PHYS 516: Methods of Computational Physics
ASSIGNMENT 3- MC Simulation of the Ising Model

Anup V Kanale
February 7, 2017

1 Theoretical Foundation of Metropolis Algorithm

Consider a set of N states, $\Gamma_1, \Gamma_2, ..., \Gamma_N$ and let the probability to find the system in the m-th state, Γ_m, be ρ_m. Here, we prove that the probability distribution is a fixed point of the metropolis transition matrix defined below, i.e., $\Pi\rho = \rho$.

$$\text{(Metropolis transition matrix)}\pi_{mn} = \begin{cases} \alpha_{mn} & \rho_m \geq \rho_n m \neq n \\ \frac{\rho_m}{\rho_n}\alpha_{mn} & \rho_m \leq \rho_n m \neq n \\ 1 - \sum_{m' \neq n} \pi_{m'n} & \end{cases}$$

Here, π_{mn} are elements of the matrix Π, ρ_m are the elements of vector ρ, and α_{mn} are elements of a symmetric attempt matrix, i.e., $\alpha_{mn} = \alpha_{nm}$.

This can be proved by enforcing the Detailed Balance condition. Consider a pair of states m and n. Assuming $\rho_m < \rho_n$,

$$\pi_{nm}\rho_m = \alpha_{nm}\rho_m = \alpha_{mn}\rho_m \tag{1}$$

where the second equality holds because of the symmetric attempt. For the same case,

$$\pi_{mn}\rho_n = \frac{\rho_m}{\rho_n}\alpha_{mn}\rho_n = \alpha_{mn}\rho_m \tag{2}$$

Using equations 1 and 2,

$$\pi_{nm}\rho_m = \pi_{mn}\rho_n \tag{3}$$

The left-hand side is a flux of probability that the current state is n and that the next state is m, and the right-hand side is a flux from m to n.

In order to show that above detailed balance condition is sufficient for the unit-eigenvalue relation, sum the both side over n.

$$\sum_{n=1}^{N} \pi_{mn}\rho_n = \underbrace{\sum_{n=1}^{N} \pi_{nm}}_{=1} \rho_m \tag{4}$$

$$\implies \sum_{n=1}^{N} \pi_{mn}\rho_n = \rho_m \tag{5}$$

$$\therefore \Pi\rho = \rho \tag{6}$$

2 2D Ising Model

Ising model is used for modelling ferromagnetic materials. This model represents a lattice consisting of atoms which have quantum mechanical spin, which can be ± 1. When these individual magnetic fields are aligned in the same direction, it gives rise to macroscopic magnetic field. This strong alignment arises from exchange interactions between electrons.

2.1 Computer Simulation

A program was written in C to simulate the 2D Ising model.

```c
/* Monte Carlo Simulation of 2D Ising Model */
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>

// Define global variables
#define L 20 //lattice size
int s[L][L]; //Spins s[i][j]=+-1
double exp_dV[2][5];
double JdivT; // J/kBT
double HdivT; // H/kBT

void table_set(){  //function to set up the table
  int sDash,sneighbor,k,l;
  for (k=0;k<2;k++){
    sDash = 2*k-1;
    for(l=0;l<5;l++){
      sneighbor = 2*l-4;
      exp_dV[k][l] = exp(2*sDash*(JdivT*sneighbor + HdivT)); // check formula
    }
  }
}

int main() {
  double runM;
  double sumM=0.0, sumM2=0.0;
  double exp_val, avgM, sigM;
  int snew,sneighbor,Sta_step;
  int i,j,step,k,l,im,ip,jm,jp;
  FILE *f = fopen("Magnetization_data.txt", "w");

  printf("Input J/kBT\n");
  scanf("%le",&JdivT);

  HdivT = 0.0;
  Sta_step = 2000000;
```

```c

  table_set();  // Set up the look-up table for the exponent calculation

  for(i=0;i<L;i++) {  //Cold start- start with all spins up configuration
    for(j=0;j<L;j++) {
      s[i][j] = 1;
    }
  }
  runM=1.0*L*L;

  for(step=0; step<Sta_step; step++) {
    i=rand()%L;    j=rand()%L;
    snew=-s[i][j];

    // Figure out which element of the table is to be looked up
    im = (i + L - 1) % L;
    ip = (i + 1) % L;
    jm = (j + L - 1) % L;
    jp = (j + 1) % L;
    k = (snew+1)/2;
    sneighbor = s[im][j] + s[ip][j] + s[i][jm] + s[i][jp];
    l = (sneighbor+4)/2;

    //Change in Pot Energy wth flip
    exp_val=exp_dV[k][l];
    // Accept or reject flip conditionally
    if (exp_val>1.0) {
      s[i][j] = snew;
      runM += 2*snew; //update value of magnetization
    }
    else if(rand()/(double)RAND_MAX < exp_val) {
      s[i][j] = snew;
      runM += 2*snew; //update value of magnetization
    }
    sumM += runM;
    sumM2 += runM*runM;
    fprintf(f, "%f\n", runM);
  }
  fclose(f);
  avgM = sumM/Sta_step;    //Mean Magnetization
  sigM = sqrt(sumM2/Sta_step-avgM*avgM);  //Standard deviation
  printf("Mean Magnetization %le, Standard Deviation %le \n", fabs(avgM), sigM);
}
```

IsingModel.c

2.2 Results

The plots below show the Mean magnetization and its standard deviation as a function of the exchange coupling, and a histogram showing the number of MC samples for every value of Magnetization. The matlab script used to generate these plots is shown below.

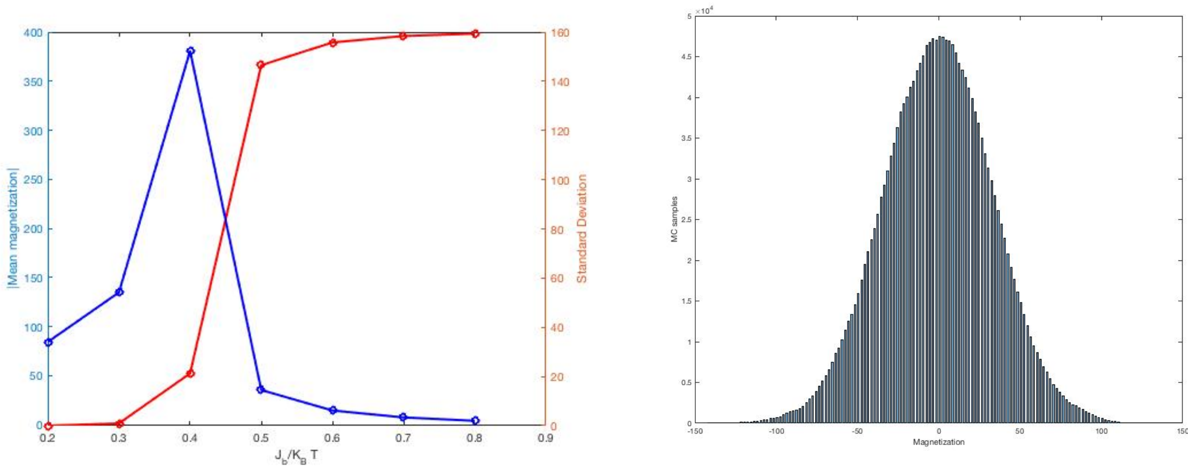

Figure 1: Plot of Magnetization and S.D vs J/K_BT, and Histogram for $J/K_BT = 0.2$

```matlab
close all; clear all;

%% Scatter Plots
JDivT = [0.2:0.1:0.8];
M = [-9.427350e-01,1.361261e+00 ,5.233627e+01,...
    3.662830e+02,3.894057e+02,3.959868e+02,3.984030e+02];
yyaxis left;
plot(JDivT, M, 'r-o','LineWidth', 2);
xlabel('J_b/K_B T'); ylabel('|Mean magnetization|');

sd = [3.403888e+01, 5.418066e+01,1.523472e+02,...
    1.449735e+01,6.180052e+00,3.360630e+00,1.993231e+00];
yyaxis right;
plot(JDivT, sd, 'b-o','LineWidth', 2);
xlabel('J_b/K_B T'); ylabel('Standard Deviation');

%% Histogram
figure()
A = importdata('Magnetization_data.txt', '\n', 0);
histogram(A);
ylabel('MC samples'); xlabel('Magnetization');
xlim([-150 150]);
```

Ising_MagnetizationPlots.m