TD centrale 2 analyse et géométrie

I Fiche AN et PLOT

I.1 Résolution d'équations

Exercice 1 *Soit* $f : x \mapsto x^3 - 3x - 1$

1. *Représenter graphiquement* f *sur l'intervalle* $[-3, 3]$.
2. *Donner des valeurs approchées des solutions de l'équation* $f(x) = 0$.

Exercice 2 *Utilisation de la fonction* `root`

1. *Donner une valeur approchée d'une solution du système* $\left\{ \begin{array}{l} x^2 + y^2 = \\ x^4 + y^4 = \end{array} \right.$
2. *Déterminer une matrice* U *orthogonale et une matrice* S *symétrique vérifiant* $SU = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$.

Solution de l'exercice:

```
##Q1 fsolve
import scipy.optimize as resol
def f(v) :
    return v[0]**2 + v[1]**2 - 5, v[0]**4 + v[1]**4 - 20
sol = resol.root(f, [1,1])# un False dans le resultat indique que
# le point de depart [0,0] de l'algorithme n'est pas bon
sol = resol.root(f, [1,1])
print(sol.x)
print(f(sol.x)) # v\U{e9}rification: \U{e7}a fait presque (0,0)
```

I.2 Intégration, équations différentielles, séries entières, graphiques

Exercice 3 *(centrale 2 extrait): On pose* $f_p(x) = \int_0^1 \sqrt{x^2 + t^p}\,dt$.

1. *Représenter graphiquement* f_2 .
2. *Représenter graphiquement les fonctions* f_i *sur un même graphique pour* $i \in [|1, 5|]$.

Solution de l'exercice:

```
#Q1
def g(x):
    def f(t):
        return np.sqrt(x**2+t**2)
    return integr.quad(f,0,1)
X = np.arange(0,2, 0.01)
Y=[g(x) for x in X]
plt.plot(X,Y)
plt.show()

#Q2
X = np.arange(0,2, 0.01)
for i in range(1,6):
    def g(x):
        def f(t):
            return np.sqrt(x**2+t**i)
        return integr.quad(f,0,1)
    Y=[g(x) for x in X]
    plt.plot(X,Y)
plt.show()
```

Exercice 4 *Tracer la courbe représentant la solution de* $\begin{cases} y' + e^x y = 1 \\ y(0) = 1 \end{cases}$ *sur l'intervalle* $[0,1]$.

Solution de l'exercice:

```
import math
def f(x, t) :
    return 1-math.exp(t)*x
T = np.arange(0, 1.01, 0.01)# condition de Cauchy en T[0]=0
X = integr.odeint(f, 1, T)# deuxieme argument 1: valeur de la solution en 0
plt.plot(T,X)
plt.show()
```

Exercice 5 *(centrale 2 extrait): On considère l'équation* $(1-x)\,y'' = y$ *sur* $]-\infty, 1[$.

1. *(maths) Montrer qu'il existe une et une seule solution* f *de cette équation vérifiant* $f(0) = 0$ *et* $f'(0) = 1$.
2. *(python) Représenter la fonction* f *sur l'intervalle* $[-2; 0,95]$.
3. *Soit* (a_n) *la suite définie par* $a_0 = 0$, $a_1 = 1$ *et* $\forall n \in \mathbb{N} \setminus \{0,1\}$, $a_n = \dfrac{n-2}{n} a_{n-1} + \dfrac{1}{n(n-1)} a_{n-2}$.
   (a) *(python) Représenter graphiquement* a_n *pour* $n \in [0, 100]$.
   (b) *Montrer que le rayon de convergence de* $\sum a_n x^n$ *est supérieur ou égal à 1.*
   (c) *(python) Représenter graphiquement la fonction* $x \mapsto \sum_{n=0}^{100} a_k x^k$*.sur* $[-2; 0,95]$. *Que constate-on?*
   (d) *Démontrer le résultat.*

Solution de l'exercice:

On a $(E) = (1-x)\,y'' = y$ sur $]1, +\infty[\Leftrightarrow y'' - \dfrac{1}{(1-x)} y = 0$. D'après le théorème de Cauchy, 'il existe une et une seule solution f de cette équation vérifiant $f(0) = 0$ et $f'(0) = 1$.

```
""" Q2: representation de la solution f"""
def f(X,t): # fonction de l'equa diff X'=f(X,t)
    return [X[1],1/(1-t)*X[0]]
from scipy.integrate import odeint
import numpy as np
import matplotlib.pyplot as plt
# sur [0,0.95]
liste_t=np.linspace(0,0.95,100)
sol=odeint(f,[0,1],liste_t) # X[0]=[0,1]
plt.plot(liste_t,sol[:,0]) # sol array 100,2: premiere col: valeurs de y, deuxieme: valeurs de y'
# sur [-2,0]
liste_t=np.linspace(0,-2,100) # liste des temps de 0 a -2 (temps parcouru en sens inverse)
sol=odeint(f,[0,1],liste_t) # X[0]=[0,1]
plt.plot(liste_t,sol[:,0])
plt.show()

"""Q3a: representation des an"""
liste_n=[n for n in range(101)]
liste_a=[0,1]
a,b=0,1
for n in range(2,101):
    c=(n-2)/n*b+a/(n*(n-1))
    liste_a.append(c)
    a,b=b,c
plt.plot(liste_n,liste_a) # on constate 0<=an<=1
```

Montrons par récurrence sur n que $0 \leq a_n \leq 1$. C'est vrai pour $n = 0, 1$.

Soit $n \geq 2$. Supposons $0 \leq a_{n-2} \leq 1$ et $0 \leq a_{n-1} \leq 1$. On a $a_n = \dfrac{n-2}{n} a_{n-1} + \dfrac{1}{n(n-1)} a_{n-2} \leq \dfrac{n-2}{n} + \dfrac{1}{n(n-1)} \leq \dfrac{n-2}{n} + \dfrac{1}{n} < 1$ car $n - 1 \geq 1$ et $a_n \geq 0$ donc $0 \leq a_n \leq 1$. Posons $b_n = 1$. La série entière $\sum b_n x^n$ est de rayon 1 et $0 \leq a_n \leq b_n$ donc le le rayon de convergence de $\sum a_n x^n$ est supérieur ou égal à 1.

```
""" representation de S100(x)
def S100(x):
    S,p=0,1
    for i in range(101):
        S=S+liste_a[i]*p
        p=p*x
    return S
liste_S=[S100(t) for t in liste_t]
plt.plot(liste_t,liste_S)
plt.plot(liste_t,sol[:,0])
```

Pour tout $n \geq 0$, $n(n-1) a_n - (n-2)(n-1) a_{n-1} + a_{n-2} = 0$ donc si $|x| < 1$, $\sum\limits_{n=2}^{+\infty} n(n-1) a_n x^{n-2} - \sum\limits_{n=2}^{+\infty} (n-2)(n-1) a_{n-1} x^{n-2} + \sum\limits_{n=2}^{+\infty} a_{n-2} x^{n-2} = 0$ (toutes les séries ont même rayon que $\sum a_n x^n$). Or $\sum\limits_{n=2}^{+\infty} n(n-1) a_n x^{n-1} = x \sum\limits_{n=2}^{+\infty} n(n-1) a_n x^{n-2} = x f''(x)$; $\sum\limits_{n=2}^{+\infty} (n-2)(n-1) a_{n-1} x^{n-2} = x \sum\limits_{n=2}^{+\infty} (n-2)(n-1) a_{n-1} x^{n-3} = x \sum\limits_{n=3}^{+\infty} n(n-1) a_n x^{n-2} = x f'(x) - a_2 = x f'(x)$ car $a_2 = 0$.et $\sum\limits_{n=2}^{+\infty} a_{n-2} x^{n-2} = f(x)$ donc $S(x)$ vérifie l'équation (E) et comme $S(0) = a_0 = 0$ et $S'(0) = a_1 = 1$, on en déduit que $S = f$.

II Tracé de courbes et surfaces

Exercice 6 *(centrale 2 extrait): Soit $M \in \mathcal{M}_n(\mathbb{R})$. On définit $\varphi_M : \mathbb{R}^n \to \mathbb{R}$ définie par $\varphi_M(X) = {}^tXMX$. On suppose que $M = \begin{pmatrix} -2 & 4 \\ -1 & 2 \end{pmatrix}$ et $X = \begin{pmatrix} x \\ y \end{pmatrix}$. Tracer la surface d'équation $z = \varphi_M(x, y)$ pour $(x, y) \in [-2, 2]^2$.*

Solution de l'exercice:

On a $\varphi_M(x, y) = {}^tXMX = (x, y) \begin{pmatrix} -2x + 4y \\ -x + 2y \end{pmatrix} = x(-2x + 4y) + y(-x + 2y) = (2x + y)(-x + 2y) = -2x^2 + 3xy + 2y^2$.

```
import matplotlib.pyplot as plt
import numpy as np
from mpl_toolkits.mplot3d import Axes3D
ax=Axes3D(plt.figure())
def f(x,y):
    return (2*x+y)*(-x+2*y)
X=np.arange(-2,2,0.02)
Y=np.arange(-2,2,0.02)
X,Y=np.meshgrid(X,Y)
Z=f(X,Y)
ax.plot_surface(X,Y,Z)
plt.show()
# On a eu un probleme mardi avec ce script (a verifier)
```

Exercice 7 *Tracer la courbe paramétrée plane définie par* $\begin{cases} x(t) = \cos(t) \\ y(t) = \sin(2t) \end{cases}$

Exercice 8 *Tracer la courbe du plan d'équation $x^2 + y^4 = \lambda$ pour $\lambda \in \{1, 4, 10\}$.*

Solution de l'exercice:

```
def f(x,y) :
    return x**2 + y**4
X = np.arange(-10, 10, 0.01)
Y = np.arange(-10, 10, 0.01)
X, Y = np.meshgrid(X, Y)
Z = f(X, Y)
plt.axis('equal')
plt.contour(X, Y, Z, [1,4,10])
```

III Enoncé complet

Exercice 9 *On pose, pour tout* $n \in \mathbb{N}^*, A_n : x \mapsto \sum_{k=1}^{n} \frac{x^k}{k}$.

1. *Montrer que pour tout* $y \in \mathbb{R}^+$*, il existe un unique* $x \in \mathbb{R}^+$ *tel que* $A_n(x) = y$*. On note* $f_n(y)$ *cette unique solution.*
2. *Ecrire une fonction Python qui prend en argument* n *et* x *et qui renvoie* $A_n(x)$*.*
3. *Utiliser la fonction fsolve pour obtenir* $f_n(y)$*.*
4. *Tracer, pour différentes valeurs de* y*, les valeurs* $(f_n(y))_{n\leq 100}$*.*
5. *Montrer que la suite* $(f_n(y))_n$ *converge.*

Solution :

1 La fonction A_n est un polynôme. Elle est donc continue sur $\mathbb{R}^+$, et dérivable sur $\mathbb{R}^+$:

$$\forall x \geq 0, A_n'(x) = \sum_{k=1}^{n} x^{k-1} = \sum_{k=0}^{n-1} x^k \geq 0$$

Donc A_n est strictement croissante sur $\mathbb{R}^+$.
Comme de plus $A_n(0) = 0$ et $A_n \underset{+\infty}{\longrightarrow} +\infty$, il vient que A_n réalise une bijection de $\mathbb{R}^+$ dans $\mathbb{R}^+$.
Et donc pour tout $y \in \mathbb{R}^+$il existe un unique $x \in \mathbb{R}^+$tel que $A_n(x) = y$.
2

```
def \QTR{rm}{A}(\QTR{rm}{n},\QTR{rm}{x}) :
    s=  0
    puiss $=1$.
    for $k$ in range ( 1,n+1 ):
        puiss *= x
        s += puiss / k
    return s
```

3

```
from scipy.optimize import fsolve
def f(n,y):
return fsolve( lambda x : A(n,x)-y , 1)[0]
```

Remarque : si vous vous demandez pourquoi j'ai pris 1 comme second paramà¨tre de fsolve au lieu du naturel 0, la réponse vous attend à la question suivante.

```
import matplotlib.pyplot as plt
Y = [1,2,3,4]
couleurs = ['red','green','blue','cyan']
les_n = [n for n in range(1, 101)]
for k in range(len(Y)):
    y = Y[k]
    coul = couleurs[k]
    les_f = [f(n,y) for n in les_n]
    plt.plot(les_n,les_f,'o',color=coul,label='y='+str(y))
plt.legend()
plt.show()
```

On dirait bien que la suite $(f_n(y))_n$ est décroissante et convergente vers une limite $\ell(y)$, et que $y \mapsto \ell(y)$ est croissante.
Remarque : j'avais dans un premier temps pris 0 comme second paramètre pour l'utilisation de fsolve et j'avais constaté un problème j'ai changé le point de départ de l'algorithme de résolution en 1, avec un résultat probant.

5 Fixons un $y > 0$ et notons pour alléger la rédaction $u_n = f_n(y)$. Trà¨s classiquement :

$$\begin{aligned} A_n(u_{n+1}) &= A_{n+1}(u_{n+1}) - \frac{u_{n+1}^{n+1}}{n+1} \\ &\leq A_{n+1}(u_{n+1}) \\ &\leq y \\ &\leq A_n(u_n) \end{aligned}$$

Et puisque la fonction A_n est croissante, il vient que $u_{n+1} \leq u_n$.
Comme d'autre part on a $u_n \geq 0$, la suite (u_n) est décroissante et minorée. Elle est donc convergente.
6 C'est un peu étonnant qu'on ne nous demande pas de déterminer la limite de $(f_n(y))$. Je soupà§onne que la question venait Â«en directÂ» lors de l'oral.
Commençons par travailler un peu au corps la fonction $A_n(x)$:

$$\begin{aligned}
A_n(x) &= \int_0^x \sum_{k=0}^{n-1} t^k \, \mathrm{d}t \\
&= \int_0^x \frac{1-t^n}{1-t} \, \mathrm{d}t \\
&= -\ln(1-x) - \int_0^x \frac{t^n}{1-t} \, \mathrm{d}t
\end{aligned}$$

En particulier :

$$y = A_n(u_n) = -\ln(1-u_n) - \int_0^{u_n} \frac{t^n}{1-t} \, \mathrm{d}t$$

Notons ℓ la limite de la suite (u_n) (avec les mà^{a}mes notations qu'à la question précédente). Supposons que $\ell \geq 1$. Alors on a :

$$\forall n \in \mathbb{N}, y = \sum_{k=1}^{n} \frac{u_n^k}{k} \geq \sum_{k=1}^{n} \frac{l^k}{k} \geq \sum_{k=1}^{n} \frac{1}{k}$$

Ce qui est impossible du fait de la divergence de la série harmonique.
Donc $\ell < 1$.
On peut alors majorer l'intégrale qui nous pourrit les calculs :

$$0 \leq \int_0^{u_n} \frac{t^n}{1-t} \, \mathrm{d}t \leq \int_0^{u_n} \frac{u_n^n}{1-u_n} \, \mathrm{d}t = \frac{u_n^{n+1}}{1-u_n} \longrightarrow 0$$

(car à partir d'un certain rang on a $u_n \leq \frac{\ell+1}{2} < 1$. On voit ici pourquoi j'ai voulu montrer que $\ell < 1$)
On a donc :

$$-\ln(1-u_n) \longrightarrow y$$

Ou encore :

$$u_n \longrightarrow 1 - e^{-y}$$