléennes peuvent stocker des valeurs vraies ou fausses. Dans Scilab, true s'écrit avec %t ou %T et false avec %f ou %F.

```

1 --> etat=%t
2   etat =
3     T
4
5 --> etat=%f
6   etat =
7     F

```

### 2.5.2 Nombres complexes

Scilab fournit des nombres complexes, qui sont stockés sous forme de paires de nombres à virgule flottante. La variable prédéfinie %i représente le nombre imaginaire  $i$  qui satisfait  $i^2 = -1$ .

```

1 -->x= 1+%i
2       x=
3       1. + i
4 -->isreal(x)
5       ans =
6         F
7 -->y=1-%i
8       y=
9       1. - i
10 -->real(y)
11      ans =
12        1.
13 -->imag(y)
14      ans =
15        - 1.

```

### 2.5.3 Entiers

Nous pouvons créer différents types de variables entières avec Scilab. Il existe un lien direct entre le nombre de bits utilisés

int8	int16	int32
uint8	uint16	uint32

FIGURE 6 – Types de données entières

pour stocker un nombre entier et la plage de valeurs que l'entier peut gérer. La plage d'une variable entière dépend du nombre de ses bits.

- Un entier signé  $n$  bits prend ses valeurs de la plage  $[-2^{n-1}, 2^{n-1} - 1]$
- Un entier non signé  $n$  bits prend ses valeurs de la plage  $[0, 2^n - 1]$

### 2.5.4 Conversions entre entiers

Il existe des fonctions qui permettent de convertir des types de données entiers.

La fonction inttype permet de se renseigner sur le type d'une variable entière. Selon le type, la fonction renvoie une valeur correspondante

iconvert	conversion to integer representation
inttype	type of integers

FIGURE 7 – Conversions entre entiers

inttype(x)	Type
1	8-bit signed integer
2	16-bit signed integer
4	32-bit signed integer
11	8-bit unsigned integer
12	16-bit unsigned integer
14	32-bit unsigned integer

FIGURE 8 – Types d'entiers renvoyés par la fonction inttype.

```

1 --> a=int8(3)
2   a =
3     3
4
5 --> b=uint8(3)
6   b =
7     3
8
9 --> c=int32(3)
10  c =
11   3
12
13 --> d=uint32(3)
14  d =
15   3
16
17 --> inttype(a)
18  ans =
19   1.
20
21 --> inttype(b)
22  ans =
23  11.
24
25 --> inttype(c)
26  ans =
27   4.
28
29 --> inttype(d)
30  ans =
31  14.

```



### 2.5.5 Virgules flottantes

Par défaut les valeurs numériques dans Scilab sont codées sur des mots double précision (64 bits). Cela signifie que les nombres sont représentés avec une précision relative de l'ordre de  $10^{-16}$ . Donc, contrairement aux logiciels de calcul formel, les résultats obtenus par Scilab ne sont pas exacts ; toutefois, ils sont suffisamment précis pour la plupart des applications.

### 2.5.6 Chaines de caractères

Les chaînes peuvent être stockées dans des variables, à condition qu'elles soient délimitées par des guillemets " ". L'opération de coordination est disponible à partir de l'opérateur " + "

```

1  -->x = "foo"
2      x=
3      foo
4  -->y="bar"
5      y=
6      bar
7  -->x+y
8      ans =
9      foobar

```

## 2.6 Fonctions de lecture et ecriture

### 2.6.1 disp

la fonction disp permet l'affichage de variables de toute nature (nombres, matrices, texte) :  
 disp(x1,[x2,...xn])

```

1  --> disp('hello')
2  hello
3
4  --> a=3;disp(a)
5  3.
6
7  --> disp('salam',a)
8  3.
9  salam

```

### 2.6.2 input

La fonction input fonction provoque une interruption dans le programme courant. L'utilisateur est invité à entrer des caractères au clavier et à terminer avec <Enter>. Lorsque cette touche est enfoncée, l'exécution du programme se poursuit, et la fonction fournit en retour une valeur correspondant à ce que l'utilisateur a entré. Cette valeur peut alors être assignée à une variable quelconque, convertie, etc.

Pour lire une valeur numérique :

```

1  --> x=input("saisir_une_valeur")
2  saisir une valeur
3  4.5
4  x =
5  4.5
6
7  --> x
8  x =
9  4.5

```

pour lire une chaîne de caractères :

```
1 --> nom=input("Saisir_votre_Nom:_", "string")
2 Saisir votre Nom :
3 ESSADDOUKI
4   nom =
5
6   ESSADDOUKI
```