Laboratory 3: Projectile motion under the action of air resistance

ismisebrendan

March 28, 2023

1 Introduction

The aim of the lab is to determine projectile motions under air resistance for a number of different starting conditions, including differing initial velocities, angles, heights and masses.

Air resistance acts in the opposite direction to the motion of an object and it is dependent on the velocity of the object. It can be written mathematically as:

$$-f(V)\hat{u} \tag{1.1}$$

Where $\hat{u}$ is the unit vector of the velocity, and $f(V)$ is some function of the velocity. The function $f(V)$ can be approximated as having a linear and quadratic term in it.

$$f(V) = bV + cV^2 \tag{1.2}$$

The coefficients b and c in the function f are dependent on the size and shape of the object. For a spherical object of diameter D they are given by

$$b = BD \tag{1.3}$$

$$c = CD^2 \tag{1.4}$$

B and C depend on the nature of the medium. For air, B = 1.6×10^{-4} Nsm^{-2}, and C = B = 0.25 Ns^2m^{-4}. It is obvious that $f(v)$ in Eq. 1.2 can also be written as a function of VD

$$f(VD) = b(VD) + c(VD) \tag{1.5}$$

Where

$$b(VD) = BDV \tag{1.6}$$

$$c(VD) = C(VD)^2 \tag{1.7}$$

At certain values of VD the linear contribution from $b(VD)$ or the quadratic contribution from $c(VD)$ can be neglected. At the point that the quadratic term can be neglected Newton's second law can be expressed as:

$$\frac{dV_y}{dt} = -g - \frac{b}{m}V_y \tag{1.8}$$

And approximating for a small, finite value of dt, Δt,

$$\Delta V_y = -g\Delta t - \frac{b}{m}V_y\Delta t \tag{1.9}$$

The horizontal velocity, V_x is very similar to the vertical velocity, the only difference being that there is no dependence on gravity for the horizontal velocity, leading to the following equation

$$\frac{dV_x}{dt} = -\frac{b}{m}V_x \tag{1.10}$$

Which can be approximated for a small, finite value of dt, Δt, as

$$\Delta V_x = -\frac{b}{m}V_x\Delta t \tag{1.11}$$

In the case where there is no resistance both b and c can be let equal 0.

As velocity is the rate of change of position with respect to time the change in position of a particle can be numerically calculated by Eq. 1.12. The same can be applied to horizontal position and velocity, X and V_x respectively.

$$\Delta Y = V * \Delta t \tag{1.12}$$

There is an analytical solution to the vertical velocity of an object under air resistance from the linear term and it is given below

$$V_y = \frac{mg}{b}\left(e^{-bt/m} - 1\right) \tag{1.13}$$

If the quadratic term, the $c(VD)$ term is not neglected the horizontal and vertical velocities (V_x and V_y respectively) can be numerically modelled as

$$\frac{dV_x}{dt} = -\frac{c}{m}\sqrt{V_x^2 + V_y^2}V_x \tag{1.14}$$

$$\frac{dV_y}{dt} = -g - \frac{c}{m}\sqrt{V_x^2 + V_y^2}V_y \tag{1.15}$$

Which can respectively be approximated for a small, finite value of dt, Δt, as

$$\Delta V_x = -\frac{c}{m}\sqrt{V_x^2 + V_y^2}V_x\Delta t \tag{1.16}$$

$$\Delta V_y = -g\Delta t - \frac{c}{m}\sqrt{V_x^2 + V_y^2}V_y\Delta t \tag{1.17}$$

2 Methodology

2.1 Scaling of Air Resistance with Velocity

The code for this section can be found in *Appendix A: Code, 5.1.1 Scaling of Air Resistance with Velocity* below.

The functions $b(VD)$, $c(VD)$ and $f(VD)$ were defined as laid out above in Eqs. 1.5, 1.6 and 1.7.

Through changing the value of VD inputted into the function the ranges over which the linear term could be neglected, the quadratic term could be neglected or where they are both significant were found.

The graphs of $b(VD)$ and $c(VD)$ were plotted for the ranges found above.

Through inputting the given values for the different projectiles into $b(VD)$, $c(VD)$ and $f(VD)$ the ideal form of $f(V)$ was found for the three projectiles.

2.2 Vertical Motion Under the Action of Air Resistance

The code for this section can be found in *Appendix A: Code, 5.1.2 Vertical Motion Under the Action of Air Resistance* below.

As the value of VD used in much of this experiment tends to be less than $10^{-5}\ m^2s^{-1}$, as the diameter itself is 10^{-5} m, the quadratic term is neglected.

For a spherical dust grain of density $2 \times 10^3\ kgm^{-3}$ and the diameter stated above the vertical velocity of the object was a function of time was determined through implementation of Eq. 1.9 in the program, simulated over 1 second and a graph of V_y against time was plotted for this dust grain.

Graphs of V_y against time were then plotted for different values of masses between 10^{-10} kg and 10 kg and a fixed value of resistance, $b = 1.6 \times 10^{-8}\ Nsm^{-2}$. And then for different values of b between $1.6 \times 10^{-6}\ Nsm^{-2}$ and $1.6 \times 10^{-15}\ Nsm^{-2}$ for a fixed mass the same as the mass of the spherical dust grain used above. The effect of increasing the mass was compared to the effect of decreasing the resistance.

The analytical solution to the vertical motion of the particle as shown in Eq. 1.13 was implemented and the motion of the mass for both the numerical and analytical solutions was compared, as was the error between them.

The time taken for the particle to fall a distance of 5 metres as a function of mass was measured through implementation of Eq. 1.12 with an initial value for Y of 5 m and adding ΔY to Y after each time step. A number of masses from 10^{-8} kg to 100 kg were investigated.

2.3 Projectile Motion Under the Action of Air Resistance - Restricted to Linear Terms

The code for this section can be found in *Appendix A: Code, 5.1.3 Projectile Motion Under the Action of Air Resistance - Restricted to Linear Terms* below.

Two functions to calculate the trajectories of particles of mass m $= 1.047 \times 10^{-9}$ kg, one subject to air resistance and the other not subject to it respectively were initialised in the program, where Eqs 1.9 and 1.11 were used for the trajectory under air resistance, while in the case of no air resistance b was set to zero. The values of the vertical and horizontal positions, Y and X respectively, were updated accordingly until the simulated particle returned to the ground. The two trajectories were plotted for particles with initial velocities of 1 m/s launched at an angle of $\pi/4$ to the horizontal.

Through the use of a loop the optimum launching angle to maximise the horizontal displacement for a particle of the mass above was determined to an accuracy of 0.01 rad.

Through another loop the dependence of the optimum angle on the mass of the particle was determined for a number of masses between 10^{-9} kg and 10^{-3} kg, and the relationship was plotted.

2.4 Projectile Motion Under the Action of Air Resistance

The code for this section can be found in *Appendix A: Code, 5.1.4 Projectile Motion Under the Action of Air Resistance* below.

The code used in *2.3 Projectile Motion Under the Action of Air Resistance - Restricted to Linear Terms* was adapted to also include calculations of the trajectory according to Eqs 1.16 and 1.17. The trajectories of the objects were plotted for a number of masses between 10^{-10} kg and 10^{-2} kg, a number of initial angles between 15° and 90°, and initial speeds between 1 m/s and 20 m/s.

3 Results

The graphs are available in *5.2 Appendix B: Graphs* below.

3.1 Scaling of Air Resistance with Velocity

The graphs of $b(VD)$, $c(VD)$ and $f(VD)$ are shown in Figure 5.1 for a range over which both the linear and quadratic contributions to air resistance are significant.

The graphs of $b(VD)$, $c(VD)$ and $f(VD)$ are shown in Figure 5.2 for a range over which only the linear contribution to air resistance is significant.

The graphs of $b(VD)$, $c(VD)$ and $f(VD)$ are shown in Figure 5.3 for a range over which only the quadratic contribution to air resistance is significant.

It was found that when the product of the velocity and diameter, VD, is below approximately $10^{-5}\ m^2s^{-1}$ only the linear term is relevant in the calculation of the air resistance. Between values of approximately $10^{-5}\ m^2s^{-1}$ and $10^{-3}\ m^2s^{-1}$ for VD both terms are relevant, and above $10^{-3}\ m^2s^{-1}$ only the quadratic term is relevant.

The results of the calculations for the three projectiles are given in Table 3.1. As can be seen from the table the quadratic term is a sufficient approximation for the motion of the baseball, the linear term is a sufficient approximation for the motion of the oil drop, and both terms must be used to model the motion of the rain drop.

Projectile	b(VD) (N)	c(VD) (N)	f(VD) (N)
Baseball	5.6×10^{-5}	0.030625	0.030681
Oil drop	1.2×10^{-14}	1.40625×10^{-21}	1.2×10^{-14}
Rain drop	1.6×10^{-7}	2.5×10^{-7}	4.1×10^{-7}

Table 3.1: The results of the calculations for the three projectiles

3.2 Vertical Motion Under the Action of Air Resistance

The graph of the velocity of the simulated dust grain for time steps of 0.0001 s over one second is shown in Figure 5.4, the velocity is seen to increase and then approach its terminal velocity.

As seen for the graphs of velocity against time for different masses and fixed b in Figure 5.5 when the mass gets significantly large the linear resistance term breaks down and approaches zero. This also leads to large speeds at which the quadratic term in the Eq. 1.2 becomes relevant once again.

As seen from the graphs in Figures 5.5 and 5.6 a decrease in mass has the same effect on the graph of the velocity against time as an increase in the coefficient of resistance, and vice-versa.

The graphs in Figure 5.7 show that the numerical approximation used here is very good as the graphs of the numerical result (Figure 5.7a) and the analytical result (Figure 5.7b) appear very similar, and the error between them (Figure 5.7c) is very small at all points, and decreases as the time increases apart from an initial spike.

The graph in Figure 5.8 shows the dependence of the time taken to fall 5 m on the mass of the particles of uniform cross-section. Above masses in the range of 10^{-5} kg the particles all fall with the same acceleration as is generally expected.

3.3 Projectile Motion Under the Action of Air Resistance - Restricted to Linear Terms

The two trajectories were plotted for particles with initial velocities of 1 m/s launched at an angle of $\pi/4$ to the horizontal and can be seen in Figure 5.9. The particle which was not subject to air resistance clearly travelled much further than the one which was subject to it.

As shown in Figure 5.10 the horizontal displacement depends heavily on the initial launching angle and the angle which provides the maximum horizontal displacement is 0.56 rad.

As seen in the graph in Figure 5.11 the optimum launching angle for a particle generally increases with mass and reaches a value of 0.79 rad. This is what $\frac{\pi}{4}$ rounds to to two decimal places. $\frac{\pi}{4}$ is the optimum angle for launching an object in a vacuum, this may suggest that at higher masses the optimum launching angle approaches that of a particle launched in a vacuum.

3.4 Projectile Motion Under the Action of Air Resistance

A large number of trajectories were plotted, however only a small number of plots taken to be demonstrative of certain phenomena are included in this report.

In no case is the horizontal displacement or maximum vertical displacement of the trajectory in a vacuum less than that of either of the other trajectories.

In most cases with a small mass, the trajectory which follows the quadratic rule has a larger horizontal displacement than the trajectory followed by the trajectory following the linear rule, this is seen the plot in Figure 5.12.

For more massive particles the relative positions of the trajectories given by the quadratic and linear functions can switch depending on the initial velocity, as illustrated in the graphs in Figure 5.13.

In some cases the trajectory given by the quadratic function can reach a lower maximum height than the that given by the linear function and yet still travel a greater horizontal distance. This is illustrated in Figure 5.14

When the mass of the particle increases more for lower initial speeds both trajectories subject to air resistance approximate the trajectory in vacuum very well, it is only at higher velocities that they deviate significantly from the trajectory in vacuum as illustrated in the plots in Figure 5.15

Finally, once the mass increases enough, above at least 10^{-4} kg there is no visible difference in the plots of the trajectories subject to air resistance as compared to the trajectory in a vacuum. This is shown in the two graphs in Figure 5.16. It should be noted at this point that the diameter of the spherical dust grain is being held constant.

4 Conclusions

When the product of the velocity and diameter, VD, is below approximately $10^{-5}\ m^2s^{-1}$ only the linear term is relevant in the calculation of the air resistance. Between values of approximately $10^{-5}\ m^2s^{-1}$ and $10^{-3}\ m^2s^{-1}$ for VD both terms are relevant, and above $10^{-3}\ m^2s^{-1}$ only the quadratic term is relevant.

At least when using the linear approximation for the air resistance a decrease in mass has the same effect on the graph of the velocity against time as an increase in the coefficient of resistance, and vice-versa.

For the conditions used in these programs the numerical approximation described in Eq. 1.9 is a very good approximation to the analytical result described in Eq. 1.13

Above a mass of approximately 10^{-5} kg particles of uniform cross-section all fall with the same acceleration as is generally expected. However to conclusively say this for all objects more investigation must be given to objects of different shapes and different cross-sectional areas.

The optimum launching angle for a particle generally increases with mass and appears to approach a value of $\frac{\pi}{4}$, the optimum launching angle for a particle in a vacuum.

For as an object becomes denser its trajectories in vacuum and when subject to air resistance become practically indistinguishable.

These numerical methods appear to be very good methods for modelling the trajectories of different spherical bodies. With further adaptation they could be used to model the trajectory of objects of different shapes also.

5 Appendices

5.1 Appendix A: Code

5.1.1 Scaling of Air Resistance with Velocity

```
# -*- coding: utf-8 -*-
"""
Created on Tue Mar 14 14:01:25 2023

@author: wattersb
"""
import numpy as np
import matplotlib.pyplot as plt

# f(V) = bV + cV^2
# b = BD
# c = CD^2

B = 1.6*10**(-4) # Ns/m^2
C = 0.25 # Ns^2/m^4

# bV
def b(VD):
```

```
    return B * VD

# cV^2
def c(VD):
    return C * VD**2

# f(V)
def f(V):
    return b(V) + c(V)


############################
#                          #
#   RANGE OF VD VALUES     #
#                          #
############################


##################
# BOTH RELEVANT #
##################

x = np.arange(0, 10**(-3), 10**(-6))

# PLOT bV AND cV AGAINST VD
plt.title("Graph of the contributions to f(V) against VD")
plt.plot(x, b(x), label="bV", color="blue")
plt.plot(x, c(x), label="cV$^2$", color="red")
plt.legend()
plt.xlabel("V $\\times$ D")
plt.ylabel("bV, cV$^2$")
plt.savefig("images/Exercise1A_together_both.png", dpi=300)
plt.show()

plt.title("Graph of the B against VD")
plt.plot(x, b(x), label="bV", color="blue")
plt.legend()
plt.xlabel("V $\\times$ D")
plt.ylabel("bV")
plt.savefig("images/Exercise1A_B_both.png", dpi=300)
plt.show()

plt.title("Graph of the C against VD")
plt.plot(x, c(x), label="cV$^2$", color="red")
plt.legend()
plt.xlabel("V $\\times$ D")
plt.ylabel("cV$^2$")
plt.savefig("images/Exercise1A_C_both.png", dpi=300)
plt.show()

###################
# LINEAR RELEVANT #
```

```
####################

x = np.arange(0, 10**(-5), 10**(-8))

# PLOT bV AND cV AGAINST VD
plt.title("Graph of the contributions to f(V) against VD")
plt.plot(x, b(x), label="bV", color="blue")
plt.plot(x, c(x), label="cV$^2$", color="red")
plt.legend()
plt.xlabel("V $\\times$ D")
plt.ylabel("bV, cV$^2$")
plt.savefig("images/Exercise1A_together_linear.png", dpi=300)
plt.show()

plt.title("Graph of the B against VD")
plt.plot(x, b(x), label="bV", color="blue")
plt.legend()
plt.xlabel("V $\\times$ D")
plt.ylabel("bV")
plt.savefig("images/Exercise1A_B_linear.png", dpi=300)
plt.show()

plt.title("Graph of the C against VD")
plt.plot(x, c(x), label="cV$^2$", color="red")
plt.legend()
plt.xlabel("V $\\times$ D")
plt.ylabel("cV$^2$")
plt.savefig("images/Exercise1A_C_linear.png", dpi=300)
plt.show()

######################
# QUADRATIC RELEVANT #
######################

x = np.arange(0, 10**(-1), 10**(-3))

# PLOT bV AND cV AGAINST VD
plt.title("Graph of the contributions to f(V) against VD")
plt.plot(x, b(x), label="bV", color="blue")
plt.plot(x, c(x), label="cV$^2$", color="red")
plt.legend()
plt.xlabel("V $\\times$ D")
plt.ylabel("bV, cV$^2$")
plt.savefig("images/Exercise1A_together_quadratic.png", dpi=300)
plt.show()

plt.title("Graph of the B against VD")
plt.plot(x, b(x), label="bV", color="blue")
plt.legend()
plt.xlabel("V $\\times$ D")
plt.ylabel("bV")
plt.savefig("images/Exercise1A_B_quadratic.png", dpi=300)
```

```
plt.show()

plt.title("Graph of the C against VD")
plt.plot(x, c(x), label="cV$^2$", color="red")
plt.legend()
plt.xlabel("V $\\times$ D")
plt.ylabel("cV$^2$")
plt.savefig("images/Exercise1A_C_quadratic.png", dpi=300)
plt.show()


########################
#                      #
#   DIFFERENT CASES    #
#                      #
########################


#############
# Baseball #
#############

D = 0.07 # m
V = 5 # m/s

print("\n Baseball:")
print("Linear term:" + str(b(V)))
print("Quadratic term:" + str(c(V)))
print("Total:" + str(f(V)))

#############
# Oil drop #
#############

D = 1.5*10**(-6) # m
V = 5*10**(-5) # m/s

print("\n Oil drop:")
print("Linear term:" + str(b(V)))
print("Quadratic term:" + str(c(V)))
print("Total:" + str(f(V)))

#############
# Raindrop #
#############

D = 10**(-3) # m
V = 1 # m/s

print("\n Rain drop:")
print("Linear term:" + str(b(V)))
print("Quadratic term:" + str(c(V)))
```

```
print("Total:" + str(f(V)))
```

5.1.2 Vertical Motion Under the Action of Air Resistance

```
# -*- coding: utf-8 -*-
"""
Created on Tue Mar 14 14:28:51 2023

@author: wattersb
"""
import numpy as np
import matplotlib.pyplot as plt

# f(V) = bV + cV^2
# b = BD
# c = CD^2

# INITIAL QUANTITIES
B = 1.6*10**(-4) # Ns/m^2
D = 10**(-4) # m
b = B*D
rho = 2*10**3 # kg/m^3
m = 4/3 * np.pi * (D/2)**3 * rho # kg
V = 0 # m/s
t = 0 # s
g = 9.81 # m/s^2

dt = 0.0001

# MAXIMUM TIME
tmax = 1 # s

# LIST OF VELOCITIES AND TIMES
V_list = np.array([])
t_list = np.array([])

while t < tmax:
    dV = -g*dt - b/m * V * dt
    V = V + dV
    t = t + dt

    V_list = np.append(V_list, V)
    t_list = np.append(t_list, t)

plt.plot(t_list, V_list)
plt.title("Graph of V$_y$ against time")
plt.xlabel("Time (s)")
plt.ylabel("V$_y$ (m/s)")
plt.savefig("images/Exercise2B.png", dpi=300)
plt.show()


############################
```

```
#                              #
#    FOR DIFFERENT MASSES      #
#                              #
################################


# INITIAL MASS
m = 10**(-10) # kg

# MAXIMUM TIME
tmax = 10 # s

while m <= 10:

    # RESET
    V = 0 # m/s
    t = 0 # s
    V_list = np.array([])
    t_list = np.array([])

    while t < tmax:
        dV = -g*dt - b/m * V * dt
        V = V + dV
        t = t + dt

        V_list = np.append(V_list, V)
        t_list = np.append(t_list, t)

    plt.plot(t_list, V_list)
    plt.title("Graph of V$_y$ against time for mass =" + str(m) + "kg")
    plt.xlabel("Time (s)")
    plt.ylabel("V$_y$ (m/s)")
    plt.savefig("images/Exercise2C_mass"+str(m)+".png", dpi=300)
    plt.show()

    m = m * 10


###################################
#                                 #
#    FOR DIFFERENT RESISTANCES    #
#                                 #
###################################


# INITIAL MASS
m = 4/3 * np.pi * (D/2)**3 * rho # kg

# INITIAL RESISTANCE
b = 1.6*10**(-6)

while b >= 1.6*10**(-15):
```

```
102
103     # RESET
104     V = 0 # m/s
105     t = 0 # s
106     V_list = np.array([])
107     t_list = np.array([])
108
109     while t < tmax:
110         dV = -g*dt - b/m * V * dt
111         V = V + dV
112         t = t + dt
113
114         V_list = np.append(V_list, V)
115         t_list = np.append(t_list, t)
116
117     plt.plot(t_list, V_list)
118     plt.title("Graph of V$_y$ against time for resistance =" + str(b) +
    "N s/m")
119     plt.xlabel("Time (s)")
120     plt.ylabel("V$_y$ (m/s)")
121     plt.savefig("images/Exercise2C_resistance"+str(b)+".png", dpi=300)
122     plt.show()
123
124     b = b / 10
125
126
127 ##################################################
128 #                                                #
129 #   COMPARE NUMERICAL AND ANALYTICAL RESULTS     #
130 #                                                #
131 ##################################################
132
133
134 ####################
135 # NUMERICAL RESULT #
136 ####################
137
138 # LIST OF VELOCITIES AND TIMES
139 V_list = np.array([])
140 t_list = np.array([])
141
142 tmax = 1 # s
143
144 # RESET V, TIME
145 V = 0 # m/s
146 t = 0 # s
147 b = B*D
148 m = 4/3 * np.pi * (D/2)**3 * rho # kg
149
150 while t < tmax:
151     dV = -g*dt - b/m * V * dt
152     V = V + dV
```

```
    t = t + dt

    V_list = np.append(V_list, V)
    t_list = np.append(t_list, t)

plt.plot(t_list, V_list)
plt.title("Graph of V$_y$ against time - numerical result")
plt.xlabel("Time (s)")
plt.ylabel("V$_y$ (m/s)")
plt.savefig("images/Exercise2D_numerical.png", dpi=300)
plt.show()

######################
# ANALYTICAL RESULT #
######################

# RESET VARIABLES
V = 0 # m/s
b = B*D
m = 4/3 * np.pi * (D/2)**3 * rho # kg

# LIST OF TIME VALUES
t = np.arange(0, 1, 0.0001)

def Vy(t):
    return m * g / b * (np.exp(-b*t/m) - 1)

plt.plot(t, Vy(t))
plt.title("Graph of V$_y$ against time - analytical result")
plt.xlabel("Time (s)")
plt.ylabel("V$_y$ (m/s)")
plt.savefig("images/Exercise2D_analytical.png", dpi=300)
plt.show()

#########
# ERROR #
#########

plt.plot(t_list, Vy(t_list)-V_list)
plt.title("Graph of the error from the numerical result against time")
plt.xlabel("Time (s)")
plt.ylabel("Error")
plt.savefig("images/Exercise2D_error.png", dpi=300)
plt.show()


############################
#                          #
#   TIME FALLING V MASS    #
#                          #
############################

```

```
# SET MASS
m = 10**(-8)

# SET LISTS
mass_list = np.array([])
time_list = np.array([])

while m <= 100:
    # RESET TIME, SPEED, HEIGHT
    t = 0 # s
    Y = 5
    V = 0

    while Y >= 0:
        dV = -g*dt - b/m * V * dt
        V = V + dV
        t = t + dt
        Y += V*dt

    # APPEND TO ARRAYS
    time_list = np.append(time_list, t)
    mass_list = np.append(mass_list, m)

    # INCREASE MASS
    m = m * 2

plt.plot(np.log10(mass_list), time_list)
plt.scatter(np.log10(mass_list), time_list)
plt.title("Graph of the time taken to fall 5 m against the mass of a
   particle")
plt.ylabel("Time (s)")
plt.xlabel("log$_{10}$ mass")
plt.savefig("images/Exercise2E.png", dpi=300)
plt.show()
```

5.1.3 Projectile Motion Under the Action of Air Resistance - Restricted to Linear Terms

```
# -*- coding: utf-8 -*-
"""
Created on Tue Mar 21 16:20:39 2023

@author: wattersb
"""
import numpy as np
import matplotlib.pyplot as plt

# INITIAL QUANTITIES
B = 1.6*10**(-4) # Ns/m^2
D = 10**(-4) # m
b = B*D
rho = 2*10**3 # kg/m^3
m = 4/3 * np.pi * (D/2)**3 * rho # kg
```

```
g = 9.81 # m/s^2


################################
#                              #
#   TRAJECTORIES OF OBJECTS    #
#                              #
################################


####################
# DEFINE FUNCTIONS #
####################

# POSITION WITHOUT RESISTANCE
def Pos(x0, y0, v0, theta, m): # INITIAL X, INITIAL Y, INITIAL SPEED, INITIAL ANGLE, MASS

    t = 0
    dt = 0.0001

    X = x0
    Y = y0
    Vx = v0 * np.cos(theta)
    Vy = v0 * np.sin(theta)

    # LIST OF POSITIONS, VELOCITIES AND TIMES
    Vx_list = np.array([])
    X_list = np.array([])
    Vy_list = np.array([])
    Y_list = np.array([])
    t_list = np.array([])

    # WHILE ABOVE THE GROUND
    while Y >= 0:
        # UPDATE Y STUFF
        dVy = -g*dt
        Vy = Vy + dVy
        Y += Vy*dt

        Vy_list = np.append(Vy_list, Vy)
        Y_list = np.append(Y_list, Y)

        # UPDATE X STUFF
        X += Vx*dt

        Vx_list = np.append(Vx_list, Vx)
        X_list = np.append(X_list, X)

        # UPDATE TIME
        t = t + dt
        t_list = np.append(t_list, t)
```

```
67
68      return Vx_list, X_list, Vy_list, Y_list, t_list
69
70  # POSITION WITH AIR RESISTANCE
71  def PosResistance(x0, y0, v0, theta, m): # INITIAL X, INITIAL Y,
        INITIAL SPEED, INITIAL ANGLE, MASS
72
73      t = 0
74      dt = 0.0001
75
76      X = x0
77      Y = y0
78      Vx = v0 * np.cos(theta)
79      Vy = v0 * np.sin(theta)
80
81      # LIST OF POSITIONS, VELOCITIES AND TIMES
82      Vx_list = np.array([])
83      X_list = np.array([])
84      Vy_list = np.array([])
85      Y_list = np.array([])
86      t_list = np.array([])
87
88      # WHILE ABOVE THE GROUND
89      while Y >= 0:
90          # UPDATE Y STUFF
91          dVy = -g*dt - b/m * Vy * dt
92          Vy = Vy + dVy
93          Y += Vy*dt
94
95          Vy_list = np.append(Vy_list, Vy)
96          Y_list = np.append(Y_list, Y)
97
98          # UPDATE X STUFF
99          dVx = - b/m * Vx * dt
100         Vx = Vx + dVx
101         X += Vx*dt
102
103         Vx_list = np.append(Vx_list, Vx)
104         X_list = np.append(X_list, X)
105
106         # UPDATE TIME
107         t = t + dt
108         t_list = np.append(t_list, t)
109
110     return Vx_list, X_list, Vy_list, Y_list, t_list
111
112 # GET THE VALUES
113 Vx_list_res, X_list_res, Vy_list_res, Y_list_res, t_list_res =
        PosResistance(0, 0, 1, np.pi/4, m)
114 Vx_list, X_list, Vy_list, Y_list, t_list = Pos(0, 0, 1, np.pi/4, m)
115
116 plt.plot(X_list_res, Y_list_res, label="Trajectory with air resistance"
```

```
    )
plt.plot(X_list, Y_list, label="Trajectory without air resistance")
plt.legend()
plt.title("Graph of Y against X")
plt.ylabel("Y")
plt.xlabel("X")
plt.savefig("images/Exercise3A.png", dpi=300)
plt.show()


################################################
#                                              #
#   TEST DIFFERENT ANGLES FOR CONSTANT MASS    #
#                                              #
################################################


# SET THE MASS
m = 4/3 * np.pi * (D/2)**3 * rho # kg

# SET LISTS OF ANGLES AND HORIZONTAL DISTANCES
angle = np.arange(0,np.pi/2,0.01)
x_list = np.array([])

for i in range(len(angle)):
    Vx_list_res, X_list_res, Vy_list_res, Y_list_res, t_list_res =
    PosResistance(0, 0, 1, angle[i], m)
    x_list = np.append(x_list, max(X_list_res))

plt.plot(angle, x_list)
plt.title("Graph of horizontal displacement against initial angle")
plt.ylabel("Horizontal displacement (m)")
plt.xlabel("Initial angle (rad)")

# FIND THE ANGLE OF MAXIMUM DISPLACEMENT AND PRINT AND PLOT IT
maxX = np.where(x_list == max(x_list))
maxAngle = angle[maxX]
print("The maximum displacement occurs with an initial angle of " + str
    (maxAngle) + " radians")
plt.scatter(maxAngle, x_list[maxX], color='black', label="Angle of
    Maximum Displacement")
plt.legend()
plt.savefig("images/Exercise3B_constant_mass.png", dpi=300)
plt.show()


################################################
#                                              #
#   TEST DIFFERENT ANGLES FOR CHANGING MASS    #
#                                              #
################################################

```

```
165
166 # LIST OF OPTIMUM ANGLES
167 theta_optimum = np.array([])
168
169 # INITIAL MASS
170 m = 10**(-9) # kg
171
172 mass_list = np.array([])
173
174 while m <= 10**(-3):
175
176     # RESET LIST OF X VALUES
177     x_list = np.array([])
178
179     # CHECK DIFFERENT ANGLES FOR GIVEN MASS
180     for i in range(len(angle)):
181         Vx_list_res, X_list_res, Vy_list_res, Y_list_res, t_list_res =
    PosResistance(0, 0, 1,angle[i], m)
182         x_list = np.append(x_list, max(X_list_res))
183
184     # MAX ANGLE FOR GIVEN MASS
185     maxX = np.where(x_list == max(x_list))
186     maxAngle = angle[maxX]
187     theta_optimum = np.append(theta_optimum, maxAngle)
188
189     # RECORD MASS
190     mass_list = np.append(mass_list, m)
191
192     # INCREASE THE MASS
193     m = m*2
194
195 plt.plot(np.log10(mass_list), theta_optimum)
196 plt.plot(np.log10(mass_list), np.pi/4 * np.ones(len(mass_list)), color=
    "black", label="$\\theta_0$ = $\\frac{\pi}{4}$")
197 plt.scatter(np.log10(mass_list), theta_optimum, s=5, label="Calculated
    points")
198 plt.title("Graph of optimum initial angle against mass")
199 plt.xlabel("log$_{10}$Mass")
200 plt.ylabel("Optimum initial angle (rad)")
201 plt.legend()
202 plt.savefig("images/Exercise3B_angles_for_masses.png", dpi=300)
203 plt.show()
```

5.1.4 Projectile Motion Under the Action of Air Resistance

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Mon Mar 27 21:27:34 2023
4
5 @author: wattersb
6 """
7 import numpy as np
8 import matplotlib.pyplot as plt
```

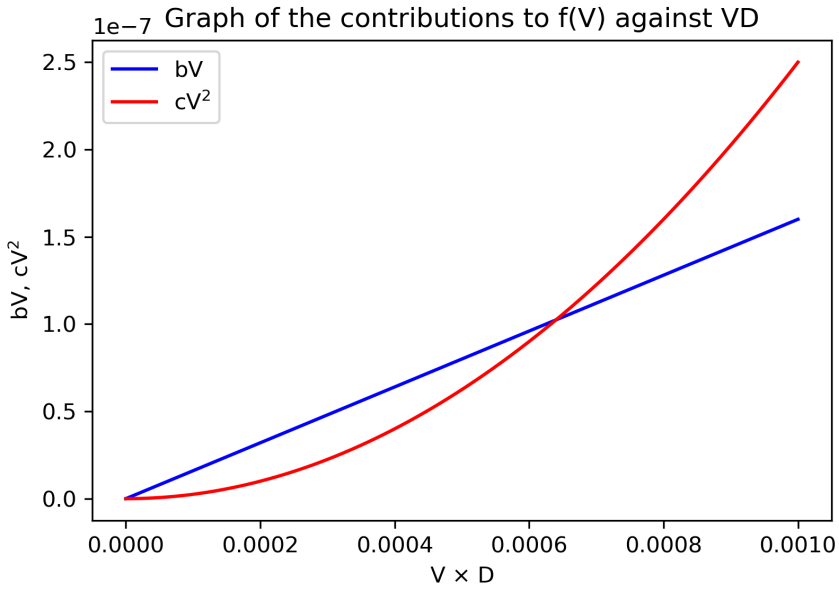
```
9
10 # INITIAL QUANTITIES
11 B = 1.6*10**(-4) # Ns/m^2
12 C = 0.25 # Ns^2/m^4
13 D = 10**(-4) # m
14 b = B*D
15 c = C*D**2
16 g = 9.81 # m/s^2
17
18
19 ########################
20 #                      #
21 #   DEFINE FUNCTIONS   #
22 #                      #
23 ########################
24
25
26 # POSITION WITH AIR RESISTANCE, QUADRATIC DEPENDENCE
27 def PosResistanceQuadratic(x0, y0, v0, theta, m): # INITIAL X, INITIAL
       Y, INITIAL SPEED, INITIAL ANGLE, MASS
28
29     t = 0
30     dt = 0.0001
31
32     X = x0
33     Y = y0
34     Vx = v0 * np.cos(theta)
35     Vy = v0 * np.sin(theta)
36
37     # LIST OF POSITIONS, VELOCITIES AND TIMES
38     Vx_list = np.array([])
39     X_list = np.array([])
40     Vy_list = np.array([])
41     Y_list = np.array([])
42     t_list = np.array([])
43
44     # WHILE ABOVE THE GROUND
45     while Y >= 0:
46         # UPDATE Y STUFF
47         dVy = -g*dt - c/m * np.sqrt(Vx**2+Vy**2) * Vx * dt
48         Vy = Vy + dVy
49         Y += Vy*dt
50
51         Vy_list = np.append(Vy_list, Vy)
52         Y_list = np.append(Y_list, Y)
53
54         # UPDATE X STUFF
55         dVx = -c/m * np.sqrt(Vx**2+Vy**2) * Vx * dt
56         Vx = Vx + dVx
57         X += Vx*dt
58
59         Vx_list = np.append(Vx_list, Vx)
```

```
        X_list = np.append(X_list, X)

        # UPDATE TIME
        t = t + dt
        t_list = np.append(t_list, t)

    return Vx_list, X_list, Vy_list, Y_list, t_list

# POSITION WITH AIR RESISTANCE, LINEAR DEPENDENCE
def PosResistanceLinear(x0, y0, v0, theta, m): # INITIAL X, INITIAL Y, INITIAL SPEED, INITIAL ANGLE, MASS

    t = 0
    dt = 0.0001

    X = x0
    Y = y0
    Vx = v0 * np.cos(theta)
    Vy = v0 * np.sin(theta)

    # LIST OF POSITIONS, VELOCITIES AND TIMES
    Vx_list = np.array([])
    X_list = np.array([])
    Vy_list = np.array([])
    Y_list = np.array([])
    t_list = np.array([])

    # WHILE ABOVE THE GROUND
    while Y >= 0:
        # UPDATE Y STUFF
        dVy = -g*dt - b/m * Vy * dt
        Vy = Vy + dVy
        Y += Vy*dt

        Vy_list = np.append(Vy_list, Vy)
        Y_list = np.append(Y_list, Y)

        # UPDATE X STUFF
        dVx = - b/m * Vx * dt
        Vx = Vx + dVx
        X += Vx*dt

        Vx_list = np.append(Vx_list, Vx)
        X_list = np.append(X_list, X)

        # UPDATE TIME
        t = t + dt
        t_list = np.append(t_list, t)

    return Vx_list, X_list, Vy_list, Y_list, t_list

# POSITION WITHOUT RESISTANCE
```

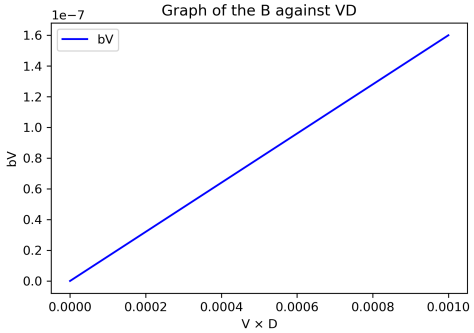
```
111 def Pos(x0, y0, v0, theta, m): # INITIAL X, INITIAL Y, INITIAL SPEED,
    INITIAL ANGLE, MASS
112
113     t = 0
114     dt = 0.0001
115
116     X = x0
117     Y = y0
118     Vx = v0 * np.cos(theta)
119     Vy = v0 * np.sin(theta)
120
121     # LIST OF POSITIONS, VELOCITIES AND TIMES
122     Vx_list = np.array([])
123     X_list = np.array([])
124     Vy_list = np.array([])
125     Y_list = np.array([])
126     t_list = np.array([])
127
128     # WHILE ABOVE THE GROUND
129     while Y >= 0:
130         # UPDATE Y STUFF
131         dVy = -g*dt
132         Vy = Vy + dVy
133         Y += Vy*dt
134
135         Vy_list = np.append(Vy_list, Vy)
136         Y_list = np.append(Y_list, Y)
137
138         # UPDATE X STUFF
139         X += Vx*dt
140
141         Vx_list = np.append(Vx_list, Vx)
142         X_list = np.append(X_list, X)
143
144         # UPDATE TIME
145         t = t + dt
146         t_list = np.append(t_list, t)
147
148     return Vx_list, X_list, Vy_list, Y_list, t_list
149
150
151 ##############################################
152 #                                            #
153 #   DIFFERENT MASSES, ANGLES, VELOCITIES     #
154 #                                            #
155 ##############################################
156
157 # DEFINE VARIABLES
158 m = 10**(-10) # kg
159 theta = 10 # degrees
160 v = 1 # m/s
161
```

```
162
163 while m <= 10**(-2):
164     while theta <= 90:
165         while v <= 100:
166             Vx_list_quad, X_list_quad, Vy_list_quad, Y_list_quad,
    t_list_quad = PosResistanceQuadratic(0, 0, v, theta * np.pi/180, m)
167             Vx_list_lin, X_list_lin, Vy_list_lin, Y_list_lin,
    t_list_lin = PosResistanceLinear(0, 0, v, theta * np.pi/180, m)
168             Vx_list, X_list, Vy_list, Y_list, t_list = Pos(0, 0, v,
    theta * np.pi/180, m)
169
170             plt.plot(X_list_quad, Y_list_quad, label="With air
    resistance, quadratic dependence")
171             plt.plot(X_list_lin, Y_list_lin, label="With air resistance
    , linear dependence")
172             plt.plot(X_list, Y_list, label="Without air resistance")
173             plt.legend()
174             plt.title("Graph of Y against X for m = " + str(m) + " kg,
    $\\theta_0$ = " + str(theta) + "$^{\circ}$, v$_0$ = " + str(v) + "
    m/s")
175             plt.ylabel("Y")
176             plt.xlabel("X")
177             plt.savefig("images/Exercise4_m"+str(m) + "_theta" + str(
    theta) + "_v" + str(v) + ".png")
178             plt.show()
179
180             if v < 10:
181                 v += 1
182             else:
183                 v += 10
184
185         theta += 10
186         v = 1
187
188     m = m*100
189     theta = 10
```

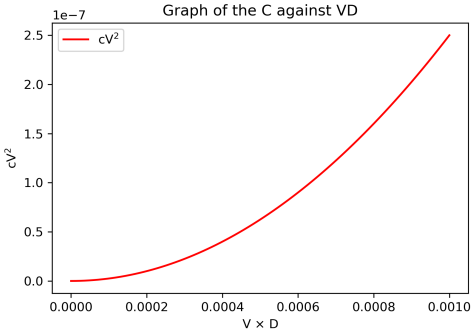
5.2 Appendix B:Graphs



(a) A graph of both $b(VD)$ and $c(VD)$

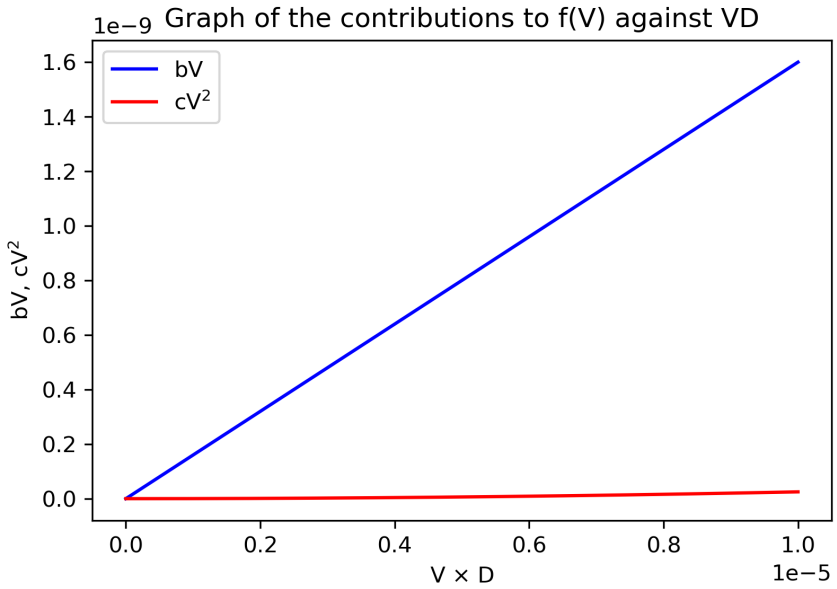


(b) $b(VD)$ graphed alone

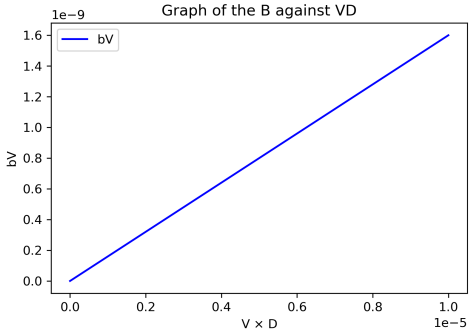


(c) $c(VD)$ graphed alone

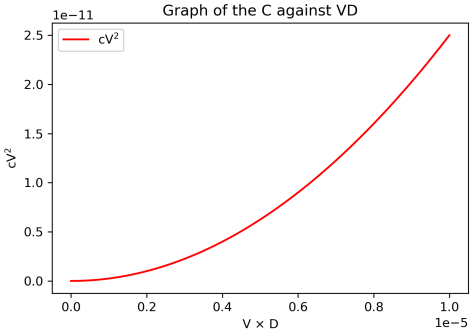
Figure 5.1: $b(VD)$ and $c(VD)$ for a range where both terms are significant



(a) A graph of both $b(VD)$ and $c(VD)$

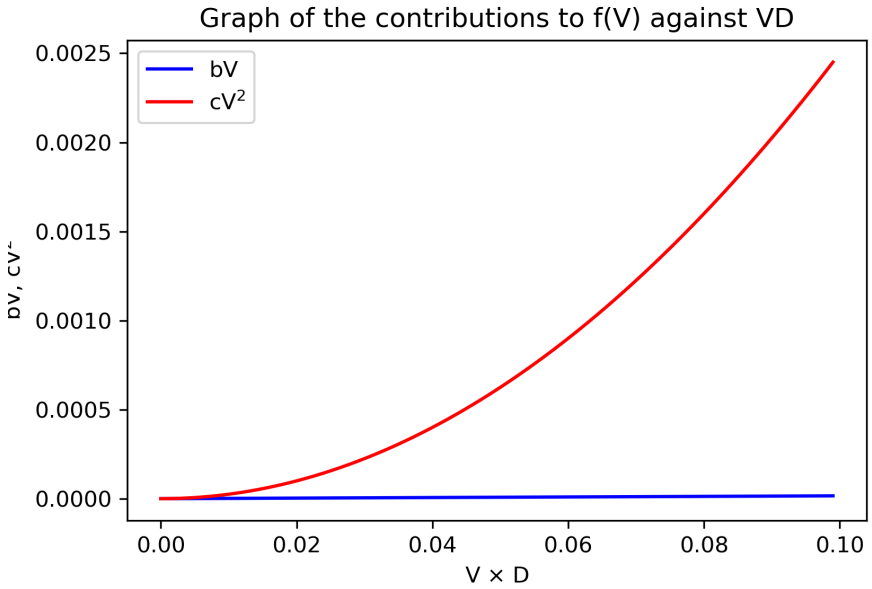


(b) $b(VD)$ graphed alone

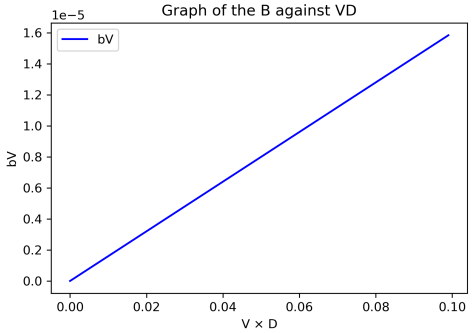


(c) $c(VD)$ graphed alone

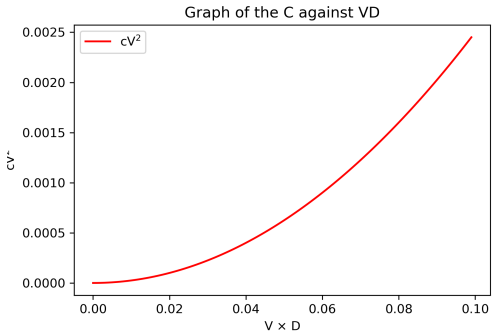
Figure 5.2: $b(VD)$ and $c(VD)$ for a range where only the linear term is significant



(a) A graph of both $b(VD)$ and $c(VD)$



(b) $b(VD)$ graphed alone



(c) $c(VD)$ graphed alone

Figure 5.3: $b(VD)$ and $c(VD)$ for a range where only the quadratic term is significant

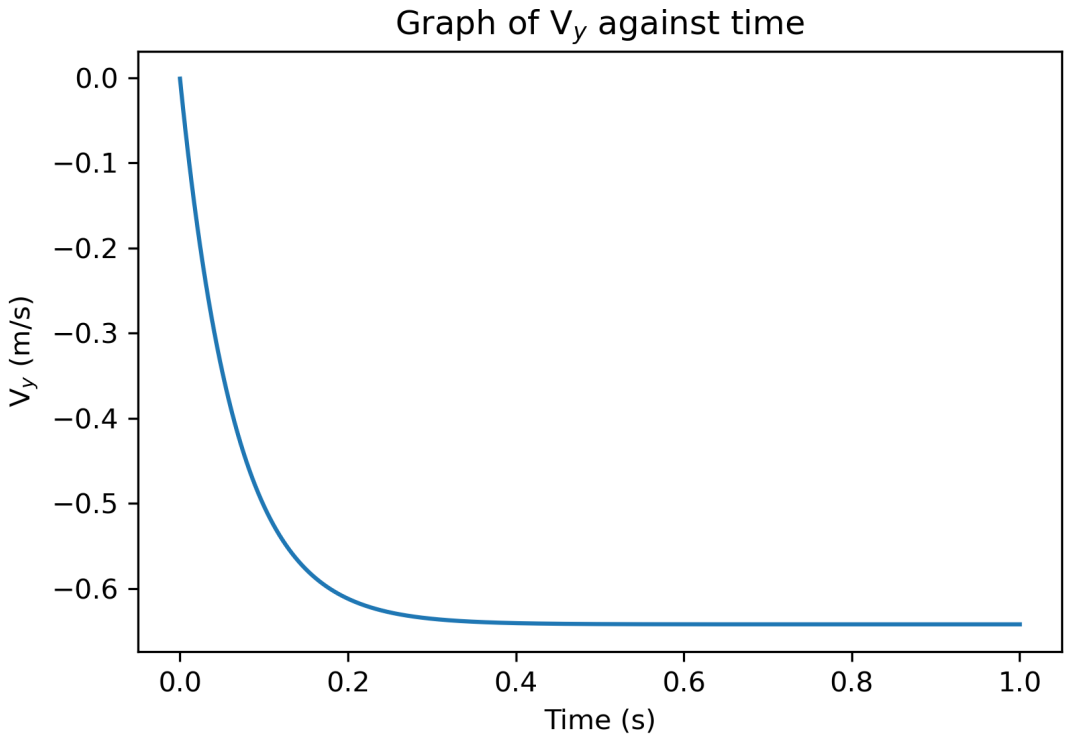
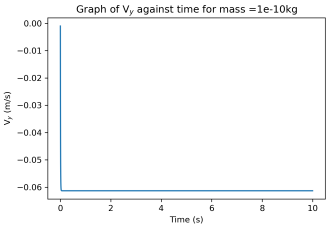
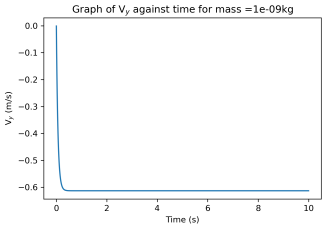


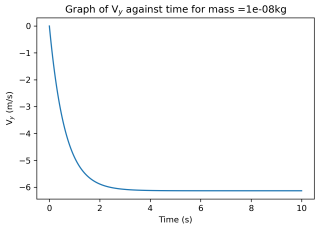
Figure 5.4: A graph of vertical velocity against time for a small spherical dust grain approximated to use only the linear dependence of air resistance.



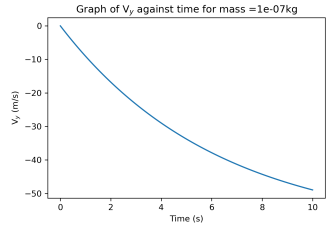
(a) V_y against time for m = 10^{-10} kg



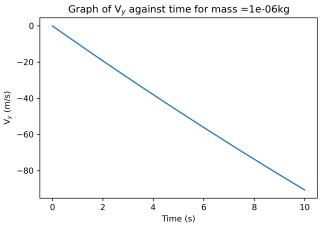
(b) V_y against time for m = 10^{-9} kg



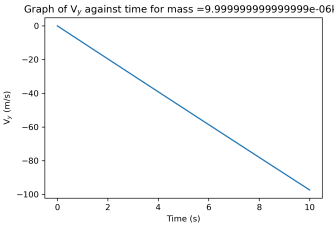
(c) V_y against time for m = 10^{-8} kg



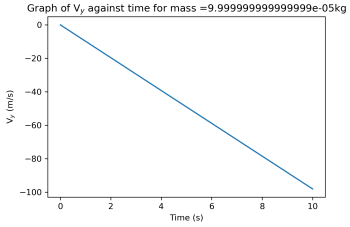
(d) V_y against time for m = 10^{-7} kg



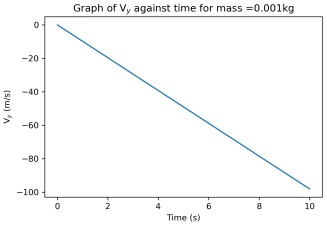
(e) V_y against time for m = 10^{-6} kg



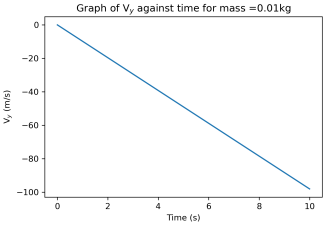
(f) V_y against time for m = 10^{-5} kg



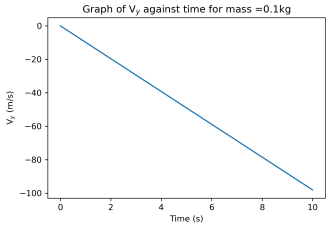
(g) V_y against time for m = 10^{-4} kg



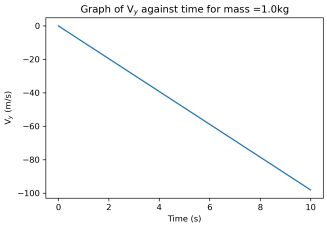
(h) V_y against time for m = 0.001 kg



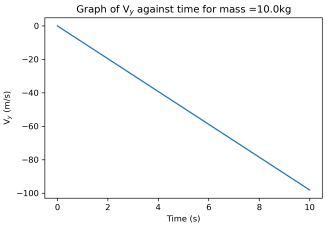
(i) V_y against time for m = 0.01 kg



(j) V_y against time for m =0.1 kg

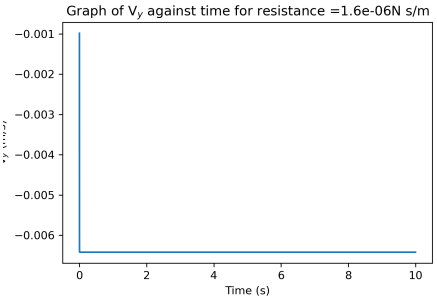


(k) V_y against time for m = 1.0 kg

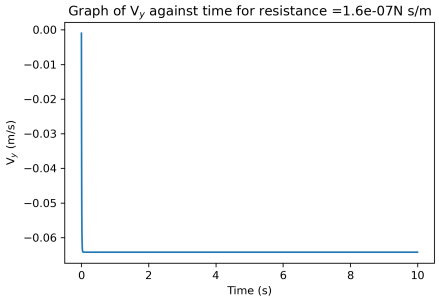


(l) V_y against time for m = 10.0 kg

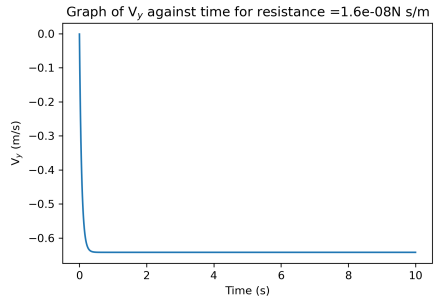
Figure 5.5: V_y against time for different masses and constant resistance b = 1.6×10^{-8} Nsm^{-2}.



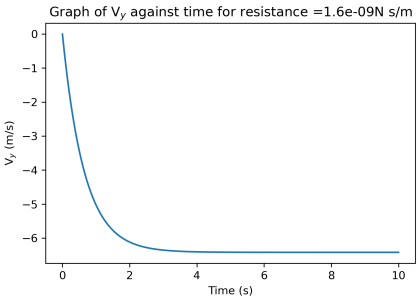
(a) V_y against time for b $= 1.6 \times 10^{-6}$ kg



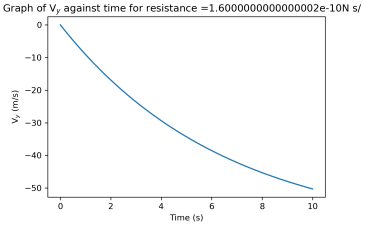
(b) V_y against time for b $= 1.6 \times 10^{-7}$ kg



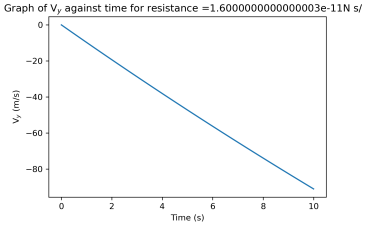
(c) V_y against time for b $= 1.6 \times 10^{-8}$ kg



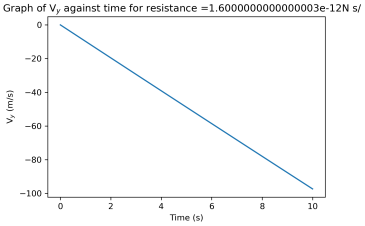
(d) V_y against time for b $= 1.6 \times 10^{-9}$ kg



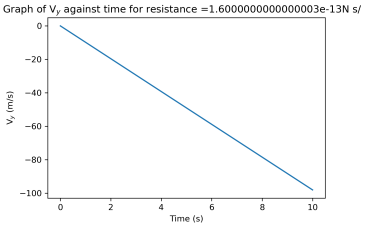
(e) V_y against time for b $= 1.6 \times 10^{-10}$ kg



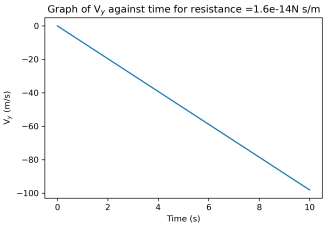
(f) V_y against time for b $= 1.6 \times 10^{-11}$ kg



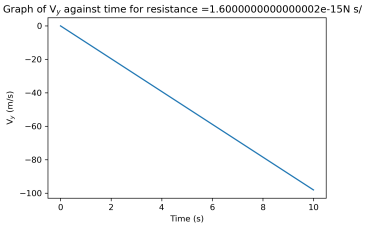
(g) V_y against time for b $= 1.6 \times 10^{-12}$ kg



(h) V_y against time for b $= 1.6 \times 10^{-13}$ kg

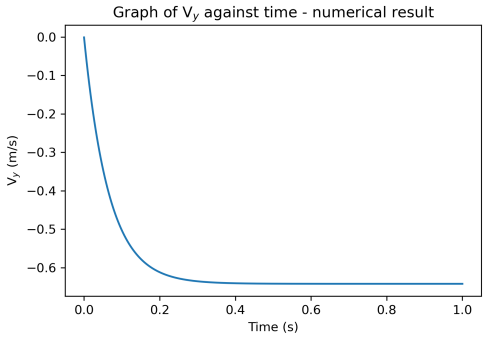


(i) V_y against time for b $= 1.6 \times 10^{-14}$ kg

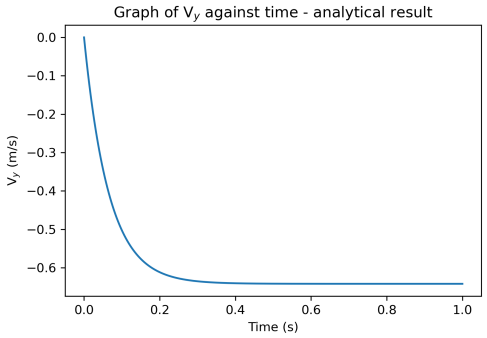


(j) V_y against time for b $= 1.6 \times 10^{-15}$ kg

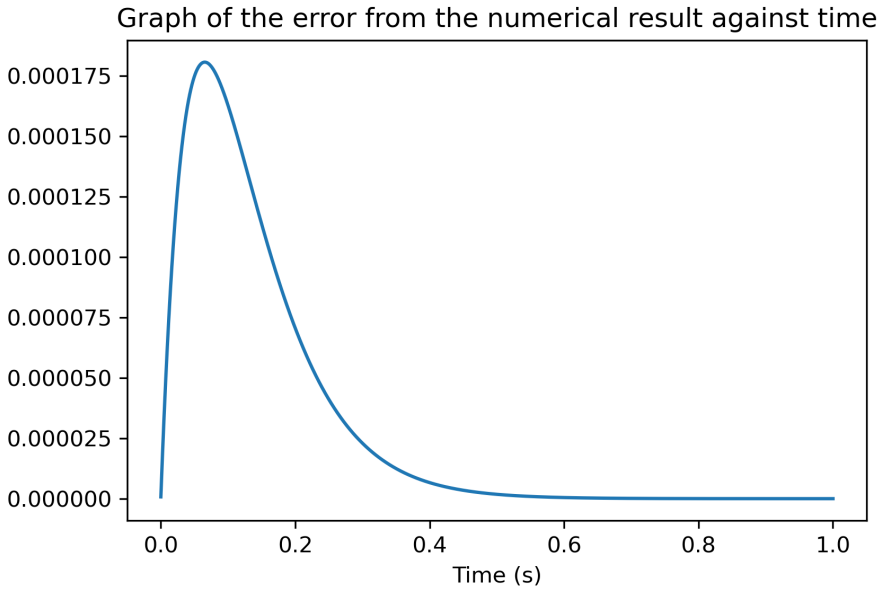
Figure 5.6: V_y against time for different resistances and constant mass, m $= 1.047 \times 10^{-9}$ kg.



(a) The numerical solution to the vertical velocity of the dust grain against time



(b) The analytical solution to the vertical velocity of the dust grain against time



(c) A graph of error between the numerical and analytical methods of determining the velocity of the particle

Figure 5.7: The numerical and analytical solutions of the vertical velocity of the dust grain along with the error between the two methods

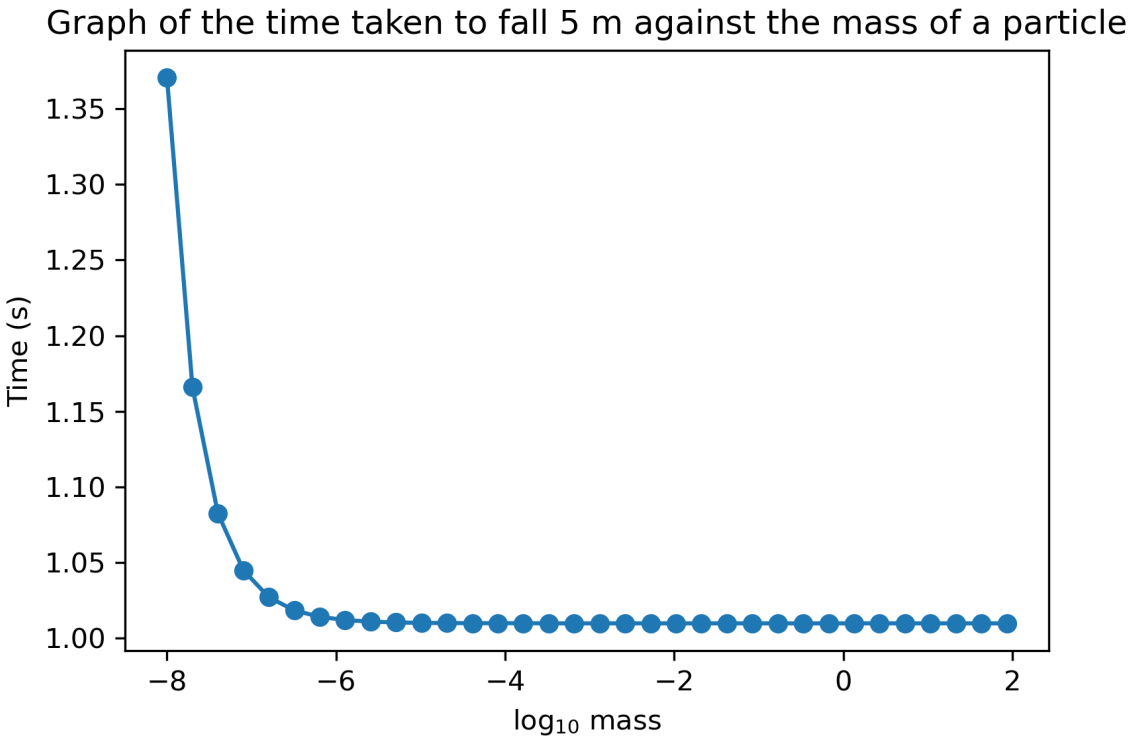


Figure 5.8: The graph of the time taken to fall 5 m against the mass of the falling object. Above approximately 10^{-5} kg all objects of uniform cross-section appear to fall at the same rate.

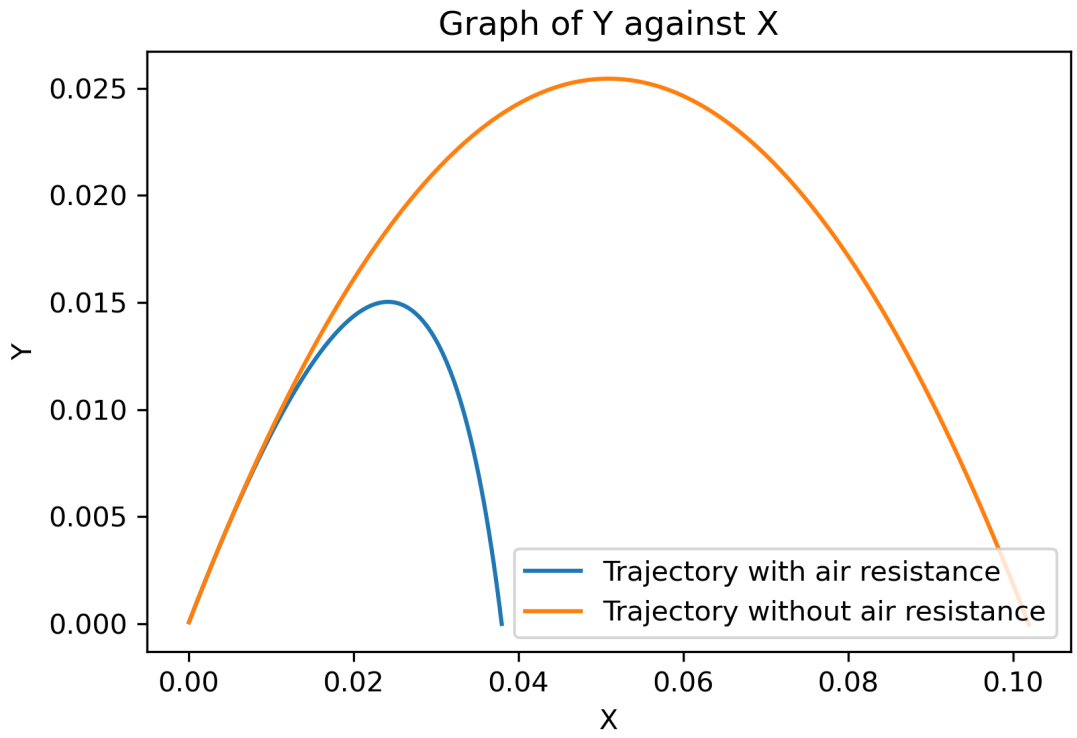


Figure 5.9: The trajectories of two particles both launched with an initial velocity of 1 m/s at an angle of $\pi/4$ to the horizontal.

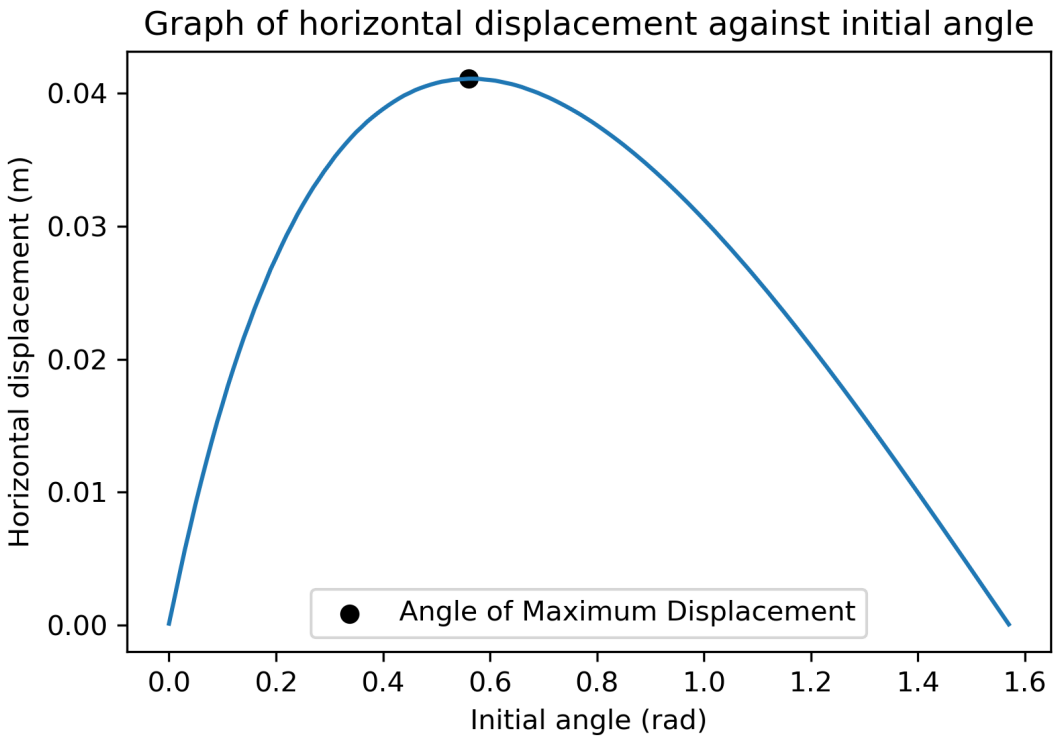


Figure 5.10: Horizontal displacement as a function of launching angle for a particle of constant mass 1.047×10^{-9} kg and launch speed of 1 m/s.

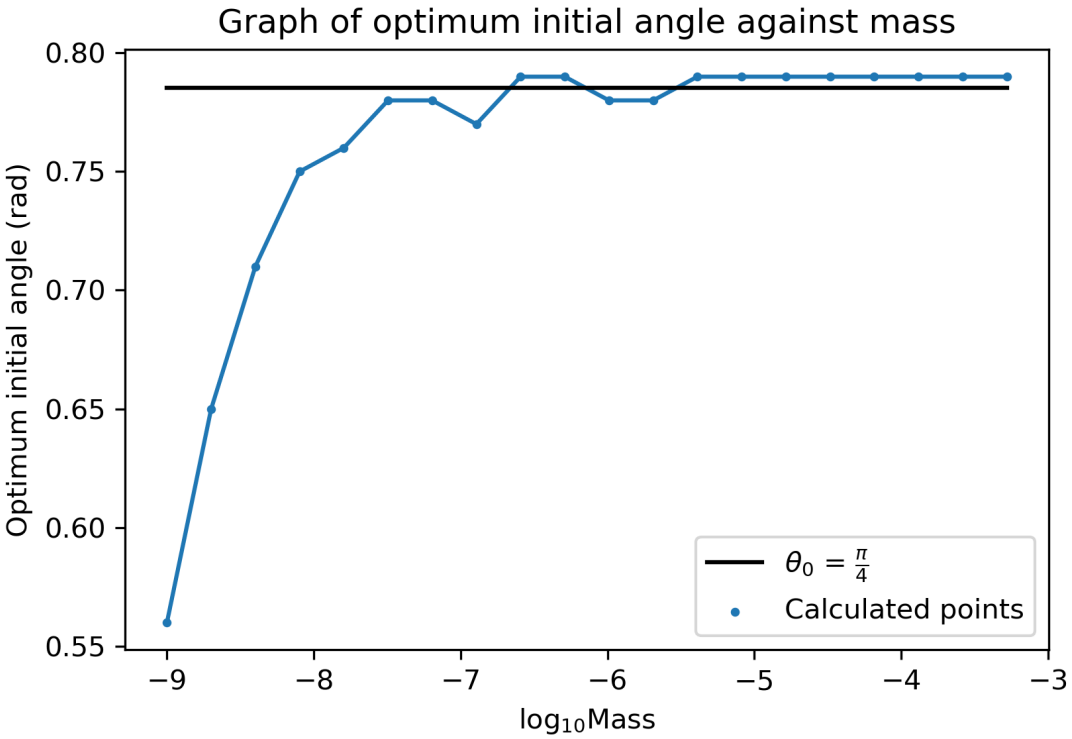


Figure 5.11: The optimum launching angle plotted against the mass of the projectile.

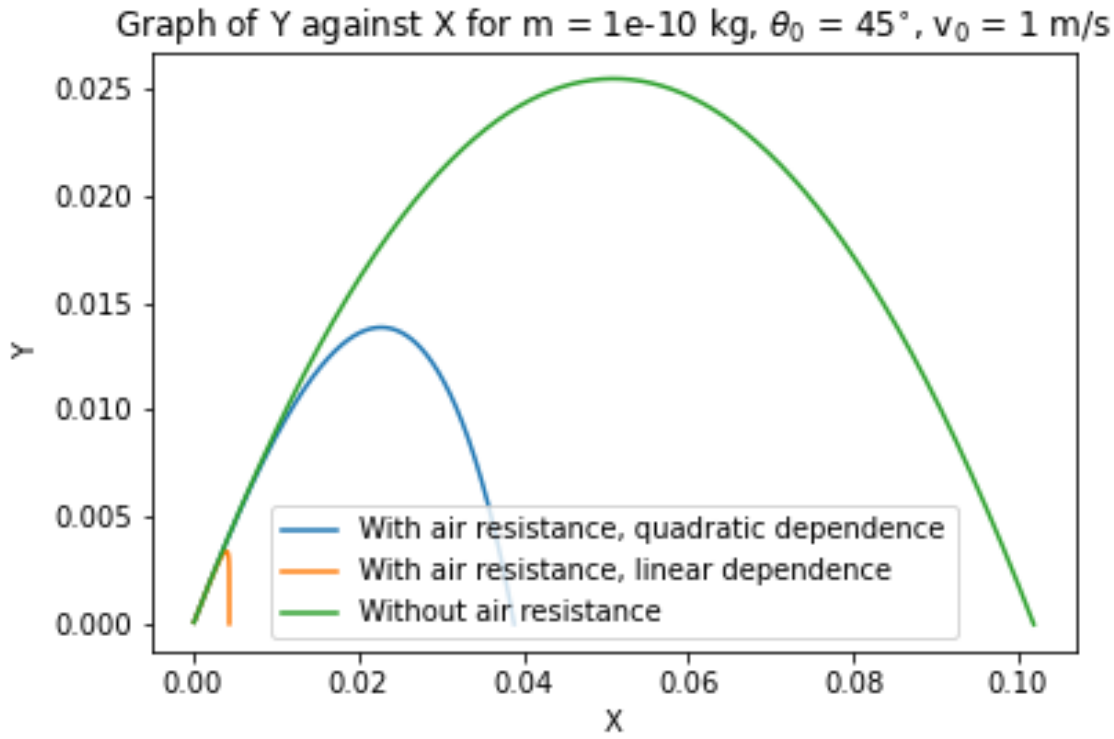
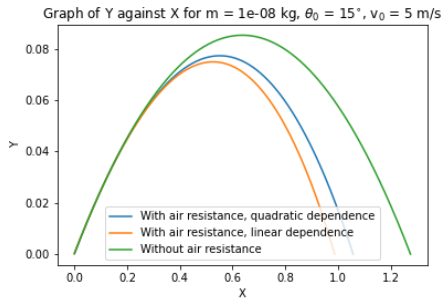
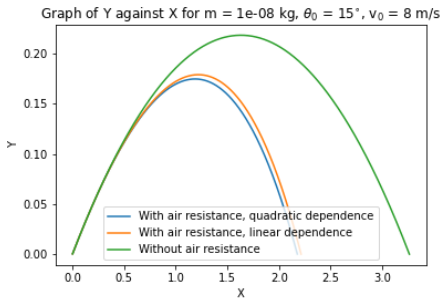


Figure 5.12: A plot of the different trajectories for a particle of mass 10^{-10} with an initial velocity of 1 m/s at an angle of 45° with the horizontal. The trajectory in a vacuum has the greatest horizontal displacement, followed by the trajectory following the quadratic rule and finally the trajectory following the linear rule.



(a) Initial speed of 5 m/s



(b) Initial speed of 8 m/s

Figure 5.13: Plots of the different trajectories for a particle of mass 10^{-8} with an initial angle of 15° with the horizontal with different starting speeds.

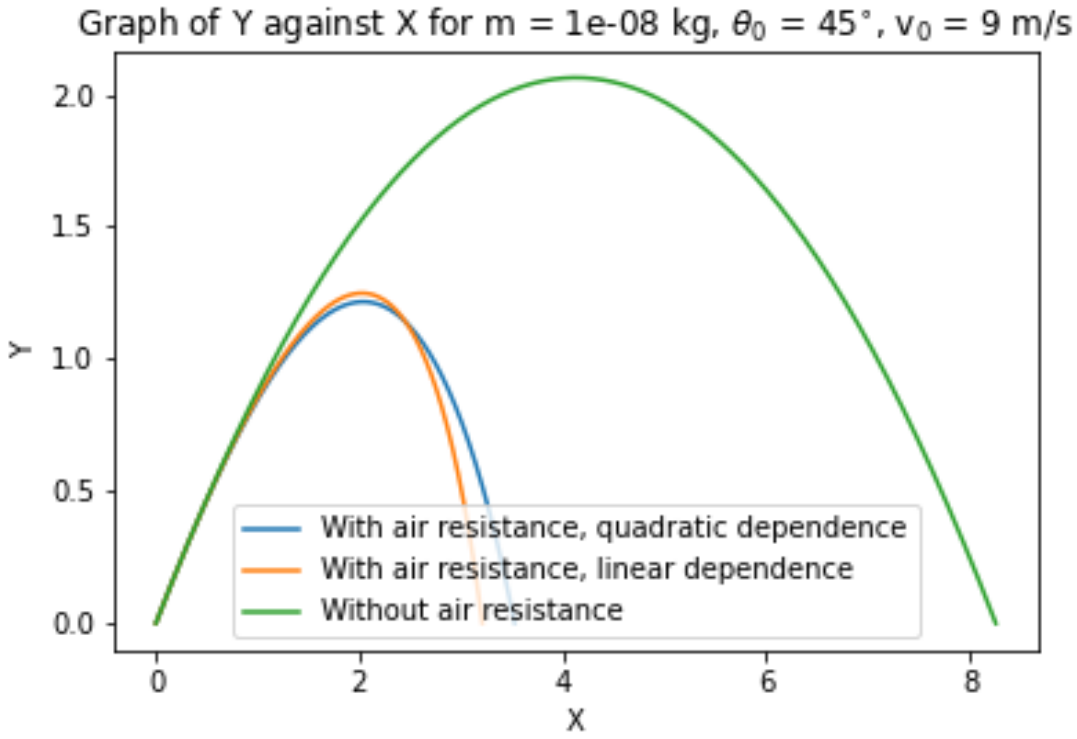
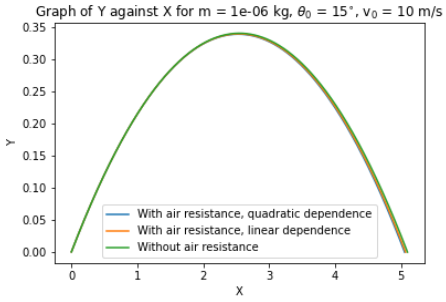
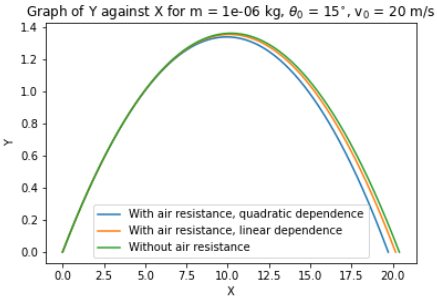


Figure 5.14: A plot of the different trajectories for a particle of mass 10^{-8} with an initial velocity of 9 m/s at an angle of 45° with the horizontal. Despite reaching a lower maximum height than the trajectory following the linear function for air resistance the trajectory following the quadratic function for air resistance has a greater maximum horizontal displacement.

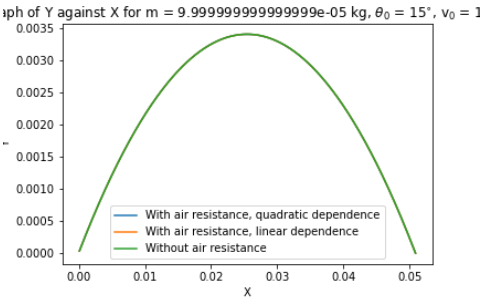


(a) Initial speed of 10 m/s

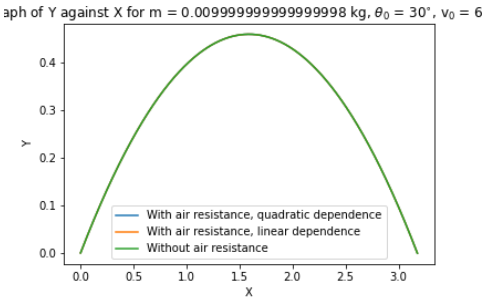


(b) Initial speed of 20 m/s

Figure 5.15: Plots of the different trajectories for a particle of mass 10^{-6} with an initial angle of 15° with the horizontal with different starting speeds. The larger initial speed causes a deviation of the two trajectories subject to air resistance from the trajectory in a vacuum



(a) Initial velocity of 1 m/s at 15° to the horizontal and a mass of 10^{-4}



(b) Initial velocity of 6 m/s at 30° to the horizontal and a mass of 10^{-2}

Figure 5.16: At larger masses there is no visible difference between the trajectories of the particles subject to air resistance and the particle in a vacuum.