ALGORITHMIQUE
&
MATHEMATIQUES

Travaux pratiques et Applications Scilab
pour le lycée et la licence

José OUIN
Ingénieur INSA Toulouse
Professeur agrégé de Génie civil
Professeur agrégé de Mathématiques

ellipses

ISBN : 978-2-7298-5439-3

www.editions-ellipses.fr

Avant-Propos

Depuis la rentrée 2009, l'algorithmique fait partie des programmes de mathématiques du lycée.
Cet ouvrage répond aux attentes de tous ceux qui découvrent l'algorithmique ou qui sont à la recherche d'un ensemble de ressources "prêtes à l'emploi".

A partir de définitions concises et d'exemples concrets, cet ouvrage a pour objectif d'accompagner les enseignants, les étudiants et les élèves et de leur permettre ainsi :

- d'écrire des algorithmes et des programmes ;
- de découvrir l'outil de calcul scientifique Scilab ;
- de conjecturer des évolutions ou des solutions à des problèmes concrets.

Les travaux pratiques sont destinés à être traités à l'aide d'algorithmes, aussi il n'est pas toujours nécessaire que le cours théorique correspondant ait été abordé auparavant en classe. Certains d'entre eux peuvent même constituer des activités préparatoires au cours magistral.
Ainsi chaque activité peut être effectuée à tout moment de l'année scolaire et ce, pour les trois niveaux du lycée.

J'espère que cet ouvrage aidera les professeurs dans leurs missions d'enseignement et les étudiants dans leur apprentissage des outils de calculs scientifiques.
Pour moi, l'algorithmique est un moyen de rendre les mathématiques encore plus attrayantes, grâce aux possibilités des logiciels et à la puissance de calcul des ordinateurs qui permettent d'émettre des conjectures ou d'encadrer les valeurs numériques de solutions suite à un grand nombre d'itérations.

Je dédie ce livre à tous les amoureux des mathématiques appliquées, à tous ceux qui aiment calculer, modéliser ou conjecturer, à tous ceux qui souhaitent découvrir un langage de programmation et à tous ceux qui découvriront, grâce aux algorithmes, le plaisir de faire des mathématiques.

José OUIN.

Les sites Internet dont je suis l'auteur :

- http://www.joseouin.net : Site Internet comportant les logiciels que j'ai développés ainsi que les programmes Scilab de cet ouvrage.
- http://www.refleteducation.fr : Site Internet destiné à promouvoir l'intégration de logiciels libres pour l'éducation (gestionnaire de contenus, plate-forme de formation, didacticiels vidéos, etc.).

SOMMAIRE

L'algorithmique

Première partie – Enoncés des travaux pratiques

A – Fonctions et résolution d'équations

B – Probabilités

C – Suites numériques

D – Arithmétique

E – Géométrie

Deuxième partie – Solutions des travaux pratiques

A – Fonctions et résolution d'équations

B – Probabilités

C – Suites numériques

D – Arithmétique

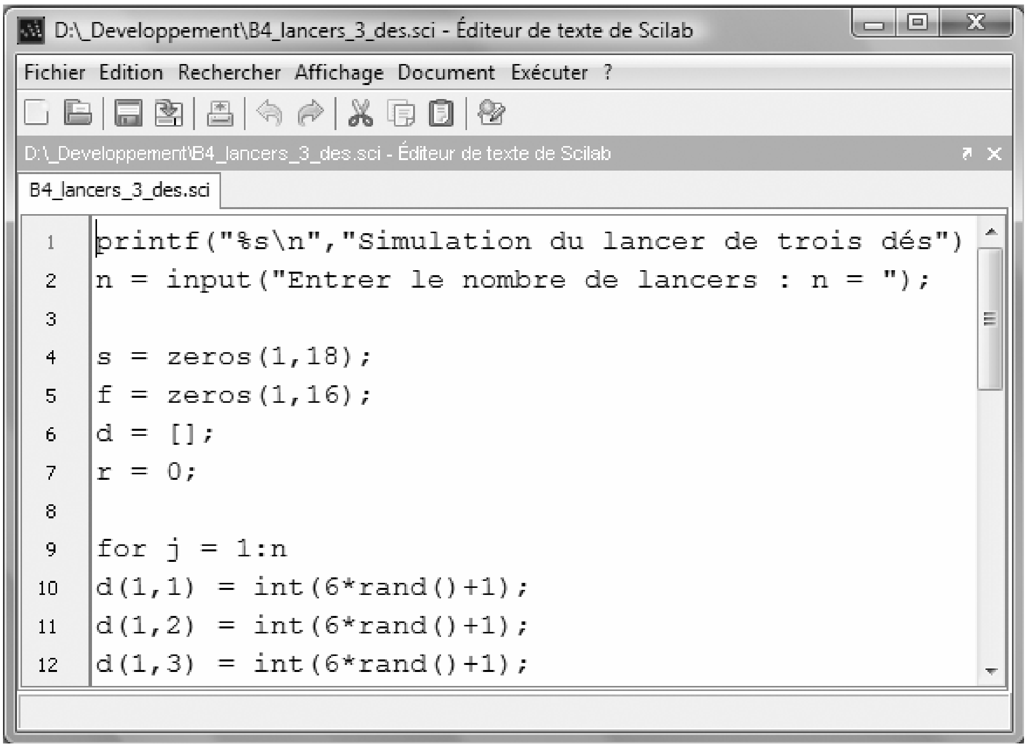
E – Géométrie

Troisième partie – Le logiciel Scilab en 10 étapes

➔ L'éditeur

L'éditeur permet à l'utilisateur de saisir les lignes de codes d'un programme ou de définir des fonctions.

Remarque : On peut saisir directement des instructions dans la console Scilab, mais il est plus simple de grouper ces commandes dans l'éditeur qui seront exécutées ligne par ligne par le logiciel Scilab. Ces commandes peuvent être sauvegardées dans un fichier texte (extension .sci ou .sce).



Les lignes de codes écrites dans l'éditeur Scilab sont présentées ainsi :

```
Editeur :
x = linspace(a,b,p);
y= ones(x)*d;

// Surface
scf();
xset('colormap',hotcolormap(128))
fplot3d1(x,x,f)
```

4- Les instructions du langage Scilab

4-1. Lecture et écriture des données

Cette partie comporte la liste des fonctions Scilab utilisées dans cet ouvrage. Chaque fonction comporte une définition succincte mais suffisante pour être utilisée dans les travaux pratiques. Pour plus d'informations sur les différents paramètres de ces fonctions, prière de se reporter au fichier d'aide du logiciel Scilab.

4-1.1 INPUT

La fonction input() permet de saisir une variable utilisateur.

```
Editeur :
w = input("Entrer votre prénom : ","string");
u = input("Entrer les bornes : [a,b]= ");
p = input("Entrer le nombre de points : p = ");
```

```
Console :
Entrer votre prénom : Pierre
Entrer les bornes : [a,b]= [5,2]
Entrer le nombre de points : p = 7
```

Remarques :
La variable w contient une chaîne de caractère (string)
La variable u est un vecteur : u(1) = 5 et u(2) = 2 . u(1) peut s'écrire u(1,1) et u(2), u(1,2).
La variable p est un réel. Ici p = 7.

4-1.2 DISP

La commande disp() permet d'afficher un résultat dans la console Scilab.

```
Editeur :
a=5;
disp(a)
```

```
Console :
5.
```

Première partie

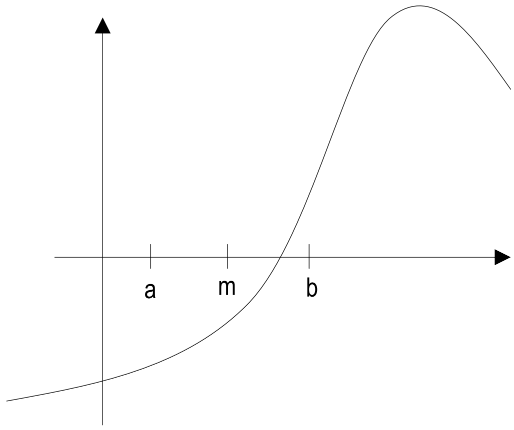
Enoncés des travaux pratiques

2- Travail pratique
Résolution de l'équation f(x) = 0 – Méthode de dichotomie

Soit f une fonction strictement monotone sur un intervalle [a ; b] où les valeurs de a et de b sont telles que f change de signe entre a et b.
Il s'agit de résoudre l'équation (E) sur cet intervalle :

$$f(x) = 0 \quad \text{(E)}$$

De nombreuses situations donnent lieu à la résolution numérique d'équations et d'inéquations. Les calculatrices et les outils logiciels intègrent les fonctionnalités numériques ou formelles permettant cette résolution.
On se propose ici d'utiliser une des méthodes de résolution appelée méthode de dichotomie.



Présentation de la méthode de dichotomie
Cette méthode consiste, en choisissant à chaque fois la valeur située au milieu de l'intervalle en cours, à réduire de moitié l'amplitude de l'intervalle dans lequel se trouve le nombre α tel que $f(\alpha) = 0$.

Au bout de n essais, l'intervalle a pour amplitude : $\frac{b-a}{2^n}$

2-1. Enoncé

On considère la fonction suivante : $f(x) = -x^2 + 10x - 23$. Il s'agit de résoudre l'équation (E) :

$$f(x) = 0 \quad \text{(E)}$$

On souhaite déterminer la solution x_1 appartenant à l'intervalle [2 ; 4] avec une précision p donnée, ainsi que la solution x_2 appartenant à l'intervalle [5 ; 7] avec la même précision.

2-2. Travail demandé

1] Ecrire l'algorithme permettant de calculer les solutions de l'équation (E).
2] Ecrire le programme correspondant permettant de saisir la précision p désirée.

2-3. Algorithmique

2-3.1 Les données

- f : La fonction à étudier.
- p : La précision désirée.

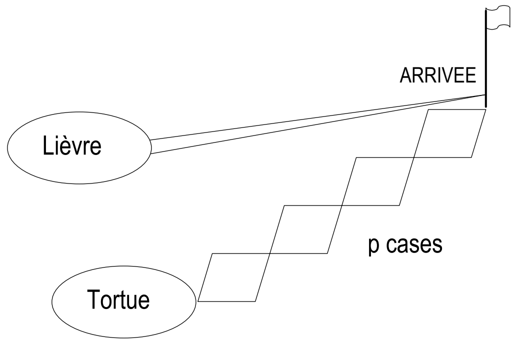
2-3.2 Les valeurs à déterminer

Les solutions de l'équation (E).

1- Travail pratique

Le jeu du lièvre et de la tortue

1-1. Enoncé



Un lièvre et une tortue sont sur la ligne de départ. La tortue doit avancer de p cases pour atteindre la ligne d'arrivée.
Règle du jeu :
Une partie comporte au plus p tours. À chaque tour, on lance un dé. Si le 6 sort, alors le lièvre gagne la partie, sinon la tortue avance d'une case.
La tortue gagne quand elle a avancé p fois.

Soit T l'évènement : "La tortue a gagné la partie", il s'agit de déterminer, à l'aide d'une simulation de ce jeu, la fréquence de l'évènement T afin de pouvoir conjecturer, suivant les valeurs de p, si le jeu est à l'avantage du lièvre ou de la tortue.

1-2. Travail demandé

1] Ecrire l'algorithme permettant d'effectuer la simulation de *n* parties et de compter le nombre de parties *t* gagnées par la tortue. En déduire la fréquence $\frac{t}{n}$ de l'évènement T.
2] Ecrire le programme correspondant permettant de saisir le nombre de parties *n* ainsi que le nombre de cases p à parcourir par la tortue.
3] Conjecturer, suivant les valeurs de p, si le jeu est à l'avantage du lièvre ou de la tortue.

1-3. Algorithmique

1-3.1 Les données

- *n* : Le nombre de parties à jouer.
- p : Le nombre de cases à parcourir par la tortue.

1-3.2 Les valeurs à déterminer

- *t* : Le nombre de parties gagnées par la tortue.
- $\frac{t}{n}$: La fréquence de l'évènement T.

1-3.3 La méthode utilisée

- La simulation de chaque lancer de dé est obtenue par la génération d'un nombre entier aléatoire compris entre 1 et 6.
- Pour chaque partie, on lance le dé au maximum p fois. Par exemple, si le 6 sort au second lancer alors la tortue a perdu et on recommence alors une nouvelle partie.
- Pour chaque partie gagnée par la tortue, on incrémente la valeur de *t*.
- Au bout de *n* parties on calcule le quotient $\frac{t}{n}$ correspondant à la fréquence de l'évènement T.

7- Travail pratique

Nombres à moyenne harmonique entière

7-1. Enoncé

Un nombre à moyenne harmonique entière est un entier naturel p dont la moyenne harmonique m de ses n diviseurs positifs est un entier.

Si on note $d_1, d_2, ..., d_n$ les n diviseurs positifs de l'entier naturel p, alors le nombre m suivant est un entier :

$$m = \frac{n}{\frac{1}{d_1} + \frac{1}{d_2} + ... + \frac{1}{d_n}} = \frac{n}{\sum_{k=1}^{k=n} \frac{1}{d_k}}$$

Exemple :

On considère le nombre p = 6. Les 4 diviseurs positifs de 6 sont {1, 2, 3, 6}.

$m = \frac{4}{\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{6}} = \frac{4}{2} = 2$; m est un entier donc 6 est un nombre à moyenne harmonique entière.

7-2. Travail demandé

1] Ecrire l'algorithme permettant de déterminer si un entier naturel p est à moyenne harmonique entière.
2] Ecrire le programme correspondant permettant de saisir la valeur de p.
3] Vérifier que les nombres suivants sont à moyenne harmonique entière : 6 ; 1638 ; 6200 ; 8128 ; 8190.
4] On définit les entiers : $q_n = 2^{n+1} - 1$ et $p_n = 2^n . q_n$, où n est un entier naturel. Donner quelques valeurs de n pour lesquelles l'entier p_n est un entier à moyenne harmonique entière. Quelle conjecture peut-on faire ?

7-3. Algorithmique

7-3.1 Les données

p : L'entier naturel à étudier.

7-3.2 Les valeurs à déterminer

Il s'agit de déterminer la moyenne harmonique m de l'entier naturel p.

7-3.3 La méthode utilisée

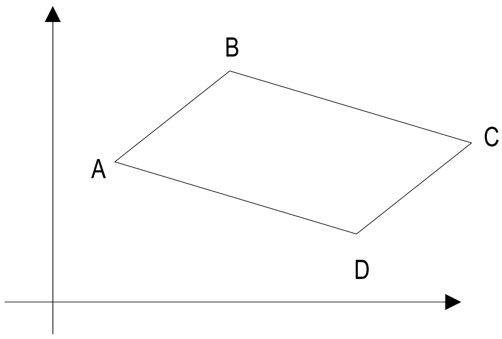
- On saisit la valeur de p.
- On détermine l'ensemble des diviseurs positifs de p, puis on calcule la valeur de m.
- On affiche le résultat.

7-3.4 Les fonctions et structures

➔ Structure répétitive
TantQue p <> 1 **Faire**
{Traitement 1}
FinTantQue

2- Travail pratique

Etude d'un parallélogramme



2-1. Enoncé

A partir des coordonnées de trois points A, B et C, on souhaite déterminer les coordonnées du point D tel que ABCD soit un parallélogramme.

2-2. Travail demandé

1] Ecrire l'algorithme permettant de déterminer les coordonnées du point D.
2] Ecrire le programme correspondant permettant de saisir les coordonnées des points A, B et C.
3] Représenter le parallélogramme ABCD dans une fenêtre graphique.

2-3. Algorithmique

2-3.1 Les données

Les coordonnées des points A, B et C.

2-3.2 Les valeurs à déterminer

Les coordonnées du point D.

2-3.3 La méthode utilisée

- On saisit les coordonnées des points A, B et C.
- On calcule les coordonnées du point I, milieu de [AC].
- On calcule les coordonnées du point D, symétrique du point B par rapport au point I.
- On affiche le résultat.

2-3.4 Les fonctions et structures

➔ vecteur a = [5,8]
a est un vecteur qui représente les coordonnées du point A(5 ; 8).
a(1) ou a(1,1) a pour valeur 5. a(2) ou a(1,2) a pour valeur 8.

n = a/2 ; n est un vecteur et n = [2.5,4]
Si b = [2,10] alors d = a + b est un vecteur et d = [7,18]

➔ Fonction gca()
a=gca() retourne l'identifiant de l'axe courant.
➔ Fonction isoview
a.isoview() = "on" : Permet d'obtenir un repère isométrique (avec les mêmes échelles).

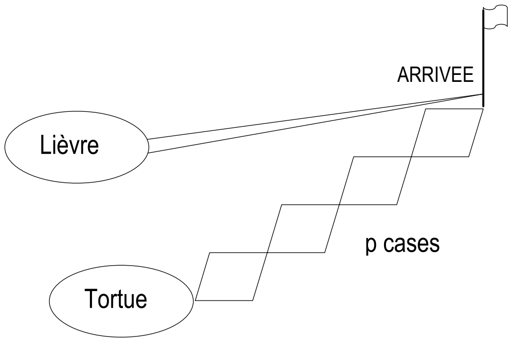
➔ Fonction plot() : Trace le graphe d'une fonction.

Deuxième partie

Solutions des travaux pratiques

1- Travail pratique – Solution
Le jeu du lièvre et de la tortue

1-1. L'algorithme



Variables
```
n , p , c , t , d , k
```

Entrées
```
Saisir n , p
```

Traitement
```
c prend la valeur 0 (Numéro de la case où se trouve la tortue)
t prend la valeur 0 (Nombre de parties gagnées par la tortue)
k prend la valeur n (On utilise la variable k pour le comptage des
parties)

TantQue k <> 0 Faire

d prend une valeur entière aléatoire comprise entre 1 et 6.

Si d <> 6 Alors
c prend la valeur c + 1
       Si c = p Alors
       t prend la valeur t + 1
       c prend la valeur 0
       k prend la valeur k - 1
       FinSi
Sinon
c prend la valeur 0
k prend la valeur k - 1
FinSi

FinTantQue
```

Sorties
```
Afficher la fréquence : t/n
Afficher la fréquence théorique : (5/6)^p
```

1-2. Le programme

```
n = input("Entrer le nombre n de parties à jouer .........      : ")
p = input("Entrer le nombre p de cases à parcourir par la tortue : ")


k = n;
t = 0;
c = 0;

while k <> 0
d = int(6 * rand() + 1);
  if d <> 6 then
  c = c + 1;
        if c == p then
        t = t + 1;
        c = 0;
        k = k - 1;
        end
  else
  c = 0;
  k = k - 1;
  end
end



printf("%s\n","RESULTATS :")
printf ("Fréquence des parties gagnées par la tortue : %f\n",t/n)
printf ("Fréquence théorique                         : %f\n",(5/6)^p)
```

1-3. Les résultats numériques

Les résultats ci-dessous correspondent à des simulations de 10 000 parties pour différentes valeurs du nombre de cases p.

Console Scilab :

```
Entrer le nombre n de parties à jouer ..........     : 10000
Entrer le nombre p de cases à parcourir par la tortue :     1
RESULTATS :
Fréquence des parties gagnées par la tortue : 0.838300
Fréquence théorique                          : 0.833333

Entrer le nombre n de parties à jouer ..........     : 10000
Entrer le nombre p de cases à parcourir par la tortue :     2
RESULTATS :
Fréquence des parties gagnées par la tortue : 0.696900
Fréquence théorique                          : 0.694444

Entrer le nombre n de parties à jouer ..........     : 10000
Entrer le nombre p de cases à parcourir par la tortue :     3
RESULTATS :
Fréquence des parties gagnées par la tortue : 0.573300
Fréquence théorique                          : 0.578704

Entrer le nombre n de parties à jouer ..........     : 10000
Entrer le nombre p de cases à parcourir par la tortue :     4
RESULTATS :
Fréquence des parties gagnées par la tortue : 0.487200
Fréquence théorique                          : 0.482253

Entrer le nombre n de parties à jouer ..........     : 10000
Entrer le nombre p de cases à parcourir par la tortue :     5
RESULTATS :
Fréquence des parties gagnées par la tortue : 0.402900
Fréquence théorique                          : 0.401878

Entrer le nombre n de parties à jouer ..........     : 10000
Entrer le nombre p de cases à parcourir par la tortue :     6
RESULTATS :
Fréquence des parties gagnées par la tortue : 0.332200
Fréquence théorique                          : 0.334898

Entrer le nombre n de parties à jouer ..........     : 10000
Entrer le nombre p de cases à parcourir par la tortue :     7
RESULTATS :
Fréquence des parties gagnées par la tortue : 0.288900
Fréquence théorique                          : 0.279082
```

Conjecture : Le jeu est à l'avantage de la tortue lorsque le nombre de cases est inférieur ou égal à 3.

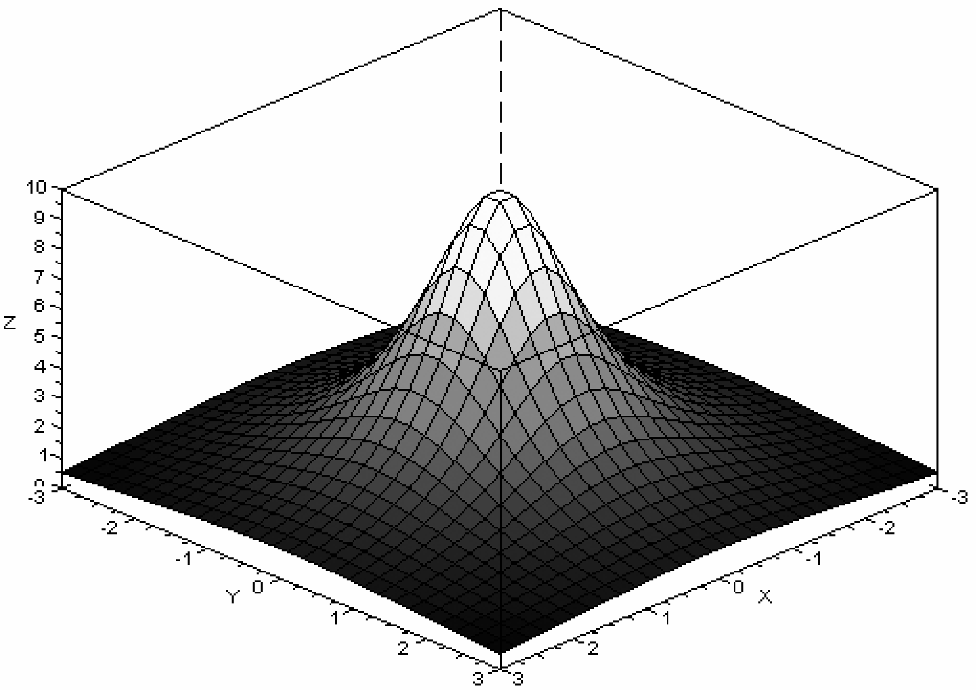
4-3. Les résultats numériques

3] Représentation graphique des surfaces :

$$(S_1) : \ z = \frac{10}{x^2 + y^2 + 1}$$

Représentation graphique d'une surface d'équation z = f(x,y)

Entrer l'expression de la fonction f(x,y). z= 10/(x^2+y^2+1)
Entrer les bornes de l'intervalle. [a,b]= [-3,3]
Entrer le nombre de points à considérer. p= 30



Remarque : Le bouton "Rotation"  de la barre d'outils de la fenêtre graphique permet de visualiser la surface sous différents angles de vue.

Troisième partie

Le logiciel Scilab en 10 étapes

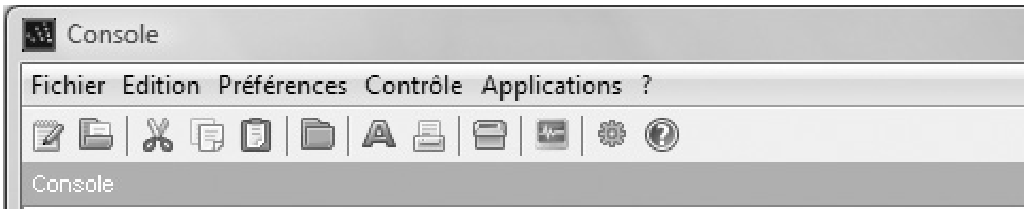
Cette troisième partie est destinée à permettre une prise en main rapide du logiciel Scilab. Il est entendu que toutes les fonctionnalités du logiciel n'y sont pas abordées. Il s'agit simplement d'acquérir une connaissance de base pour pouvoir traiter les travaux pratiques dans de bonnes conditions. Chaque étape propose des exercices à réaliser ainsi que les solutions correspondantes.

1- L'environnement Scilab

Le logiciel Scilab se compose principalement d'une console, d'un éditeur et de fenêtres graphiques.

1-1. La console

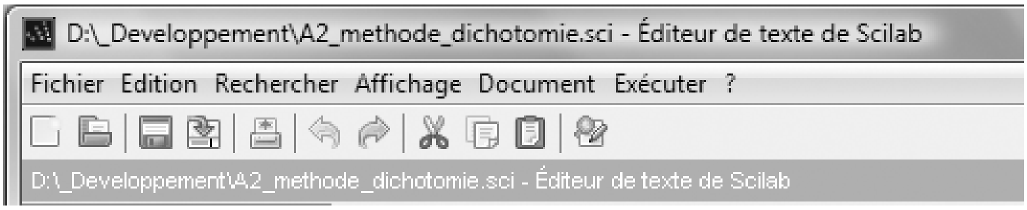
Il s'agit de la fenêtre principale de Scilab. La console permet à l'utilisateur de saisir toutes les commandes Scilab directement au clavier.



➔ Pour accéder à l'éditeur : Menu de la console : **"Applications / Editeur"**

1-2. L'éditeur

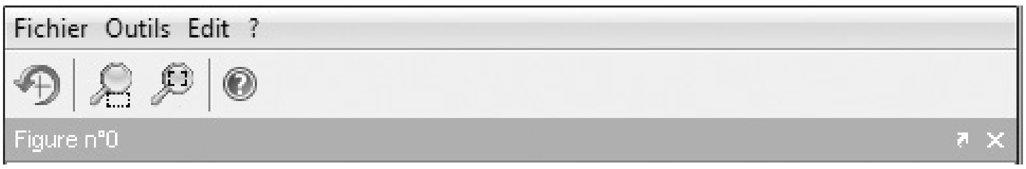
L'éditeur permet à l'utilisateur de saisir les lignes de codes d'un programme ou de définir des fonctions.



➔ Pour exécuter un programme : Menu de l'éditeur : **"Exécuter / Charger dans Scilab"** ou **"Exécuter / Exécuter le fichier dans Scilab"**.

1-3. Les fenêtres graphiques

Les fenêtres graphiques affichent les représentations graphiques créées par les programmes ou les fonctions définies par l'utilisateur. Il est possible de définir plusieurs fenêtres graphiques pour un même programme. La barre d'outils permet d'effectuer une rotation du graphique, des zooms avant ou arrière et d'accéder au fichier d'aide du logiciel.



2- Utiliser la console et l'éditeur

Exercice 1
Effectuer les opérations suivantes dans la console Scilab :
- 22/7
- 4^2 + 5
- (5+4^2)/(7+4^3)

Solution 1

```
Console :
-->22/7
 ans  =
    3.1428571

-->4^2+5
 ans  =
    21.

-->(5+4^2)/(7+4^3)
 ans  =
     0.2957746
```

Exercice 2
Effectuer les opérations suivantes dans l'éditeur Scilab et afficher les résultats dans la console :
- 22/7
- 4^2 + 5
- (5+4^2)/(7+4^3)

Solution 2

```
Editeur :
a = 22/7;
b = 4^2 + 5;
c = (5+4^2)/(7+4^3);
disp(a)
disp(b)
disp(c)
```

```
Console :
    3.1428571

    21.

    0.2957746
```

Remarque : On lance les calculs à l'aide du menu de l'éditeur : **"Exécuter / Charger dans Scilab"** ou **"Exécuter / Exécuter le fichier dans Scilab".**

3- Saisir et afficher des données

Exercice 1
Ecrire un programme dans l'éditeur Scilab effectuant les opérations suivantes :
- Saisir une valeur a dans la console ;
- Calculer le carré de la valeur a ;
- Afficher le résultat.

Solution 1

```
Editeur :
a = input("Entrer la valeur de a : ");
r = a^2;
printf("Résultat : %f\n",r)
```

```
Console :
Entrer la valeur de a : 5
Résultat : 25.000000
```

Exercice 2
Ecrire un programme dans l'éditeur Scilab effectuant les opérations suivantes :
- Saisir deux valeurs a et b dans la console ;
- Calculer le quotient a/b ;
- Afficher le résultat avec 4 décimales.

Solution 2

```
Editeur :
s = input("Entrer les valeurs de a et de b : [a,b]= ");
r = s(1)/s(2);
printf("Résultat : %0.4f\n",r)
```

```
Console :
Entrer les valeurs de a et de b : [a,b]= [22,7]
Résultat : 3.1429
```