Python 6
Observation des trajectoires de systèmes différentiels

L'objectif de cette fiche est l'observation de trajectoires de systèmes différentiels linéaires à coefficients constants de **taille 2**. Soit $A \in \mathcal{M}_2(\mathbb{R})$ et $X : \mathbb{R} \to \mathcal{M}_{2,1}(\mathbb{R})$ une application $\mathscr{C}^1$ sur $\mathbb{R}$. On note S le système différentiel $X' = AX$ et $X = \begin{pmatrix} x \\ y \end{pmatrix}$. Rappels sur les trajectoires de S :

1. On considère la matrice $A = \begin{pmatrix} -1 & 0 \\ 0 & -2 \end{pmatrix}$.

   1.a. Résoudre le système $X' = AX$.

   1.b. On impose la condition $X(0) = (1, 0)$. Donner alors l'unique solution convenant ainsi que la trajectoire associée à cette unique solution. Représenter cette trajectoire dans le plan. Représenter également l'allure de chacune des deux composantes de la solution.

   > ☞ **Rappel...**
   > C'est le théorème de Cauchy-Lipschitz qui garantit alors l'existence et l'unicité de la solution.

   1.c. Mêmes questions en imposant $X(-1) = (1, 0)$.

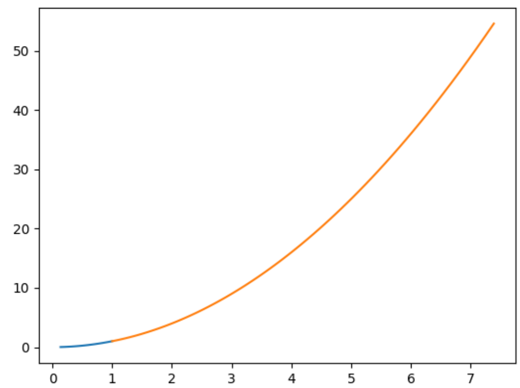
> ✘ **Attention !**
> Une même trajectoire peut être commune à deux solutions différentes... On sait en effet que deux trajectoires sont soit d'intersection vide, soit confondues. Dans les deux exemples choisies, les trajectoires ont toutes deux le point de coordonnées $(1, 0)$ en commun, elles sont donc confondues.

1.d. Mêmes questions en imposant $X(0) = (1, 1)$.

1.e. On considère le programme suivant :

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

def systdiff(X,t,a,b,c,d):
    x,y=X[0],X[1] #ou x,y=X
    dXdt=[a*x+b*y,c*x+d*y]
    return dXdt

a,b=-1,0
c,d=0,-2
X0=[1,1]

t=np.linspace(0,2,100)
sol=odeint(systdiff,X0,t,args=(a,b,c,d))
plt.plot(sol[:,0],sol[:,1])

t=np.linspace(0,-2,100)
sol=odeint(systdiff,X0,t,args=(a,b,c,d))
plt.plot(sol[:,0],sol[:,1])
plt.show()
```

qui permet d'obtenir le graphique :

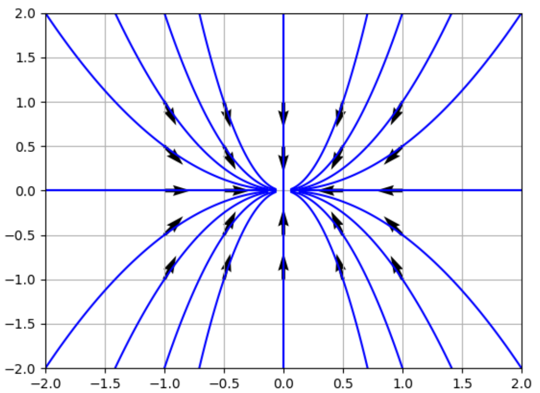


Que représente ce graphique ?

1.f. On considère le programme suivant :

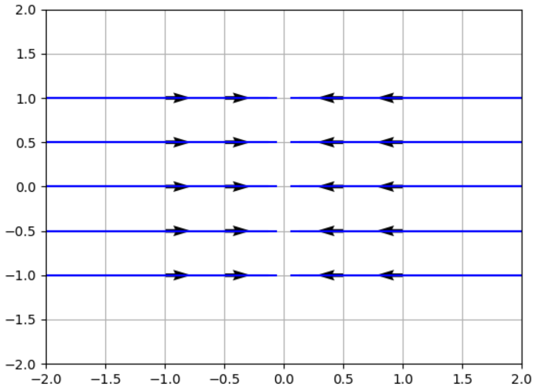
```python
import matplotlib.pyplot as plt
import numpy as np
from scipy.integrate import odeint

def systdiff(X,t,a,b,c,d):
    x,y=X[0],X[1] #ou x,y=X
    dXdt=[a*x+b*y,c*x+d*y]
    return dXdt

a,b=-1,0
c,d=0,-2
t=np.linspace(0,2,100)
for x0 in np.linspace(-1,1,5):
    for y0 in np.linspace(-1,1,5):
        X0=[x0,y0]
        sol=odeint(systdiff,X0,t,args=(a,b,c,d))
        plt.plot(sol[:,0],sol[:,1],'b')

t=np.linspace(0,-2,100)
for x0 in np.linspace(-1,1,5):
    for y0 in np.linspace(-1,1,5):
        X0=[x0,y0]
        sol=odeint(systdiff,X0,t,args=(a,b,c,d))
        plt.plot(sol[:,0],sol[:,1],'b')
        A=np.array([[a,b],[c,d]])
        Y=np.dot(A,X0)
        plt.quiver(X0[0],X0[1],Y[0],Y[1])

plt.axis([-2,2,-2,2])
plt.grid()
plt.show()
```

qui permet d'obtenir le graphique :



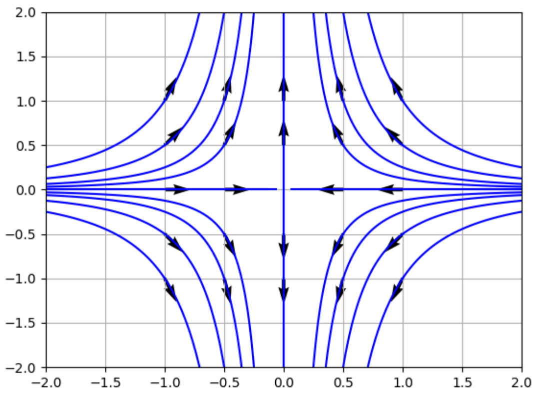
Que représente ce graphique ?

2. Dans le cas où $A = \begin{pmatrix} -1 & 0 \\ 0 & 0 \end{pmatrix}$, on obtient le graphique suivant :



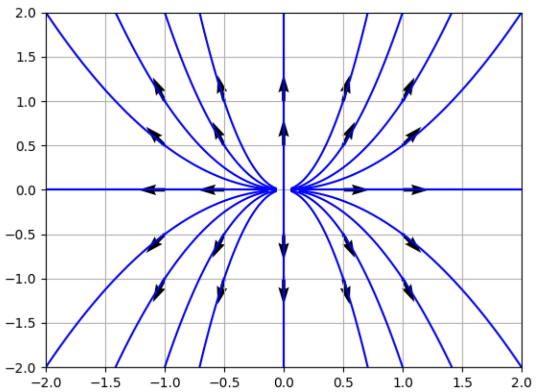
Interpréter.

3. Dans le cas où $A = \begin{pmatrix} -1 & 0 \\ 0 & 2 \end{pmatrix}$, on obtient le graphique suivant :



Interpréter.

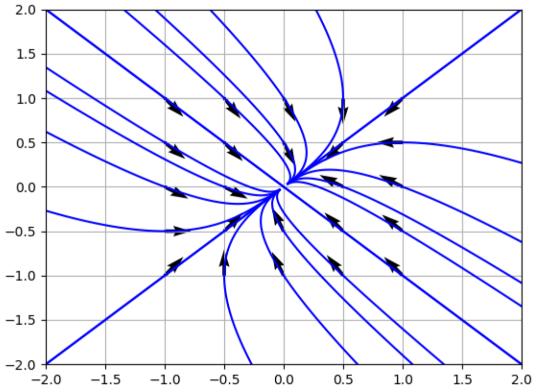
4. Dans le cas où $A = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$, on obtient le graphique suivant :



Interpréter.

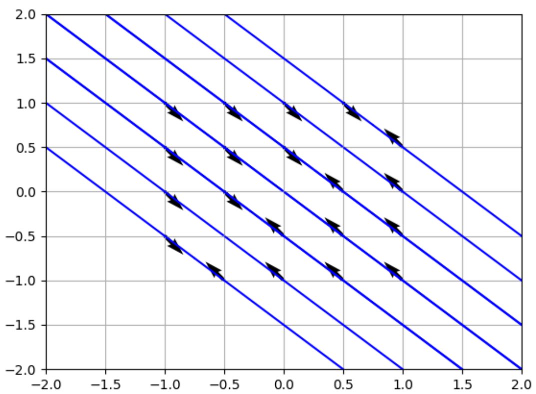
5. Dessiner l'allure du graphique obtenu dans le cas où $A = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$.

6. Dans le cas où $A = \begin{pmatrix} -2 & 1 \\ 1 & -2 \end{pmatrix}$, on obtient le graphique suivant :



Interpréter.

7. Pour une certaine matrice, on obtient le graphique suivant :



Interpréter.