

Fiche cours 7 : la bibliothèque numpy et fonctions (mathématiques)

Points abordés

- Bibliothèque.
 - Fonctions (mathématiques).
-

1 Introduction

De nombreuses fonctions sont très utilisées et par différentes personnes. Pour éviter que chacun n'ait à coder (et recoder) celles-ci, il existe des bibliothèques qui contiennent des fonctions usuelles. Par exemple, il n'est pas nécessaire de programmer soi-même la fonction exponentielle pour pouvoir l'utiliser. Il suffit de faire appel à la bibliothèque `numpy`. Si on veut générer de l'aléatoire on pourra utiliser la bibliothèque `random` etc. Il existe des bibliothèques pour beaucoup d'autres domaines : jeux, traitement des bases de données, calcul symbolique etc.

2 les bibliothèques

2.1 Importer une bibliothèque

Pour pouvoir utiliser une bibliothèque, il faut dans un premier temps l'importer. Par exemple, si on veut utiliser la bibliothèque `numpy`, il suffit d'écrire :

```
import numpy
```

De manière générale, pour importer une bibliothèque portant le nom `bibli`, on saisit

```
import bibli
```

Parfois, saisir le nom d'une bibliothèque devient rapidement fastidieux. On peut alors lui donner un autre nom (un alias). Par exemple, la bibliothèque `bibliothequeFrancoisMitterand` (qui n'existe pas en python) possède un nom un peu long, on pourra l'importer sous cette forme :

```
import bibliothequeFrancoisMitterand as bnf
```

Il est courant d'utiliser l'alias `np` pour `numpy`. On trouve alors généralement en en-tête :

```
import numpy as np
```

2.2 Utiliser une bibliothèque

La bibliothèque `numpy` contient de nombreuses fonctions usuelles. Par exemple, si on veut utiliser la fonction exponentielle, on procède de la manière suivante :

```
import numpy as np    # on importe la bibliothèque numpy sous le nom np

x=np.exp(3)           # la variable x contient maintenant la valeur exp(3).
```

Avoir accès à une bibliothèque ne signifie pas savoir l'utiliser. Heureusement, toute bibliothèque est accompagnée d'une documentation que l'on peut consulter à l'aide de la ponctuation `?`. Par exemple,

```
np.exp?
```

permet de voir une documentation expliquant la fonction `exp` et comment l'utiliser. De plus, il est possible de voir la manière dont est programmée la fonction à l'aide de `??`. Par exemple, en saisissant

```
np.exp??
```

on voit le code source de la fonction `exp`.

Remarque 1. Parfois la documentation peut aussi être obscure... dans ce cas, il y a google les forums etc. En pratique, nous ne la consultons que sporadiquement

3 `numpy` et `matplotlib.pyplot`

La bibliothèque `numpy` sera utilisée lorsque l'on veut utiliser les fonctions numériques classiques. Mais elle ne permet pas de tracer de courbes. Pour cela, on a besoin d'une autre bibliothèque, `matplotlib.pyplot`.

En pratique, il nous suffit de mettre en en-tête :

```
import numpy as np
import matplotlib.pyplot as plt
```

On considère le code source suivant :

```
import numpy as np
import matplotlib.pyplot as plt

L=np.linspace(0,10)
K=[np.exp(x) for x in L]
plt.plot(L,K)
```

On obtient alors le tracé de la fonction exponentielle sur l'intervalle $[0, 10]$. Expliquons les différentes commandes.

1. `np.linspace(0,10)` construit une liste de nombres compris entre 0 et 10 répartis de manière uniforme sur $[0, 10]$.
2. `plt.plot(L,K)` construit le graphe obtenu en joignant $(L[0], K[0])$ à $(L[1], K[1])$, $(L[1], K[1])$ à $(L[2], K[2])$ etc. Pour pouvoir utiliser cette fonction, il faut que les deux listes aient le même nombre d'éléments.