import random

import matplotlib.pyplot as plt

import matplotlib.colors as mcolors

import numpy as np

fig, ax = plt.subplots(2,3)

bigList1 = []

bigList2 = []

bigList3 = []

variables1 = [1,-1]

variables3 = [-1, (1-np.sqrt(3))/2, (1+np.sqrt(3))/2]

numberOfRuns = 1000

numberOfSteps = 1000

for j in range(numberOfRuns):

    list1 = []

    list2 = []

    list3 = []

    value1 = 0

    value2 = 0

    value3 = 0

    for i in range(numberOfSteps):

        value1 = value1 + random.choice(variables1)

        value2 = value2 + np.random.normal(0,1)

        value3 = value3 + random.choice(variables3)

        list1.append(value1)

        list2.append(value2)

        list3.append(value3)

    bigList1.append(list1)

    bigList2.append(list2)

    bigList3.append(list3)

numberList = range(0,numberOfSteps)

for i in range(numberOfRuns):

    ax[0,0].plot(bigList1[i],numberList, linewidth = 0.01, color = "b")

    ax[0,1].plot(bigList2[i],numberList, linewidth = 0.01, color = "g")

    ax[0,2].plot(bigList3[i],numberList, linewidth = 0.01, color = "r")

finalNumber1 = []

finalNumber2 = []

finalNumber3 = []

for i in range(numberOfRuns):

    finalNumber1.append(bigList1[i][numberOfSteps-1])

    finalNumber2.append(bigList2[i][numberOfSteps-1])

    finalNumber3.append(bigList3[i][numberOfSteps-1])

ax[1,0].hist(finalNumber1, bins=30, color = "b")

ax[1,1].hist(finalNumber2, bins=30, color = "g")

ax[1,2].hist(finalNumber3, bins=30, color = "r")

plt.show()

import random

import matplotlib.pyplot as plt

import matplotlib.colors as mcolors

import numpy as np

fig, ax = plt.subplots(2,3)

t1 = 0.01

t1Steps = int(5/t1)

t2 = 0.05

t2Steps = int(5/t2)

t3 = 0.1

t3Steps = int(5/t3)

bigList1 = []

bigList2 = []

bigList3 = []

numberOfRuns = 10000

for j in range(numberOfRuns):

    list1 = []

    list2 = []

    list3 = []

    value1 = 0

    value2 = 0

    value3 = 0

    for i in range(t1Steps):

        value1 = value1 + np.random.normal(0,1)\*np.sqrt(t1)

        list1.append(value1)

    for i in range(t2Steps):

        value2 = value2 + np.random.normal(0,1)\*np.sqrt(t2)

        list2.append(value2)

    for i in range(t3Steps):

        value3 = value3 + np.random.normal(0,1)\*np.sqrt(t3)

        list3.append(value3)

    bigList1.append(list1)

    bigList2.append(list2)

    bigList3.append(list3)

numberList1 = np.linspace(0,5,num=t1Steps)

numberList2 = np.linspace(0,5,num=t2Steps)

numberList3 = np.linspace(0,5,num=t3Steps)

for i in range(50):

    ax[0,0].plot(numberList1, bigList1[i], linewidth = 0.1, color = "b")

    ax[0,1].plot(numberList2, bigList2[i], linewidth = 0.1, color = "g")

    ax[0,2].plot(numberList3, bigList3[i], linewidth = 0.1, color = "r")

mean1 = []

for j in range(t1Steps):

    mean = 0

    for i in range(numberOfRuns):

        mean += (bigList1[i][j]-bigList1[i][0])\*\*2

    mean1.append(mean/numberOfRuns)

mean2 = []

for j in range(t2Steps):

    mean = 0

    for i in range(numberOfRuns):

        mean += (bigList2[i][j]-bigList2[i][0])\*\*2

    mean2.append(mean/numberOfRuns)

mean3 = []

for j in range(t3Steps):

    mean = 0

    for i in range(numberOfRuns):

        mean += (bigList3[i][j]-bigList3[i][0])\*\*2

    mean3.append(mean/numberOfRuns)

ax[1,0].plot(numberList1,mean1, color = "b")

ax[1,1].plot(numberList2,mean2, color = "g")

ax[1,2].plot(numberList3,mean3, color = "r")

plt.show()

import numpy as np

import matplotlib.pyplot as plt

fig, ax = plt.subplots(1,3)

temp = 300

R = 0.000001

eta = 0.001

m = 0.0000000000000111

kB = 0.00000000000000000000001380649

gamma = 6\*np.pi\*eta\*R

tau = m/gamma

t = 0.01\*tau

timesteps = int(1\*tau/t)

x1 = []

x2 = []

x1new = []

x2new = []

x1.append(0)

x1.append(0)

x2.append(0)

x2.append(0)

W=np.random.randn(timesteps)

for i in range(2,timesteps+2):

    x1.append((2+t\*(gamma/m))/(1+t\*(gamma/m))\*x1[i-1]-x1[i-2]/(1+t\*(gamma/m))+ np.sqrt(2\*kB\*temp\*gamma)/(m\*(1+t\*(gamma/m)))\*np.power(t,3/2)\*W[i-2])

    x2.append(x2[i-1]+np.sqrt((2\*kB\*temp\*t)/gamma)\*W[i-2])

    x1new.append(x1[i]\*1e9)

    x2new.append(x2[i]\*1e9)

xaxis = []

for i in range(timesteps):

    xaxis.append(i\*0.01)

ax[0].plot(xaxis,x1new, "k--")

ax[0].plot(xaxis,x2new)

t = 0.01\*tau

timesteps = int(100\*tau/t)

x1 = []

x2 = []

x2new = []

x1new = []

W=np.random.randn(timesteps)

x1.append(0)

x1.append(0)

x2.append(0)

x2.append(0)

for i in range(2,timesteps+2):

    x1.append((2+t\*(gamma/m))/(1+t\*(gamma/m))\*x1[i-1]-x1[i-2]/(1+t\*(gamma/m))+ np.sqrt(2\*kB\*temp\*gamma)/(m\*(1+t\*(gamma/m)))\*np.power(t,3/2)\*W[i-2])

    x2.append(x2[i-1]+np.sqrt((2\*kB\*temp\*t)/gamma)\*W[i-2])

    x1new.append(x1[i]\*1e9)

    x2new.append(x2[i]\*1e9)

xaxis = []

for i in range(timesteps):

    xaxis.append(i\*0.01)

ax[1].plot(xaxis,x1new, "k--")

ax[1].plot(xaxis,x2new)

numberOfParticles = 1000

bigList1 = []

bigList2 = []

for j in range(numberOfParticles):

    x1 = []

    x2 = []

    x1new = []

    x2new = []

    x1.append(0)

    x1.append(0)

    x2.append(0)

    x2.append(0)

    for i in range(2,timesteps+2):

        r = np.random.randn(1)

        x1.append((2+t\*(gamma/m))/(1+t\*(gamma/m))\*x1[i-1]-x1[i-2]/(1+t\*(gamma/m))+ np.sqrt(2\*kB\*temp\*gamma)/(m\*(1+t\*(gamma/m)))\*np.power(t,3/2)\*r)

        x2.append(x2[i-1]+np.sqrt((2\*kB\*temp\*t)/gamma)\*r)

        x1new.append(x1[i])

        x2new.append(x2[i])

    bigList1.append(x1new)

    bigList2.append(x2new)

mean1 = []

mean2 = []

for j in range(timesteps):

    tempmean1 = 0

    tempmean2 = 0

    for i in range(numberOfParticles):

        tempmean1 += (bigList1[i][j]-bigList1[i][0])\*\*2

        tempmean2 += (bigList2[i][j]-bigList2[i][0])\*\*2

    mean1.append(tempmean1/numberOfParticles)

    mean2.append(tempmean2/numberOfParticles)

xaxis = []

for i in range(timesteps):

    xaxis.append(np.log10(i/100))

    mean1[i] = np.log10(mean1[i]\*1e18)

    mean2[i] = np.log10(mean2[i]\*1e18)

ax[2].plot(xaxis,mean1, "k--")

ax[2].scatter(xaxis,mean2)

plt.show()

import numpy as np

from matplotlib import pyplot as plt

import math

fig, ax = plt.subplots(1,4)

temp = 300

R = 0.000001

eta = 0.001

m = 0.0000000000000111

kB = 0.00000000000000000000001380649

gamma = 6\*np.pi\*eta\*R

tau = m/gamma

kx = 1e-6             # Stiffness of the optical trap along x

ky = 0.25e-6              # Stiffness of the optical trap along y

t = 0.01

N = int(1e5)

x = np.zeros(N)    # Initiated trajectory array x

y = np.zeros(N)    # Initiated trajectory array y

Wx=np.random.randn(N)  # Gaussian distributed random numbers

Wy=np.random.randn(N)  # Gaussian distributed random numbers

for i in range(N-1):

    x[i+1] = x[i] - kx\*x[i]\*t/gamma + np.sqrt(2\*kB\*temp\*t/gamma)\*Wx[i]     # Overdamped Langevin equation x

    y[i+1] = y[i] - ky\*y[i]\*t/gamma + np.sqrt(2\*kB\*temp\*t/gamma)\*Wy[i]      # Overdamped Langevin equation y

avgxList = []

avgyList = []

axisList = []

for j in range(300):

    axisList.append(j\*t)

    avgx = 0

    avgy = 0

    for i in range(N-j):

        avgx += x[i]\*x[i+j]

        avgy += y[i]\*y[i+j]

    avgxList.append(abs(avgx/(N-j)))

    avgyList.append(abs(avgy/(N-j)))

ax[2].plot(axisList,avgxList)

ax[2].plot(axisList,avgyList)

k = np.linspace(0, 3,300)

o1 = []

o2 = []

# for i in range(len(k)):

#     o1.append((kB\*temp)/kx\*math.exp((-kx\*k[i])/gamma))

#     o2.append((kB\*temp)/ky\*math.exp((-ky\*k[i])/gamma))

for i in k:

    o1.append((kB\*temp)/kx\*math.exp((-kx\*i)/gamma))

    o2.append((kB\*temp)/ky\*math.exp((-ky\*i)/gamma))

ax[2].plot(k,o1,"k--")

ax[2].plot(k,o2,"k--")

ax[2].set\_xlim([0,0.3])

ax[0].plot(x\*1e9,y\*1e9,'.',markersize=0.6)

val, axis, \_ = ax[1].hist(x, bins = 100, density = True, color = "lightblue", histtype = "step", linewidth = 2)

a = np.linspace(max(axis),min(axis),100)

b = []

for i in range(len(a)):

    b.append(math.exp(((kx\*a[i]\*\*2)/2)/(-kB\*temp))\*max(val))

    a[i] = a[i]

ax[1].plot(a,b, "k--")

val = []

axis = []

a = []

val, axis, \_ = ax[1].hist(y, bins = 100, density = True,  color = "g", histtype = "step", linewidth = 2)

a = np.linspace(max(axis),min(axis),100)

b = []

for i in range(len(a)):

    b.append(math.exp(((ky\*a[i]\*\*2)/2)/(-kB\*temp))\*max(val))

    a[i] = a[i]

ax[1].plot(a,b,"k--")

a = 0

g = []

b = [1,2,3,5,10]

for j in b:

    a = 0

    x = []

    x.append(0)

    for i in range(N):

        x.append(x[i] - j\*1e-9\*x[i]\*t/gamma + np.sqrt(2\*kB\*temp\*t/gamma)\*Wx[i])

        a += abs(x[i])

    g.append(a\*\*2/(N))

    #     a += (x[i+1]-x[i])

    # g.append((a/(N-1))\*\*2\*1e12)

ax[3].scatter(b,g)

s = []

v = []

xaxis = np.linspace(0.1,10,N)

for i in range(N):

    v.append(1e14\*kB\*temp/xaxis[i])

ax[3].plot(xaxis,v)

plt.show()