TP7

Systèmes différentiels

Compétences attendues

- Illustrer les concepts vus dans le cours de maths.
- Aucune connaissance n'est exigible au concours.

On s'intéresse au système différentiel $X' = AX$ avec A appartenant à $\mathcal{M}_3(\mathbb{R})$. On rappelle qu'une solution de ce système est donnée par trois fonctions réelles x, y, z.

On souhaite tracer plusieurs trajectoires de solutions pour observer leur comportement vis à vis des conditions initiales et des états d'équilibres du système.

Cependant les trajectoires évoluant dans un espace de dimension 3 ($\mathbb{R}^3$), il n'est pas aisé de les représenter. Ainsi nous choisirons de les représenter en traçant dans un même repère les trois graphes des fonctions x, y et z.

7.1 Cas où A possède 3 valeurs propres strictement négatives

On pose $A = \begin{pmatrix} -1 & 1 & -1 \\ -4 & -5 & 4 \\ -2 & -1 & 0 \end{pmatrix}$.

1. Recopier le script Python suivant et l'exécuter :

```python
import numpy as np
import numpy.linalg as al
A = np.array([[-1,1,-1],[-4,-5,4],[-2,-1,0]])
L = al.eig(A)
print(L[0])
```

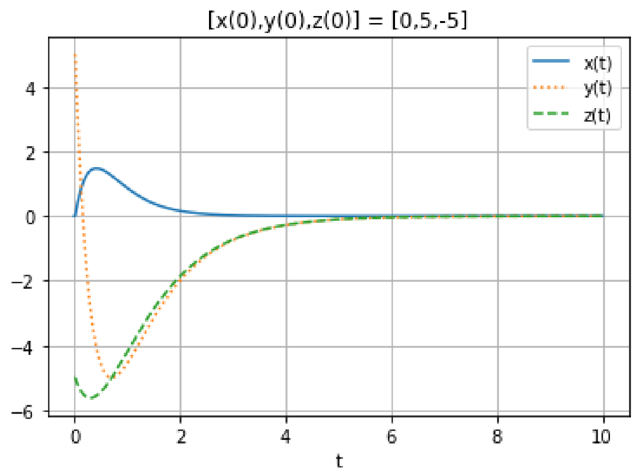
   Que peut-on en déduire ?

2. On note $X(t) = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix}$.

   Pour représenter les graphes de chacune des fonctions x, y, z, on utilise le script suivant :

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import solve_ivp

# Definition du systeme
def systeme(t, Y):
    x = Y[0]
    y = Y[1]
    z = Y[2]

    dx_dt = -x+y-z
    dy_dt = -4*x-5*y+4*z
    dz_dt = -2*x-y

    return [dx_dt, dy_dt, dz_dt]

# Conditions initiales
x0 = 0
y0 = 5
z0 = -5

# Creation de la solution X
solution = solve_ivp(systeme, [0, 10], [x0, y0, z0], method='RK45', args=(), max_step=0.01)

# Recuperation des resultats
a = solution.y[0]
b = solution.y[1]
c = solution.y[2]

# taille de la solution
n=len(a)

# Intervalle de temps pour le trace
t = np.linspace(0,10,n)

# Configuration du trace
plt.plot(t,a,'-',label='x(t)')
plt.plot(t,b,':', label='y(t)')
plt.plot(t,c,'--', label='z(t)')

plt.title(f'[x(0),y(0),z(0)] = [{x0},{y0},{z0}]')
plt.legend(loc='best')
plt.xlabel('t')

plt.grid()
plt.show()
```

Ce qui donne le graphique suivant :



3. Faire varier les conditions initiales. Que remarque-t-on ?

7.2 Cas où A possède 1 valeur propre strictement positive et 2 autres strictement négatives

On pose $A = \begin{pmatrix} 0 & 1 & -1 \\ -6 & -5 & 6 \\ -4 & -2 & 3 \end{pmatrix}$.

1. Recopier le script suivant et l'exécuter :

```python
import numpy as np
import numpy.linalg as al
A = np.array([[0,1,-1],[-6,-5,6],[-4,-2,3]])
I = np.eye(3)
print(al.matrix_rank(A-I))
print(al.matrix_rank(A+I))
print(al.matrix_rank(A+2*I))
```

   Que peut-on en déduire ?

2. Reprendre le programme de la partie précédente en :
   • mettant à jour la matrice A
   • en mettant à jour l'intervalle de temps de $[0,10]$ à $[0,2]$.
   Faire varier les conditions initiales puis décrire le comportement lorsque $t+\infty$ des fonctions x, y et z.
   On testera en particulier les conditions initiales $[1,0,1]$, $[-1,2,0]$ et $[0,1,2]$.

3. On souhaite comprendre cette observation via les résultats théoriques du cours et on admet que :
   • $U_1 = \begin{pmatrix} -1 \\ 2 \\ -1 \end{pmatrix}$ est un vecteur propre de A associé à $\lambda = 1$
   • $U_2 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$ est un vecteur propre de A associé à $\lambda = -1$
   • $U_3 = \begin{pmatrix} 1 \\ -2 \\ 0 \end{pmatrix}$ est un vecteur propre de A associé à $\lambda = -2$

   (a) Montrer que la forme générale des solutions du système $X' = AX$ peut s'écrire sous la forme :
   $$X(t) = \alpha e^t U_1 + \beta e^{-t} U_2 + \gamma e^{-2t} U_2.$$
   On passera par le système réduit puis on reviendra au système initial en posant $X = PY$ où P est la matrice de passage de la base canonique de $\mathcal{M}_{2,1}(\mathbb{R})$ à la base de vecteurs propre de A, puis enfin on écrira la solution générale sous la forme proposée.

   (b) Expliciter $x(t)$, $y(t)$ et $z(t)$ puis vérifier l'observation faite précédemment.

   Voici un exemple de trajectoires :