```
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
import matplotlib.animation as animation
from math import *
from matplotlib import patches

r = 2 # mm
R = 50 # mm
g = 9810 # mm/s^2
h0 = 300 # mm
al = 2*g/((R/r)**4 - 1) #coefficient de décroissance mm^(1/2)/s
Tf = 2*sqrt(h0)/al #durée du vidage
nbre_images = 200
H = 100

liste_t = np.linspace(0,Tf,nbre_images)
liste = []

nz = 80
zj = np.linspace(H,0,nz)

def unJet(vv):
    xjl = []
    for i in range(nz):
        xx = 2*R + min(4*R-1,vv*sqrt(2*(H-zj[i])/g)) # limite
"intérieure" de la cuve
        xjl.append(xx)
    return xjl
v = sqrt(2*g*h0)
xj = unJet(v)

fig,ax = plt.subplots()

n_jet = 0
jet, = ax.plot(xj,zj,'b',zorder = 1) #jet = ax.plot(xj,zj,'b',zorder
= 1)[0]
liste.append(jet)

n_support = 1
support = plt.Rectangle([0,0],2*R,H,0.0,edgecolor =
'xkcd:brown',facecolor = 'xkcd:light brown',zorder = 2)
ax.add_patch(support)
liste.append(support)

n_liquide = 2
liquide = plt.Rectangle([0,H],2*R,h0,0.0,edgecolor = 'b',facecolor =
'xkcd:sky blue',zorder = 3)
ax.add_patch(liquide)
liste.append(liquide)

n_boite = 3
boite = plt.Rectangle([0,H],2*R,h0,0.0,fill = False,color =
'k',zorder = 5)
```

```
ax.add_patch(boite)
liste.append(boite)

n_verse = 4
verse = plt.Rectangle([2*R,0],4*R,0,0.0,edgecolor = 'b',facecolor =
'xkcd:sky blue',zorder = 4)
ax.add_patch(verse)
liste.append(verse)

n_cuve = 5
cuve = plt.Rectangle([2*R,0],4*R,h0/4,0.0,fill = False,color =
'k',zorder = 6)
ax.add_patch(cuve)
liste.append(cuve)


plt.axis("equal")
plt.axis("image")

def animate(num):
        t = liste_t[num]
        h = (sqrt(h0)-0.5*al*t)**2
        v = sqrt(2*g*h)
        xj = unJet(v)
        liste[n_jet].set_data(xj,zj)
        liste[n_liquide].set_height(h)
        liste[n_verse].set_height((h0-h)/4)
        return liste

anim = animation.FuncAnimation(fig,animate,len(liste_t),blit=True)
plt.show()
```