TD n°05 : Tracer de graphes sous Python

Rédigé par Pascal Delahaye

14 octobre 2016

1 La fonction PLOT de matplotlib.pylab

Utilisation de la fonction plot() de matplotlib.pylab

Cette fonction `plot()` permet de représenter des points $M_0(x_0,\ y_0)$, $M_1(x_1,\ y_1)$, $M_2(x_2,\ y_2)$, $\ldots M_n(x_n,\ y_n)$:

- Les abscisses sont stockées dans une liste `X=[`x_0`,`x_1`,`$x_2\ldots$`,`x_n`]`
- Les ordonnées sont stockées dans une liste `Y=[`y_0`,`y_1`,`$y_2\ldots$`,`y_n`]`
- On lance alors l'instruction :

```
plot(X,Y)
```

- On peut souvent construire les listes X et Y à l'aide d'une boucle 'for".
  Cependant, il est parfois possible de gagner du temps en procédant comme ci-dessous :

- Pour obtenir X :
  Une instruction très utile pour obtenir une liste d'abscisses : `X = arange(a,b,p)`
  Elle permet de construire une liste X contenant toutes les valeurs de "a" à "b" par pas de "p"
  Cette fonction est disponible dans la bibliothèque `maplotlib.pylab`.

- Pour obtenir Y :
  Une instruction très utile pour obtenir la liste des ordonnées correspondantes : `Y = f(X)`
  X est ici la liste des abscisses et f une fonction construite exclusivement à partir de fonctions de la bibliothèque `matplotlib.pylab`

1. ⚠ Lorsqu'on travaille dans l'éditeur, les graphes n'apparaissent qu'après avoir tapé : `show()`

Pour tracer plusieurs courbes sur un même dessin, il suffit d'exécuter la fonction ***`plot()`*** *pour chacune des courbes.*

Exercice : 1

(∗) Donner la représentation graphique au voisinage de 0 de la fonction f définie par $f(x) = |x \sin \frac{1}{x}|$.
Sous ***`pylab`*** *la fonction donnant la valeur absolue est* ***`absolute()`****.*

Exercice : 2

(∗) Donner sur un même graphe les représentations graphiques de f et g définies par $\begin{cases} f(x) = \lfloor x \rfloor \\ g(x) = \lfloor x \rfloor - (x - \lfloor x \rfloor)^2 \end{cases}$.

Exercice : 3

(∗∗) Construire une fonction permettant de représenter dans le plan complexe les racines nième de l'unité.

Exercice : 4

(∗∗) Construire une fonction permettant de représenter le graphe de la fonction f_N d'expression $f_N(x) = \sum_{k=0}^{N} \frac{\cos(3x)}{2^k}$.

Exercice : 5

(∗) Construire la courbe paramétrée d'équation $\begin{cases} x = \cos(t)(1-\cos t) \\ y = \sin(t)(1-\cos t) \end{cases}$ pour $t \in [0,\ 2\pi]$.

Exercice : 6

(∗∗) Construire le graphe de l'image d'une droite $D : y = ax + b$ par la fonction exponentielle complexe.

Exercice : 7

(∗∗) Un joueur joue à un jeu de pile ou face : s'il gagne, il remporte x euros et s'il perd, il doit donner y euros.
Il décide de jouer n parties.
Représenter sur un graphe, l'évolution de ses gains au cours du temps pour différentes valeurs de x, y et n.

On pourra pour effectuer la simulation, utiliser la fonction `randint()` *de la bibliothèque* `random`

2 La fonction PLOT de sympy

Utilisation de la fonction plot() de sympy

Cette fonction `plot()` permet de tracer les graphes de fonctions tirées de la bibliothèque sympy.

- Il faut commencer par définir la variable par l'instruction : `x = symbols('x')`
  Ne pas oublier d'importer également cette fonction `symbols()` depuis la bibliothèque `sympy`.
- On applique alors la fonction `plot()` de la façon suivante :

  ```
  plot(f1(x),f2(x),f3(x),(x,a,b))
  ```

  On trace ainsi sur un même dessin les différentes représentations graphiques pour $x \in [a, b]$
- ⚠ les fonctions f_1, f_2... doivent impérativement être construites à l'aide des fonctions usuelles de `sympy`.

Exercice : 8

(∗) Visualiser sur un graphe la nature des solutions de l'équation $\tan x = x$.

Exercice : 9

(∗∗) Représenter sur un même graphe les approximations de la fonction exponentielle au voisinage de 0 suivantes :

$$\begin{cases} f_1(x) = 1 + x \\ f_2(x) = 1 + x + \frac{x^2}{2} \\ f_3(x) = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} \end{cases}$$

Exercice : 10

(∗) Représenter la famille de droites $D_m : (1-m)x + 2y - m = 0$ pour $m \in [\![-10, 10]\!]$. Que constatez-vous ?

Pour cela, vous pourrez créer la liste F des fonctions correspondantes puis taper l'instruction `plot(*F)`