VII – Graphisme avec Python

Sauf précision contraire, les commandes utilisées dans ce chapitre sont contenues dans la bibliothèque `matplotlib.pyplot`, qu'il faut donc importer en début de programme.

1/ Premiers tracés

Voici déjà quelques commandes fondamentales pour effectuer des tracés :

`plot(X,Y)`	place les points dont les abscisses forment la liste X et les ordonnées forment la liste Y.
`text(x,y,"texte")`	affiche le texte au point de coordonnées (x, y).
`savefig("chemin.eps")`	enregistre le graphique obtenu.
`show()`	affiche le graphique.

Exemple : représentons la fonction carré sur $[0;3]$ grâce à des compréhensions de listes.

```
X=range(4)
Y=[x**2 for x in X]
plot(X,Y)
show()
```

```
X=[0.03*i for i in range(101)]
Y=[x**2 for x in X]
plot(X,Y)
show()
```

On observe que, par défaut, Python trace une ligne brisée bleue reliant les points donnés. Pour obtenir l'illusion d'une courbe, il suffira de placer suffisamment de points.

Remarques :

❶ Si l'on ne fournit qu'une seule liste en argument à `plot()`, Python suppose que c'est la liste des ordonnées et prend alors pour abscisses les entiers naturels successifs. Ceci permet de représenter une suite.

❷ L'instruction `savefig()` se place **après** `plot()` et **avant** `show()`.

❸ Ce qui est placé après le `show()` se trouve sur un nouveau graphique.

Un petit bonus

Voici une commande de la bibliothèque `numpy` qui facilite les créations de listes :

`linspace(a,b,n)`	liste de n valeurs régulièrement espacées de a à b inclus.

Pour tracer la courbe d'une fonction f sur un intervalle $[a;b]$, il suffit alors de taper les instructions suivantes :

```
X=linspace(a,b,100)
Y=[f(x) for x in X]
plot(X,Y)
```

2/ Mise en forme

2.1) Axes et fenêtre graphique

Lors du tracé, Python détermine les valeurs extrêmes prises par les coordonnées et ajuste automatiquement la fenêtre graphique. On peut cependant reprendre le contrôle :

`figure(figsize=(a,b))`	la fenêtre graphique est de $a \times b$ pouces (avec a et b entiers).
`xlim(a,b)`	les abscisses vont de a à b.
`ylim(a,b)`	les ordonnées vont de a à b.
`axis('off')`	les axes et la fenêtre graphique ne sont pas tracés.
`axis('scaled')`	impose la même échelle aux deux axes.
`grid(True)`	trace le quadrillage

Comme les graduations automatiques de `Python` ne sont pas toujours pertinentes, il est bon de pouvoir les gérer soi-même :

`xticks(L)`	place les graduations de l'axe (Ox) aux abscisses de la liste L.
`xticks(L1,L2)`	L1 : abscisses des graduations, L2 : labels associés.
`yticks()`	même chose avec l'axe (Oy).

Exemple : dessiner le cercle trigonométrique à l'aide de l'instruction `axis('scaled')`. Sans cette dernière, on obtient une ellipse car les unités graphiques diffèrent selon les axes.

2.2) Titres et labels

On enjolive les dessins en rajoutant un peu de texte

`xlabel('texte')`	affiche un texte le long de l'axe des abscisses.
`ylabel('texte')`	même chose le long de l'axe des ordonnées.
`title('texte')`	ajoute un titre au dessin.

En enchaînant les `plot()`, on trace plusieurs courbes sur la même figure. Les couleurs changent alors automatiquement, et l'on peut en outre identifier les différentes courbes en leur associant des labels ainsi :

`plot(x,y,label='texte')`	donne un nom à la courbe.
`legend()`	affiche la légende.

Exemple : tracer sur une même figure les courbes des fonctions sinus et cosinus, en utilisant un titre, des labels et une légende. On utilisera des graduations adaptées à la situation.

Remarque : en mettant le label `'$\pi$'` à la place du label `'pi'`, on affiche la lettre grecque.

2.3) Options graphiques

On a déjà vu l'option `label` de la fonction `plot()`. Voici quatre autres options pratiques :

`linewidth=x`	fixe l'épaisseur de la courbe à x pt.
`linestyle='symbole'`	modifie le style de la courbe reliant les points.
`marker='symbole'`	modifie le marqueur utilisé pour placer les points.
`color='lettre'`	modifie la couleur de la courbe.

Options pour `linestyle`

`''`	les points ne sont pas reliés.
`'-'`	tracé en trait plein.
`'--'`	tracé avec des tirets.
`':'`	tracé en pointillés.
`'-.'`	alternance de tirets et de points.

Options pour `marker`

`','` / `'.'` / `'o'`	pixel / point / disque.
`'^'` / `'v'` / `'<'` / `'>'`	triangle vers le haut / le bas / la gauche / la droite.
`'d'` / `'D'`	petit losange / grand losange.
`'h'` / `'H'`	petit hexagone / grand hexagone.
`'s'` / `'p'` / `'8'`	carré / pentagone / octogone.
`'*'` / `'+'` / `'x'`	astérisque / croix droite / croix penchée.

Options pour `color`

`'r'`	`'b'`	`'g'`	`'c'`	`'m'`	`'y'`	`'k'`	`'w'`
rouge	bleu	vert	cyan	magenta	jaune	noir	blanc

Remarques :

- Ces différentes options peuvent aussi se mettre dans une chaîne de caractères fournie en argument à `plot()`, comme dans `plot(X,Y,'r--*')`.
- Si l'on modifie le marqueur dans la chaîne, alors par défaut `linestyle=''`.
  Il faudra impérativement penser à le préciser si l'on veut relier les marqueurs.

Exemple : tracer les courbes des fonctions racine, carré et cube sur $[0;1]$ en se lâchant.

3/ Application : ma première fractale

La courbe de Von Koch est définie par récurrence de la façon suivante :

- On part d'un segment [AB] quelconque (mais quand même pas réduit à un seul point !).
- On le divise en trois parties égales [AC], [CE] et [EB].
- On construit un triangle équilatéral direct CED sans base au-dessus du segment [CE].
- C'est la **transformation de Von Koch**, qui nous donne la ligne brisée [A,C,D,E,B]. On réitère ensuite ce procédé avec chacun des segments [AC], [CD], [DE] et [EB].

Pour dessiner cette courbe en `Python`, on représentera ses points à l'aide de leurs affixes. Les courbes de Von Koch de différents niveaux seront donc des listes de nombres complexes construites à l'aide des deux fonctions suivantes :

❶ La fonction `transfoS(A,B)` applique la transformation de Von Koch au segment [A,B] et renvoie la ligne brisée [C,D,E,B]. On a exclu le point A pour éviter les doublons.

❷ La fonction `transfoLigne(L)` applique la fonction précédente à chaque segment de la ligne brisée L et renvoie la ligne brisée obtenue.

Voici les codes de ces deux fonctions :

```
w=(1+1j*3**0.5)/2

def transfoS(A,B):
    h=(B-A)/3
    C,E = A+h,B-h
    D=C+h*w
    return [C,D,E,B]
```

```
def transfoLigne(L):
    R=[L[0]]
    n=len(L)
    for i in range(n-1):
        R=R+transfoS(L[i],L[i+1])
    return R
```

En partant de `L=[-1+0j,1+0j]` par exemple et en répétant l'instruction `L=transfoLigne(L)`, on obtient les courbes de Von Koch de différents niveaux. Il ne reste plus qu'à les tracer en capturant d'abord les parties réelles et imaginaires des affixes :

```
def trace(L):
    X=[z.real for z in L]
    Y=[z.imag for z in L]
    plot(X,Y)
    show()
```

Exercices

Exercice 1.
Tracer les courbes des fonctions suivantes :

1) $f : x \longmapsto \sqrt{1-x^2}$ sur $[\,-1\,;1\,]$.
2) $g : x \longmapsto \mathrm{e}^{-1/x^2}$ sur $[\,-2\,;2\,] \setminus \{0\}$. *(Comment gérer la valeur interdite ?)*
3) $h : x \longmapsto \sin x - x$ sur $[\,0\,;6\pi\,]$.
4) $k : x \longmapsto \mathrm{e}^{-x/(x^2+1)}$ sur $[\,-12\,;12\,]$.

Exercice 2.
Tracer les courbes paramétrées suivantes :

1) $x(t) = \sin(2t)$ et $y(t) = \sin(3t)$ pour $t \in [\,0\,;2\pi\,]$. *(Courbe de Lissajous)*.
2) $x(t) = \sin(t) + \cos(t)$ et $y(t) = \cos(3t)$ pour $t \in [\,0\,;2\pi\,]$. *(Courbe rigolote)*
3) $x(t) = \cos^3(t)$ et $y(t) = \sin^3(t)$ pour $t \in [\,0\,;2\pi\,]$. *(Astroïde)*.
4) $x(t) = t/(1+t^4)$ et $y(t) = t^3/(1+t^4)$ pour $t \in \mathbb{R}$. *(Lemniscate de Bernoulli)*.

Exercice 3.
Même chose avec des coordonnées polaires ; cette fois, $x = \rho\cos\theta$ et $y = \rho\sin\theta$:

1) $\rho = 1 + \cos(\theta)$ pour $\theta \in [\,0\,;2\pi\,]$. *(Cardioïde)*.
2) $\rho = 1 + 2\cos(3\theta)$ pour $\theta \in [\,0\,;2\pi\,]$. *(Folioïde)*
3) $\rho = 1 + \tan(\theta/2)$ pour $\theta \in]\,-\pi\,;\pi\,[$. *(Bizarroïde)*
4) $\rho = \theta - \sqrt{\theta^2-4}$ pour $\theta \geqslant 2$. *(Spirale de Schwarz)*.
5) $\rho = (\theta - 10\pi)^2$ pour $\theta \in [\,0\,;20\pi\,]$. *(Spirale de Kandinsky)*.

Exercice 4.
Créer une fonction permettant de tracer un polygone régulier à n côtés.
On reliera de plus chaque sommet au centre.

Exercice 5.
On va dessiner quelques étoiles amusantes.

1) Commencer par tracer un pentacle rouge inscrit dans un cercle rouge.
   Pour cela, relier les points d'un pentagone régulier en sautant un point à chaque fois.
2) Faire la même chose avec une étoile à 7 branches : on prend un heptagone régulier et on saute 2 points à chaque fois.

Exercice 6.
Représenter sur une même figure les fonctions $f_\alpha : x \longmapsto x^\alpha$ pour $\alpha \in \{-2, -1, 0.5, 1, 2, 3\}$.
On se restreindra à l'intervalle $[\,0\,;2\,]$.

Exercice 7.
Représenter sur une même figure les courbes des fonctions exp, ln et Id.
Attention à bien choisir les intervalles pour chaque courbe.

Qu'observe-t-on ?

Exercice 8.
On définit sur $\mathbb{R}$ les fonctions $f : x \longmapsto \dfrac{\mathrm{e}^x - \mathrm{e}^{-x}}{2}$ et $g : x \longmapsto \ln\left(x + \sqrt{x^2+1}\right)$.

1) Représenter sur une même figure les courbes des fonctions f, g et Id.

   Qu'observe-t-on ?

2) Afficher des listes de valeurs de $f \circ g$ et $g \circ f$ arrondies à deux décimales.

   Que peut-on en conclure ?

Exercice 9.
Un objet de masse $m = 1$ kg est lancé par une fenêtre située à 10 mètres de haut.
Il est soumis à son poids $\overrightarrow{\mathrm{P}} = m\overrightarrow{g}$ et à la force de frottement $\overrightarrow{\mathrm{F}} = -k \left\| \overrightarrow{v} \right\| \overrightarrow{v}$.
Sa vitesse initiale est de $1\ \mathrm{m\,s^{-1}}$ vers la droite et $2\ \mathrm{m\,s^{-1}}$ vers le haut.
De plus, on donne $\left\| \overrightarrow{g} \right\| = 9.81\ \mathrm{m\,s^{-2}}$ et $k = 0.2\ \mathrm{kg\,m^{-1}}$.

1) Calculer l'accélération, la vitesse instantanée et la position (horizontale et verticale) de notre mobile toutes les 0.02 secondes depuis le lancer.

   On s'arrêtera quand l'altitude devient négative (le sol est atteint), et l'on stockera toutes les données calculées dans des listes.

2) Représenter sur différents graphiques :

   (a) la trajectoire de notre projectile

   (b) sa vitesse et son altitude en fonction du temps

3) À quelle distance du mur se situe donc le point de chute ?