TP 10 : tracé de courbes, fonction W, TVI

Formulaire

Vous aurez (probablement mais pas obligatoirement) besoin des commandes suivantes :

```
import numpy as np
if ... :
[.... for k in ....]
for k in range(....)
```

```
while .... :
print('message')
abs
exp
```

Tracé de courbes avec matplotlib

Le module `matplotlib` contient de nombreux outils pour tracer des courbes de fonctions. Nous utiliserons une toute petite partie de ce module : la méthode `pyplot`, contenant elle-même la méthode `plot`. Vous inclurez dans tous vos codes les lignes montrées ci-contre.

```
import numpy as np
from math import *
import matplotlib.pyplot as plt
```

Ceci qui nous donnera accès à tous les outils de mathématiques (dans le module `math`), de calcul numérique (dans le module `numpy`) et de tracé de courbes. La syntaxe d'utilisation de `pyplot` est montrée ci-contre. Cette méthode affiche un point du plan par élément de la liste donnée en argument à `plt.plot()`. Les abscisses sont alors automatiquement les positions dans la liste en question.

```
ordo=[exp(k) for k in range(1,30,3)]
plt.plot(ordo)
plt.show()
```

On peut vouloir préciser les abscisses soi-même. Il suffit alors de donner deux arguments à la fonction `plot` : abscisses et ordonnées. Tant que la fenêtre graphique reste ouverte, Python attend une action. Fermez la fenêtre graphique qui a été ouverte par Python puis modifiez votre code de la façon montrée ci-contre, avant de relancer votre code.

```
abscis=[k for k in range(1,30,3)]
ordo=[exp(k) for k in abscis]
plt.plot(abscis,ordo)
plt.show()
```

Vous remarquez qu'il faut deux lignes de code pour afficher le graphique : une utilisation de `plt.plot` suivie d'un appel de `plt.show()`. Cela permet de différer l'affichage afin de faire des modifications au graphique initial généré par `plt.plot`. Par exemple on peut vouloir tracer plusieurs courbes sur un même graphique. Modifiez encore votre code de la façon montré ci-contre.

```
abscis=[0.1+k*0.01 for k in range
    (200)]
ordo=[k**k for k in abscis]
plt.plot(abscis,ordo)
plt.plot(abscis,abscis)
plt.show()
```

Un dernier outil est pratique pour générer facilement des listes de points utilisés comme abscisses : la fonction `np.linspace`. Elle prend trois arguments : réel de départ a, réel d'arrivée b et nombre de points n. Elle renvoie un `array` numpy contenant exactement n points dont le premier est a et le dernier est b, répartis de manière uniforme entre a et b. Ils seront donc espacés de $(b-a)/(n-1)$.

```
x=np.linspace(0.001,100,10000)
y=[log(k) for k in x]
plt.plot(x,y)
plt.show()
```

Exercice 1.

1. Créez une fonction Python qui à x associe $x \times \mathrm{e}^x$.
   Testez votre fonction en vérifiant qu'elle renvoie le réel e lorsqu'on calcule l'image de 1.
2. Tracer le graphe de votre fonction pour des abscisses variant entre -8 et 1. Vous afficherez exactement 1000 points : utilisez `linspace` obligatoirement.

Application numérique du thm de la bijection : tracé de la courbe de W

La fonction $x \mapsto x\mathrm{e}^x$ est continue et dérivable sur $\mathbb{R}$, de dérivée $x \mapsto \mathrm{e}^x(1+x)$, strictement positive sur l'intervalle $]-1;+\infty[$. Elle réalise donc, d'après le théorème de la bijection, une bijection de $[-1;+\infty[$ dans l'image de cet intervalle. Par produit de limites on a $\lim\limits_{x\to+\infty} x\mathrm{e}^x = +\infty$ et donc l'intervalle image est $[-1/\mathrm{e};+\infty[$.

On va essayer de calculer la bijection réciproque de cette fonction à l'aide de l'ordinateur.
On appelle cette bijection réciproque la **fonction W de Lambert**.

Exercice 2 (Calcul de $W(1)$)**.**

1. Écrivez une fonction `tvi` d'arguments `f`,`a`,`b`,`eps` qui renvoie (par la méthode de Dichotomie vue en classe lors de la démonstration du théorème des valeurs intermédiaires) une valeur approchée à `eps` près d'une solution de l'équation $f(x) = 0$ située entre a et b (qu'on suppose existante).

   Indications :
   — *On utilisera deux variables u et v qui contiendront les valeurs successives des deux bornes de l'intervalle de recherche, initialisées à a et b, ainsi que $c = (u+v)/2$.*
   — *La recherche devra continuer tant que $|f(c)| >$ `eps`.*
   — *Il faudra distinguer les cas où $f(c)$ et $f(u)$ ont le même signe et les cas où $f(c)$ et $f(u)$ sont de signes opposés.*
2. Testez votre fonction sur la fonction $x \mapsto x\mathrm{e}^x - 1$. On pourra démarrer la recherche sur l'intervalle $[0;1]$ (voyez-vous pourquoi ?), et on vérifiera que la solution est environ 0.5671386718 (toutes les décimales montrées sont exactes).

Exercice 3 (Calcul de $W(y)$ pour y quelconque)**.**

1. Modifiez votre fonction `tvi` en ajoutant un argument `y` à sa liste d'arguments, de sorte que la fonction renvoie la solution de $f(x) = y$ au lieu de la solution de $f(x) = 0$.
2. Vérifiez qu'on obtient $W(7) \approx 1.52434520$ (toutes les décimales montrées sont exactes). sur [1,2]

Exercice 4 (Tracé de la courbe de W)**.**

1. Définissez une fonction `W` d'arguments `x` et `eps` qui renvoie la valeur de $W(x)$ calculée à l'aide de la fonction `tvi` à `eps` près.
2. Tracez sur un même graphique les courbes des fonctions $x \mapsto x\mathrm{e}^x$, $x \mapsto W(x)$ ainsi que $x \mapsto x$, sur l'intervalle $[0;1]$ avec un nombre suffisant de points et une précision suffisante pour que les courbes paraissent lisses.
3. Observez-vous la symétrie des deux courbes des deux fonctions bijections réciproques l'une de l'autre par rapport à la symétrie d'axe la droite d'équation $y = x$?
   Proposez une explication à ce phénomène.