第三次计算物理作业

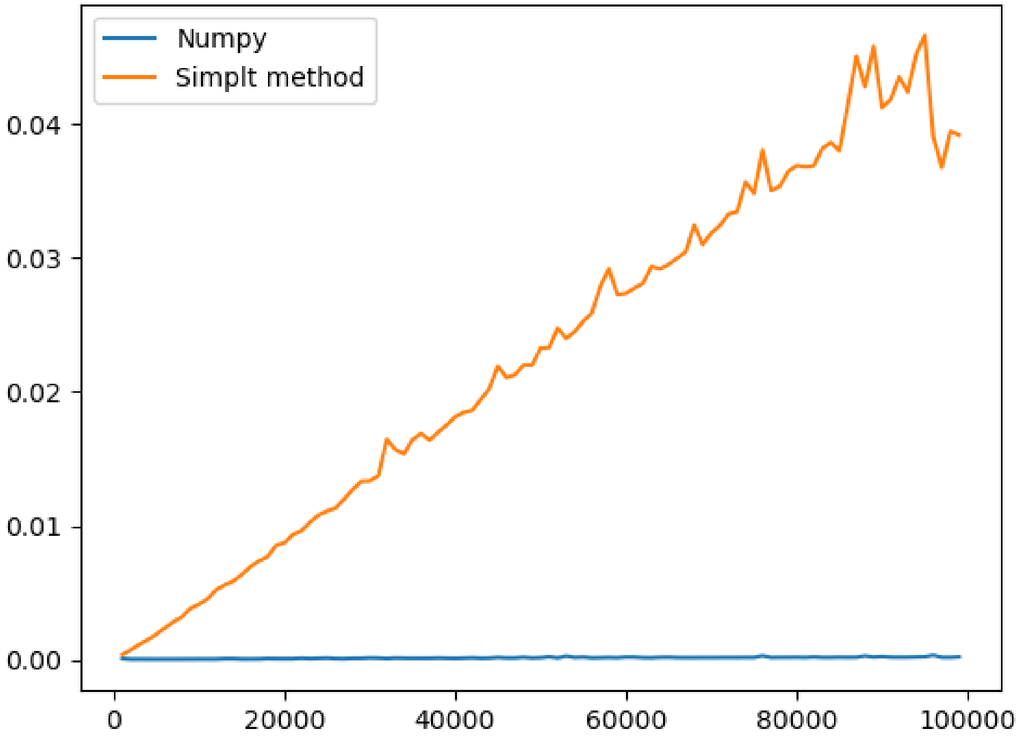
吴远清 2018300001031

第一题

代码:

```
import numpy as np
import time
import matplotlib.pyplot as plt

def numpyUpSum(n):
    s = np.arange(1, n + 1, dtype = np.float32)
    up = 1/s
    return up.sum()

def upSum(n):
    sum = np.float32(0)
    for i in range(1, n + 1):
        sum += np.float32(1/i)
    return sum

def timeTest(func, n):
    st = time.time()
    func(n)
    ed = time.time()
    return ed-st

if __name__ == '__main__':
    numpyTime = []
    normalTime = []
    n = np.array(range(1, 100)) * 1000
    for i in n:
        numpyTime.append(timeTest(numpyUpSum, i))
        normalTime.append(timeTest(upSum, i))
        print(i)
    plt.plot(n, numpyTime, label = 'Numpy')
    plt.plot(n, normalTime, label = 'Simplt method')
    plt.legend()
    plt.show()
```

运行结果:

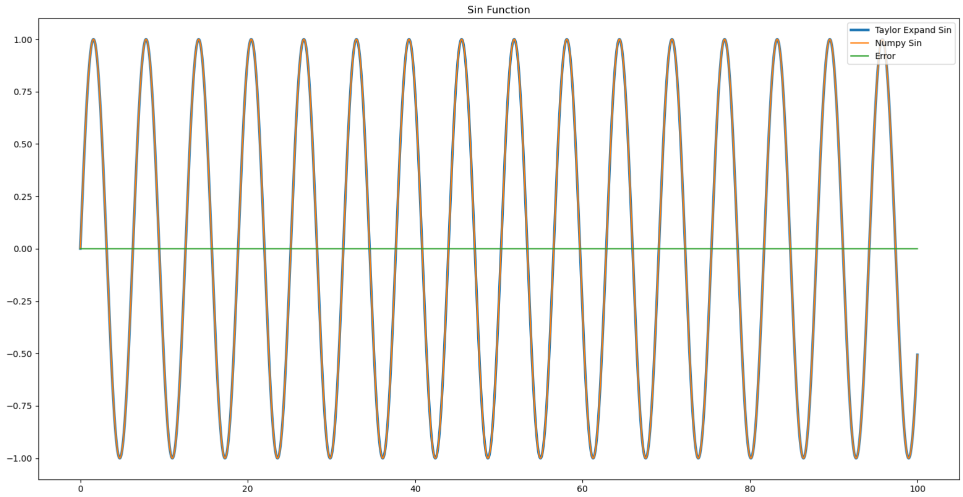


第二题

代码:

```
import numpy as np
import matplotlib.pyplot as plt

def taylorCos(x):
    eps = np.float32(10**-8)
    sum = np.float32(1)
    term = np.float32(1)
    n = 0
    while True:
        n += 1
        term = -x*x*term/(2*n*(2*n-1))
        sum += term
        if abs(term/sum) < eps:
            break
    return sum

def taylorSin(x):
    if x<0:
        return -1 * taylorSin(abs(x))
    elif x > 2*np.pi:
        return taylorSin(x % (2*np.pi))
    elif x > np.pi:
        return -1 * taylorSin(2*np.pi - x)
    elif x > np.pi/2:
        return taylorSin(np.pi - x)
    elif x > np.pi/4:
        return taylorCos(abs(x - np.pi/2))
    else:
        eps = np.float32(10**-8)
        term = np.float32(x)
        sum = np.float32(x)
        n = 1
        while(True):
            n += 1
            term = -x*x*term/((2*n-1)*(2*n-2))
            sum += term
            if np.abs(term/sum) < eps:
                break
        return sum

if __name__ == '__main__':
    x = np.linspace(0.001, 100, 10000)
    y = []
    for i in x:
        y.append(taylorSin(i))
    y = np.array(y)
    yn = np.sin(x)
    plt.title('Sin Function')
    plt.plot(x,y,label = 'Taylor Expand Sin',linewidth=3)
    plt.plot(x, yn, label = 'Numpy Sin')
    plt.plot(x, np.abs(yn - y), label = 'Error')
    plt.legend()
    plt.show()
```

运行结果:



说明:

使用递归调用来将$x \in \mathbb{R}$上的Sin函数映射到$x \in [0, \pi/4]$上的SIn和Cos函数.

Cos函数仿照Sin函数来求值.