FYS-MEK1110 Oblig 1

Samuel Bigirimana

February 1, 2021

Euler-Cromer Method for the sprinter

The programing made for a 100m run:

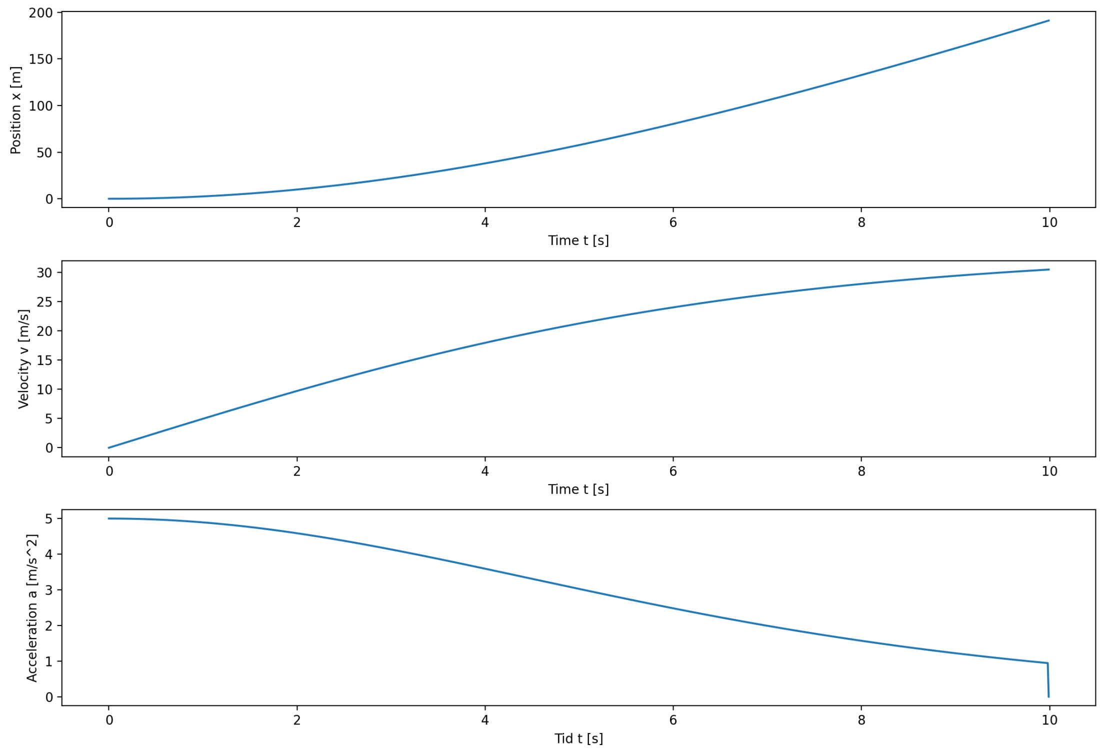
```
# Import Python libraries
import numpy as np
import matplotlib.pyplot as plt

# Define the constants and variables we need
m = 80.0  # the runners mass in Kg
F = 400.0 # N

# Values to calculate the air resistance force
P = 1.293 # kg/m^3
Cd = 1.2  #
A_0 = 0.45  # Areal surface of the runner m^2
w = 0.0   # Wind velocity

time = 10 # the maximum time we look at in seconds
dt = 0.01 # time step in seconds

# Here we are using the numpy ceil funtion to get the number of array elements
n = int(np.ceil(time/dt))

print(f'{n}') # Printing the amount of time steps we take in the terminal window

# Define the arrays we need for the time t, position x, velocity v and acceleration a
t = np.zeros(n,float)
x = np.zeros(n,float)
v = np.zeros(n,float)
a = np.zeros(n,float)

# Here we use the initial conditions for the first vector elements
x[0] = 0.0  # at the start, the person is at x = 0
t[0] = 0.0  # We start the clock at t = 0 s
v[0] = 0.0  # no velocity at the start, v0 = 0 m/s
a[0] = 0.0  # at the start, the speed = 0

# Now we calculate everything
```

```python
for i in range(0, n-1):
  # Now we calculate the air resistance.
  D = -0.5 * P * Cd * A_0 * (v[i] - w)**2   # Calculated air resistence
  #Fd = -D*v[i]*abs(v[i]) # it works against the direction of motion!
  # Now that we have all the force components, we can use Newton's 2nd law
  #a[i] = (F + Fd)/m # calculating the acceleration:
  a[i] = (F + D)/m # calculating the acceleration:
  # Calculating the velocity and the position with the Euler-Cromer method:
  v[i+1] = v[i] + a[i]*dt
  x[i+1] = x[i] + v[i+1]*dt
  t[i+1] = t[i] + dt


# In the first "window", we plot how the position x evolves with time
plt.subplot(3,1,1)
plt.plot(t,x)
plt.xlabel('Time t [s]')
plt.ylabel('Position x [m]')

# Now switch to the next "window", index = 2
# Here we want to plot how the velocity v varies with time t
plt.subplot(3,1,2)
plt.plot(t,v)
plt.xlabel('Time t [s]')
plt.ylabel('Velocity v [m/s]')

# Now switch to the next "window", index = 3
# Here we want to plot how the acceleration a varies with time t
plt.subplot(3,1,3)
plt.plot(t,a)
plt.xlabel('Tid t [s]')
plt.ylabel('Acceleration a [m/s^2]')

# Show everything:
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:

```python
# Import Python libraries
import numpy as np
import matplotlib.pyplot as plt
from math import exp, sqrt

# Define the constants and variables we need
m = 80.0  # the runners mass in Kg
F = 400.0 # N

# Values to calculate the air resistance force
P = 1.293 # kg/m^3
Cd = 1.2  #
A_0 = 0.45  # Areal surface of the runner m^2
w = 0.0   # Wind velocity
Fd = 25.8   # sN/m

# Values to calculate other forces on the sprinter
f_c = 488   # N
t_c = 0.67  # s

time = 10 # the maximum time we look at in seconds
dt = 0.01 # time step in seconds

# Here we are using the numpy ceil funtion to get the number of array elements
n = int(np.ceil(time/dt))

print(f'{n}') # Printing the amount of time steps we take in the terminal window

# Define the arrays we need for the time t, position x, velocity v and acceleration a
t = np.zeros(n,float)
x = np.zeros(n,float)
v = np.zeros(n,float)
a = np.zeros(n,float)

# Here we use the initial conditions for the first vector elements
x[0] = 0.0  # at the start, the person is at x = 0
t[0] = 0.0  # We start the clock at t = 0 s
v[0] = 0.0  # no velocity at the start, v0 = 0 m/s
a[0] = 0.0  # at the start, the speed = 0

# Now we calculate everything
for i in range(0, n-1):
  # We calculate the air resistance force
  D = 0.5 * A_0 * (1 - 0.25 * exp(-(t[i]/t_c)**2)) * P * Cd * (v[i]-w)**2 # New calculated air resistence
  # Now that we have all the force components, we can use Newton's 2nd law
  # calculating the acceleration:
  a[i] = (F + f_c * exp(-(t[i]/t_c)**2) - (Fd * v[i]) - D)/m
  # Calculating the velocity and the position with the Euler-Cromer method:
  v[i+1] = v[i] + a[i]*dt
```

```python
  x[i+1] = x[i] + v[i+1]*dt
  t[i+1] = t[i] + dt

V_t = sqrt(2*F/(P*Cd*A_0)) # The formula for terminal velocity
    based on drag forve "D"
V_t2 = F/Fd     #The formula for terminal velocity based on only
    driving force
print(V_t)  # Printing out the calculated terminal velocity
print(V_t2) # Printing out the calculated terminal velocity based
    on driving force

# In the first "window", we plot how the position x evolves with
    time
plt.subplot(3,1,1)
plt.plot(t,x)
plt.xlabel('Time t [s]')
plt.ylabel('Position x [m]')

# Now switch to the next "window", index = 2
# Here we want to plot how the velocity v varies with time t
plt.subplot(3,1,2)
plt.plot(t,v)
plt.xlabel('Time t [s]')
plt.ylabel('Velocity v [m/s]')

# Now switch to the next "window", index = 3
# Here we want to plot how the acceleration a varies with time t
plt.subplot(3,1,3)
plt.plot(t,a)
plt.xlabel('Tid t [s]')
plt.ylabel('Acceleration a [m/s^2]')

# Show everything:
plt.show()
```

The new graph for the runner