

计算物理——Homework3

曾郅琛 PB20071431

摘要：利用C++语言以及Python中mpl_toolkits.mplot3d解决以下问题：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。本实验利用球面上的均匀分布，即单位立体角均匀分布，对 (ρ, θ, φ) 进行抽样，最后转化为 (x, y, z) 坐标表示，并绘制图像。

1 算法及实现

1.1 (ρ, θ, φ) 的抽样以及球面直角坐标表示

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：

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

所以在球面上均匀分布点的概率分布函数：

$$F(\theta, \phi) = \frac{1 - \cos(\theta)}{4\pi} \phi = \frac{1 - \cos(\theta)}{2} * \frac{\phi}{2\pi}$$

根据直接抽样方法，将左右两式分别取为抽样：

$$\xi_1 = \frac{1 - \cos(\theta)}{2}$$
$$\xi_2 = \frac{\phi}{2\pi}$$

所以在球坐标下表示可得：

$$\theta = \cos^{-1}(1 - 2\xi_1)$$
$$\phi = 2\pi\xi_2$$

最后利用直角坐标系和球坐标系转化公式，即可得到球面均匀坐标点：

$$x = \sin(\theta)\cos(\phi)$$
$$y = \sin(\theta)\sin(\phi)$$
$$z = \cos(\theta)$$

1.2 随机数生成与抽样

在实验一的基础上，利用实验一的16807随机数生成器，生成`Save_Random(Seed(),N,"Ran_1.txt")`随机数作为第一次抽样，之后利用`Cos(int N, const char *url)`读取随机数并计算得到 θ ；同样的，对 φ 的生成也是如此，`Save_Random(Seed(),N,"Ran_2.txt")`生成随机数作为 φ 的抽样，最后将计算得到的 φ 保存在文件内，下为核心代码：

```
f.open(url,ios::out); //将\xi_1抽样来表示cos(theta)
while (!fp.eof()){
    getline(fp,line);
```

```

        istream sin(line);
        string Waypoints;
        getline(sin,Waypoints);
        stringstream sx; //transform string to double
        sx << Waypoints;
        sx >> k;
        cos = 1-2*k; //cos
        theta = acos(cos); //反三角函数
        f<<theta<<endl;
    }

f.open(url,ios::out); //将\Xi_2抽样表示phi=2*PI*y
while (!fp.eof()){
    getline(fp,line);
    istream sin(line);
    string Waypoints;
    getline(sin,Waypoints);
    stringstream sx; //transform string to double
    sx << Waypoints;
    sx >> k;
    phi=2*PI*k; //phi计算
    f<<phi<<endl;
}

```

1.3 计算直角坐标系坐标值

```

void Generate_pot(char const* p){ //利用生成的theta和phi, 计算坐标值
(x,y,z)
    double x,y,z,theta,phi;
    FILE *fp1 = fopen("theta.txt","r");
    FILE *fp2 = fopen("phi.txt","r");
    FILE *f = fopen(p,"w"); //按行存入坐标值(x,y,z)

    while( (fgetc(fp1)) != EOF&&(fgetc(fp2)) != EOF){
        fscanf(fp1,"%lf",&theta);
        fscanf(fp2,"%lf",&phi);
        x= sin(theta)* cos(phi); //球坐标系下坐标(x,y,z)
        y= sin(theta)* sin(phi);
        z= cos(theta);
        fprintf(f,"%lf,%lf,%lf\n",x,y,z);
    }
    fclose(fp1);
    fclose(fp2);
    fclose(f);
}

```

按照1.1中推导的公式，将 θ 和 φ 代入计算得到直角坐标系坐标，并存在一个文件R_Generate.txt内。

2 实验结果分析及讨论

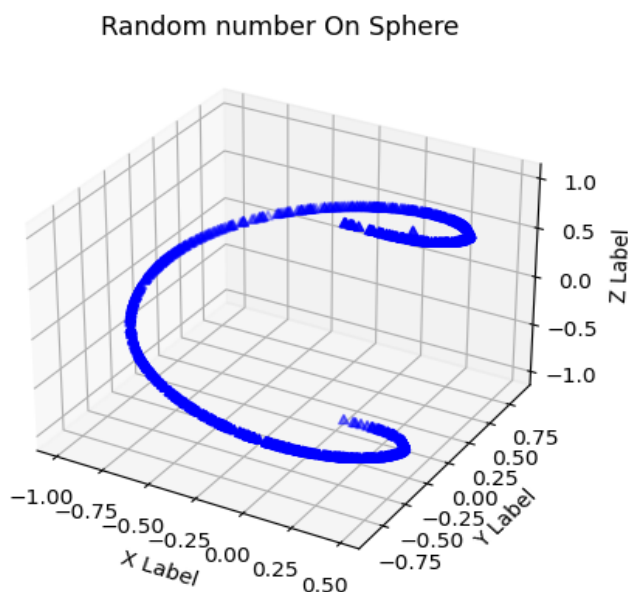
2.1 利用 `mplot3d` 画图

```
input_txt = ".\\cmake-build-debug\\R_Generate.txt"
x = []
y = []
z = []
f = open(input_txt)
for line in f:
    line = line.strip('\n')
    line = line.split(',')
    # print(line)
    x.append(float(line[0]))          # 读取x
    y.append(float(line[1]))          # 读取y
    z.append(float(line[2]))          # 读取z
ax = plt.figure().add_subplot(111, projection = '3d')
ax.set_xlabel('X Label')
ax.set_ylabel('Y Label')
ax.set_zlabel('Z Label')

ax.scatter(x, y, z, c = 'b', s=1, marker = '^')

plt.title("Random number On Sphere_10000")      # 数字与cpp文件对应
plt.savefig("Random number On Sphere_10000.png") # 数字与cpp文件对应
plt.show()
```

在python中调用包绘图，在第一次尝试时出现如下情况：



起初一直以为是代码写错了，原本为球形形状，出现了螺旋型。正想着去debug看看，然后偶然打开了生成的随机数文件，忽然发现生成的"Ran_1.txt"和"Ran_2.txt"竟然完全一样！！！！

Ran_1.txt - 记事本			Ran_2.txt - 记事本		
文件	编辑	查看	文件	编辑	查看
0.296164			0.296164		
0.624812			0.624812		
0.212103			0.212103		
0.814738			0.814738		
0.295973			0.295973		
0.424242			0.424242		
0.228255			0.228255		
0.285042			0.285042		
0.700474			0.700474		
0.860804			0.860804		
0.525884			0.525884		
0.530375			0.530375		
0.0111045			0.0111045		
0.633526			0.633526		
0.670168			0.670168		
0.506292			0.506292		
0.243996			0.243996		
0.848129			0.848129		
0.509284			0.509284		
0.531025			0.531025		
行 11, 列 9			行 1, 列 1		

于是就发现了问题所在，由于16807随机数生成器的种子值取决于系统时间，在生成随机数过程中两步是连续进行，所以seed值并没有区别，所有数据完全一样也不稀奇了。

2.2 改进

为了避免时间一样，那采取的手段必然是让系统sleep一小会zzz：

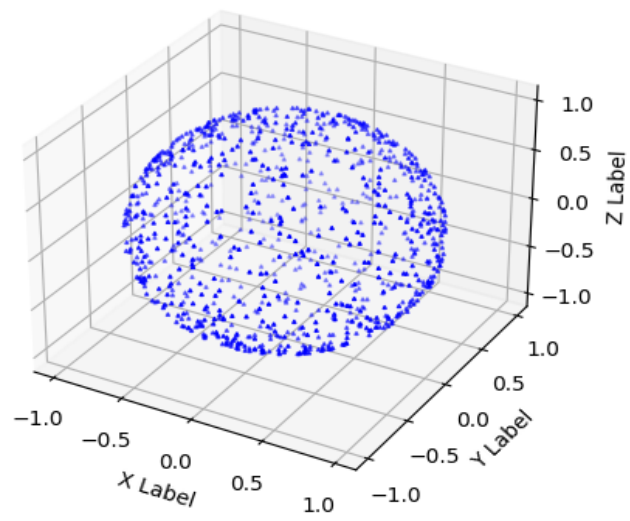
```

s.Cos(N,"theta.txt");           //生成theta.txt
Sleep(5000);                    //sleep一段时间非常非常非常重要！！具体在报告解释
s.Phi(N,"phi.txt");             //生成phi.txt
Generate_pot("R_Generate.txt"); //利用theta, phi生成坐标值

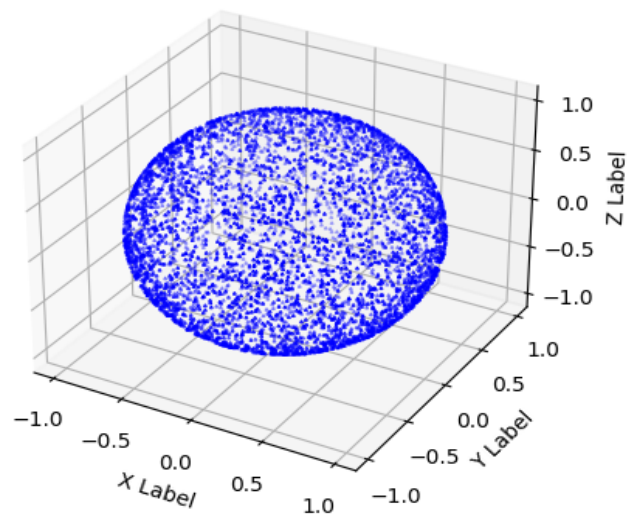
```

再次利用python画图，选取不同的N值：

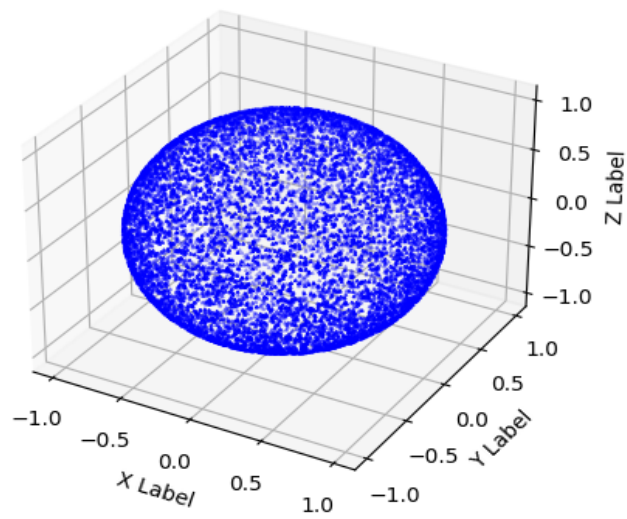
Random number On Sphere_1000



Random number On Sphere_5000



Random number On Sphere_10000



最后可以得到非常均匀的球面点。

3 总结与收获

在此次实验中，整个过程并非完全顺畅，中间遇到与预期结果不符合的情况。反过来仔细思考，因为生成随机数一样， θ 和 φ 应该存在某种关系，而螺线型方程正好也是此种关系，所以必须在两次生成随机数中间插入 `Sleep()` 休眠语句。

总的来说，收获较多，第一次尝试三维画图以及在错误结果中思考bug出处，这些都会给自己以后带来很好的宝贵经验。