

Romberg求积算法与应用

徐锦慧 U202011675 CS2011

代码:

```
#实现Romberg求积算法，并用于求解具体积分实例

from cmath import pi
import numpy as np
import math

#下述函数实现Romberg求积算法

#任务一：请在此处添加代码完成该函数，实现Romberg算法求积计算功能
#####Begin_1#####
def Romberg_quadrature_algorithm(func,a,b,TOL):
    """
    龙贝格求积算法
    求函数func在区间[a,b]上的积分值
    TOL: 误差
    """
    T=[]
    S=[0]
    C=[0,0]
    R=[0,0,0]
    h=b-a    #每次迭代计算更新h步长
    half_list=np.array([(a+b)/2]) #二分点列表

    T.append(0.5*h*(func(a)+func(b)))

    for k in range(1,200):
        T.append(0.5*T[k-1]+0.5*h*np.sum(func(half_list)))
        S.append((4*T[k]-T[k-1])/3)
        if(k>=2):
            C.append((16*S[k]-S[k-1])/15)
        if(k>=3):
            R.append((64*C[k]-C[k-1])/63)
        if(k>=4):
            if(R[k]-R[k-1]<=TOL):
                return R[k]

        h=0.5*h # 更新步长h为原来的一半
        half_list = np.linspace(0, 2 ** k, 2 ** k, endpoint=False) # 计算下一轮所
        # 需的二分点，一共有2^k个点,0,1,2,...,2^k-1
        half_list = (b-a) * (2 * half_list + 1) / (2 ** (k + 1)) + a

    #####End_1#####
```

```

# 任务二：对于函数 $f(x)=4/(1+x^2)$ ，其在 $[0,1]$ 区间的积分等于圆周率，试完善下述代码，
# 调用上述函数求不同精度要求的圆周率的近似值，并打印输出。

#####Begin_2#####

def func1(x):
    y=4/(1 + x**2)
    return y

print('误差<0.1时，圆周率近似值为： ',Romberg_quadrature_algorithm(func1,0,1,0.1))
print('误差<0.01时，圆周率近似值为： ',Romberg_quadrature_algorithm(func1,0,1,0.01))
print('误差<0.001时，圆周率近似值为： ',
      ,Romberg_quadrature_algorithm(func1,0,1,0.001))
print('误差<0.0001时，圆周率近似值为： ',
      ,Romberg_quadrature_algorithm(func1,0,1,0.0001))
print('误差<0.00001时，圆周率近似值为： ',
      ,Romberg_quadrature_algorithm(func1,0,1,0.00001))
print('误差<0.000001时，圆周率近似值为： ',
      ,Romberg_quadrature_algorithm(func1,0,1,0.000001))
print('误差<0.0000001时，圆周率近似值为： ',
      ,Romberg_quadrature_algorithm(func1,0,1,0.0000001))
print('误差<0.00000001时，圆周率近似值为： ',
      ,Romberg_quadrature_algorithm(func1,0,1,0.00000001))
print('误差<0.000000001时，圆周率近似值为： ',
      ,Romberg_quadrature_algorithm(func1,0,1,0.000000001))

#####End_2#####

```

```

# 任务三：人造地球卫星的轨道可视为平面上的椭圆，通过卫星地面控制中心可以测出卫星的
# 近地点距离与远地点距离，进而估计人造地球卫星的轨道长度。
# 我国第一颗人造地球卫星，
# 它的近地点距离为 $h_1=439\text{km}$ ，远地点距离为 $h_2=2384\text{km}$ ，地球半径 $R$ 约为 $6371\text{km}$ ，试建立数学模型，
# 使用积分法求出该人造地球卫星的轨道长度。在下方补充代码，调用Romberg_quadrature_alg.，
# 实现此计算功能。

```

```

#####Begin_3#####

def func2(x):
    a=(439+2384+2*6371)/2
    b=(2384-439)/2
    return (a**2*np.cos(x)**2+b**2*np.sin(x)**2)**(0.5)

print('\n')
print ('人造地球卫星的轨道长度近似值为： ',
4*Romberg_quadrature_algorithm(func2,0,pi/2,0.000000001))

#####End_3#####

```

运行结果:

误差<0.1时, 圆周率近似值为: 3.141592638396796
误差<0.01时, 圆周率近似值为: 3.141592638396796
误差<0.001时, 圆周率近似值为: 3.141592638396796
误差<0.0001时, 圆周率近似值为: 3.141592638396796
误差<0.00001时, 圆周率近似值为: 3.141592638396796
误差<0.000001时, 圆周率近似值为: 3.1415926535900294
误差<0.0000001时, 圆周率近似值为: 3.1415926535900294
误差<0.00000001时, 圆周率近似值为: 3.1415926535897936
误差<0.000000001时, 圆周率近似值为: 3.1415926535897936
误差<0.0000000001时, 圆周率近似值为: 3.1415926535897936

人造地球卫星的轨道长度近似值为: 31853.846345396923