

Feuille de TP n° 7

Représentation graphique avec Scilab

Ce TP accompagne le chapitre 3 (Informatique et Algorithmique) : **Représentation graphique en Scilab**. Créez le dossier `..\ECS1B_TPIInfo\TP7\` et faites-en le répertoire courant de Scilab. Tous les scripts devront être sauvegardés dans ce dossier.

I Entraînement

1) plot VS plot2d

- Exécuter les commandes suivantes dans la console :

```
-->X=[0,1,4,11,10,15,5,4,2,1.5,1,0.5,0];
-->Y=[0,2,4,4,7,10,10,8,7.5,7.5,9.5,8.5,8.5];
-->plot(X,Y)
-->plot(Y,X)
```

Par défaut, les tracés successifs se superposent. Il faut penser à utiliser la commande `clf()` ; pour effacer le contenu de la fenêtre graphique avant de tracer une nouvelle figure.

- Exécuter les commandes suivantes dans la console :

```
-->clf(); plot(X,Y,'r')
-->clf(); plot(X,Y,'g')
-->clf(); plot(X,Y,'x')
-->clf(); plot(X,Y,'o')
-->clf(); plot(X,Y,'--')
-->clf(); plot(X,Y,':')
-->clf(); plot(X,Y,'c*')
-->clf(); plot(X,Y,'rx')
-->clf(); plot(X,Y,'go:')
```

- Exécuter les commandes suivantes dans la console :

```
-->clf(); plot(X,Y,-X,Y)
-->clf(); plot(X,Y,'k',-X,Y,'k')
-->clf(); plot(X,Y,'r-',-X,Y,'g*')
```

- Lorsque l'on désire tracer plusieurs courbes ayant le même vecteur d'abscisses X, on peut n'écrire qu'une seule fois X et regrouper les vecteurs ordonnées. Exécuter les commandes suivantes :

```
-->X=[5,9,11,13,14,17,20,21];
-->EM=[23.5,24,23,23.5,22,24,25,24];
-->MLP=[23.5,24,23,22.5,22,23,20,21.8];
-->FF=[19,17,19,20,19,19.5,19,19.5];
-->JLM=[17,18,17,18.5,20,18,19,19.3];
-->clf(); plot(X,[EM;MLP;FF;JLM])
```

Il s'agit de courbes de sondages (*elles correspondent aux intentions de votes pour les quatre candidats de tête à l'élection présidentielle de 2017 relevées à 8 dates d'avril 2017*). Les couleurs choisies par Scilab ne correspondent pas vraiment aux couleurs usuellement attribuées aux candidats. Hélas avec cette syntaxe on ne peut pas changer les couleurs. Plusieurs options s'offrent à nous :

— L'ancienne syntaxe :

```
| -->clf(); plot(X,EM,'c',X,MLP,'k',X,FF,'b',X,JLM,'r')
```

— L'utilisation de la fonction `plot2d` qui fonctionne essentiellement comme `plot` mais avec des variantes dans la syntaxe, notamment pour la gestion des options (les couleurs sont symbolisées par des numéros) :

```
| -->clf(); plot2d(X',[EM',MLP',FF',JLM'],style=[4,1,2,5])
```

Le apostrophes après chaque vecteur servent à les transposer (c'est-à-dire les transformer en vecteurs lignes).

On liera attentivement le chapitre 4 pour d'autres détails sur les fonctions `plot` et `plot2d` et sur leurs différences (il faut savoir les utiliser toutes les deux).

• Un intérêt de `plot2d` est que l'on peut modifier la zone graphique et insérer des légendes directement dans la syntaxe de la fonction. Exécuter les commandes suivantes :

```
| -->legende="Macron@Le Pen@Fillon@Mélenchon";  
| -->clf();  
| -->plot2d(X',[EM',MLP',FF',JLM'],style=[4,1,2,5],rect=[5,16,21,25],leg=legende)
```

• Exécuter les commandes suivantes :

```
| -->Z=[0,3,-2,-5,-1,2,4,7]  
| -->clf(); plot2d(X,Z);  
| -->clf(); plot2d2(X,Z);  
| -->clf(); plot2d3(X,Z);
```

2) Tracé de courbes représentatives de fonctions

Tracer la courbe représentative d'une fonction sur un intervalle, revient à tracer une infinité de points. Or il n'est évidemment pas possible pour un ordinateur de réaliser une infinité d'instructions. Pour tracer une courbe avec Scilab, on va représenter un nombre fini de points et de les relier par une ligne continue.

• Exécuter sans écho dans la console le script `fct1.sci` du TP précédent. Exécuter ensuite les commandes suivantes :

```
| -->x=linspace(-15,15,10);  
| -->clf(); plot(x,fct1(x),'r')
```

• Il faut bien sûr un plus grand nombre de points pour que l'approximation de la courbe par une ligne brisée donne l'illusion de courbe.

```
| -->x=linspace(-15,15,20); clf(); plot(x,fct1(x),'r')  
| -->x=linspace(-15,15,30); clf(); plot(x,fct1(x),'r')  
| -->x=linspace(-15,15,1000); clf(); plot(x,fct1(x),'r')
```

• Il y a d'autres syntaxes pour tracer la courbe représentative d'une fonction. Exécuter ensuite les commandes suivantes :

```
| -->x=linspace(-15,15,1000);  
| -->clf(); plot(x,fct1,'r')  
| -->clf(); plot2d(x,fct1(x),5)  
| -->clf(); fplot2d(x,fct1,5)
```

3) Diagrammes à barres et histogrammes

- Exécuter ensuite les commandes suivantes :

```
-->x=1:5; y=[10,5,9,7,1];
-->clf(); bar(x,y,'r')
-->z=[6,3,3,5,2];
-->clf(); bar(x,[y',z'])
-->clf(); bar(x,[y',z'],'stacked')
```

- Exécuter plusieurs fois les commandes suivantes :

```
-->x=2*rand(1:50)-1;
-->clf(); histplot(5,x)
-->clf(); histplot(10,x)
-->clf(); histplot(10,x,normalized=%f)
-->clf(); histplot([-1,-0.5,0.5,0.75,1],x)
```

4) Personnalisation des fenêtres graphiques

- Exécuter les commandes suivantes :

```
-->clf(); x=linspace(-%pi,%pi,1000);
-->plot2d(x,[cos(x)',(1-x.^2/2)'])
```

Réduire la zone du graphique visible à l'intervalle $[-1,1]$ à l'aide la commande `square`. Ajouter ensuite un quadrillage, un titre, une légende et annoter les axes (à l'aide des fonctions `xgrid`, `title`, `legend`, `xlabel`, `ylabel` respectivement).

- Exécuter les commandes suivantes :

```
-->x=2*rand(1:50)-1;
-->scf();
-->subplot(2,3,1)
-->histplot(5,x)
-->subplot(2,3,3)
-->histplot(10,x)
-->subplot(2,3,5)
-->histplot(20,x)
```

Que fait `subplot` ? Que fait la commande `scf()` ?

II Exercices

Commencez par taper `clear` dans la console. Pour vous y retrouver plus facilement, veuillez écrire

- en commentaire au début de chaque programme et chaque fonction le numéro du TP et de l'exercice concerné.
- sur cette feuille de TP, le nom que vous avez donné au programme (extension en `.sce`) ou à la fonction (extension en `.sci`)

Vous testerez dans la console Scilab tous les programmes et fonctions que vous écrirez (sans oublier de les exécuter au préalable).

Exercice 1. Tracer la courbe représentative des cinq fonctions de l'exercice 3 du TP 6, sur des intervalles bien choisis et sur cinq graphiques différents.

Exercice 2. Pour tout $a > 0$, posons $\gamma_a : x \mapsto \begin{cases} 0 & \text{si } x < 0 \\ x^{a-1}e^{-x} & \text{si } x > 0 \end{cases}$.

- 1) Implémenter une fonction en Scilab qui prend en entrée x et a et qui calcule $\gamma_a(x)$ (et qui soit compatible avec un traitement vectoriel pour la première coordonnée).
- 2) Écrire un programme qui représente, sur une même fenêtre graphique, les fonctions g_a pour $a \in \left\{ \frac{1}{100}, \frac{1}{2}, 1, \frac{3}{2}, 2, \frac{5}{2}, 3, 100 \right\}$, sur l'intervalle $[-1, 5]$.

On répartira les tracés sur deux lignes et on utilisera une boucle for.

Exercice 3. Pour tout $n \in \mathbb{N}^*$, posons $f_n : x \mapsto \begin{cases} 0 & \text{si } x < 0 \\ x^n & \text{si } x \in [0, 1] \\ 1 & \text{si } x > 1 \end{cases}$ puis $f : x \mapsto \lim_{n \rightarrow +\infty} f_n(x)$.

Sur un même graphique, représenter les fonctions f et f_n pour $n \in \{1, 2, 5, 10, 20, 50\}$, sur l'intervalle $[-1, 2]$. Commenter.

Exercice 4. Dans le TP6, nous avons implémenté en Scilab les fonctions $S_n : x \mapsto \sum_{k=0}^n \frac{(-1)^k x^{2k}}{(2k)!}$,

$n \in \mathbb{N}$, sous le nom `sommecos.sci`, et nous les avons comparées avec la fonction cosinus. Sur une même fenêtre graphique découpée en cinq zones, représenter

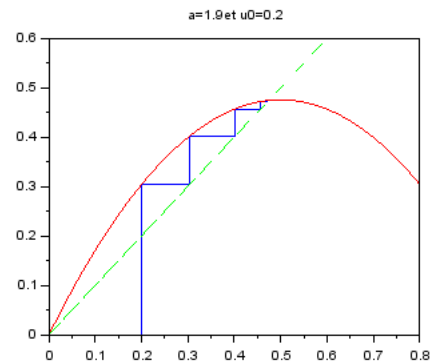
- sur la première ligne : les fonctions `cos` et S_n pour $n \in \{5, 6, 10, 20\}$, sur l'intervalle $[-2\pi, 2\pi]$.
- sur la deuxième ligne : les fonctions `cos` et S_n pour $n \in \{20, 30, 40, 50\}$, sur l'intervalle $[-10\pi, 10\pi]$.

Exercice 5. Pour tout $a \in \mathbb{R}_+^*$, considérons la fonction $f_a : x \mapsto ax(1-x)$ sur $[0, 1]$.

- 1) Implémenter une fonction en Scilab qui prend en entrée x et a et qui calcule $f_a(x)$ (et qui soit compatible avec un traitement vectoriel pour la première coordonnée).

- 2) Le graphique ci-contre représente la trajectoire (en bleu) d'une suite $(u_n)_{n \in \mathbb{N}}$ définie par $u_0 \in [0, 1]$ et, pour tout $n \in \mathbb{N}$, $u_{n+1} = f_a(u_n)$. La fonction f_a est représentée en rouge et la droite $y = x$ en pointillés verts.

La courbe bleue est la ligne brisée qui joint les points $(0, u_0), (u_0, u_1), (u_1, u_1), (u_1, u_2), \dots$



- a) Écrire une fonction qui prend en entrée $a \in \mathbb{R}_+^*$, $u_0 \in [0, 1]$ et $n \in \mathbb{N}^*$ et qui reproduit la figure ci-contre, la trajectoire bleue s'arrêtant au point (u_n, u_{n+1}) .
- b) Tracer huit graphiques (répartis sur quatre lignes dans une même fenêtre graphique) pour $n = 100$ et $(a, u_0) \in \{(0.9, 0.6), (1.7, 0.9), (2.3, 0.2), (2.8, 0.9), (3, 0.1), (3.4, 0.7), (3.5, 0.8), (3.9, 0.1)\}$.

Exercice 6 (transition avec un prochain TP...). Représenter le diagramme à barres traduisant le tableau ci-dessous :

k	2	3	4	5	6	7	8	9	10	11	12
p_k	$\frac{1}{36}$	$\frac{1}{18}$	$\frac{1}{12}$	$\frac{1}{9}$	$\frac{5}{36}$	$\frac{1}{6}$	$\frac{5}{36}$	$\frac{1}{9}$	$\frac{1}{12}$	$\frac{1}{18}$	$\frac{1}{36}$

Nous avons vu dans le chapitre *Variables aléatoires réelles finies* que $(p_k)_{2 \leq k \leq 12}$ est la loi d'une variable aléatoire représentant la somme des chiffres lorsqu'on lance deux dés.

Exercice 7. Pour tout $n \in \mathbb{N}^*$, notons $S_n = \sum_{k=1}^n \frac{1}{k^2}$.

1) Écrire un script qui représente les 1000 premiers termes de la suite $(S_n)_{n \geq 1}$.

On utilisera une boucle `for` pour calculer les termes successifs de la somme et on les stockera dans un vecteur `V`. Il suffit alors d'utiliser `plot(1:1000, V)` pour les représenter.

2) Superposer la droite d'équation $y = \frac{\pi^2}{6}$.

On pourra utiliser le vecteur `ones(1:1000)%pi^2/6`.*