

Python td 3 - fonctions et suites : représentations graphiques et approximations

I. Définir une fonction

I.1) Fonctions usuelles définies dans le package numpy

Nom	Fonction
np.log	logarithme népérien
np.exp	exponentielle
np.sqrt	racine carrée
np.abs	valeur absolue
np.floor	partie entière
np.sin	sinus
np.cos	cosinus
np.tan	tangente

Ces fonctions peuvent s'appliquer à une variable numérique ou directement à une matrice.

I.2) Définir une nouvelle fonction

```
def f(x):  
    return valeur en fonction de x
```

On peut appliquer la fonction ainsi définie à une valeur ou directement à une matrice.

Exemple

```
def f(x):  
    return x**2
```

```
L=np.arange(1,11,1)  
print(f(L))
```

II. Tracé de la courbe d'une fonction

```
import matplotlib.pyplot as plt
```

Puis si x est la matrice des abscisses et si y est la matrice des ordonnées, la commande

```
plt.plot(x,y)  
plt.show() #pour afficher la figure
```

relie les points de coordonnées x et y

Exemple

Courbe de la fonction sinus sur $[-2\pi, 2\pi]$:

```
x=np.arange(-2*np.pi,2*np.pi,0.01) #on discrétise l'intervalle [-2pi,2pi]  
y=np.sin(x)  
plt.plot(x,y)  
plt.show()
```

III. Suites

III.1) Suites explicites du types $u_n = f(n)$

Si la suite est définie à partir d'une fonction $\forall n \in \mathbb{N}, u_n = f(n)$, alors on définit f comme précédemment et en posant $x=np.arange(0,n+1,1)$ et $u=f(x)$, u est la matrice ligne dont les coordonnées sont les $n+1$ premières valeurs $u_0, \dots, u_n$. On peut alors utiliser `plt.plot(x,u,"o")` pour représenter les termes de la suite (le "o" sert à indiquer que l'on veut mettre un point pour les termes de la suite et ne pas les relier entre eux).

Exemple

On considère la suite définie par : $\forall n \in \mathbb{N}, u_n = \frac{n^2}{2^n}$. Le programme suivant permet de tracer les premiers termes de la suite :

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

```
def f(x):  
    return (x**2)/(2**x)
```

```
x=np.arange(0,12,1)  
y=f(x)  
plt.plot(x,y,"o")  
plt.show()
```

Exécuter ce programme. Quelle conjecture peut-on émettre quand à la monotonie de cette suite ?

III.2) Suites récurrentes du types $u_{n+1} = g(u_n)$

Si la suite est définie par récurrence, on utilisera en général une boucle `for`.

On considère une suite définie par $u_0 = a$ et pour tout $n \in \mathbb{N}, u_{n+1} = g(u_n)$.

Première méthode : nous calculons les valeurs de la suite en les effaçant au fur et à mesure (pas de tracé possible), et on affiche le terme u_n .

```
n=int(input("Entrer n"))  
u=a  
for k in range(1,n+1):  
    u=g(u)  
print(u)
```

Deuxième méthode : stocker les valeurs dans une matrice au fur et à mesure

Nous pouvons ensuite représenter graphiquement les premiers termes de la suite.

```
n=int(input('Entrer n: '))  
u=np.zeros([n+1]) # on définit une matrice unidimensionnelle nulle à n+1 cases  
u[0]=a  
for k in range(1,n+1):  
    u[k]=g(u[k-1])  
print(u)  
plt.plot(np.arange(0,n+1,1),u,"o")
```

Exercice 1

On considère la suite définie par $u_0 = 0$ et pour tout $n \in \mathbb{N}, u_{n+1} = \sqrt{2 + u_n}$.

A l'aide de la première méthode, calculer le n -ième terme de cette suite. Puis à l'aide de la deuxième méthode, calculer et afficher les termes $u_0, \dots, u_n$.

Que peut-on conjecturer pour cette suite ? Comment pourrait-on le démontrer ?

IV. Exercices

Exercice 2

Tracés de courbes

- On considère la fonction g définie par : $\forall x \in]0, +\infty[\quad g(x) = \frac{\ln(x)}{x}$.
 - Montrer que g admet un unique point d'inflexion en une abscisse x_0 et calculer x_0 .
 - Tracer la courbe de g sur $[1, 10]$ en utilisant Python et constater graphiquement votre résultat précédent.
- On considère la fonction h définie par : $\forall x \in \mathbb{R} \quad h(x) = x + 1 + e^{-x}$
 - Tracer la courbe de h sur $[-2, 3]$ en utilisant Python.
 - Montrer que la courbe de h admet une asymptote au voisinage de $+\infty$ et préciser une équation de cette droite.
 - Illustrer ce résultat en traçant sur le même graphe la courbe de h et cette droite sur $[-2, 3]$.

Exercice 3

Tracé d'une courbe d'une bijection réciproque

1. La fonction arctangente :

- Tracer la courbe de la fonction tangente sur l'intervalle $\left[-\frac{\pi}{2} + 0.1; \frac{\pi}{2} - 0.1\right]$.
- Tracer la courbe de la fonction arctangente ainsi que la droite d'équation $y = x$ sur le même graphe.

2. Un autre exemple

On note $\forall x \in \mathbb{R} \quad f(x) = \frac{e^x - e^{-x}}{2}$.

- Montrer que f est une bijection de $\mathbb{R}$ dans un intervalle à préciser (cette fonction est la fonction "sinus hyperbolique", notée sh).
- Tracer la courbe de f sur $[-2, 2]$ et la courbe de son application réciproque en Python.

Exercice 4

Illustration d'un développement limité

1. Le développement limité de $x \rightarrow \ln(1+x)$ au voisinage de 0

- Tracer la courbe de $x \rightarrow \ln(1+x)$ sur $[-0.999, 4]$ en utilisant Python.
- Rappeler le développement limité à l'ordre de 2 de $x \rightarrow \ln(1+x)$ au voisinage de 0.
- Tracer la courbe de la partie régulière de ce DL sur la même figure Python qu'au-dessus, toujours sur $[-0.999, 4]$.
Que peut-on dire quand à l'approximation de $\ln(1+x)$ par son DL ?

2. Le développement limité de $x \rightarrow e^x$ au voisinage de 0

- Tracer la courbe de $x \rightarrow e^x$ sur $[-1, 1]$ en utilisant Python.
- Rappeler le développement limité à l'ordre de 2 de $x \rightarrow e^x$ au voisinage de 0.
- Illustrer ce résultat grâce à Python sur $[-1, 1]$. Que peut-on dire quand à l'approximation de e^x par son DL ?

Exercice 5

Suites adjacentes

On considère le programme suivant :

```
import numpy as np
import matplotlib.pyplot as plt
n=30
x=np.arange(1,n+1)
y=np.zeros(n)
a=1
for i in range(n):
    y[i]=a/(i+1)
    a=a*(-1)
z=np.cumsum(y)
plt.axis([0,30,0.35,1.1])
plt.plot(x,z,'*')
plt.show()
```

- Quel est le contenu des variables x , y , z ?
- Recopier et exécuter le programme ci-dessus. Le graphe obtenu suggère que deux suites sont adjacentes : lesquelles ?
-> Ceci avait été prouvé dans le TD du Chapitre 1 ! Il s'agit de montrer que la série est convergente.

Exercice 6

Méthode de dichotomie et suite implicite

On considère la fonction $f : x \mapsto x \exp(x)$.

- Etudier les variations de f sur $\mathbb{R}$.
- Définir en Python la fonction f et tracer la courbe de f sur $[-4, 1.5]$.
- Montrer que pour tout $n \in \mathbb{N}^*$, l'équation $f(x) = n$ admet une unique solution que l'on notera u_n et vérifier que $u_n \in [0, n]$.
- Compléter la fonction suivante, intitulée `def u(n):`, afin qu'elle renvoie une valeur approchée de u_n à 10^{-3} près par la méthode de dichotomie.

```
def u(n):
    a= ...
    b= ...
    while ... :
        c=(a+b)/2
        if .... :
            a=...
        else:
            b=...
    return ...
```

- Le morceau de code suivant permet alors de représenter les 10 premiers termes de la suite (u_n) . Recopier et exécuter ce code.

```
x=np.arange(1,11,1)
y=np.zeros(10)
for k in range(0,10):
    y[k]=u(k+1)
plt.plot(x,y,"*")
plt.show()
```

Que peut-on conjecturer quand à la suite (u_n) ? Démontrer ce résultat en utilisant les propriétés de la fonction W , qui est la bijection réciproque de la fonction f restreinte à $\mathbb{R}_+$.