

**FACULTY OF ELECTRICAL ENGINEERING**

**UNIVERSITI TEKNOLOGI MALAYSIA**

**SKEE 1033 – SCIENTIFIC PROGRAMMING**

**IN CLASS ACTIVITY 1**

|  |  |  |
| --- | --- | --- |
| LECTURER: | DR. MUHAMMAD ARIFF BAHARUDIN | |
| GROUP NAME: | Connecting…. | |
| GROUP MEMBER | 1. LIM YEOW SHENG  2. SOO YU XIANG  3. PANG BIAO XIAN  4. LEE KOCK ZONG | A24KE0144  A24KE2088  A24KE0404  A24KE0129 |
| SECTION | 05 & 06 | |
| SUBMISSION DATE | 10/12/2024 | |

**Problem 1**

import matplotlib.pyplot as plt

import numpy as np

# Problem 1

t = np.arange(0, 11, 1)

a = 2

v = a \* t

plt.plot(t, v)

plt.xlabel('Time(t)')

plt.ylabel('Velocity(v)')

plt.title('Velocity(v) over time(t)')

plt.show()

**Problem 2**

import numpy as np

import matplotlib.pyplot as plt

t = np.arange(0, 11, 1)

y1 = 3 \* t + 2

y2 = 2 \* t\*\*2 + 1

plt.plot(y1,t,y2,t)

plt.xlabel('Plant')

plt.ylabel('Days')

plt.title('Plant growth over the days')

plt.legend(["Plant A","Plant B"])

plt.show()

**#Problem 3**

import matplotlib.pyplot as plt

import numpy as np

A = 5

ω= 2

#Displacement x(t)

#x-axis

t=np.arange(0, 11, 1)

#y-axis

x= A \* np.sin(ω\*t)

plt.subplot(1,3,1)

plt.plot(t,x)

plt.title("Displacement x(t)")

plt.xlabel("Time(t)")

plt.ylabel("Displacement(x)")

#Velocity v(t)

#x-axis

t=np.arange(0, 11, 1)

#y-axis

v= A \* ω \* np.cos(ω\*t)

plt.subplot(1,3,3)

plt.plot(t,v)

plt.title("Velocity v(t)")

plt.xlabel("Time(t)")

plt.ylabel("Velocity(v)")

plt.show()

**#Problem 4**

import matplotlib.pyplot as plt

import numpy as np

temperatures = [15,17,19,21,23,25,27,28,27,26,24,22,20]

time = ["6AM", "7AM",  "8AM", "9AM", "10AM", "11AM", "12PM", "1PM", "2PM", "3PM", "4PM", "5PM", "6PM"]

plt.plot(time, temperatures, "-.rx")

plt.title(f"Temperatures (\N{DEGREE SIGN}C) recorded hourly. Stem Plot")

xlabel = len(time)

plt.xticks(time, time)

plt.xlabel("Time")

plt.ylabel("Temperature")

**#Problem 5**

import matplotlib.pyplot as plt

import numpy as np

t = np.arange(1, 11)

bacteria = 10

growth = bacteria \*\* t

plt.semilogy(t, growth, "-mo")

plt.title("Logarithmic Plot")

plt.xlabel("Time, (hours)")

plt.xticks(t)

plt.ylabel("Bacterial Growth")

plt.tight\_layout()

plt.show()

**#Problem 6**

import numpy as np

import matplotlib.pyplot as plt

mean = 70

std\_dev = 10

num\_of\_students = 1000

test\_scores = np.random.normal(mean, std\_dev, num\_of\_students)

#Histogram

plt.hist(test\_scores, bins=20)

plt.title("Histogram of Student's Test Scores")

plt.xlabel('Test Scores')

plt.ylabel("Numbers of students")

plt.grid(True)

plt.show()

**#Problem 7**

import numpy as np

import matplotlib.pyplot as plt

mean\_height\_teamA= 180

std\_dev\_height\_teamA= 6

mean\_height\_teamB= 190

std\_dev\_height\_teamB= 6

num\_of\_players=50

height\_teamA=np.random.normal(mean\_height\_teamA, std\_dev\_height\_teamA, num\_of\_players)

height\_teamB=np.random.normal(mean\_height\_teamB, std\_dev\_height\_teamB, num\_of\_players)

height\_data=[height\_teamA,height\_teamB]

#box plot

plt.figure()

plt.boxplot(height\_data,labels=["TeamA","TeamB"])

plt.title("Box plot of basketball players's height")

plt.ylabel("Height(cm)")

plt.show()

**#Problem 8**

import numpy as np

import matplotlib.pyplot as plt

data = np.random.normal(1.2, 0.2, 30)

plt.figure()

plt.boxplot(data)

numDataPoints = len(data)

x = np.ones(numDataPoints)

x\_scattered = x + 0.2 \* np.random.randn(numDataPoints)

plt.scatter(x\_scattered, data)

plt.title('Scatter Box Plot of the weight of the 30 fruits(kg)')

plt.xlabel('Weight')

plt.ylabel('The weight of 30 fruits(kg)')

plt.show()

**#Problem 9**

import numpy as np

import matplotlib.pyplot as plt

C = 1e-6

R = 10e3

f = np.linspace(1, 10e4, 10)

H = 1 / (1 + 1j \* 2 \* np.pi \* f \* R \* C)

Magnitude = 20 \* np.log10(np.abs(H))

Phase = np.angle(H)

#Magnitude plot

plt.subplot(2, 1, 1)

plt.semilogx(f, Magnitude)

plt.xlabel('Frequency, f (Hz)')

plt.ylabel('Magnitude (dB)')

plt.title('Magnitude over Frequency,f (Hz) ')

#Phase plot

plt.subplot(2, 1, 2)

plt.semilogx(f, Phase / np.pi)

plt.xlabel('Frequency, f (Hz)')

plt.ylabel('Phase (π rad)')

plt.yticks(np.arange(-1, 1.5, 0.5), [f'{tick:.1f}π' for tick in np.arange(-1, 1.5, 0.5)])

plt.tight\_layout()

plt.show()

import matplotlib.pyplot as plt

import numpy as np

**#Comprehensive Problem**

# Step 1: 2D line plot

t= np.arange(0, 61, 1)

A=np.sin(0.1\*np.pi\*t)

plt.plot(t, A)

plt.xlabel("Seismic amplitude A(t)")

plt.ylabel("Time(t)")

plt.title("Seismic amplitude A(t) over time(t)")

plt.show()

# Step 2: Stem plot

t= np.arange(0, 61, 1)

A=np.sin(0.1\*np.pi\*t)

plt.stem(t, A)

plt.xlabel("Seismic amplitude A(t)")

plt.ylabel("Time(t)")

plt.title("Seismic amplitude A(t) over time(t)")

plt.show()

# Step 3: Normal plot vs log-log plot

EE=[10\*\*n for n in range(1,11,1)]

t=np.arange(1,11,1)

plt.subplot(1, 2, 1)

plt.plot(t,EE)

plt.xlabel("Time (t)")

plt.ylabel("Energy Released (EE)/j")

plt.title("Energy released(EE)/j over time(t)")

plt.subplot(1, 3, 3)

plt.loglog(t,EE)

plt.xlabel("Time (log t)")

plt.ylabel("Energy Released (log EE)")

plt.title("Energy released (log EE) over time(log t)")

plt.show()

# Step 4: Log-log plot

LL=[30, 35, 50, 40, 45, 55,50, 60, 65, 70, 75, 80, 85, 90, 95, 100, 95, 85, 80, 75, 70, 65, 60, 55]

t=np.arange(1,25,1)

plt.loglog(t,LL)

plt.xlabel("Time (log t)")

plt.ylabel("Sound Intensity (log LL)")

plt.title("Sound Intensity (log LL) over Time (log t))")

plt.show()

# Step 5: Bode plot

R=10000

C=1e-6

f=np.arange(1,10001,1000)

H = 1/(1 + 1j \*2\*np.pi\* f \* R \* C)

magnitude=(20\*np.log10(np.abs(H)))

phase = np.angle(H)

#Magnitude plot

plt.subplot(2, 1, 1)

plt.semilogx(f, magnitude)

plt.xlabel("Frequency,f (Hz)")

plt.ylabel("Magnitude (dB)")

plt.title("Magnitude(dB) over freqyency(Hz)")

#Phase plot

plt.subplot(2, 1, 2)

plt.semilogx(f, phase/np.pi)

plt.xlabel("Frequency,f (Hz)")

plt.ylabel("Phase (π rad)")

plt.yticks(np.arange(-1, 1.5, 0.5), [f'{tick:.1f}π' for tick in np.arange(-1, 1.5, 0.5)])

plt.title("Phase (π rad) over frequency(Hz)")

plt.show()

# Step 6: Histogram

mean = 4

std\_dev = 0.5

num\_of\_earthquake = 100

test\_scores = np.random.normal(mean, std\_dev, num\_of\_earthquake)

plt.hist(test\_scores, bins=10)

plt.title("Histogram of magnitude distribution")

plt.xlabel("Magnitude distribution(M)")

plt.ylabel("Numbers of earthquake")

plt.show()

# Step 7: Box plot

mean\_north= 10

std\_dev\_north=2

mean\_south=12

std\_dev\_south=3

num\_of\_earthquake=100

north=np.random.normal(mean\_north, std\_dev\_north, num\_of\_earthquake)

south=np.random.normal(mean\_south, std\_dev\_south, num\_of\_earthquake)

data=[north,south]

plt.figure()

plt.boxplot(data,labels=["North","South"])

plt.title("Box plot of North and South sensor measurement(X)")

plt.ylabel("Sensor measurement (X)")

plt.show()