Diagram

Description automatically generated

**Python Code**

#Program to create Figure 2.7 from Giordano

from math import cos,sin,exp,sqrt,pi,radians

from matplotlib import pyplot as plt

#lists of data from calculations

xlist=[]

ylist=[]

xheadlist=[]

yheadlist=[]

xtaillist=[]

ytaillist=[]

#variables

x = 0

y = 1

xhead = 0

yhead = 1

xtail = 0

ytail = 1

velocity = 49 #initial velocity

vy = velocity\*sin(radians(35)) #initial velocity in y direction

vx = velocity\*cos(radians(35)) #initial velocity in x direction

Vd = 35

delta = 5

gravity = 9.8

timestep = 0.01

xlist.append(x)

ylist.append(y)

xheadlist.append(xhead)

yheadlist.append(yhead)

xtaillist.append(xtail)

ytaillist.append(ytail)

def getNewV (Vx, Vy, Vwind, xy):

if xy == 0:

WindDif = Vx-Vwind

V = Vx

else:

WindDif = Vy

V = Vy

V -= (gravity\*timestep)

Vadj = sqrt((Vx - Vwind)\*\*2+Vy\*\*2)

V -= windDrag(Vadj, WindDif)\*(0.0039 + (0.0058/(1 + exp((Vadj - Vd)/delta))))\*timestep

return V;

def windDrag (Vadj, Winddif):

drag = Vadj\*Winddif

return drag;

def reset():

global x

x = 0.0

global y

y = 1.0

global vTot

vTot = 49

global vx

vx = velocity\*cos(radians(35))

global vy

vy = velocity\*sin(radians(35))

return;

#Euler method for velocity without drag

for i in range(1000):

x += vx\*timestep

y += vy\*timestep

vx = getNewV(vx, vy, 0, 0)

vy = getNewV(vx, vy, 0, 1)

xlist.append(x) #adds data to a list

ylist.append(y) #adds data to a list

if y <= 0:

break

reset()

for i in range(1000):

x += vx\*timestep

y += vy\*timestep

vx = getNewV(vx, vy, -4.4704, 0)

vy = getNewV(vx, vy, -4.4704, 1)

xheadlist.append(x) #adds data to a list

yheadlist.append(y) #adds data to a list

if y <= 0:

break

reset()

for i in range(1000):

x += vx\*timestep

y += vy\*timestep

vx = getNewV(vx, vy, 4.4704, 0)

vy = getNewV(vx, vy, 4.4704, 1)

xtaillist.append(x) #adds data to a list

ytaillist.append(y) #adds data to a list

if y <= 0:

break

#Creates a plot with all three sets of data

%matplotlib

fig = plt.gcf()

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

plt.plot(xlist, ylist, 'k', lw=1.5)

plt.plot(xheadlist, yheadlist, 'k-.', lw=1.5)

plt.plot(xtaillist, ytaillist, 'k:', lw=1.5)

plt.xlim(0,150)

plt.ylim(0,30)

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

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

plt.title("Trajectory of a batted baseball")

plt.xlabel("x (m)")

plt.ylabel("y (m)")

plt.annotate('tailwind', xy=(110.5,17.5), xytext=(125,23),

arrowprops=dict(arrowstyle='->', shrinkA=3,

connectionstyle='arc3,rad=-0.1'), ha='center', va='top', fontsize=14)

plt.annotate('no wind', xy=(115, 8), xytext=(135,13),

arrowprops=dict(arrowstyle='->', shrinkA=3,

connectionstyle='arc3,rad=-0.1'), ha='center', va='top', fontsize=14)

plt.annotate('headwind', xy=(103.5, 12), xytext=(80,6),

arrowprops=dict(arrowstyle='->', shrinkA=3,

connectionstyle='arc3,rad=-0.1'), ha='center', va='top', fontsize=14)

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()