

Introduction à Scilab II¹

Fabien Navarro

1 Programmation de fonctions

Scilab permet de définir de nouvelles fonctions utilisateurs, la syntaxe est la suivante :

```
function [<arguments renvoyés>]=<nom_de_la_fonction>(<arguments d'entrée>)  
...instructions ...  
endfunction
```

Tester l'exemple suivant :

```
function y=f(x)  
y=x^2-1  
endfunction  
f(2)  
z=f(100)
```



Une fonction ne modifie pas les valeurs des variables définies en dehors de la fonction.

2 Graphiques

Afin de vous familiariser avec les commandes concernant les graphiques, voici quelques instructions à tester.

Pour effectuer un graphique en deux dimensions, on utilise les commandes `plot` ou `plot2d` (ou encore `fplot2d` pour les fonctions). Par exemple : `x=[1,3,4,9]` ;

```
y=[2,7,-6,-3];  
plot2d(x,y)
```

Cela relie les points de coordonnées $(x(1), y(1))$, $(x(2), y(2))$, ...

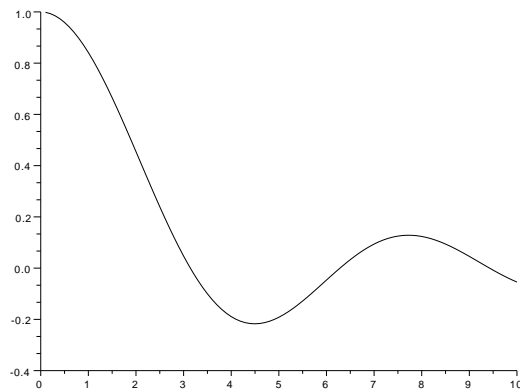
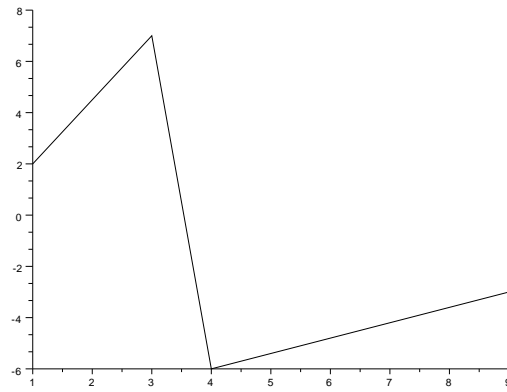
Un autre exemple,

```
x=0.1:0.1:10;  
y=sin(x)./x;  
clf(); // efface le contenu de la fenetre graphique  
// on peut aussi utiliser xbas() avec scilab4  
plot2d(x,y)
```

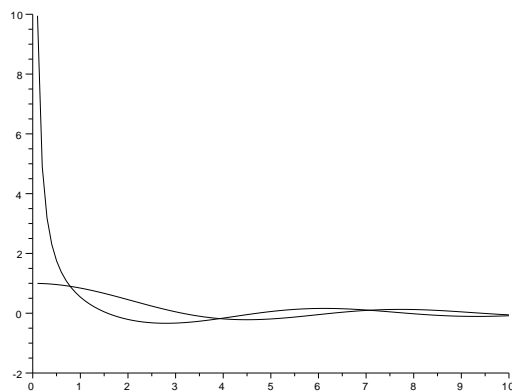
Il est également possible de tracer plusieurs courbes sur un même graphique (par superposition ou simultanément) :

```
x=0.1:0.1:10;  
y=sin(x)./x;  
z=cos(x)./x;
```

1. Très largement inspiré du cours : Introduction à Scilab, Graphiques, fonctions Scilab, programmation, saisie de données Jean-Philippe Chancelier & Michel De Lara cermics, École des Ponts, 2003.



```
xset("window",1); (ou scf(1))
// ouvre une nouvelle fenetre 1
// (la fenetre par default est 0)
plot2d(x,y);plot2d(x,z) // les deux graphiques sont superposes
//ou en utilisant plot
plot(x,y,x,z) //dans ce cas Scilab attribue une couleur a chaque courbe
```



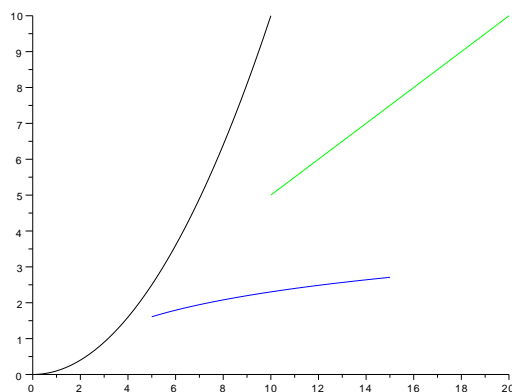
Même exemple mais simultanément :
 $x=x'$; $y=y'$; $z=z'$;

```

// x, y et z sont maintenant des vecteurs colonnes
// ce qui est obligatoire quand on veut dessiner plus de deux courbes.
// Noter qu'il est preferable de travailler avec des
// vecteurs colonnes!
// et pour s'assurer qu'un vecteur est bien en colonnes on peut utiliser :
// i.e. x=x(:)
clf();plot2d(x,[y z])
// idem, mais les deux graphiques sont traces chacun avec sa couleur

On peut aussi tracer plusieurs courbes (simultanément) :
M=[(0:0.1:10)' (5:0.1:15)' (10:0.1:20)'];
N=[0.1*M(:,1).^2 log(M(:,2)) 0.5*M(:,3)];
xbasc();plot2d(M,N);

```



On peut également tracer plusieurs courbes dans une même figure mais sur des fenêtres graphiques différentes en utilisant `subplot`.

```

x=1:10;
...
plot2d2(x,x);
...
plot2d3(x,x);

```

Exercice 0. A titre d'exercice, chercher de l'aide sur la fonction `subplot` et reproduire la figure 1

2.1 Titre et légende

```

annees=[1985 1986 1987 1988 1989 1990 1991 1992];
annee_ref=2004 ;
prix_annee_ref=[1.27 1.05 1.03 1.00 1.04 1.04 1.01 0.96];
prix_courant=[0.85 0.72 0.73 0.73 0.78 0.81 0.81 0.79];
clf();plot2d2(annees',[prix_annee_ref ; prix_courant]');
xlabel('Litres de supercarburant ...
en euros de l'annee ' +string(annee_ref),...
'(annees)', '(prix)')
// Noter le " pour signifier l'apostrophe dans l'annee.
// Dans une instruction trop longue pour tenir dans une ligne,

```

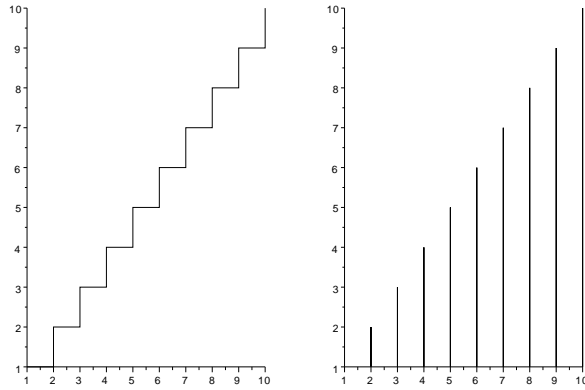
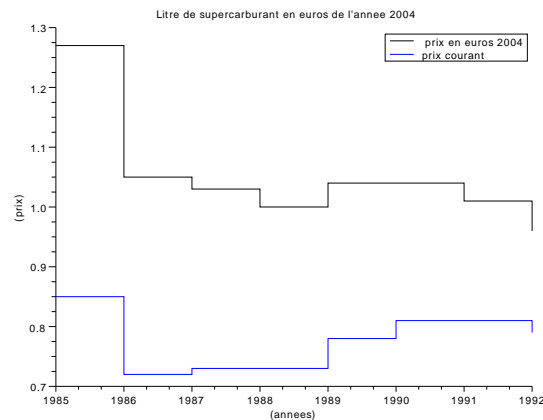


FIGURE 1 – Créneaux plot2d2, barres plot2d3

```
// mettre ... avant de passer a la ligne
legends(['prix en euros '+ string(annee_ref); 'prix courant'], [1,2], 'ur')
```



La fonction *Scilab* `plot2d` a de nombreux arguments optionnels permettant de spécifier des attributs d'un graphique : couleur, style et épaisseur des traits, échelle, dimensions du cadre, etc. Pour découvrir ces arguments optionnels, taper `help plot2d`.

Pour découvrir les attributs associés aux objets graphiques, utiliser le menu `Edit` de la fenêtre graphique. A travers ce menu, on peut également modifier en temps réel des attributs.

A titre indicatif, noter que les manipulations des attributs d'un graphique peuvent également être effectuées au moyen de commandes *Scilab* (taper `help graphics_entities`)

L'option `style` permet notamment de changer la couleur :

```
x=(1:100)';
y=rand(100,1);
clf();plot2d(x,y)
// Par défaut, les points sont reliés par des droites de couleur noire
clf();plot2d(x,y,style=5)
// Avec l'option style=5, on obtient du rouge.
// Le code par défaut des couleurs se découvre par la commande getcolor()
Ce changement de couleur peut également être obtenu en utilisant le menu Edit de la fenêtre graphique.
```

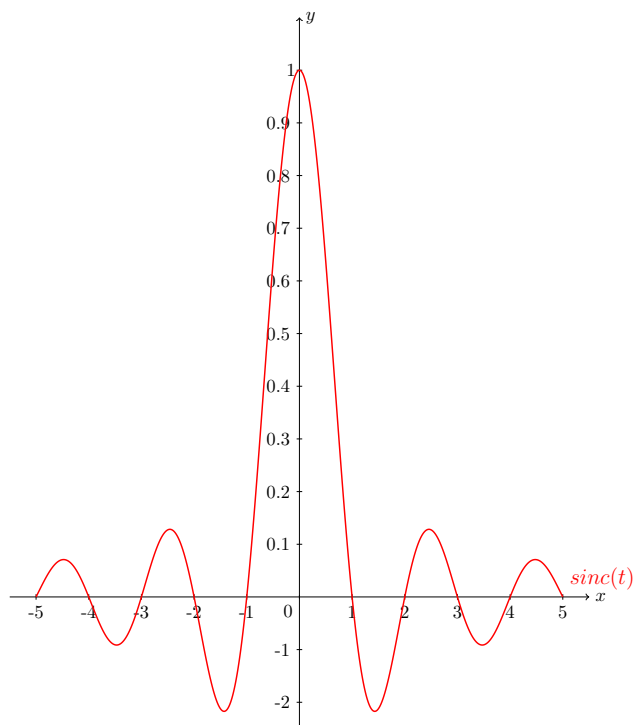
l'option `rect` permet de fixer les échelles.

```
y=[6:-1:0,1:5] ; x =[y,y,y,y,y];  
clf();plot2d(1:60,x);  
// Un graphique simple dans la fenetre courante.  
  
Tester également :  
xset('window',2);  
// Ouvre une deuxieme fenetre graphique.  
clf();plot2d(1:60,x,rect=[0,0,70,10]);  
// Les echelles sont fixees par rect =[xmin,ymin,xmax,ymax]
```

3 Exercices

Exercice 1.

- 1) Ecrire une fonction *sinc* définie par : $\text{sinc}(t) = \frac{\sin(t)}{t}$.
- 2) Représenter graphiquement la fonction *sinc* en utilisant par exemple la fonction `plot` ou `fplot2d`. Attention à l'ensemble de définition de la fonction : on suppose que la fonction vaut 1 si $t = 0$, réaliser par exemple un test avec une instruction avec l'aide de la fonction `find` : `if (expression) instruction 1; else instruction 2;` (`help if` pour la syntaxe).
- 3) Ajouter un titre et une légende (pour avoir plus de détails sur les options graphiques taper `help set`, `help gcf`, `help gca`,...).



Exercice 2.

Soit $A \in \mathcal{M}_n(\mathbb{R})$

$$A = \begin{pmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -1 \\ 0 & \dots & 0 & -1 & 2 \end{pmatrix}$$

- 1) Ecrire une fonction `matriceA` qui prend en argument la taille de la matrice A et qui retourne A .
- 2) En prenant $n = 100$ résoudre le système $Ax = b$, avec $b = (1, \dots, 1)^T$ en utilisant :
 - a) la fonction `inv`;
 - b) la fonction `linsolve`;
 - c) la fonction `\` (backslash).
- 3) Chercher de l'aide sur la fonction `timer` et comparer les temps de calculs pour chacune des différentes méthodes avec $n = 100, 200, 500, 1000$.

Exercice 3.

On considère un marché correspondant à un produit P quelconque. On connaît, pour P , la courbe de demande de l'ensemble des consommateurs en fonction du prix de vente, ainsi que la courbe d'offre de la part des producteurs en fonction du prix d'achat. Les courbes d'offre et de demande sont les suivantes :

$$D(p) = k_0 \left(1 + \frac{1}{p}\right) \quad (1)$$

$$O(p) = k_1 p^\alpha \quad (2)$$

Dans la suite, on prendra $k_0 = 2$, $k_1 = 75$ et $\alpha = \frac{3}{2}$.

- 1) Trouver le prix d'équilibre p^* du produit P sur le marché étudié (utiliser la fonction `fsolve` qui permet de trouver les zéros d'une fonction).
- 2) Représenter sur un même graphique (avec `fplot2d` par exemple) les fonctions D et O .
- 3) Ajouter le point p^* sur la graphique précédent (par exemple `plot(x,y,'o')`).