

蒙特卡洛算法实验报告

2023200440

2024 年 10 月 24 日

1 实验目的

本次实验是通过蒙特卡洛算法计算 π 的近似值，了解概率统计理论在实际计算问题中的应用。

2 实验环境

机器：windows11

解释器：python3.12

编辑器：pycharm

3 实验方法

蒙特卡洛方法：利用计算机随机生成横纵坐标在 $[0, 1]$ 内的随机点，统计落在单位圆内的点的个数，计算这些点占所有点的比例乘以 4 即得到 π 的估计值

3.1 实验步骤

1. 通过 random 库随机生成坐标范围在 $[0, 1]$ 内的点
2. 计算每个点是否在单位圆内，如果在，points 数 +1
3. 计算 π 的近似值： $(\text{points} / \text{num_samples}) * 4$

4 代码细节

1. 代码如下

```
import random

def monte_carlo_pi(num_samples):
    """
    通过蒙特卡洛方法计算  $\pi$  的近似值
    参数:
        num_samples: 生成随机点的个数
    """
    points = 0    # 生成的随机点在圆内的点的数目
    for _ in range(num_samples):
        x, y = random.random(), random.random() # 在[0, 1]区间内随机生成一个点
        if x**2 + y**2 <= 1: # 如果点在圆内
            points += 1

    return (points / num_samples) * 4 # 圆面积与正方形面积的比乘以4即为  $\pi$  的近似值

if __name__ == "__main__":

    num_samples = 1000000 # 生成点的样本数
    estimate_pi = monte_carlo_pi(num_samples)
    print(f" $\pi$  的近似值是: {estimate_pi}")
```

5 实验结果

5.1 结果展示

实验结果如下:

num_samples	estimate_pi
10	2.4
100	3.2
1000	3.192
10000	3.1408
100000	3.14416
1000000	3.143128
10000000	3.1416868

表 1: Caption

```

D:\Anaconda\envs\env1\python.exe D:\CODE_REPOSITORY\AI引论\作业\2023200440_李应_lab4\code.py
点的样本数为: 1000000, pi的近似值是: 3.139156
进程已结束, 退出代码为 0

```

图 1: 结果

6 实验结果分析

总体上, 当 num_samples 越大, 计算出的 π 值越精确