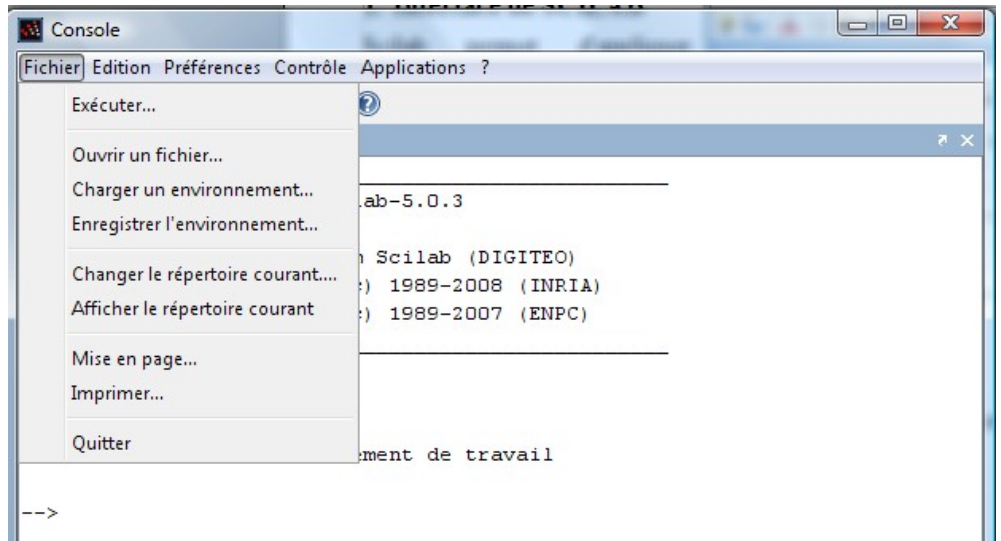


Programmes sous SCILAB

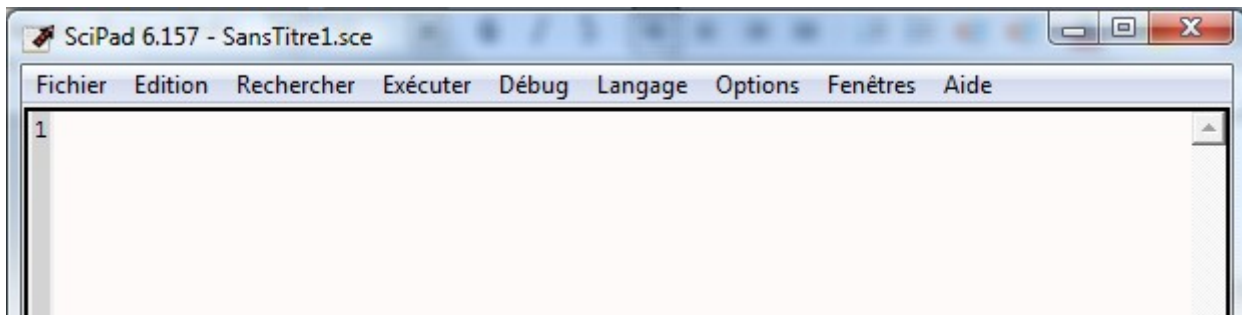
1. Interface de SCILAB

Il existe deux moyens d'utiliser un programme (ou une fonction) :

Scilab permet d'appliquer directement les instructions qui sont rentrées les unes après les autres après le prompt `-->`. L'utilisateur remarquera la similitude de fonctionnement avec une calculatrice.



L'utilisateur peut aussi démarrer un éditeur que l'on peut trouver dans l'onglet Applications (le Bloc Note de Windows ou tout autre éditeur simple peut également convenir). Une fois le programme ou la suite de fonctions enregistrés dans un fichier (normalement au format `.sce` ou `.sci` mais Scilab peut ouvrir les fichiers `.txt`), il faut aller dans l'onglet Fichier de Scilab pour l'exécuter ou bien dans l'onglet Exécuter de l'éditeur.



Dans le cas de l'utilisation de l'éditeur Scipad, les instructions sont soit séparées d'une virgule (et dans ce cas, l'affichage du contenu va s'effectuer), soit elles sont séparées d'un point virgule et dans ce cas, les affichages successifs des calculs ne sont pas permis.

Fonctions et programmes peuvent se côtoyer dans un même fichier. Il suffit de présenter une fonction en encadrant son contenu par la syntaxe :

```
function y=nom(x)          (function [y1,y2]=nom(x1,x2) dans le cas de plusieurs paramètres
...                        en entrée et/ou en sortie)
...
endfunction
```

Une remarque importante dans l'utilisation de Scilab, la flèche vers le haut du clavier (\uparrow) permet de rappeler les lignes précédentes sans qu'il soit utile de les retaper, comme par exemple l'exécution du fichier de programme sauvegardé depuis l'éditeur.

2. Ecrire et tester un programme simple

Reprenons le programme de calcul de valeurs d'une fonction par balayage :

Algobox	Comparatif	SCILAB
<p>Code de l'algorithme</p> <pre> VARIABLES ├── x EST_DU_TYPE NOMBRE ├── y EST_DU_TYPE NOMBRE └── DEBUT_ALGORITHME ├── x PREND_LA_VALEUR 1 ├── TANT_QUE (x<=2) FAIRE │ ├── DEBUT_TANT_QUE │ ├── y PREND_LA_VALEUR x-1-1/x │ ├── AFFICHER x │ ├── AFFICHER "-> " │ └── AFFICHER y ├── x PREND_LA_VALEUR x+0.1 └── FIN_TANT_QUE FIN_ALGORITHME </pre>	<p>x reçoit la valeur 1. L'affectation est désignée par = alors que le test d'égalité est désigné par ==</p> <p>On affiche le résultat par disp (pour display).</p> <p>Disp ne permettant pas un affichage sur une ligne, on transforme ici les valeurs des variables en chaînes de caractères grâce à l'instruction string.</p> <p>La concaténation entre chaînes est assurée par +.</p> <p>Le programme ci-contre a été écrit dans l'éditeur de Scilab ; soit on utilise le menu Exécuter de l'éditeur, soit on l'enregistre dans un fichier pour l'exécuter dans Scilab.</p>	<pre> Fichier Edition Rechercher Exécuter Débug La 1 x=1; 2 while x<=2 do 3 y=x-1-1/x; 4 //l'instruction disp(x,"-->",y) ne 5 //permet pas d'afficher le tout sur 6 //une ligne 7 disp(string(x)+"-->"+string(y)) 8 //on transforme le tout en caractères 9 x=x+0.1; 10 end 11 --> 1.1-->-0.8090909 1.2-->-0.6333333 1.3-->-0.4692308 1.4-->-0.3142857 1.5-->-0.1666667 1.6-->-0.025 1.7-->0.1117647 1.8-->0.2444444 1.9-->0.3736842 --> </pre>

3. Programmes et paramètres

Le programme ci-dessus peut s'améliorer par l'utilisation de paramètres pour constituer une fonction :

- les bornes a et b de l'intervalle de définition de la fonction;
- le pas h du balayage;

L'expression de la fonction est elle-même définie dans une autre fonction.

<pre> Fichier Edition Rechercher Exécuter Débug 1 function [y]=f(x) 2 y=x-1-1/x; 3 endfunction 4 5 function balayage(a,b,h) 6 x=a; 7 while x<=b do 8 y=f(x) 9 disp(string(x)+"-->"+string(y)) 10 x=x+h; 11 end 12 endfunction 13 14 balayage(3,10,1) 15 </pre>	<p>Scilab a exécuté le programme de gauche et l'a réalisé sur la capture d'écran ci-contre, à droite.</p> <p>Une fois le programme réalisé, le prompt attend d'autres instructions.</p>	<pre> 3-->1.6666667 4-->2.75 5-->3.8 6-->4.8333333 7-->5.8571429 8-->6.875 9-->7.8888889 10-->8.9 --> </pre>
---	---	--

4. Autres algorithmes

a) Deviner un nombre

Algobox	SCILAB
<p>de de l'algorithme</p> <ul style="list-style-type: none"> ▼ VARIABLES <ul style="list-style-type: none"> n EST_DU_TYPE NOMBRE m EST_DU_TYPE NOMBRE ▼ DEBUT_ALGORITHME <ul style="list-style-type: none"> n PREND_LA_VALEUR floor(random()*1000)+1 LIRE m ▼ TANT_QUE (m!=n) FAIRE <ul style="list-style-type: none"> DEBUT_TANT_QUE ▼ SI (m>n) ALORS <ul style="list-style-type: none"> DEBUT_SI AFFICHER m AFFICHER " est trop grand" FIN_SI ▼ SI (m<n) ALORS <ul style="list-style-type: none"> DEBUT_SI AFFICHER m AFFICHER " est trop petit" FIN_SI LIRE m FIN_TANT_QUE AFFICHER "Bravo !" ▼ FIN_ALGORITHME 	<p>SciPad 6.157 - devinette</p> <p>Fichier Edition Rechercher Exécuter Débug Langage</p> <pre> 1 n=floor(rand()*100)+1; 2 m=0; 3 while m<>n do 4 m=input('Entrez votre valeur'); 5 if m>n then disp("trop grand") 6 elseif m<n then disp("trop petit") 7 else disp("gagné") 8 end 9 end 10 </pre> <p>Entrez votre valeur: 12</p> <p>trop petit</p> <p>Entrez votre valeur: 50</p> <p>trop grand</p> <p>Entrez votre valeur:</p>

b) Maximum de deux nombres

Algobox	SCILAB
<p>Code de l'algorithme</p> <ul style="list-style-type: none"> ▼ VARIABLES <ul style="list-style-type: none"> a EST_DU_TYPE NOMBRE b EST_DU_TYPE NOMBRE m EST_DU_TYPE NOMBRE ▼ DEBUT_ALGORITHME <ul style="list-style-type: none"> LIRE a LIRE b ▼ SI (a<b) ALORS <ul style="list-style-type: none"> DEBUT_SI m PREND_LA_VALEUR b FIN_SI ▼ SINON <ul style="list-style-type: none"> DEBUT_SINON m PREND_LA_VALEUR a FIN_SINON AFFICHER m ▼ FIN_ALGORITHME 	<p>Fichier Edition Rechercher Exécuter</p> <pre> 1 function c=maximum(a,b) 2 if a<b then c=b 3 else c=a 4 end 5 endfunction 6 </pre> <p>-->maximum(1,12)</p> <p>ans =</p> <p>12.</p> <p>--> </p> <p>On peut remarquer dans cette exécution de fonction le même principe que sur une calculatrice, c'est-à-dire que le résultat est attribué à une variable ans (pour answer) si l'utilisateur n'affecte pas le résultat de la fonction à une variable.</p>

c) Dessin d'une rosace sous SCILAB

```
Fichier  Edition  Rechercher  Exécuter
1 function rosace(n)
2   square(-4,-4,4,4);
3   for i=1:n do
4     x=cos(2*i*pi/n)
5     y=sin(2*i*pi/n)
6     xarc(x-2,y+2,4,4,0,360*64)
7   end
8 endfunction
9
```

L'instruction square permet, comme pour une calculatrice, de créer un repère orthogonal dans lequel on désigne successivement xmin, ymin, xmax, ymax.

Les constantes sous Scilab (π , i , e , ...) sont désignées par l'utilisation de %.

l'instruction xarc définit un arc d'ellipse (ici un cercle) dans la fenêtre, inscrit par un rectangle défini par le point en haut à gauche, puis sa largeur et sa hauteur. De plus, cette instruction fonctionne par 1/64 de degré, donc un tour complet pour l'arc revient à le tracer pour un paramètre entre 0 et 360*64.

L'exécution dans Scilab de cette fonction par l'instruction rosace(30), permet d'obtenir une fenêtre graphique dans laquelle on peut voir :

