蒙特卡洛方法求解 Ising 模型 (二)

何翼成 *

May 23, 2022

Project 1

一 题目分析

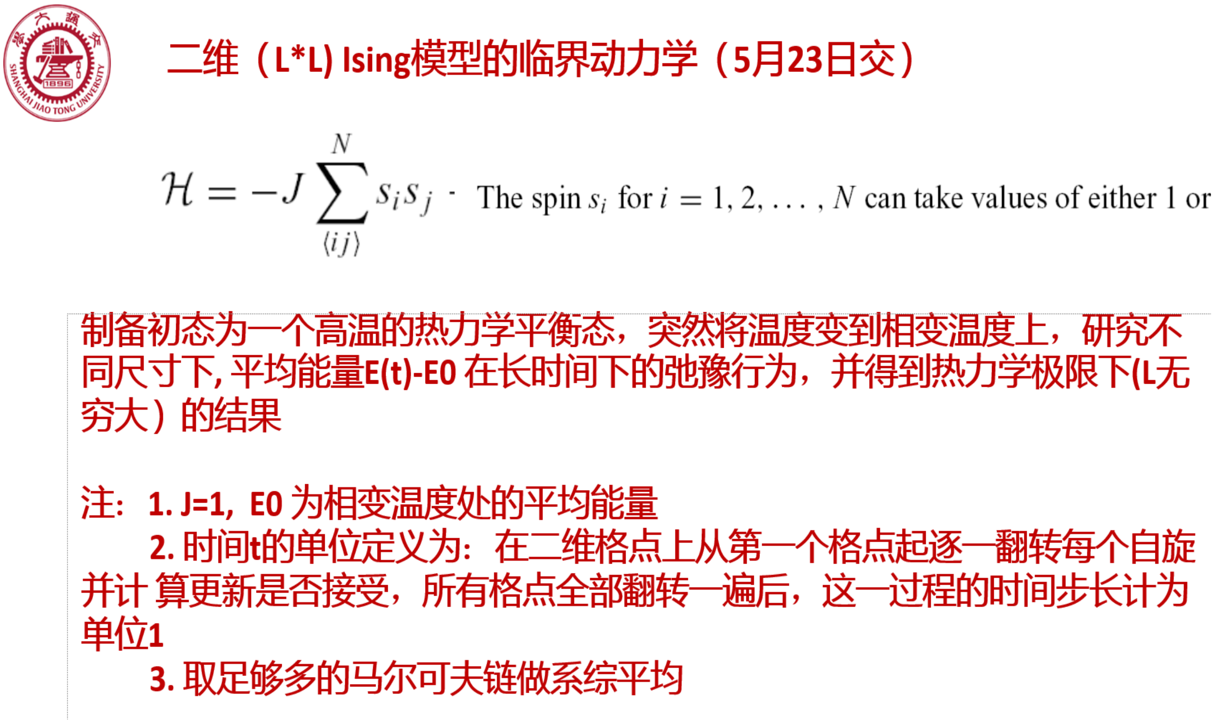

Figure 1: 题目总览

二 代码展示

```
%创建方格矩阵
Ls=[50];Tspan=linspace(100,100,2000);T0=2.269;num=800;Num=3e4;%计算步数
A=zeros(length(Ls),num);%储存不同L下的能量-时间曲线
for in=1:length(Ls)
```

*学号:520072910043;
邮箱地址: heyicheng@sjtu. edu. cn

```matlab
    L=Ls(in);
    c=cputime;
    %设定初态的温度
    for tn=1:length(Tspan)
        T=Tspan(tn);
        Es=zeros(length(Tspan),num);
        %得到高温的热力学平衡态
        %确认尺寸，生成随机矩阵
        M=randi([0,1],L,L)*2-1;
        %逐行扫描元素，完全扫描整个矩阵为一个时间单位
        %演化num步数
        for tt=1:num
            for jj=1:L   %y
                for ii=1:L %x
                    M_1=M;%保留磁矩矩阵的原信息
                    M_2=M;M_2(ii,jj)=-M_1(ii,jj);%翻转后的磁矩矩阵
                    pudM=pud(M);
                    deltaE=2*M(ii,jj)*(pudM(ii+2,jj+1)+pudM(ii,jj+1)+pudM(ii+1,jj+2)+pudM(ii+1,jj));
                    %判断状态是否保留
                    flag=Metro(deltaE,T0);
                    if flag==1
                        M=M_2;
                    else
                        M=M_1;
                    end
                end
            end
        Es(tn,tt)=H(M);%计算一次能量并且储存
        cc=cputime;
        disp("计算长度为"+L+",温度进度为"+tn/length(Tspan)*100+"%, 已完成进度"+tt/num*100+"%,已耗时"+num2str((c
        end
    end
    Eavg=sum(Es)/length(Tspan);
    A(in,:)=Eavg;
end

%计算T0时的平均能量
    E0=AverageH(L,T,Num,10);
    plot(1:num,A-E0)
    legend("L="+L)
    xlabel("LogTime")
    ylabel("Log(E-E_{0})")

%%
%Metropolis算法函数的定义
function flag=Metro(deltaE,T)
    beta=1/T;
    if deltaE<=0
        p=1;
```

```
        else
            p=exp(-beta*deltaE);
        end
        z=rand();
        if z<p
           flag=1;
        else
            flag=0;
        end
    end
    %%
    %Pud，辅助计算。扩展为(L+2)**2的矩阵
    function pudM=pud(M)
    L=size(M,1);
    pudM=zeros(L+2,L+2);%分配储存空间
    %pudding，采用周期性边界条件
    pudM(2:(L+1),2:(L+1))=M;
    %行的移动
    pudM(1,2:(L+1))=M(L,:);
    pudM(L+2,2:(L+1))=M(1,:);
    %列的移动
    pudM(2:(L+1),1)=M(:,L);
    pudM(2:(L+1),L+2)=M(:,1);
    end

    %能量计算
    function E=H(M)
    L=size(M,1);
    pudM=pud(M);
    H=0;
    for i=2:L
        for j=2:L
            H=H-pudM(i,j)*(pudM(i,j-1)+pudM(i,j+1)+pudM(i-1,j)+pudM(i+1,j));
        end
    end
    E=H/2;
    end

    %%
    %平均能量计算.输入尺寸，温度,循环次数,并行链条数，计算该温度下的矩阵的平均能量
    function avgH=AverageH(L,T,Num,s)
    %创建随机矩阵
    Matrix=randi([0,1],L,L)*2-1;
    %创建空数组记录能量值
    Hels=zeros(s,Num);
    for ss=1:s
    for mm=1:Num
        locs=randi([1,L],1,2);%创建随机坐标
        x=locs(1);y=locs(2);
```

```
        Matrix_1=Matrix;%保留磁矩矩阵的原信息
        Matrix_2=Matrix;Matrix_2(x,y)=-Matrix_1(x,y);%翻转后的磁矩矩阵
        pudMatrix=pud(Matrix);
        deltaE=2*Matrix(x,y)*(pudMatrix(x+2,y+1)+pudMatrix(x,y+1)+pudMatrix(x+1,y+2)+pudMatrix(x+1,y));
        %判断状态是否保留
        flag=Metro(deltaE,T);
        if flag==1
            Matrix=Matrix_2;
        else
            Matrix=Matrix_1;
        end
        Hels(ss,Num)=H(Matrix);
    end
    end
    avgH=sum(Hels,"all")/(Num*s);
    end
```

三　结果分析与结论

3.1　原图像

3.2　取对数轴进行逼近

由于笔记本 CPU 性能问题, 在计算过程中即使采取了多种近似和精简方法, 在 L 增大至 50 左右时, 所需要的时间就开始急速增长. 所以本文中分析过程也会局限于 L<50 的范围.

根据图像分析可知, 随着 L 取值的增大, 其图线的意义逐渐趋向于热力学极限. 从原始图像来看, 就是原本高温的平衡态逐渐平衡在了 $T_0 = 2.269K$. 为了方便分析, 我们对代表时间意义的 Time 轴和平均能量均取对数, 从而得到了一个弯曲的曲线. 我们可以合理推论: 当 L 趋向于无穷大时, 其图线将会趋于一条直线.

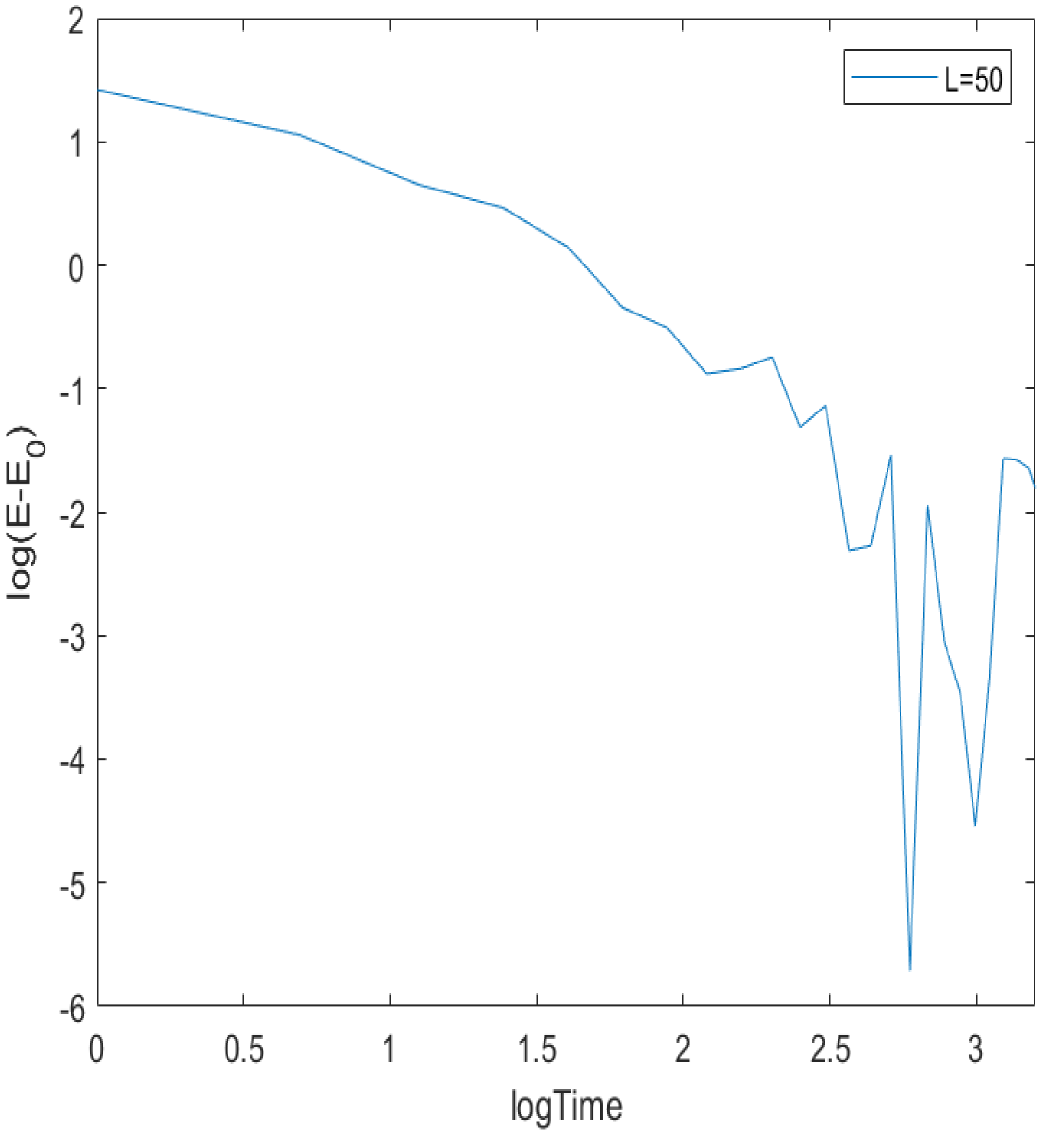

Figure 2: <H>-Time with L=50

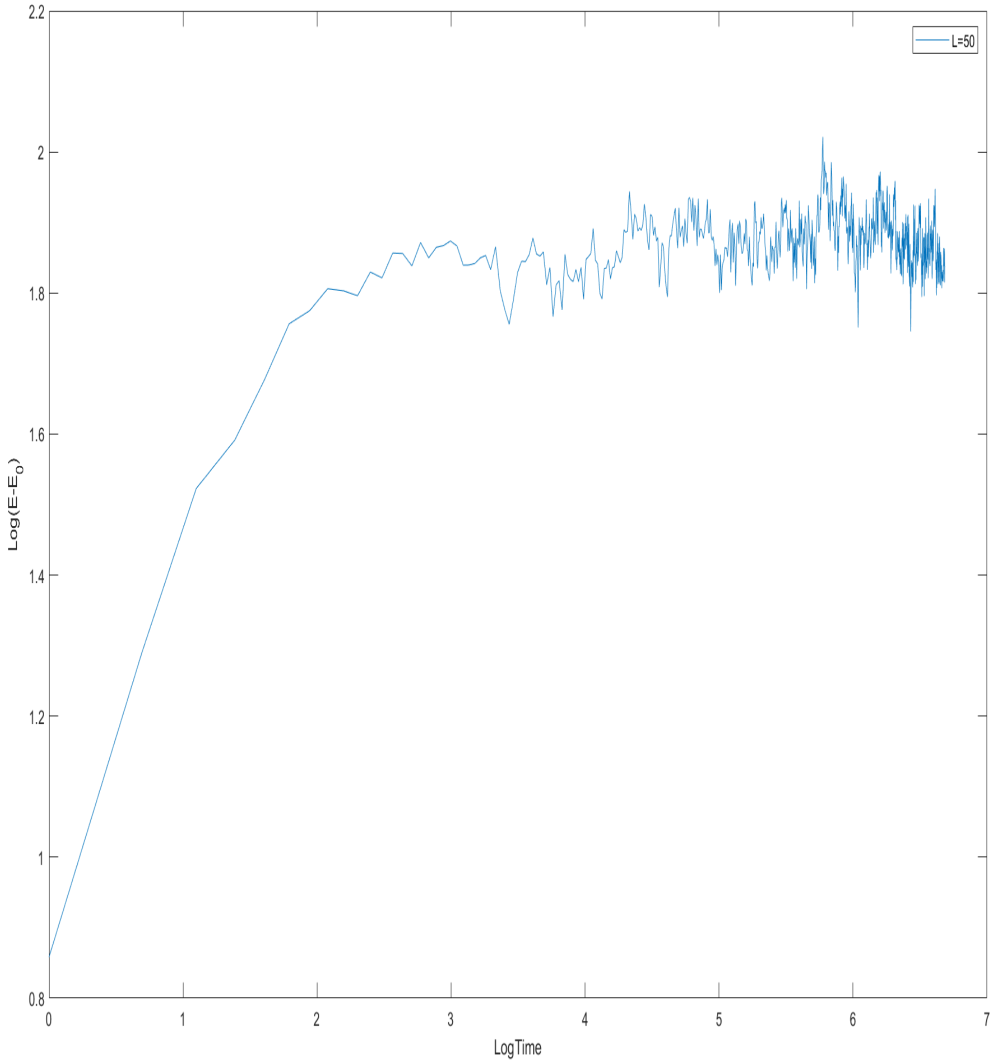

Figure 3: log<H>-logTime