

Informatique en CPGE (2014-2015)
TP : introduction à Scilab

1 Introduction

SCILAB est un système interactif pour le calcul numérique, les calculs d'ingénierie, la programmation et la visualisation. Le logiciel SCILAB est un environnement complet. On peut résoudre des calculs complexes très rapidement. C'est ce que l'on peut appeler un environnement pour la résolution de problèmes. Ses avantages sont : une programmation facile, des algorithmes performants, de larges domaines d'application, le tracé de graphiques faciles.

Il a un autre avantage non négligeable sur d'autres logiciels du même type, c'est qu'il est libre et gratuit. On peut le télécharger sur <http://www.scilab.org>, ainsi qu'une nombreuse documentation gratuite. Scilab tourne sous UNIX, LINUX, Window et Mac OSX. C'est un logiciel écrit à l'origine par des chercheurs de l'INRIA (Institut National de Recherche en Informatique et Automatique).

Il existe des logiciels de calcul symbolique (ou calcul formel), Maple, Mathematica, Derive. Ils sont parfaits pour résoudre des équations par exemple, avec Derive :

$$\begin{aligned}
& \text{SOLVE}(x^3 - x^2 + x + 1, x) \\
& x = -\frac{(3\sqrt{33}-17)^{1/3}}{6} + \frac{(3\sqrt{33}+17)^{1/3}}{6} + \frac{1}{3} - i \cdot \left(\frac{(27\sqrt{11}-51\sqrt{3})^{1/3}}{6} + \frac{(27\sqrt{11}+51\sqrt{3})^{1/3}}{6} \right) \vee x = -\frac{(3\sqrt{33}-17)^{1/3}}{6} + \frac{(3\sqrt{33}+17)^{1/3}}{6} + \\
& \frac{1}{3} + i \cdot \left(\frac{(27\sqrt{11}-51\sqrt{3})^{1/3}}{6} + \frac{(27\sqrt{11}+51\sqrt{3})^{1/3}}{6} \right) \vee x = \frac{(3\sqrt{33}-17)^{1/3}}{3} - \frac{(3\sqrt{33}+17)^{1/3}}{3} + \frac{1}{3} \\
& \text{NSOLVE}(x^3 - x^2 + x + 1, x) \\
& x = 0.7718445063 - 1.115142508 \cdot i \vee x = 0.7718445063 + 1.115142508 \cdot i \vee x = -0.5436890126
\end{aligned}$$

Le problème, c'est que dès le degré 5, il n'y a pas de solution analytique générale, ce qui est encore plus courant pour de simples équations différentielles. Le calcul symbolique ne suffit pas et il est nécessaire de travailler avec des valeurs approchées. Avec Scilab, tous les calculs sont numériques et suivent la norme IEEE-754 (représentation des nombres dans la machine).

Rappel : Double précision = $2 \times 32 = 64$ bits.

Soit s = signe (1 bit), e = exposant (11 bits), m = mantisse (52 bits) :

nombre = $(-1)^s \times 2^{e-1023} \times 1.m$

nombre min : $2^{-1022} = 2,225 \dots 10^{-308}$

nombre max : $1,999 \dots \times 2^{1023} = 1,797 \dots 10^{308}$

précision : $2^{-52} = 2,22 \dots 10^{-16}$, soit 16 chiffres au maximum.

2 Premiers pas

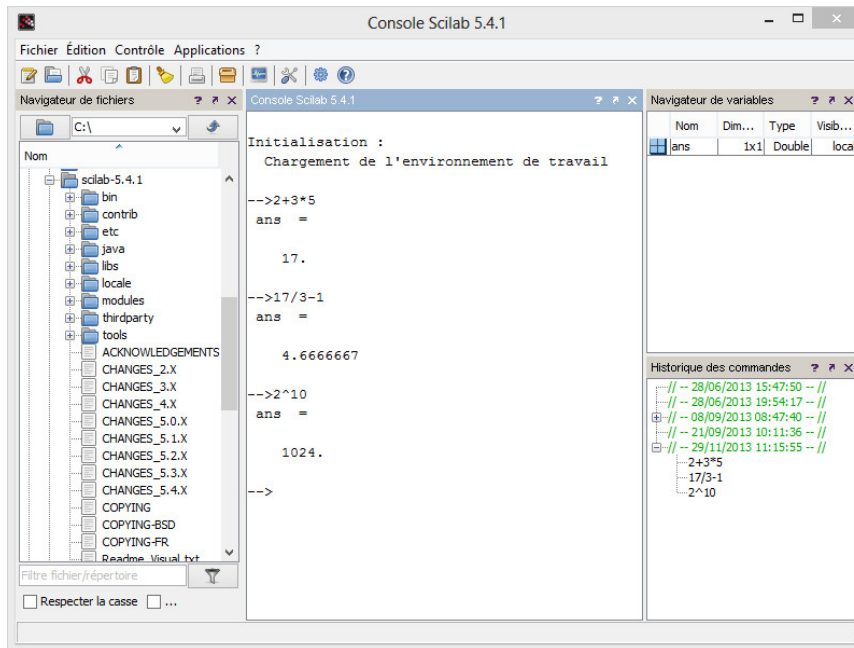
2.1 La console

Après l'invite de commande " -> ", il suffit de saisir une expression ou des instructions et d'appuyer sur la touche ↵. Il est possible de se déplacer à tout moment, avec les flèches du clavier ou avec la souris.

Tester les opérations habituelles, ainsi que les commandes et nombres particuliers qui suivent.

Par exemple :

ans, (= answer pour réponse), indique la réponse donné par Scilab.



```
-->sqrt(25)  
ans =  
  
5.  
  
-->%e  
%e =  
  
2.7182818284590450907956  
  
-->exp(1)  
ans =  
  
2.7182818284590450907956  
  
-->log(exp(1)) (Attention log est le logarithme népérien!)  
ans =  
  
1.  
  
-->%pi  
%pi =  
  
3.1415926535897931159980  
  
-->sqrt(-1)  
ans =  
  
i  
  
-->(1+2*i)^2  
ans =  
  
- 3. + 4.i  
  
-->sin(%pi)  
ans =  
  
1.225D-16  
  
10-16 équivaut ici à 0.  
  
-->%eps  
%eps =  
  
2.220D-16
```

c'est la précision minimale.

Calculs :

```
-->format (25)
-->1/10
ans =

    0.1000000000000000055511

-->1+0.1
ans =

    1.1000000000000000888178
```

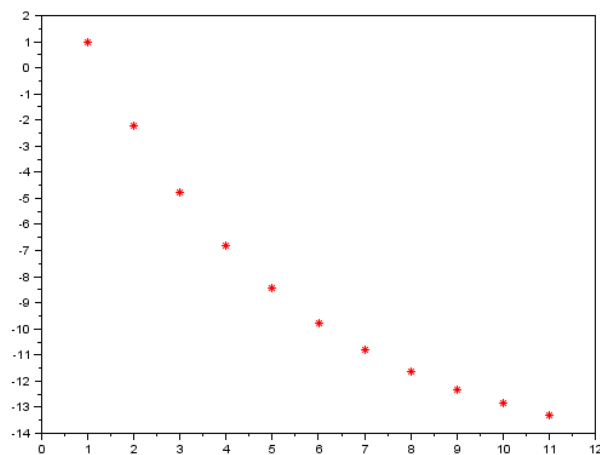
Les fonctions disponibles sont listées dans l'aide accessible en cliquant dans la barre de menus sur : ? > Aide de Scilab

2.2 L'éditeur

On peut ouvrir un éditeur à partir de la console : **Applications > SciNotes**. L'avantage est de pouvoir enregistrer (**Fichier > Enregistrer sous** avec l'extension ".sce " à la fin du nom de fichier) et enchaîner plusieurs instructions. L'éditeur se comporte comme n'importe quel logiciel de traitement de texte. L'indentation est automatique et il est possible d'ajouter des commentaires.

La définition d'une fonction commence par **function** et finit par **endfunction**. L'affectation se fait avec le signe " = ". Tester le programme suivant qui calcule les termes d'une suite avec ou sans le graphique :

```
//Calcul des termes d'une suite
u(1)=1;
//ou bien u(1)=input("valeur de u(1) ?");
function y=f(x);
    y=0.8*x-3;
endfunction
for n=1:10
    // sans fonction u(n+1)=0.8*u(n)-3;
    u(n+1)=f(u(n));
    disp([n,u(n)])
end
//plot(u,"*r") pour affichage graphique
```



3 Fenêtre graphique

Exemple du tracé d'une fonction : on écrit la commande **linspace** pour définir les valeurs de x , en écrivant : `x=linspace(a,b,n)` ;

a est la plus petite valeur de la variable x , b est la plus grande valeur de x , et n le nombre de valeurs qui seront calculées entre a et b .

Les tracés dans le plan se font avec la commande **plot**. Tester le programme qui suit.

```
function y=f(x)
    y=(x^2+2*x)*exp(-x)
endfunction
x=linspace(-2,5,50);
plot(x,f,"r")
function y=g(x)
    y=sin(x/2)
endfunction
x=linspace(-2,5,50);
clf //pour effacer la figure précédente
plot(x,g,"g")
//plot(x,f,"r",x,g,"g") pour avoir les 2 courbes
```

Les couleurs : "b" = bleu (par défaut), "k" = noir, "r" = rouge, "g" = vert, "c" = cyan, "m" = magenta, "y" = jaune, "w" = blanc

Les styles de points : relié (par défaut), ou ".", "+", "o", "x", "*"; d'autres options sont disponibles (à tester) avec : "s", "d", "v", "<", et ">".

La fonction **surf** permet de tracer une surface et **feval(x,y,f)** retourne la matrice $m \times n$ dont l'élément ij est $f(x_i, y_j)$ que l'on va transposer en utilisant l'apostrophe « ' » ; pour tracer la surface définie par une fonction du type $z = f(x, y)$:

```
function z=f(x,y)
    z=2*x^2+y^2;
endfunction
x=linspace(-1,1,100);
y=linspace(-2,2,200);
z=(feval(x,y,f))'; //calcul de z
clf
surf(x,y,z)
```

4 Structures de base pour la programmation

Test :

```
x=input("valeur de x ?");
if x>10 then
    y=x;
else
    y=x^2
end
disp y=
disp ([y])
disp ("y = " + string(y))
```

Boucle :

```
for i=1:10
    x(i)=i^2;
end
h=1.2;
while h<7
    h=h+0.3;
end
```

5 Exercices

Anniversaire : en ne tenant pas compte des années de naissance, la probabilité pour que, dans un groupe de k personnes, deux au moins aient la même date d'anniversaire est beaucoup plus élevée que ce que l'on pourrait croire.

```
function proba = anniversaire1(k)
    jours = 365;
    p = 1;
    for n = 1:k-1
        p = p * (jours - n)/jours;
    end
    proba = 1 - p;
endfunction

disp(anniversaire1(23))
```

Vérifier que la réponse est bien 0.5072972.

Tracer la courbe de la probabilité de coïncidence en fonction du nombre d'individus de 1 à 365 avec la commande **plot(feval(1 :365, anniversaire1))**.

Visualiser à partir de quel nombre d'individus elle dépasse 50%, 75% ... (il faudra zoomer)

Autre manière à tester :

```
function proba2 = anniversaire2(k)
    jours = 365;
    p = prod(1:-1/jours:1-(k-1)/jours); //prod(début : pas : fin)
    //ou bien p= prod(jours-k+1:jours)/jours^k;
    // p est la probabilité pour qu'aucune date ne coïncide
    proba2 = 1 - p;
endfunction

disp(anniversaire2(23))
```

Vérifier que la réponse est bien 0.5072972 puis tracer la courbe de la probabilité de coïncidence en fonction du nombre d'individus de 1 à 365 avec la commande **plot(feval(1 :365, anniversaire2))**.

Remarque : on peut éviter l'utilisation de la fonction **feval** : remplacer **prod** par **cumprod** dans anniversaire2 (sauvegarder et recharger !) puis écrire après l'invite de commande :
-> clf; plot(anniversaire2(70))

Compléments : cliquer sur ? en haut dans la barre des menus puis sur **Aide de Scilab** ou sur **Démonstrations** pour voir ce qu'il est possible de faire avec Scilab.