

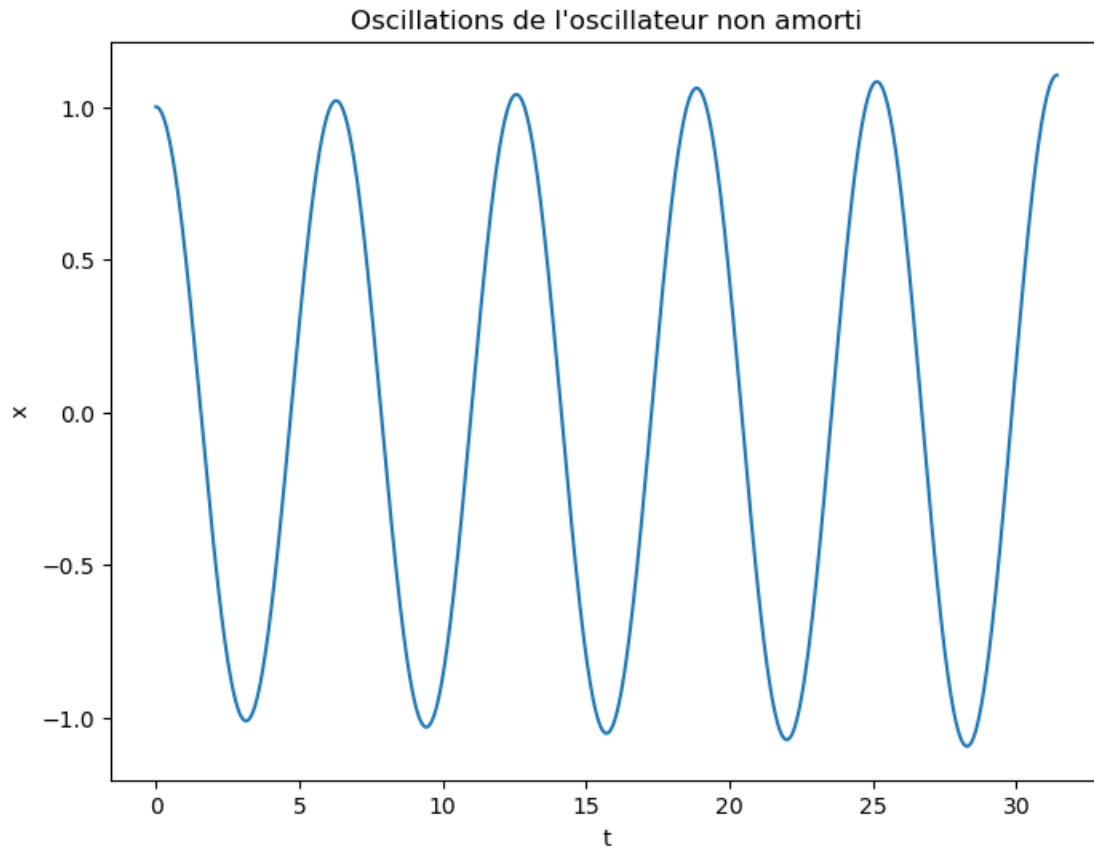
Oscillateur harmonique

```
[1]: import numpy as np
import matplotlib.pyplot as plt
```

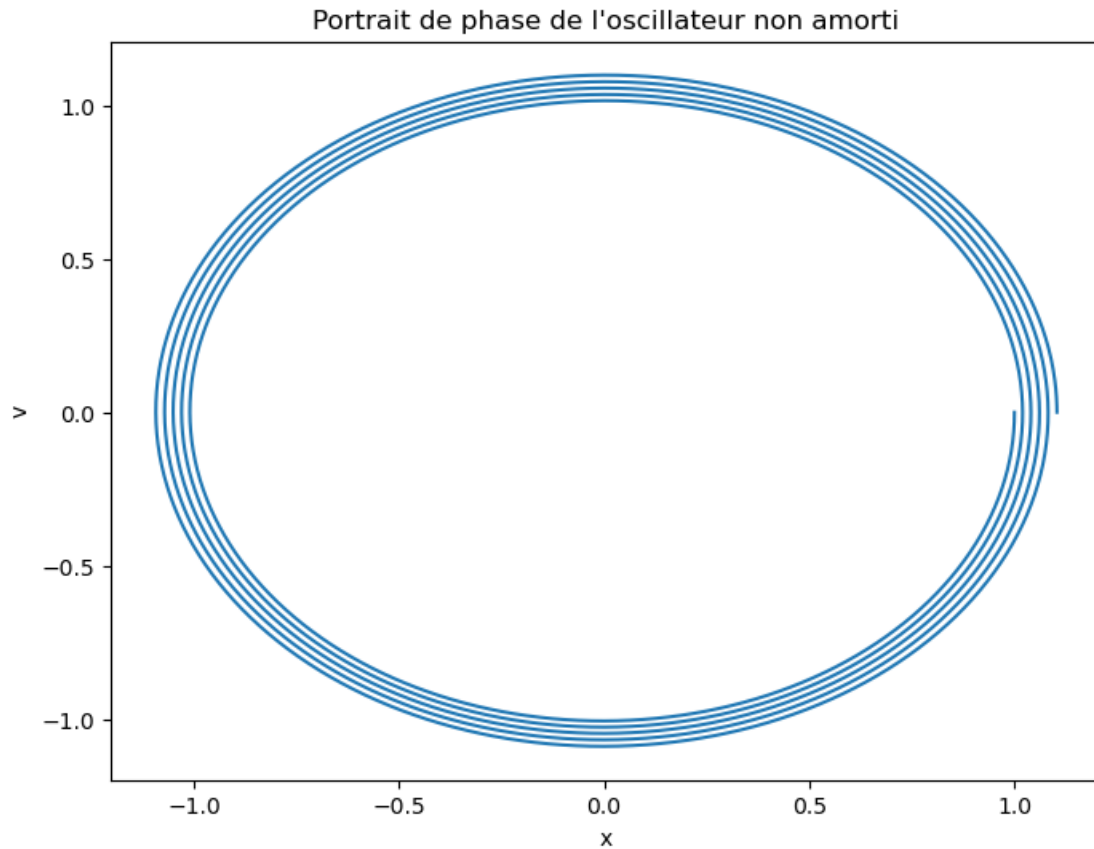
```
m,k=1,1
w0=np.sqrt(k/m)
T0=2*np.pi/w0
```

```
[2]: def acceleration(t,x,v):
    return -w0**2*x
pas=T0/1000
tmax=5*T0
N=int(tmax/pas)
tab_t=np.zeros(N+1)
tab_x=np.zeros(N+1)
tab_v=np.zeros(N+1)
tab_t[0],tab_x[0],tab_v[0]=0,1,0
for n in range(N):
    tab_t[n+1]=tab_t[n]+pas
    tab_x[n+1]=tab_x[n]+pas*tab_v[n]
    tab_v[n+1]=tab_v[n]+pas*acceleration(tab_t[n],tab_x[n],tab_v[n])

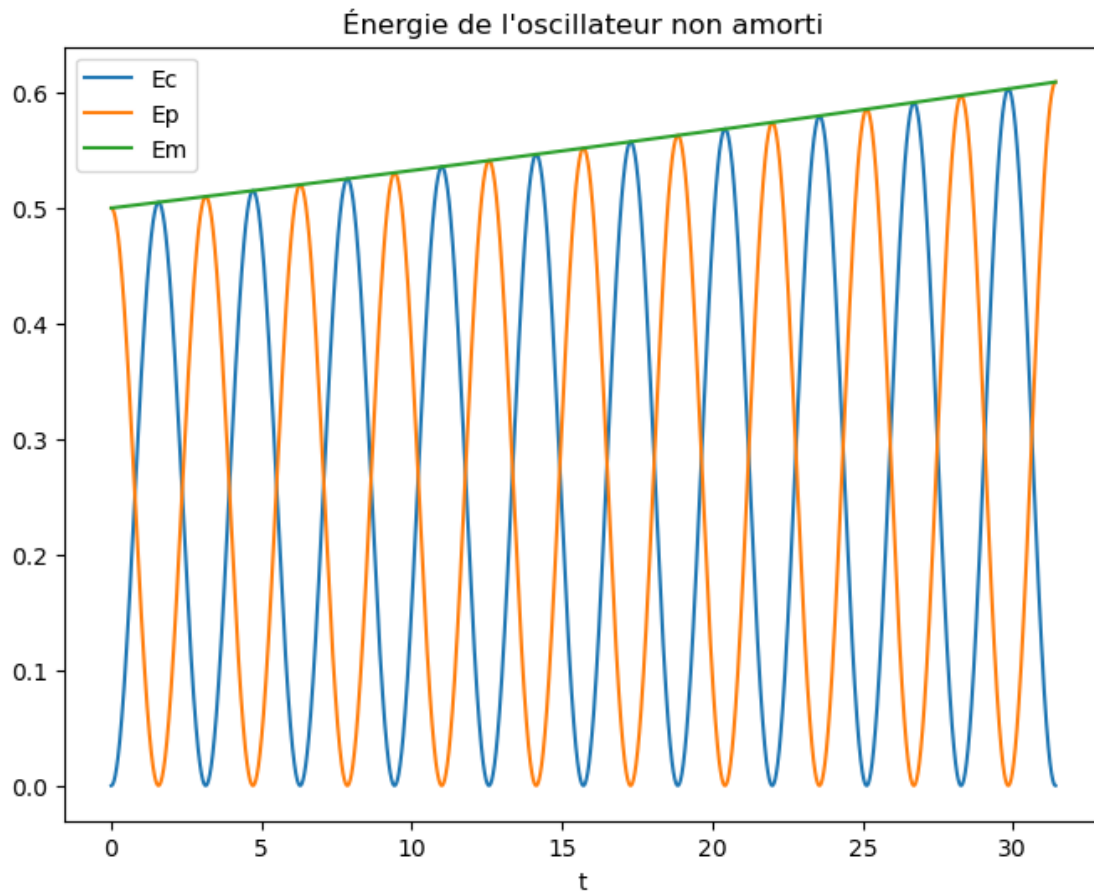
plt.figure(figsize=(8,6))
plt.plot(tab_t,tab_x)
plt.xlabel("t")
plt.ylabel("x")
plt.title("Oscillations de l'oscillateur non amorti")
plt.show()
```



```
[3]: plt.figure(figsize=(8,6))
plt.plot(tab_x,tab_v)
plt.xlabel("x")
plt.ylabel("v")
plt.title("Portrait de phase de l'oscillateur non amorti")
plt.show()
```



```
[4]: tab_Ec=0.5*m*tab_v**2
      tab_Ep=0.5*k*tab_x**2
      tab_Em=tab_Ec+tab_Ep
      plt.figure(figsize=(8,6))
      plt.plot(tab_t,tab_Ec,label="Ec")
      plt.plot(tab_t,tab_Ep,label="Ep")
      plt.plot(tab_t,tab_Em,label="Em")
      plt.xlabel("t")
      plt.legend()
      plt.title("Énergie de l'oscillateur non amorti")
      plt.show()
```



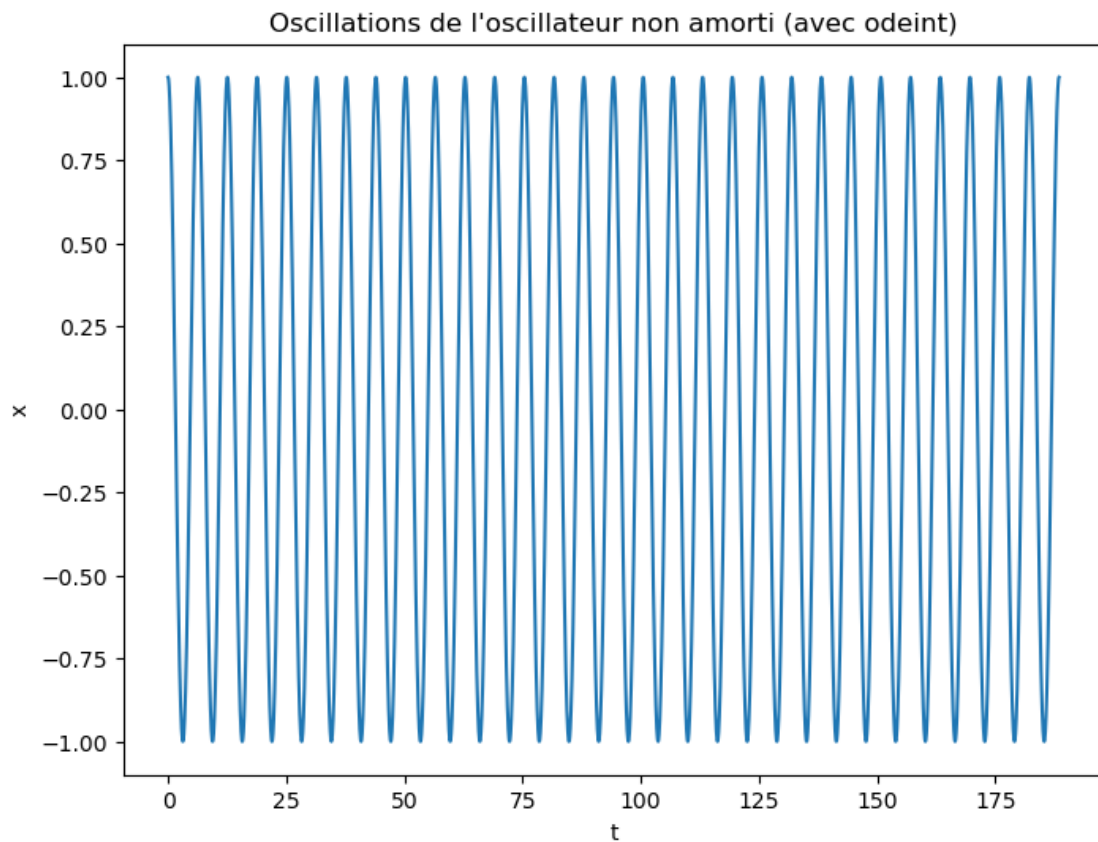
E_m croît alors qu'elle devrait rester constante; c'est un défaut de la méthode d'Euler.

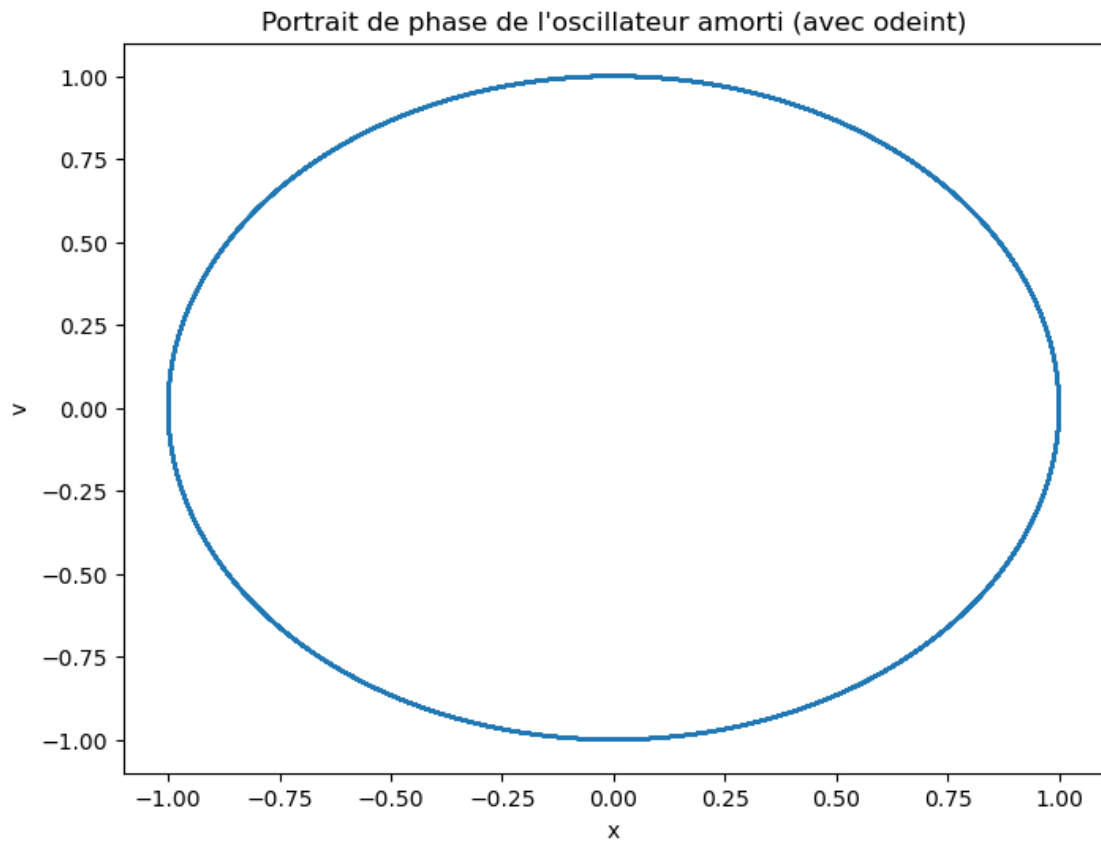
```
[5]: from scipy.integrate import odeint
def derivee(inconnues,t):
    x=inconnues[0]
    v=inconnues[1]
    a=-w0**2*x
    return [v,a]
ci=[1,0]
tab_t=np.linspace(0,30*T0,5000)
solution=odeint(derivee,ci,tab_t)
tab_x=solution[:,0]
tab_v=solution[:,1]
plt.figure(figsize=(8,6))
plt.plot(tab_t,tab_x)
plt.xlabel("t")
plt.ylabel("x")
plt.title("Oscillations de l'oscillateur non amorti (avec odeint)")
plt.show()
```

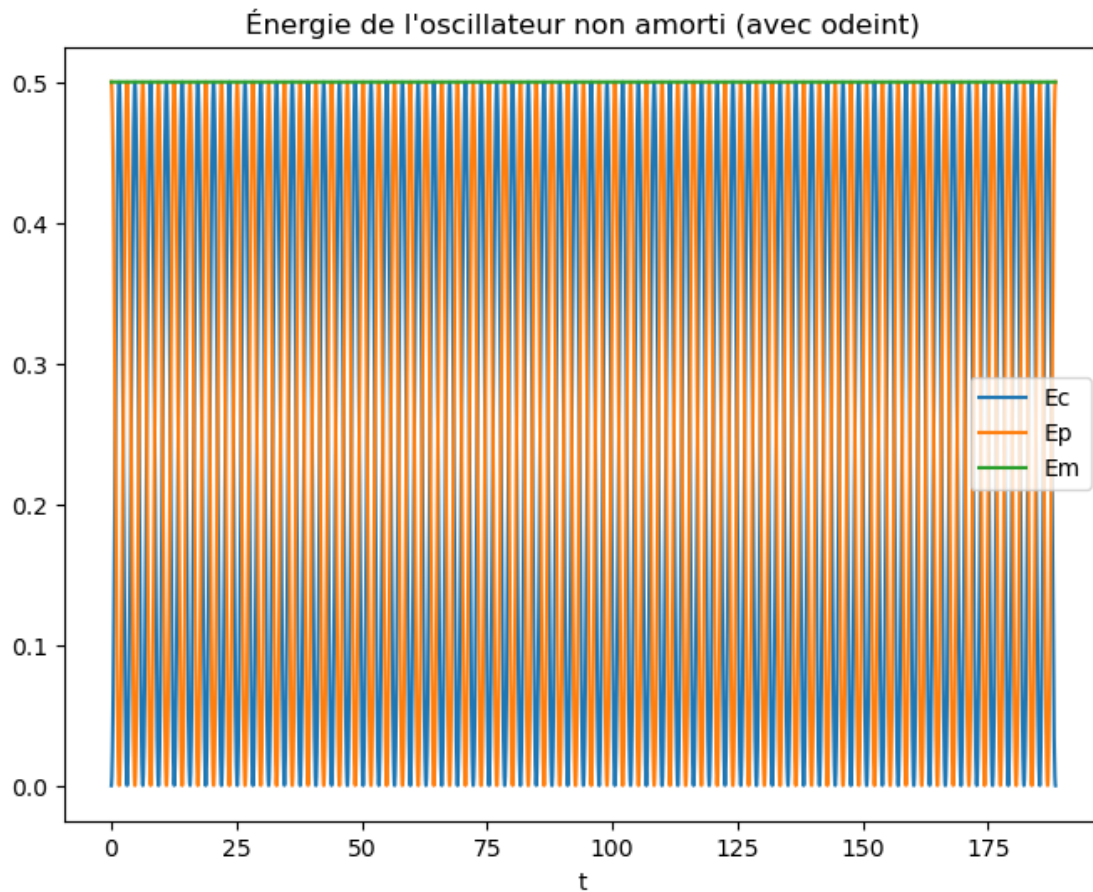
```

plt.figure(figsize=(8,6))
plt.plot(tab_x,tab_v)
plt.xlabel("x")
plt.ylabel("v")
plt.title("Portrait de phase de l'oscillateur amorti (avec odeint)")
plt.show()
tab_Ec=0.5*m*tab_v**2
tab_Ep=0.5*k*tab_x**2
tab_Em=tab_Ec+tab_Ep
plt.figure(figsize=(8,6))
plt.plot(tab_t,tab_Ec,label="Ec")
plt.plot(tab_t,tab_Ep,label="Ep")
plt.plot(tab_t,tab_Em,label="Em")
plt.xlabel("t")
plt.legend()
plt.title("Énergie de l'oscillateur non amorti (avec odeint)")
plt.show()

```





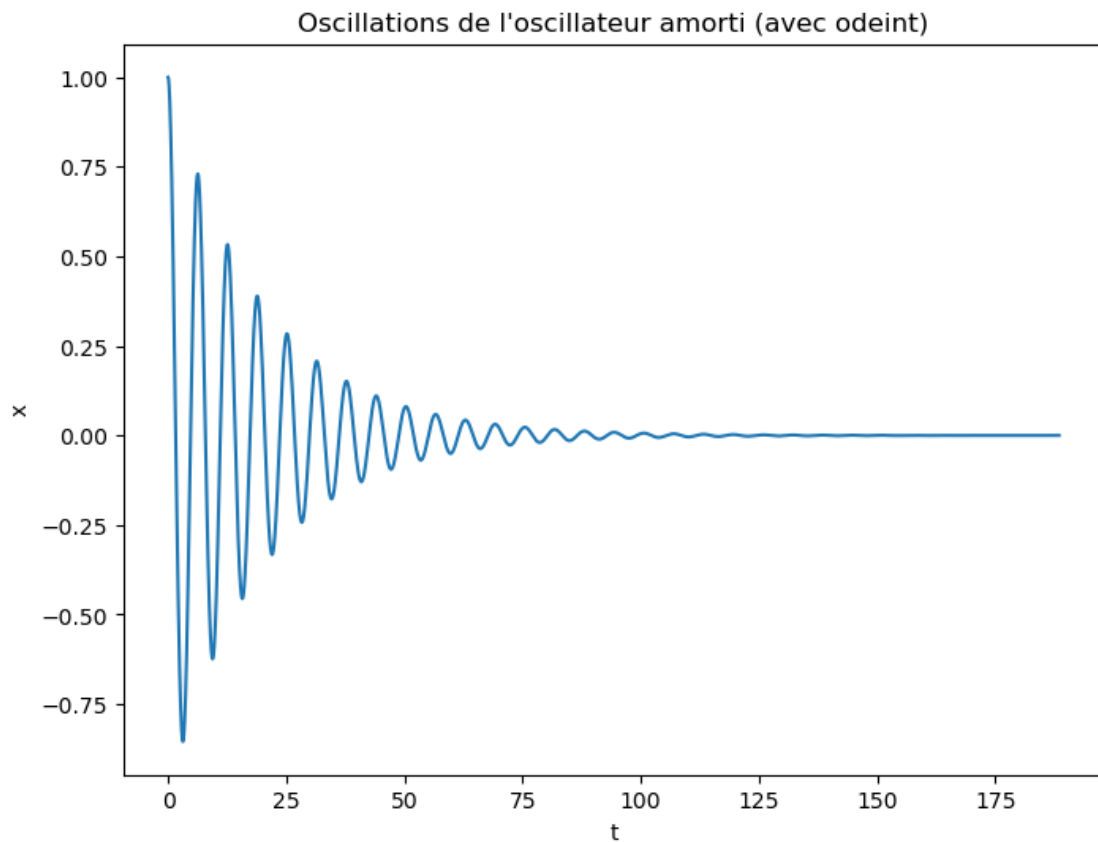


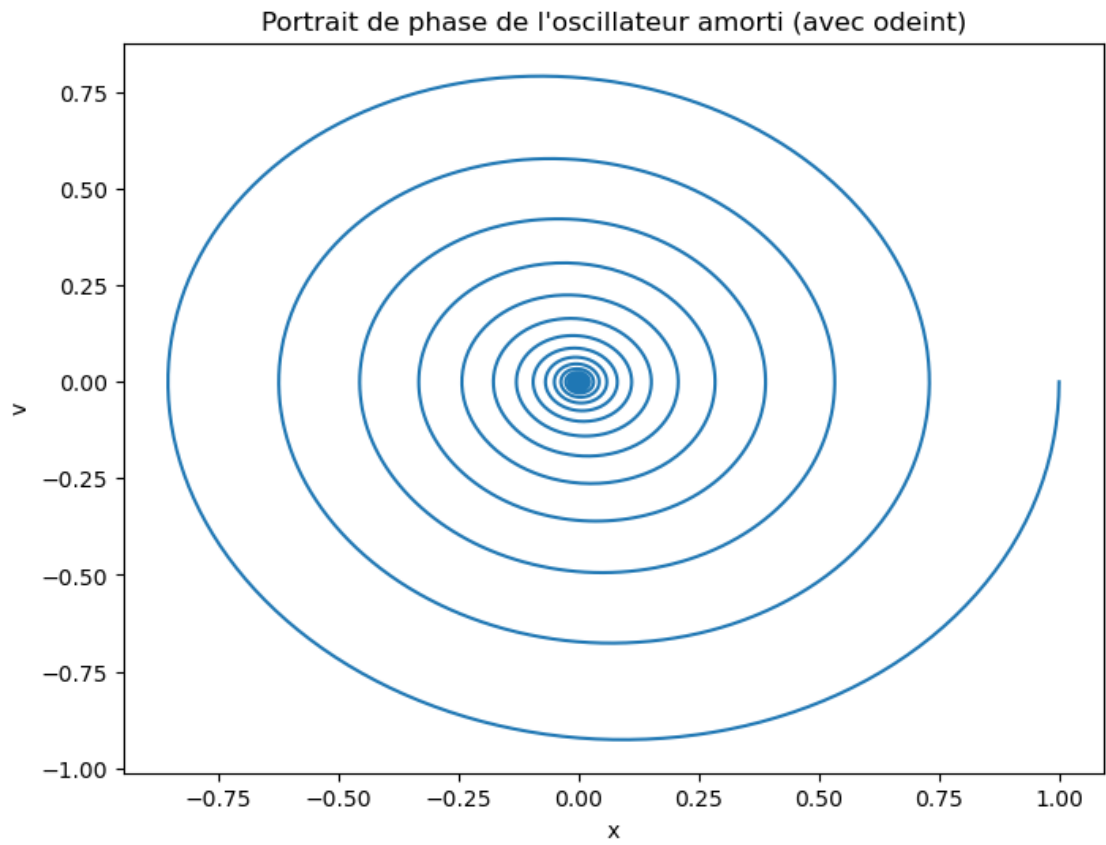
```
[6]: from scipy.integrate import odeint
Q=10
def derivee(inconnues,t):
    x=inconnues[0]
    v=inconnues[1]
    a=-w0**2*x-w0/Q*v
    return [v,a]
ci=[1,0]
tab_t=np.linspace(0,30*T0,5000)
solution=odeint(derivee,ci,tab_t)
tab_x=solution[:,0]
tab_v=solution[:,1]
plt.figure(figsize=(8,6))
plt.plot(tab_t,tab_x)
plt.xlabel("t")
plt.ylabel("x")
plt.title("Oscillations de l'oscillateur amorti (avec odeint)")
plt.show()
```

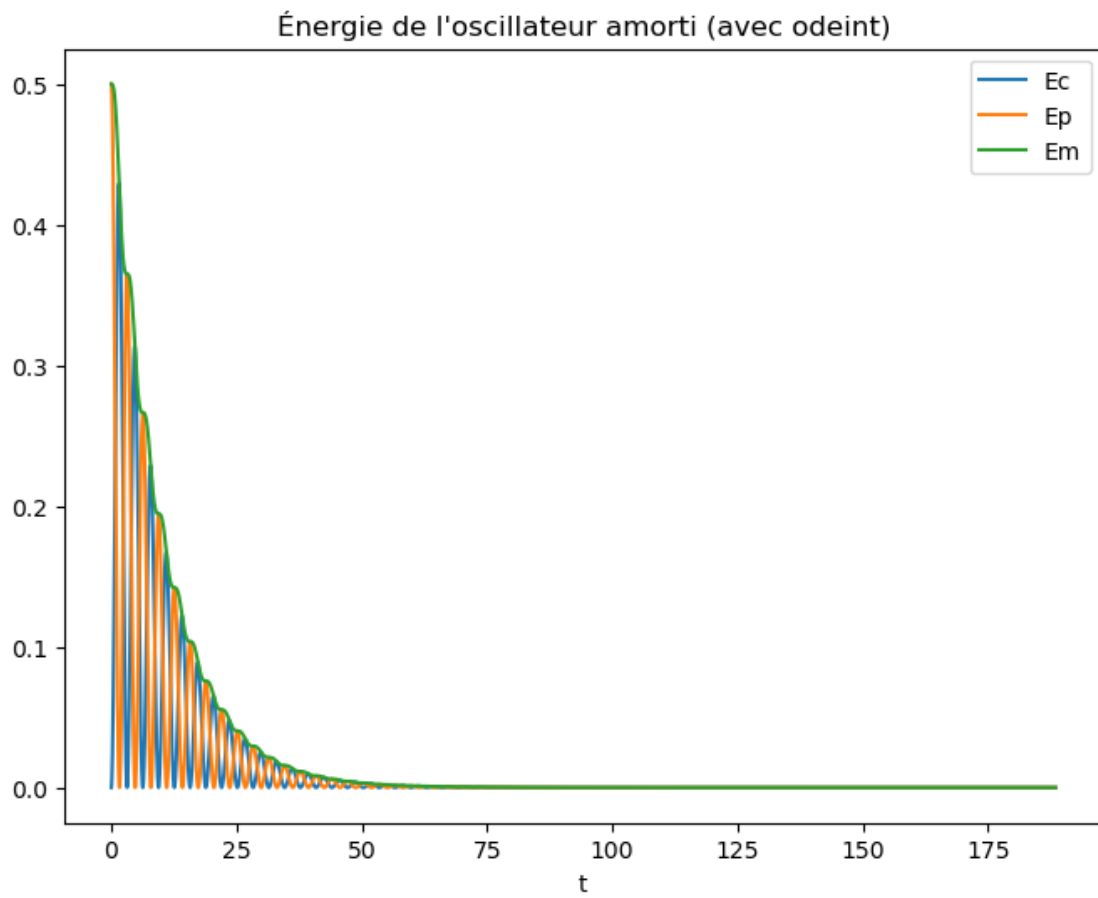
```

plt.figure(figsize=(8,6))
plt.plot(tab_x,tab_v)
plt.xlabel("x")
plt.ylabel("v")
plt.title("Portrait de phase de l'oscillateur amorti (avec odeint)")
plt.show()
tab_Ec=0.5*m*tab_v**2
tab_Ep=0.5*k*tab_x**2
tab_Em=tab_Ec+tab_Ep
plt.figure(figsize=(8,6))
plt.plot(tab_t,tab_Ec,label="Ec")
plt.plot(tab_t,tab_Ep,label="Ep")
plt.plot(tab_t,tab_Em,label="Em")
plt.xlabel("t")
plt.legend()
plt.title("Énergie de l'oscillateur amorti (avec odeint)")
plt.show()

```







[]: