Exercice 1: Euler – Ordre 1 – 1 équation

Question 1: Déterminer l'équation différentielle régissant le mouvement la masse et la mettre sous la forme $v'(t) = f(v(t), t)$

$$\sum \overrightarrow{F_{ext}} = ma(t)$$
$$-mg + kv^2(t) = ma(t)$$
$$a(t) = -g + \frac{k}{m}v^2(t)$$
$$v'(t) = \frac{k}{m}v^2(t) - g = f(v(t), t)$$

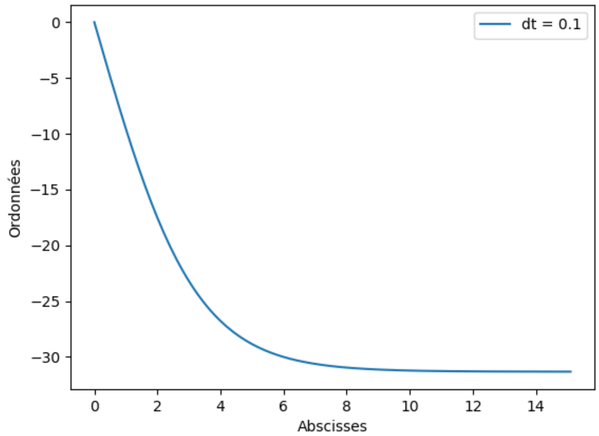
Question 2: Proposer une fonction Euler_Explicite dont les arguments sont à préciser, permettant de résoudre l'équation différentielle $y' = f(y, t)$ avec comme condition initiale $y(t_0) = y_0$, par itérations successives de pas de temps dt sur l'intervalle $[t_0, t_1]$ en utilisant la méthode d'Euler explicite, et renvoyant la liste des temps et la liste des solutions approchées associées

$$y(t + T) = Ty'(t) + y(t)$$

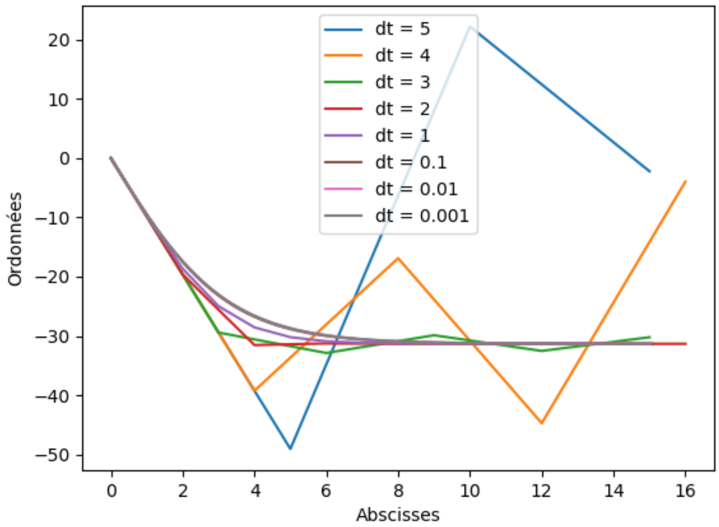
```python
def Euler_Explicite(f,y0,t0,t1,dt):
    t = t0
    y = y0
    T = [t]
    Y = [y]
    while t < t1:
        yp = f(y,t)
        y = y + yp*dt
        t += dt
        T.append(t)
        Y.append(y)
    return T,Y
```

Question 3: Proposer le code Python permettant de déterminer l'évolution de la vitesse de la masse en chute libre pendant 15 secondes avec un pas de temps de 0.1 s et qui affiche la courbe v en fonction de t

```python
# Euler

# Fonctions d'Euler

def Euler_Explicite(f,y0,t0,t1,dt):
    t = t0
    y = y0
    T = [t]
    Y = [y]
    while t < t1:
        yp = f(y,t)
        y = y + yp*dt
        t += dt
        T.append(t)
        Y.append(y)
    return T,Y

def f(y,t):
    k = 0.001
    g = 9.81
    m = 0.1
    yp = (k/m)*(y**2)-g
    return yp

# Résolution

v0 = 0
t0 = 0
t1 = 15
dt = 0.1
T,V = Euler_Explicite(f,v0,t0,t1,dt)

# Affichage de la solution

import matplotlib.pyplot as plt
def f_Affiche_liste(fig_i,Liste_X,Liste_Y,Legende):
    fig = plt.figure(fig_i)
    plt.plot(Liste_X,Liste_Y,label=Legende)
    plt.xlabel('Abscisses')
    plt.ylabel('Ordonnées')
    plt.legend()
    plt.show()

fig = 0
f_Affiche_liste(fig,T,V,'dt = '+ str(dt))
```



Question 4: Créer les quelques lignes de code permettant de tracer sur une même figure la solution explicite pour dt dans la liste [5,4,3,2,1,0.1,0.01,0.001] et conclure sur une valeur acceptable de dt permettant à la solution d'être réaliste



On voit qu'à partir de 0,01, la solution approchée n'évolue plus beaucoup et est donc proche de la solution réelle.

Question 5: En supposant que $v(t+dt) \approx v(t)$, que faut-il changer pour que la résolution soit implicite ?

$$y(t+T) = Ty'(t+T) + y(t)$$

$$y(t+T) = Tf(v(t+dt), t+dt) + y(t) \approx Tf(v(t), t+dt) + y(t)$$

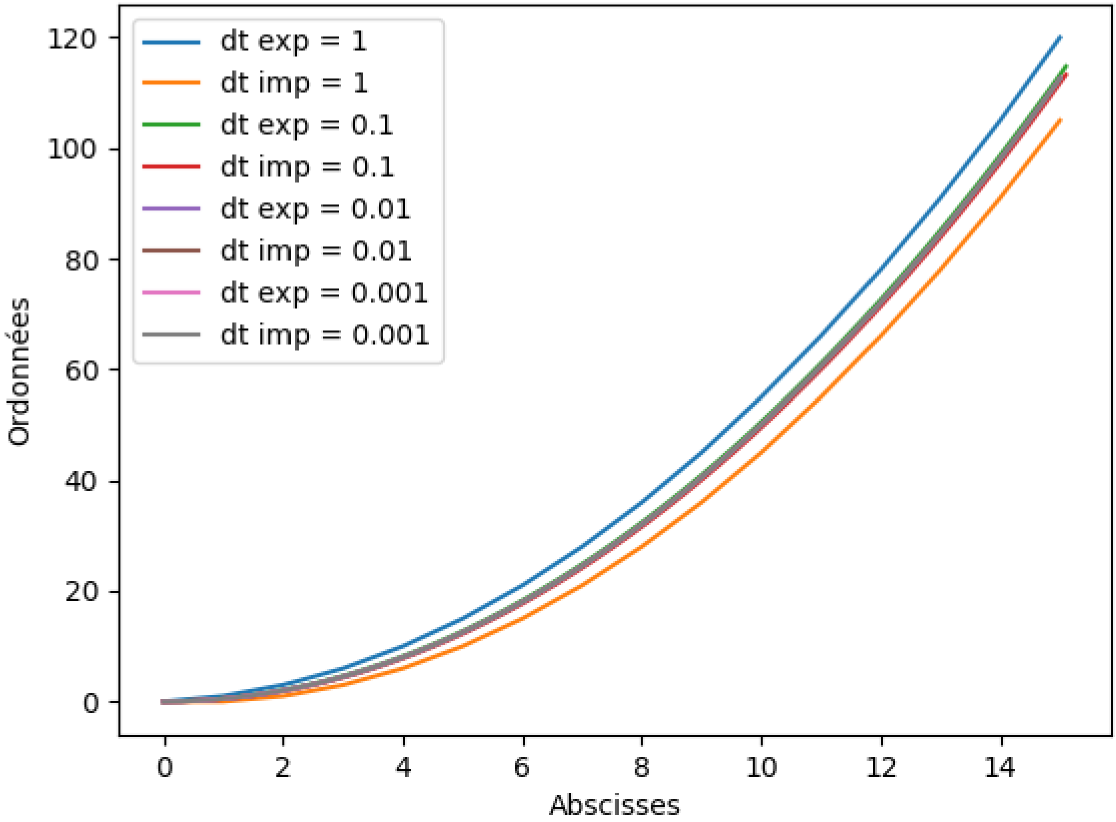
Il y a plusieurs solutions, la plus simple étant :

```
def Euler_Implicite(f,y0,t0,t1,dt):
    t = t0
    y = y0
    T = [t]
    Y = [y]
    while t < t1:
        yp = f(y,t+dt)
        y = y + yp*dt
        t += dt
        T.append(t)
        Y.append(y)
    return T,Y
```

On peut aussi intervertir les lignes ou le temps est mis à jour et ou est calculé le prochain y, auquel cas on n'écrit pas t+dt dans f

Quand on utilise cette fonction, cela n'a pas d'influence car la fonction f n'utilise pas t…

Comparaison des résolutions implicite et explicite :



La solution, c'est $y(t) = t^2$, on tend vers cette solution par le dessous, ou le dessus.