Lycée Marcelin BERTHELOT
BCPST 1C – Mme MOREL

Représentations graphiques

1 Introduction

La représentation des résultats d'un programme informatique passe très souvent par une illustration graphique.
Les bibliothèques `matplotlib.pyplot` et `numpy` destinées à tracer et visualiser des données sous forme graphique, permet en effet de représenter des points, des courbes et des surfaces (fonctions, trajectoires, nuages de points, ...); d'animer des systèmes dynamiques; d'organiser des données en histogramme, en "camembert", et bien d'autres choses encore...
Les alias usuels de ces deux bibliothèques sont respectivement `plt` et `np`
Comment charger ces bibliothèques ?

Pour créer les **abscisses**, on peut créer une liste ou par exemple utiliser dans `numpy` :

- `np.linspace(a,b,n)` : génère un tableau ligne de n valeurs uniformément réparties entre a et b INCLUS. (il y a a et b dans ces n valeurs)
  Par exemple, `linspace(0,6,4)` contient les valeurs $0, 2, 4, 6$, que contient `np.linspace(-1,1,5)`?
- `np.arange(a,b,r)` : génère un tableau ligne dont les valeurs vont de a (inclus) à b EXCLUS, et le pas entre chaque valeur vaut r.
  On obtient ainsi une suite arithmétique de premier terme `a` et de raison r, qui s'arrête juste avant `b` (le réel `b` n'est pas atteint).
  Par exemple, que contient `np.arange(0,1,0.2)` ?
  On note que `arange` donne un tableau qui peut contenir des flottants ; alors que la commande `range()` ne peur contenir que des entiers. De plus, avec `arange`, on peut choisir une raison r égale à un flottant assez petit , de façon à obtenir un dessin bien lisse.

Pour créer le graphe, on peut utiliser dans la bibliothèque `matplotlib.pyplot`:

- `plt.pyplot(x,y,'options')` : crée le graphe d'abscisses contenues dans `x` et d'ordonnées contenues dans `y`, avec des options comme la couleur, pointillés, symboles, etc ...
- `plt.show()` : affiche le graphe créé précédemment.

2 Représentation de points (segments qui les relient)

2.1 Syntaxe

(1) Soit Y une **liste** dont on note n la longueur.

```
plt.plot(Y)
```

Cette commande permet de placer n points dans un repère orthonormé, dont les abscisses respectives sont $0, 1, , \ldots, n-1$ et les ordonnées respectives sont les éléments de la liste Y , et de les relier par des segments.

(2) Soient X et Y **deux listes de même longueur** n.

```
plt.plot(X,Y)
```

Cette commande permet de placer n points dans un repère orthonormé, dont les abscisses respectives sont les éléments de la liste X et les ordonnées respectives sont les éléments de la liste Y , et de les relier par des segments.

Exemple 1 Entrer les lignes de commandes suivantes :

```
L=[1,2,5]
M=[1,4,9]
plt.plot(L)
plt.plot(L,M)
plt.plot(L,M,'r')
plt.plot(L,M,'b')
plt.show()
```

2.2 Suites

Exemple 2 (suite explicite)
On se donne la suite définie par : $\forall n \in \mathbb{N}$, $u_n = 1 + 2^n$. Comment représenter les points $u_0, \ldots, u_{10}$ sur un graphe ?

Exemple 3 (suite récurrente)
On se donne la suite définie par :

$$u_0 = 1 \quad \text{et} \quad \forall n \in \mathbb{N}\,,\ u_{n+1} = \sqrt{u_n + 2}$$

Comment représenter les points $u_0, \ldots, u_{10}$ sur un graphe ?

Remarque 1 Le dynamisme des suites récurrentes (escalier ou spirale) sera vu en TP.

3 Graphe d'une fonction

La courbe d'une fonction est un ensemble infini de points. Pour tracer ce graphe, Python place quelques points (éventuellement beaucoup) et les relie par des traits.
Plus précisément, il commence par calculer les valeurs de f pour des abscisses discrétisées $x_1, \ldots, x_n$ puis il dessine la famille de points $(x_i, f(x_i))_{1 \leqslant i \leqslant n}$, avant de relier ces points par des segments.
La qualité de la représentation dépend donc du nombre de points choisis. Plus il y en a, plus on a l'impression que la courbe est lisse.

3.1 Syntaxe

Création des abscisses: voir l'introduction.

`x=np.linspace(a,b,n)` ou `x=np.arange(a,b,r)`

Création des ordonnées: la famille des ordonnées des points est créée par la commande:

`y=np.fonction(x)`

Création du graphique: on utilise la commande:

```
plt.plot(x,y)
```

Afficher le graphique: on utilise la commande:

```
plt.show()
```

Exemple 4 Exécuter les instructions suivantes :

```
x=np.arange(0,4*pi,0.01)
y1=sin(x) ; y2=cos(x)
plt.xlim(0,4*pi) ; plt.ylim(-1,1)
plt.plot(x,y1) ; plt.plot(x,y2)
plt.show()
```

Les commandes `plt.xlim(0,4*pi)` et `plt.ylim(-1,1)` fixe les borne des axes des abscisses et des ordonnées.

Exercice 1 :
Écrire un code Python permettant de tracer le graphe de la fonction $p : x \mapsto \ln\left(\pi + \exp\left(3\sin^2(x)\right)\right)$ sur $[-2, 5]$ avec 200 points de tracé.

Exercice 2 :
Représenter le graphe de la fonction arctangente.
Représenter à nouveau le graphe de la fonction arctangente sans utiliser la commande `arctan`