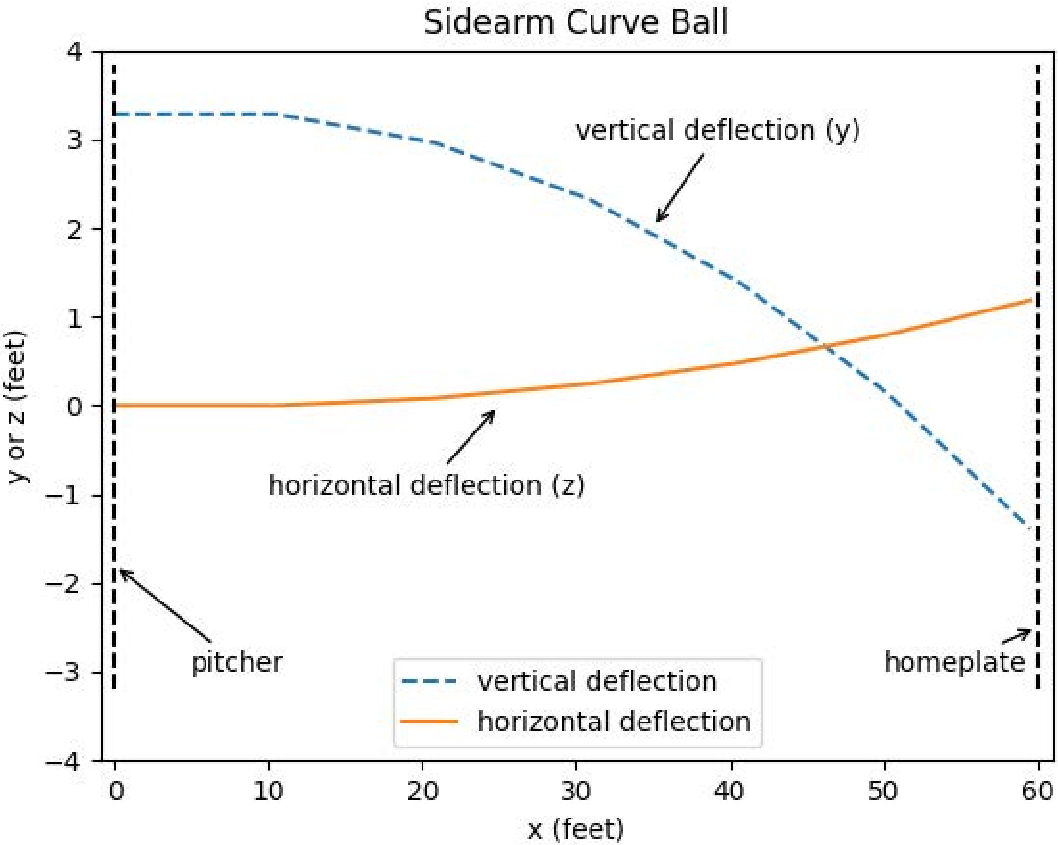
Sidearm Curve Ball
vertical deflection (y)
horizontal deflection (z)
pitcher
homeplate
vertical deflection
horizontal deflection
y or z (feet)
x (feet)
4
3
2
1
0
−1
−2
−3
−4
0
10
20
30
40
50
60

```
import matplotlib.pyplot as plt
import math
import time

X = 0
Y = 1
Z = 2
VX = 3
VY = 4
VZ = 5

def main(v_wind=0, S=0, Ï‰=0):
    for i in range(0, N):
        x = data[i][X]
        y = data[i][Y]
        vx = data[i][VX]
        vy = data[i][VY]
        z = data[i][Z]
        vz = data[i][VZ]

        v = math.sqrt((vx-v_wind)**2 + vy**2)
        B = (0.0039 + 0.0058/(1+math.exp((v - v_drag)/delta))) * m
        F_drag = (B/m) * v

        x_i = x + vx * dt
        vx_i = vx - (F_drag*(vx - v_wind)) * dt

        y_i = y + vy * dt
        vy_i = vy - g * dt - (F_drag*vy) * dt

        z_i = z + vz * dt
        vz_i = vz + S * vx * Ï‰ * dt
        if(S == 0):
            if(y_i <= 0):
                break
        else:
            if(x >= 17):
                break
        data[i+1] = [x_i, y_i, z_i, vx_i, vy_i, vz_i]



def plot(x, y, title, labelx, labely, style='-', legendLabel=None):
    if(labelx is not None):
        plt.xlabel(labelx)
    if(labely is not None):
        plt.ylabel(labely)
    if(title is not None):
        plt.title(title)
    if(legendLabel is None):
        legendLabel = title
    line = plt.plot(x,y,style,label=legendLabel)
    plt.legend()
```

```
    return line

def populate(v_wind=0, z_i=0, v_z=0):
    data.clear()
    data[0] = [x_i, y_i, z_i, v_x, v_y, v_z]

    main(v_wind)

    x = []
    y = []
    for key in data.keys():
        x.append(data[key][X])
        y.append(data[key][Y])
    return (x, y)

if __name__ == '__main__':
    start_time = time.time()
    data = dict()
    #Gravity constant
    g=9.81
    #ball mass
    m=0.145
    #inital velocity magnitude
    v_i = 50.512
    #inital angle
    angle_i = 0.6109
    #inital velocity x component
    v_x = v_i * math.cos(angle_i)
    #inital velocity y component
    v_y = v_i * math.sin(angle_i)

    #inital x position
    x_i = 0
    #inital y position
    y_i = 0

    #Drag Velocity
    v_drag = 35
    #Delta
    delta = 5

    #Increments of 0.1
    dt = 0.1
    #2000 iterations
    N = 2000

    x,y = populate()
    plot(x, y, 'Trajectory of a batted baseball', 'x (m)', 'y (m)',legendLabel='no wind')

    x,y = populate(4.592)
    plot(x, y, 'Trajectory of a batted baseball', 'x (m)', 'y (m)',style='--',legendLabel='tail wind')

    x,y = populate(-4.592)
    plot(x, y, 'Trajectory of a batted baseball', 'x (m)', 'y (m)',style='-.',legendLabel='head wind')
```

```
plt.show()

#inital velocity
v_i = 32.144
#Constant average drag force over the face of the ball
S = 4.1 * 10 ** -4
#Omega
Ï‰ = 30 * 2 * math.pi
#Inital Y position
y_i = 1
#inital Z position
z_i = 0
#inital velocity vector
v_x = v_i
v_y = 0
v_z = 0

data.clear()
data[0] = [x_i, y_i, z_i, v_x, v_y, v_z]

main(S=S, Ï‰=Ï‰)
x = []
y = []
z = []
for key in data.keys():
    x.append(data[key][X]/0.3048)
    y.append(data[key][Y]/0.3048)
    z.append(data[key][Z]/0.3048)
plot(x, y, 'Sidearm Curve Ball', 'x (feet)', 'y or z (feet)',style='--',legendLabel='vertical deflection')
plot(x, z, 'Sidearm Curve Ball', 'x (feet)', 'y or z (feet)',style='-',legendLabel='horizontal deflection')
plt.axvline(x=0, ymin=0.1, ymax=0.98, label='pitcher', color='k',linestyle='--')
plt.axvline(x=60,ymin =0.1, ymax=0.98, label='homeplate', color='k',linestyle='--')
plt.annotate('pitcher', xy=(0,-1.8), xytext=(5, -3), arrowprops=dict(arrowstyle="->"))
plt.annotate('homeplate', xy=(60,-2.5), xytext=(50, -3), arrowprops=dict(arrowstyle="->"))
plt.annotate('horizontal deflection (z)', xy=(25, 0), xytext=(10, -1), arrowprops=dict(arrowstyle="->"))
plt.annotate('vertical deflection (y)', xy=(35, 2), xytext=(30, 3), arrowprops=dict(arrowstyle="->"))
plt.ylim([-4, 4])
plt.xlim([-1,60.9])
plt.show()
print("--- %s seconds ---" % (time.time() - start_time))
```