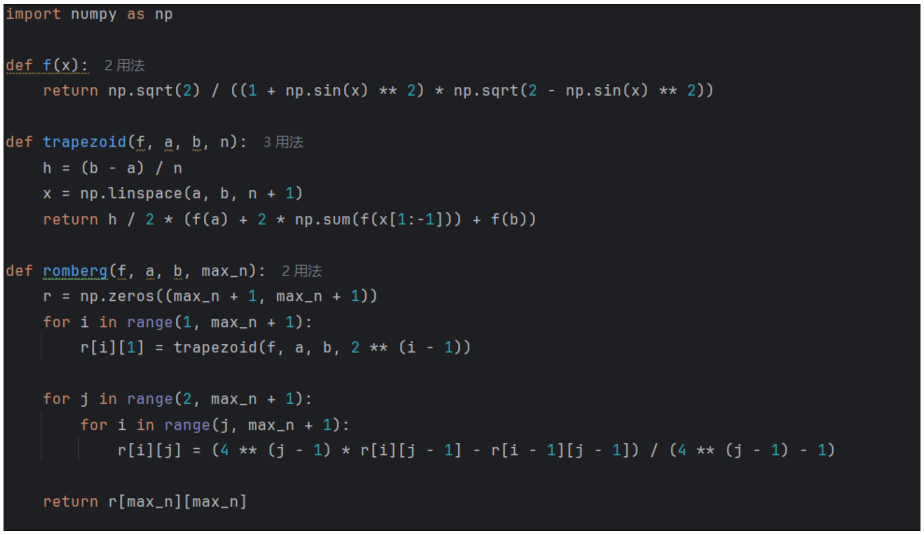
第五章作业-2

李显

2024 年 11 月 4 日

5.7:

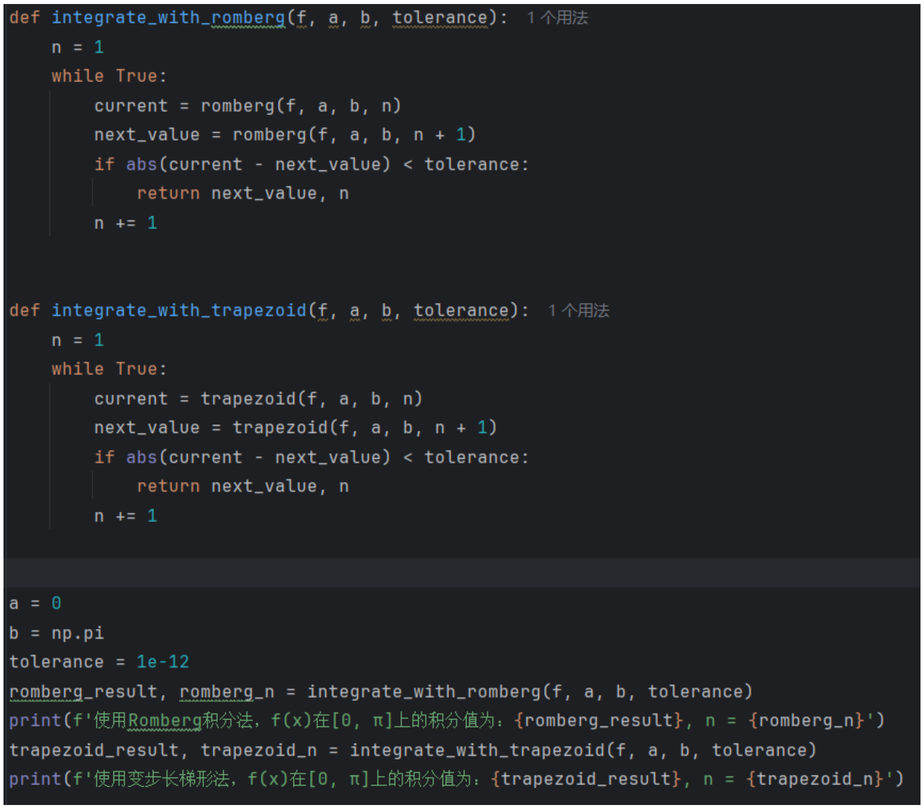
编写函数部分 python 代码如图：



```python
import numpy as np

def f(x):
    return np.sqrt(2) / ((1 + np.sin(x) ** 2) * np.sqrt(2 - np.sin(x) ** 2))

def trapezoid(f, a, b, n):
    h = (b - a) / n
    x = np.linspace(a, b, n + 1)
    return h / 2 * (f(a) + 2 * np.sum(f(x[1:-1])) + f(b))

def romberg(f, a, b, max_n):
    r = np.zeros((max_n + 1, max_n + 1))
    for i in range(1, max_n + 1):
        r[i][1] = trapezoid(f, a, b, 2 ** (i - 1))

    for j in range(2, max_n + 1):
        for i in range(j, max_n + 1):
            r[i][j] = (4 ** (j - 1) * r[i][j - 1] - r[i - 1][j - 1]) / (4 ** (j - 1) - 1)

    return r[max_n][max_n]
```

图 1: 函数代码

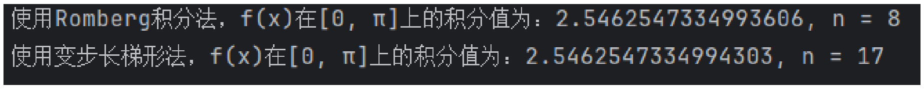
再编写积分代码，并进行运行：



```
def integrate_with_romberg(f, a, b, tolerance):
    n = 1
    while True:
        current = romberg(f, a, b, n)
        next_value = romberg(f, a, b, n + 1)
        if abs(current - next_value) < tolerance:
            return next_value, n
        n += 1


def integrate_with_trapezoid(f, a, b, tolerance):
    n = 1
    while True:
        current = trapezoid(f, a, b, n)
        next_value = trapezoid(f, a, b, n + 1)
        if abs(current - next_value) < tolerance:
            return next_value, n
        n += 1


a = 0
b = np.pi
tolerance = 1e-12
romberg_result, romberg_n = integrate_with_romberg(f, a, b, tolerance)
print(f'使用Romberg积分法，f(x)在[0, π]上的积分值为：{romberg_result}, n = {romberg_n}')
trapezoid_result, trapezoid_n = integrate_with_trapezoid(f, a, b, tolerance)
print(f'使用变步长梯形法，f(x)在[0, π]上的积分值为：{trapezoid_result}, n = {trapezoid_n}')
```

图 2: 积分代码

运行结果如图:



```
使用Romberg积分法，f(x)在[0, π]上的积分值为：2.5462547334993606, n = 8
使用变步长梯形法，f(x)在[0, π]上的积分值为：2.5462547334994303, n = 17
```

图 3: 结果