

设计 MonteCarlo 法计算圆周率的 Tcl 程序并观察收敛性质

——景宇 1216073606 18362972083

一、需求分析

1. 仿真目标说明

圆周率是圆的周长与直径之比，是数学中的一个重要常数，也是迄今为止计算时间最长，计算得到位数最多的一个常数。人们之所以对 π 的计算进行研究，不仅是为了找到更加精准的圆周率，更重要的是为了找到一些新的计算方法、改进计算的手段，并且 π 值是用来来检验计算机可靠性、精准性、运算速度及计算容量的有力办法、手段和衡量计算进展。本文设计 MonteCarlo 法计算圆周率的 Tcl 程序并观察收敛性质。

2. 结果预期

设计用计算机进行模拟随机试验，以此事件发生的频率估计概率，进而得到 π 的近似值。并使用大量的计算次数来观察计算结果，看其是否接近 π 的真实值，收敛性质如何。

二、仿真设计

1. Monte Carlo 方法介绍

蒙特卡罗方法又称统计模拟法、随机抽样技术，是一种随机模拟方法，以概率和统计理论方法为基础的一种计算方法，是使用随机数(或更常见的伪随机数)来解决很多计算问题的方法。将所求解的问题同一定的概率模型相联系，用电子计算机实现统计模拟或抽样，以获得问题的近似解。

Monte Carlo 方法的基本思想很早以前就被人们所发现和利用。早在 17 世纪，人们就知道用事件发生的“频率”来决定事件的“概率”。19 世纪人们用投针试验的方法来决定圆周率 π 。本世纪 40 年代电子计算机的出现，特别是近年来高速电子计算机的出现，使得用数学方法在计算机上大量、快速地模拟这样的试验成为可能。

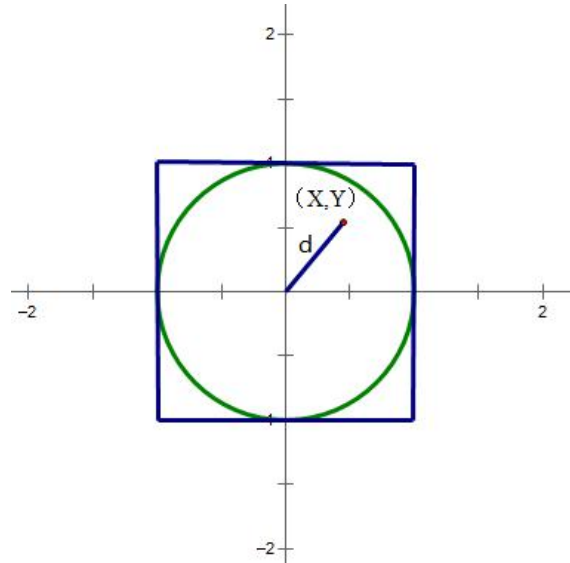
2. π 计算模型

通过蒙特卡罗算法计算圆周率的主导思想是：统计学（概率）。

一个正方形有一个内切圆，向这个正方形内随机的画点，则点落在圆内的概率 P 为圆面积与正方形面积的商。

(1) 在一个平面直角坐标系下，在点 (0, 0) 处画一个半径为 $R=1$ 的圆, 以这个圆画一个外接正方形，其边长为 $2R=2$ ($R=1$ 时，圆面积即 π)。

(2) 随机取一点 (X, Y) 使得 $-R \leq X \leq R$ 并且 $-R \leq Y \leq R$ ，即随机点在正方形内。



(3) 判断点是否在圆内，通过公式 $d^2 = X^2 + Y^2 \leq R^2$ 计算。

(4) 设所有点的个数为 N，落在圆内的点的个数为 M，则

$$P = M/N = 4 * R * R / \pi * R * R = \pi/4$$

$$\pi = 4 * M/N$$

当实验次数越多（N 越大），所计算出的 Pi 也越准确。

3. Tcl 仿真程序

```
proc pi-eval t {
    set c 0
    for {set i 0} {$i < $t} {incr i} {
        set x [expr rand()]
        set y [expr rand()]
        set d2 [expr $x * $x + $y * $y]
        if {$d2 < 1.0} {
            incr c
        }
    }
    return [expr 4 * double($c) / $t]
}
set round 10
puts "$round\t[pi-eval $round]"
```

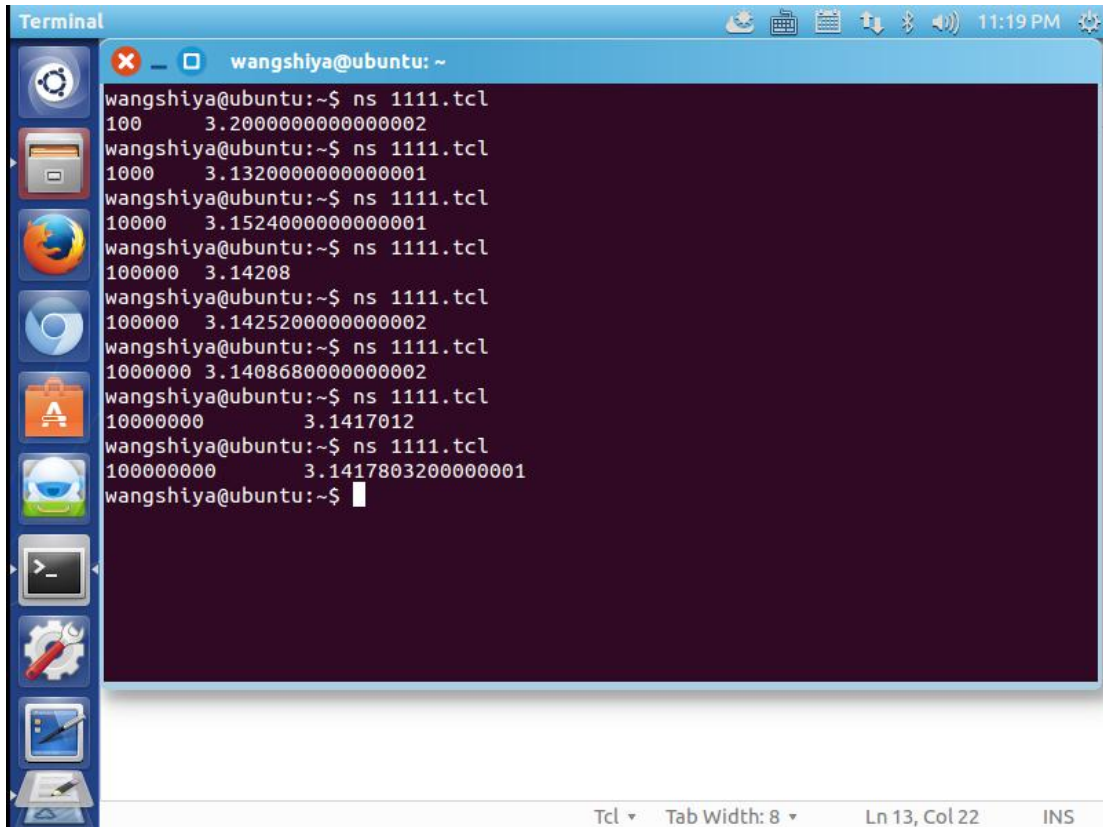
三、结果与分析

1. 仿真结果

100	3.20000000
1000	3.13200000

10000	3.15240000
100000	3.14252000
1000000	3.14086800
10000000	3.14170120
100000000	3.14178032

2. 仿真结果截图



```
wangshiya@ubuntu:~$ ns 1111.tcl
100      3.2000000000000002
wangshiya@ubuntu:~$ ns 1111.tcl
1000     3.1320000000000001
wangshiya@ubuntu:~$ ns 1111.tcl
10000    3.1524000000000001
wangshiya@ubuntu:~$ ns 1111.tcl
100000   3.14208
wangshiya@ubuntu:~$ ns 1111.tcl
100000   3.1425200000000002
wangshiya@ubuntu:~$ ns 1111.tcl
1000000  3.1408680000000002
wangshiya@ubuntu:~$ ns 1111.tcl
10000000 3.1417012
wangshiya@ubuntu:~$ ns 1111.tcl
100000000 3.1417803200000001
wangshiya@ubuntu:~$
```

3. 结果分析

根据实验数据可以隐约看出，当实验次数越多（N 越大），所计算出的 Pi 也越准确，逐渐逼近 π 的值。但是尽管 N 的数值已经很大，计算值也还是有一点差距。

这是因为计算机上的随机数毕竟是伪随机数，当取值超过一定值，也会出现不随机现象，因为伪随机数是周期函数。随机数产生器在 NS2 中是非常重要的，举凡网络节点在 NAM 中的位置或者是应用程序在何时开始传送或结束数据传输，都会用到随机数产生器。随机数产生器所产生的数值是由种子和分布所控制的，不同的种子或者是分布就会产生出不同的随机数。

这样就会影响 Pi 的计算结果的准确度，想要达到我们预期的效果，我们还需要编写真正的随机数产生器，还要学习很多。