Assignment_1_Numerical_321A

September 21, 2019

Spencer Plovie. Student Number: V00886526. PHYS321A Fall 2019, Section A01.

0.1 Assignment 1 Numerical Problem

Parachutist of m=70kg jumps from plane at an altitude of 32km above Earth's surface. Parachute doesn't open, so he is in free fall. Neglect horizontal motion and assume initial velocity is zero.

a) Calculate the time of fall (accurate to 1s) until ground impact, given no air resistance and a constant g value.

```
In [1]: #Importing all necessary libraries
        import numpy as np
        import scipy.integrate
        from scipy.integrate import odeint
        from scipy.interpolate import interp1d
        import matplotlib.pyplot as plt
```

In order to derive the formula to solve for time, we start with the conservation of energy:

$$E_p = E_k$$
$$E_p = mgh$$
$$E_k = \frac{1}{2}mv^2$$

Where v is the final velocity just before impact. Re-arranging to solve for v:

$$v = \sqrt{\frac{2E_k}{m}}$$

And plugging in E_p as a substitute for E_k:

$$v_f = \sqrt{\frac{2mgh}{m}}$$
$$v_f = \sqrt{2gh}$$

Then, using a kinematics equation, and knowing that v_o=0m/s:

$$
\begin{aligned}
v_f &= v_o + at \\
v_f &= gt \\
t &= \frac{v_f}{g} \\
t &= \sqrt{\frac{2h}{g}}
\end{aligned}
$$

```
In [2]: m = 70 #kilograms (kg)
        h = 32000 #meters (m)
        v_0 = 0 #meters/second (m/s)
        g = 9.81 #meters/second*second (m/s^2)

        E_p = m*g*h

        E_k = 0.5*m*np.power(v_0,2)

        t1 = np.sqrt((2*h)/(g))

        print("The time to fall until ground impact is %.0f seconds." % (t1))
```

```
The time to fall until ground impact is 81 seconds.
```

b) Calculate time of fall (accurate to 1s) until ground impact, given constant g and a force of air resistance given by $F(v) = -c_2 v|v|$, where c_2 is a constant, 0.5, in SI units for a falling man. In order to solve for time, in this case we must use a differential equation with a linear term for air resistance, $F(v)$. We start by using Newton's second law to find the forces relevant to this situation:

$$
\begin{aligned}
\sum F &= ma \\
mg - c_2 v|v| &= ma \\
\frac{dv}{dt} &= a \\
\frac{dv}{dt} &= g - \frac{c_2}{m} v|v|
\end{aligned}
$$

Since the first derivative of the position, h, (which we can consider to be y since we are neglecting horizontal motion in this case) would be the velocity, v, and the second derivative of y would be acceleration, a, we can simplify these to be seen as:

$$\dot{y} = v$$
$$\ddot{y} = a$$

In [3]:
```
y0 = [h, 0] #Paired in one list so that odeint will accept it
c_2 = 0.5
b = -c_2/m #Simplifying it

def dv_dt(y0,t,g,b):
    y,v = y0 #v = derivative of y, y,v are tuple so ODEINT can take it in
    dvdt = [-v, g-b*v*np.abs(v)] #First-order function (will not work if not FO)
    return dvdt

step=0.01
t = np.arange(0,900,step)
#t is in the range of [0,900] in steps of 0.01 (used later for a gradient)

odeint1 = odeint(dv_dt, y0, t, args=(g,-b)) #Returns position and velocity
interp = interp1d(odeint1[:,0], t)
#Interpolating the data to find when the parachutist hits the ground
#[] means slicing odeint1, ':' represents taking all data of the array
#'0' specifies only the position, y
```

In [4]:
```
print("The time to fall until ground impact is %.0f seconds." % (interp(0)))
#Find the index where y = 0
```

```
The time to fall until ground impact is 866 seconds.
```

c) Calculate time of fall (accurate to 1s) until ground impact, given that c_2 scales with atmospheric density as $c_2 = 0.5e^{-y/H}$, where $H = 8km$ is the scale height of the atmosphere and y is the height above ground. Furthermore, assume that g is no longer constant and is given by $g = 9.81/(1 + (y/R_e))^2 m/s^2$, where $R_e = 6370km$.

In [5]:
```
H = 8000 #meters
R_e = 6370000 #Radius of the Earth (meters)

def Dv_dt(y0,t,g,c_2,H,R_e):
    y,v = y0 #v = derivative of y, y,v are tuple so ODEINT can take it in
    c_2 = 0.5*np.exp(-y/H) #Modified c2
    g_mod = g/((1+y/R_e)**2) #Modified gravity
    dvdt = [v, -g_mod - (c_2/m)*v*np.abs(v)]
    #First-order function (will not work if not FO)
    return dvdt

odeint2 = odeint(Dv_dt, y0, t, args=(g,c_2,H,R_e))
interp2 = interp1d(odeint2[:,0], t)
```

```
In [6]: print("The time to fall until ground impact is %.0f seconds." % (interp2(0)))
```

```
The time to fall until ground impact is 381 seconds.
```

d) For case (c), plot acceleration, velocity, and altitude of the parachutist as a function of time. Explain why the acceleration becomes positive as the parachutist falls.

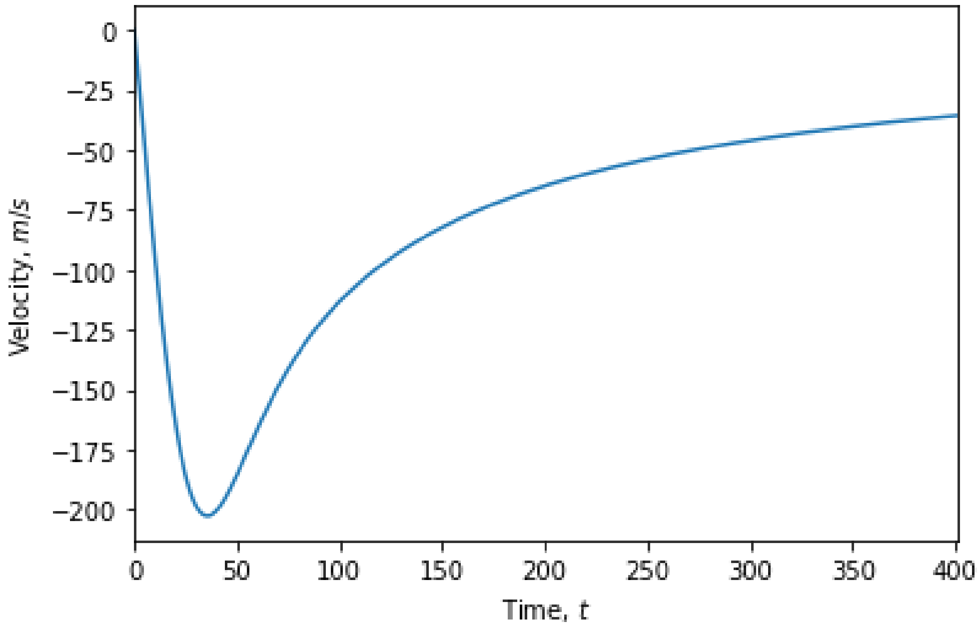
```
In [7]: plt.figure(1)
        plt.plot(t,odeint2[:,0])
        plt.title("(Fig. 1) Height of the Falling Parachutist until Ground Impact")
        plt.xlabel('Time, $t$')
        plt.ylabel('Position, $y$')
        plt.xlim([0,interp2(0)+20]) #Included to cut off the negative height (ignored)
        plt.ylim([0,h+400]) #Slightly above initial height to show full function clearly
```

```
Out[7]: (0, 32400)
```

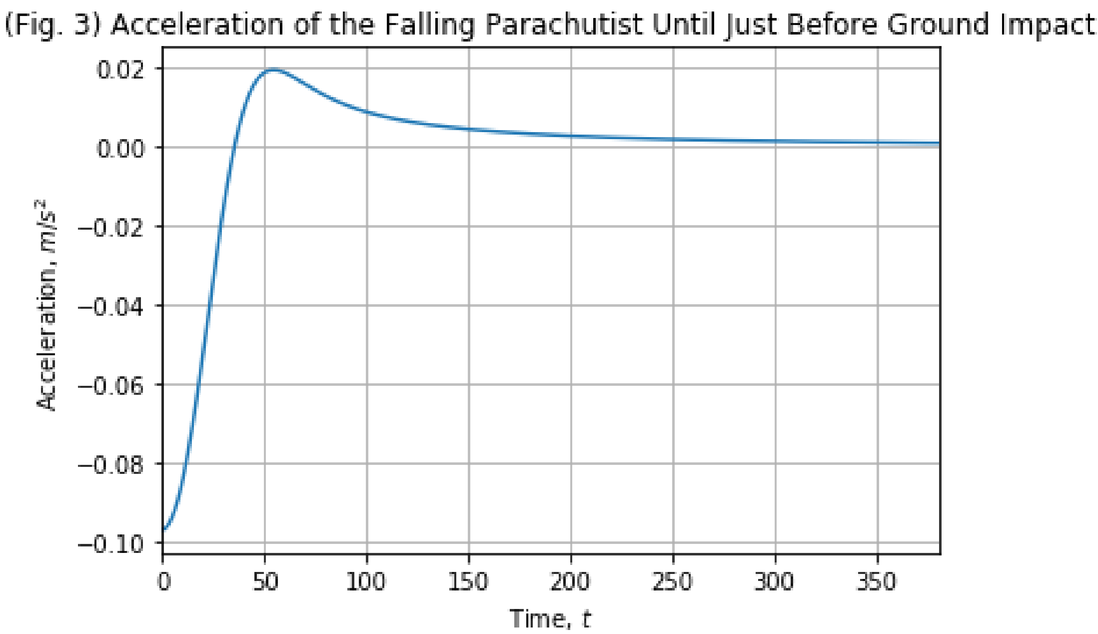

```
In [8]: plt.figure(2)
        plt.plot(t,odeint2[:,1])
        plt.title("(Fig. 2) Velocity of the Falling Parachutist \
        until Just Before Ground Impact")
        plt.xlabel('Time, $t$')
        plt.ylabel('Velocity, $m/s$')
```

```
plt.xlim([0,interp2(0)+20])
plt.show()
```

(Fig. 2) Velocity of the Falling Parachutist until Just Before Ground Impact




```
In [9]: plt.figure(3)
        plt.plot(t, np.gradient(odeint2[:,1]))
        #"Derivative" of the velocity (delta v) divided by time (delta t)
        #Used to approximate the acceleration
        plt.title("(Fig. 3) Acceleration of the Falling Parachutist \
        Until Just Before Ground Impact")
        plt.xlabel('Time, $t$')
        plt.ylabel('Acceleration, $m/s^2$')
        plt.xlim([0,interp2(0+20)])
        plt.grid() #To emphasize that acceleration approaches zero
        plt.show()
```



The acceleration becomes positive as the parachutist falls because the exponential drag force is much stronger closer to the Earth's surface - this is because the atmosphere becomes thicker as the parachutist falls down to the Earth.

We can also see that the acceleration approaches $0m/s^2$ as the opposing gravitational and drag forces approach an equilibrium.

```
In [ ]:
```