**(a)** L'équation

$$(1+x^2)y''(x)+xy'(x)-\frac{1}{4}y(x)=0 \qquad (E)$$

est une équation différentielle linéaire et homogène du second ordre. Les coefficients sont continus sur $\mathbb{R}$ et le coefficient de $y''(x)$ ne s'annule jamais. On peut donc appliquer le Théorème de Cauchy-Lipschitz : quels que soient les réels $x_0$, $a$ et $b$, il existe une, et une seule, solution $f$ de (E) de classe $\mathscr{C}^2$ sur $\mathbb{R}$ telle que

$$f(x_0)=a \quad \text{et} \quad f'(x_0)=b.$$

En particulier, il existe une, et une seule, solution sur $]-1,1[$ telle que $y(0)=0$ et $y'(0)=\sqrt{2}$.

**(b)** La résolution numérique d'une équation différentielle demande qu'on l'écrive sous forme résoluble. L'équation (E) devient alors

$$\forall\, x\in\mathbb{R},\quad Y'(x)=\begin{pmatrix} y'(x) \\ \frac{1}{1+x^2}\cdot(\frac{1}{4}y(x)-xy'(x)) \end{pmatrix}=f\big(Y(x),x\big).$$

```
import numpy as np
from scipy.integrate import odeint

def f(Y, x):
   y, y_prime = Y[0], Y[1]
   y_seconde = (y/4-x*y_prime)/(1+x**2)
   return np.array([y_prime, y_seconde])
```

❧ La résolution numérique ne demande alors plus que de choisir un intervalle de temps $(x_i)_{0\leqslant i<n}$ et une condition initiale $(y_0,v_0)$.

```
X = np.arange(0, 1.01, 0.01)
CI = np.array([0, np.sqrt(2)])
Y = odeint(f, CI, X)
positions, vitesses = Y[:,0], Y[:,1]
plt.plot(X, positions)
```

Le tableau `Y` produit par `odeint` est de la forme $(Y_{i,j})_{0\leqslant i<n,0\leqslant j<2}$ où $Y_{i,0}\approx y(x_i)$ et $Y_{i,1}\approx y'(x_i)$.

❧ Mais l'approximation de la solution ainsi obtenue n'est définie que sur l'intervalle $[0,1]$ et pas sur $[-1,1]$ !

Une première possibilité consiste à calculer l'équation vérifiée par $z=[x\mapsto y(-x)]$. Comme $z(x)=y(-x)$, $z'(x)=-y'(-x)$ et $z''(x)=y''(-x)$ et que $y$ est une solution de (E) sur l'intervalle $[x_1,x_2]$, alors $z$ est une solution de (E) sur l'intervalle $[-x_2,-x_1]$ :

$$\begin{aligned}(1+x^2)z''(x)+xz'(x)-\frac{1}{4}z(x)&\\ &=(1+x^2)y''(-x)-xy'(-x)-\frac{1}{4}y(-x)\\ &=[1+(-x)^2]y''(-x)+(-x)y'(-x)-\frac{1}{4}y(-x)\\ &=0.\end{aligned}$$

Par conséquent, $y$ est solution sur $[-1,0]$ si, et seulement si, $z$ est solution sur $[0,1]$ avec la condition initiale

$$\big(z(0),z'(0)\big)=\big(y(0),-y'(0)\big)=(0,-\sqrt{2})=-(0,\sqrt{2}).$$

```
Z = odeint(f, -CI, X) # on résout sur [0,1]
plt.plot(-X, Z[:,0]) # on trace sur [-1,0]
```

REMARQUE.— En général, les fonctions $y$ et $z$ *ne vérifient pas* la même équation différentielle...

❧ On peut procéder de manière un peu différente en tirant parti de la structure particulière de l'équation différentielle.

Considérons la solution $y$ de (E) définie sur $\mathbb{R}$ associée à la condition initiale $(a, b) = (0, b)$ et la fonction $z = [x \mapsto -y(-x)]$. On vérifie (comme plus haut) que $z$ est une solution de (E) associée à la même condition initiale : comme le Théorème de Cauchy-Lipschitz s'applique, on en déduit que

$$\forall\, x \in \mathbb{R}, \quad z(x) = -y(-x) = y(x).$$

Autrement dit, la solution $y$ est impaire et son graphe admet donc l'origine pour centre de symétrie et il est donc inutile d'invoquer deux fois la fonction `odeint` !

```
plt.figure()
plt.plot(X, positions)
plt.plot(-X, -positions)
```