Physics Practice Homework #5

"""

Calculations without air resistance:

Time to reach max height:
vf = vo - g*t
=> t = vo/g = 37.66
Max height:
x = 0.5 (vo + vf)t = 0.5 vo t = 0.5 vo^2 /g = 6956.47
Velocity upon impact:
vf^2 = vo^2 - 2gx
=> vf = - sqrt(2gx) = 369.44
(can also be trivially proven by conservation of energy, meaning the kinetic energy and hence the velocity at launch and at impact is the same)

"""

```
import numpy as np
import matplotlib.pyplot as plt
from mpmath import mp
import time
mp.dps = 10


# Constants
g = 9.81
mass = mp.mpf(0.00603)
Cd = mp.mpf(0.295)
rho = mp.mpf(1.293)
diameter = mp.mpf(7.82e-3)
A = mp.pi * (diameter / 2) ** 2
v0 = mp.mpf(369.44)
dt = mp.mpf(0.001)

def drag_force(v):
    return 0.5 * Cd * rho * A * v**2

def simulate_motion_without_drag(v0, dt):
    y = 0
    v = v0
    t = 0
```

```
    positions = []
    velocities = []
    times = []

    while y >= 0:
        F_gravity = mass * g
        F_net = F_gravity

        a = -F_net / mass

        v += a * dt
        y += v * dt
        t += dt

        positions.append(float(y))
        velocities.append(float(v))
        times.append(float(t))

    return np.array(times), np.array(positions), np.array(velocities)

def simulate_motion_with_drag(v0, dt):
    y = 0
    v = v0
    t = 0
    positions = []
    velocities = []
    times = []

    while y >= 0:
        F_gravity = mass * g
        F_drag = drag_force(v)
        if v >= 0:
            F_net = F_gravity + F_drag
        elif v < 0:
            F_net = F_gravity - F_drag

        a = -F_net / mass

        v += a * dt
        y += v * dt
        t += dt

        positions.append(float(y))
        velocities.append(float(v))
        times.append(float(t))

    return np.array(times), np.array(positions), np.array(velocities)
```

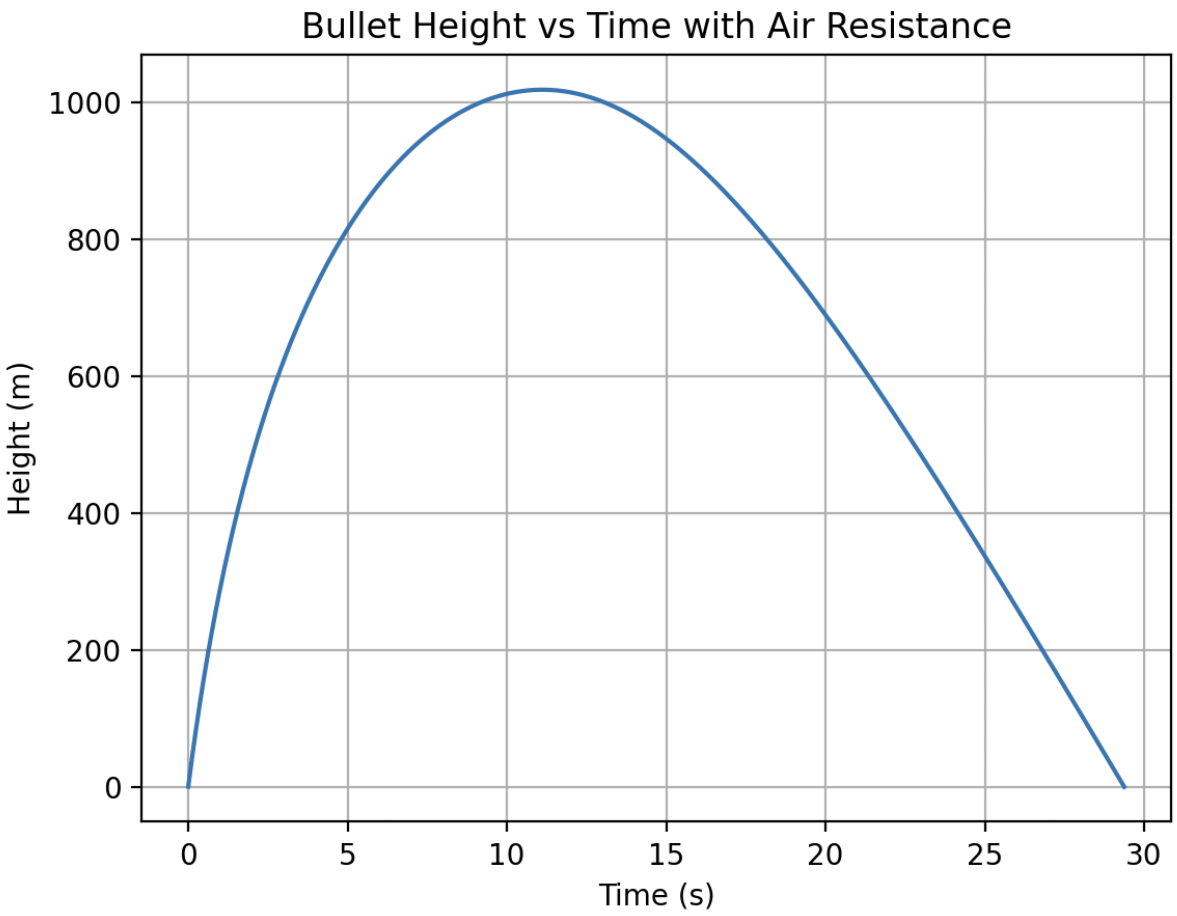
```
drag_bool = input("With air resistance (True or False): ").strip().lower() == 'true'

if drag_bool:
    times, positions, velocities = simulate_motion_with_drag(v0, dt)
elif not drag_bool:
    times, positions, velocities = simulate_motion_without_drag(v0, dt)


max_height = np.max(positions)
time_to_ground = times[-1]
velocity_on_impact = velocities[-1]

print(f"Maximum height: {max_height:.3f} m")
print(f"Time to hit the ground: {time_to_ground:.3f} s")
print(f"Velocity upon impact: {abs(velocity_on_impact):.3f} m/s")

# Plot the height vs time graph
plt.plot(times, positions)
plt.title('Bullet Height vs Time with Air Resistance')
plt.xlabel('Time (s)')
plt.ylabel('Height (m)')
plt.grid(True)
plt.show()
```


Bullet Height vs Time with Air Resistance
Height (m)
Time (s)
0
200
400
600
800
1000
0
5
10
15
20
25
30