

FYS-MEK1110 Oblig 1

Samuel Bigirimana

February 1, 2021

Euler-Cromer Method for the sprinter

The programing made for a 100m run:

```
1 # Import Python libraries
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 # Define the constants and variables we need
6 m = 80.0 # the runners mass in Kg
7 F = 400.0 # N
8
9 # Values to calculate the air resistance force
10 P = 1.293 # kg/m^3
11 Cd = 1.2 #
12 A_0 = 0.45 # Areal surface of the runner m^2
13 w = 0.0 # Wind velocity
14
15 time = 10 # the maximum time we look at in seconds
16 dt = 0.01 # time step in seconds
17
18 # Here we are using the numpy ceil funtion to get the number of
19 # array elements
20 n = int(np.ceil(time/dt))
21
22 print(f'{n}') # Printing the amount of time steps we take in the
23 # terminal window
24
25 # Define the arrays we need for the time t, position x, velocity v
26 # and acceleration a
27 t = np.zeros(n,float)
28 x = np.zeros(n,float)
29 v = np.zeros(n,float)
30 a = np.zeros(n,float)
31
32 # Here we use the initial conditions for the first vector elements
33 x[0] = 0.0 # at the start, the person is at x = 0
34 t[0] = 0.0 # We start the clock at t = 0 s
35 v[0] = 0.0 # no velocity at the start, v0 = 0 m/s
36 a[0] = 0.0 # at the start, the speed = 0
37
38 # Now we calculate everything
```

```

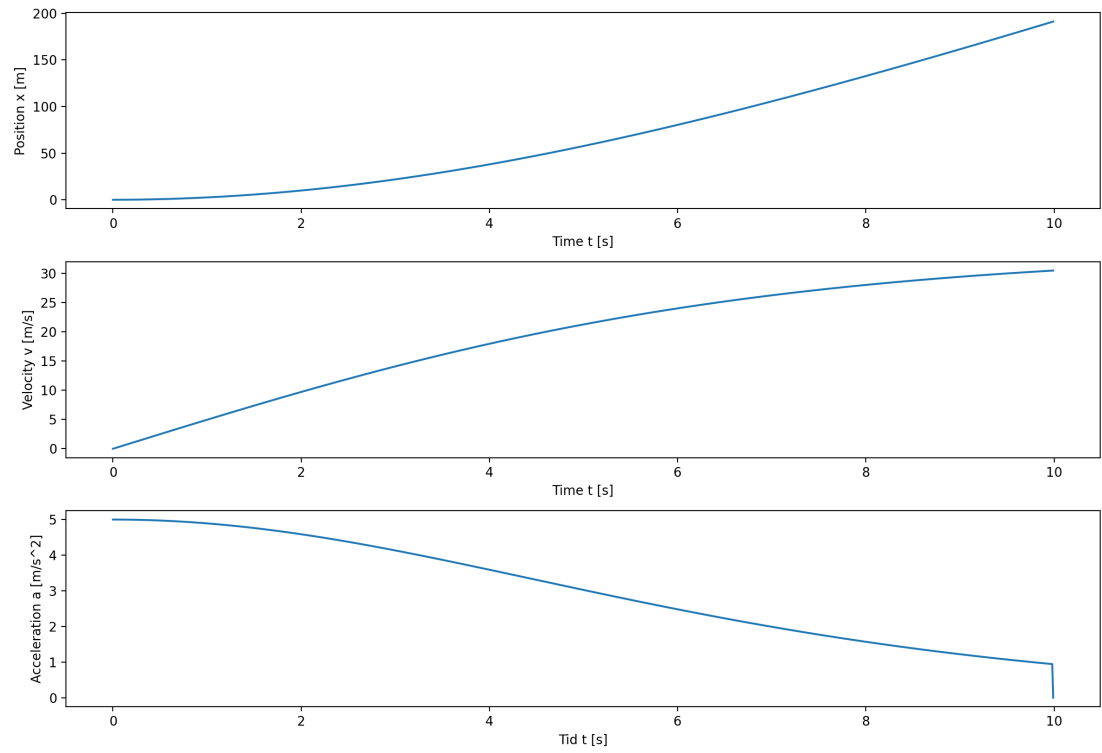
36 for i in range(0, n-1):
37     # Now we calculate the air resistance.
38     D = -0.5 * P * Cd * A_0 * (v[i] - w)**2    # Calculated air
           resistance
39     #Fd = -D*v[i]*abs(v[i]) # it works against the direction of
           motion!
40     # Now that we have all the force components, we can use Newton's
           2nd law
41     #a[i] = (F + Fd)/m # calculating the acceleration:
42     a[i] = (F + D)/m # calculating the acceleration:
43     # Calculating the velocity and the position with the Euler-Cromer
           method:
44     v[i+1] = v[i] + a[i]*dt
45     x[i+1] = x[i] + v[i+1]*dt
46     t[i+1] = t[i] + dt
47
48
49 # In the first "window", we plot how the position x evolves with
           time
50 plt.subplot(3,1,1)
51 plt.plot(t,x)
52 plt.xlabel('Time t [s]')
53 plt.ylabel('Position x [m]')
54
55 # Now switch to the next "window", index = 2
56 # Here we want to plot how the velocity v varies with time t
57 plt.subplot(3,1,2)
58 plt.plot(t,v)
59 plt.xlabel('Time t [s]')
60 plt.ylabel('Velocity v [m/s]')
61
62 # Now switch to the next "window", index = 3
63 # Here we want to plot how the acceleration a varies with time t
64 plt.subplot(3,1,3)
65 plt.plot(t,a)
66 plt.xlabel('Time t [s]')
67 plt.ylabel('Acceleration a [m/s^2]')
68
69 # Show everything:
70 plt.show()

```

Printed time steps we take in the terminal window:

10 000

The graphs for the runner



The second modeling of the runner with other forces added:

```
1 # Import Python libraries
2 import numpy as np
3 import matplotlib.pyplot as plt
4 from math import exp, sqrt
5
6 # Define the constants and variables we need
7 m = 80.0 # the runners mass in Kg
8 F = 400.0 # N
9
10 # Values to calculate the air resistance force
11 P = 1.293 # kg/m^3
12 Cd = 1.2 #
13 A_0 = 0.45 # Areal surface of the runner m^2
14 w = 0.0 # Wind velocity
15 Fd = 25.8 # sN/m
16
17 # Values to calculate other forces on the sprinter
18 f_c = 488 # N
19 t_c = 0.67 # s
20
21 time = 10 # the maximum time we look at in seconds
22 dt = 0.01 # time step in seconds
23
24 # Here we are using the numpy ceil funtion to get the number of
    array elements
25 n = int(np.ceil(time/dt))
26
27 print(f'{n}') # Printing the amount of time steps we take in the
    terminal window
28
29 # Define the arrays we need for the time t, position x, velocity v
    and acceleration a
30 t = np.zeros(n,float)
31 x = np.zeros(n,float)
32 v = np.zeros(n,float)
33 a = np.zeros(n,float)
34
35 # Here we use the initial conditions for the first vector elements
36 x[0] = 0.0 # at the start, the person is at x = 0
37 t[0] = 0.0 # We start the clock at t = 0 s
38 v[0] = 0.0 # no velocity at the start, v0 = 0 m/s
39 a[0] = 0.0 # at the start, the speed = 0
40
41 # Now we calculate everything
42 for i in range(0, n-1):
43     # We calculate the air resistance force
44     D = 0.5 * A_0 * (1 - 0.25 * exp(-(t[i]/t_c)**2)) * P * Cd * (v[i]
        ]-w)**2 # New calculated air resistance
45     # Now that we have all the force components, we can use Newton's
        2nd law
46     # calculating the acceleration:
47     a[i] = (F + f_c * exp(-(t[i]/t_c)**2) - (Fd * v[i]) - D)/m
48     # Calculating the velocity and the position with the Euler-Cromer
        method:
49     v[i+1] = v[i] + a[i]*dt
```

```

50 x[i+1] = x[i] + v[i+1]*dt
51 t[i+1] = t[i] + dt
52
53 V_t = sqrt(2*F/(P*Cd*A_0)) # The formula for terminal velocity
    based on drag force "D"
54 V_t2 = F/Fd #The formula for terminal velocity based on only
    driving force
55 print(V_t) # Printing out the calculated terminal velocity
56 print(V_t2) # Printing out the calculated terminal velocity based
    on driving force
57
58 # In the first "window", we plot how the position x evolves with
    time
59 plt.subplot(3,1,1)
60 plt.plot(t,x)
61 plt.xlabel('Time t [s]')
62 plt.ylabel('Position x [m]')
63
64 # Now switch to the next "window", index = 2
65 # Here we want to plot how the velocity v varies with time t
66 plt.subplot(3,1,2)
67 plt.plot(t,v)
68 plt.xlabel('Time t [s]')
69 plt.ylabel('Velocity v [m/s]')
70
71 # Now switch to the next "window", index = 3
72 # Here we want to plot how the acceleration a varies with time t
73 plt.subplot(3,1,3)
74 plt.plot(t,a)
75 plt.xlabel('Time t [s]')
76 plt.ylabel('Acceleration a [m/s^2]')
77
78 # Show everything:
79 plt.show()

```

The new graph for the runner

