计算物理第一次作业

姓名：姚星宇　　学号：PB21000188　　班级：2021 级少年班学院四班

2022 年 9 月 30 日

1 题目重述

1.1 Schrage 方法

用 Schrage 方法编写随机数子程序，用指定间隔（非连续 $l>1$）两个随机数作为点的坐标值绘出若干点的平面分布图。再用 $<x^k>$ 测试均匀性（取不同量级的 N 值，讨论偏差与 N 的关系）、$C(l)$ 测试其 2 维独立性（总点数 $N>10^7$）。

1.2 序列检测

用 16807 产生器测试随机数序列中满足关系 $X_{n-1}>X_{n+1}>X_n$ 的比重。讨论 Fibonacci 延迟产生器中出现这种关系的比重。

1.3 球坐标抽样

在球坐标系 (ρ,θ,ϕ) 下，产生上半球面上均匀分布的随机坐标点，给出其直接抽样方法。

1.4 pdf 函数抽样

设 pdf 函数满足关系式：

$$p'(x)=a\delta(x)+bexp(-cx),x\in[-1,1] \tag{1}$$

讨论该函数性质并给出抽样方法。

1.5 Marsaglia 抽样

对于球面上均匀分布的随机坐标点，给出它们在（x, y）平面上投影的几率分布函数。并由此验证 Marsaglia 抽样方法 $x=2u\sqrt{1-r^2},y=2v\sqrt{1-r^2},z=1-2r^2$ 确为球面上均匀分布的随机抽样。

2 题目分析

本作业使用 Python（Anaconda 3）进行代码编写。

2.1 第一题分析

Schrage 方法的关键如下：令 $q=[\frac{m}{a}],r=m \mod a$, 则有：

$$az \mod m=\{\frac{z}{q}(aq+r)-\frac{rz}{q}\} \mod m=a(z \mod q)-r[\frac{z}{q}] \tag{2}$$

即可得到由原随机数得到新的随机数。

由 Python 语言的特性，本作业采用 yeild 关键字来实现迭代式生成随机数，以避免生成两组相同的随机数列表浪费内存。生成随机数时先计算出 q 与 r，然后代入公式，即可得到新的随机数。

对于画出平面分布图，若储存随机数的 list 为 m，则可直接将 m[::2] 与 m[1::2] 作为输入，利用 matplotlib 绘制散点图。

对于计算 C(l)，则可直接将 m 转换为 nd_array 数据类型并进行形如 $a = m[l:], b = m[:-l]$ 的切片，并计算平均值 $< x_n >= b.sum()/N, < x_n x_{n+l} >= (a * b).sum()/N, < x_n^2 >= (b * b).sum()/N$ 即可得到各个平均值。需要注意的是，这里如果 N 较大，相乘求和后的结果容易超过（Anaconda 3 中的）整型储存上限，故可以先相除再相乘求和，以避免这种情况。

对于计算 $< x^k >$，可直接将 m 转换为 nd_array 数据类型，随后为了防止溢出而首先归一化（$m/m.max()$）然后进行 k 次方平均数 ($(m * *k).sum()/N$)。需要注意的是，此处的期望值为

$$\bar{(}x) = \int_0^1 x^k dx = \frac{1}{k+1} \tag{3}$$

故最后算得与期望值的相对偏差为 $(k+1) < x^k > -1$。

2.2 第二题分析

对于 16807 产生器，可直接对存储在 nd_array 中的随机数序列切片得到 $delta1 = m[:-2] - m[2:], delta2 = m[2:] - m[1:-1]$，并计算 $(np.absolute(delta1)/delta1 - 1) * (np.absolute(delta2)/delta2 - 1)$ 以得到符合条件的个数。

对于 Fibonacci 随机数产生器，产生该关系的比重取决于具体的抽样方法。例如，最原始的 Fibonacci 数列进行随机数生成 $I_n = I_{n-1} + I_{n-2} \mod 2^{32}, I_1 = 1, I_2 = 1$，则产生的比重约为 $\frac{1}{\log_{1.62} 2^{32}} = 0.022$

2.3 第三题分析

对于上半球面上按**立体角（或表面积）**均匀分布的随机坐标点，有：

$$dP(\theta \in [0, \theta], \phi \in [0, \phi]) = \frac{d\Omega}{\Omega} = \frac{1}{2\pi} \sin\theta d\theta d\phi \tag{4}$$

即有 $p(\theta, \phi) = \frac{1}{2\pi} \sin\theta$, 则可得：

$$\begin{cases} \xi_1 = \int_0^{2\pi} d\phi \int_0^{\xi_\theta} \frac{1}{2\pi} \sin\theta = 1 - \cos\xi_\theta \\ \xi_2 = \int_0^{\xi_\phi} d\phi \int_0^{\frac{1}{2}\pi} \frac{1}{2\pi} \sin\theta = \frac{\xi_\phi}{2\pi} \end{cases}$$

即：

$$\begin{cases} \xi_\phi = 2\pi\xi_2 \\ \xi_\theta = \arccos\xi_1 \end{cases}$$

其中 ξ_1, ξ_2 为 [0,1] 间均匀分布的随机变量，生成方式可为上述的 16807 随机数生成器除以 $(2^{31} - 1)$ 来得到。再调用 matplotlib 的 3 维绘图库，即可得到结果。

2.4 第四题分析

对于题给 pdf 函数 $p'(x) = a\delta(x) + bexp(-cx), x \in [-1,1]$，首先有归一化关系：

$$\frac{2b}{c}\sinh c + a = 1 \tag{5}$$

这里可令 $\xi_1 \in [0,1]$, 若 $\xi_1 < a$, 则 $\xi_x = 0$；若 $\xi_1 > a$，则令 $\xi_2 = \frac{\xi_1 - a}{1-a}$ ，则有 $\int_{-1}^{\xi_x} bexp(-cx)dx = \xi_2 \int_{-1}^{1} bexp(-cx)dx$, 即有 $\xi_x = -\frac{\ln(e^c - 2\xi_2 \sinh c)}{c}$，即可得到对 pdf 函数的直接抽样。

2.5 第五题分析

如第三题的分析，可得 $p(\theta,\phi) = \frac{1}{2\pi}\sin\theta$，又有：$\theta = \arcsin\sqrt{x^2+y^2}, \phi = \arctan\frac{y}{x}$。

则有：

$$p(x,y) = p(\theta,\phi)|\frac{\partial(\theta,\phi)}{\partial(x,y)}| = \frac{1}{2\pi\sqrt{1-x^2-y^2}} \tag{6}$$

此即题中所求分布函数。

抽样的方法严格按讲义中的 Marsaglia 抽样进行，在此不赘述。

3 代码

```
# -*- coding: utf-8 -*-
import matplotlib.pyplot as plt
import numpy as np
from mpl_toolkits.mplot3d import Axes3D


def sch_random(N, M = 1, a = 16807, b = 0, m = 2**31 - 1, seed = 1):#N 为生成个数，M 为生成间隔
    q, r = m // a, m % a    #得到 p，r
    for i in range(N):
        for j in range(M):
            seed = a * (seed % q) -r * (seed // q) #进行 schrage 方法
            if seed < 0:
                seed += m
        yield seed  #产生可迭代对象，简化代码


def distribution(f_random, seed = 114, N = 100000, M = 1):#传入函数以减少耦合，下同
    l = list(f_random(2*N, M, seed = seed))#使用 list 将可迭代对象转换成 list 对象，下同
    x, y = l[::2], l[1::2] #进行切片，得到相邻的列，便于绘图
```

```
    fig, ax = plt.subplots(1, 1, figsize = (60,60))
    ax.plot(x, y, color = 'r', linestyle = 'None', marker = '.')
    fig.show()
    return True


def compare(f_random, seed = 1919, N = 100000):
    l = list(f_random(3*N, seed = seed))
    x, y, z = np.array(l[::3]), np.array(l[1::3]), np.array(l[2::3])#进行切片
    res = (x-z)/np.absolute(x-z)+(z-y)/np.absolute(z-y)#使用 numpy 自带的矩阵计算简化代码
    res = (res +np.absolute(res)).sum()/4#使用绝对值函数直接获得个数
    return res/N


def powered_k(f_random,seed=1919,N=100000,k=1):
    m = np.array(list(f_random(N,seed = seed)))
    m = m/m.max()
    return (m**k).sum()*(k+1)/N-1


def independence(f_random, seed =810, N=100000, l0 = 1):#C(l) 检验
    l = list(f_random(N+l0, seed = seed))
    a, b = np.array(l[l0:])/N, np.array(l[:-l0])/N# 在此处先除以 N 是为了防止溢出
    avg1, avg2, avg3 = b.sum() , (b*b).sum()*N , (a*b).sum()*N
    return (avg3 - avg1 ** 2) / (avg2 - avg1 ** 2)


def sph_sam(f_random,seed1 = 114514,seed2 = 1919810,N=1000):#上半球面抽样
    theta = np.arccos(np.array(list(f_random(N, seed = seed1)))/(2**31 - 1))#具体推导见文档
    phi = 2*3.1415926535*(np.array(list(f_random(N, seed = seed2)))/(2**31 - 1))
    x = np.cos(phi)*np.sin(theta)
    y = np.sin(phi)*np.sin(theta)
    z = np.cos(theta)
    fig = plt.figure()
    ax = fig.add_subplot(111, projection='3d')
    plt.rcParams['savefig.dpi']=1000
```

```python
    plt.rcParams['figure.dpi']=1000
    ax.scatter(x, y, z, c='r', marker='*')
    ax.set_xlabel('X')
    ax.set_ylabel('Y')
    ax.set_zlabel('Z')
    plt.show()


def mar_sam(f_random,seed1=114514,seed2=1919810,N=1000):#Marsaglia 抽样
    u = (np.array(list(f_random(N=int(1.27324*N), seed = seed1)))/(2**31 - 1))*2-1
    v = (np.array(list(f_random(N=int(1.27324*N), seed = seed2)))/(2**31 - 1))*2-1
    r2 = u**2+v**2
    t = (1-(r2-1)/np.absolute(r2-1))/2
    u,v,r2 = u*t,v*t,r2*t
    x,y = u*np.sqrt(1-r2),v*np.sqrt(1-r2)
    fig, ax = plt.subplots(1, 1, figsize = (10,10))
    ax.plot(x, y, color = 'r', linestyle = 'None', marker = '*')
    fig.show()


if __name__ == '__main__':
    distribution(sch_random,N=int(1e7),l=1)
    distribution(sch_random,N=int(1e7),l=4)
    independence(sch_random,l0=1)
    independence(sch_random,l0=4)
    print(powered_k(sch_random,k=1,N=int(1e8)))
    sph_sam(sch_random)
    mar_sam(sch_random)
```

3.1 结果

相邻随机数作为坐标如图 1 所示。

$< x^k >$ 分析表示，对于 16807 产生器，N 越大，相对偏差越小。如 $N \sim 10^5$ 时 $\frac{<x^k>}{\bar{x}} - 1 \sim 10^{-4}$，$N \sim 10^7$ 时 $\frac{<x^k>}{\bar{x}} - 1 \sim 10^{-5}$ 故 16807 产生器具有相当好的均一性。

C(l) 分析表示，N=1e7 时，C(1)=-0.00136，C(2)=-0.00136,C(5)=-0.00150, 故均较小，可认为具有相当好的独立性。

N=1e7 时，满足关系 $X_{n-1} > X_{n+1} > X_n$ 的比重约为 0.1666。

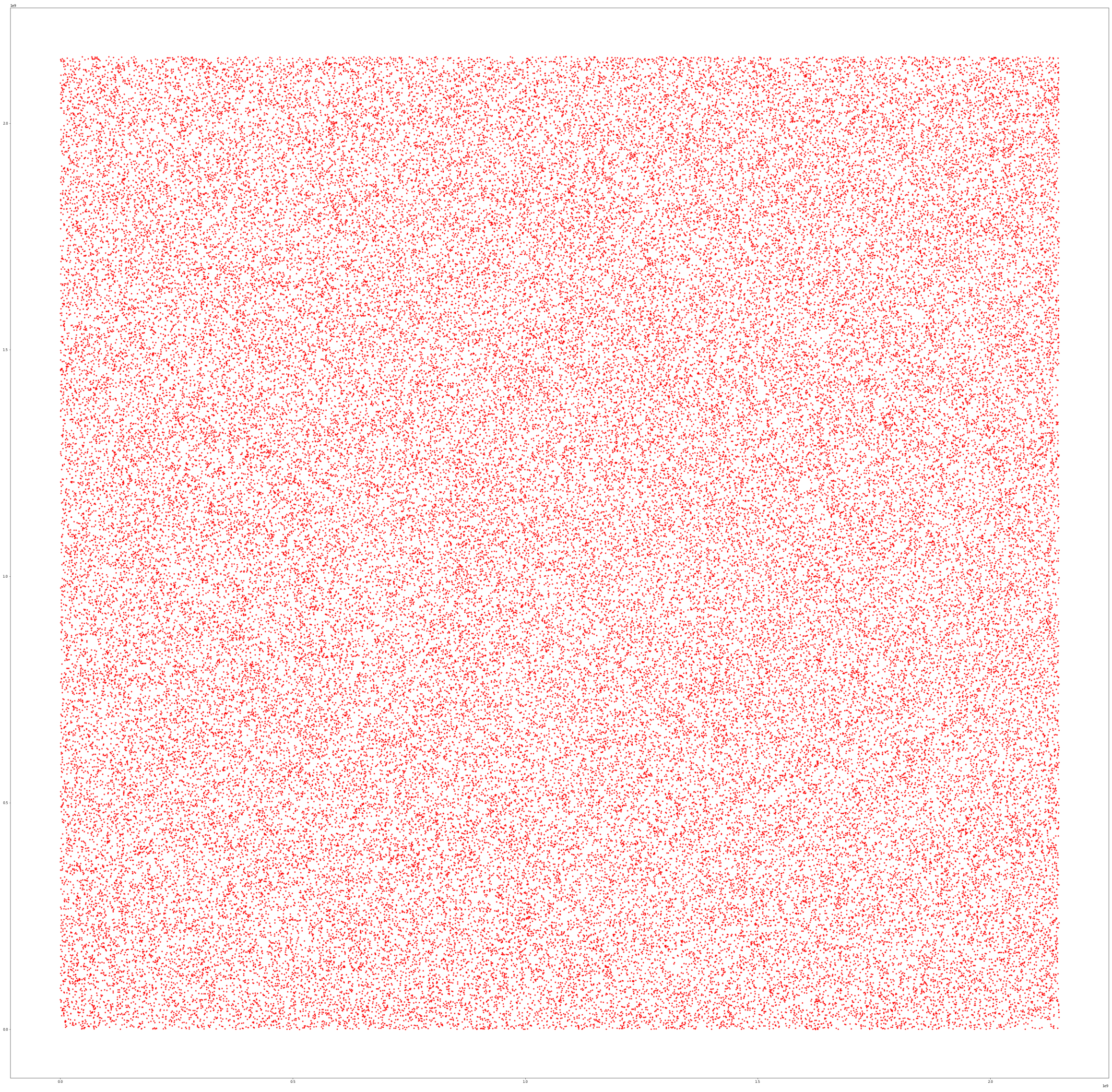

图 1: 以相邻随机数作为坐标得到的绘图

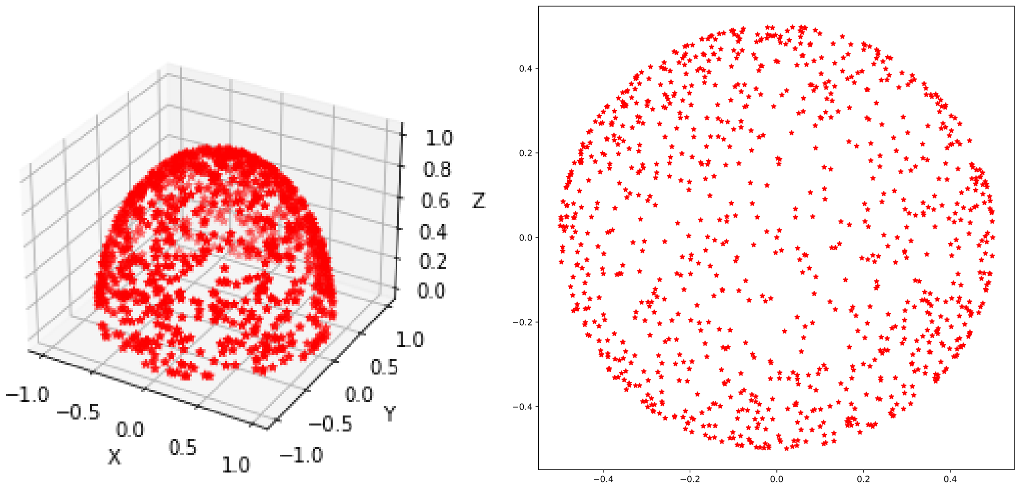

图 2: 上半球直接抽样，Marsaglia 抽样

抽样如图所示，可直观看出具有良好的均匀性，较好地达到了目标需求。

参考文献

[1] 丁泽军. 计算物理讲义 [M]