Lab File

AS1104 Computational Mathematics

PREPARED BY

**Manas Acharya (2021BCA013)**

FACULTY GUIDE

**Dr. Jaya Gupta**



**Department of Computer Science Engineering**

**Institute of Engineering and Technology (IET)**

**JK Lakshmipat University Jaipur**

**April 2022**

INDEX

|  |  |  |  |
| --- | --- | --- | --- |
| **SN** | **Name of Experiment** | **Date** | **Sign/Comment** |
| 1. | Write a python code to calculate the addition of two matrixes. | 3-feb |  |
| 2. | Write a python code to calculate the subtraction of two matrixes. | 3-feb |  |
| 3. | Write a python code to calculate the multiplication of two matrixes. | 10-feb |  |
| 4. | Write a python code to calculate the determinant of a matrix. | 17-feb |  |
| 5. | Write a python code to calculate the inverse of a matrix. | 24-feb |  |
| 6. | Write a python code to calculate the numeric solution using bisection method. | 3-march |  |
| 7. | Write a python code to calculate the numeric solution using regula falsi method. | 10-march |  |
| 8. | Write a python code to calculate the numeric solution using newton raphson method. | 17-march |  |
| 9. | Write a program to calculate forward interpolation | 24-march |  |
| 10. | Write program to calculate integration using trapezoidal  rule. | 31-march |  |
| 11. | Write program to calculate integration using Simpson 3/8 | 21- April |  |

**Code:- Matrix Addition**

print("Enter order of 1st matrix:")

m,n = list(map(int,input().split()))

print("Enter Row wise values")

matrix1 = []

for i in range(m) :

    print("Enter row",i,"value:")

    elements = list(map(int,input().split()))

    matrix1.append(elements)

print("Enter order of 2nd matrix:")

p,q = list(map(int,input().split()))

print("Enter Row wise values")

matrix2 = []

for j in range(p) :

    print("Enter row",j,"value:")

    elements = list(map(int,input().split()))

    matrix2.append(elements)

print("Matrix 1:",matrix1)

print("Matrix 2:",matrix2)

res = []

for i in range(m):

    elements = []

    for j in range(q):

        elements.append(0)

    res.append(elements)

print("Matrix Addition: ")

for i in range(m):

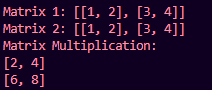
    for j in range(len(matrix1[0])):

        res[i][j] = matrix1[i][j] + matrix2[i][j]

for elements in res:

    print(elements)

**Output:-**

****

**Code:- Matrix Subtraction**

print("Enter order of 1st matrix:")

m,n = list(map(int,input().split()))

print("Enter Row wise values")

matrix1 = []

for i in range(m) :

    print("Enter row",i,"value:")

    elements = list(map(int,input().split()))

    matrix1.append(elements)

print("Enter order of 2nd matrix:")

p,q = list(map(int,input().split()))

print("Enter Row wise values")

matrix2 = []

for j in range(p) :

    print("Enter row",j,"value:")

    elements = list(map(int,input().split()))

    matrix2.append(elements)

print("Matrix 1:",matrix1)

print("Matrix 2:",matrix2)

res = []

for i in range(m):

    elements = []

    for j in range(q):

        elements.append(0)

    res.append(elements)

print("Matrix Subtraction: ")

for i in range(m):

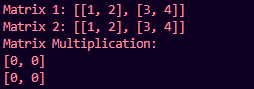
    for j in range(len(matrix1[0])):

        res[i][j] = matrix1[i][j] - matrix2[i][j]

for elements in res:

    print(elements)

**Output:-**

****

**Code :- Matrix Multiplication**

print("Enter order of 1st matrix:")

m,n = list(map(int,input().split()))

print("Enter Row wise values")

matrix1 = []

for i in range(m) :

    print("Enter row",i,"value:")

    elements = list(map(int,input().split()))

    matrix1.append(elements)

print("Enter order of 2nd matrix:")

p,q = list(map(int,input().split()))

print("Enter Row wise values")

matrix2 = []

for j in range(p) :

    print("Enter row",j,"value:")

    elements = list(map(int,input().split()))

    matrix2.append(elements)

print("Matrix 1:",matrix1)

print("Matrix 2:",matrix2)

res = []

for i in range(m):

    elements = []

    for j in range(q):

        elements.append(0)

    res.append(elements)

print("Matrix Multiplication: ")

for i in range(m):

    for j in range(q):

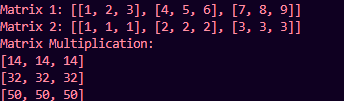
        for k in range(n) :

            res[i][j] += matrix1[i][k] \* matrix2[k][j]

for elements in res:

    print(elements)

**Output :-**



**Code :- Matrix Determinant**

import numpy as np

print("Enter order of the matrix:")

m,n = list(map(int,input().split()))

print("Enter Row wise values")

matrix = []

for i in range(m) :

    print("Enter row",i,"value:")

    elements = list(map(int,input().split()))

    matrix.append(elements)

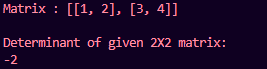
m = np.array(matrix)

det = np.linalg.det(matrix)

print("\nDeterminant of given 2X2 matrix:")

print(round(det))

**Output :-**

****

**Code :- Matrix Inverse**

import numpy as np

print("Enter order of the matrix:")

m,n = list(map(int,input().split()))

print("Enter Row wise values")

matrix = []

for i in range(m) :

    print("Enter row",i,"value:")

    elements = list(map(int,input().split()))

    matrix.append(elements)

m = np.array(matrix)

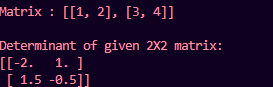
inv = np.linalg.inv(matrix)

print("Matrix :",matrix)

print("\nDeterminant of given 2X2 matrix:")

print((inv))

**Output:-**

****

**Code :- Bisection Method**

import math

def eq(x):

    return x\*\*3 -2\*x -5

*# Prints root of func(x)*

def bisection(x0,x1):

    count =0

    if (eq(x0) \* eq(x1) >= 0):

        print("You have not assumed right x0 and x1\n")

        return

    x2 = x0

    while ((x1-x0) >= 0.0001):

*# Find middle point*

        x2 = (x0+x1)/2

        print("x2 is ",x2, end="")

        print("f(x2) is ",eq(x2))

        if (eq(x2) == 0.0):

            break

        count +=1

        if (eq(x2)\*eq(x0) < 0):

            x1 = x2

            print("x1 is",x1)

        else:

            x0 = x2

            print("x0 is ",x0)

    print("The value of root is : ","%.4f"%x2)

    print("No of iteration : ",count)

x0 = 2

x1 = 3

bisection(x0, x1)

**Output :-**

****

**Code:- Regula Falsi**

MAX\_ITER = 1000000

def func( x ):

    return (x\*x\*x - x\*2 - 5)

def regulaFalsi( a , b):

    if func(a) \* func(b) >= 0:

        print("You have not assumed right a and b")

        return -1

    c = a

    for i in range(MAX\_ITER):

        c = (a \* func(b) - b \* func(a))/ (func(b) - func(a))

        if func(c) == 0:

            break

        elif func(c) \* func(a) < 0:

            b = c

        else:

            a = c

    print("The value of root is : " , '%.4f' %c)

a = 2

b = 3

regulaFalsi(a, b)

**Output :-**

****

**Code : - Newton Raphson**

def func( x ):

    return x \* x \* x - 2 \* x -5

*# Derivative of the above function*

*# which is 3\*x^x - 2\*x*

def derivFunc( x ):

    return 3 \* x \* x - 2

*# Function to find the root*

def newtonRaphson( x ):

    y = func(x) / derivFunc(x)

    while abs(y) >= 0.0001:

        y = func(x)/derivFunc(x)

        x = x - y

    print("The value of the root is : ","%.4f"% x)

x0 = 2

newtonRaphson(x0)

**Output :-**

****

**Code :- Forward Interpolation**

def u\_c(u, n):

    tp = u

    for i in range(1, n):

        tp = tp \* (u - i)

    return tp

def fact(n):

    f = 1

    for i in range(2, n + 1):

        f \*= i

    return f

n = 6

x = [ 1910, 1915, 1920, 1925, 1930, 1935 ]

y = [[0 for i in range(n)]

        for j in range(n)]

y[0][0] = 5

y[1][0] = 7

y[2][0] = 10

y[3][0] = 15

y[4][0] = 22

y[5][0] = 30

for i in range(1, n):

    for j in range(n - i):

        y[j][i] = y[j + 1][i - 1] - y[j][i - 1]

for i in range(n):

    print(x[i], end = "     ")

    for j in range(n - i):

        print(y[i][j], end = "      ")

    print("")

value = 1917

sum = y[0][0]

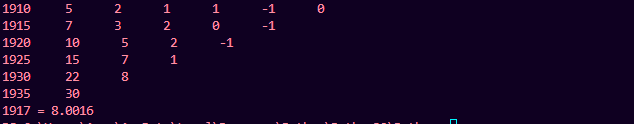
u = (value - x[0]) / (x[1] - x[0])

for i in range(1,n):

    sum = sum + (u\_c(u, i) \* y[0][i]) / fact(i)

print( value,"=", round(sum, 6))

**Output :-**

****

**Trapezoidal rule**

from cmath import pi

import math

def eq(x):

    return math.sin(x)

def trap(x0, xn, inter):

    h  = (xn-x0)/inter

    integration = eq(x0) + eq(xn)

    for i in range (1,inter):

         k = x0 + i\*h

         integration = integration + 2 \* eq(k)

    integration = integration \* h/2

    return integration

lower = 0

upper = pi

n = 6

print(trap(lower,upper,n))

**Output**

****

**Simpson 1/3 –**

from cmath import pi

import math

def eq(x):

    return math.sin(x)

def simp13(x0, xn, inter):

    h  = (xn-x0)/inter

    integration = eq(x0) + eq(xn)

    for i in range(1,n):

        k = x0 + i\*h

        if i%2 == 0:

            integration = integration + 2 \* eq(k)

        else:

            integration = integration + 4 \* eq(k)

*# Finding final integration value*

    integration = integration \* h/3

    return integration

lower = 0

upper = pi

n = 6

print("Integration = : %0.6f"%(simp13(lower,upper,n)))

**Output**

****