Diagram

Description automatically generated

**Python Code**

#Program to create Figure 2.2 from Giordano

import pandas as pd

from matplotlib import pyplot as plt

velocitylist=[] #list of data from calculations

timelist=[] #list of data from calculations

velocitydraglist=[] #list of data from calculations

timedraglist=[] #list of data from calculations

#variables

velocity = 4 #initial velocity in m/s

velocitydrag = 4 #initial velocity in m/s

Power = 400 #power generated by cyclist in Watts

mass = 70 #combined mass of cyclist and bicycle in kg

dragcoeff = 0.5 #drag coefficient

airdensity = 1.225 #air density

Area = 0.33 #cross-sectional area of cyclist and bicycle in square meters

time = 0 #initial time

timedrag = 0 #initial time

timestep = 0.1

#Euler method for velocity without drag

for i in range(2000):

velocity = velocity + ((Power\*timestep)/(mass\*velocity))

time = time + timestep

velocitylist.append(velocity) #adds data to a list

timelist.append(time) #adds data to a list

#Euler method for velocity with drag

for i in range(2000):

velocitydrag = velocitydrag + ((Power\*timestep)/(mass\*velocitydrag)) - ((dragcoeff\*airdensity\*Area\*timestep\*velocitydrag\*velocitydrag)/(mass))

timedrag = timedrag + timestep

velocitydraglist.append(velocitydrag) #adds data to a list

timedraglist.append(timedrag) #adds data to a list

#Creates a plot with both sets of data

%matplotlib

fig = plt.gcf()

fig.set\_size\_inches(6.05, 5)

plt.plot(timelist, velocitylist, 'k', lw=1.5)

plt.plot(timedraglist, velocitydraglist, 'k:', lw=1.5)

plt.xlim(0,200)

plt.ylim(0,50)

plt.title("Bicycle simulation: velocity vs. time")

plt.xlabel("time (s)")

plt.ylabel("velocity (m/s)")

plt.text(75, 9, "With air resistance", fontsize=14)

plt.text(18,35, "No air resistance", fontsize=14)

plt.xticks(ticks = (0,50,100,150,200))

plt.yticks(ticks = (0,10, 20, 30, 40, 50))

plt.tick\_params(direction='in', bottom=True, top=True, left=True, right=True)

plt.tick\_params(labelbottom=True, labeltop=False, labelleft=True, labelright=False)

plt.show()