Diagram, engineering drawing

Description automatically generated

Python Code

#Program to create Figure 4.6 left from Giordano

import pandas as pd

from matplotlib import pyplot as plt

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

import sys

#Lists of x and y positions

xleftlist=[]

yleftlist=[]

xrightlist=[]

yrightlist=[]

#Variables

xleft = 1 #x position in AU left

yleft = 0 #y position in AU left

vxleft = 0 #x velocity left

vyleft = 5 #y velocity left

betaleft = 2.1 #value to adjust inverse square law left

xright = 1 #x position in AU right

yright = 0 #y position in AU right

vxright = 0 #x velocity right

vyright = 5 #y velocity right

betaright = 2.01 #value to adjust inverse square law right

dt = 0.001 #time step in years

xleftlist.append(xleft)

yleftlist.append(yleft)

xrightlist.append(xright)

yrightlist.append(yright)

#Euler-Cromer method to calculate x and y positions and velocities for left figure

for i in range(1675):

rleft = sqrt(xleft\*\*2 + yleft\*\*2)

vxleft -= 4\*(pi\*\*2)\*(xleft/(rleft\*rleft\*\*betaleft))\*dt

xleft += vxleft\*dt

vyleft -= 4\*(pi\*\*2)\*(yleft/(rleft\*rleft\*\*betaleft))\*dt

yleft += vyleft\*dt

xleftlist.append(xleft)

yleftlist.append(yleft)

#Euler-Cromer method to calculate x and y positions and velocities for right figure

for i in range(5000):

rright = sqrt(xright\*\*2 + yright\*\*2)

vxright -= 4\*(pi\*\*2)\*(xright/(rright\*rright\*\*betaright))\*dt

xright += vxright\*dt

vyright -= 4\*(pi\*\*2)\*(yright/(rright\*rright\*\*betaright))\*dt

yright += vyright\*dt

xrightlist.append(xright)

yrightlist.append(yright)

#Creates a plot with both sets of data

%matplotlib

fig, (ax1, ax2) = plt.subplots(1, 2)

ax1.margins(0)

ax1.plot(xleftlist, yleftlist, 'k:', dashes=(1,0.25))

ax1.set\_xticks([-1,-0.5,0,0.5,1])

ax1.set\_xticklabels([-1,-0.5,0,0.5,1], fontsize=12)

ax1.set\_yticks([-1,-0.5,0,0.5,1])

ax1.set\_yticklabels([-1,-0.5,0,0.5,1], fontsize=12)

ax1.set\_title("Simulation of elliptical orbit", fontsize=14)

ax1.set(xlabel='x (AU)', ylabel='y (AU)')

ax1.text(-0.68, 0.8, "\u03B2 = 2.10", fontsize=10, horizontalalignment='left', verticalalignment='bottom')

ax1.tick\_params(direction = 'in', bottom=True, top=True, left=True, right=True)

ax1.tick\_params(labelbottom=True, labeltop=False, labelleft=True, labelright=False)

ax1.axvline(x = 0, ymin = 0, ymax = 1, color = 'black', linestyle = ':', lw = 0.5)

ax1.axhline(y = 0, xmin = 0, xmax = 1, color = 'black', linestyle = ':', lw = 0.5)

ax2.plot(xrightlist, yrightlist, 'k:', dashes=(1.5,0.25))

ax2.set\_xticks([-1,-0.5,0,0.5,1])

ax2.set\_xticklabels([-1,-0.5,0,0.5,1], fontsize=12)

ax2.set\_yticks([-1,-0.5,0,0.5,1])

ax2.set\_yticklabels([-1,-0.5,0,0.5,1], fontsize=12)

ax2.set\_title("Simulation of elliptical orbit", fontsize=14)

ax2.set(xlabel='x (AU)', ylabel='y (AU)')

ax2.text(-0.68, 0.8, "\u03B2 = 2.01", fontsize=10, horizontalalignment='left', verticalalignment='bottom')

ax2.tick\_params(direction = 'in', bottom=True, top=True, left=True, right=True)

ax2.tick\_params(labelbottom=True, labeltop=False, labelleft=True, labelright=False)

ax2.axvline(x = 0, ymin = 0, ymax = 1, color = 'black', linestyle = ':', lw = 0.5)

ax2.axhline(y = 0, xmin = 0, xmax = 1, color = 'black', linestyle = ':', lw = 0.5)

plt.show()