计算物理——Homework5

曾郅琛 PB20071431

摘要： 利用C++语言以及Python中mpl_toolkits.mplot3d解决以下问题：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抽样方法确为球面上均匀分布的随机抽样。首先推导球面上均匀分布的随机坐标点在（x, y）平面上投影的几率分布函数，之后在实验中采用舍选法抽样在python中画出三维图以验证Marsaglia方法，并将其投影到xy、yz、xz坐标与理论推导比对，得出结果。

1 算法及实现

1.1 球面均匀点（x, y）平面上投影的几率分布函数推导

对于球面上的均匀分布，根据数学知识应该用单位立体角点数均匀来表示，在单位球面上：

$$d\Omega = r^2 sin(\theta)d\theta d\phi = sin(\theta)d\theta d\phi$$

均匀分布则要求：

$$\frac{d\Omega}{\Omega} = \frac{dN}{N},\ and\ \Omega = 4\pi$$

所以在球面上均匀分布点的概率密度函数：

$$g(\theta,\phi)d\theta d\phi = \frac{sin(\theta)}{4\pi}d\theta d\phi$$

根据Jacobi变换：

$$g(\theta,\phi)d\theta d\phi = f(x,y)dxdy$$
$$\frac{\partial(\theta,\phi)}{\partial(x,y)} = \frac{1}{sin(\theta)cos(\theta)}$$

所以在(x, y)下表示有概率密度函数：

$$f(x,y) = g(\theta,\phi)|\frac{\partial(\theta,\phi)}{\partial(x,y)}| = 2\cdot\frac{sin(\theta)}{4\pi}\frac{1}{sin(\theta)cos(\theta)} = \frac{1}{2\pi cos(\theta)}$$

最后利用直角坐标系和球坐标系转化公式，即可得到(x, y)下表示有概率密度函数：

$$z = cos(\theta)$$
$$=> f(x,y) = \frac{1}{2\pi z} = \frac{1}{2\pi\sqrt{1-x^2-y^2}}$$

1.2 Marsaglia抽样方法

在实验一的基础上，利用实验一的16807随机数生成器，生成"Random_u.txt"、"Random_v.txt"随机数作为初始随机数，通过判断u^2+v^2与1的大小关系，若大于1则重新抽样，直到小于1为止，再通过Marsaglia抽样方法公式计算(x, y, z)保存在generate_dot_N.txt中，其中N为抽样点数。下为核心代码：

```
FILE *fp1 = fopen("Random_u.txt","r");                    //生成u随机数
FILE *fp2 = fopen("Random_v.txt","r");                    //生成v随机数
FILE *f1 = fopen("generate_dot_10000.txt","w");                //存储(x,y,z)点
while( i ){
    fscanf(fp1, "%lf", &u);
```

```
        fscanf(fp2, "%lf", &v);
        r2 = u * u + v * v;                                          //r^2值
        if (r2 > 1) ;                                         //判断：若r^2>1则重新取样
        else {
            x = 2 * u * pow(1 - r2, 0.5);
            y = 2 * v * pow(1 - r2, 0.5);
            z = 1 - 2 * r2;
            fprintf(f1, "%lf,%lf,%lf\n", x,y,z);                       //计算并返回(x,y,z)
        i--;
        }
        j++;
    }
std::cout<<"Marsaglia抽样方法效率："<<(double)i0*1.0/j<<endl;
```

同时记录抽样过程，打印抽样效率。

2 实验结果分析及讨论

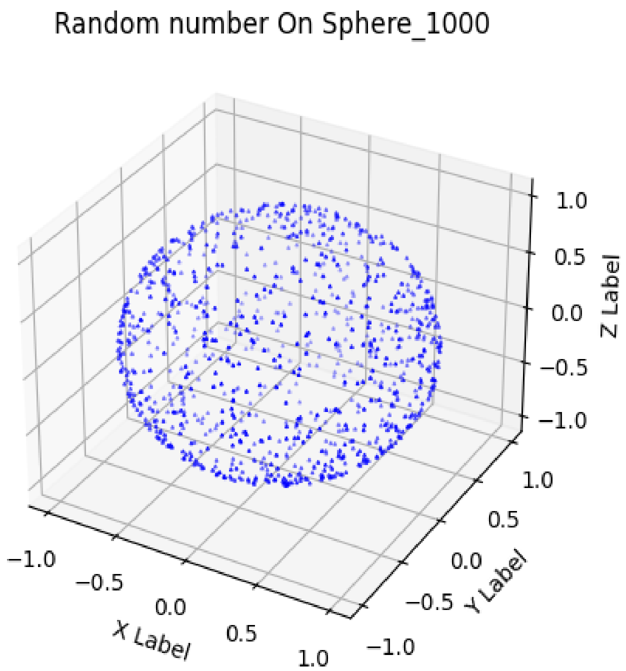
2.1 Marsaglia抽样方法效率

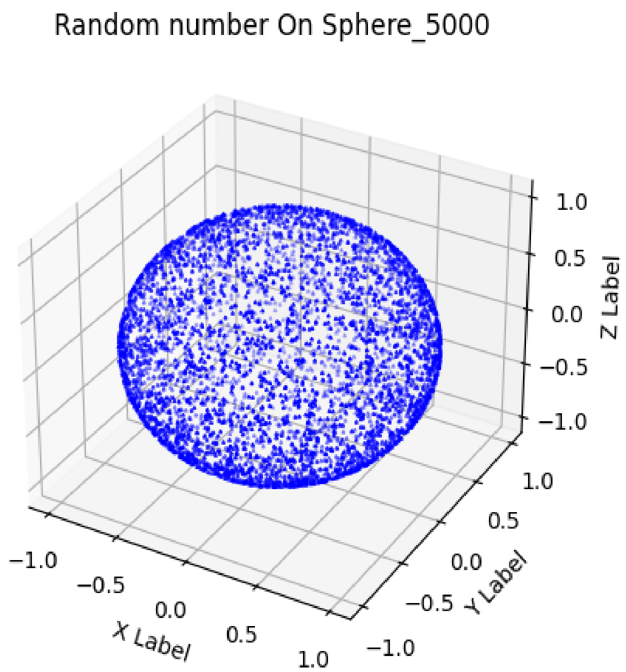
```
Please input N:1000
舍选法效率：0.803859

Please input N:5000
舍选法效率：0.783576

Please input N:10000
舍选法效率：0.792959
```

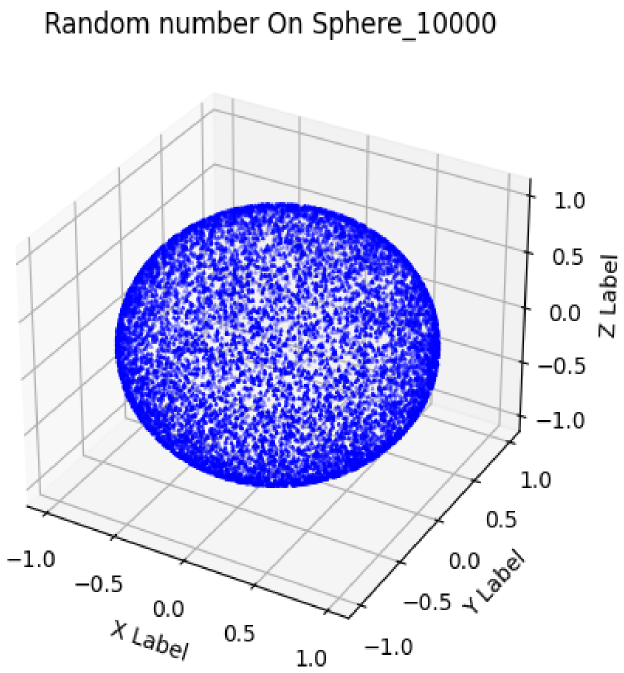
通过选取$u^2+v^2>1$的数据得到想要的抽样点，最终抽样效率都接近79%左右，而理论取样是从边长为2的正方形取出单位圆，抽样效率$\pi/4 \approx 0.785398$，与实际吻合较好，说明抽样成功！

2.2 三维画图

与前几次实验相同，读取文件中点坐标，在python中调用包绘图：首先是三维图，验证Marsaglia抽样方法确为球面上均匀分布的随机抽样：


Random number On Sphere_1000
X Label
Y Label
Z Label
−1.0
−0.5
0.0
0.5
1.0

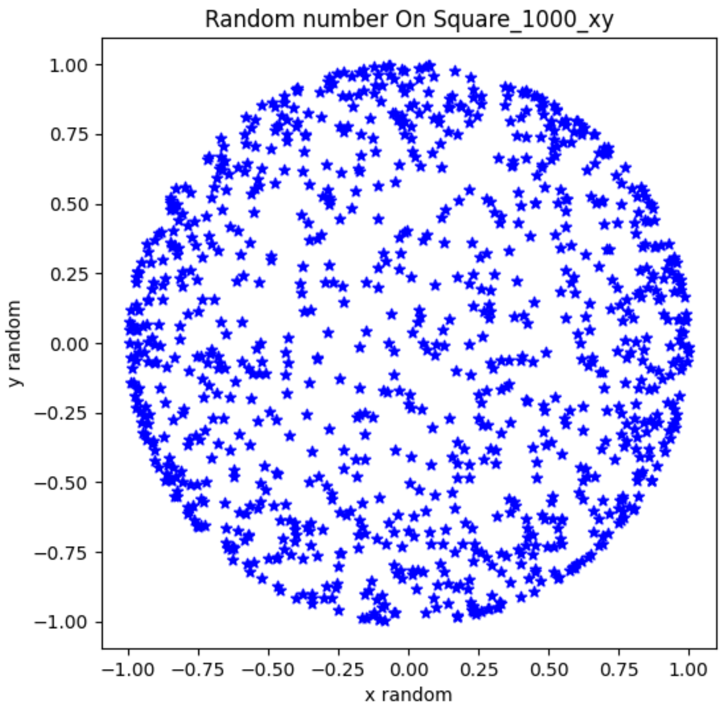

Random number On Sphere_5000
X Label
Y Label
Z Label
−1.0
−0.5
0.0
0.5
1.0

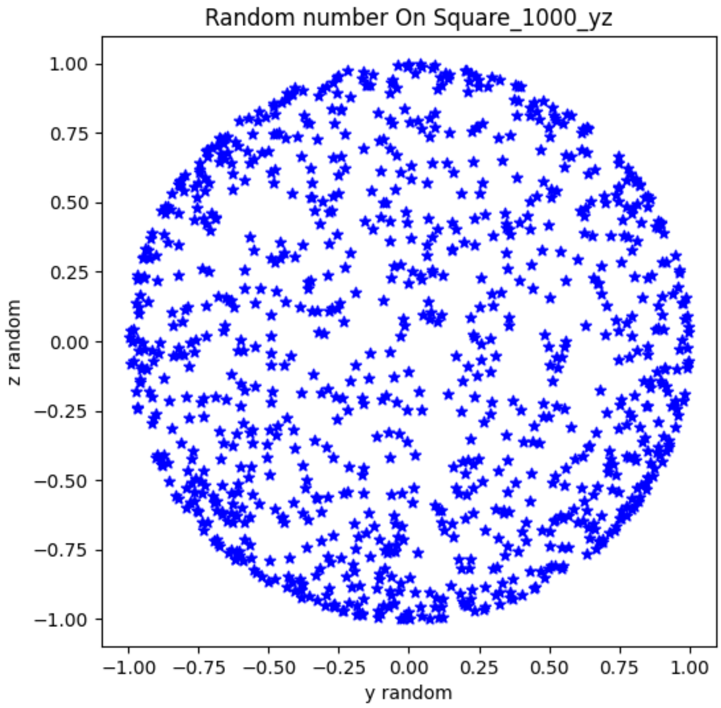


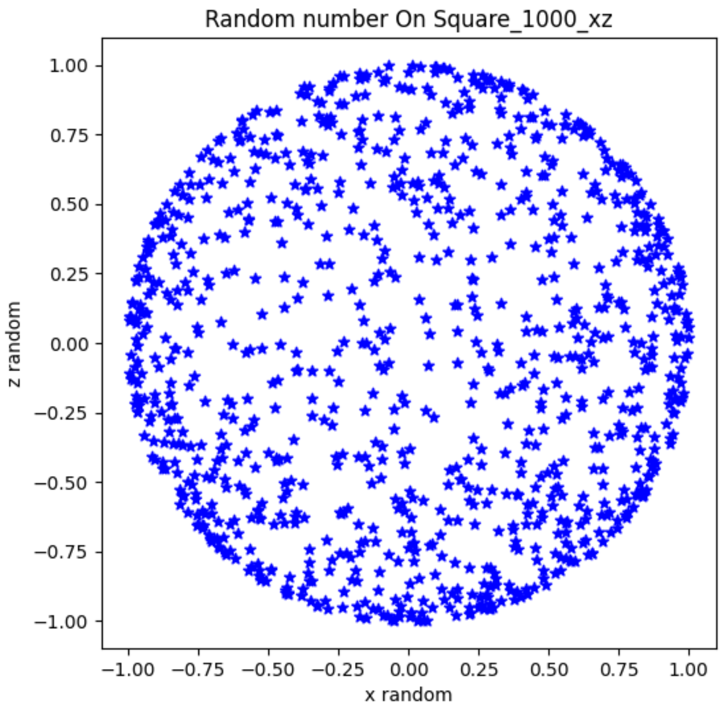
由图可以看出，在不同N取值时，球面上点确实为较为均匀的分布。

2.3 二维投影图

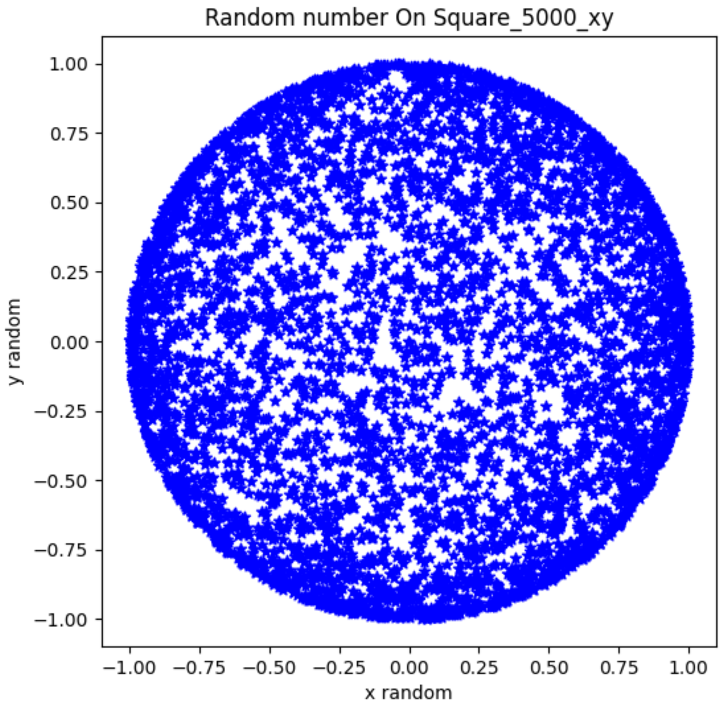
2.3.1 N=1000

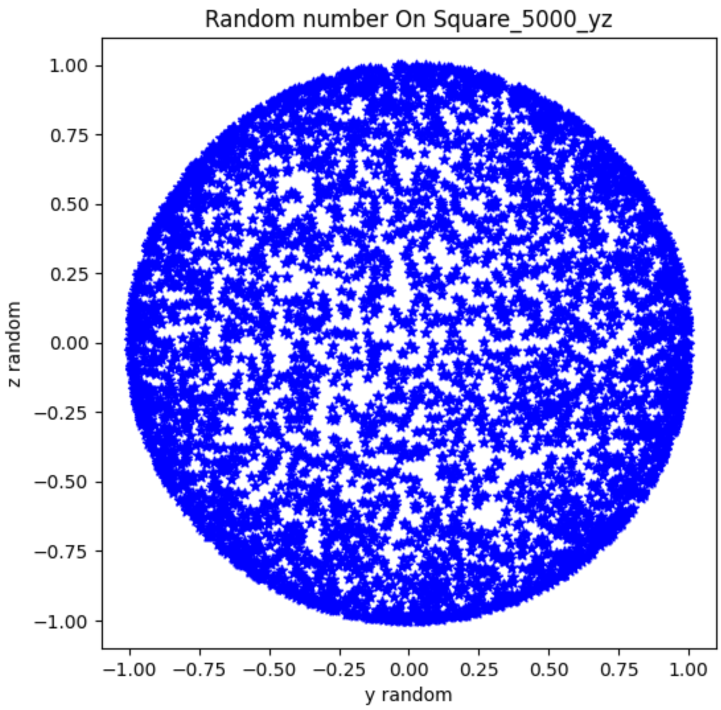


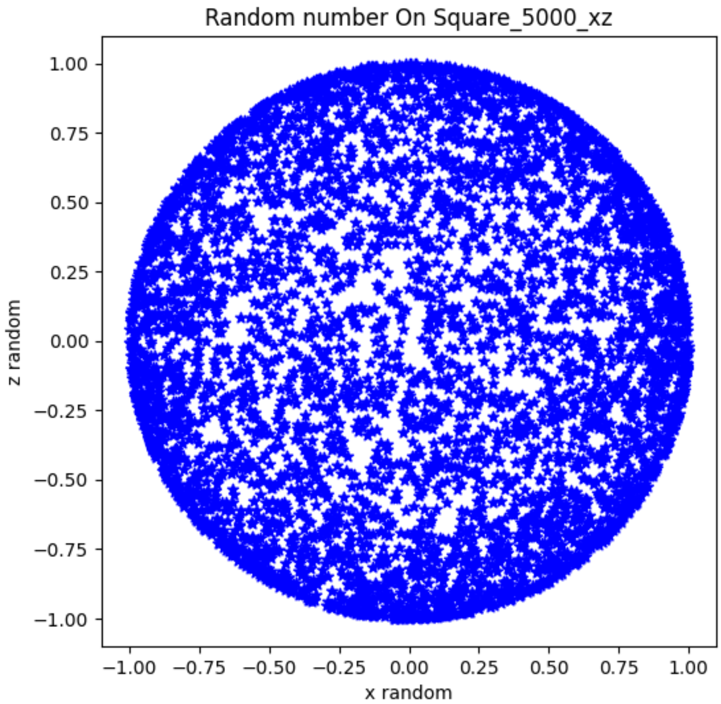

Random number On Square_1000_yz
z random
y random
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00 −0.75 −0.50 −0.25 0.00 0.25 0.50 0.75 1.00


Random number On Square_1000_xz
z random
x random
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00 −0.75 −0.50 −0.25 0.00 0.25 0.50 0.75 1.00

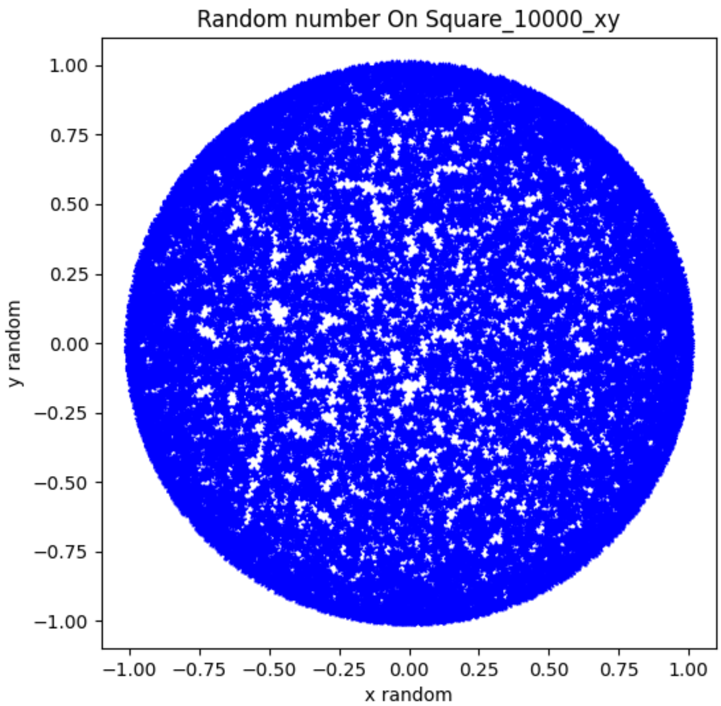
2.3.2 N=5000

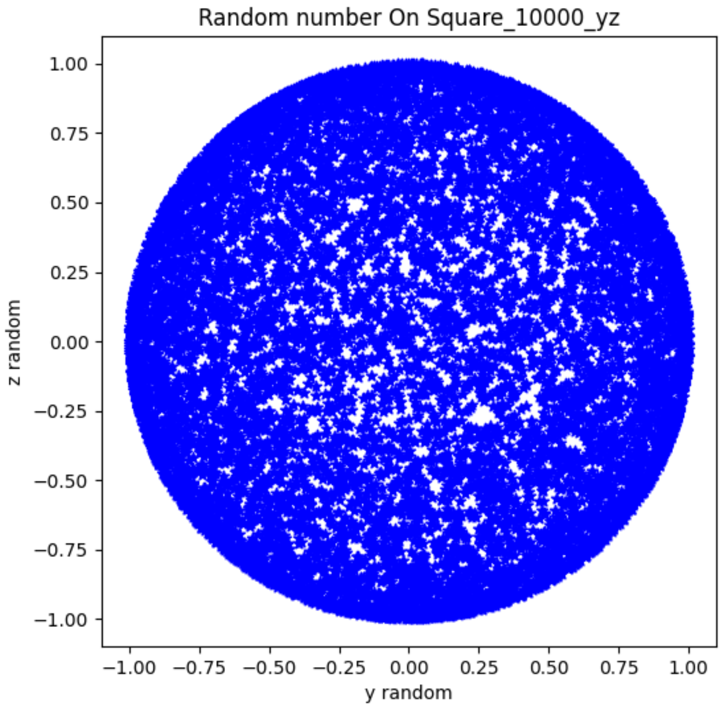

Random number On Square_5000_xy
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
y random
−1.00 −0.75 −0.50 −0.25 0.00 0.25 0.50 0.75 1.00
x random

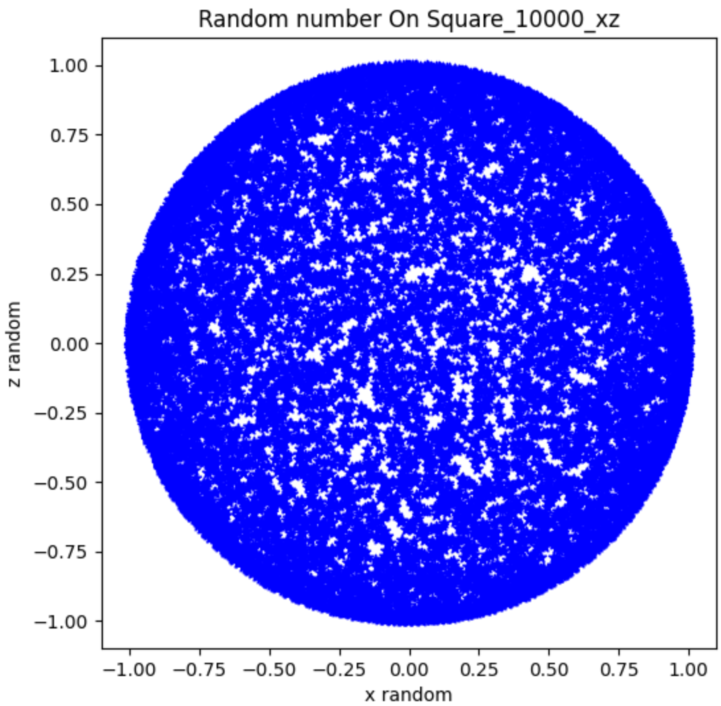

Random number On Square_5000_yz
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
z random
−1.00 −0.75 −0.50 −0.25 0.00 0.25 0.50 0.75 1.00
y random


Random number On Square_5000_xz
z random
x random
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

2.3.3 N=10000


Random number On Square_10000_xy
y random
x random
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00





2.3.4 与理论概率密度函数对比后结论

球面上均匀分布投影到(x, y)下表示，有概率密度函数：

$$f(x,y)=\frac{1}{2\pi\sqrt{1-x^2-y^2}}$$

可以明显看出，随着距离圆心越远，概率密度函数越大，在散点图中点数越多；而实际上，在我们的实验结果中，确实反映如此，可以看到越靠近边缘，分布的密度越大。

从而从另一个唯象角度验证了Marsaglia抽样方法确为球面上均匀分布的随机抽样。

3 总结与收获

在此次实验中，通过之前的积累代码与新的思路，越来越对抽样以及随机数模拟产生感觉，逐渐上手了这样一套方法。

总的来说，第一次验证一种舍选法以及投影图画法，收获颇多。