

# 计算物理第八次作业

李柏轩 20300200004

January 1, 2023

## 1 题目 1: Volume of Hypersphere

### 1.1 题目描述

The interior of a  $d$ -dimensional hypersphere of unit radius is defined by the condition  $x_1^2 + x_2^2 + \cdots + x_d^2 \leq 1$ .

1. Write a program that finds the volume of a hypersphere using a Monte Carlo method. Test your program for  $d = 2$  and  $d = 3$  and then calculate the volume for  $d = 4$  and  $d = 5$ , compare your results with the exact results.

### 1.2 程序描述

本程序使用 Monte Carlo 方法求解高维单位超球体的体积。利用随机数在  $\{(x_1, x_2, \cdots, x_d) | x_i \in [-1, 1]\}$  范围内投点, 若  $x_1^2 + x_2^2 + \cdots + x_d^2 \leq 1$  则点落在超球体内部, 计数加 1。由于落在超球体内部的概率为  $p = \frac{n}{N_{\text{tot}}} = \frac{V_{\text{sphere}}}{V_{\text{tot}}} = \frac{V_{\text{sphere}}}{2^d}$ , 则  $V_{\text{sphere}} = \frac{n \cdot 2^d}{N_{\text{tot}}}$ 。单位超球体体积公式为

$$V_n = \frac{\pi^{\frac{n}{2}}}{\Gamma\left(\frac{n}{2} + 1\right)} \quad (1)$$

本程序源文件为 HyperSphere.py, 运行依赖 Python 第三方库 Numpy、Matplotlib 和 Math。在终端进入当前目录, 使用命令 `python -u HyperSphere.py` 运行本程序。运行后输出 Fig.1 和 Table.1 所示超球体理论值、计算值以及计算误差, 以及 Fig.2 所示 Monte Carlo 方法收敛过程。

### 1.3 伪代码

Monte Carlo 方法求超球体体积的伪代码如 Alg.1 所示。

### 1.4 输入输出示例

投点个数为  $1 \times 10^6$ , 程序输出 Fig.1 和 Table.1 所示超球体理论值、计算值以及计算误差, 以及 Fig.2 所示 Monte Carlo 方法收敛过程。由 Fig.2 可知 Monte Carlo 作为积分方法时收敛情况 (精度) 与维数无关而只与投点个数有关。

---

**Algorithm 1** Monte Carlo Method

---

**Input:** Dimension of the required hypersphere  $d$  and the number of points  $N$

**Output:** The volume of the  $d$ -dimension hypersphere.

```
1:  $n \leftarrow 0$ 
2: for  $i \leftarrow 1, 2, \dots, N$  do
3:   Randomly generate  $\vec{x} \leftarrow (x_1, x_2, \dots, x_d)$ 
4:   if  $|\vec{x}| \leq 1$  then
5:      $n \leftarrow n + 1$ 
6:   end if
7: end for
8: return  $\frac{n \cdot 2^d}{N}$ 
```

---

```
2D:Theoretical: 3.141592653589793 ,Numerical: 3.145166666666667 , Relative Error: 0.0011376436957190698
3D:Theoretical: 4.1887902047863905 ,Numerical: 4.2035 , Relative Error: 0.0035117049301731794
4D:Theoretical: 4.934802200544679 ,Numerical: 4.976 , Relative Error: 0.00834841960854556
5D:Theoretical: 5.263789013914325 ,Numerical: 5.237 , Relative Error: 0.005089302372019575
```

Figure 1: 超球体体积理论值、计算值以及计算误差

Table 1: 超球体体积理论值、计算值以及计算误差

维数	理论值	计算值	计算误差 (%)
2	3.14159	3.14538	0.114
3	4.18879	4.2035	0.351
4	4.93480	4.976	0.835
5	5.26379	5.237	0.509

## 2 题目 2: 求解 3-D fcc 晶格 Heisenberg 模型的相变温度

### 2.1 题目描述

Write a MC code for a 3D Face-Centered Cubic lattice using the Heisenberg spin model (adopt periodic boundary condition). Estimate the ferromagnetic Curie temperature.

$$H = -J \sum_{\langle i,j \rangle} \vec{S}_i \cdot \vec{S}_j \quad (2)$$

where  $J = 1, |\vec{S}_i| = 1$ .

### 2.2 程序描述

本程序使用 Metropolis 算法求解 3-D fcc 晶格 Heisenberg 模型的相变温度。Metropolis 算法是一种基于 Markov Chain Monte Carlo (MCMC) 的单自旋翻转算法，首先在给定温度下初始化晶格，再进行 Markov Chain 演化：每次随机选择一个格点改变其上的自旋，再根据能量变化求出接受该改变的概率。

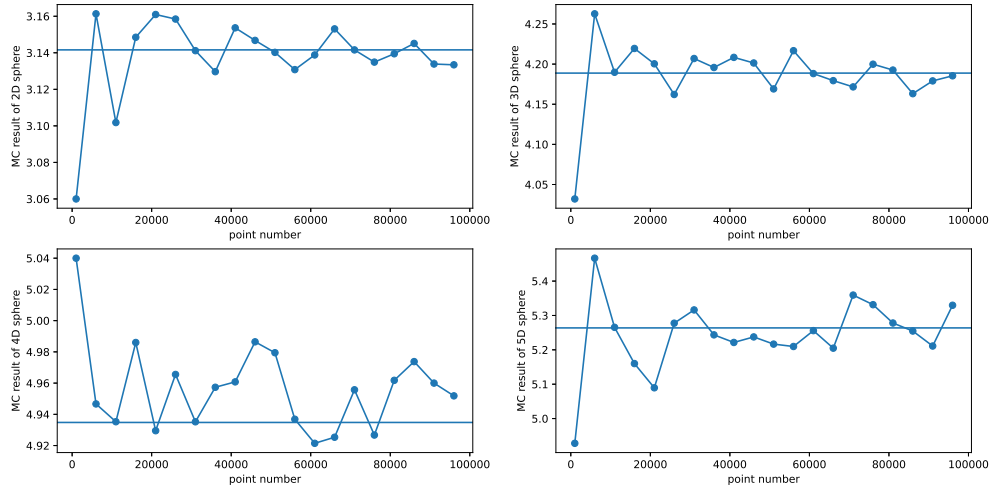


Figure 2: Monte Carlo 方法收敛过程（其中水平直线为理论值）

率，并以该概率决定是否接受该改变；为了防止相邻两次采样之间的相关性太强，我们每进行  $n$  次 Markov Chain 演化作为一次采样，共进行  $N$  次采样，对这  $N$  次采样的结果求平均值得到想要的热力学量。我们需要抛弃前  $K$  次采样，因为那时 Markov Chain 尚未达到收敛。

本程序难点之一是以何种方式存储 fcc 晶格自旋信息并寻找每个格点的最近邻格点。在周期性边条下要使得模拟范围内的每个格点都有 12 个配位数，则晶格应如 Fig.3所示（其中空心圆形表示周期性边条）。

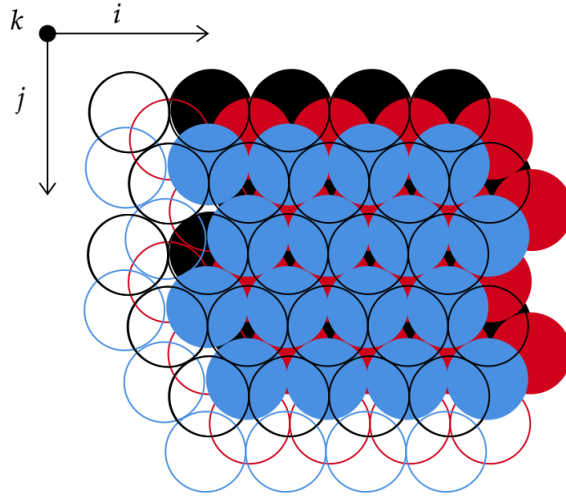


Figure 3: fcc 晶格示意图

本程序源文件为 Metropolis\_HeisenbergFCC.py，运行依赖 Python 第三方库 Numpy、math 和 Matplotlib。在终端进入当前目录，使用命令 `python -u Metropolis_HeisenbergFCC.py` 运行本程序。程序输出 Fig.4所示磁化强度  $m$  随温度  $T$  的关系

## 2.3 伪代码

Metropolis 算法的伪代码如 Alg.2所示。

---

**Algorithm 2** Metropolis Algorithm

---

**Input:** The lattice point number in  $x, y, z$  direction  $n_x, n_y, n_z$ , temperature  $T$ , number of evolution in one bin  $n$ , total sampling number  $N$ , and abandoned sampling number  $K$ .

**Output:** The magnetization intensity  $M$

```
1: Initialize the spin lattice array
2: for  $i \leftarrow -K, -K + 1, \dots, 0, 1, \dots, N$  do
3:   for  $j \leftarrow 1, 2, \dots, n$  do
4:     Randomly choose one lattice point  $\vec{x} \leftarrow (i, j, k)$ 
5:     Randomly generate one magnetic spin  $\vec{m}$ 
6:      $\Delta E \leftarrow -J \sum_{\text{n.n. of } \vec{x}} \vec{m}(\vec{y}) \cdot [\vec{m} - \vec{m}(\vec{x})]$ 
7:      $p \leftarrow \min(1, e^{-\Delta E/T})$ 
8:     Randomly generate a number  $u$  in  $[0, 1)$ 
9:     if  $u < p$  then
10:       $\vec{m}(\vec{x}) \leftarrow \vec{m}$ 
11:    end if
12:  end for
13:  if  $i > 0$  then
14:     $m[i] \leftarrow \sum_{\vec{x}} \vec{m}(\vec{x})$ 
15:  end if
16: end for
17: return  $\langle \vec{m} \rangle$ 
```

---

## 2.4 输入输出示例

设置  $(n_x, n_y, n_z) = (4, 4, 3)$ ,  $n = 48$ ,  $N = 6000$ ,  $K = 1000$  并取  $J = 1, k_B = 1$ , 输出磁化强度  $m$  随温度  $T$  的关系如 Fig.4所示。由图估计 Curie 温度约为 3.0K。

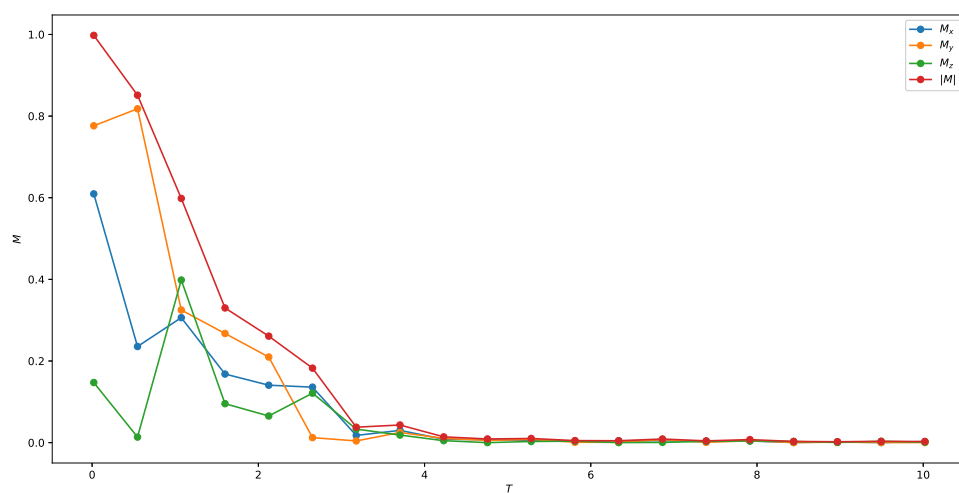


Figure 4: 3-D fcc 晶格 Heisenberg 模型的 M-T 相图