

Méthode d'Euler en programmation Python

La méthode d'Euler est une méthode numérique de résolution des équations différentielles.

En mécanique par exemple, nous devons résoudre l'équation : $v = \frac{dx}{dt}$.

Nous faisons l'approximation : $v(t) = \left(\frac{dx}{dt} \right)_t \approx \frac{\Delta x}{\Delta t} = \frac{x(t+\Delta t) - x(t)}{\Delta t} \Rightarrow x(t+\Delta t) = v(t) \cdot \Delta t + x(t)$

Autrement dit :

- si v est connue à chaque instant t
- en se donnant une durée Δt (le pas d'itération),
- en fixant une abscisse initiale x_0 ,

nous pouvons calculer avec cette relation, les valeurs de x après une durée Δt .

Exemple : mouvement parabolique d'un objet ponctuel, dans le champ de pesanteur uniforme.

Exprimer les accélérations suivant les axes Ox horizontal et Oy vertical.

Écrire avec la méthodes d'Euler, les relations permettant de calculer $v_x(t)$, $v_y(t)$, $x(t)$ et $y(t)$.

```
1 import matplotlib.pyplot as plt
2 import math
3 import matplotlib.pyplot as plt
4 import numpy as np
5
6 g = 9.8
7 h = float(input("altitude initiale"))
8 V0 = float(input("Valeur de la vitesse initiale :"))
9 alpha = float(input("valeur (en °) de l'angle initial de la trajectoire :"))
10
11 ax = []
12 ay = []
13 vx = [V0 * math.cos(alpha * math.pi / 180)]
14 vy = [V0 * math.sin(alpha * math.pi / 180)]
15 x = []
16 y = []
17 ythéo = []
18 t = []
19 i = 1
20 Dt = 0.1
21
22 while y[i-1] >= 0:
23     t.append(t[i-1] + Dt)
24     ax.append(0)
25     ay.append(-g)
26     vx.append(vx[i-1] + ax[i-1] * Dt)
27     vy.append(vy[i-1] + ay[i-1] * Dt)
28     x.append(x[i-1] + vx[i-1] * Dt)
29     y.append(y[i-1] + vy[i-1] * Dt)
30     ythéo.append(-0.5 * g * x[i]**2 / V0**2 / math.cos(math.pi * alpha / 180)**2 + math.tan(alpha * math.pi / 180) * x[i] + h)
31     plt.plot(x, ythéo, 'g-')
32     plt.plot(x, y, 'bo')
33     plt.show()
34     plt.pause(1)
35     i = i + 1
36
37 print(t, ax, ay, vx, vy, x, y)
```

La programmation Python ci-dessus permet de tracer la trajectoire du mobile.

- lignes 1 à 4 : que désignent les bibliothèques importées ?
- Comment sont définis les paramètres du mouvement ?
- Que désignent les formulations des variables t , a_x , a_y , v_x , v_y , x et y ? Compléter les lignes 11 à 18 afin de donner les valeurs initiales de ces variables.
- Comment sont calculées les tableaux de valeurs calculées de t , a_x , a_y , v_x , v_y ?
- Compléter les lignes 28 et 29 pour obtenir les tableaux de valeurs de x et y .
- Expliquer le choix de critère ligne 22.
- Traduire les lignes 31 à 34.
- Que calcule la ligne 30 ?
- Exécuter le programme. Comparer les courbes obtenues. Comment pourrait-on améliorer le programme ? Vérifier votre hypothèse.

Correction :

```
import matplotlib.pyplot as plt
import math
import matplotlib.pyplot as plt
import numpy as np

g = 9.8
h = float(input("altitude initiale"))
V0 = float (input("Valeur de la vitesse initiale : "))
alpha = float(input("valeur (en °)de l'angle initial de la trajectoire :"))

ax = [0]

ay = [-g]
vx = [V0*math.cos(alpha*math.pi/180)]
vy = [V0*math.sin(alpha*math.pi/180)]
vx = [0]
y = [h]
ythéo = [h]
t = [0]
i = 1
Dt = 0.01

while y[i-1] >= 0 :
    t.append(t[i-1]+Dt)
    ax.append(0)
    ay.append(-g)
    vx.append(vx[i-1]+ax[i-1]*Dt)
    vy.append (vy[i-1]+ay[i-1]*Dt)
    x.append (x[i-1]+vx[i-1]*Dt)
    y.append (y[i-1]+vy[i-1]*Dt)
    ythéo.append(-0.5*g*x[i]**2/V0**2/math.cos(math.pi*alpha/180)**2+math.tan(alpha*math.pi/180)*x[i]+h)
    plt.plot(x,ythéo,'g-')
    plt.plot(x,y,'bo')
    plt.show()
    plt.pause(1)
    i=i+1
print (t, ax, ay,vx,vy,x,y)
```