

```

import numpy as np
from scipy.integrate import quad
import matplotlib.pyplot as plt

def Exp(x):
    return np.exp(x)

def Sin(x):
    return np.sin(x)

def integrand(x, f, i):
    return f(x) * np.power(x, i)

def interpolate(func, n):
    x1 = np.arange(0, n+1)
    x2 = (np.arange(0, n+1)).reshape(-1, 1)
    A = 1/(x1 + x2 + 1)
    b = np.array([quad(integrand, 0, 1, args=(func, i))[0] for i in range(n+1)])
    cis = np.linalg.solve(A, b)
    return cis

def pnOutput(coff, inputx):
    pw = np.arange(coff.shape[0])
    mid = np.power(inputx.reshape(-1, 1), pw)
    return mid @ coff

def getOutput(func, n, funcName="GroundTruth", savefig=False):
    coff = interpolate(func, n)
    inputx = np.arange(0, 10, 0.2)
    outputy = pnOutput(coff, inputx)
    groundTruth = func(inputx)
    error = np.abs(groundTruth - outputy)
    errorNorm = np.linalg.norm(error)
    plt.plot(inputx, outputy, label=f"approximation order {n}", color="red")
    plt.plot(inputx, groundTruth, label=f"{funcName}", color="blue")
    plt.title(f"n = {n}, error = {errorNorm}")
    plt.legend()
    if savefig:
        plt.savefig(f"hw6q2_{funcName}_order{n}.jpg")
    plt.show()

def getError(func, n):
    coff = interpolate(func, n)
    inputx = np.arange(0, 1, 0.02)
    outputy = pnOutput(coff, inputx)
    groundTruth = func(inputx)
    error = np.abs(groundTruth - outputy)
    errorNorm = np.linalg.norm(error)
    # plt.plot(inputx, outputy, label=f"approximation order {n}", color="red")
    # plt.plot(inputx, groundTruth, label=f"groundTruth", color="blue")
    # plt.title(f"n = {n}, error = {errorNorm}")
    # plt.legend()
    # plt.show()
    return errorNorm

def hw6q2():
    n = np.arange(30)
    sinError = np.array([getError(Sin, i) for i in n])
    expError = np.array([getError(Exp, i) for i in n])
    plt.plot(n, sinError, label = "SinError", color="red")
    plt.plot(n, expError, label = "ExpError", color="blue")
    plt.legend()
    plt.xlabel("Order")
    plt.ylabel("Error")
    plt.title("Error Map")
    plt.savefig("hw6q2.jpg")

```

```
plt.show()
```

```
hw6q2()
```