

# 第九次作业

尹朝阳 物理学系

## 1 题目 1

### 1.1 题目描述

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

本题目要求用 Monte-Carlo method 进行 Poisson 求解单位超球体的体积。

求解思路为，采用每一个维度上的在  $[-1, 1]$  区间上的独立的均匀分布 (uniform distribution)，生成大量的随机坐标点  $X_i = (x_{i0}, x_{i1}, \dots, x_{id}), i = 1, 2, \dots, n$ ，其中  $n$  为点的总数， $d$  为超球体的几何维度。对每一个随机点判断是否有

$$x_{i1}^2 + x_{i2}^2 + \dots + x_{id}^2 \leq 1$$

若上式成立，则取该点 (即  $\text{accept} \leftarrow \text{accept} + 1$ )，等到将所有数据点遍历一遍后，计算  $p = \text{accept}/n$  即为落在单位超球体内的概率，最后该值乘上所取的立方空间的体积  $2^d$ ，则可认为是单位超球体的体积。即体积的估计公式为

$$V = \text{accept}/n * 2^d$$

对于单位超球体，显然  $n = 2$  时即为单位圆，其体积 (面积) 为  $\pi$ ，而  $n = 3$  时为单位球，其体积为  $\frac{4\pi}{3}$ 。对于更高维的情况，有以下理论值公式 [1]

$$V = \frac{\pi^{n/2}}{\Gamma(\frac{n}{2} + 1)}$$

容易计算得， $n = 4$  时体积为  $\frac{\pi^2}{2}$ ， $n = 5$  时体积为  $\frac{8\pi^2}{15}$ 。

本题目的源程序为 `VolumeofHypersphere/calculate_volume_of_hypersphere_MC_method.py`，导入的第三方库有 `numpy`。

### 1.3 伪代码

---

**Algorithm 1** calculate the volume of a hypersphere using a Monte Carlo method

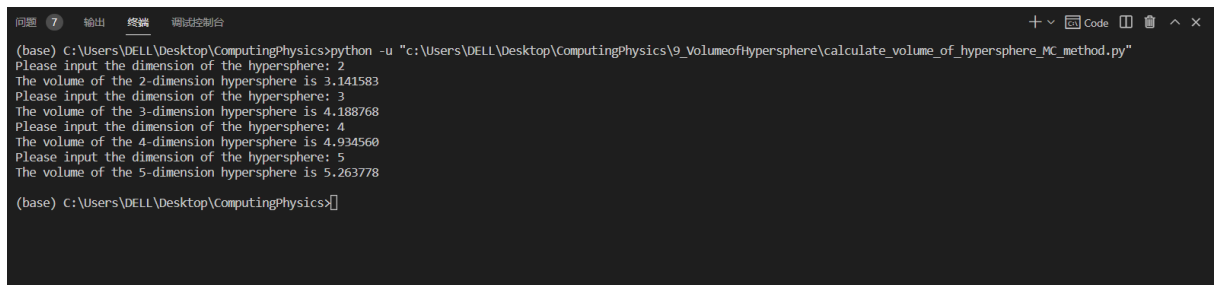
---

**Output :** the volume of the  $d$ -dimension hypersphere  $V$ 

```
1: set the total number  $n$  to accept or reject and the dimension  $d$ 
2: for  $i \leftarrow 1$  to  $n$  do
3:   for  $j \leftarrow 1$  to  $d$  do
4:      $rand_{ij} \leftarrow$  a random number chosen from a uniform distribution of  $[0, 1]$ 
5:   end for
6: end for
7:  $accept \leftarrow 0$ 
8: for  $k \leftarrow 1$  to  $n$  do
9:   if  $\|rand_i\| \leq 1$  then
10:     $accept \leftarrow accept + 1$ 
11:   end if
12: end for
13:  $V_{cube} \leftarrow 2^d$ 
14:  $V \leftarrow accept/n * V_{cube}$ 
15: return  $V$ 
```

---

## 1.4 测试用例



```
(base) C:\Users\DELL\Desktop\ComputingPhysics>python -u "c:\Users\DELL\Desktop\ComputingPhysics\9_volumeofhypersphere\calculate_volume_of_hypersphere_MC_method.py"
Please input the dimension of the hypersphere: 2
The volume of the 2-dimension hypersphere is 3.141583
Please input the dimension of the hypersphere: 3
The volume of the 3-dimension hypersphere is 4.188768
Please input the dimension of the hypersphere: 4
The volume of the 4-dimension hypersphere is 4.934560
Please input the dimension of the hypersphere: 5
The volume of the 5-dimension hypersphere is 5.263778

(base) C:\Users\DELL\Desktop\computingPhysics>
```

图 1: the volume of the  $d$ -dimensional hypersphere

与理论值比较可以发现，误差均较小。

dimension	2	3	4	5
理论值	3.141593	4.188790	4.934802	5.263789
模拟值	3.141583	4.188768	4.934560	5.263778

## 2 题目 2

### 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.

$$H = -J \sum_{\langle i,j \rangle_{NN}} \vec{S}_i \cdot \vec{S}_j, \quad J = 1, \quad |\vec{S}_i| = 1$$

### 2.2 程序描述

本题目要求用 Monte Carlo 方法模拟三维面心立方晶格的海森堡自旋模型，并估计铁磁性材料的居里温度。

居里温度可以从磁化强度随温度的变化曲线  $M(T)$  中读出，对于海森堡自旋模型，已知其  $M(T)$  曲线的大致变化趋势如下图 2 所示。

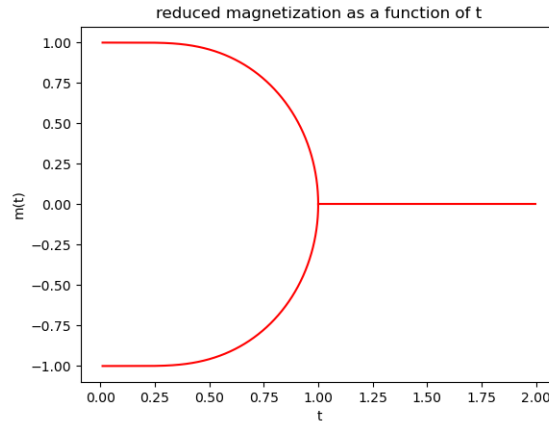


图 2:  $M(T)$  (注: 图源自作业 2)

则铁磁性材料的居里点为上图的三条曲线的交汇点，也即平均磁化强度从开始下降为 0 的点，其温度为临界温度，即居里温度。

本题采用上述的方法估计居里温度，即先绘制磁化强度随温度的变化曲线，再从图上估计出居里温度。

对于每个温度  $T$ ，求解该温度下的平均磁化强度的方法如下。

首先，生成面心立方晶格。将三维坐标系的原点建在最边缘的立方体的最外部的顶点处， $x$ 、 $y$ 、 $z$  轴分别与小立方体的三条棱互相平行。则对于最左下角的小立方体，其 8 个顶点上分别有一个自旋，坐标分别为  $(0, 0, 0)$ 、 $(0, 0, 2)$ 、 $(0, 2, 0)$ 、 $(0, 2, 2)$ 、 $(2, 0, 0)$ 、 $(2, 0, 2)$ 、 $(2, 2, 0)$ 、 $(2, 2, 2)$ ，此外每个面的中心分别有一个自旋，其坐标分别为  $(1, 1, 0)$ 、 $(1, 1, 2)$ 、 $(0, 1, 1)$ 、 $(2, 1, 1)$ 、 $(1, 0, 1)$ 、 $(1, 2, 1)$ 。同理可以构建其他自旋的坐标。我们发现，在该格点坐标系中，面心立方晶格的自旋只存在于  $x_i + y_i + z_i$  为偶数的格点位置，而可以设置其他位置的自旋为 0，不对能量和磁化强度产生贡献。

然后，对于存在自旋的位置，需要设置其自旋矢量，要求我们生成均匀球面分布的矢量。生成方法如下：

$$x = r \sin \theta \cos \phi$$

$$y = r \sin \theta \sin \phi$$

$$z = r \cos \theta$$

我们生成特定分布的  $\theta$  和  $\phi$ ，然后将其映射到直角坐标系  $(x, y, z)$  上，使得点在球面上均匀分布。该特定分布可以用如下函数生成 [2]：

$$u = \text{uniform distribution of } [0, 1]$$

$$v = \text{uniform distribution of } [0, 1]$$

$$\theta = 2\pi * u$$

$$\phi = \arccos(2 * v - 1)$$

我们取 10000 个点，用上述的映射关系随机取得各点的坐标（具体见 code2.1 伪代码），绘制三维分布图如下图 3 所示。可以看到成功生成了三维球面上的均匀分布。

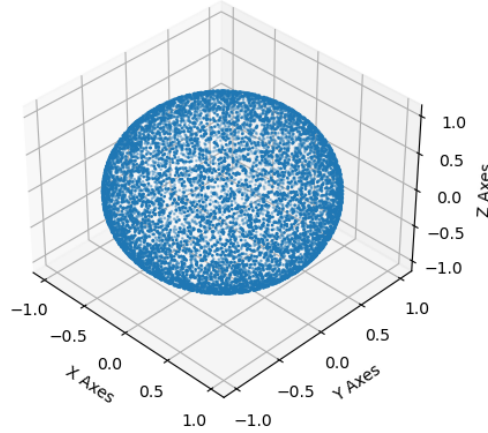


图 3: uniform distribution on a 3D sphere

在得到初始 3 维 FCC 海森堡自旋模型后，下面计算系统达到稳态时的平均磁化强度。

随机选取一个自旋  $\vec{S}_i$ ，设置随机转过一个角度（即用 code2.1 伪代码中的函数随机生成一个新的三维单位矢量赋值给这个自旋）变为  $\vec{S}_i'$ ，计算能量变化。由于我们只需要考虑最近邻的自旋对该自旋的影响，面心立方中最近邻自旋有 12 个，则能量变化  $\Delta E$  满足：

$$\Delta E = (-J * \vec{S}_i' \cdot \sum_{12} \vec{S}_j) - (-J * \vec{S}_i \cdot \sum_{12} \vec{S}_j)$$

其中对于坐标为  $(i, j, k)$  的自旋，其最近邻自旋的坐标为  $(i, j-1, k-1)$ 、 $(i, j+1, k-1)$ 、 $(i, j-1, k+1)$ 、 $(i, j+1, k+1)$ 、 $(i-1, j, k-1)$ 、 $(i-1, j, k+1)$ 、 $(i+1, j, k-1)$ 、 $(i+1, j, k+1)$ 、

$(i-1, j-1, k)$ 、 $(i-1, j+1, k)$ 、 $(i+1, j-1, k)$ 、 $(i+1, j+1, k)$ 。考虑周期性边界条件, 当  $i = 1$  时, 取  $i-1 = i_{max}$ ; 当  $i = i_{max}$  时, 取  $i+1 = 1$ 。对于  $j$ 、 $k$  同理。

然后用 reject or accept 方法评估是否要接受这个改变作为一个新的自旋。计算概率

$$p = e^{-\Delta E/(k_B T)}$$

再从  $[0, 1]$  的均匀分布中随机选取一个数  $PRN$ , 如果  $p \geq PRN$ , 就接受这个改变, 否则若  $p < PRN$  就拒绝这次随机变化。

重复上述过程, 直至系统基本达到平衡状态, 这里认为状态改变 10000 次之后基本达到稳定。取最后的 8000-10000 次, 每 20 次记录下系统的平均磁化矢量, 最后取平均值并计算平均后的磁化矢量的模, 即为该温度下的磁化强度。其中磁化强度由  $M = \|(M_x, M_y, M_z)\|$  给出。

对于不同温度分别计算磁化强度, 最终得到平均磁化强度随温度的变化关系, 并得到临界居里温度。

本题目的源程序为 CurieTemperature-MCmethod/FccFMCurieTemperature\_HeisenbergSpinModel\_MCmethod.py, 其他两个程序 plot\_magnetization-timesteps\_consider GivenTemperature.py 和 uniformDistribution\_Ona3DSphere.py 分别绘制在给定温度下磁化强度随时间的变化关系, 以及绘制三维球面上的均匀分布。导入的第三方库有 numpy 和 matplotlib。

注: 标准的海森堡模型要求无限大, 这里我们选取  $15 \times 15 \times 15$  个小晶格拼成的三维自旋系统, 并设定周期性边界条件, 来进行较为粗略的模拟。

## 2.3 伪代码

由于本题目代码较长, 各代码块之间有较强的非耦合性, 这里将全部代码拆分为多个伪代码块给出。

code2.1、2.2、2.3、2.4 分别实现“产生三维球面上均匀分布的随机矢量”、“建立三维面心立方晶格的海森堡自旋模型”、“改变系统状态并用 reject or accept 方法判断该改变是否有效”以及“计算磁化强度并绘制磁化强度随温度的变化关系”。

由于伪代码较多, 将放在最后“参考文献”之后的部分。

## 2.4 测试用例

首先, 我们取  $J = 1$ , 由于计算概率  $p = e^{-\Delta E/(k_B T)}$  时, 若取玻尔兹曼常数  $k_B = 1.380649 \times 10^{-23} J/K$ , 而  $\Delta E$  在 0.1 到 1 的量级上, 会导致概率  $p \rightarrow 0 (\Delta E > 0)$  或者  $p \rightarrow +\infty (\Delta E < 0)$ , 与  $PRN$  比较意义不大, 故这里选取  $k_B = 0.01$  (临界温度大约在几百 K) 使得量级基本相同。

首先可以看一下在特定温度 (这里选取为 80K) 下磁化强度随时间的变化关系。设置初始自旋均为  $(1, 0, 0)$ 。

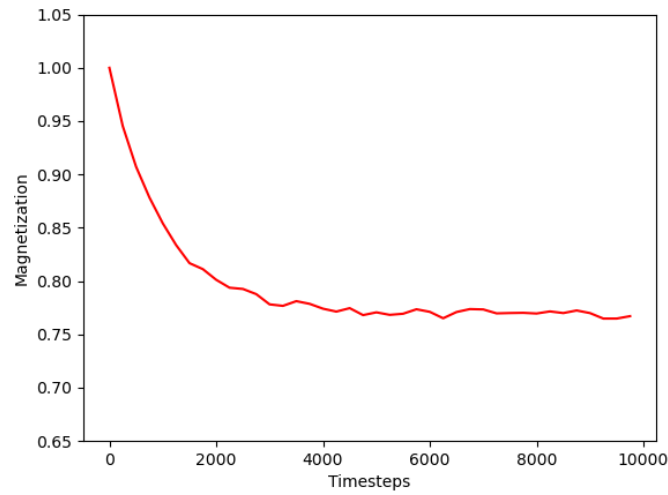


图 4: Magnetization-time graph under a given temperature

可以看到，磁化强度逐渐降低，在 6000 步之后基本不变，此时可以选择 8000-10000 步的磁化强度做平均后作为该温度下的平均磁化强度。

对 0-200K 的温度，每 5K 进行平均磁化强度的计算，绘制平均磁化强度随温度的变化曲线如图 5 所示。

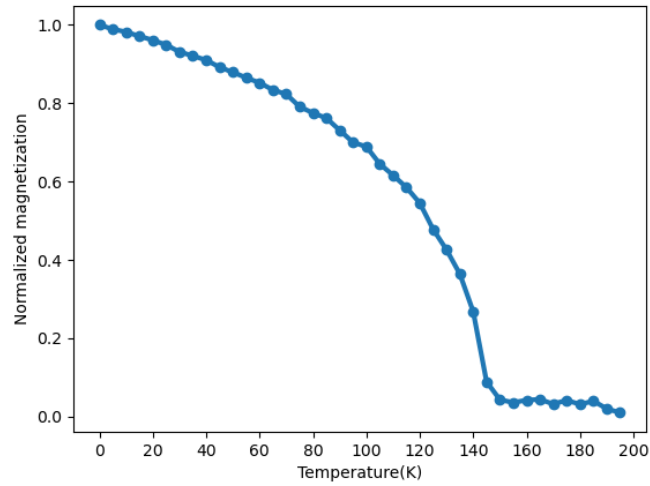


图 5: Magnetization-Temperature graph

可以看到，在大约 145K 左右，平均磁化强度降为接近 0，可认为该点为临界点，则临界居里温度为 145K。

下面以 fcc-Ni 为例，估计其居里温度。

对于 fcc-Ni,  $J$  大约为  $2.93 \times 10^{-21} J/link$ [3]。

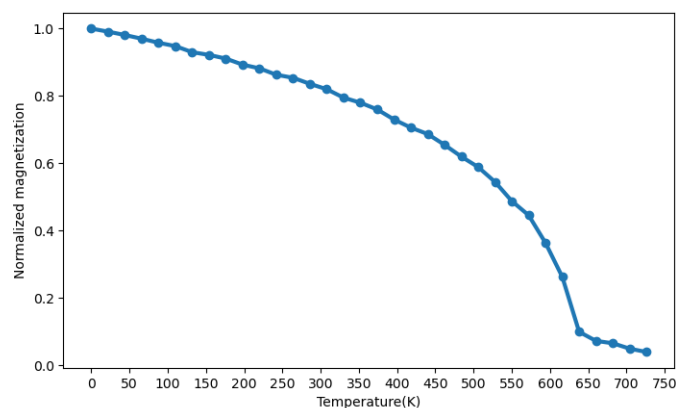


图 6: Magnetization of fcc-Ni – Temperature graph

从上图可以估计出，居里温度大约为 640K，接近实验上测量得的居里温度（大约为 620K）。认为产生误差的原因有系统由  $15 \times 15 \times 15$  个晶格组成，而标准海森堡模型要求系统趋于无穷大；此外只考虑最近邻的自旋的影响也会产生误差。

## 参考文献

- [1] Volume of Hypersphere[OL]. <https://www.chegg.com/learn/calculus/calculus/volume-of-hypersphere>.
- [2] Cory Simon. Generating uniformly distributed numbers on a sphere[OL]. <http://corysimon.github.io/articles/uniformdistn-on-sphere/>. February 27, 2015.
- [3] Jin, X., Kudrnovský, J., Wang, D. and Bruno, Patrick. (2008). Curie temperatures of fcc and bcc nickel and permalloy: Supercell and Green's function methods. Physical Review B. 77. 10.1103/PhysRevB.77.054431.

## 附录

---

**Algorithm 2.1** generate a randomly oriented 3D vector

---

**Output :**  $\vec{S}$  the vector

```
1:  $u \leftarrow$  a random number chosen from an uniform distribution of  $[0, 1]$ 
2:  $v \leftarrow$  a random number chosen from an uniform distribution of  $[0, 1]$ 
3:  $\theta \leftarrow 2\pi * u$ 
4:  $\phi \leftarrow \arccos(2 * v - 1)$ 
5:  $x \leftarrow \sin \theta * \sin \phi$ 
6:  $y \leftarrow \sin \theta * \cos \phi$ 
7:  $z \leftarrow \cos \phi$ 
8: return  $\vec{S} \leftarrow (x, y, z)$ 
```

---

---

**Algorithm 2.2** generate a fcc lattice using the heisenberg spin model

---

**Output :** the fcc heisenberg spin model  $spin$

```
1: for  $i \leftarrow 1$  to  $i_{max}$  do
2:   for  $j \leftarrow 1$  to  $j_{max}$  do
3:     for  $k \leftarrow 1$  to  $k_{max}$  do
4:       if  $(i + j + k) \% 2 = 0$  then
5:          $spin_{ijk} \leftarrow \text{Randomly\_Orientated\_3D\_Vector}()$  ▷ code2.1
6:       else
7:          $spin_{ijk} \leftarrow (0, 0, 0)$ 
8:       end if
9:     end for
10:   end for
11: end for
12: return  $spin$ 
```

---



---

**Algorithm 2.3** randomly choose and change a spin and decide whether to accept the change

---

```

1: Expand the  $spin$  to a  $((i_{max} + 2) \times (k_{max} + 2) \times (k_{max} + 2))$  matrix
2:  $spin[0, :, :] \leftarrow spin[i_{max}, :, :]$  ▷ line 2 – 7: adopt the periodic boundary condition
3:  $spin[:, 0, :] \leftarrow spin[:, j_{max}, :]$ 
4:  $spin[:, :, 0] \leftarrow spin[:, :, k_{max}]$ 
5:  $spin[i_{max} + 1, :, :] \leftarrow spin[1, :, :]$ 
6:  $spin[:, j_{max} + 1, :] \leftarrow spin[:, 1, :]$ 
7:  $spin[:, :, k_{max} + 1] \leftarrow spin[:, :, 1]$ 
8:  $spin' \leftarrow spin$ 
9:  $i \leftarrow 0, \quad j \leftarrow 0, \quad k \leftarrow 1$ 
10: while  $(i + j + k) \% 2 \neq 0$  do
11:    $i \leftarrow$  a random integer from  $[1, i_{max}]$ 
12:    $j \leftarrow$  a random integer from  $[1, j_{max}]$ 
13:    $k \leftarrow$  a random integer from  $[1, k_{max}]$ 
14: end while
15:  $spin'_{ijk} \leftarrow \text{Randomly\_Orientated\_3D\_Vector}()$  ▷ code2.1
16:  $E_{old} \leftarrow -J * (spin_{ijk} \cdot spin_{i,j-1,k-1} + spin_{ijk} \cdot spin_{i,j+1,k-1} + spin_{ijk} \cdot spin_{i,j+1,k+1} + spin_{ijk} \cdot$   

 $spin_{i,j-1,k+1} + spin_{ijk} \cdot spin_{i-1,j,k-1} + spin_{ijk} \cdot spin_{i+1,j,k-1} + spin_{ijk} \cdot spin_{i-1,j,k+1} + spin_{ijk} \cdot$   

 $spin_{i+1,j,k+1} + spin_{ijk} \cdot spin_{i-1,j-1,k} + spin_{ijk} \cdot spin_{i+1,j-1,k} + spin_{ijk} \cdot spin_{i-1,j+1,k} + spin_{ijk} \cdot$   

 $spin_{i+1,j+1,k})$ 
17:  $E_{new} \leftarrow -J * (spin'_{ijk} \cdot spin'_{i,j-1,k-1} + spin'_{ijk} \cdot spin'_{i,j+1,k-1} + spin'_{ijk} \cdot spin'_{i,j+1,k+1} + spin'_{ijk} \cdot$   

 $spin'_{i,j-1,k+1} + spin'_{ijk} \cdot spin'_{i-1,j,k-1} + spin'_{ijk} \cdot spin'_{i+1,j,k-1} + spin'_{ijk} \cdot spin'_{i-1,j,k+1} + spin'_{ijk} \cdot$   

 $spin'_{i+1,j,k+1} + spin'_{ijk} \cdot spin'_{i-1,j-1,k} + spin'_{ijk} \cdot spin'_{i+1,j-1,k} + spin'_{ijk} \cdot spin'_{i-1,j+1,k} + spin'_{ijk} \cdot$   

 $spin'_{i+1,j+1,k})$ 
18:  $\Delta E = E_{new} - E_{old}$ 
19:  $p \leftarrow e^{-\Delta E / (k_B * T)}$ 
20:  $PRN \leftarrow$  a random number from the uniform distribution of  $[0, 1]$ 
21: if  $p \geq PRN$  then ▷ accept or reject method
22:    $newSpin \leftarrow$  the unexpanded  $spin'$ 
23: else
24:    $newSpin \leftarrow$  the unexpanded  $spin$ 
25: end if
26: return  $newSpin$ 

```

---

---

**Algorithm 2.4** calculate the magnetization for every temperature and plot  $M(T)$

---

```

1: for  $T \leftarrow T_{min}$  to  $T_{max}$  do
2:    $spin \leftarrow \text{Fcc\_Heisenberg\_Spin\_Model}()$  ▷ code2.2
3:   for  $t \leftarrow start$  to  $stop$  do
4:      $spin \leftarrow \text{Decide\_The\_New\_Spins}(spin, T)$  ▷ code2.3
5:      $S_x \leftarrow 0$ 
6:      $S_y \leftarrow 0$ 
7:      $S_z \leftarrow 0$ 
8:     for  $i \leftarrow 1$  to  $i_{max}$  do
9:       for  $j \leftarrow 1$  to  $j_{max}$  do
10:        for  $k \leftarrow 1$  to  $k_{max}$  do
11:           $S_x \leftarrow S_x + spin_{ijk,1}$ 
12:           $S_y \leftarrow S_y + spin_{ijk,2}$ 
13:           $S_z \leftarrow S_z + spin_{ijk,3}$ 
14:        end for
15:      end for
16:    end for
17:     $M(t) \leftarrow ||(S_x, S_y, S_z)||$ 
18:  end for
19:   $loop \leftarrow 0$ 
20:  for  $t_i \leftarrow (0.8 * (stop - start) + start)$  to  $stop$ ,  $step$  do ▷ 8000 – 10000, 20 as time step
21:     $loop \leftarrow loop + 1$ 
22:     $M\_mean(T) \leftarrow M\_mean(T) + M(t_i)$ 
23:  end for
24:   $M\_mean(T) \leftarrow M\_mean(T)/loop$ 
25: end for
26: Plot  $M\_mean(T)$  with the temperature  $T$  changes

```

---