

# 量子多体第六次作业

高岚 2019102775

本次作业：

1:编写经典Monte Carlo程序模拟二维Ising模型，计算 $m^2$ 的期望值及其标准差，并作图。

2:计算 $m^2$ 、比热、和磁化率（值和误差），并进行data collapse确定相变温度和临界指数。

[本次计算单位制取  $\hbar = m_e = e = 1$ ，常数取  $k_B = 1$ ， $\mu_B = 1$ ]

## 目录

目录	1
1.蒙特卡洛及 Ising 模型简介	2
2.Metropolis算法	2
3.蒙特卡洛方法的代码实现	3
4.蒙特卡洛方法求解2维Ising模型	6
4.1 体系热力学量公式推导	6
4.2 程序设计	7
4.3 误差分析与结果展示	9
4.4 临界指数确定与data collapse	14

## 1.蒙特卡洛及 Ising 模型简介

蒙特卡罗方法是非常强大的一种方法，在各个领域都有重要的应用。核心思想就是利用随机数来代替概率来进行抽样计算平均值。本次作业针对二维 Ising 模型进行计算，采用经典蒙特卡罗方法，对二维 Ising 模型的  $m^2$ 、比热和磁化率（值和误差）进行计算。我们知道 Ising 模型的哈密顿量为

$$H = -J \sum_{ij} \sigma_i^z \sigma_j^z \quad (J=1) \quad (1)$$

考虑体系的热力学性质，我们需要考虑配分函数  $Z = \sum_n e^{-\beta E_n}$ ，这里  $n$  表示每一个 Ising 构型。这样我们可以得到所关心的观测量可以写作

$$\langle O \rangle = \frac{1}{Z} \sum_n e^{-\beta E_n} O_n \quad (2)$$

若将  $p_n = \frac{e^{-\beta E_n}}{Z}$  看作几率，这就是一个概率加权求和的问题，因此可以很好的运用蒙特卡罗方法来求解。

## 2.Metropolis算法

生成构型序列的算法我们用的是 Metropolis 算法。算法中任意翻转构型中的一个自旋  $\sigma(1)$ ，之后每个根据所需分布函数判断为成功翻转而得到的  $\sigma(k+1)$  都被随机的从  $\sigma(1)$  得到。我们只需要存储现有的构型。通常我们将  $N$  次翻转定义为一次蒙特卡洛 Sweep。这个算法建立在细致平衡理论的基础上，这是随机过程的常用理论，记从 A 构型变为 B 构型的概率为  $P(A \rightarrow B)$ ，细致平衡条件要求

$$\frac{P(A \rightarrow B)}{P(B \rightarrow A)} = \frac{W(B)}{W(A)} \quad (3)$$

在满足  $P(A \rightarrow B) > 0$  时对于任意两个构型 A、B 都成立。其中  $W$  即为概率分布。另外，取样应该是各态历经的。

因此更新算法可以表示为

$$P(\sigma \rightarrow \sigma^{-i}) = P_{select}(i)P_{accept}(\sigma^{-i}) \quad (4)$$

其中  $P_{select}(i)$  即为随意选取的一个自旋  $i$ ，在这里应该始终等于  $\frac{1}{N}$ 。

Metropolis 算法中接受概率取为

$$P_{accept}(\sigma^{-i}) = \min[\frac{W_{\sigma^{-i}}}{W_{\sigma}}, 1] \quad (5)$$

### 3.蒙特卡洛方法的代码实现

接下来运用程序实现蒙特卡罗算法。首先我们建立初始的格点  $N=L*L$ ，即  $N$  为总格点数，并赋予初值，我们这里给的初值为每个格点自旋取 1。程序代码为：

```
L = 8
shape = (L, L)
N = L*L
moment = 1
delta=0.05
# External magnetic field
field = np.full(shape, 0)
# Interaction (ferromagnetic if positive, antiferromagnetic if
negative)
interaction = 1
```

这里 field 为外场，我们取的是 0，若取有限值可以计算横场 Ising 模型。然后我们定义接受的概率为  $P = e^{-\beta(E_m - E_n)/T}$ ，因此我们需要定义概率函数和获取能量的函数，代码为：

```
def get_probability(delta_energy, beta):
    return np.exp(-delta_energy*beta)

def get_energy():
    return -np.sum(
        interaction * spins * np.roll(spins, 1, axis=0) +
```

```

interaction * spins * np.roll(spins, -1, axis=0) +
interaction * spins * np.roll(spins, 1, axis=1) +
interaction * spins * np.roll(spins, -1, axis=1)
)/2 - moment * np.sum(field * spins)

```

这个获取能量的函数将构型中所有的自旋间相互作用全部计算了一遍并求和，是按照哈密顿量直接算的，计算量较大，因此我们只会在第一次计算能量时调用他，之后每次更新我们都去计算更新前后的能量差，然后去加上能量差得到新构型的能量。

接下来便可以定义更新的函数，这里我们采用了全局变量，在更新函数中维护自旋构型、当前能量和自旋的和，避免每次更新都需要复制自旋构型进行翻转，同时也避免计算磁矩时总要对全部自旋求和以及计算能量时重新对哈密顿量进行计算，因此采用全局变量可以大大减少程序的计算量。我们随机选取一个自旋进行翻转，然后判断是否接受，进而完成更新，这部分为 Monte Carlo 算法的核心部分。这部分代码为：

```

def update(beta):
    global spins
    global current_energy
    global spins_sum
    i = np.random.randint(spins.shape[0])
    j = np.random.randint(spins.shape[1])
    delta_energy=2*spins[i,j]*(spins[(i+1)%L,j]+spins[(i-1)%L,j]+spins[i,(j+1)%L]+spins[i,(j-1)%L])
    if get_probability(delta_energy, beta) > np.random.random():
        spins[i,j] *= -1
        current_energy += delta_energy
        spins_sum += spins[i,j]*2

```

这里我们可以先看到预热一百万步后构型为

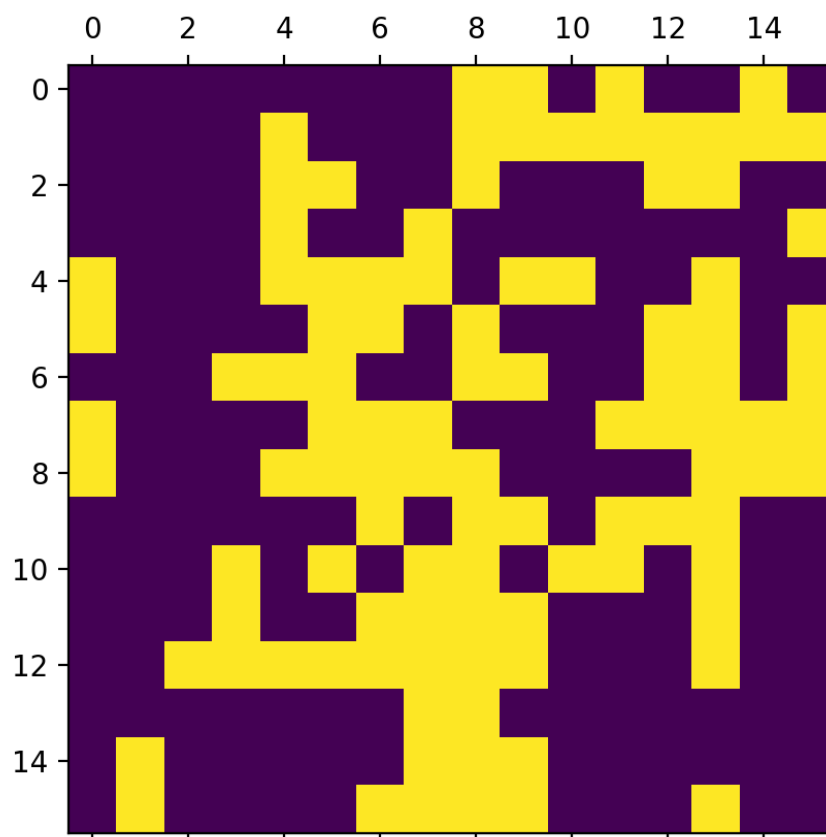


图 1. 预热一百万步

最后我们再定义两个循环函数，一个蒙特卡洛sweep，另一个进行一个 bin 内的所有sweep，代码如下：

```
def isweep(N,beta):
    for i in range(N):
        update(beta)

def
bin_sweep(N_per_bin,beta,mag_sum,mag2_sum,energy1_sum,energy2_sum):
    for i in range(N_per_bin):
        isweep(N,beta)
        mag_sum += abs(spins_sum)/N
        mag2_sum += (spins_sum/N)**2
        energy1_sum += current_energy
        energy2_sum += current_energy**2
    return mag_sum,mag2_sum,energy1_sum,energy2_sum
```

## 4.蒙特卡洛方法求解2维Ising模型

将以上这些函数定义好后，我们便可以开始进行 Monte Carlo 模拟来计算体系的物理量。计算过程中我们先进行预热总步数的 1/10 使模拟达到平衡，再每隔一个  $N$  步进行一次取样，这里  $N$  即总的格点数，也就是 Monte Carlo 算法中的一个 Sweep。

### 4.1 体系热力学量公式推导

下面计算  $\langle m^2 \rangle$ ，每一个样本计算的公式为

$$\langle m^2 \rangle = (\sum_i s_i)^2 / N \quad (6)$$

在每一个给定温度下，对所取全部样本进行计算，再取平均值，便可得到该温度下的  $\langle m^2 \rangle$ 。

接下来就用类似的步骤去计算平均格点比热容  $c$  和磁化率  $\chi$ 。

由配分函数  $Z = \sum e^{-\beta H}$ ，有内能  $\langle E \rangle = -\frac{\partial \ln Z}{\partial \beta}$ ，又由比热容的公式为

$C_v = \frac{\partial \langle E \rangle}{\partial T}$ ，可以得到

$$\begin{aligned} C_v &= -\frac{\partial}{\partial T} \frac{\partial \ln Z}{\partial \beta} = -\frac{\partial}{\partial \beta} \frac{\partial \beta}{\partial T} \frac{\partial \ln Z}{\partial \beta} \\ &= \frac{1}{k_B T^2} \frac{\partial}{\partial \beta} \left( \frac{1}{Z} \frac{\partial Z}{\partial \beta} \right) \\ &= \frac{1}{k_B T^2} \left( \frac{1}{Z} \frac{\partial^2 Z}{\partial \beta^2} - \left( \frac{1}{Z} \frac{\partial Z}{\partial \beta} \right)^2 \right) \\ &= \frac{(\langle E \rangle^2 - \langle E^2 \rangle)}{k_B T^2} \end{aligned} \quad (7)$$

平均格点比热容为  $c = \frac{C_v}{N} = \frac{(\langle E \rangle^2 - \langle E^2 \rangle)}{N k_B T^2}$ 。我们取  $k_B = 1$  则有

$$c = \frac{(\langle E \rangle^2 - \langle E^2 \rangle)}{N T^2} \quad (8)$$

接下来推导磁化率的公式。我们有磁化强度

$$\begin{aligned}
M = Nm &= \sum \vec{\mu}_s = \frac{2\mu_B}{\hbar} \sum \vec{S} \\
&= \mu_B \sum \sigma = \mu_B \frac{\text{Tr}(\sum \sigma e^{-\beta \hat{H}})}{\text{Tr}(e^{-\beta \hat{H}})}
\end{aligned}$$

则磁化率为

$$\begin{aligned}
\chi &= \frac{\partial M}{\partial H} = \mu_1 \mu_B \frac{\partial}{\partial B} \frac{\text{Tr}(\sum \sigma e^{-\beta \hat{H}})}{\text{Tr}(e^{-\beta \hat{H}})} \\
&= \mu_1 \mu_B^2 \beta \left( \frac{\text{Tr}(\sum \sigma \sum \sigma e^{-\beta \hat{H}})}{\text{Tr}(e^{-\beta \hat{H}})} - \left( \frac{\text{Tr}(\sum \sigma e^{-\beta \hat{H}})}{\text{Tr}(e^{-\beta \hat{H}})} \right)^2 \right) \\
&= \mu_1 \beta (\langle M^2 \rangle - \langle M \rangle^2) \\
&= \frac{(\langle m^2 \rangle - \langle |m| \rangle^2) N}{T}
\end{aligned} \tag{9}$$

其中有  $\mu_1 = 1$ ,  $k_B = 1$ 。

## 4.2 程序设计

得到表达式后，我们可以设计这部分代码为：

```

global spins
spins = np.full(shape,1)
global spins_sum
    global current_energy
        current_energy=get_energy()
m2    T_list=[]
        C_list=[]
M2_list=[]
M_susceptibility_list=[]
M2_std=[]
C_std=[]
M_suscep_std=[]
N_bin=50
N_per_bin=5000
N_total=N_bin*N_per_bin
for temperature in np.arange(1.5,4,delta):

```

```

beta=1/temperature
T_list.append(temperature)
M2_bin=np.zeros((N_bin))
C_bin=np.zeros((N_bin))
Chi_bin=np.zeros((N_bin))
spins_sum=np.sum(spins)
isweep(int(N_total/10),beta)
for i in range(N_bin):
    magnetization_sum=0
    magnetization2_sum=0
    energy_sum=0
    energy_2_sum=0

magnetization_sum,magnetization2_sum,energy_sum,energy_2_sum=bin_sweep(N_per_bin,beta,magnetization_sum,magnetization2_sum,energy_sum,energy_2_sum)
    magnetization_b=magnetization_sum/N_per_bin
    magnetization2_b=magnetization2_sum/N_per_bin
    energy_b=energy_sum/N_per_bin
    energy2_b=energy_2_sum/N_per_bin
    heat_capacity_b=(energy2_b-energy_b**2)*(beta**2)/N
    magnetic_susceptibility_b=(magnetization2_b-magnetization_b**2)*N*beta
    M2_bin[i]=magnetization2_b
    C_bin[i]=heat_capacity_b
    Chi_bin[i]=magnetic_susceptibility_b

magnetization2=np.mean(M2_bin)
stantard_deviation_M2=np.std(M2_bin)/sqrt(N_bin-1)

heat_capacity=np.mean(C_bin)
stantard_deviation_C=np.std(C_bin)/sqrt(N_bin-1)

magnetic_susceptibility=np.mean(Chi_bin)
stantard_deviation_Msus=np.std(Chi_bin)/sqrt(N_bin-1)

M2_list.append(magnetization2)
M2_std.append(stantard_deviation_M2)
C_list.append(heat_capacity)
C_std.append(stantard_deviation_C)

```



```
M_susceptibility_list.append(magnetic_susceptibility)
M_suscep_std.append(stantard_deviation_Msus)
```

主程序代码思路为，首先对温度  $T$  进行一重循环，对于每个给定的温度，我们再对所有的 bin 进行循环，因此循环  $N_{bin}$  次，就得到了  $N_{bin}$  个 bin。每个 bin 的循环内调用  $N_{per\_bin}$  次之前写好的单个 bin 循环的函数，每调用一次得到一个样本值，存储在相应的列表中，这样每个 bin 内有  $N_{per\_bin}$  个样本。因此我们总的样本数为  $N_{total} = N_{bin} * N_{per\_bin}$ 。再对每个 bin 运用之前推导的公式去计算均值和标准差，得到数据点和误差。

### 4.3 误差分析与结果展示

在计算标准差时，我们需要先清楚蒙卡模拟的过程和误差产生的原因。在进行蒙卡模拟时，每次取样得到的值都是一个随机变量，属于某个分布，而我们所有的取样属于同一个分布。因此，若我们进行了  $N_{total}$  次取样，这  $N_{total}$  个值各自是一个随机变量，同属于一个分布。我们想要用这  $N_{total}$  个样本来近似估计这个分布的标准差，就是通过计算这所有样本的均值和样本标准差，这个式子为

$$\sigma^2 = \frac{1}{N_{total} - 1} \sum (\bar{Q}_b - \bar{Q})^2 \quad (10)$$

而进一步的，由于蒙卡模拟我们是将所有样本取平均后的值来作为我们最终的结果，而这个平均值同样是一个随机变量，同样服从一个分布。那么我们最终蒙卡的误差应当是这个平均值所满足的分布的标准差，这个标准差的估计就应该用到更多的平均值，而运用中心极限定理，我们有分布的标准差就等于单个样本的分布的标准差除以根号  $n$ ，因此有

$$\sigma^2 = \frac{1}{N_{total}(N_{total} - 1)} \sum (\bar{Q}_b - \bar{Q})^2 \quad (11)$$

又因为蒙特卡洛模拟中各样本间有关联，我们运用 bin 方法将多个样本的均值作为一个样本，这样这些样本间就可以看作近似独立的，也就满足了我们之前讨论的原理。我们将所得到的所有样本分成若干个 bins，对每个 bin 内的样本计算均值，得到一个  $\bar{Q}_b$ ，再对这些 bin 的均值求一个均值，即

$$\bar{Q} = \frac{1}{B} \sum_{b=1}^B \bar{Q}_b \quad (12)$$

这里 B 即 bin 的个数。因此此时独立样本的个数即为 B，我们标准差的公式也就变为

$$\sigma_Q^2 = \frac{1}{B(B-1)} \sum_{b=1}^B (\bar{Q}_b - \bar{Q})^2 \quad (13)$$

$\sigma_Q$  就是标准差，即 errorbar，也就是在该温度下 Monte Carlo 模拟所得物理量值的误差。

我们分别对  $L=8$ 、 $L=12$  和  $L=16$  进行了计算，我们将温度以  $T_0 = \frac{J}{k_B} = J$  为单位，所得结果为

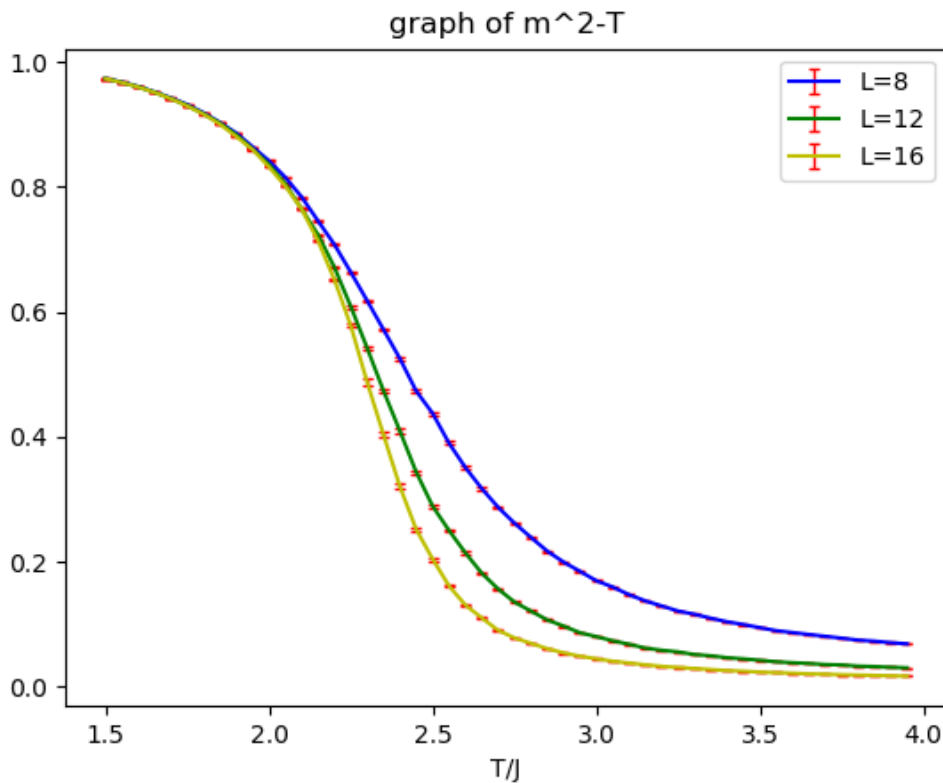


图2.  $\langle m^2 \rangle - T$

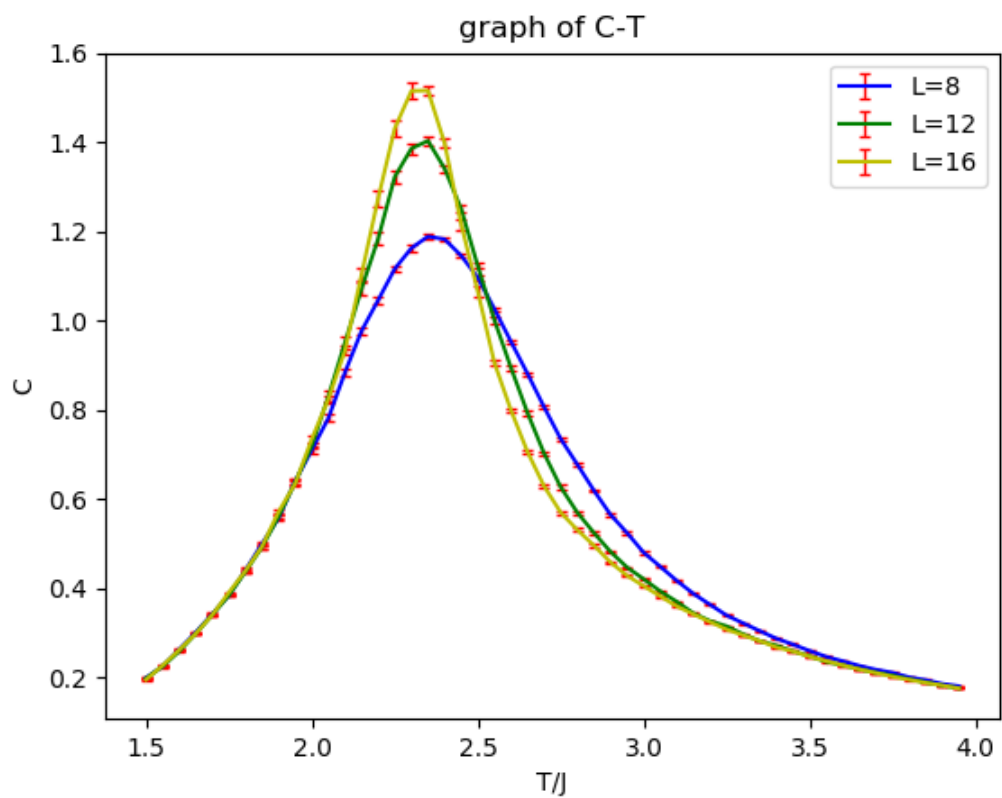


图3.  $C - T$

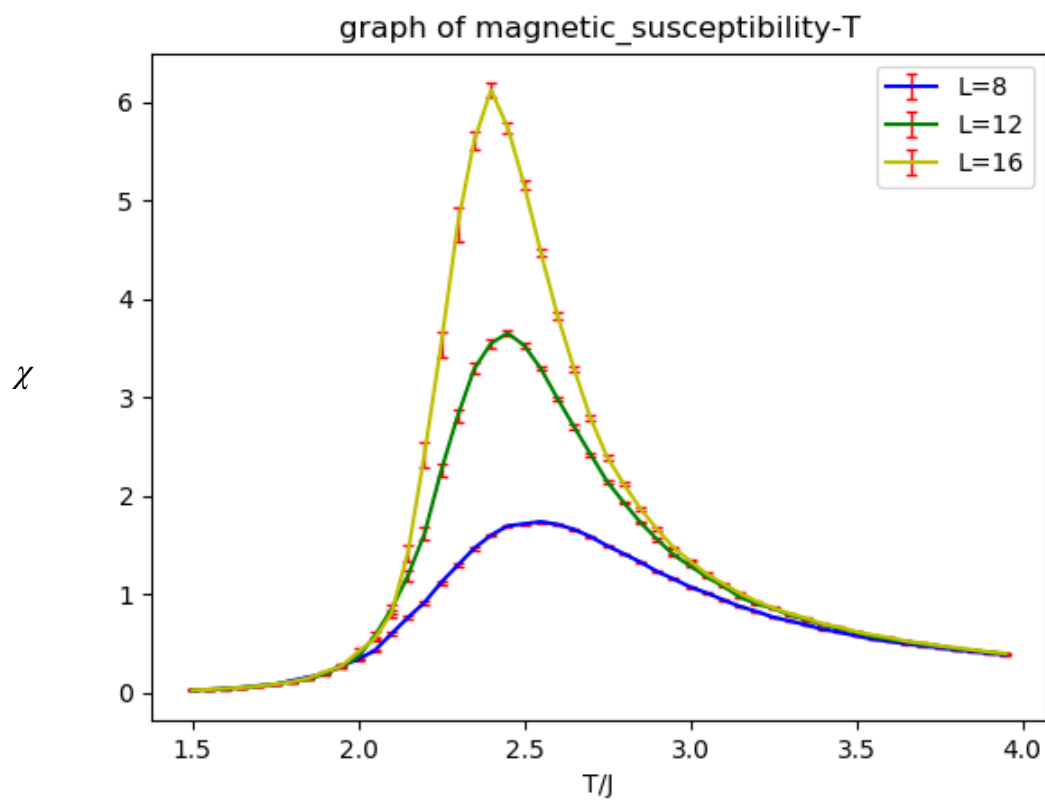


图4.  $\chi - T$

由上述结果可以看出，2D Ising 模型相变温度在 2.4 J 附近，还需要进一步增加格点来得到更好的结果。

最后，为了证明所得结果中误差分析是正确的，我们分析标准差的计算中，根据中心极限定理，当  $N_{\text{per\_bin}}$  较大时，bin 平均值的分布是高斯分布，bin 的均值可以看作是统计上相互独立的。在这个情况下，errorbar 应该只依赖于总样本数，即  $N_{\text{total}}$ ，且关于总样本数是  $1/2$  次幂衰减的：

$$\sigma_Q \propto \frac{1}{\sqrt{N_{\text{perbin}} * N_{\text{bin}}}} = \frac{1}{\sqrt{N_{\text{total}}}} \quad (14)$$

其中比例系数则正比于蒙卡样本的自关联时间的  $1/2$  次幂，即  $\propto \sqrt{\tau_Q}$ 。这里需要先定义自关联函数

$$A_Q(t) = \frac{\langle Q(i+t)Q(i) \rangle - \langle Q \rangle^2}{\langle Q^2 \rangle - \langle Q \rangle^2} \quad (15)$$

其中  $t$  和  $i$  为进行蒙卡模拟的时间，通常以蒙卡 sweep 数为单位。由归一化要求  $A_Q(0) = 1$  和  $A_Q(t \rightarrow \infty) = 0$ ，可定义渐进行为  $A_Q(t) \sim e^{-t/\tau_Q}$ ，此即自关联时间  $\tau_Q$  的定义。通常我们使用积分自关联时间来进行分析，即

$$\tau_Q^{\text{int}} = \frac{1}{2} + \sum_{t=1}^{\infty} A_Q(t) \quad (16)$$

由于有  $A_Q(t \rightarrow \infty) = 0$ ，因此当蒙卡模拟足够多的步数后，自关联时间将不再增大，而是逐渐稳定在一个常数。

因此，若我们固定 bin 的个数  $N_{\text{bin}}$ ，而去逐渐增大每个 bin 内的样本数  $N_{\text{per\_bin}}$ ，那么模拟的情况就由样本间关联比较严重逐渐变为关联消失，而当关联较严重时，误差会被低估，随着  $N_{\text{per\_bin}}$  的增大关联减弱，误差渐渐准确，最终趋于一个常数。

而如果我们固定单个 bin 内的样本数，例如取  $N_{\text{per\_bin}} = 4096$ ，那么样本间的关联就一直是可忽略的，随着 bin 的数目  $N_{\text{bin}}$  的增大，也就是样本数的增大，误差值则应该先波动最后稳定在一个常数值。

因此我们采用的验证方法是在给定温度和体系格点数下，将  $\text{errorbar}$  与总样本数的 $1/2$ 次幂  $\sqrt{N_{total}}$  作为纵坐标，去观察其随着总样本数  $N_{total}$  变化的趋势图像。我们先取

$T=4\text{ J}$  ,  $L=4$  ,  $N_{bin} = 16384$  ,  $N_{perbin} = [1,2,4,8,32,64,128,256]$  , 计算  $m^2$  的标准差。

之后再尝试同样的  $T$  和  $L$  下取

$N_{perbin} = 4096$  ,  $N_{bin} = [2,4,8,32,64,256,512,666,888,1024]$  , 计算  $m^2$  的标准差。

得到的结果为

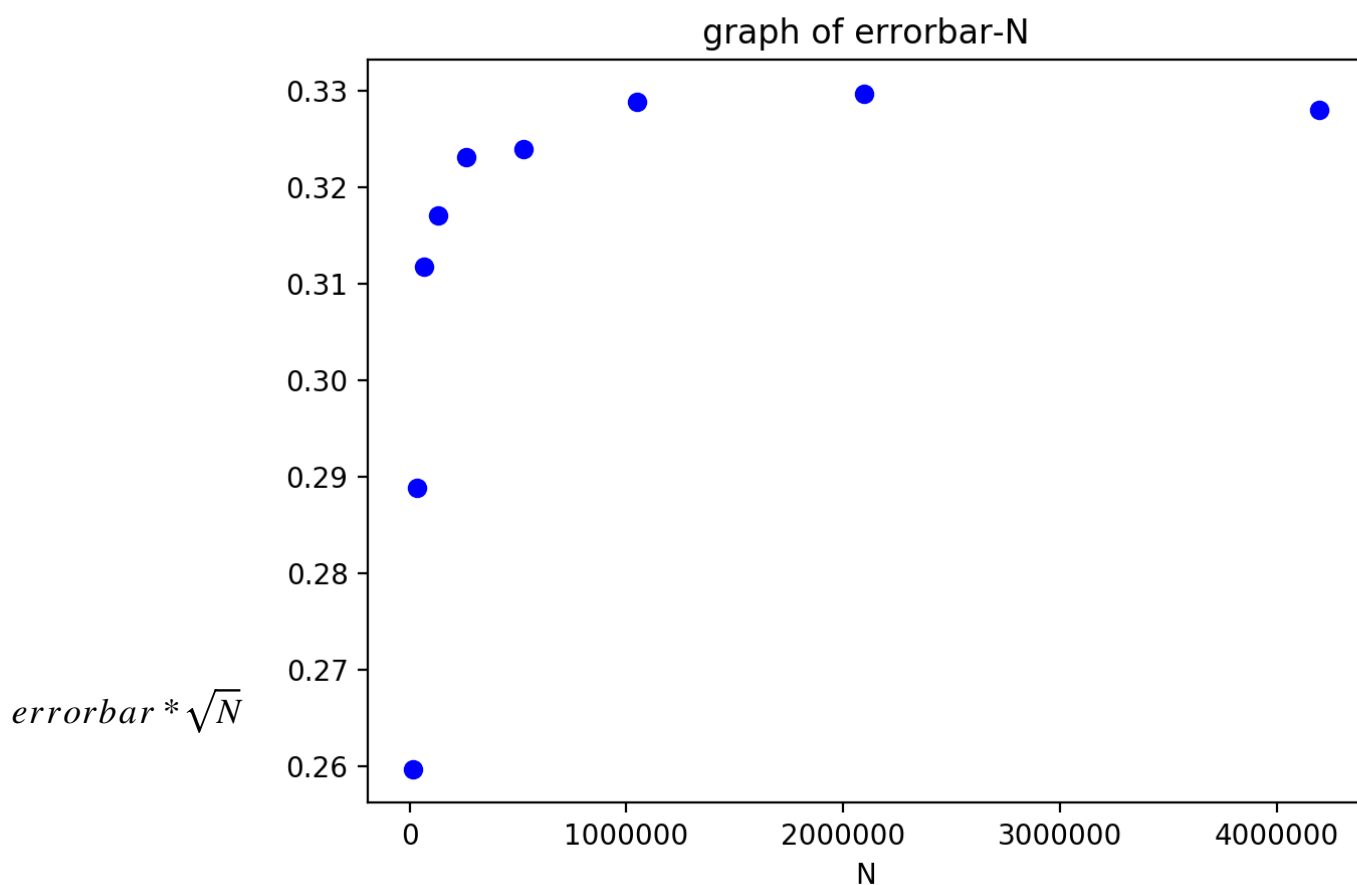


图 5 .固定bin的数目，逐渐增大每个bin内的样本所得的误差关于总样本数的趋势图

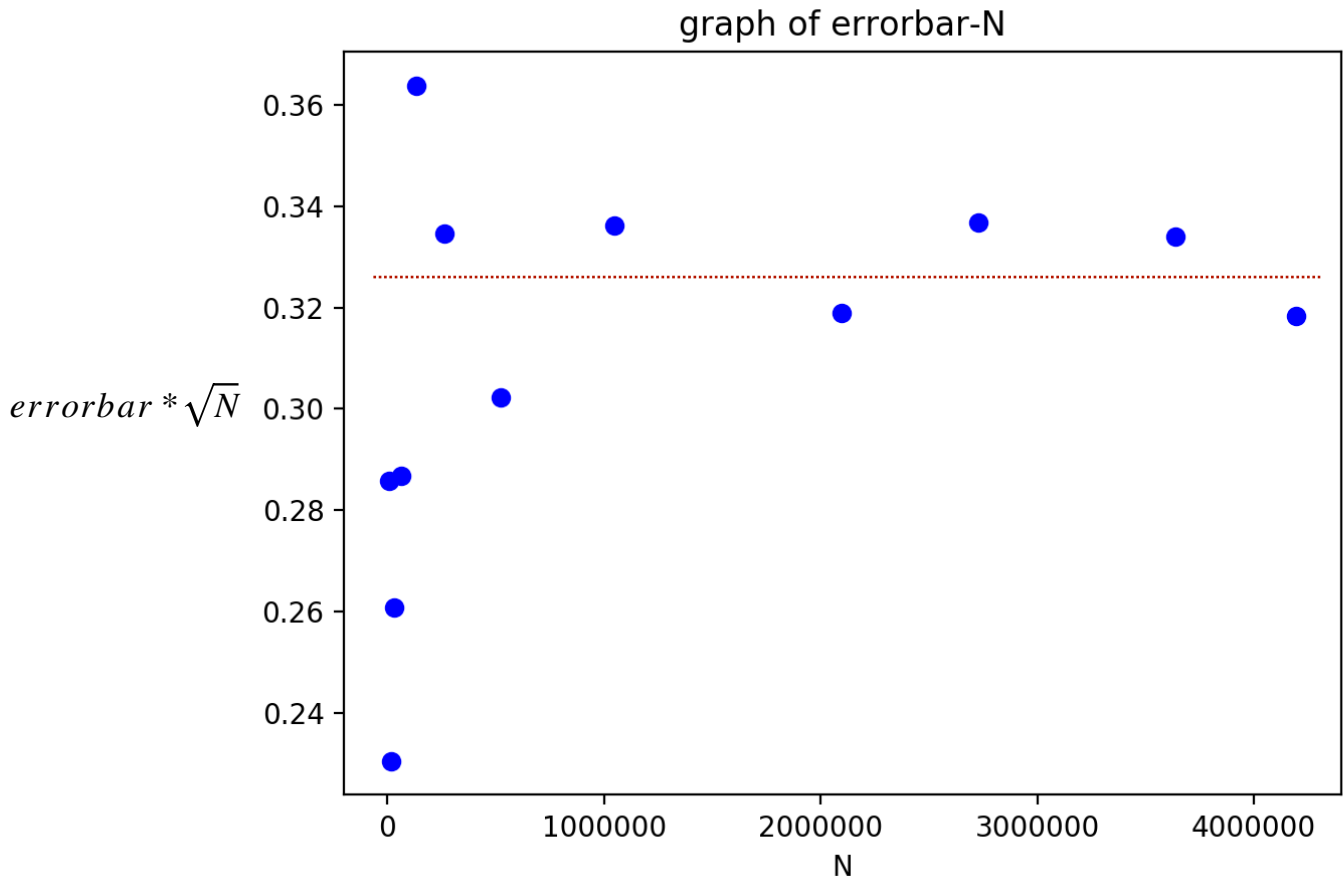


图 6 .固定每个bin内样本数，逐渐增大bin的数目所得的误差关于总样本数的趋势图

由该图可以看出，在图5所示情况中，我们得到的  $errorbar * \sqrt{N}$  在  $N$  较小时逐渐增大，而随着  $N$  增大后基本稳定在常数值，而理论分析中  $errorbar * \sqrt{N}$  应该是  $\propto \sqrt{\tau_Q}$ ，随着  $N_{total}$  的增大， $\tau_Q$  先增大后逐渐饱和，稳定在一个常数。图6所示情况则是先波动再逐渐稳定于一个常数，可见两图情况皆与之前的理论分析吻合。因此可以证明我们所得到的  $errorbar$  结果是正确的。

## 4.4 临界指数确定与data collapse

### 4.1.1 理论推导

为了进一步讨论有限尺寸体系在趋近临界点处的行为，我们引入关联长度，这是相变和临界现象中很重要的概念。我们知道在有限体系中，趋近临界温度时关联长度会以幂次发散，即

$$\xi \sim |t|^{-\nu} \quad (17)$$

我们知道序参量在临界点处的行为为

$$\langle m \rangle \sim |t|^\beta \quad (18)$$

以及磁化率公式为

$$\chi = \frac{N}{T}(\langle m^2 \rangle - \langle |m| \rangle^2) \quad (19)$$

由该表达式可以看出磁化率随着序参量的涨落，在趋于临界温度时会趋于发散，可定义

$$\chi \sim |t|^{-\gamma} \quad (20)$$

因此，在趋近临界点时，系统变的对微小的外场的扰动非常敏感。

再考虑比热容，有

$$C \sim |t|^{-\alpha} \quad (21)$$

对于二维Ising模型有  $\alpha = 0$ ，但在  $T_c$  处仍有微弱的对数发散。

对于这些临界指数，我们需要进行标度变换来进一步确定。考虑任意在  $T_c$  有幂次发散行为的物理量  $Q$ （取约化温度  $t = \frac{T - T_c}{T_c}$ ）

$$Q \sim |t|^{-\kappa} \quad (22)$$

再由 (17) 式代入有

$$Q \sim \xi^{\kappa/\nu} \quad (23)$$

该式在  $\xi \ll L$  成立。因此有在有限格点上  $Q$  最大值为

$$Q_{max}(L) \sim L^{\kappa/\nu} \quad (24)$$

同理有 (17) 式中最大的约化温度满足

$$|t_{max}(L)| \sim L^{-1/\nu} \quad (25)$$

前面的比例系数由物理量决定。

由标度变换理论，任意在临界温度有奇异性的观测量都满足  $L$  的幂次乘上一个关于  $\xi/L$  的非奇异函数，即

$$Q(t, L) = L^\sigma f(\xi/L)$$

由 (17) 式有

$$Q(t, L) = L^\sigma g(tL^{1/\nu})$$

当温度处于临界温度即  $t=0$  时，有  $Q(0, L) \sim L^\sigma$ ，当体系尺寸无限大有

$$Q(t, L \rightarrow \infty) \sim |t|^{-\kappa}$$

其中  $\kappa$  为负（例如对于序参量有  $\kappa = -\beta$ ）。因此标度变换函数  $g(x)$  满足当  $x \rightarrow \infty$  时有  $g(x) \sim x^{-\kappa}$ ，因此取  $\sigma = \kappa/\nu$ ，即

$$Q(t, L) = L^{\kappa/\nu} g(tL^{1/\nu}) \quad (26)$$

为了求出标度变换函数，我们可以定义

$$y_L = Q(t, L)L^{-\kappa/\nu}, \quad x_L = tL^{1/\nu} \quad (27)$$

#### 4.4.2 代码实现

由上面推导所得到的 (27) 式，运用我们之前通过蒙特卡洛得到的数据，代入式子并对不同尺寸的体系进行多项式拟合，我们这里取得是 5 次多项式，设拟合的结果为  $y_1$ ，数据点为  $y$ ，我们的误差函数取均方误差

$$Error = \frac{\sum (y - y_1)^2}{n^2}$$

其中  $n$  为数据点的个数。通过使误差最小，我们可以找到最适合的指数，对于磁化率为  $\gamma$ ，对于比热容为  $\alpha$ 。

具体代码实现时，我们首先定义了产生我们所需的横纵坐标的函数，即将原本通过蒙特卡洛得到的数据分别作相应的标度变换，得到新的  $x$ 、 $y$ ，代码如下：

```
def generate_xLyL(gamma, mu):
    for t in T_list:
        index=T_list.index(t)
        reduce_T=(t-Tc)/Tc
```



```

x1=reduce_T*L1**( $1/\mu$ )
x2=reduce_T*L2**( $1/\mu$ )
x3=reduce_T*L3**( $1/\mu$ )
if abs(x1)<=0.5:
    tL_list.append(x1)
    c1=C_list_8[index]
    C_list_new.append(c1*L1**(-gamma/ $\mu$ ))
if abs(x2)<=0.5:
    tL_list.append(x2)
    c2=C_list_12[index]
    C_list_new.append(c2*L2**(-gamma/ $\mu$ ))
if abs(x3)<=0.5:
    tL_list.append(x3)
    c3=C_list_16[index]
    C_list_new.append(c3*L3**(-gamma/ $\mu$ ))
# print(tL_list)
return tL_list,C_list_new

```

接下来对待定的临界指数  $\gamma$  或  $\alpha$  和  $\nu$  进行循环，遍历某个范围，对已有的数据点进行 5 次多项式拟合，再用得到的拟合值来计算相应的误差，寻找误差最小的值所对应的临界指数，就得到我们所要求的。代码如下

```

error_list=[]
exponents=[]
for gamma in np.arange(0,0.5,0.01):
    for mu in np.arange(0.9,1.3,0.01):
        tL_list,C_list_new=generate_xLyL(gamma,mu)
        tL_array=np.array(tL_list)
        C_array=np.array(C_list_new)
        poly=np.polyfit(tL_array,C_array,5)
        length=len(C_list_new)
        error=0
        for x in tL_list:
            indexnew=tL_list.index(x)
            y1=np.polyval(poly,x)
            y=C_list_new[indexnew]
            error += (y-y1)**2/length**2
        exponents.append((gamma,mu))

```

```

error_list.append(error)
index_min=error_list.index(min(error_list))
gamma_mu=exponents[index_min]

```

### 4.3.3 临界指数及data collapse结果展示

用上述方法，我们得到 2 维 Ising 模型的临界指数为  $\alpha = 0$ ， $\nu = 0.99$ 。磁化率所对应的指数  $\gamma$  还未得到。这里我们取严格解  $\gamma = 7/4$ ，临界温度  $T_c = 2.269 \text{ J}$ ，所得到的磁化率图像为

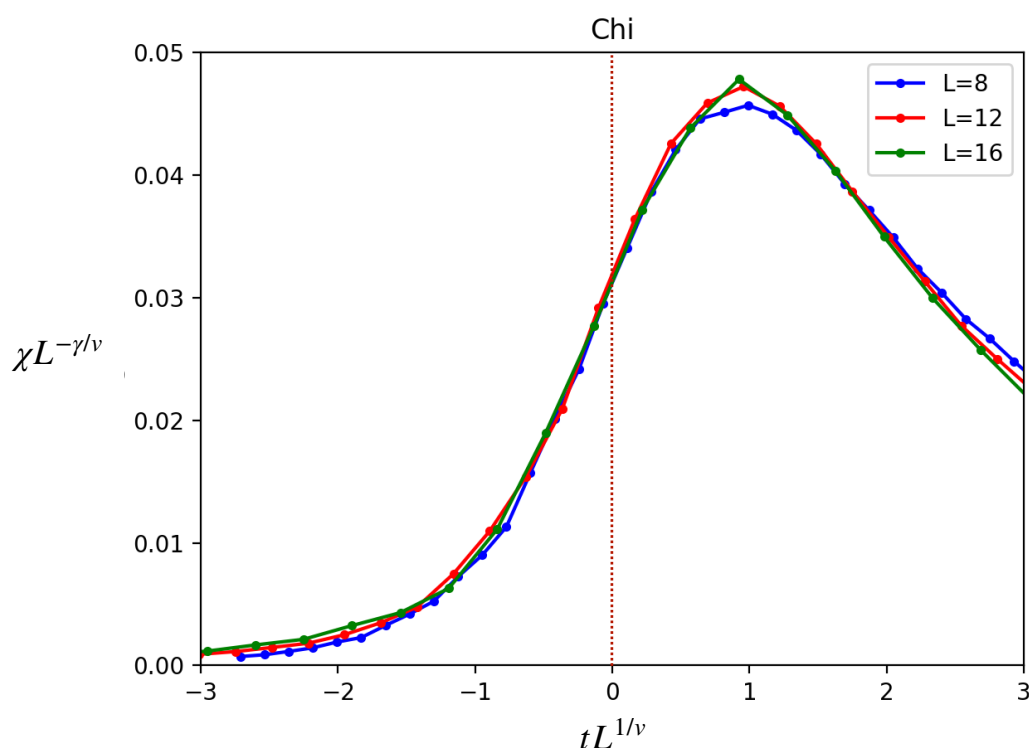


图 7. 标度变换后磁化率图像

观察该图可以发现，在约化温度  $t$  为 0 时，不同尺寸的磁化率图像重叠在一起，这被称为 data collapse。这里便可以确定临界温度  $T_c = 2.269 \text{ J}$ 。进一步的，选取相变温度附近的部分，可以更好的看出标度变换的 collapse，如下图

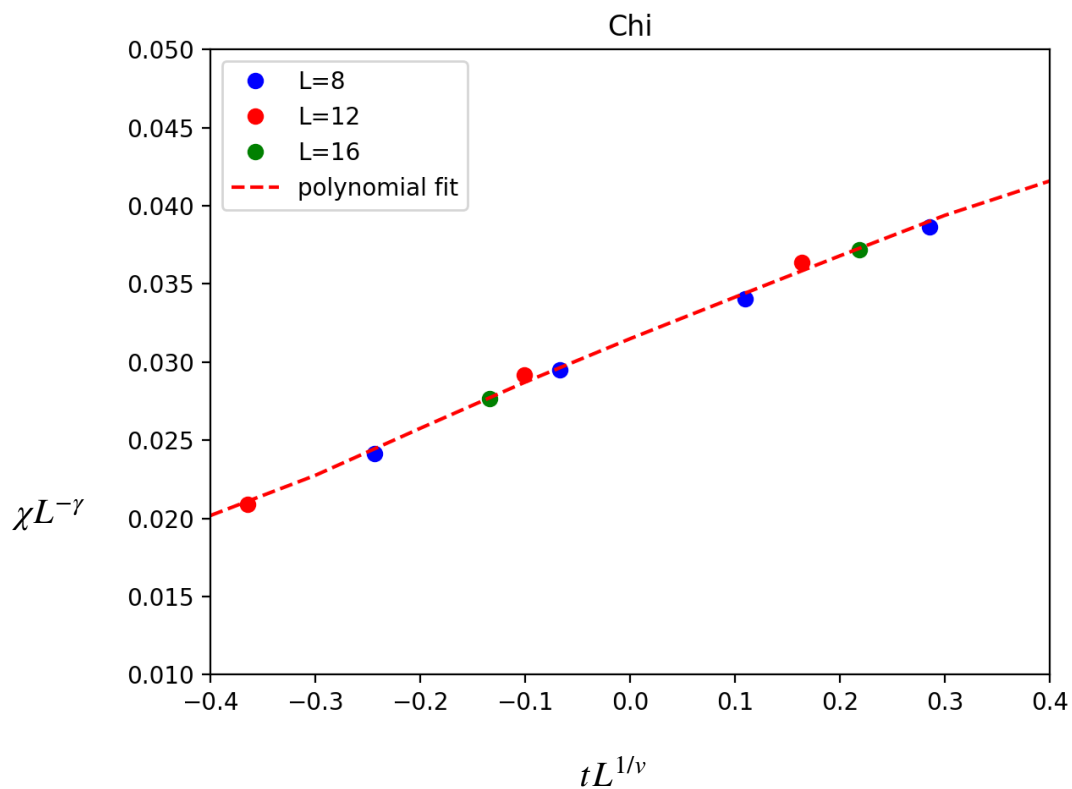
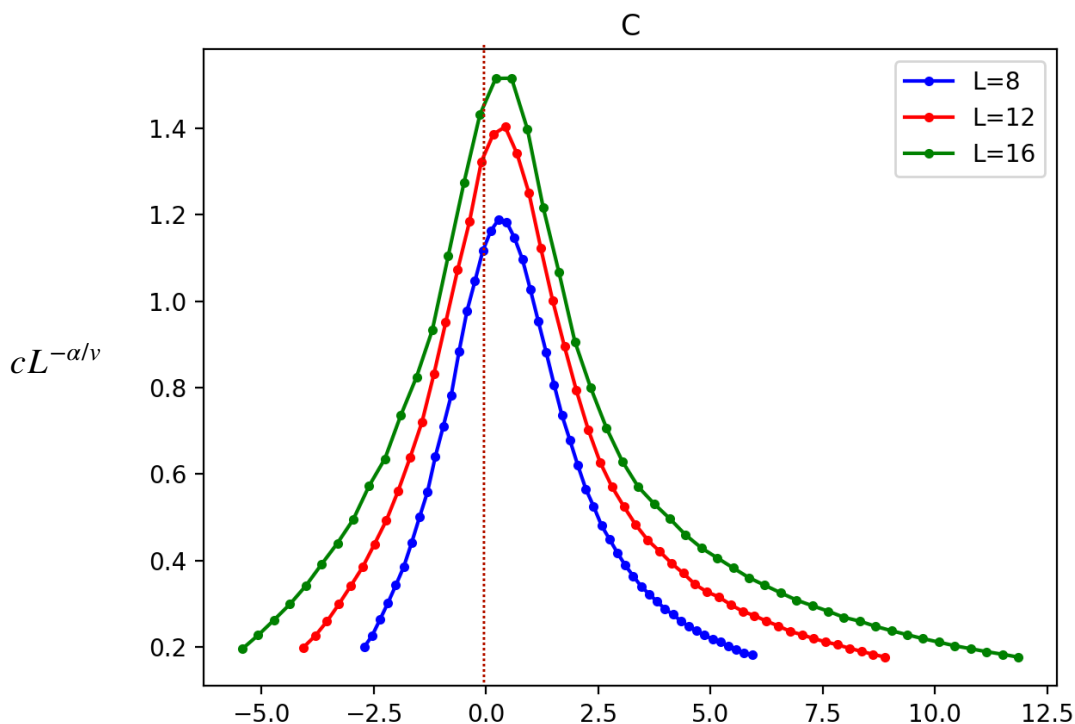


图 8. 标度变换后磁化率图像

其中虚线则为 5 次多项式拟合的结果。进一步的，我们可以通过对数据点增加高斯扰动来重复进行多次拟合，来得到相应的误差 errorbar。事实上这里应该采用更大的体系来进行拟合观察，才能得到更稳定的结果，否则标度修正将不能被忽略，我们这里暂时不考虑这一点。

接下来再看比热在标度变换后的图像：



$$tL^{1/\nu}$$

图 9 . 标度变换后比热图像

在比热的图中，可以看出我们需要计算更大的体系来更好的观察相变点的行为，以我们目前的图像，并不能很好的看出在约化温度  $t=0$  处比热随着体系的增大而最大值逐渐向  $t=0$  处移动，且在相变温度有发散的趋势。我们将尝试计算更大的体系。

### 【参考文献】

- [1]Anders W. Sandvik, Computational Studies of Quantum Spin Systems, arXiv:1101.3281.
- [2]Richard Fitzpatrick.Computational Physics:2006-03-29.